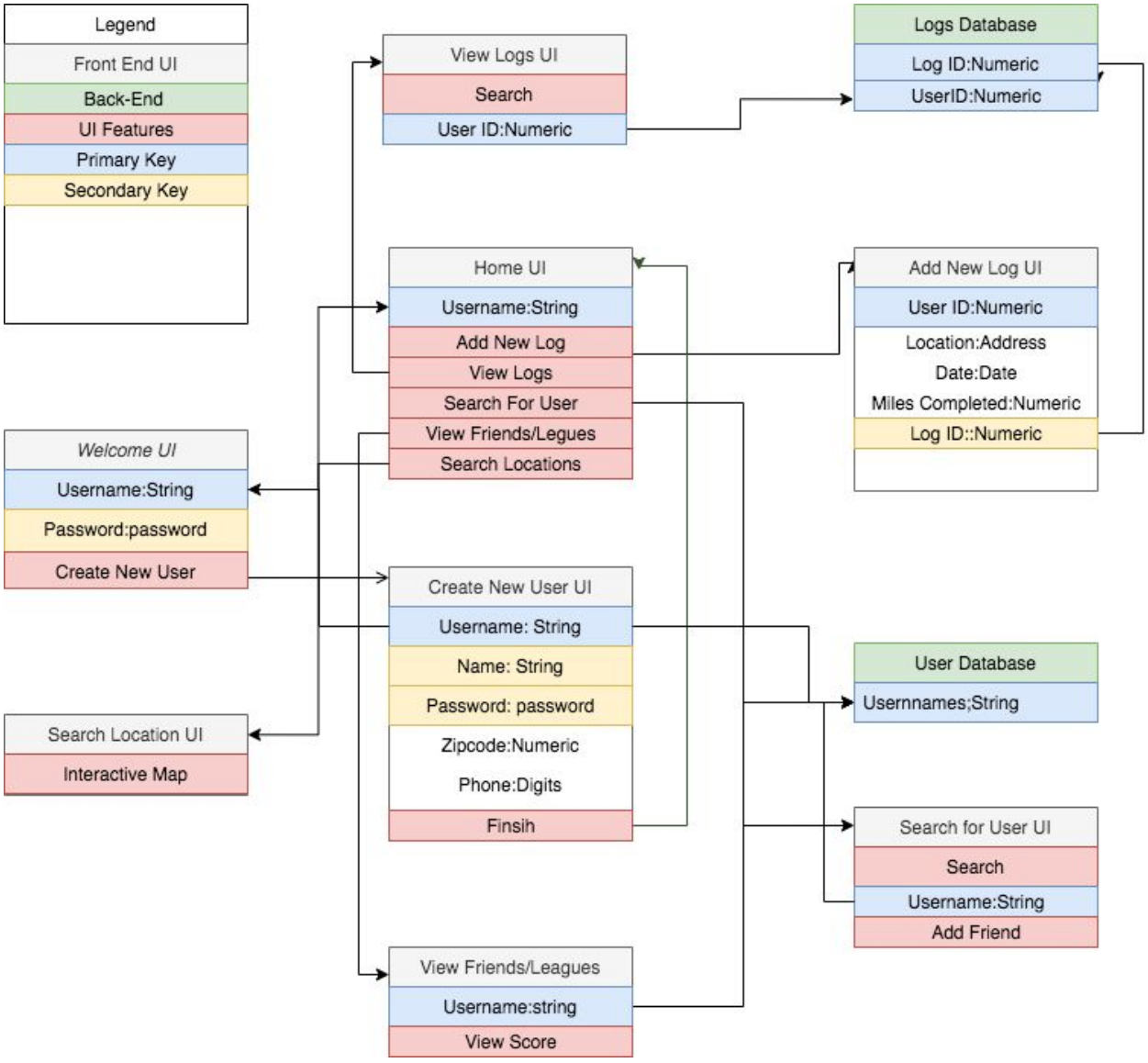
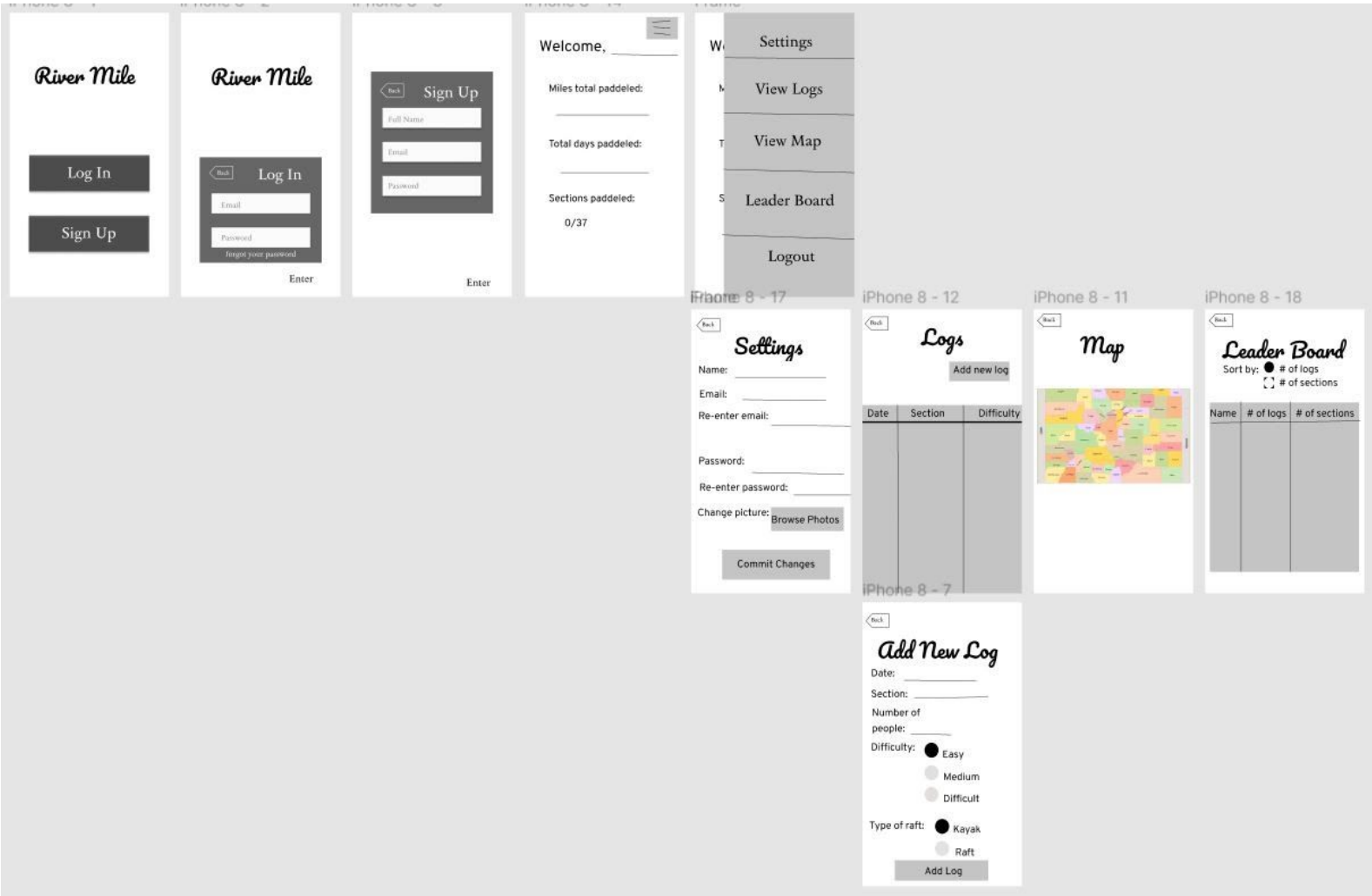
REVISED LIST OF Features:

- Personal User Accounts
    - Take input from users and store it so they are able to see all information which they have entered throughout the usage of the application. (Milage, Days on water, etc.)
- Users can input usage data
    - Users can input and store how far they boated on a given day and other relevant data.
- Users can view usage history
    - Individuals can go back and view their recorded data to see how many miles they have boated in a certain period of time , how many days they have spent on the water, ect.
- Users can see sections on a map
    - Users can compare mileage, days on water, and conditions during their trip, this can be shown on a map to indicate the route and potential hazards which could be encountered during a specific route.
- Users can share section data
    - Users should be able to share data of specific sections of rivers and be able to generate and share data on specific sections of whitewater, which others can use to navigate.
- Users can see live water levels
    - Current water levels for sections based on USGS gauges will be displayed for users.
- Users can compare their data with other users.
    - Leaderboard of mileage displayed on users profile pages.

Architecture Diagram:



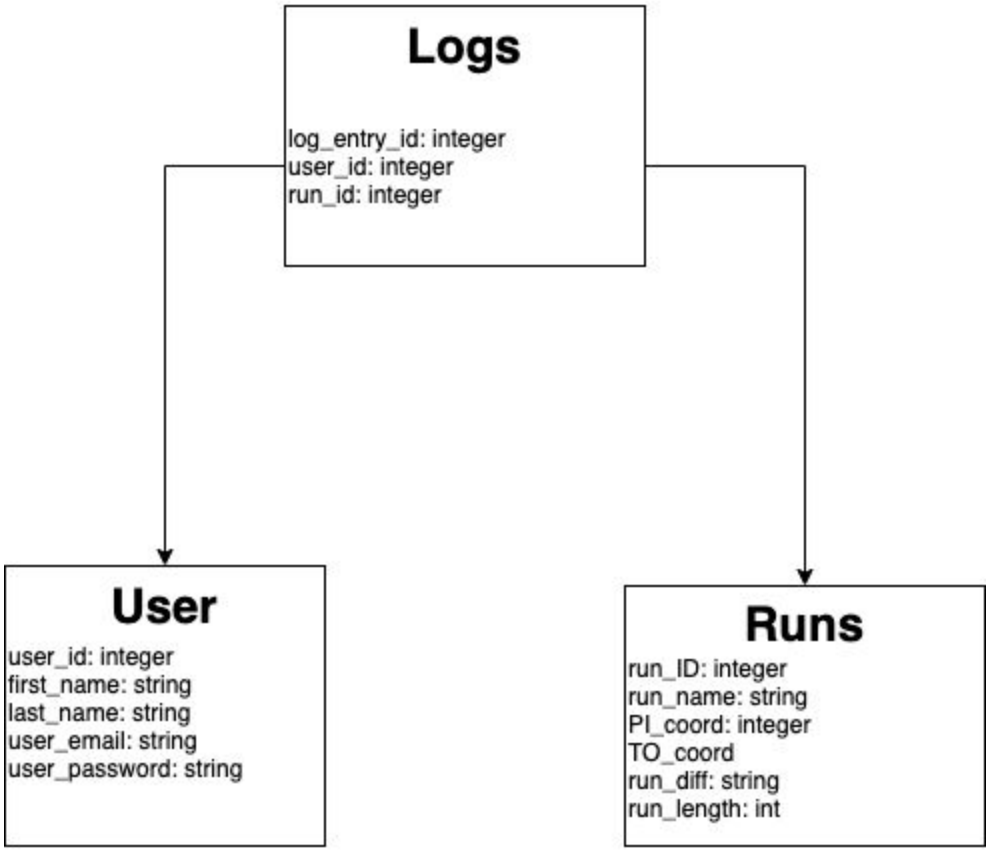
Front End Design:



Web Service Design: N/A

Database Design:

This is a SQLite database. The Logs database is the core of our app, as it is where all of the user recorded runs will be stored. Log will reference the database of already created runs (runs must be created before they can be logged) and the users. This allows stats to be tracked by user (comparing mileage, number of runs, avg diff, ect) or by run (popularity, number of visits, ect). user_id and run_id are both forign keys in the logs database, allowing for quick and compact referencing.



Project Plan:

https://docs.google.com/spreadsheets/d/1WeA_c3Ca2mbwgcZTd9i0lM4tJRQGgeW47oYFBYDnbho/edit?usp=sharing