

KISDarchiv

REALISATION – OPERATION – EVOLUTION

ARK_RECORD_EXAM_TYPE	Bachelor
ARK_RECORD_SUBEXAM	Nebenthema 2
ARK_RECORD_LLECTURER	Prof. Björn Bartholdy
ARK_LLECTURER_AREA	Audiovisuelle Medien
ARK_RECORD_AUTHOR	Willem Rabe
ARK_AUTHOR_EMAIL	willem@rabe.li
ARK_AUTHOR_MNR	11069940
ARK_AUTHOR_DEGREE	BIDes
ARK_RECORD_YEAR	18. März 2014

HINWEISE Aus Gründen der besseren Lesbarkeit wird auf die zusätzliche Formulierung der weiblichen Form verzichtet. Die ausschließliche Verwendung der männlichen Form soll explizit als geschlechtsunabhängig verstanden werden.

Der Lesbarkeit halber wird die Plattform »KISDspaces« samt Artikel geführt.

Die Begriffe »KISDarchiv« und »Neues digitales Archiv / NeDA« werden synonym verwendet, wobei letzterer die temporäre Bezeichnung des Projekts während seiner Entwicklung darstellt.

Alle Screenshots des Systems wurden zwischen Februar 2013 und März 2014 via Chrome (Version 31.0.1650) unter OS X 10.9 angefertigt. Ausnahme bilden »historische« Screenshots und Abbildungen, die im Rahmen der Entwicklung angefertigt wurden.

Die zum Zeitpunkt der Entstehung dieser Arbeit betriebene WordPress-Version trägt die Nummer 3.6.1. Mit dem Update auf Version 3.8.x werden Backend-Ansichten visuell von in dieser Arbeit abgebildeten Ansichten abweichen, die Anordnung und Funktion der Elemente ist davon nicht betroffen.

Alle Urheberrechte an abgebildeten Arbeiten liegen bei deren Autoren.
Änderungen am System vorbehalten.

Gesetzt ist die Arbeit in Tobias Frere-Jones' **Gotham** und **Gotham Narrow**.
Druck und Bindung: Hundt Druck GmbH, Köln.

Präambel & Resümee

»Das Chaos will als solches erkannt und erfahren werden, bevor es sich in eine neue Ordnung umwandeln läßt.«*

Hermann Hesse

Michaela Kunze und Peter Neuhaus wählten obiges Zitat, um im Jahr 2001 ihre Arbeit zum digitalen Archiv der KISD zu eröffnen. Fast elf Jahre später, als die Initiatoren des selbstinitiierten Projekts »Arche. Reanimation des Archivs.« mich beiläufig fragten, ob ich nicht Lust habe, mich als Programmierer an ebendiesem zu beteiligen, war ich des Chaos noch nicht gewahr.

Inzwischen habe ich es erfahren, und erkannt, dass Chaos und Ordnung vereinbar sind. Mehr noch: Die Ordnung inhärenter Bestandteil des Chaos und die Umwandlung, von der Hesse spricht, lediglich unsere Wahrnehmung desselben verändert – nicht etwa das Chaos selbst.

Insofern stellt das in dieser Arbeit dokumentierte KISDarchiv eine Linse dar, die dem Chaos als maskengleiche Instanz vorgeschaltet ist, um die Erfahrung replizierbar zu gestalten.

Doch dies soll keine Abhandlung über das Chaos sein. Die Arbeit dokumentiert die Entwicklung des KISDarchivs und bildet ein Kompendium für alle, die hinter die Maske schauen wollen, um einen Blick auf die Mechanismen zu werfen, welche die Transformation des Chaos begleiten. Sie soll als Einladung zur Auseinandersetzung, als Fundament zur Weiterentwicklung, als Anregung für zukünftige Entwicklungen und als Beispiel für maßlosen Schachtelsatzbau und den inflationären Gebrauch des Verbs »implementieren« dienen.**

Gegliedert ist die Arbeit in drei Teile:

1. **Realisation:** Der erste Teil beschreibt die Entwicklung des KISDarchivs.
2. **Operation:** Der zweite Teil dient als Handbuch und API.
3. **Evolution:** Der dritte Teil richtet den Blick auf die Zukunft des Systems.

* zitiert aus: Archibald. Konzept für ein digitales Archiv des Fachbereichs Design der Kölner Fachhochschule, Vordiplom Hauptthema von Michaela Kunze und Peter Neuhaus, 2001.

** Summa summarum tritt das Wort 70 Mal auf.

Zum Zeitpunkt, da diese Arbeit geschrieben wird, befindet sich das hier dokumentierte KISDarchiv nach wie vor im Betastadium. Tatsächlich trug die abermalige Reflektion in Textform dazu bei, Abläufe zu optimieren und Anregungen für die Weiterentwicklung des Systems zu gewinnen.

Die Arbeit ist von Redundanz geprägt, da sie, wie das System, welches sie dokumentiert, verschiedene Perspektiven auf den Betrachtungsgegenstand anwendet. Sie ist somit beliebig lesbar und will erfahren werden.

Ich hoffe, dass die Arbeit dazu beiträgt, das KISDarchiv als festen Teil des virtuellen Handwerkszeugs der Studierendenschaft zu etablieren, und künftige Hüter, AG-Mitglieder, Studierende und KISD-Menschen dazu motiviert, das analoge und digitale Erbe der KISD zu bewahren und daraus zu schöpfen.

Köln, im März 2014.

1.0 Dank

Diese Arbeit und das ihr zugrunde liegende KISDarchiv wäre ohne die folgenden Menschen nicht zustande gekommen. Meinen herzlicher Dank gilt

- **Andreas Johannes Katona**, Archivar a. D., für unzählige Stunden gedankenvoller, sorgfältiger Auseinandersetzung mit dem Archiv in allen seinen Formen und der Hochschule, die es umgibt. Für wertvolle Einsichten, seine Beteiligung am Betatest und die Idee, diese Arbeit zu schreiben.
- **Vitus Schuhwerk**, für außerordentliches Engagement als Betatester, schlaue Einsichten in die Abläufe der digitalen KISD, Ansporn und Zucchini-Suppe.
- Den Mitgliedern des SI »Arche«: **Johannes Eisbrenner, Lea Fumy, Joshua Horz, Andreas Johannes Katona, Philipp Mies, Laura Schraven, Carla Zimmermann** und **Tim Zähres** für ihren Einsatz im Dienste des analogen und digitalen Erbes der KISD und ihr anhaltendes Vertrauen.
- **Anne Schirner**, für nützliche Einblicke in die internen Abläufe des Archivs.
- **Till Maria Jürgens**, für Orientierung in den Untiefen der KISDspaces, virtuelle Allmacht und vorbehaltloses Vertrauen in meine guten Absichten.
- **Clemens Wahlig**, für MacGyver'sches Geschick in der Aufrechterhaltung der technischen Infrastruktur trotz aller Widrigkeiten.
- **Prof. Andreas Muxel**, für Interesse und Anteilnahme an diesem Projekt.
- **Prof. Andreas Wrede**, für Vertrauen und Interesse.
- **Prof. Björn Bartholdy**, für die Betreuung dieser Arbeit.
- **Richard Jungkunz**, für die Vermittlung eines Betreuers für diese Arbeit.
- **Elisabeth Kaliva**, für Vertrauen und »grünes Licht«.
- **David Grieshammer** und **Colin Joy** für wertvolle Einsichten in die KISDspaces.
- **Andreas Beutel**, für zuverlässige Korrespondenz und technischen Support.
- Der **Archiv-AG** der KISD, für vergangene und zukünftige Kooperation.

Inhalt

I	REALISATION	15
1	Einleitung	17
2	Entwicklungschronologie	19
2.0	Einleitung	19
2.1	Pre Alpha	19
2.2	Alpha	20
2.3	Closed Beta	24
2.3.1	Streamlining	24
2.4	Limited Public Beta	28
2.4.1	Sandbox-Umzug	28
2.5	Pre Public Beta	29
2.5.1	Umzug auf das Live-System	29
2.6	Public Beta	30
2.6.1	Inbetriebnahme	30
3	Altes digitales Archiv (AdA)	31
3.0	Einleitung	31
3.1	Frontend	31
3.2	Gruppenordner	32
3.3	Alte Metadatenbank (AMD)	33
4	Vergleichbare Konzepte	34
4.0	Archibald	34
4.1	Blended Studies	34
5	SI »Arche«	35
5.0	Einleitung	35
5.1	Analoges Archiv	35
5.2	Digitales Archiv	36
6	Vorüberlegungen	37
6.0	Einleitung	37
6.1	Lehren	37
6.1.1	Das Wireframe-Dilemma	38
6.1.2	Flexibilität durch Nonlinearität	38
6.2	Technische Voraussetzungen	39
6.2.1	Plattform	39
6.2.2	Framework	39
6.3	Konventionen	40
6.3.1	Sprache	40
6.3.2	Nomenklatur	40
6.3.3	Dateinamen/-pfade	41
6.3.4	Permalinks	41

7	Schaffung der Datenstruktur	42
7.0	Einleitung	42
7.1	Datenbankmodell	42
7.2	Neue Metadatenbank (NMD)	43
7.3	Entwicklung von Posttypen	43
8	Backend-Anpassung	44
8.0	Einleitung	44
8.1	Eintrags-Einzelansichten	45
8.2	Eintrags-Listenansichten	46
8.3	Navigation	47
8.4	Dashboard	47
9	Datenimport	48
9.0	Einleitung	48
9.1	Datenquellen	48
9.2	Mapping und Datenaufbereitung	49
9.2.1	Mapping	50
9.2.2	Konsolidierung	54
9.3	Einlesevorgang	54
9.4	Datenabgleich & Zweitimport	54
9.5	Datenannektion	56
9.6	Iterative Defragmentierung	56
9.7	Datenbankpflege	57
10	Frontend-Entwicklung	58
10.0	Einleitung	58
10.1	Typografie	59
10.2	Wortmarke	61
10.3	Formularelemente	61
10.3.1	Suchfeld	61
10.3.2	Metasuchfeld	61
10.3.3	Textfeld	61
10.3.4	Dropdown	62
10.3.5	Tagfeld	62
10.3.6	Button	62
10.4	Elemente	63
10.4.1	Elementtypen	63
10.5	PDF-Ansicht	64
10.6	Piktogramme	64
10.7	Interaktive Headline	65
10.7.1	Schlagwort-Mehrfachansicht	65
10.7.2	Schlagwort-Einzelansicht	65
10.7.3	Autoren-Einzelansicht	65
10.7.4	Lehrenden-Einzelansicht	66
10.7.5	Lesezeichen-Seite	66
10.8	Farben	67
10.8.1	Wortmarke	67
10.8.2	Hervorhebungen, Hinterlegungen	67
10.8.3	Eintrags-Farbkodierung	67

10.9	Ansichten	69
10.9.1	Einzelansichten	69
10.9.2	Mehrfachansichten	70
10.9.3	Mehrfachansicht Detail	71
10.9.4	Mehrfachansicht Karten	72
10.9.5	Mehrfachansicht Tabelle	74
10.10	Templates und Module	75
10.10.1	Startseite	76
10.10.2	Navigators	77
10.10.3	Suche	81
10.10.4	Metasuche	83
10.10.5	Metasuche / Kontextuelle Hilfe	88
10.10.6	Lesezeichen	89
10.10.7	Schlagworte	91
10.10.8	Autorenindex	93
10.10.9	Integrität	94
10.10.10	Statistik	97
10.10.11	Arbeit einreichen	98
10.10.12	Das analoge Archiv	108
10.10.13	Bugs-Seite	109
II	OPERATION	111
1	Posttypen	114
1.0	Einleitung	114
1.0.1	Posttyp-Struktur	114
1.1	ark_record / Eintrag	115
1.2	ark_area / Lehrgebiet	118
1.3	ark_author / Autor/in	119
1.4	ark_lecturer / Prüfer/in	120
2	Taxonomien	121
2.0	Einleitung	121
2.1	ark_author_groups	121
2.2	ark_lecturer_role	121
2.3	ark_lecturer_groups	121
3	Funktionsübersicht / API	122
3.0	Einleitung	122
3.0.1	Funktionsstypen	122
3.0.2	Funktionsaufruf	122
3.0.3	Beispiel Funktionsdokumentation	123
3.1	Templatefunktionen	124
3.1.1	ark_get_area_by_ID()	125
3.1.2	ark_get_area_meta()	125
3.1.3	ark_get_author_by_ID()	126
3.1.4	ark_get_author_degree()	127
3.1.5	ark_get_author_name()	127
3.1.6	ark_get_author_meta()	128
3.1.7	ark_get_author_for_permalink()	128

3.1.8	ark_get_elements()	129
3.1.9	ark_get_excerpt_by_id()	129
3.1.10	ark_get_lecturer_by_ID()	130
3.1.11	ark_get_lecturer_meta()	130
3.1.12	ark_get_record_subexam()	131
3.1.13	ark_get_tile()	131
3.1.14	ark_greeting()	132
3.1.15	ark_search_wisecrack()	132
3.1.16	ark_show_bug_button()	132
3.1.17	ark_show_view_modes()	133
3.1.18	ark_view_table()	133
3.2	Helferfunktionen	134
3.2.1	ark_custom_menu_order()	136
3.2.2	ark_add_dashboard_widget()	136
3.2.3	ark_add_post_meta_boxes()	136
3.2.4	ark_cpt_icons()	136
3.2.5	ark_create_menu()	137
3.2.6	ark_author_set_taxonomy()	137
3.2.7	ark_lecturer_set_taxonomy()	137
3.2.8	ark_register_settings()	137
3.2.9	declare_ark_area()	138
3.2.10	declare_ark_author()	138
3.2.11	declare_ark_lecturer()	138
3.2.12	declare_ark_record()	138
3.2.13	ark_remove_menu_entries()	139
3.2.14	ark_get_author_by_name()	139
3.2.15	ark_integrity_meta_box()	139
3.2.16	ark_generate_note_xml()	140
3.2.17	ark_generate_pdf_url()	140
3.2.18	ark_cmp()	140
3.2.19	ark_get_ID_by_title()	141
3.2.20	ark_get_lecturer_by_name()	141
3.2.21	ark_get_lecturer_role()	141
3.2.22	ark_get_new_author_meta()	142
3.2.23	ark_get_valid_year_range()	142
3.2.24	ark_queue()	142
3.2.25	ark_current_page_url()	143
3.2.26	ark_sort_record_by_year()	143
3.2.27	ark_record_post_title()	143
3.2.28	ark_zahl2wort()	144
3.2.29	utf8_strrev()	144
3.2.30	utf8_strshuffle()	144
3.2.31	ark_acf_relationship_result()	145
3.2.32	ark_acf_generate_search_meta()	145
3.2.33	ark_filter_wp_title()	145
3.2.34	ark_acf_load_value()	146
3.2.35	ark_acf_update_value()	146
3.2.36	ark_permalinks()	147
3.2.37	ark_query_post_type()	147
3.2.38	ark_remove_mass_edit_bulk_action()	147

3.2.39	ark_relationship_query()	148
3.2.40	ark_set_title()	148
3.2.41	ark_update_slug()	148
3.2.42	ark_remove_row_actions()	149
3.2.43	ark_replace_excerpt_more()	149
3.2.44	ark_set_default_title()	149
3.2.45	ark_wpautop()	150
3.2.46	ark_add_rewrite_rules()	150
3.2.47	ark_admin_header()	150
3.2.48	ark_deregister_styles()	150
3.2.49	ark_dashboard_widget_function()	151
3.2.50	ark_post_meta_boxes_setup()	151
3.2.51	ark_remove_media_controls()	151
3.2.52	ark_settings_page()	151
3.2.53	ark_setup()	152
3.2.54	ark_fix_attachment_meta()	152
3.2.55	ark_fix_permalinks()	152
3.2.56	ark_generate_search_meta()	152
3.3	Includes	153
3.3.1	ark-dir-ext.php	154
3.3.2	ark-get-adv-search-results.php	154
3.3.3	ark-get-author-degree-by-name.php	154
3.3.4	ark-get-lecturer-area-by-name.php	155
3.3.5	ark-get-lecturer-id-by-name.php	155
3.3.6	ark-get-word-from-number.php	155
3.3.7	ark-pdf-url-check.php	156
3.3.8	ark-record-author-dir.php	156
3.3.9	ark-search-author.php	156
3.3.10	search_ark_record_author.php	157
3.3.11	search_ark_record_lecturer.php	157
3.3.12	search_ark_record_tag.php	157
4	Cookies	158
4.0	Einleitung	158
4.1	ark_view	158
4.2	ark_queue	158
5	JavaScript	159
5.0	Einleitung	159
5.1	Bibliotheken und Plugins	159
5.1.1	jQuery	160
5.1.2	jQuery Migrate	160
5.1.3	jQuery UI	160
5.1.4	mapAttributes jQuery Plugin	160
5.1.5	jQuery Columnview	160
5.1.6	jQuery Dropdown	161
5.1.7	jQuery Tagsinput	161
5.1.8	jQuery Cookie	161
5.1.9	jQuery fancybox	161
5.1.10	jQuery Validation Plugin	161

5.1.11	jQuery Bitly	162
5.1.12	jQuery sortElements	162
5.1.13	jqBarGraph	162
5.1.14	Google Charts	162
5.1.15	jQuery Iframe Transport Plugin	162
5.1.16	jQuery File Upload Plugin	163
5.1.17	jQuery Highlight Plugin	163
5.1.18	jQuery BBQ: Back Button & jQuery Library	163
5.1.19	jQuery hashchange event	163
6	Templates	164
6.0	Einleitung	164
6.0.1	Seitentemplates	164
6.0.2	Subtemplates	164
6.1	KISDarchiv-Templates	167
6.1.1	ark_index.php	168
6.1.2	ark_navigator.php	169
6.1.3	ark_search.php	170
6.1.4	ark_tags.php	171
6.1.5	single-ark_author.php	172
6.1.6	single-ark_lecturer.php	173
6.1.7	ark_author_directory.php	174
6.1.8	ark_statistics.php	175
6.1.9	ark_queue.php	176
6.1.10	ark_pdf_view.php	177
6.1.11	ark_upload.php	178
6.1.12	ark_integrity.php	179
6.1.13	ark_about.php	180
6.1.14	ark_bugs.php	181
6.1.15	ark_upload_complete.php	182
6.1.16	ark_help.php	183
6.2	WordPress Templates	184
6.2.1	search.php	185
6.2.2	tag.php	186
6.2.3	404.php	187
6.2.4	single.php	188
6.3	KISDarchiv-Subtemplates	189
6.3.1	ark_author.php	190
6.3.2	ark_lecturer.php	191
6.3.3	ark_view_detail.php	192
6.3.4	ark_view_table.php	193
6.3.5	ark_view_tiles.php	194
6.3.6	pdf.php	195
6.3.7	no_pdf.php	196
6.4	WordPress-Subtemplates	197
6.4.1	header.php	198
6.4.2	footer.php	198
6.4.3	searchform.php	198
6.4.4	entry.php	199
6.4.5	entry-content.php	200

6.4.6	entry-summary.php	200
6.4.7	entry-meta.php	200
6.4.8	entry-footer-single.php	200
6.4.9	nav-below.php	201
7	WordPress	202
7.0	Einleitung	202
7.1	Dateistruktur	202
7.2	Datenbank	203
7.2.1	Struktur	203
7.3	WordPress-Plugins	203
7.3.1	Advanced Custom Fields (ACF)	204
7.3.2	ACF Flexible Content Field	204
7.3.3	Codepress Admin Columns	204
7.3.4	KISDarchive	204
7.3.5	Relevanssi	204
7.3.6	Revision Control	205
7.3.7	Right Now Reloaded	205
7.3.8	wp-Typography	205
7.3.9	wp Permalauts	205
7.3.10	WP Ultimate CSV Importer	205
7.4	Benutzerrollen	206
7.4.1	Super-Administrator	206
7.4.2	Administrator	206
7.4.3	Editor	206
7.4.4	Author	206
7.4.5	Contributor	206
7.4.6	Subscriber	206
8	Abläufe	207
8.0	Einleitung	207
8.1	Eintrag (ark_record)	207
8.1.1	Einträge durchsuchen	207
8.1.2	Einträge anlegen / einreichen	208
8.1.3	Einträge bearbeiten	209
8.2	Autor/in (ark_author)	211
8.2.1	Autoren durchsuchen	211
8.2.2	Autoren anlegen	211
8.2.3	Autoren bearbeiten / löschen	213
8.3	Prüfer/in (ark_lecturer)	214
8.3.1	Prüfer durchsuchen	214
8.3.2	Prüfer anlegen	215
8.3.3	Prüfer bearbeiten / löschen	216
8.4	Lehrgebiet (ark_area)	217
8.4.1	Lehrgebiete durchsuchen	217
8.4.2	Lehrgebiete anlegen	218
8.4.3	Lehrgebiete bearbeiten	219
8.5	Studiengang	220
8.5.1	Studiengang anlegen	220

III	EVOLUTION	223
1	Einleitung	225
2	Daten	225
2.0	Einleitung	225
2.1	Aufbereitung vorhandener Daten	225
2.1.1	PDF-Datenbestand	225
2.1.2	Metadatenbestand	226
2.2	Digitalisierung von Altdaten	226
3	Struktur	227
3.0	Einleitung	227
3.1	Erweiterungen	227
3.1.1	Hilfe / Feedback-System	227
3.1.2	Studiengangsverwaltung	227
3.1.3	Verflechtung mit KISDspaces	227
3.1.4	Lehrveranstaltungsdocumentationen	228
3.1.5	Modulare Startseite	228
3.1.6	Wartungstools	228
4	TODO-Liste	229
IV	APPENDIX	231
1	Wireframes	232
1.0	Einleitung	232
1.1	SI »Arche«	232
2	Diagramme und Modelle	244
2.0	Einleitung	244
2.1	UML-Diagramme	244
2.2	Datenbankmodelle	246
2.2.1	WordPress	246
2.2.2	KISDarchiv (Pre Alpha)	247
2.3	Integritätsdiagramme	248
3	Code	250
3.0	Einleitung	250
4	Daten	250
4.0	Einleitung	250
5	Quellen	251
5.0	Einleitung	251
5.1	spaces.kisd.de/NeDA	251
5.2	Videos	253
6	Index	255

Erster Teil

REALISATION

1 Einleitung

Im ersten Teil der Arbeit werden alle Entscheidungsprozesse der Entwicklung dargestellt, sowie die technischen Grundlagen erläutert.

Dabei orientiert sich dieser Teil grob an der chronologischen Entwicklung des KISDarchivs, welche sich im ersten Schritt den bereits vorhandenen Daten zuwendet, um daraus zum einen Modelle für das neue digitale Archiv abzuleiten, zum anderen in der Arbeit mit den Daten Anforderungen an die im zweiten Schritt behandelte Entwicklung des Frontends ableitet.

Diese Vorgehensweise bekommt bei einem System wie dem KISDarchiv mit vielen Tausend Datensätzen besonderes Gewicht. Als Gestaltern sollte uns bewusst sein, dass jedes Interface lediglich eine Repräsentation einer unterliegenden Datenstruktur nach einem willkürlich gewählten Muster darstellt. Jede im Interface gewünschte Funktionalität muss in der zugrundeliegenden Struktur abzubilden sein. Im Falle dieses Projekts kam hinzu, dass bereits tausende Metadaten vorlagen, die für die Überführung in die neue Struktur aufbereitet werden mussten. Auch dies wird hier beschrieben.

Die Entwicklung lässt sich in mehrere Phasen aufteilen:

PRE ALPHA	OKTOBER 2011 - FEBRUAR 2013 Theoretische Auseinandersetzung.
ALPHA	FEBRUAR 2013 - AUGUST 2013 Entwicklung der Grundfunktionen.
CLOSED BETA	AUGUST 2013 - OKTOBER 2013 Entwicklung des Frontends, Erweiterung des Funktionsumfangs.
LIMITED PUBLIC BETA	NOVEMBER 2013 Interne Testphase unter Realbedingungen, Bugfixes.
PRE PUBLIC BETA	NOVEMBER 2013 - FEBRUAR 2014 Vorbereitung der Öffnung gegenüber KISD-Menschen, Bugfixes.
PUBLIC BETA	FEBRUAR 2014 - KISD-interne Öffnung des KISDarchivs.

2 Entwicklungsschronologie

2.0 Einleitung

An dieser Stelle wird in chronologischer Folge fragmentarisch die Entwicklung des KISDarchivs dargestellt.

2.1 Pre Alpha

OKTOBER 2011

11. 10. 2011 Das erste Treffen des selbst initiierten Projekts »Arche. Reanimation des Archivs« findet in Raum 126 statt.

DEZEMBER 2011

1. 12. 2011 In einem Treffen mit Tim Zähres und Laura Schraven vom SI »Arche«, sowie Elisabeth Kaliva, Colin Joy, David Grieshammer und mir werden die von Tim und Laura konzipierten Wireframes und die Verknüpfung von digitalem Archiv und KISDspaces besprochen.

3. 12. 2011 Ein erstes relationales Datenbankmodell bildet die künftige Datenstruktur und alle darauf geplanten Funktionalitäten ab. Das Modell geht von einer komplett WordPress-autarken Datenbank aus – Eine Idee, die später zugunsten einer integrierten Lösung verworfen werden sollte.

15. 12. 2011 In einem Treffen mit Mitgliedern des SI »Arche« werden die überarbeiteten Wireframes besprochen.

ABB. 1 Wireframe-Besprechung in Raum 126.

v.l.n.r:
Willem Rabe, Tim Zähres,
Philipp Mies.



JANUAR 2012

20. 1. 2012 Der erste Versuch einer KISDarchiv-Umsetzung auf WordPress beginnt. Diese Iteration verfügt über ein Backend-Modul zur Importierung von Datensätzen direkt aus der Alten Metadatenbank (AMD). Die Implementierung gestaltet sich aufgrund großer Inkonsistenzen im Datenbestand derart komplex, dass mangels zeitlicher Ressourcen die Umsetzung des Projekts vorerst auf Eis gelegt wird. Ein Jahr wird vergehen, bis die Umsetzung ein weiteres Mal und auf einer neuen WordPress-Version erneut in Angriff genommen wird.

2.2 Alpha

FEBRUAR 2013

7. 2. 2013

Das SI »Arche« verkündet im KISD Intranet, dass alle digital vorliegenden Arbeiten auf dem Gruppenordner geladen wurden und somit online vorliegen.

Der KISDarchiv-Entwicklungsspace (spaces.kisd.de/NeDA/) geht online:

ABB. 2 Entwicklungsspace;
spaces.kisd.de/NeDA/



Der Space wird über die nächsten zwölf Monate als Entwicklungstagebuch für die Arbeit am **Neuen digitalen Archiv (NeDA)**, wie das KISDarchiv während seiner Entwicklung genannt wird, dienen.

Die Analyse der vorliegenden Daten und die darauf basierende Konzeption der Posttypen bildet den Ausgangspunkt für die Arbeit am KISDarchiv.

- 8. 2. 2013 Da der Zugriff auf den deutschen Webserver aus Japan mit erhöhten Latenzzeiten verbunden ist, wird eine lokale Entwicklungsumgebung auf einem alten Laptop aufgesetzt, welcher bis zur Limited Public Beta im September 2013 das KISDarchiv behausen wird. Das Dateisystem des lokalen Servers wird live in die Cloud gespiegelt, regelmäßige Datenbank-Backups sollen zudem den Erhalt Der Datenbasis im Falle eines Systemausfalls sicherstellen.
- 8. 2. 2013 Der Import der Altdaten aus AMD und NMD in die KISDarchiv-Datenbank beginnt. Das Such-Plugin »Relevanssi« wird implementiert.
- 9. 2. 2013 Die Archiv-AG vermeldet im KISD Intranet, dass alle gedruckten und digitalen Dokumente eingepflegt seien. Die KISDarchiv-Posttypen erfahren diverse Modifikationen um neue Funktionalitäten abbilden zu können.
- 11. 2. 2013 Der Datenimport der AMD- und NMD-Daten wird abgeschlossen. Backend-Eingabemasken und Listenansichten sowie die Navigation erfahren Anpassungen, um die Posttypen besser abzubilden.
- 12. 2. 2013 Die Eingabe- und Bearbeitungsmaske des Posttyps **ark_record** wird in erster Instanz vorgestellt und im NeDA-Entwicklungsspace besprochen. Der Datenabgleich zwischen AMD, NMD und Gruppenordner beginnt. Siehe **Datenimport** (S. 48).
- 19. 2. 2013 Prof. Andreas Muxel tritt dem NeDA-Entwicklungsspace bei.
- 27. 2. 2013 Der importierte Schlagwortbestand der Altdaten wird analysiert und mit der Aufbereitung und partiellen Konsolidierung der Schlagworte begonnen.

28. 2. 2013 Die Schlagwort-Aufbereitung ist abgeschlossen.
Die Entwicklung des Frontends beginnt.

MÄRZ 2013

3. 3. 2013 In einem ersten Post zum Frontend im NeDA-Entwicklungsspace wird der Gedankengang dargelegt, erst mit den Arbeiten am Frontend zu beginnen, sobald eine solide Datenbasis vorhanden ist, auf Basis derer alle die Gestaltung betreffenden Entscheidungen validiert werden können.
5. 3. 2013 Der Fokus in der Entwicklung der Frontend-Ansichten und Module wird auf die Darstellung der über 4000 Prüfungsarbeiten gelegt, da diese die LV-Dokumentationen quantitativ um den Faktor 20 übertreffen und über wesentlich konsistentere Metadaten verfügen.

ABB. 3 Startseite (alpha),
März 2013.



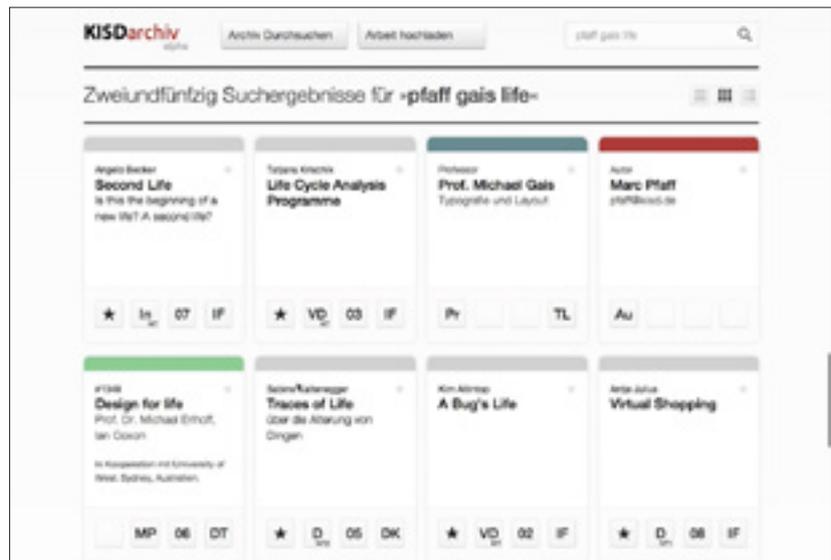
6. 3. 2013 Eine erste Demonstration in Videoform zeigt die Startseite, den Navigator, die Eintrags-Einzelansicht, die Eintrags-Bearbeitungsmaske samt Integritäts-Metabox im Backend, die diversen Mehrfachansichten, die Suche und die Schlagwort-Einzelansicht.
7. 3. 2013 Unterliegende Darstellungsfunktionen erfahren grundlegende Optimierungen. Die Einträge der Posttypen **ark_record** und **ark_lecturer** werden um Taxonomien erweitert, um das Geschlecht der Personen erfassen zu können.
9. 3. 2013 Code-Konsolidierungen und -Optimierungen werden durchgeführt um kompaktere Funktionen zu schaffen, die in ihrer parametrischen Form eine höhere Modularität und Dokumentierbarkeit in einer hypothetischen API erlauben. Eine Prioritätenliste wird erstellt.
10. 3. 2013 Das **Briefing zur Einpflege digitaler Dokumente** wird im Arbeitsspace der Archiv-AG veröffentlicht. Das Dokument instruiert die AG-Mitglieder über den Umgang mit der NMD und die korrekte Ablage von Daten auf dem Gruppenordner um bis zur Fertigstellung des KISDarchivs weiterhin nutzbare Metadaten zu generieren. Überlegungen zur Erfassung des Studiengangs in Metadatenform werden angestellt und im NeDA-Entwicklungsspace besprochen.
13. 3. 2013 Prof. Andreas Wrede tritt dem NeDA-Entwicklungsspace bei.

APRIL 2013

1. 4. 2013 In einer Demonstration in Videoform wird die überarbeitete Kartenansicht

vorgestellt, welche diverse Posttypen differenzierter abbildet:

ABB. 4 Kartenansicht (alpha), April 2013.



Diese frühe Iteration der Kartendarstellung zeigt den inzwischen entlassenen Posttyp **ark_course** («Lehrveranstaltung»), welcher zu diesem Zeitpunkt im System erfassten Projekt- und Seminardokumentationen zugeordnet ist. Einträge diesen Typs werden mit grünen Farbbalken farbkodiert. Die farbliche Unterscheidung der einzelnen Prüfungsstufen ist noch nicht implementiert.

Ebenfalls vorgestellt wird die erste Realisierung der Lesezeichen-Seite:

ABB. 5 Lesezeichenseite mitsamt PDF-Ansicht (alpha), April 2013.



Die erste Iteration der Lesezeichen-Seite zeichnet sich durch eine linksbündige Liste der mit Lesezeichen versehenen Arbeiten und eine gerahmte Darstellung der PDF-Inhalte aus. Diese Art der Darstellung wird sich als zu sperrig erweisen – Die Lesezeichen-Liste selbst kann nur eine begrenzte Anzahl Einträge in der einspaltigen Darstellung unterbringen, die PDF-Darstellung leidet unter dem begrenzten ihr zur Verfügung stehenden Raum. Dies soll später durch die separate Darstellung von Liste und PDF-Darstellung gelöst werden.

Der Lesezeichen-Indikator, welcher die Anzahl mit Lesezeichen versehener Einträge abbildet, ist bereits in dieser frühen Iteration implementiert. Ebenso weist die Lesezeichenliste bereits die Bezeichner »laden«, »speichern« und »alle löschen« auf – Indiz für die geplanten Lesezeichen-Funktionalitäten.

Die erste Iteration der Metasuche, damals noch »Erweiterte Suche« genannt, bildet den Abschluss des Demo-Videos:

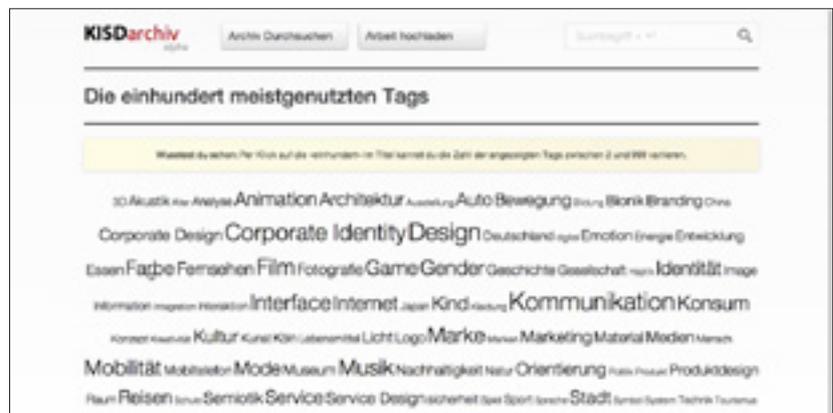
ABB. 6 Metasuche (alpha), April 2013.



Die erste Version der Metasuche ist als übergroßes Formular gestaltet, welches die Bestandteile der Suchanfrage in diverse Auswahl- und Eingabefelder aufspaltet. Dies bietet einen guten Überblick über die durchsuchbaren Kriterien, hat allerdings zum Nachteil, dass die Formulierung komplexer Anfragen eine hohe Anzahl an Klicks erfordert (zwei pro Kriterium). Die Eingabe-Elemente in dieser Form werden nicht weiter verwendet, dienen allerdings als Inspiration für das Feature der interaktiven Headline.

2. 4. 2013 Die Schlagwortübersicht wird vorgestellt, welche sich erstmals des Bedienelements der interaktiven Headline bedient:

ABB. 7 Schlagwortübersicht (alpha) mit nicht hervorgehobener interaktiver Headline, April 2013.



Die Entwicklung des KISDarchivs ist mit Ende der Alpha-Phase an einem Punkt angelangt, an dem das System über viele der elementaren Funktionen verfügt. Noch scheint jedoch kein Ende in Sicht – Mit der Arbeit einreichen-Maske steht eine technische Herausforderung bevor, deren Ausmaß schwer zu erahnen ist, gleichzeitig befindet sich der Metadatenbestand der LV-Dokumentationen nach wie vor in einem desolaten Zustand. Erschwerend hinzu kommt, dass die für das Projekt veranschlagte Zeit mit dem bevorstehenden Ende des Auslandsaufenthalts seines Programmierers knapp wird. Mangels zeitlicher Ressourcen wird das Projekt von April bis Anfang September 2013 wieder einmal auf Eis gelegt.

2.3 Closed Beta

AUGUST 2013

08./2013 Nach der Rückkehr der Entwicklungsabteilung nach Köln werden die Arbeiten am KISDarchiv wieder aufgenommen.

SEPTEMBER 2013

4. 9. 2013 In einem Treffen mit Till Maria Jürgens werden die nächsten Schritte besprochen. Das Streamlining des KISDarchivs wird beschlossen.

2.3.1 Streamlining

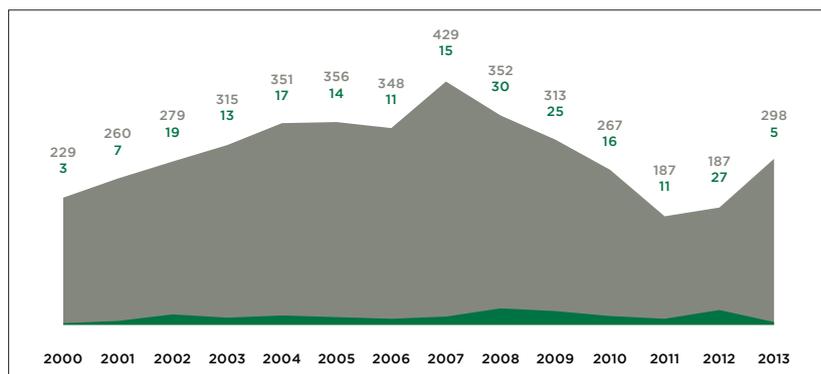
Das Alte digitale Archiv hatte sowohl Prüfungs- als auch Lehrgebietsdokumentationen («LV-Dokus») geführt. Es erschien also als logischer Schritt, diese Daten auch im KISDarchiv abzubilden.

Die Gestaltung der Posttypen und die ihnen zugewiesenen Bezeichnungen spiegeln dies wider: Ein Posttyp **ark_course** für Veranstaltungen war geschaffen worden, der Posttyp **ark_record** zweigeteilt angelegt, um Prüfungs- und LV-Dokumentationen in einem gemeinsamen Posttyp abbilden zu können.

ABB. 8 Digital vorliegende Datensätze nach Jahren, 2000–13.

Das Flächendiagramm zeigt Prüfungsarbeiten (grau) versus Lehrveranstaltungs-dokumentationen (grün).

Gesamtanzahl (2000–13):
4171 Prüfungsarbeiten
213 LV-Dokumentationen



Zu diesem Zeitpunkt in der Entwicklung zeichnet sich jedoch ab, dass die Zweiteilung des KISDarchivs in dieser Form nicht praktikabel ist.

Einige Gründe dafür:

- Die Entwicklung eines zweigeteilten Systems bedeutet einen deutlich erhöhten zeitlichen Aufwand, der an dieser Stelle nicht zu erbringen ist.
- Die Prüfungsarbeiten sind den LV-Dokumentationen quantitativ bereits um den Faktor 20 überlegen. Diese Diskrepanz würde sich angesichts der stagnierenden Einreichungszahlen der LV-Dokumentationen voraussichtlich noch weiter ausprägen, womit Projekt- und Seminararbeiten ein Nischendasein im KISDarchiv fristen würden. Ihr Erscheinen innerhalb von Prüfungsarbeiten dominierten Mehrfachansichten würde ihnen irritativen Charakter verleihen.
- Zu den LV-Dokumentationen liegen kaum nutzbare Metadaten vor, was eine manuelle Rekonstruktion derselben nach sich gezogen hätte.

- Das Interesse der Studierendenschaft an Prüfungsdokumentationen wurde gegenüber LV-Dokumentationen als deutlich höher bewertet, da diese in der Recherche für die eigenen Prüfungsthemen relevant sind.
- Seitens des Blended-Teams wurde der Trend aufgezeigt, dass mehr und mehr LV-Dokumentationen direkt über die KISDspaces erbracht werden, die »klassische« PDF-Dokumentation hingegen praktisch extinkt ist.
- Eine Verschlinkung des Systems auf Prüfungsdokumentationen würde sowohl Backend-Oberflächen zur Pflege von Arbeiten als auch Frontend-Templates deutlich vereinfachen. Ein derartiger Fokus auf Prüfungsarbeiten würde zudem eine differenziertere, maßgeschneiderte Erfahrung ermöglichen.
- Eine Verschlinkung des Systems würde den Wartungsaufwand verringern, da das KISDarchiv nicht über eine eigene Veranstaltungsdatenbank verfügen müsste, die ständiger Pflege bedürfte.

Die Veranstaltungsdaten, der Posttyp **ark_course** und alle Einträge, welche Projekt- und Semindokumentationen zugeordnet waren, wurden aus der KISDarchiv-Datenbank entfernt und die Frontend-Templates angepasst.

Die Frage nach dem künftigen Umgang mit LV-Dokumentationen soll im dritten Teil dieser Arbeit besprochen werden, mehr dazu auf Seite 228.

27. 9. 2013 Die Arbeit einreichen-Maske wird in Videoform vorgestellt.

ABB. 9 Backend; Navigator-Cache Einstellungen (admin)



28. 9. 2013 Der Navigator wird um einen Cache erweitert, sodass die komplexe Berechnung der ihm zugrunde liegenden Daten nur in gewissen Intervallen stattfinden hat. Die Cache-Gültigkeit wird auf 25 Tage gesetzt. Eine Einstellungsseite im Backend erlaubt Administratoren die Variation der Cachedauer sowie die Neugenerierung des Caches.

OKTOBER 2013

3. 10. 2013 Die interaktive Headline wird in weiteren Templates implementiert. Der Navigator wird um eine Ladeanimation ergänzt.

4. 10. 2013 Das asynchrone Auslesen und Speichern des Lesezeichen-Cookies via AJAX wird implementiert. Dies äußert sich in einem deutlichen Geschwindigkeitszuwachs in der Bedienung, da Nutzer nach der Betätigung des Lesezeichen-Buttons nicht mehr auf das Neu-Laden der Seiteninhalte warten müssen.

Die Farbkodierung von Einträgen nach Vorbild des analogen Archivs wird innerhalb von Mehrfachansichten implementiert.

Die Danke-Seite nach Upload-Abschluss, die seitenweite korrekte Paginierung innerhalb aller Mehrfachansichten, die interaktive Headline innerhalb der Autoren-Einzelansicht und die Funktion, Lesezeichenlisten zu leeren werden implementiert. Die Eintrags-Bearbeitungsmaske im Backend erfährt Optimierungen. Die Eintragseingabemaske versucht nun anhand der Eingabe von Autoren bzw. Prüfern mittels einer systemfremden Datenbank Rückschlüsse auf Studiengang und Lehrgebiet zu ziehen und wählt diese automatisch aus. Formular-Eingabefelder werden nun einheitlich(er) gestylt. Die Arbeit einreichen-Maske wird um eine Funktion mit Datenbankverknüpfung ergänzt, welche feststellt, ob eingegebene Autoren bereits im KISDarchiv existieren und diese bei Bedarf, sofern in der externen Datenquelle aufzufinden, mitsamt all ihrer Metadaten automatisch neu anlegt.

- 14. 10. 2013** Der PDF-Upload aus dem Frontend wird implementiert. Die konditionalen Eingabefelder der Uploadmaske werden vorgestellt. Im Rahmen eines Archiv-AG-Treffens findet die erste Live-Präsentation des KISDarchivs statt.

ABB. 12 Metasuche, zweite Iteration



- 15. 10. 2013** Die zweite Iteration der Metasuche wird vorgestellt. Die Eingabe der Kriterien ist nun in Form natürlicher Sprache möglich. Die Statistikseite wird vorgestellt.
- 16. 10. 2013** Die Metasuche erfährt Bugfixes und Erweiterungen.
- 17. 10. 2013** Die Hervorhebung des Suchbegriffs innerhalb der Suchergebnisse wird neu umgesetzt. Die Metasuche wird weiteren Bugfixes unterzogen.
- 18. 10. 2013** Die Validierung der Eingabefelder der Arbeit einreichen-Maske sowie die maximale Dateigröße für hochgeladene PDF-Dokumente und die Formatüberprüfung werden implementiert. Die Integritätsseite wird vorgestellt.
- 21. 10. 2013** Weitere Arbeiten an der Integritätsseite findet statt. Bugfixes an der Metasuche werden durchgeführt. Die relationalen Eingabefelder im Backend erfahren Modifikationen. Der Autorenindex wird vorgestellt.
- 22. 10. 2013** Die Arbeit-Einstellen-Maske im Frontend überprüft nun die Eingabe von mindestens fünf Tags und Autor/Prüfer. Dank der Ersetzung von Umlauten via Datenbankquery wird der Autorenindex nun korrekt sortiert.
- 24. 10. 2013** 498 auf alten Metadaten basierende Permalinks werden neu generiert.

Die 404-Seite schreibt nun eine E-Mail an den Admin, wenn der Besucher einem Deadlink gefolgt ist. Der Eintragseditor im Backend wird angepasst. Das Schlagwort-Feld in der Uploadmaske (Frontend) verarbeitet nun auch Tags, die per copy & paste ins Feld gelangen. Die Validierung der Eingabefelder in der Arbeit einreichen-Maske wird um Piktogramme ergänzt, welche visuelles Feedback zur Validität der Eingabe anzeigen.

25.10.2013 Die Arbeit einreichen-Maske wird finalisiert und die ersten 60 Hauptthemen darüber eingegeben. Durchschnittlich dauert dies 33 Sekunden pro Eintrag.

Der Umzug des KISDarchivs auf die Sandbox wird vorbereitet.

28.10.2013 Eine E-Mail an die Administratoren wird verschickt:

Hallo liebe Admins,

ich möchte das neue digitale Archiv (NeDA) gerne auf einem KISD-Server installieren. Es handelt sich um eine eigenständige WordPress-Installation. Dazu bedarf es einer Quota von 10GB zum Start, PHP 5.5.x samt MySQL, einer Subdomain archiv.kisd.de, die den Zugriff per .htaccess authentifiziert (wie derzeit my.kisd.de, es findet kein Login über die Spaces statt)*

**Alle bisherigen Daten auf dem Gruppenordner (groups.kisd.de/doku-archiv/) bleiben erhalten und werden referenziert, künftig werden die Daten direkt via WordPress hochgeladen und in einer menschenlesbaren Struktur abgelegt werden. Der Platzbedarf wird jährlich um 5GB wachsen.*

Sollte es Rückfragen geben bin ich von Mittwoch an wieder an der KISD, ansonsten per E-Mail erreichbar.

*Viele Grüße,
Willem*

2.4 Limited Public Beta

Die Limited Public Beta bedeutet einen großen Schritt für die Entwicklung des KISDarchivs: Erstmals seit Beginn der Arbeiten wird das System auf einem KISD-eigenen Server installiert und kann unter Realbedingungen getestet werden. Mit Andreas J. Katona und Vitus Schuhwerk können zwei hochqualifizierte und engagierte Betatester gewonnen werden, welche über wertvolle Einsichten in die innersten Abläufe des Archivs, der KISD und ihrer Server verfügen. Binnen zwei intensiver Wochen wird das System auf Herz und Nieren getestet, jede noch so minuziöse Anmerkung katalogisiert und in Quasi-Echtzeit verarbeitet.

2.4.1 Sandbox-Umzug

31.10.2013 Statt das KISDarchiv, wie in der oben zitierten E-Mail beschrieben, als eigenständige WordPress-Installation aufzusetzen, wird dieses nach Rücksprache

mit Till Maria Jürgens auf der KISDspaces Sandbox installiert.

Der Umzug auf Multisite-Umgebung der KISDspaces verläuft größtenteils reibungslos. Viele der potenziellen Fallstricke können durch vorausschauende Programmierung umgangen werden. Tatsächlich bedeutet der Umzug von der lokalen Entwicklungsumgebung auf einen dedizierten Server einen deutlichen Zuwachs in der Systemperformanz. Doch die Multisite-Umgebung weist auch Tücken auf: Die eigenwillige Weise, in welcher WordPress die hochgeladenen Dateien der diversen Blogs verwaltet, lässt sich nicht mit der Upload-Routine vereinbaren, die auf der lokalen Singlesite-Installation funktioniert hatte. Die Implementierung einer Lösung, welche die korrekte Verortung aus dem Front- und Backend hochgeladener Dateien nach einem strengen Regelwerk durchzusetzen hat, wird sich noch über Wochen hinziehen.

Mehr dazu im Kapitel **Arbeit einreichen**, S. 98.

NOVEMBER 2013

- 1. 11. 2013 Die kontextuelle Hilfe auf der Seite Metasuche wird implementiert, die Arbeit einreichen-Maske um ein Anmerkungsfeld erweitert. Die Bugs-Seite nimmt ihren Betrieb auf.
- 2. 11. 2013 Die Kategorisierung der Bugs-Seite wird eingeführt, um der Flut von Anmerkungen gerecht zu werden. Das Responsive Design wird angepasst.
- 3. 11. 2013 Die Statistikseite wird auf Google Charts umgestellt.
- 5. 11. 2013 Diverse Bugfixes werden umgesetzt, einige Hundert Permalinks von Master- und Vordiplomen neu generiert.
- 9. 11. 2013 Der PDF-Datenbestand wird statistisch untersucht.
- 13. 11. 2013 Die Limited Public Beta endet. Über 100 Bugfixes und Features wurden in den zwei Wochen seit ihrem Beginn umgesetzt.

2.5 Pre Public Beta

2.5.1 Umzug auf das Live-System

Nach zwei Wochen auf der Sandbox wird der Umzug auf das KISDspaces Live-System vollzogen. Wie schon beim Umzug auf die Sandbox ergeben sich keine größeren Probleme. Kopfzerbrechen bereitet lediglich die sehr kritische Speicherplatzsituation des Servers. Dieser ist zeitweise so nah an seiner Kapazitätsgrenze, dass WordPress-Grundfunktionen, wie der Dateiupload, systemweit versagen.

Aufgrund des Platzmangels wird angedacht, den KISDarchiv-Datenbestand an externer Stelle zu hosten. Außerdem zeichnet sich ab, dass der Gruppenordner in absehbarer Zeit nicht mehr als Speicherort für den rund 70 GB umfassenden Altdatenbestand fungieren würde. Ein komplettes Backup des Alt- und Neudatenbestands wird angefertigt.

Die Entwicklung liegt in Erwartung der Lösung der Speicherplatzfrage auf Eis.

FEBRUAR 2014

- 3. 2. 2014 Clemens Wahlig rettet das KISDarchiv mit der Anschaffung neuer Speichermedien. Der Server verfügt nun über genug Kapazität, um das KISDarchiv bei gleichbleibender Studierendenzahl der KISD bis ins Jahr 2154* zu hosten.
- 5. / 6. 2. 2014 Das komplette Alt- und Neudatenbackup mit einem Umfang von insgesamt 75 GB fließt durch glühende Leitungen auf die makellosen Festplatten des KISDspaces-Servers.
- 7. 2. 2014 Die Dateipfade in der Datenbank werden korrigiert, um die neue physische Position des Datenbestands abzubilden.
- 8. 2. 2014 Der Autor verbringt zwei Tage damit, einen Fehler zu beheben, welcher verhindert, dass aus dem Front- und Backend hochgeladene Dateien in einer menschenlesbaren, jedoch nicht menschenzugänglichen Dateistruktur verortet werden. Überdies ergibt sich ein Problem beim Zugriff auf innerhalb des Livesystem-eigenen Dateisystems abgelegten PDF-Dokumenten, welches Überraschung und Schrecken auslöst und letztendlich in der kompletten Neuschreibung des PDF-Ansichts-Templates mündet.

2.6 Public Beta

- 22. 2. 2014 Das KISDarchiv befindet sich 54 Wochen und konservativ geschätzten 1500 Arbeitsstunden nach Einläutung der Alpha-Phase in einem Zustand, der als *»für die KISD-Öffentlichkeit zumutbar«* bewertet wird. Zur Vorstellung des Produkts wird ein Video-Tutorial produziert, welches die Grundfunktionen dessen erläutert. Ein weiteres Video, in welchem den künftigen Prüflingen die Arbeit einreichen-Maske nahegebracht wird, folgt.

2.6.1 Inbetriebnahme

- 23. 2. 2014 Das KISDarchiv wird via KISDspaces, KISD-Intranet und E-Mail vorgestellt.
- 24. 2. 2014 Der erste Nutzer lädt (unaufgefordert) eine Arbeit über das Frontend hoch.

MÄRZ 2013

- 3 / 2014 Die ersten Intermediate- und Bachelorarbeiten werden von den Prüflingen in das KISDarchiv eingegeben.

* Schätzwert

3 Altes digitales Archiv (AdA)

3.0 Einleitung

Das Alte digitale Archiv (AdA) stellt den Vorläufer zum KISDarchiv dar. Es wurde im Jahr 2004 von Marian Steinbach entwickelt. Es markiert die erste Anstrengung der digitalen Archivierung von KISD-Prüfungsarbeiten inklusive Metadaten. Dem AdA verdanken wir einen neun Jahre umfassenden Metadatenbestand, welcher als Basis für die Datenbank des KISDarchivs diente.

Das AdA bestand aus drei verknüpften Modulen:

- Die Eingabemaske (»**Frontend**«), mit welcher sich zu den im Gruppenordner abgelegten Dateien Metadaten abrufen und manipulieren ließen.
- Die Ordnerstruktur (»**Gruppenordner**«), in der Prüfungs- und Lehrgebietsdokumentationen sowie sonstige Dokumente (Jahrbücher, Veranstaltungskalender,...) abgelegt wurden.
- Die Meta-Datenbank (»**AMD**«), welche die im Frontend eingegebenen Metadaten in Tabellenform speicherte.

3.1 Frontend

ABB. 13 Eintrags-Listendarstellung im alten digitalen Archiv.

Titel	Status	Werkzeuge	Typ	Erstellt	Modifiziert	Größe	Gruppenordner	Erstellt	Modifiziert
Handbuch der Schullehrkräfte (1984)	aktiv	Handbuch	Handbuch	1984	1984	1000	Handbuch	1984	1984
Handbuch der Schullehrkräfte (1985)	aktiv	Handbuch	Handbuch	1985	1985	1000	Handbuch	1985	1985
Handbuch der Schullehrkräfte (1986)	aktiv	Handbuch	Handbuch	1986	1986	1000	Handbuch	1986	1986
Handbuch der Schullehrkräfte (1987)	aktiv	Handbuch	Handbuch	1987	1987	1000	Handbuch	1987	1987
Handbuch der Schullehrkräfte (1988)	aktiv	Handbuch	Handbuch	1988	1988	1000	Handbuch	1988	1988
Handbuch der Schullehrkräfte (1989)	aktiv	Handbuch	Handbuch	1989	1989	1000	Handbuch	1989	1989
Handbuch der Schullehrkräfte (1990)	aktiv	Handbuch	Handbuch	1990	1990	1000	Handbuch	1990	1990
Handbuch der Schullehrkräfte (1991)	aktiv	Handbuch	Handbuch	1991	1991	1000	Handbuch	1991	1991
Handbuch der Schullehrkräfte (1992)	aktiv	Handbuch	Handbuch	1992	1992	1000	Handbuch	1992	1992
Handbuch der Schullehrkräfte (1993)	aktiv	Handbuch	Handbuch	1993	1993	1000	Handbuch	1993	1993
Handbuch der Schullehrkräfte (1994)	aktiv	Handbuch	Handbuch	1994	1994	1000	Handbuch	1994	1994
Handbuch der Schullehrkräfte (1995)	aktiv	Handbuch	Handbuch	1995	1995	1000	Handbuch	1995	1995
Handbuch der Schullehrkräfte (1996)	aktiv	Handbuch	Handbuch	1996	1996	1000	Handbuch	1996	1996
Handbuch der Schullehrkräfte (1997)	aktiv	Handbuch	Handbuch	1997	1997	1000	Handbuch	1997	1997
Handbuch der Schullehrkräfte (1998)	aktiv	Handbuch	Handbuch	1998	1998	1000	Handbuch	1998	1998
Handbuch der Schullehrkräfte (1999)	aktiv	Handbuch	Handbuch	1999	1999	1000	Handbuch	1999	1999
Handbuch der Schullehrkräfte (2000)	aktiv	Handbuch	Handbuch	2000	2000	1000	Handbuch	2000	2000

Das Frontend des AdA bot diverse rudimentäre Eingabeformulare, die es erlaubten, Dateien in das digitale Archiv einzugeben und zu diesen Dateien Metadaten zu erfassen. In diesem Prozess wurden die Dateien in einer Ordnerstruktur, dem so genannten Gruppenordner, abgelegt, und die physische Position innerhalb dieser Ordnerstruktur innerhalb der Metadaten. Das Frontend verknüpfte also die Alte Metadatenbank (AMD) mit dem Gruppenordner. Überdies erlaubte das Frontend das gezielte Durchsuchen des Metadatenbestands und war damit wichtigstes Recherchetool, um sinnvoll mit dem digitalen Datenbestand zu arbeiten. Die Datenbankoperationen erwiesen sich jedoch mit wachsendem Datenbestand als große Belastung für den Server, sodass Teile des Frontends 2009 effektiv außer Betrieb genommen wurden.

3.2 Gruppenordner

ABB. 14 Darstellung der Dateistruktur auf dem Gruppenordner.

Name	Last Modified	Size	Description
Karens_Directory		-	
Dokumentationen_Projekte	18-Jan-2014 11:06	-	
Dokumentationen_Seminar-Klassenarbeiten	08-Feb-2013 14:06	-	
Dokumentationen_Sonstige	01-Dec-2013 11:29	-	
Verknuepfungskalender_SV_08_WS_11-12	28-Feb-2013 14:55	-	
Bachelor	25-Mar-2013 13:29	-	
Diplom	22-Apr-2013 14:51	-	
Intermediat	28-Mar-2013 18:29	-	
Master	07-Feb-2013 19:06	-	
Yordiplom	22-Apr-2013 13:40	-	
Prüfungen_Listen_1998_bis_heute	04-Sep-2012 11:08	-	
Zusammenfassung_2000-2007	20-Jan-2012 16:15	-	
Sonstige_ETB-Archivierungen	30-Jan-2013 14:15	-	

Der so genannte »Gruppenordner« trägt seinen Namen, da der komplette Archiv-Datenbestand innerhalb derselben Struktur abgelegt war, die auch einigen hundert Intranetgruppen als Dateistruktur diente: Dem KISD-internen Server »HAL«. Unter der Adresse <http://groups.kisd.de/doku-archiv/> war die Ordnerstruktur via Browser einsehbar.

Die Struktur folgte folgender Hierarchie:

1. EBENE	PRÜFUNGSSTUFE	- Diplom
2. EBENE	JAHR	└ 2010
3. EBENE	AUTOR	└└ 12345678_mustermann_max
4. EBENE	PDF-DATEI	└└└ 12345678_ht.pdf
"	ABGABEFORMULAR	└└└ 12345678_ht_note.pdf

Der Gruppenordner war ursprünglich lediglich Speicherort für Dateien gewesen, die aus dem Frontend des AdA referenziert wurden. Hier lagen physische Dateien, zumeist in PDF-Form, welche von den Prüflingen eingereicht worden waren. Für sich betrachtet war diese Sammlung von Dateien wenig nützlich. Aus den Dateinamen war nicht einmal der Titel der Arbeit zu ersehen. Erst im Zusammenspiel mit den zwei anderen Modulen, welche die Verknüpfung der physischen Dateien mit Metadaten erlaubten, entstand ein nützliches Tool.

Mit der teilweisen Außerbetriebnahme des Frontends 2009 fiel dem Gruppenordner die Rolle des primären Archiv-Interfaces zu. Von 2009–12 wurde dieser als von der AMD getrennte Entität betrachtet. Die Verknüpfung zwischen AMD-Metadaten und physischen Dateien, welche lediglich darin bestand, innerhalb der Metadaten einer Arbeit deren Position auf dem Gruppenordner zu notieren, geriet in Vergessenheit. Der Fokus der AG lag nun auf dem Gruppenordner und dessen Übersichtlichkeit gegenüber Besuchern. Im Rahmen des SI »Arche« wurden Anstrengungen unternommen, den Gruppenordner neu zu organisieren. Dies beinhaltete das Verschieben und Umbenennen von Dateien und Ordern. Dies führte zur steigender Diskrepanz zwischen dem durch die AMD abgebildeten Bild des Gruppenordners und den tatsächlichen Positionen der physischen Dateien. Die AMD glich nunmehr einem veraltetem Telefonbuch, dessen Angaben nicht mehr aktuelle sind.

3.3 Alte Metadatenbank (AMD)

Als **Alte Metadatenbank (AMD)** wird die Datenbank bezeichnet, welche die im AdA-Frontend eingegebenen Metadaten fasste.

ABB. 15 Strukturansicht der Tabelle archivag_eingabe, dem Kernstück der AMD.

Field	Typ	Attribute	Null	Standard	Extra
id	INT(4)	unsigned	Yes		auto_incr
pdf_gleid	varchar(255)		Yes	NULL	
anliegen	enum('K', 'Nicht')		Yes	NULL	
titel	varchar(255)		Yes	NULL	
autor	varchar(255)		Yes	NULL	
matrikelnummer	INT(4)	unsigned	Yes	NULL	
typ	enum('Bf', 'Vordr', 'Master', 'Bachelor', 'Dipl', 'Wd', 'Wf', 'Uhr')		Yes	NULL	
arbeit	enum('E', 'N')		Yes	NULL	
betreuung	varchar(255)		Yes	NULL	
lehrgebiet	varchar(5)		Yes	NULL	
jahr	year(4)		Yes	NULL	
sprache	enum('D', 'E', 'W', 'I')		Yes	NULL	
abstract	text		Yes	NULL	
keywords	text		Yes	NULL	
datengroesse	INT(4)	unsigned	Yes	NULL	
pdf_annotierung	varchar(255)		Yes	NULL	
pdf_anzahl	varchar(255)		Yes	NULL	

Die AMD bestand im Wesentlichen aus einer einzelnen Tabelle, die sämtliche Eintrags-Metadaten parallel abbildete. Dies brachte gewisse Vorteile mit sich. Datenbankabfragen ließen sich besonders kurz fassen:

```
SELECT * FROM archivag_eingabe WHERE autor LIKE "%Max Kuwertz%";
```

Alle Einträge aus der Tabelle zu beziehen, deren Autor »Max Kuwertz« heißt, bedurfte lediglich einer Zeile SQL-Code. Die redundante Speicherung in Klartextform barg jedoch die Gefahr diverse Schreibweisen ein und derselben Person und gestaltete die Wartung der Daten unnötig aufwändig.

GENERELLE TABELLE	<pre>ID ; titel ; autor ; lehrgebiet 1 ; Arbeit Eins ; Max Kuwertz ; PT 2 ; Thema Zwei ; Maximilian Kuwertz ; ÖD</pre>
--------------------------	--

Eine gezielte Spezialisierung bot sich an:

TABELLE »EINTRAG«	<pre>EID ; titel ; fAID ; fLID 1 ; Arbeit Eins ; 1 ; 1 2 ; Thema Zwei ; 1 ; 2</pre>
--------------------------	---

TABELLE »AUTOR«	<pre>AID ; vorname ; nachname ; mnr 1 ; Max ; Kuwertz ; 12345678</pre>
------------------------	--

TABELLE »LEHRGEBIET«	<pre>LID ; kuerzel ; name 1 ; PT ; Produktionstechnologie 2 ; ÖD ; Ökologie und Design</pre>
-----------------------------	--

Mehr zur Spezialisierung: Siehe **Posttypen** (S. 114).

4 Vergleichbare Konzepte

4.0 Archibald

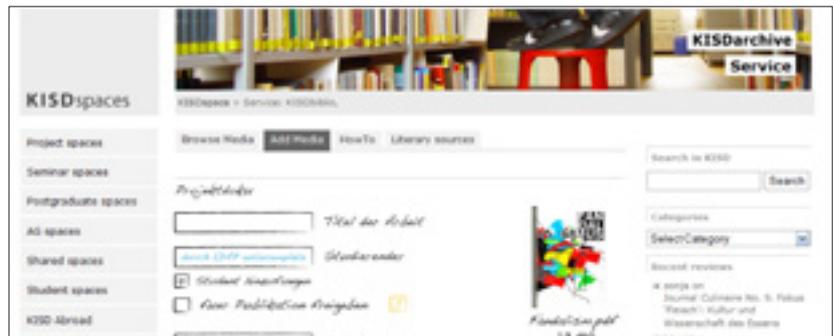
ABB. 16 Auszug aus **Archibald. Konzept für ein digitales Archiv des Fachbereichs Design der Kölner Fachhochschule.**
M. Kunze / P. Neuhaus, 2001



Bereits im Jahr 2001 versuchten sich Michaela Kunze und Peter Neuhaus in einem Vordiplom-Hauptthema unter dem Titel »Archibald« an einem Konzept für ein Intranet-basiertes Archiv für die KISD. Obwohl dieses streckenweise recht konkrete Vorschläge bis hin zu Interface-Mockups enthielt, wurde es nie umgesetzt. Dennoch bietet die Arbeit interessante Einblicke in den Archiv-Betrieb der frühen 2000er-Jahre.

4.1 Blended Studies

ABB. 17 Konzept eines KISDarchivs auf Basis der KISDspaces.
D. Grieshammer, 25.3.2010



Im Rahmen eines gemeinsamen Treffens mit dem Team der Blended Studies stellte David Grieshammer ein Konzept vor, welches eine Implementierung des KISDarchivs auf Basis der KISDspaces skizzierte.

Das Konzept setzte auf eine hohe Verknüpfung zwischen KISDarchiv und diversen Lehrveranstaltungsspaces und orientierte sich darin an KISDbiblio, welches die Referenzierung von Datensätzen in Posts via Shortcode ermöglicht. Diese enge Integration scheint sich vor allem dort anzubieten, wo direkte Bezüge zwischen archivierten Inhalten und Spaces abbildbar sind; in der Dokumentation von Lehrveranstaltungen also. Das Konzept wurde nicht umgesetzt. Die Möglichkeit, Lehrveranstaltungen zu dokumentieren wurde auf andere Weise implementiert.

5 SI »Arche«

5.0 Einleitung

»Ziel ist es, ein Konzept für eine klar strukturierte Überarbeitung des bestehenden Archivs zu entwickeln. Dieses Konzept soll vor allem unter dem Gesichtspunkt der Benutzerfreundlichkeit, Hilfe und Unterstützung für Studierende und Lehrende leisten. Gegenstand der Betrachtung sind sowohl das digitale Archiv als auch das bestehende analoge Archiv.

*Durch die Rekonstruktion der Historie, einer Analyse des Status quo sowie der Erarbeitung von potentiellen Nutzungsmustern und Anforderungen an ein künftiges Archiv wird eine Basis für konzeptionelle Ansätze geschaffen. Ein weiterer Teil des Projektes wird außerdem sein, die zukünftigen Aufgaben der Archiv-AG zu fassen und die Zukunft des »physischen« Archivs, also der Räumlichkeit, zu überdenken und eventuell neu zu definieren. Im Vordergrund steht die Nutzenmaximierung des Archiv für alle Studierenden an der KISD in Form einer nachhaltigen Änderung der bestehenden Strukturen.«**

5.1 Analoges Archiv

ABB. 18 Archiv (R.126)
13.10.2011



ABB. 19 Archiv (R.126)
27.09.2013



* aus KISDarchiv. Rückblick, Einblick, Ausblick 2011, Dokumentation des SI »Arche« im WS 2011/12.

Die Transformation, welche das analoge Archiv im Rahmen des SI »Arche« erfuhr, kommt in zwei im Abstand von zwei Jahren aufgenommenen Fotos zum Ausdruck. Sie ist Zeugnis akribischer Erfahrung des Chaos laut Hesse und war in vielerlei Hinsicht maßgeblich für die Ansprüche an die digitale Inkarnation des Archivs, welche erst Form anzunehmen begann, als der Raum 126 schon zu den saubersten der KISD zählte. Sie verdient außerdem eine gesonderte Dokumentation, weshalb ich mir nicht anmaße, die Geschehnisse der 102 Wochen zwischen den beiden Aufnahmen zu umschreiben.

5.2 Digitales Archiv

ABB. 20 T. Zähres / L. Schraven:
Wireframe; Einstellmaske.
(Dezember 2011)

The image shows a wireframe of a settings mask for 'KISD Archiv'. The form is organized into several sections with colored markers (purple, blue, red) on the left side. The fields include:

- Lehrveranstaltungsnummer (ID):** A text input field.
- Titel der Veranstaltung:** A text input field.
- Name der Dokumental- (Einzel)gruppe:** A text input field.
- Autorin & Matrikelnummer:** A text input field with a red 'X' icon and a link 'Neue Autoren hinzufügen'.
- Semester:** A dropdown menu.
- Keywords:** A text input field with the placeholder 'Begriff, Information, Person, Max Mustermann'.
- E-Mail Adresse der Projektgruppe:** A text input field with a note: 'Projektbetreiber werden per Mail benachrichtigt, dass ein »Projektitem« im Archiv erstellt wurde'.
- Sprache:** A dropdown menu.
- Lehrveranstaltung:** A dropdown menu with a note: 'Wo finde ich die Lehrveranstaltungsnummer?'.
- Projekt:** A dropdown menu with a note: 'Wo finde ich die Lehrveranstaltungsnummer?'.
- Matrikelnummer:** A dropdown menu.
- Additional fields:** A text input field containing 'ind_datenupload' and a checkbox 'Weitere Medien im Archiv verfügbar'.

At the bottom, there are two buttons: 'Arbeit hochladen' and 'Vorschau'.

Ein Konzept für eine neue Implementierung des Online-Archivs auf Basis der KISDspaces wurde während des SI »Arche« von Tim Zähres und Laura Schraven ausgearbeitet. In mehreren Treffen im Rahmen dieses Konzepts wurden Ende 2011 Überlegungen zur Bedienung angestellt und Wireframes entwickelt. In den Wireframes wurden die zwei Grundfunktionen »Arbeit suchen« und »Arbeit einstellen« skizziert, sowie die Möglichkeit, Arbeiten einer Merkliste hinzuzufügen. Dabei ging man von einer Integration in das bestehende Layout der KISDspaces aus, angelehnt an das Tool KISDbiblio mit seiner prominenten Darstellung von Buchcovern.

Die Wireframes erwiesen sich als äußerst hilfreiches Anschauungsmaterial um Funktionalitäten in der Gruppe zu besprechen. Da zu diesem Zeitpunkt noch keine Auseinandersetzung mit dem zugrundeliegenden Datenbestand stattgefunden hatte, konnten die Ansätze jedoch nicht qualifiziert werden.

6 Vorüberlegungen

6.0 Einleitung

Bevor ich mich an die Umsetzung des KISDarchivs begab, stellte ich Überlegungen an, welche die Art der Entwicklung und Erscheinung sowie die technischen Voraussetzungen betrafen. Einige dieser Überlegungen werden hier zur Niederschrift gebracht werden.

6.1 Lehren

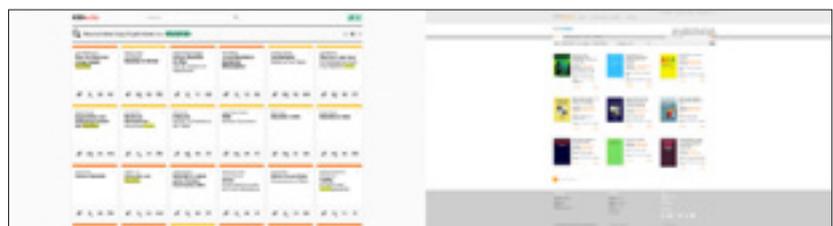
So unterschiedlich die bestehenden Konzepte zum Thema KISDarchiv in ihren Ideen und visuellen Ansätzen auftraten, so hatten sie einiges gemeinsam:

1. Alle Konzepte basieren auf bestehenden Strukturen respektive Interfaces: Das Konzept »Archibald« orientierte sich am KISD-Intranet, die darauf folgenden Konzepte an den KISDspaces. Die auf den KISDspaces basierenden Konzepte orientierten sich darüber hinaus am Tool KISDbiblio.
2. Alle Konzepte waren durchdacht und klar von einander abzugrenzen, sie lieferten Einsichten in die sich verändernden Abläufe an der KISD.
3. Die Konzepte wurden allesamt mehr oder weniger konkret in Wireframes und Mockups visualisiert.
4. Keines der Konzepte wurde umgesetzt.

Der erste Punkt scheint mir der gewichtigste: Die Verknüpfung und damit der Zugang über die KISDspaces erschien mir als einzig sinnvolle Implementierung. Ich erwählte mir KISDbiblio zum Vorbild, um zu evaluieren, ob sich die Vorzüge seiner viel zitierten Darstellung auf das KISDarchiv übertragen ließen.

Ein Blick auf KISDbiblio zeigt die Herausforderungen, die von starren Rahmen ausgehen. Das Layout funktioniert gut ab etwa 1000 Pixel Fensterbreite, sobald das Fenster 1600 Pixel Breite überschreitet, wirkt es etwas verloren:

ABB. 21 Gegenüberstellung von KISDarchiv und KISDbiblio; Viewport-Breite: 2560 px.



Das Element der gut zu differenzierenden Buchcover lässt KISDbiblio in dieser Form funktionieren. Mitunter, etwa bei Taschenbüchern aus dem Suhrkamp Verlag, ist der Titel auf dem lediglich 163 Pixel hohen Cover sogar besser zu erkennen, als der daneben in Textform dargestellte Titel.

ABB. 22 Darstellung von Einträgen innerhalb von KISDbiblio.



Die auf den KISDspaces basierenden Konzepte griffen die Cover-Metapher auf, ohne zu hinterfragen, was im Rahmen des KISDarchivs an ihre Stelle treten sollte. Wo ein Konzept die Generierung der Cover beim Upload andenkt, schlägt das andere gar vor, die Prüflinge eine separate Datei für das Cover einreichen zu lassen. Die Tatsache, dass der komplette Altdatenbestand von über 4000 PDFs auf der Suche nach für Cover geeigneten Seiten durchforstet und diese in ein entsprechendes Format hätten gebracht werden müssen, ließ mich von der Cover-Idee Abstand nehmen.

6.1.1 Das Wireframe-Dilemma

Der dritte Punkt in der obigen Liste sollte sich als gravierend erweisen: Die Wireframes und Photoshop-Mockups wiesen insofern eine erhebliche Schwachstelle auf, als dass sie die Extreme nicht berücksichtigen konnten – Lediglich einige wenige Datensätze, oftmals mit selbsterdachtem »Blindtext« gefüllt, zierten die Layouts. Der Diversität der Daten, welche später durch sie Abbildung finden sollten, berücksichtigten die Wireframes nicht. Dabei handelt es sich um ein grundlegendes Problem vieler modellhafter Ansätze. Dies wurde in der Auseinandersetzung mit den vorliegenden Daten offenbar. Es erscheint mir im Falle dieses Projekts umso gravierender, da das System nicht von Grund auf konzipiert wurde, sondern auf Basis bestehender Strukturen, denen es Rechnung tragen sollte.

6.1.2 Flexibilität durch Nonlinearität

In der Bindung an ein bestehendes Interface müssen wir nolens volens mit Einbußen in der Flexibilität unseres, einem neuen Zweck verschriebenen Interfaces rechnen. Ich vermute, dass sich besagte Einbußen umso stärker bemerkbar machen, je größer die Unterschiede zwischen den im originalen Interface dargestellten Daten und den Neudaten sind.

Das KISDarchiv sollte auf mehreren Ebenen flexibel gehalten sein:

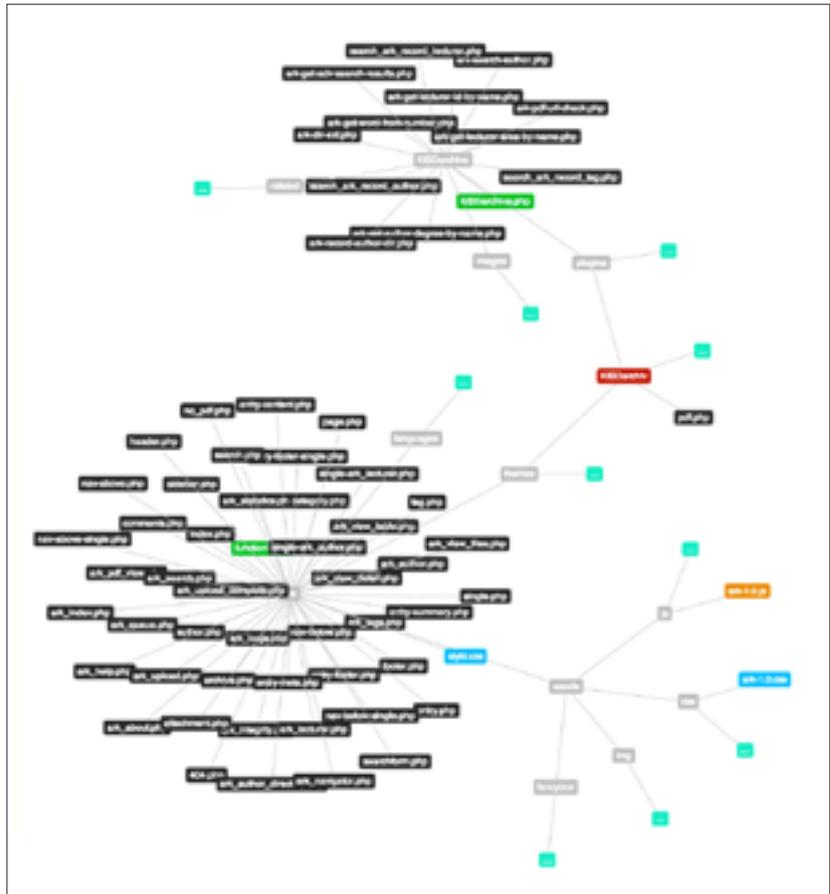
- Flexible Entwicklung: Die Konzeption des gesamten Systems sollte keinen Wireframes oder Mockups folgen, sondern seine Form, seine Module und Templates lediglich aus dem Umgang mit der Datenbasis ableiten.
- Flexibler Rahmen: Das zugrundeliegende Raster sollte sich je nach zur Verfügung stehender Fläche flexibel anpassen, und somit das analoge Archiv als Musterbeispiel optimierter Flächennutzung referenzieren.
- Flexible Interfaces: Diverse Module und Repräsentationen (»Ansichten«) sollten den Nutzern die Erschließung der Daten auf vielseitige Weise ermöglichen und ihnen keine starren Arbeitsabläufe auferlegen.

6.2 Technische Voraussetzungen

6.2.1 Plattform

ABB. 23 KISDarchiv-Dateistruktur basierend auf WordPress.

rot: Grundordner,
 grau: Unterordner,
 schwarz: (Sub-)Templates,
 blau: CSS-Dateien,
 orange: JavaScript-Dateien,
 grün: Plugin & Theme-Files
 türkis: ausgeblendete Dateien.



Die Wahl von WordPress als unterliegendem System lag auf der Hand: Einerseits operierten die KISDspaces ebenfalls auf WordPress, andererseits gab es schlichtweg kein System mit einer vergleichbaren Nutzer- oder Entwicklerbasis auf dem ein Projekt dieser Größenordnung denkbar gewesen wäre. Dass WordPress bereits seit 2005 mein Quasi-CMS der Wahl war, erleichterte die Entscheidungsfindung. Das KISDarchiv wurde von vorneherein so konzipiert, dass es sich nahtlos in die WordPress-Architektur einfügte, ohne Veränderungen an dieser vorauszusetzen. Dies war erforderlich, um in einer komplexen Umgebung wie den KISDspaces keine Konflikte mit anderen Installationen oder Plugins zu erzeugen und den Wartungsaufwand bei WordPress-Updates zu minimieren.

6.2.2 Framework

Die Fahl eines geeigneten CSS-Frameworks fiel auf das von Nathan Smith entwickelte »Unsemantic«. Ausschlaggebend für die Wahl war die gute Dokumentation, was »Unsemantic« von seinen Konkurrenten wie Joni Korpis »frameless« und dem »Golden Grid System« abhob.

LINK unsemantic.com

6.3 Konventionen

Komplexe Projekte verlangen nach rigorisen Konventionen und einer präzisen Syntax. Anders ist es nicht möglich, den Überblick über die vielen Dateien, Datenbanktabellen, Funktionen, Variablen und Links zu behalten.

6.3.1 Sprache

Alle Elemente des Frontends, Links, Tooltips und sonstige Interface-Bestandteile sind in deutscher Sprache beschriftet. Tutorial-Videos sind in deutscher Sprache gehalten. Englische Sprache taucht nur da auf, wo sie seitens der KISD explizit verwendet wird, etwa in der Bezeichnung von Teilprüfungen (»Final Thesis«). Zudem werden Ordner, Dateien, Programmbestandteile, Funktionen und Variablen in englischer Sprache benannt.

6.3.2 Nomenklatur

Eine sorgfältige Benennung aller innerhalb der Software agierenden Bestandteile ist notwendig. Sie hilft dabei, KISDarchiv-eigene Komponenten zu identifizieren, grenzt diese sowohl von den WordPress eigenen Bestandteilen ab, als auch von unbekanntem, etwa durch externe Plugins zukünftig zugeführten Bezeichnern. Neben dieser präventiven Konfliktprävention helfen eindeutige Bezeichner dem systemfremden Operator, sich in diesem zurechtzufinden.

Programmbestandteile

Das Präfix **ark_** wird Ordnern, Dateien, Funktionen, Cookies, Variablen, CSS-Klassen und IDs, Posttypen und sonstigen Entitäten vorangestellt, wenn diese in einem Kontext auftreten, innerhalb dessen sie ungepräftigt einer gewissen Verwechslungsgefahr unterliegen

Allen Variablen, Posttypen, Funktionen, Cookies, sonstigen Bezeichnern, Ordnern und Dateien ist das Präfix **ark_** vorangestellt, wenn der Kontext dies erfordert, um diese eindeutig als dem KISDarchiv zugehörig zu kennzeichnen.

Variablen eines Posttyps setzen sich wie folgt zusammen:

SYNTAX `[Präfix]_[Posttyp]_[Variable]`

BEISPIEL Autorenfeld im Posttyp Eintrag: **ark_record_author**.

Funktionen werden so benannt, dass aus dem Namen ersichtlich ist, was die Funktion zu leisten imstande ist.

BEISPIEL Die Funktion **ark_get_author_name()** bekommt eine ID oder ein Array von IDs übergeben und gibt die assoziierten Namen zurück.

Darüber hinaus wird auf Großschreibung verzichtet, Ausnahme bildet JavaScript, wo eine Notation ohne Unterstriche Anwendung findet (**arkRecordSubexam** statt **ark_record_subexam**).

Die Begriffe **Arbeit**, **Eintrag (vom Posttyp `ark_record`)** werden gleichbedeutend betrachtet. Ist nur von einem Eintrag die Rede, so meint das einen Eintrag vom Typ **`ark_record`**. Einträge anderen Typs werden als solche bezeichnet.

6.3.3 Dateinamen/-pfade

PDFs werden nach einem bestimmten Muster in der Dateistruktur verortet:

SYNTAX `[kisdarchiv_url]/ark-upload/files/[pruefungsstufe]/[jahr]/[nachname_vorname]/[teilpruefung_titel.pdf]`

BEISPIEL `http://spaces.kisd.de/kisdarchiv/ark-upload/files/bachelor/2013/wischmeier_timo/ht_plastik-through-the-gyre.pdf`

Siehe `ark_generate_pdf_url()` (S. 140).

6.3.4 Permalinks

Permalinks werden in Abhängigkeit vom Posttyp generiert.

`ark_record`

Der Permalink für Einträge des Typs **`ark_record`** setzt sich aus den Metadaten nach folgendem Muster zusammen:

SYNTAX `[kisdarchiv_url]/eintrag/[pruefungsstufe]/[jahr]/[autoren]/[teilpruefung]_[titel]-[ID]/`

Liegen die entsprechenden Metadaten nicht vor, werden diese im Permalink durch generische Bezeichner ersetzt. Bei mehreren Autoren werden die Vornamen in Kurzform dargestellt.

BEISPIEL `http://spaces.kisd.de/kisdarchiv/eintrag/in/2012/a-j-katona-m-j-nimmesgern/ht_abwesenheit~22470/`

`ark_author`

Der **`ark_author`**-Permalink führt den vollständigen Namen des Autoren.

SYNTAX `[kisdarchiv_url]/autor/[vollstaendiger_name]/`

BEISPIEL `http://spaces.kisd.de/kisdarchiv/autor/andreas-johannes-katona/`

`ark_lecturer`

Einträge vom Typ **`ark_lecturer`** führen den akademischen Titel und Namen.

SYNTAX `[kisdarchiv_url]/dozent/[titel_und_vollstaendiger_name]/`

BEISPIEL `http://spaces.kisd.de/kisdarchiv/dozent/prof-dr-michael-erlhoff/`

7 Schaffung der Datenstruktur

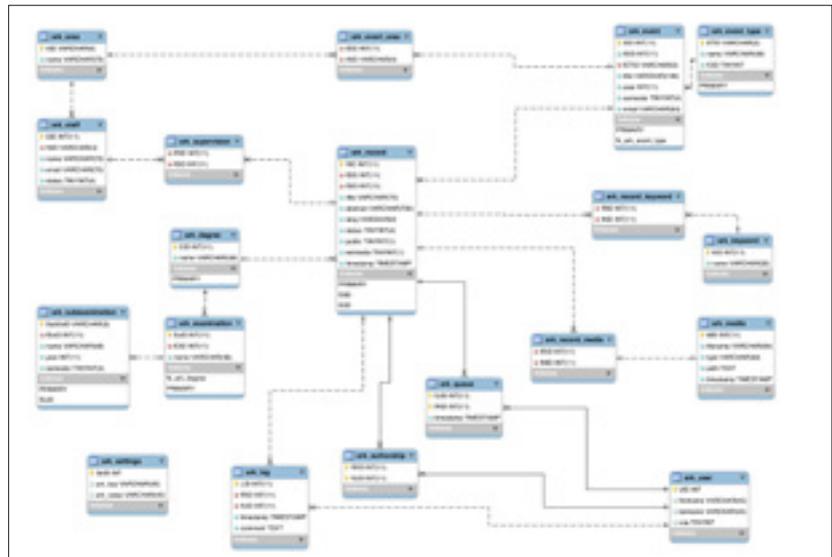
7.0 Einleitung

Der Beginn der Arbeit am Projekt KISDarchiv markierte in vielerlei Hinsicht einen kritischen Punkt. Es mussten Entscheidungen bezüglich der Datenstrukturen gefällt werden. Da diese Strukturen unmittelbar nach ihrer Schaffung tausende Datensätze aus dem Datenbestand des AdA behausen würden, war nicht daran zu denken, diese nach dem Import der Daten noch einmal größeren Änderungen zu unterziehen. Die Struktur wollte also wohl überlegt sein.

7.1 Datenbankmodell

Bereits ein Jahr vor Beginn der Arbeiten am neuen digitalen Archiv hatte ich im Rahmen des SI »Arche« Überlegungen bezüglich der Beschaffenheit einer Datenbankstruktur für das neue digitale Archiv angestellt.

ABB. 24 Frühes Datenbankmodell zur Evaluation der Herangehensweise (Dez. 2011)



Die Formulierung eines relationalen Datenbankmodells, seinerzeit noch ohne WordPress-Integration, hatte ich bereits im Dezember 2011 abgeschlossen. Im Zentrum der Aufmerksamkeit stand damals schon die Entität **ark_record**, welche 14 Monate später in den gleichnamigen Posttyp übergehen sollte. Von der Idee, eine komplett von WordPress autarke Datenbankstruktur zu konzipieren, kam ich allerdings schnell ab – Eine eigenständige Datenbank hätte zwar hocheffiziente Abfragen ermöglicht und den Daten eine Struktur auferlegt, die in sich selbst bereits Fehleingaben großteils ausgeschlossen hätte. Andererseits wäre es ebendieser starre Rahmen gewesen, der bei kleinen strukturellen Änderungen im System grundlegende Manipulationen an der Datenbankstruktur erfordert hätte. Ich entschied mich also dazu, die WordPress-eigenen Custom Post Types zu nutzen, um eine Handvoll flexibler Posttypen zu konzipieren, die dem System aufgrund ihrer parallelen Datenspeicherung weniger starre Strukturen auferlegten. Dies würde zwar vereinzelt komplexere Datenbankabfragen bedeuten, was ich allerdings angesichts der gewonnenen Flexibilität gerne in Kauf nahm.

7.2 Neue Metadatenbank (NMD)

ABB. 25 Neue Metadatenbank (NMD) auf Google Drive

Autor/in	Autor/in	Prüfer/in	Prüfer/in	Titel	Kurzinhaltangabe	Spr. Beitraggeber	Prüfer/in	Lehrgebiet	Bemerkung
Infaf	Maria	Bachelor 2013 NT 2	Food Trends	Die Welt infaf sich mit den Food Trends auf dem Aufbruchsschritt. Aktuelle und vergangene Food Trends werden analysiert und anhand emotionaler Kriterien bewertet.	DE	Food Trend, Aufbruchsschritt	Prof. Sascha	TI, Typographie und Layout	
Infaf	Maria	Bachelor 2013 NT 1	Marken	Markenungen transformieren ein Produkt, die Produktgeschichte verändert sich dadurch. Die Markenung ist eine Identität, welche die Nutzer identifizieren soll.	DE	Markenungen, Prof. Hermann	DD (Design und Design)		
Auf der Meise	Oliver	Bachelor 2013 NT 2	Neu - Strängen	Verarbeitungsverfahren (eigener Vektor: Annot & Co. GmbH)	DE	Annot, Co. GmbH, Prof. Hermann	PF (Produktionstechnologie)	Nur in der digitalen Welt möglich	
Auf der Meise	Oliver	Bachelor 2013 NT 1	Goldberg	Goldberg im Prof. Hermann, Designer	DE	Goldberg, Prof. Hermann	TI, Typographie und Layout	Nur in der	

Die neue Metadatenbank (»NMD«) wurde Ende April 2012 als Übergangslösung geschaffen, um der Archiv AG während der andauernden Arbeiten am KISDarchiv die Möglichkeit zu bieten, Metadaten zu Dateien zu erfassen, welche nach wie vor auf dem Gruppenordner abgelegt wurden.

Dies erschien erforderlich, da der Zugriff auf die Alte Metadatenbank (AMD) nicht mehr funktional war, der Zugriff auf den Gruppenordner – schreiben-derweise seitens der AG und lesen-derweise seitens der Studierendenschaft – jedoch weiterhin gegeben war. Somit bestand ein Anreiz, weiterhin Dateien auf diesen zu laden.

Zum Zweck der kontinuierlichen Metadatenerfassung wurde ein Tabellendokument auf der Plattform Google Drive (ehemals: Google Docs) angelegt. Dieses konnte entweder über eine der Tabelle vorgeschalteten Eingabemaske oder via Direktzugriff auf die Tabelle selbst beschrieben werden.

Die auf diese Weise erfassten Daten würden mir im weiteren Entwicklungsverlauf als Ausgangspunkt dienen, um seit Aussetzen der AMD abgelegte Dateien mit Metadaten zu versehen.

Die NMD wurde von der Archiv-AG in den Jahren 2012/13 mit 550 Datensätzen befüllt, welche im Rahmen von Datenaufbereitung und -import herangezogen werden konnten.

7.3 Entwicklung von Posttypen

Die Posttypen hatte ich bereits im Rahmen der Entwicklung des ersten Datenbank-Modells im Dezember 2011 formuliert. Folgende Posttypen adaptierte ich aus dem ER/RD-Modell auf WordPress:

ARK_RECORD	Eintrag
ARK_AUTHOR	Autor/in
ARK_AREA	Lehrgebiet
ARK_LECTURER	Prüfer/in (ehemals: Lehrende/r)
ARK_COURSE	Veranstaltung (ehemaliger Posttyp)

Siehe **Posttypen** (S. 114).

8 Backend-Anpassung

8.0 Einleitung

Das Backend stellte die erste Baustelle des KISDarchivs dar. Lange bevor daran zu denken war, sich dem Frontend zu widmen, mussten hier die Grundlagen geschaffen werden, um mit den Daten agieren zu können. Das WordPress-Backend bot eine solide Basis. Den potenziellen Nutzern dürfte der Umgang mit WordPress aus der Arbeit mit den KISDspaces bereits vertraut sein. Darum sollten sich alle alle Eingriffe organisch in die bekannte Erfahrung des Backends einfügen.

Die korrekte Abbildung der KISDarchiv-spezifischen Posttypen in diversen Kontexten stand dabei im Fokus. Diese mussten den kompletten Datenbestand so übersichtlich wie möglich darstellen können, bevor die Entwicklung des Frontends in Angriff genommen werden konnte. Mit der Entwicklung des Frontends und dessen Modulen würde sich im weiteren Verlauf abzeichnen, inwieweit Funktionalitäten und Abbildungen des Frontends im Backend repliziert werden mussten, um diese auf die Bedürfnisse der Archiv-AG zuschneiden zu können.

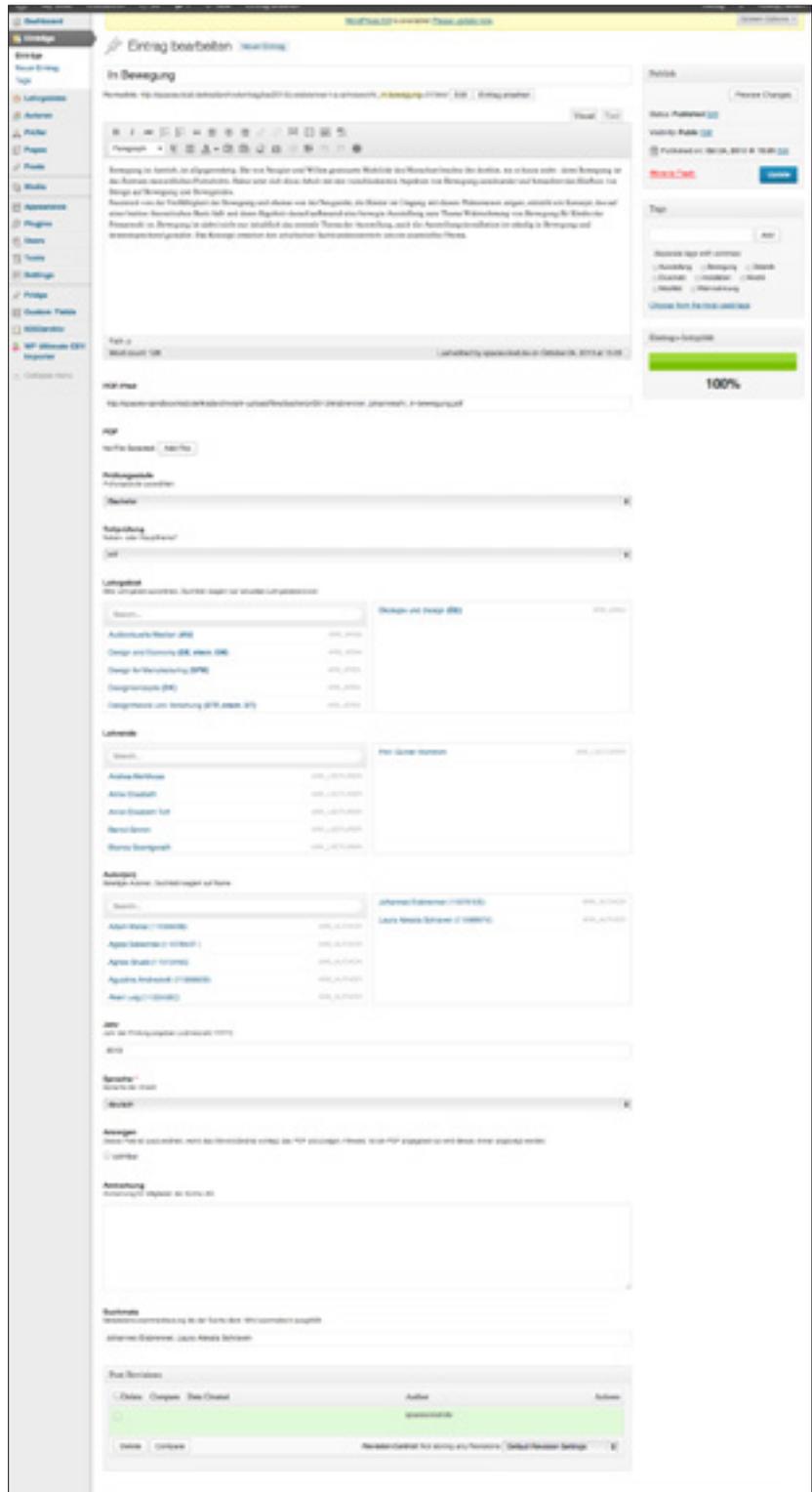
Letztendlich sollten die hier vorgestellten anfänglichen Erweiterungen des Backends als ausreichend bewertet werden. Die frontendseitigen Module für den Umgang mit dem Datenbestand konnten aufgrund ihrer maßgeschneiderten Format überzeugen und sollten in der Datenpflege flexiblere Werkzeuge darstellen, als es ein modifiziertes WordPress-Backend sein könnte.

Im Folgenden werden die diversen Backend-Kontexte beschrieben, innerhalb derer Modifikationen respektive Erweiterungen implementiert wurden.

8.1 Eintrags-Einzelansichten

ABB. 26 Eintrags-Eingabemaske im Backend.

vgl. Arbeit einreichen-Maske im Frontend, Seite 98



Für die korrekte Abbildung der Metadaten innerhalb der diversen Einzelansichten der KISDarchiv-spezifischen Posttypen nutzte ich das WordPress-Plugin »Advanced Custom Fields« (ACF) von Elliot Condon. Dieses ermöglichte es mir, die WordPress-eigene Eingabemaske um Formularelemente zu erweitern, welche diverse Formen annehmen können. Die einzelnen Formularfelder korrespondieren dabei mit WordPress Custom Fields, also den Metadaten-

feldern der Posttypen. Gegenüber den Formularfeldern, wie sie in der später implementierten Frontend-Eingabemaske auf der Arbeit einreichen-Seite Verwendung finden sollten, verzichtete ich an dieser Stelle auf eine strenge Validierung der Felder. Dies hat den Grund, dass zwischen vier- und fünftausend Datensätze ins System eingespeist werden würden, die vorbei an den Eingabemethoden des Backends direkt via CSV-Import ihren Weg in die Datenbank finden würden. Diesen Daten nachträglich eine Validierung auferlegen zu wollen, wäre keine viable Lösung gewesen.

Innerhalb der Lehrenden- und Autoren-Einzelansichten wurden überdies die Titel-Eingabefelder gesperrt, da diese Posttypen die Eintragstitel aus diversen Metadatenfeldern zusammensetzen.

8.2 Eintrags-Listenansichten

ABB. 27 Eintrags-Listenansicht des Posttyps `ark_lecturer`. (Ausschnitt)

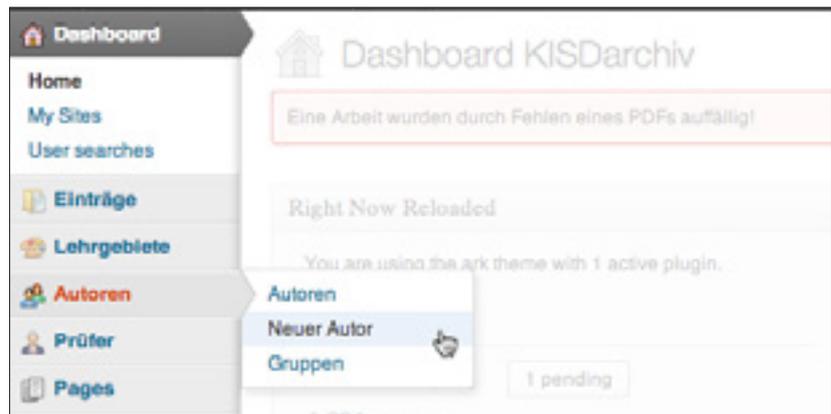
ID	Title	First Name	Last Name	Nickname	Username
78	Prof. Ryan Bartholdy	Prof.	Ryan	Bartholdy	ry
80	Prof. Dr. Ute Brandes	Prof. Dr.	Ute	Brandes	brandes
108	Prof. Dr. Michael Ehrhoff	Prof. Dr.	Michael	Ehrhoff	ehhoff
109	Prof. Michael Gels	Prof.	Michael	Gels	gels

In den diversen Eintrags-Listenansichten wurde das Plugin »Admin Columns« von Codepress eingesetzt. Dieses ermöglicht der Erweiterung der standardmäßig vorhandenen Tabellenspalten um selbstdefinierte Spalten, welche sich aus Metadatenfeldern speisen. Dies erlaubt eine weitaus bessere Übersicht und Dichte in den abgebildeten Daten. Die Sortierung der Einträge eines Typs nach diversen Metadaten erlaubt das Aufspüren unvollständiger Datensätze.

Einige WordPress-Standardfunktionen, welche das massenhafte (»Bulk«) respektive Schnell-Editieren von Einträgen in der Listenansicht erlauben, wurden unterdessen deaktiviert, da diese Funktionen dem gewissenhaften Pflegen von Datensätzen nicht zuträglich sind und diverse komplexe Prozesse, welche sich beim Editieren in der Einzelansicht ergeben, innerhalb der Bulk- und Quick-Edit-Vorgänge nicht ausgeführt würden.

8.3 Navigation

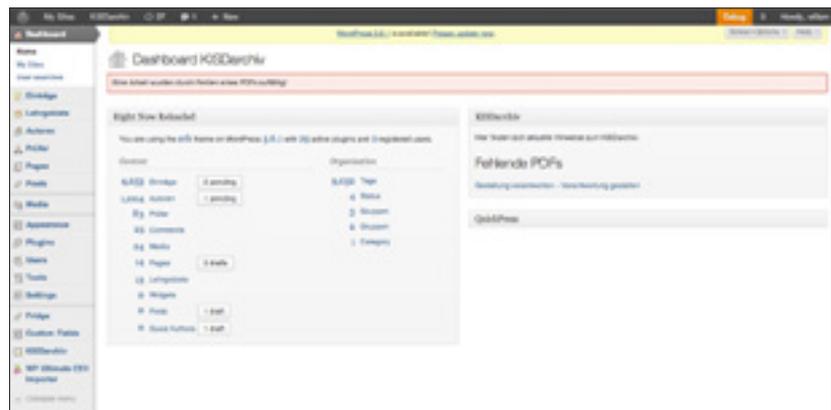
ABB. 28 Backend; Navigation mit KISDarchiv-Posttypen



Die Navigation wurde um Einträge für die KISDarchiv-spezifischen Posttypen erweitert. Für den Betrieb des KISDarchivs unnötige Punkte wurden entfernt.

8.4 Dashboard

ABB. 29 Modifiziertes WordPress-Dashboard im Backend (Administratoren-Ansicht).



Das WordPress-eigene Dashboard wurde um ein Widget erweitert, welches Arbeiten auflistet, die von Nutzern aufgerufen wurden und unerwarteter Weise über keine PDF-Datei verfügten.

Außerdem zeigt das Widget »Right Now Reloaded« (links im Bild) an, wann immer neue Einträge und Autoren über das Frontend eingegeben wurden. Darüber können AG-Mitglieder mit einem Klick die zu zu prüfenden Einträge überblicken und diese mit einem weiteren Klick freischalten.

9 Datenimport

9.0 Einleitung

Eine Herausforderung bestand darin, den recht fragmentierten Altdatenbestand aus diversen Datenquellen in ein mit den neu geschaffenen Posttypen kompatibles Format zu überführen. Dies bedeutete unter anderem die Aufspaltung der redundant gespeicherten Autoren-, Prüfer- und Lehrgebietsdaten in eigene Posttypen, welche im Vorhinein eingelesen werden mussten, um anschließend in den Eintragsdaten referenziert werden zu können.

Der Importprozess gestaltete sich aufgrund der diversen Quellen und Überschneidungen respektive Inkonsistenzen zwischen denselben recht umfangreich. Er wird an dieser Stelle in chronologischer Folge dargestellt.

9.1 Datenquellen

Folgende Daten lagen vor:

- **Alter digitaler Metadatenbestand / Alte Metadatenbank (AMD)**
Der Metadatenbestand aus der AMD umfasste etwa 9000 Einträge, welche Dokumente auf dem Gruppenordner referenzierten.
- **Neuer digitaler Metadatenbestand / Neue Metadatenbank (NMD)**
Der neue Metadatenbestand der NMD umfasste 550 Einträge welche nach Ausserbetriebnahme des AdA auf den Gruppenordner geladene PDF-Dokumente referenzierten.
- **Analoger Metadatenbestand / Archiv-AG-Ordner**
Der analoge Metadatenbestand bestand aus mehr als 800 zu großen Teilen handschriftlich ausgefüllten Abgabeformularen (»Offline Notes«), welche von der AG digitalisiert worden waren und teilweise auf dem Gruppenordner vorhandene Dokumente referenzierten.
- **»Externer« digitaler Metadatenbestand / Drittquellen**
Die Veranstaltungs- und Studierendendatenbank des Intranets, diverse KISD-eigene Datenbanken, Prüfungslisten und quantitative Aufstellungen wurden ergänzend hinzugezogen.
- **Digitaler Dokument-Datenbestand / Gruppenordner**
Der Dokument-Datenbestand basierte auf allen eingereichten Dokumenten, hauptsächlich also PDF-Dateien und teilweise digitalen Abgabeformularen (»Notes«), welche für das Nachtragen von Metadaten herangezogen werden konnten. Der Bestand umfasste etwa 70 Gigabyte. Als problematisch erwies sich hier, dass die menschenlesbare Dateistruktur von der AG manuell reorganisiert und in diesem Prozess Dateien und Ordner verschoben und umbenannt worden waren. Dies erhöhte die Übersicht der Struktur bei direktem Zugriff, hatte jedoch zur Folge, dass große Teile des Alten Metadatenbestandes nun Pfade referenzierten, die in dieser Form nicht mehr auf dem Gruppenordner abgebildet wurden.

Führen wir uns an dieser Stelle vor Augen, was diese verschiedenen Datenquellen bedeuteten, respektive in welchem Zusammenhang sie zu einander standen. Wir haben es mit vier Metadaten-Beständen zu tun, welche in diverser Form Metadaten zu physischen Dateien auf dem Gruppenordner enthielten. Es war davon auszugehen, dass die diversen Metadatenbestände Inkonsistenzen und Überschneidungen aufwiesen.

Ausschlaggebend für den künftigen Datenbestand war dabei, alle physisch vorliegenden Dateien mit qualitativen Metadaten zu versehen. Um dies zu erreichen, wurden die auf dem Gruppenordner vorliegenden Dateien in diverse Klassen unterteilt (basierend auf den in den Metadaten angegebenen PDF-Pfaden):

- Dateien, welche in der AMD referenziert wurden.
- Dateien, welche in der AMD referenziert wurden, jedoch verschoben wurden.
- Dateien, welche nachträglich auf den Server geladen worden waren und in der Neuen Metadatenbank (NMD) erfasst worden waren.
- Dateien, welche nachträglich auf den Server geladen worden waren und über keinerlei Metadaten verfügten.

Es galt also, die Metadaten erst einmal in Gänze in Einträge innerhalb der KISDarchiv-Datenbank zu übertragen und anschließend die Redundanzen innerhalb der Datensätze zu eliminieren und per Abgleich mit dem physischen Datenbestand Metadaten für jene Dateien zu generieren, die noch über keine Metadaten verfügten. Eine weitere Konsolidierung aller nun im KISDarchiv vorhandener Metadaten würde dann den Abgleich der verschobene Dateien referenzierenden Einträge mit den neu generierten Einträgen ermöglichen.

9.2 Mapping und Datenaufbereitung

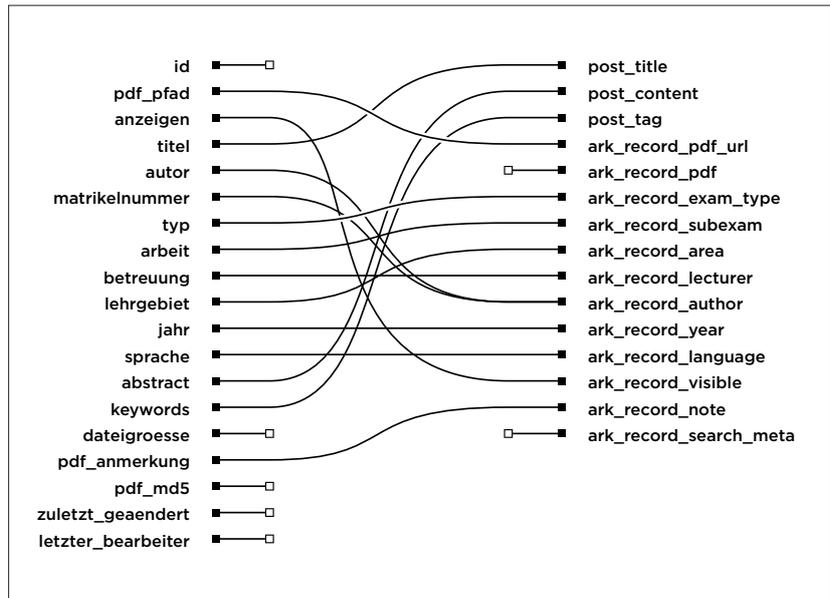
Das Mapping bildete den ersten Schritt bei der Importierung der Altdaten. Beim Mapping wurde die Struktur der AMD herangezogen, und eine Abbildung der bestehenden Datenbankfelder auf ihre Entsprechungen innerhalb der neu geschaffenen Posttypen zu realisieren.

Die im Mapping-Prozess inhärente Datenaufbereitung diente dem Zweck, die AMD-Daten in ein Format zu überführen, welches den Feld-Vorgaben innerhalb der Posttypen entspricht.

Praktisch gestaltete sich das Mapping und die Datenaufbereitung so, dass im ersten Schritt die gesamte AMD in Tabellenform exportiert wurde, um in dieser Form die Formatanpassungen und Spaltenzuweisungen der Datensätze vorzunehmen. Anschließend wurde der so umformatierte Datenbestand über das WordPress Plugin »WP Ultimate CSV Importer« via Stapelverarbeitung in Posts entsprechenden Posttyps eingelesen. Dieser Schritt erforderte besondere Sorgfalt, da der CSV-Import Daten ohne jegliche Validierung in die Datenbank übernahm. Aufgrund des hohen Fehlerpotenzials einer solchen Methode wurde letztendlich keine Möglichkeit implementiert, weiterhin Posts auf diese Weise in den Datenbestand einzulesen.

9.2.1 Mapping

ABB. 30 Mapping der Datenbankfelder der AMD auf KISDarchiv-Posttypen respektive Datenfelder.



Die folgende Tabelle bildet die Struktur der AMD ab. In der linken Spalte stehen die Feldnamen der AMD, darauf folgt deren Datentyp. Die dritte Spalte zeigt den Posttyp, welcher die Werte im KISDarchiv abbildet und die vierte und letzte Spalte schließlich die Variablen innerhalb des betreffenden Posttypen, die den Wert des AMD-Felds übernehmen.

FELD (AMD)	DATENTYP	POSTTYP (KA)	MAPPING / FELD (KA)
ID	int(10)	ark_record	ark_record_legacy_id
—	—	ark_record	ark_record_type
PDF_PFAD	varchar(255)	ark_record	ark_record_pdf_url
ANZEIGEN	enum('ja','nein')	ark_record	ark_record_visible
TITEL	varchar(255)	ark_record	post_title
AUTOR	varchar(255)	ark_author	ark_author_firstname, ark_author_lastname
MATRIKELNUMMER	int(10)	ark_author	ark_author_mnr
TYP	enum('dipl','[...]')	ark_record	ark_record_exam_type
ARBEIT	enum('ht','nt')	ark_record	ark_record_subexam
BETREUUNG	varchar(255)	ark_lecturer	ark_lecturer_firstname, ark_lecturer_lastname
LEHRGEBIET	varchar(5)	ark_area	ark_area_ID
JAHR	year(4)	ark_record	ark_record_year
SPRACHE	enum('de','en','[...]')	ark_record	ark_record_language
ABSTRACT	text	ark_record	post_content
KEYWORDS	text	ark_record	post_tag
DATEIGROESSE	int(10)	—	—
PDF_ANMERKUNG	varchar(255)	ark_record	ark_record_note
PDF_MD5	varchar(32)	—	—
ZULETZT_GEAENDERT	datetime	—	—
LETZTER_BEARBEITER	varchar(12)	—	—

Im Folgenden betrachten wir das Mapping Feld für Feld. Es gilt das Muster [AMD-Feldname] > [KISDarchiv-Feldname].

id › ark_record_legacy_id

Die in der AMD geführte ID wurde in einem entsprechendem Feld mitgeführt, um importierten Einträgen im KISDarchiv einwandfrei AMD-Einträge zuordnen zu können. Nach erfolgreichem Abschluss des Datenimports wurde die Anzeige der Legacy ID im Backend ausgesetzt. Diese kann allerdings weiterhin ausgelesen werden, aus dem Frontend etwa via

```
<?php echo get_post_meta( $post->ID, 'ark_record_legacy_id', TRUE ); ?>
```

— › ark_record_type

Da die AMD Prüfungsarbeiten und Lehrveranstaltungsdokumentationen nicht unterschied, wurde an dieser Stelle ein neues Feld geschaffen, welches diese Unterscheidung abbilden konnte. Die vorhandenen Datensätzen wurden in diesem Zug als 'pru' respektive 'pro' redeclariert. Dieses Feld wurde allerdings mit Entfernung sämtlicher LV-Dokumentationen im Rahmen des Streamlining-Prozesses ausgesetzt und spielt im derzeitigen Archiv keine Rolle mehr.

pdf_pfad › ark_record_pdf_url

Das Feld **ark_record_pdf_url** übernahm sämtliche in der AMD hinterlegten PDF-Pfade. Diese wurden im Rahmen des Importprozesses von relativen auf absolute Pfade umgeschrieben. Dieser Prozess sollte im Laufe der Entwicklung noch einige Male wiederholt werden (erstmal beim Umzug auf die Sandbox, ein weiteres Mal beim Umzug auf das Live-System, ein drittes Mal beim Umzug sämtlicher PDFs vom Gruppenordner auf den Spaces-Server). Bei dieser, ersten Manipulation sämtlicher Pfade wurde bereits teilweise der manuell durch die Archiv AG veränderten Gruppenordnerstruktur Rechnung getragen, um Inkonsistenzen zwischen AMD und Gruppenordner zu reduzieren. Dadurch konnte der späterere Korrekturaufwand etwas reduziert werden.

anzeigen › ark_record_visible

Das Anzeigen-Flag, welches über die Darstellungsberechtigung von PDFs im Frontend Auskunft gibt, wurde unverändert übernommen. Dabei wurden die hinterlegten Werte ('ja' / 'nein') in boolesche Logik (TRUE/FALSE) übersetzt.

titel › post_title

Das AMD-Feld **titel** wurde auf das WordPress-eigene Titelfeld des Post-typs **ark_record** übertragen. Die im Titelfeld geführten Sonderzeichen und Umlaute wurden für den reibungslosen Import in ihre HTML-Entsprechungen übersetzt (Der »WP Ultimate CSV Importer« interpretierte seinerzeit keine Unicode-Zeichen). Nach erfolgtem Import wurden die Sonderzeichen via Datenbank-Query wieder in Unicode übersetzt.

autor, matrikelnummer › **ark_record_author**

Während in der AMD die Autorenschaft in Einträgen anhand der Autorennamen und -Matrikelnummern gefasst wurde, erforderte die Aufspaltung der Autoren in einen eigenen Posttyp **ark_author** an dieser Stelle die Konsolidierung zweier Felder. Eine separate Tabelle wurde angelegt, in welcher die Namen und Matrikelnummern aller in der AMD erfassten Autoren zusammengetragen wurden. Diese wurden wiederum auf die Felder des Datentyps **ark_author** gemappt und separat in WordPress importiert. Die im Verlauf des Imports von WordPress vergebenen IDs der Einträge des **ark_author**-Posttyps wurden daraufhin als Referenz in die Eintragstabelle gespeist, sodass diese anstelle von Namen und Matrikelnummern lediglich numerische Referenzen auf bereits im System vorhandene Autoren enthielt.

typ › **ark_record_exam_type**

Das Feld **typ** enthielt die Prüfungsstufen-Bezeichner, welche ohne größere Veränderungen auf ihre Entsprechung innerhalb des **ark_record**-Posttyps übertragen werden konnten.

arbeit › **ark_record_subexam_type**

Das Feld **arbeit** unterschied in der AMD lediglich zwischen Hauptthema ('ht') und Nebenthema ('nt'). Hier wurde eine feingliedrigere Trennung ('ht', 'nt', 'nt1', 'nt2') eingeführt. Dies würde später eine feinere Differenzierung in der Anzeige von Einträgen erlauben. Um die feingliedrigere Unterscheidung auf den bestehenden Datenbestand anwenden zu können, wurden die PDF-URLs herangezogen, aus welchen sich oftmals ablesen ließ, ob es sich etwa bei einer vormals als 'nt' deklarierten Diplomarbeit um ein NT 1 oder 2 handelte.

betreuung › **ark_record_lecturer**

Die Aufbereitung des Felds **betreuung** ging ähnlich vonstatten wie die Konsolidierung der Autorenfelder – Da die Lehrenden in einen separaten Datentyp gekapselt wurden, war eine Extrahierung aller in diesem Feld geführten Namen in eigene Datensätze vonnöten, um diese an dieser Stelle numerisch referenzieren zu können. Da die Zuordnung der Lehrenden im AdA über ein Textfeld geschehen war, wies dieses Feld erhebliche Inkonsistenzen auf. So fanden sich Dozenten, die unter 6 verschiedenen Schreibweisen geführt waren (ein respektive zwei akademische Titel, abgekürzter Vorname, lediglich der Nachname, als Kürzel, komplett falsch geschrieben). Durch die Ersetzung all dieser unterschiedlichen Deklarationsformen durch eine numerische ID wird das fehlerkorrigierende Potenzial einer solchen Generalisierung deutlich.

lehrgebiet › **ark_record_area**

Alle im Feld **lehrgebiet** geführten Kürzel wurden durch numerische Referenzen auf neu geschaffene Einträge vom Typ **ark_area** ersetzt.

jahr > **ark_record_year**

Die Jahreszahlen des AMD-Feld **jahr** konnten ohne größeren Korrekturaufwand übernommen werden.

sprache > **ark_record_language**

Die Sprache der Arbeit, welche bereits in RFC 1766-konformer Weise in der AMD vorlag, wurde in das Feld **ark_record_language** übernommen.

abstract > **post_content**

Das Feld **abstract** wurde in das **post_content**-Feld des Posttyps **ark_record** übertragen. Zwecks korrekter Importierung wurden alle Umlaute und Sonderzeichen durch ihre HTML-Entsprechungen ersetzt und nach Abschluss des Imports via SQL-Query wieder in ihre Unicode-Entsprechungen übersetzt.

keywords > **post_tag**

Die Schlagwörter wurden innerhalb der AMD neben der gewünschten, kommaseparierten Trennung durch Semikola oder lediglich Leerzeichen getrennt angegeben. Diese mussten händisch separiert werden, um korrekt auf die WordPress-eigene Schlagwort-Taxonomie übertragen werden zu können.

dateigroesse > —

Das Feld **dateigroesse** wurde nicht übernommen, da diese innerhalb des KISDarchivs nicht angezeigt wird und bei Bedarf live ausgelesen werden kann.

pdf_anmerkung > **ark_record_note**

Das Anmerkungsfield wurde 1:1 auf seine Entsprechung übertragen. Mit Implementierung der Arbeit einreichen-Maske wurde den Nutzern überdies die Möglichkeit geschaffen, Anmerkungen zu den von ihnen eingereichten Datensätzen in dieses Feld zu schreiben und auf diesem Wege mit der Archiv-AG zu kommunizieren.

pdf_md5 > —

Das Feld **pdf_md5** wurde nicht übernommen.

zuletzt_geaendert > —

Das Feld **zuletzt_geaendert** wurde nicht übernommen, da diese Information über WordPress-eigene Felder abgebildet werden kann.

letzter_bearbeiter > —

Das Feld **letzter_bearbeiter** wurde nicht übernommen, da diese Information über WordPress-eigene Felder abgebildet werden kann.

9.2.2 Konsolidierung

ABB. 31 Konsolidierung doppelter Datensätze vor Einlesevorgang.

uri	title	author	exam	subject	area	year	tag
04.pdf	Daily Agent. Blind to handle - Konzeption eines Kommunikat	####	dipi	ht		8161 64 2004	0 76
04.pdf	Daily Agent. Blind to handle - Konzeption eines Kommunikat	####	dipi	ht		8161 64 2004	0 76
ht.pdf	urban tools	####	dipi	ht		8170 1161 2000	0 ##
ht.pdf	urban tools	####	dipi	ht		8170 1161 2000	0 ##
ht.pdf	smove – Audiovisuelle Kommunikation für eine Die	####	dipi	ht		8195 1159 2000	3 ##
ht.pdf	smove – Audiovisuelle Kommunikation für eine Die	####	dipi	ht		8195 1159 2000	3 ##
ht.pdf	Imagefilm für die Kün International School of Design	####	vordipi	ht		76 53 2004	0
ht.pdf	How to use a Hoensaal	####	vordipi	ht		1166 2000	3 32
ht.pdf	Personalisierte Informationssysteme in Bibliotheken	####	dipi	ht		8195 1159 2008	0
ht.pdf	Personalisierte Informationssysteme in Bibliotheken	####	dipi	ht		8167 1159 2008	0

Mit der soeben beschriebenen Zuordnung und Zusammenfassung von Feldern der AMD in Posttyp-Felder des KISDarchivs war die Grundlage für die Konsolidierung der Datensätze geschaffen. Tabellarisch gesprochen diente das Mapping der korrekten Zuordnung der Spalten. Nun galt es, sich den Zeilen zu widmen. Im ersten Schritt wurden alle Tabellenzeilen herausgefiltert, die keine Einträge referenzierten. Etwa 1300 Kurzbeschreibungs-Dateien und sonstige Referenzen wurden auf diese Weise extrahiert. Im zweiten Schritt galt es, Arbeiten, welche in Form mehrerer Einträge abgelegt waren, in jeweils einen Eintrag zu bündeln.

9.3 Einlesevorgang

An diesem Punkt im Importvorgang lagen die Daten aus dem AMD-Metadatenbestand in gepappter, aufbereiteter und konsolidierter Tabellenform vor.

Die aufbereiteten Daten wurden mithilfe des WordPress-Plugins »WP Ultimate CSV Importer« eingelese. Im ersten Schritt wurden die aus den AMD-Daten generierten Autoren- / Lehrenden- und Lehrgebietsdaten in die korrespondierenden Posttypen übertragen und die so generierten Posts anhand ihrer IDs in den Eintragsdaten referenziert. Anschließend wurden alle Eintragsdaten auf die selbe Weise ins KISDarchiv eingelese.

9.4 Datenabgleich & Zweitimport

Nach dem Einlesen der AMD-Metadaten in eigenständige KISDarchiv-Einträge mussten ebendiese Einträge mit dem physischen Datenbestand, welchen sie referenzierten, abgeglichen werden. Dieser war über einen längeren Zeitraum hinweg Manipulationen ausgesetzt gewesen, sodass die in den Metadaten erfassten Pfadangaben nicht mehr als zuverlässig zu betrachten waren. Überdies existierten, wie bereits geschildert, Dateien auf dem Gruppenordner, die noch gar nicht in der AMD erfasst gewesen waren.

All dies machte einen Datenabgleich zwingend erforderlich. Dieser umfasste:

1. Die Erstellung einer Liste L_A , welche alle aus der AMD ins KISDarchiv überführten PDF-Pfade enthielt.
2. Die Generierung einer Liste L_G , die sämtliche auf dem Gruppenordner befindlichen Dateien samt Pfad enthielt, sowie und die Filterung dieser Liste,

um lediglich die für den Abgleich relevanten Dateien abzubilden.

3. Die Gegenüberstellung der Listen L_A und L_G , wobei in erster Instanz alle übereinstimmenden Pfade der beiden Listen, sprich: »Dateien, die dort liegen, wo sie laut Pfad liegen sollten«, entfernt wurden. Im zweiten Schritt wurden alle lediglich in L_G geführten Pfade, also »Dateien, die physisch vorhanden sind, jedoch nicht unter dem angegebenen Pfad in der AMD geführt werden«, in zwei neue Listen aufgeteilt:
 - Liste L_V – **Verschobene Dateien** (L_G -Pfad weicht von L_A -Pfad ab)
 - Liste L_N – **Neue Dateien** (L_G -Pfad hat keine Entsprechung in L_A)

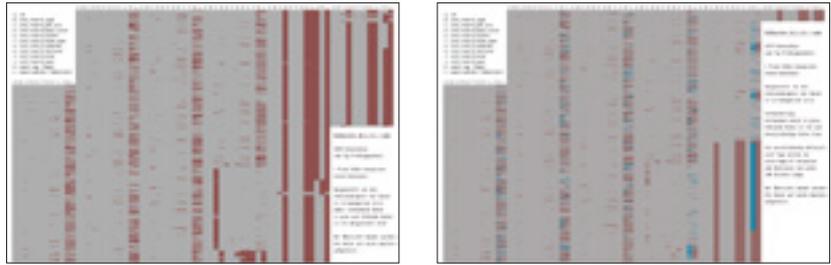
Die L_V -Liste wurde ein weiteres mal mit der Liste L_A abgeglichen. Dies setzte die Erzeugung approximierter Pfade voraus die aus den vorliegenden Metadaten *reverse engineered* wurden, um jede verschobene Datei einem Datensatz zuordnen zu können. Dieser Prozess sollte eine Woche in Anspruch nehmen.

Die L_N -Liste, (»Nicht in der AMD geführte, wohl aber im Gruppenordner vorhandene Dateien«) umfasste 658 Arbeiten, welche auf dem Gruppenordner abgelegt wurden, ohne eine Eintragung in der AMD vorzunehmen. Die Liste bildete den Ausgangspunkt für eine Tabelle, welche nach und nach mit Metadaten befüllt wurde. Die Autoren dieser Arbeiten wurden mit bereits im KISDarchiv vorhandenen Autorendaten abgeglichen, 26 neu hinzugekommene Autoren in dieses nachgetragen. Die Metadaten-Anreicherung schritt voran, einige Angaben ließen sich aus den PDF-Pfaden ableiten, andere mussten händisch recherchiert werden. Der resultierende Metadaten-Neubestand wurde endlich ebenfalls ins KISDarchiv eingelesen, welches nun alle physisch im Gruppenordner vorliegenden Dateien in Form von Datensätzen abbildete.

Eine gewisse Redundanz ließ sich in diesem Prozess nicht vermeiden, da nicht alle überführten Einträge über PDF-Pfade verfügten. Diese würden später konsolidiert und ergänzt werden müssen.

Im nächsten Schritt wurden 300 Datensätze aus der NMD ins KISDarchiv eingelesen, welche sich durch fehlende PDF-Pfade auszeichneten. Diese trugen also zur Redundanz bei. Es sollte sich überdies zeigen, dass Arbeiten, welche von mehreren Autoren eingereicht worden waren, ebenfalls redundant auf dem Server abgelegt worden waren. Weiterhin konnten Duplikate auftreten, wenn eine Arbeit beispielsweise als Vordiplom und dann erneut als Intermediate-Arbeit abgelegt worden war.

ABB. 34 Zwei Integritätsdiagramme des Datenbestands, erstellt im Abstand von drei Tagen im Februar 2013. Der Rückgang roter Farbe zeugt von steigender Eintragsintegrität.



Auf diesem Wege ließ sich über einen gewissen Zeitraum beobachten, wie die Eintragsquantität nach und nach abnahm während die Qualität bedingt durch Konsolidierungen zu steigen begann. Eine Woche nach Beginn der Datenaufbereitung lag die Konsistenz des Datenbestands bei etwa 83%, zum jetzigen Zeitpunkt (März 2014) liegt der Wert bei 92%.

Innerhalb der Eintragskonsistenz machte sich vor allem das Fehlen von Schlagwörtern und Abstracts bemerkbar, da diese Daten sich nicht ohne größeren Aufwand rekonstruieren ließen, wie das etwa mit den bereits in den PDF-Pfaden abgebildeten Datenfeldern (Prüfungsstufe, Teilprüfung, Autor oder Jahr) der Fall war. Die Integritätsbestimmung erwies sich als nützliches Werkzeug und wurde sowohl back- als auch frontendseitig implementiert.

Siehe **Integrität** (S. 94) /

Siehe **Integritätsdiagramme** (S. 248).

9.7 Datenbankpflege

Nach der systematischen Aufbereitung der Daten mithilfe der Integritätsüberprüfung bedurfte es diversen Anpassungen am Metadatenbestand. Diese wurden direkt über ein MySQL-Interface auf der Datenbank vorgenommen und umfassten:

- Die Übersetzung von HTML-Sonderzeichen in ihre Unicode-Entsprechungen,
- Die Korrektur von PDF- und Attachment-Pfaden durch methodische Ersetzung aller Vorkommnisse der Zeichenkette `localhost` respektive `sandbox` bei Verschiebung des Systems auf neue Serverumgebungen,
- Das halbautomatisch Einfügen korrekter Delimiter zwischen Titel und Untertitel durch Aufspüren »verdächtiger« Zeichen wie Trennstrichen und Punkten,
- Das Auffinden und Aufspalten von Schlagwortketten ohne Delimiter,
- Die Bereinigung ungenutzter Felder und Posttypen nach deren Ausscheiden aus dem KISDarchiv,
- Die Vereinheitlichung von Anführungszeichen im Datenbestand.

10 Frontend-Entwicklung

ABB. 35 Diverse Frontend-Templates.



10.0 Einleitung

Die Entwicklung des Frontends ist keinesfalls als linearer Prozess zu begreifen. Die steigende Komplexität des KISDarchivs als Ganzes informierte in vielerlei Hinsicht immer wieder grundlegende Bestandteile. Eine chronologische Darstellung der Entwicklung wäre dementsprechend schwer zu begreifen.

Stattdessen möchte ich mich dem Frontend aus verschiedenen Betrachtungsabständen, die ich als Rahmen bezeichne, nähern.

- Der **erste Rahmen** umfasst Grundbausteine wie Typografie, grundlegende Elemente, interaktive Elemente sowie Farben.
- Der **zweite Rahmen** behandelt Ansichten, welche Einträgen quantitativ und in diversen Kontexten abbilden.
- Der **dritte Rahmen** umfasst die diversen Templates, welche die Module des Frontends bilden. In diesem Rahmen werden auch Abläufe, wie etwa das Einreichen von Einträgen, besprochen.

Diese Rahmen finden sich in ähnlicher Form im zweiten Teil dieser Arbeit wieder, welcher eine technischere Annäherung beinhaltet.

10.1 Typografie

Die Typografie innerhalb des KISDarchivs wurde durch Evaluierung der vorliegenden Altdaten unter Ableitung einer Bedeutungshierarchie in verschiedenen Kontexten konstruiert. Es folgt ein Einblick in die Methodik.

ABB. 36 Vergleich der Darstellung desselben Eintragsstils in diversen Kontexten:

- A Eintrageinzelsicht
- B Mehrfachansicht **Detail**
- C PDF-Ansicht, Infofenster
- D Mehrfachansicht **Tabelle**
- E Mehrfachansicht **Karten**
- F Integrität, Tooltip (invertiert)
- G PDF-Ansicht, Liste
- H Navigator, deaktive Zeile

Der Kontrast der Darstellung wurde angeglichen, um eine bessere Vergleichbarkeit zu gewährleisten.



Diese Abbildung zeigt die acht Kontexte, innerhalb derer ein Eintragsstil im Frontend dargestellt wird. Jede dieser Applikationen folgt anderen Rahmenbedingungen: Der Titel steht in Detail-, Tabellen und Kartendarstellungen, in Tooltips, ein- und mehrzeiliger Darstellung, allein und in Gesellschaft anderer Titel und Daten. Damit noch nicht genug: Das dem KISDarchiv zugrunde liegende Grid-System ist mit fünf Breakpoints ausgestattet, an denen das Layout sich verändernder Viewportgröße flexibel anpasst. Anders gesagt: Wollte man diese hier gezeigten acht Applikationen wiederum noch einmal innerhalb der sechs Auflösungsbereiche illustrieren, käme man nur für die Titeldarstellung auf vierzig verschiedene Abbildungen.

Die Konstruktion eines typografischen Regelwerks in einem solch volatilen Umfeld ist durch massiven Einsatz von Trial and Error geprägt.

Ich begann damit, die Eintrageinzelsicht und die diversen Mehrfachansichten zu entwickeln. Ausgehend von einer gewissen Bildschirmgröße wollte ich eine überschaubare Anzahl an Einträgen auf einer Bildschirmseite abbilden. Aus der zur Verfügung stehenden Fläche minus dem als hinreichend betrachteten Weißraum entstanden innerhalb der Mehrfachansicht wiederum kleinere Flächen, aus denen später die Kartendarstellung entstehen sollte. Innerhalb der Karten platzierte ich alle Elemente in einer Bedeutungshierarchie:

Titel > Untertitel > Autor(en) > sonstige Metadaten

Die Elemente wurden anschließend innerhalb der Hierarchie skaliert. Hier gereichte mir zum Vorteil, dass ich bereits die komplette Datenbasis vor

mir hatte, was jede Designentscheidung unmittelbar qualifizierbar machte. Ich ermittelte die Extreme; die kürzesten und längsten Namen und Titel:

Namen (ark_author : post_title)

1 Jin Xu
 2 Pia Alt
 ...
 1004 Jessica Patricial Vidal Masias de Brunne
 1005 Valentine Mayuran Emmanuel Gnanapragasam

Minimum: 6 / Durchschnitt: 15 / Maximum: 40 Zeichen.

Titel (ark_record : post_title)

1 X
 2 LED
 ...
 4254 Unterhaltung am stillen Örtchen - Auf der Suche nach Unterhaltungsmöglichkeiten auf öffentlichen Toiletten der Kölner Bars/Restaurants/Cafes und Reaktionen ihrer Besucher

Minimum: 1 / Durchschnitt: 41 / Maximum: 170 Zeichen.

Nun justierte ich die Elemente innerhalb ihrer Hierarchie, bis ich einen Punkt erreicht hatte, an dem bei einer normalen Bildschirmdarstellung (Fensterbreite: 1200px) 95% der Arbeiten innerhalb der diversen Kontexte abgebildet werden konnten, ohne diese abreviiieren zu müssen. Dieser Wert hätte noch gesteigert werden können, wenn ich eine schmalere laufende Schrift gewählt hätte. Die neue Helvetica schien jedoch die geeignete Wahl, da sie – und die ihr ähnlichen Schriften, welche als ihre Alternative dienen – auf einer Vielzahl von Plattformen vorliegt und von diversen Browsern und Engines angenehm gerendert wird.

Neben der Bedeutungshierarchie, die den diversen typografischen Elementen auferlegt wurde, spielte die Abstraktionsstufe eine entscheidende Rolle in der Ermittlung der Relationen. Will heißen: Je mehr Einträge auf einer Fläche XY abgebildet werden, desto höher ist in der Regel der Betrachtungs- und damit Abstraktionsgrad zum einzelnen Datensatz. Es findet eine Priorisierung statt, die sich in der Generalisierung und/oder Kürzung dargestellter Informationen äußert. So musste evaluiert werden, in welchen Kontexten ein oder mehrere Namen ausgeschrieben dargestellt werden konnten, und an welcher Stelle die Vornamen gekürzt werden mussten.

All dies im Nachhinein zu beschreiben ist jedoch müßig und wenig hilfreich – Wichtig ist lediglich die Feststellung, dass Realdaten im Designprozess immer über Blindtext siegen, dass diese über Extreme verfügen, die zu berücksichtigen zu einer Abbildung führen wird, welche künftige Extreme berücksichtigt.

10.2 Wortmarke

ABB. 37 KISDarchiv-Wortmarke im Seitenkopf.



Die KISDarchiv-Wortmarke im Seitenkopf dient als Link zur Startseite. Sie taucht neben dem Seitenkopf ebenfalls im Seitenfuß und im Bedienelement der PDF-Ansicht auf. Gesetzt ist sie in Myriad Pro bold und regular.

10.3 Formularelemente

10.3.1 Suchfeld

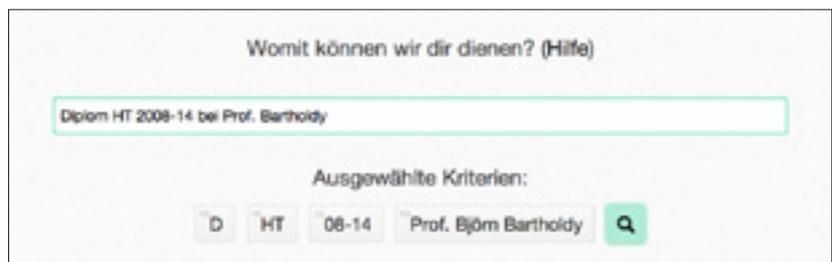
ABB. 38 Suchfeld im Seitenkopf.



Das Suchfeld bildet ein ordinäres Texteingabefeld, dessen Wert bei Betätigen der Eingabetaste oder Klick auf die Lupe im Feld abgesandt wird.

10.3.2 Metasuchfeld

ABB. 39 Metasuchfeld mit Kriterienanzeige; Metasuche.



Das Metasuchfeld fungiert in Konjunktion mit einer Kriterienanzeige, welche die live-geparsten Parameter und einen Absende-Button darstellt.

10.3.3 Textfeld

ABB. 40 Texteingabefelder; Arbeit einreichen-Maske.



Das reguläre Textfeld sowie das Textarea-Feld zeichnen sich durch eine Hervorhebung in ausgewähltem Zustand aus. In speziellen Fällen, wie etwa bei der Eingabe des Prüfungsjahrs, lassen die Felder nur numerische Eingaben zu.

10.3.4 Dropdown

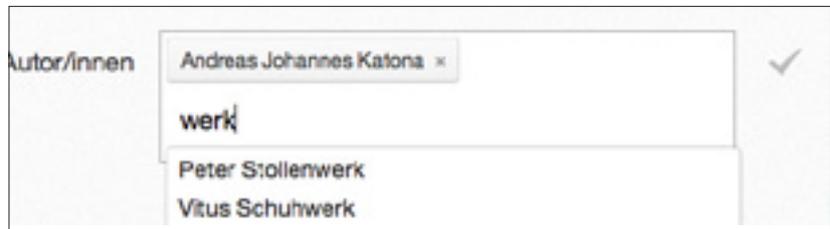
ABB. 41 Dropdown-Feld;
Arbeit einreichen-Maske.



Dropdownfelder werden sowohl in ihrer gewöhnlichen Form sowie als konditionale Dropdowns eingesetzt. Letztere zeichnen sich dadurch aus, dass sie in linearer Folge ausgefüllt werden müssen, um das jeweils nächste Feld zur Auswahl freizuschalten. Desweiteren bedingt die Auswahl eines Felds die möglichen auswählbaren Werte des nächsten Felds.

10.3.5 Tagfeld

ABB. 42 Tagsinput-Feld mit
Autocomplete-Funktion;
Arbeit einreichen-Maske.



Tagfelder kommen bei Auswahl von Schlagwörtern, Autoren- und Lehrenden vor und bedienen sich teilweise einer datenbankgestützten Autocomplete-Funktion, welche dem Nutzer eine Liste möglicher Werte ausgibt.

Sobald ein Name aus der Liste ausgewählt wird, erscheint dieser in Buttonform oberhalb der Eingabefläche im Tagfeld. Entfernt werden kann der so ausgewählte Button per Klick auf das **x** am rechten Buttonrand.

10.3.6 Button

ABB. 43 Button,
Upload-Fortschrittsanzeige,
Send-Button (deaktiviert),
Send-Button (gehighlightet);
Arbeit einreichen-Maske.



Button-Elemente existieren in normaler sowie gehighlighteter Ausführung. Darüber hinaus können Buttons in Sonderfällen, etwa während des Datei-Uploads, deaktiviert dargestellt werden.

10.4 Elemente

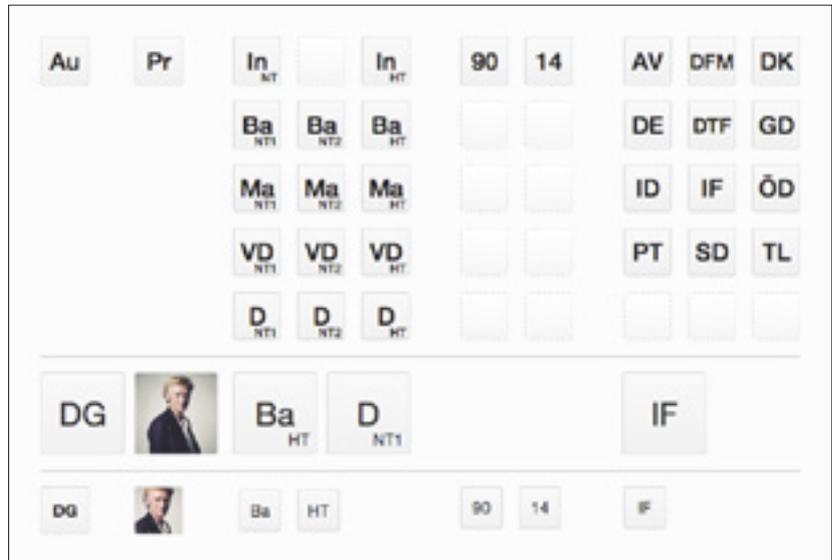
Die Tatsache, dass zwischen Prüfungsstufen-, Teilprüfungs- und Lehrgebietskurzformen keine Überschneidungen existieren, ermöglichte die Elemente :

ABB. 44 Element-Übersicht:

oben: Autoren-, Prüfer-, Prüfungsstufen/Teilprüfungs-, Jahres- und Lehrgebiets-elemente.

Darauf folgend in großer Abbildung: Autoren-, Prüfer- Prüfungsstufen/Teilprüfungs- und Lehrgebiets-element.

unten: Elemente für die Darstellung im Kontext der Mehrfachansicht **Tabelle**.



Die Elemente erscheinen in Mehrfachansichten und der Eintrags-Einzelansicht. Je nach zur Verfügung stehendem Raum und Informationsdichte wurden die Elemente selbst und die Lettern darin skaliert, um in allen Kontexten eine gute Lesbarkeit zu gewährleisten.

Die Anordnung der Elemente stellt dabei sicher, dass ähnliche Daten an der selben Position stehen. Siehe **Mehrfachansicht Karten** (S. 72).

10.4.1 Elementtypen

Im Kontext der Mehrfachansichten erscheinen die Personen-Bezeichner **Au** (Autor) und **Pr** (Prüfer/Professor). In der Mehrfachansicht **Tabelle** sowie der Eintrags-Einzelansicht erscheint das Initial- (Autoren und Prüfer) und Portraitelement (Prüfer). In Mehrfachansichten und der Eintrags-Einzelansicht erscheint das Prüfungsstufenelement samt Teilprüfung; in der Mehrfachansicht **Tabelle** aus Platzgründen getrennt als Prüfungsstufen- und separates Teilprüfungselement. In Mehrfachansichten sowie der Eintrags-Einzelansicht finden überdies das Jahreselement und das Lehrgebietselement Abbildung.

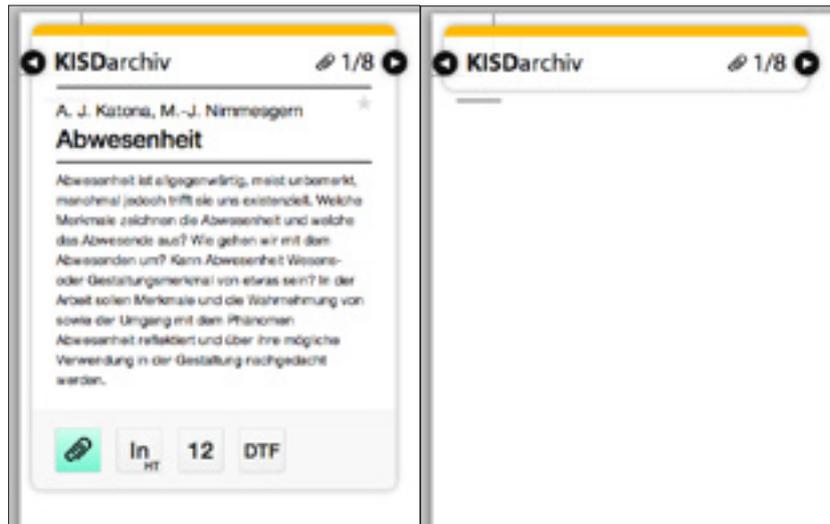
Tooltips geben bei Mouseover Auskunft über die Bedeutung hinter Elementen:

ABB. 45 Tooltip bei Mouseover über ein Element in der Mehrfach-Darstellung **Karten**.



10.5 PDF-Ansicht

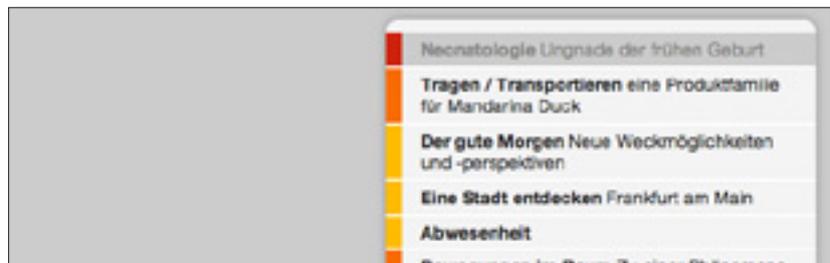
ABB. 46 PDF-Ansicht;
Lesezeichen-Bedienelement
in ausgefahrenem sowie
verborgenen Zustand.



Das Lesezeichen-Bedienelement orientiert sich an den Karten, welche im Rahmen der gleichnamigen Mehrfachansicht erscheinen. Das Element ist etwas größer dimensioniert, um zusätzlich zu Titel und Elementleiste noch einen Einblick ins Abstract geben zu können.

Im oberen Bereich des Elements finden sich zwei Pfeil-Buttons, welche die vorige, respektive nächste Arbeit in der Lesezeichenliste aufrufen. Zudem ist dort die KISDarchiv-Wortmarke platziert, welche als Link auf die Startseite dient und ein Button in Büroklammer-Form, der die Lesezeichenliste aufruft:

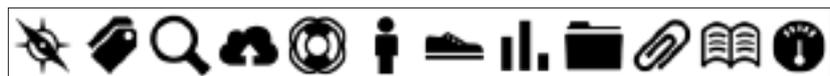
ABB. 47 PDF-Ansicht;
Lesezeichenliste.



Die Lesezeichenliste bildet alle mit Lesezeichen versehenen Einträge farbko-
diert ab, der aktuell angezeigte Eintrag ist grau hinterlegt. Ein Klick auf einen
Titel ruft die entsprechende Arbeit auf.

10.6 Piktogramme

ABB. 48 Helveticons (Auswahl)



Als Piktogramme kommen die »Helveticons« vom schwedischen Designstudio
»Goodbye Horses« zum Einsatz. Deren Einsatz variiert je nach Kontext:

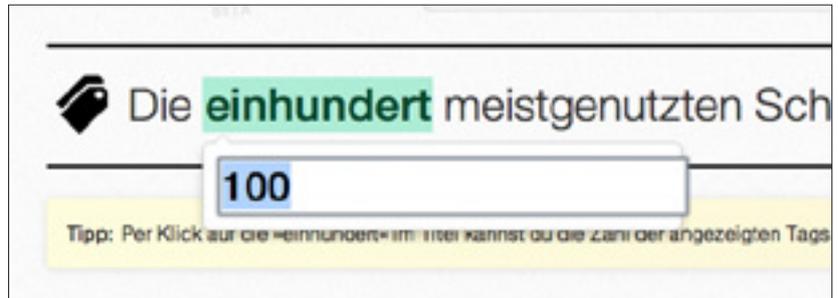
- Auf der Startseite grenzen die Piktogramme die Module von einander ab.
- Auf einzelnen Unterseiten werden mithilfe der Piktogramme die jeweiligen Seiteninhalte kontextuell verortet.

10.7 Interaktive Headline

Die Interaktive Headline ist ein Link in Form einer farbigen Hinterlegung welcher es ermöglicht, die auf der jeweiligen Seite angezeigten Inhalte zu modifizieren. In folgenden Kontexten tritt das Element in Erscheinung:

10.7.1 Schlagwort-Mehrfachansicht

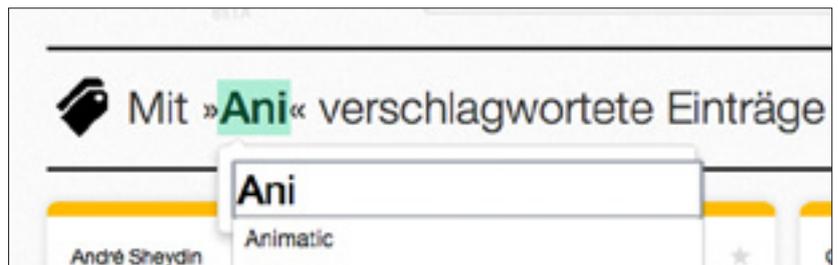
ABB. 49 Interaktive Headline; Schlagwortseite



In der Schlagwort-Mehrfachansicht lässt sich via interaktiver Headline die Zahl der angezeigten meistgenutzten Schlagwörter zwischen 2 und 999 variieren.

10.7.2 Schlagwort-Einzelansicht

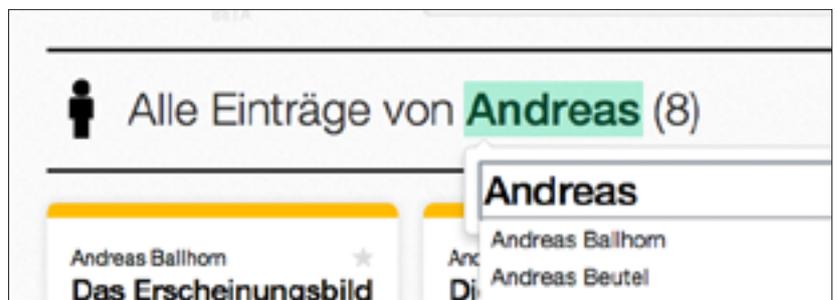
ABB. 50 Interaktive Headline; Schlagwort Einzelansicht



Die interaktive Headline in der Schlagwort-Einzelansicht ermöglicht es dem Besucher, das referenzierte Schlagwort zu ändern. Das Eingabefeld an dieser Stelle verfügt über eine Autovervollständigung, welches sich aus einem Verzeichnis aller vorhandenen Schlagwörter speist.

10.7.3 Autoren-Einzelansicht

ABB. 51 Interaktive Headline; Autoren-Einzelansicht



In der Autoren-Einzelansicht erfüllt die interaktive Headline den Zweck, zwischen Autoren umzuschalten. Ein Eingabefeld samt Autovervollständigung soll dabei gewährleisten, dass nur in der Datenbank erfasste Autoren auch ausgewählt werden.

10.7.4 Lehrenden-Einzelansicht

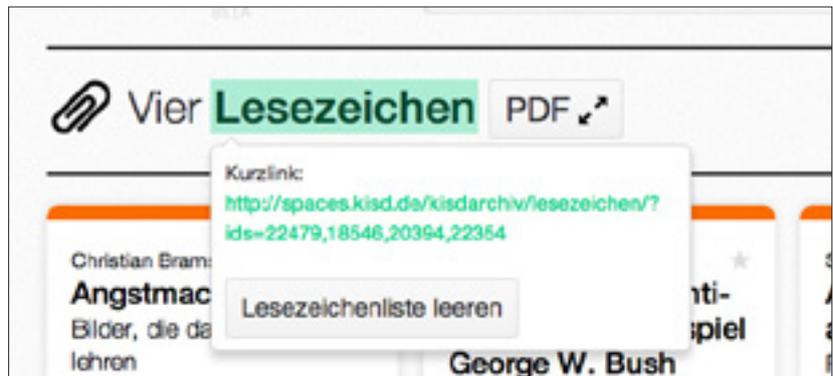
ABB. 52 Interaktive Headline;
Lehrenden-Einzelansicht



Die interaktive Headline der Lehrenden-Einzelansicht dient dem Besucher dazu, zwischen Lehrenden zu wechseln. Das entsprechende Eingabefeld greift via Autovervollständigung auf alle hinterlegten Lehrenden zu.

10.7.5 Lesezeichen-Seite

ABB. 53 Interaktive Headline;
Lesezeichenseite



Die interaktive Headline auf der Lesezeichen-Seite öffnet ein Popup-Fenster, welches einerseits einen Shortlink zur aktuell dargestellten Lesezeichenliste ermöglicht, andererseits einen Button, welcher die Lesezeichenliste leert.

10.8 Farben

10.8.1 Wortmarke

Der Begriff »archiv« der KISDarchiv-Wortmarke erscheint in **Monza Rot**.

ABB. 54 Monza-Rot

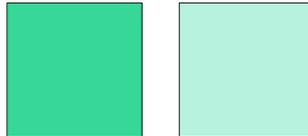


ZUORDNUNG	FARBE	WEBSAFE	HEX	R	G	B	A
HERVORHEBUNG	Monza (100%)	# CC 00 00	# DD 00 00	221	0	0	1

10.8.2 Hervorhebungen, Hinterlegungen

Interaktive Elemente, Links und diverse Buttons werden in der Farbe **Shamrock** dargestellt. Hinterlegungen nutzen 35-prozentige Deckkraft:

ABB. 55 Hervorhebungs- und Hinterlegungsfarbe Shamrock.



ZUORDNUNG	FARBE	WEBSAFE	HEX	R	G	B	A
HERVORHEBUNG	Shamrock (100%)	# 33 CC 99	# 28 D7 94	40	215	148	1
HINTERLEGUNG	Shamrock (35%)	# CC FF CC	# B4 F1 DA	40	215	148	.35

10.8.3 Eintrags-Farbkodierung

2012 wurden im analogen Archiv Farbauszeichnungen eingeführt. Um eine bessere Unterscheidbarkeit in der Bildschirmdarstellung zu erreichen, habe ich die Farbwerte angepasst. In Anlehnung an die Regalbeklebungen im analogen Archiv wurde auch im digitalen Archiv eine Farbkodierung von Einträgen und Personengruppen implementiert.

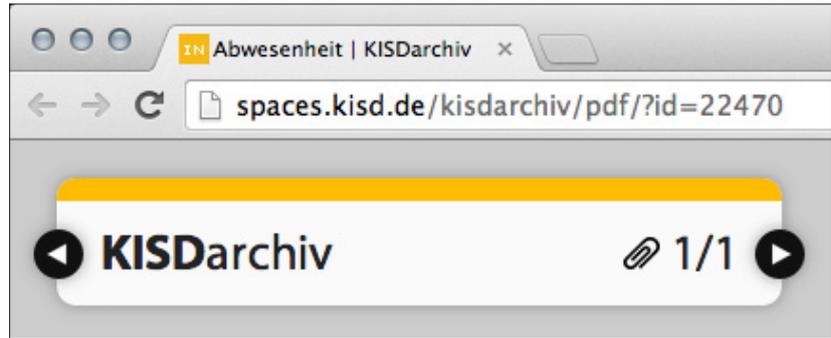
ABB. 56 Farbkodierte Favicons



Diese äußert sich in folgenden Kontexten:

- Farbstreifen in der Mehrfachansicht **Karten**
- Farbstreifen in der PDF-Ansicht (Info-Fenster sowie Lesezeichen-Liste)
- Favicon-Hinterlegung in Einzelansicht sowie PDF-Ansicht
- Statistik-Seite (leicht abgewandelte Farbigkeit zwecks Differenzierbarkeit)

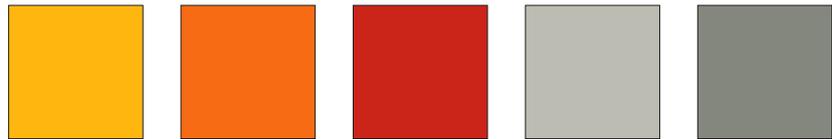
ABB. 57 Farbkodierung einer Intermediate-Arbeit in der PDF-Ansicht.



Die folgende Tabelle zeigt die Farbauszeichnungen laut Archiv-AG Handbuch (Stand 10.10.2013) und daneben die von mir abgeleiteten Farben für die Bildschirmdarstellung (entnommen aus der **ark-1.0.css**).

Keine Entsprechung haben nicht im digitalen Archiv abgelegte Inhalte, dafür sollen an dieser Stelle noch die nicht im AG-Handbuch kodierten Farben für Autoren und Lehrende Erwähnung finden, welche im Kontext der Mehrfachansicht **Karten** Verwendung finden.

ABB. 58 Farbkodierung (v.l.n.r.):
In/VD, Ba/D, Ma,
Autoren, Lehrende



ZUORDNUNG	FARBE	XF5*	HEX	R	G	B
DOKUMENTATIONEN	Grün	5067	—	—	—	—
INTERMEDIATE/VORDIPLOM	Gelb-Orange	5024	# FF BA 00	255	186	0
BACHELOR/DIPLOM	Orange	5028	# F6 6B 13	246	107	19
MASTERARBEITEN	Röt-Orange	5051	# CB 24 19	203	36	25
PUBLIKATIONEN	Blau	5050	—	—	—	—
ARCHIV INTERN	Grau	5019	—	—	—	—
AUTOREN	Hellgrau	—	# BC BC B2	188	188	178
LEHRENDE	Dunkelgrau	—	# 84 87 7E	132	135	126

*Die Nummer der Farbfolie, die Werte beziehen sich auf den X-Film XF5.

10.9 Ansichten

Der Begriff »Ansicht« (engl. »View«, auch: »Darstellung«, »Repräsentation«) bezeichnet die Darstellungen von Datensätzen und generischen Inhalten (»Seiten«). Datensätze selbst sind dabei bestimmten Typs. In der Darstellung wird zwischen Einzel- und Mehrfach-Darstellungen differenziert. Während erstere zumeist an einen Eintrags-Posttyp gebunden sind, bilden letztere verschiedene Darstellungsmodi einer Anzahl von Einträgen verschiedenen Typs ab. Die Ansichten kommen innerhalb diverser Templates zum Einsatz. Siehe **Templates und Module** (S. 75).

10.9.1 Einzelansichten

Der Begriff »Einzelansicht« bezeichnet eine Eintragsseite bestimmten Posttyps, die lediglich den Inhalt eines Eintrags darstellt. Die prototypische Einzelansicht zum Posttyp **ark_record** etwa zeigt die vollständigen Metadaten eines Eintrags an. Damit bilden die Einzelansichten das Gegenstück zu den Mehrfachansichten, welche jeweils mehrere Einträge auf einmal darstellen.

Zu beachten – und im Rahmen ihrer Templates dargestellt – sind die Einzelansichten zu den Posttypen **ark_author** und **ark_lecturer** welche sich dadurch auszeichnen, dass sie nicht die Metadaten der jeweiligen Autoren bzw. Lehrenden abbilden, sondern alle mit den entsprechenden Autoren und Lehrenden assoziierten Einträge. Die Assoziation an dieser Stelle bezeichnet das Verhältnis: »Autor/in – Alle von dieser Person verfassten Arbeiten« respektive »Prüfer/in – alle von dieser Person geprüften Arbeiten«. Sie sind damit als Mehrfachansichten mit Referenz auf jeweils eine Person zu verstehen.

Die Entwicklung des Frontends ging von der Darstellung einzelner Einträge des Posttyps **ark_record** aus.

Als Ausgangspunkt diente dabei die Einzelansicht eines Eintrags, der Fokus auf einer einzigen Arbeit. Innerhalb dieses Rahmens mussten alle Daten so untergebracht sein, dass sie sich dem Leser schnell erschließen konnten.

Die Einzelansichten werden im Rahmen ihrer Templates detailliert beschrieben.

Siehe **Templates und Module** (S. 75).

10.9.2 Mehrfachansichten

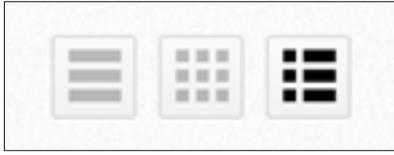


ABB. 59 Mehrfachansichts-Buttons
(Ansicht **Detail** aktiviert)

Die Implementierung dreier verschiedener Ansichten auf ein- und derselben Datenbasis entstand aus der Überlegung heraus, den Besuchern eine höchstmögliche Flexibilität im Umgang mit den Daten zu ermöglichen.

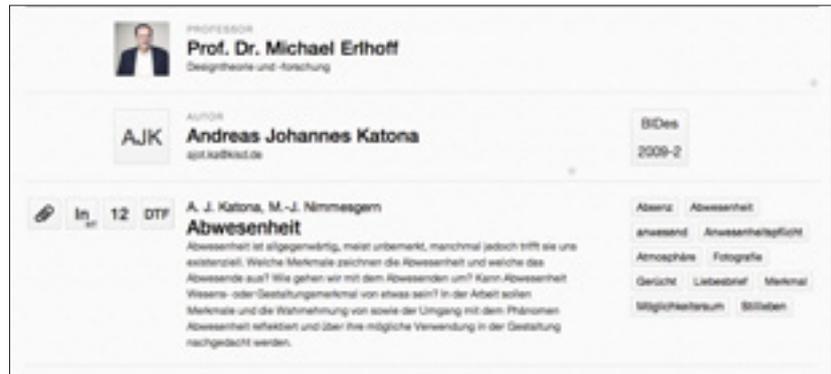
Siehe **Flexibilität durch Nonlinearität** (S. 38).

Die drei Mehrfachansichten **Tabelle**, **Karten** und **Detail** treten analog und überall dort in Erscheinung, wo mehr als ein Datensatz gleichzeitig dargestellt wird. Über neben der Headline positionierte Bedienelemente kann zwischen den drei Ansichten umgeschaltet werden. Wird auf diese Weise eine Präferenz angegeben, speichert das KISDarchiv die Auswahl im Cookie **ark_view**, sodass diese beim nächsten Aufruf einer Mehrfachansicht respektive einem erneuten Seitenbesuch erhalten bleibt.

Siehe **ark_view** (S. 158).

10.9.3 Mehrfachansicht Detail

ABB. 60 Anzeige von Einträgen der Datentypen `ark_lecturer`, `ark_author` und `ark_record` in der Detail-Mehrfachansicht



Die Mehrfachansicht **Detail** bildet Einträge unterschiedlichen Datentyps in mehrspaltiger, detaillierter Darstellung ab.

Abbildung

Betrachten wir die Abbildung von Metadaten in der Detailansicht pro Posttyp:

Posttyp `ark_record`

Elementleiste:

- Interaktiver **Lesezeichen-Button**,
- **Prüfungsstufen-** und **Teilprüfungskürzel**,
- **Jahr der Arbeit** als zweistellige Jahreszahl,
- Zwei- bis dreistelliges **Lehrgebietskürzel** sowie
- **Autorennamen** in etwaig gekürzter Form, welche als Hyperlink auf die Autoren-Einzelansichten fungieren.
- **Titel, Untertitel** und **Abstract**
- Liste dem Eintrag zugeordneter **Schlagwörter**

Posttyp `ark_author`

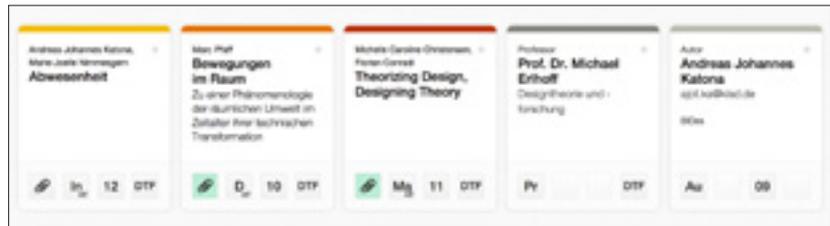
- Element mit den **Initialien** des Autors
- Klassifizierung **Autor/in**, je nach hinterlegtem Geschlecht
- **Vollständiger Name**
- **E-Mail-Adresse**, sofern hinterlegt
- Element mit **Studiengang** sowie **Studienbeginn**

Posttyp `ark_lecturer`

- **Portrait** des Prüfers, falls vorhanden, ansonsten **Initialien**
- Klassifizierung **Professor/in** oder **Prüfer/in**, je nach Status und Geschlecht
- **Vollständiger Name** des Prüfers mitsamt Titel
- **Lehrgebiet** des Prüfers, falls zugeordnet

10.9.4 Mehrfachansicht **Karten**

ABB. 61 Einträge diverser Prüfungsstufen und Posttypen in der Kartendarstellung



Die Mehrfachansicht **Karten** stellt Datensätze als Karteikarten dar, die in einer Matrix angeordnet sind. Je nach Viewport-Breite können so bis zu sechs Karten pro Reihe dargestellt werden.

Abbildung

Die Abbildung diverser Posttypen in Kartenform variiert, da die personenbezogenen Metadaten nicht deckungsgleich mit denen des Posttyps **ark_record** sind. Im Folgenden werden kurz die drei Darstellungen der Eintrags-, Autoren- und Lehrenden-Abbildung erläutert.

Posttyp **ark_record**

Die Eintragskarte zum Posttyp **ark_record** umfasst folgende Elemente:

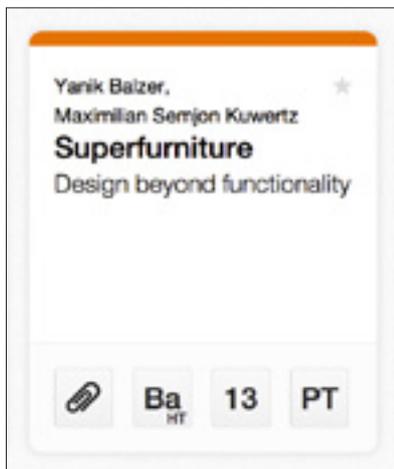


ABB. 62 Eintragskarte (ark_record)

Farbbalken: Die Karte für den Posttyp **ark_record** trägt einen farbigen Balken welcher die Prüfungsstufe in Anlehnung an die Farbkodierung des analogen Archivs anzeigt. Siehe **Eintrags-Farbkodierung** (S. 67).

Autorenname/n, in ausgeschriebener, kommaseparierter Form, welche als Hyperlinks auf die jeweilige Autoren-Einzelansicht fungieren.

Ein **Editier-Button in Sternform**, der als Link zur Eintrags-Bearbeitungsmaske im Backend führt, sofern der aktuell angemeldete WordPress-Benutzer über die Berechtigung der Eintrags-Editierung verfügt.

Der **Titel der Arbeit** sowie ein abgesetzter **Untertitel**, welche als Link auf die Eintrags-Einzelansicht dienen.

Die **Elementleiste:**

- Der interaktive **Lesezeichen-Button**,
- Das **Prüfungsstufen-** und **Teilprüfungskürzel**,
- Das **Jahr der Arbeit** als zweistellige Jahreszahl,
- Das zwei- bis dreistellige **Lehrgebietskürzel**

Diese Elemente sind sämtlich mit Tooltip-Funktionalität ausgestattet, um dem den Abkürzungen nicht vertrauten Benutzer deren Bedeutung zu erschließen.

Posttyp `ark_author`

Die Autoren-Eintragskarte umfasst folgende Elemente:

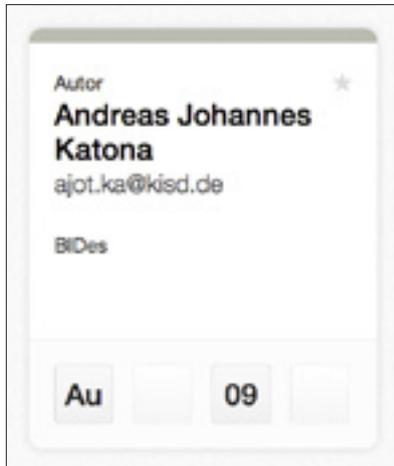


ABB. 63 Eintragskarte (`ark_author`)

Den **Farbbalken**, der hellgrau hinterlegt die Autorenkarte von den farbigen Balken des Posttyps `ark_record` abgrenzt.

Die Klassifizierung **Autor/in**, je nach Wert der `ark_author_gender`-Variable.

Den **Editier-Button in Sternform**, der als Link zur Autoren-Bearbeitungsmaske im Backend führt, sofern der aktuell angemeldete WordPress-Benutzer über die Berechtigung der Eintrags-Editierung verfügt.

Den **vollständigen Namen des Autoren**.

Die **E-Mail-Adresse des Autoren**, sofern hinterlegt.

Einen oder mehrere **Studiengänge des Autoren**, sofern hinterlegt.

Die **Elementleiste**:

- Das **Autorenkürzel** »Au«,
- Ein **Leerelement**, um die Darstellung konsistent zu halten,
- Das **Jahr des Studienbeginns**, sofern hinterlegt,
- Ein **Leerelement**, um die Darstellung konsistent zu halten.

Posttyp `ark_lecturer`

Die Prüfer-Eintragskarte umfasst folgende Elemente:

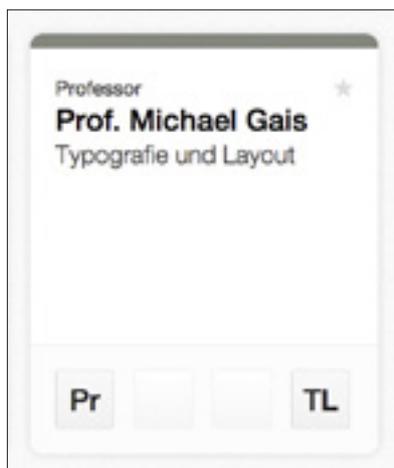


ABB. 64 Eintragskarte (`ark_lecturer`)

Den **Farbbalken**, der dunkelgrauer Balken grenzt die Prüfer von den farbigen Balken des Posttyps `ark_record` ab.

Die Klassifizierung **Professor/in** beziehungsweise **Prüfer/in**, je nach hinterlegtem Geschlecht und Status.

Den **Editier-Button in Sternform**, der als Link zur Prüfer-Bearbeitungsmaske im Backend führt, sofern der aktuell angemeldete WordPress-Benutzer über die Berechtigung der Eintrags-Editierung verfügt.

Den **vollständigen Namen des Prüfers samt Titel**.

Das **Lehrgebiet des Prüfers**, sofern zugeordnet.

Die **Elementleiste**:

- Das **Prüfer- und Professorenkürzel** »Pr«,
- Ein **Leerelement**, um die Darstellung konsistent zu halten,
- Ein **Leerelement**, um die Darstellung konsistent zu halten,
- Das **Lehrgebietskürzel**, sofern zugeordnet.

10.9.5 Mehrfachansicht **Tabelle**

ABB. 65 Einträge v. Typ `ark_lecturer`, `ark_author` und `ark_record` in der tabellarischen Darstellung

Titel	Autor/in	LG	Prüfer/in	PS	TP	JG
 Prof. Dr. Michael Erhoff			DTF			
 Andreas Johannes Katona						08
 Abwesenheit	A. J. Katona M.-J. Nimmegen		DTF Prof. Dr. Michael Erhoff	In	HT	12
 Bewegungen im Raum Zu einer Phänomenologie der räumlichen Umwelt im Zusatz zur ihrer technischen Transformation	Mani Patel		DTF Prof. Dr. Michael Erhoff	D	HT	10
 Theorizing Design, Designing Theory	M. C. Christensen F. Conrad		DTF Prof. Dr. Uta Brandes Prof. Dr. Michael Erhoff	Ma	HT	11
 Haute Cuisine Die Verköstigung von Passagieren an Bord von	Philippe Meas		DTF Prof. Hatto Griesse	In	HT	11

Die Mehrfachansicht **Tabelle** bildet Datensätze diversen Posttyps in einer tabellarischen Matrix ab. Gegenüber den anderen Mehrfachansichten zeichnet sich diese Darstellung dadurch aus, dass sie eine schnellere Vergleichbarkeit gewährleistet, da pro Bildschirmseite mehr Einträge Platz finden.

Posttyp `ark_record`

SPALTE 1 »-«	Interaktiver Lesezeichen-Button
SPALTE 2 »TITEL«	Titel und Untertitel der Arbeit
SPALTE 3 »AUTOR/IN«	Autor/en der Arbeit, gekürzte Vornamen bei mehreren Autoren
SPALTE 4 »LG«	Lehrgebiet , Kurzform
SPALTE 5 »PRÜFER/IN«	Prüfer der Arbeit, Hyperlink auf Prüfer-Einzelansicht
SPALTE 6 »PS«	Prüfungsstufe , Kurzform
SPALTE 7 »TP«	Teilprüfung , Kurzform
SPALTE 8 »JG«	Jahr der Arbeit, zweistellige Notation

Posttyp `ark_author`

SPALTE 1 »-«	Initialen des Autors
SPALTE 2 »TITEL«	Vollständiger Name des Autors
SPALTE 3 »AUTOR/IN«	-
SPALTE 4 »LG«	-
SPALTE 5 »PRÜFER/IN«	-
SPALTE 6 »PS«	-
SPALTE 7 »TP«	-
SPALTE 8 »JG«	Studienbeginn , zweistellige Notation falls vorhanden

Posttyp `ark_lecturer`

SPALTE 1 »-«	Portrait oder Initialen des Prüfers
SPALTE 2 »TITEL«	Akademischer Titel und Name des Prüfers
SPALTE 3 »AUTOR/IN«	-
SPALTE 4 »LG«	Lehrgebiet , Kurzform, falls dem Prüfer ein LG zugeordnet wurde
SPALTE 5 »PRÜFER/IN«	-
SPALTE 6 »PS«	-
SPALTE 7 »TP«	-
SPALTE 8 »JG«	-

10.10 Templates und Module

In diesem Teil werden die Templates inhaltlich besprochen, sowie die Auseinandersetzung mit ihnen im Rahmen der Entwicklung. Im zweiten Teil dieser Arbeit werden die technischen Aspekte der Templates behandelt.

Der Begriff »Modul« beschreibt die sinnhafte Gruppierung von Templates.

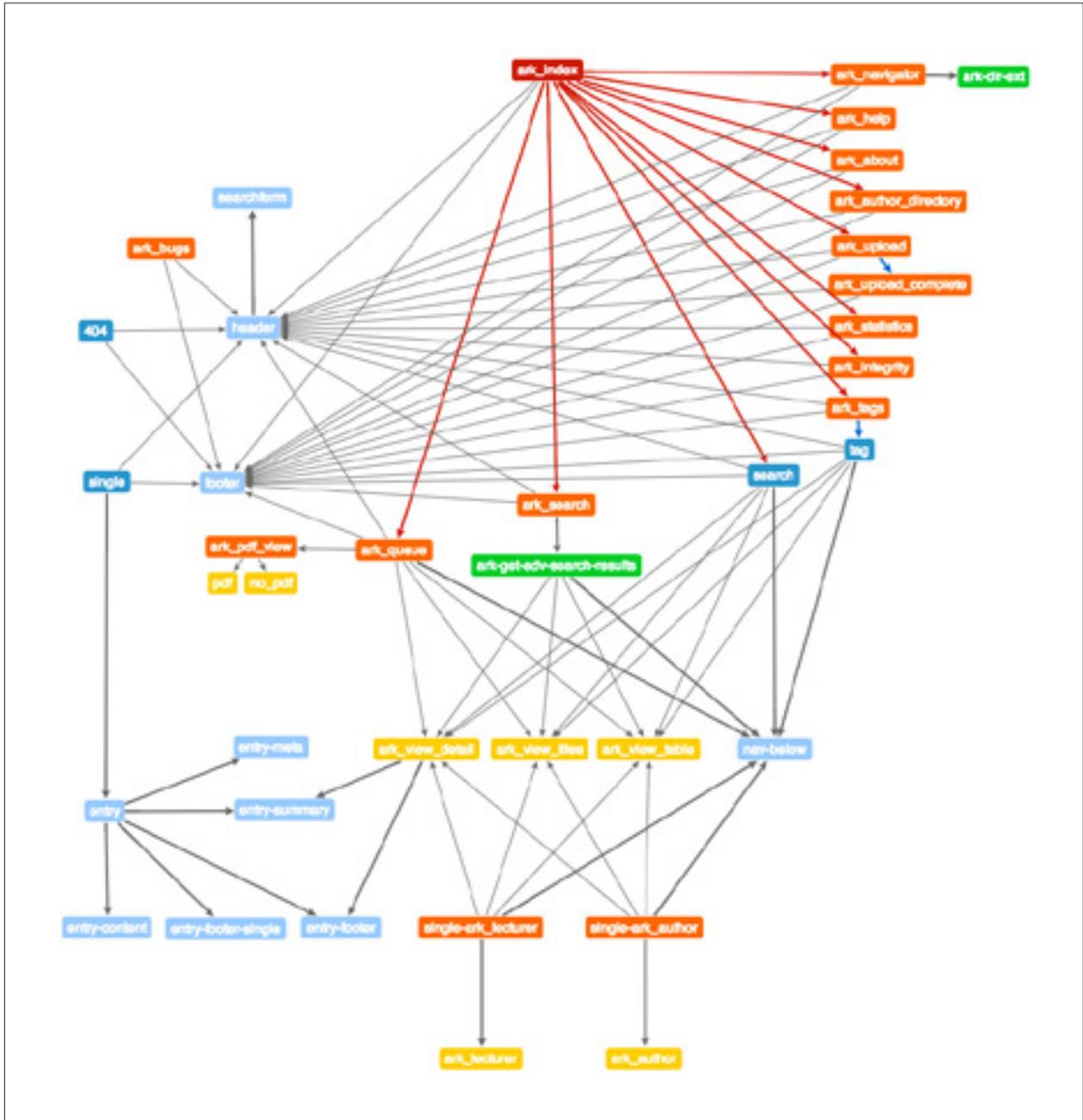


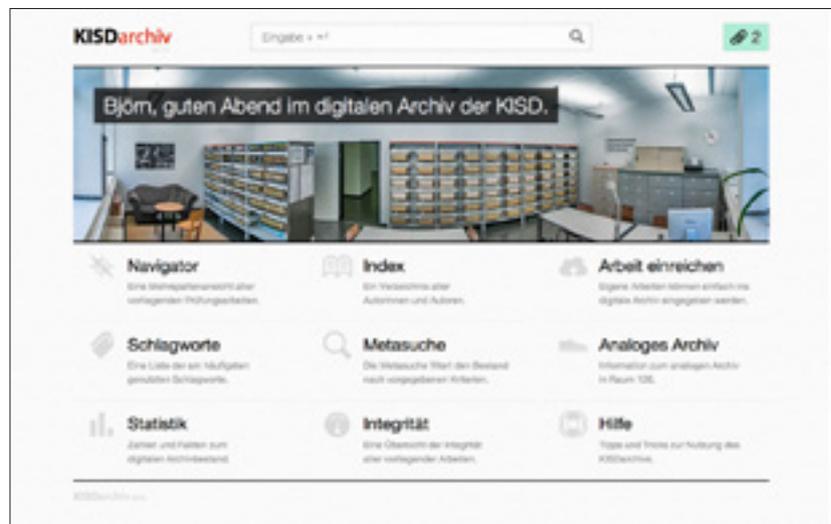
ABB. 66 KISDarchiv Template-Struktur:

Startseite samt Links in rot,
 KISDarchiv-Templates orange
 KISDarchiv-Subtemplates gelb
 WordPress-Templates dunkel-
 bzw. -Subtemplates hellblau,
 Includes in grün.

10.10.1 Startseite

Übersicht und Erklärung der Module.

ABB. 67 Startseite (Januar 2014)



Die Startseite wurde als Portal auf alle Module konzipiert. Wie Abb. 66 auf Seite 75 zeigt, erlaubt die Startseite den direkten Zugriff auf zehn KISDarchiv-Templates sowie die Suche. Die Startseite ist von jeder Unterseite aus mit einem Klick über die KISDarchiv-Wortmarke zu erreichen.

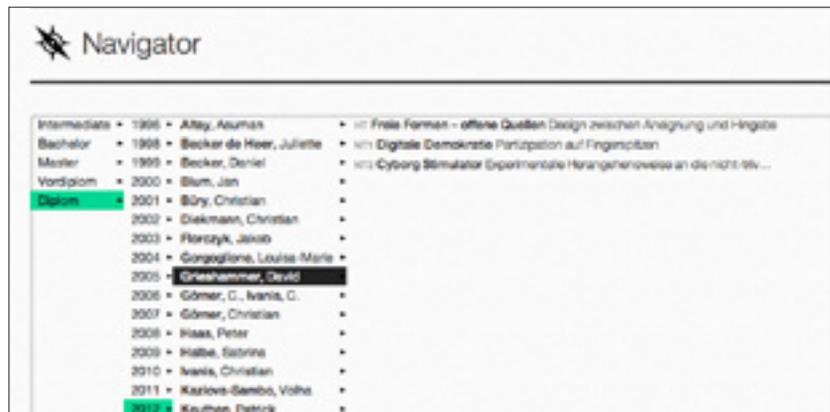
Die Startseite unterstreicht in ihrer nahezu hierarchielosen Darstellung die Philosophie hinter dem KISDarchiv selbst: Der Nutzer wird eingeladen, das Interface zu erfahren und sich zu eigen zu machen. Dies wird auch im Fehlen einer traditionellen Seitennavigation abseits der Startseite kommuniziert.

Die prominente Platzierung des Suchfelds, welches mit Ausnahme der PDF-Ansicht auf allen Seitentemplates im Seitenkopf erscheint, erlaubt den direkten Zugriff auf das zentrale Werkzeug des KISDarchivs. Sie zitiert die klassische Suchmaschinen-Erfahrung: Die Durchsuchbarkeit eines enormen Datenkorpus durch ein simples Eingabefeld.

10.10.2 Navigator

Verschachtelte Eintragsübersicht mit »Miller-Columns« alias »Einträge wie bisher durchsuchen, nur besser.«

ABB. 68 Navigator



Einleitung

Zum Zeitpunkt des Beginn meiner Arbeit hatte sich der Gruppenordner, welcher einst lediglich als Dateistruktur für das AdA gedient hatte, als primäres Interface für das digitale Archiv etabliert. Dieser Umstand war der Tatsache geschuldet, dass das eigentliche Interface des AdA nicht mehr in Betrieb war. Bei der Darstellung handelte es sich im Übrigen um eine generische Listenansicht, wie sie serverseitig vorgegeben ist.

Mir war daran gelegen, eine Implementierung der hierarchischen Ordnerstruktur wie sie der Gruppenordner bot auch im neuen digitalen Archiv zur Verfügung zu stellen. Für diese Darstellung eines virtuellen Verzeichnisbaums wählte ich den Namen »Navigator«. Diese Art der Darstellung im neuen digitalen Archiv aufzugreifen, hatte mehrere Gründe: Zum einen hatte man sich seitens der Studierendenschaft bereits an diese Art des Interfaces gewöhnt, zum anderen bot diese Art der Darstellung einen schnellen Überblick über große Datenmengen. Allerdings wies die Gruppenordner-Ansicht einige Schwachstellen auf, die es im Zuge der Adaptierung auf das neue digitale Archiv zu beseitigen galt:

Eigenverortung innerhalb der Hierarchie

Bewegte man sich durch die verschachtelten Ordner des Gruppenordners, so musste man sich ständig vergegenwärtigen, auf welcher Ebene man sich gerade befand. Ich wollte eine Möglichkeit schaffen, die Verschachtelung besser zu visualisieren und somit ein schnelleres »Auftauchen« aus den Tiefen der Struktur zu ermöglichen. Um dies zu erreichen bediente ich mich der Visualisierungstechnik der »Miller Columns«, die 1980 von Mark S. Miller entwickelt wurden, um hierarchische Baumstrukturen in einer Spaltenansicht darzustellen. Dieses Paradigma dürfte den meisten Nutzern aus seiner Integration im Mac OSX Finder bekannt sein. Die Implementierung wurde über ein JavaScript-Plugin realisiert, welches sowohl die Bedienung via Maus als auch über die Pfeiltasten der Tastatur ermöglicht.

Verbesserung der Darstellung und Sortierung einzelner Arbeiten

In der Gruppenordner-Ansicht waren Arbeiten nach der vorgegebenen Syntax `matrikelnummer_pruefungsstufe_teilpruefung.pdf` benannt. Aus dieser Benennung war nicht ohne weiteres ersichtlich, welchen Titel die einzelne Arbeit trägt. Zudem erschlossen sich Prüfungsstufe, Jahr und Autor nur aus den zuvor aufgerufenen Ordnern. Obendrein lagen im Gruppenordner neben den Arbeiten selbst noch so genannte »Notes«-Dokumente, in denen Metadaten zu den Arbeiten abgelegt waren. Im Gruppenordner wurden Arbeiten mehrerer Autoren oft redundant abgelegt, hier bot sich eine gemeinsame Anzeige an.

Struktur

Der Gruppenordner folgte in seiner Anzeige der folgenden Hierarchie:

Gruppenordner-Struktur

1. EBENE	Prüfungsstufe	(»Diplom«)
2. EBENE	Jahr	(»2010«)
3. EBENE	Autor/in	(»12345678_mustermann_max«)
4. EBENE	Arbeit(en)	(»12345678_d_ht.pdf«)

Diese wurde weitestgehend beibehalten, mit dem Unterschied, dass auf der 4. Ebene zusätzliche Metadaten angezeigt werden:

Navigator-Struktur

1. EBENE	Prüfungsstufe	(»Bachelor«)
2. EBENE	Jahr	(»2013«)
3. EBENE	Autor(en)	(»Eisbrenner, Johannes«)
4. EBENE	Pr.stufe / Titel	(NT1 / Das physische Modell im Gestaltungsprozess)

Funktionsweise

Die erste Iteration des Navigators wurde in der ersten Märzwoche 2013 implementiert. Ich werde an dieser Stelle die technischen Herausforderungen in der Umsetzung dieses Moduls beschreiben.

Der Gruppenordner, nach dessen Vorbild der Navigator aufgebaut ist, bildet eine physische Dateistruktur ab, wie sie auf dem Dateisystem des Servers vorliegt. Dies ist ein recht geradliniger Prozess, es muss lediglich die Dateistruktur ausgelesen und im Browser ausgegeben werden.

Der Navigator dagegen liest nicht etwa ein Dateisystem aus, sondern generiert die hierarchische Darstellung aus Metadaten.

Die Herausforderung besteht darin, dass die Metadaten selbst in WordPress keine Hierarchie abbilden, sondern parallel abgelegt sind.

An dieser Stelle soll grob die Funktionsweise umrissen werden:

1. Eine Datenbankabfrage bezieht zu allen Datensätze vom Typ **ark_record**, welche den Status »veröffentlicht« innehaben, die folgenden Metadaten:

- Eindeutige ID **ID**
- Titel der Arbeit **post_title**
- Prüfungsstufe **ark_record_exam_type**
- Autoren **ark_record_author**
- Teilprüfung **ark_record_subexam**
- Jahr **ark_record_year**

Die Metadaten werden in ein multidimensionales Array gespeichert.

2. Das Array wird rekursiv durchlaufen und folgende Operationen ausgeführt:

- Der Prüfungsstufe wird ein Wert zugewiesen, welcher der Sortierung dient.

- Der Autorenwert (Referenz auf Eintrag vom Posttyp **ark_author**) wird in Abhängigkeit von der Autorenanzahl in Namen umgewandelt.

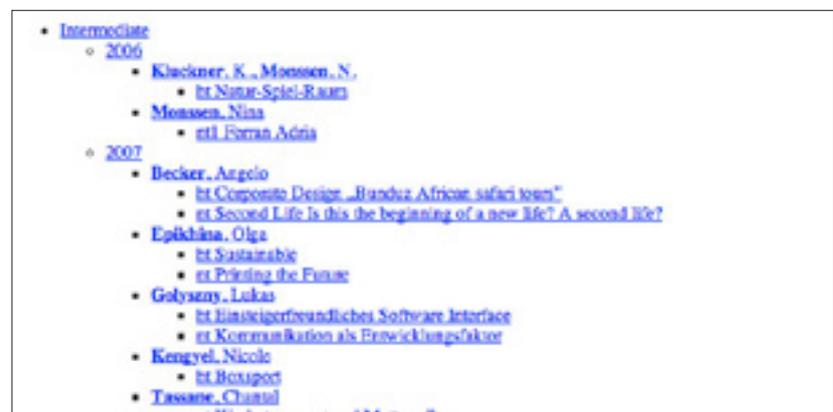
Syntax: `Nachname, Vorname` oder `Nachname, V. / Nachname, V.`

- Das Array wird rekursiv nach Prüfungsstufe, Jahr, Autor & Teilprüfung sortiert.

3. Das sortierte Array wird in einer verschachtelten Liste in HTML ausgegeben.

4. Die verschachtelte Liste wird per jQuery Plugin »jQuery Columnview« in einen Spaltenbrowser umgewandelt. Siehe **jQuery Columnview** (S. 160).

ABB. 69 Ausgabe der Navigator-Inhalte als vierdimensionale Liste vor Spaltenbrowser-Umwandlung.



Herausforderungen

Wie sich schnell zeigte, ist mit der Generierung einer Ausgabe auf Basis von Metadaten ein erheblicher Rechenaufwand verbunden. Dieser liegt darin begründet, dass zum einen sämtliche **ark_record**-Einträge durchlaufen und weiterhin für jeden einzelnen Datensatz noch einmal die Namen der zugehörigen Autoren ermittelt werden müssen. Die abschließende Sortierung des Arrays nach vier Dimensionen stellt sich ebenfalls als rechenintensiv dar. Die Größe des ausgegebenen Markups liegt bei 1,3 MB purem HTML.

In der ersten Iteration des Navigators versuchte ich, die Serverlast dadurch zu reduzieren, dass ich der Anzeige jeglicher Daten eine Abfrage voranstellte, welche Prüfungsstufe angezeigt werden sollte, um nur eine Teilmenge der Datensätze abfragen zu müssen. Dieser Schritt erschien mir jedoch als umständlich und stand dem Vorbild Gruppenordner insofern nach, als dass er nicht die selbe Menge an Daten auf einmal darstellen konnte.

Die im Navigator dargestellten Informationen bleiben im Regelbetrieb größtenteils unverändert. Änderungen treten hier auf, wenn Datensätze modifiziert oder hinzugefügt werden. Darum erschien mir eine Lösung sinnvoll, die die komplexen Operationen zur Generation des Navigators einmalig ausführte, um die dabei ermittelte Darstellung auf dem Server zwischenspeichern. Beim abermaligen Aufruf der Seite würden dann lediglich die bereits berechneten Darstellungsdaten übertragen, die Berechnung derselben wäre bereits erfolgt. Dieses Caching-Verfahren implementierte ich so, dass die gesamte Navigatorausgabe in einer Cache-Datei mit einer Halbwertszeit von 25 Tagen gespeichert wird, wonach der erste Aufruf des Navigators zur Neugenerierung des kompletten Datenbaums führt.

ABB. 70 KISDarchiv-Einstellungsseite; Navigator-Cache. (Administrator-Ansicht)



Besteht überdies das Bedürfnis, den Cache vor Ablauf zu regenerieren, so ist den administrativen Nutzern des KISDarchivs die Möglichkeit gegeben, über das Backend die aktuelle Cache-Datei zu tilgen und neu berechnen zu lassen. So kann etwa unmittelbar nach einer Abgabefrist dafür gesorgt werden, dass die neu im System vorhandenen Arbeiten zeitnah im Navigator sichtbar sind.

10.10.3 Suche

»Das Herzstück des KISDarchivs.«

ABB. 71 KISDarchiv-Startseite mit Suchfeld, mobile Ansicht.



Die Suche bildet das Herzstück des digitalen Archivs. Die Qualität der Ergebnisse bestimmt über Erfolg und Misserfolg in der Recherche. Zum Einsatz kommt das von Mikko Saari entwickelte WordPress-Plugin »Relevanssi«. Dieses bietet gegenüber der WordPress-eigenen Suche diverse Vorteile und Einstellungsmöglichkeiten, die hier dargestellt werden sollen.

Gewichtung / Fuzzy Matching

»Relevanssi« gibt die Suchergebnisse nach Relevanz sortiert aus, wobei eine Gewichtung angewandt wird. Taucht der gesuchte Begriff im Titel einer Arbeit auf so zählt das fünf mal so viel, wie wenn diese mit dem Begriff verschlagwortet ist. Darüber hinaus werden erst einmal exakte Übereinstimmungen bevorzugt, um bei keinem direkten Treffer auf Fuzzy Matching zu setzen, welches den Begriff innerhalb zusammengesetzter Begriffe sucht.

Indizierung von Posttypen

Die Suche ist so konfiguriert, dass lediglich Einträge vom typ **ark_record**, **ark_lecturer** und **ark_author** durchsucht und in den Ergebnissen angezeigt werden. Lehrgebiete oder generische Seiten sind somit nicht in den Suchergebnissen vertreten, was angesichts ihrer verschwindend geringen Menge gegenüber Einträgen anderer Posttypen nur logisch erscheint.

Indizierung von Metadatenfeldern / Taxonomien

Gegenüber der WordPress-eigenen Suche bietet »Relevanssi« die Option, Metadatenfelder zu indizieren und Einträge damit nicht nur aufgrund von Übereinstimmungen mit Suchbegriffen in den **post_title**- und **post_content**-Feldern sowie dem Eintrag zugeordneten Tags anzuzeigen. Dies ist von großer Bedeutung, da die Metadatenfelder wichtige Informationen zur Arbeit enthalten, welche andernfalls nicht durchsuchbar wären.

An dieser Stelle möchte ich auf zwei Felder besonders eingehen, deren Indizierung wesentlich dazu beiträgt, Einträge auffindbar zu machen:

10.10.4 Metasuche

Den Datenbestand nach festgelegten Kriterien durchsuchen.

ABB. 73 Metasuche (Ausschnitt)

Womit können wir dir dienen? (Hilfe)

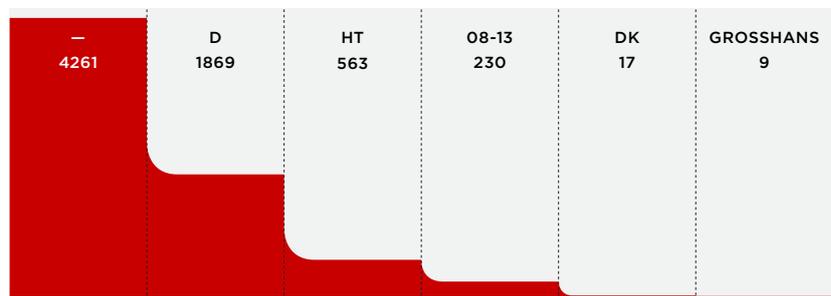
Ba HT 2008-13 bei Prof. Erhoff

Ausgewählte Kriterien:

Ba HT 08-13 Prof. Dr. Michael Erhoff Q

Die Metasuche bildet eine Ergänzung zur regulären Suchfunktion. Wie der Name vermuten lässt, lassen sich mit ihr gezielt Metadaten durchsuchen. Die Metasuche akzeptiert eine Suchanfrage in natürlicher Sprache. Der wesentliche Unterschied zur regulären Suche besteht dabei darin, dass die Metasuche nicht nach gezielten Begriffen in Titel oder Verschlagwortung suchen lässt, sondern nach jenen Metadaten, die größere Überschneidungen abbilden, wie etwa Lehrgebiets- oder Jahreszuordnungen. Eine andere Form, Metasuche und reguläre Suche zu differenzieren wäre, die Metasuche als Kombination von Filtern zu begreifen, mit denen sich der Datenbestand iterativ durchsieben lässt.

ABB. 74 Metasuche: Funktionsweise bei iterativer Eingabe von Kriterien (Stand: 25.2.2014).



Die Abbildung zeigt beispielhaft die Funktionsweise der Metasuche als filternde Instanz. Jede Spalte bildet die Ergebnismenge der kumulativen Kriterien der vorangegangenen Spalten ab, die Spalte ganz links den Gesamtbestand. Der gesamte Datenbestand – 4261 Einträge stark – wird durch Eingabe des Kriteriums **Diplom** (2. Spalte) auf 1869 angezeigte Einträge verjüngt. Unter Zugabe des Kriteriums **Hauptthema** reduziert sich Anzahl der Ergebnisse auf 563. Fügen wir die Einschränkung **Jahr 2008-13** hinzu, vermindert sich die Anzahl auf 230 Arbeiten, die weitere Kriterien-Hinzufügung des Lehrgebiets **DK** lässt die Anzahl noch einmal auf 17 Einträge schrumpfen bis durch die Eingabe des Prüfers **Prof. Großhans** noch einmal neun Einträge zurückbleiben.

Die Parameter können dabei in beliebiger Folge und selbstverständlich auch kollektiv eingegeben werden, die Grafik soll lediglich die Funktionsweise der Metasuche illustrieren.

Funktionsweise

Die Bedienung der Metasuche lässt sich in zwei Stufen unterteilen:

1. Die **Eingabe einer Suchanfrage** seitens des Besuchers
2. Die **Verarbeitung** (der Anfrage) **und Ausgabe** von Ergebnissen

Betrachten wir beide Stufen im Detail:

Eingabe

Statt den Benutzer mit einem Formular zu konfrontieren, welches ihn für jedes Kriterium um eine Auswahl bittet, entschied ich mich dazu, die Metasuche als einzelnes Textfeld zu implementieren, welches aus den Eingaben der Besucher mittels Mustererkennung die eingegebenen Kriterien ableiten kann. Priorität hatten dabei das verzögerungsfreie Feedback zur Eingabe noch während diese getätigt wird und eine hohe Flexibilität in der Eingabe der Parameter, welche verschiedenste Schreibweisen zu erkennen vermag.

Dies sollte sich als Herausforderung erweisen, da mit wachsender Anzahl gültiger Eingaben für Parameter die Wahrscheinlichkeit von Überschneidungen zwischen Parametern respektive versehentlicher Eingaben ansteigt.

Es entstand ein System, welches die im Suchfeld eingegebene Zeichenkette clientseitig via Javascript parst, erkannte Parameter in eine Datenbankabfrage fasst und zutreffende Ergebnisse zurückliefert.

ABB. 75 Gesamtheit der Kriterien mitsamt verborgener Parsing-Maske.



Die eingegebene Zeichenkette wird während der Eingabe bei Betätigen der Leertaste, Eingabetaste oder nach Verstreichen einiger Sekunden ohne Eingabe evaluiert. Während der Evaluation erkannte Parameter werden unterhalb des Eingabefelds in grafischer Form dargestellt. Ein Klick Auf den Such-Button beziehungsweise das Betätigen der Eingabetaste führt dann zum Absenden der generierten Anfrage.

Die Reihenfolge der eingegebenen Kriterien ist beliebig. Darüber hinaus sind für jedes Kriterium eine Anzahl von Schreibweisen zulässig, um eine möglichst große Diversität in Formulierungen zu berücksichtigen.

Die Folgende Kriterien werden von der Metasuche begriffen:

Prüfungsstufe

ABB. 76 Metasuche; Kriterium Prüfungsstufe



Die Angabe der Prüfungsstufe ist möglich in abgekürzter Form (»In«, »Ba«), ausgeschrieben (»Diplomarbeit«, »Vordiplom«) und im Zusammenhang mit der Teilprüfung (»Bachelorhauptthemen«, »Intermediatenebenthemen«).

Teilprüfung

ABB. 77 Metasuche; Kriterium Teilprüfung



Die Angabe der Teilprüfung ist möglich in abgekürzter Form (»HT«, »NT«), als auch ausgeschrieben (»Hauptthema«, »Nebenthemen«) und in Verbindung mit der Prüfungsstufe (»Masternebenthemen«).

Lehrgebiet

ABB. 78 Metasuche; Kriterium Lehrgebiet



Die Angabe des Lehrgebiets kann als Kürzel (»AV«) oder ausgeschrieben (»Service Design«) erfolgen.

Jahr

ABB. 79 Metasuche; Kriterium Jahr



Das Jahr kann als zwei- oder vierstellige Jahreszahl angegeben werden (»09«, »2013«). Darüber hinaus lassen sich Zeiträume angeben (»2004-12«, »seit 2010«, »bis 2012«, »03-05«).

Prüfer/in

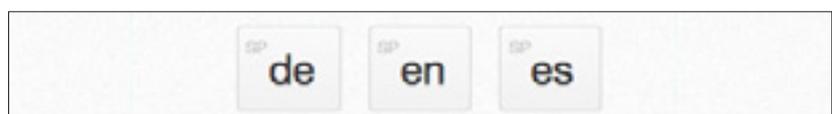
ABB. 80 Metasuche; Kriterium Prüfer/in



Der Angabe der prüfenden Person muss ein »bei« oder »von« vorangestellt werden. Darauf folgt der Name, der Titel ist optional (»bei Prof. Erhoff«, »geprüft von Prof. Hatto Grosse«).

Sprache

ABB. 81 Metasuche; Kriterium Sprache



Sprachangabe in Kurz- (»de«, »en«) oder Langform (»englisch«, »english«).

Verarbeitung und Ausgabe

Die Verarbeitung der Eingabe erfolgt während der Eingabe mit jeder Eingabetasten-Betätigung, Leerzeichen-Eingabe, Betätigen der Backspace-Taste oder bei zweisekündigem Verharren des Eingebenden. Diese Evaluation findet in JavaScript, also clientseitig statt, und umfasst die Untersuchung der im Meta-suchfeld eingegebenen Zeichenkette via regular expressions (regex).

ABB. 82 ark_search.php (Z.273ff)

```
273 //ark_record_exam_type
274 if (/^\b[I][Nn]\b|\b[Ii]ntermediate(\w*)\b/g.test(pointer)) {
    $('#aRET_ip').attr('selected', true);
    $('#aret_ip').addClass('active');
}
```

Das hier abgebildete Code-Fragment bildet eine regular expression ab, die die Zeichenkette (**pointer**) auf das Vorkommen von allen zulässigen einem Intermediate zugeordneten Schreibweisen hin überprüft. Sofern eine der Schreibweisen gefunden wird, selektiert die Evaluierungsfunktion ein verborgenes Auswahlfeld (**#aRET_ip**) innerhalb eines Formulars, welches im Verlauf der Funktion zur Generierung der Datenbankabfrage hinzugezogen wird. Das Element, welches dem Nutzer suggeriert, dass der Begriff identifiziert wurde (**#aret_ip**), wird währenddessen eingeblendet.

Für jede Prüfungsstufe, jede Teilprüfung, jede Sprache und jedes Lehrgebiet findet sich eine if-Abfrage samt regular expression wie die oben abgebildete innerhalb der Evaluierungsfunktion. Je nachdem, wie viele Schreibungen diese erkennen sollen, steigt die Komplexität der regex an, wie das folgende Beispiel zeigt:

ABB. 83 ark_search.php (Z.296ff)

```
296 if (/^\b[D][E]\b|\b[Dd][Mm]\b|\b[Dd]esign(\s?[UuAa]?n?d?\s?)
[Ee]conomy\b/g.test(pointer)){
297     $('#aRA_de').attr('selected', true);
    $('#ara_de').addClass('active');
298 }
```

Diese regex registriert sowohl das ehemalige Kürzel des Lehrgebiets (»**DM**«), als auch das aktuelle Kürzel in diversen Schreibweisen (»**DE**«), übrigens nicht in Kleinschreibung, da dieses mit der Sprach-Angabe kollidieren würde. Ebenfalls zulässig ist die Schreibweise des Lehrgebietsnamens mit »**and**« als auch »**und**« in Groß- oder Kleinschreibung.

Einen Sonderfall bildet die Erkennung des Prüfernamenten innerhalb der Meta-suchanfrage. Da das System derzeit 83 Prüfer enthält (Stand: Februar 2014), und jederzeit eine neue prüfungsberechtigte Person hinzukommen kann, war die Hardkodierung aller Prüfernamenten in Form von regular expressions keine viable Lösung. Es galt also, innerhalb der Evaluierungsfunktion die Datenbank abzufragen um Prüfer zu identifizieren und wiederum der Datenbankabfrage für die Ergebnisabfrage die ID des Prüfers liefern zu können. Ich führte also zwei Begriffe ein, namentlich die Wörter »(**geprüft**) von« und »**bei**«, welche via regex gesucht werden. Dies war notwendig, da nicht jeder Prüfer

ein »Prof« im Namen trägt, beziehungsweise nicht sicherzustellen ist, dass der Suchende den Titel – falls vorhanden – überhaupt eingibt. Wenn also ein »von« oder »bei« im Suchstring gefunden wird, so versucht das System, alle darauf folgenden Titel herauszufiltern und den darauf folgenden Namen mit der Prüferdatenbank abzugleichen:

ABB. 84 ark_search.php (Z.319ff)

```
319 if(\/b([Vv]on\s|[Bb]ei\s)([Pp]rof\.?e?s?s?o?r?i?n?\s)?([Dd]r.\?
s)?([A-Z][a-züäöß]+\s)?([A-Z][a-z]+\s)?([a-züäöß]+\s)?([a-züäöß]
{4,})?)\b/g.test(pointer)){
```

Die Abfrage bedient sich einiger Kniffe, etwa muss der Name mindestens vier Zeichen umfassen bevor die Datenbank befragt wird. Zum Erstaunen des Autors führen eine Vielzahl an Eingabevariationen (z.B.: »bei Erlhoff«, »bei Prof. Dr. Michael Erlhoff«, »bei Professor erlhoff«, »bei Michael«) zur korrekten Zuordnung des Prüfers. Da jedoch die Länge des Prüfernamenten bei diversen möglichen Schreibweisen nicht mit Sicherheit zu bestimmen ist, sollte dieser am Ende einer Anfrage stehen, um keine anderen Kriterien zu schlucken. Wie bereits angedeutet, durchläuft die Evaluierungsfunktion die vorgestellten Parameter und überprüft deren Vorkommen innerhalb der Suchmeta-Eingabe.

Für jedes zutreffende Kriterium wird dieses in ein unsichtbares Formular eingetragen, welches so ausgelegt ist, dass es je nach Kriterium nur einen oder mehrere Werte aufnimmt. Das Formular fasst die Kriterien bereits im Format, welches sie in der Datenbank einnehmen, Prof. Dr. Michael Erlhoff etwa würde im verborgenen Formular unter seiner ID (118) geführt werden. Gleichzeitig wird dem Nutzer visuell angezeigt, welche Werte im verborgenen Formular gesetzt sind. Wenn dieser nun die Suchabfrage abschickt, werden die dort abgelegten Werte sowie der Parameter **offset** (aktuelle Seite der Suchergebnisse) sowie die aktive Mehrfachansicht **ark_view** via AJAX-Abfrage an eine ausgelagerte Funktion (**ark-get-adv-search-results.php**) übergeben, die via Datenbankabfrage alle zutreffenden Einträge ermittelt und sie paginiert und in der korrekten Mehrfachansicht ausgibt.

HINWEIS Die Eingabe von '**meta [Anfrage]**' im regulären Suchfeld leitet die Suche die Anfrage zur Metasuche weiter.

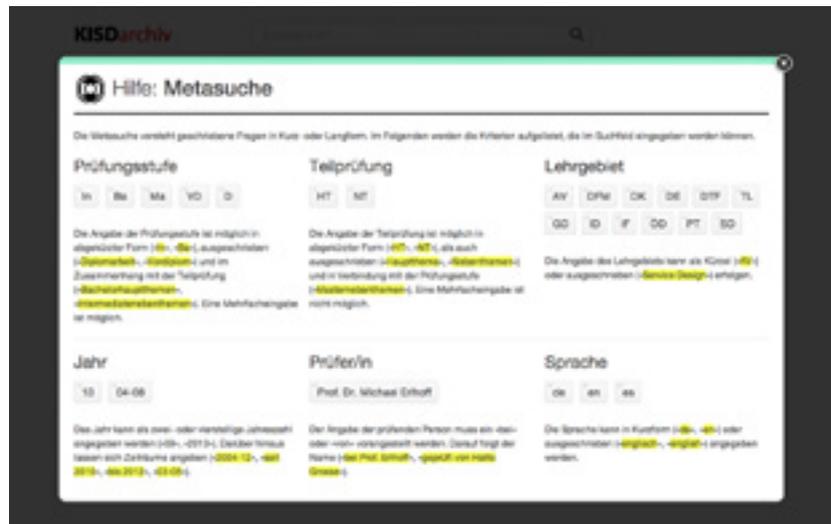
HINWEIS Die Eingabe von '**Zeige mir alles**' erwirkt die Ausgabe sämtlicher Einträge.

HINWEIS Die Eingabe einer leeren Anfrage blendet das Hilfefenster ein.

10.10.5 Metasuche / Kontextuelle Hilfe

Überlagertes Hilfefenster mit kontextuellen Tipps und Tricks.

ABB. 85 Hilfefenster innerhalb der Seite **Metasuche**



Die Metasuche besticht durch ein sehr reduziertes Interface, welches abseits gewohnter Paradigmen operiert. Um die Nutzer an dieses ungewohnte Interface zu gewöhnen, implementierte ich ein kontextuelles Hilfefenster, welches per Knopfdruck eingeblendet werden kann und immer dann in Erscheinung tritt, wenn die Metasuche ohne eingegebene Kriterien abgesendet wird.

Das Hilfefenster wird via fancybox-Plugin eingeblendet, es orientiert sich am selben Raster wie die unterliegende Seite und passt sich wie diese auch dem zur Verfügung stehenden Platz im Viewport an. Innerhalb des Hilfefensters wird eine Übersicht aller Kriterien dargestellt, welche im Metasuchfeld eingegeben werden können, sowie zu jedem Parameter eine Übersicht der gültigen Eingaben dargestellt.

Aus der kontextuellen Hilfe heraus entstand die Idee, eine zentrale Hilfeseite einzurichten, welche zum einen eine Art FAQ (Frequently Asked Questions) enthält, zum anderen die modulspezifischen Hilfe-Einträge zentral zugänglich macht. Diese zentrale Hilfe wird zu einem späteren Zeitpunkt implementiert werden und Nutzern die Möglichkeit geben, Fragen zum KISDarchiv direkt über ein Formular zu stellen. Darüber hinaus kann eine Feedback-Funktion innerhalb der Eintrags-Einzelansicht AG-fremden Nutzern die Möglichkeit bieten, Korrekturen und Ergänzungen zu Einträgen einzureichen, und die Datenpflege somit teilweise der Studierendenschaft anvertrauen.

Diese Ansätze sollen an dieser Stelle jedoch nicht vertieft werden, da sie thematisch bereits in den dritten Teil dieser Arbeit fallen.

10.10.6 Lesezeichen

Die Möglichkeit, Einträge zum Zwecke der zeitversetzten Lektüre oder der Weiterempfehlung einer Merkliste hinzuzufügen.

ABB. 86 Lesezeichen-Seite



Die Möglichkeit, Arbeiten beim Durchstreifen des KISDarchivs einer Merkliste hinzuzufügen, wurde bereits im Rahmen des SI »Arche« skizziert.

Die Funktionalität der »Lesezeichen«, wie die Einträge auf einer solchen Liste innerhalb des KISDarchiv bezeichnet werden, ist in diesem Modul gefasst.

Elemente

Das Lesezeichen-Modul besteht aus folgenden Bestandteilen:

Lesezeichen-Seite

Die Lesezeichen-Seite enthält die Lesezeichenliste, welche alle zu einem bestimmten Zeitpunkt mit Lesezeichen versehenen Einträge anzeigt. Neben ihrer Funktion als Übersichtsseite dient diese Seite als Portal zur PDF-Ansicht.

Lesezeichen-Cookie

Der Lesezeichen-Cookie bildet nutzerseitig eine Liste aller Einträge ab, die mit Lesezeichen versehen wurden.

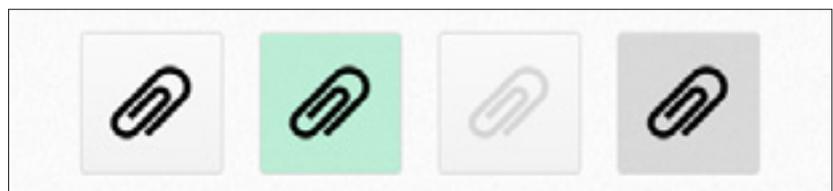
PDF-Ansicht

Die PDF-Ansicht ist von der Lesezeichen-Seite aus erreichbar und zeigt die denen auf der Liste befindlichen Einträgen zugeordneten PDF-Dateien an. Die Ansicht erlaubt dabei das Hin- und Her Springen zwischen Arbeiten auf der Liste und das Entfernen von Einträgen von derselben.

Lesezeichen-Buttons

Lesezeichen-Buttons werden im Kontext der Mehrfachansichten dargestellt.

ABB. 87 Lesezeichen-Button;
Zustände (v.l.n.r.):
deaktiv,
aktiv,
deaktiv (PDF-Datei fehlt),
aktiv (PDF-Datei fehlt)



Per Klick auf den Lesezeichen-Button wird der zum Button gehörige Eintrag mit einem Lesezeichen versehen, oder von der Lesezeichenliste entfernt, sofern diese bereits auf dieser vertreten ist. Anhand einer farbigen Hinterlegung zeigt der Lesezeichen-Button an, ob der Eintrag bereits auf der Lesezeichen-Liste vorhanden ist. Eine graue Hinterlegung weist überdies darauf hin, dass die den Lesezeichen hinzugefügte Arbeit über keine PDF-Datei verfügt.



ABB. 88 Lesezeichen-Indikator

Lesezeichen-Indikator

Der Lesezeichen-Indikator fungiert als dynamischer Link zur Lesezeichen-Seite, sobald diese mindestens einen Eintrag enthält. Er erscheint in der oberen rechten Bildschirmcke in Form eines farbig hinterlegten Buttons, der die Anzahl derzeit auf der Lesezeichen-Liste befindlicher Einträge darstellt.

Funktionsweise

Die Lesezeichenliste wird userseitig in einem Cookie namens **ark_queue** gespeichert und besteht aus einer kommaseparierten Liste von Eintrags-IDs. Der Cookie selbst ist mit einer Gültigkeitsdauer von 31 Tagen versehen. Darüber hinaus kann der Zustand der Liste in Linkform gespeichert werden. Siehe **ark_queue** (S. 158).

In der ersten Iteration der Lesezeichenfunktion resultierte ein Klick auf den Lesezeichen-Button einer Arbeit im Übergeben einer HTTP POST-Variable an den Server, welche einen Parameter ('add'/'remove') und die ID des entsprechenden Eintrags enthielt. Der Cookie wurde dann – sofern vorhanden – vom Client ausgelesen, die Operation durchgeführt und der Cookie wieder geschrieben. Negativ bemerkbar machte sich bei dieser Lösung die Wartezeit, die mit dem erneuten Laden der Seite beim Klicken auf einen Lesezeichen-Button verbunden war. In der zweiten Iteration setzte ich also auf eine JavaScript-gestützte Implementierung der Lesezeichen-Funktion, welche mir den asynchronen Zugriff auf den Lesezeichen-Cookie ermöglichte. Dies macht sich bei der Lesezeichenvergabe dadurch bemerkbar, dass sich nach Betätigung des Lesezeichen-Buttons keinerlei Wartezeit mehr ergibt.

Fragestellungen

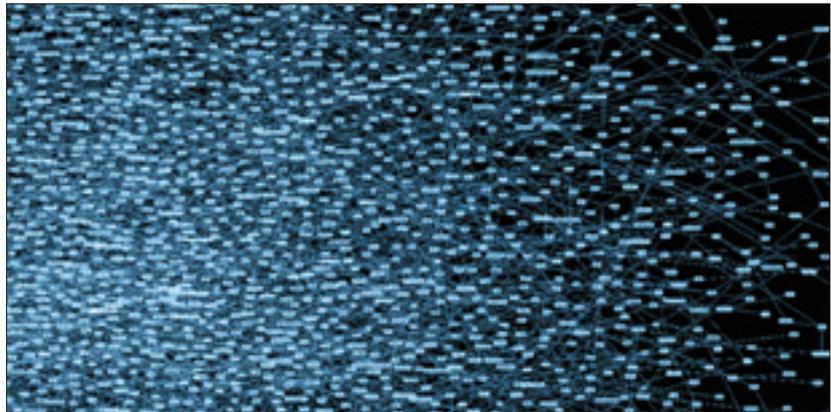
Einige der Fragen, die sich während der Modulentwicklung stellten:

- Die seitenübergreifende Verwaltung von Lesezeichen gestaltete sich als Herausforderung, da Inhalte etwa in der Metasuche live generiert werden, was die asynchrone Lesezeichenverwaltung vor gewisse Herausforderungen stellte.
- Der Umgang mit Einträgen, die über keine PDF-Dateien verfügen, war ein weiterer Diskussionspunkt. Letztendlich wurde eine Lösung gefunden, die erlaubt, diese den Lesezeichen hinzuzufügen. Der Lesezeichen-Button signifiziert an dieser Stelle durch eine graue Einfärbung, dass keine PDF-Datei vorliegt. In diesem Kontext war die Verknüpfung von Lesezeichen-Seite und PDF-Ansicht Thema. Hier wurde eine Lösung implementiert, welche das Aufrufen von PDFs aus der Eintragsansicht heraus erlaubt, ohne diese im Voraus mit Lesezeichen zu versehen.
- Der Umgang mit Lesezeichen-Links, die bei bereits bestehenden Lesezeichen geöffnet werden, wurde über ein Dialogfenster gelöst. Diese bietet an, zwei Sets von Lesezeichen zu kombinieren, oder eines der beiden zu bevorzugen während das andere verworfen wird.

10.10.7 Schlagworte

ABB. 89 Konnektivitätsdiagramm des KISDarchivs (März 2013)

Abgebildet sind etwa 4831 Relationen zwischen 2246 Einträgen und Schlagwörtern im Datenbestand (Ausschnitt).



Mit dem Datenimport (siehe Seite 48) wurden, wie bereits geschildert, alle Schlagworte aus dem Datenbestand der AMD in die WordPress Schlagwort-Taxonomie übernommen. Das KISDarchiv zählt derzeit etwa 9000 Schlagwörter. Die Abbildung am Seitenende zeigt die Verteilung des Schlagwortbestands auf den Eintragsbestand. 6812 Schlagwörter sind lediglich einem Eintrag zugeordnet. Durchschnittlich kommen auf ein Schlagwort 1,85 Einträge.

Schlagwörter erfüllen im KISDarchiv eine Doppelrolle:

- **Relation:** Schlagworte bilden Sinnverknüpfungen zwischen Einträgen: Sie erlauben dem Nutzer, in der Eintragsansicht verwandte Einträge zu sehen. Umgekehrt bilden Einträge Verknüpfungen zwischen Schlagworten: In der Schlagwortübersicht sieht der Nutzer alle einem Schlagwort zugeordneten Einträge und wiederum deren Verschlagwortung.
- **Suchbegriff:** Schlagworte werden neben diversen anderen Metadaten im Suchindex geführt und erfüllen damit die wichtige Funktion, Einträge auffindbar machen. Bedenkt man den Stellenwert der Suchfunktion innerhalb der diversen Module des KISDarchivs, so sollte diese Rolle immer in die Bewertung jeglicher Maßnahmen auf dem Schlagwortbestand einbezogen werden.

ABB. 90 Einträge pro Schlagwort (Februar 2013)

X-Achse:
Einträge pro Schlagwort
Y-Achse:
Schlagwort-Anzahl
(logarithmische Skala)

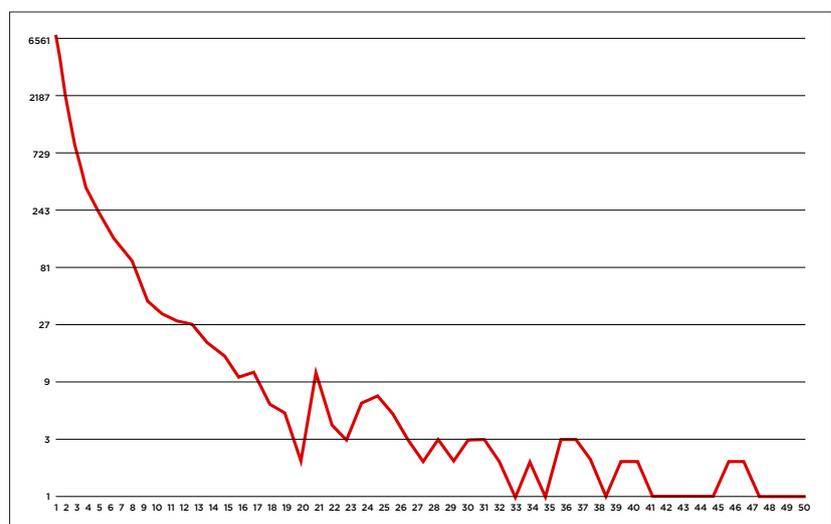


ABB. 91 Schlagwortdiagramm zum Begriff »Typografie« (März 2013)

schwarz: Mit »Typografie« verschlagwortete Einträge

grau: Schlagworte der verschlagworteten Einträge (1. Grad)

grün: Relationen zwischen Schlagworten ersten Grades



Was macht ein gutes Schlagwort aus?

Die Frage ist schwierig zu beantworten, da Schlagworte nicht einzeln sondern nur in ihrer Konnektivität mit Einträgen und darüber mit anderen Schlagworten qualifiziert werden. Überdies ändern sich mit einem wachsenden Datenbestand die Anforderungen an den Schlagwortbestand.

Es ließe sich argumentieren, dass ein Schlagwort nicht zu speziell gefasst sein sollte. Wie sich jedoch zeigt, bietet ein Datenbestand von über viertausend Einträgen durchaus Potenzial für Konnektivität bei exotischeren Begriffen wie Mass Customization (5), Milch (3) oder Chronobiologie (3). Selbst ein Schlagwort, welches keine Verbindung zwischen Einträgen bildet, kann durchaus das Potenzial dazu haben. Der optimal verschlagwortete Eintrag sollte demnach eine Mischung aus eher generell gefassten und speziellen Begriffen abbilden um der Doppelrolle der Schlagwörter im KISDarchiv gerecht zu werden.

Optimieren lässt sich der Schlagwortbestand durch die Umsetzung gewisser syntaktischer Regeln, wie etwa der Verwendung von Begriffen im Singular. Dies erhöht die Konnektivität und damit den Nutzen der Schlagwort-Seiten.

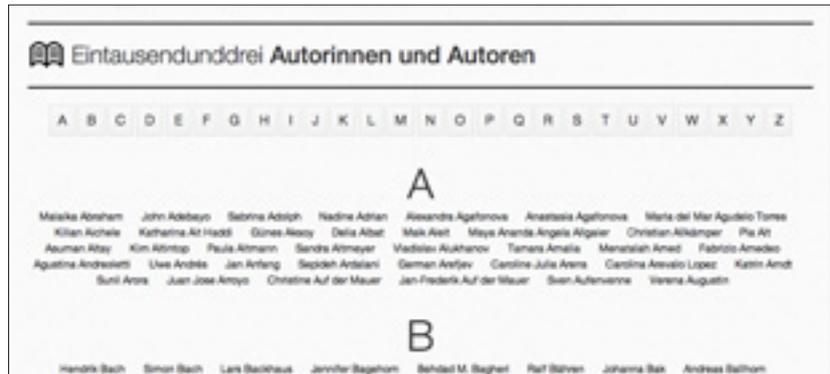
Von größeren Eingriffen in den Schlagwortbestand ist jedoch abzuraten. Immerhin wurde dieser von Designstudierenden formuliert und sollte somit bereits in gewissem Maße auf das Denken der Nutzerbasis zugeschnitten sein.

10.10.8 Autorenindex

Übersicht aller Autorinnen und Autoren

Es erschien naheliegend, eine Darstellung zu schaffen, welche einen Zugang zu sämtlichen Einträgen über die ihnen zugeordneten Autoren zulässt. Dies wurde in Form des Autorenverzeichnisses alias »Index« umgesetzt.

ABB. 92 Autorenverzeichnis

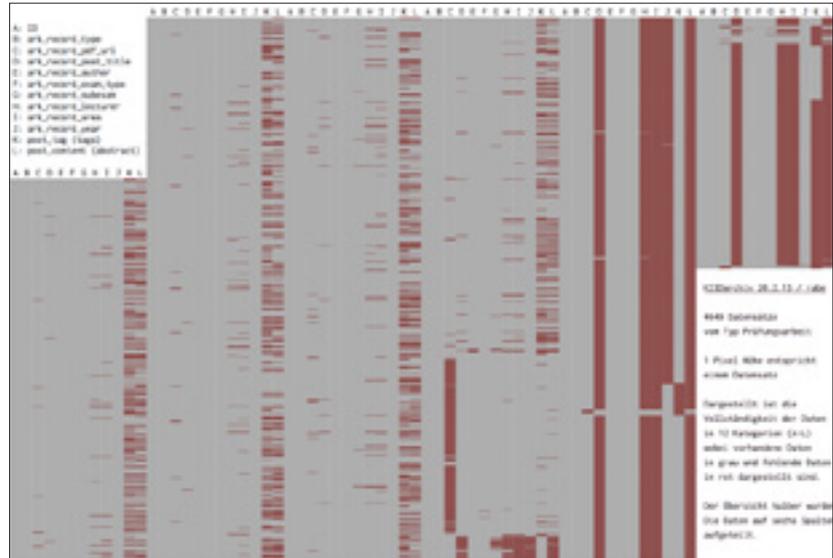


Der Index bildet ein Verzeichnis sämtlicher im KISDarchiv vertretenen Autorinnen und Autoren ab. Diese werden nach Nachname alphabetisch sortiert ausgegeben wobei eine statische horizontale Navigation das schnelle Navigieren zwischen den einzelnen Lettern ermöglicht. Diese können ebenso per Eingabe des jeweiligen Buchstaben auf der Tastatur angesteuert werden. Die Namen verlinken auf die jeweilige Autoren-Einzelansicht.

10.10.9 Integrität

Das Modul »Integrität« setzt sich aus einer Seite im Frontend und einem Backend-Widget zusammen.

ABB. 93 Integritätsdiagramm: Übersicht der Integrität der im KISDarchiv vertretenen Datensätze nach 12 Kategorien. Stand: 20.2.2013



Die Frage nach der Eintragsintegrität, also der Vollständigkeit der Metadaten der im KISDarchiv abgelegten Arbeiten, stellte sich erstmals im Rahmen des Imports der Altdaten aus dem AdA-Bestand. Die Eintragsintegrität zu bestimmen, erwies sich sowohl auf der **Makroebene** (»Wie ist es um die Vollständigkeit des Gesamtdatenbestands bestellt?«) als auch auf der **Mikroebene** (»Wie vollständig ist der mir vorliegende Eintrag?«) als sinnvoll.

Integritätsberechnung

Die quantitative Bestimmung der Metadaten-Vollständigkeit alias »Integrität« eines Eintrags bedeutet, einen Zustand der Sättigung festzulegen, bei dessen Erreichen ein Eintrag als vollständig gilt. Im ersten Schritt wählte ich dafür zehn Metadatenfelder aus, die für die Berechnung herangezogen werden. Diese gehen mit gleicher Gewichtung in die Bestimmung ein, jedes der Felder kann also maximal 10% zur Eintrags-Integrität beitragen:

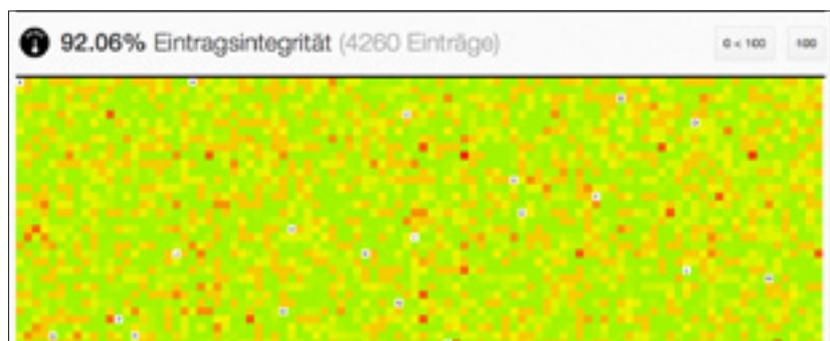
	DATENFELD	GEWICHTUNG (MAX. 10%)	KRITERIUM
PRÜFUNGSSTUFE	ark_record_exam_type	10 %	Vorhandensein
TEILPRÜFUNG	ark_record_subexam	10 %	Vorhandensein
LEHRGEBIET	ark_record_area	10 %	Vorhandensein
JAHR	ark_record_year	10 %	Vorhandensein
PRÜFER	ark_record_lecturer	10 %	Vorhandensein
SPRACHE	ark_record_language	10 %	Vorhandensein
PDF	ark_record_pdf_url	10 %	Vorhandensein
AUTOR	ark_record_author	10 %	Vorhandensein
ABSTRACT	post_content	0,1 % pro Zeichen	min. 100 Zeichen
SCHLAGWORTE	post_tag	2 % pro Schlagwort	min. 5 Schlagwörter

Die Berechnung folgt bei den ersten acht aufgeführten Feldern einer binären Logik, das Vorhandensein eines Werts wird hier als Erfüllung des Kriteriums bewertet. Das Abstract hingegen wird ab 100 Zeichen Länge als ausgefüllt. Ab fünf vorhandenen Schlagwörtern werden diese als vollständig bewertet.

Wie bereits geschildert, findet keine qualitative Überprüfung der Daten statt: Hundertprozentige Integrität bedeutet nicht, dass das Abstract aussagekräftig ist, oder die Schlagwörter gut gewählt wurden. Ebenso garantiert es nicht die Korrektheit der eingegebenen Daten. Die Bestimmung der Integrität ist somit als bloßer Richtwert zu begreifen. Versteht man diese der Integritätsberechnung zugrunde liegenden Einschränkungen, kann diese eine wertvolle Ergänzung des AG-Toolkits darstellen.

Integritätsanzeige, Frontend

ABB. 94 Integritätsseite im Frontend (Ausschnitt)



Die Integritätsanzeige im Frontend bildet auf einer Bildschirmseite die Vollständigkeit sämtlicher vorhandenen Einträge ab. Sie orientiert sich damit am Integritätsdiagramm auf der vorherigen Seite. Für die Seitendarstellung bei rund 4000 Datensätzen werden immerhin 50000 Metadaten ausgelesen.

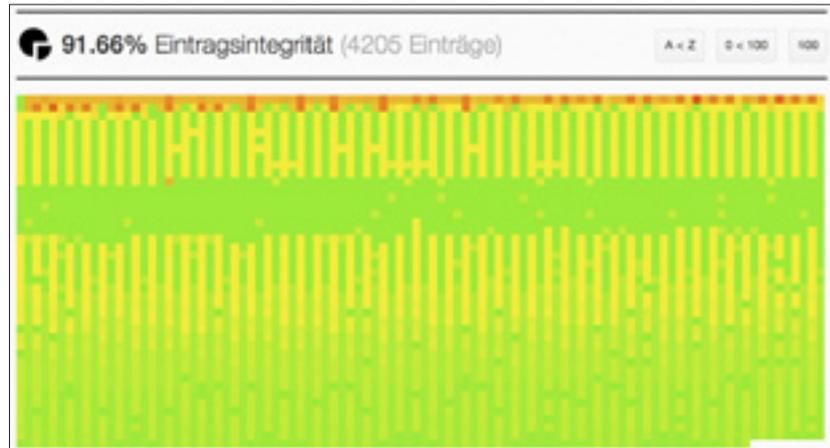
Jeder einzelne Eintrag bildet ein Quadrat mit einer Kantenlänge von 12 Pixeln welches je nach Vollständigkeit der zugrunde liegenden Eintrags-Metadaten eingefärbt ist – 50% Eintrags-Integrität entspricht einer Rotfärbung, Einträge mit 100% Integrität erscheinen in einem kräftigem Grün.

Fährt der Benutzer mit der Maus über einen der Einträge, wird in einem Tooltip der Titel und die prozentuale Integrität des Eintrags angezeigt. Ein Klick auf den Eintrag öffnet diesen im Backend (sofern der Besucher die Berechtigung besitzt, diesen zu editieren), beziehungsweise im Frontend (für den Fall, dass ebendiese Berechtigung nicht vorliegt).

Standardmäßig werden die Einträge alphabetisch nach Titel sortiert gezeigt. Darüber hinaus besteht die Möglichkeit, über zwei Buttons in der Headline alle Einträge mit hundertprozentiger Integrität auszublenden und/oder die dargestellten Einträge nach Vollständigkeit zu sortieren.

Die Reduktion der Darstellung auf einen farbigen Pixel gegenüber der differenzierteren Darstellung des auf der vorigen Seite abgebildeten Diagramms soll die Lesbarkeit vereinfachen und unterstreicht die quantitative Natur der unterliegenden Berechnung.

ABB. 95 Die Integritätsseite schafft Pixel Art; ein Bug im Code führt zu einer inkorrekten, wenn auch visuell betörenden Sortierung (20.10.2013)



Integritätsmetabox (IMB), Backend

Die Integritätsmetabox wurde geschaffen, um dem Bearbeiter eines Eintrags im Backend ein schnelles Überblicken der Vollständigkeit der Metadaten zu ermöglichen. Sie lässt sich in der Eintrags-Bearbeitungsansicht einblenden.

ABB. 96 Drei Integritätsmetaboxen im Vergleich. (Collage)



Da die Eintrags-Bearbeitungsmaske sich mit gemessenen 2670 Pixeln Höhe über einen nur durch Scrollen zu überblickenden Bereich erstreckt, erfüllt die IMB eine nützliche Funktion für den geneigten Bearbeiter: Einen quantitativen Überblick über die Eintragsmetadaten.

Aufbau

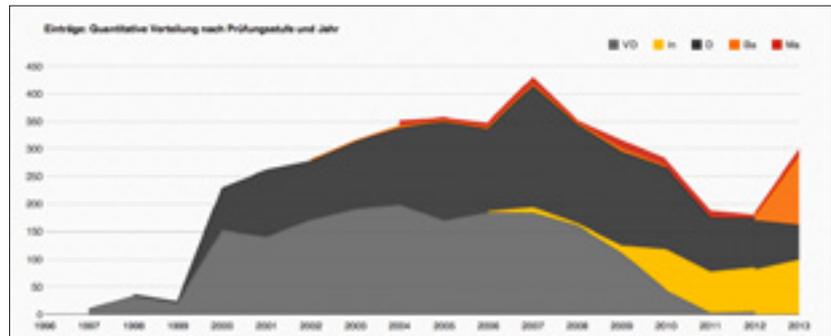
Das auffälligste Element der IMB bildet ein grafischer Fortschrittsbalken, welcher die Integrität des Eintrags abbildet. Im Falle einer Integrität unter 100% findet sich unterhalb des Balkens eine Auflistung der fehlenden Angaben in Form von Links, welche die auszufüllenden Formularelemente referenzieren. Die IMB berechnet den Integritätswert mit Abspeichern des Eintrags und dem damit verbundenen Aktualisieren der Seite neu.

10.10.10 Statistik

Diagramm-Darstellung des Datenbestands

Die Schaffung einer Diagrammdarstellung wurde aus dem Wunsch heraus geboren, dem digitalen Archiv-Besucher ein Gefühl dafür zu vermitteln, wie die quantitative Verteilung der Einträge nach Prüfungsstufe und Jahr aussieht. Dies bildet gewissermaßen die digitale Entsprechung zum Blick auf die reich gefüllten Regale des analogen Archivs.

ABB. 97 Statistikseite im Frontend (Ausschnitt)

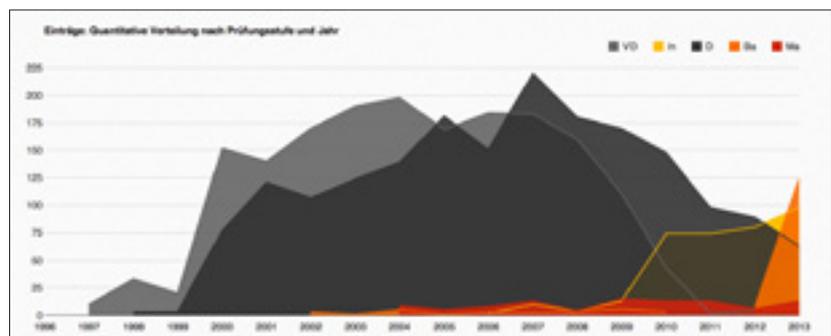


Das Modul Statistik bietet eine grafische Darstellung des gesamten Eintrags-Datenbestands in Form eines Flächendiagramms. Die Einträge werden dabei nach Prüfungsstufe geclustert und auf die Zeitachse geplottet. Die Farbigkeit der Flächen entspricht den Prüfungsstufen zugewiesenen Farben. Siehe **Eintrags-Farbkodierung** (S. 67). Ausnahme bilden die Teilprüfungen »Vordiplom« und »Diplom« des Diplomstudiengangs Design, welche zur besseren Differenzierung in Graustufen dargestellt werden.

Die Anordnung der Flächen wurde so gewählt, dass korrespondierende Prüfungsstufen an einander angrenzen. So lässt sich erkennen, wie die Prüfungsstufe »Vordiplom« in die Prüfungsstufe »Intermediate« übergeht.

Ein rechts oben in der Headline positionierter Button bietet die Möglichkeit, von der kaskadierten auf eine überlagerte Flächendarstellung zu wechseln.

ABB. 98 Statistikseite im Frontend, Alternative Darstellung (Ausschnitt)



Die Darstellung aktualisiert sich selbstverständlich mit jeder Veränderung im Datenbestand selbst, da die Daten mit jedem Aufruf aus der Datenbank bezogen werden. Die Achsen skalieren sich automatisch wenn die Maximalwerte überschritten werden sollten.

10.10.11 Arbeit einreichen

Arbeiten ins aus dem Frontend ins KISDarchiv einstellen.

ABB. 99 Arbeit einreichen-Maske, Frontend (Januar 2013)

Das Arbeit einreichen-Modul bildet die Antwort auf zwei Fragen, die auf den ersten Blick nicht die selbe Antwort haben sollten:

- Wie, wenn überhaupt, können Studierende Arbeiten ins KISDarchiv eingeben?
- Wie können Mitglieder der Archiv-AG Arbeiten ins KISDarchiv eingeben?

Um den Hintergrund dieser Fragen besser zu begreifen, betrachten wir an dieser Stelle den bisherigen Modus Operandi:

Alter Abgabevorgang (bis 2013)

1. Die Prüflinge reichten zum Abgabetermin ihre Arbeit in mehrfacher Ausführung (Prüferexemplar(e), Archivexemplar) samt digitaler Kopie (auf CD-ROM)

und Abgabeformular (Note) bei Frau Lindner im Sekretariat ein.

2. Die gesammelten Archivexemplare samt CD-ROM und Note wurden an die Archiv-AG übergeben.
3. Die Archiv-AG sortierte die Archivexemplare samt gedrucktem Reiterschild in eine Hängetasche, pflegte Arbeit und Note digital ins AdA ein, die im Note enthaltenen Metadaten in die (neue) Metadatenbank.
4. Die Archiv-AG vernichtete die Notes sowie die CD-ROMs, benachrichtigte gegebenenfalls Prüflinge, deren Daten nicht vorlagen und bat diese um Einsendung derselben.

Reflektion zum Abgabevorgang

Betrachten wir den bisherigen Abgabevorgang, so wird schnell klar, dass dieser einen enormen Aufwand materieller und personeller Ressourcen bedeutet. Die mehrfache und medienübergreifende Übertragung von (Meta-)daten birgt hohes Fehlerpotenzial.

Einzel- versus Batch-Eingabe

Die Eingabe hunderter oder gar tausender Einträge in einem Schwung ist bezeichnend für die Art und Weise, wie die Altdaten ins KISDarchiv überführt wurden. Die Eingabe eines ganzen Jahrgangs an Prüfungsarbeiten ließe sich ebenfalls als Batch-Eingabe charakterisieren. Während diese Art der Eingabe ihre Daseinsberechtigung hat, ist nicht daran zu denken, eine solche Eingabe für den Regelbetrieb des KISDarchivs zu implementieren. Dies hat den Grund, dass der Batch-Import die Daten in Tabellenform einliest, ohne einzelne Eingaben zu validieren. Wir hätten es also mit einem sehr fehleranfälligen System zu tun.

Begreifen wir die Abgabe als »Erfassen neuer Einträge« unabhängig von Kontext und einpflegender Person, so wird ein Eingabemodus denkbar, welcher AG-Mitgliedern und Prüflingen gleichermaßen zur Verfügung steht. Dieser ist also im Frontend zu verorten. Eine solche Universallösung hat den Vorteil, dass sie einfacher zu warten ist, und durch ihre Form als reine Eingabemaske gegenüber den Eingabe- und Bearbeitungsmasken des Backends eine strengere Validierung erzwingen kann. Dies hat den einfachen Grund, dass die Auf-erlegung einer Validierung auf noch zu erfassender Daten – im Rahmen der Eingabe – sich sehr viel einfacher gestaltet, als die nachträgliche Validierung bereits bestehender Einträge im Rahmen der Bearbeitung.

Die Abgabe lässt sich grob in zwei Schritte unterteilen:

1. Die **Eingabe** von Metadaten in Verbindung mit dem Upload einer PDF-Datei
2. Die **Überprüfung**, etwaige Korrektur und Freischaltung der Metadaten.

Unterziehen wir die zwei Schritte einer detaillierten Betrachtung:

Eingabe von Metadaten und PDF-Upload

Der Eingabevorgang lässt sich illustrieren, indem wir die Eingabefelder der Arbeit einreichen-Maske in ihrer sequenziellen Anordnung besprechen.

1. Titel

ABB.100 Titel-Eingabefeld

Dieses Feld fasst den Titel der Arbeit. Da dieser keinen syntaktischen Einschränkungen unterliegt, wird das Feld lediglich auf das Vorhandensein von mindestens zwei Zeichen überprüft

TYP	Textfeld
ASSOZIATION	post_title
VALIDIERUNG	Das Feld wird als ausgefüllt betrachtet, wenn es mindestens 2 Zeichen enthält.

2. Untertitel

ABB.101 Untertitel-Feld

Das Untertitelfeld stellt kein Pflichtfeld dar. Die hier getätigte Eingabe wird mitsamt der Eingabe des Titelfelds delimitersepariert in das **post_title**-Feld geschrieben. Siehe **ark_record / Eintrag** (S. 115).

TYP	Textfeld
ASSOZIATION	post_title
VALIDIERUNG	-

3. Abstract

ABB.102 Abstract-Feld

Das Abstract-Feld ist in Form einer Textarea gehalten. Diese überprüft bei erfolgter Eingabe, ob mindestens 100 Zeichen eingegeben wurden, was als Mindestlänge für das Abstract definiert wurde.

TYP	Textfeld
ASSOZIATION	post_content
VALIDIERUNG	Das Feld muss mindestens 100 Zeichen enthalten.

4. Schlagwörter

ABB. 103 Schlagwörter-Feld

Das Schlagwort-Feld akzeptiert eine beliebige Anzahl, mindestens jedoch fünf mit Komma abgetrennte Begriffe. Die Begriffe können von Hand eingegeben oder als kommaseparierte Liste ins Feld kopiert werden. Das Feld lässt keine Doppeleingaben von Schlagworten zu.

TYP	tagsInput-Feld
ASSOZIATION	post_tag
VALIDIERUNG	mindestens fünf Schlagworte müssen eingegeben werden

5. Autor/innen

ABB. 104 Autor/innen-Feld

Das Autor/innen-Feld ist via Autocomplete-Funktion sowohl mit der Autoren-Datenbank des KISDarchivs verbunden, als auch mit einer externen Datenquelle welche aktuell Studierende mitsamt ihrer Metadaten enthält. Sobald der Nutzer beginnt, einen Namen einzugeben, prüft das Feld, ob dieser sich bereits in der KISDarchiv-eigenen Autoredatenbank befindet. Ist dies nicht der Fall, werden die Autoredaten aus der externen Datenquelle herangezogen, um den Benutzer neu anzulegen. In jedem Fall sollte ein von der Autocomplete-Funktion angebotener Name in dieses Feld geschrieben werden, da ein beliebiger – also weder in der Autoredatenbank noch in der externen Datenbank vorhandener – Name vom System nicht zugeordnet werden kann. Dieser müsste folglich von der AG manuell angelegt werden. Das Feld erlaubt die Eingabe mehrerer Autoren und beschränkt diese auch nicht, da zwar davon auszugehen ist, dass unter gewissen Bedingungen – etwa beim Ablegen eines Intermediate-Nebenthemas – nur eine Person involviert ist, die Praxis jedoch gezeigt hat, dass selbst solche Arbeiten von mehr als einer Person eingereicht werden. Eine strenge Validierung würde die Eingabe solcher Arbeiten verhindern und die Daten verfälschen. Mit der Auswahl eines Autors wird der diesem zugeordnete Datensatz, genauer gesagt das Feld **ark_author_degree** ausgelesen. Kann der Autor auf diesem Wege mit einem Studiengang assoziiert werden, so wird der entsprechendem Studiengang im folgenden Feld bereits ausgewählt.

TYP	tagsInput-Feld mit Autocomplete
DATENQUELLEN	Studierenden-Datenbestand; externe Studierendendatenbank
ASSOZIATION	ark_record_author
VALIDIERUNG	-
FOLGE	Automatische Studiengangsauswahl sofern dem Autor ein Studiengang zuzuordnen ist (via ark_author_degree).

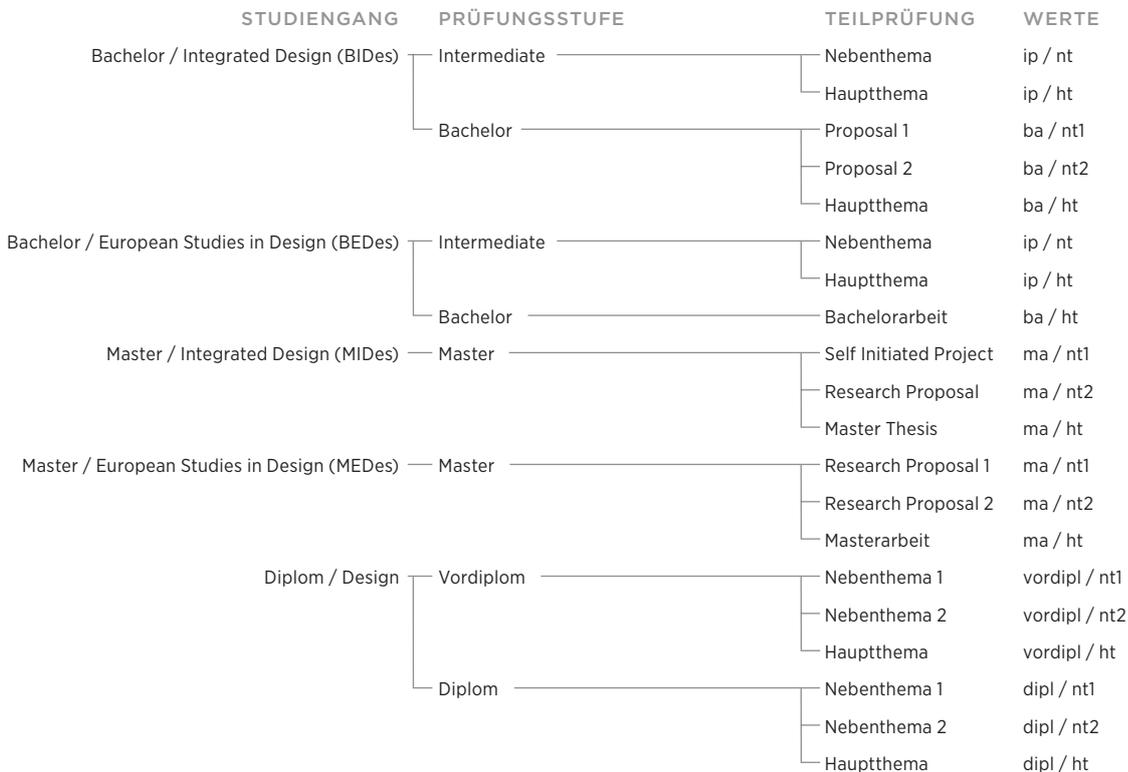
6. Studiengang / Prüfungsstufe / Teilprüfung

ABB.105 Studiengang-, Prüfungsstufe- und Teilprüfungs-Felder



Die drei Dropdown-Felder bedingen einander und werden deshalb an dieser Stelle zusammen erläutert.

Die Validierung jedes einzelnen dieser drei Felder ist dadurch gewährleistet, dass es sich um Dropdownfelder mit vorgegebenen Optionen handelt. Jedoch ergeben sich aus der Auswahl des Studiengangs nur bestimmte zulässige Eingaben für die Prüfungsstufe und innerhalb einer Prüfungsstufe wiederum nur bestimmte Auswahlmöglichkeiten für eine Teilprüfung, welche zudem von Studiengang zu Studiengang unterschiedlich benannt sein kann. Es musste also eine Reihe von konditionalen Dropdowns geschaffen werden, welche sich lediglich sequenziell ausfüllen lassen und welche mit jeder Auswahl die Folgefelder mit zulässigen Werten bestückt und zur Auswahl freigibt. Die folgende Struktur gibt einen Überblick über alle möglichen, also auswählbaren Kombinationen der drei Felder und der resultierenden Werte:



Im Diagramm sehen wir einige Kombinationen, namentlich die Master-Studiengänge, welche jeweils nur über eine auswählbare Prüfungsstufe verfügen. Wird einer dieser Studiengänge ausgewählt, so aktiviert die Maske direkt das Teilprüfungsdropdown und erspart dem Nutzer die Prüfungsstufenauswahl.

Selbiges gilt für den Fall, dass ein Nutzer innerhalb des Studiengangs BEDes die Prüfungsstufe Bachelor auswählt: Da dieser nur über eine Teilprüfung verfügt, wird diese bereits für den Nutzer vorausgewählt und zum nächsten Feld, also der Auswahl des Prüfers, gesprungen.

Um Fehleingaben zu verhindern, verfügen diese drei Felder zudem über eine rückwärtige Prüfroutine: Wird nach Auswahl der Teilprüfung die Auswahl des Studiengangs verändert, setzt die Maske nicht mit dem neu ausgewählten Studiengang konforme Prüfungsstufen- und Teilprüfungs-Werte zurück.

TYP	Konditionale Dropdown-Felder
DATENQUELLEN	hardcoded (ark-upload.php)
ASSOZIATION	ark_author_degree / ark_record_exam_type / ark_record_subexam
VALIDIERUNG	diverse Prüfroutinen, s. o.

9. Prüfer/in

ABB. 106 Eingabefeld Prüfer/in

Die Eingabe von Prüfer/in erfolgt wie auch die Autoreneingabe über ein Feld mit Autocomplete-Funktion. Dieses liest via Datenbankabfrage alle im KISD-archiv vertretenen Dozenten aus. Bei Auswahl der prüfenden Person aus der vom Autocomplete-Feature generierten Liste wird das dieser Person zugeordnete Lehrgebiet im folgenden Feld vorausgewählt.

TYP	tagsInput-Feld mit Autocomplete
DATENQUELLEN	Dozenten-Datenbestand
ASSOZIATION	ark_record_lecturer
VALIDIERUNG	-
FOLGE	Automatische Lehrgebietsauswahl sofern der prüfenden Person ein Lehrgebiet zugeordnet ist (via ark_lecturer_area).

10. Lehrgebiet

ABB. 107 Eingabefeld »Lehrgebiet«

Falls das Lehrgebiet nicht bereits durch Auswahl eines mit einem Lehrgebiet assoziierten Prüfer vorausgewählt wurde, werden an dieser Stelle alle Lehrgebiete angezeigt, deren Anzeigen-Flag (**ark_area_show**) gesetzt wurde.

TYP	tagsInput-Feld mit Autocomplete
DATENQUELLE	Lehrgebiets-Datenbestand
ASSOZIATION	ark_record_area
VALIDIERUNG	-

11. Jahr

ABB.108 Eingabefeld Jahr

 A screenshot of a web form input field labeled 'Jahr'. The field contains the text '2013' and has a small checkmark icon on the right side, indicating that the input is valid.

Das Jahres-Eingabefeld fasst standardmäßig die aktuelle Jahreszahl als Wert. Darüber hinaus findet eine strenge Validierung Anwendung, welche nur zwei- oder vierstellige Jahreszahlen als Eingabe akzeptiert, Eingaben vor 1990 auf das Gründungsjahr der KISD setzt und Jahreszahlen, welche in der Zukunft angesiedelt sind auf das aktuelle Jahr.

TYP	Numerisches Textfeld
ASSOZIATION	ark_record_year
VALIDIERUNG	zwei- oder vierstellige Jahreszahl, s. o.

12. Sprache

ABB.109 Eingabefeld Sprache

 A screenshot of a web form dropdown menu labeled 'Sprache'. The dropdown is open, showing 'deutsch' as the selected option. A small checkmark icon is visible on the right side of the dropdown.

Das Sprach-Eingabefeld wird mit deutsch als Sprache vorausgewählt geladen. Darüber hinaus sind englisch oder spanisch wählbar, eine Nicht-Auswahl ist auszuschließen und damit keine Validierung notwendig.

TYP	Dropdown-Feld
ASSOZIATION	ark_record_year
VALIDIERUNG	-

13. PDF-Datei

ABB.110 PDF-Datei

 A screenshot of a web form interface for uploading a PDF file. It features a title 'PDF-Datei' followed by a disclaimer: 'Mit dem Hochladen eines PDFs erklärst du dich damit einverstanden, dass deine Arbeit innerhalb des digitalen Archivs sichtbar ist.' Below this is a yellow tip box that says: 'Tipp: Ziehe das PDF einfach in dieses Browserfenster. Die maximale Dateigröße beträgt 10 MB.' At the bottom, there is a button labeled 'PDF-Datei auswählen' and a partially visible input field.

Der Upload einer PDF-Datei direkt über die Arbeit einreichen-Maske kann via Betätigung des entsprechenden Buttons oder durch Ziehen einer Datei in das Browserfenster geschehen. Bevor die Datei dem Upload-Handler übergeben wird, überprüft das System, ob es sich dabei um eine PDF-Datei handelt und diese die maximale Dateigröße nicht überschreitet. Wird eines dieser Kriterien nicht erfüllt, erhält der Nutzer eine Meldung mit der Aufforderung, eine Datei korrekten Formats oder korrekter Größe einzureichen. Die gültige Datei wird in ein Zwischenverzeichnis auf dem Server übertragen und ihre Position in der

Variable **ark_record_pdf_url** geschrieben. Während des Uploads wird der »Arbeit einreichen«-Button nicht klickbar geschaltet, um zu verhindern, dass der Eintrag eingereicht wird, bevor die Datei komplett auf den Server übertragen werden konnte. Bei der späteren Freischaltung des Eintrags durch die AG wird die Datei umbenannt und an die korrekte Stelle innerhalb der Ordnerstruktur verschoben.

Der Prüfling wird nicht gezwungen, eine PDF-Datei hochzuladen. Sofern eine Datei hochgeladen wird, setzt das System die Sichtbarkeitsvariable des PDFs (**ark_record_visible**) auf **TRUE**. Wird keine PDF-Datei eingereicht, erhält die Variable den Wert **FALSE**.

TYP	Dateiupload-Feld
ASSOZIATION	ark_record_pdf_url / ark_record_visible
VALIDIERUNG	maximale Dateigröße / Dateiformat: PDF

14. Anmerkung

ABB.111 Anmerkungsfeld



Ein Anmerkungsfeld am Formularende gibt den Prüflingen die Möglichkeit, der Arbeit eine Notiz anzuhängen, welche der AG bei der Prüfung des Eintrags im Backend angezeigt wird. Das Feld unterliegt keiner Validierung.

TYP	Textfeld
ASSOZIATION	ark_record_note
VALIDIERUNG	-

14. Arbeit einreichen

ABB.112 »Arbeit einreichen«-Button



Mit Absenden des Formulars werden die eingegebenen Daten einem neuen Eintrag zugeordnet, welcher unter dem Status »pending review« im Backend sichtbar ist. Für den Fall, dass ein noch nicht im System vorhandener Autor angegeben wurde (siehe Feld 5) wird dieser ebenfalls als unter dem Status »pending review« im Backend angelegt.

Dem Eingebenden im Frontend wird währenddessen ein Ausgabeformular generiert, welches seine soeben eingegebenen Metadaten für die Druckausgabe aufbereitet darstellt. Siehe **ark_upload_complete.php** (S. 182).

Überprüfung, Korrektur und Freischaltung

Mit Absenden des »Arbeit einreichen«-Formulars im Frontend erscheint der Eintrag im Dashboard des Backends unter dem Status »pending review«. Ist dem Eintrag überdies ein Autor zugewiesen worden, welcher noch nicht mit einer Arbeit im System vertreten ist, so wird dieser ebenso geführt.

ABB. 113 Backend, Dashboard: Ansicht ausstehender Einträge (Ausschnitt)

Content		Organization	
4,253 Einträge	2 pending	9,052 Tags	
1,004 Autoren	1 pending	4 Status	
300 Drucker		3 Gruppen	

Per Klick auf die als **pending** ausgezeichneten Buttons kann das AG-Mitglied die zu überprüfenden Einträge in einer Listendarstellung aufrufen. Innerhalb der Listendarstellung sind bereits ein Großteil der vom Prüfling eingegebenen Metadaten abgebildet:

ABB. 114 Backend, Eintragsübersicht ausstehender Einträge (Status: »pending«)

Anforderung	ID	Titel	Tag	Autor	Subjekt	Status	PDF Pfad	Datei	Date
Ullmer Markier 2.0 - Entwicklung eines neuartigen Stanzbals	1234	Ullmer Markier 2.0 - Entwicklung eines neuartigen Stanzbals	Ullmer Markier, Stanzballe, Entwicklung, Stanzballe, Ullmer, Markier, 2.0, Ullmer, Markier	Ullmer Markier					
Ullmer Markier 2.0 - Entwicklung eines neuartigen Stanzbals	5678	Ullmer Markier 2.0 - Entwicklung eines neuartigen Stanzbals	Ullmer Markier, Stanzballe, Entwicklung, Stanzballe, Ullmer, Markier, 2.0, Ullmer, Markier	Ullmer Markier					

Ein Klick auf den Arbeitstitel öffnet die Arbeit in der Eintrags-Einzelansicht:

ABB. 115 Eintrags-Einzelansicht eines ausstehenden Eintrags

Eintrag bearbeiten
Ullmer Markier 2.0 - Entwicklung eines neuartigen Stanzbals
Integritätsmetabox: 100%
PDF-Pfad: Ullmer Markier 2.0 - Entwicklung eines neuartigen Stanzbals.pdf
Publish

Innerhalb der Eintrags-Einzelansicht kann das AG-Mitglied mit einem Blick auf die Integritätsmetabox die Vollständigkeit der Metadaten ansehen. Das Feld **PDF-Pfad** zeigt an, dass sich die PDF-Datei im temporären Ordner befindet. Nachdem sich das AG-Mitglied nun von der Korrektheit der eingegebenen Metadaten überzeugt hat, genügt ein Klick auf den Button **Publish**, um den Eintrag im KISDarchiv zu veröffentlichen. In diesem Zug zieht das System automatisch die soeben vom AG-Mitglied validierten Metadaten heran um die PDF-Arbeit an die korrekte Stelle innerhalb der Dateistruktur zu relokalisieren und einen Permalink zu erzeugen.

HINWEIS Grundsätzlich sollten zuerst neu angelegte Autoren auf diese Weise öffentlich geschaltet werden, bevor ihnen zugeordneten Einträge freigeschaltet werden.

Fassen wir den soeben beschriebenen Vorgang noch einmal zusammen:

Neuer Abgabevorgang

1. Der Prüfling gibt die Metadaten zur Arbeit ein und lädt die PDF-Datei hoch. Das KISDarchiv quittiert die Eingabe mit der automatischen Generierung des Abgabeformulars, welches der Prüfling den Druckexemplaren beilegen kann.
2. Die AG überprüft die eingegebenen Daten, ergänzt oder korrigiert diese gegebenenfalls und schaltet den Eintrag im KISDarchiv frei. Das System sorgt für die korrekte Verortung der hochgeladenen Datei auf dem Server.

Vorteile des neuen Abgabevorgangs

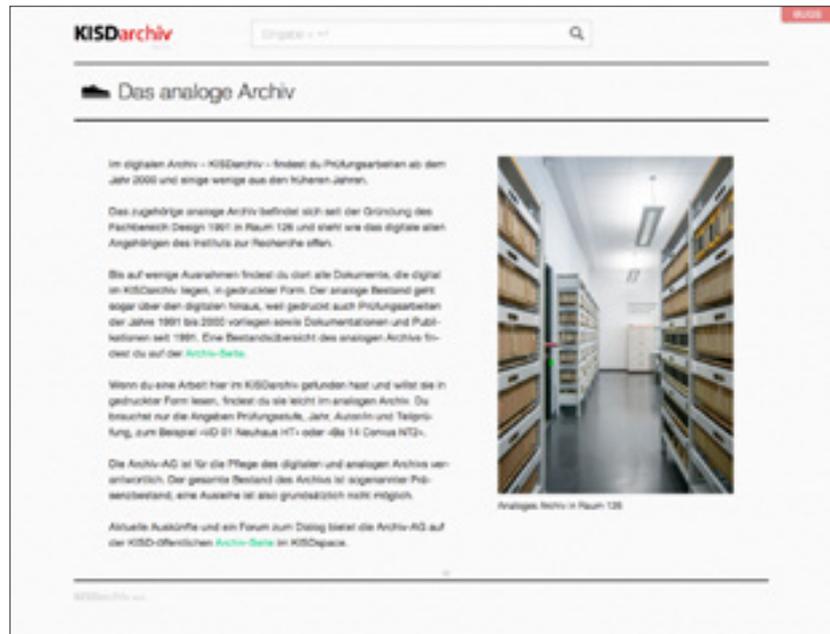
Der neue Abgabevorgang weist eine Reihe von Vorteilen gegenüber dem bisherigen Modus Operandi auf:

- **Vereinfachter Ablauf:** Statt die PDF-Datei auf eine CD zu brennen, diese zum Sekretariat zu tragen, von wo aus sie nach Lagerung ins Archiv gegeben wird, wo sie nach abermaliger Lagerung ausgelesen und auf den Server geladen wird, wird die PDF-Datei im KISDarchiv direkt auf den Server geladen.
- **Ökologischer(er) Ablauf:** Pro Abgabebjahr werden 175-200 CDs eingespart.
- **Sicherheit:** CDs mit sensiblen Prüfungsinhalten müssen nicht mehr »bewacht« und vernichtet werden, unlesbare CDs sind passé. Fehleingaben und Fehler in der Übertragung von Metadaten werden drastisch reduziert.
- **Zeitersparnis:**
 - Der Prüfling spart sich die Zeit, CD-Rohlinge anzuschaffen und zu bespielen. Die Eingabe der Arbeit im digitalen Archiv dauert nicht länger als das Ausfüllen des bislang gebräuchlichen PDF-Formulars und resultiert in der Generierung desselben.
 - Frau Lindner spart Zeit, sie muss nicht mehr auf das Vorhandensein von CD-Rohlingen prüfen.
 - Die AG spart Zeit, da die Einpflege nun den Prüflingen überantwortet wird. Die AG kann sich auf die Prüfung und Aufwertung der eingegebenen Daten konzentrieren.
- **Überprüfbarkeit:** Mit Vorlage des vom KISDarchiv generierten Abgabeformulars erbringt der Prüfling bereits den Nachweis, seine Arbeit digital eingereicht zu haben. Die Nachpflege von digitalen Arbeiten wird erleichtert, da die Prüflinge Arbeiten direkt ins KISDarchiv einreichen können. Die Vereinfachung und relative Unabhängigkeit von der AG, welche hierbei eine überblickende Rolle einnimmt, erhöht zudem die Chance, Arbeiten nachträglich zu erhalten.

10.10.12 Das analoge Archiv

Schnittstelle zwischen analogem und digitalem Archiv.

ABB. 116 Seite Das analoge Archiv



Die Seite **Das analoge Archiv** bildet die Schnittstelle zwischen analogem und digitalem Archiv. An dieser Stelle wird auf die Existenz des Institutsarchivs in Raum 126 hingewiesen, sowie auf die Archiv-Informationseite innerhalb der KISDspaces verwiesen.

Die Seite ist bewusst minimal gehalten und enthält einen Text, der auch bei mehrjähriger Nichtbeachtung noch den Eindruck von Aktualität vermittelt.

10.10.13 Bugs-Seite

Interaktive ToDo-Liste für Beta-Tester.

ABB.117 Bugs-Seite in vier Abschnitten.



Die Bugs-Seite wurde Beginn der Limited Public Beta erstellt, um den zwei Betatestern eine komfortable Möglichkeit zu geben, Anregungen und Bugs direkt aus dem KISDarchiv heraus einzureichen.

Die Seite bestand aus einem zweiseitigen Layout. Die rechte Spalte umfasste eine Eingabemaske über welche Anregungen in Textform eingegeben und abgeschickt werden konnten. Diese erschienen unmittelbar unterhalb der Maske in chronologisch absteigender Form sortiert. Die linke Spalte enthielt Anregungen, die bereits einer von sechs Kategorien zugeordnet wurden:

KATEGORIE	KODIERUNG	BESCHREIBUNG
UNKLAR (LOOK INTO)	hellgrau	Keiner Kategorie zuzuordnen
MUST HAVE (FEATURES)	dunkelblau	Ideen für funktionale, wesentliche Erweiterungen
SHOULDN'T HAVE (BUGS)	rot	Bugs und Fehler jeglicher Art
NICE TO HAVE (FEATURES)	türkis	Anregungen für fakultative Erweiterungen
INHALTLICHES / STRUKTUR / DATEN	violett	Anmerkungen zu konkreten Datensätzen im Bestand
WORTLAUT	dunkelgrau	Anmerkungen zur Nomenklatur des Frontends

Jede Kategorie bildete eine eigene Liste mit Einträgen, jeder Eintrag enthielt die Textanmerkung sowie den Namen des Betatesters, der diese eingesandt hatte. Mit Korrektur oder Umsetzung der Anmerkungen wurden die Einträge in eine chronologische Liste »Behobene Bugs« unterhalb der kategorischen Listen verschoben und als bearbeitet gekennzeichnet.

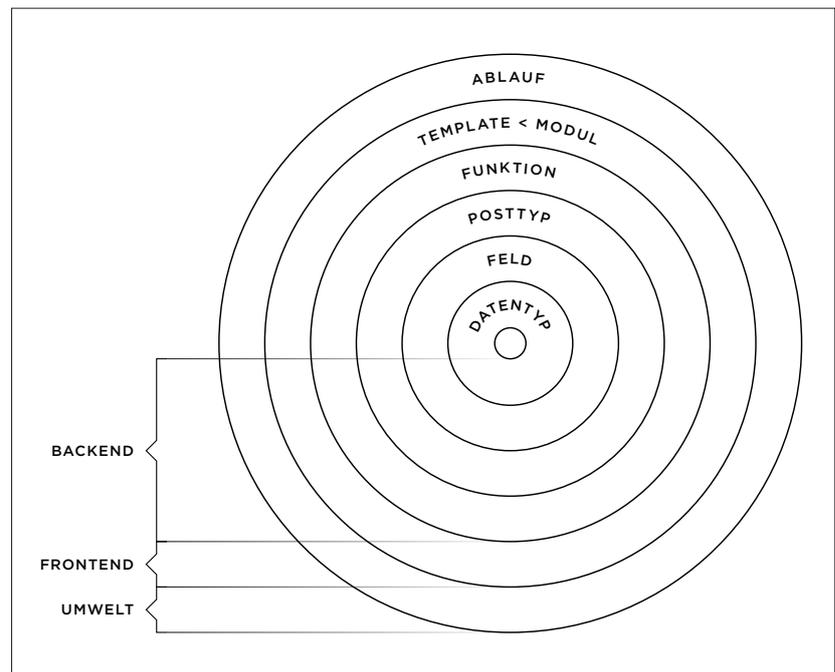
Das Bugs-Modul umfasste einen seitenweit angezeigten »Bug-Button«, welcher über die Funktion `ark_show_bug_button()` eingebunden wurde. Mit Ende der Limited Public Beta wurde die Seite sowie der Button ausgesetzt.

Zweiter Teil

OPERATION

Das Operationshandbuch richtet sich in erster Linie an jene, die auf administrativer Ebene mit dem KISDarchiv zu tun haben. Es erläutert auf diversen Ebenen die unterliegenden Mechanismen, welche das System ausmachen.

ABB. 118 Operatives Ebenenmodell des KISDarchivs.



Dieser Teil orientiert sich am hier abgebildeten operativen Ebenenmodell, wobei von innen nach außen vorgegangen wird:

- DATENTYPEN** Datentypen bilden die kleinste Einheit des Systems. Sie klassifizieren in der Datenbank angelegte Felder. Datentypen beschreiben numerische Werte, Textinhalte («Strings»/«Zeichenketten») und boolesche Logik (true/false). Siehe **Posttyp-Struktur** (S. 114).
- FELDER** Felder sind datentypgebundene Variablen, welche ein bestimmtes Attribut abbilden. Das WordPress-eigene Feld **post_title** etwa fasst den Titel eines Posts, abgebildet im Datentyp **Text**.
- POSTTYPEN** Posttypen kombinieren Felder und bilden darauf Sinneinheiten («Einträge»). Auf dieser Ebene können die enthaltenen Felder als Referenzen auf Posts anderer Posttypen agieren und so Zusammenhänge zwischen Posts abbilden.
- FUNKTIONEN** Funktionen dienen der Anzeige und Manipulation von Einträgen. Einige Funktionen fungieren dabei posttypübergreifend wohingegen andere auf einen bestimmten Posttyp zugeschnitten sind.
- TEMPLATES** Templates kapseln mindestens eine und zumeist mehrere Funktionen im Frontend-basierten Kontext. Sie treten mitunter verschachtelt auf.
- MODULE** Module stellen sinnstiftende Gruppierungen von Templates dar.
- ABLÄUFE** Abläufe umfassen mehrere Personen oder Gruppen, umspannen Front- und Backend und bilden die Schnittstelle zu anderen KISD-internen Prozessen.

1 Posttypen

1.0 Einleitung

Die vier KISDarchiv-eigenen Posttypen (in WordPress »Custom Post Types«) bilden die zentralen Entitäten innerhalb des Archivs und die Beziehungen zwischen diesen ab. Die KISDarchiv-Posttypen ergänzen die WordPress-eigenen Posttypen zu denen **post** und **page** (Post-Eintrag und Seite) zählen.

- Der Posttyp **ark_record** enthält Einträge, welche alle Informationen (Metadaten) zu jeweils einer Prüfungsarbeit fassen.
- Der Posttyp **ark_area** enthält Lehrgebiete, in welchen Einträge und Dozenten verortbar sind.
- Der Posttyp **ark_author** enthält alle mit Arbeiten vertretenen Autorinnen und Autoren, welche Einträgen zuordenbar sind.
- Der Posttyp **ark_lecturer** enthält alle Personen, welche aus Einträgen heraus in ihrer Position als Prüfer und Prüferinnen referenziert werden.

1.0.1 Posttyp-Struktur

Posttypen setzen sich aus Feldern zusammen, die mitunter als »Variablen« oder »Metadaten« oder »Metadatenfelder« bezeichnet werden. Jedem Feld ist genau ein Datentyp zugeordnet. Dieser bestimmt maßgeblich, welche Art von Daten das jeweilige Feld zu fassen imstande ist. Folgende Datentypen sind für die Felder der KISDarchiv-Posttypen von Relevanz:

DATENTYP	BESCHREIBUNG
TEXT	Ein Feld, welches eine beliebige Zeichenkette enthält.
FILE	Eine Referenz auf ein WordPress-Attachment; also eine Datei.
SELECT	Ein Dropdown-Feld mit vorgegebenen Auswahl-Werten.
BOOLE	Ein Auswahlfeld welches eine binäre wahr/falsch-Logik abbildet.
REFERENZ	Eine Referenz auf einen Post bestimmten Posttyps.
CHECKBOX	Ein Auswahlfeld mit vorgegebenem Wert.

KISDarchiv-eigene Posttyp-Variablen tragen stets den Namen des Posttyps selbst im Namen um ihre Posttyp-Zuordnung zu erleichtern. So ist bereits aus dem Bezeichner **ark_record_area** ersichtlich, dass es sich um eine Variable des Posttyps **ark_record** handelt, die sich auf das Lehrgebiet bezieht. Die Benennung von Templates und Funktionen folgt ebendieser Logik.

Auf den folgenden Seiten finden sich Beschreibungen der vier KISDarchiv-Posttypen anhand der ihnen eigenen Feldern. Jeder Posttyp wird kurz vorgestellt, darauf folgt eine tabellarische Übersicht der posttyp-eigenen Felder und schließlich eine detaillierte Beschreibung der einzelnen Felder.

1.1 ark_record / Eintrag

Der zentrale Posttyp innerhalb des KISDarchivs. Dieser fasst jede Prüfungsarbeit in Form genau eines Eintrags.

Feldübersicht

BESCHREIBUNG	NAME	TYP	WP-FELD
TITEL DER ARBEIT	post_title	text	×
ABSTRACT	post_content	text	×
SCHLAGWÖRTER	post_tag	referenz	×
PDF-PFAD	ark_record_pdf_url	text	
PDF	ark_record_pdf	file	
PRÜFUNGSSTUFE	ark_record_exam_type	select	
TEILPRÜFUNG	ark_record_subexam	select	
LEHRGEBIET	ark_record_area	referenz	
PRÜFER	ark_record_lecturer	referenz	
AUTOR/EN	ark_record_author	referenz	
JAHR	ark_record_year	text	
SPRACHE	ark_record_language	select	
PDF ANZEIGEN	ark_record_visible	checkbox	
ANMERKUNG	ark_record_note	text	
SUCHMETA	ark_record_search_meta	text	

Felder

TITEL DER ARBEIT	<p>post_title</p> <p>Das WordPress-eigene Titelfeld fasst den Eintragstitel. Dieses Feld weist die Besonderheit auf, dass ein Gedankenstrich (»-«) hier als Trennzeichen (Delimiter) zwischen Titel und Untertitel verstanden wird. Soll ein Titel mehrere Gedankenstriche enthalten, so werden die übrigen Gedankenstriche über ihre HTML-Entsprechung (&ndash;) eingegeben. Dies ist vonnöten, da nicht davon ausgegangen werden kann, dass es sich beim ersten auftretende Gedankenstrich um einen Delimiter handelt.</p> <p>Beispiel: Rom - Stadt &ndash; Metropole &ndash; Zeichen</p>
ABSTRACT	<p>post_content</p> <p>Das WordPress-eigene Inhaltsfeld dient der Speicherung des Abstracts. In Templates wird dieses wie gewohnt via the_content()-Funktion abgerufen, in Mehrfachansichten findet die Funktion ark_get_excerpt_by_id() Einsatz.</p>
SCHLAGWÖRTER	<p>post_tag</p> <p>Der ark_record-Posttyp bedient sich der WordPress-eigenen Schlüsselwort-Taxonomie zur Speicherung von Tags.</p>
PDF-PFAD	<p>ark_record_pdf_url</p> <p>Dieses Feld enthält den absoluten Pfad zu der dem Datensatz zugeordneten PDF-Datei und wird in der Regel vom KISDarchiv mit Freischaltung eines Eintrags nach deren Upload über das Frontend generiert.</p>

PDF ark_record_pdf

Das Feld PDF dient als Schnittstelle zur WordPress-eigenen Media-Verwaltung, erlaubt den Upload von Dateien und speichert temporär die Attachment-ID des Uploads.

PRÜFUNGSSTUFE ark_record_exam_type

Die Prüfungsstufe fasst Optionen in Form eines Dropdown-Felds.

Hinweis: Die internen Kürzel einzelner Prüfungsstufen sind historisch begründet und dienen nur als API-interne Identifier.

Folgende Selektionsmöglichkeiten sind definiert:

WERT	LABEL
Null	Bitte auswählen
'ip'	Intermediate
'ba'	Bachelor
'ma'	Master
'vordipl'	Vordiplom
'dipl'	Diplom

TEILPRÜFUNG ark_record_subexam

Das Feld Teilprüfung bildet alle möglichen Teilprüfungen ab, namentlich ein einzelnes Nebenthema (NT), zwei Nebenthemen (NT1, NT2) und ein Hauptthema (HT). Siehe **6. Studiengang / Prüfungsstufe / Teilprüfung** (S. 102).

Folgende Selektionsmöglichkeiten sind definiert:

WERT	LABEL
Null	Bitte auswählen
'nt1'	NT1
'nt2'	NT2
'nt'	NT
'ht'	HT

LEHRGEBIET ark_record_area

Das Feld **ark_record_area** enthält eine Referenz auf einen Post vom Posttyp **ark_area**. Das Lehrgebiet wird anhand seiner ID geführt.

PRÜFER ark_record_lecturer

Das Prüfer-Feld enthält eine Referenz auf einen oder mehrere Lehrenden (Post-Typ **ark_lecturer**). Diese werden als ein Array von IDs gespeichert.

Anmerkung: Der Name des Felds ist historisch begründet und stammt aus einer Zeit, da dieses Feld sowohl Prüfer als auch Lehrende führte.

AUTOR/EN ark_record_author

Das Autorenfeld fasst Referenzen auf eine/n oder mehrere Autoren/innen in Form derer IDs.

JAHR ark_record_year

Das Jahresfeld fasst den Jahrgang der Einreichung einer Arbeit in der Form (»YYYY«)

SPRACHE ark_record_language

Dieses Feld fasst die Sprache der Arbeit, auszuwählen via Dropdown-Feld.

Folgende Selektionsmöglichkeiten sind definiert:

WERT	LABEL
'de'	deutsch
'en'	englisch
'es'	spanisch

PDF ANZEIGEN ark_record_visible

Dieses Feld enthält einen Wert der anzeigt, ob das PDF öffentlich gezeigt werden darf. Diese Variable wird mit Upload einer PDF-Datei aus dem Frontend auf **TRUE** gesetzt.

ANMERKUNG ark_record_note

Dieses Feld bietet die Möglichkeit, eine Anmerkung zu speichern. Das Feld nimmt damit eine Doppelrolle ein: Die Prüflinge haben beim Upload im Frontend die Möglichkeit, hier eine Nachricht zu speichern, die Bearbeiter im Backend können das Feld nutzen, um einander Hinweise zu hinterlegen.

SUCHMETA ark_record_search_meta

Das Suchmeta-Feld dient der Speicherung von für die Suche relevanten Begriffen im Klartextformat. Um Einträge über die Suche durch Eingabe des Autorennamens auffindbar zu machen, muss dieser entweder im PDF-Pfad oder in dieser Variable kodiert werden. Dies geschieht automatisch.

1.2 ark_area / Lehrgebiet

Der **ark_area**-Posttyp fasst alle KISD-eigenen Lehrgebiete. Einträge diesen Posttyps werden aus dem Posttyp **ark_record** heraus via **ark_record_area** referenziert, sowie aus dem Posttyp **ark_lecturer** via dem Feld **ark_lecturer_area**.

Feldübersicht

BESCHREIBUNG	NAME	TYP	WP-FELD	DEPRECATED
LEHRGEBIETSNAME	post_title	text	x	
KURZFORM	ark_area_ID	text		
BESCHREIBUNG	ark_area_description	text		x
ANZEIGEN	ark_area_show	boole		
VERALTETE KÜRZEL	ark_area_former_ID	text		

Felder

LEHRGEBIETSNAME	post_title	Das Titelfeld fasst die ausgeschriebene Form des Lehrgebietsnamens. Beispiel: <code>Designtheorie und -geschichte</code>
KURZFORM	ark_area_ID	Das Kürzel-Feld fasst das jeweilige Lehrgebietskürzel. Beispiel: <code>DTF</code>
BESCHREIBUNG	ark_area_description <u>DEPRECATED</u>	Ein Beschreibungs-Textfeld, welches derzeit keine Verwendung findet.
ANZEIGEN	ark_area_show	Dieses Feld reguliert die Anzeige des Lehrgebiets im Frontend.
VERALTETE KÜRZEL	ark_area_former_ID	Dieses Feld fasst in kommaseparierter Notation ehemalige Lehrgebietskürzel. Beispiel: <code>DT</code>

1.3 ark_author / Autor/in

Im Posttyp **ark_author** werden alle im KISDarchiv mit Arbeiten vertretenen Autoren/innen gefasst. Diese werden in einem eigenen Posttyp gefasst, da Autoren historisch betrachtet über keine KISDspaces-Accounts verfügen und Einträge von diversen Parteien eingepflegt werden.

Feldübersicht

BESCHREIBUNG	NAME	TYP	WP-FELD
NAME	post_title	text	x
KISD ID	ark_author_ID	text	
VORNAME	ark_author_firstname	text	
NACHNAME	ark_author_lastname	text	
MATRIKELNUMMER	ark_author_mnr	text	
STUDIENGANG	ark_author_degree	select	
STUDIENBEGINN	ark_author_start	text	

Felder

NAME	post_title	Im Titelfeld wird der vollständige Studierendenname abgelegt. Der Feldinhalt wird aus den Eingaben für Vor- und Nachname generiert.
KISD ID	ark_author_ID	Die KISD-ID, wie sie in der @KISD.de-E-Mail-Adresse vorkommt.
VORNAME	ark_author_firstname	Der Vorname des Autors/der Autorin.
NACHNAME	ark_author_lastname	Der Nachname des Autors/der Autorin.
MATRIKELNUMMER	ark_author_mnr	Dieses Feld fasst die Matrikelnummer des Autors/der Autorin.
STUDIENGANG	ark_author_degree	Der Studiengang, in welchem der Studierende matriculiert ist
	WERT	LABEL
	Null	Bitte auswählen
	'bides'	BIDes
	'bedes'	BEDes
	'mides'	MIDes
	'medes'	MEDes
	'diplom'	Diplomstudiengang
STUDIENBEGINN	ark_author_start	Der Studienbeginn des Studierenden im Format YYYY-#. Beispiel: 2009-1.

1.4 ark_lecturer / Prüfer/in

Dieser Posttyp fasst alle prüfungsberechtigten Personen an der KISD. Einträge diesen Posttyps werden aus dem Posttyp **ark_record** heraus via dem Feld **ark_record_lecturer** referenziert.

Feldübersicht

BESCHREIBUNG	NAME	TYP	WP-FELD
NAME	post_title	text	x
TITEL	ark_lecturer_title	text	
VORNAME	ark_lecturer_firstname	text	
NACHNAME	ark_lecturer_lastname	text	
KURZNAME	ark_lecturer_shortcode	text	
E-MAIL-ADRESSE	ark_lecturer_email	text	
LEHRGEBIET	ark_lecturer_area	referenz	

Felder

NAME **post_title**

Im Titelfeld wird der vollständige Prüfervorname samt Titel abgelegt.

Der Feldinhalt wird aus den Eingaben für Titel, Vor- und Nachname generiert.

TITEL **ark_lecturer_title**

Das Feld fasst den Prüfer-Titel:

WERT	LABEL
Null	Bitte wählen
'Dr.'	Dr.
'Prof.'	Prof.
'Prof. Dr.'	Prof. Dr.
'Dipl.-Des.'	Dipl.-Des.
'Dipl.-Ing.'	Dipl.-Ing.
'Dipl.-Inf.'	Dipl.-Inf.
'Dipl.-Medienpäd.'	Dipl.-Medienpäd.

VORNAME **ark_lecturer_firstname**

Der Vorname des Prüfers/der Prüferin.

NACHNAME **ark_lecturer_lastname**

Der Nachname des Prüfers/der Prüferin.

KURZNAME **ark_lecturer_shortcode**

Das Kürzel des Prüfers/der Prüferin.

E-MAIL-ADRESSE **ark_lecturer_email**

Die E-Mail-Adresse des Prüfers/der Prüferin.

LEHRGEBIET **ark_lecturer_area**

Zuordnung eines Lehrgebiets (Referenz auf Eintrag vom Posttyp **ark_area**).

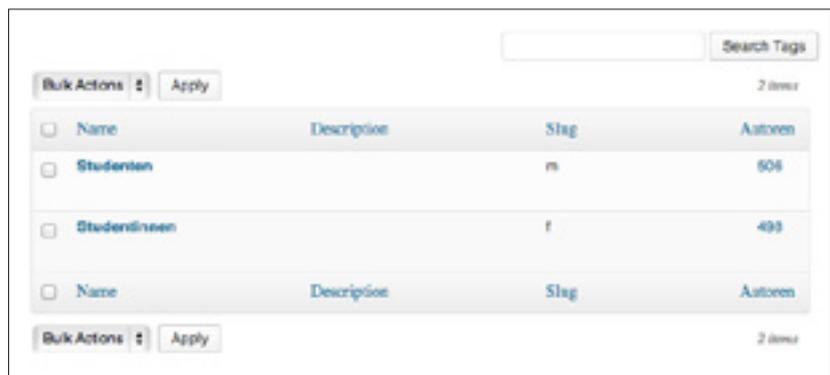
2 Taxonomien

2.0 Einleitung

Die WordPress-Kategorien («categories») und Schlagworte («tags») basieren auf Taxonomien, welche die Gruppierung von Posts erlauben. Hier werden Taxonomien herangezogen, um Posts vom Typ **ark_author** und **ark_lecturer** nach Geschlecht respektive Funktion innerhalb der KISD zu gruppieren.

2.1 ark_author_groups

ABB. 119 Darstellung der Begriffe der **ark_author_groups**-Taxonomie im Backend.



Die Taxonomie **ark_author_groups** wird genutzt, um Autoren und Autorinnen je nach Geschlecht der Gruppe »Studenten« oder »Studentinnen« zuzuordnen.

Diese Zuordnung ermöglicht den Frontend-Templates die korrekte Anzeigenderspezifischer Suffixe.

2.2 ark_lecturer_role

Die **ark_lecturer_role**-Taxonomie erlaubt die Zuordnung von Lehrenden in Gruppen, die deren Rolle innerhalb der KISD spezifizieren:

- Alumni
- Gastdozenten
- Mitarbeiter
- Profs

Dies dient in erster Linie der Sortierbarkeit der Lehrenden-Datensätze im Backend und somit der besseren Übersicht, funktional ergeben sich durch die Zuordnung zu einer Gruppe keine Unterschiede.

2.3 ark_lecturer_groups

Ähnlich der **ark_author_groups**-Taxonomie dient diese Taxonomie der Zuordnung von Gender-Metadaten zu Lehrenden-Datensätzen um diese im Frontend korrekt abzubilden.

3 Funktionsübersicht / API

3.0 Einleitung

Die Funktionsübersicht oder API (Application Programming Interface) fasst alle Funktionen, welche innerhalb des KISDarchivs zum Einsatz kommen.

3.0.1 Funktionstypen

Die Funktionen lassen sich in mehrere Gruppen fassen:

- **Template-Funktionen**, welche direkt aus Templates heraus aufgerufen werden, zumeist als Schnittstelle zu den KISDarchiv-Posttypen agieren und hierarchisch keiner anderen Funktion unterzuordnen sind. Dies bedeutet nicht, dass sie nicht aus anderen Funktionen heraus aufgerufen werden.
- **Helferfunktionen**, die eine Vielzahl von Aufgaben übernehmen. Darunter fällt die Deklaration von Posttypen, die Modifikation WordPress-eigenen Verhaltens, Die Generierung von Metadaten, die Manipulation von Cookies, String Operationen, das Verschieben hochgeladener Dateien und die Darstellung von Widgets.
Helferfunktionen sind oftmals Template-Funktionen hierarchisch unterzuordnen und können darüber hinaus einander untergeordnet sein.
- **Includes**, die im Grunde dateiegekapselte Funktionen bilden, und welche zumeist via AJAX/jQuery dynamisch aufgerufen werden.

3.0.2 Funktionsaufruf

Funktionen werden auf diverse Art und Weise aufgerufen:

- **Direkter Aufruf**: Template-Funktionen werden direkt aus einem Template heraus, zumeist also innerhalb des WordPress-Loops, aufgerufen und agieren mit Post-eigenen (Meta-)Daten.
- **Verschachtelter Aufruf**: Funktionen, die aus anderen Template- oder Helferfunktionen heraus aufgerufen werden.
- **Gehookter Aufruf**: Funktionen, die sich den WordPress- oder Plugin-eigenen Action- und Filter-Hooks bedienen, um sich an bestimmter Stelle in bestehende Prozesse einzuhaken.
- **Manueller Aufruf**: Helferfunktionen, welche Aufgaben erfüllen, die im Regelbetrieb keine Rolle spielen. Sie erlauben zum Beispiel die flächendeckende Neugenerierung von Metadaten.
- **Aufruf via jQuery/AJAX**: Includes werden dynamisch aus JavaScript heraus aufgerufen oder per PHP-Include() eingebunden.

3.0.3 Beispiel Funktionsdokumentation

Die Funktionen innerhalb dieser API werden nach dem folgenden Muster dargestellt:

ark_function_name()

Ein kurzer Beschreibungstext gibt Aufschluss darüber, was die Funktion tut.

FUNKTIONSAUFRUF	Der Funktionsaufruf zeigt anhand von Funktionsname und Parametern, wie die Funktion aus PHP heraus aufgerufen wird.
PARAMETER	Eine tabellarische Aufstellung aller zulässigen Funktionsparameter in der Reihenfolge, wie die Funktion diese übernimmt. Pro Parameter werden der Datentyp, der Variablenname sowie eine kurze Beschreibung angegeben. An dieser Stelle ist auch gekennzeichnet, ob Parameter über Standardwerte verfügen, was deren potenzielle Aussparung im Funktionsaufruf bedeutet.
RÜCKGABEWERT	Sofern die Funktion über einen Rückgabewert verfügt, ist dieser samt seines Datentyps vermerkt. Hinweis: Beachte den Unterschied zwischen Rückgabe und Ausgabe. Eine Funktion kann eine Ausgabe erzeugen, aber über keinen Rückgabewert verfügen.
BEISPIEL	Sofern angebracht, werden Beispiele gegeben, die einen Funktionsaufruf und dessen Ausgabe veranschaulichen. Auch hier ist zu beachten, ob eine Aus- oder Rückgabe gekennzeichnet ist. Eine Rückgabe wird in PHP erst via echo-Aufruf zur Ausgabe.
ANMERKUNGEN	Zusätzliche Anmerkungen zur Funktion oder zum Beispiel werden angeführt.
HOOK	Sofern der Aufruf via Hook erfolgt, ist dieser hier vermerkt. Der Unterschied zwischen Action- und Filter-Hook ist zu beachten.
QUELLDATEI	Die Datei, innerhalb welcher die Funktion definiert ist, wird hier referenziert.
ÄHNLICHE FUNKTIONEN	Funktionen, welche ähnliche Rollen einnehmen, werden hier aufgelistet.

Die oben aufgeführten Kategorien bilden die Gesamtheit derselben ab, will heißen: Für die einzelne Funktion werden nur jene Kategorien beschrieben, die im Kontext der Funktion relevant sind. Fehlt einer Funktion die Parameter-Beschreibung, so ist davon auszugehen, dass diese keine Parameter annimmt. Gleiches gilt für den Rückgabewert.

Eine Übersicht aller Funktionen sowie deren Einordnung nach Funktionstyp findet sich auf der folgenden Seite.

3.1 Templatefunktionen

3.1.1	ark_get_area_by_ID()	125
3.1.2	ark_get_area_meta()	125
3.1.3	ark_get_author_by_ID()	126
3.1.4	ark_get_author_degree()	127
3.1.5	ark_get_author_name()	127
3.1.6	ark_get_author_meta()	128
3.1.7	ark_get_author_for_permalink()	128
3.1.8	ark_get_elements()	129
3.1.9	ark_get_excerpt_by_id()	129
3.1.10	ark_get_lecturer_by_ID()	130
3.1.11	ark_get_lecturer_meta()	130
3.1.12	ark_get_record_subexam()	131
3.1.13	ark_get_tile()	131
3.1.14	ark_greeting()	132
3.1.15	ark_search_wisecrack()	132
3.1.16	ark_show_bug_button()	132
3.1.17	ark_show_view_modes()	133
3.1.18	ark_view_table()	133

3.1.1 ark_get_area_by_ID()

Diese Funktion übernimmt die numerische ID eines Posts vom Eintrag **ark_area** und gibt das Lehrgebietskürzel oder dessen Titel zurück.

FUNKTIONSAUFRUF `ark_get_area_by_ID($arid, $type)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$arid	Numerische ID eines Lehrgebiets-Eintrags
	string	\$type	Rückgabebetyp: 'ID' Lehrgebietskürzel (Standardwert) 'name' Lehrgebietsname

RÜCKGABEWERT string Lehrgebietskürzel oder -name

BEISPIELE Aufruf: `ark_get_area_by_ID(1153, 'ID')`
Rückgabe: DTF

Aufruf: `ark_get_area_by_ID(1153, 'full')`
Rückgabe: Designtheorie und -forschung

QUELLDATEI /wp-content/themes/ark/functions.php

ÄHNLICHE FUNKTIONEN ark_get_author_by_ID(), ark_get_lecturer_by_ID()

3.1.2 ark_get_area_meta()

Die Funktion gibt das Meta-Element für das Lehrgebiet eines Eintrags aus, wie es in der Eintrags-Einzelansicht vom Posttyp **ark_record** zu finden ist.

FUNKTIONSAUFRUF `ark_get_area_meta($arid)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$arid	Numerische ID eines Lehrgebiets-Eintrags

RÜCKGABEWERT -

BEISPIEL `ark_get_area_meta(1161)`



ABB. 120 Ausgabe der Funktion `ark_get_area_meta()`.

ANMERKUNGEN Die hier illustrierte Ausgabe entspricht der Darstellung innerhalb der Eintrags-Einzelansicht.

QUELLDATEI /wp-content/themes/ark/functions.php

ÄHNLICHE FUNKTIONEN ark_get_author_meta()
ark_get_lecturer_meta()

3.1.3 ark_get_author_by_ID()

Die Funktion übernimmt eine ID oder ein Array von Autoren-IDs und gibt die zugehörigen Namen der Autoren aus. Parameter umfassen den Separator, welcher mehrere Autoren von einander abtrennt, die Verlinkung der Autorennamen, das Format der Namensausgabe, sowie, ob diese nach dem Muster »Nachname, Vorname«, also umgekehrt, geschehen soll.

FUNKTIONSAUFRUF `ark_get_author_by_ID($aid, $separator, $link, $format)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int/array(i)	\$aid	ID des Autoren / Array von IDs
	string	\$sep.	Trennzeichenkette (Standard: ',')
	bool	\$link	Verlinkung (Standard: true)
	string	\$format	Ausgabeformat: 'initials' Anfangsbuchstaben 'short' Kurzname (Standardwert) 'full' Vollständiger Name 'permalink' Permalinkgerechter Name 'lastname' Nur Nachname 'firstname' Nur Vorname
	bool	\$reverse	Umgekehrte Ausgabe (Standard: false)

RÜCKGABEWERT string Autorendaten

BEISPIELE Aufruf:

```
ark_get_author_by_ID( 23761 )
```

Rückgabe:

```
<a href="[]" title="[]">Johannes Eisbrenner</a>
```

Aufruf:

```
ark_get_author_by_ID(array(23761,23804), ', ', false, 'short', true);
```

Rückgabe:

```
Eisbrenner, J., Schraven, L. A.
```

ANMERKUNGEN Die Funktion bedient sich der **ark_get_author_name()**-Funktion. Der Parameter **\$format** wird an ebendiese Funktion übergeben und tritt lediglich in Kraft, wenn mehrere IDs in Form eines Arrays übergeben werden, einzelne Namen gibt die Funktion immer als vollständigen Namen aus. Möchte man einzelne Namen in anderen Formaten ausgeben, so sollte man sich der Funktion **ark_get_author_name()** bedienen.

QUELLDATEI /wp-content/themes/ark/functions.php

ÄHNLICHE FUNKTIONEN ark_get_lecturer_by_ID()
ark_get_area_by_ID()

3.1.4 ark_get_author_degree()

Diese Funktion übernimmt die numerische ID eines Autoren und gibt alle dieser Person zugeordneten Studiengänge (komma-)separiert aus.

FUNKTIONSAUFRUF	<code>ark_get_author_degree(\$aid, \$separator)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	int/array(i)	\$aid	Autoren-ID oder Array von IDs
	string	\$sep.	Trennzeichenkette (Standard: ', ')
RÜCKGABEWERT	string		Studiengang
BEISPIEL	Aufruf:	<code>ark_get_author_degree(23761, ', ')</code>	
	Ausgabe:	BIDes	
ANMERKUNGEN	Funktion findet innerhalb der ark_get_tile() -Funktion Verwendung.		
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.1.5 ark_get_author_name()

Diese Funktion übernimmt die ID eines Posts vom Typ **ark_author** und gibt den Namen des Autoren zurück.

FUNKTIONSAUFRUF	<code>ark_get_author_name(\$aid, \$flag, \$reverse)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$aid	ID eines Autoren
	string	\$flag	Ausgabeformat: 'initials' Anfangsbuchstaben 'short' Kurzname (Standardwert) 'full' Vollständiger Name 'permalink' Permalinkgerechter Name 'lastname' Nur Nachname 'firstname' Nur Vorname
	bool	\$reverse	Umgekehrte Ausgabe (Standard: false)
RÜCKGABEWERT	string		Autorenname
BEISPIELE	Aufruf:	<code>ark_get_author_name(1, 'initials', false)</code>	
	Rückgabe:	AJK	
	Aufruf:	<code>ark_get_author_name(1, 'full', true)</code>	
	Rückgabe:	Katona, Andreas Johannes	
ANMERKUNGEN	Funktion wird in ark_get_author_by_ID() -Funktion verwendet.		
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.1.6 ark_get_author_meta()

Funktion gibt das Meta-Element für den Autoren eines Eintrags aus. Der Parameter **\$type** bestimmt hierbei den Kontext.

FUNKTIONSAUFRUF `ark_get_author_meta($aid, $a, $b, $type, $title)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$aid	Autoren-ID
	string	\$a	Breite Icon (Standard: '25')
	string	\$b	Breite Text (Standard: '75')
	string	\$type	Kontext: 'full' Vollausgabe 'element' Ausgabe in komprimierter Form default: Sidebar-Ausgabe (Standardwert)
	bool	\$title	Linktitel-Ausgabe (Standard: true)

RÜCKGABEWERT -

BEISPIEL `ark_get_author_meta(12785)`



ABB. 121 Ausgabe der Funktion `ark_get_author_meta()`.

ANMERKUNGEN Die hier illustrierte Ausgabe entspricht der Darstellung innerhalb der Eintrags-Einzelansicht.

QUELLDATEI `/wp-content/themes/ark/functions.php`

ÄHNLICHE FUNKTIONEN `ark_get_area_meta()`
`ark_get_lecturer_meta()`

3.1.7 ark_get_author_for_permalink()

Funktion übernimmt **ark_record_author**-Wert und gibt den Namen aller einer Arbeit zugeordneten Autoren in permalinktauglicher Form zurück.

FUNKTIONSAUFRUF `ark_get_author_for_permalink($aid, $separator)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int/array(i)	\$aid	Autoren-ID oder Array von IDs
	string	\$sep.	Trennzeichenkette (Standard: '-und-')
RÜCKGABEWERT	string		Autorennamen für Permalink-Inklusion.

ANMERKUNGEN Findet innerhalb der Funktion `ark_permalinks()` Verwendung.

QUELLDATEI `/wp-content/themes/ark/functions.php`

3.1.8 ark_get_elements()

Funktion übernimmt die ID eines Eintrags beliebigen Datentyps und gibt die Element-Leiste aus.

FUNKTIONSAUFRUF `ark_get_elements($post_ID, $size, $float)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$p.ID	Post-ID
	string	\$size	'medium' Stummer Parameter (Standardwert)
	string	\$float	Float: 'left' Element floated links (Standardwert) 'right' Element floated rechts

RÜCKGABEWERT string HTML-Output

BEISPIELE `ark_get_elements(23479, 'medium', 'left')`



ABB. 122 Ausgabe der Funktion `ark_get_elements()`, Posttyp `ark_record`.

`ark_get_elements(118, 'medium', 'left')`



ABB. 123 Ausgabe der Funktion `ark_get_elements()`, Posttyp `ark_lecturer`.

ANMERKUNGEN Beispiel oben: Eintrag v. Typ `ark_record` / unten: Eintrag v. Typ `ark_lecturer`.

QUELLDATEI `/wp-content/themes/ark/functions.php`

3.1.9 ark_get_excerpt_by_id()

Eine maßgeschneiderte Excerpt-Funktion für die Anwendung außerhalb des Loops. Übergeben werden die Post ID und die Wortanzahl für das Excerpt.

FUNKTIONSAUFRUF `ark_get_excerpt_by_id($post_id, $wordcount)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$p.id	Post-ID
	int	\$wordc.	Wortanzahl des Excerpts (Standardwert: 50)

RÜCKGABEWERT string Zeichenkette welche das Excerpt in einem HTML-Paragraph der Klasse `ark_excerpt` kapselt.

QUELLDATEI `/wp-content/themes/ark/functions.php`

TODO `ark_td_00` (Siehe S. 229)

3.1.10 ark_get_lecturer_by_ID()

Funktion übernimmt die ID eines Posts vom Typ **ark_lecturer** oder ein Array von IDs und gibt die Namen der Lehrenden zurück.

FUNKTIONSAUFRUF `ark_get_lecturer_by_ID($lid, $separator, $link)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int/array(i)	\$lid	Lehrenden-ID
	string	\$sep.	Trennzeichenkette (Standardwert: ',')
	bool	\$link	Verlinkung d. Lehrendennamen (Standardwert: false)

RÜCKGABEWERT string Lehrendenname(n)

BEISPIEL
 Aufruf: `ark_get_lecturer_by_ID(118, ', ', true)`
 Rückgabe: `Prof. Dr. Michael Erhoff`

QUELLDATEI /wp-content/themes/ark/functions.php

ÄHNLICHE FUNKTIONEN `ark_get_area_by_ID()`
`ark_get_author_by_ID()`

3.1.11 ark_get_lecturer_meta()

Funktion gibt das Meta-Element für den Lehrenden eines Eintrags aus. Der Parameter **\$type** bestimmt hierbei den Kontext.

FUNKTIONSAUFRUF `ark_get_lecturer_meta($lid, $a, $b, $type)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$lid	Lehrenden-ID
	string	\$a	Breite Icon (Standard: '25')
	string	\$b	Breite Text (Standard: '75')
	string	\$type	Kontext: 'full' Vollausgabe 'element' Ausgabe in komprimierter Form default: Sidebar-Ausgabe (Standardwert)

RÜCKGABEWERT -

BEISPIEL `ark_get_lecturer_meta(8170)`



ABB. 124 Ausgabe der Funktion `ark_get_lecturer_meta()`.

QUELLDATEI /wp-content/themes/ark/functions.php

ÄHNLICHE FUNKTIONEN `ark_get_area_meta()`
`ark_get_author_meta()`

3.1.12 ark_get_record_subexam()

Die Funktion gibt die korrekte Bezeichnung für eine eintragsseigene Teilprüfung in Abhängigkeit von Studiengang und Prüfungsstufe aus.

FUNKTIONSAUFRUF `ark_get_record_subexam($rid, $output)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$rid	Eintrags-ID
	string	\$output	Ausgabe (Standardwert: 'span')

RÜCKGABEWERT string Prüfungsstufe

BEISPIELE

Aufruf: `ark_get_record_subexam(22470, 'span')`
 Rückgabe: `HT`

Aufruf: `ark_get_record_subexam(30665, 'span')`
 Rückgabe: `NT2`

QUELLDATEI /wp-content/themes/ark/functions.php

3.1.13 ark_get_tile()

Die Funktion übernimmt die Post-ID und prozentuale Breite und generiert eine posttypabhängige Karte innerhalb der Mehrfachansicht »Karten«.

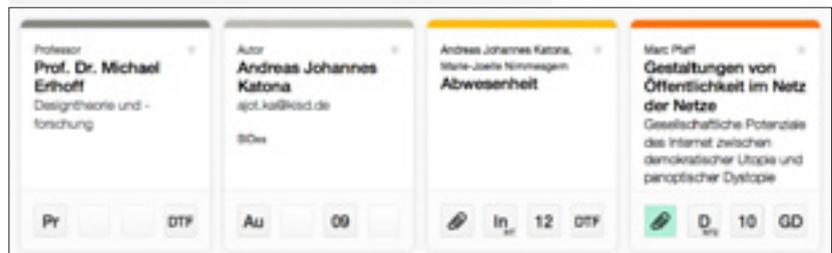
FUNKTIONSAUFRUF `ark_get_tile($post_ID, $tile_width)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$p.ID	Eintrags-ID
	string	\$t.width	Ausgabebreite in Prozent (Standardwert: '25')

RÜCKGABEWERT -

BEISPIEL `ark_get_tile(array(118,1,22470,23178), '25')`

ABB. 125 Ausgabe der Funktion `ark_get_tile()`.



ANMERKUNGEN Die Funktion wird in der Regel mit einem einzelnen ID-Parameter aufgerufen, die Darstellung von vier Karten im Beispiel dient der Illustration der diversen Ausgabe-Erzeugnisse. Haupt-Einsatzort ist das Template `ark_view_tiles.php`.

QUELLDATEI /wp-content/themes/ark/functions.php

3.1.14 ark_greeting()

Die Funktion ermittelt den Vornamen des aktuell angemeldeten Benutzers und generiert in Abhängigkeit zur Tageszeit eine personalisierte Grußzeile.

FUNKTIONSAUFRUF	<code>ark_greeting()</code>		
PARAMETER	-		
RÜCKGABEWERT	string	Zeichenkette, welche personalisierte Grußzeile enthält.	
BEISPIEL	Aufruf:	<code>ark_greeting()</code>	
	Rückgabe:	<code>Andreas, guten Tag im digitalen Archiv der KISD.</code>	
ANMERKUNGEN	Findet im Startseiten-Template (<code>ark_index.php</code>) Verwendung.		
QUELLDATEI	<code>/wp-content/themes/ark/functions.php</code>		

3.1.15 ark_search_wisecrack()

Funktion gibt eine zufällige suchbegriffbezogene Anmerkung aus.

FUNKTIONSAUFRUF	<code>ark_search_wisecrack(\$search_query)</code> DEPRECATED		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	<code>\$s.query</code>	Der eingegebene Suchbegriff
RÜCKGABEWERT	string	Antwort	
ANMERKUNGEN	Funktion ist als deprecated gekennzeichnet, sie wird nicht mehr verwendet.		
QUELLDATEI	<code>/wp-content/themes/ark/functions.php</code>		

3.1.16 ark_show_bug_button()

Funktion gibt einen CSS-platzierten Button mit Link zur Bugs-Seite aus.

FUNKTIONSAUFRUF	<code>ark_show_bug_button()</code> DEPRECATED		
PARAMETER	-		
RÜCKGABEWERT	-		
ANMERKUNGEN	Funktion ist als deprecated gekennzeichnet, sie wird nicht mehr verwendet.		
QUELLDATEI	<code>/wp-content/themes/ark/functions.php</code>		

3.1.17 ark_show_view_modes()

Diese Funktion liest den view-Parameter und den entsprechenden Cookie aus, stellt die drei Mehrfachansichten-Auswahlbuttons dar, und gibt die aktuelle Ansicht als Zeichenkette zurück.

FUNKTIONSAUFRUF `ark_show_view_modes($width)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$width	Der Funktion werden die CSS-Klassen für die Anzeige innerhalb des unsemantic-Frameworks übergeben. (Standardw.: 'grid-20 tablet-grid-20 mobile-grid-100')

RÜCKGABEWERT string Aktuelle Ansicht ('table' / 'tiles' / 'detail')

QUELLDATEI /wp-content/themes/ark/functions.php

3.1.18 ark_view_table()

Die Funktion gibt entweder den Tabellen-Kopf oder -Fuß der Mehrfachansicht **Tabelle** aus.

FUNKTIONSAUFRUF `ark_view_table($section, $parent)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$section	Tabellen-Element: 'head' Tabellenkopf 'foot' Tabellenfuß
	string	\$parent	CSS-Flag (Standardwert: 'grid-parent')

RÜCKGABEWERT -

ANMERKUNGEN Die Funktion wird für die Darstellung der Mehrfachansicht **Tabelle** genutzt.

QUELLDATEI /wp-content/themes/ark/functions.php

3.2 Helferfunktionen

3.2.1	ark_custom_menu_order()	136
3.2.2	ark_add_dashboard_widget()	136
3.2.3	ark_add_post_meta_boxes()	136
3.2.4	ark_cpt_icons()	136
3.2.5	ark_create_menu()	137
3.2.6	ark_author_set_taxonomy()	137
3.2.7	ark_lecturer_set_taxonomy()	137
3.2.8	ark_register_settings()	137
3.2.9	declare_ark_area()	138
3.2.10	declare_ark_author()	138
3.2.11	declare_ark_lecturer()	138
3.2.12	declare_ark_record()	138
3.2.13	ark_remove_menu_entries()	139
3.2.14	ark_get_author_by_name()	139
3.2.15	ark_integrity_meta_box()	139
3.2.16	ark_generate_note_xml()	140
3.2.17	ark_generate_pdf_url()	140
3.2.18	ark_cmp()	140
3.2.19	ark_get_ID_by_title()	141
3.2.20	ark_get_lecturer_by_name()	141
3.2.21	ark_get_lecturer_role()	141
3.2.22	ark_get_new_author_meta()	142
3.2.23	ark_get_valid_year_range()	142
3.2.24	ark_queue()	142
3.2.25	ark_current_page_url()	143
3.2.26	ark_sort_record_by_year()	143
3.2.27	ark_record_post_title()	143
3.2.28	ark_zahl2wort()	144

3.2.29	utf8_strrev()	144
3.2.30	utf8_strshuffle()	144
3.2.31	ark_acf_relationship_result()	145
3.2.32	ark_acf_generate_search_meta()	145
3.2.33	ark_filter_wp_title()	145
3.2.34	ark_acf_load_value()	146
3.2.35	ark_acf_update_value()	146
3.2.36	ark_permalinks()	147
3.2.37	ark_query_post_type()	147
3.2.38	ark_remove_mass_edit_bulk_action()	147
3.2.39	ark_relationship_query()	148
3.2.40	ark_set_title()	148
3.2.41	ark_update_slug()	148
3.2.42	ark_remove_row_actions()	149
3.2.43	ark_replace_excerpt_more()	149
3.2.44	ark_set_default_title()	149
3.2.45	ark_wpautop()	150
3.2.46	ark_add_rewrite_rules()	150
3.2.47	ark_admin_header()	150
3.2.48	ark_deregister_styles()	150
3.2.49	ark_dashboard_widget_function()	151
3.2.50	ark_post_meta_boxes_setup()	151
3.2.51	ark_remove_media_controls()	151
3.2.52	ark_settings_page()	151
3.2.53	ark_setup()	152
3.2.54	ark_fix_attachment_meta()	152
3.2.55	ark_fix_permalinks()	152
3.2.56	ark_generate_search_meta()	152

3.2.1 ark_custom_menu_order()

Diese Funktion implementiert eine Neusortierung der Backend-Navigation.

FUNKTIONSAUFRUF	<code>ark_custom_menu_order(\$menu_ord)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	array	\$m.ord	Array, welches eine Liste von Links zur Anzeige in der Backend-Seitennavigation enthält
RÜCKGABEWERT	array		Navigation
HOOK	Filter: custom_menu_order, menu_order		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.2 ark_add_dashboard_widget()

Diese Funktion bedient sich der Funktion `wp_add_dashboard_widget()` um ein Dashboard-Widget (Slug: `ark_dashboard_widget`) zu deklarieren.

FUNKTIONSAUFRUF	<code>ark_add_dashboard_widget()</code>		
HOOK	Action: wp_dashboard_setup		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.3 ark_add_post_meta_boxes()

Diese Funktion enthält die Deklaration der Eintrags-Integritäts-Metabox (IMB) zur Anzeige im Kontext der Eintragsmaske des Posttyps `ark_record`.

FUNKTIONSAUFRUF	<code>ark_add_post_meta_boxes()</code>		
HOOK	via <code>ark_post_meta_boxes_setup()</code>		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.4 ark_cpt_icons()

Die Funktion inkludiert CSS-Code zur korrekten Darstellung der CPT-Icons.

FUNKTIONSAUFRUF	<code>ark_cpt_icons()</code>		
HOOK	Action: admin_head		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.5 ark_create_menu()

Die Funktion fügt den Menü-Eintrag zur KISDarchiv-Einstellungsseite im Backend hinzu und initialisiert die **ark_register_settings()**-Funktion.

FUNKTIONSAUFRUF	<code>ark_create_menu()</code>
HOOK	Action: admin_menu, admin_init
QUELLDATEI	/wp-content/plugins/KISDarchiv/KISDarchive.php

3.2.6 ark_author_set_taxonomy()

Funktion deklariert die **ark_author**-eigene Taxonomie.

FUNKTIONSAUFRUF	<code>ark_author_set_taxonomy()</code>
HOOK	Action: init
QUELLDATEI	/wp-content/plugins/KISDarchiv/KISDarchive.php

3.2.7 ark_lecturer_set_taxonomy()

Funktion registriert **ark_lecturer**-eigene Taxonomie **ark_lecturer_groups**.

FUNKTIONSAUFRUF	<code>ark_lecturer_set_taxonomy()</code>
HOOK	Action: init
QUELLDATEI	/wp-content/plugins/KISDarchiv/KISDarchive.php

3.2.8 ark_register_settings()

Die Funktion deklariert die KISDarchiv eigenen Einstellungs-Datenfelder der Gruppe **ark-settings-group**:

- ark_setting_navigator_cache_time
- ark_setting_navigator_phrase
- ark_record_pdf_missing_list

FUNKTIONSAUFRUF	<code>ark_register_settings()</code>
HOOK	via ark_create_menu()
QUELLDATEI	/wp-content/plugins/KISDarchiv/KISDarchive.php

3.2.9 declare_ark_area()

Diese Funktion deklariert den Posttyp **ark_area**.

FUNKTIONSAUFRUF `declare_ark_area()`

HOOK Action: init

QUELLDATEI `/wp-content/plugins/KISDarchive/KISDarchive.php`

3.2.10 declare_ark_author()

Diese Funktion deklariert den Posttyp **ark_author**.

FUNKTIONSAUFRUF `declare_ark_author()`

HOOK Action: init

QUELLDATEI `/wp-content/plugins/KISDarchive/KISDarchive.php`

3.2.11 declare_ark_lecturer()

Diese Funktion deklariert den Posttyp **ark_lecturer**.

FUNKTIONSAUFRUF `declare_ark_lecturer()`

HOOK Action: init

QUELLDATEI `/wp-content/plugins/KISDarchive/KISDarchive.php`

3.2.12 declare_ark_record()

Diese Funktion deklariert den Posttyp **ark_record**.

FUNKTIONSAUFRUF `declare_ark_record()`

HOOK Action: init

QUELLDATEI `/wp-content/plugins/KISDarchive/KISDarchive.php`

3.2.13 ark_remove_menu_entries()

Die Funktion bereinigt die Backend-Navigation um ungenutzte Einträge.

FUNKTIONSAUFRUF `ark_remove_menu_entries()`

HOOK Action: admin_menu

QUELLDATEI /wp-content/themes/ark/functions.php

3.2.14 ark_get_author_by_name()

Funktion nimmt Autoren-Namen und gibt die korrespondierende ID zurück.

FUNKTIONSAUFRUF `ark_get_author_by_name($page_title, $output)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$p.title	Name eines Autors
	object	\$output	Postobjekt

RÜCKGABEWERT int Erste zutreffende Autoren-ID

BEISPIEL Aufruf: `ark_get_author_by_name('Philipp Mies')`
Rückgabe: `23789`

QUELLDATEI /wp-content/themes/ark/functions.php

3.2.15 ark_integrity_meta_box()

Die Funktion bestimmt die Eintragsintegrität und generiert Markup sowie CSS- und JS-Code für die korrekte Anzeige und Animation der Metabox.

FUNKTIONSAUFRUF `ark_integrity_meta_box($object, $box)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	object	\$object	Post-Objekt
	array	\$box	Metabox-Array

RÜCKGABEWERT -

HOOK via `ark_add_post_meta_boxes()`

QUELLDATEI /wp-content/plugins/KISDarchive/KISDarchive.php

3.2.16 ark_generate_note_xml()

Diese Funktion generiert eine XML-Datei mit sämtlichen Eintrags-Metadaten welche mitsamt der PDF-Datei abgelegt wird.

FUNKTIONSAUFRUF	<code>ark_generate_note_xml(\$pdf_url)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$pdf_url	URL zur PDF-Datei
RÜCKGABEWERT	-		
HOOK	via <code>ark_acf_update_value()</code>		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.17 ark_generate_pdf_url()

Die Funktion erstellt aus Metadaten den Pfad für die korrekte PDF-Ablage.

FUNKTIONSAUFRUF	<code>ark_generate_pdf_url(\$post_id, \$ark_record_pdf_url)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$p.id	Post-ID
	string	\$a.url	Pfad zur PDF-Datei
RÜCKGABEWERT	string		Generierte PDF-URL
HOOK	via <code>ark_acf_update_value()</code>		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.18 ark_cmp()

Funktion vergleicht zwei übergebene numerische Werte (a,b) und gibt einen Vergleichswert zurück.

FUNKTIONSAUFRUF	<code>ark_cmp(\$a, \$b)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$a	Numerischer Wert
	int	\$b	Numerischer Wert
RÜCKGABEWERT	int		Vergleichswert
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.19 ark_get_ID_by_title()

Die Funktion erhält einen Post-Titel und gibt die korrespondierende ID zurück.

FUNKTIONSAUFRUF	<code>ark_get_ID_by_title(\$title)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$title	Post-Titel
RÜCKGABEWERT	int		Post-ID
BEISPIEL	Aufruf:	<code>ark_get_ID_by_title('Abwesenheit')</code>	
	Rückgabe:	22470	
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.20 ark_get_lecturer_by_name()

Funktion nimmt Lehrenden-Namen und gibt die korrespondierende ID zurück.

FUNKTIONSAUFRUF	<code>ark_get_lecturer_by_name(\$page_title, \$output)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$p.title	Name eines Lehrenden
	object	\$output	Post-Objekt
RÜCKGABEWERT	int		Erste ID, welche auf Lehrendennamen zutrifft
BEISPIEL	Aufruf:	<code>ark_get_lecturer_by_name('Prof. Andreas Muxel')</code>	
	Rückgabe:	21486	
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.21 ark_get_lecturer_role()

Die Funktion nimmt ID von Lehrenden und gibt die `ark_lecturer_role` zurück.

FUNKTIONSAUFRUF	<code>ark_get_lecturer_role(\$post_ID)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$p.ID	Post-ID
RÜCKGABEWERT	string		Lehrenden-Gruppenzugehörigkeit
BEISPIEL	Aufruf:	<code>ark_get_lecturer_role(118)</code>	
	Rückgabe:	Professor	
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.22 ark_get_new_author_meta()

Die Funktion bekommt einen vollständigen Autoren-Namen übergeben und führt einen Abgleich mit einer externen Datenbank durch, um Metadaten zum eingegebenen Namen zu ermitteln (Vorname, Nachname, Matrikelnummer, Studienbeginn, Geschlecht), welche die Funktion in Arrayform zurückgibt.

FUNKTIONSAUFRUF `ark_get_new_author_meta($name)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$name	Vollständiger Name des Autors

RÜCKGABEWERT	array	Autorendaten	
---------------------	-------	--------------	--

BEISPIEL

Aufruf: `ark_get_new_author_meta('Andreas Johannes Katona')`

Rückgabe: `array(6) {`

```

    ["ark_author_firstname"]=> "Andreas Johannes"
    ["ark_author_lastname"]=> "Katona"
    ["ark_author_mnr"]=> "#####"
    ["ark_author_degree"]=> "DSN"
    ["ark_author_start"]=> "01-Sep-09"
    ["ark_author_gender"]=> "M"
  }
```

QUELLDATEI `/wp-content/themes/ark/functions.php`

3.2.23 ark_get_valid_year_range()

Funktion gibt kleinsten und größten **ark_record_year**-Wert als Array zurück.

FUNKTIONSAUFRUF `ark_get_valid_year_range($output)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	objekt	\$output	Output-Objekt

RÜCKGABEWERT	array	Jahreswerte	
---------------------	-------	-------------	--

QUELLDATEI `/wp-content/themes/ark/functions.php`

3.2.24 ark_queue()

Die Funktion bezieht die Parameter für die **add**, **remove** und **clear**-Variablen und führt entsprechende Operationen auf dem **ark_queue**-Cookie durch.

FUNKTIONSAUFRUF `ark_queue()`

QUELLDATEI `/wp-content/themes/ark/functions.php`

3.2.25 ark_current_page_url()

Funktion bezieht die aktuelle URL und gibt diese zurück.

FUNKTIONSAUFRUF	<code>ark_current_page_url()</code>		
PARAMETER	-		
RÜCKGABEWERT	string		Aktuelle Seiten-URL.
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.26 ark_sort_record_by_year()

Funktion vergleicht den für **ark_record_year** hinterlegten Wert zweier Einträge via strcmp-Funktion.

FUNKTIONSAUFRUF	<code>ark_sort_record_by_year(\$a, \$b)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	array	\$a	Eintrags-Array
	array	\$b	Eintrags-Array
RÜCKGABEWERT	int		Vergleichswert
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.27 ark_record_post_title()

Die Funktion modifiziert die Ausgabe des Post-Titel in diversen Kontexten. Für den Posttyp **ark_author** im Kontext Backend wird den Titel in der Ausgabe das Feld **ark_author_mnr** angehängt. Daneben regelt die Funktion die Trennung von Titel & Untertitel via Delimiter im Frontend, der Untertitel wird in ein span-Element der Klasse **subtitle** gekapselt.

FUNKTIONSAUFRUF	<code>ark_record_post_title(\$title, \$id)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$title	Post-Titel
	int	\$id	Post-ID
RÜCKGABEWERT	string		Post-Titel
HOOK	Filter: the_title		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.28 ark_zahl2wort()

Funktion erhält numerische Eingabe und gibt ausgeschriebene Zahl zurück.

FUNKTIONSAUFRUF	<code>ark_zahl2wort(\$int, \$genus)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$int	Numerischer Wert
	string	\$genus	Genus (Standardwert: 's')
RÜCKGABEWERT	string		Geschriebener Zahlenwert
BEISPIEL	Aufruf:	<code>ark_zahl2wort(123)</code>	
	Rückgabe:	Einhundertdreiundzwanzig	
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.29 utf8_strrev()

Funktion gibt eingegebene Zeichenkette in umgekehrter Reihenfolge zurück.

FUNKTIONSAUFRUF	<code>utf8_strrev(\$str)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$str	Zeichenkette
RÜCKGABEWERT	string		Zeichenkette
BEISPIEL	Aufruf:	<code>utf8_strshuffle('Archiv')</code>	
	Rückgabe:	vihcrA	
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.30 utf8_strshuffle()

Funktion gibt eingegebene Zeichenkette in zufälliger Reihenfolge zurück.

FUNKTIONSAUFRUF	<code>utf8_strshuffle(\$str)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$str	Zeichenkette
RÜCKGABEWERT	string		Zeichenkette
BEISPIELE	Aufruf:	<code>utf8_strshuffle('KISD')</code>	
	Rückgabe:	DKSI	
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.31 ark_acf_relationship_result()

Die Funktion implementiert die Ausgabe der Lehrgebieteskennungen im Lehrgebiets-Relationshipfeld der Eintrags-Eingabemaske.

FUNKTIONSAUFRUF `ark_acf_relationship_result($html, $post)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$html	Feldwert
	object	\$post	Post-Objekt

RÜCKGABEWERT string Feldwert

HOOK Filter: acf/fields/relationship/result

QUELLDATEI /wp-content/plugins/KISDarchive/KISDarchive.php

3.2.32 ark_acf_generate_search_meta()

Die Funktion generiert Suchmeta-Feldinhalt bei Aufruf der Eintragsmaske.

FUNKTIONSAUFRUF `ark_acf_generate_search_meta($value, $post_id, $field)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$value	Wert des Felds wie via \$_POST übergeben.
	int	\$post_id	Post-ID
	array	\$field	Feldobjekt als Array

RÜCKGABEWERT string Feldwert

HOOK Filter: acf/load_value/name=ark_record_search_meta

QUELLDATEI /wp-content/plugins/KISDarchive/KISDarchive.php

3.2.33 ark_filter_wp_title()

Filterfunktion welche sich in die Ausgabe der **wp_title()**-Funktion einhakt.

FUNKTIONSAUFRUF `ark_filter_wp_title($title)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$title	Post-Titel

RÜCKGABEWERT string Post-Titel

HOOK Filter: wp_title

QUELLDATEI /wp-content/themes/ark/functions.php

3.2.34 ark_acf_load_value()

Die Funktion stellt beim Laden des **ark_record_pdf_url**-Datenfelds fest, ob die via URL referenzierte Datei an vorgegebener Stelle vorhanden ist.

FUNKTIONSAUFRUF	<code>ark_acf_load_value(\$value, \$post_id, \$field)</code> DEPRECATED		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$value	Wert des Felds wie via \$_POST übergeben.
	int	\$post_id	Post-ID
	array	\$field	Feldobjekt als Array
RÜCKGABEWERT	string		Feldwert
ANMERKUNGEN	Funktion wird nicht mehr genutzt und gilt als deprecated.		
HOOK	Filter: acf/load_value/name=ark_record_pdf_url, Funktion deprecated		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.35 ark_acf_update_value()

Die Funktion wird mit Aktualisierung des Felds **ark_record_pdf_url** in der Backend-Eintragseingabemaske aufgerufen. Sie dient der Verschiebung der PDF-Datei in eine vorgegebene Ordnerstruktur sowie der Aktualisierung der die Datei referenzierenden Metadaten. Dabei macht sie Gebrauch von den **ark_generate_pdf_url()**- und **ark_generate_note_xml()**-Funktionen.

FUNKTIONSAUFRUF	<code>ark_acf_update_value(\$value, \$post_id, \$field)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$value	Wert des Felds wie via \$_POST übergeben.
	int	\$p.id	Post-ID
	array	\$field	Feldobjekt als Array
RÜCKGABEWERT	string		Feldwert für ark_record_pdf_url
HOOK	Filter: acf/update_value/name=ark_record_pdf_url		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.36 ark_permalink()

Die Funktion bezieht alle notwendigen Metadaten zur Generierung des Permalinks für Einträge des Typs **ark_record** und gibt den Link zurück.

FUNKTIONSAUFRUF `ark_permalink($permalink, $post, $leavename)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$p.link	Post-Permalink
	object	\$post	Post-Objekt (optional)
	bool	\$l.name	Postname beibehalten (Standardwert: false)

RÜCKGABEWERT string Post-Permalink

HOOK Filter: post_type_link

QUELLDATEI /wp-content/themes/ark/functions.php

3.2.37 ark_query_post_type()

Eine Helferfunktion für die WordPress query-Variable **ark_record**.

FUNKTIONSAUFRUF `query_post_type($query)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	object	\$query	Query-Objekt

RÜCKGABEWERT object \$query Query-Objekt

HOOK Filter: pre_get_posts

QUELLDATEI /wp-content/themes/ark/functions.php

3.2.38 ark_remove_mass_edit_bulk_action()

Die Funktion deaktiviert die »Mass Edit«-Funktion von Einträgen im Backend. Dadurch wird die korrekte Funktion KISDarchiv-eigener Funktionen gesichert.

FUNKTIONSAUFRUF `ark_remove_mass_edit_bulk_action($actions)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	array	\$actions	Array mit Dropdown-Aktionen

RÜCKGABEWERT array Array mit Dropdown-Aktionen

HOOK Filter: bulk_actions-edit-ark_record

QUELLDATEI /wp-content/themes/ark/functions.php

3.2.39 ark_relationship_query()

Modifikation des Relationship-Felds der Custom Post Types um Lehrgebiete via Eingabe des Lehrgebietskürzels (**ark_area_ID**) suchen zu können.

FUNKTIONSAUFRUF	<code>ark_relationship_query(\$args, \$field, \$post)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	array	\$args	Array mit Query-Variablen
	array	\$field	Feld aller Attribute und Einstellungen
	object	\$post	Post-Objekt
RÜCKGABEWERT	\$args		Array mit Query-Variablen
HOOK	Filter: acf/fields/relationship/query/name=ark_record_area		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.40 ark_set_title()

post_title-Felddaten-Generierung für **ark_author** und **ark_lecturer**-Masken.

FUNKTIONSAUFRUF	<code>ark_set_title(\$title)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$title	Post-Titel
RÜCKGABEWERT	string		Post-Titel
HOOK	Filter: title_save_pre		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.41 ark_update_slug()

Diese Funktion sorgt für die Generierung von Permalinks bei Post-Update.

FUNKTIONSAUFRUF	<code>ark_update_slug(\$data, \$postarr)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	array	\$data	Bereinigte Postdaten
	array	\$postarr	Post-Rohdaten
RÜCKGABEWERT	array		Postdaten
HOOK	Filter: wp_insert_post_data		
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.42 ark_remove_row_actions()

Diese Funktion entfernt die »Inline-Edit«-Funktionalität in der Listenansicht von Einträgen des Typs **ark_record** im Backend.

FUNKTIONSAUFRUF	<code>ark_remove_row_actions(\$actions, \$post)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	array	\$actions	Array mit möglichen Dropdown-Aktionen
RÜCKGABEWERT	array		Array mit möglichen Dropdown-Aktionen
HOOK	Filter: post_row_actions		
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.43 ark_replace_excerpt_more()

Die Funktion ersetzt die standardmäßige Kennzeichnung am Excerpt-Ende durch einen KISDarchiv-eigenen Bezeichner.

FUNKTIONSAUFRUF	<code>ark_replace_excerpt_more(\$more)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$more	Kennzeichnung
RÜCKGABEWERT	string		Kennzeichnung
HOOK	Filter: excerpt_more		
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.44 ark_set_default_title()

Die Funktion ersetzt den Titel-Standardwert der Posttypen **ark_author** und **ark_lecturer** durch den Hinweis, nur untenstehende Felder auszufüllen.

FUNKTIONSAUFRUF	<code>ark_set_default_title(\$title, \$post)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$title	Post-Titel
	object	\$post	Post-Objekt
RÜCKGABEWERT	string		Post-Titel
HOOK	Filter default_title		
QUELLDATEI	/wp-content/plugins/KISDarchive/KISDarchive.php		

3.2.45 ark_wpautop()

Die Funktion deaktiviert automatische Zeilenumbrüche im Rich-Text-Editor.

FUNKTIONSAUFRUF	<code>ark_wpautop(\$content)</code>		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$content	Inhalt des Felds post_content
RÜCKGABEWERT	string	Inhalt	
HOOK	Filter: the_content		
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.46 ark_add_rewrite_rules()

Die Funktion definiert WordPress **rewrite_tags** zu Metadatenfeldern und modifiziert die Permalink-Struktur des Datentyps ark_record.

FUNKTIONSAUFRUF	<code>ark_add_rewrite_rules()</code>		
HOOK	Action: init		
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.47 ark_admin_header()

Diese Funktion verhindert via jQuery-Intervention, dass die **post_title**-Felder der Posttypen **ark_author** und **ark_lecturer** im Backend schreibbar sind, da diese Felder ihren Wert aus diversen Metadaten-Felder beziehen.

FUNKTIONSAUFRUF	<code>ark_admin_header()</code>		
HOOK	Action: admin_head		
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.48 ark_deregister_styles()

Funktion zur Deaktivierung diverser ungewollter CSS-Einträge im Backend.

FUNKTIONSAUFRUF	<code>ark_deregister_styles()</code>		
HOOK	Action: wp_print_styles		
QUELLDATEI	/wp-content/themes/ark/functions.php		

3.2.49 ark_dashboard_widget_function()

Diese Funktion liest das Optionsfeld **ark_record_pdf_missing_list** aus, und bereitet die darin als IDs enthaltenen Einträge für die Anzeige innerhalb eines entsprechenden Dashboard-Widgets auf.

FUNKTIONSAUFRUF	<code>ark_dashboard_widget_function()</code>
HOOK	via <code>ark_add_dashboard_widget()</code>
QUELLDATEI	<code>/wp-content/plugins/KISDarchive/KISDarchive.php</code>

3.2.50 ark_post_meta_boxes_setup()

Funktion deklariert die Funktion `ark_add_post_meta_boxes()` via `add_meta_boxes`-Hook.

FUNKTIONSAUFRUF	<code>ark_post_meta_boxes_setup()</code>
HOOK	<code>load-post.php</code> , <code>load-post-new.php</code>
QUELLDATEI	<code>/wp-content/plugins/KISDarchive/KISDarchive.php</code>

3.2.51 ark_remove_media_controls()

Funktion verhindert die Ausgabe der Media-Buttons im Eintrags-Editor des Backends sofern der bearbeitete Eintrag den Posttyp **ark_record** trägt.

FUNKTIONSAUFRUF	<code>ark_remove_media_controls()</code>
HOOK	Action: <code>admin_head</code>
QUELLDATEI	<code>/wp-content/plugins/KISDarchive/KISDarchive.php</code>

3.2.52 ark_settings_page()

Funktion zur Darstellung der KISDarchiv-Einstellungsseite im Backend.

FUNKTIONSAUFRUF	<code>ark_settings_page()</code>
HOOK	via <code>ark_create_menu()</code>
QUELLDATEI	<code>/wp-content/plugins/KISDarchive/KISDarchive.php</code>

3.2.53 ark_setup()

Funktion initialisiert Text-Domäne und Post-Thumbnail-Theme-Support.

FUNKTIONSAUFRUF `ark_setup()`

HOOK Action: `after_setup_theme`

QUELLDATEI `/wp-content/themes/ark/functions.php`

3.2.54 ark_fix_attachment_meta()

Funktion zur Neugenerierung von Attachment-Metadaten in der Datenbank.

FUNKTIONSAUFRUF `ark_fix_attachment_meta()`

ANMERKUNGEN Wartungsfunktion, findet im Regelbetrieb keine Verwendung.

QUELLDATEI `/wp-content/themes/ark/functions.php`

3.2.55 ark_fix_permalinks()

Die Funktion generiert Permalinks zu als ID-Array übergebenen Einträgen und schreibt diese in die WordPress Post-Tabelle.

FUNKTIONSAUFRUF `ark_fix_permalinks($ids)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	array(int)	\$ids	Array numerischer Eintrags-IDs

ANMERKUNGEN Wartungsfunktion, findet im Regelbetrieb keine Verwendung.

QUELLDATEI `/wp-content/themes/ark/functions.php`

3.2.56 ark_generate_search_meta()

Die Funktion erhält ein Array von Post-IDs und schreibt die Suchmeta-Daten aller übergebenen Posts in die Datenbank (Feld: `ark_record_search_meta`).

FUNKTIONSAUFRUF `ark_generate_search_meta($ids)`

PARAMETER	TYP	NAME	BESCHREIBUNG
	array(int)	\$ids	Array numerischer Eintrags-IDs

ANMERKUNGEN Wartungsfunktion, findet im Regelbetrieb keine Verwendung.

QUELLDATEI `/wp-content/themes/ark/functions.php`

3.3 Includes

3.3.1	ark-dir-ext.php	154
3.3.2	ark-get-adv-search-results.php	154
3.3.3	ark-get-author-degree-by-name.php	154
3.3.4	ark-get-lecturer-area-by-name.php	155
3.3.5	ark-get-lecturer-id-by-name.php	155
3.3.6	ark-get-word-from-number.php	155
3.3.7	ark-pdf-url-check.php	156
3.3.8	ark-record-author-dir.php	156
3.3.9	ark-search-author.php	156
3.3.10	search_ark_record_author.php	157
3.3.11	search_ark_record_lecturer.php	157
3.3.12	search_ark_record_tag.php	157

3.3.1 ark-dir-ext.php

Generierung einer verschachtelten Liste aller im KISDarchiv erfassten Einträge zur Anzeige im Navigator.

AUFRUF	via PHP Include
IMPLEMENTIERUNG	Navigator / ark_navigator.php:11
POSITION	/wp-content/plugins/KISDarchive/ark-dir-ext.php

3.3.2 ark-get-adv-search-results.php

Diese Funktion wird von der Metasuche aufgerufen, um deren Suchergebnisse zu generieren. Sie durchläuft mehrere Stufen:

1. Alle queryrelevanten POST-Variablen werden ausgelesen.
2. Ein MySQL-Query wird zusammengefügt
3. Das Query wird ausgeführt
4. Je nach gesetzten view- und offset-Parametern wird eine Teilansicht der Ergebnisse in entsprechender Mehrfachansicht generiert und ausgegeben.
5. Die Paginierung und der jQuery-Code für die korrekte Verarbeitung von Klickbefehlen innerhalb der Ergebnisse wird erzeugt.

AUFRUF	via jQuery/AJAX
IMPLEMENTIERUNG	Metasuche / ark_search.php:565
POSITION	/wp-content/plugins/KISDarchive/ark-get-adv-search-results.php

3.3.3 ark-get-author-degree-by-name.php

Helferfunktion, welche den Namen eines Autors als Parameter akzeptiert und den dem Autor zugeordneten Studiengang ausgibt.

AUFRUF	via jQuery/AJAX						
PARAMETER	<table> <thead> <tr> <th>TYP</th> <th>NAME</th> <th>BESCHREIBUNG</th> </tr> </thead> <tbody> <tr> <td>string</td> <td>\$name</td> <td>Name des Autoren/der Autorin</td> </tr> </tbody> </table>	TYP	NAME	BESCHREIBUNG	string	\$name	Name des Autoren/der Autorin
TYP	NAME	BESCHREIBUNG					
string	\$name	Name des Autoren/der Autorin					
RÜCKGABE	-						
IMPLEMENTIERUNG	Arbeit einreichen-Maske / ark-1.0.js:773						
POSITION	/wp-content/plugins/KISDarchive/ark-get-author-degree-by-name.php						

3.3.4 ark-get-lecturer-area-by-name.php

Helferfunktion, die den Namen eines Lehrenden als Argument erhält und die ID des dem Lehrenden zugeordneten Lehrgebiets zurückgibt.

AUFRUF	via jQuery/AJAX		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$name	Name des Lehrenden
RÜCKGABE	-		
IMPLEMENTIERUNG	Arbeit einreichen-Maske / ark-1.0.js:836		
POSITION	/wp-content/plugins/KISDarchive/ark-get-lecturer-area-by-name.php		

3.3.5 ark-get-lecturer-id-by-name.php

Helferfunktion, welche den Namen eines Lehrenden als Parameter akzeptiert und die zugehörige Post-ID samt Namen zurückgibt.

AUFRUF	via jQuery/AJAX		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$name	Name des Lehrenden
RÜCKGABE	-		
IMPLEMENTIERUNG	Metasuche / ark_search.php:245		
POSITION	/wp-content/plugins/KISDarchive/ark-get-lecturer-id-by-name.php		

3.3.6 ark-get-word-from-number.php

Implementierung der **ark_zahl2wort()**-Funktion für den Aufruf via jQuery.

AUFRUF	via jQuery/AJAX		
PARAMETER	TYP	NAME	BESCHREIBUNG
	int	\$int	Zahlenwert
RÜCKGABE	-		
IMPLEMENTIERUNG	Lesezeichenseite / ark-1.0.js:1090		
POSITION	/wp-content/plugins/KISDarchive/ark-get-word-from-number.php		

3.3.7 ark-pdf-url-check.php

Seitenweiter PDF-URL-Linkchecker

AUFRUF direkt / via HTTP **DEPRECATED**

POSITION /wp-content/plugins/KISDarchive/ark-pdf-url-check.php

3.3.8 ark-record-author-dir.php

Diese Datei generiert eine tabellarische Auflistung der externen Autoren-datenbank. Sie spielt im Regelbetrieb keine Rolle.

AUFRUF direkt / via HTTP

IMPLEMENTIERUNG -

POSITION /wp-content/plugins/KISDarchive/ark-record-author-dir.php

3.3.9 ark-search-author.php

Diese Funktion findet innerhalb der Uploadmaske Verwendung. Sie führt eine Datenbankabfrage auf zwei separate Tabellen aus – der KISDarchiv-eigenen und einer externen Autoredatenquelle – und fungiert als Autocomplete-Datenquelle für das Autorenfeld der Arbeit einreichen-Maske.

AUFRUF via jQuery/AJAX

PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$term	Autorenname

RÜCKGABE -

IMPLEMENTIERUNG Arbeit einreichen-Maske / ark-1.0.js:767

POSITION /wp-content/plugins/KISDarchive/ark-search-author.php

TODO ark_td_01 (Siehe S. 229)

3.3.10 search_ark_record_author.php

Das Include generiert ein Autoren-JSON-Array auf Basis einer Feldeingabe zur Verarbeitung innerhalb der jQuery-Autocomplete-Funktionen.

AUFRUF	via jQuery/AJAX		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$term	Feldeingabe (Name eines Lehrenden) des Besuchers
RÜCKGABE	array		Array mit Datenpaaren 'id' (ID) und 'label' (post_title)
IMPLEMENTIERUNG	Autoren-Einzelansicht / ark-1.0.js:231		
POSITION	/wp-content/plugins/KISDarchive/search_ark_record_author.php		

3.3.11 search_ark_record_lecturer.php

Das Include generiert ein Prüfer-JSON-Array auf Basis einer Feldeingabe zur Verarbeitung innerhalb der jQuery-Autocomplete-Funktionen.

AUFRUF	via jQuery/AJAX		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$term	Feldeingabe (Name eines Lehrenden) des Besuchers
RÜCKGABE	array		Array mit Datenpaaren 'id' (ID) und 'label' (post_title)
IMPLEMENTIERUNG	Prüfer-Einzelansicht / ark-1.0.js:281 Arbeit einreichen-Maske / ark-1.0.js:813		
POSITION	/wp-content/plugins/KISDarchive/search_ark_record_lecturer.php		

3.3.12 search_ark_record_tag.php

Das Include generiert ein Schlagwort-JSON-Array auf Basis einer Feldeingabe zur Verarbeitung innerhalb der jQuery-Autocomplete-Funktionen.

AUFRUF	via jQuery/AJAX		
PARAMETER	TYP	NAME	BESCHREIBUNG
	string	\$term	Feldeingabe (Schlagwort) des Besuchers
RÜCKGABE	array		Array mit Datenpaaren 'id' (ID) und 'label' (post_title)
IMPLEMENTIERUNG	Schlagwort-Einzelansicht / ark-1.0.js:328		
POSITION	/wp-content/plugins/KISDarchive/search_ark_record_tag.php		

4 Cookies

4.0 Einleitung

An zweierlei Stelle macht das KISDarchiv von Browser-Cookies Gebrauch, um Daten nutzerseitig abzulegen. Dies soll an dieser Stelle dokumentiert werden. Die Cookies werden mit einem Verfallsdatum von einem Monat geschrieben.

4.1 ark_view

Aktive Mehrfachansicht

Der Cookie **ark_view** fasst die aktive Mehrfachansicht.

WERT Aktive Mehrfachansicht
 Beispiel: `tiles`
 Mögliche Werte: `detail, table, tiles`

Der Cookie wird geschrieben, sobald ein Nutzer von einer zur anderen Mehrfachansicht umschaltet. Beim nächsten Laden einer sich Mehrfachansichten bedienenden Seite wird der Cookie wiederum ausgelesen um eine konsistente Darstellung zu gewährleisten. Der Cookie verfällt nach 31 Tagen.

4.2 ark_queue

Lesezeichen-Liste

WERT Kommaseparierte Eintrags-IDs
 Beispiel: `22420%2C22421%2C23196%2C23178%2C19283`
 Hinweis: `'%2C'` entspricht der hexadezimalen Notation eines Kommas.

Der **ark_queue**-Cookie fasst Arbeiten, welche den Lesezeichen hinzugefügt wurden, in Form ihrer IDs. Die Interaktion mit der Lesezeichen-Auswahl geschieht über die eintragungseigenen Lesezeichen-Buttons, welche via jQuery Cookie Plugin asynchron Operationen auf dem Cookie durchführen, um Wartezeiten durch häufiges Neu-Laden der Seite zu eliminieren.

Der Cookie wird beim ersten Hinzufügen einer Arbeit zu den Lesezeichen geschrieben und beim Entfernen der letzten Arbeit oder Leeren der Lesezeichenliste via der interaktiven Headline auf der Lesezeichen-Seite entfernt. Leert der Nutzer die Lesezeichen-Liste nicht, so bleibt der Cookie vom Zeitpunkt seiner Erstellung an für 31 Tage gültig. Darüber hinaus lässt sich die aktuelle Lesezeichen-Liste via interaktiver Headline in Linkform exportieren, um sie zu einem späteren Zeitpunkt wieder aufzurufen oder zu verschicken.

IMPLEMENTIERUNG Die Hervorhebungsfunktion, welche die Lesezeichen-Buttons entsprechend der im Cookie enthaltenen IDs in aktivierten Zustand versetzt, deren Aufruf, sowie die Funktion zur Verarbeitung aller übrigen diesen Cookie betreffenden Funktionen finden sich in der JavaScript-Datei **ark-1.0.js** (Zeile 943-1117).

5 JavaScript

5.0 Einleitung

Das KISDarchiv macht vielerorts Gebrauch von clientseitigen Skripten. Diese sind teilweise in der zentralen JavaScript-Datei des ark-Themes zu finden (`wp-content/themes/ark/assets/ark-1.0.js`), teilweise Inline in einzelnen Templates verortet; etwa im Falle der Arbeit einreichen-Maske oder der Metasuche.

Die clientseitigen Skripte sind der Übersicht halber mit den Tempaltes innerhalb derer sie auftreten zusammen dokumentiert. An dieser Stelle werden lediglich die JavaScript-Bibliotheken und Plugins referenziert, welche im Header inkludiert die Grundlage für alle clientseitigen Operationen bilden.

5.1 Bibliotheken und Plugins

Folgende Bibliotheken und Plugins kommen im KISDarchiv zum Einsatz:

NAME	DATEI	INKLUDIERT IN	SEITE
JQUERY	jquery-1.9.1.min.js	header.php	160
JQUERY MIGRATE	jquery-migrate-1.1.1.min.js	header.php	160
JQUERY UI	jquery-ui.js	header.php	161
MAPATTRIBUTES JQUERY PLUGIN	jquery.columnview.js (Z.1-25)	header.php	160
JQUERY COLUMNVIEW	jquery.columnview.js	header.php	160
JQUERY DROPDOWN	jquery.dropdown.js	header.php	161
JQUERY TAGSINPUT	jquery.tagsinput.min.js	header.php	161
JQUERY COOKIE	jquery.cookie.js	header.php	161
JQUERY FANCYBOX	jquery.fancybox.pack.js	header.php	161
JQUERY VALIDATION PLUGIN	jquery.validate.min.js	header.php	161
JQUERY BITLY	jquery.bitly.js	header.php	162
JQUERY SORTELEMENTS	jQuery.fn.sortelements	ark_integrity.php	162
JQBARGRAPH	jqBarGraph.1.1.min.js	ark_statistics.php	162
GOOGLE CHARTS	google.com/jsapi	ark_statistics.php	162
JQUERY IFRAME TRANSPORT PLUGIN	jquery.iframe.transport.js	ark_upload.php	162
JQUERY FILE UPLOAD PLUGIN	jquery.fileupload.js	ark_upload.php	163
JQUERY HIGHLIGHT PLUGIN	ark-1.0.js (Z. 1289-1356)	-	163
JQUERY BBQ LIBRARY	ark-1.0.js (Z. 1398-1406)	-	163
JQUERY HASHCHANGE EVENT	ark-1.0.js (Z. 1407-1415)	-	163

5.1.1 jQuery

Das Standard jQuery Framework, welches die Grundlage für alle verwendeten Bibliotheken und Plugins bildet.

VERSION	1.9.1
AUTOR	jQuery Foundation, Inc.
DATEI	/wp-content/themes/ark/assets/js/jquery-1.9.1.min.js
INKLUSIONSPUNKT	header.php

5.1.2 jQuery Migrate

jQuery Migrate stellt Funktionalitäten älterer jQuery-Versionen wieder her.

VERSION	1.1.1
AUTOR	jQuery Foundation, Inc.
DATEI	/wp-content/themes/ark/assets/js/jquery-migrate-1.1.1.min.js
INKLUSIONSPUNKT	header.php

5.1.3 jQuery UI

Die jQuery User Interface Bibliothek

VERSION	1.10.1
AUTOR	jQuery Foundation, Inc.
DATEI	/wp-content/themes/ark/assets/js/jquery-ui.js
INKLUSIONSPUNKT	header.php

5.1.4 mapAttributes jQuery Plugin

Plugin, welches alle Attribute eines Elements anzeigt (vgl. PHP var_dump)

VERSION	1.0.0
AUTOR	Michael Riddle
DATEI	/wp-content/themes/ark/assets/js/jquery.columnview.js (Z.1-25)
INKLUSIONSPUNKT	header.php

5.1.5 jQuery Columnview

Das jQuery Columnview-Plugin wird herangezogen, um die Mehrspaltenansicht im Navigator zu generieren.

VERSION	1.3
AUTOR	Chris Yates / Manuel Odendahl / James Roberts
DATEI	/wp-content/themes/ark/assets/js/jquery.columnview.js
INKLUSIONSPUNKT	header.php

5.1.6 jQuery Dropdown

jQuery Dropdown erweitert das KISDarchiv um kontextuelle Pop-up-Menüs welche die Basis für die interaktive Headline bilden.

VERSION	1.0.0
AUTOR	Cory LaViska
DATEI	/wp-content/themes/ark/assets/js/jquery.dropdown.js
INKLUSIONSPUNKT	header.php

5.1.7 jQuery Tagsinput

Das jQuery-Plugin für Schlagwort-Eingabeelement im Frontend.

VERSION	1.3.3
AUTOR	XOXCO, Inc.
DATEI	/wp-content/themes/ark/assets/js/jquery.tagsinput.min.js
INKLUSIONSPUNKT	header.php

5.1.8 jQuery Cookie

Ein jQuery Plugin zum Lesen, Schreiben und Löschen von Cookies.

VERSION	1.3.1
AUTOR	Klaus Hartl
DATEI	/wp-content/themes/ark/assets/js/jquery.cookie.js
INKLUSIONSPUNKT	header.php

5.1.9 jQuery fancybox

jQuery-Portierung des Lightbox-Plugins fancybox.

VERSION	2.1.5
AUTOR	Janis Skarnelis
DATEI	/wp-content/themes/ark/assets/js/jquery.fancybox.pack.js
INKLUSIONSPUNKT	header.php

5.1.10 jQuery Validation Plugin

jQuery-Plugin zur Validierung von Eingabefeldern.

VERSION	1.11.1
AUTOR	Jörn Zaefferer
DATEI	/wp-content/themes/ark/assets/js/jquery.validate.min.js
INKLUSIONSPUNKT	header.php

5.1.11 jQuery Bitly

jQuery-Plugin zur Shortlink-Generierung via bit.ly. Das Plugin soll für die Generierung von Lesezeichen-Shortlinks herangezogen werden.

VERSION 0.5b
AUTOR Ruslanas Balčiūnas
DATEI /wp-content/themes/ark/assets/js/query.bitly.js
INKLUSIONSPUNKT header.php

5.1.12 jQuery sortElements

Funktion zur Sortierung von DOM-Elementen.

VERSION 0.11
AUTOR James Padosley
DATEI jQuery.fn.sortelements via Inline-Definition in
/wp-content/themes/ark/ark_integrity.php (Z.166-230)

5.1.13 jqBarGraph

jQuery-Plugin zur Generierung von Balkendiagrammen.
Das Plugin findet derzeit keine Verwendung im KISDarchiv.

VERSION 1.1
AUTOR Ivan Lazarevic
DATEI /wp-content/themes/ark/assets/js/jqBarGraph.1.1.min.js
INKLUSIONSPUNKT ark_statistics.php (Z.5)

5.1.14 Google Charts

Googles Bibliothek zur Generierung von Diagrammen.
Das Plugin kommt auf der Statistik-Seite zum Einsatz

AUTOR Google, Inc.
DATEI <https://www.google.com/jsapi>
INKLUSIONSPUNKT ark_statistics.php (Z.159)

5.1.15 jQuery Iframe Transport Plugin

Unterstützung von Datenupload via iFrame.

VERSION 1.7
AUTOR Sebastian Tschan
DATEI /wp-content/themes/ark/assets/js/jquery.iframe.transport.js
INKLUSIONSPUNKT ark_upload.php (Z.599)

5.1.16 jQuery File Upload Plugin

Dateiupload-Widget, welches die Auswahl mehrerer Dateien, Drag und Drop, Fortschrittsbalken, Validierung, Vorschaubilder, Audio und Video erlaubt.

VERSION 5.40.0
AUTOR Sebastian Tschan
DATEI /wp-content/themes/ark/assets/js/jquery.fileupload.js
INKLUSIONSPUNKT ark_upload.php (Z.601)

5.1.17 jQuery Highlight Plugin

Plugin zur Hervorhebung bestimmter DOM-Elemente nach Textinhalt.

VERSION 4
AUTOR Johann Burkhard
DATEI /wp-content/themes/ark/assets/js/ark-1.0.js (Z.1289-1356)

5.1.18 jQuery BBQ: Back Button & jQuery Library

Plugin für bookmarkbare HTML5-basierte Hashchange-History.
Das Plugin findet derzeit keine Verwendung.

VERSION 1.2.1
AUTOR Ben Alman
DATEI /wp-content/themes/ark/assets/js/ark-1.0.js (Z.1398-1406)

5.1.19 jQuery hashchange event

Ein Subplugin von jQuery BBQ.

VERSION 1.2
AUTOR Ben Alman
DATEI /wp-content/themes/ark/assets/js/ark-1.0.js (Z.1407-1415)

6 Templates

6.0 Einleitung

WordPress-Themes setzen sich aus einer Reihe von Templates zusammen, welche sich in der so genannten Template-Hierarchie angeordnet lassen. Wir unterscheiden zwischen vier Template-Typen in zwei Kategorien:

6.0.1 Seitentemplates

Seitentemplates bilden die Basis für die Generierung kompletter Seiten.

WordPress-Templates

WordPress-Templates bilden die Grundbestandteile des Themes. Innerhalb der Template-Hierarchie dienen sie als Schablonen, um bestimmte Seitentypen darzustellen. So existieren standardmäßig Templates, welche die Darstellung von Seiten, Posts nach Schlagwort, Kategorie oder Chronologie, oder etwa Suchergebnisse abbilden.

KISDarchiv-Templates

KISDarchiv-Templates ergänzen die WordPress-Templates um spezifische Seitentypen. So wurden beispielsweise für die Darstellung der diversen Posttypen (**ark_author**, **ark_lecturer**) spezielle Templates geschaffen. Module wie Navigator oder Uploadmaske wurden ebenfalls auf diese Weise implementiert.

6.0.2 Subtemplates

Subtemplates werden in Seitentemplates referenziert und bilden wiederkehrende Funktionalitäten ab.

WordPress-Subtemplates

WordPress-Subtemplates bilden Funktionalitäten ab, die innerhalb anderer Templates immer wieder Verwendung finden. So sind etwa der Seitenkopf beziehungsweise -fuß in eigene Subtemplates gekapselt, sodass Modifikationen an diesen Funktionen an zentraler Stelle ausgeführt werden können, was sich wiederum auf sämtliche Templates auswirkt, innerhalb derer diese eingebunden werden.

KISDarchiv-Subtemplates

KISDarchiv-Subtemplates bilden spezielle wiederkehrende Elemente ab. So wurden etwa die drei Mehrfachansichten **Detail/Liste/Karten** als eigene Templates realisiert, die wiederum von Seitentemplates aus bedarfsweise referenziert werden können.

Beispiel **Templatedokumentation**

Die Templates innerhalb dieser API werden nach dem folgenden Muster dargestellt:

ark_template_name.php

Templatenname

Ein Beschreibungstext gibt Aufschluss über den Template-Zweck.

BEZÜGE Eine Liste von Templates und Includes, die aus diesem Template heraus eingebunden werden. Dargestellt sind Einbettungen ersten Grades.

JAVASCRIPT Sofern das Template von clientseitigen Skripten Gebrauch macht, werden diese hier aufgeführt. Dabei steht an erster Stelle der Ort des Skripts, also ob dieses innerhalb des Templates selbst (»Inline«) oder von externer Stelle referenziert wird. Externe Referenzen werden anhand ihrer Dateinamen aufgeführt. Daneben findet sich eine numerische Angabe des relevanten Zeilenbereichs und eine kurze Schilderung der dort abgebildeten Funktionalität.

INKLUDIERT IN Handelt es sich um ein Subtemplate, so werden an dieser Stelle die Templates referenziert, die sich auf dieses Subtemplate beziehen. Diese Angabe bildet somit das Gegenstück zum Bezüge-Wert.

POSITION Die Position verortet das Template innerhalb der WordPress-Dateistruktur.

TODO Mitunter enthalten Templates bereits einen Verweis auf ihnen zugeordnete TODO-Einträge im vierten Teil dieser Arbeit. (Siehe S. 229)

Eine Übersicht aller Templates sowie deren Einordnung nach Template-Typ findet sich auf der folgenden Seite. Die Templatetypen werden dabei wie folgt abgekürzt:

KT KISDarchiv-Template
WT WordPress-Template
KST KISDarchiv-Subtemplate
WST WordPress-Subtemplate

Template-Übersicht

	DATEI	TYP	MAPPING	SEITE
STARTSEITE	ark_index.php	KT	/	168
NAVIGATOR	ark_navigator.php	KT	/navigator/	169
METASUCHE	ark_search.php	KT	/metasuche/	170
SCHLAGWORT-ÜBERSICHT	ark_tags.php	KT	/tags/	171
EINZELANSICHT AUTOR	single-ark_author.php	KT	/autor/~	172
EINZELANSICHT PRÜFER	single-ark_lecturer.php	KT	/dozent/~	173
AUTORENVERZEICHNIS	ark_author_directory.php	KT	/autorinnen-und-autoren/	174
STATISTIK	ark_statistics.php	KT	/statistik/	175
LESEZEICHEN	ark_queue.php	KT	/lesezeichen/	176
PDF-ANSICHT	ark_pdf_view.php	KT	/pdf/	177
ARBEIT EINREICHEN	ark_upload.php	KT	/upload/	178
INTEGRITÄT	ark_integrity.php	KT	/integritaet/	179
ANALOGES ARCHIV	ark_about.php	KT	/das-analoge-archiv/	180
BUGS	ark_bugs.php	KT	/bugs/	181
UPLOAD ABGESCHLOSSEN	ark_upload_complete.php	KT	/upload/danke/	182
	DATEI	TYP		SEITE
SUCHERGEBNISSE	search.php	WT		185
EINZELANSICHT SCHLAGWORT	tag.php	WT		186
FEHLERSEITE	404.php	WT		187
EINZELANSICHT	single.php	WT		188
	DATEI	TYP		SEITE
TEMPLATE AUTOR	ark_author.php	KST		190
TEMPLATE LEHRENDER	ark_lecturer.php	KST		191
MEHRFACHANSICHT: DETAIL	ark_view_detail.php	KST		192
MEHRFACHANSICHT: LISTE	ark_view_table.php	KST		193
MEHRFACHANSICHT: KARTEN	ark_view_tiles.php	KST		194
PDF	pdf.php	KST		195
KEIN PDF	no_pdf.php	KST		196
	DATEI	TYP		SEITE
SEITENKOPF	header.php	WST		198
SEITENFUSS	footer.php	WST		198
SUCHFELD	searchform.php	WST		198
EINTRAG	entry.php	WST		199
EINTRAGS-INHALT	entry-content.php	WST		200
EINTRAGS-ZUSAMMENFASSUNG	entry-summary.php	WST		200
EINTRAGS-META	entry-meta.php	WST		200
EINTRAGS-FUSS, EINZELANSICHT	entry-footer-single.php	WST		200
PAGINIERUNG UNTEN	nav-below.php	WST		201

6.1 KISDarchiv-Templates

	DATEI	MAPPING	SEITE
STARTSEITE	ark_index.php	/	168
NAVIGATOR	ark_navigator.php	/navigator/	169
METASUCHE	ark_search.php	/metasuche/	170
SCHLAGWORT-ÜBERSICHT	ark_tags.php	/tags/	171
EINZELANSICHT AUTOR	single-ark_author.php	/autor/-	172
EINZELANSICHT PRÜFER	single-ark_lecturer.php	/dozent/-	173
AUTORENVERZEICHNIS	ark_author_directory.php	/autorinnen-und-autoren/	174
STATISTIK	ark_statistics.php	/statistik/	175
LESEZEICHEN	ark_queue.php	/lesezeichen/	176
PDF-ANSICHT	ark_pdf_view.php	/pdf/	177
ARBEIT EINREICHEN	ark_upload.php	/upload/	178
INTEGRITÄT	ark_integrity.php	/integritaet/	179
ANALOGES ARCHIV	ark_about.php	/das-analoge-archiv/	180
BUGS	ark_bugs.php	/bugs/	181
UPLOAD ABGESCHLOSSEN	ark_upload_complete.php	/upload/danke/	182

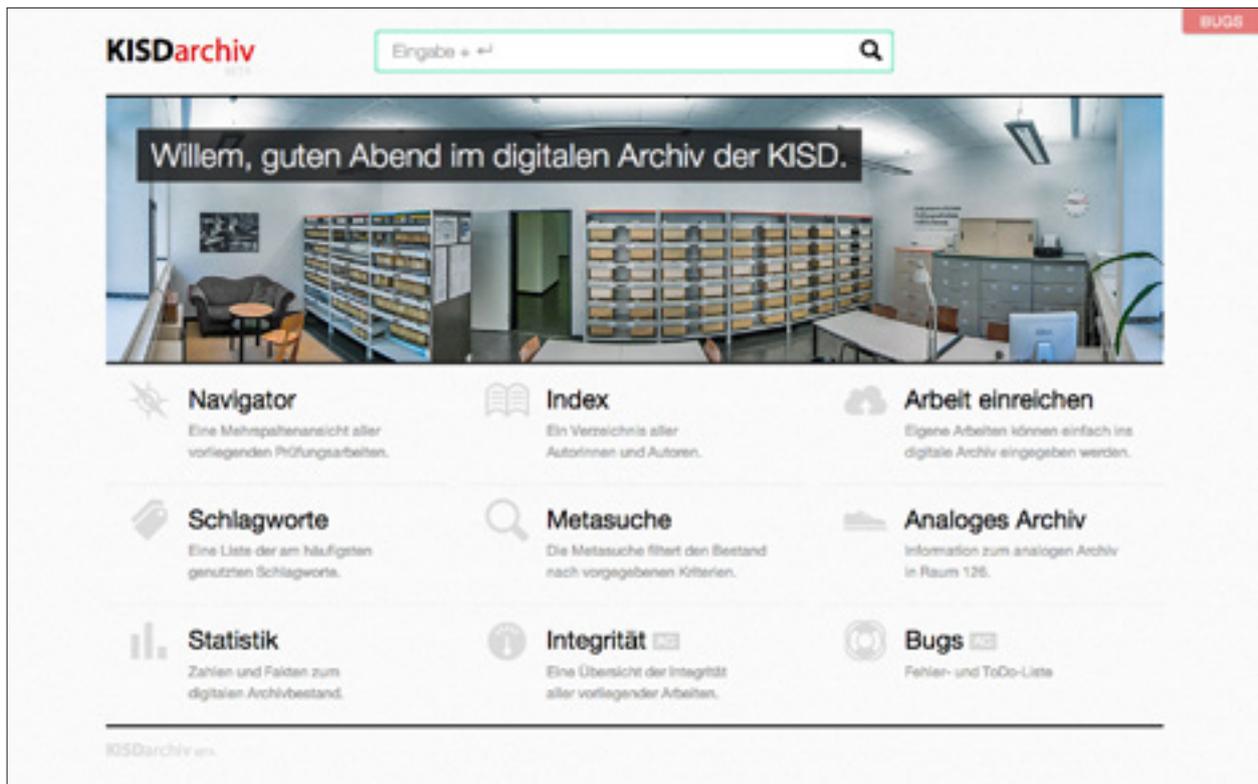


ABB.126 Startseite

6.1.1 ark_index.php Startseite

Das Startseitentemplate enthält die Headergrafik inklusive personalisiertem Gruß und Uhrzeitanzeige sowie ein Raster mit Links zu den diversen Modulen.

Das Template enthält Inline CSS- und JavaScript-Code zur korrekten Darstellung der analogen Uhr in der Header-Grafik.

BEZÜGE header.php
footer.php

JAVASCRIPT Inline (Z.41-74) Funktion der analogen Uhr innerhalb der Headergrafik.

POSITION /wp-content/themes/ark/ark_index.php

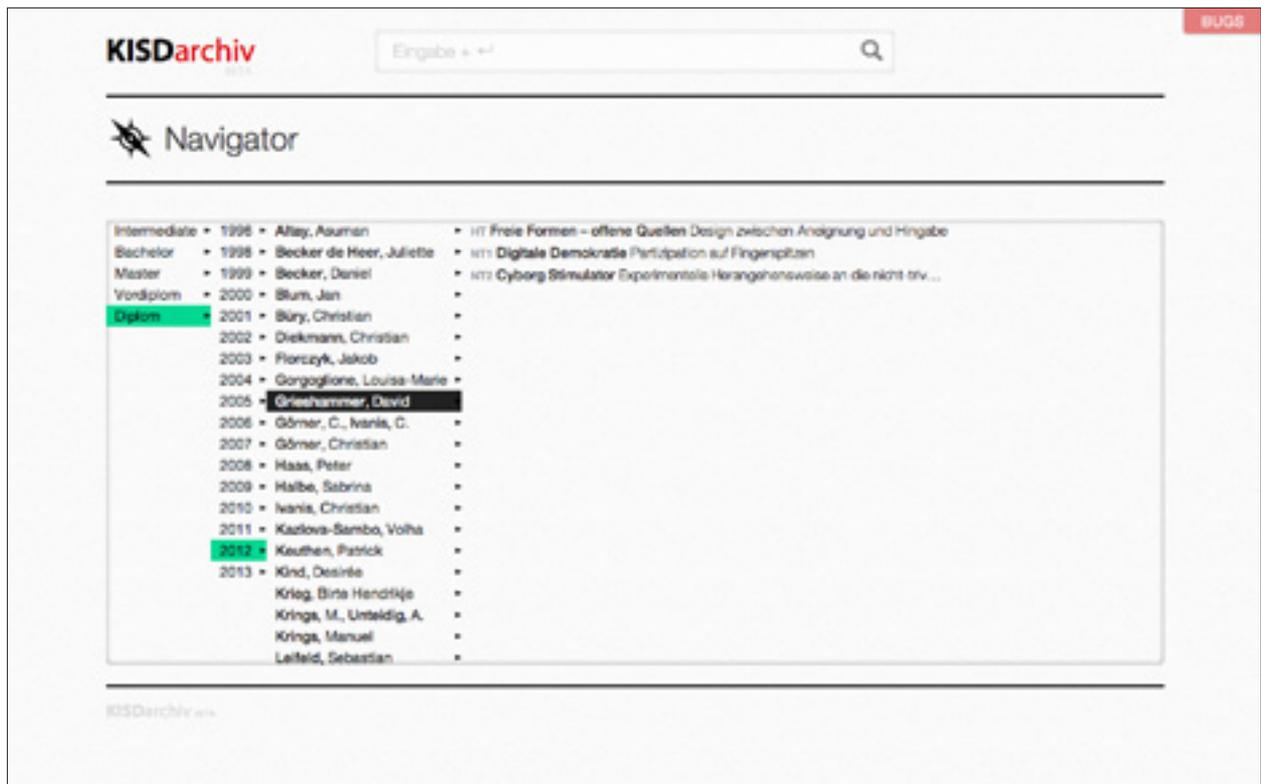


ABB.127 Navigator

6.1.2 ark_navigator.php

Navigator

Das Navigator-Template zeigt lediglich eine Ladeanimation an und inkludiert desweiteren die Datei **ark-dir-ext.php** aus dem Plugin-Verzeichnis, welche alle weiteren Operationen in der Navigator-Darstellung übernimmt.

BEZÜGE header.php
 footer.php
 ark-dir-ext.php (inkludiert)

JAVASCRIPT ark-1.0.js (Z.107-111) Initialisierung des Columnview-Plugins,
 Ausblenden der Lade-Animation,
 Einblenden der Spaltenansicht.

POSITION /wp-content/themes/ark/ark_navigator.php

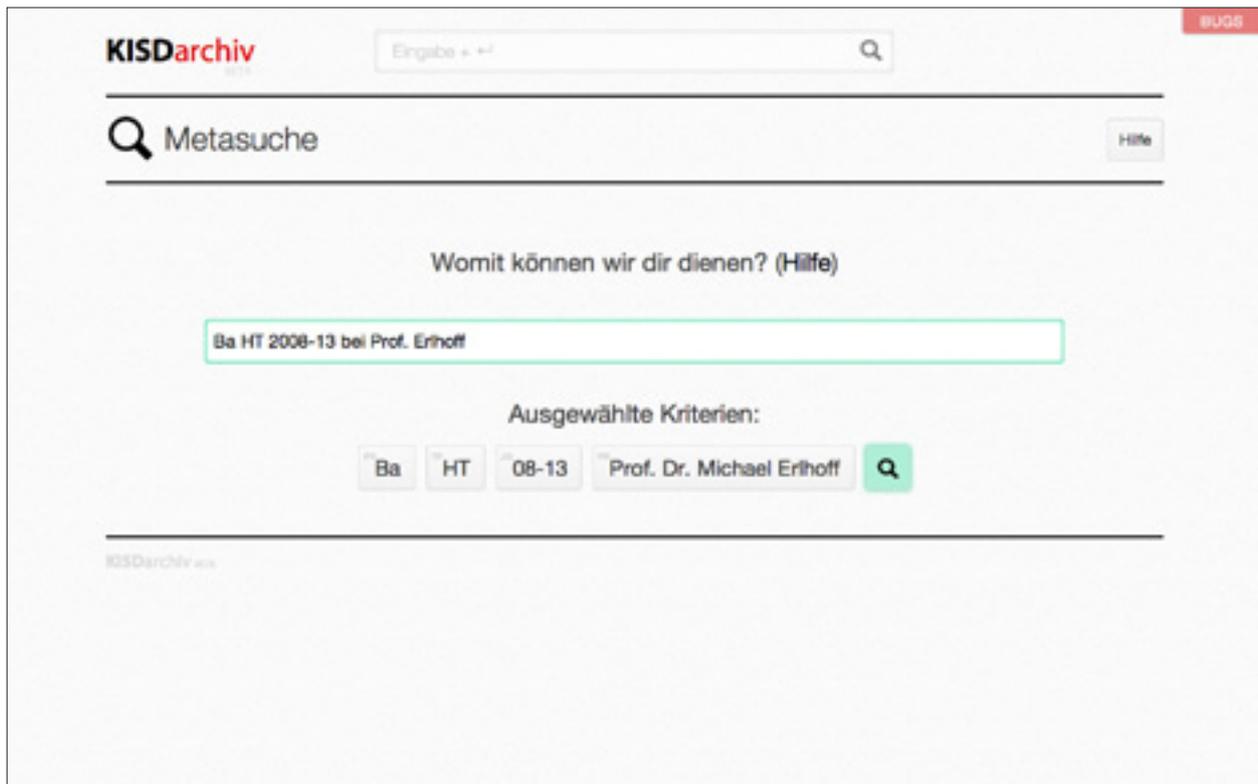


ABB.128 Metasuche

6.1.3 ark_search.php Metasuche

Das Metasuche-Template kapselt die Eingabemaske, sowie die Kriterienanzeige und das verborgene Formular, welches die Kriterien in eine Datenbankabfrage übersetzt. Das Hilfenfenster, sowie der größte Teil des für die Verarbeitung der Abfrage verantwortlichen JavaScript-Codes, ist inline inkludiert.

BEZÜGE header.php
 footer.php
 ark-get-adv-search-results.php (via AJAX)

JAVASCRIPT Inline (Z.182-679) Parsing-Routine der Metasuche.
 ark-1.0.js (Z.365-557) Autocomplete-Initialisierung für Lehrenden-Abgleich innerhalb der Metasuche.

POSITION /wp-content/themes/ark/ark_search.php

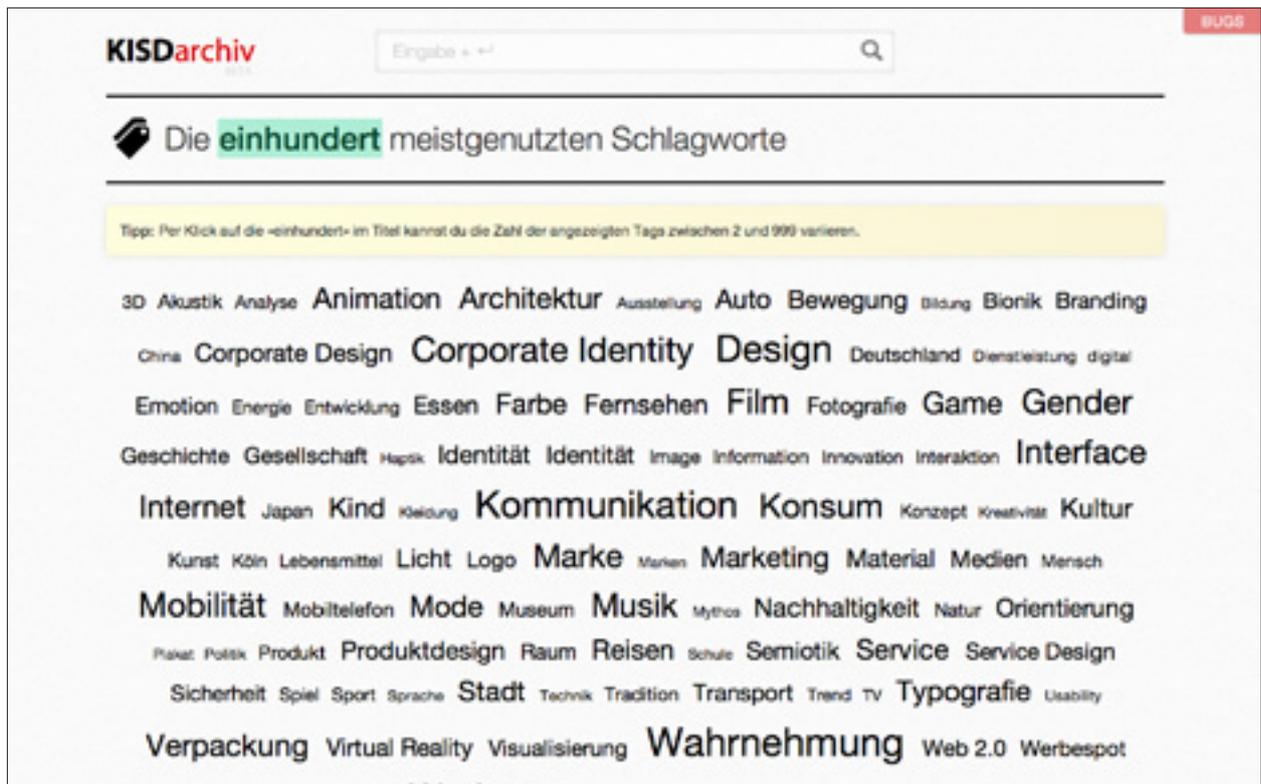
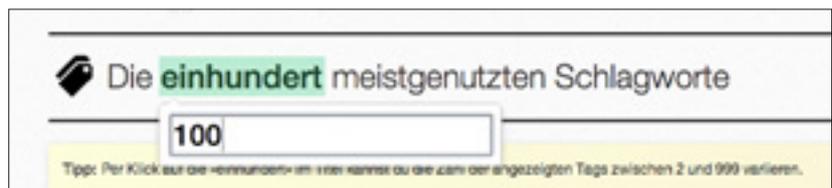


ABB.129 Schlagwortübersicht

6.1.4 ark_tags.php Schlagwortübersicht

Dieses Template bedient sich der Funktion **wp_tag_cloud()** um eine Tag Cloud der am häufigsten verwendeten Schlagwörter zu generieren. Jedes Schlagwort fungiert dabei als Link auf die jeweilige Einzelansicht. Die Anzahl angezeigter Schlagwörter ist via Interaktiver Headline variierbar.

ABB.130 Interaktive Headline im Kontext der Schlagwortübersicht



BEZÜGE header.php
footer.php

JAVASCRIPT ark-1.0.js (Z.175-217) Implementierung der Interaktiven Headline.

POSITION /wp-content/themes/ark/ark_tags.php

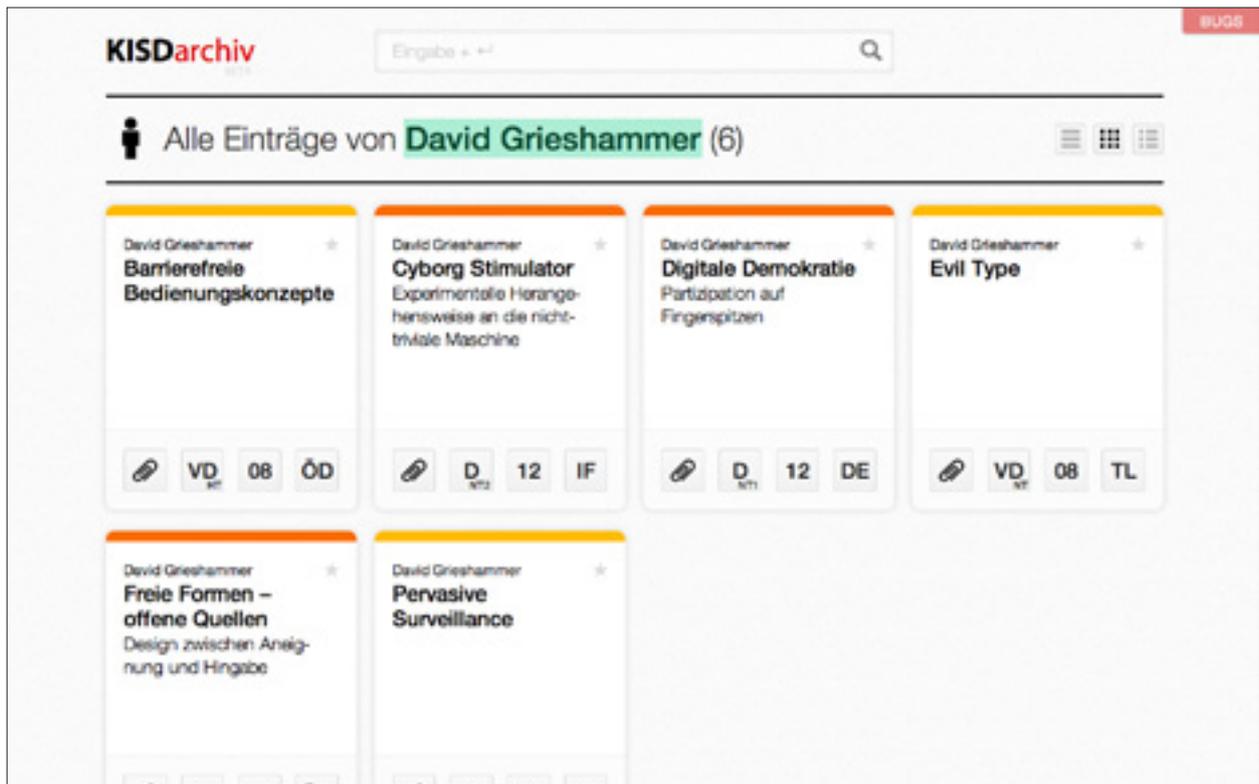


ABB. 131 Autoren-Einzelansicht

6.1.5 single-ark_author.php

Autoren-Einzelansicht

Die Autoren-Einzelansicht kapselt lediglich das Template **ark_author.php**. An dieser Stelle wird überdies die POST-Variablen **ark_aut** ausgelesen, welche innerhalb der Interaktiven Headline im Autoren-Template gesetzt wird.

BEZÜGE header.php
 footer.php
 ark_author.php
 ark_view_[detail | table | tiles].php (je nach aktiver Ansicht)
 nav-below.php

JAVASCRIPT ark-1.0.js (Z.220-261) Autocomplete-Funktionalität innerhalb der interaktiven Headline sowie deren Implementierung.

POSITION /wp-content/themes/ark/single-ark_author.php

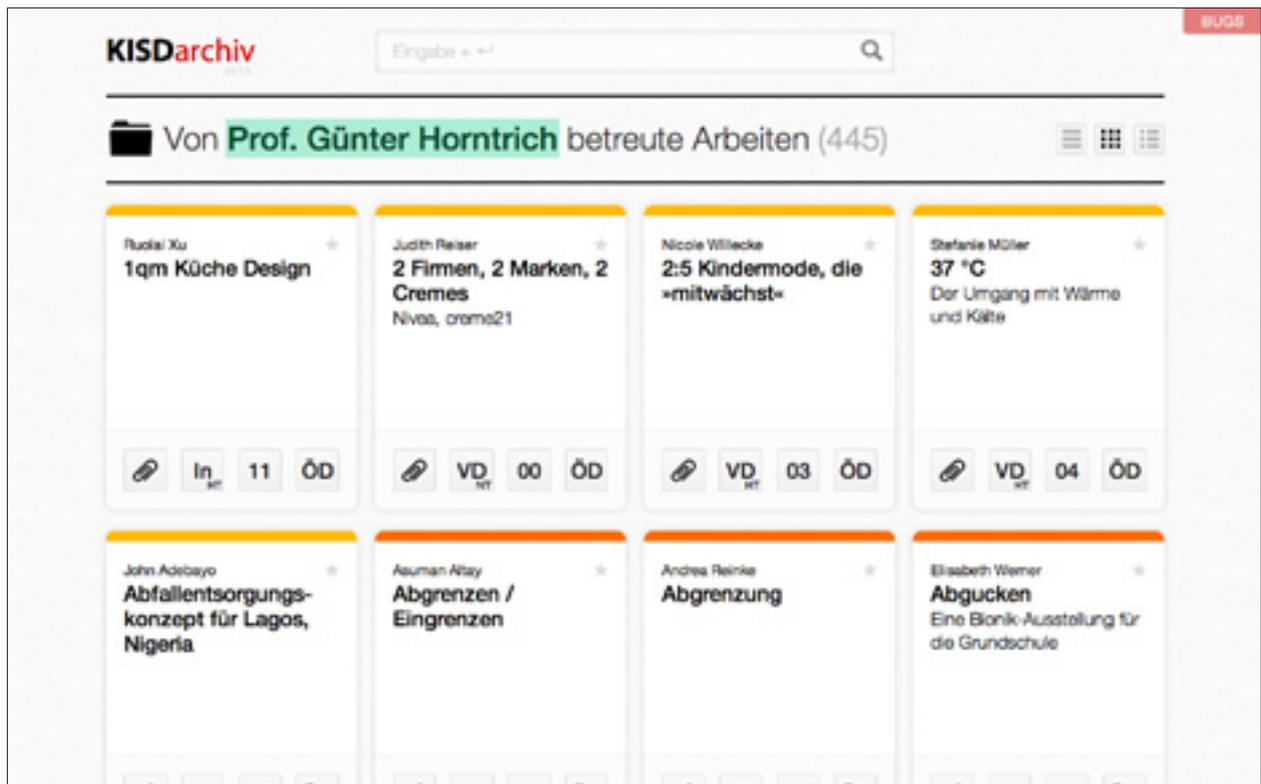


ABB.132 Prüfer-Einzelansicht

6.1.6 single-ark_lecturer.php Prüfer-Einzelansicht

Die Lehrenden-Einzelansicht kapselt lediglich das Template **ark_lecturer.php**. An dieser Stelle wird überdies die POST-Variablen **ark_lect** ausgelesen, welche innerhalb der Interaktiven Headline im Lehrenden-Template gesetzt wird.

BEZÜGE header.php
 footer.php
 ark_lecturer.php
 ark_view_[detail | table | tiles].php (je nach aktiver Ansicht)
 nav-below.php

JAVASCRIPT ark-1.0.js (Z.266-311) Autocomplete-Funktionalität innerhalb der interaktiven Headline sowie deren Implementierung.

POSITION /wp-content/themes/ark/single-ark_lecturer.php

The screenshot shows the KISDarchiv website interface. At the top left is the logo 'KISDarchiv'. To its right is a search bar with the placeholder text 'Eingabe + ↵' and a magnifying glass icon. In the top right corner, there is a red button labeled 'BÜGS'. Below the search bar is a horizontal line, followed by a book icon and the text 'Eintausendunddrei Autorinnen und Autoren'. Below this is another horizontal line, and then a row of 26 buttons representing the alphabet from A to Z. The letter 'A' is highlighted. Underneath, a list of authors is displayed, including Malika Abraham, John Adebayo, Sabrina Adolph, Nadine Adrian, Alexandra Agatonova, Anastasia Agatonova, Merla del Mar Agudelo Torres, Kilian Alchale, Katharina Alt Haddi, Günes Akay, Della Albat, Malik Aleit, Mays Ananda, Angela Allgaler, Christian Alkämper, Pia Alt, Asuman Altay, Kim Altintop, Paula Altmann, Sandra Altmeyer, Vladislav Alukhanov, Tamara Amalia, Menataiah Amed, Fabrizio Amedeo, Agustina Andreolotti, Uwe Anclès, Jan Antang, Sepideh Ardalani, German Artyjev, Caroline Julia Arens, Carolina Arevalo Lopez, Katrin Arndt, Sunil Arora, Juan Jose Arroyo, Christine Auf der Mauer, Jan-Frederik Auf der Mauer, Sven Aufenvenne, Verena Augustin.

The letter 'B' is also highlighted, and a list of authors follows, including Hendrik Bach, Simon Bach, Lars Backhaus, Jennifer Bagehom, Behdad M. Bagheri, Ralf Bähren, Johanna Bak, Andreas Ballhom, Conrad Ballschmiter, Faden Baloglu, Henrik Balzer, Yanik Balzer, Boris Bandyopadhyay, Sharmila Bandyopadhyay, Nils Mark Bamer, Corinna Barfels, Nadine Bartz, Sebastian Baum, Lars Baumann, Daniel Bayer, Tobias Beck, Angelo Becker, Christina Becker, Daniel Becker, Katharina Becker, Miriam Becker, Julietta Beckerd de Heer, Marlen Beckmann, Marcel Defort, Michael Behr, Jan-Philp Behrens, Tania Beifuß, Christiane Beltz, Bianca Bender, Jochen Bender, Ralf Bender, Marc Benz, Nicole Berens, Niklas Berger, Nico Bergmann, Manuel Berninger, Kai Stefan Berthold, Sarah Bessner, Dirk Beushausen, Andreas Beutel, Till Beudling, Nanna Beyer, Sven-Anwar Bibi, Thomas Bieling, Yael Biessack, Martin Bienschek, Alija Bilik, Dörenberger, Nancy Birkhöfer, Nicole Birkenbach, Benno Bittmayer, Franz Blach, Frouke Blazy, Jan Blum, Silke Bochat, Lena Boehm, Uta Bogenrieder, Patricia Bogesitz, Aufschneider, Lukas Böhle, Janine Böhley, Lea Lin Böhrner, Lars Bonasu, Eva Bongers, Annika Börker, Stefanie Bormann, Schidof, Manuel Bom, Martin Herbert Bom, Judith Bosbach, Bastian Boss, Urs Bösawetter, Christian Bramspe, Bettina Braun, Janina Braun, Franz Braunagel, Eva Brauner, Andreas Breilmann, Susanne Breiffeld, Cordula Brenzel, Isabelle Britzy, Alexandra Brücher, Svenja Brüggemann.

ABB.133 Autorenverzeichnis

6.1.7 ark_author_directory.php Autorenverzeichnis

Dieses Template gibt ein alphabetisches Verzeichnis aller Autoren aus, welche im KISDarchiv über veröffentlichte Einträge verfügen.

Das Template enthält Inline JavaScript-Code für die korrekte Funktion und Darstellung der alphabetischen Navigation.

BEZÜGE header.php
footer.php

JAVASCRIPT Inline (Z.39-180) Darstellung / Funktion der alphabetischen Navigation.

POSITION /wp-content/themes/ark/ark_author_directory.php

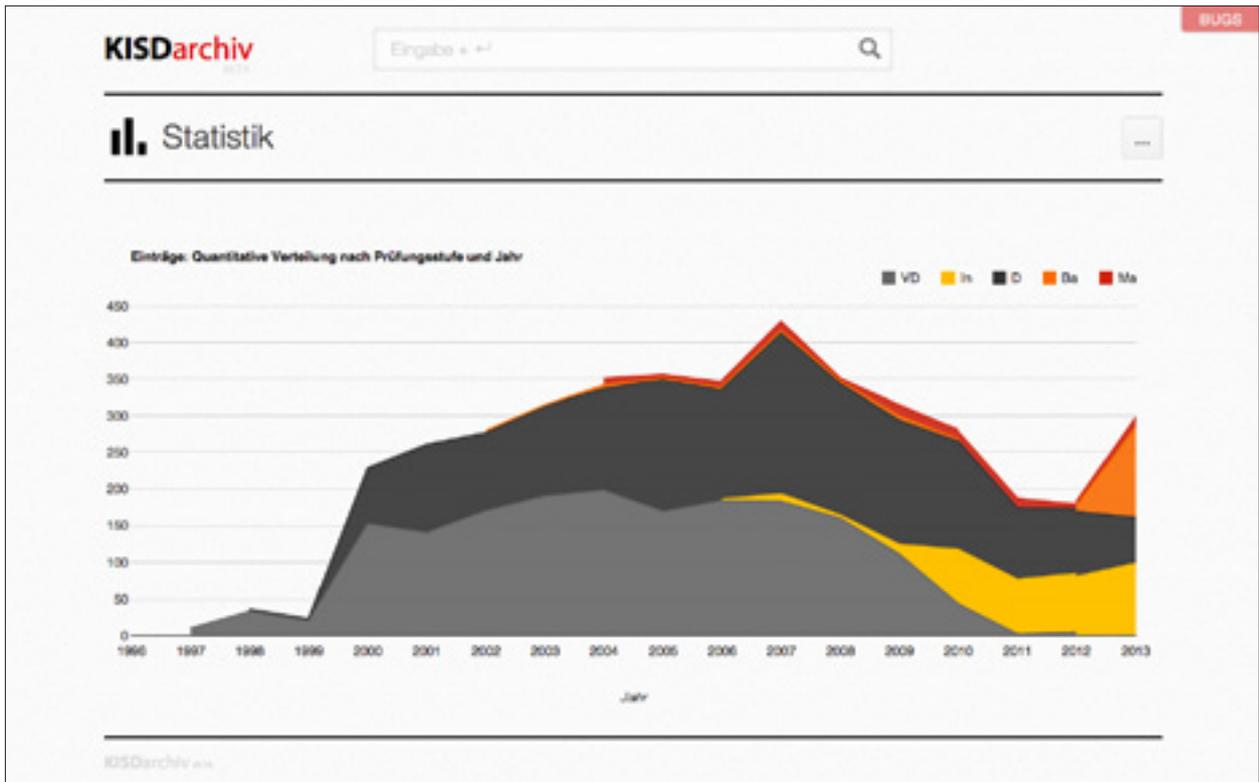


ABB.134 Statistik

6.1.8 ark_statistics.php Statistik

Dieses Template ist für die Darstellung der Statistik-Seite verantwortlich. Das Statistik-Template enthält Inline JavaScript-Code innerhalb dessen eine Datenbankabfrage ein Ergebnis-Array befüllt.

BEZÜGE header.php
footer.php

JAVASCRIPT Inline (Z.160-320) Initialisierung des Flächendiagramms.

Inklusion der Google Chart Tools API via google.com/jsapi/

POSITION /wp-content/themes/ark/ark_statistics.php

Titel	Autor/in	LG	Prüfer/in	PS	TP	JG
Abwesenheit	A. J. Katona, M.-J. Nimmesgern	DTF	Prof. Dr. Michael Erhoff	In	HT	12
Bewegungen im Raum Zu einer Phänomenologie der räumlichen Umwelt im Zeitalter ihrer technischen Transformation	Marc Pfaff	DTF	Prof. Dr. Michael Erhoff	D	HT	10
Architektur, Raum und Poesie	Anna Magdalena Lipp	OO	Prof. Günter Horntrich	D	HT	11
Abgrenzen / Eingrenzen	Asuman Altay	OO	Prof. Günter Horntrich	D	HT	12
Ähnlichkeit Überlegungen zur Gestaltung der erkennbaren Welt	Marc Pfaff	ID	Prof. Andreas Wrede	D	NT1	10
Methoden zwischen Mensch und Ding Reflexionen über eine Operationalisierung des Menschen	Johannes Schott	ID	Prof. Andreas Wrede	D	NT1	11
Haute Cuisine Die Verköstigung von Passagieren an Bord von Verkehrsflugzeugen	Philipp Mies	DFM	Prof. Hatto Grosse	In	HT	11
In Bewegung	J. Esbrenner, L. A. Schreven	OO	Prof. Günter Horntrich	Ba	HT	13

ABB. 135 Lesezeichen-Seite

6.1.9 ark_queue.php Lesezeichen

Die Lesezeichen-Seite bildet alle mit einem Lesezeichen versehenen Einträge in Mehrfachdarstellung ab. Dazu wird der Cookie **ark_queue** ausgelesen. Eine interaktive Headline bietet die Möglichkeit, die Lesezeichen in Link-Form zu exportieren, oder aber die Liste zu leeren. Überdies kann über die Headline die PDF-Ansicht aufgerufen werden.

ABB. 136 Interaktive Headline im Kontext der Lesezeichen-Seite



BEZÜGE header.php
 footer.php
 ark_author.php
 ark_view_[detail | table | tiles].php (je nach aktiver Ansicht)
 nav-below.php

JAVASCRIPT ark-1.0.js (Z.929-941) Implementierung »Lesezeichenliste leeren«-Button.
 (Z.1164-1268) Verhalten bei Aufruf von Shortlink-URLs.

POSITION /wp-content/themes/ark/ark_queue.php

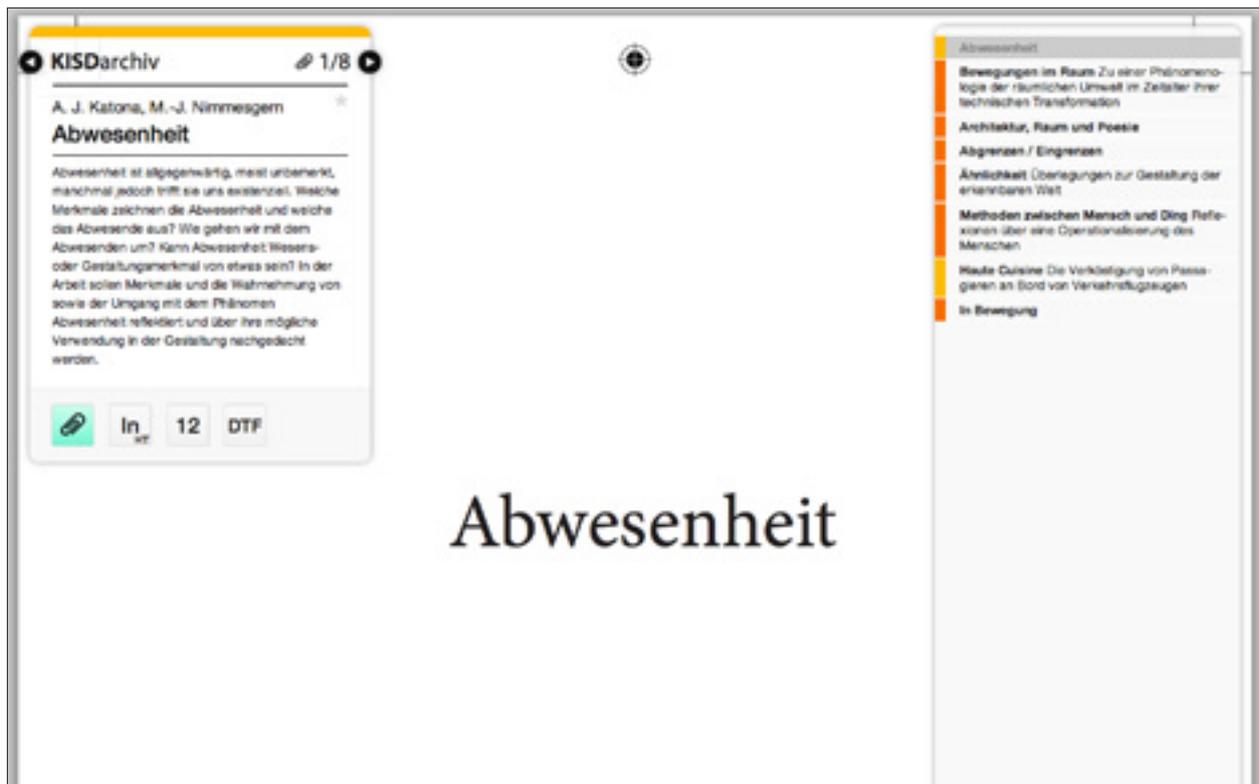


ABB.137 PDF-Ansicht

6.1.10 ark_pdf_view.php PDF-Ansicht

Die PDF-Ansicht, welche von der Lesezeichen-Seite aus referenziert wird, besteht aus zwei über einander gelegten Ebenen:

Die **untere Ebene** bildet ein Browserfenster-umspannender iframe, welcher die aktiv angezeigte PDF-Datei über das browsereigene PDF-Plugin streamt.

Die **obere Ebene** bilden Bedienelemente, welche Metadaten zum aktuelle angezeigten PDF darstellen und die Navigation zwischen mit Lesezeichen versehenen Einträgen Arbeit ermöglichen.

Dieses Template bildet die obere Ebene ab. Es enthält Inline CSS- und JavaScript-Code für die korrekte Darstellung und Funktion der Bedienelemente, sowie Lese- und Schreibvorgänge auf dem Lesezeichen-Cooke **ark_queue**.

Dieses Template verzichtet auf die Seitenkopf und -fuß-Subtemplates.

BEZÜGE pdf.php (inkludiert)

JAVASCRIPT Inline (Z.53-260) Funktion der Bedienelemente.
(Z.925-960) Funktion der Bedienelemente.

POSITION /wp-content/themes/ark/ark_pdf_view.php

ABB.138 Arbeit einreichen

6.1.1 ark_upload.php

Arbeit einreichen

Abhängigkeiten: jquery.iframe-transport.js, jquery.fileupload.js

Das Arbeit einreichen Template zählt zu den komplexesten Templates im KISDarchiv. Es bildet eine Eingabemaske ab, welche es dem Nutzer ermöglicht, Einträge anzulegen. Eine Vielzahl von mit interagierenden Eingabefeldern sind sequenziell auszufüllen, überdies bietet das Template die Möglichkeit, eine PDF-Datei hochzuladen.

Das Template fasst Inline JavaScript-Code zur Live-Evaluation der Eingaben.

BEZÜGE header.php
footer.php

JAVASCRIPT ark-1.0.js (Z.563-927) Konfiguration der konditionalen Prüfungsstufen- und Teilprüfungs-Dropdowns, Initialisierung des tagsInput-Plugins für Schlagwort-, Autoren- und Prüfer-Eingabe. Validierung des Jahr-Eingabefelds.

jquery.custom.js Initialisierung des jQuery Validation Plugins.

Inline (Z.603-801) Kapselung der Inline-Validierung der Eingabefelder. Initialisierung des jQuery File Upload Plugins.

POSITION /wp-content/themes/ark/ark_upload.php



ABB.139 Integrität

6.1.12 ark_integrity.php Integrität

Das Template dient der Darstellung der Integritätsanzeige.

Die Datenbankabfrage, auf Basis derer die Integrität bestimmt wird, findet direkt im Template statt. Das Template enthält Inline CSS-Code zur korrekten Darstellung des Diagramms.

BEZÜGE header.php
footer.php

JAVASCRIPT Inline (Z.164-267) Implementierung der Animation und Interaktion des Integritätsdiagramms (Sortieren und Ausblenden).

POSITION /wp-content/themes/ark/ark_integrity.php

KISDarchiv

Eingabe + ↵

BLOG

🏠

Das analoge Archiv

Im digitalen Archiv – KISDarchiv – findest du Prüfungsarbeiten ab dem Jahr 2000 und einige wenige aus den früheren Jahren.

Das zugehörige analoge Archiv befindet sich seit der Gründung des Fachbereich Design 1991 in Raum 126 und steht wie das digitale allen Angehörigen des Instituts zur Recherche offen.

Bis auf wenige Ausnahmen findest du dort alle Dokumente, die digital im KISDarchiv liegen, in gedruckter Form. Der analoge Bestand geht sogar über den digitalen hinaus, weil gedruckt auch Prüfungsarbeiten der Jahre 1991 bis 2000 vorliegen sowie Dokumentationen und Publikationen seit 1991. Eine Bestandsübersicht des analogen Archivs findest du auf der [Archiv-Seite](#).

Wenn du eine Arbeit hier im KISDarchiv gefunden hast und willst sie in gedruckter Form lesen, findest du sie leicht im analogen Archiv. Du brauchst nur die Angaben Prüfungstufe, Jahr, Autor/in und Teilprüfung, zum Beispiel -VD 01 Neuhaus HT- oder -Ba 14 Corvus NT2-.

Die Archiv-AG ist für die Pflege des digitalen und analogen Archivs verantwortlich. Der gesamte Bestand des Archivs ist sogenannter Präsenzbestand, eine Ausleihe ist also grundsätzlich nicht möglich.



Analoges Archiv in Raum 126

ABB.140 Das analoge Archiv

6.1.13 ark_about.php

Das analoge Archiv

Das Template dient der Darstellung von Informationen zum analogen Archiv. Die Seite besteht aus einem Textblock und einem hochformatigen Foto.

BEZÜGE header.php
 footer.php

POSITION /wp-content/themes/ark/ark_about.php

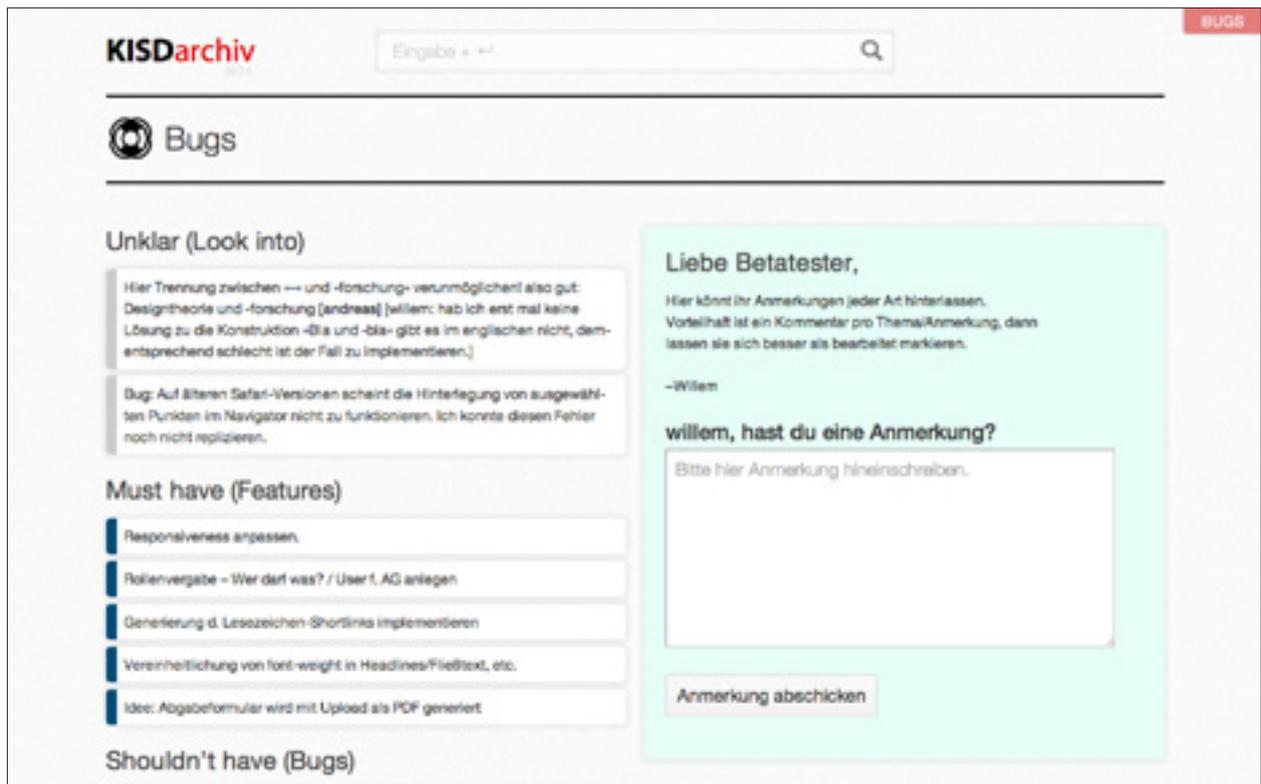


ABB.141 Bugs

6.1.14 ark_bugs.php

Bugs

Das Bugs-Template wurde für die Limited Public Beta Phase geschaffen und dient den Betatestern als Anlaufstelle um Bugs in Textform festzuhalten, welche anschließend von meiner Seite in Kategorien verortet wurden.

Das Template enthält Inline CSS-Code der dem korrekten Styling der Bug-Listendarstellung dient.

Das Template spielt im Regelbetrieb keine Rolle.

BEZÜGE header.php
 footer.php

POSITION /wp-content/themes/ark/ark_bugs.php

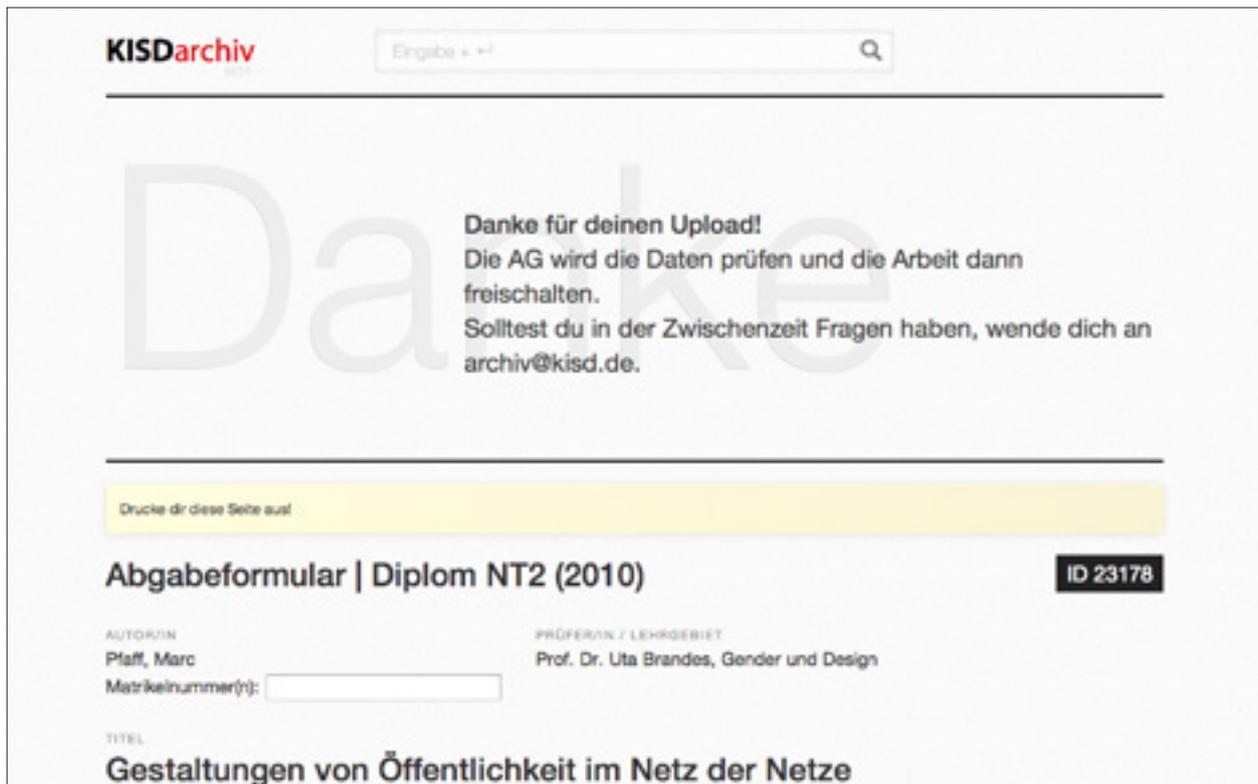


ABB.142 Danke-Seite

6.1.15 ark_upload_complete.php Danke-Seite

Dieses Template stellt die Seite dar, auf welche der Nutzer weitergeleitet wird nachdem er die Arbeit einreichen-Maske erfolgreich ausgefüllt hat.

Hierarchisch ist sie der Upload-Seite untergeordnet:

http://spaces.kisd.de/kisdarchiv/upload/danke/?ark_record_ID=23178

In der URL selbst wird die ID des soeben erstellten Eintrags übergeben, die Seite bezieht daraufhin alle diesem Eintrag zugeordneten Metadaten aus der Datenbank und stellt das Abgabeformular zur Arbeit dar. Die Seite kann so in gedruckter Form als Beleg fungieren, eine Arbeit digital eingereicht zu haben.

Sie erlaubt überdies das nachträgliche Generieren von Abgabeformularen, indem die in der URL übergebene ID modifiziert wird.

Da praktisch alle bis 2013 eingereichten Arbeiten im Dateinamen die Matrikelnummer des Prüflings enthalten, werden Zahlenfolgen in der Ausgabe des Dateinamens im Formular geschwärzt.

Das Template enthält Inline CSS-Code für die Darstellung der Seiteninhalte in der gedruckten Ausgabe.

BEZÜGE header.php
footer.php

POSITION /wp-content/themes/ark/ark_upload_complete.php

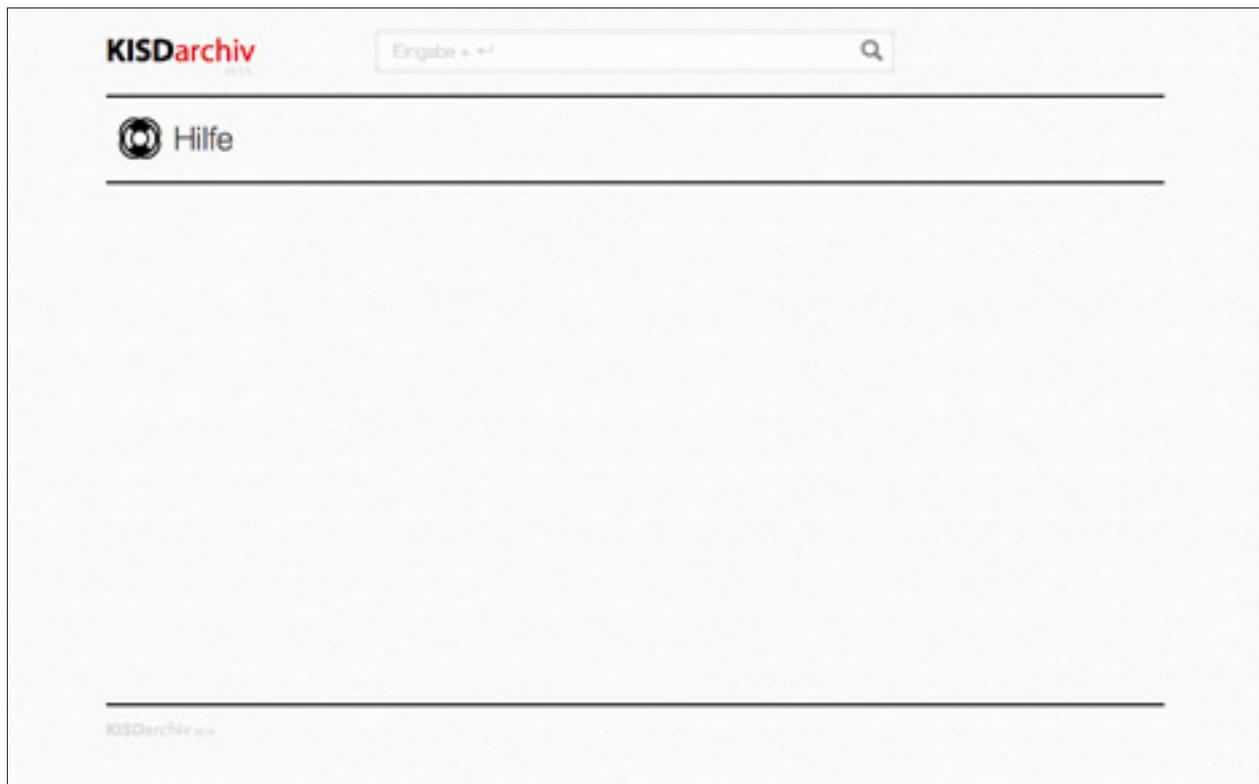


ABB.143 Hilfe

6.1.16 ark_help.php

Hilfe

Das Template für die Darstellung der Hilfe.

Die Hilfe befindet sich zum Zeitpunkt, da diese Arbeit geschrieben wird, noch in der Entwicklung.

BEZÜGE header.php
footer.php

POSITION /wp-content/themes/ark/ark_help.php

6.2 WordPress Templates

	DATEI	SEITE
SUCHERGEBNISSE	search.php	185
EINZELANSICHT SCHLAGWORT	tag.php	186
FEHLERSEITE	404.php	187
EINZELANSICHT	single.php	188

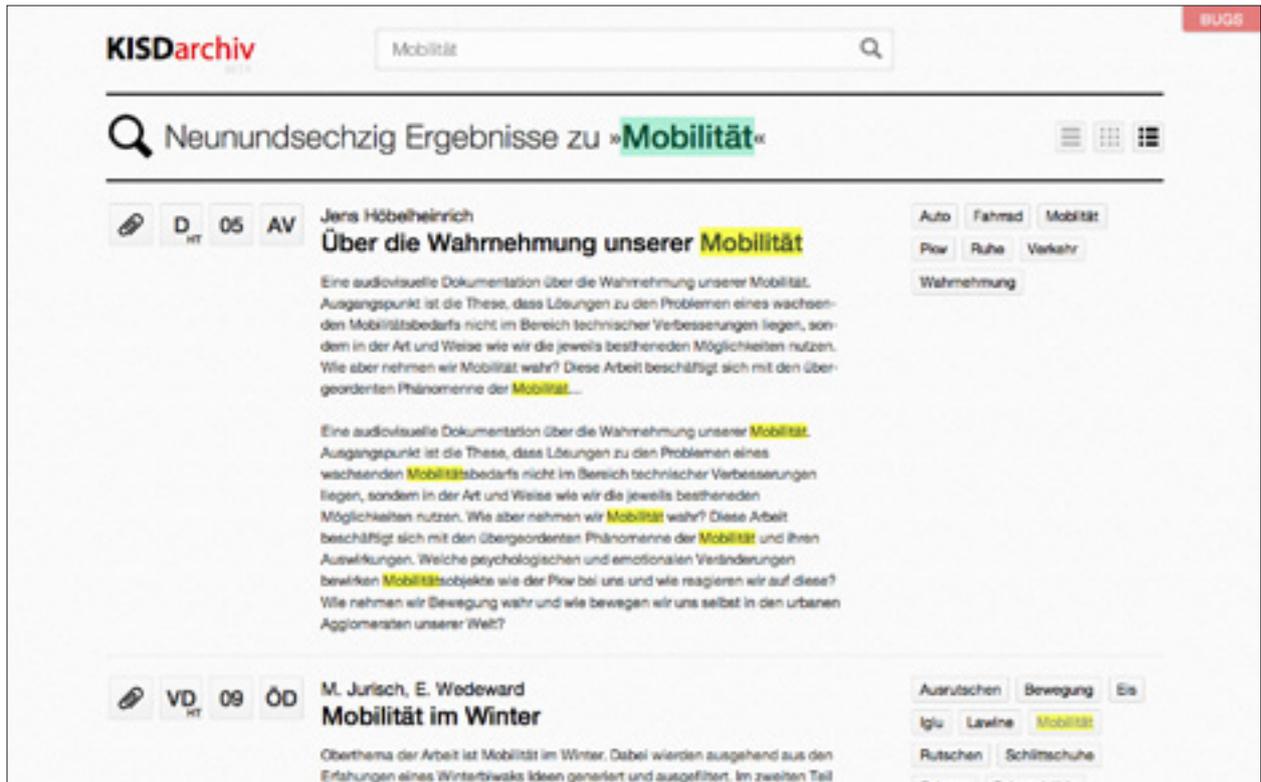
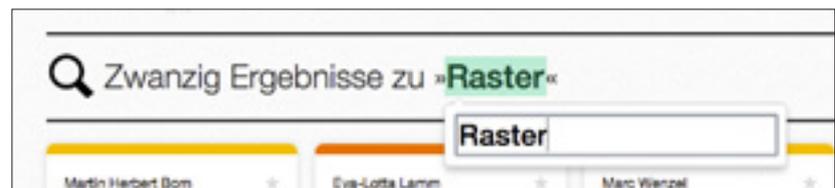


ABB.144 Suchergebnisse

6.2.1 search.php Suchergebnisse

Das Template zur Anzeige der Suchergebnisse liest den Cookie `ark_view` aus und zeigt die Suchergebnisse entsprechend der darin deklarierten Mehrfachansicht an. Diese Template verfügt über eine interaktive Headline, welche es dem Nutzer erlaubt, den Suchbegriff direkt auf der Ergebnisseite zu variieren.

ABB.145 Interaktive Headline im Kontext einer Suchergebnis-Seite



BEZÜGE header.php
 footer.php
 ark_view_[detail | table | tiles],.php (je nach aktiver Ansicht)
 nav-below.php

JAVASCRIPT ark-1.0.js (Z.113-121) Austausch von textlicher durch numerische Suchergebnis-Anzahl via Klick auf diesselbe.
 (Z.126-171) Hervorhebung von Suchbegriff in Ergebnissen, Implementierung der Interaktiven Headline.

POSITION /wp-content/themes/ark/search.php

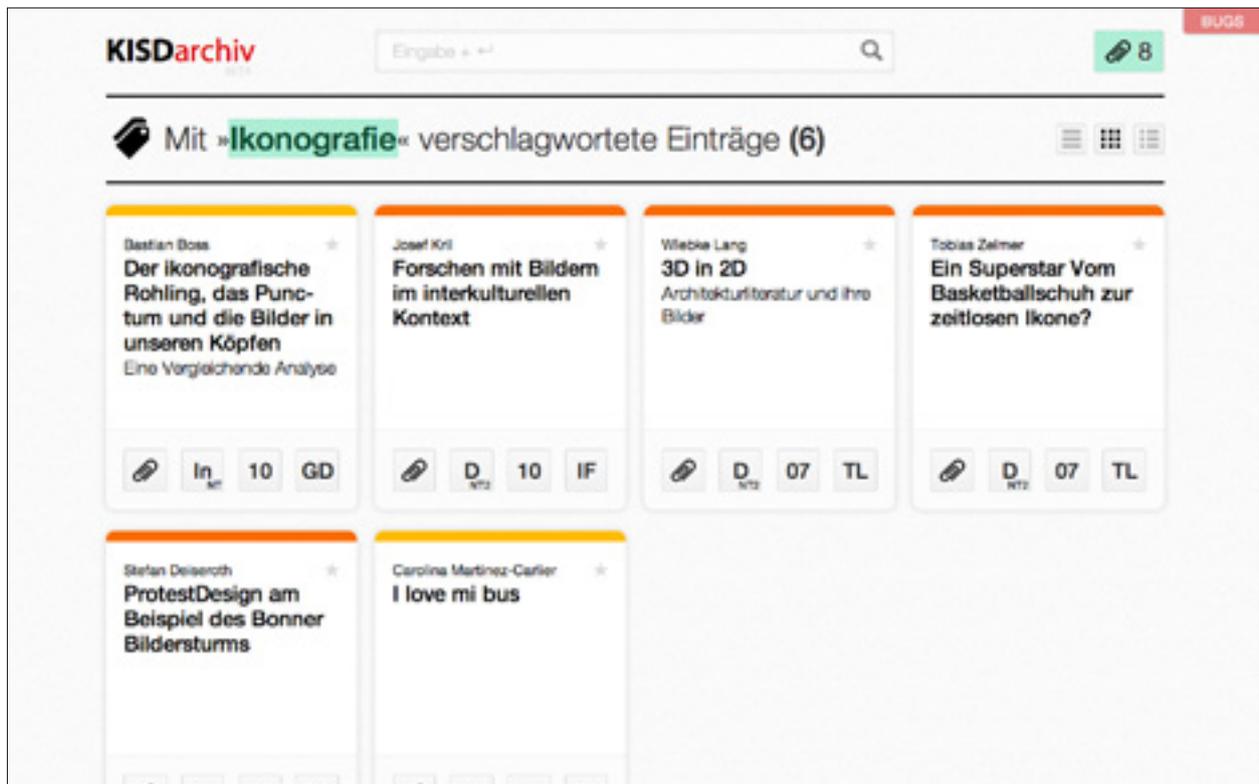


ABB.146 Schlagwort-Einzelansicht

6.2.2 tag.php

Schlagwort-Einzelansicht

Die Schlagwort-Einzelansicht zeigt alle einem Schlagwort zugeordneten Einträge des Typs `ark_record`. Das angezeigte Schlagwort lässt sich innerhalb dieser Ansicht via Interaktiver Headline verändern. Die Einträge werden entsprechend der in der Variable `$ark_view` gesetzten Ansicht dargestellt.

BEZÜGE header.php
 footer.php
 ark_view_[detail | table | tiles].php (je nach aktiver Ansicht)

JAVASCRIPT ark-1.0.js (Z.314-361) Autocomplete-Funktionalität der interaktiven Headline sowie deren Implementierung.

POSITION /wp-content/themes/ark/tag.php

TODO ark_td_07 (Siehe S. 229)

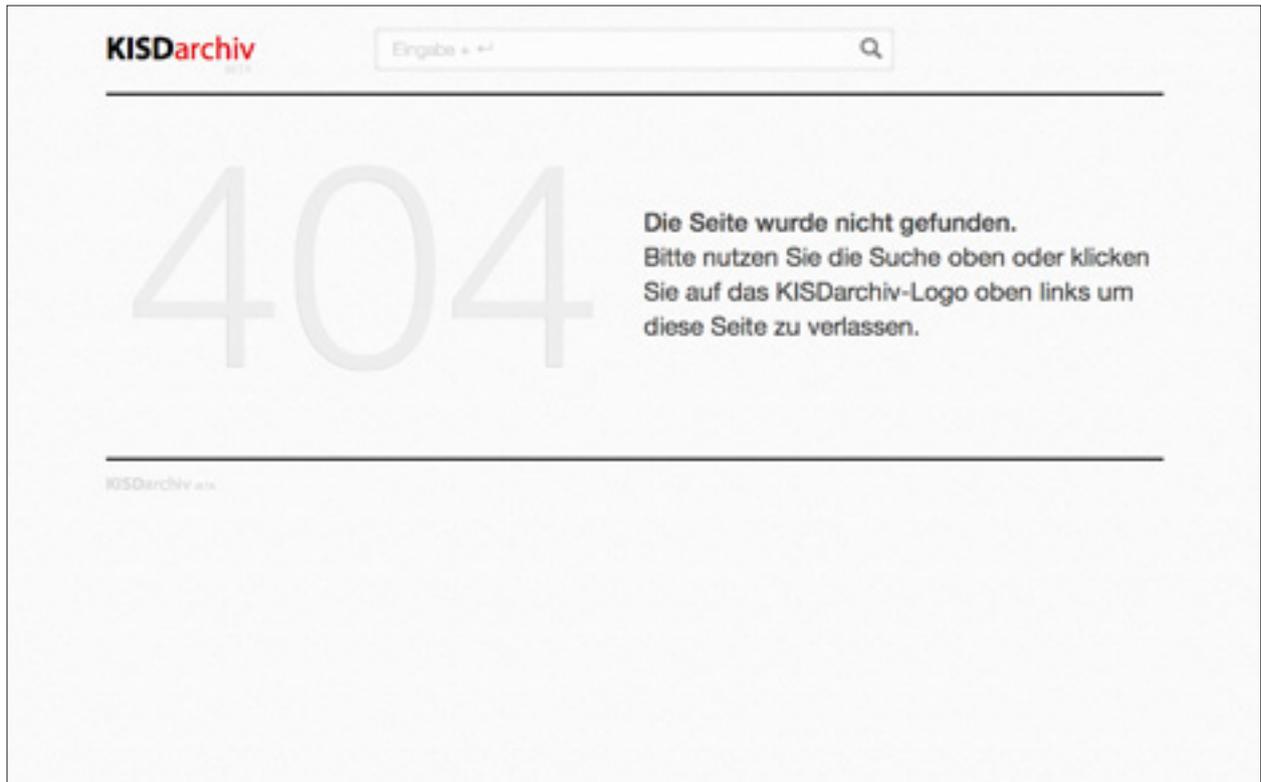


ABB.147 Fehlerseite

6.2.3 404.php Fehlerseite

Das 404-Template dient der Darstellung einer Fehlerseite, sobald der Nutzer sich auf eine nicht-existente URL begeben hat.

Das Template überprüft die Server-Variable HTTP_REFERER um festzustellen, ob der Nutzer sich durch eigenes Verschulden – sprich: Die manuelle Eingabe einer ungültigen Adresse – auf die Fehlerseite begeben hat. Ist dem nicht der Fall und der Nutzer somit einem ungültigen Link gefolgt, wird eine E-Mail an die für den Admin hinterlegte E-Mail-Adresse abgeschickt, in welcher der Vorfall detailliert geschildert wird.

BEZÜGE header.php
footer.php

JAVASCRIPT Inline (Z.44-63) Animation des Fehlerteufels je nach Cursorposition.

POSITION /wp-content/themes/ark/404.php

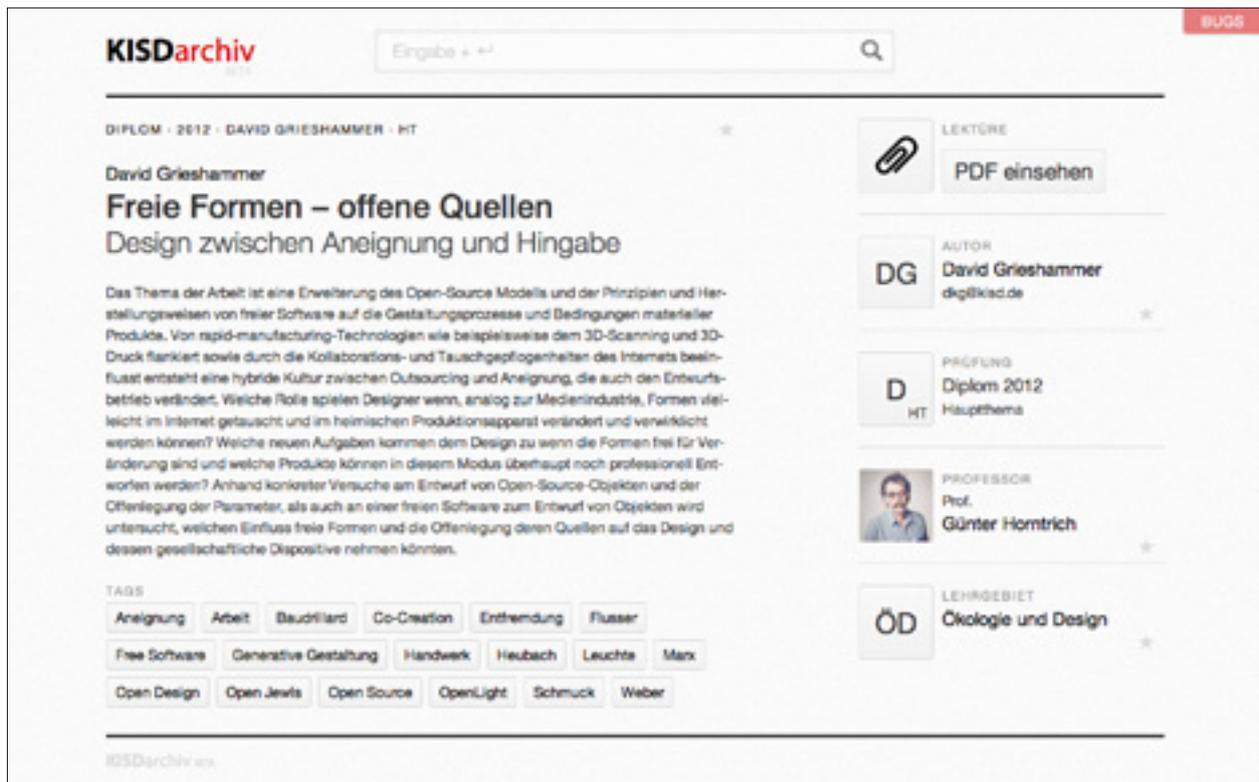


ABB. 148 Eintrags-Einzelansicht

6.2.4 single.php Eintrags-Einzelansicht

Das WordPress Standard-Template zur Darstellung einzelner Posts.

BEZÜGE header.php
footer.php
entry.php

POSITION /wp-content/themes/ark/single.php

6.3 KISDarchiv-Subtemplates

	DATEI	SEITE
TEMPLATE AUTOR	ark_author.php	190
TEMPLATE LEHRENDER	ark_lecturer.php	191
MEHRFACHANSICHT: DETAIL	ark_view_detail.php	192
MEHRFACHANSICHT: LISTE	ark_view_table.php	193
MEHRFACHANSICHT: KARTEN	ark_view_tiles.php	194
PDF	pdf.php	195
KEIN PDF	no_pdf.php	196

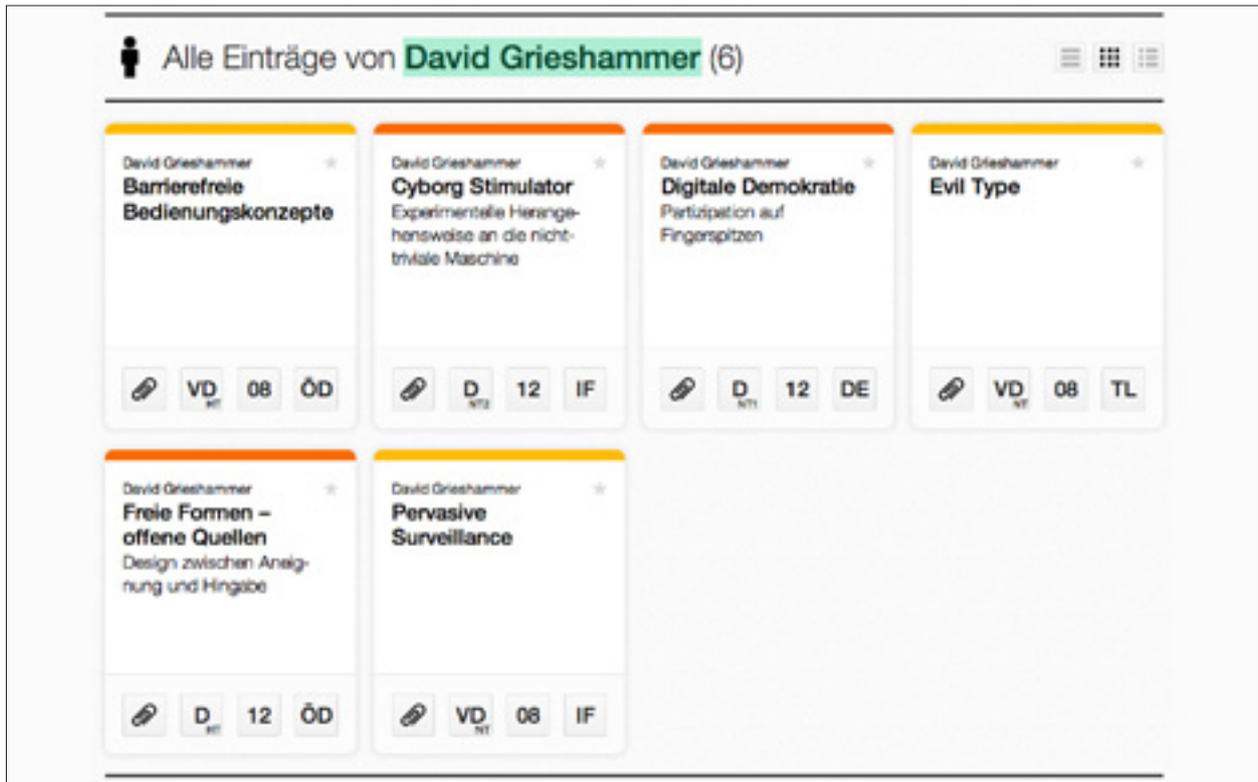


ABB. 149 Subtemplate Autor

6.3.1 ark_author.php Autoren-Subtemplate

Das Autorentemplate enthält eine maßgeschneiderte Datenbankabfrage welche die alle einem speziellen Autor zugeordneten Einträge ausliest und je nach ausgewählter Mehrfachansicht ausgibt. Eine Interaktive Headline erlaubt das Wechseln zwischen Autoren innerhalb dieser Seite.

INKLUDIERT IN single-ark_author.php

POSITION /wp-content/themes/ark/ark_author.php

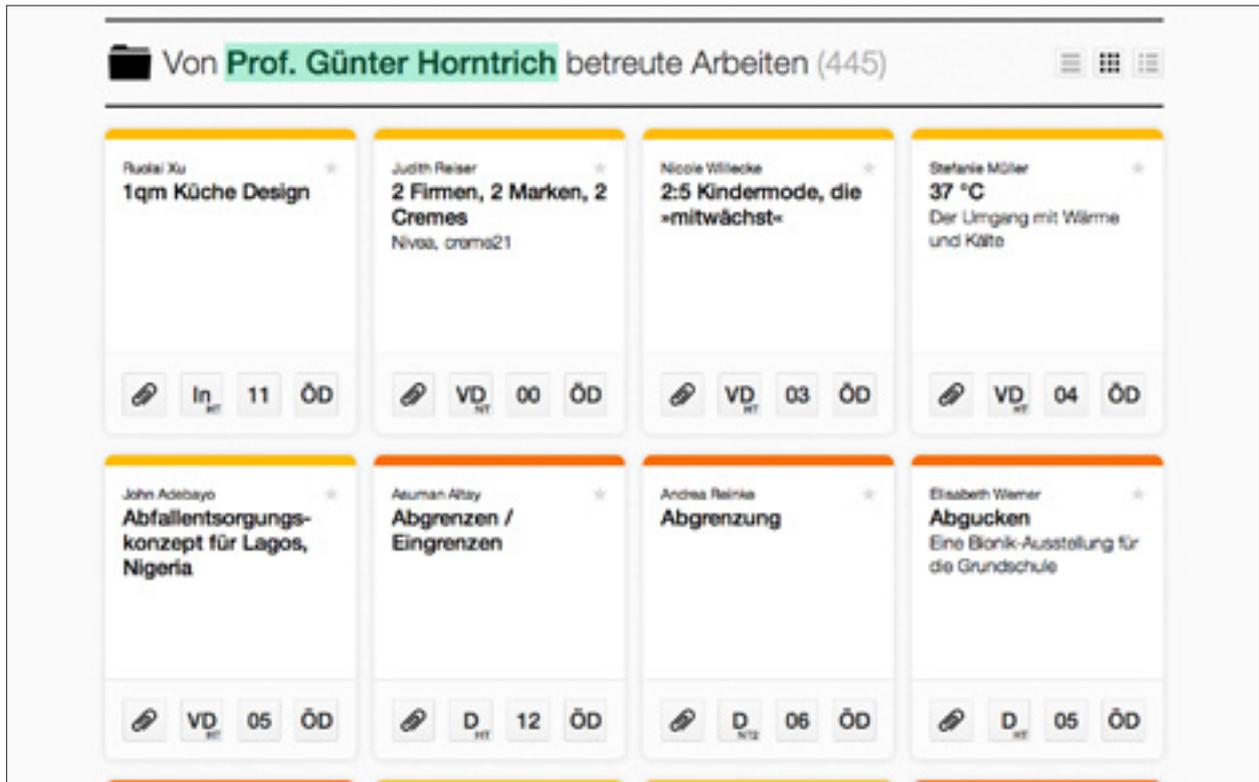


ABB. 150 Subtemplate Prüfer

6.3.2 ark_lecturer.php Prüfer-Subtemplate

Das Lehrentemplate enthält eine maßgeschneiderte Datenbankabfrage welche die alle einem Lehrenden zugeordneten Einträge ausliest und je nach ausgewählter Mehrfachansicht ausgibt. Eine Interaktive Headline erlaubt das Wechseln zwischen Lehrenden innerhalb dieser Seite.

INKLUDIERT IN single-ark_lecturer.php

POSITION /wp-content/themes/ark/ark_lecturer.php

The screenshot displays two article entries in a multi-view detail view. The top entry is by Jens Höbelheirich, titled "Über die Wahrnehmung unserer Mobilität". It features a tag cloud with "Auto", "Fahrad", "Mobilität", "Pkw", "Ruhe", and "Verkehr". The bottom entry is by M. Jurisch, E. Wedeward, titled "Mobilität im Winter". It features a tag cloud with "Ausrutschen", "Bewegung", "Eis", "Iglu", "Lawine", "Mobilität", "Rutschen", "Schlittschuhe", "Schnee", "Schneehöhe", "Schuhschle", "Ski", "Snowboard", "Winter", and "Winterreifen".

ABB. 151 Mehrfachansicht Detail

6.3.3 ark_view_detail.php Mehrfachansicht Detail

Das Template für die Anzeige der Detaildarstellung innerhalb von Eintrags-Mehrfachansichten. Das Template liest den Posttyp der darzustellenden Einträge aus und zeigt daraufhin die adäquaten Metadaten an.

INKLUDIERT IN /wp-content/themes/ark/ark_view_detail.php

POSITION /wp-content/themes/ark/ark_author.php

	Titel	Autor/in	LG	Prüfer/in	PS	TP	JG
	Farbe	Kolja Schäfer	PT	Prof. Walter Dohmen	VD	NT	00
	Die Farbe Haut	Jasmin Jochum	GD	Prof. Dr. Uta Brandes	D	NT2	07
	Rosa Betrachtung einer Farbe und ihrer Nuancen	Barbara Wirtz	DTF	Prof. Dr. Michael Erhoff	D	NT1	06
	Farbdesign Wirkung von Farbe im Patientenzimmer	Andrea Döhling	DTF	Prof. Dr. Michael Erhoff	VD	NT	05
	Pink ist keine Farbe, sondern eine Lebenseinstellung	Alexandra Wolf	TL	Prof. Michael Gais	VD	HT	08
	Farbe im kulturellen Kontext	Philipp Thesen	DE	Prof. Dr. Brigitte Wolf	Ma	NT1	07
	Wechselwirkung von Farbe	Karl Wissing	"		VD	NT	99
	Farbe im interkulturellen Kontext	Philipp Thesen	ID	Prof. Helner Jacob	Ma	NT1	07
	Olivgreen Die Farbe Oliv-Grün	Denise Laser	ID	Prof. Walter Dohmen	VD	NT	00
	Verführung Farbe Das Auge isst mit!	Anke Kossak	CO	Prof. Günter Hornbich	D	NT1	04
	Die Farben der Farben Eine facettenreiche Welt	Stefanie Weidner	PT	Prof. Wolfgang Laubersheimer	VD	NT	08
	Rosa vs. Pink	Constanze Kletsch	GD	Prof. Dr. Uta Brandes	D	NT1	04

ABB.152 Mehrfachansicht Tabelle

6.3.4 ark_view_table.php Mehrfachansicht Tabelle

Das Template für die Anzeige der Listendarstellung innerhalb von Eintrags-Mehrfachansichten.

Das Template liest für jeden Eintrag den Posttyp aus und erstellt daraufhin die Tabellenzeilen welche die posttypspezifischen Metadaten enthalten.

INKLUDIERT IN single-ark_lecturer.php

POSITION /wp-content/themes/ark/ark_view_table.php

TODD ark_td_02 (Siehe S. 229)

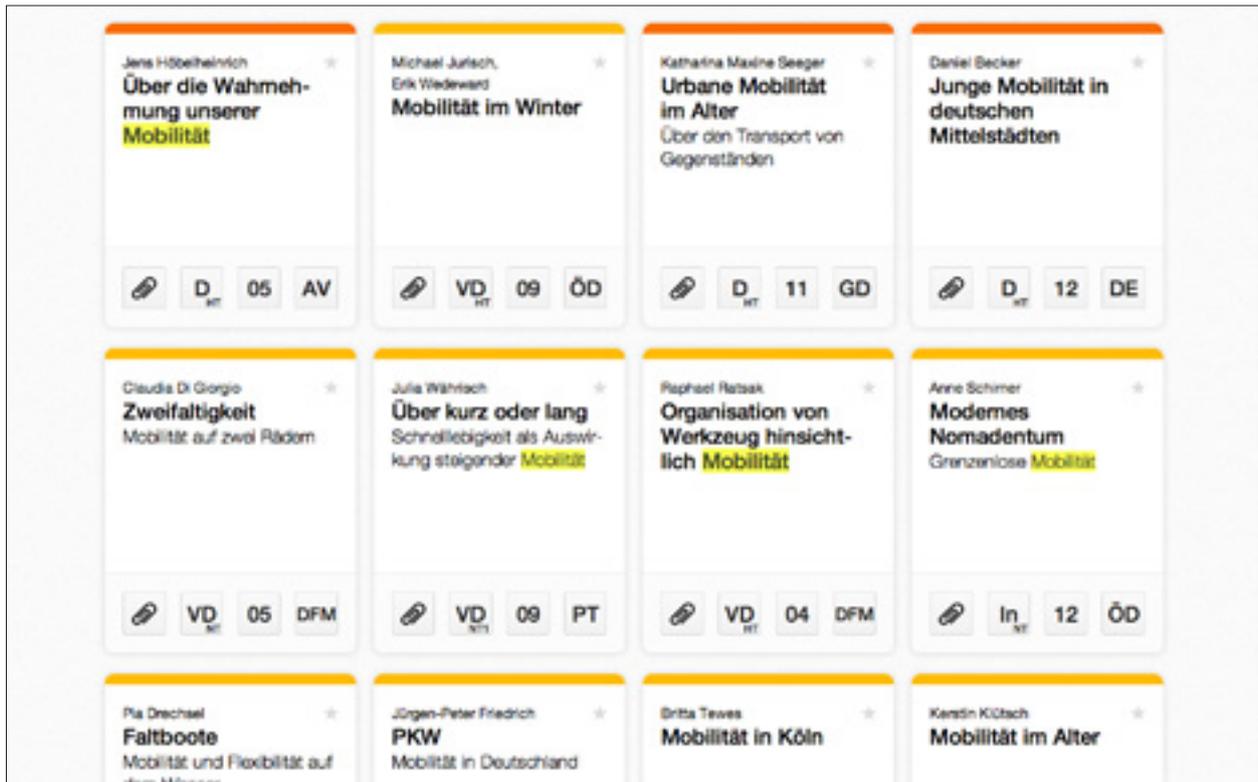


ABB. 153 Mehrfachansicht Karten

6.3.5 ark_view_tiles.php Mehrfachansicht Karten

Das Template für die Anzeige der Kartendarstellung innerhalb von Eintrags-Mehrfachansichten.

Dieses Template enthält lediglich den Aufruf zur Funktion `ark_get_tile()`, welche sich für die Generierung der Karten verantwortlich zeichnet.

INKLUDIERT IN `single-ark_author.php`

POSITION `/wp-content/themes/ark/ark_view_tiles.php`



ABB. 154 Subtemplate PDF

6.3.6 pdf.php PDF-Subtemplate

Das PDF-Subtemplate kommt innerhalb der PDF-Ansicht zum Einsatz. Es erhält die aktuelle Eintrags-ID in Form der HTTP POST-Variable `id`. Daraufhin wird die übergebene ID als einem Eintrag vom Typ `ark_record` zugehörig überprüft, die eintragseigene Variable `ark_record_pdf_url` ausgelesen und zur Ausgabe gebracht.

Jeder Lesevorgang wird in einer Log-Datei protokolliert, welche im Root-Verzeichnis abgelegt ist (`/var/www/spaces/htdocs/ark.log`).

- INKLUDIERT IN `ark_pdf_view.php`
- POSITION `/pdf.php`
- TODD `ark_td_03` (Siehe S. 229)



ABB. 155 PDF-Datei fehlt

6.3.7 no_pdf.php PDF-fehlt-Subtemplate

Dieses Template wird innerhalb des Templates **pdf.php** eingebunden, sofern dieses feststellt, dass zu einem anzuzeigenden Eintrag keine zugehörige PDF-Datei vorhanden ist.

Das Template fragt die **ark_record_visible**-Variable jeweiligen Eintrags ab, um festzustellen, ob das Nicht-Vorhandensein der PDF-Datei zu tolerieren ist (**ark_record_visible** hat in diesem Fall den Wert `›false‹/0`). Besteht keine Einwilligung, die PDF-Datei zu zeigen – und damit keine Notwendigkeit ihrer Existenz auf dem Server – wird ein entsprechender Hinweis ausgegeben, der anzeigt, dass die entsprechende Zustimmung des Autors nicht vorliegt. Besteht jedoch der Fall, dass die Zustimmung gegeben wurde (**ark_record_visible** hält dann den Wert `›true‹/1`), wird die ID des Eintrags dem Optionsfeld **ark_record_pdf_missing_list** hinzugefügt, was zur Folge hat, dass der Eintrag in einem Dashboard-Widget im Backend angezeigt wird. Damit wird die Archiv-AG auf den Umstand hingewiesen, dass die PDF-Datei nicht vorhanden ist und nachgepflegt werden sollte. Gleichzeitig zeigt das Template dem Nutzer einen Hinweis an, dass die AG über den Umstand der fehlenden Datei informiert wurde.

Das Template enthält Inline CSS-Code für die korrekte Seitendarstellung.

INKLUDIERT IN ark_pdf_view.php (bedarfsweise inkludiert)

POSITION /wp-content/themes/ark/no_pdf.php

6.4 WordPress-Subtemplates

	DATEI	SEITE
SEITENKOPF	header.php	198
SEITENFUSS	footer.php	198
SUCHFELD	searchform.php	198
EINTRAG	entry.php	199
EINTRAGS-INHALT	entry-content.php	200
EINTRAGS-ZUSAMMENFASSUNG	entry-summary.php	200
EINTRAGS-META	entry-meta.php	200
EINTRAGS-FUSS, EINZELANSICHT	entry-footer-single.php	200
PAGINIERUNG UNTEN	nav-below.php	201



ABB.156 Seitenkopf

6.4.1 header.php

Seitenkopf

Das Header-Template enthält den Kopfbereich des KISDarchiv-Themes. Alle notwendigen JavaScript- und CSS-Dateien sind hier inkludiert. Handelt es sich bei der aktuellen Seite um eine Eintrags-Einzelansicht, wird an dieser Stelle überdies ein prüfungsstufenspezifisches favicon eingebunden.

INKLUDIERT IN alle Seitentemplates

POSITION /wp-content/themes/ark/header.php



ABB.157 Seitenfuß

6.4.2 footer.php

Seitenfuß

Der Footer bildet den Seitenabschluss. Er enthält die Fußzeile.

INKLUDIERT IN alle Seitentemplates

JAVASCRIPT Inline (Z.49-81) Animation des Seitenfußes bei Mouseover.

POSITION /wp-content/themes/ark/footer.php



ABB.158 Suchfeld

6.4.3 searchform.php

Suchfeld

Dieses Subtemplate gibt ein Suchformular (ID: **searchform**) in Form eines Texteingabefelds aus, welches im Seitenkopf-Template eingebunden wird. Das Template enthält etwas Inline-JavaScript-Code welcher das Feld auf Eingaben überprüft und bedarfsweise einen Platzhaltertext einfügt.

INKLUDIERT IN header.php

JAVASCRIPT ark-1.0.js (Z.37-78) Einblenden der kontextuellen Hilfe im Falle des Absendens eines leeren Suchfelds.
(Z.80-88) Absenden des Felds bei Klick auf Lupensymbol.

POSITION /wp-content/themes/ark/footer.php

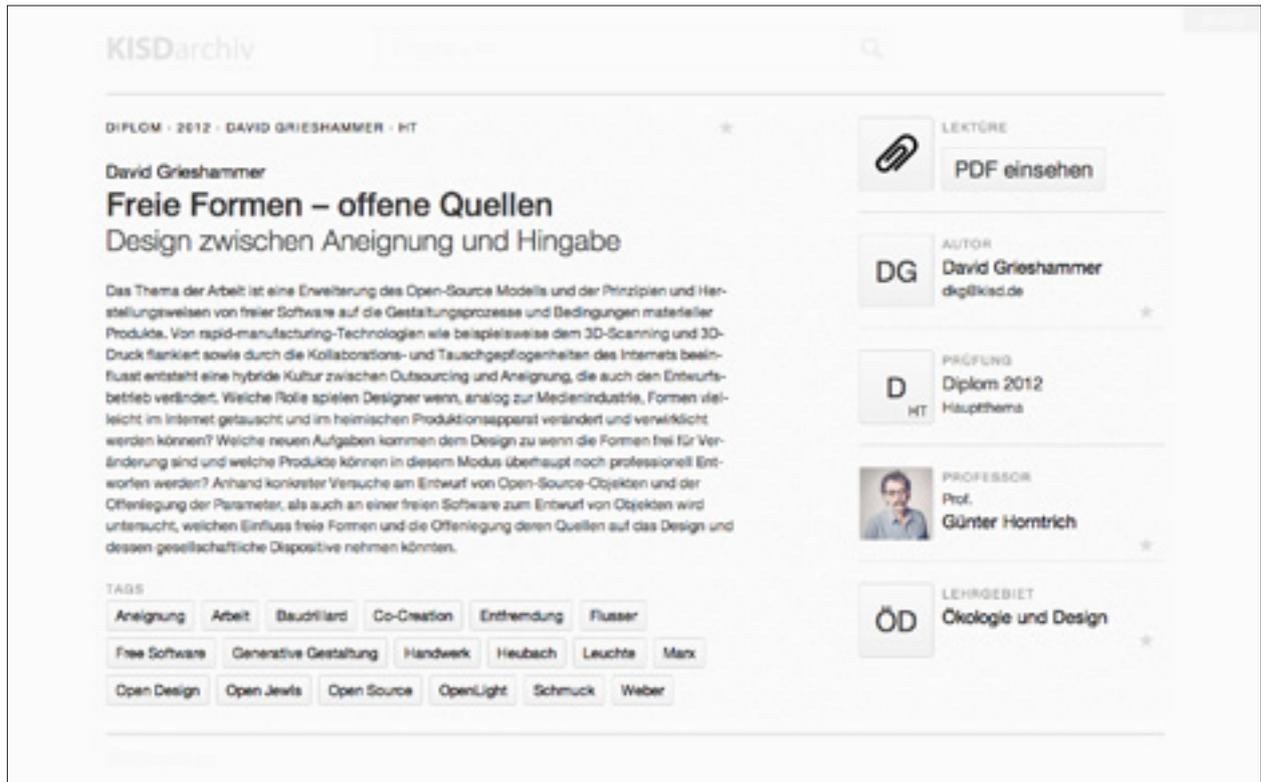


ABB.159 Eintrags-Einzelansicht

6.4.4 entry.php Eintrag

Das Eintragstemplate kommt bei der Anzeige einzelner Einträge sowie der Darstellung von Eintrags-Metadaten innerhalb der Suchergebnisse zum Tragen. Das Template referenziert eintragsbezogene Subtemplates, welche die Generierung von Abstract, Seitenleiste und Schlagwort-Liste übernehmen.

BEZÜGE entry-content.php
entry-summary.php
entry-meta.php
entry-footer.php
entry-footer-single.php

INKLUDIERT IN single-ark_author.php

POSITION /wp-content/themes/ark/ark_author.php

6.4.5 entry-content.php

Eintragsinhalt

Der Eintragsinhalt stellt das Abstract in der Eintrags-Einzelansicht dar. Liegt kein Abstract vor, so wird ein entsprechender Hinweis angezeigt, welcher den Autor der Arbeit dazu aufruft, dieses einzusenden.

INKLUDIERT IN entry.php

POSITION /wp-content/themes/ark/entry-content.php

6.4.6 entry-summary.php

Eintrags-Zusammenfassung

Dieses Template wird innerhalb von Suchergebnissen variabel eingebunden und sorgt für die Ausgabe eines verkürzten Abstracts.

INKLUDIERT IN entry.php

POSITION /wp-content/themes/ark/entry-content.php

6.4.7 entry-meta.php

Eintrags-Meta

Dieses Subtemplate wird innerhalb der Eintrags-Einzelansicht eingebunden und erzeugt die Darstellung der Metadaten-Seitenleiste.

INKLUDIERT IN entry.php

POSITION /wp-content/themes/ark/entry-meta.php

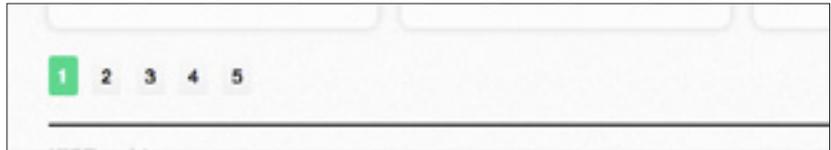
6.4.8 entry-footer-single.php

Eintrags-Fuß, Einzelansicht

Der Eintragsfuß in der Einzelansicht dient der Darstellung der einem Eintrag zugeordneten Schlagwörter.

INKLUDIERT IN entry.php

POSITION /wp-content/themes/ark/entry-footer-single.php

ABB.160 Paginierung unten

6.4.9 nav-below.php

Paginierung unten / Mehrfachansicht

Dieses Subtemplate zeichnet sich für die Darstellung der Paginierung zwischen Mehrfachansicht und Seitenfuß verantwortlich.

INKLUDIERT IN Mehrfachansichts-Seiten

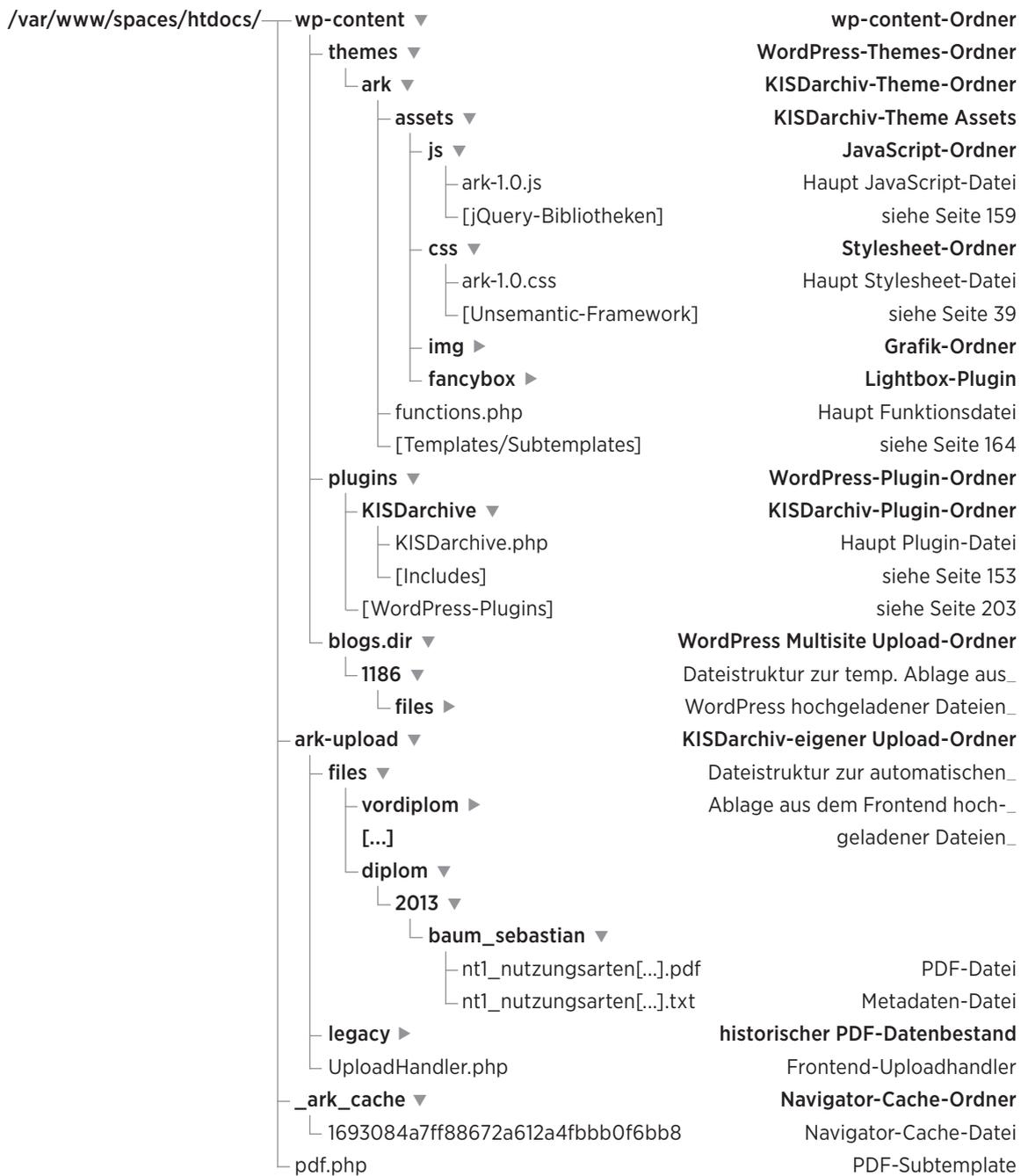
POSITION /wp-content/themes/ark/nav-above.php

7 WordPress

7.0 Einleitung

WordPress ist die wohl am besten dokumentierte CMS-ähnliche Plattform. Der Fokus an dieser Stelle soll auf jenen Strukturen liegen, die das KISDarchiv den WordPress-eigenen Gefügen anbringt. Es werden die KISDarchiv-eigenen Ordner- und Dateistrukturen, seine Datenbank, das KISDarchiv-Theme und -Plugin sowie die übrigen eingesetzten WordPress-Plugins Erwähnung finden.

7.1 Dateistruktur



7.2 Datenbank

7.2.1 Struktur

TABELLE	BESCHREIBUNG
WP_1186_COMMENTMETA	-
WP_1186_COMMENTS	Kommentare (Bugs-Seite)
WP_1186_LINKS	-
WP_1186_OPTIONS	Optionstabelle
WP_1186_POSTS	Posttabelle (Seiten, Einträge aller Posttypen, Anhänge,...)
WP_1186_POSTMETA	Metadaten von Posts und Plugins
WP_1186_RELEVANSSI	Suche / Begriffsindex
WP_1186_RELEVANSSI_CACHE	-
WP_1186_RELEVANSSI_EXCERPT	-
WP_1186_RELEVANSSI_LOG	Suche / Log vergangener Suchbegriffe
WP_1186_RELEVANSSI_STOPWORDS	Suche / Stopwords
WP_1186_STUDENTEN2	Externe Autorenatenquelle
WP_1186_TERMS	Kategorien, Schlagwörter und sonstige Taxonomien
WP_1186_TERM_RELATIONSHIPS	Zuordnung von Taxonomien zu Posts
WP_1186_TERM_TAXONOMY	Taxonomie-Beschreibungen (Kategorie, Tag,...) der wp_1186_terms

Die Datenbankstruktur entspricht großteils der WordPress-eigenen Struktur und bedarf darum an dieser Stelle keiner ausführlichen Erläuterung.

Besonderheiten bilden die Tabellen, welche **relevanssi** im Namen führen. Diese werden automatisch vom gleichnamigen Such-Plugin verwaltet.

Überdies sollte die Tabelle **wp_1186_studenten2** erwähnt werden, die Studierendendaten enthält, welche von der Arbeit einreichen-Maske herangezogen werden, um noch nicht im System vorhandene Autoren mit korrekten Metadaten zu versehen. Es handelt sich dabei um eine Kopie einer KISD-eigenen Studierenden-Datenbank, die Struktur wurde 1:1 übernommen und kann exakt so interpretiert werden. Die Referenz auf die Originaldatenbank sollte jedoch umgesetzt oder aber mit jedem neuen Jahrgang der aktuelle Stand auf diese Tabelle übertragen werden. Siehe **ark-search-author.php** (S. 156).

7.3 WordPress-Plugins

Die folgende Auflistung enthält sämtliche Plugins, die innerhalb des KISD-archivs zum Einsatz kommen und für dessen Betrieb von gesteigerter Bedeutung sind. Dies betrifft besonders die Plugins auf der folgenden Seite. Die Abschaltung eines der auf der folgenden Seite genannten Plugins hätte erhebliche Beeinträchtigungen der KISDarchiv-Funktionen zufolge. Darum sollte darauf geachtet werden, dass vor einem WordPress-Update die Kompatibilität mit sämtlichen hier aufgeführten Plugins gewährleistet wird.

7.3.1 Advanced Custom Fields (ACF)

Das Plugin ermöglicht das Anlegen maßgeschneiderter Eingabemasken im Backend, welche als Schnittstelle zu Datenfeldern dienen.

VERSION 4.1.6
AUTOR Elliot Condon
POSITION /wp-content/plugins/advanced-custom-fields/

7.3.2 ACF Flexible Content Field

Ein Sub-Plugin, welches ACF um ein flexibles Eingabefeld erweitert.

VERSION 1.0.2
AUTOR Elliot Condon
POSITION /wp-content/plugins/acf-flexible-content/

7.3.3 Codepress Admin Columns

Admin Columns erlaubt die Abbildung zusätzlicher Metafelder in Spaltenform innerhalb von Eintrags-Listenansichten im Backend.

VERSION 1.4.9
AUTOR Codepress
POSITION /wp-content/plugins/codepress-admin-columns/

7.3.4 KISDarchive

KISDarchiv-Kernfunktionalitäten sind in diesem Plugin gekapselt.

VERSION 1.0b
AUTOR Willem Rabe
POSITION /wp-content/plugins/KISDarchive/

7.3.5 Relevanssi

Relevanssi fungiert anstelle der WordPress-eigenen Suche und bietet unter anderem eine relevanz-basierte Ausgabe von Suchergebnissen.

VERSION 3.1.4
AUTOR Mikko Saari
POSITION /wp-content/plugins/relevanssi/

7.3.6 Revision Control

Das Plugin erlaubt Feintuning der posttypspezifischen Revisionseinstellungen

VERSION 2.1
AUTOR Dion Hulse
POSITION /wp-content/plugins/revision-control/

7.3.7 Right Now Reloaded

Right Now Reloaded stellt eine Erweiterung des »Right Now« Dashboard-Widgets dar. Es zeigt Statistiken zu Custom Post Types an.

VERSION 2.1
AUTOR Michael Dance
POSITION /wp-content/plugins/right-now-reloaded/

7.3.8 wp-Typography

wp-Typography ist ein umfangreiches Typografie-Plugin inklusive Silbentrennung, Witwenschutz, automatischer Zeichenersetzung bei Anführungszeichen, Strichen, mathematischen Symbolen et cetera.

VERSION 2.0.4
AUTOR Jeffrey D. King
POSITION /wp-content/plugins/wp-typography/

7.3.9 wp Permalauts

WP Permalauts dient der korrekten Darstellung von Umlauten innerhalb von Permalinks. Da WordPress diese von Haus aus nicht in ihre Entsprechungen (Bsp.: ä -> ae) umzuwandeln weiß, ergibt sich auf deutschsprachigen Seiten die Notwendigkeit eines solchen Plugins.

VERSION 1.0.2
AUTOR Christoph Grabo
POSITION /wp-content/plugins/wp-permalauts/

7.3.10 WP Ultimate CSV Importer

Dieses Plugin ermöglicht das massenhafte Einlesen von Daten im CSV-Format. Es wurde für den Import der Altdaten verwandt.

VERSION 3.2.2
AUTOR smackcoders.com
POSITION /wp-content/plugins/wp-ultimate-csv-importer/

7.4 Benutzerrollen

Die WordPress-Benutzerrollen bestimmen, welche Möglichkeiten der Einflussnahme ein Benutzer innerhalb des Systems besitzt. Interessant für den KISD-Archiv-Regelbetrieb sind dabei lediglich die Rolle des Super-Administrators in Form des Teams der Blended Studies, sowie die Editor-Rolle, welche an Mitglieder der Archiv-AG vergeben wird. Studentische Nutzer des Archivs müssen über keine spezifische Nutzerrolle verfügen, da das Einreichen von Arbeiten über die gleichnamige Maske jedem mit Zugriff auf den Space möglich ist.

7.4.1 Super-Administrator

Super-Administratoren haben Zugriff auf die KISDspaces-weiten Netzwerk-Administrations-Einstellungen und sämtliche andere Einstellungen. Dieser Rang wird gewöhnlich den Blended Studies Mitgliedern zuteil, die sich systemweiter administrative Aufgaben annehmen.

7.4.2 Administrator

Die Administrator-Rolle verschafft ihrem Besitzer sämtliche administrativen Befugnisse innerhalb einer einzelnen WordPress Installation der KISDspaces. Diese Rolle wird innerhalb des KISDarchivs nicht zwangsläufig vergeben, da die Super-Administratoren administrative Aufgaben abdecken.

7.4.3 Editor

Nutzer mit Editor-Privilegien sind imstande, Einträge verschiedenen Datentyps zu veröffentlichen und editieren, unabhängig davon, wer diese Einträge angelegt hat. Diese Rolle fällt im KISDarchiv-Regelbetrieb den AG-Mitgliedern zu, welche für die Wartung des digitalen Datenbestands verantwortlich sind.

7.4.4 Author

Autoren-Privilegien bedeuten die Fähigkeiten, eigene Einträge zu schreiben und zu bearbeiten, jedoch keine Möglichkeit, diese zu veröffentlichen. Diese Stufe spielt im KISDarchiv-Regelbetrieb keine Rolle.

7.4.5 Contributor

Der Contributor-Rang ist eine abgeschwächte Variante der Rolle Author und spielt im KISD-Archiv Regelbetrieb keine Rolle.

7.4.6 Subscriber

Subscriber können innerhalb einer WordPress-Installation lediglich ihr Profil verwalten. Die Rolle spielt im KISDarchiv-Regelbetrieb keine Rolle.

8 Abläufe

8.0 Einleitung

Im Folgenden werden diverse Abläufe dargestellt, die zumeist im direkten Bezug zu Posttypen stehen. Diese Abläufe beschreiben das Durchsuchen, Anlegen, Bearbeiten und Löschen von Einträgen, Autoren, Prüfern und Lehrgebieten aus Sicht der Nutzer, AG-Mitglieder und Administratoren. Für letztere enthält dieses Kapitel überdies Anleitungen, die sich mit der Erweiterung des Systems und dessen Wartung auseinandersetzen.

Die Trennung von Abläufen in Front- und Backend suggeriert dabei keinesfalls, dass AG-Mitglieder mit Backendzugang lediglich darin arbeiten. Viele Module des Frontends sind auf die optimale Arbeit mit dem Datenbestand hin konzipiert worden und bieten einen komfortableren Zugang, als es die Word-Press-eigenen Backend-Tools ermöglichen würden.

8.1 Eintrag (ark_record)

Der Umgang mit Einträgen macht den Hauptbestandteil der Arbeit mit dem KISDarchiv aus. Dementsprechend diverse Möglichkeiten der Interaktion mit diesem Posttyp stehen zur Verfügung.

8.1.1 Einträge durchsuchen

Den Eintragsbestand zu durchsuchen, ist auf vielerlei Art und Weise denkbar.

Frontend

Die folgende Tabelle zeigt, welche Module den Datenbestand nach welchen Kriterien durchsuchen (hierarchische/alternative Darstellung nummeriert):

	T	AB	S	PU	PS	TP	LG	PR	AU	JG	SP	I
SUCHE	X	X	X	X	X	(PU)	-	-	X	(PU)	-	-
METASUCHE	-	-	-	-	X	X	X	X	-	X	X	-
NAVIGATOR	5	-	-	-	1	4	-	-	3	2	-	-
AUTORENINDEX	-	-	-	-	-	-	-	-	X	-	-	-
SCHLAGWORTSEITE	-	-	X	-	-	-	-	-	-	-	-	-
INTEGRITÄTSSSEITE	X	-	-	-	-	-	-	-	-	-	-	X
AUTOREN-EINZELANSICHT	2	-	-	-	-	-	-	-	1	-	-	-
LEHRENDEN-EINZELANSICHT	2	-	-	-	-	-	-	1	-	-	-	-
SCHLAGWORT-EINZELANSICHT	2	-	1	-	-	-	-	-	-	-	-	-
	TITEL	ABSTRACT	SCHLAGWORT	PDF-PFAD	PRÜFUNGSSTUFE	TEILPRÜFUNG	LEHRGEBIET	PRÜFER	AUTOR	JAHR	SPRACHE	INTEGRITÄT

Backend

ABB. 161 Backend; Übersicht aller Datensätze des Posttyps `ark_record` (Ausschnitt).

Anmeldung	ID	Titel	Typ	Autor	Lehrgebiet	Dozent	Stat. Such	WZ	Seit	Date
Ergebnis	01707	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Artikel	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...
	01708	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Artikel	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...
	01709	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Artikel	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...
	01710	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Artikel	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...
	01711	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Artikel	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...
	01712	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Artikel	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...
	01713	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Artikel	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...
	01714	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Artikel	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...
	01715	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Artikel	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...	Wider Thesen und die... Wider Thesen und die... Wider Thesen und die...

Die Eintrags-Listenansicht im Backend bietet die Möglichkeit, Einträge nach ihren diversen Metadatenfeldern sortiert in paginierter Form auszugeben. Ein Suchfeld in der oberen rechten Ecke lässt den Datenbestand nach den selben Kriterien durchsuchen, wie es die Suche im Frontend tut. Ein Klick auf den jeweiligen Spaltenkopf sortiert die Darstellung aufsteigend nach dieser Spalte, ein weiterer Klick darauf absteigend. Über die aufsteigende Sortierung lassen sich somit Einträge identifizieren, die in bestimmten Feldern keine Werte führen. Angemerkt werden sollte an dieser Stelle, dass die Sortierung der Spalten Autor, Lehrgebiet und Dozent diese bei Klick nicht alphabetisch sortiert, da die in diesen Feldern gespeicherten Werte numerische IDs in Arrayform beinhalten.

8.1.2 Einträge anlegen / einreichen

Frontend

Das Anlegen neuer Einträge sollte in jedem Fall über die Arbeit einreichen-Maske im Frontend geschehen. Diese wurde eigens zu diesem Zweck konzipiert und bietet gegenüber der Eintragsmaske im Backend eine überlegene Validierung, die Fehleingaben vermeiden soll.

Ein ausführliche Beschreibung der Arbeit einreichen Maske findet sich im ersten Teil. Siehe **Arbeit einreichen** (S. 98).

Backend

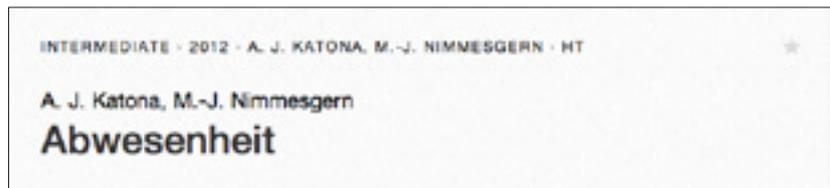
Während das Anlegen von Einträgen im Backend möglich ist, wird dazu geraten, die Eintrag einreichen-Maske im Frontend zu benutzen, da diese eine komfortablere Eingabe inklusive Validierung ermöglicht. Die Backend Eintragsmaske sollte nur in Ausnahmefällen genutzt werden. Sie wird darum in diesem Kontext nicht weiter beschrieben.

8.1.3 Einträge bearbeiten

Frontend

Die Bearbeitung von Einträgen im Frontend ist nicht möglich. Für AG-Mitglieder mit ausreichenden Berechtigungen besteht allerdings die Möglichkeit, per Klick auf den Stern-Button rechts oben im Inhaltsblock eines Eintrags oder in einer der Mehrfachansichten in die Bearbeitungsmaske desselben im Backend zu gelangen.

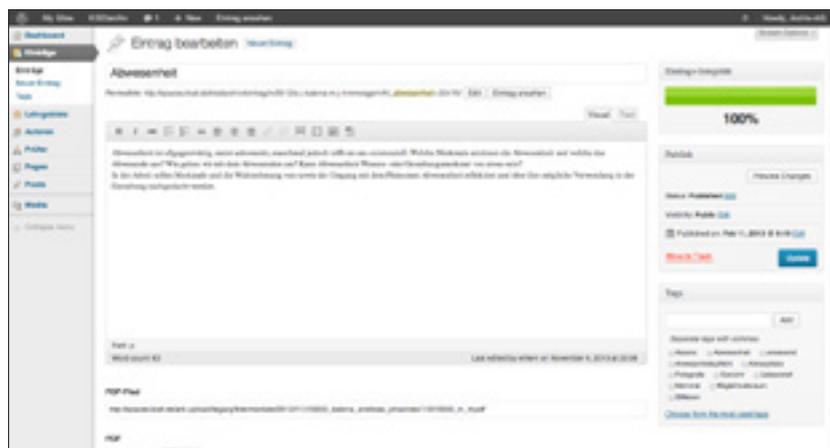
ABB. 162 Inhaltsblock mitsamt des Editier-Buttons in Sternform in der Eintrags-Einzelansicht im Frontend (Ausschnitt)



Backend

Die Eintrags-Bearbeitungsmaske im Backend bildet die Metadaten eines Eintrags vollständig ab.

ABB. 163 Eintrags-Bearbeitungsmaske im Backend (Ausschnitt)



PDF nachpflegen, austauschen

Die Eintrags-Bearbeitungsmaske bietet dem AG-Mitglied die Möglichkeit, Einträge nachträglich mit PDF-Dateien zu versehen oder vorhandene PDF-Dateien auszutauschen. Gehe dazu wie folgt vor:

- Sofern etwas im Feld **PDF-Pfad** steht, entferne es.
- Wähle den Button **Add File** darunter aus, und lade eine Datei über den WordPress-eigenen Upload-Manager hoch. Die Datei wird anschließend unter PDF abgebildet
- Klicke den Button **Update**. Das neu hochgeladene PDF ist nun mit dem Eintrag assoziiert.

HINWEIS Beim Upload von PDFs über das Backend ist die Begrenzung der maximalen Dateigröße höher angesetzt, als dies beim Hochladen über das Frontend der

Fall ist. Dies ist so angelegt, um Dateien ins System übertragen zu können, welche technisch bedingt nicht unter das Limit der Arbeit einreichen-Maske zu bringen sind. Es ist keinesfalls als Aufforderung zu verstehen, gedankenlos große Dateien ins System zu speisen!

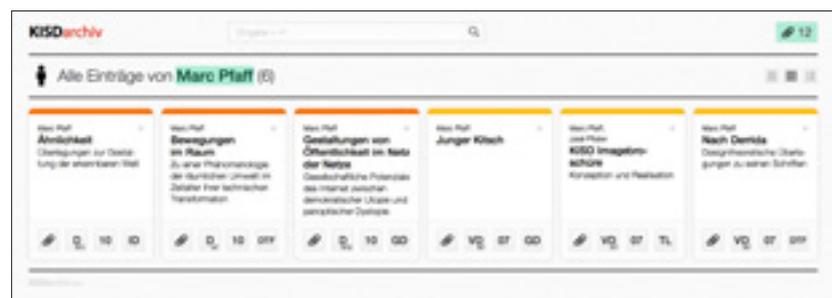
Einträge vervollständigen

Die bereits vorgestellte Eintrags-Bearbeitungsmaske sollte genutzt werden, um Metadaten unvollständiger Einträge nachzutragen. Dabei ist es von großem Vorteil, das gedruckte Exemplar der Arbeit hinzuzuziehen, da die Angaben in diesem mit höherer Wahrscheinlichkeit korrekt sind, als die Metadaten, welche teilweise aus unbekanntem Quellen stammen.

Unvollständige Einträge können auf diverse Art und Weise aufgespürt werden. Bereits in diesem Kapitel beschrieben wurde die Eintrags-Listenansicht im Backend, welche sich nach Metadatenfeldern sortieren und damit nicht ausgefüllte Metadaten in Einträgen hervorheben kann. Spezifische Aufbereitungsmaßnahmen, wie die Ersetzung übergroßer PDF-Dateien, erfordern speziell erstellte Listen von Einträgen. Einige solche Listen liegen mir bereits vor und können auf Anfrage bezogen werden. Aus Datenschutzgründen kann ich sie nicht im Anhang dieser Arbeit abdrucken.

Darüber hinaus hilft eine gute Kenntnis der Studiengangstruktur und die Fähigkeit, Muster zu erkennen. So lassen sich über die Anzeige aller vorliegenden Arbeiten eines Autors bereits mit hoher Wahrscheinlichkeit ablesen, ob etwa die Prüfungsstufen- und Teilprüfungs-Metadaten korrekt eingegeben wurden. Liegen zwei Vordiplom- und drei Bachelorthemen vor, sollte das stutzig machen und deutet darauf hin, dass die Intermediate-Themen zu einer Zeit eingereicht wurden, als diese mitsamt Vordiplom-Abgabeformularen eingereicht wurden.

ABB. 164 Vordiplom- und Diplom-Neben- und Hauptthemen eines Autors.



Ein weiteres Indiz für Nachpflege-Bedarf ist das Vorhandensein von weniger als sechs Arbeiten pro graduiertem Diplom-Prüfling (VD NT1, NT2, HT + D NT1, NT2, HT). Ebenso können mehr als sechs Einträge auf einen Prüfling kommen, beispielsweise bei nicht bestandenem und wiederholten Prüfungen.

Derartige Muster lassen sich schwer katalogisieren. Sie bedürfen der Auseinandersetzung mit dem Archiv in seinen diversen Formen, dem Austausch mit AG-Teilnehmern älterer Semester.

8.2 Autor/in (ark_author)

8.2.1 Autoren durchsuchen

Frontend

Einzelne Einträge lassen sich über die **Suche** finden. Autoren nach Prüfungsstufe und Jahr über den **Navigator**. Autoren nach alphabetisch nach Namen über den **Autorenindex**.

Backend

ABB.165 Backend; Übersicht aller Datensätze des Posttyps ark_author

ID	Name	Type	Taxonomie	Gruppen
10000	Wolfgang Amadeus Mozart	Person	Person	Person
10001	Wolfgang Amadeus Mozart	Person	Person	Person
10002	Wolfgang Amadeus Mozart	Person	Person	Person
10003	Wolfgang Amadeus Mozart	Person	Person	Person
10004	Wolfgang Amadeus Mozart	Person	Person	Person
10005	Wolfgang Amadeus Mozart	Person	Person	Person
10006	Wolfgang Amadeus Mozart	Person	Person	Person
10007	Wolfgang Amadeus Mozart	Person	Person	Person
10008	Wolfgang Amadeus Mozart	Person	Person	Person
10009	Wolfgang Amadeus Mozart	Person	Person	Person
10010	Wolfgang Amadeus Mozart	Person	Person	Person

Per Klick auf den Menüpunkt **Autoren** im Backend öffnet sich eine Übersicht aller im KISDarchiv vertretenen Autorinnen und Autoren. Diese bildet tabellarisch die vorhandenen Metadaten ab. Ein Suchfeld in der oberen rechten Ecke erlaubt das Durchsuchen aller Autoren-Datensätze nach Namen, darüber hinaus lässt sich die Tabelle nach jeder ihrer Spalten sortieren (ein Klick auf den jeweiligen Spaltenkopf für aufsteigende, ein weiterer Klick für absteigende Sortierung). Ein Klick auf die taxonomische Zuordnung in der Spalte **Gruppen** zeigt lediglich Datensätze an, die einer bestimmten Gruppe – in diesem Fall also **Studentinnen** oder **Studenten** – zugeordnet wurde.

8.2.2 Autoren anlegen

Frontend

Das direkte Anlegen von Autoren im Frontend ist nicht möglich. Die Arbeit einreichen Maske verfügt über die Möglichkeit, Autoren, welche nicht im System, wohl aber in einer externen Datenbank abgelegt sind, automatisch anzulegen. Dies sollte jedoch nur genutzt werden, wenn ohnehin ein Eintrag angelegt wird. Soll lediglich ein Autor angelegt werden, so nutze das Backend:

Backend

Autoren aus dem Backend heraus anlegen, gestaltet sich wie folgt:

ABB. 166 Backend; Autoren direkt aus der Seitennavigation heraus anlegen.



Über die Seitennavigation des Backends lässt sich der Punkt **Neuer Autor** auswählen, woraufhin die entsprechende Eingabemaske geladen wird:

ABB. 167 Backend; Eingabe- und Bearbeitungsmaske des Datentyps **ark_author**.



Die folgenden Felder sind auszufüllen. Siehe **ark_author / Autor/in** (S. 119).

KISD ID	Die KISD-ID des Studierenden alias »das, was in der E-Mail-Adresse vor dem @kisd.de steht«. Entspricht dem Datenfeld ark_author_email .
VORNAME	Der Vorname des Studierenden. Entspricht dem Datenfeld ark_author_firstname .
NACHNAME	Der Nachname des Studierenden. Entspricht dem Datenfeld ark_author_lastname .
MATRIKELNUMMER	Die Matrikelnummer des Studierenden. Entspricht dem Datenfeld ark_author_mnr .
STUDIENGANG	Der Studiengang des Studierenden. Das Auswahlfeld lässt eine Mehrfachauswahl zu. Entspricht dem Datenfeld ark_author_degree .
BEGINN	Der Zeitpunkt des Studienbeginns. Entspricht dem Datenfeld ark_author_start .
GRUPPEN	Die Zuordnung zur Taxonomie ark_author_groups . Zulässige Werte: studenten , studentinnen .

Nach Ausfüllen dieser Felder genügt ein Klick auf den **Publish**-Button, um den Autoren-Datensatz freizuschalten.

8.2.3 Autoren bearbeiten / löschen

Das Bearbeiten und Löschen von Autoren-Datensätzen.

Frontend

Die Bearbeitung von Autoren-Datensätzen im Frontend ist nicht möglich. Für AG-Mitglieder mit ausreichenden Berechtigungen besteht allerdings die Möglichkeit, per Klick auf den Stern-Button neben dem einen Eintrag zugeordneten Autor, in die Bearbeitungsmaske im Backend zu gelangen.

ABB. 168 Autoren-Metadaten in der Eintrags-Einzelansicht im Frontend.



Backend

Von der Autoren-Übersicht oder dem Frontend aus lässt sich die Bearbeitungs- und Eingabemaske des Posttyps **ark_author** aufrufen, welche bereits auf Seite 211 vorgestellt wurde. Diese erfüllt die Funktion, Autoren Daten einzutragen, über die Arbeit einreichen-Maske generierte Autoren freizuschalten und bestehende Autoren Daten zu editieren oder zu löschen.

HINWEIS

In den seltensten Fällen ist es vonnöten, einen Autoren zu löschen. Dies ist dem Umstand geschuldet, dass potenziell jede studierende Person, welche eine Prüfung an der KISD abgelegt hat, als Autor infrage kommt. Findet man sich trotzdem in der Situation wieder, einen Autoren löschen zu wollen – weil etwa versehentlich eine Person doppelt angelegt wurde und nun das Duplikat entfernt werden soll – so sollte man vor Löschen des Autors sicherstellen, dass dieser Person keine Einträge zugeordnet sind, da eine Löschung sonst Inkonsistenzen innerhalb der Datenbank zufolge hätte. Dies kann überprüft werden, indem der zu löschende Autor aus der Bearbeitungsmaske heraus über den Button **Autoren ansehen** aufgerufen wird.

ABB. 169 Autoren ansehen-Button in der Autoren-Bearbeitungs-maske im Backend.



8.3 Prüfer/in (ark_lecturer)

8.3.1 Prüfer durchsuchen

Frontend

Das Suchen von Prüfern über das Frontend ist nicht direkt möglich, da Prüfer selbst über keine eigene Seite verfügen, sondern lediglich anhand der von ihnen geprüften Arbeiten repräsentiert werden.

Über die reguläre Suche lassen sich alle Prüfungsberechtigten Personen finden und per Klick auf ihren Namen in diversen Kontexten alle mit ihnen assoziierten Einträge anzeigen.

HINWEIS Über die Eingabe des Suchbegriffs **profs** wird eine Liste aller Professorinnen und Professoren ausgegeben.

Backend

ABB. 170 Backend; Übersicht aller Einträge vom Posttyp **ark_lecturer** (Ausschnitt).

ID	Name	Title	Taxonomy	Nickname	Email	Submitter	Author	Slug
4278	Prof. Dr. Oliver Baren	Prof. Dr.	Oliver	Baren	o.baren@uni-wuerzburg.de	Design und Technologie	Prof.	prof-dr-oliver-baren
4279	Prof. Dr. Oliver	Prof.	Dr.	Oliver	o.oliver@uni-wuerzburg.de	Technik und Lernen	Prof.	prof-dr-oliver
4280	Prof. Dr. Ingrid Wagner	Prof.	Ingrid	Wagner	i.wagner@uni-wuerzburg.de	Service Design	Prof.	prof-dr-ingrid-wagner
4281	Prof. Dr. Ingrid Wagner	Prof.	Ingrid	Wagner	i.wagner@uni-wuerzburg.de	Design und Technologie	Prof.	prof-dr-ingrid-wagner
4282	Prof. Dr. Ingrid Wagner	Prof.	Ingrid	Wagner	i.wagner@uni-wuerzburg.de	Service Design	Prof.	prof-dr-ingrid-wagner

Per Klick auf den Punkt **Prüfer** im Backend erscheint eine Liste aller Einträge vom Posttyp **ark_lecturer**. Diese bildet tabellarisch die vorhandenen Metadaten ab. Ein Suchfeld in der oberen rechten Ecke erlaubt das Durchsuchen aller Datensätze nach Namen, darüber hinaus lässt sich die Tabelle nach jeder ihrer Spalten sortieren (ein Klick auf den jeweiligen Spaltenkopf für aufsteigende, ein weiterer Klick für absteigende Sortierung).

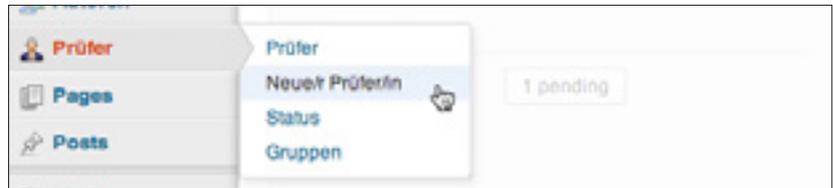
8.3.2 Prüfer anlegen

Frontend

Aus dem Frontend heraus besteht keine Möglichkeit, Prüfer anzulegen.

Backend

ABB. 171 Backend; Prüfer direkt aus der Seitennavigation heraus anlegen.



Neue Prüfer können per Klick auf den gleichnamigen Menüpunkt im Backend angelegt werden. Daraufhin öffnet sich die Eingabe- und Bearbeitungsmaske für Einträge des Posttyps **ark_lecturer**:

ABB. 172 Backend; Eingabe- und Bearbeitungsmaske des Posttyps **ark_lecturer**.



Die folgenden Felder sind auszufüllen. Die Eingabefelder entsprechen Datenfeldern innerhalb des Posttyps **ark_lecturer**.

Siehe **ark_lecturer / Prüfer/in** (S. 120).

- TITEL** Der akademische Titel des Prüfers.
Entspricht dem Datenfeld **ark_lecturer_title**.
- VORNAME** Der Vorname des Prüfers.
Entspricht dem Datenfeld **ark_lecturer_firstname**.
- NACHNAME** Der Nachname des Prüfers.
Entspricht dem Datenfeld **ark_lecturer_lastname**.
- KÜRZEL** Das Kürzel des Prüfers.
Entspricht dem Datenfeld **ark_author_shortcode**.
- E-MAIL-ADRESSE** Die E-Mail-Adresse des Prüfers.
Entspricht dem Datenfeld **ark_author_email**.

LEHRGEBIET	Die Zuordnung des Lehrgebiets des Prüfers. Entspricht dem Datenfeld ark_lecturer_area .
STATUS	Die Zuordnung zur Taxonomie ark_lecturer_role .
GRUPPEN	Die Zuordnung zur Taxonomie ark_lecturer_groups . Zulässige Werte: <ul style="list-style-type: none"> – Prüfer – Prüferinnen

Nach Ausfüllen dieser Felder genügt ein Klick auf den Publish-Button, um den Prüfer-Datensatz freizuschalten.

8.3.3 Prüfer bearbeiten / löschen

Frontend

Die Bearbeitung von Prüfer-Datensätzen im Frontend ist nicht möglich. Für AG-Mitglieder mit ausreichenden Berechtigungen besteht allerdings die Möglichkeit, per Klick auf den Stern-Button neben dem einen Eintrag zugeordneten Prüfer, in die Bearbeitungsmaske im Backend zu gelangen.

ABB. 173 Prüfer-Metadaten in der Eintrags-Einzelansicht im Frontend.



Backend

Von der Prüfer-Übersicht oder dem Frontend aus lässt sich die Bearbeitungs- und Eingabemaske des Posttyps **ark_lecturer** aufrufen, welche bereits auf Seite 215 vorgestellt wurde. Diese erfüllt die Funktion, Prüferdaten einzutragen und bestehende Prüfer zu editieren oder zu löschen.

HINWEIS In den seltensten Fällen ist es vonnöten, einen Prüfer zu löschen. Möchte man dies trotzdem tun, so sollte man vor Löschen des Prüfers sicherstellen, dass dieser Person keine Einträge zugeordnet sind, da eine Löschung sonst Inkonsistenzen innerhalb der Datenbank zufolge hätte. Dies kann überprüft werden, indem der zu löschende Prüfer aus der Bearbeitungsmaske heraus über den Button **Prüfer/in ansehen** aufgerufen wird.

ABB. 174 Prüfer/in ansehen-Button in der Prüfer-Bearbeitungs- maske im Backend.



8.4 Lehrgebiet (ark_area)

8.4.1 Lehrgebiete durchsuchen

Frontend

Das Suchen von Lehrgebieten über das Frontend ist nicht direkt möglich, da Lehrgebiete selbst über keine eigene Seite verfügen, sondern lediglich anhand der von ihnen geprüften Arbeiten repräsentiert werden.

Über die Metasuche lassen sich alle einem Lehrgebiet zugeordneten Einträge via Eingabe des Lehrgebietskürzels oder -namens anzeigen.

Backend

ABB. 175 Backend; Übersicht aller Einträge des Typs `ark_area` (Ausschnitt).

ID	Name	Typ	Status	Erstellt	Geändert
81	Archivische Medien	Strukturierte	Planerisch	1	
82	Design and Economics	Strukturierte	Planerisch	1	08
83	Design for Manufacturing	Strukturierte	Planerisch	1	
84	Designlogik	Strukturierte	Planerisch	1	
188	Designtheorie und -entwicklung	Strukturierte	Planerisch	1	07
205	Photografie	Strukturierte	Planerisch	1	

Per Klick auf den Punkt **Lehrgebiete** im Backend erscheint eine Liste aller Einträge vom Posttyp `ark_area`. Diese bildet tabellarisch die vorhandenen Metadaten ab. Ein Suchfeld in der oberen rechten Ecke erlaubt das Durchsuchen aller Datensätze nach Namen, darüber hinaus lässt sich die Tabelle nach jeder ihrer Spalten sortieren (ein Klick auf den jeweiligen Spaltenkopf für aufsteigende, ein weiterer Klick für absteigende Sortierung).

8.4.2 Lehrgebiete anlegen

Frontend

Das Anlegen von Lehrgebieten aus dem Frontend heraus ist nicht möglich.

Backend

ABB. 176 Backend; Lehrgebiet direkt aus der Seitennavigation heraus anlegen.



Neue Lehrgebiete können per Klick auf den gleichnamigen Menüpunkt im Backend angelegt werden. Daraufhin öffnet sich die Eingabe- und Bearbeitungsmaske für Einträge des Posttyps **ark_area**:

ABB. 177 Backend; Eingabe- und Bearbeitungsmaske des Posttyps **ark_area**.



Die folgenden Felder sind auszufüllen. Die Eingabefelder entsprechen Datenfeldern innerhalb des Posttyps **ark_lecturer**.

Siehe **ark_area / Lehrgebiet** (S. 118).

TITELFELD	Die ausgeschriebene Form des Lehrgebietsnamens. Entspricht dem Datenfeld post_title .
KURZFORM	Das Lehrgebietskürzel Entspricht dem Datenfeld ark_area_ID .
BESCHREIBUNG	Ein Beschreibungstext (findet derzeit keine Verwendung). Entspricht dem Datenfeld ark_area_description .
ANZEIGEN	Das Anzeige-Flag für die Darstellung im Frontend. Entspricht dem Datenfeld ark_area_show .
VERALTETE KÜRZEL	Ehemalige, dem Lehrgebiet zugeordnete Kürzel. Entspricht dem Datenfeld ark_area_former_ID .

Nach Ausfüllen dieser Felder genügt ein Klick auf den Publish-Button, um den Lehrgebiets-Datensatz freizuschalten.

Implementierung neuer Lehrgebiete innerhalb der Metasuche

Um etwaige neue Lehrgebiete innerhalb der Metasuche suchbar zu machen, muss diese um entsprechenden Code erweitert werden, da die Lehrgebiete dort hardcoded vorkommen. Da besagte Code-Erweiterung mit einem gewissen Zeitaufwand einher geht, der der Dokumentation seiner selbst gleichkommt und die Wahrscheinlichkeit eines neuen Lehrgebiets gering scheint, wird an dieser Stelle auf die Dokumentation einer solchen Erweiterung verzichtet. Für den unwahrscheinlichen Fall, dass jedoch ein Lehrgebiet hinzukommen sollte, ist der Autor dieses Dokuments erreichbar und imstande, die Modifikation am Code vorzunehmen.

8.4.3 Lehrgebiete bearbeiten

Frontend

Die einzige Möglichkeit, vom Frontend aus in die Bearbeitungsmaske eines Lehrgebiets zu gelangen, ist, einen Eintrag welcher ebendiesem Lehrgebiet zugeordnet ist, zu öffnen, und über den sternförmigen Editierbutton dieses im Backend in seiner Bearbeitungsansicht zu öffnen.

Backend

Von der Lehrgebiets-Übersicht oder dem Frontend aus lässt sich die Bearbeitungs- und Eingabemaske des Posttyps **ark_area** aufrufen, welche bereits auf Seite 217 vorgestellt wurde. Diese erfüllt die Funktion, Lehrgebiete einzutragen und bestehende Lehrgebiete zu editieren.

HINWEIS Es gibt keinen Grund, ein Lehrgebiet zu löschen. Diese sollten immer erhalten bleiben. Findet eine Umbenennung eines Lehrgebiets statt, wie dies etwa im Falle des Lehrgebiets **Identität und Design** geschehen ist, so muss diese Umbenennung im Datenfeld **veraltete Kürzel** abgebildet werden. Wird hingegen ein neues Lehrgebiet geschaffen, welches nicht aus einem bestehenden Lehrgebiet hervorgeht, so kann dieses auch neu angelegt werden. Die Aussetzung eines aktuellen Lehrgebiets ist jedoch kein Lösungsgrund, da die dem Lehrgebiet zugeordneten Datensätze weiterhin korrekt abgebildet werden sollen.

8.5 Studiengang

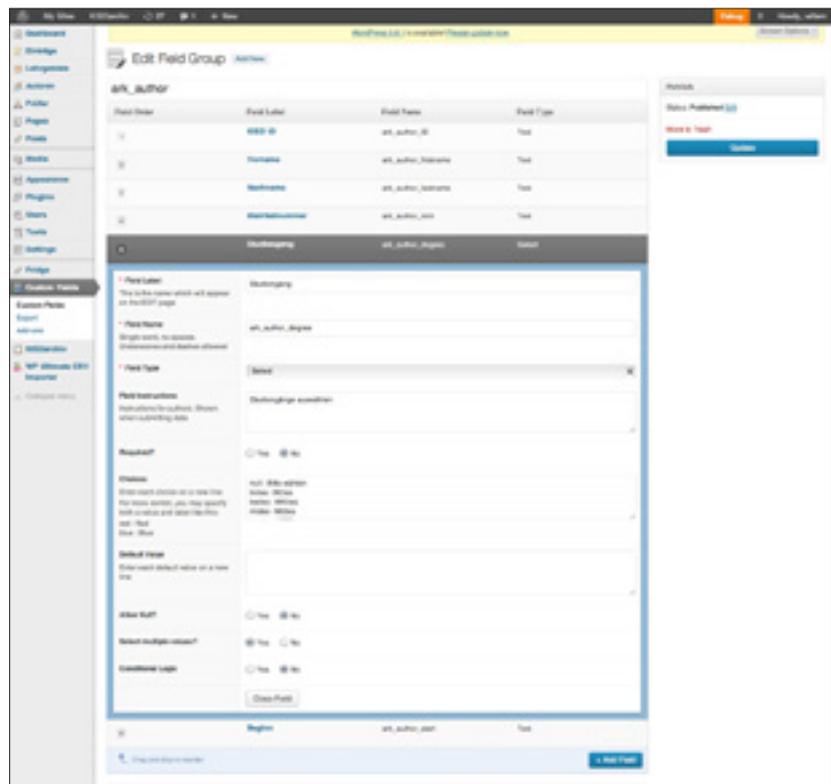
8.5.1 Studiengang anlegen

HINWEIS Dieser Punkt richtet sich an Menschen, welche administrativ mit dem System zu tun haben. Die Umsetzung einer wie hier beschriebenen Änderung sollte von jemandem vorgenommen werden, der langjährige WordPress-Erfahrung und einen Hintergrund in der Informatik hat.

Es ist damit zu rechnen, dass an einem gewissen Punkt die Erweiterung des KISDarchivs um einen neuen Studiengang bevorsteht. Die Implementierung einer solch tiefgreifenden Veränderung in der Struktur stellt sich nicht ganz unkompliziert dar. Ideal wäre eine dynamische Studiengangsverwaltung im Backend, die die Definition aller Studiengänge mit samt ihrer Teilprüfungen und deren Abhängigkeiten erlaubt. Die Entwicklung einer solchen Lösung steht in keinem Verhältnis zum daraus zu ziehenden Nutzen, weshalb an dieser Stelle beschrieben werden soll, an welchen Stellen Eingriffe nötig sind, um einen Studiengang manuell zu implementieren.

Backend: Custom Fields

ABB. 178 Backend; Einstellungsseite Custom Fields / ark_author / ark_author_degree.



Für Administratoren zugänglich bietet der Menüpunkt **Custom Fields** den Zugang zur Verwaltung der diversen Eintrags-Bearbeitungsmasken des Backends. Hier kann die Maske für den Posttyp **ark_author** aufgerufen und angepasst werden. Im Detail muss hier das Feld **ark_author_degree** geöffnet und unter dem Punkt Choices eine Deklaration nach dem Muster `wert : Name` hinzugefügt werden. Die bisherigen Werte im Feld können als Beispiel dienen.

Frontend: Arbeit Einreichen-Maske

Die Arbeit einreichen-Maske bildet Studiengänge, Prüfungsstufen und Teilprüfungen in Abhängigkeit von einander ab. Die Implementierung ist in den Dateien **functions.php** und **ark_upload.php** zu finden. Grundsätzlich sollte der bestehende Code als anschauliches Beispiel dienen. Eine verzeichnisweite Suche nach der Zeichenkette 'bides' etwa wird alle Stellen offenlegen, an denen BIDES-spezifischer Code exekutiert wird, was mit hoher Wahrscheinlichkeit bedeutet, dass ein neuer Studiengang an dieser Stelle ebenfalls Erwähnung finden sollte.

HINWEIS Bei Fragen stehe ich gerne zur Verfügung: willem@kisd.de.

Mehrfachansicht »Detail«

Die Mehrfachansicht Detail enthält studiengangsspezifischen Code innerhalb des Templates (**ark_view_detail.php**, Z. 66-71).

Funktionen **ark_get_author_degree()**, **ark_get_record_subexam()**

Die Funktion zur Ausgabe des Studiengangs, deklariert in der **functions.php**, sollte ebenfalls angepasst werden, um die korrekte Ausgabe zu gewährleisten. Selbiges gilt für die Funktion zur Ausgabe von Teilprüfungen, welche diese in Abhängigkeit von Studiengang benennt.

JavaScript

An diversen Stellen innerhalb der Haupt-JavaScript-Datei wird auf den Studiengang Bezug genommen (Betreffende Datei: **ark-1.0.js** (zu finden im **assets**-Ordner, Betreffende Zeilen: 591-645, 778-793).

Sonstiges

Werden überdies neue Prüfungsstufen eingeführt, erweitert sich die Zahl der zu modifizierenden Dateien. In diesem Falle bietet es sich an, den Code nach Vorkommnissen der bestehenden Prüfungsstufen (**ark_record_exam_type**) zu durchsuchen. Betroffen wäre dann auch das Template der Metasuche sowie die regular expressions, welche die Nutzereingaben in ein Datenbank-Query übersetzt.

HINWEIS Die Löschung eines Studiengangs sollte in keinem Fall geschehen, da dadurch Assoziationen des Studiengangs mit Studierenden korrumpiert würden.

Dritter Teil

EVOLUTION

1 Einleitung

Das Projekt befindet sich in einem fortgeschrittenen Stadium. Die Tatsache, dass es dennoch nach wie vor unter dem Bezeichner »beta« operiert, will als Aufforderung an die künftigen Hüter des KISDarchivs verstanden werden, das System weiter auszubauen und die zugrundeliegende Datenbasis aufzuwerten. Im dritten Teil dieser Arbeit möchte ich Möglichkeiten skizzieren, das System zu verbessern.

2 Daten

2.0 Einleitung

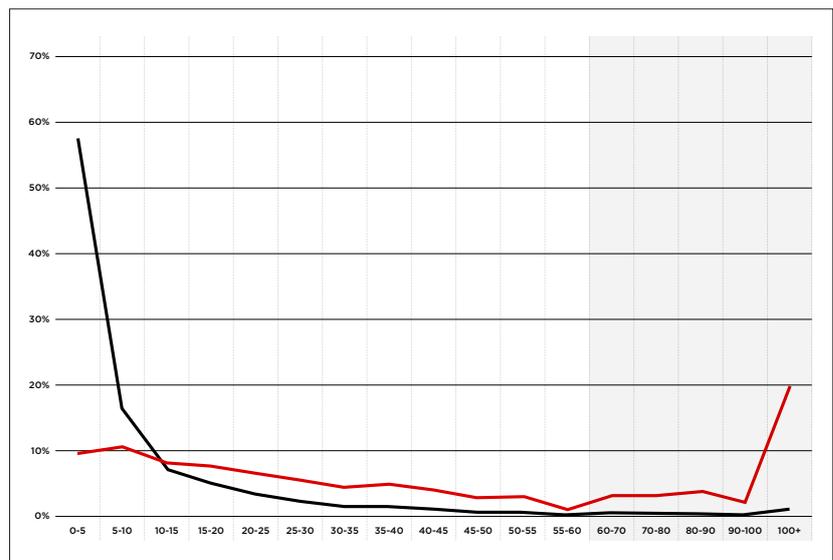
Die Pflege des digitalen Datenbestands ist Aufgabe der Archiv-AG. An dieser Stelle sollen die wichtigsten Baustellen identifiziert werden, welche die Aufmerksamkeit der AG in unmittelbarer Zukunft verdienen.

2.1 Aufbereitung vorhandener Daten

2.1.1 PDF-Datenbestand

ABB. 179 PDF-Datenbestand der Prüfungsarbeiten (November 2013)

X-Achse:
Dateigröße in Megabyte
Y-Achse:
Prozentualer Anteil am Gesamtbestand
Schwarzer Graph:
Anteil an der Gesamtmenge
Roter Graph:
Anteil an der Gesamtgröße



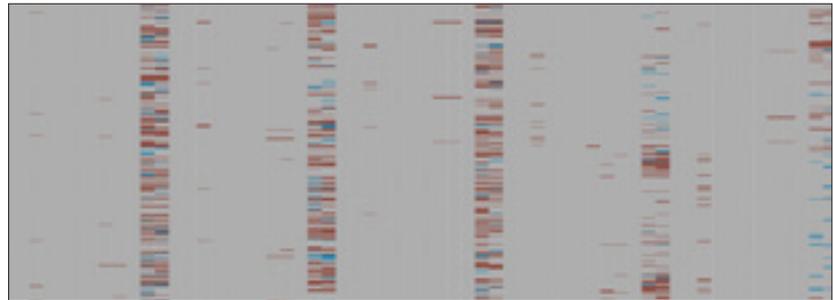
Eine Erhebung der Dateigrößen im PDF-Datenbestand ergab eine durchschnittliche Dateigröße von 11,19 MB. 74% des Gesamtdatenbestands liegt unter der Grenze von 10 MB. Diese 3151 Arbeiten machen 20% der Gesamtdatenmenge aus, während die fünf Prozent der größten Arbeiten ein Drittel der Gesamtdateigröße ausmachen.

Derzeit überschreitet ein Viertel des Datenbestands – über 1000 Dateien – eine Dateigröße, die für die Web-Darstellung als wünschenswert betrachtet wird. Es erscheint als Gewinn für die Nutzbarkeit des KISDarchivs, wenn die Archiv-AG Ressourcen bereitstellte, um sich diesem Problem anzunehmen und den Altdatenbestand systematisch zu optimieren. Eine Liste aller auf im

Datenbestand vorhandener PDF-Dateien nach Größe liegt vor. Es sind nun diverse Herangehensweisen denkbar. Naheliegender wäre es, am oberen Ende der Skala zu beginnen und sich in erster Instanz den größten vorliegenden Arbeiten zu widmen. Bereits eine Optimierung der 100 größten Dateien auf Dateigrößen unterhalb der 10 Megabyte-Grenze hätte eine Platzersparnis von über 12 Gigabyte oder eine Verringerung des Gesamtdatenbestands um etwa 15% zur Folge. So ließe sich inkrementell der PDF-Bestand verschlanken.

2.1.2 Metadatenbestand

ABB. 180 Integritätsdiagramm
(Ausschnitt, Nov. 2013)



Die Optimierung des Metadatenbestands hat das Potenzial, die Nützlichkeit des KISDarchivs drastisch zu steigern. Zwar liegt die Integrität des Metadatenbestands derzeit (März 2014) über 92%, doch diese Zahl kann täuschen:

- 2791 Einträge gelten als unvollständig.
- 1364 davon verfügen kein ausreichendes Abstract,
- 1152 davon wiederum über keinerlei Abstract.

Arbeiten ohne Metadaten sind nicht auffindbar. Sobald eine Arbeit über ein Abstract verfügt, erhöht das ihre Chance, gefunden und angezeigt zu werden.

Die gewissenhafte Ergänzung der Metadaten durch die Archiv-AG ist die wichtigste Aufgabe derselben im KISDarchiv. Die AG ist dafür optimal ausgestattet: Fehlende Metadaten können im Zweifelsfall direkt im analogen Bestand recherchiert und nachgetragen werden.

2.2 Digitalisierung von Altdaten

Die Aufbereitung vorhandener Daten stellt wie bereits dargestellt die wichtigste Aufgabe der Archiv-AG innerhalb des KISDarchivs dar. Doch diese Aufgabe ist rechnerisch endlich. So unwahrscheinlich es ist, dass das Integritätsdiagramm einmal eine ungebrochene grüne Fläche darstellt und sich damit selbst obsolet macht – Es wäre nur der Anfang gemacht.

Die Digitalisierung und Einpflege aller nicht digital vorliegenden Dokumente wäre ein lohnenswertes Unterfangen, sofern alle vorliegenden Daten wie im vorigen Punkt beschrieben aufbereitet wurden. Dabei müsste man sich Gedanken über eine Form der Digitalisierung machen, welche Dokumente erzeugt, die im Idealfall durchsuchbar sind und die Dateigröße der Arbeit einreichen-Maske nicht überschreiten.

3 Struktur

3.0 Einleitung

Das KISDarchiv in seiner jetzigen Form ist das Resultat der »Erfahrung des Chaos«. Es ist ein Werkzeug, geschaffen um mit großen Mengen an Daten umzugehen. Doch es ist ständiger Veränderung ausgesetzt. Das Team der Blended Studies wird sich der strukturellen Pflege und Erweiterung des KISDarchivs annehmen.

An dieser Stelle möchte ich einige Anregungen zu möglichen Erweiterungen anbringen, deren Umsetzung sich als sinnvoll erweisen könnte.

3.1 Erweiterungen

3.1.1 Hilfe / Feedback-System

Die Metasuche verfügt bereits über eine kontextuelle Hilfe. Wünschenswert wäre eine seitenweit implementierte Hilfe in Verknüpfung mit einer Feedback-Maske. Diese könnte einerseits als FAQ dienen welches den Nutzern die Möglichkeit bietet, Fragen einzusenden, andererseits ist eine eintragspezifische Feedbackmaske denkbar, welche Nutzern und AG-Mitgliedern gleichermaßen ermöglicht, unvollständige oder inkorrekte Einträge direkt aus dem Frontend zu korrigieren und somit die Datenpflege via Crowdsourcing teilweise auszulagern. Die AG hätte dabei selbstverständlich die Kontrolle über jegliche Änderungen, bevor diese in Kraft treten.

In diesem Rahmen wäre auch die Produktion weiterer Tutorial-Videos denkbar, wie sie bereits in zweifacher Ausführung geschaffen wurden.

3.1.2 Studiengangsverwaltung

Die derzeitigen Studiengänge werden nicht die Einzigen bleiben – Derzeit bedeutet die Implementierung eines neuen Studiengangs ins System noch einen gewissen Aufwand, was der Tatsache geschuldet ist, dass die Struktur der Studiengänge selbst recht unterschiedlich ist, was etwa in der Arbeit einreichen-Maske abgebildet werden will. Ein System, welches diese Abhängigkeiten mittels eines Interfaces verwaltbar macht und seitenweit umzusetzen weiß, wäre eine deutliche Arbeitserleichterung bei der Implementierung neuer Studiengänge im KISDarchiv.

3.1.3 Verflechtung mit KISDspaces

Eine Verflechtung mit anderen Seiten innerhalb der KISDspaces ist etwa über einen WordPress-Shortcode denkbar, wie dies bereits im Rahmen von KISDbiblio implementiert ist. Auf diese Weise könnten etwa die User Spaces recht

einfach über Referenzen zu vorhandenen Arbeiten der jeweiligen Autoren im KISDarchiv erweitert werden, die als Schnittstelle zwischen diesen und der Eintrags-Einzelansicht im KISDarchiv dienen würden.

3.1.4 Lehrveranstaltungsdokumentationen

Es wurde bereits dargelegt, dass die derzeit 220 digital vorhandenen Projekt- und Seminardokumentationen nur einen Bruchteil des gesamten digitalen Datenbestands abbilden, was maßgeblich dafür verantwortlich zu machen ist, warum diese zum jetzigen Zeitpunkt nicht im KISDarchiv geführt werden. LV-Dokumentationen folgen überdies einer gegenüber Prüfungsarbeiten anderen strukturellen Logik: Sie sind zumeist größeren Gruppen von Autoren zugeordnet und überdies auf eine Veranstaltungsdatenbank angewiesen, um korrekt abgebildet werden zu können.

Für den Augenblick werden die LV-Dokumentationen weiterhin über eine Listenansicht im Gruppenordner, verlinkt über den Archiv-Space, zugänglich gemacht. Darüber hinaus scheint es erforderlich, die vorhandenen Dokumentationen zu sichten und auf ihre Qualität hin zu überprüfen, um diese in die KISDspaces, welche schließlich die digitale Repräsentation der Lehrveranstaltungen bilden, zu integrieren. Eine Aufnahme von LV-Dokumentationen in den KISDarchiv-Bestand scheint keine glückliche Lösung, wie bereits im Kapitel Streamlining dargestellt wurde. Siehe **Streamlining** (S. 24).

3.1.5 Modulare Startseite

Eine Startseite, welche dem Nutzer die flexible Anordnung der KISDarchiv-Module erlaubt, wäre eine willkommene Erweiterung. Überdies sind eine Reihe alternativer Module denkbar, etwa eine Entsprechung des Autoren-Index für Lehrende und Lehrgebiete beziehungsweise eine Übersicht von Autoren gruppiert nach Jahrgängen. Letzteres setzt eine Datenintegrität der Autordaten voraus, die derzeit nicht gegeben ist.

3.1.6 Wartungstools

Davon ausgehend, dass die Autordatenbank eine gewisse Integrität erreicht haben wird, wäre ein Wartungstool denkbar, welches auf Basis des eines Autoren zugeordneten Studiengangs ermittelt, ob zwischen Typ und Anzahl dem Autoren zugeordneter Einträge und der erwarteten Menge eine Diskrepanz besteht. Die Archiv-AG würde somit auf Lücken im Datenbestand respektive fälschlich abgelegte Arbeiten – etwa nicht bestandene Arbeiten – aufmerksam gemacht und könnte diese systematisch ausgleichen.

4 TODO-Liste

Eine annotierte Liste offener TODO-Items.

Die folgende Liste enthält Punkte, die während oder unmittelbar vor Bearbeitung dieser Arbeit aufgetreten sind. Es handelt sich dabei nicht um Bugs deren Nichtbeachtung die Systemstabilität gefährden könnte, sondern vielmehr um kleine Inkonsistenzen, die in ihrer Bedeutung dem nicht sichtbaren Staub auf den Regalen des analogen Archivs gleichkommen. Die Liste dient also eher als Erinnerung, dass die Arbeit am System nicht abgeschlossen ist.

TODO-ID	BESCHREIBUNG
ARK_TD_00	Siehe <code>ark_get_excerpt_by_id()</code> (S. 129). Der Konsistenz halber sollte 'ID' im Funktionsaufruf groß geschrieben werden.
ARK_TD_01	Siehe <code>ark-search-author.php</code> (S. 156). Die Funktion greift derzeit auf die Tabelle <code>wp_1186_students2</code> zu. Diese Tabelle spiegelt eine KISD-interne Studierendendatenbank. Die Originaldatenbank sollte anstelle des Duplikats referenziert werden, um auch in Zukunft aktuelle Studierendendaten zur Verfügung zu stellen.
ARK_TD_02	Siehe <code>ark_view_table.php</code> (S. 193). Die Tabellenzeilendarstellung sollte in eine eigene Funktion gekapselt werden.
ARK_TD_03	Siehe <code>pdf.php</code> (S. 195). Die Template-Datei sollte in den Theme-Ordner relokalisiert werden.
ARK_TD_04	Responsiveness Die seitenweite Anpassung diverser Gestaltungselemente, insbesondere deren Skalierung in diversen Auflösungskontexten, wäre ratsam.
ARK_TD_05	Funktionskonsolidierung Die derzeitige Code-Basis ist das Produkt einer langen Entwicklung und damit nicht als optimal zu betrachten. Die Generalisierung bestimmter Funktionen, und damit Verschlinkung des Projekts wäre denkbar.
ARK_TD_06	SVG-Icons Die Ersetzung der Raster-Piktogramme durch SVG-Varianten würde den seitenweiten und auflösungsunabhängigen Einsatz derselben erlauben.
ARK_TD_07	Siehe <code>tag.php</code> (S. 186). Die Sortierung der Einträge in der Schlagwort-Einzelansicht sollte angepasst werden um den anderen Ansichten folgend alphabetisch zu erscheinen.

Vierter Teil

APPENDIX

1 Wireframes

1.0 Einleitung

In dieser Arbeit finden sich keine Wireframes oder Photoshop-Mockups, die das finale Design des KISDarchivs abbilden. Der Grund dafür ist, dass keine solchen Dokumente angefertigt wurden. Der Grund dafür wiederum ist, dass ich der Überzeugung war und bin, ein System dieser Größenordnung nur im kontinuierlichen Umgang mit dem darin gefassten Daten so begreifen zu können, dass ich mich in der Lage sehe, ein Interface zu gestalten, welches diesen Daten eine Form auferlegt.

Dies soll nicht gegen Wireframes sprechen, welche ihre Daseinsberechtigung haben – Gerade in Projekten die mehrere Parteien über längere Zeiträume hinweg involvieren. Die hier gezeigten Wireframes bildeten den Ausgang für die Auseinandersetzung mit dem KISDarchiv.

1.1 SI »Arche«

Die folgenden Seiten beinhalten die Wireframes, welche im Rahmen des selbstinitiierten Projekts »Arche. Reanimation des Archivs« von Tim Zähres und Laura Schraven konzipiert wurden.



ABB. 181 Wireframes Digital Archiv – Einstellprozess / Legende / 5.12.2011 (S. 1/22)

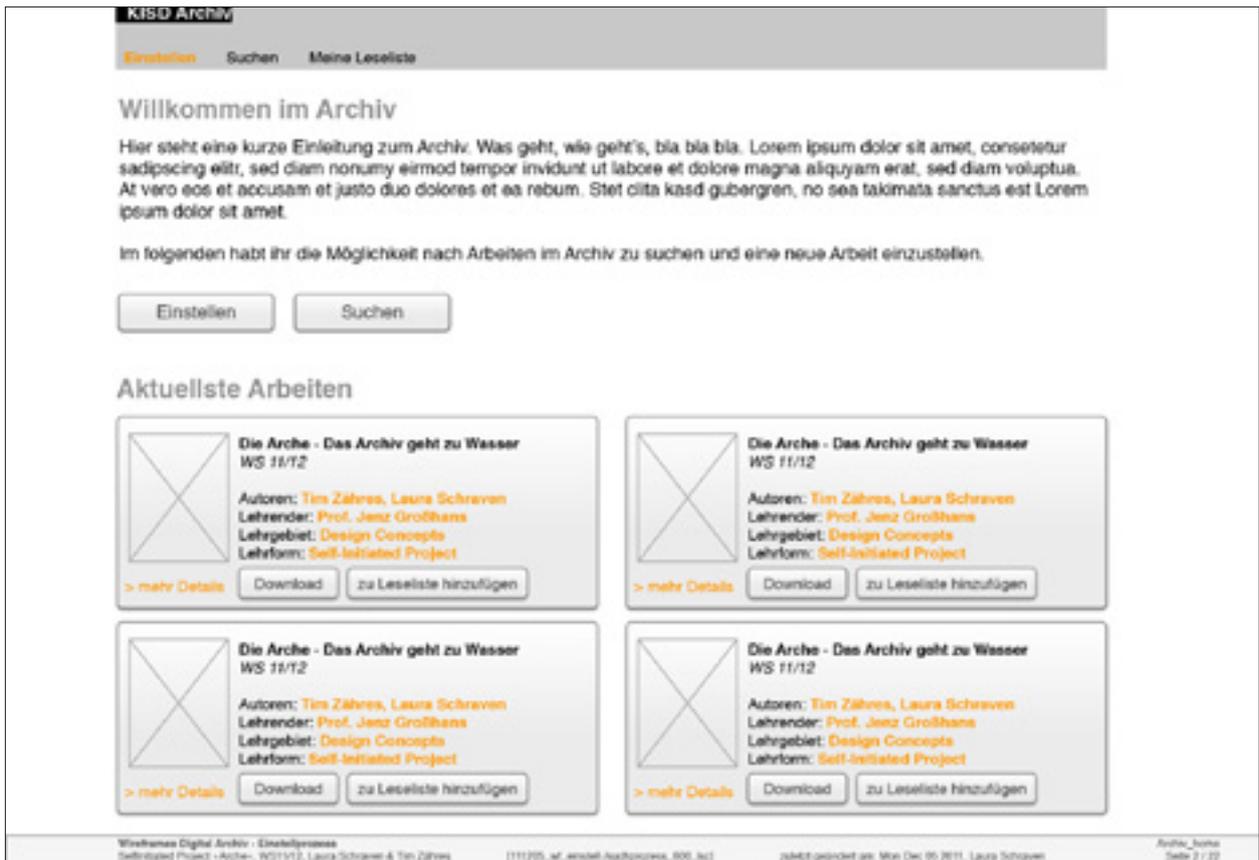


ABB.182 Wireframes Digital Archiv – Einstellprozess / Startseite / 5.12.2011 (S. 2/22)

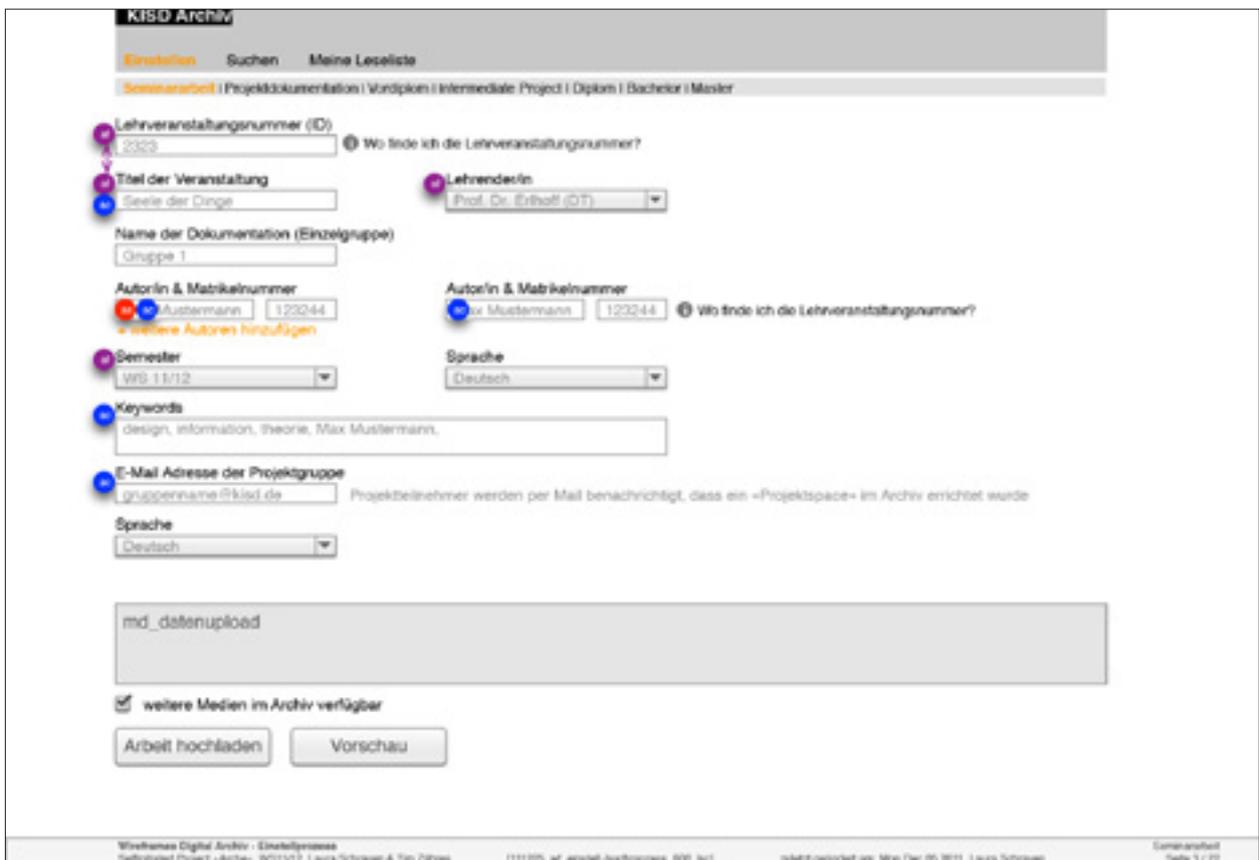


ABB.183 Wireframes Digital Archiv – Einstellprozess / Einstellmaske, Seminararbeit / 5.12.2011 (S. 3/22)

KISD Archiv
 Einstellen Suchen Meine Leseliste
 Seminararbeit | **Projektdokumentation** | Vordiplom | Intermediate Project | Diplom | Bachelor | Master

Lehrveranstaltungsnummer (ID) Wo finde ich die Lehrveranstaltungsnummer?

Titel der Veranstaltung Lehrende/in Projektart

Name der Dokumentation (Einzelgruppe)

Autorin & Matrikelnummer Wo finde ich die Lehrveranstaltungsnummer?
+ weitere Autoren hinzufügen

Semester Sprache

Keywords

E-Mail Adresse der Projektgruppe Projektteilnehmer werden per Mail benachrichtigt, dass ein -Projekt-space- im Archiv errichtet wurde

Sprache

weitere Medien im Archiv verfügbar

Wireframes Digital Archiv - Einzelprozess
 Definiertes Projekt - Archiv, WS11/12, Laura Schraier & Tim Zährer [111205, wt_arsinet_halfprocess_800_3c] juM&F gedruckt am: Mo 04 Dec 05 2011, Laura Schraier ProjektInfo Seite 4 / 22

ABB.184 Wireframes Digital Archiv – Einstellprozess / Einstellmaske, Projektdokumentation / 5.12.2011 (S. 4/22)

KISD Archiv
 Einstellen Suchen Meine Leseliste
 Seminararbeit | Projektdokumentation | **Vordiplom** | Intermediate Project | Diplom | Bachelor | Master

Hauptthema Nebenthema 1 Nebenthema 2

Titel der Arbeit Prüfende(r)

Autorin & Matrikelnummer weitere(r) Autorin nur bei HT möglich Studiengang

+ weitere Autoren hinzufügen

Abgabesemester

Keywords

Kurzbeschreibung

Sprache

Ich möchte das meine Arbeit vertraulich behandelt wird und nur das Abstract über die Suchfunktion im Archiv angezeigt wird Alle Prüfungs müssen dem zustimmen!

weitere Medien im Archiv verfügbar

Wireframes Digital Archiv - Einzelprozess
 Definiertes Projekt - Archiv, WS11/12, Laura Schraier & Tim Zährer [111205, wt_arsinet_halfprocess_800_3c] juM&F gedruckt am: Mo 04 Dec 05 2011, Laura Schraier Vordiplom Seite 1 / 22

ABB.185 Wireframes Digital Archiv – Einstellprozess / Einstellmaske, Vordiplom / 5.12.2011 (S. 5/22)

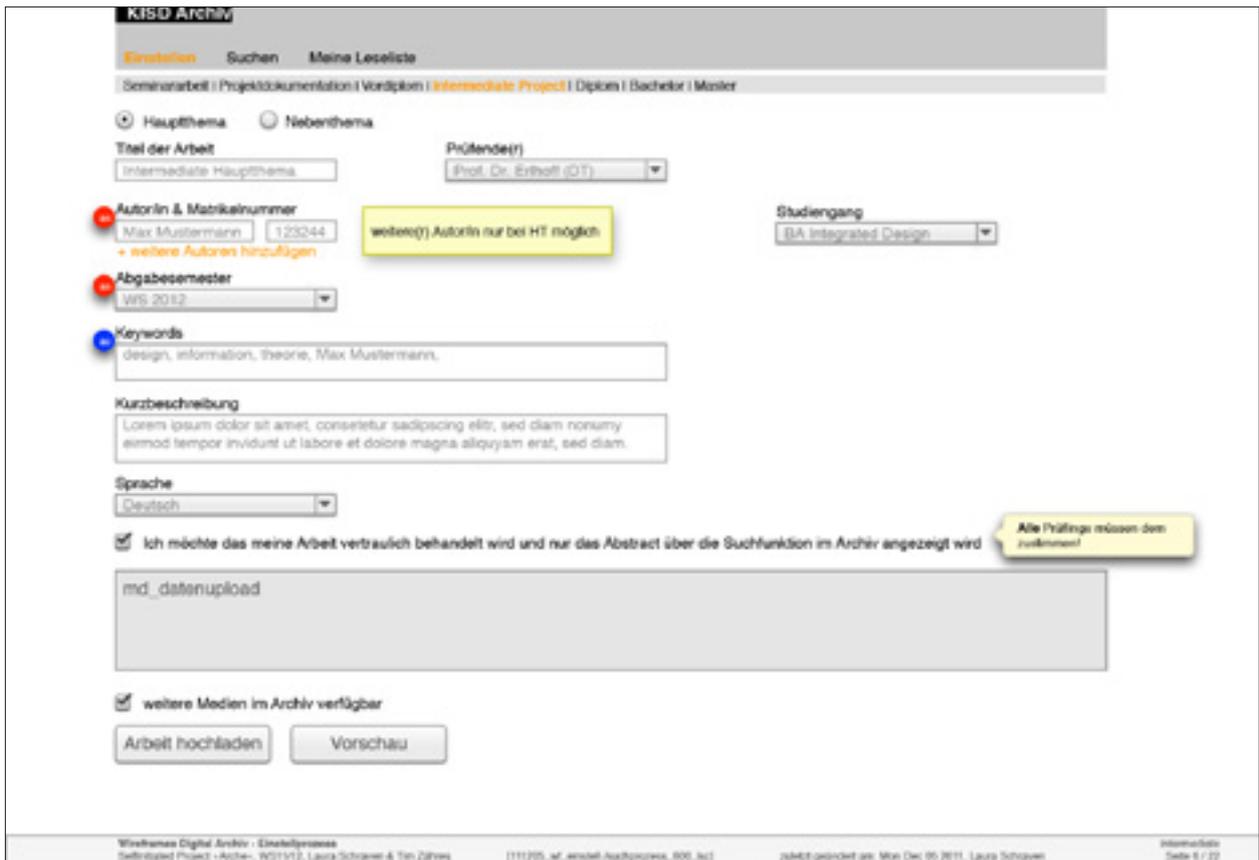


ABB.186 Wireframes Digital Archiv – Einstellprozess / Einstellmaske, Intermediate / 5.12.2011 (S. 6/22)

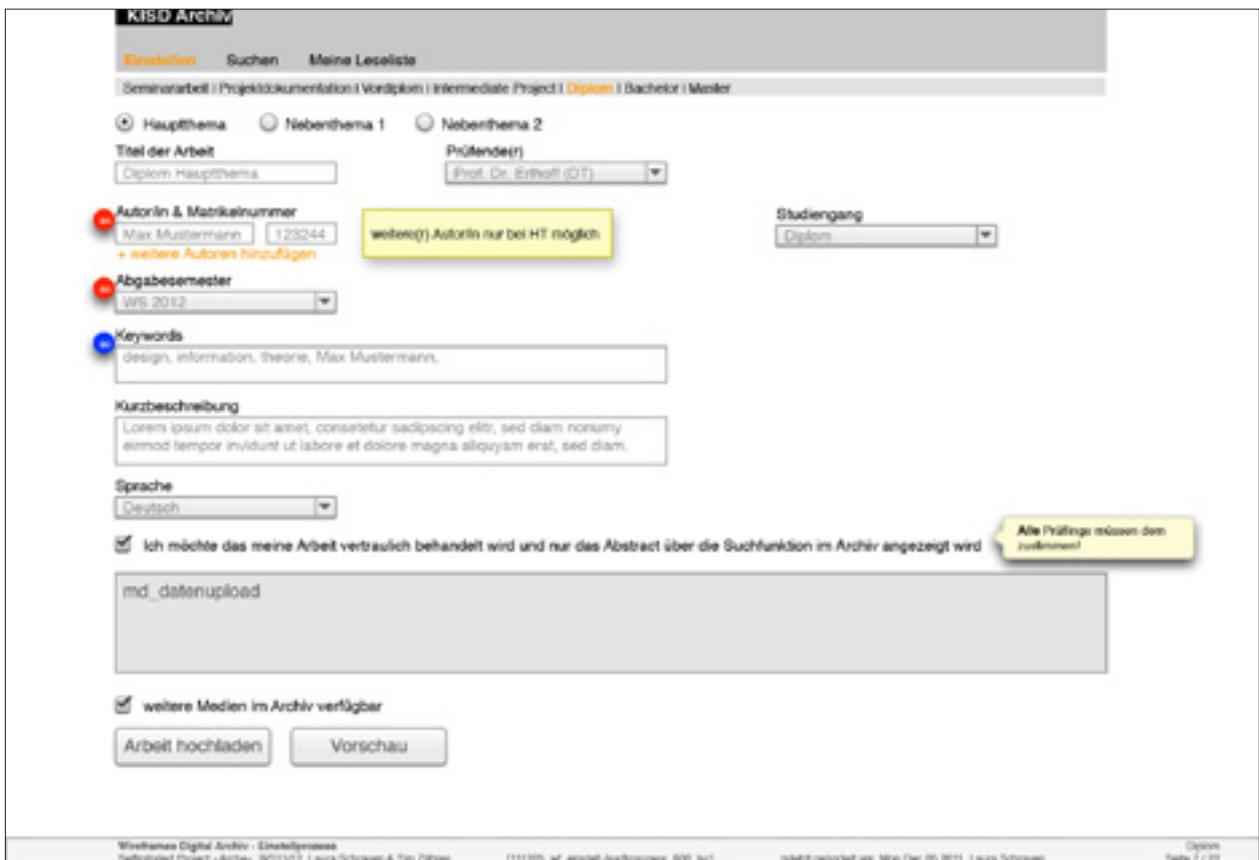


ABB.187 Wireframes Digital Archiv – Einstellprozess / Einstellmaske, Diplom / 5.12.2011 (S. 7/22)

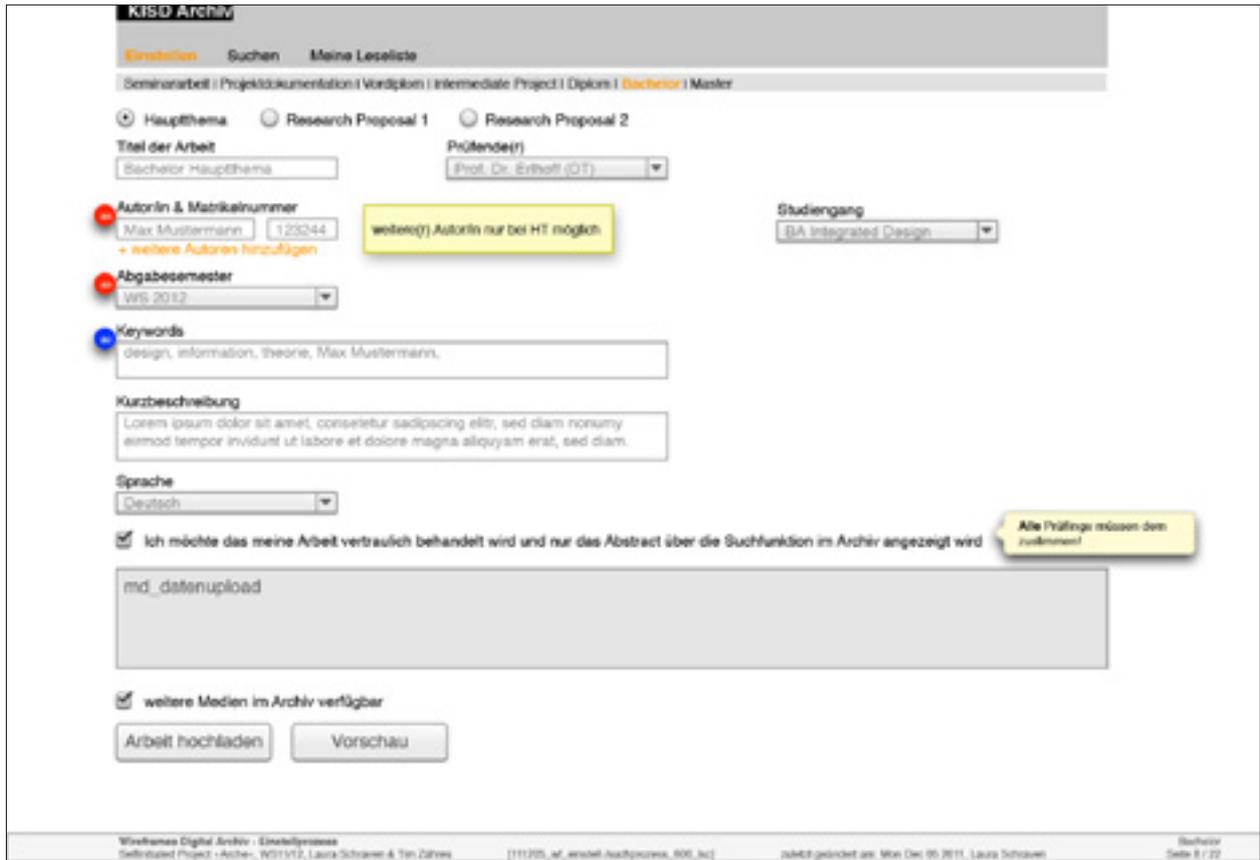


ABB.188 Wireframes Digital Archiv – Einstellprozess / Einstellmaske, Bachelor / 5.12.2011 (S. 8/22)

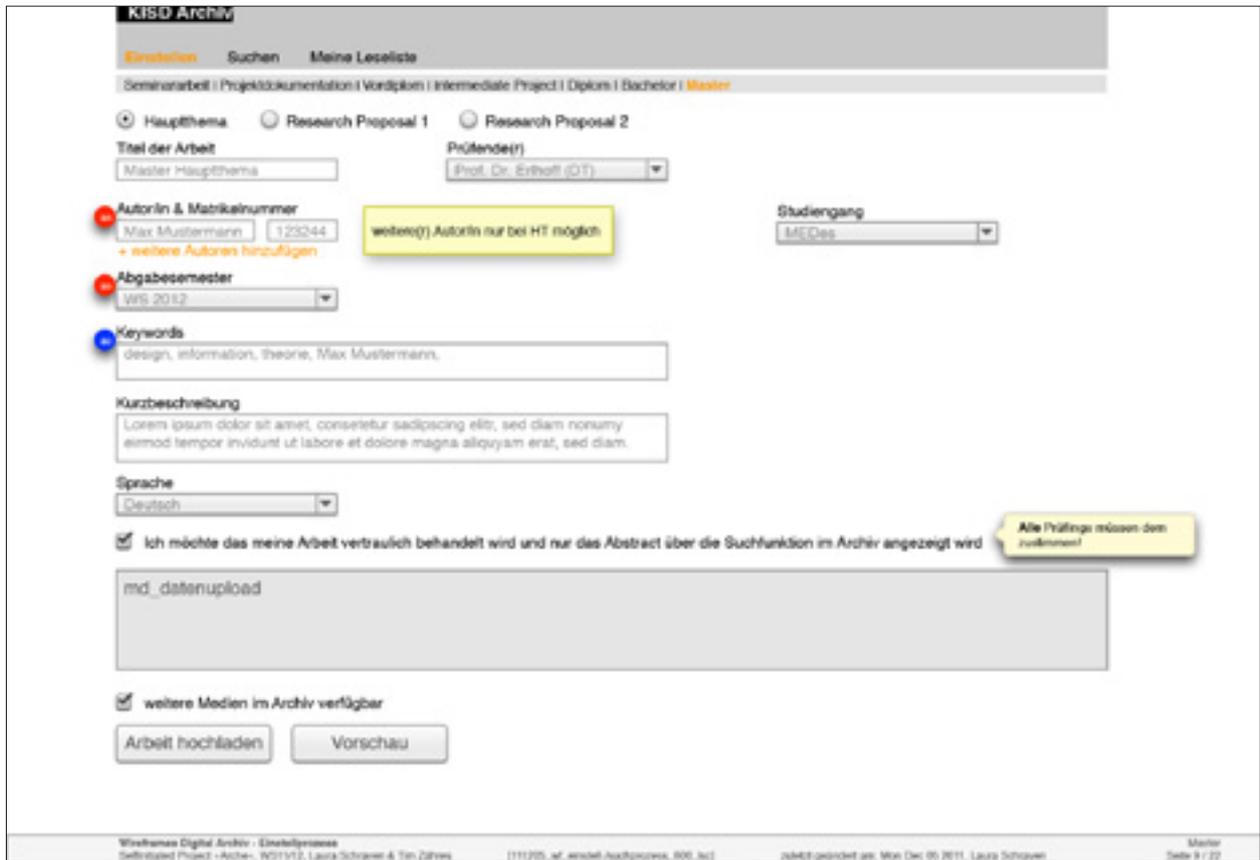


ABB.189 Wireframes Digital Archiv – Einstellprozess / Einstellmaske, Master / 5.12.2011 (S. 9/22)

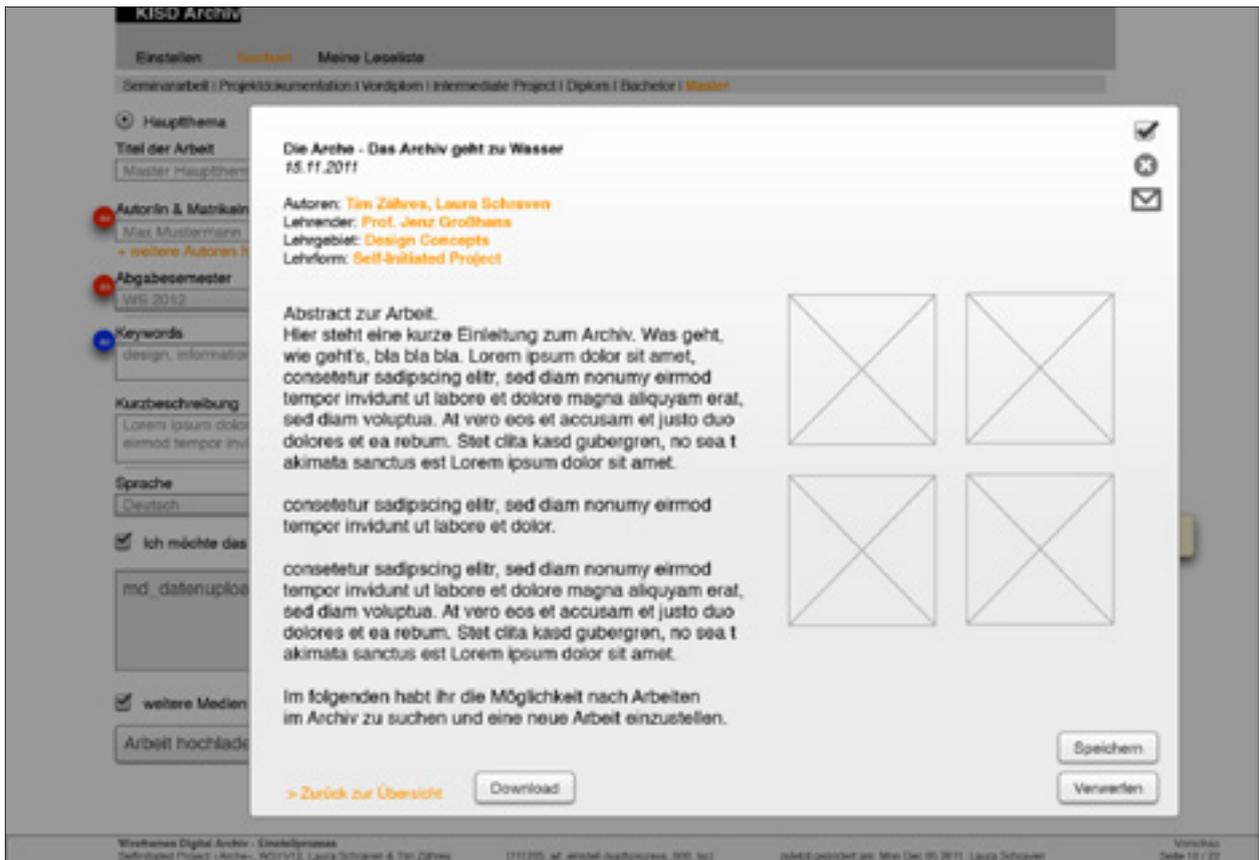


ABB.190 Wireframes Digital Archiv – Einstellprozess / Einstellmaske, Vorschau / 5.12.2011 (S. 10/22)

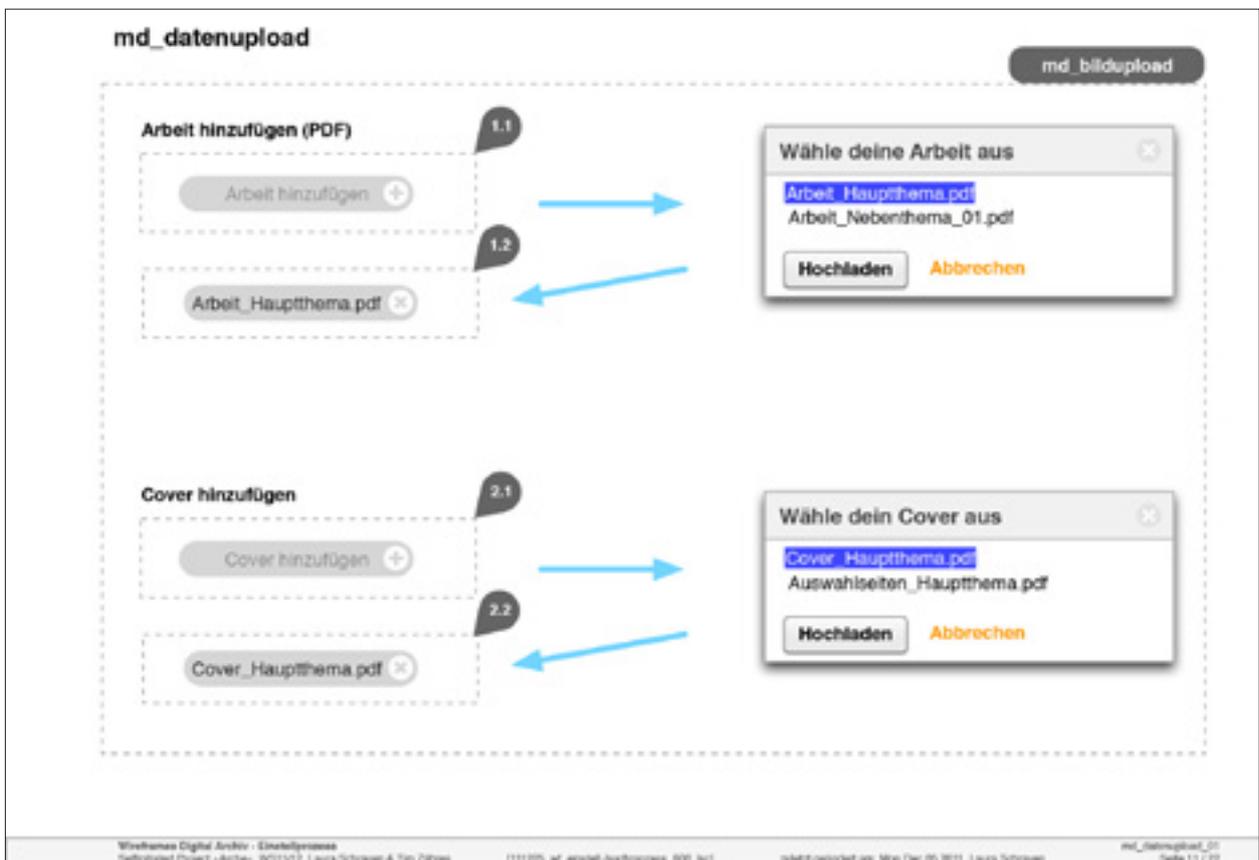


ABB.191 Wireframes Digital Archiv – Einstellprozess / Diagramm, Datenupload (1/2) / 5.12.2011 (S. 11/22)

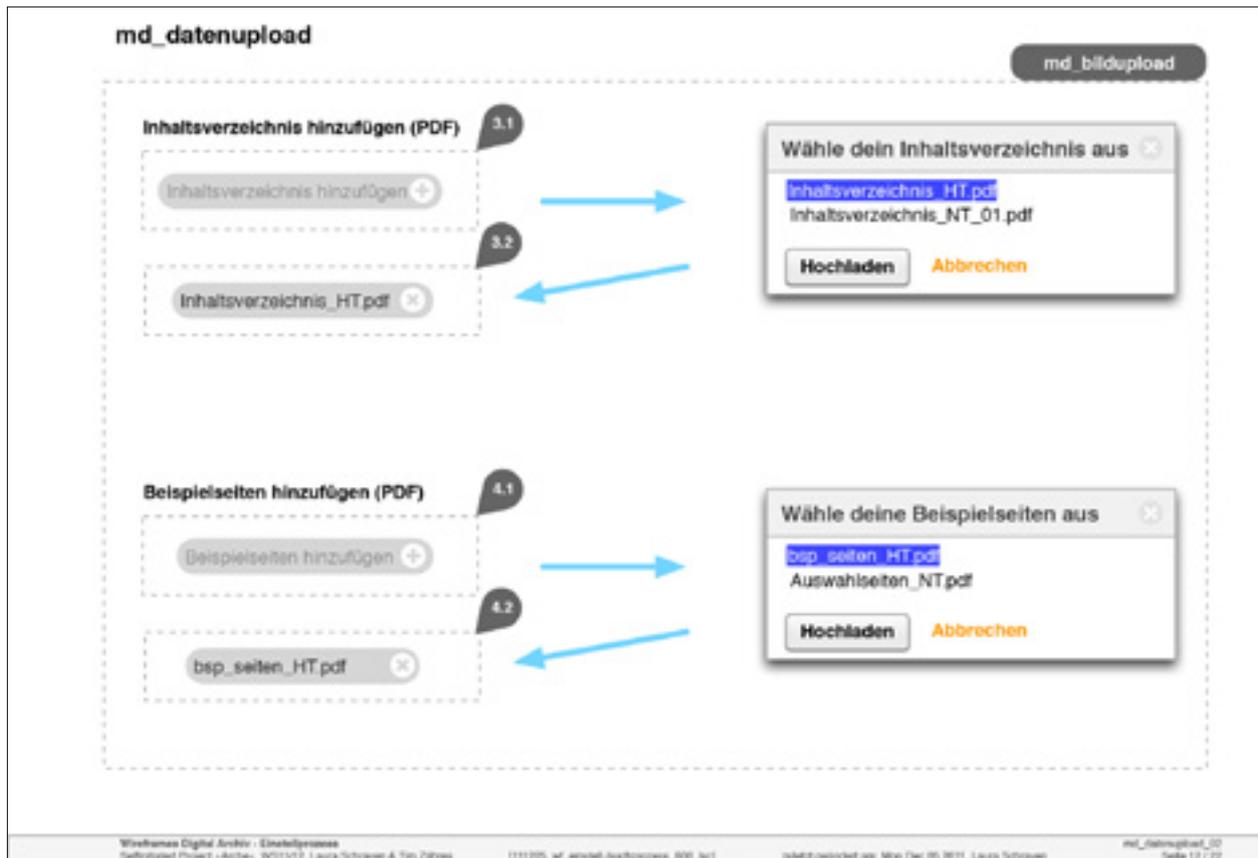


ABB.192 Wireframes Digital Archiv – Einstellprozess / Diagramm, Datenupload (2/2) / 5.12.2011 (S. 12/22)

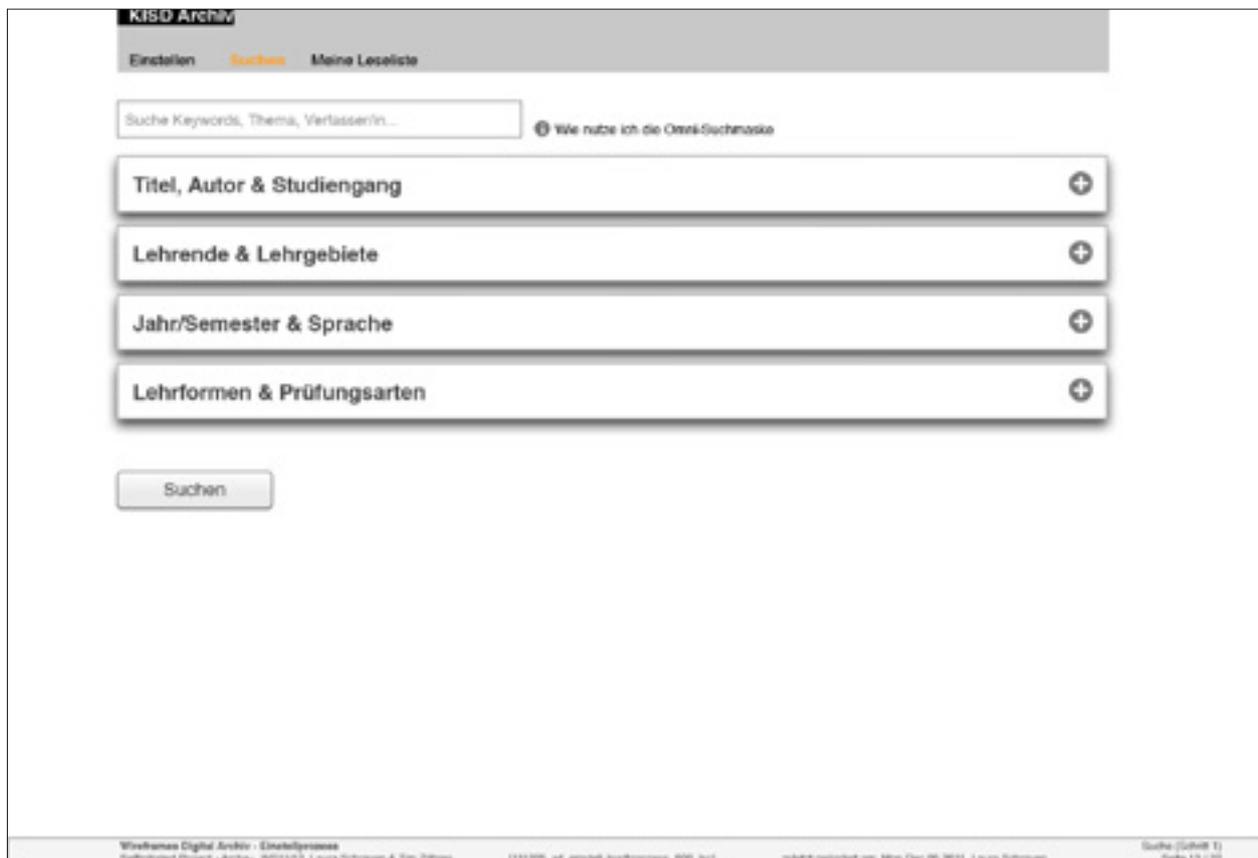


ABB.193 Wireframes Digital Archiv – Einstellprozess / Suche (Schritt 1) / 5.12.2011 (S. 13/22)

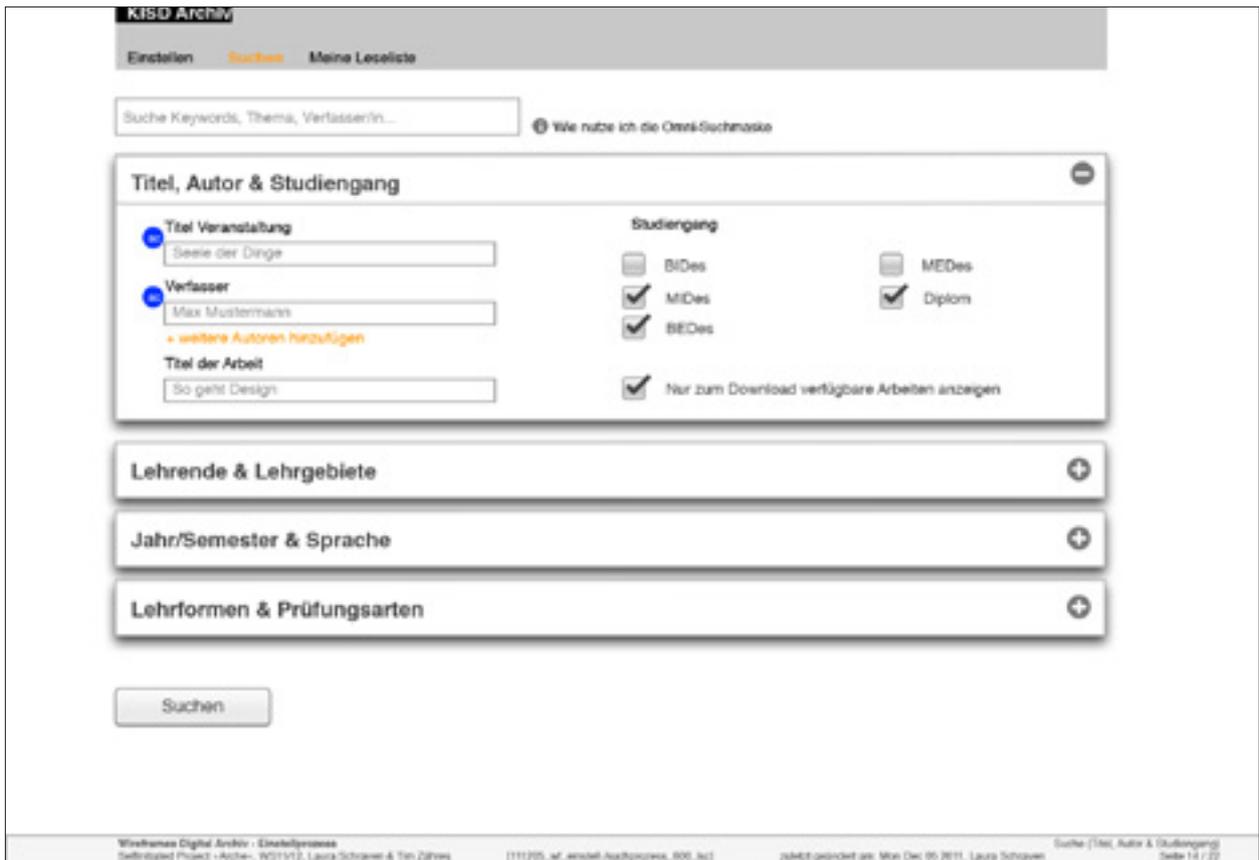


ABB.194 Wireframes Digital Archiv – Einstellprozess / Suche: Titel, Autor & Studiengang / 5.12.2011 (S. 14/22)

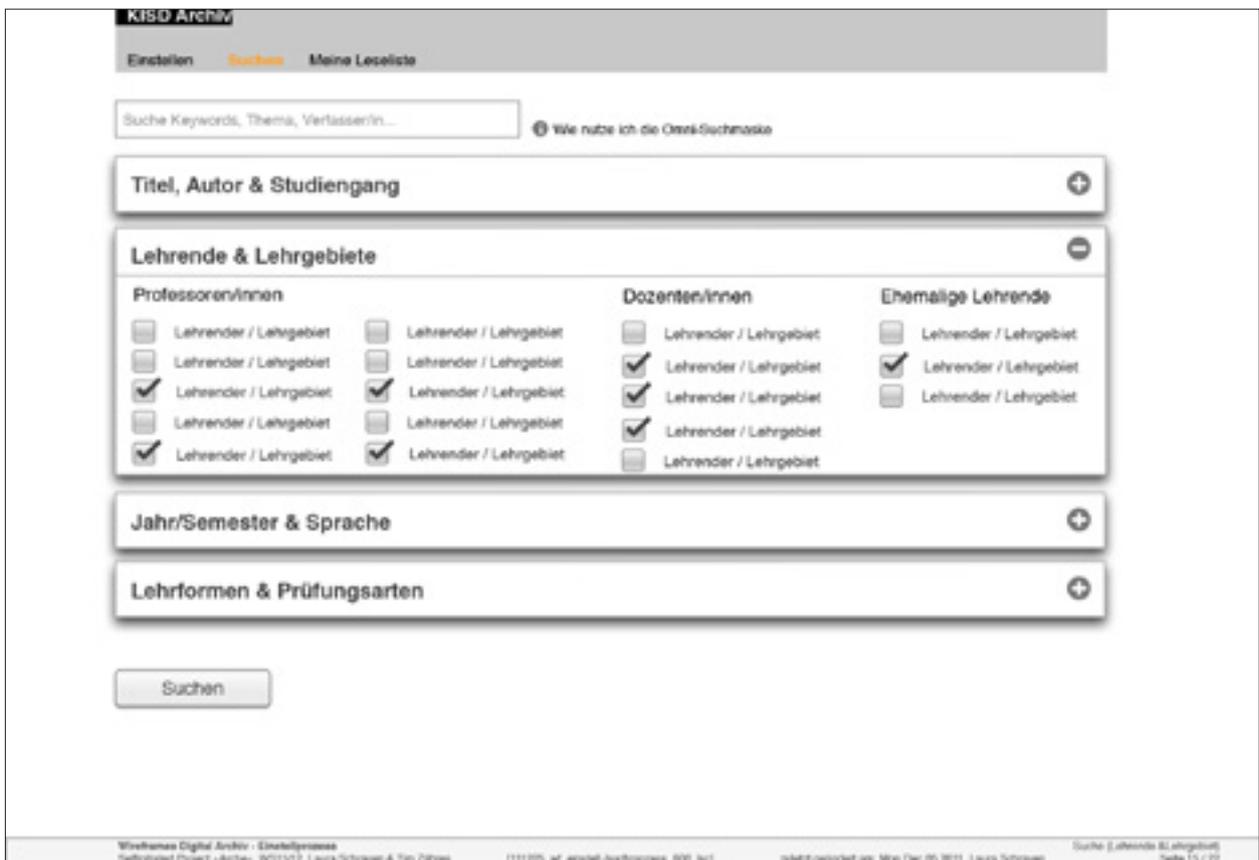


ABB.195 Wireframes Digital Archiv – Einstellprozess / Suche: Lehrende & Lehrgebiete / 5.12.2011 (S. 15/22)

KISD Archiv

Einstellen **Suchen** Meine Leseliste

Suche Keywords, Thema, Verfasser/n... [Wie nutze ich die Onlei-Suchmaske](#)

Titel, Autor & Studiengang +

Lehrende & Lehrgebiete +

Jahr/Semester & Sprache -

Jahr oder Semester

von bis

Sprache

Deutsch
 Englisch
 Spanisch

Lehrformen & Prüfungsarten +

Suchen

Wireframes Digital Archiv - Einstellprozess
 Definiertes Projekt - Archiv - WS11/12, Laura Schramm & Tim Zährner [111205, wt_arsinet-leadprocess, 800, 3x] jshM2 geändert am: Mon Dec 05 2011, Laura Schramm Suche (Jahr / Sprache) Seite 16 / 22

ABB.196 Wireframes Digital Archiv – Einstellprozess / Suche: Jahr & Sprache / 5.12.2011 (S. 16/22)

KISD Archiv

Einstellen **Suchen** Meine Leseliste

Suche Keywords, Thema, Verfasser/n... [Wie nutze ich die Onlei-Suchmaske](#)

Titel, Autor & Studiengang +

Lehrende & Lehrgebiete +

Jahr/Semester & Sprache +

Lehrformen & Prüfungsarten -

Seminardokumentationen

Seminararbeit

Projektdokumentationen

Projektdoku KP
 Projektdoku MP
 Projektdoku LP
 Projektdoku WP
 Projektdoku IP
 Projektdoku RS

Prüfungsarbeiten

Vordiplom HT
 Vordiplom NT
 Bachelor NT
 Intermediate HT
 Intermediate NT

Master RP
 Diplom HT
 Diplom NT

Suchen

Wireframes Digital Archiv - Einstellprozess
 Definiertes Projekt - Archiv - WS11/12, Laura Schramm & Tim Zährner [111205, wt_arsinet-leadprocess, 800, 3x] jshM2 geändert am: Mon Dec 05 2011, Laura Schramm Suche (Projektform) Seite 17 / 22

ABB.197 Wireframes Digital Archiv – Einstellprozess / Suche: Lehrformen & Prüfungsarten / 5.12.2011 (S. 17/22)

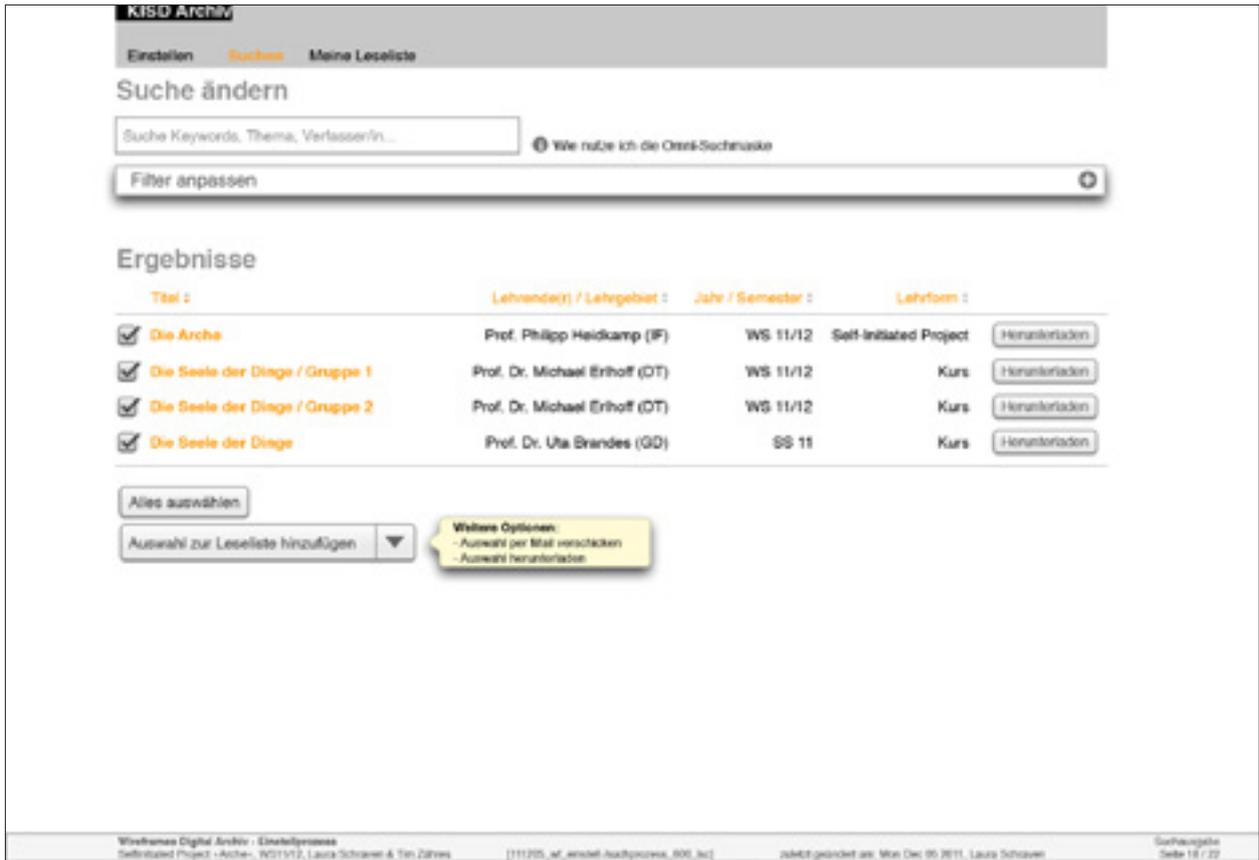


ABB.198 Wireframes Digital Archiv – Einstellprozess / Suchausgabe / 5.12.2011 (S. 18/22)

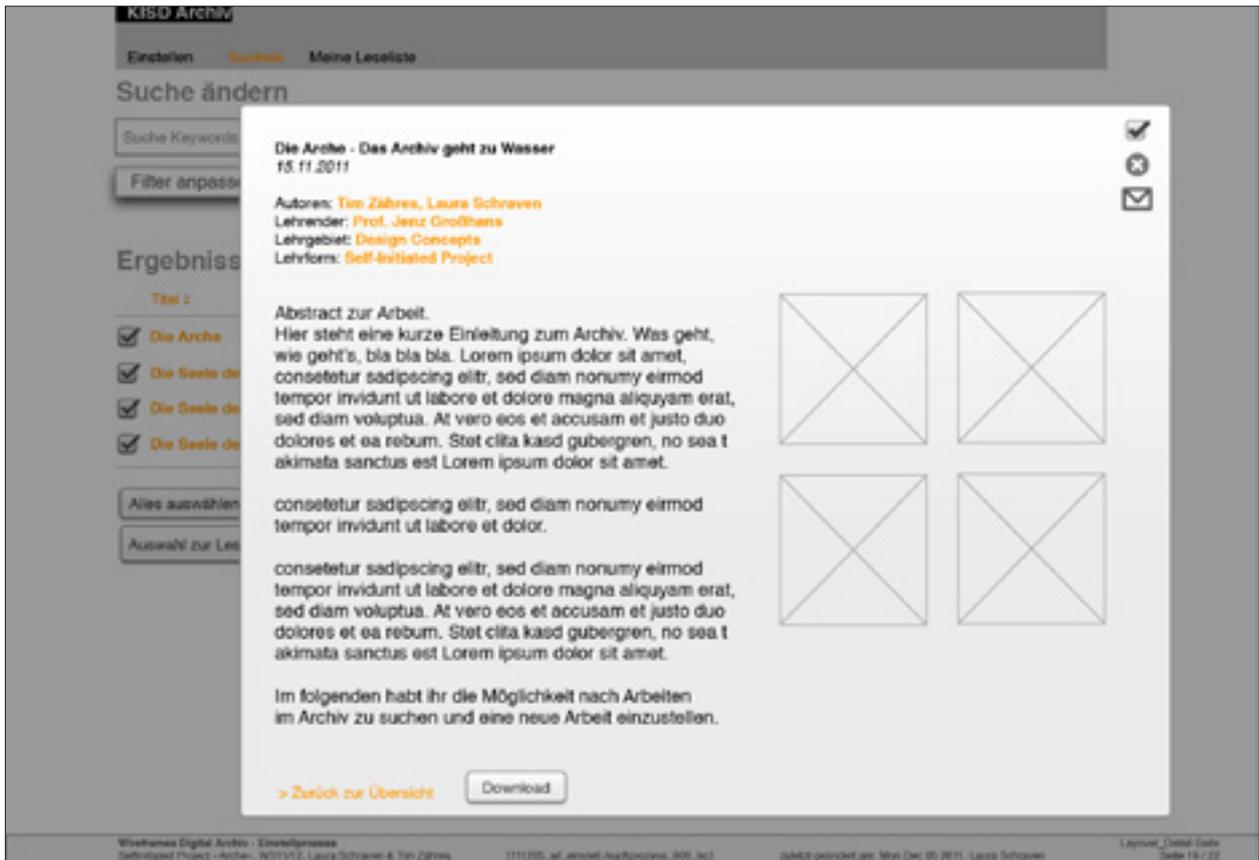


ABB.199 Wireframes Digital Archiv – Einstellprozess / Layover_Detail-Seite / 5.12.2011 (S. 19/22)

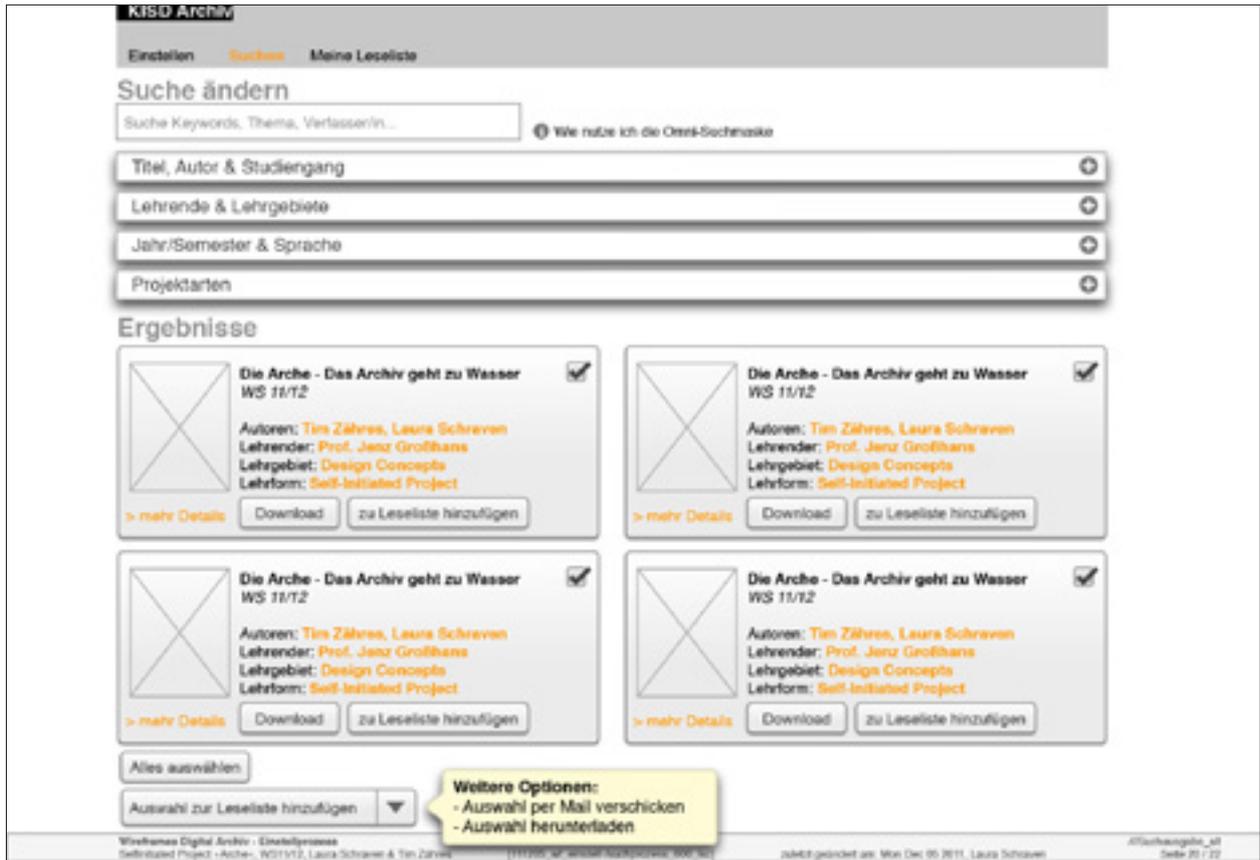


ABB.200 Wireframes Digital Archiv – Einstellprozess / Suchausgabe_alt / 5.12.2011 (S. 20/22)

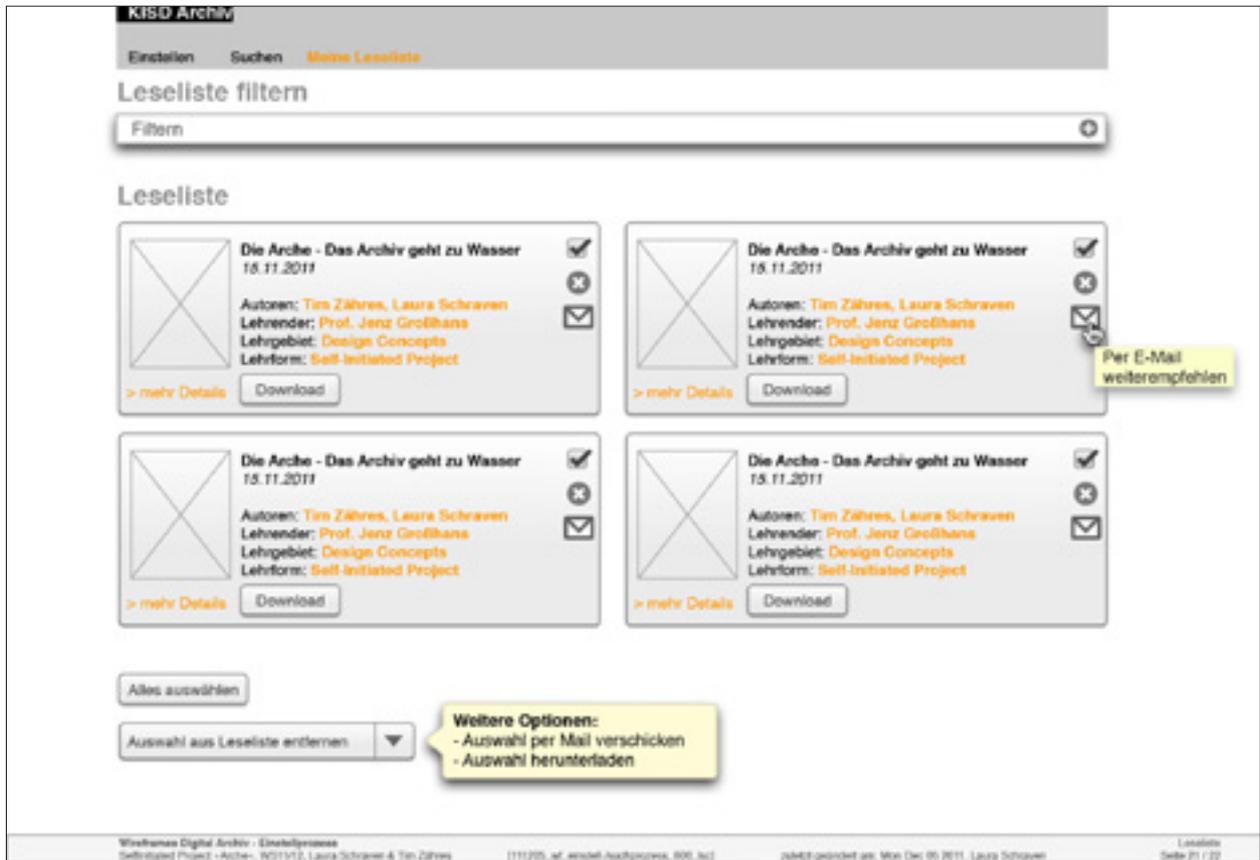


ABB.201 Wireframes Digital Archiv – Einstellprozess / Leseliste / 5.12.2011 (S. 21/22)



ABB. 202 Wireframes Digital Archiv – Einstellprozess / Leseliste Lightbox Gallery / 5.12.2011 (S. 22/22)

2 Diagramme und Modelle

2.0 Einleitung

Diese Sektion umfasst diverse Diagramme mit direktem Bezug zur Arbeit.

2.1 UML-Diagramme

Diese UML-Diagramme wurden erstellt, um zu evaluieren, ob eine diagrammatische Darstellung bestimmter Abläufe dem Leser einen besseren Überblick über komplexe Abläufe ermöglichen würde. Letztendlich entschied ich mich für eine Beschreibung dieser Maske in Form von Screenshots und Text. Siehe **Arbeit einreichen** (S. 98).

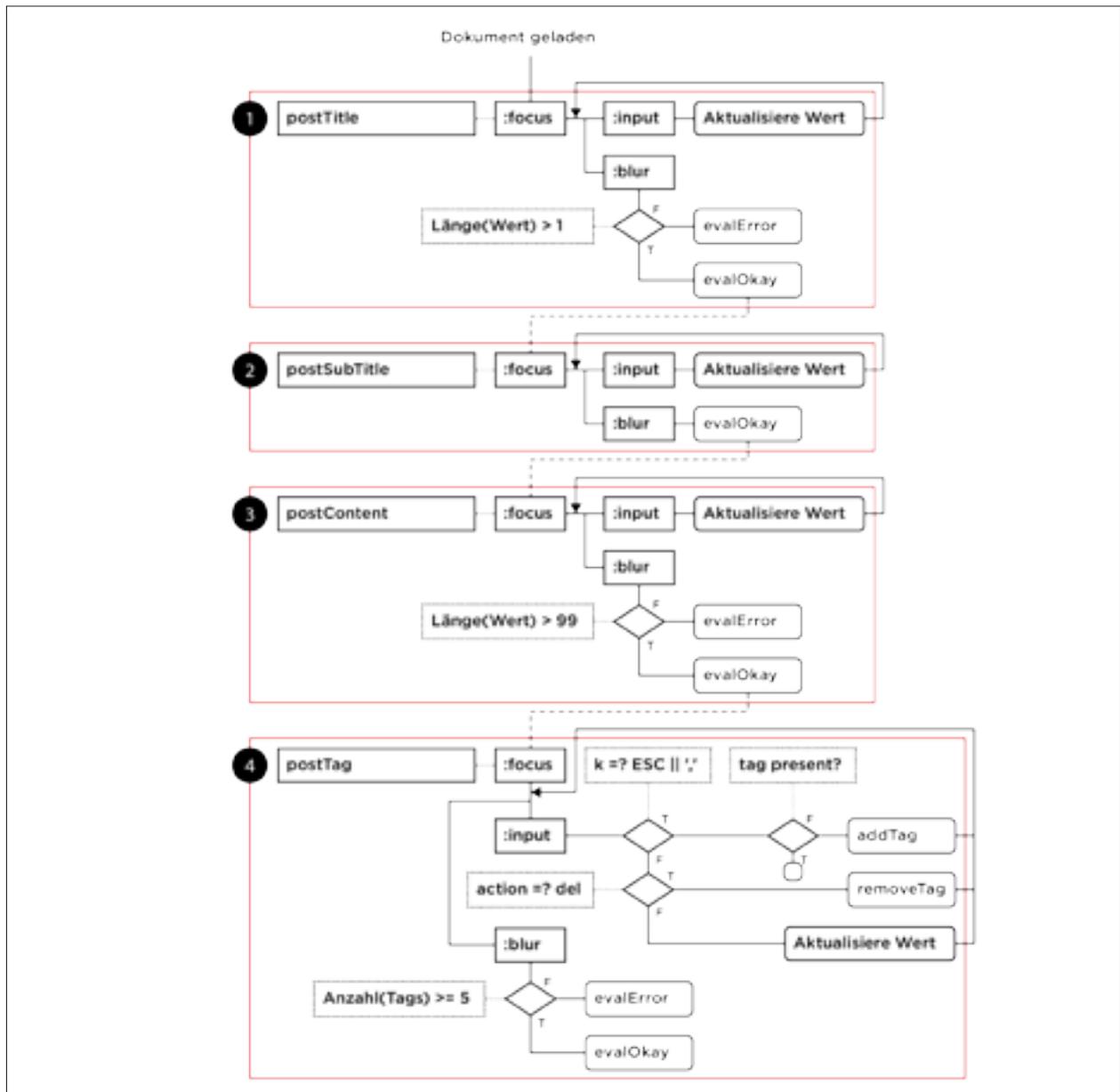


ABB. 203 Partielles Ablaufdiagramm »Ausfüllen einreichen Maske« (Felder 1-4)

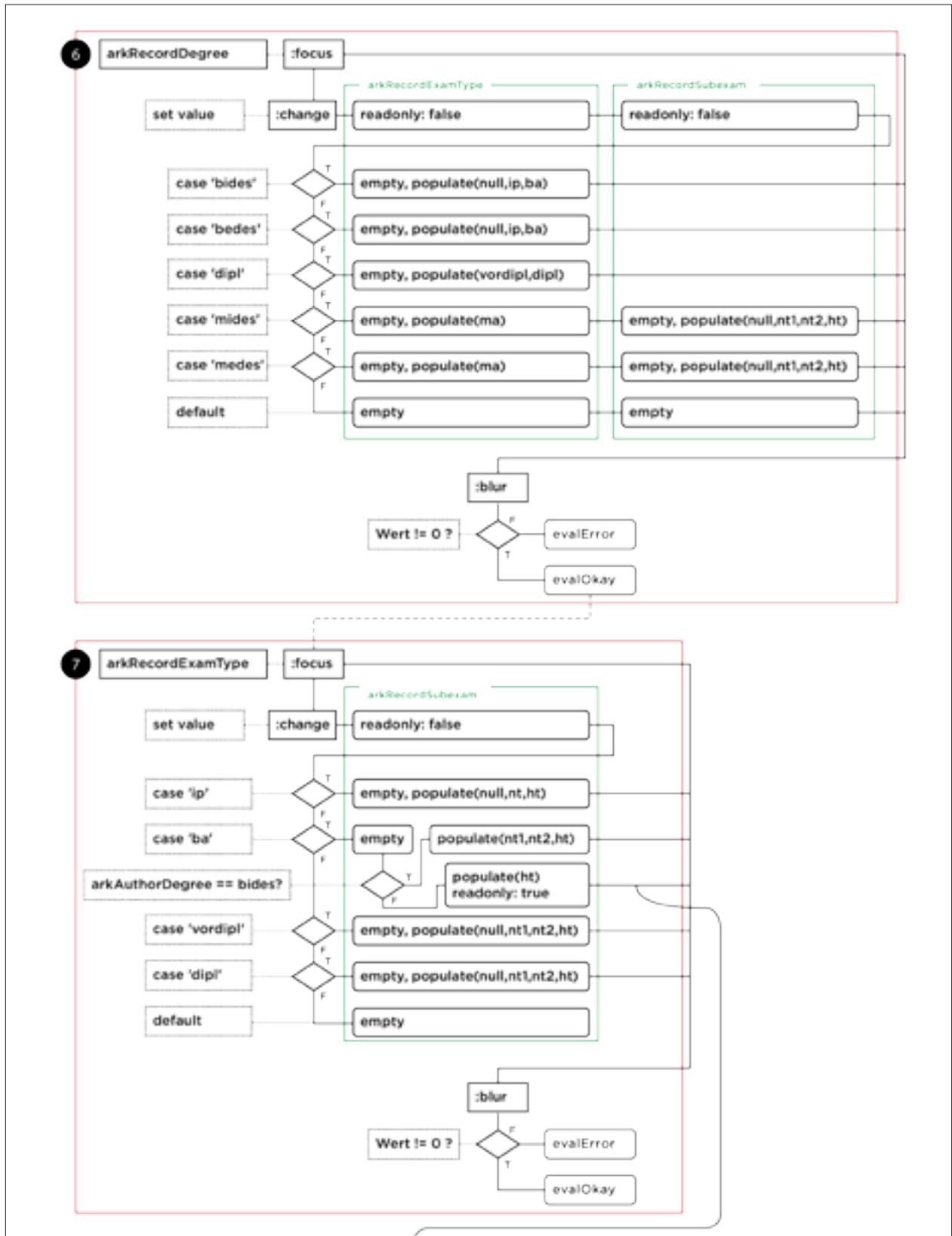


ABB. 204 Partielles Ablaufdiagramm »Arbeit einreichen Maske« (Felder 6-7)

2.2 Datenbankmodelle

2.2.1 WordPress

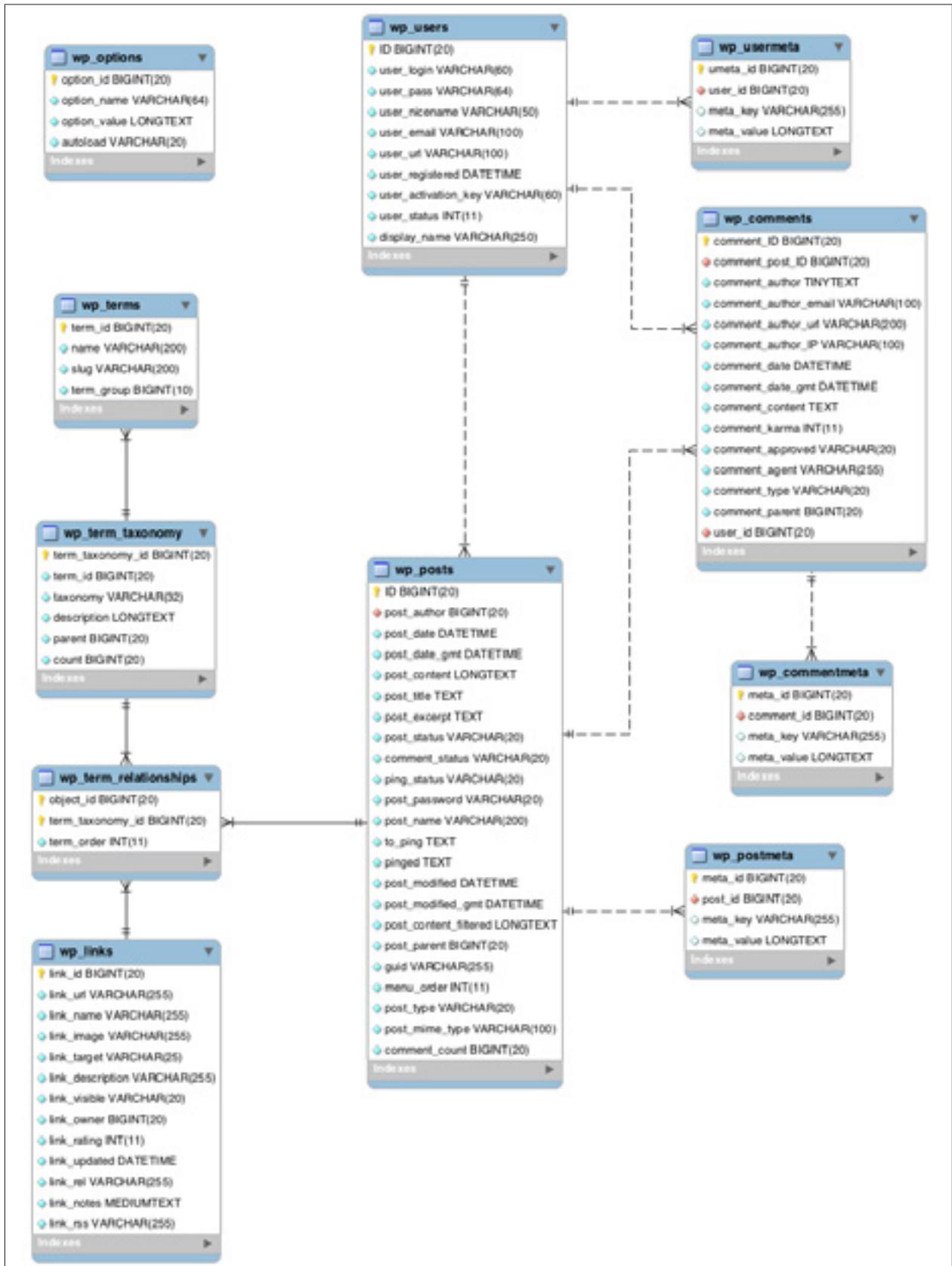


ABB. 205 WordPress-Datenbankmodell (Quelle: http://codex.wordpress.org/Database_Description/).

2.3 Integritätsdiagramme

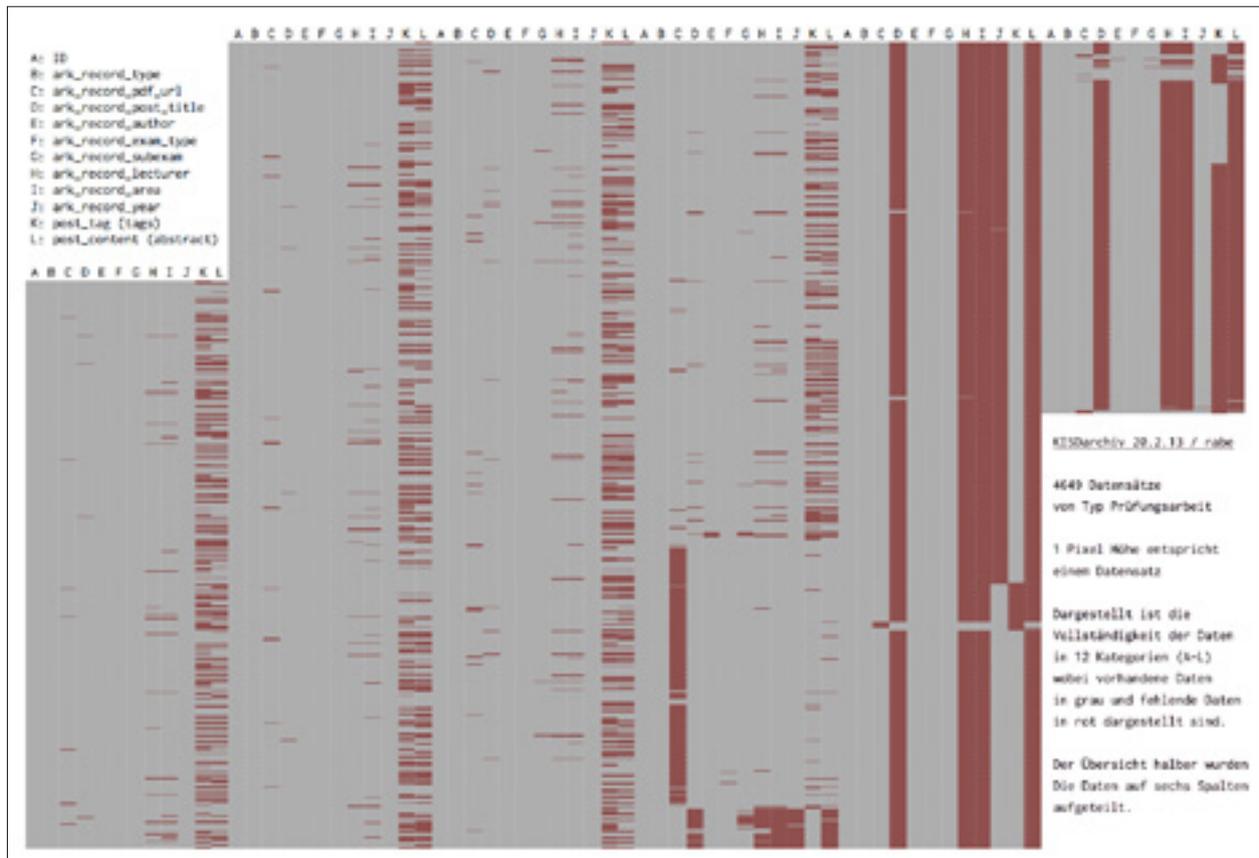
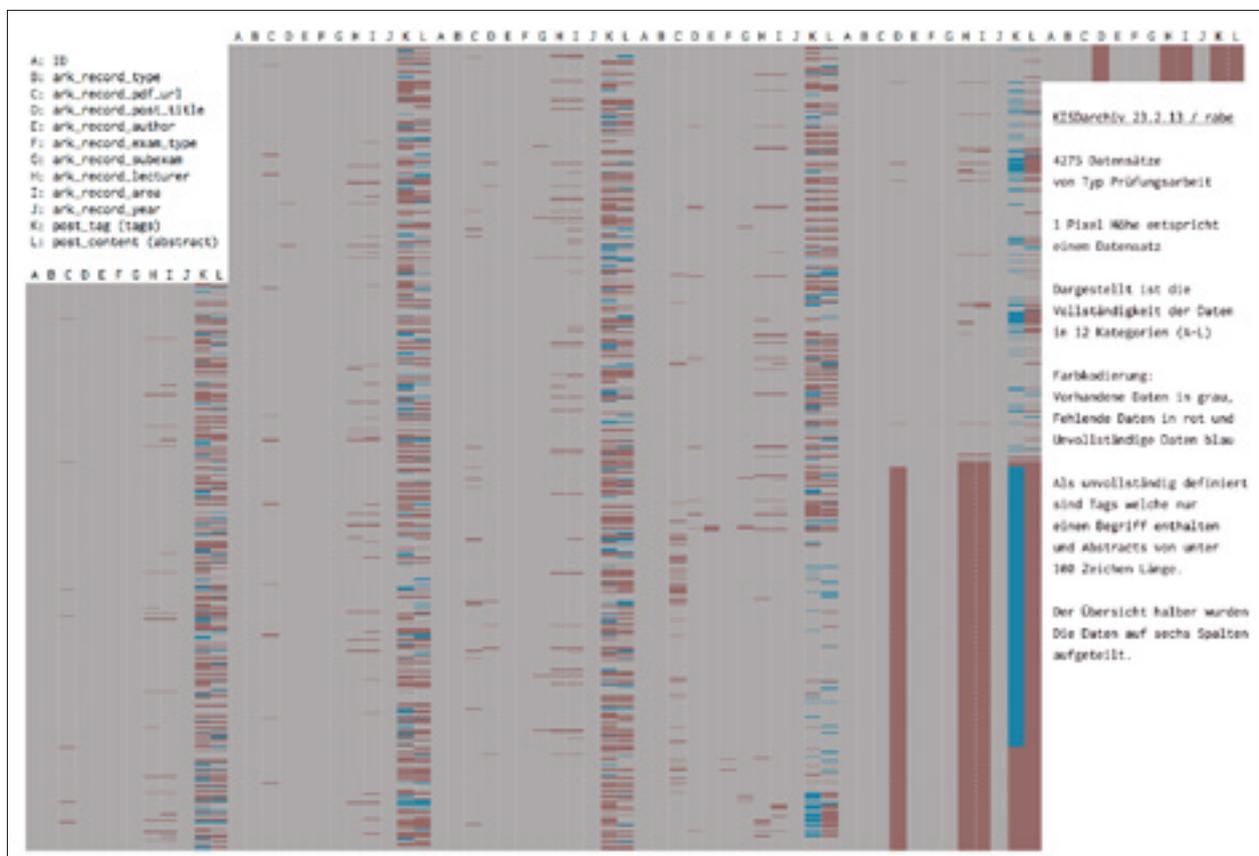


ABB. 207 Integritätsdiagramm 20.2.2013 (oben)

ABB. 208 Integritätsdiagramm 23.2.2013 (unten)



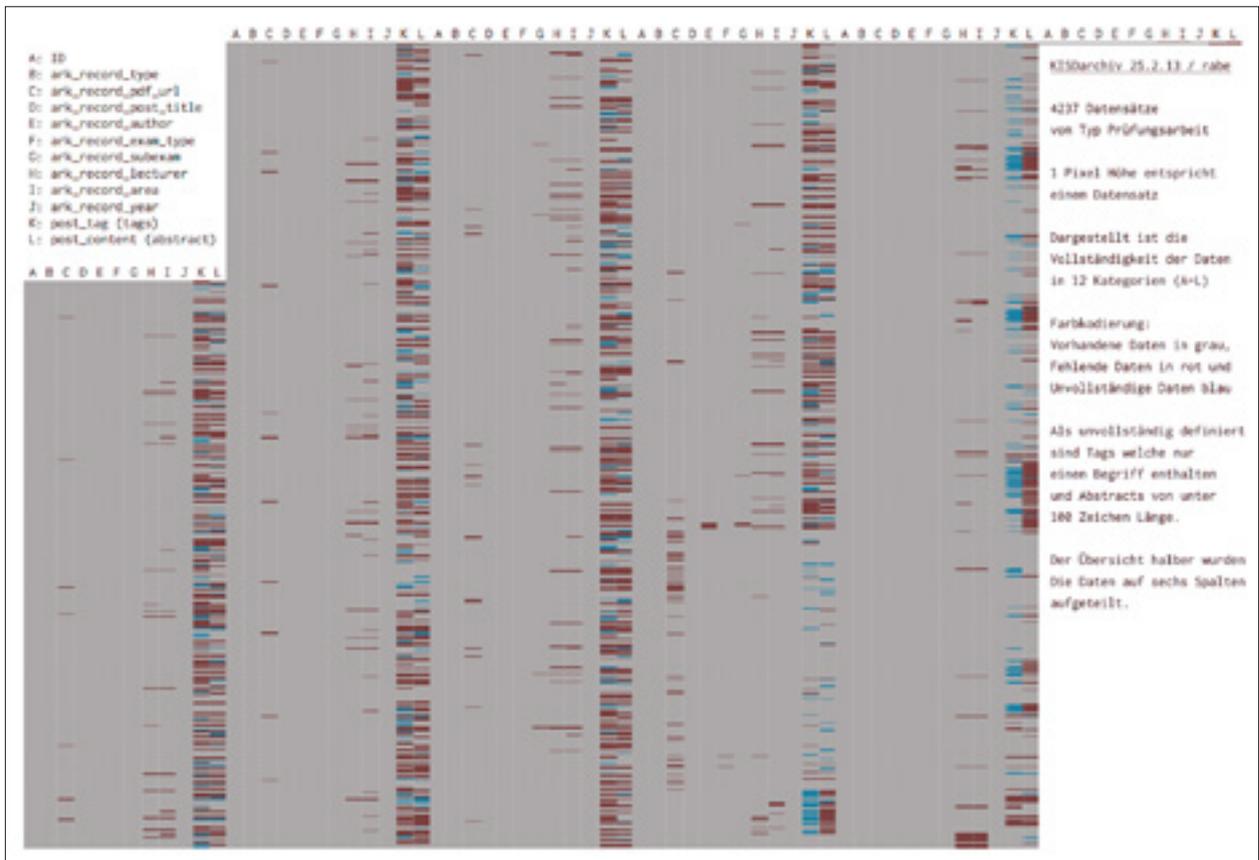
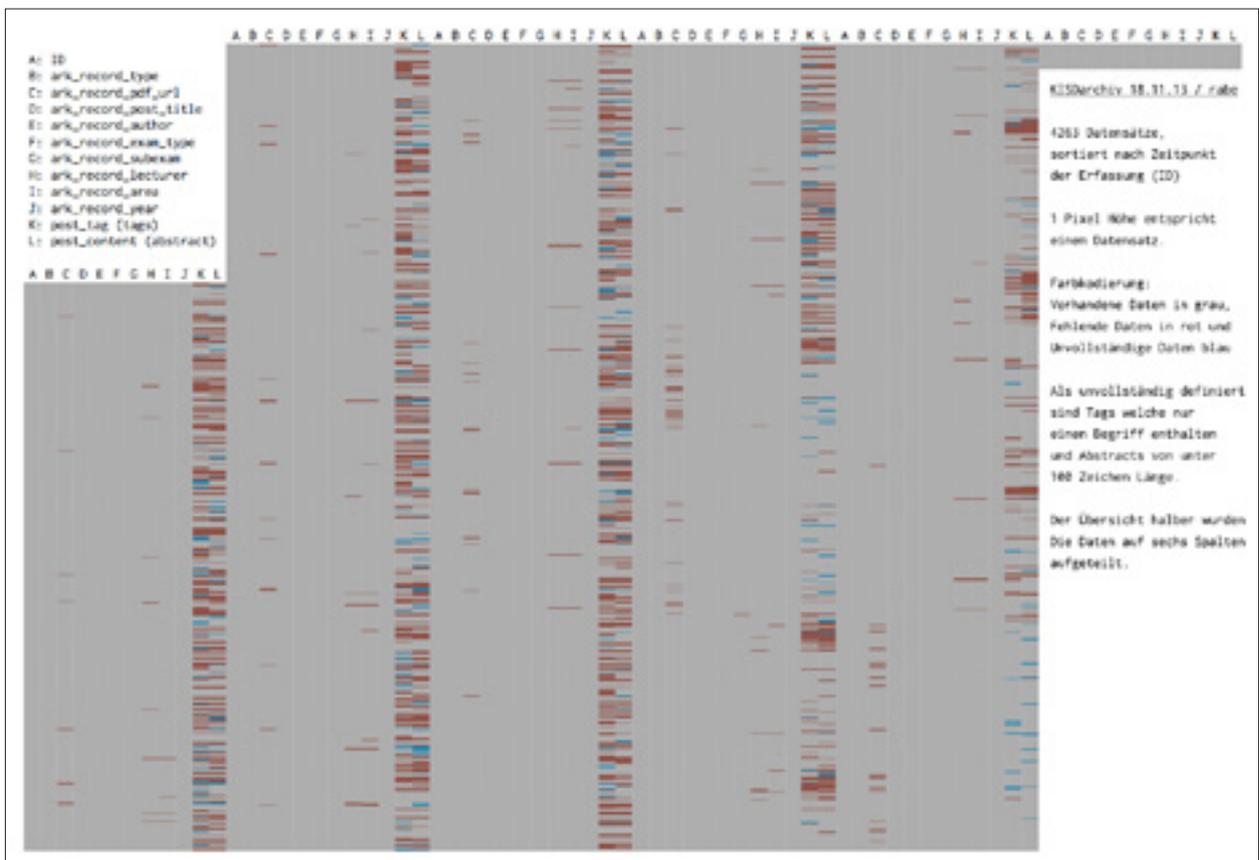


ABB. 209 Integritätsdiagramm 25.2.2013 (oben)

ABB. 210 Integritätsdiagramm 18.11.2013 (unten)



3 Code

3.0 Einleitung

Bevor ich begann, diese Arbeit zu schreiben, hatte ich eine Idee: Ich wollte den kompletten Code des KISDarchivs hier abdrucken. Die haptische Erfahrungsmachung des Projekts durch seine Überführung in Papierform.

Eine Überschlagung ergab die Summe von 15000 Zeilen eigens für das KISDarchiv geschriebenem Code (50% davon PHP, je 25% JavaScript und CSS). Gesetzt auf einer Größe, die diesem Text entspräche, würde der Code damit 300 Seiten einnehmen. Die Idee des Abdrucks erschien angesichts des bereits ohne Code-Abbild beachtlichen Umfangs der Arbeit den Rahmen zu sprengen und wurde verworfen.

4 Daten

4.0 Einleitung

Ursprünglich war ebenfalls geplant, an dieser Stelle ein komplettes Datenbank-Abbild zu drucken. Dies hätte jedoch selbst bei einer Schriftgröße von 6 Punkt ohne jegliche Formatierung etwa 968 Seiten eingenommen.

Überdies war angedacht, hier diverse Datenbank-Queries abzudrucken, die im Wartungsbetrieb genutzt wurden. Abfragen »von der Stange« anzubieten, schien jedoch der Datenintegrität nicht gerade zuträglich. Stattdessen sollen an dieser Stelle einige aktuelle (Februar 2014) Zahlen genannt werden:

TABELLE	ZEILEN	BESCHREIBUNG
WP_1186_COMMENTMETA	0	
WP_1186_COMMENTS	26	Kommentare (Bugs-Seite)
WP_1186_LINKS	0	
WP_1186_OPTIONS	318	Optionstabelle
WP_1186_POSTS	5820	Posttabelle (Seiten, Einträge aller Posttypen, Anhänge,...)
WP_1186_POSTMETA	159044	Metadaten von Posts und Plugins
WP_1186_RELEVANSSI	171561	Suche / Begriffsindex
WP_1186_RELEVANSSI_CACHE	0	
WP_1186_RELEVANSSI_EXCERPT	0	
WP_1186_RELEVANSSI_LOG	1472	Suche / Log vergangener Suchbegriffe
WP_1186_RELEVANSSI_STOPWORDS	1337	Suche / Stopwords
WP_1186_STUDENTEN2	598	Externe Studierenden-Metadatenquelle
WP_1186_TERMS	9074	Kategorien, Schlagwörter und sonstige Taxonomien
WP_1186_TERM_RELATIONSHIPS	19310	Zuordnung von Taxonomien zu Posts
WP_1186_TERM_TAXONOMY	9074	Taxonomie-Beschreibungen (Kategorie, Tag,...) der wp_terms

5 Quellen

5.0 Einleitung

Verständlicherweise bestehen die Quellen aus selbstgeschriebenem Material. Dies umfasst den Entwicklungsspace und eigenproduziertes Videomaterial.

5.1 spaces.kisd.de/NeDA

Der Entwicklungsspace, welcher Anfang Februar 2013 geschaffen wurde, um die Arbeit am KISDarchiv zu dokumentieren, bildet gewissermaßen die Grundlage für diese Arbeit. Dort sind alle Entwicklungsstufen des Projekts beschrieben. Der Space wird mit Abgabe dieser Arbeit auf KISD-öffentlich geschaltet werden, um Interessierten einen tieferen Einblick in die Entwicklung des KISDarchivs zu gewähren.

7.2.2013	Daten, Daten, Daten
HTTP://SPACES.KISD.DE/NEDA/	2013/02/07/daten-daten-daten/
8.2.2013	Daten-Import nach WordPress / Suchfunktion
HTTP://SPACES.KISD.DE/NEDA/	2013/02/08/daten-import-nach-wordpress/
9.2.2013	Datenerfassung
HTTP://SPACES.KISD.DE/NEDA/	2013/02/09/datenerfassung/
10.2.2013	Notizen des Herrn K: a) Dateibenennung b) Notes_offline
HTTP://SPACES.KISD.DE/NEDA/	2013/02/10/notizen-des-herrn-k-a-dateibenennung-b-notes_offline/
11.2.2013	Backend: Eingabemasken, Datenverwaltung
HTTP://SPACES.KISD.DE/NEDA/	2013/02/11/backend-eingabemasken-datenverwaltung/
12.2.2013	Eingabemaske (ark_record)
HTTP://SPACES.KISD.DE/NEDA/	2013/02/12/eingabemaske-ark_record/
12.2.2013	Prüfungsformulare 2012-10-17
HTTP://SPACES.KISD.DE/NEDA/	2013/02/12/pruefungsformulare-2012-10-17/
13.2.2013	Anm. zu a) Erfassung analoger Bestand, b) Spezialfall RS
HTTP://SPACES.KISD.DE/NEDA/	2013/02/13/anm-zu-erfassung-analoger-bestand/
17.2.2013	NOTES-offline: Originaldaten / Bearbeitete
HTTP://SPACES.KISD.DE/NEDA/	2013/02/17/notes-offline-originaldaten/
19.2.2013	Hallo Prof. Muxel!
HTTP://SPACES.KISD.DE/NEDA/	2013/02/19/hallo-prof-muxel/
26.2.2013	Workflows: Abgabe / Einpflegen / Aktualisierung »FAQ Archiv«
HTTP://SPACES.KISD.DE/NEDA/	2013/02/26/aktualisierung-faq-archiv-handbuch-der-archiv-ag/

27.2.2013	Datenabgleich, -import & -konsistenz
HTTP://SPACES.KISD.DE/NEDA/2013/02/27/doku-archiv-abgleich/	
27.2.2013	Tags & Suche
HTTP://SPACES.KISD.DE/NEDA/2013/02/27/tags-suche/	
27.2.2013	Prüfungsarbeiten CLOSED: 1. Nicht bestanden 2. Widerspruch (anlg./digit.)
HTTP://SPACES.KISD.DE/NEDA/2013/02/27/pruefungsarbeiten-closed-1-nicht-bestanden/	
2.3.2013	NeDA-bezogene Aufgaben für Archiv-AG SoSe 2013 (2013-1)
HTTP://SPACES.KISD.DE/NEDA/2013/03/02/neda-bezogene-aufgaben-fur-archiv-ag-sose-2013/	
3.3.2013	Frontend
HTTP://SPACES.KISD.DE/NEDA/2013/03/03/frontend/	
10.3.2013	Erfassung Studiengang
HTTP://SPACES.KISD.DE/NEDA/2013/03/10/erfassung-studiengang/	
13.3.2013	Willkommen Prof. Wrede!
HTTP://SPACES.KISD.DE/NEDA/2013/03/13/willkommen-prof-wrede/	
1.4.2013	Frontend: Suche/Lesezeichen/Erw. Suche/Tagcloud/Interakt. Headline
HTTP://SPACES.KISD.DE/NEDA/2013/04/01/frontend-suchelesezeichenerweiterte-suche/	
8.4.2013	Kurze Frage zu Abgabe 2014
HTTP://SPACES.KISD.DE/NEDA/2013/04/08/kurze-frage-zu-abgabe-2014/	
27.4.2013	Kurze Frage zu »Notes offline«
HTTP://SPACES.KISD.DE/NEDA/2013/04/27/kurze-frage-zu-notes-offline/	
16.7.2013	Unterscheidung Masterarbeiten MED/MID?
HTTP://SPACES.KISD.DE/NEDA/2013/07/16/unterscheidung-masterarbeiten-medmid/	
27.9.2013	Frontend / Arbeit einstellen (Upload)
HTTP://SPACES.KISD.DE/NEDA/2013/09/27/frontend-arbeit-einstellen-upload/	
28.9.2013	NeDA / Tagebuch 28.9.-14.10.
HTTP://SPACES.KISD.DE/NEDA/2013/09/28/neda-aktuelles/	
14.10.2013	NeDA / Tagebuch 14.10.-12.11.
HTTP://SPACES.KISD.DE/NEDA/2013/10/14/neda-tagebuch-14-10-12-11/	
25.10.2013	Uploadmaske (final)
HTTP://SPACES.KISD.DE/NEDA/2013/10/25/uploadmaske-final/	
31.10.2013	NeDA => Limited Public Beta
HTTP://SPACES.KISD.DE/NEDA/2013/10/31/neda-limited-public-beta/	
12.2.2014	NeDA Inbetriebnahme: KW8/9 – Update: online.
HTTP://SPACES.KISD.DE/NEDA/2014/02/12/neda-inbetriebnahme-18-2/	

5.2 Videos

Im Rahmen der Entwicklung des KISDarchivs produzierte ich 19 Videos, welche mir als Mittel dienten, aktuelle Entwicklungen in der sonst unzugänglichen lokalen Entwicklungsumgebung zu kommunizieren.

Mit Lancierung des KISDarchivs im Rahmen der Public Beta machte ich zudem zwei Tutorials öffentlich, welche die Funktionen der potenziellen Nutzerschaft zu erklären suchen.

10.2.2013	KISDarchiv / Eintrags-Eingabemaske / Sneak Preview
HTTPS://VIMEO.COM/	59378231 (Passwort: arche)
11.2.2013	KISDarchiv / Backend / Verwaltung, Sortierung, Suche von Einträgen
HTTPS://WWW.YOUTUBE.COM/	watch?v=Y4LdqX1xZdM
27.2.2013	KISDarchiv : Backend / Eintrag / Integritäts-Metabox
HTTPS://WWW.YOUTUBE.COM/	watch?v=KA5ZJhNprzE
6.3.2013	KISDarchiv Frontend / Sneak Preview
HTTPS://WWW.YOUTUBE.COM/	watch?v=9D2ER8oEKXc
1.4.2013	KISDarchiv / Frontend / Vorschau: Suche + Lesezeichen + Erweiterte Suche
HTTPS://WWW.YOUTUBE.COM/	watch?v=yli9wX-b_4U
2.4.2013	Frontend / Tag-Übersicht
HTTPS://WWW.YOUTUBE.COM/	watch?v=4HyJbkpajLw
2.4.2013	Frontend / Einzelansicht Lehrende, »Interaktive Headline«
HTTPS://WWW.YOUTUBE.COM/	watch?v=epGfdl-XmAl
27.9.2013	NeDA / Frontend / Uploadformular
HTTPS://WWW.YOUTUBE.COM/	watch?v=Hz1hG1E6sGs
27.9.2013	NeDA / Frontend / Navigator (cache)
HTTPS://WWW.YOUTUBE.COM/	watch?v=O1G300xBvOU
3.10.2013	NeDA / Frontend / Interaktive Headlines, Multiviews
HTTPS://WWW.YOUTUBE.COM/	watch?v=zn_3_YaGCAk
5.10.2013	NeDA / Frontend / Asynchrone Lesezeichen
HTTPS://WWW.YOUTUBE.COM/	watch?v=5atdWEEobj4
7.10.2013	NeDA / Front- und Backend / Uploadmaske / Dateiverwaltung
HTTPS://WWW.YOUTUBE.COM/	watch?v=TT9HEs5ly8w
9.10.2013	NeDA / Frontend / PDF Betrachter
HTTPS://WWW.YOUTUBE.COM/	watch?v=NCANnKK3_IY

14.10.2013	NeDA / Frontend / Intelligente Eingabemaske
HTTPS://WWW.YOUTUBE.COM/	watch?v=Z4fapQXJdTE
15.10.2013	NeDA / Frontend / Uploadmaske / Automatische Autorenkreation
HTTPS://WWW.YOUTUBE.COM/	watch?v=qQ7738LCfo4
15.10.2013	Frontend / Erweiterte Suche / Parse (Entwicklung)
HTTPS://WWW.YOUTUBE.COM/	watch?v=sjsxbyjWPXA
18.10.2013	NeDA / Frontend / Metasuche
HTTPS://WWW.YOUTUBE.COM/	watch?v=NI9M9XMKnJA
21.10.2013	NeDA / Frontend / Autorenindex
HTTPS://WWW.YOUTUBE.COM/	watch?v=fOiklX3bglk
25.10.2013	NeDA / Uploadvorgang / Simulation
HTTPS://WWW.YOUTUBE.COM/	watch?v=RqQBxIWDNh8
23.2.2014	KISDarchiv / Demonstration wesentlicher Funktionen u. Berührungspunkte
HTTP://SPACES.KISD.DE/	kisdarchiv/hilfe/
23.2.2014	KISDarchiv / Arbeit einreichen, eine Demonstration
HTTP://SPACES.KISD.DE/	kisdarchiv/hilfe/

6 Index

Symbols

– 51

404.php 187

A

Abgabevorgang 99

Abläufe 113

abstract 53

Abstract 100, 115

ACF 204

AdA 31

Administrator 206

AMD 33, 48

Analoges Archiv 180

Anmerkung 105, 115

Ansicht 69

anzeigen 51, 118

API 122

arbeit 52

Arbeit einreichen 98, 178

Arche 35

Archibald 34

ark-1.0.css 202

ark-1.0.js 202

ark_about.php 180

ark_acf_generate_search_meta() 145

ark_acf_load_value() 146

ark_acf_relationship_result() 145

ark_acf_update_value() 146

ark_add_dashboard_widget() 136

ark_add_post_meta_boxes() 136

ark_add_rewrite_rules() 150

ark_admin_header() 150

ark_area_description 118

ark_area_former_ID 118

ark_area_ID 118

ark_area_show 118

ark_aut 172

ark_author_degree 119

ark_author_directory.php 174

ark_author_firstname 119

ark_author_groups 121

ark_author_ID 119

ark_author_lastname 119

ark_author_mnr 119

ark_author.php 190

ark_author_set_taxonomy() 137

ark_author_start 119

ark_bugs.php 181

ark_cache 202

ark_cmp() 140

ark_cpt_icons() 136

ark_create_menu() 137

ark_current_page_url() 143

ark_custom_menu_order() 136

ark_dashboard_widget_function() 151

ark_deregister_styles() 150

ark_dir_ext.php 154

ark_filter_wp_title() 145

ark_fix_attachment_meta() 152

ark_fix_permalinks() 152

ark_generate_note_xml() 140

ark_generate_pdf_url() 140

ark_generate_search_meta() 152

ark-get-adv-search-results.php 154

ark_get_area_by_ID() 125

ark_get_area_meta() 125

ark_get_author_by_ID() 126

ark_get_author_by_name() 139

ark_get_author_degree() 127

ark-get-author-degree-by-name.php 154

ark_get_author_for_permalink() 128

ark_get_author_meta() 128

ark_get_author_name() 127

ark_get_elements() 129

ark_get_excerpt_by_id() 129

ark_get_ID_by_title() 141

ark-get-lecturer-area-by-name.php 155

ark_get_lecturer_by_ID() 130

ark_get_lecturer_by_name() 141

ark-get-lecturer-id-by-name.php 155

ark_get_lecturer_meta() 130

ark_get_lecturer_role() 141

ark_get_new_author_meta() 142

ark_get_record_subexam() 131

ark_get_tile() 131

ark_get_valid_year_range() 142

ark-get-word-from-number.php 155

ark_greeting() 132

ark_help.php 183

ark_index.php 168

ark_integrity_meta_box() 139

ark_integrity.php 179

ark_lct 173

ark_lecturer_area

ark_lecturer_email 120

ark_lecturer_firstname 120

ark_lecturer_groups 121

ark_lecturer_lastname 120

ark_lecturer.php 191

ark_lecturer_role 121

ark_lecturer_set_taxonomy() 137

ark_lecturer_shortcode 120

ark_lecturer_title 120

ark.log 195

ark_navigator.php 169

ark-pdf-url-check.php 156

ark_pdf_view.php 177

ark_permalinks() 147

ark_post_meta_boxes_setup() 151

ark_query_post_type() 147

ark_queue 90

ark_queue() 142

ark_queue (Cookie) 158

ark_queue.php 176

ark_record_area 52, 116

ark_record_author 52, 116

ark-record-author-dir.php 156

ark_record_exam_type 52, 116

ark_record_language 53, 117

ark_record_lecturer 52, 116

ark_record_legacy_id 51

ark_record_note 53, 117

ark_record_pdf 116

ark_record_pdf_url 51, 115

ark_record_post_title() 143

ark_record_search_meta 117

ark_record_subexam 116

ark_record_subexam_type 52

ark_record_type 51

ark_record_visible 51, 117

ark_record_year 53, 116

ark_register_settings() 137

ark_relationship_query() 148

ark_remove_mass_edit_bulk_action() 147

ark_remove_media_controls() 151

ark_remove_menu_entries() 139

ark_remove_row_actions() 149

ark_replace_excerpt_more() 149

ark-search-author.php 156

ark_search.php 170

ark_search_wisecrack() 132

ark_set_default_title() 149

ark_settings_page() 151

ark_set_title() 148

ark_setup() 152
 ark_show_bug_button() 109, 132
 ark_show_view_modes() 133
 ark_sort_record_by_year() 143
 ark_statistics.php 175
 ark_tags.php 171
 ark_update_slug() 148
 ark-upload 202
 ark_upload_complete.php 182
 ark_upload.php 178
 ark_view (Cookie) 158
 ark_view_detail.php 192
 ark_view_table() 133
 ark_view_table.php 193
 ark_view_tiles.php 194
 ark_wpautop() 150
 ark_zahl2wort() 144
 assets 202
 Au 63
 Author 206
 autor 52
 Autor 101
 Autor/en 115
 Autoren anlegen 211
 Autoren ansehen 213
 Autoren bearbeiten / löschen 213
 Autoren durchsuchen 211
 Autoren-Einzelansicht 172
 Autorenindex 93
 Autoren-Subtemplate 190
 Autorenverzeichnis 174
 Autor/in 119

B

Bachelor 102
 Bachelorarbeit 102
 Backend 44
 BEdes 102
 Benutzer 206
 Beschreibung 118
 betreuung 52
 Bibliothek 159
 BIdes 102
 Blended 34
 blogs.dir 202
 Boole 114
 Bugs 109, 181
 Büroklammer 64
 Button 62

C

categories 121
 Checkbox 114
 Contributor 206
 Cookies 158
 CSS 250
 Custom Fields 220
 Custom Post Types 114

D

Danke-Seite 182
 Darstellung 69
 Das analoge Archiv 108
 Dashboard 47
 dateigroesse 53
 Dateinamen 41
 Datenbank 203
 Datenbankmodell 246
 Datentyp 113
 declare_ark_area() 138
 declare_ark_author() 138
 declare_ark_lecturer() 138
 declare_ark_record() 138
 Detail 71
 Diagramm 97
 Diplom 102
 doku-archiv 32
 Dropdown 62
 Dunkelgrau 68

E

Editor 206
 Eintrag 115, 199
 Einträge anlegen / einreichen 208
 Einträge durchsuchen 207
 Eintrags-Einzelansicht 188
 Eintrags-Fuß 200
 Eintragsinhalt 200
 Eintrags-Meta 200
 Eintrags-Zusammenfassung 200
 Elemente 63
 Elementleiste 72
 E-Mail-Adresse 120
 entry-content.php 200
 entry-footer-single.php 200
 entry-meta.php 200
 entry.php 199

entry-summary.php 200
 Entwicklungsspace 251

F

Farben 67
 Farbkodierung 67
 Fehlerseite 187
 Feld 113
 File 114
 footer.php 198
 Formular 61
 Freischaltung 106
 Frontend 58
 functions.php 202
 Funktion 113
 Funktionsaufruf 122, 123

G

Gelb-Orange 68
 Google Charts 162
 Gruppenordner 32

H

HAL 32
 Hauptthema 102
 header.php 198
 Helferfunktion 122
 Hellgrau 68
 Hervorhebung 67, 163
 Hesse 3
 Hilfe 88, 183
 Hinterlegung 67
 Hook 123

I

id 51
 IMB 96
 Include 122
 Inline 165
 Integrität 94, 179, 248
 Integritätsanzeige 95
 Integritätsberechnung 94
 Integritätsmetabox 95, 96
 Interaktive Headline 65
 Intermediate 102
 Intial 63

J

jahr 53
 Jahr 85, 104, 115
 JavaScript 159
 JG 74
 jQuery 160
 jQuery Cookie 161
 jQuery Dropdown 161
 jQuery fancybox 161
 jQuery Migrate 160
 jQuery Tagsinput 161
 jQuery UI 160

K

Karten 72
 keywords 53
 KISDarchive 202
 KISDarchive.php 202
 KISD ID 119
 KST 165
 KT 165
 Kurzform 118
 Kurzname 120

L

legacy 202
 lehrgebiet 52
 Lehrgebiet 85, 115, 118, 120
 Lehrgebiete anlegen 218
 Lehrgebiete bearbeiten 219
 Lehrgebiete durchsuchen 217
 Lehrgebietsname 118
 Lesezeichen 89, 176
 Lesezeichen-Button 89
 Lesezeichen-Cookie 89
 Lesezeichen-Indikator 90
 Lesezeichen-Liste 158
 Lesezeichen-Seite 89
 letzter_bearbeiter 53
 LG 74
 Lightbox 161

M

mapAttributes 160
 Mapping 50
 Master 102

Masterarbeit 102
 Master Thesis 102
 matrikelnummer 52
 Matrikelnummer 119
 MEDes 102
 Mehrfachansicht Detail 71, 192
 Mehrfachansichten 70
 Mehrfachansicht Karten 72, 194
 Mehrfachansicht Tabelle 74, 193
 meta [Anfrage] 87
 Metasuche 83, 170
 Metasuchfeld 61
 MIDes 102
 Modul 113

N

Nachname 119, 120
 Name 119, 120
 nav-below.php 201
 Navigator 77, 169
 Nebenthema 102
 NeDA 20
 Neuer Autor 212
 NMD 48
 Nomenklatur 40
 no_pdf.php 196

O

Orange 68

P

Paginierung 201
 Parameter 123, 165
 Parsing 84
 PDF 115
 pdf_anmerkung 53
 PDF-Ansicht 64, 89, 177
 PDF anzeigen 115
 PDF-Datei 104
 PDF-fehlt-Subtemplate 196
 pdf_md5 53
 pdf_pfad 51
 PDF-Pfad 115
 pdf.php 195
 PDF-Subtemplate 195
 pending 106
 pending review 105

Permalinks 41
 PHP 250
 Piktogramm 64
 Pixel Art 96
 Plugin 159
 plugins 202
 Plugins 203
 post_content 53, 115
 post_tag 53, 115
 post_title 51, 115, 118, 119, 120
 Posttyp 113
 Posttypen 114
 Pr 63
 Präfix 40
 Proposal 102
 Prüfer 115
 Prüfer anlegen 215
 Prüfer bearbeiten / löschen 216
 Prüfer durchsuchen 214
 Prüfer-Einzelansicht 173
 Prüfer/in 85, 120
 Prüfer-Subtemplate 191
 Prüfungsstufe 85, 102, 115
 PS 74
 Publish 106

R

Referenz 114
 regex 86
 regular expressions 86
 Relevansi 81, 204
 Research Proposal 102
 Rot-Orange 68
 Rückgabewert 123, 165

S

Sandbox 29
 Schlagwort-Einzelansicht 186
 Schlagwörter 101, 115
 Schlagwortübersicht 171
 search_ark_record_author.php 157
 search_ark_record_lecturer.php 157
 search_ark_record_tag.php 157
 searchform.php 198
 search.php 185
 Seitenfuß 198
 Seitenkopf 198
 Seitentemplate 164

Select 114
 Self Initiated Project 102
 SI Arche 232
 Silbentrennung 205
 single_ark_author.php 172
 single_ark_lecturer.php 173
 single.php 188
 Spaltenbrowser 79
 sprache 53
 Sprache 40, 85, 104, 115
 Startseite 76, 168
 Statistik 97, 175
 Streamlining 24
 Studenten 121
 Studentinnen 121
 Studienbeginn 119
 Studiengang 102, 119
 Studiengang anlegen 220
 Subscriber 206
 Suche 81
 Suchergebnisse 185
 Suchfeld 61, 198
 Suchmeta 115
 Super-Admin 206

T

Tabelle 74
 Tagfeld 62
 tag.php 186
 tags 121
 Taxonomien 121
 Teilprüfung 85, 102, 115
 Template 113
 Template-Funktion 122
 Templates 75, 164
 Text 114
 Textfeld 61
 themes 202
 titel 51
 Titel 100, 120
 Titel der Arbeit 115
 TODO 229
 TP 74
 typ 52
 Typografie 59

U

Umlaute 205

UML-Diagramm 244
 Unsemantic 39
 Untertitel 100
 utf8_strrev() 144
 utf8_strshuffle() 144

V

Validation 161
 veraltete kürzel 118
 Videos 253
 View 69
 Vordiplom 102
 Vorname 119, 120

W

Wireframes 232
 WordPress 202
 Wortmarke 61
 wp_1186_commentmeta 203
 wp_1186_comments 203
 wp_1186_links 203
 wp_1186_options 203
 wp_1186_postmeta 203
 wp_1186_posts 203
 wp_1186_relevanssi 203
 wp_1186_relevanssi_cache 203
 wp_1186_relevanssi_excerpt 203
 wp_1186_relevanssi_log 203
 wp_1186_relevanssi_stopwords 203
 wp_1186_studenten2 203
 wp_1186_term_relationships 203
 wp_1186_terms 203
 wp_1186_term_taxonomy 203
 wp-content 202
 WST 165
 WT 165

Z

Zeige mir alles 87
 zuletzt_geaendert 53

Versicherung

Hiermit versichere ich, die Arbeit selbstständig angefertigt und keine anderen als die angegebenen und bei Zitaten kenntlich gemachten Quellen und Hilfsmittel benutzt zu haben.

Köln, 18.3.2014