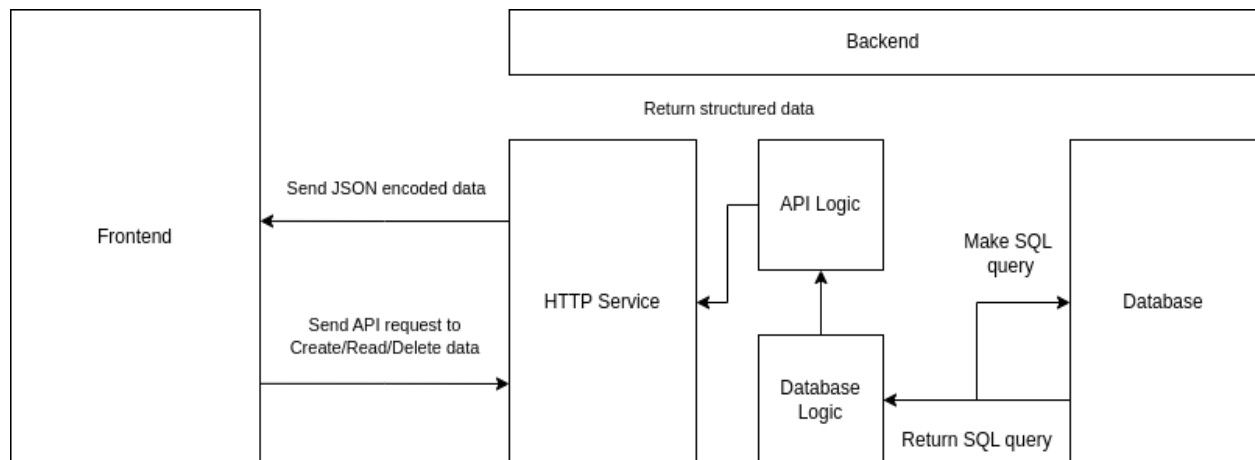# Deliverable 2

Group 2

Members:
Gilbert Martinez
Jon Brooks
Andrew Heckman

System Structure

The system is divided into two major components, one that handles the *user interface* and another that supports the UI with persistent data storage and lookup using a *database*. We call them *frontend,* and *backend* respectively.

- **Frontend:** The *frontend* utilizes a framework to organize and present a coherent web based interface. It can be displayed on any device with an up-to-date internet browser.
- **Backend:** The *backend* is broken up into 3 subsystems. One is written to encapsulate the logic that is needed to perform *database* operations, one handles HTTP requests that serve as the communication protocol between the *frontend* and *backend*. The last module shapes the data returned from the *database* into what is being requested from the *frontend*.
    - **Main (HTTP):** Handles HTTP requests and responses using the Flask library. Other HTTP server features are handled such as static file uploads for user content. Uses API as a dependency
    - **API:** Queries data from the *database*, returning it shaped the way it needs to be in the *frontend*. Uses SQL as a dependency
    - **SQL:** Defines the logic for the smaller order *database* queries the API module may need.
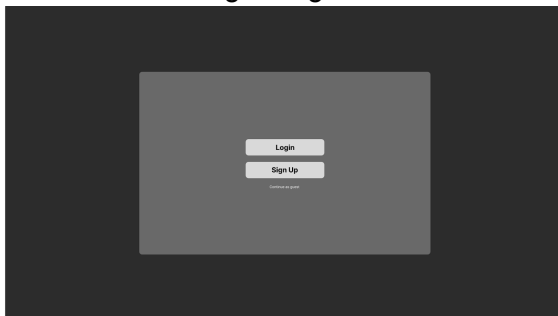
Functional Requirements

- Recipe storage
  - All recipes posted by the users will be stored on a SQLite3 database.
- Recipe rating system
  - Users will rate recipes on a scale of 1 to 5 stars then system will report the average rating of each recipe.
- Recipe sorting
  - Users will be able to sort the recipes by the most recently added and by the highest user rated
- Recipe search
  - Users will be able to search for recipes by their names and ingredients
- User storage
  - The system will store all the users in the SQLite3 database
- User's saved recipes
  - The users will be able to save recipes and the database will keep track of each user's saved recipes
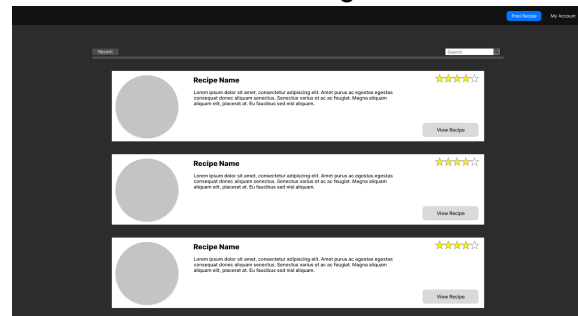
Non-Functional Requirements

- Simplicity
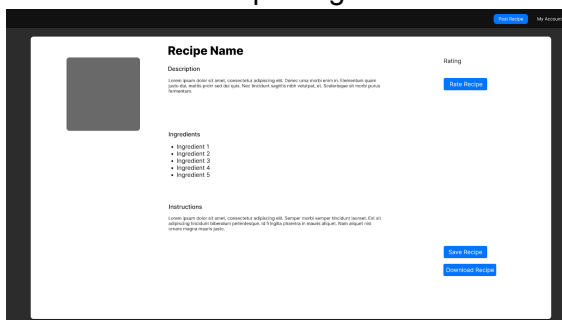- Minimal
- Reliable
- Usability

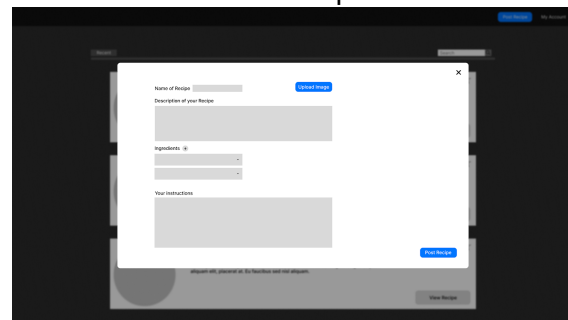# User Interfaces

Login Page

Home Page
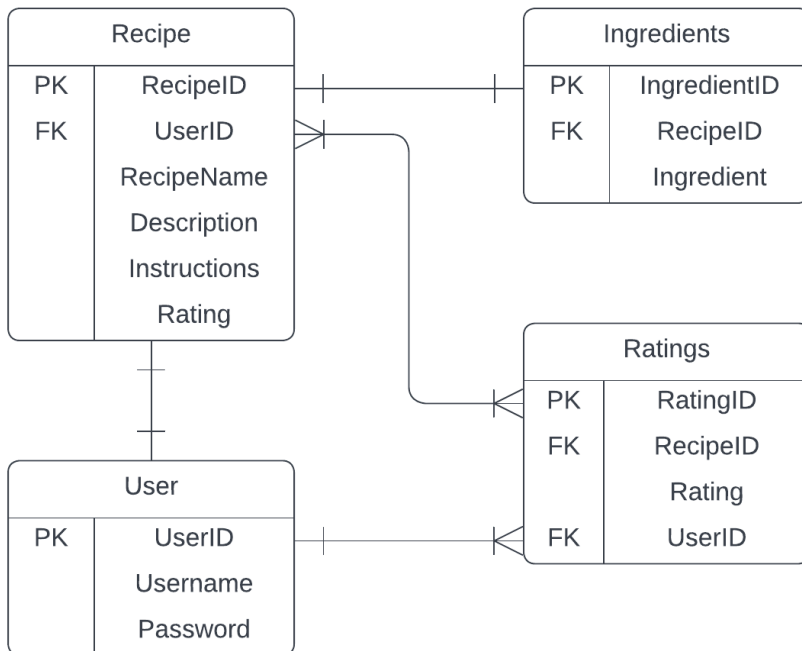

Recipe Page

Post Recipe

*Note: Color schemes and styling of the user interface will change, but the overall layout will remain the same.

Database Interface

Development Phases

Feature	Description	Phase
Recipe Storage	Store user-entered recipes in SQLite3 database	1
Recipe Rating	Store user-entered ratings from 1-5 stars, display average rating	2
Recipe Sorting	Display recipes in order of most recent or highest rated	2
Recipe Search	Search for recipe by name	3

User Storage	Store user information in SQLite3 database	1
Recipe Saving	Users can save specific recipes	3