TEAM 19

Drinks with Friends

Tim Buss - busst@purdue.edu

Rodrigo Gerbasi - rgerbasi@purdue.edu

Nick Leuer - nleuer@purdue.edu

Paul Miller - mill2136@purdue.edu

E.J. Wennerberg - ewennerb@purdue.edu

Index

Purpose

Introduction

Everyone has their favorite drinks, and convincing yourself to try new drinks and stepping out of your comfort zone is very difficult. Adding on to this, you can save yourself the embarrassment of approaching a bar and being the person holding up the line trying to figure out what drink looks good or you can solve the issue of trying to make a drink at home with what little ingredients are left, avoiding a late night store run.

These problems can now be a worry of the past with our web app Drinks with Friends. Drinks With Friends will allow you to quickly come up with the name of a new drink, see your friends' favorite drinks, and also be able to see the menu of businesses such as bars or restaurants. By liking and disliking drinks from the database, our algorithm will be able to help suggest drinks you have never heard of, but might become your new favorite.To ease the process of choosing a drink even further, our web app allows users to let accounts pin one all time favorite drink to their profile.

There are some web apps that exist that have the same founding idea as Drinks with Friends. What distinguishes Drinks with Friends from others is our easy to use, clean user interface, and the implementation of a social media aspect to the web app. Other sites like The Webtender or CocktailDB are simply databases of established cocktails and mixed drinks. Our app not only allows you to have friends, but also allows you to like and dislike drinks so that we can suggest new drinks that have a higher probability that the user will like.

Project Objectives

- Create a web app that organizes and presents the recipes and ingredients for a wide variety of mixed drinks.
- Allow users to create an account with a username and optional profile picture.
- Allow users that are logged in to save their favorite drinks and recipes to their profile for their own access.

- Recommend new drinks to users that have similar components to the drinks that they have favorited.
- Allow users to add and publish their own drink recipes to their profiles.

Functional Requirements

1. User Account

   As a user,

   a. I would like to create/delete an account
   b. I would like to login/logout of my profile
   c. I would like to add my all time favorite drink
   d. I would like to view my profile page
   e. I would like to see a dark and light mode
   f. I would like to make/change/reset my password
   g. I would like to receive emails to change my password
   h. I would like to change my username
   i. I would like my username/password to be remembered on this device
   j. I would like to add a profile picture
   k. I would like to have and edit a bio
   l. I would like to add other users to a friends list

   As a developer,

   a. I want to ensure that no accounts are made with people under 21

2. Drinks

   As a user,

   a. I would like to add drink instructions
   b. I would like to like/dislike drinks
   c. I would like to like/dislike ingredients
   d. I would like to see the username of a drink's publisher
   e. I would like to see pictures of drinks
   f. I would like the option to geo-tag drinks that I have had

g. I would like to have a map with all my geo-tagged drinks visible on my profile

h. I would like to have an all time favorite drink

i. I would like to access drinks without logging in

3. Adding and Editing Drink Information

As a user,

a. I would like to add drink recipes to a list of my liked drinks

b. I would like to remove drinks from my list of liked drinks

c. I would like to edit my drinks

d. I would like to dislike a drink

e. I would like to add and publish my own drink recipe

4. Searching

As a user,

a. I would like to search by user profile

b. I would like to search for bars/restaurants

c. I would like to search by drink

d. I would like to search by ingredients

e. I would like to see a list of all drinks

f. I would like to only search for drinks that are official (i.e. not added by another user)

g. I would like to search and view other user's profile pages

h. I would like to view other people's favorite drink

i. I would like to see other user's geo-tagged drinks when viewing their profile

j. I would like to see similar drinks when searching

5. Sharing

As a user,

a. I want to be notified when accounts I follow have posted

b. I would like to share posts via email/Twitter/Facebook

c. I would like to share drink recipes

d. I want a tab with drinks in the news

e. I would like to create a post with text/image

f. I would like to delete a post I made

g. I would like to search for a post

As a developer,

a. I want to implement a web scraper to find new stories about drinks

6. Bars

As a user,

a. I would like to tag bars/restaurants I have visited

b. I would like to see price comparisons for bars/restaurants

c. I would like to tag bars/restaurants I have visited

d. I would like to see Yelp reviews for each bar/restaurant

As a bar/restaurant

e. I would like to have data on most popular drinks of the night

f. I would like to share pictures, drinks, and recipes

g. I would like to display our location

h. I would like to publish our menu

i. I would like to access and edit my account

j. I would like to create a company account

k. I would like to update my menu

l. I would like to include the prices of drinks

7. Recommendations

As a user,

a. I would like to see a random drink suggestions

b. I would like to receive drink suggestions based on drinks I have favorited

c. I would like to see the top drinks that users have favorited

As a developer,

a. I would like to categorize drinks

Non-Functional Requirements

1. Client Requirements

   As a developer,

   a. I would like the application to be able to function on any browser

   b. I would like the application to be mobile friendly

2. Server Requirements

   As a developer,

   a. I would like to be able to save user/drink data to a local database

   b. I would like to save user preferences to the database

3. Performance Requirements

   As a developer,

   a. I would like the application to run without crashing

   b. I would like the server to handle 25,000 requests simultaneously

   c. I would like the server response to be under 2 seconds

   d. I would like a page to be loaded under 5 seconds

4. Appearance Requirements

   As a developer,

   a. I would like the interface to be self-explanatory and easy to navigate

   b. I would like the user to have some power to customize their screen

   c. I would like the website to appear clear and organized to users

5. Security Requirements

   As a developer,

   a. I would like users to be able to choose if the application can access their location

   b. I would like user's data to be protected between server and client communication

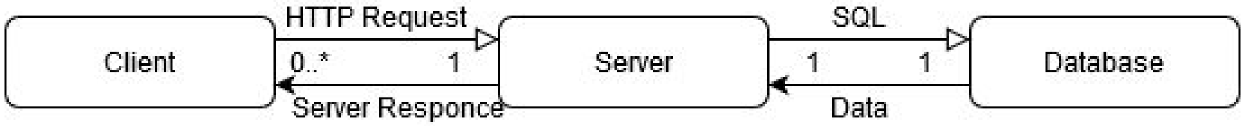
   c. I would like to protect user's personal information from other users

   d. I would like to clean user input to prevent malicious attacks

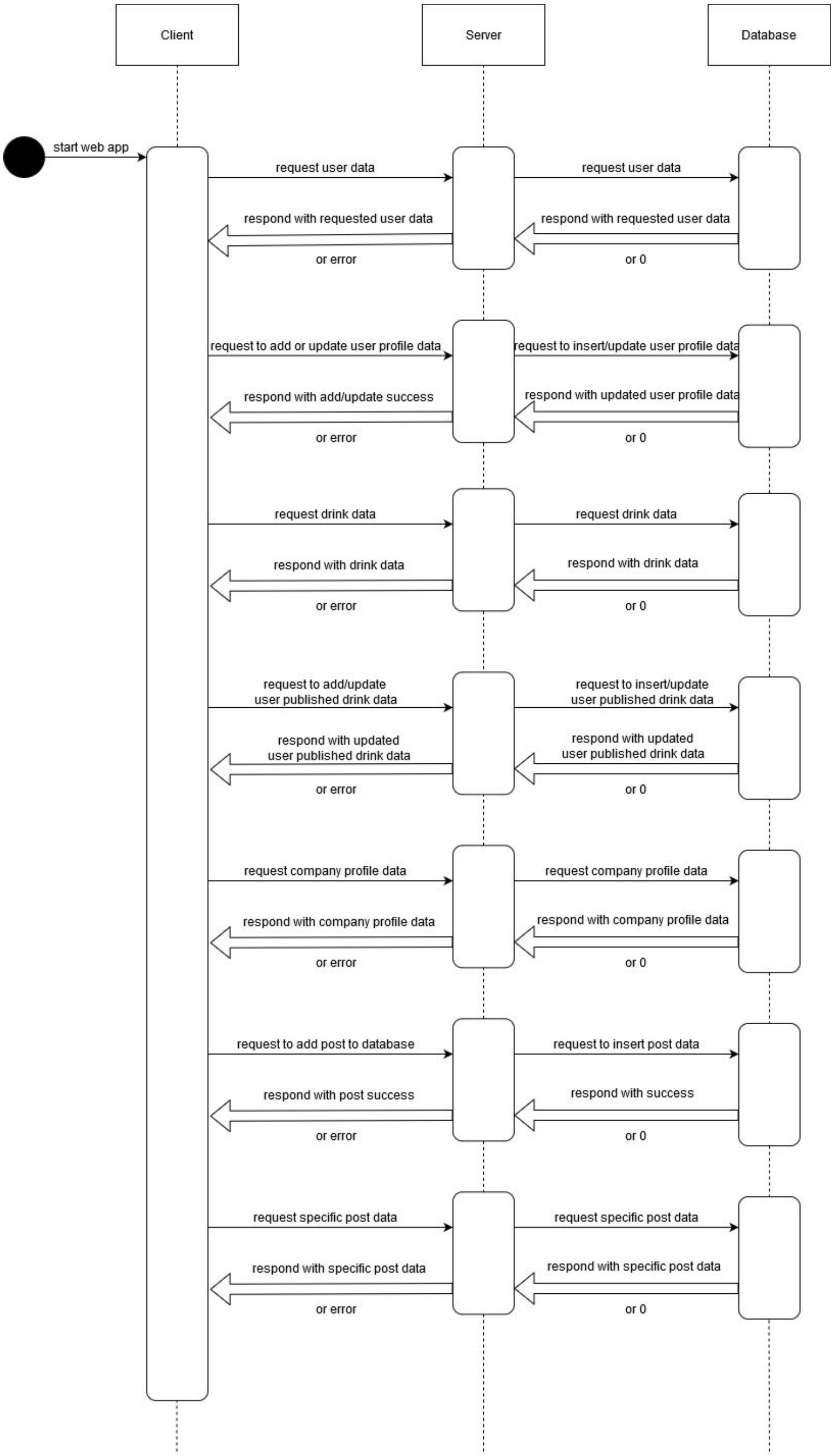
Design Outline

High-Level Overview

This project will be a web application that allows users to create and search drink recipes and text posts and follow other user’s accounts. This application features a client-server model to serve many concurrent users at once. The user is able to navigate about the client, with the client making requests as needed. The client then communicates the request to the server. When the server receives a request, it will either query the database for information and send back either data, an error, or a success, or it will save the information, returning an error or success. The server will be able to hold general information that will be accessed by many users to reduce the amount of queries into the database.

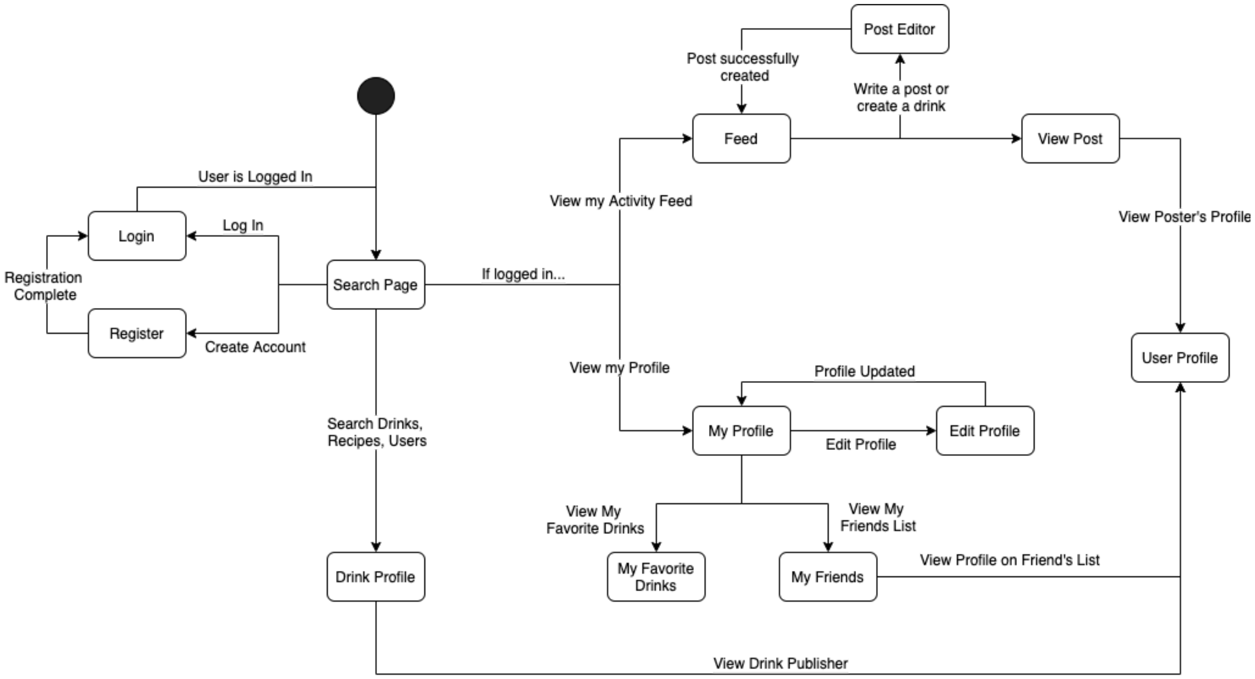


Event Sequence Diagram


Client
Server
Database
start web app
request user data
request user data
respond with requested user data
respond with requested user data
or error
or 0
request to add or update user profile data
request to insert/update user profile data
respond with add/update success
respond with updated user profile data
or error
or 0
request drink data
request drink data
respond with drink data
respond with drink data
or error
or 0
request to add/update user published drink data
request to insert/update user published drink data
respond with updated user published drink data
respond with updated user published drink data
or error
or 0
request company profile data
request company profile data
respond with company profile data
respond with company profile data
or error
or 0
request to add post to database
request to insert post data
respond with post success
respond with success
or error
or 0
request specific post data
request specific post data
respond with specific post data
respond with specific post data
or error
or 0

State Diagram

The following diagram shows an overview of the different states of the application's user interface. When the user first opens the application they will be taken to the homepage containing a search bar and links to login or create an account. Users of the application are able to search for drinks regardless of whether or not they are logged in. A search returns a list of drinks that contain a picture, list of ingredients, recipe, and the user that created it. By clicking on the username, one can see the poster's profile. If a user is logged in, there are additional pages that can be navigated to. Navigating to the activity feed will show recent posts from the profiles that the user is following in addition to the ability to create a post. A logged in user can create their own post from this page as well. In addition to their feed, a logged in user can also view and edit their own profile information, containing a profile picture, all-time favorite drink, username, and a list of their liked and disliked drinks. Finally, from any of these states, the user is able to return to the search page or 'homepage'.



Design Issues

Functional Issues

1. How should we differentiate between subscribers and friends?
   - Option 1: Each user has separate lists of subscribers and friends
   - Option 2: Define friends as mutual subscribers
   - Option 3: Have business accounts be a different kind of user because they do not need a friends list

   Decision -- Option 2

   Discussion: At first we did not see the difference between the two, however we kept saying the word friend's list and realized that there needs to a way to differentiate between businesses and normal users, but to the most simple option was just to have a follow/subscribe option and if two users follow each other mutually then they are friends and that can be the friends list. Instead of having to deal with other lists, and having to differentiate user accounts.

2. How can a user utilize our webapp without having to create an account?
   - Option 1: Force the users to create an account
   - Option 2: Show random posts for someone not logged in, and when they log in show the posts of people they follow/subscribe to
   - Option 3: Have a different home screen for people that are logged in

   Decision -- Option 3

   Discussion: One of our goals with the project is the availability for anyone over 21 to be able to search for drinks and not having to create an account if the user is in a rush to search for a drink. Our hope, is that eventually, with the continued usage of our project, the user finds other users he would like to follow or subscribe to and happily decide to make an account. Since the goal is to search drinks we decided to show a different homepage with a search bar for anyone that confirms their age, instead of showing random posts or forcing anyone to create an account.

3. How should we utilize location services?
   - Option 1: Do we collect location services from only bars/restaurants?
   - Option 2: Do we collect location services from users?
   - Option 3: Do we collect location services from bars/restaurants and people?

   Decision -- Option 1

   Discussion: We elected to collect locations from only bars/restaurants in order to make a map of locations that drinks can be linked to. Users are able to tag bars/restaurants they have been to and associate drinks with the restaurants/bars. The location of the bars/restaurants will be visible on the map. Option 2 and 3 required too much data storage and would have complicated the web app significantly.

4. When and how should we check to be sure users are 21?
   - Option 1: Make users put their age in once they make an account.
   - Option 2: Force users to create an account in order to verify age.
   - Option 3: Have a banner appear that appears that makes the user acknowledge that they must be at least 21 years of age in order to drink alcohol.

   Decision -- Options 1 and 3.

   Discussion: As a team, we wanted to give users the ability to use the web app without having to create an account or log in. Once a user navigates to our web app, they must click and acknowledge that they must be 21 years of age. Additionally, when users create an account they will be required to enter the date of birth. We will not allow users to create an account if they are younger than 21.

5. Should we have the users' login name be the same as their display name?
    - Option 1: Have the users' login name be different than their display name.
    - Option 2: Have the users' login name be the same as their display name.
    - Option 3: Have the users login with their email and then choose a display name.

    Decision -- Option 3

    Discussion: We decided to proceed with option 3. This is because keeping track of an email, a login name, and a display name would be too much data to maintain. Options 1 and 2 would require us to store three data fields whereas Option 3 allows us to only store two. For simplicity's sake and in order to not waste space, the user will login with the email they used when they created the account and their password.

Non-Functional Issues

1. How should we host our application?
    - Option 1: DigitalOcean
    - Option 2: Heroku

    Decision -- Option 1

    Discussion: We elected to proceed with Option 1. While Option 1 requires payment to host applications, university students are able to redeem $ 50.00 worth of hosting time through the GitHub Student Developer Pack. Option 2 is another hosting service that is both free and paid. The free version of Heroku has a slightly slower start-up time than some of the other options in consideration, and any running server will deactivate after 30 minutes of inactivity. Because this is not ideal for our use case, especially during the early stages of our application's deployment, we will need a reliable hosting service that will keep our application online when we need it to be online.

2. What framework will we use to develop our frontend?

- Option 1: jQuery
- Option 2: HTML and CSS
- Option 3: Javascript with the React framework

Decision -- Option 3

Discussion: Since some of us have experience with the React framework we have decided to work with that because the members of the team that have experience also find it more appealing than the other options, based on terms of organization and previous experience with certain libraries. React makes handling HTML and CSS a lot easier so we decided on that option.

3. What are we going to use to implement our back end?
    - Option 1: PHP
    - Option 2: Java
    - Option 3: Python

Decision -- Option 2

Discussion: Since everyone on the team has experience with Java due to the curriculum at Purdue, we decided to use Java to implement our backend. One of our members also has experience with Java that is built on top of a mySQL database, which we will use to store user profiles and drink recipes for our application.

4. What are we going to use for the database?

    - Option 1: MySQL
    - Option 2: Firebase
    - Option 3: MongoDB

Decision -- Option 1

Discussion: While Firebase is very easy to implement and use, by default the database that it sets up is non-relational. Because all of our classes are connected to each other in some fashion, we decided that we wanted something more customizable to fit our use case. As far as Option 2 is concerned, we looked into using MongoDB as opposed to MySQL, but we ultimately decided to proceed with MySQL as one of our team members has a lot of experience working with it in conjunction with a Java backend.

5. Is the webapp going to be mobile?

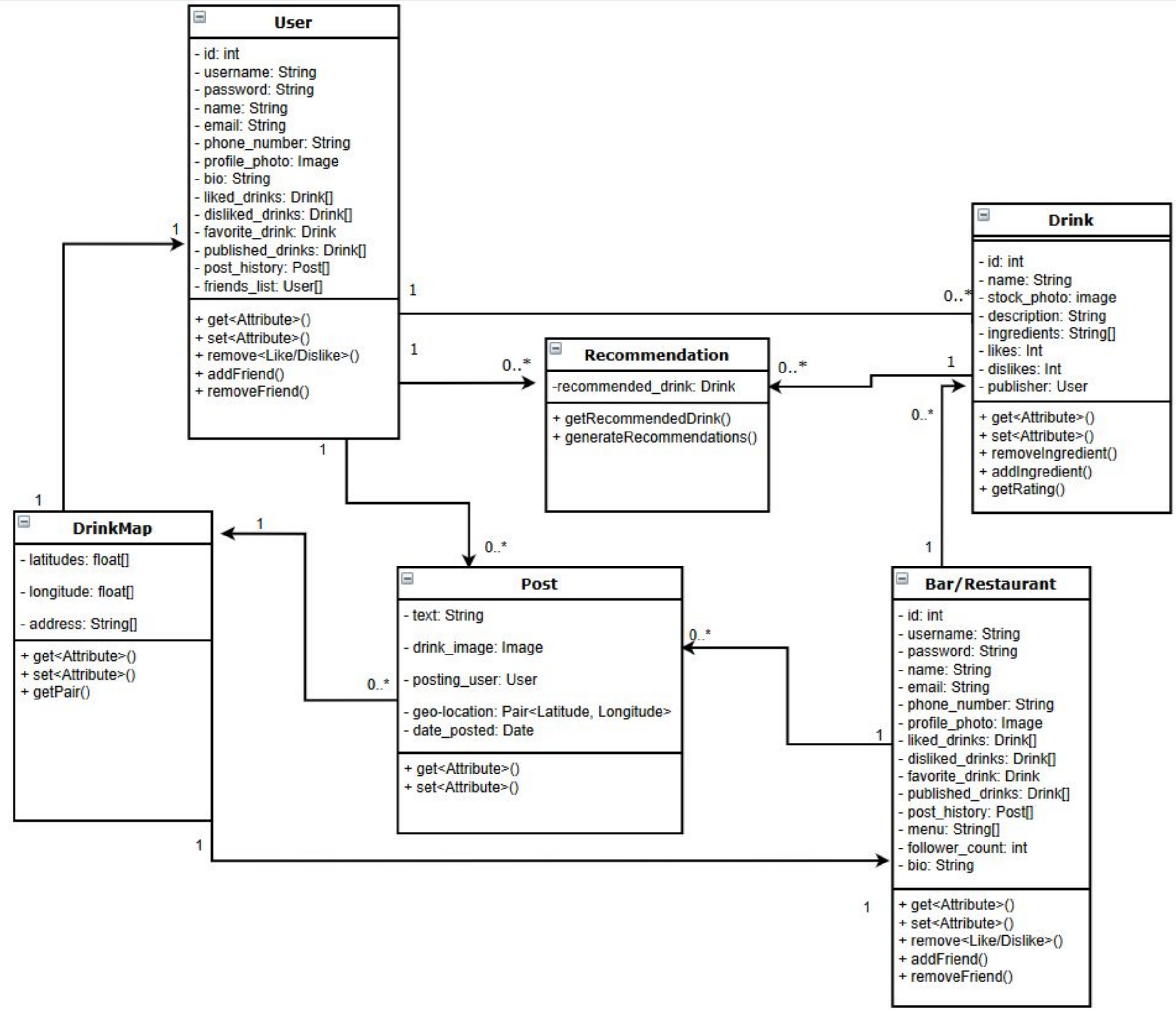
- Option 1: No
- Option 2: Yes
- Option 3: Make it mobile accessible

Decision: Option 3

Discussion: None of us have background with mobile apps. This would be a great edition if time allows, but it is not a feasible priority of the project. React has some components to help with making the website mobile friendly so it is not necessary for us to create an app for it. Since it would be convenient to use this site on the go, we are going to make it mobile friendly, using React, if time allows.

Design Details

Data Class Level Design



Description of Data Models/Classes and Interactions

USER

- A user is created when a person signs up for the website
- A user isn't required, but it gives a unique experience
- Each user has a unique username
- Each user enters there DOB, password, email, name, bio, profile picture when creating their account
- A user will have their location, liked and dislike drinks, friends, posts, published drinks, favorite drink saved dynamically when using the site
- A user will be able to see and edit information fields from their profile page
- A user will be able to see recommendations based off their profiles
- A user will be able to follow accounts, upgrading to a friendship if they follow back
- A user can create a drink, adding it to their published drinks
- A user can post on the feed, adding posts to their post list

DRINK

- A drink represents a name, recipe, process created by a user
- A drink has a unique name derived from the publisher and name
- A drink has a recipe, name, directions, photo assigned when created
- A drink has a rating from the likes and dislikes it gets from users
- A drink is editable only from its publisher

POST

- A post represents a text and image post from a user displayed on the feed
- A post has a text component, an image component, a publisher and a location when created
- A post is able to link to a profile when a user clicks on their name/profile
- Users will be able to see where a post is from with the location

DRINKMAP

- A drinkmap represent a map for a user with tagged locations
- Each location is pinned to the drinkmap, showing places a user has been to
- Other users are able to see the pinned locations of other users and the user depending on if the user allows their location to be seen

RECOMMENDATION

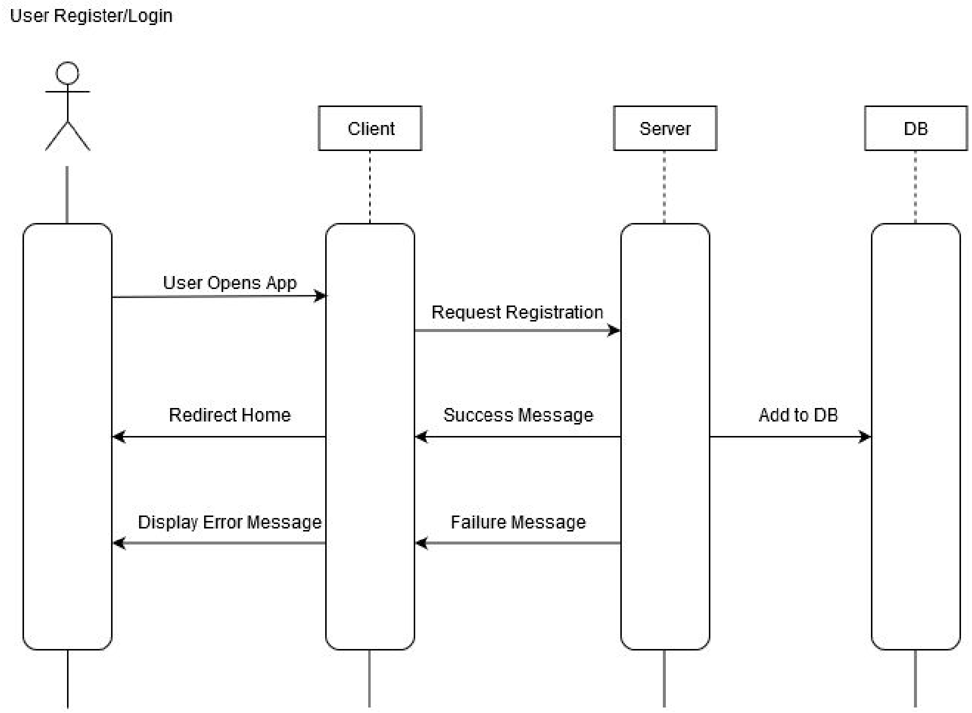
- Recommendation represents recommended drinks for a user
- A recommendation holds a list drinks recommended based on a user profile

BAR/RESTAURANT

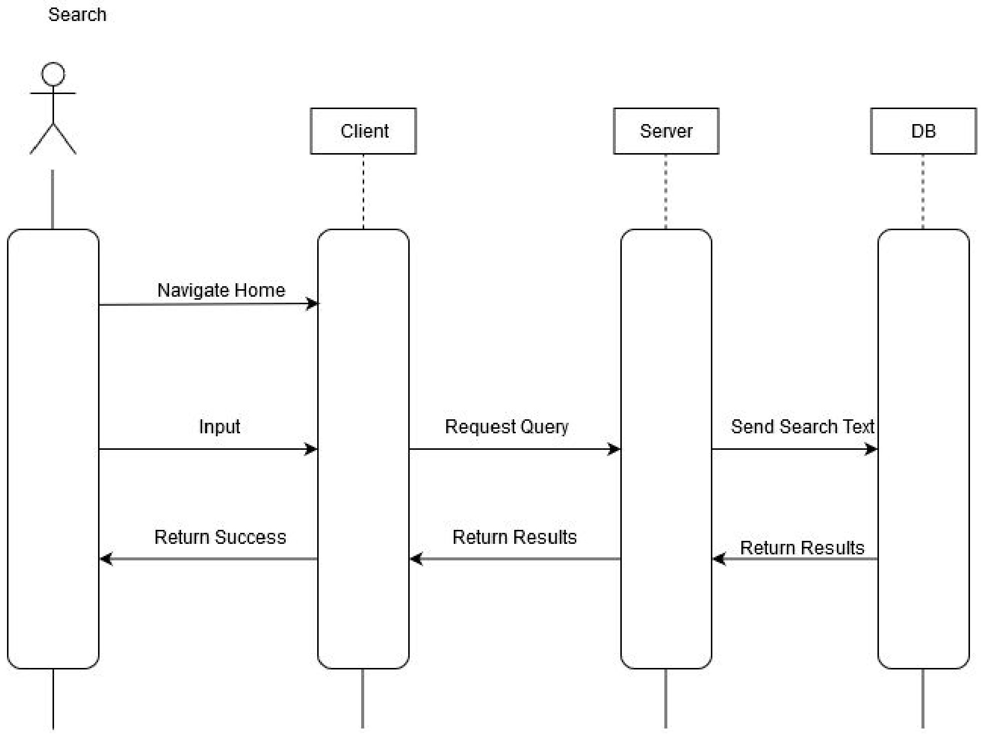
- Bar/restaurant represents a business that opens an account on the app
- A business has a unique username
- A business has a name, menu, email, phone number, profile picture, bio, a location, and password on creation
- A business is able to update and change their menu, name, email, phone, profile picture, bio, password from their homepage
- A business can make posts to all their followers
- A business will appear different in the feed compared to regular users

Sequence Diagrams

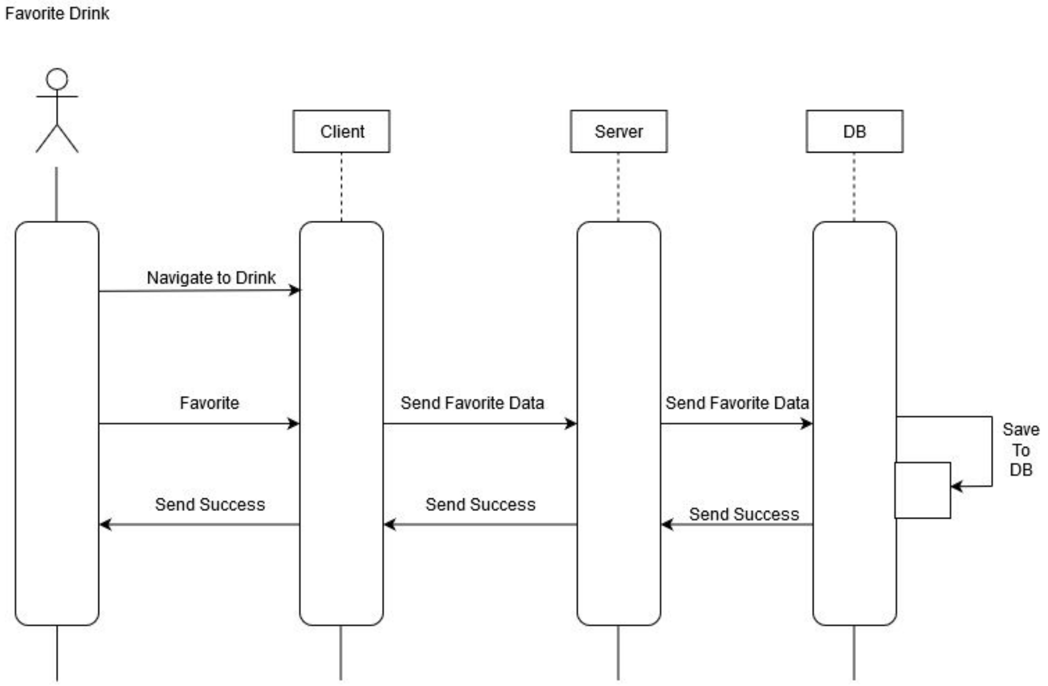
- Sequence of events when a user creates an account



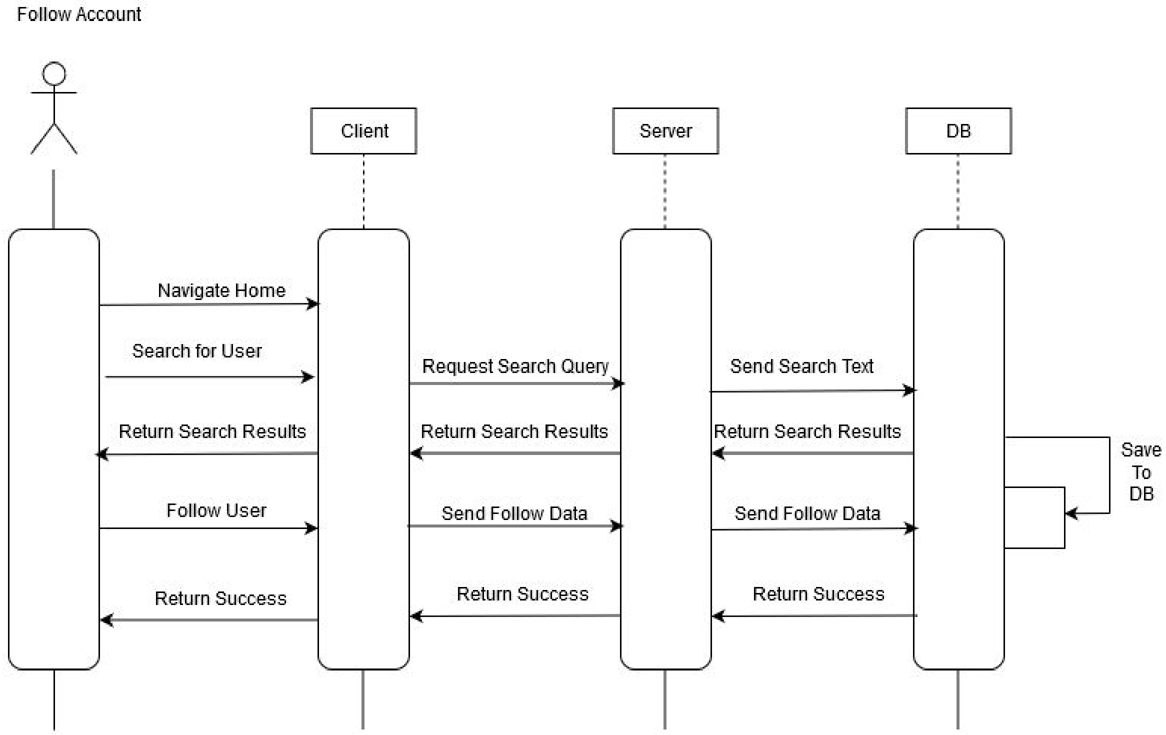
- Sequence of events when a user searches for a drink



- Sequence of events when a user likes a drink

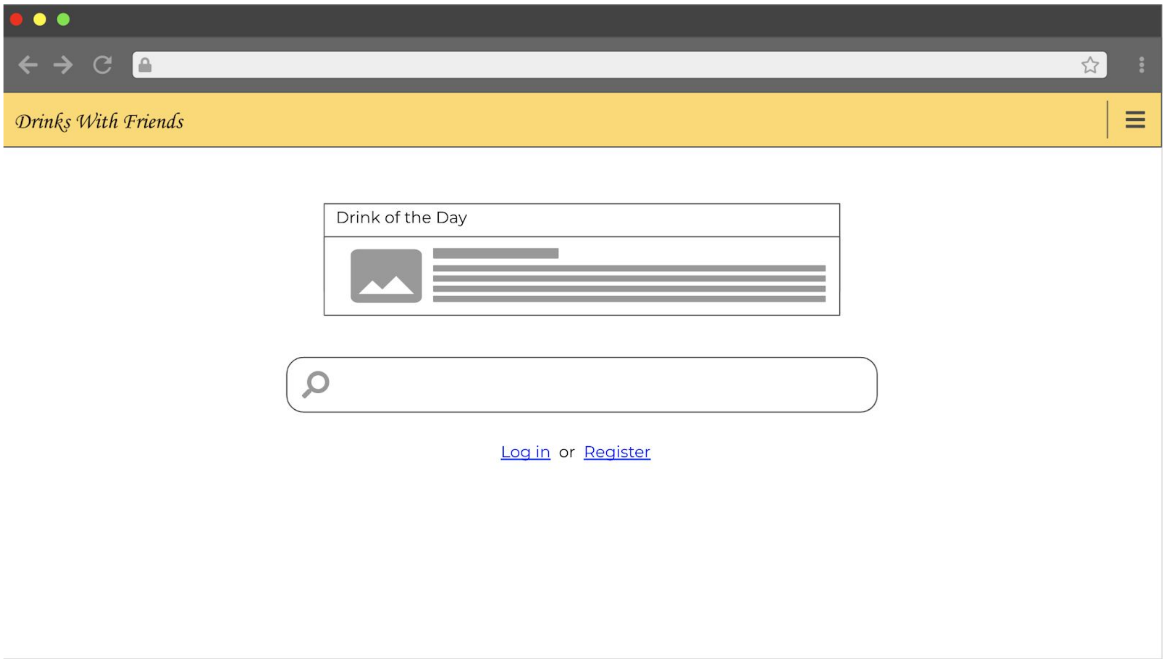


- Sequence of events when a user likes a drink

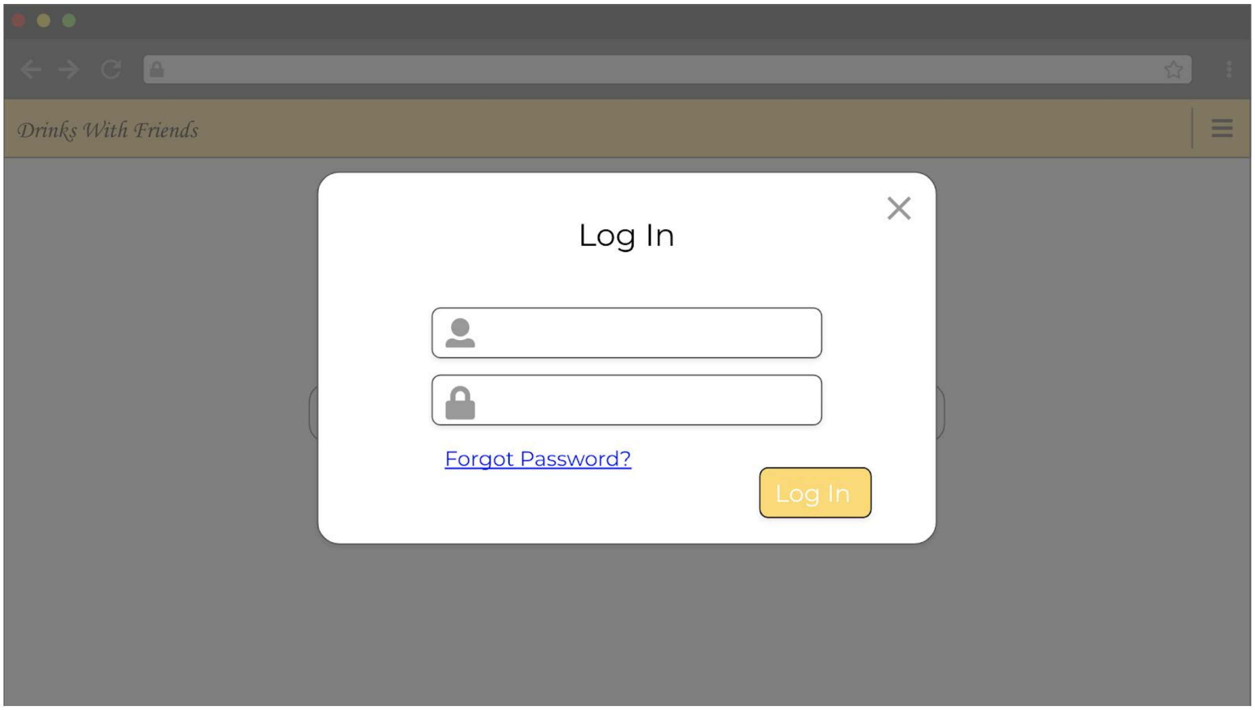


UI Mockup

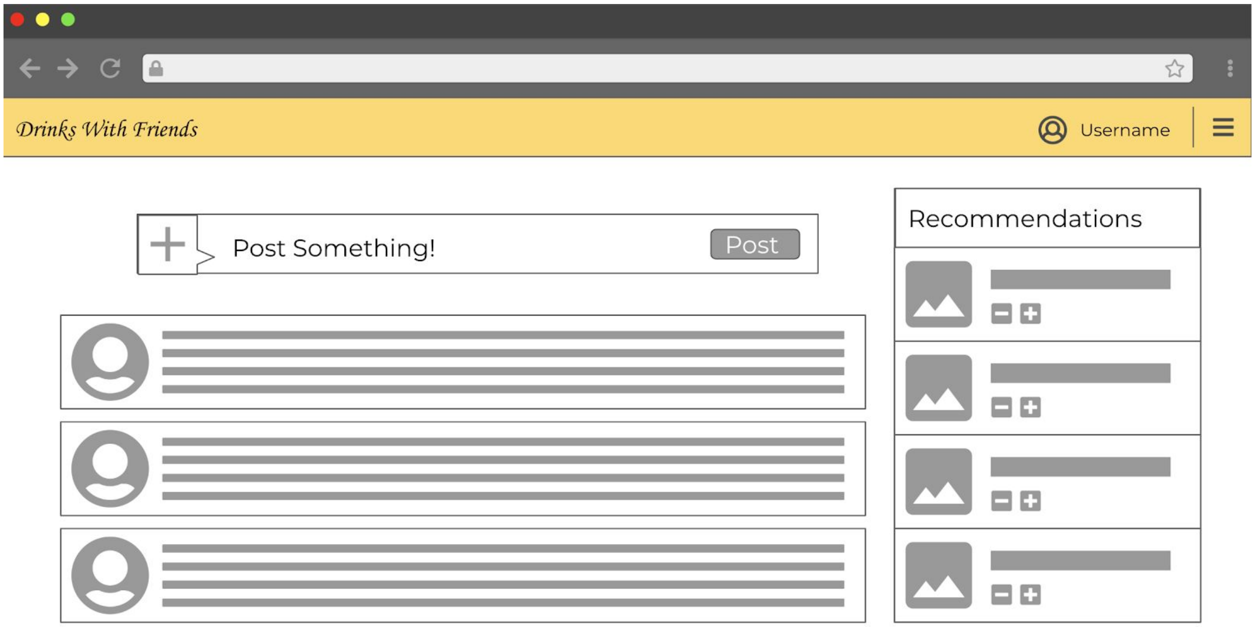
Homepage/Search



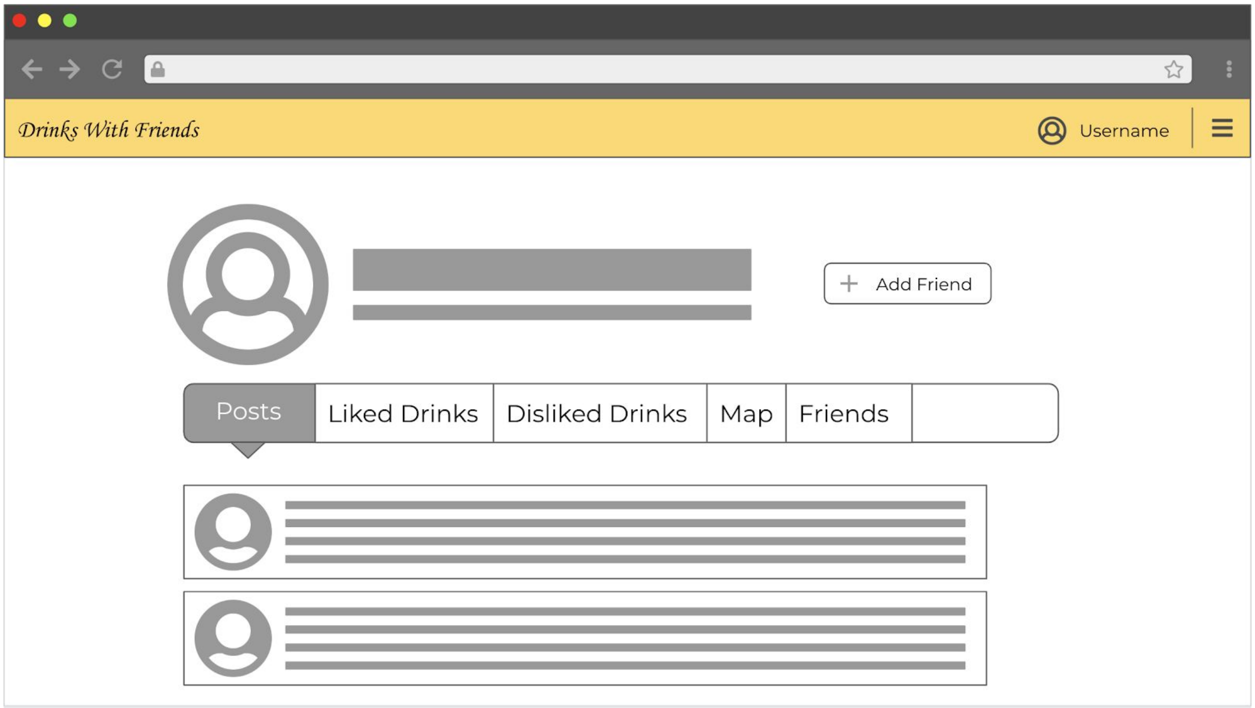
Login



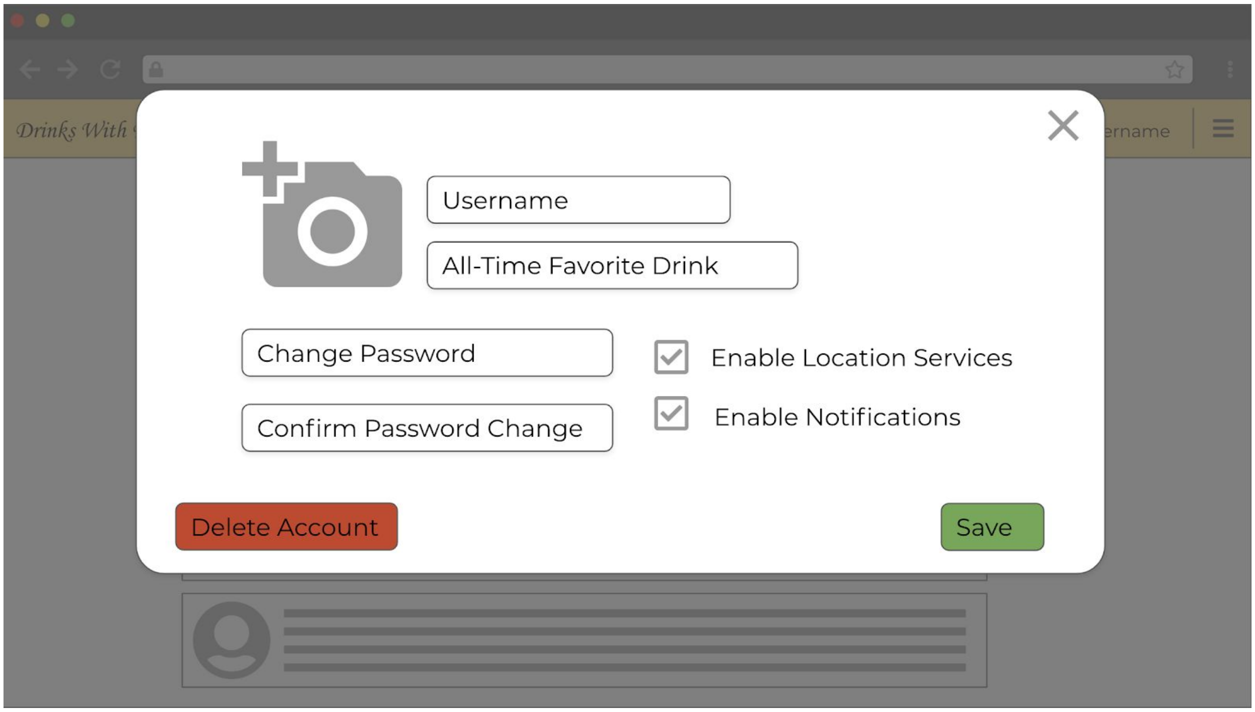
Post Feed



User Profile



Edit Profile



☑ Enable Location Services

☑ Enable Notifications