

Dokumentation - Abschlussprojekt

Bartender[©] by SorbTech

Autor des Dokuments	Simon Scapan	Erstellt am	07.02.2020
Dateiname	Documentation.doc		
Seitenanzahl	7	© 2020 Simon Scapan	Matrikelnummer: 6699329

Einleitung

Dieses Dokument dient der Befriedigung der Anforderung zur Dokumentation des Abschlussprojektes der Vorlesung Web Programmierung.

Zuerst werden die Anforderungen für das Projekt spezifiziert und ausformuliert. Diese basieren auf stichpunktartigen Informationen, die vom Dozenten als Status Quo vorgegeben wurden.

In den darauffolgenden beiden Punkten, Paper Prototype und Klassendiagramm werden jeweils nur die Diagramme aufgezeigt. Eine Erläuterung und Erklärung dieser folgt unter Konzept.

Viel Spaß beim Lesen der Dokumentation und dem dazugehörigen Quelltext.

Requirements

Vorgaben

Cocktail App:

- 3 Views: Zutaten Eingabe, Rezeptübersicht, Rezept (jeweils über eine Route angezeigt)
- Zutaten Eingabe:
 - Zutaten hinzufügen/auswählen (es muss eine Liste an vorhandenen Zutaten geben)
 - Ja/Nein Frage nach zusätzlichem Einkauf
- Rezeptübersicht:
 - Darstellung aller Rezepte die mit den gegebenen Zutaten erstellt werden können (mit Error und Matching-rate)
- Rezept:
 - Zutaten
 - Zubereitungsschritte (mit Checkbox)

Autor des Dokuments	Simon Scapan	Erstellt am	07.02.2020
Dateiname	Documentation.doc		
Seitenanzahl	7	© 2020 Simon Scapan	Matrikelnummer: 6699329

Ausformulierung

Die Applikation Bartender wird in drei Sichten eingeteilt. Jede dieser Sichten soll im oberen Bereich eine Navigationsleiste haben, um zwischen den Sichten springen zu können. Aus der Navigationsleiste heraus soll man mit jeweils einem Klick zum Rezept, zur Rezeptübersicht bzw. zum Hinzufügen der Zutaten gelangen.

Folgt werden die Einzelnen sichten detaillierter erklärt.

Zutaten Eingabe:

Die Komponente Zutaten Eingabe soll dem Nutzer alle vorhandenen Zutaten anzeigen. Diese soll er mittels einer Checkbox an bzw. abwählen können. Wenn Zutaten abgewählt sind, soll sich die spätere Anzeige der Rezepte so verhalten, als wären diese Zutaten nicht vorhanden. Wenn der Nutzer dann seine Wahl getroffen hat, sollen ihm alle Rezepte angezeigt werden, welche aus den Zutaten hergestellt werden können. Zu dieser Übersicht gelangt der Nutzer indem er die Schaltfläche "Drinks anzeigen" betätigt.

Rezeptübersicht:

In der Rezeptübersicht soll der Nutzer, wie die Bezeichnung schon verrät, einen Überblick über die möglichen Rezepte bekommen. Diese sollen mit ihren Namen und einem Beispielbild untereinander aufgelistet werden.

Beim Klick auf eines der Listenelemente, also auf einen Namen/Drink, soll der Nutzer zur Seite Rezept weitergeleitet werden und dort das Rezept einsehen können.

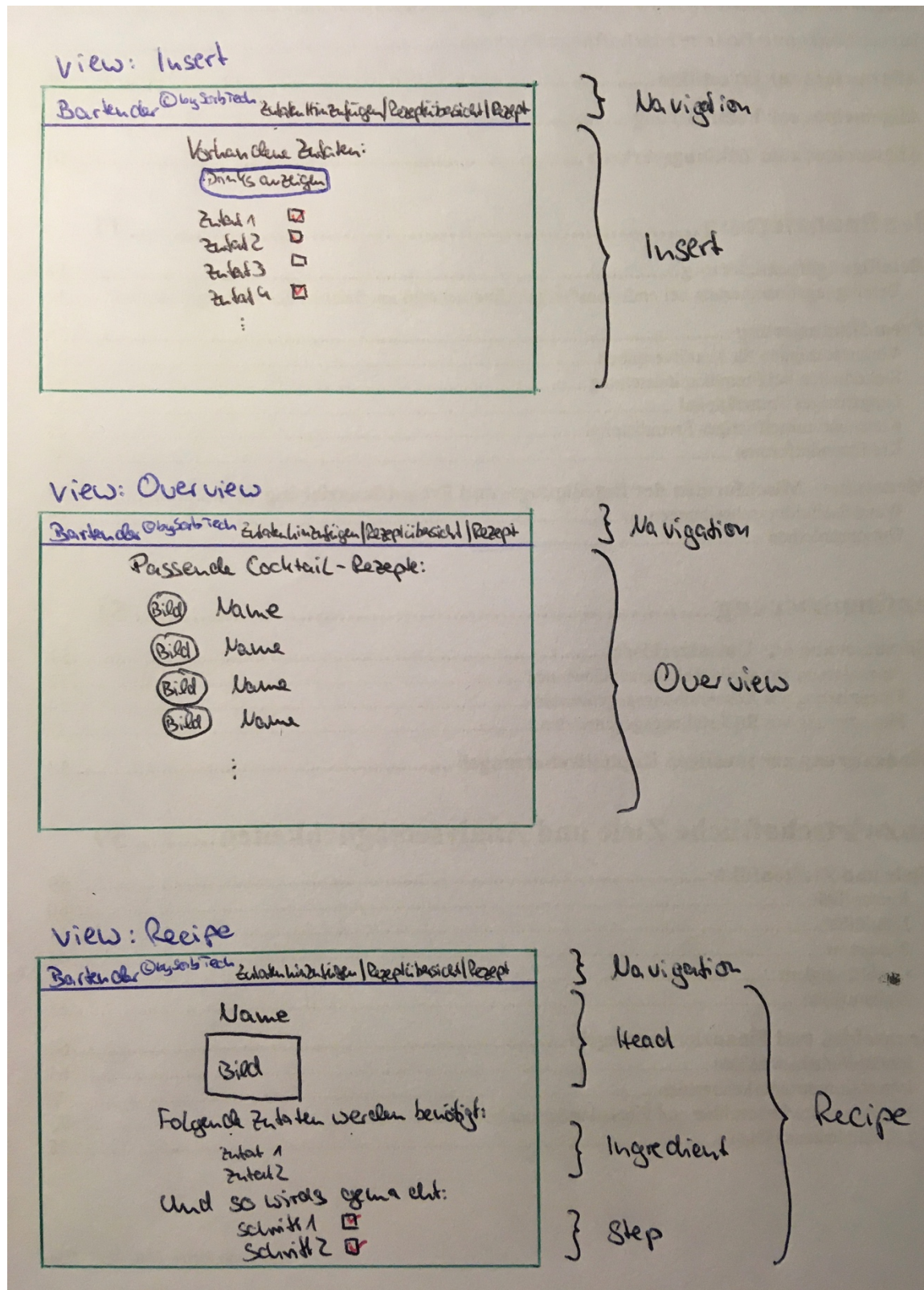
Rezept:

Aus der Rezeptübersicht kommend sieht der Nutzer dann das Komplette Rezept. Dieses setzt sich aus Bild, Namen, Zutaten und Zubereitungsschritten zusammen.

Im oberen Bereich stehen zuerst der Name und das Bild des Getränks. Darunter sollen sich die benötigten Zutaten befinden und darunter wiederum die Schritte bis zum fertigen Cocktail. Diese Schritte sollen mit Checkboxes ausgestattet sein, sodass man die einzelnen Schritte abhacken kann und damit eine fälschliche Zubereitung des Drinks ausschließen kann.

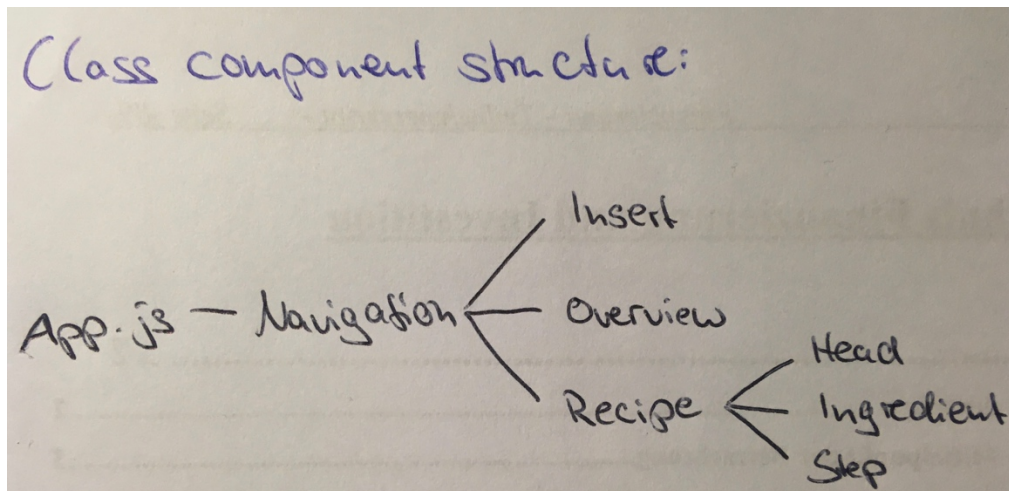
Autor des Dokuments	Simon Scapan	Erstellt am	07.02.2020
Dateiname	Documentation.doc		
Seitenanzahl	7	© 2020 Simon Scapan	Matrikelnummer: 6699329

Paper Prototype



Autor des Dokuments	Simon Scapan	Erstellt am	07.02.2020
Dateiname	Documentation.doc		
Seitenanzahl	7	© 2020 Simon Scapan	Matrikelnummer: 6699329

Klassendiagramm



Konzept

Nach dem Konkretisieren der Anforderungen schreiten wir mit der Konzeptionierung des Projektes fort. Zuerst wird auf die Einteilung der Sichten in Komponenten eingegangen, später dann auf die Interaktion zwischen den Komponenten. Dazu werden die beiden Diagramme in den darüber liegenden Punkten hinzugezogen.

Die Sichten wurden in den Anforderungen schon konkret beschrieben und so auch von der Funktionalität implementiert. Folgt wird jede Sicht wieder einzeln betrachtet.

Zuvor sei noch gesagt, dass jede Komponente ohne die Navigation betrachtet wird, da diese immer da ist und als eigene Komponente implementiert wurde. Der Grund hinter der Entscheidung ist, dass bei einem Seitenladefehler der Komponenten die Navigation noch da ist und vom DAU (Dümmster Anzunehmender User) ausgehend es aus Usability gründen angenehmer ist einen Button erneut zu drücken als die URL nochmals auszuführen.

Autor des Dokuments	Simon Scapan	Erstellt am	07.02.2020
Dateiname	Documentation.doc		
Seitenanzahl	7	© 2020 Simon Scapan	Matrikelnummer: 6699329

Insert:

Die Komponente Insert beinhaltet die zuvor erklärte Eingabe der Zutaten. Von einer tieferen Zerteilung der Komponente wurde abgesehen, da es sich in der Komponente nur um Text, einen Button und eine Liste mit Elementen handelt.

Overview:

Hier wurde ebenfalls von einer Zergliederung abgesehen, da nur eine Liste an Rezeptnamen mit deren Icon erstellt wird.

Recipe:

Die Komponente Recipe wurde hingegen um eine zusätzliche Schicht erweitert. Diese zerteilt die Komponente Recipe nochmals in Head, Ingredient und Step. Das ist insofern notwendig, da verschiedene Daten übergeben werden und unterschiedlich verarbeitet werden. Die einzelnen Komponenten werden dann in Recipe zusammengeführt und gesammelt angezeigt.

Die Struktur der Komponenten ist aus dem Klassendiagramm ersichtlich. Aus der App.js ausgehend liegt die Navigation dahinter, welche dann in Insert, Overview und Recipe aufteilt. Die Komponente Recipe wird nochmals in Head, Ingredient und Step aufgegliedert, wie oben bereits beschrieben.

Autor des Dokuments	Simon Scapan	Erstellt am	07.02.2020
Dateiname	Documentation.doc		
Seitenanzahl	7	© 2020 Simon Scapan	Matrikelnummer: 6699329

Abschluss

Das Projekt verlief weitestgehend nach Plan und Vorgabe.

Kritische bzw. Problematische Punkte waren zum einen die Einbindung der Material-UI Komponenten, da sich diese merkbar von den Standard-Komponenten unterscheiden. Weiterhin gab es während des Entwicklungsprozesses immer wieder Schwierigkeiten mit dem Handling der Daten. Diese Probleme wurden freundlicherweise vom Dozenten im Dialog aufgegriffen und nochmals erklärt. Somit war dies im weiteren Verlauf des Projektes kein Problem mehr.

Anfangs war angedacht die Daten nicht direkt in der App.js abzulegen, sondern diese aus einer Firebase Datenbank zu laden. Leider ist dieses Vorhaben an der Connection gescheitert. Diese konnte nicht hergestellt werden, da sie mit vielen weiteren Problemen verbunden war. Das Vorhaben wurde bei Projektfortschritt 20% abgebrochen.

Im Weiteren ist alles soweit wie geplant gelaufen. Auf die Positiven Sachen wird folglich nicht weiter tiefer eingegangen, da es den Rahmen überschreiten würde. Allgemein kann man aber sagen, dass alles, was nicht im Absatz zuvor unter Problemen aufgeführt wurde, zu Erfolg geführt hat und auch ohne Problematiken zu lösen war.

Autor des Dokuments	Simon Scapan	Erstellt am	07.02.2020
Dateiname	Documentation.doc		
Seitenanzahl	7	© 2020 Simon Scapan	Matrikelnummer: 6699329