Tatiana Nazarova, 6328375; Philip Spahn, 6043168

Dokumentation: Aufgabenblatt 4

Aufgabe 1

Teilaufgabe a

1. Auslöser der Fehrlermeldung wird in der Fehlermeldung nicht genannt. Man weiß also nicht, wie man das Problem lösen sollte, wenn eins auftritt.

2. Abkürzungsalgorithmus wurde nicht eingehalten bzw. die Abkürzungen sind uneinheitlich, das verwirrt den User:
z.B. beim 1., 3. und 4. Befehl ist der Algorithmus folgendermaßen: Man nimmt die ersten Buchstaben der einzelnen Wörter:
db: **d**elete **b**ook
cb: **c**hange **b**ook
ub: **u**pdate **b**ook

Da der Befehl *createBook* durch die Anwendung dieses Abkürzungsalgorithmus mit dem Befehl *cb* identisch wäre, müsste man sich hier eine neue Regel ausdenken und nicht einfach die Abkürzung weglassen. Da das den Befehl zu lang macht.
Die Befehle *dp* und *up* sind aufgrund des oben angewendeten Algorithmus irreführend abgekürzt:
up: da die Abkürzung *u* in einem vorherigen Befehl bereits für update verwendet wurde, assozieiert man ihn nicht mit dem Befehl 'löschen'.
Der Buchstabe *p* wurde in einem oberen Befehl bereits für price verwendet, daher würde der User den Buchstaben hier auch damit assozieieren. Für ihn wäre daher die Interpretation des Befehls als 'update price' naheliegender als die eigentliche Funktion des Löschens der kompletten Datenbank.
dp: Der Buchstabe *d* wurde in einem vorherigen Befehl bereits für 'delete' verwendet, daher würde der User ihn wieder mit dieser Funktion assoziieren. Somit wäre die naheliegendste Interpretation des Befehls laut dem obigen Algorithmus 'delete price' und nicht die eigentliche Funktion des Befehls, das würde den User verwirren.

3. Der Befehl *ub* wird häufig fehlschlagen, da die User den Befehl *dp* zum gezielten Updaten des Preises verwenden werden, um Laufzeit zu sparen, wenn bekannt ist, das sonst alles aktuell ist. Will man nun doch überprüfen, ob alles aktuell ist, so wird der Befehl *ub* fehlschlagen, wenn man vorher den Befehl *dp* ausgeführt hat, da der Preis selten zwei Mal hintereinander in kürzester Zeit geändert wird und somit in kurzer Zeit nicht zwei geupdatet werden kann.

Teilaufgabe b

1. Beschreibung des Fehlers, falls einer auftritt, statt der wenig aussagekräftigen Fehlermeldung *'What do you mean by this command?'*

2. Besserer Abkürzungsalgorithmus, beziehungsweise verwendet man den oben bereits beschriebenen Algorithmus richtig. Somit würden sich folgende Abkürzungen ergeben:
db: **d**elete **b**ook
cb: **c**reate **b**ook
si: **s**wap **i**ds
ub: **u**pdate **b**ook
up: **u**pdate **p**rice
da: **d**elete **a**ll

Dadurch bräuchte man keine Außnahmeregel, da man den Algorthmus auf alle Befehle anwenden kann. Außerdem steht hier jeder Buchstabe für eine Aktion oder einen Eintrag und wechselt nicht seine Bedeutung.

3. Man müsste den Befehl *ub*, dahingehend verbessern, dass er keine Fehlermeldung mehr werfen würde, wenn man direkt vorher den Befehl textitdp ausführen würde.

Aufgabe 2

Drückt man auf den Knopf *Filter* so erscheint über dem Bücherkatalog eine Suchmaske in der für jedes Suchkriterium ein Feld vorhanden ist. Man muss den Titel sowhohl als auch den oder die Autoren genauso eingeben, wie es in dem Kopf der Buchkarte steht (auf Groß- und Kleinschreibung achten), gibt es mehrere Autoren, so müssen ihre Namen durch Kommata oder unds getrennt, wie in den Buchkartenüberschriften angegeben werden. Werden Suchbegriffe eingegeben, die in dem Katalog nicht existieren, wird nichts angezeigt, das gleiche passiert, wenn man zwei Kriterien eingibt die nicht zusammenpassen, zum Beispiel man sucht einen Titel und gibt dazu die falsche ISBN in die Suchmaske ein. Drückt man nach dem Suchen erneut *Filter* und schließt die Maske, so erscheint der Katalog wieder vollständig.

Die Suche funktioniert durch eine for-Schleife, die über alle Bücher iteriert und so die passenden Suchkiteruen mit dem Suchwort abgleicht (wegen diesem Abgleich müssen die Titel genauso wie in den Buchkarten geschrieben werden). Bücher deren Kriterien nicht denen entsprechen nach denen gesucht wird, werden mit Hilfe von CSS ausgeblendet.

Aufgabe 3

Teilaufgabe a

Probleme:
1. Gute (schnelle und einfache) Erlernbarkeit ist nicht möglich:
Dadurch, dass die Menübuttons auf der rechten Seite oft ihre Position wechseln, muss sich der User nach jedem Klick ihre neue Position einprägen. Außerdem sind finden die Positionswechsel nicht nach dem Klicken jedes Buttons statt, sondern nur nach gewissen.
Wählt man zum Beispiel von der Hauptseite aus die Speisekarte auf, so tauschen die Buttons links und die Buttons unten den Platz. Wählt man dann den Punkt Anreise aus, so verlinkt der Button nicht nur auf die falsche Seite, sondern die Buttons wechseln auch auf die rechte Seite.

2. Schlechte Verständlichkeit der Webseite
Außerdem sind die Buttons falsch verlinkt, sowohl der 'Über uns'-Button als auch der Impressum-Button verweisen beide auf die Anreise und der Anreise-Button verweißt auf das Impressum. Diese falsche Verlinkung trägt zur Verwirrung des Nutzers bei und erschwert ihm somit die Bedienung der Seite zu lernen und zu verstehen, da die falsche Verlinkung logisch nicht ersichtlich ist.

3. Vollständigkeit wird verletzt
Die Über-uns-Seite ist nicht erreichbar, da der Button auf die Anreise verlinkt und es keinen weiteren Button gibt, durch den man die Über-uns-Seite erreichen kann.

4. Bildschirmfläche wird bei Speisekarte bei einem großem Bildschirm nicht optimal benutzt
Es bleibt viel ungenutzte Bildschrimfläche auf dem rechten Bildschirmrand übrig, das sieht nicht nur unästetisch aus, sondern der User muss auch noch unnötig lang scrollen, aufgrund der in die Länge gezogenen Speisekarteneinträgen.

5. Lila Schrift ist auf Lila intergrund nicht gut lesbar

6. Die Seite ist nicht responsive
Wenn man die Seite verkleinert, so überlappen die Buttons, beispielsweise bei der Speisekarte, die Einträge.

Lösungsvorschläge:
Zu Problem nr. 1:
Man ordnet den Buttons einen festen Ort zu und behält den bei der Navigation auf der Seite bei. Dadurch, dass die Buttons nicht unkoordiniert den Platz tauschen oder wechseln, wird es dem User leichter fallen sich ihre Position einzuprägen und die Webseite efffizient zu benutzen.

Zu Problem nr. 2: Man verlinkt die Buttons zu den richtigen entsprechenden Seiten, dadurch, wird es dem User auch leichter fallen die Funktion der Buttons zu verstehen.

Zu Problem nr. 3:
Man schreibt eine Über-uns Seite und verlinkt sie mit dem Über-uns-Button. Somit werden alle Seiten durch die Webseite erreichbar sein und sie wird vollständig.

Zu Problem nr 4:
Man zieht die Speisekarteneinträge in die Länge, bzw. wählt eine größere Column-Größe für die Karte. Dadurch, wird nicht so viel Platz verschwendet und der User muss nicht so viel scrollen.

Zu Problem nr.5:
Andere Schriftfarbe wählen, die einen höheren Kontrast zum Hintergrund aufweist.

Zu Problem nr.6:
Die Seite im Sinne des responsive Design implementieren, bzw. das Format der websiete je nach Bildcshrimgröße verändern.

Teilaufgabe b

Fehler des Styleguides:
1. Der Hauptbereich der Seite befindet sich nicht in der optischen Mitte, da er nach links versetzt ist.

2. Es gibt Kontrast-Probleme, da die Schriftfarbe *6b007c* auf der Hintergrundfarbe *800080* nur schwer zu erkennen ist.

Was noch fehlt:
1.Es wird zu wenig auf die Gestaltung des Logos eingegangen und welche Farben dafür verwendet wurden, würde man nun anhand des Styleguides versuchen es zu platzieren oder zu reproduzieren, so würde man es schwer haben.

2.Es wird nicht auf das responsive Design eingegangen, würde man die seite weiterentwickeln wollen, so würde man nicht wissen, wie der Formatwechsel von statten zu gehen hat.

3. Es wird nicht genau gesagt, wo die Buttons platziert sein sollen und wo im Vergleich dazu das Logo und der Hauptteil sein soll. Dafür wären zum Beispiel Skizzen hilfreich.

4. Die verwendete Schriftart wird nicht genannt.

5. Auf die Quelle und den Ort der Einbindung der Bilder wird nicht eingegangen.

Teilaufgabe c

Wenn man einen unvolständigen Styleguide hat, ist es schwer für andere zum Beispiel ein Logo oder Buttons bei der Weiterentwicklung der Webseite richtig zu platzieren. Dadurch wird das Design inkonsistent und schwerer für die User zu erlernen, somit wird die Usability verschlechtert und man verliert das Vertrauen der Kunden.
Hat man zum Beispiel einen nicht vollständigen Stylguide für die Entwicklung eines Logos, so kann es passieren, dass der Wiedererkennungswert des Unternehmens sinkt, da das nach dem Styleguide produzierte Logo vom Aussehen her nicht dem vorherigen entspricht.

Aufgabe 4

Teilaufgabe a

Bei einem CRUD-Interface handelt es sich um meine Benutzerschnittstelle bei der man die Funktionen *create, read, update* und *delete* an einem Datensatz ausführen kann. Ein Beispiel für ein CRUD-Userinterface ist MySQl-Workbench, durch diese Applikation kann man Datensätze erstellen aus Datenbanken abfragen, löschen und Änderungen updaten.

Teilaufgabe b

Damit das CLI die Anforderungen eines CRUD-Interfaces erfüllt, muss man noch Befehle für die Read-Funktionalität einfügen, da die restlichen bereits durch die Befehle db, da, cb, up und ub abgedeckt sind.
Befehlsliste:

db < buch-id > : Löscht das Buch mit der ID aus der Datenbank
cb <buch-data>: Legt das Buch mit der buch-data in der Datenbank unter der nächsten freien ID ab
si <buch-id1> <buch-id2>: Vertauscht die ID vom Buch mit buch-id1 mit der ID vom Buch mit buch-id2
ub <buch-id> <buch-data>: Updated das Buch unter buch-id mit buch-data.
up <buch-id> <preis> : Updated den Preis des Buchs mit der ID buch-id
da : Löscht alle Bücher aus der Datenbank.
la : Listet alle Bücher in der Datenbank auf.
lb <book-id>: Listet Buchinformationen des Buchs mit der Buch-ID auf.

Bücher werden im folgenden Format, nach JSON Syntax, gespeichert und verwaltet:
{ " i d " : " 1 2 3 4 5 " ,
" t i t l e " : "Lorem Ipsum ",
" i s b n " : 2 5 3 1 2 3 1 4 4 2 ,
" p r i c e " : " 1 2. 9 9 " ,
" au th o r " : " John Smith " ,
" d e s c r i p t i o n " : "Lorem ipsum d ol o r s i t amet , [. . .] " }

Schlägt ein Befehl fehl, so wird eine Fehlermeldung mit einer Umschreibung des Fehlers ausgegeben.