

# **Dokumentation Cocktailapp**

## **Jannik Fischer Matrikelnr. 7296816**

### **Einleitung:**

Dieses Dokument stellt die offizielle Dokumentation meines Projektes dar. Die Dokumentation hat den Nutzen, dass die User einen guten Überblick bekommen können, wie ich bei dem Projekt vorgegangen bin und wie ich alles aufgebaut habe. Nachfolgend werden die folgenden Inhalte meines Projektes erläutert:

- Requirements
- Paper Prototype
- Klassendiagramm inklusive Erläuterung
- Konzept
- Abschluss

### **Requirements:**

Es soll eine Website für eine Cocktail Bar entwickelt werden. Der User soll die Möglichkeit bekommen, verschiedene Zutaten für seinen Cocktail auszuwählen. Die Website soll den User darauf Cocktails vorschlagen, die er mit den ausgewählten Zutaten herstellen kann. Beim Vorschlagen der Cocktails soll differenziert werden, ob der User noch zusätzlich einkaufen gehen möchte, oder ob er nur die Zutaten nutzen möchte, die er auch ausgewählt hat.

Die App soll insgesamt 3 verschiedene Views geben:

- Die Zutateneingabe
- Die Rezeptübersicht
- Die Rezepte

Die Navigation dieser 3 Views sollen über einen Router gelöst werden.

Zutateneingabe:

- Der User soll verschiedene Zutaten auswählen können, die er zuhause hat.
- Zudem soll der User zusätzlich die Option haben, noch einkaufen zu gehen

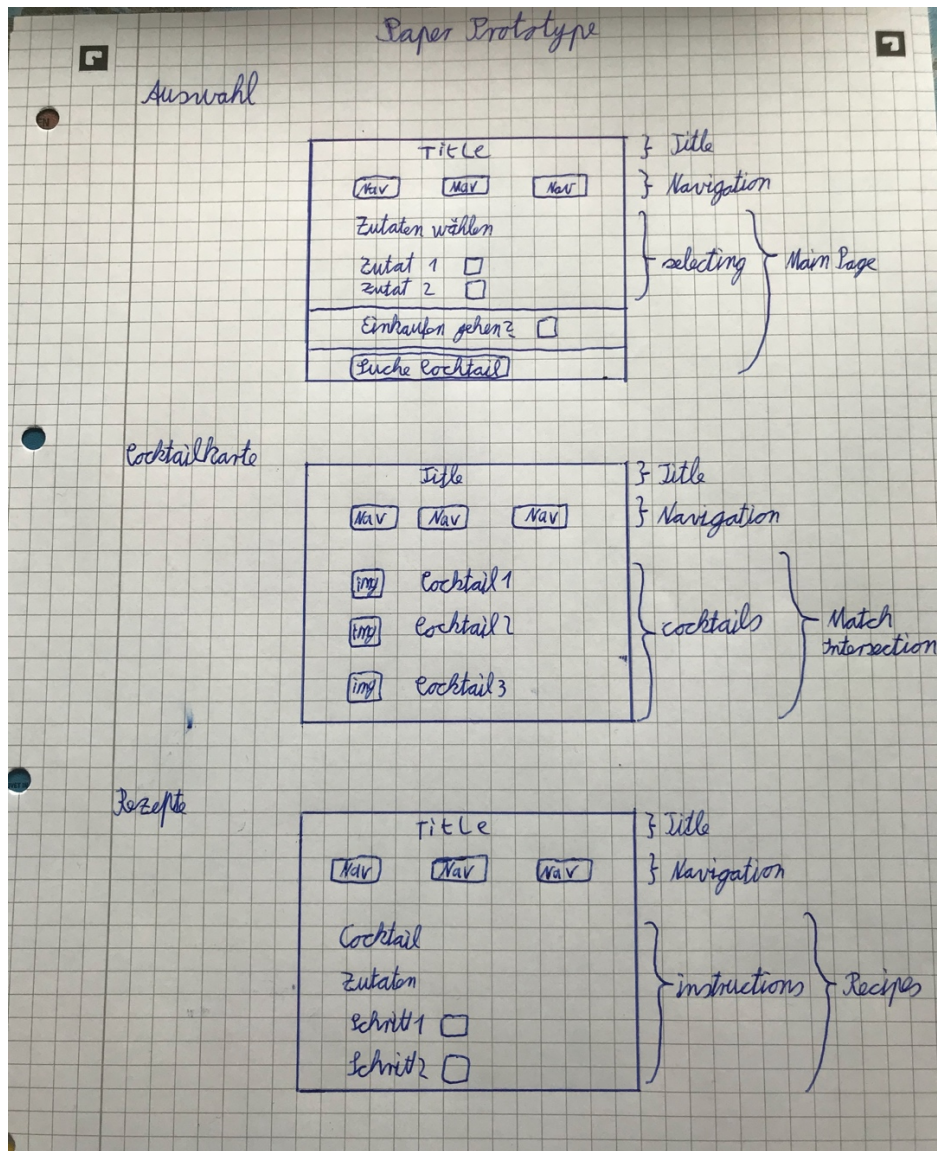
Rezeptübersicht:

- Die Rezeptübersicht soll alle Cocktails zeigen, die der User mit seinen Zutaten herstellen kann. Sollte der User nicht genügend Zutaten für keinen Cocktail haben, soll ein Error auftauchen
- Hat der User die Option „Einkaufen gehen“ angeklickt, so soll bei jedem Cocktail eine Matchingrate auftauchen, zu wie viel Prozent jeder Cocktail in Abhängigkeit zu den ausgewählten Zutaten matched

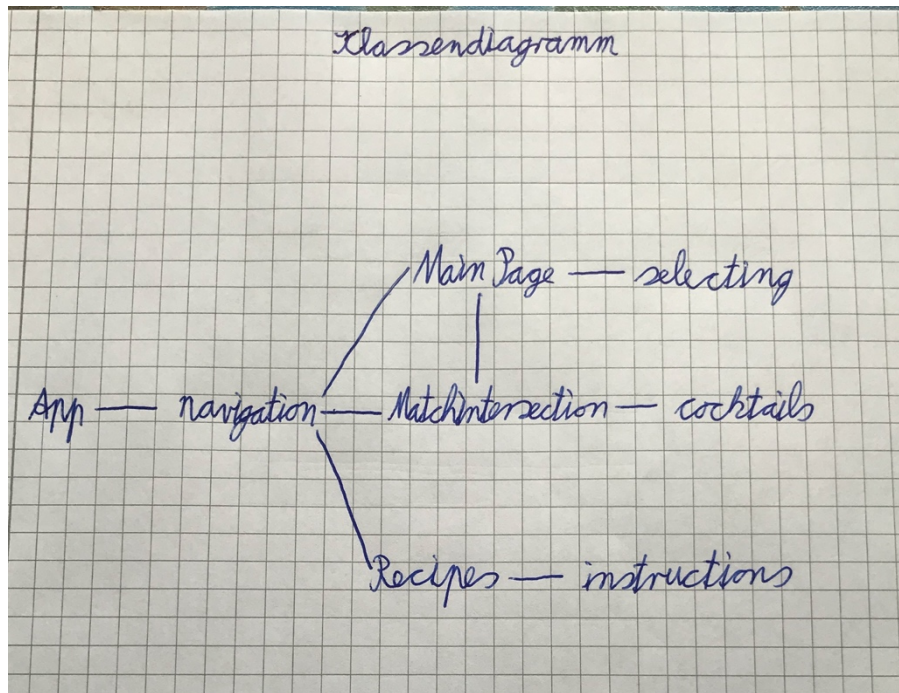
Rezepte:

- Hier sollen die Rezepte der gematchten Cocktails angezeigt werden
- Es sollen die Zutaten angezeigt werden, die zu diesem Cocktail gehören
- Zudem sollen auch die Zubereitungsschritte für den Cocktail angezeigt werden, die der User abhaken kann.

## Paper Prototype:



## Klassendiagramm:



## Klassendiagramm Beschreibung:

### Title:

Der Title ist eine starre Komponente, die auf jedem View erscheinen soll. Diese Komponente hat keine Funktionalität, sondern soll nur aus stylischen Gründen erscheinen. Aufgrund der Trivialität dieser Komponente wird diese als Header in der App implementiert, sodass sie keine einzelne Komponente darstellt.

### Navigation:

Die Navigation besteht aus drei verschiedenen Buttons, mit dem der User zwischen den einzelnen Views navigieren kann.

### Auswahl (MainPage):

Hier werden dem User verschiedene Zutaten angezeigt, die er über eine Checkbox auswählen kann. Zudem hat er über eine weitere Checkbox die Möglichkeit, zusätzlich einkaufen zu gehen. Auf dem Button „Suche passenden Cocktail“ werden dem User die gematchten Cocktails angezeigt.

### Menükarte (Matchintersection):

Hier bekommt der User einen Überblick über alle Cocktails, die die App anbietet, inklusive Bild und einer kleinen Beschreibung. Klickt der User auf das Bild, so gelangt er direkt zu den Rezepten.

### Rezepte (Recipes):

Hier werden dem User die Zubereitungsschritte angezeigt, die er für die Zubereitung benötigt. Diese sind über eine Checkbox anklickbar.

### **Konzept:**

#### App:

In der App sollen die gesamten Cocktails, inklusive deren Zutaten und Zubereitungsschritte gespeichert werden, die es insgesamt gibt. Zudem sollen hier auch alle Zutaten gespeichert sein, die der User auswählen kann. Soweit die Zutaten nicht ausgewählt sind, liegen diese als false vor.

Daneben stellt die App die Parent Klasse für alle anderen Klassen dar.

Außerdem wird hier auch der statische Titel implementiert, damit dieser auf jedem View angezeigt werden kann.

Datenbasis:

Cocktails: Cocktailname, Zutaten, Rezept, Beschreibung

Zutaten: Zutatenname, ausgewählt (wahr, falsch)

### Navigation:

Die Navigation vereint die drei großen Komponenten MainPage, Matchintersection und Recipes. Sie ist für die Navigation zwischen den einzelnen Views zuständig. Für jeden View gibt es einen Button, mit dem der User zum beliebigen View navigieren kann. Die Buttons sollen für den User jederzeit verfügbar sein, um eine bequeme Anwendung zu ermöglichen. Die Navigation funktioniert über einen Router zwischen den drei Komponenten.

### MainPage:

Die Komponente MainPage zeigt alle Zutaten an, die der User auswählen kann. Über eine Checkbox kann der User die Komponenten auswählen, die er im Bestand hat. Über die Child Komponente Selecting wird der Status der Zutaten auf True gesetzt, sobald eine Zutat angeklickt wird, sodass diese für das Matching einbezogen werden können. Unter den Zutaten steht die separate Checkbox „Einkaufen gehen.“ Klickt der User diese an, so wird diesem angezeigt, wie die Cocktails zu seinen Zutaten matchen. Sobald der User auf den Bestätigungsbutton drückt, werden ihm entweder die gematchten Cocktails angezeigt oder eine Fehlermeldung, wenn kein Cocktail matcht. Hat der User „Einkaufen gehen“ angekreuzt, so wird ihm eine Matchingrate vorgeschlagen.

### Matchintersection:

Über die Child Komponente Cocktails sollen alle Cocktails inklusive eines Bildes dargestellt werden. Diese zeigt den kompletten Bestand aller Cocktails. Die Matchintersection Komponente dient dabei als Schnittstelle zwischen der Cocktails-Komponente und der MainPage. Dadurch kann der Bestand mit den ausgewählten Zutaten abgeglichen werden.

### Recipes:

Diese Komponente zeigt die Rezepte der Cocktails. Über die Child Komponente Items wird zu jedem Schritt, den der User für die Zubereitung machen muss, eine Checkbox hinzugefügt. Diese Checkboxes haben keine Funktionalität, sie dienen lediglich der optischen Darstellung.

### Abschluss:

Abschließend betrachtet ist das Projekt erfolgreich gewesen. Es konnten alle gewünschten Anforderungen bei der Website einbezogen werden. Bezüglich des Time-Managements war stets ein angemessener Fortschritt im Projekt zu sehen, sodass nie die Gefahr bestand, das Projekt zu spät abzugeben. Die Website funktioniert gut, sieht aber noch ein wenig nutzerunfreundlich aus. Als Ausblick hätte man das Layout noch ein bisschen schöner machen können, sodass der User ein besseres Feeling bei der Anwendung bekommt. Außerdem wäre es noch gut gewesen, eine Datenbank für eine bessere Performance anzubinden. Da beide Aspekte jedoch einen zu hohen Aufwand bringen und nicht explizit vom Auftraggeber gefragt wurden, wurden diese Aspekte ausgelassen.

Ein hohes Risiko war jedoch, dass die Aufgabenstellung und der damit verbundene Aufwand deutlich komplexer und intensiver waren, als angenommen, sodass deutlich mehr Zeit in das Projekt investiert werden musste, als geplant. Auch die Klausurenphase und andere Projekte haben das Projekt teilweise stillgelegt, sodass es häufig zu größeren Pausen im Projekt kam.

Aus stylischen Gründen wurde am Ende des Projektes noch ein Impressum hinzugefügt. Geplant war, dass dieses in die App eingefügt wird, sodass es auf jeder Seite unten rechts in der Ecke zu sehen ist. Da dies aber nicht so gut aussah, wie erwartet, wurde das Impressum als eigene Komponente mit eigenem Button und View hinzugefügt. Da es sich um eine sehr triviale Komponente handelt, die lediglich den Ersteller inklusive Matrikelnummer anzeigt und dies auch nicht explizit vom Auftraggeber gefordert war (lediglich eine On-Top Komponente für eine bessere Anwendung), wurde das Klassendiagramm und der Paper Prototype im Nachhinein nicht verändert. Dies würden die eigentlichen Anforderungen unübersichtlich machen, und da das Impressum als Komponente nicht zu den gewünschten Anforderungen zählt, wird diese auch nicht in den Diagrammen erwähnt.

Abschließend kann gesagt werden, dass das Projekt erfolgreich war und gut verlaufen ist.

Mannheim, 24.02.2020

Ort, Datum

A handwritten signature in blue ink, appearing to read 'F. Fischer', with a stylized, cursive script.

Unterschrift