

Fachbereich Media
Studiengang Informationswissenschaft

Aufgabenstellung für das Modul 5030

Programmierwerkstatt

betreut von
Prof. Dr. E. Lang, Prof. Dr. B. Thull

8. Oktober 2018

Inhaltsverzeichnis

1	Ziel	2
2	Aufgabenstellung	2
2.1	Anforderungen	2
2.2	Randbedingungen	3
3	Anmerkungen	4
3.1	Anmerkungen zum Arbeitsaufwand	4
3.2	Anmerkung im Falle von Einergruppen	4
3.3	Anmerkung zur Verwendung von HTML-Frameworks	4
3.4	Anmerkung zur Verwendung von Javascript	5
4	Ablauf des Projekts	6
4.1	Meilensteine	6
4.2	Meilenstein 1: Planung (Ende der 5. Woche)	6
4.3	Meilenstein 2: Implementierung (Ende der 9. Woche)	7
4.4	Meilenstein 3: Ausgestaltung (Ende der 12. Woche)	8
4.5	Abschluss (Ende der 14. Woche)	8
5	Dokumentation	8
5.1	Aufbau	10
5.2	Titelblatt	10
5.3	Anhang	11
5.4	Umfang	11
5.5	Form	11
6	Bewertung	12
6.1	Funktionsumfang	12
6.2	Gestaltung	12
6.3	Dokumentation	12
6.4	Programmierung	12

1 Ziel

Das Ziel der *Programmierwerkstatt* ist die Entwicklung eines webbasierten Informationssystems auf Basis der in den Modulen *Relationale Datenbanken*, *Webskripting* und *Anwendungsentwicklung* vermittelten Werkzeuge und Methoden. Insbesondere sollen Sie eine bestimmte Anwendungsdomäne sinnvoll modellieren und die so repräsentierten Daten über eine Website interaktiv zugänglich machen. Dazu ist es erforderlich, dass Nutzer Daten anlegen, sich diese Daten anschauen, Daten ändern und auch wieder löschen können (*Datenlebenszyklus*, CRUD¹). Darüber hinaus soll die Autorenschaft für Daten nachvollziehbar sein, was eine wesentliche Voraussetzung für eine gute Datenqualität ist. Dazu sollen sich Nutzer, die Daten anlegen oder manipulieren möchten, in die Website einloggen müssen.

Wir gehen davon aus, dass Sie zum ersten Mal eine Website entwickeln. Daher beschränken wir uns bewusst auf die in den oben genannten Modulen vermittelten Inhalte und erwarten als Ergebnis eine typische "Einsteigerwebsite". Eine solche Website kann aber trotzdem qualitativ hochwertig sein, d.h. einen breiten Funktionsumfang abdecken, fehlerfrei funktionieren und eine ansprechende Gestaltung (*look*) verbunden mit einer guten Nutzbarkeit (*feel*) aufweisen.

2 Aufgabenstellung

2.1 Anforderungen

Entwickeln Sie eine Website für eine Anwendung Ihrer Wahl, die folgende Funktionen enthält:

- Es gibt dynamische Inhalte (Informationsobjekte), für die ein Datenlebenszyklus auf der Basis einer nicht-trivialen Datenmodellierung implementiert ist (CRUD-Operationen).
- Es gibt eine Suchfunktion.
- Es gibt Medien, die eingestellt und gelöscht werden können.²
- Es gibt öffentliche Bereiche und geschützte Bereiche, in denen Daten manipuliert werden können, sodass die Autorschaft der eingegebenen Daten nachvollzogen werden kann.

Wann ist eine Datenmodellierung trivial? Die für diese Veranstaltung zur Anschauung entwickelte Anwendung *Logbuch*³ beispielsweise weist aus didaktischen Gründen ein sehr einfaches Datenmodell aus, das lediglich aus drei Tabellen und zwei Beziehungen zwischen diesen Tabellen besteht. Eine solche Modellierung ist trivial. Würde man dieses Modell um Ka-

¹CRUD steht für *Create, Retrieve, Update, Delete* und beschreibt den Lebenszyklus von Informationsobjekten, die typischerweise in Datenbanken abgelegt werden.

²Wir verzichten auf die Implementierung der Möglichkeit, hochgeladene Medien nachträglich zu ändern, was technisch sehr anspruchsvoll und aufwändig ist.

³Siehe Thema *Beispielanwendung* im Moodle-Kurs *Programmierwerkstatt*.

tegorien bzw. Schlüsselwörter (so genannten *tags*), z.B. zur Charakterisierung von Einträgen oder Reisen, erweitern, entstünde ein nicht-triviales Modell.

Medien können z.B. Fotos, Videos, Audiodateien oder auch PDF-Dokumente sein (siehe MIME-Types für weitere Beispiele), die typischerweise eher nicht in Datenbanken abgelegt, sondern im Dateisystem eines Servers verwaltet werden.

Zur Implementierung von öffentlichen und geschützten Bereichen ist es nicht notwendig, Registrierungsdialoge zu implementieren. Das ist aufwändig und kostet einfach zu viel Zeit. Es geht lediglich darum, Beispielnutzern ein Einloggen in Ihre Anwendung zu ermöglichen, damit die Autorenschaft für Änderungen nachvollzogen werden kann. Dazu reicht es völlig aus, einige Beispielnutzer von Hand in der Datenbank anzulegen.

Bisher im Rahmen der Programmierwerkstatt entwickelte Beispiele für solche Anwendungen waren webbasierte Fotoalben für Familien, Websites für kleine Museen, Websites für einen Instrumentenbaukurs oder, zunehmend beliebt, Websites rund ums Kochen oder Backen. In allen Beispielen gab es Bereiche der Websites, die öffentlich waren, und solche, die z.B. bestimmten Familienmitgliedern, einer Administratorin oder einem Redakteur vorbehalten waren. Es gab die Möglichkeit Fotos oder Videos in die Websites hoch zu laden und auch wieder zu löschen, und sie z.B. mit Bildunterschriften zu versehen, sie um erläuternde Texte zu ergänzen oder sie zu kommentieren. Weitere Beispiele wären etwa Blogs, FAQ- oder auch Dokumentenmanagementsysteme.

2.2 Randbedingungen

- Die Aufgabenstellung wird in Zweiergruppen bearbeitet. Dreiergruppen sind nicht zugelassen. Ein einzelner Bearbeiter oder eine einzelne Bearbeiterin bleibt die Ausnahme und muss abgesprochen werden. Siehe hierzu auch die Anmerkung im Falle von Einergruppen (Unterabschnitt 3.2).
- Gestaltung auf Basis des *Web Style Guide* (<http://webstyleguide.com>), d.h. nach Prinzipien der *Universal Usability*⁴, Standards und Konventionen. Abweichungen müssen in der Dokumentation begründet werden.
- Technologien PHP, HTML5, CSS3
- Javascript außerhalb eines Frameworks wie z.B. *Twitter Bootstrap* darf nur unter bestimmten Bedingungen verwendet werden: Der Einsatz muss gut begründet sein und beide Mitglieder der Gruppe müssen den Javascript-Code erklären können. Siehe hierzu auch die Anmerkung zur Verwendung von HTML-Frameworks (Unterabschnitt 3.3) und zur Verwendung von Javascript (Unterabschnitt 3.4).
- Die Lösung muss auf dem dafür vorgegebenen Server demonstriert werden können (siehe Bereich *Technische Information* des Kurses).

⁴Insbesondere werden im Gegensatz zu früheren Projekten i.d.R. keine Personas mehr eingesetzt.

- Sie sind selbst dafür verantwortlich, dass Sie ausschließlich Bild-, Video- oder Tonmaterial verwenden, für das Sie die entsprechenden Rechte haben. Das können z.B. selbst erstellte Materialien sein, oder solche, die mit einer **Creative Commons**-Lizenz ausgestattet sind.

3 Anmerkungen

3.1 Anmerkungen zum Arbeitsaufwand

Für ein Projekt werden Ihnen 10 so genannte *Credit Points* (CP) angerechnet. Jeder *Credit Point* entspricht einem Arbeitsaufwand von etwa 25–30 Stunden. Dieses Projekt fordert also etwa 250 Arbeitsstunden von jedem von Ihnen. Umgerechnet auf die Vorlesungszeit in einem Semester von typischerweise 16 Wochen sind das etwa 16 Stunden pro Woche, oder zwei volle Arbeitstage. Die Erfahrung der letzten Semester hat gezeigt, dass diese Größenordnung realistisch ist.

Statistische Erhebungen für die Bachelorprogramme in Deutschland haben gezeigt, dass Studierende typischerweise eher etwa die Hälfte der veranschlagten Zeit zur Erbringung der *Credit Points* leisten. Das trifft wohl auch auf viele Veranstaltungen in unserem Studienprogramm zu, aber nicht auf die Programmierwerkstatt. Wenn Sie also permanent das Gefühl haben, diese Veranstaltung fordere zu viel Zeit von Ihnen, dann mag Ihre subjektive Empfindung so sein; in Wirklichkeit entspricht der Zeitaufwand den Bologna-Anforderungen an ein Bachelorstudienprogramm.

Wenn Sie eine Woche trödeln, brauchen Sie in der Woche darauf bereits 3–4 Tage für das Projekt. Zwei Wochen Trödeln führt dazu, dass Sie eine komplette Woche einplanen müssen. Diesen Rückstand können Sie irgendwann nicht mehr aufholen. Nehmen Sie also die Zeitanforderung Ernst, einfach nur mal im Labor zu erscheinen reicht auf keinen Fall.

3.2 Anmerkung im Falle von Einergruppen

Im Falle von Einergruppen darf die Aufgabenstellung reduziert werden. Dazu kann er oder sie entweder auf die dritte oder die vierte Anforderung aus der Aufgabenstellung verzichten, d.h. die Möglichkeit, Medien einzustellen und zu löschen, bzw. öffentliche und geschützte Bereiche einzurichten. Die erste und zweite Anforderung (Implementierung von CRUD-Operationen, Suche) ist in jedem Fall zu erfüllen.

3.3 Anmerkung zur Verwendung von HTML-Frameworks

Wir möchten Sie ausdrücklich dazu ermuntern, ein HTML-Framework wie z.B. **Twitter Bootstrap**⁵ zu verwenden. Der Einsatz solcher Frameworks entspricht dem aktuellen Stand in der Entwicklung von Websites, ist effektiver als die manuelle Codierung aller Komponenten mit

⁵<http://getbootstrap.com>

HTML und CSS und führt zu ansprechenden Webseiten, bei deren Gestaltung gängige Konventionen leicht zu berücksichtigen sind.

Solche Frameworks sind allerdings komplexe Systeme, die nur richtig funktionieren, wenn sie wie vorgesehen eingesetzt werden. Verwenden Sie also z.B. **Twitter Bootstrap** bitte genau so, wie auf der **entsprechenden Website**⁶ dokumentiert. Sehen Sie bitte davon ab, nur einzelne Komponenten zu verwenden, oder z.B. in den Style-Sheets einzelner Komponenten herum zu basteln.⁷

3.4 Anmerkung zur Verwendung von Javascript

Die Verwendung von Javascript ist immer wieder Gegenstand von Diskussionen in den Arbeitsgruppen. Zunächst einmal: Javascript ist ein tolles Werkzeug, das überhaupt erst die Entwicklung reichhaltiger Nutzungserfahrungen im Web ermöglicht (siehe hierzu auch: W3C Web Design and Applications⁸). Allerdings ist die Entwicklung einer Einsteigerwebsite ein explizites Ziel der Programmierwerkstatt, nicht die Entwicklung einer möglicherweise marktgängigen, professionellen Website, oder einer Website auf dem allerletzten Stand der Technik. "Profis" unter Ihnen, die das Gefühl haben, nicht ohne Javascript auskommen zu können, sollten sich drei Fragen stellen: (1) Was hat meine Partnerin oder mein Partner davon (... außer dass er oder sie ziemlich beeindruckt sein wird)? (2) Wäre es nicht auch mal eine Herausforderung, eine gute Website ausschließlich mit Hilfe von modernem HTML und CSS zu entwickeln, nur um zu sehen, wie weit man damit inzwischen kommt? In der Beschränkung zeigen sich die Meister! (3) Was tun Sie eigentlich, wenn Sie eine Website im Intranet eines Unternehmens entwickeln sollen, in dem Javascript aus Sicherheitsgründen grundsätzlich abgeschaltet ist? Ein letzter Punkt: Wer Javascript wirklich intensiv lernen oder ausprobieren möchte, kann das im Rahmen eines turnusmäßigen Wahlpflichtangebots *Interaktive Webanwendungen* (Prof. Wiedling) tun.

Eine Ausnahme bildet die Verwendung eines HTML-Frameworks wie z.B. **Twitter Bootstrap**. JavaScript-Bibliotheken sind in solchen Frameworks ein inhärenter Bestandteil, ohne die diese Systeme nicht funktionieren können. Die Verwendung dieser Frameworks führt aber typischerweise nicht dazu, dass Anwendungsentwickler selbst JavaScript-Code bearbeiten müssen. Die Verwendung solcher Frameworks ist ausdrücklich erwünscht.

⁶<http://getbootstrap.com/docs/4.1/getting-started/introduction/>

⁷Selbstverständlich können Sie sich auch ein eigenes, so genanntes *Theme* entwickeln. Siehe dazu <http://getbootstrap.com/docs/4.1/getting-started/theming/>. Weil es sich dabei praktisch um ein eigenes Projekt handeln würde, raten wir Ihnen dringend davon ab, dies im Rahmen der Programmierwerkstatt zu tun. Einfacher wäre es, sich stattdessen für ein alternatives, fertiges *Theme* zu entscheiden (siehe z.B. <https://bootswatch.com/>).

⁸<https://www.w3.org/standards/webdesign/>

4 Ablauf des Projekts

4.1 Meilensteine

Abbildung 1 zeigt einen typischen und empirisch bewährten Projektablauf. Bitte beachten Sie, dass die Planungsphase insgesamt etwa ein Drittel des Projektzeitraums in Anspruch nimmt. Am Ende der jeweiligen Phasen stehen entsprechende Meilensteine.

Woche	#	Phase	Infodesign	DB-Design	Skripting
1-2	2	Kick off	Ideensammlung, Einarbeitung Bootstrap, Einarbeitung GitLab, Beispiel <i>Logbuch</i> verstehen		
3-5	3	Planung	Mission statement, Fragen- und Aufgabenkatalog, Site-Diagramm, Wireframes	ER-Diagramm, Aktivitätsdiagramme	Überprüfen der Skripting-Fähigkeiten: Datenbankanbindung, Formularverarbeitung, Sessions
		Meilenstein 1: Die Planung ist abgeschlossen, dokumentiert und besprochen. Mit der Abgabe der Planung melden sich die Gruppen für die Abnahme ihres Projekts (Prüfung) an.			
6-9	4	Implementierung	Inhalte	SQL-Abfragen, Datenbank mit Beispieldaten gefüllt	Skripting der HTML-Strukturen, Anbindung der Datenbank
		Meilenstein 2: Die Implementierung ist weitestgehend abgeschlossen.			
10-12	3	Ausgestaltung	Raster (<i>grid</i>), Gestaltungsrichtlinien oder <i>Theme</i>	-	Fertigstellung PHP-Skripte
		Meilenstein 3: Die Arbeit an der Website ist weitestgehend abgeschlossen.			
13-14	2	Abschluss	Dokumentation		Test und Debugging
		Meilenstein 4: Abnahme des Projekts			

Abbildung 1 Ablauf des Projekts, seine Phasen, bewährte Dauern dieser Phasen, die Tätigkeiten während dieser Phasen sowie die Meilensteine. Erläuterungen siehe Text.

4.2 Meilenstein 1: Planung (Ende der 5. Woche)

Zu diesem Meilenstein haben Sie eine präzise Vorstellung davon, wie Ihre Website später einmal aussehen und was sie können wird. Sie können für jede Ihrer geplanten Webseiten bereits SQL-Statements entwickeln und so überprüfen, ob Ihr geplantes Datenmodell wirklich vollständig ist und die Website auch tatsächlich geeignet unterstützen kann.

Der erste Meilenstein hat eine besondere Bedeutung. Wenn Sie z.B. die Rolle einer Informationsarchitektin oder eines Informationsarchitekten in einer Firma ausüben, dann wäre Ihre Aufgabe mit diesem Meilenstein schon wieder zu Ende. Ihre Planungsunterlagen würden an die IT- und die Designabteilung bzw. externe Dienstleister zur weiteren Bearbeitung gehen.

Auf der Basis der Site-Diagramme, der Aktivitätsdiagramme, der Wireframes und des Entity-Relationship-Diagramms würden Programmierer die Datenbank und die Skripte zur Anzeige der HTML-Gerüste und der Daten erstellen. Auf Basis des Mission Statements, der Aufgaben- und Fragenkataloge, der Site-Diagramme und der Wireframes würden Designer beispielsweise Raster und Stilrichtlinien festlegen sowie CSS-Dateien entwickeln, die sich auf die von Ihnen festgelegten HTML-Gerüste beziehen. Am Ende muss das alles zusammenpassen, damit eine funktionsfähige Website entsteht. Überlegen Sie sich, wie diese beiden Gruppen Ihre Dokumentation dieses Meilensteins lesen und verstehen können. Reicht die Dokumentation? Ist sie vollständig? Passt am Ende alles zusammen?

Aufgrund seiner Bedeutung für den weiteren Verlauf Ihres Projekts prüfen wir diesen Meilenstein ab und geben Ihnen eine Rückmeldung. Wir erwarten von Ihnen eine Dokumentation Ihrer Planung der Website, die die folgende Bestandteile haben soll (siehe Unterabschnitt 5.1):

- Mission statement
- Aufgaben- oder Fragenkatalog
- Site-Diagramm
- Wireframes
- Aktivitätsdiagramme
- *Entity Relationship*-Diagramm

Im weiteren Verlauf des Projekts werden Sie die Planungsdocumentation fortschreiben und zur Abschlussdocumentation weiterentwickeln. Um nicht unnötig Zeit zu verlieren, können und sollten Sie selbstverständlich unmittelbar nach Abgabe Ihrer Planungsdokumente mit der Implementierung Ihres Projekts beginnen, und zwar unabhängig von dem Termin, an dem wir Ihnen unsere Rückmeldung auf Ihre Planung geben.

Bitte beachten Sie: Die Planungsdocumentation stellt eine Teilleistung für das Modul “Programmierwerkstatt” dar. Mit der Abgabe einer Planungsdocumentation melden Sie sich daher formal zur Abnahme ihrer Lösung, d.h. zur Prüfung in diesem Modul, an. Ein Abbruch der “Programmierwerkstatt” nach diesem Meilenstein führt zu einem Fehlversuch.

4.3 Meilenstein 2: Implementierung (Ende der 9. Woche)

Zu diesem Meilenstein haben Sie Ihre Datenbank mit Beispieldaten gefüllt und bereits Medien (z.B. Bilder oder Videos) hinterlegt. Ebenso haben Sie die HTML-Struktur für alle Webseiten sowie die nötigen PHP-Skripte entwickelt, die diese HTML-Strukturen zusammen mit den aus der Datenbank abgerufenen Daten erzeugen. Die Website enthält aber durchaus noch Blindtext und Platzhalterbilder. Für den zweiten Meilenstein sollten Sie Ihre bereits angelegte Dokumentation fortschreiben. Wir erwarten von Ihnen, dass Sie insbesondere unsere

Rückmeldungen auf Ihre bisherige Dokumentation berücksichtigen und die erforderlichen Änderungen in die aktualisierte Fassung Ihrer Dokumentation einarbeiten.

4.4 Meilenstein 3: Ausgestaltung (Ende der 12. Woche)

Zu diesem Meilenstein haben Sie eine weitestgehend funktionsfähige Website mit Datenbankzugriffen und exemplarischem Inhalt, die auch schon ansprechend aussieht.

Auch für den dritten und letzten Meilenstein sollten Sie Ihre Dokumentation fortschreiben. Insbesondere sollten Sie für diesen Meilenstein Ihre Dokumentation um die **Darstellung des Rasters (*grid*) und der Stilrichtlinien (*style guide*)** bzw. im Falle der Verwendung von Bootstrap der Auswahl des *theme* ergänzen.

4.5 Abschluss (Ende der 14. Woche)

Zum Abschluss des Projekts erwarten wir von Ihnen:

1. Eine **Projektdokumentation**. Sie besteht aus der im Laufe des Semesters entstandenen Dokumentationen, sowie einem neuen Kapitel *Lessons Learned*, das Sie naturgemäß erst am Ende des Projekts schreiben können.
2. Eine **funktionstüchtige Website auf dem Übungsserver**. Bitte räumen Sie Ihren Ordner auf und löschen Sie nicht benötigte Dateien.
3. Eine **Videoaufnahme Ihrer Website**. Demonstrieren Sie Ihre Website und zeichnen Sie die wesentliche Interaktivität Ihrer Website mit Hilfe einer **Screencasting-Software** auf. Unter Windows können Sie dazu beispielsweise Camtasia verwenden (siehe Hinweis im Bereich *Technische Information* des Kurses), unter Mac OS reicht bereits der vorhandene Quicktime Player. Eine solche Aufnahme dauert etwa **2–3 Minuten** und erlaubt es uns, Ihre Website auch dann noch zu sehen, wenn Ihre Installation auf dem Server längst nicht mehr lauffähig ist.

5 Dokumentation

Die Ergebnisse des Projekts werden, wie in den Vorlesungen *Software Engineering* und *Informationsdesign* vorgestellt, in einer Projektdokumentation festgehalten.⁹ Diese Dokumentation entsteht im Verlauf des Projekts und besteht aus den folgenden neun Abschnitten:

1. *Mission statement*
Kurze **Darstellung der Idee** Ihrer Anwendung.
2. *Aufgaben- oder Fragenkatalog*
Dieser Katalog spiegelt die Anliegen der Nutzer wider, und muss für Ihre Anwendung

⁹Bitte beachten Sie: Die Programmierwerkstatt ist als eine Übung zum Modul *Anwendungsentwicklung* angelegt. Inhaltlich beziehen wir uns also sehr intensiv auf diese beiden dort angesetzten Vorlesungen. Sollten daher Begriffe oder Zusammenhänge unklar sein, schauen Sie sich bitte noch einmal den Stoff dieser Vorlesungen an.

glaubhaft und plausibel sein. Welche Aufgaben möchten Nutzer mithilfe Ihrer Website erledigen? Welche Fragen möchten Nutzer an Ihrer Website beantworten?

3. Site-Diagramm

Die Planung Ihrer Website – genauer: der einzelnen Seiten und ihrer Verlinkung – dokumentieren Sie typischerweise mit Hilfe von Site-Diagrammen, wie sie Lynch & Horton im *Web Style Guide* vorschlagen. Mit Hilfe des Aufgaben- und Fragenkatalogs können Sie überprüfen, ob Ihre Nutzer ihre jeweiligen Fragen und Aufgaben an der geplanten Website erledigen werden können, ob die Klickpfade lang oder kurz sein werden, und ob dies für die jeweilige Aufgabe oder Frage angemessen ist oder nicht.

4. Aktivitätsdiagramme

Für das Verfolgen von Links können Site-Diagramme gezeichnet werden, die die möglichen Klickpfade zeigen. Interaktionen mit Formularen sind komplexer und entziehen sich schnell dieser Darstellungsform, z.B. dann, wenn Plausibilitätsprüfungen gemacht werden. Stellen Sie daher die formularbasierten Interaktionen Ihrer Website mit Hilfe von UML-Aktivitätsdiagrammen dar.

5. Wireframes

Wenn Ihr Site-Diagramm wirklich vollständig ist, dann haben Sie auch die Möglichkeit, den Inhalt jeder einzelnen geplanten Seite zu inventarisieren. Was ist auf welchen Seiten zu sehen? Zeichnen Sie ein Wireframe für jede Seite Ihrer Website, d.h. dokumentieren Sie für alle Seiten Ihrer Site-Diagramme die grundsätzlichen Auslegungen dieser Seiten (*wireframes*) und skizzieren Sie ihre Inhalte.

6. Entity Relationship-Diagramm

Was auf diesen Seiten ist statisch, was ist dynamisch? Da die dynamischen Inhalte in einer Datenbank gehalten werden sollen, können Sie auch das Datenmodell entwickeln und in einem Entity Relationship-Diagramm dokumentieren. Dokumentieren Sie ggf., inwiefern und warum Ihre MySQL-Datenbank von Ihrem im ERD dargestellten Entwurf bzw. von allgemeinen Designregeln abweicht.

7. Raster oder grid-System

Dokumentation des Rasters, mit dessen Hilfe die Wireframes in eine pixelgenaue Auslegung der einzelnen Webseiten übersetzt werden können (siehe z.B. *Design for Developers*¹⁰ oder Kap. 4.5 und 4.6 im *Grundkurs gutes Webdesign*¹¹). Bei Verwendung von *Twitter Bootstrap* Dokumentation der Verwendung des *grid*-Systems.

8. Stilrichtlinien (*style guide*) oder *Bootstrap theme*

Festlegung der visuellen Aspekte der Gestaltung: Typografie, Farbschema, Grafiken wie z.B. Logo oder Bilder. Auf Basis der Stilrichtlinien wird das *CSS-Style Sheet* geschrieben. Bei Verwendung von *Twitter Bootstrap* Auswahl und Begründung des verwendeten *the-*

¹⁰<http://de.slideshare.net/Wolfr/design-for-developersonlineversionlong>

¹¹Björn Rohles: *Grundkurs gutes Webdesign*, Galileo Press, 2013 (Signatur: 40/ST 252 R738)

me.

9. Lessons Learned

Was hatten Sie ursprünglich geplant? Was davon konnten Sie umsetzen? Was fanden Sie gut, was nicht so gut? Was haben Sie am eigenen Projekt gelernt? Was können wir in der nächsten Programmierwerkstatt vielleicht besser machen?

5.1 Aufbau

Die Gliederung der Dokumentation ergibt sich aus der Aufgabenstellung:

1. Ziele und Anforderungen
 - Mission statement
 - Aufgaben- und Fragenkatalog
2. Spezifikation
 - Spezifikation der Website (Site-Diagramm)
 - Spezifikation der Webseiten (Wireframes, Aktivitätsdiagramme)
 - Spezifikation der Datenbank (ER-Diagramm, optional SQL-Anfragen)
3. Umsetzung
 - Visuelle Gestaltung (Raster, Stilrichtlinien)
 - Programmierung (*optional*)
4. Lessons Learned

Im Einzelnen besteht die Dokumentation für den Meilenstein *Planung* aus den beiden Punkten 1 und 2, und die *Abschlussdokumentation*, die die Planung fortschreibt, aus allen vier Punkten 1 bis 4. Dazu fügen Sie bitte vorne noch ein Titelblatt, eine eidesstattliche Erklärung, ein Inhalts- und ein Abbildungsverzeichnis ein, und hängen hinten bei Bedarf noch ein Quellenverzeichnis und vielleicht einen Anhang an. Die *Abschlussdokumentation* entsteht während der zweiten Hälfte des Projekts, indem Sie die Dokumentation der Planung fort-schreiben.

Selbstverständlich reicht es nicht aus, einfach alle Diagramme hintereinander abzubilden. Unter bestimmten Umständen kann es nötig sein, Darstellungen und Diagramme im Text zu erläutern. Das ist z.B. dann der Fall, wenn diese Abbildungen oder Teile davon nicht selbst-erklärend sind.

5.2 Titelblatt

Auf ein korrektes Titelblatt gehört:

- Hochschule und Studiengang (“Hochschule Darmstadt”, “Studiengang Informationswissenschaft”)
- Name der Veranstaltung (“Programmierwerkstatt”) und das Semester
- Titel des Dokuments (“Spezifikation” oder “Dokumentation”)
- Nummer der Gruppe sowie Namen der Gruppenmitglieder
- Datum der Abgabe

Die Gestaltung ist Ihnen überlassen.

5.3 Anhang

Die **Dokumentation soll lesbar sein**. Zu viele Abbildungen können manchmal den Lesefluss stören und sind deshalb im Endeffekt eher hinderlich für das Textverständnis. Das kann z.B. dann vorkommen, wenn Sie sehr viele Wireframes dokumentieren wollen. In einem solchen Fall kann es sinnvoll sein, ein exemplarisches Wireframe im Fließtext zu belassen und alle anderen in einen Anhang zu überführen, auf den Sie dann im Text verweisen können. Das Gleiche könnte für große und mehrteilige Site-Diagramme, ER-Diagramme oder auch Aktivitätsdiagramme gelten.

5.4 Umfang

Arbeiten, die bisher mit der Note 1 oder 2 bewertet worden sind, haben Fließtext, der typischerweise einen Umfang zwischen 2.500 bis 3.500 Wörtern hat. Das klingt mehr als es ist. Formulieren Sie also prägnant. Dazu kommen typischerweise Abbildungen:

- Site-Diagramm(e)
- Wireframes
- Aktivitätsdiagramme
- ER-Diagramm(e)
- Raster

5.5 Form

Wie man eine gute Ausarbeitung schreibt und in Form bringt, ist in zahlreichen Büchern ausführlich beschrieben. In unserer Bibliothek haben wir einige davon, manche sind sogar elektronisch verfügbar, wie z.B. Bruno Kremer: *Vom Referat bis zur Examensarbeit*, 4. Aufl., Berlin, Heidelberg: Springer Spektrum, 2014, Online-Ausg. <http://dx.doi.org/10.1007/978-3-642-41302-5>

6 Bewertung

Für die Bewertung werden die vier Aspekte Funktionsumfang, Gestaltung, Dokumentation und Programmierung berücksichtigt. Die Benotung Ihrer Leistung erfolgt gemäß ABPO §15 Absatz 1.

6.1 Funktionsumfang

- Gibt es dynamische Inhalte, die in einer nicht-trivialen Datenbank abgelegt sind? Kann ich dynamische Inhalte erzeugen, lesen, verändern, löschen und durchsuchen?
- Gibt es Medien, die ich einstellen und wieder löschen kann?
- Gibt es öffentliche Bereiche und geschützte Bereiche, in denen die Daten manipuliert werden können, sodass die Autorschaft von Einträgen nachvollzogen werden kann?
- Funktioniert das alles fehlerfrei?

6.2 Gestaltung

- Wie stark orientiert sich die Gestaltung an Konventionen? Sind Abweichungen von Konventionen plausibel begründet?
- Ist die Gestaltung einheitlich? Wirkt sie konsistent und stimmig? (Der so genannte *Look*)
- Wie fühlt sich die Interaktion an? Glatt, schnell und intuitiv? Oder holprig, verbunden mit Überraschungen und langen Wartezeiten? (Das so genannte *Feel*)

6.3 Dokumentation

- Sind alle neun oben aufgeführten Abschnitte vorhanden?
- Sind sie prägnant und nachvollziehbar geschrieben?
- Sind die Diagramme richtig angelegt?
- Spiegelt die Dokumentation den tatsächlichen Sachverhalt auf der Website wider?

6.4 Programmierung

Bei der Bewertung der Programmierung geht es in erster Linie darum herauszufinden, ob Sie Ihren Code wirklich selbst entwickelt haben. Wir erwarten also keinen perfekten Code, sondern vor allem, dass Sie ihn selbst geschrieben haben. Dazu organisieren wir am Ende des Semesters mit jeder Gruppe eine **Abnahmesitzung von etwa 30 Minuten** Dauer, in der wir Sie sehr intensiv zur Programmierung befragen. Die Fragen sind solcher Art, dass Sie sie nicht durch Auswendiglernen des Codes beantworten können.

Viel Erfolg!