



National College of Ireland  
Higher Diploma in Science and Computing  
Web Development

2021/2022

Michelle Lally  
x20194269

Harry's Bar Interactive Cocktail Menu  
Technical Report

# Contents

Executive Summary .....	2
1.0 Introduction .....	2
1.1. Background .....	2
1.2. Aims .....	3
1.3. Technology .....	4
MERN Stack .....	4
Recommendation System .....	5
Password & Database Security .....	6
1.4. Structure .....	6
2.0 System .....	7
2.1. Business Requirements .....	7
2.2. Functional Requirements .....	8
2.3. Non-Functional Requirements .....	8
2.4. Design & Architecture .....	9
2.5. Graphical User Interface (GUI) .....	15
2.6. Testing .....	19
Black Box .....	19
White Box .....	38
3.0 Conclusions .....	43
4.0 Further Development or Research .....	44
5.0 References .....	44
6.0 Appendices .....	45
Github Repository Link .....	45
Github Issues .....	45
README.md .....	46
Python script .....	46

## Executive Summary

This report proposes how a selected cocktail bar may digitize their cocktail menu with customer interaction and present how it can be used to increase sales and attract customers while tailoring a unique experience for each user that, according to my research, cannot be found in any other local establishments.

After witnessing the reaction from some customers who are not interested in reading through a page of different concoctions that make up a cocktail menu, I discovered there could be a market for an online menu which can breakdown a list of items of most interest to the individual, while filtering out the rest.

You will also find information on the benefits of introducing a machine learning recommendation system into a native mobile application in this report and learn about the technologies that the functionality is built on.

### Product

A mobile application that gathers data from the user on their cocktail preferences which will then provide a short list of drinks explicitly related to the information provided with a recommendation system and staff login to update the menu.

### Market

This application would be of interest to businesses whose customer base would be familiar with the scanning of a QR code to access an online version of their menu.

It would also require a medium-large selection of drink items on the menu with a variety of spirits and distinctive flavour profiles to categorize each item efficiently.

### Competition

Of course, there are thousands of applications and websites which provide a digital copy of an establishment's menu, however, it is yet to be discovered if an application tailored for a single bar or restaurant that includes the additional functionality you will find disclosed in this report already exists.

## 1.0 Introduction

### 1.1. Background

This project is based from an idea I had when working in a very busy, high end cocktail bar with a wide-ranging cocktails menu which always had customers struggling to choose what they would like to drink.

This idea became a reality after I built a website for a local bar with the same issue as the one I had previously worked in. As part of an assignment last semester, I made a web application to introduce myself to the React libraries. Although I was very happy with the web app, I found that it was more suitable to be used on a laptop/PC rather than a mobile device.

This is why I chose to reproduce a similar program which would be more mobile-friendly using React Native instead to build a native mobile application instead.

Before I began learning React Native, I thought that from building a React application, that my previous experience would have been beneficial with learning the native development side, but other than the use of JavaScript, it was completely different as React uses HTML as the markup language but React Native has its own markup language which was entirely new for me.

In addition to the functionality included in my previous project, I wanted to implement a recommendation system into the mobile application which I done through the use of Googles TensorFlow.js machine learning platform and a python script to build, train a model.

This was arguably the most challenging part of completing this project, which made it the most rewarding to finally see it in action and to accomplish a long term goal of self-learning how to a machine learning system into a mobile application, which is not something that is widely done according to my research.

## 1.2. Aims

Businesses nowadays require an online presence to become noticed, continue to be noticed and to provide notices on products and services they provide.

Boosting sales and attracting customers are the basics that is expected by business owners when they build or buy a website to promote their business online.

This application aims to offers a unique experience in assisting customers and staff members with menu item recommendations, on top of the basics mentioned above.

After building a similar program as a web application in a previous module, I felt the UI was not as responsive on a mobile device as it was on a computer, but knowing that most customers wouldn't be bringing their laptops to a bar.

I feel that this is an issue that was raised from my web development specialization as I believe my application is clearly much better suited to a mobile application which is why I decided to convert it for this project.

## 1.3. Technology

### MERN Stack

#### ■ MongoDB

- A document-model database which stores data in documents which don't have to follow the exact same structure as each other record in the database which is more natural for humans to understand.
- It is also a distributed database as there is a copy of each database on multiple servers in the cloud. This means if a single server were to crash and the database you require is stored there, that it can be accessed from one of the many other servers
- MongoDB increases a systems fault tolerance, scalability and access to fast data to each and every database without any extra configuration which made it a very desirable choice for my application.

#### ■ Express

- A web lightweight but powerful framework for between a web-page and a server for Node.js applications.
- It is widely used in the production of mobile and web applications to easily handling HTTP requests and responses, providing and configuring routing components, building RESTful API's and for connecting to databases and executing queries.
- Express saves a developers and companies countless hours by providing all of this functionality in a few lines of code.

#### ■ React Native

- A JavaScript framework for building native mobile applications for iOS and Android devices without having to build separate apps for each of the platforms. A React Native application supports cross platform use.
- For this reason alone it is a huge time saver for making mobile applications, but because of its instant compiling technology during production, any changes made in the code are instantly visible on the simulator which is not commonly found with other frameworks.
- Facebook, Instagram, Pinterest, Skype and UberEats are just some of the popular applications that are built on the React Native Framework.

#### ■ Node.js

- Before Node.js, JavaScript code could only be run in a web browser. Now, it can be executed as a cross-platform runtime environment for building powerful front and back end services for full stack applications.

- NPM is a Node.js node package manager which is a public library of over 800,000 code packages used by approximately 11 million developers to assist with the development of JS applications.
- It runs on the V8 JavaScript runtime engine which converts JavaScript code into machine code which can be run on a computer without having to spend time interpreting it first.

## **Recommendation System**

### ■ Tensorflow

- This open source framework is developed by Google to provide an end to end machine learning platform which is used to design, build and train deep learning models through the use of high level API's.
- Tensorflow implements these algorithms for neural networks by inputting a multi-dimensional array of data which is also known as tensors, which flows through a set of operations to produce a model.
- Its capabilities of being able to run on a desktop, cloud or a mobile device, makes it extremely popular for all types of development. A model can be trained in the cloud to be used on many machines.

### ■ Keras

- This Python high level API is designed for humans to conveniently access and customize machine learning frameworks like Microsoft Cognitive Services, or in my case, Tensorflow.
- It runs on top of Tensorflow, to simplify its learning and implementing its deep learning neural networks, as it can be quite complex to use solely, especially for beginners like myself.
- Keras is a popular among developers as for something that is used to perform complex computation and evaluate deep learning models it is made extremely efficient, powerful and easy to use.

### ■ Python

- This high level programming language with is one of the most popular among machine learning and artificial intelligence, but it can also be used for web, mobile and desktop application development.
- It is most prevalent in the ML/AI community because of its simple syntax and makes learning data science much easier than most other languages as reading, evaluating and debugging the data.
- Python has a large community, open source with widespread library support to support machine learning processes with plenty of online help and resources.

## **Password & Database Security**

### ■ Bcrypt

- Brcypt is one of the most popular and secure password hashing algorithms which can be used when storing passwords in a database by converting the plain text to hashes, protecting it from hackers.
- It can take a string, run it through the hashing algorithm and then produce a string of characters which looks like gibberish, this is called the hash. It is nearly impossible to run a hash through the same algorithm and see the original text making this the most optimal solution to storing user passwords securely.

### ■ JSON Web Token

- JWT is used for authenticating users that has a different approach to cookie based authentication by checking that the correct user is requesting access to the system using JSON Objects.
- When a user registers for an account, the server will generate and respond with an access token with a secret key which must be provided when accessing other routes on the server.
- And when a request is made to log into a system, the user credentials are passed to the server and the JWT must be verified by checking that the signature passed is the same as the one which was defined.

## 1.4. **Structure**

This report is composed of 5 sections:

- **Section 2:** All business, functional, non-functional requirements can be found here along with vigorous test cases, numerous diagrams describing system design and architecture along with screenshots of the application.
- **Section 3:** Conclusions on what this application is capable of and what it is limited to do, the markets it can reach, improvements that could be made
- **Section 4:** Here is explained what could be done differently provided there was additional time, resources, expertise.
- **Section 5:** You can find a list of which resources were used for inspiration and help throughout this project development.
- **Section 6:** All additional information with links to the online code repository, code snippets, video demonstration sources.

## 2.0 System

### 2.1. Business Requirements

Goal	Business Goal Description	Goal achieved
1	Allow the user to complete a series of questions and gather information on their cocktail preferences of the following: Type of alcohol: Vodka, gin, whiskey, rum, tequila, other. Flavour profiles: Sweet, sour, refreshing, spicy, all	Yes. This goal was achieved, however, due to the lack of cocktails containing tequila as the main spirit, I decided to remove this option from the list, along with spicy
2	Provide the user with clear instruction on navigation through the process	Yes. The navigation component provides a back arrow and other pages include a button which allows them to navigate to another page
3	Return a list of cocktails with a description from the database which match the users desired ingredients and flavours which is displayed to the user	Yes. A list of cocktails with their names, ingredients and the glass they are served in is displayed to the user
4	Recommend cocktails similar to their chosen item	Yes. Some recommendations may not be 100% accurate, but the functionality is there but will require updating
5	Allow staff to log in with predefined log in credentials to manage the menu items	Yes. Authenticated using MongoDB, password being hashed with JWT and Bcrypt to ensure a secure login
6	Allow staff to add new cocktails to the menu	Yes. A form requesting data from the user is displayed and the data is inserted to the database
7	Allow staff make changes to existing cocktails on the menu	Yes. A form requesting data from the user is displayed and the database is updated
8	Allow staff to remove cocktails from the menu	Yes. A delete button is provided for each cocktail and when pressed, the record is removed from the database.

## 2.2. Functional Requirements

Use-case Id	Use-Case Description	Modified Y/N
1	Selecting preferred type of alcohol	Y. Removed 'Tequila' as an option
2	Selecting preferred flavour	Y. Removed 'Spicy' as an option
3	Show suggested cocktails	N
4	Show recommended cocktails	N
5	Login to the system	N
6	Show cocktail list	N
7	Add a new cocktail to the menu	N
8	Update existing cocktail	N
9	Delete existing cocktail	N
10	Display a cocktail menu.	Y. Added a page so the user can read about every cocktail on the menu without having to pick preferences

## 2.3. Non-Functional Requirements

NFR ID	NFR Description	Tested	Met
1	Application should be loaded in 5 seconds or less	Y	Y
2	Results from the database should be returned within 3 seconds or less	Y	Y
3	Successful login should be completed within 3 seconds or less	Y	Y

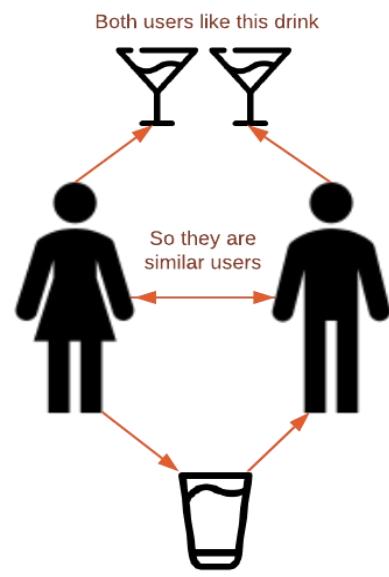
4	Recommendations should be made to the user in 3 seconds or less	Y	Y
5	Unauthorized user should not be able to read or write to the database	Y	Y
6	Key required for database access should be securely stored	N	Y
7	Keys required for API should be securely stored	N	Y
8	All user input must be validated	Y	Y
9	Database should be capable of storing more than the number of cocktails on the menu to allow for additions to the collection	Y	Y
10	Additions made to the collection after development should not effect any existing features	Y	Y

## 2.4. Design & Architecture

### Recommendation System

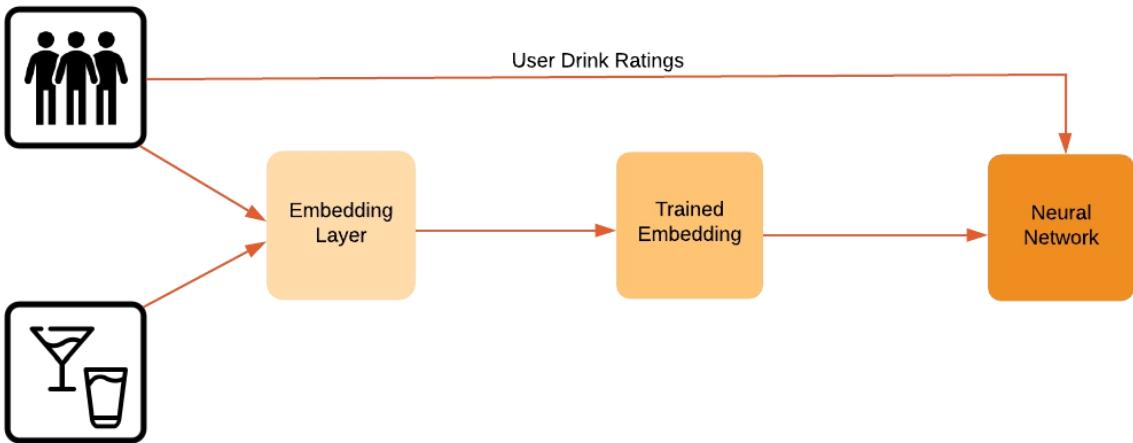
#### Collaborative Filtering System

- This type of recommendation system is based on how the user behaves. It gathers information on individual users, such as their likes and dislikes. Then, it will find other users who behaved in a similar way and offer recommendations based on what the other user liked.
- This project uses the collaborative filtering approach to building the recommendations system.



#### Embedding

- An embedding layer is used in neural networks for building a profile of user and item interactions by providing it a list of items, users which is then trained to understand the similarity between items.
- This layer, along with a list of user ratings, is how to train the neural network.



## The Data

- The dataset which I manually generated contains 40 different cocktails and 200 ratings from 40 users. The way I visualized it was that there is a user for each cocktail which represents the users favourite one, imagined what other drinks that user may like or dislike and provided ratings based on this logic.
- This set isn't very large, so the accuracy would range from 60-70% which can be improved by doubling the ratings from each user which requires time.
- The data had to be stored in a csv file which cannot contain any null fields for processing by the python script.

	Cocktail ID	Rating
Vieux Carre	31	5
	2	4
	32	5
	33	5
	4	2
	Cocktail ID	Rating
Old Fashioned	1	5
	9	5
	14	4
	30	4
	24	1
	Cocktail ID	Rating
Mai Tai	10	5
	13	4
	18	5
	17	3
	11	3
	Cocktail ID	Rating
Pornstar Martini	28	5
	23	4
	14	4
	13	4
	9	3
	Cocktail ID	Rating
Gin Bramble	18	5
	6	4
	9	3
	31	1
	2	2
	Cocktail ID	Rating
Rosemary Fizz	13	5
	27	5
	30	3
	17	3
	18	4

cocktails.csv snippet

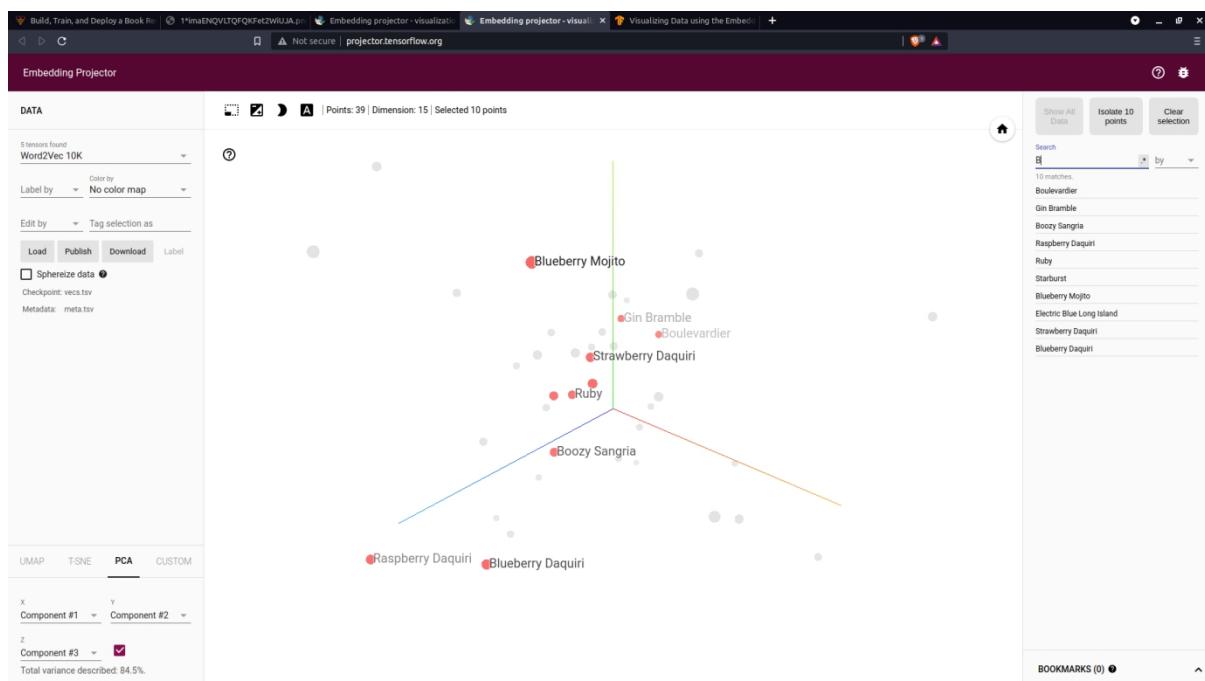
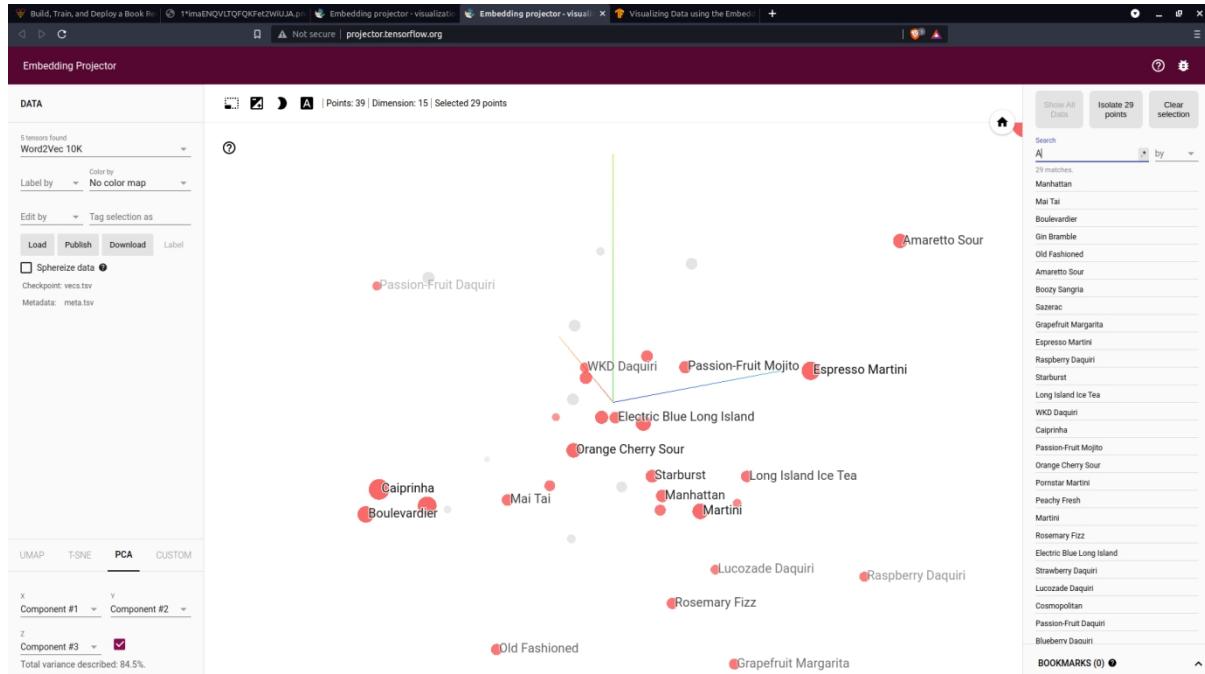
id	cocktail_id	name	ratings_coratings_1	ratings_2	ratings_3	ratings_4	ratings_5	image_url
1	1	Vieux Carr	8	0	2	1	3	2 https://rav
2	2	Old Fashic	10	1	2	0	6	1 https://rav
3	3	Mai Tai	4	0	0	2	1	1 https://rav
4	4	Pornstar M	11	0	2	2	2	5 https://rav
5	5	Gin Bramb	4	0	0	1	1	2 https://rav
6	6	Rosemary	6	0	0	3	3	0 https://rav
7	7	Whiskey S	2	2	0	0	0	0 https://rav
8	8	New York !	5	0	0	1	3	1 https://rav
9	9	Orange Cr	8	0	0	3	3	2 https://rav
10	10	Amaretto S	6	0	1	0	2	3 https://rav
11	11	Grapefruit	10	0	1	3	5	1 https://rav
12	12	Starburst	9	0	0	0	3	6 https://rav
13	13	Peachy Fr	10	0	0	1	5	4 https://rav
14	14	Espresso M	5	0	0	1	3	1 https://rav
15	15	Boozy San	2	0	0	0	0	2 https://rav

ratings.csv snippet

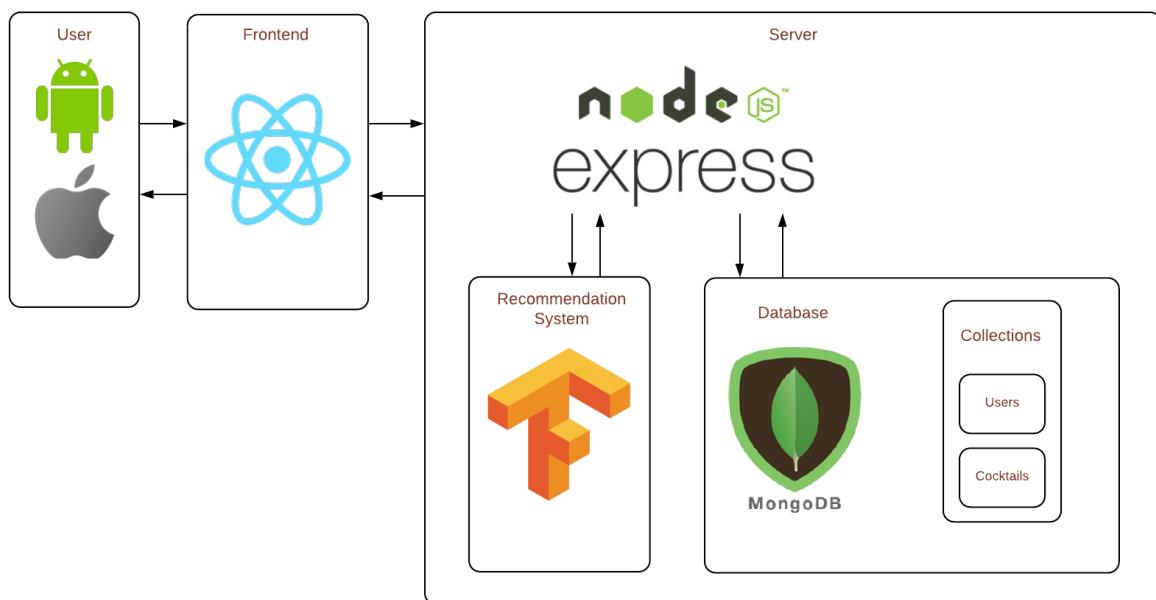
cocktail_id	user_id	rating
31	1	5
2	1	4
32	1	5
33	1	5
4	1	2
1	2	5
9	2	5
14	2	4
30	2	4

## TensorFlow Embedding Projector

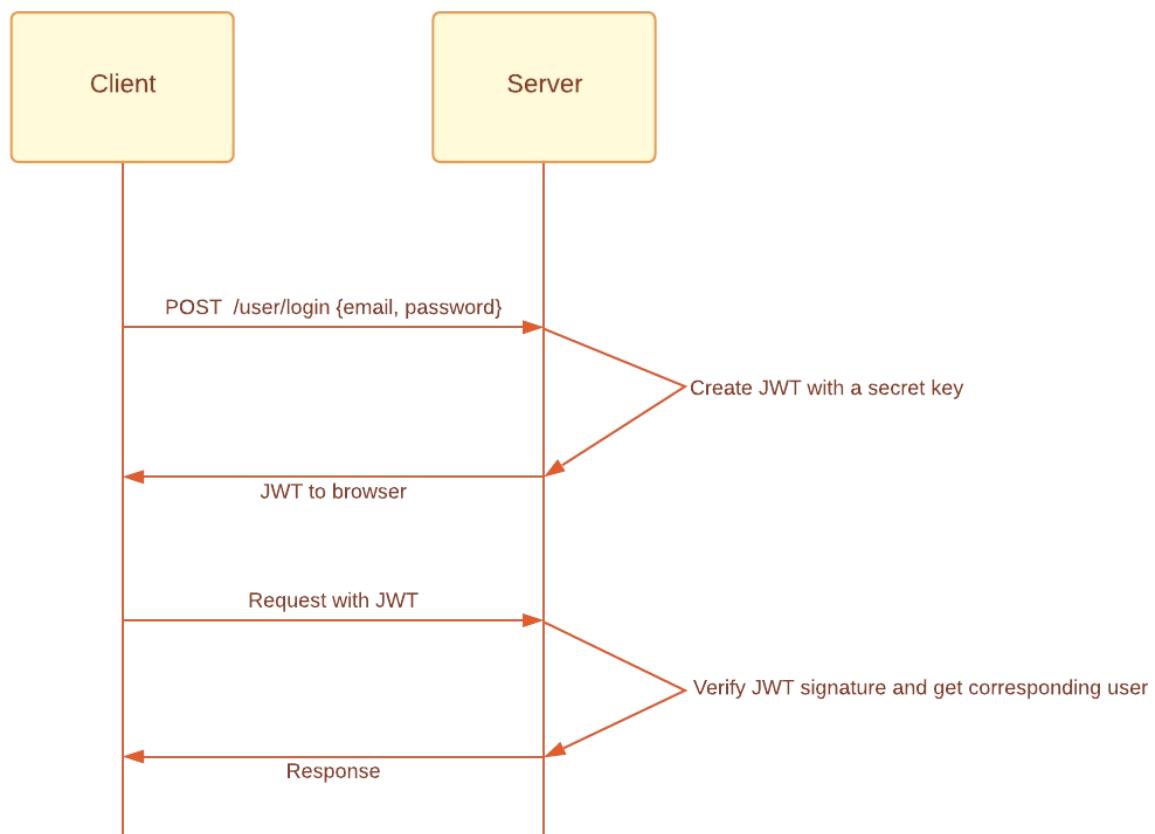
- Once the model has been trained and saved, you can download files which are then uploaded to the TensorFlow Embedding Projector to visualize how the cocktails cluster in the embedding.
- The closer the dots are, the more similar they are.



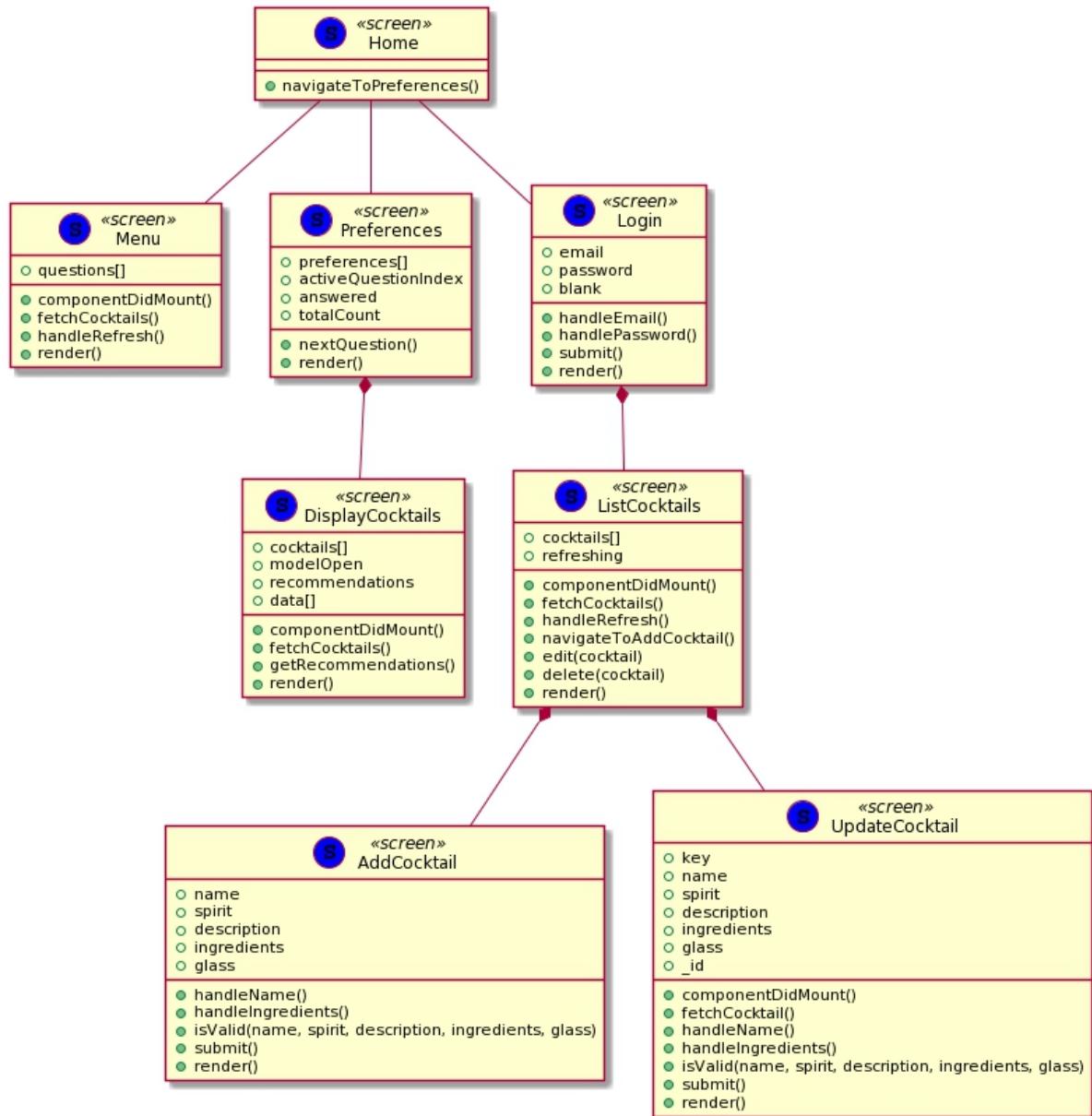
## System Architecture



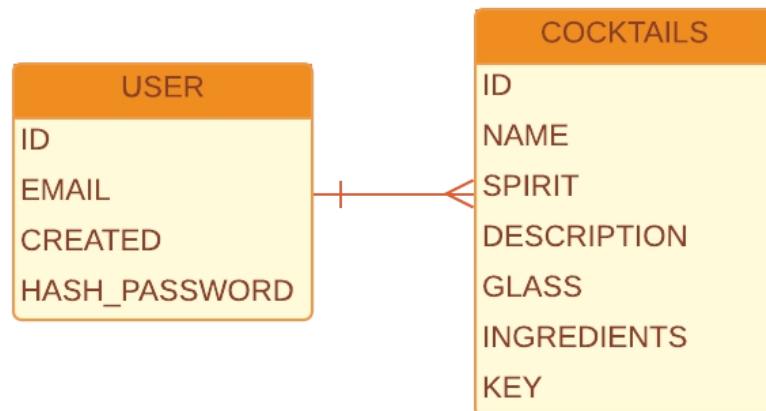
## JWT Authentication Sequence Diagram



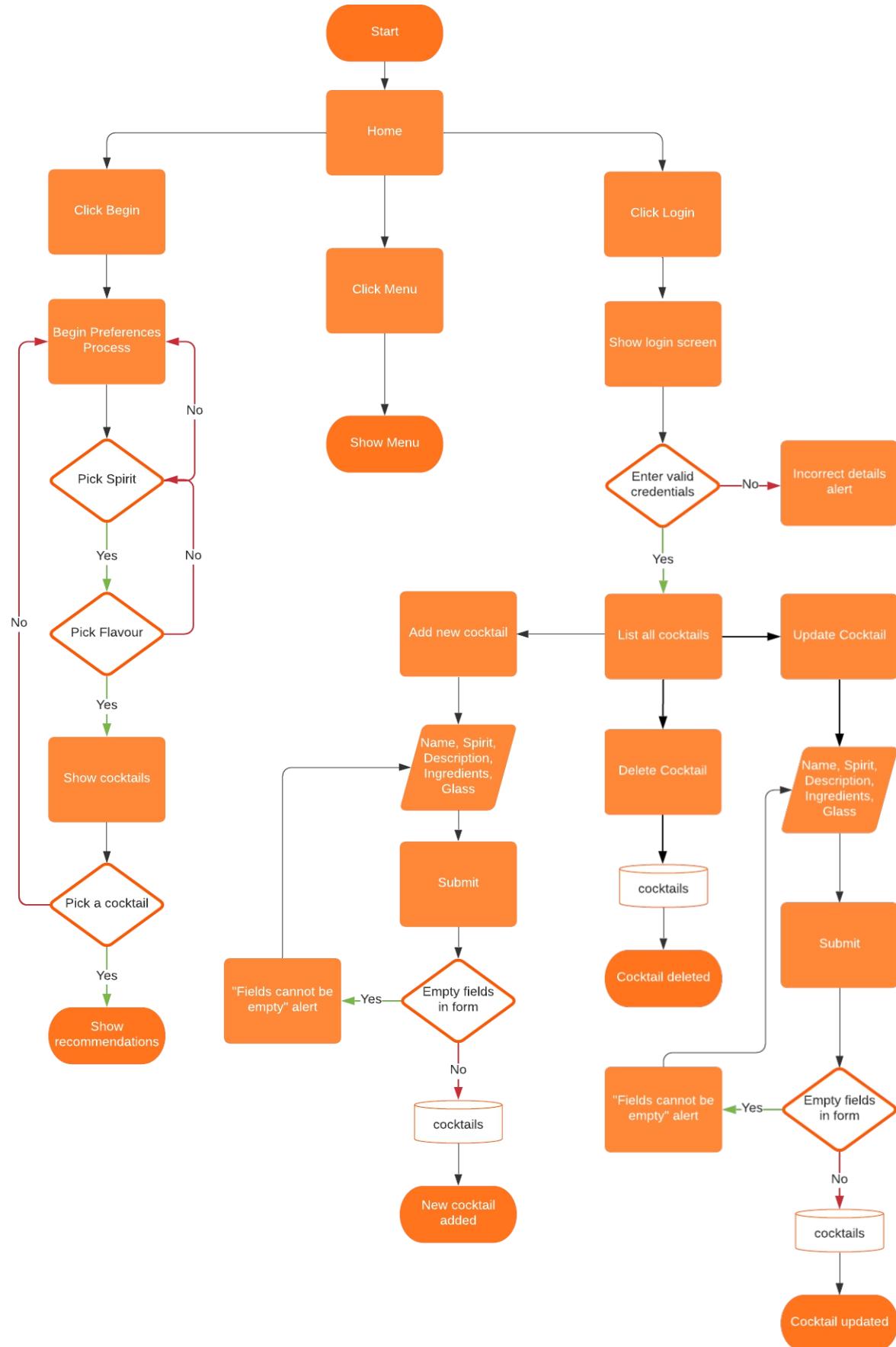
## Class Diagram



## Database Design



## Flowchart



## 2.5. Graphical User Interface (GUI)

Page	Screenshot
Landing/Home	

### Preferences

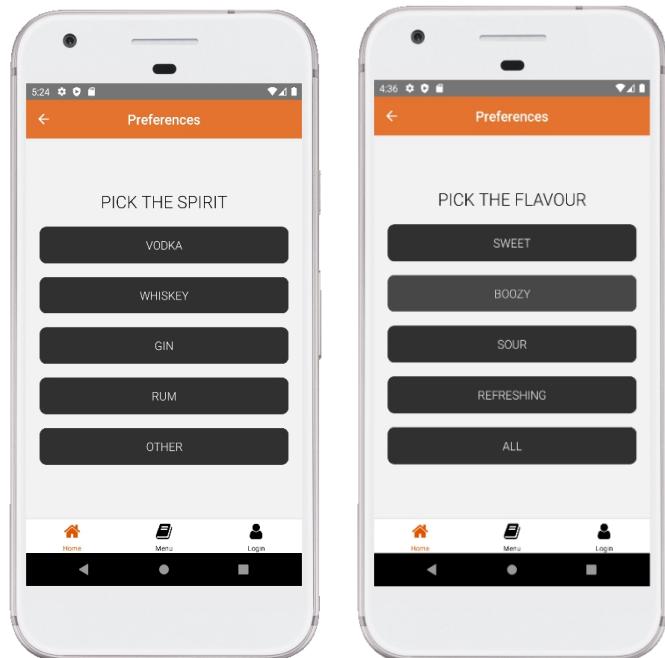
This is where the user can choose which spirit they want in their cocktail.

The page contains 5 buttons which are mapped to a JSON file containing the data

what question is being asked.

The first question is about the spirit, with Vodka, Whiskey, Gin, Rum and Other being the options the user can choose from.

The second question is where the user picks Sweet, Boozy, Sour, Refreshing or All cocktails.



## Display Cocktails

This is where the cocktails based on the chosen preferences are presented to the user.

The cocktails are displayed in a list with the name, ingredients and a sketch of the glass that they are served in.

Each sketch was hand drawn by a friend of mine who is an artist especially for this application.

The items can be clicked on to produce recommendations based on the chosen cocktail.

The user may begin the preferences process again by clicking the 'go again' button.

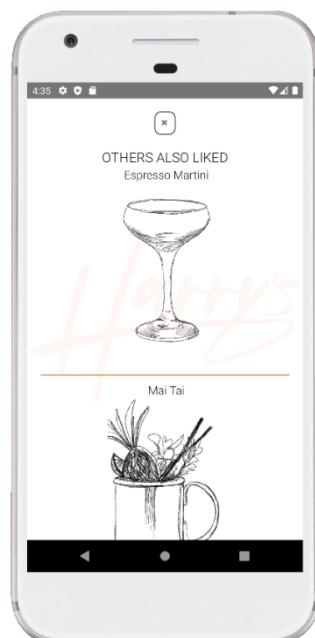
## Recommendations

This screen here is a popup modal which is displayed after a user clicks on a cocktail they like.

This is where the use of the TensorFlow model is introduced to the application.

Recommendations are based on users and their ratings where each user has a unique identifier with a corresponding cocktail identifier which is used to train the model to get an idea of what other cocktails that user may like after choosing their preferred cocktail.

This logic will be explained further in much more detail below.

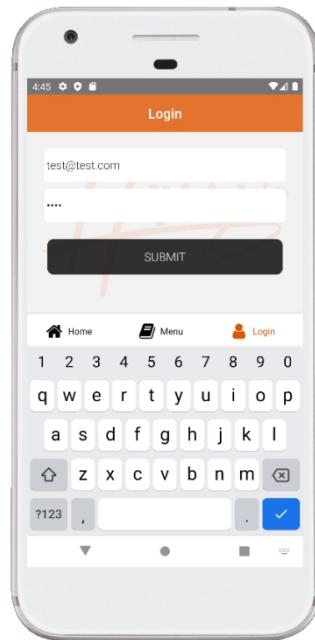


## Login

The login screen is where the staff will come to gain access to modifying the data in the application.

The credentials have been predefined and will be provided to the staff members to perform CRUD functionality on each cocktail item in the database.

An error alert will be displayed to the user if the credentials are incorrect.



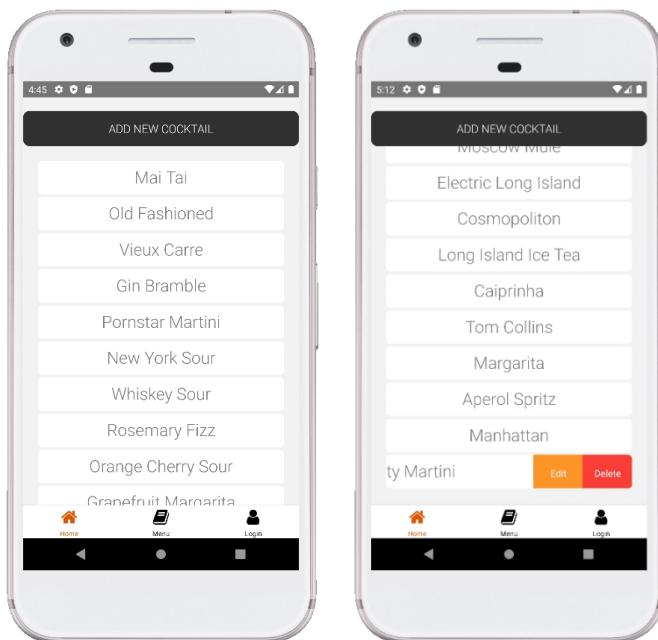
## List Cocktails

This is the screen that will be displayed if the user logs in to the application successfully.

It provides a list of all the cocktails in the database. Sliding left on any of the cocktails will show two buttons, edit and delete.

Pressing edit will direct the user to a form where they can make changes to the cocktail, while delete will then remove it from the database entirely.

The 'Add New Cocktail' button at the top of the page will allow the user to create a new cocktail record for the database.



## Add Cocktail

When a staff member is logged into the application successfully, they are provided with functionality that allows them to add new cocktails to the menu.

This screen is a form with a number of text input boxes and dropdown lists with the information required for the preferences process.

Each box must contain a value which is checked when pressing the submit button.

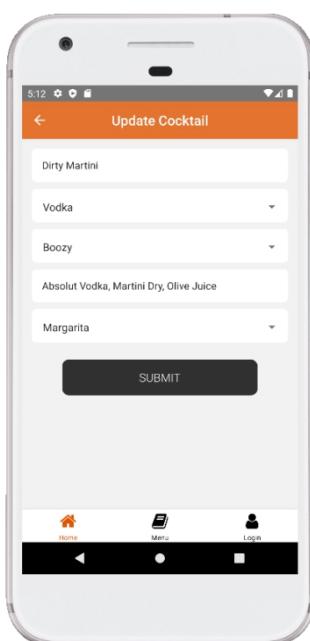
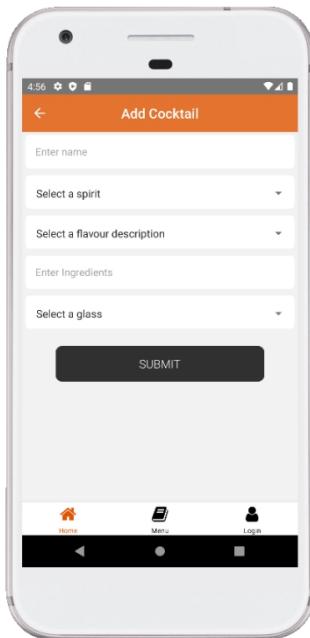
Pressing submit will also add the new item to the database.

## Update Cocktail

This page is very similar in design and functionality to the 'Add Cocktail' page.

The main difference is that it will pull the item that is being updated from the database and populate the form with the data so the user doesn't have to fill it out fully again.

The data is also validated on submit, and then updates the record in the database.



## Menu

The Menu screen is just a basic layout of all the cocktails on the menu for the user to browse on the app without having to narrow down through the preferences process.

I thought it would be necessary to include this screen due to the recommendations popup not listing the ingredients in the cocktails which was not included when training the model.



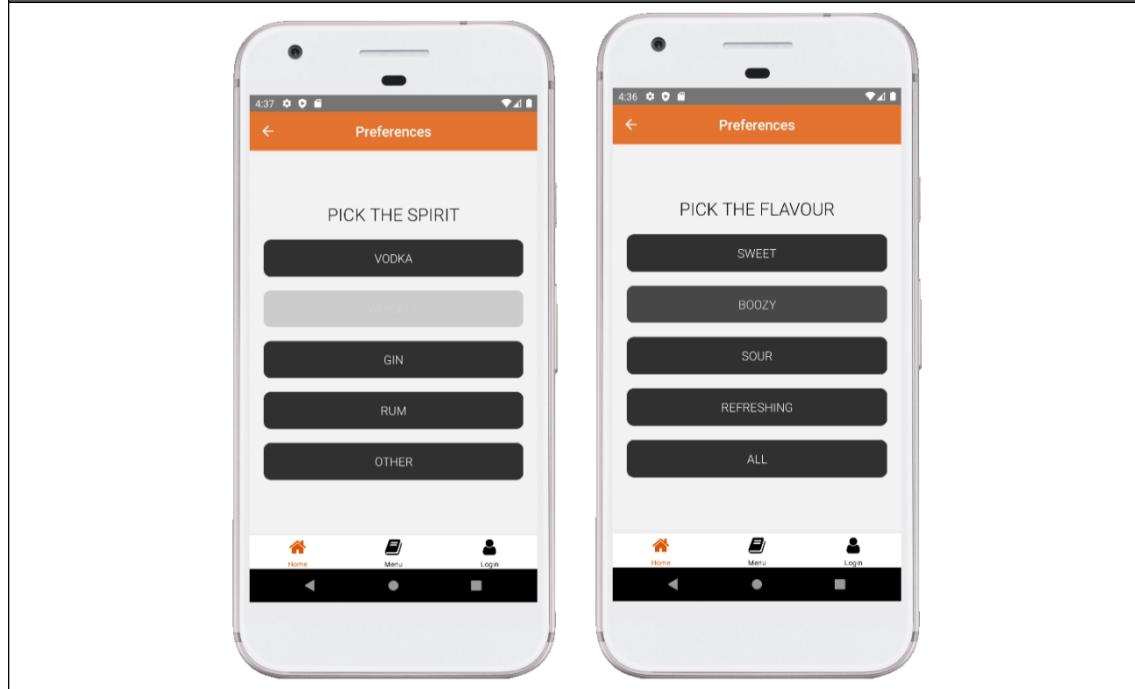
## 2.6. Testing

### Black Box

#### Normal Flow

ID	Description	Expected Result	Actual Result
1	Selecting preferred type of alcohol	Pass	Pass
No.	Steps		
1	User is presented with the 5 buttons		
2	User is presented with next set of button after clicking on 'Whiskey'		

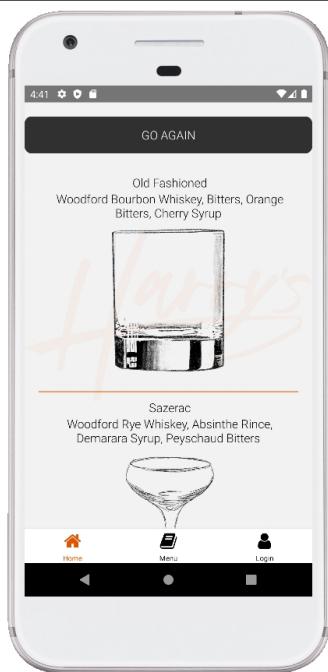
#### Screenshots



ID	Description	Expected Result	Actual Result		
2	Selecting preferred flavour	Pass	Pass		
No.	Steps				
1	User is presented with the 5 buttons				
2	User is redirected to the list of suggested cocktails				
Screenshots					
 					

ID	Description	Expected Result	Actual Result
3	Show cocktails based on preferences	Pass	Pass
No.	Steps		
1	Cocktails that contain whiskey and are boozy, e.g. Old Fashioned, is presented to the user		
2	User can click on 'Old Fashioned'		

### Screenshots



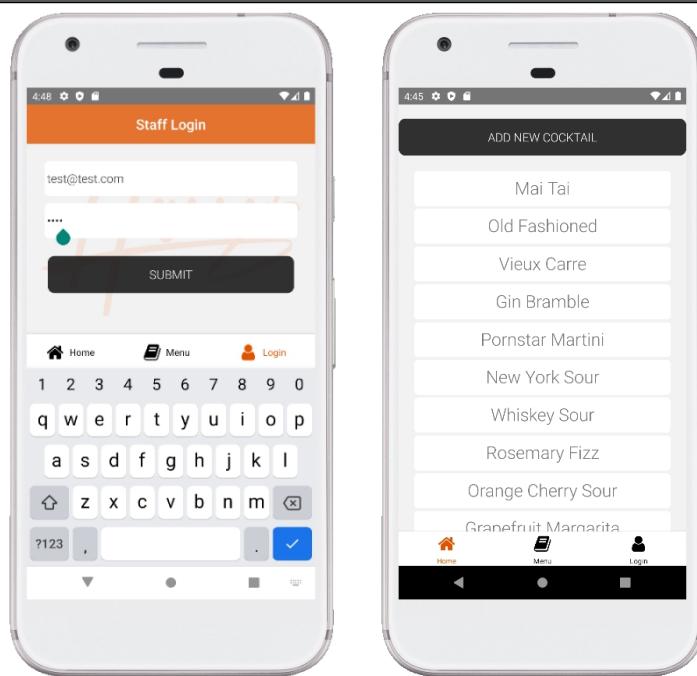
ID	Description	Expected Result	Actual Result
4	Show recommended cocktails	Pass	Pass
No.	Steps		
1	Once a user has clicked on 'Old Fashioned', they are presented with 3 recommendations		

### Screenshots



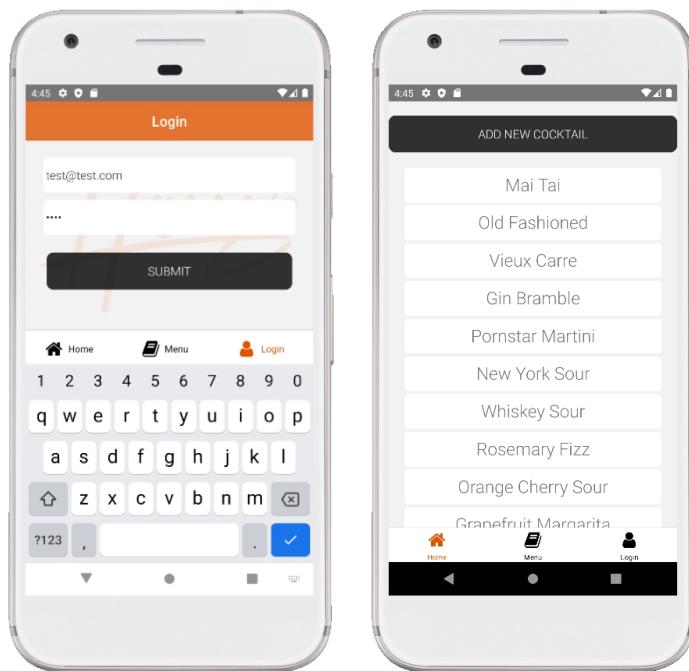
ID	Description	Expected Result	Actual Result
5	Login to the system	Pass	Pass
No.	Steps		
1	User enters <a href="mailto:test@test.com">test@test.com</a> into the email field		
2	User enters test into the password field		
3	User is successfully logged in and redirected to the next page		

### Screenshots



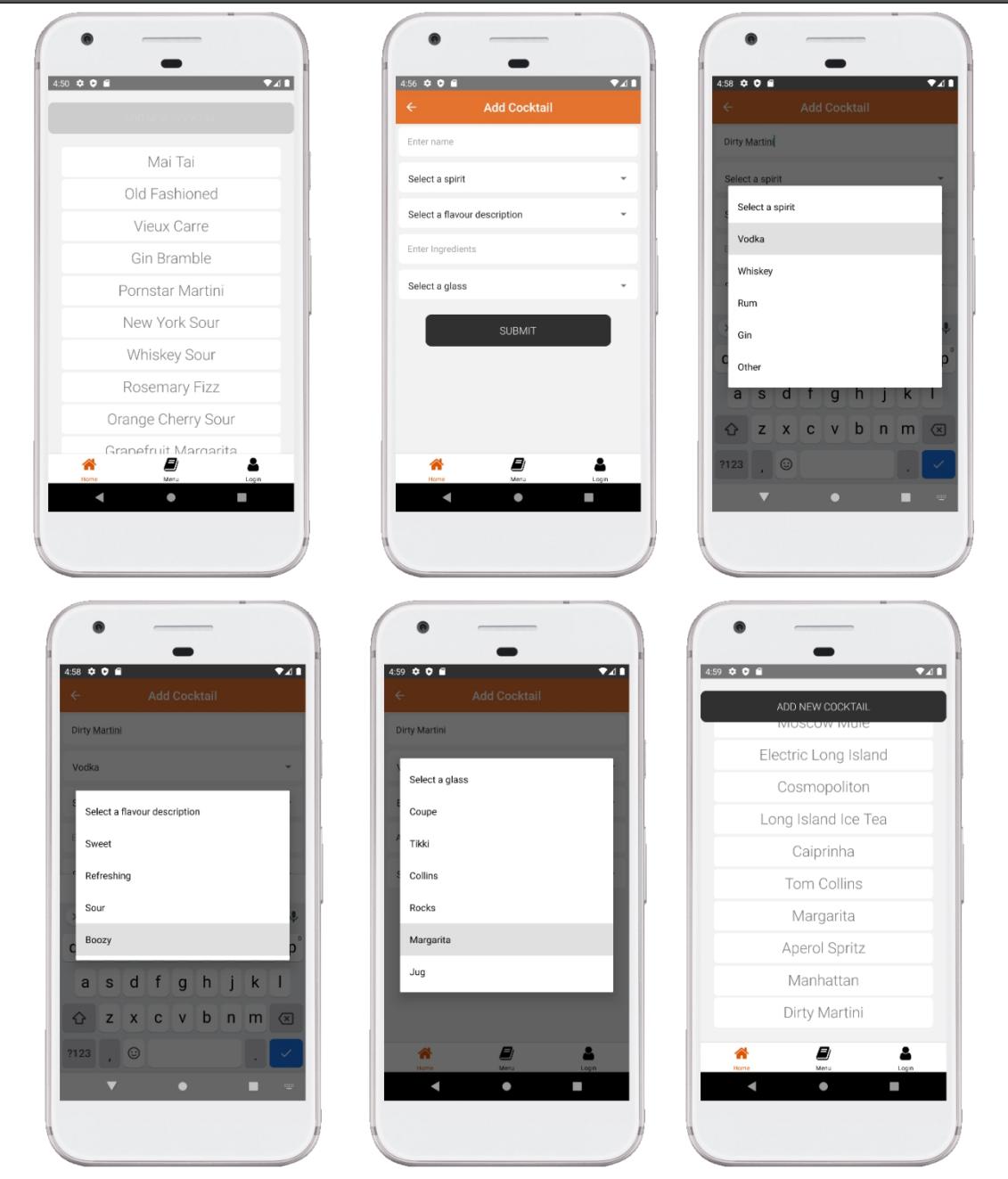
ID	Description	Expected Result	Actual Result
6	Show cocktail list	Pass	Pass
No.	Steps		
1	Logged in user is displayed a list of all cocktails on the menu		

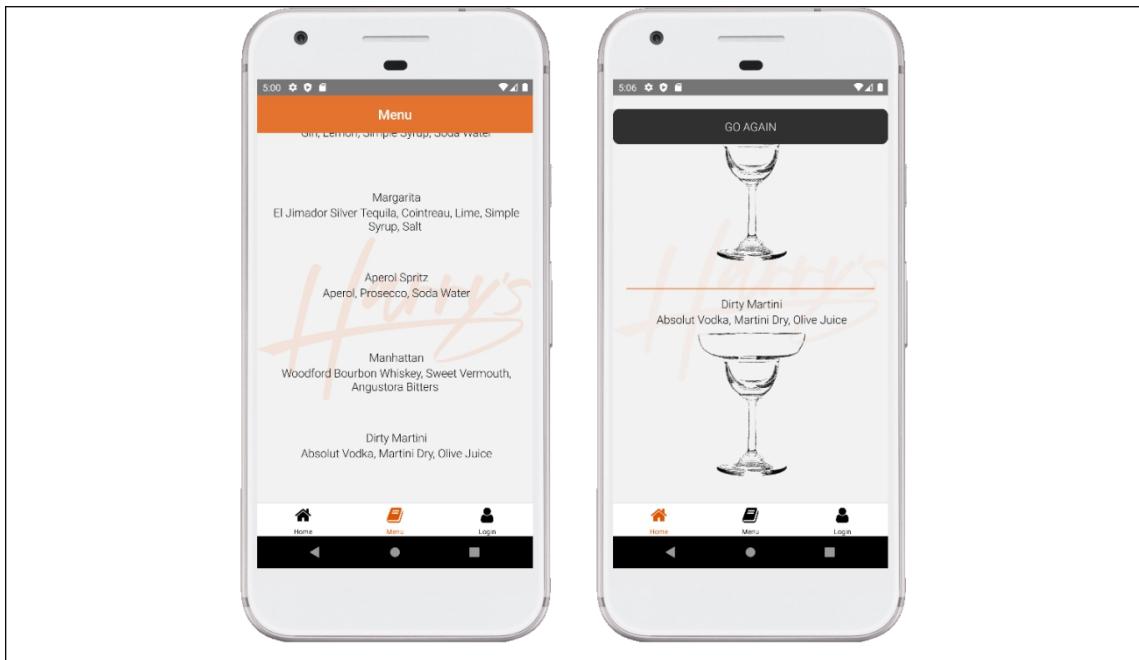
### Screenshots



ID	Description	Expected Result	Actual Result
7	Add a new cocktail to the menu	Pass	Pass
No.	Steps		
1	When user clicks on 'Add cocktail', they are presented a form to enter the data		
2	User fills each input box and can submit the data		
3	The new cocktail has been successfully added to the list and the user is redirected to an updated list of all cocktails		
4	The new cocktail can also be seen on the cocktail menu tab		

### Screenshots





ID	Description	Expected Result	Actual Result
8	Update existing cocktail	Pass	Fail
No.	Steps		
1	When slides an item in the cocktail list to the left, they are presented with an 'Edit' and 'Delete' buttons.		
	User clicks on 'Edit' button on 'Boozy Sangria'		
2	User is presented a form to enter the new data		
3	User changes 'Lemon' to 'Lime' and the glass from 'Jug' to 'Margarita'		
4	Cocktail has been successfully added to the list and the user is redirected to an updated list of all cocktails		
Screenshots			

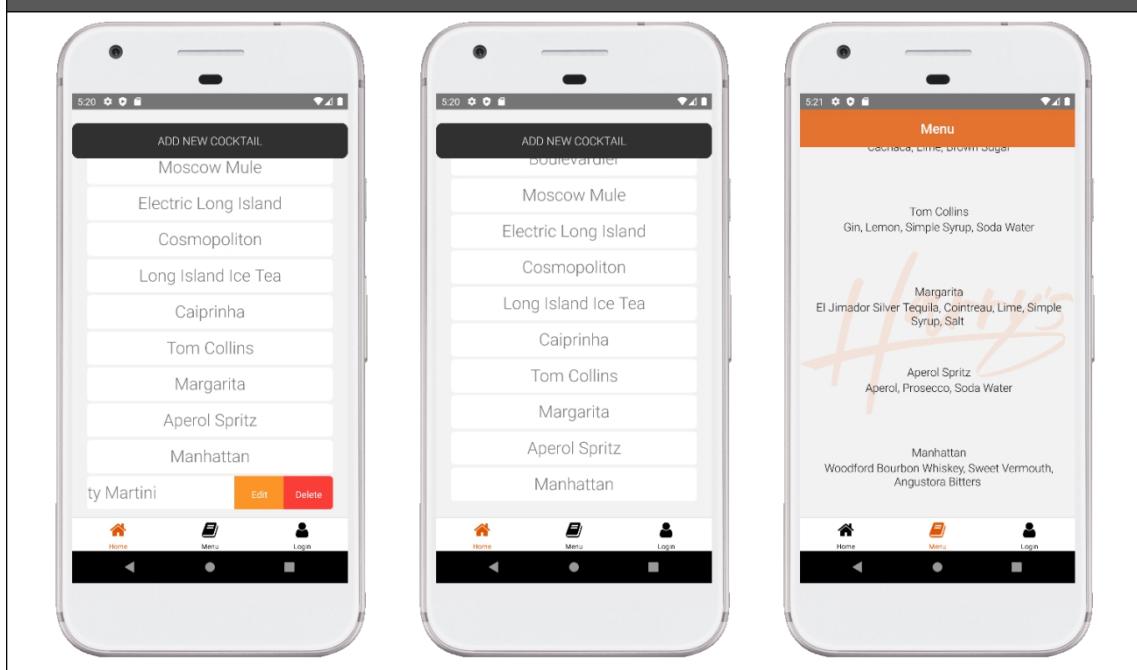


## Comments

The test passes on steps 1 and 2, but I have labeled this test case as a fail due to the list not refreshing automatically after the changes are submitted. However, the user can pull down on the screen to refresh the page after modifications are made as a way around this issue which is logged here: <https://github.com/michellelally/cocktail-app/issues/10>

ID	Description	Expected Result	Actual Result
9	Delete existing cocktail	Pass	Pass
No.	Steps		
1	When slides an item in the cocktail list to the left, they are presented with an 'Edit' and 'Delete' buttons		
2	User clicks on 'Delete' button		
2	Cocktail has been successfully deleted from the list and the user is redirected to an updated list of all cocktails		

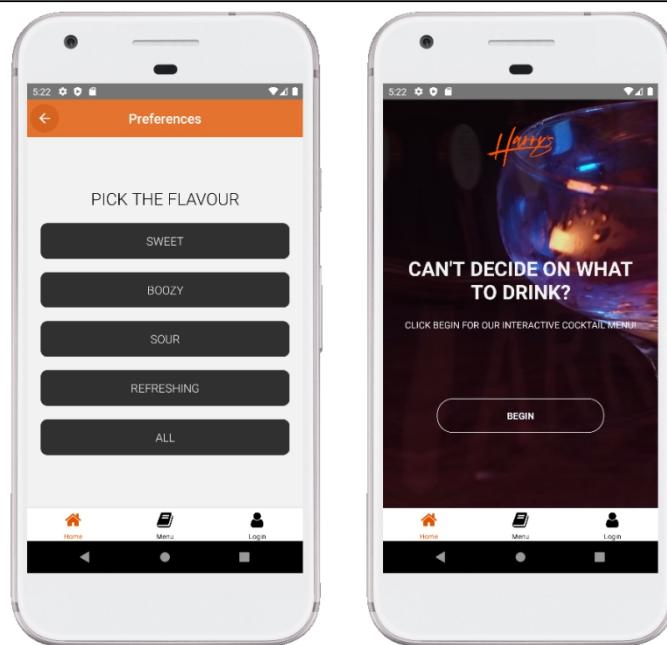
### Screenshots



## Alternate Flow

ID	Description	Expected Result	Actual Result
10	Changing spirit preference	Pass	Pass
No.	Steps		
1	User clicks on back button		
2	User is brought back to the home page to begin the preferences process again		

Screenshots



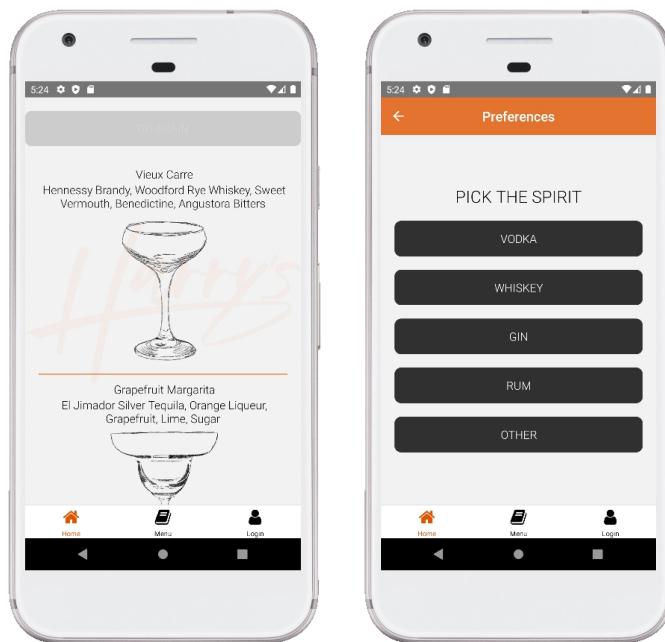
ID	Description	Expected Result	Actual Result
11	Changing spirit and flavour preference	Pass	Pass
No.	Steps		
1	User clicks on the 'go again' button		
2	User is brought back to the 'Pick your spirit' page to begin the preferences process again		

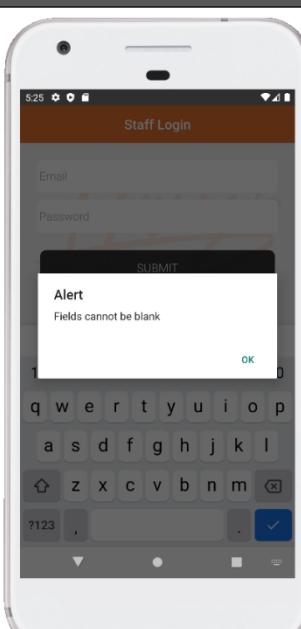
### Screenshots



ID	Description	Expected Result	Actual Result
12	User does not want to receive any recommendations	Pass	Pass
No.	Steps		
1	User clicks on back or 'go again' button		
2	User is brought back to the 'Pick your spirit' page to begin the preferences process again		

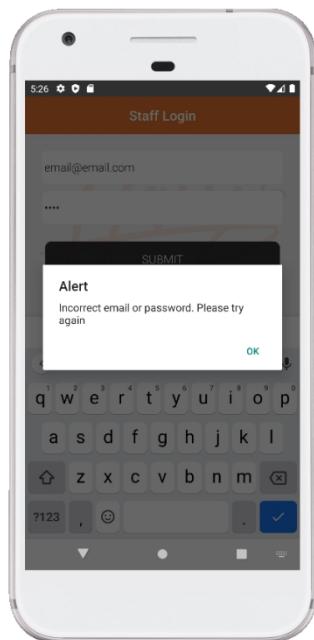
### Screenshots



ID	Description	Expected Result	Actual Result		
13	User attempts to submit empty email and password fields when logging in	Pass	Pass		
No.	Steps				
1	User clicks 'submit' button without entering any data				
2	User is alerted that fields cannot be empty				
Screenshots					
					

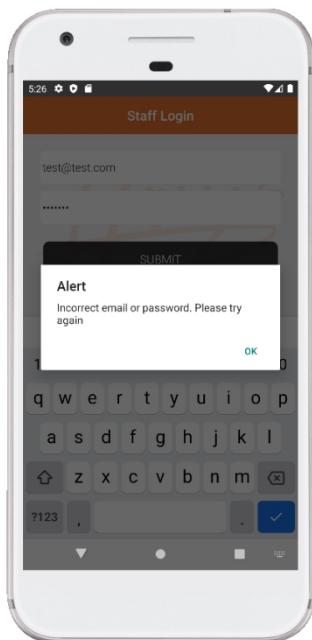
ID	Description	Expected Result	Actual Result
14	User enters incorrect email	Pass	Pass
No.	Steps		
1	User enters 'email@email.com'		
2	User is alerted that username or password is incorrect and to try again		

### Screenshots



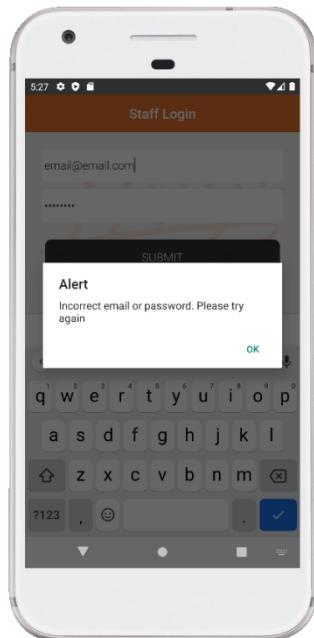
ID	Description	Expected Result	Actual Result
15	User enters incorrect password	Pass	Pass
No.	Steps		
1	User enters 'password'		
2	User is alerted that username or password is incorrect and to try again		

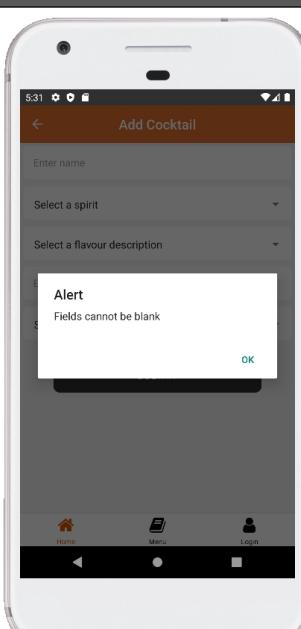
#### Screenshots



ID	Description	Expected Result	Actual Result
16	User enters incorrect email and password	Pass	Pass
No.	Steps		
1	User enters 'email@email.com' and 'password'		
2	User is alerted that username or password is incorrect and to try again		

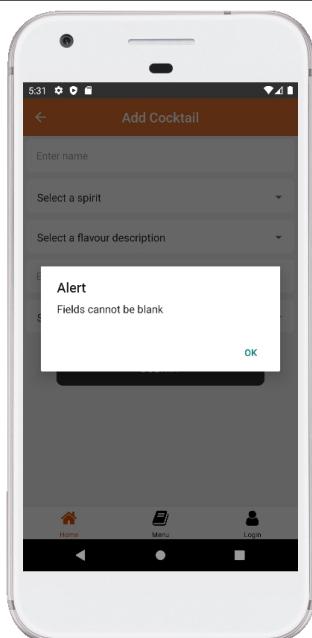
### Screenshots



ID	Description	Expected Result	Actual Result		
17	User attempts to submit empty fields when adding a new cocktail	Pass	Pass		
No.	Steps				
1	User clicks 'submit' button without entering any data				
2	User is alerted that fields cannot be empty				
Screenshots					
					

ID	Description	Expected Result	Actual Result
18	User attempts to submit empty fields when updating a cocktail	Pass	Pass
No.	Steps		
1	User clicks 'submit' button without entering any data		
2	User is alerted that fields cannot be empty		

### Screenshots



## White Box

### Normal Flow

ID	Description	Expected Result	Actual Result			
19	User is informed their chosen preferences returned no results	Pass	Fail			
No.	Steps					
1	User clicks on a spirit and flavour type but doesn't match anything in the database					
2	User is presented with a 'sorry, cocktails not found' item					
Screenshots						
						
Comments						
<p>The functionality to check if the dataset being returned from the database is empty, does not produce a 'cocktail not found' item as expected.</p> <p>The code to implement this logic has since been removed, but can be seen from line 67 onward in this version of the file:</p> <p><a href="https://github.com/michellelally/cocktail-app/blob/802fe24e8b8ac2165d6d9cc21109f79ef8836aa3/app/screens/DisplayCocktails.js">https://github.com/michellelally/cocktail-app/blob/802fe24e8b8ac2165d6d9cc21109f79ef8836aa3/app/screens/DisplayCocktails.js</a></p> <p>This code block would show the notFound data, for only a split second before it disappeared.</p> <p>The issue is logged here: <a href="https://github.com/michellelally/cocktail-app/issues/9">https://github.com/michellelally/cocktail-app/issues/9</a></p>						

ID	Description	Expected Result	Actual Result
20	User entering correct username and password does returns a JWT	Pass	Pass
No.	Steps		
1	User enters 'test@test.com' and test		
2	The bcrypt comparePassword method returns a 200 status and the generated JWT allowing the user to login securely		

## Screenshots

The screenshot shows the Postman interface with a successful API call. The request URL is `http://localhost:5000/auth/sign_in`. The request body contains:

```

{
  "email": "test@test.com",
  "password": "test"
}

```

The response is a 200 OK status with the following JSON payload:

```

{
  "token": "eyJhbGciOiJIUzI1NiIsInEiOiJzCl41IpxVVZ9...ay11mfpbC16iR1c3RAduVx05jb201LCJfaWQ1O112MaF1MaQ52aMyNzE00T142wH2YTRjNzH1LCo...0QyLChEyYyaZdBypsU0Qo2utBxqDp21cJU0h_LzeSA"
}

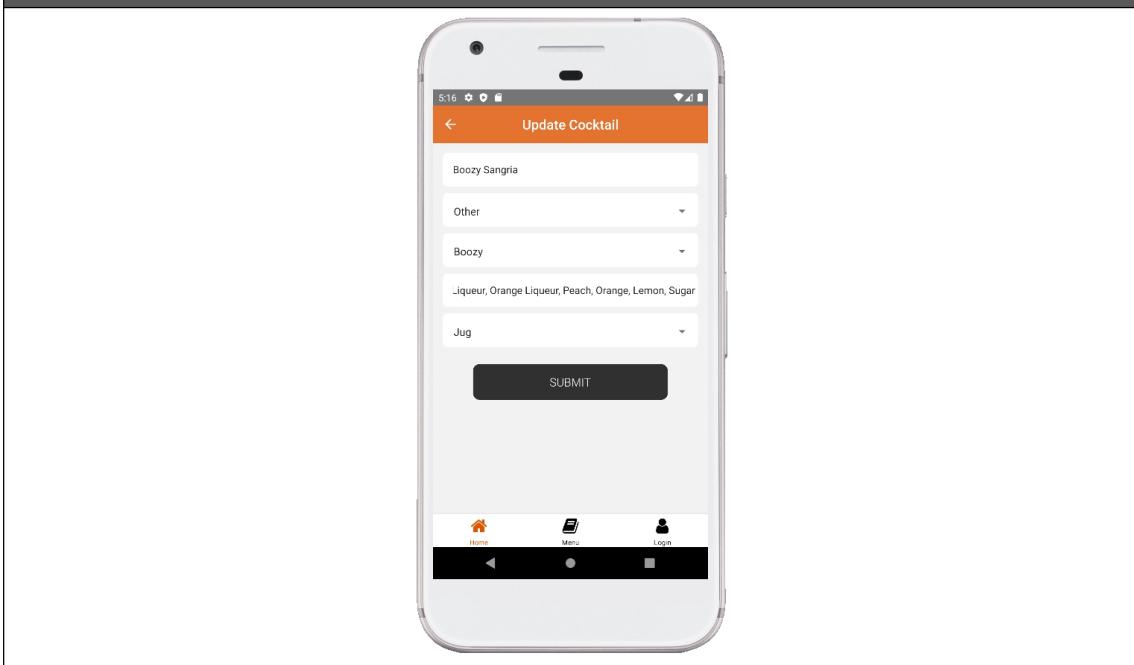
```

## Comments

Using Postman to check the status and response of the request when passing the correct credentials to the server and database

ID	Description	Expected Result	Actual Result
21	Update cocktail form auto-fills with the data from the database	Pass	Pass
No.	Steps		
1	When the user slides an item in the cocktail list to the left, they are presented with an 'Edit' and 'Delete' buttons		
2	User clicks on 'Edit' button		
3	System finds the cocktail in the database and returns the record		
3	User is presented with a populated form to save them from entering all the chosen cocktails data again		

### Screenshots



## Alternate Flow

ID	Description	Expected Result	Actual Result
22	User entering an incorrect password does not generate a JWT	Pass	Pass
No.	Steps		
1	User enters incorrect password		
2	The bcrypt comparePassword method returns a 401 status informing the system the password did not match and no JWT is generated		

### Screenshots

The screenshot shows the Postman application interface. A POST request is being made to `http://localhost:5000/auth/sign_in`. The request body contains two form-data fields: `email` with value `test@test.com` and `password` with value `wrongpassword`. The response status is 401 Unauthorized, with the message "Authentication failed. Invalid user or password." The response body is JSON:

```

1
2   "message": "Authentication failed. Invalid user or password."
3

```

ID	Description	Expected Result	Actual Result
23	Data with empty fields is not submitted to the database	Pass	Pass
No.	Steps		
1	User submits empty fields		
2	User is alerted the fields cannot be empty		
3	Database does not contain record that was attempted to be inserted		

### Screenshots

The screenshot on the left shows a mobile application interface for adding a cocktail. It has three input fields: 'Enter name', 'Select a spirit', and 'Select a flavour description'. A modal dialog box titled 'Alert' displays the message 'Fields cannot be blank'. The screenshot on the right shows a MongoDB Cloud interface with a list of cocktail documents. The first document is a Margarita cocktail with the following details:

```

{
  "_id": "61a79ed37399f83d8e1d0e0",
  "name": "margarita",
  "spirit": "tequila",
  "description": "margarita",
  "url": "https://raw.githubusercontent.com/michellelally/cocktail-app-primeapp/main/cocktails/margarita.json",
  "ingredients": "lime, tequila, simple syrup, soda water",
  "key": 38
}

```

The second document is a cocktail with the following details:

```

{
  "_id": "61a79ed37399f83d8e1d0e1",
  "name": "margarita",
  "spirit": "tequila",
  "description": "margarita",
  "url": "https://raw.githubusercontent.com/michellelally/cocktail-app-primeapp/main/cocktails/margarita.json",
  "ingredients": "lime, tequila, simple syrup, soda water",
  "key": 39
}

```

The third document is a cocktail with the following details:

```

{
  "_id": "61a79ed37399f83d8e1d0e2",
  "name": "aperol spritz",
  "spirit": "aperol",
  "description": "refreshing",
  "url": "https://raw.githubusercontent.com/michellelally/cocktail-app-primeapp/main/cocktails/aperol-spritz.json",
  "ingredients": "aperol, prosecco, soda water",
  "key": 37
}

```

### 3.0 Conclusions

If you have read my Interim Report, You may have noticed that I previously decided to use Firebase for my applications database. I made this change for a number of reasons, one being that the code for all database operations was very similar to the project I had submitted for a past module. Another reason was I had touched on MongoDB in the past and remember finding it easy to learn and implement and was looking to refresh my memory on it.

I am thrilled with my choice to make the switch. The more research I did into it and the more I learned about it, it made me appreciate the developers even more and I have most definitely learned some valuable skills and practices that will benefit me hugely in the future.

Another aspect of this project which I was very excited about was learning how to use TensorFlow, on my own, for the first time and I am very proud of myself for achieving the goal I set, as it wasn't all that easy. I still have so much to learn on the topic and I look forward to it thoroughly.

The recommendation system I built is not perfect, but it is still working with about 60-70% accuracy. Obviously there is a lot of room for improvement there, however, it is a lengthy process of generating the data, training the model and deploying it. I would need to double the amount of reviews I have provided for this application, which I didn't have the time for in the end but I will definitely continue working on.

My applications biggest flaw that I can think of, or know of, is that the re-training of the model process isn't automated. This means that any new additions to the menu, will not become a recommendation until I provide new data to the model, update a lot of the existing data, re-train the model and then provide the new model to the application.

Other than major UI improvements, I have found it hard to think of what changes I would like to make to this application. I am very pleased with how it all came together. I enjoyed learning and implementing the technologies I used and I can't imagine what else I would have done better.

The design is quite basic but I think it is effective. It looks clean and straightforward, rather than being full of colours that may not be too easy on the eyes. I would love to have more pages like the home page with a short clip as the background, but it is demanding to run the application on the simulator when on the home page and I would prefer a basic design over a slow application.

I really do believe I have a useful product that I could pitch to other bars and have them become potential customers. I have spoken to many people who said they would love an application like this, so I hope to develop some others in the future.

## 4.0 Further Development or Research

My main priority of future development would be the designing the application with more colour and animations. React Native provides such a selection of components and I only got to experiment with a fraction of them.

As I mentioned above I would love more short clips, or even some images as backgrounds, animated loading screens, a wider colour palette, and much more.

As I mentioned above in the Design and Architecture section, I used the collaborative filtering method, however, I would be interested in trying and testing some of the different types of filtering for the TensorFlow model to see which is most suited to the app as I just used the most convenient and easy to use as my first time experimenting with the library.

Another step I would like to take in the future is gather data from the user about the recommendations they were given, if they were accurate, or even give their own recommendations. This data could then be used to re-train the model which would remove the need for me manually generate the data and provide it to Tensorflow and Keras.

I haven't done any research into automating the process of training the model, but it would be extremely useful here, for the reasons also mentioned in the conclusion.

Once some of these updates are completed, I would love to upload it to the appstore for Androids and iOS mobile phones. It would be such an accomplishment to have an application that I'm really proud of making, online for everyone to download.

If this was to ever become a reality, I would look into ways of monetizing it by implementing a few ads here and there, but nothing too big that might interfere with the users experience.

## 5.0 References

Sharma, A., 2020. *NodeJS and MongoDB application authentication by JWT*. [online] Loginradius.com. Available at: <<https://www.loginradius.com/blog/async/Nodejs-and-MongoDb-application-authentication-by-JWT/>>

Saruk, U., 2019. *What is JWT and why to use it ? Simplified !*. [online] Medium. Available at: <<https://medium.com/@umeshsaruk/what-is-jwt-and-why-to-use-it-simplified-97f8ac19c332>>

Bezkoder. 2021. *React.js + Node.js + Express + MongoDB example: MERN stack CRUD App - BezKoder*. [online] Available at: <<https://www.bezkoder.com/react-node-express-mongodb-mern-stack/>>

Odeguia, R., 2020. *Build, Train, and Deploy a Book Recommender System Using Keras, TensorFlow.js*,. [online] Medium. Available at: <<https://heartbeat.comet.ml/build-train-and-deploy-a-book-recommender-system-using-keras-tensorflow-js-b96944b936a71>>

## 6.0 Appendices

### Github Repository Link

<https://github.com/michellelally/cocktail-app>

### Github Issues

Issues · michellelally/cocktail-app - Mozilla Firefox

michellelally / cocktail-app Public

Code Issues Pull requests Actions Projects Wiki Security

Label issues and pull requests for new contributors  
Now, GitHub will help potential first-time contributors discover issues labeled with [good first issue](#)

Filters is:issue is:closed Labels Milestones New issue

Clear current search query, filters, and sorts

3 Open 7 Closed

Author	Label	Projects	Milestones	Assignee	Sort	Count
	Back navigation on page displaying suggestions to user	#8 by michellelally was closed 5 days ago				1
	List page is not displaying last element in array	#6 by michellelally was closed 5 days ago				1
	Refresh after deleting cocktail from the database	#5 by michellelally was closed 6 days ago				1
	Navigation on preferences page	#4 by michellelally was closed 5 days ago				2
	undefined cocktails being returned from database	#3 by michellelally was closed 11 days ago				1
	Picker (dropdown list) Component not passing an initial value	#2 by michellelally was closed 11 days ago				1
	Network request failed	#1 by michellelally was closed 25 days ago				1

ProTip! What's not been updated in a month: [updated<2021-11-13](#).

Issues · michellelally/cocktail-app - Mozilla Firefox

michellelally / cocktail-app Public

Code Issues Pull requests Marketplace Explore

Label issues and pull requests for new contributors  
Now, GitHub will help potential first-time contributors discover issues labeled with [good first issue](#)

Filters is:open is:issue Labels Milestones New issue

Clear current search query, filters, and sorts

3 Open 7 Closed

Author	Label	Projects	Milestones	Assignee	Sort	Count
	Page not refreshing after updates	#10 opened 3 days ago by michellelally				1
	Cocktail not found not working	#9 opened 4 days ago by michellelally				1
	Tensorflow model is returning duplicate data	#7 opened 6 days ago by michellelally				1

ProTip! Type [@](#) on any issue or pull request to go back to the pull request listing page.

Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

## README.md

<https://github.com/michellelally/cocktail-app/blob/main/README.md>

## Python script

The code I used for building, training and saving the model

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import warnings

warnings.filterwarnings('ignore')
%matplotlib inline

import tensorflow.keras as tf

ratings_df = pd.read_csv('cocktail-data/ratings.csv')
cocktails_df = pd.read_csv('cocktail-data/cocktails.csv')
ratings_df.head()
cocktails_df.head()

print(ratings_df.shape)
print(ratings_df.user_id.nunique())
print(ratings_df.cocktail_id.nunique())
ratings_df.isna().sum()

from sklearn.model_selection import train_test_split
Xtrain, Xtest = train_test_split(ratings_df, test_size=0.4, random_state=1 )
print(f Shape of train data: {Xtrain.shape})
print(f Shape of test data: {Xtest.shape})

#Get the number of unique entities in cocktails and users columns
ncocktail_id = ratings_df.cocktail_id.nunique()
nuser_id = ratings_df.user_id.nunique()

input_cocktails = tf.layers.Input(shape=[1])
embed_cocktails = tf.layers.Embedding(ncocktail_id + 1, 15)(input_cocktails)
cocktails_out = tf.layers.Flatten()(embed_cocktails)

#user input network
input_users = tf.layers.Input(shape=[1])
```

```

embed_users = tf.layers.Embedding(nuser_id + 1, 15)(input_users)
users_out = tf.layers.Flatten()(embed_users)

conc_layer = tf.layers.concatenate([cocktails_out, users_out])
x = tf.layers.Dense(128, activation='relu')(conc_layer)
x_out = x = tf.layers.Dense(1, activation='relu')(x)
model = tf.Model([input_cocktails, input_users], x_out)

opt = tf.optimizers.Adam(lr=0.000000000001)
model.compile(optimizer=opt, loss='mean_squared_error')
model.summary()

hist = model.fit([Xtrain.cocktail_id, Xtrain.user_id], Xtrain.rating,
                 batch_size=1,
                 epochs=2000,
                 verbose=1,
                 validation_data=[Xtest.cocktail_id, Xtest.user_id], Xtest.rating))

train_loss = hist.history['loss']
val_loss = hist.history['val_loss']
plt.plot(train_loss, color='r', label='Train Loss')
plt.plot(val_loss, color='b', label='Validation Loss')
plt.title('Train and Validation Loss Curve')
plt.legend()
plt.show()

# Extract embeddings
cocktail_em = model.get_layer('embedding')
cocktail_em_weights = cocktail_em.get_weights()[0]
cocktail_em_weights.shape

cocktails_df_copy = cocktails_df.copy()
cocktails_df_copy = cocktails_df_copy.set_index('cocktail_id')

out_v = open('vecs.tsv', 'w')
out_m = open('meta.tsv', 'w')
for i in b_id:
    cocktail = dict_map[i]
    embeddings = cocktail_em_weights[i]
    out_m.write(cocktail + '\n')
    out_v.write('t' + ''.join([str(x) for x in embeddings]) + '\n')

```

```

out_v.close()
out_m.close()

#save the model
model.save('model')

cocktails_df_copy = cocktails_df.copy()
cocktails_df_copy = cocktails_df_copy.set_index('cocktail_id')

b_id = list(ratings_df.cocktail_id.unique())
b_id.remove(40)
dict_map = {}
for i in b_id:
    dict_map[i] = cocktails_df_copy.iloc[i]['name']

#Making recommendations for user 12
cocktail_arr = np.array(b_id) #get all cocktail IDs
user = np.array([36 for i in range(len(b_id))])
pred = model.predict([cocktail_arr, user])
pred

pred = pred.reshape(-1) #reshape to single dimension
pred_ids = (-pred).argsort()[0:5]
pred_ids

cocktails_df.iloc[pred_ids]

web_cocktail_data = cocktails_df[['id', 'name', 'image_url']]
web_cocktail_data = web_cocktail_data.sort_values('id')
web_cocktail_data.head()

web_cocktail_data.to_json(r'web_cocktail_data.json', orient='records')

```