
Aufgabe 4.1 - CLI

- a) Das CLI weist viele Schwächen und Probleme auf. Darunter fällt folgendes:
- Nicht intuitiv / schwer zu erlernen, da die Befehlsnamen kaum einen sinnvollen Hintergrund haben. (Warum db, cb, ub, dp, up?). Nur createBook ist an der Stelle ein sinnvoller Befehlsname.
 - Keine Keyword Labels wie z.B. cb from=source to=destination. Stattdessen werden nur die Argumente verwendet. Das macht das Ganze schwerer lesbar, anfällig für semantische Fehler und ist schwer zu verwenden ohne viel Übung.
 - Befehle führen nicht die Erwartungen aus. Bei cb erwartet man als User einen createBook Befehl und keinen Befehl zum Vertauschen der Bücher.
 - Es gibt keine Hierarchie Struktur und Befehle auf Datenbank Ebene (wie up, createBook) und Befehle auf Bücher Ebene (wie dp, cb) werden in einem Interface vermischt.
 - Anstatt dem Hinweis "What do you mean by this command?" sollte eine Hilfe zu dem Befehl bzw. zum Interface selbst ausgegeben werden.

Des Weiteren auch wenn nicht von Bedeutung für das CLI sollte der price des Buchs in der JSON nicht als Float gespeichert werden, da dies bei zukünftigen Berechnungen zu Rundungsfehlern aufgrund eines Überlaufs führen kann. Preise sollten immer in Cent gespeichert werden (Sprich also 1299).

b) Verbesserungsvorschlag

Eine neue Hierarchie nur für Bücher. (Vielleicht soll das Interface später noch für eine andere Verwaltung wie z.B. Kunden erweitert werden) Verwendung von klaren Befehlen, mit keyword labels. Die Syntax und die keyword labels sind immer die selben, sodass diese schnell erlernt werden kann.

```
book -create -data=<buch-data>
book -delete -id=<buch-id>
book -delete -all
book -update -id=<buch-id> -newPrice=<preis>
book -update -id=<buch-id> -data=<buch-data>
book -exchange -id1=<buch-id1> -id2=<buch-id2>
```

Ist zum Beispiel die Syntax eines Befehls falsch, wird folgende Fehlermeldung angezeigt:

Hilfe:

```
book -create -data=<buch-data>
```

Legt das Buch mit buch-data in der Datenbank unter einer zufällig bestimmten, noch nicht verwendeten, ID ab.

-data: Die Daten des Buchs, das erstellt werden soll

```
book -delete -id=<buch-id>
```

Löscht das Buch mit der ID buch-id aus der Datenbank

-id: Die Id des Buch, das gelöscht werden soll

book -delete -all

Löscht alle Bücher aus der Datenbank

book -update -id=<buch-id> -newPrice=<preis>

Updated den Preis des Buchs mit der ID buch-id

-id: Die Id des Buch, das geupdated werden soll

book -update -id=<buch-id> -data=<buch-data>

Updated das Buch unter buch-id mit buch-data. Wird nicht jeder Eintrag außer der ID geupdated, schlägt der Befehl fehl.

-id: Die Id des Buch, das geupdated werden soll

-data: Die Daten des Buch, das geupdated werden soll

book -exchange -id1=<buch-id1> -id2=<buch-id2>

Vertauscht die ID vom Buch mit buch-id1 mit der ID vom Buch mit buch-id2

-id1: Die Id des ersten Buch

-id2: Die Id des zweiten Buch

Aufgabe 4.2 - Suchmaske

Im Buchkatalog soll eine Suchmaske hinzugefügt werden. Der Nutzer soll alle Bücher nach Titel, Author, ISBN filtern können. Es soll außerdem ein Preisintervall angegeben werden, das die Bücher filtert.

Aus UX Sicht ist ein einzelnes Eingabefeld für Titel, Author und ISBN nach der gefiltert werden soll vorzuziehen, da im Buchkatalog wenig Platz verbraucht werden soll und der Nutzer die Suche intuitiv bedienen können soll.

Die Sucheingabe soll sich im Main Content Bereich oben über den Büchern aber unter dem Titel der Seite befinden.

Der Preis Slider/Filter befindet sich in der gleichen Zeile (Desktop Ansicht), zeigt Min- und Max-Preis des gesamten Katalogs bzw. den Max-Preis des aktuellen Filters an.

Die Bücher werden mit einem Bild des Buchcovers (hier Platzhalterbilder) sowie dem hervorgehobenen Titel in Blöcken dargestellt. Unter dem Titel werden die weiteren Infos gelistet: Author, ISBN, Preis

Das Cover und der Titel werden hervorgehoben, da diese wahrscheinlich eher zu einem Verkauf führen/das Interesse des Nutzers wecken als die ISBN oder der Preis.

Design-Vorlage des Buchkatalogs:



Aufgabe 4.3 - Usability

a) Usability:

Allgemein:

Die Schriftfarbe (headlines schlimmer als Fließtext) hat einen zu geringen Kontrast zur Hintergrundfarbe, wodurch Fließtext und Headlines teilweise kaum lesbar sind.

Die Animation oben rechts/das Logo enthält keine Informationen für den Besucher lenkt aber dennoch die Aufmerksamkeit auf sich, da es sich farblich stark von der Website absetzt und sich zusätzlich auch noch bewegt.

Das Footer Menu befindet sich below the fold, also nicht im sichtbaren Bereich des Users in der Desktop Version. Datenschutz und Impressum werden dort erwartet, aber nicht der Link zur *Über uns* Seite. Außerdem gibt es bereits auch einen *Über Uns* und *Anreise* Punkt im Fließtext auf der Startseite.

Nutzer sind es gewöhnt, dass man per Klick aufs Logo die Startseite erreicht, das funktioniert leider nicht auf dieser Seite.

Das Main Menu läuft in der mobilen Ansicht aus seiner Sidebar in den Main Content. Durch den geringen Kontrast der Schriftfarbe, sind überlaufende Teile der Menüeinträge kaum lesbar.

Fehler im Footer Menu:

Über uns:

Fehlerhafter Link/Content: *Anreise* Info, Nutzer erwartet *Über uns* Info

Impressum:

Irreführende Linkbezeichnung/Fehlerhafte Verlinkung zur Seite mit Anreise Infos, Nutzer erwartet *Impressum*

Fehler im Main Menu (Sidebar):

Vertauschte Menüs auf der Unterseite Speisekarte (siehe weitere Menüfehler)

Anreise:

Irreführende Linkbezeichnung/Fehlerhafte Verlinkung zur Seite mit *Impressum*, Nutzer erwartet *Anreise* Infos

Weitere Menüfehler:

Footer und Main Menu sind auf der Unterseite *Speisekarte* vertauscht. Der Nutzer ist verwirrt und findet sich vermutlich nicht mehr zurecht. Vor allem der Grund, dass der wichtige Link zurück zur *Hauptseite* nur durch scrollen zum Footer erreichbar ist, kann den Nutzer verärgern. Viele nutzen den typischerweise auf die Startseite zeigende Headerbereich/Headerlink, um sich bei unerwarteten Ereignissen wieder zurechtzufinden. Der Header ist aber leider kein Link.

Fehler auf Unterseiten:

anreise.html

Enthält Infos zum Impressum

Fehlerhafte/untypische Ausrichtung der Sidebar, die das Main Menu enthält: sie befindet sich auf einmal rechts vom Main Content Bereich

about.html & impressum.html

Enthält infos zur Anreise

anreise.html

Enthält infos zum Impressum, (Nutzer erwartet Infos zur Anreise)

card.html

Die Speisekarte enthält mehrere Speisen, allerdings nutzt sie in der Desktop Version nicht die gesamte Breite des Main Content Bereichs, sodass immer nur eine Speise schmal linksbündig ausgerichtet angezeigt wird.

Die Texte zu den Speisen sind sehr lang, sodass der User sehr viel scrollen muss und keinen umfassenden schnellen Überblick über die gesamte Speisekarte erhalten kann.

b) **Styleguide**

Farben:

Fehlerhafte Umsetzung:

Highlight des Hauptbereich Rahmens wurde nicht mit #fff sondern mit #dee2e6 umgesetzt

Hintergrundfarbe wurde nicht mit #800080 sondern mit (purple) #6f42c1 umgesetzt

Es existieren keine Buttons mit Hintergrundfarbe #800080

Schriftfarben wurden nicht mit #6b007c bzw. #000 sondern mit rgb(104, 0, 104) = #680068 und #212529 umgesetzt

Buttons wurden nicht mit der Hintergrundfarbe #0d82ff sondern mit #007bff umgesetzt.

Vollständigkeit:

Button sind nicht vollständig definiert:

Es sind zwei verschiedene Farben als Buttonfarbe definiert, wobei auch nicht genauer angegeben ist, ob es sich um Hintergrund und Schriftfarbe handelt oder nicht.

Effekte, Außenlinien etc. sind auch nicht weiter definiert.

Schriftfarben sind nicht vollständig definiert:

Es sind zwei verschiedene Farben als Schriftfarbe definiert, wobei nicht genau angegeben ist für welche Art von Text welche Farbe gilt (z.B. Absätze, Headlines..)

Es fehlen alle Angaben zu Linkfarben und evtl. Link-Effekten.

Es fehlen alle Angaben zur Schriftart und Größe für verschiedene Arten von Texten.

Argumentation:

Die Farben wurden gewählt weil sie gemeinsam auf eine Farbskala passen.

-> Das Argument ist nicht verständlich.

Die meisten Farbkombinationen besitzen einen starken Kontrast und es gibt daher keine Kontrastprobleme.

-> Es ist richtig, dass die meisten Farbkombinationen einen starken Kontrast haben, allerdings gibt es zwei sehr ähnliche Farben, die zudem vom Styleguide als Hintergrund-, Button- und auch als Schriftfarbe definiert sind.

Es wäre also erlaubt zwei sehr kontrastarme Farben als Hintergrund- und Schriftfarbe eines Elements zu nutzen, was zu erheblichen Kontrastproblemen führt.

Die Hintergrundfarbe wurde als zusätzliches Alleinstellungsmerkmal der Website gewählt.

-> die Hintergrundfarbe (#800080) hat kein Alleinstellungsmerkmal, da sie in Kombination mit #6b007c eine sehr kontrastarme Farbkombination ergibt.

Interface:

Fehlerhafte Umsetzung:

Der Hauptbereich der Website befindet sich nicht genau in sondern rechts von der optischen Mitte, angenommen das mit dem Hauptbereich nur der Bereich des Main Contents ohne Sidebar/Main Menu gemeint ist.

Enthält der Hauptbereich auch die Sidebar mit dem Main Menu und den Header so wäre der Hauptbereich zentriert um die optische Mitte platziert.

Insgesamt enthält der Styleguide kaum Angaben zum Interface, wodurch die Website wenige Umsetzungsfehler aufweist in Bezug auf das Interface.

Vollständigkeit:

Die einzelnen Bereiche sind nicht vollständig definiert (Header, Footer, Sidebar, Main Content etc.) Außerdem fehlt die Angabe wo genau sich welches Menü befinden soll und was genau den Hauptbereich der Website umfasst.

Es ist auch nicht näher definiert wie genau sich Bereiche voneinander absetzen (Farben, Schriften, Linien etc.), um dem Prinzip der gemeinsamen Region zu folgen.

Argumentation:

Jede Seite der Website besitzt zwei Menüs, eins zur Navigation der Hauptseite, Speisekarte und Anreise, ein weiteres zur Navigation zum Impressum, Datenschutz und Über uns.

-> Wurde wie im Styleguide gefordert umgesetzt

Der Hauptbereich der Website ist in der optischen Mitte gehalten und prägnant, außerdem folgt er dem Prinzip der gemeinsamen Region, sowie alle abgetrennten Bereiche der Website auch.

-> Mangels fehlender Definition des Prinzips der gemeinsamen Region in den Vorlesungsunterlagen beziehe ich mich auf die Definition nach <https://www.foto-schuhmacher.de/w/gestaltgesetze.html>.

Laut dieser Definition folgt der Main Content Bereich dem Prinzip, da er sich durch eine Außenlinien absetzt, die einen hohen Kontrast zur Hintergrundfarbe aufweist.

Die beiden Menüs folgen dem Prinzip ebenfalls, durch eine zum Hintergrund kontrastreiche Buttonfarbe. Der Header setzt sich nicht ab. Grundsätzlich sei zu vermerken, dass alle Bereiche die gleiche Hintergrundfarbe nutzen (bis auf Buttons) was dazu führt, das die meisten Bereiche bis auf Main Content und Button Menüs ineinander verschwinden.

Das Logo ist animiert, um die Aufmerksamkeit des Nutzers auf sich zu ziehen.

-> Diese Anforderung wurde umgesetzt, nur dass es sich nicht wirklich um ein Logo, sondern lediglich um eine Form mit Außenrand handelt.

c) Probleme unvollständiger Styleguide

Entwickler hat Entscheidungsfreiheit, da keine genauen Angaben existieren wo was auf der Website sein soll.

UX kann dadurch stark verändert werden, da es kein klares UI-Design gibt/eingehalten wird.

Außerdem wird die Brand Experience mitunter nicht korrekt über die Website vermittelt/Website passt nicht zur Corporate Identity.

Entwicklung bis zur vollständigen Website dauert länger, da Entwickler mehr Rückfragen stellen müssen oder Versionen aus Designgründen öfter überarbeitet werden müssen, bis sie vom Design-Team/Kunden o.ä. abgenommen werden.

Die Argumentation für das Aussehen und die Anordnung bestimmter Bereiche gegenüber dem Kunden ist schwieriger ohne vollständigen Styleguide.

Das Ziel der Website sollte eine für möglichst viele Nutzer vorteilhafte UX bieten und nicht die Umsetzung persönlicher Vorlieben des Kunden sein.

Aufgabe 4.4 - CRUD Interface

a) Beschreibung

Die Abkürzung CRUD steht für Create, Read, Update, Delete und ist ein Pattern zum Designen von Interfaces, welches meistens bei Datenbanken verwendet wird. Dies findet oft Verwendung bei der Anbindung von Datenbanken an einen Webservice, wobei mittels einer HTTP Methode (Post, Delete, Get), über eine Route eine Datenbank Interaktion ausgeführt wird.

Die Create Methode erstellt dabei einen neuen Eintrag in der Datenbank (also ein INSERT) Die Read Methode liest einen Eintrag aus der Datenbank

Die Update Methode aktualisiert dabei einen neuen Eintrag in der Datenbank ((also ein UPDATE)

Die Delete Methode löscht dabei einen neuen Eintrag in der Datenbank (also ein DELETE)

Vorteile:

Das Crud Interface wird oft verwendet, da es zum einen Schutz vor SQL Injection Attacken bietet, indem es Stored Procedures für die CRUD Befehle verwendet und zum anderen dadurch auch eine bessere Performance bietet, da der Ausführungsplan der Stored Procedure durch den SQL Server gecached wird und somit für die Zukunft bereits bekannt ist.

Beispiel: Die Webserver Lösung Django bietet ein CRUD interface an.

(See <https://stackify.com/what-are-crud-operations/>)

b) Create Interface

```
book -create -data=<buch-data>
```

Legt das Buch mit buch-data in der Datenbank unter einer zufällig bestimmten, noch nicht verwendeten, ID ab.

-data: Die Daten des Buch, das erstellt werden soll

Read Interface

book -read -id=<buch-id>

Gibt die Daten des Buch mit der ID aus der Datenbank zurück.

-id: Die Id des Buch, das angefragt wurde

Update Interface

book -update -id=<buch-id> -newData<buch-data>

Updated das Buch mit den neuen Buchdaten

-id: Die Id des Buch, das geupdated werden soll

-newData: Die geupdateten Buchdaten des Buchs

Delete Interface

book -delete -id=<buch-id>

Löscht das Buch mit der ID buch-id aus der Datenbank

-id: Die Id des Buch, das gelöscht werden soll

Das neue Interface enthält eine einheitliches Interface für alle Update, Delete und Create Operationen. Außerdem gibt es nun eine Read Operation zum Lesen eines Buch. Alle Operationen, mit Ausnahme der Create Operation, benötigen eine BuchID als Referenz auf das Buch welches in der Datenbank verändert werden soll.