Team Number: 015-1

Team Name: Seg Faults

Team Members: John Hau, Jack Gentlemen, Junyu Ding, Charlotte Donaldson, Ashley Weaver, Kent WaxmanWe

Application Name: Mixr

Application Description:
Mixr is an iOS and Android app created to help users create classic and unique drinks purely based on the ingredients at the user's disposal. Mixr also has the ability for users to upload their custom drinks for the community to see and try, or simply store the user's custom recipes for easy access.

With Mixr's abundant features, it is an application that stands out because it uses crowdsourcing to help expand the network and provide users with more fun drink options beyond the classic bartender drink options that are already available. Mixr offers the ability for users to upload their unique drink recipes, as well as get ratings from other users on the app regarding their unique concoctions. Based on these ratings, it determines the rank in which users see the drink when their search query remotely matches the ingredients of the drink.

With the ability to easily search for popular drink recipes, as well as the ability to upload and view custom drink recipes, Mixr serves as a platform for everyone to explore mixology.

Vision Statement:
For people twenty-one and older (primarily college students),
Who want to be a bartender for a day,
The Mixr app is a mixology platform,
That tells users what to make and how to make it based on ingredients they and also allows uploads of custom-made drinks,
Unlike "Make Me A Cocktail," [https://makemeacocktail.com/mybar/]
Our product allows users to upload and share their own recipes with other users. This ensures that the recipes in the app are unique and well liked, as they require at least one user to know about the recipe prior to uploading it.

Communication Plan:
Our group is currently using GroupMe as a communication app to communicate with each other. We created a group named CSCI 3308 in GroupMe which contains all members in it. It is a very

convenient app for us to use for communication because it notices us on the phone whenever one group member sends something in it.

Meeting Plan:
Our group meets every Tuesday at 5:00 p.m MT for 2 hours over zoom (could be less depending on what we are working on during the meeting). Meeting with TA is after Friday recitations, 1:15-1:35pm
Link: https://cuboulder.zoom.us/j/98799103086

Development Method:
For this project our method will be the agile/scrum development method. This methodology involves an emphasis on functional code and flexibility over deadlines and designs. For our group we will be using Jira to keep track of individual tasks and the overall development process. Every week we will meet and hold an informal scrum meeting in which we provide updates on which stories we worked on, as well as updates on which stories we will be working on within that week. The agile method allows for increased flexibility, so if certain hiccups are met within development changes can be made so circumvent these, as documentation does not need to be strictly adhered to.
Github Repo Link: https://github.com/CSCI-3308-CU-Boulder/3308SP21_015_1
Jira Board Link:
https://csci-3308-spring21-015-1.atlassian.net/jira/software/projects/S01/boards/1/backlog

Use Case Diagram:

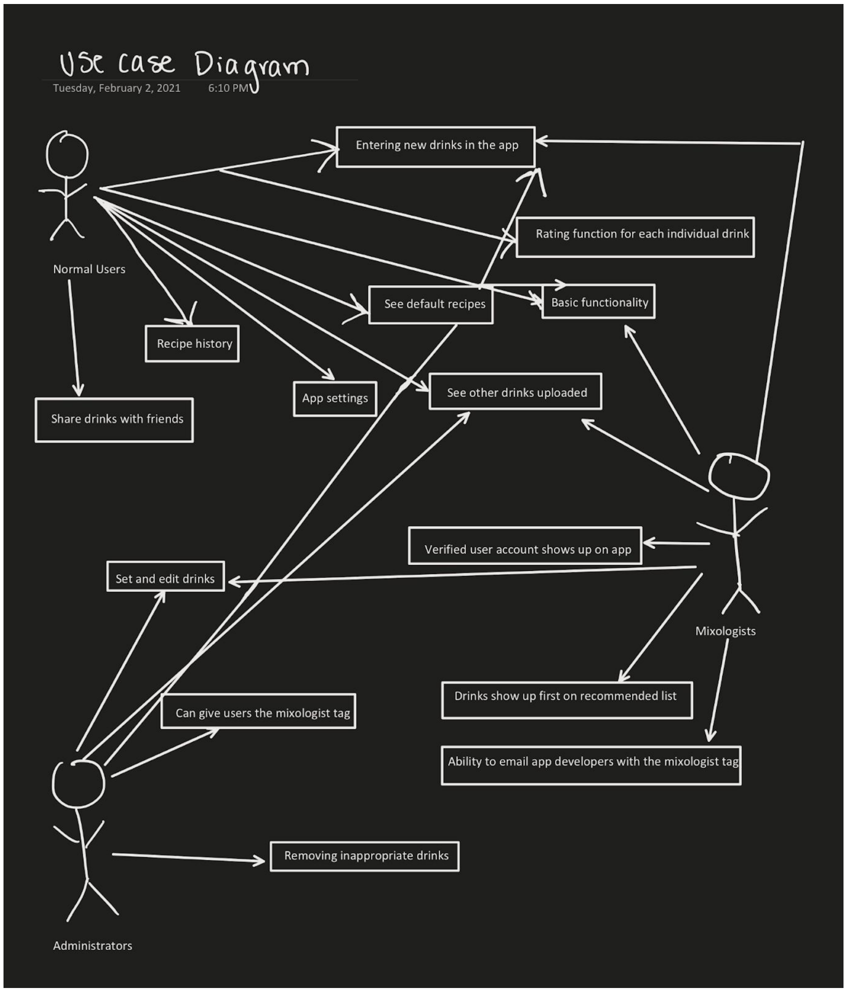
> Identify a minimum of 3 actors and 15 use cases for your project. Create and attach a use case diagram to depict them. You may change these at a later point in time as suited.

Actors:

1. Normal Users: College Students, People over the age of 21, anyone who has access to alcohol
    a. Basic functionality (check off what drinks/ingredients they have)
    b. Be able to make enter new drinks they make into the app
    c. Recipe history
    d. See default recipes (not added by users)
    e. Share drinks with their friends (send links)
    f. See other recipes uploaded by users
    g. App setting
    h. Recipe filter
    i. Rating function for each individual drink
2. Administrators: People just regulating the drinks that are being entered
    a. Administrator can come in and take off drinks that are inappropriate
    b. Administrators create the set drinks on the app and can edit pre existing drinks

c. Can give users the “Mixologist” tag, if earned (see below)

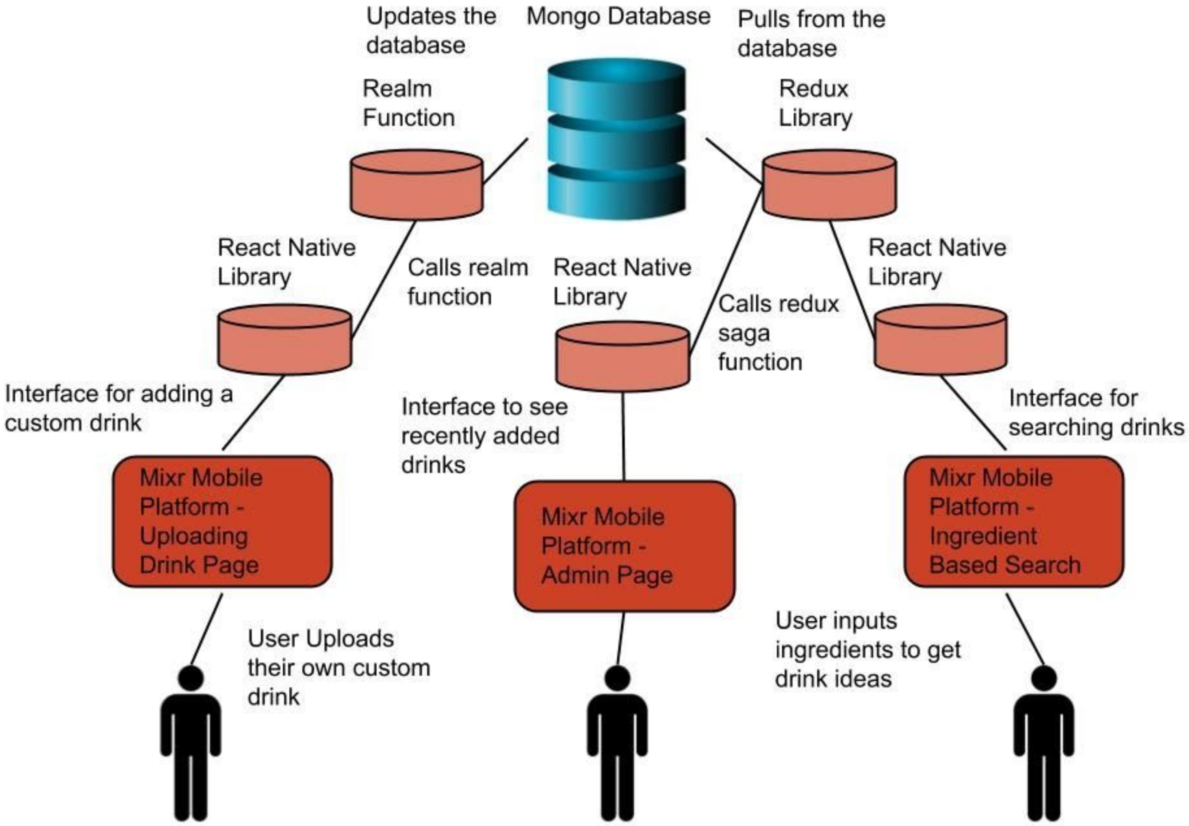
3. Mixologists (“Blue Check” like on Twitter)
    a. Basic functionality
    b. Verified user account, shows up on the app
    c. Drinks show up first in user recommended drinks
    d. Ability to email developers, showing credentials and receiving “Mixology” tag.
        i. If app becomes more popular, Mixologists may require a certain amount of followers to receive the tag
    e. Can also set and edit drinks (Not regulated first though)


Use case Diagram
Tuesday, February 2, 2021
6:10 PM
Entering new drinks in the app
Rating function for each individual drink
Normal Users
Basic functionality
See default recipes
Recipe history
App settings
See other drinks uploaded
Share drinks with friends
Verified user account shows up on app
Set and edit drinks
Mixologists
Drinks show up first on recommended list
Can give users the mixologist tag
Ability to email app developers with the mixologist tag
Removing inappropriate drinks
Administrators

(First basic use case diagram)

Proposed Architecture Plan:

The proposed architecture for this project will involve React Native for the front end, which mostly allows the app to be adapted between Android and iOS easily with the same code. The backend will use a Mongo database with Realm as the primary means for updating and altering the database from user interaction. The React Native or front end of the application will mainly involve a screen in which the user can input ingredients and certain drinks will be relayed back to the user based on those ingredients. On the back end, these drinks will be stored within a Mongo collection called drinks, which holds JSON files that contain a drink rating, drink ingredients, drink name, and possibly a username for who uploaded the drink. The front end will also have a screen for allowing the user to upload their own custom drinks. This will use Realm and specifically stitch functions to update the database and add such drinks, using the above listed criteria for each JSON file. The front end will also include some interface to allow an admin to view recently added drinks being pulled from the database collection drinks. Last, the app will have a screen for user authentication to allow for users to create an account and login, such that user interactions with the database and previous interactions can be updated to the user at login. This all will be stored in a user collection that maintains basic user information, including a username and password, as well as their favorite drinks and recently accessed drinks.


Updates the database
Mongo Database
Pulls from the database
Realm Function
Redux Library
React Native Library
Calls realm function
React Native Library
Calls redux saga function
React Native Library
Interface for adding a custom drink
Interface to see recently added drinks
Interface for searching drinks
Mixr Mobile Platform - Uploading Drink Page
Mixr Mobile Platform - Admin Page
Mixr Mobile Platform - Ingredient Based Search
User Uploads their own custom drink
User inputs ingredients to get drink ideas