Help-Me-Recipe

Design Document

Team 29

- Abigehl Slattery
- Sriharsha Valluripalli
- Ben Malone
- Andre Chang
- Guillermo del Aguila
- Annie Xu

Index

- Purpose 2
  - Functional Requirements
  - Non Functional Requirements
- Design Outline 6
  - High Level Overview
  - Sequence of Events Overview
- Design Issues 8
  - Functional Issues
  - Non-Functional Issues
- Design Detail 12
  - Class Design
  - Sequence Design
  - Navigation Flow Map
  - UI Mockup

Purpose

When looking for recipes online, people oftentimes look through numerous recipes before deciding on one, usually randomly. Deciding on a recipe randomly can result in food that turns out poorly, causing a user to waste time and money in the production. This creates a demand for a website that allows users to search more efficiently for what they want, while ensuring quality of recipes.

We believe that by creating a website which allows users to share their own recipes as well as rate other's recipes, we can ensure quality recipes for our users. As well as ensuring quality, Help-Me-Recipe will provide users with numerous search options. This will allow the user to narrow their search, providing them with recipes that better match their preferences. While there are similar websites, Help-Me-Recipe will provide users with a more social media oriented experience. Users will be able to create accounts and follow other users. The users, once they have created an account, will be able to rate recipes as well as save them to their favorites.

Functional Requirements:

1. **User can create an account, log in and edit their account:**

   As a unregistered user:

   a. I would like to be able to register for a Help-Me-Recipe account.

   As a user:

   a. I would like to set my display name.
   b. I would like to set a profile picture.
   c. I would like to be able to give personal information in my profile.
   d. I would like to be able to login to my account.
   e. I would like to be able to reset my account password if I forget it.
   f. I would like to indicate my cooking experience level.

2. **User can create recipes/meal plans and edit them**

   As a user:

   a. I would like to be able to post my own recipes.
   b. I would like to be able to see how long it takes to make a recipe.
   c. I would like to indicate the ingredients that my recipe requires.
   d. I would like to indicate the nutritional values of my recipes.
   e. I would like to view instructional videos which explain how to cook a

dish, when available.

f. I would like to be able to post/share pictures of the food I make.
g. I would like to have a checklist made out of the ingredients of a recipe.
h. I would like to be able to share/post video on the recipe.
i. I would like to create weekly meal plans.
j. I would like to share weekly meal plans for others to use.
k. I would like to view weekly meal plans shared by others.

3. User can follow other users and favorite recipes

As a user:

a. I would like to be able to save my favorite recipes.
b. I would like to be able to receive notifications about new followers/ratings.
c. I would like to be able to view my favorite recipes.
d. I would like to be able to like/dislike other user's posts.
e. I would like to see my own activity in the social feed.
f. I would like to be able to follow other users which will allow me to see their activity on my social feed.

4. User can search for different recipes

As a user:

a. I would like to be able to search for a specific recipe.
b. I would like to be able to filter out recipe prices in my search.
c. I would like to be able to filter out recipe ingredients in my search.
d. I would like to be able to filter out keywords in my search.
e. I would like to find recipes depending on the calories they have.
f. I would like to be able to search for recipes depending on my kitchen utensils.
g. I would like to be able to choose between different cuisines.

5. User can comment and do follow ups of the recipes

As a user:

a. I would like to comment on other people's recipes.
b. I would like to be able to rate other user recipes out of 5 stars.
c. I would like to be able to directly message other Help-Me-Recipe users.

6. User can see their nearby stores

As a user:

a. I would like to be shown nearby stores on google maps.
b. I would like to see whether ingredients are in stock at stores around me. (if time permits)

7. **Admin can manage all recipes and user accounts**
   As an admin:
   a. I would like to have the ability to delete posts.
   b. I would like to have the ability to ban/delete users.
   c. I would like to create groups of similar recipes for users to see.
   d. I would like to showcase recipes of my choice.

Non Functional Requirements:

1. **Platform/Hosting**
   As a developer:
   a. I would like to use HTML/CSS and Javascript with Bootstrap as the framework for front-end
   b. I would like to use flask as the framework for back-end
   c. I would like to host the web app to a cloud service
   d. I would like to use Google Maps API to find nearby stores
   e. I would like to use SQL to implement the database

2. **Security**
   As a developer:
   a. I would like to make sure that all the users' information is stored securely in the database.
   b. I would like to make sure different users have different authorizations through the website.
   c. I would like to make sure that all conversations are securely stored in the database.

3. **Usability**
   As a developer:
   a. I would like to make sure that the UI is easy to navigate and aesthetically pleasing.
   b. I would like to make sure that recipes are easy to create and that the readers can read it step by step.
   c. I would like to make sure that the search bar has different tags for advance search while staying readable.

4. Performance

As a developer:

a. I would like to make sure that our server can handle large amounts of requests.
b. I would like to make sure that our database can handle large amounts of data.
c. I would like to make sure that out application has a minimum latency.

Design Outline

High Level Overview

Help-Me-Recipe will be a web application that will use the Client-Server model. This application will enable users to post and search for recipes based on filters such as complexity level, ingredients used, time spent making recipe etc. Additionally, Help-Me-recipe will be a social media oriented platform that will let people create their own profiles and contact each other. Our server will be able to handle and process multiple clients and multiple types of requests. This will be done using the Flask framework—which uses python—to implement the requirements. The database will be implemented using SQL and the server will store and retrieve data requested by the client.

1. Client

The client running in the user's computers will provide an interface to interact with our server. This will allow them to search for recipes, post etc. The client will use an HTTP request to send and retrieve data from the server about recipes and other user's profiles. The subsequent HTTP answer from the server will be sent to the client which will interpret it and display it to the user. The client will be built on HTML/CSS and Javascript with Bootstrap.

2. Server

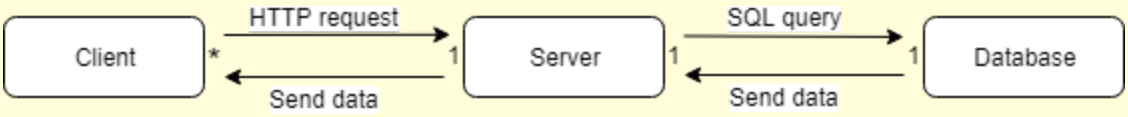
The server will receive the requests of the clients and handle them. Once the server has interpreted the client's request it will perform a search on the database, retrieve or write data about recipes, profiles etc. After the server has interacted with the database it will send a response to the client that sent the original request. The Framework used to implement the server will be flask and the programming language will be python.

3. Database

The database will store all the information regarding the recipes posted by the users. It will have to store them according to all the filters available to the users. It will also store information about the users and their profiles. This relational database will answer the queries performed by the server. SQL will used to implement this database.

Sequence of Events Overview

The client starts when the user opens the web app on his/her computer and logs in to their user profile. Once the user has logged in, the client can request and send data to the server. The server will then handle this request, performing the tasks necessary to satisfy the request. The client may send data about recipes that will be stored in the database by the server. On the other hand, there may be a data request of recipes or another user's information, the server will then have to retrieve the data from the database by performing queries and then send that data back to the client. Other actions that might be required are update requests—in this case the server will have to find the specific data stored in the database, change it and then inform the client of it. Below is a simple overview of these events.



Design Issues

Functional Issues

1. **Should user ratings be available for users to see**

   Options:

   Yes

   No

   Choice: No

   Justification: If a user had a low rating and they were able to see it, they may be discouraged from using the website. Alternatively they may be driven to create a new account, making the rating system less efficient. We decided that in order to preserve user moral, we should keep the users from viewing their user rating.

2. **How should user rating be determined**

   Options:

   All recipe ratings

   All recipe ratings and recipe popularity

   All recipe ratings, recipe popularity and number of followers

   Choice: All recipe ratings, recipe popularity and number of followers

   Justification: By combining these three elements we will get a better understanding of how popular a user is. If we just used recipe ratings, and the recipe only had one rating, it would not be a good indicator of how popular the user was. While using recipe rating and recipe popularity provides a good indicator of how popular a user is, we believe that by making a very small portion of the user score dedicated to number of followers, we will be able to get scores that better reflect a user's popularity.

3. **Should Users need to have an account to use Help-Me-Recipes**

   Options:

   Yes

   No

   Choice: No

   Justification: This will allow the website to appeal to people who do not wish to create accounts. While one of Help-Me-Recipe's most prominent features is its social media like environment, it can also be used simply for users to search for recipes. We felt that the most important feature of Help-Me-Recipe was its search function, and with that being the case,

we decided it was important to allow users to search and view trending without having an account.

4. **What information should we require in order to create an account**

   Options:

   - Email and password
   - Username and password
   - Username, password and email

   Choice: Username, password and email

   Justification: This will provide the user with the ability to change their password if they have forgotten it. While the option of email and password would also provide the user with the ability to retrieve their account information, it would limit the creativity of the user, because it would not allow them to choose their username.

5. **Should we be able to ban users**

   Options:

   - Yes
   - No

   Choice: Yes

   Justification: In the event that a user violates the forum rules, there must be a way to address the issue. The simplest way is to allow admins the ability to ban problematic users. Without the ability to ban users who are posting inappropriate comments or harassing others, it would create a hostile environment and dissuade other users from utilizing our site.

Non Functional Issues

1. **What should we use for our database**

   Options:

   - Oracle XE
   - SQL
   - MongoDB

   Choice: SQL

   Justification: We decided to use SQL because it is a relational database that is secure and more appropriate for our application. It has a great framework with multiple data views that make it easier to use. People are more familiar with SQL than other databases in our group, so the learning curve wont be as huge as the other ones.

2. **What should we use to program the backend**

    Options:

        Flask
        Django
        React.js
        Ruby on Rails

    Choice: Flask

    Justification: We decided to use Flask as our back-end framework because it's simpler with more flexibility and fine-grained control that is more appropriate for our application than its counterparts. It provides many extensions which enables a large amount of functionality while not creating a large workload for our developers.

3. **How should we host our website**

    Options:

        Digital Ocean
        Azure
        Heroku
        AWS

    Choice: Heroku

    Justification: We chose Heroku because it offers an immense amount of support for our application. The add-ons available through Heroku will assist us in scaling our website and overall management. Heroku's goal is to allow developers to focus on the app on not be distracted by the hosting services' interface, this will allow us to build a powerful and efficient site.

4. **Which framework should be used for the frontend?**

    Options:

        Bootstrap
        Semantic-UI
        Foundation
        Materialize

    Choice: Bootstrap

    Justification: Bootstrap is an in-depth platform for developing the front-end of our webapp. It offers a dynamic building experience which allows us to choose the features we need and build out front-end from there. In addition, it comes packed with a significant amount of tools

which add relevant and useful add-ons at a low time-cost to our developers.

5. **Which language should be used for the frontend?**

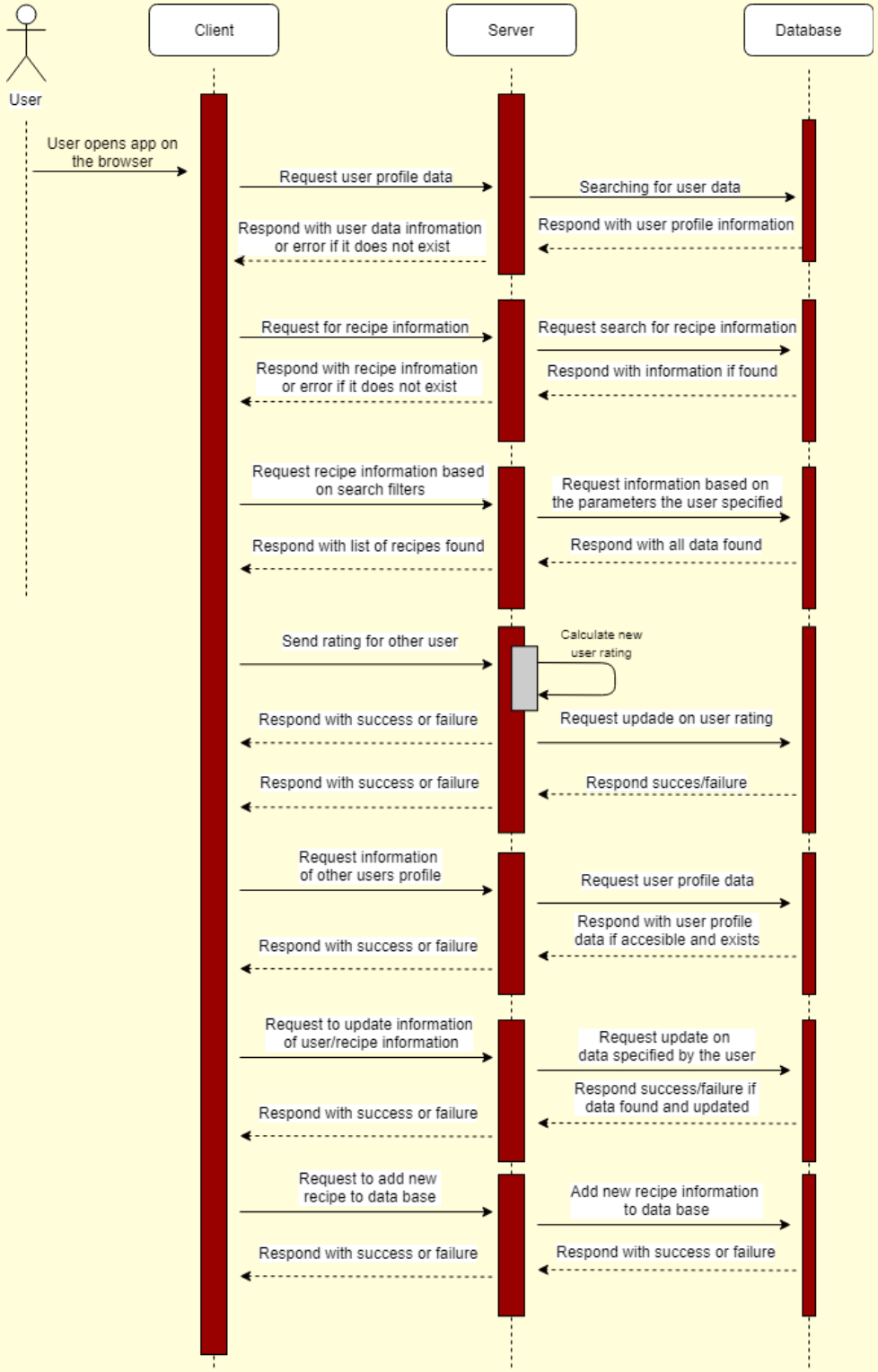
   Options:

   - HTML
   - CSS
   - Javascript

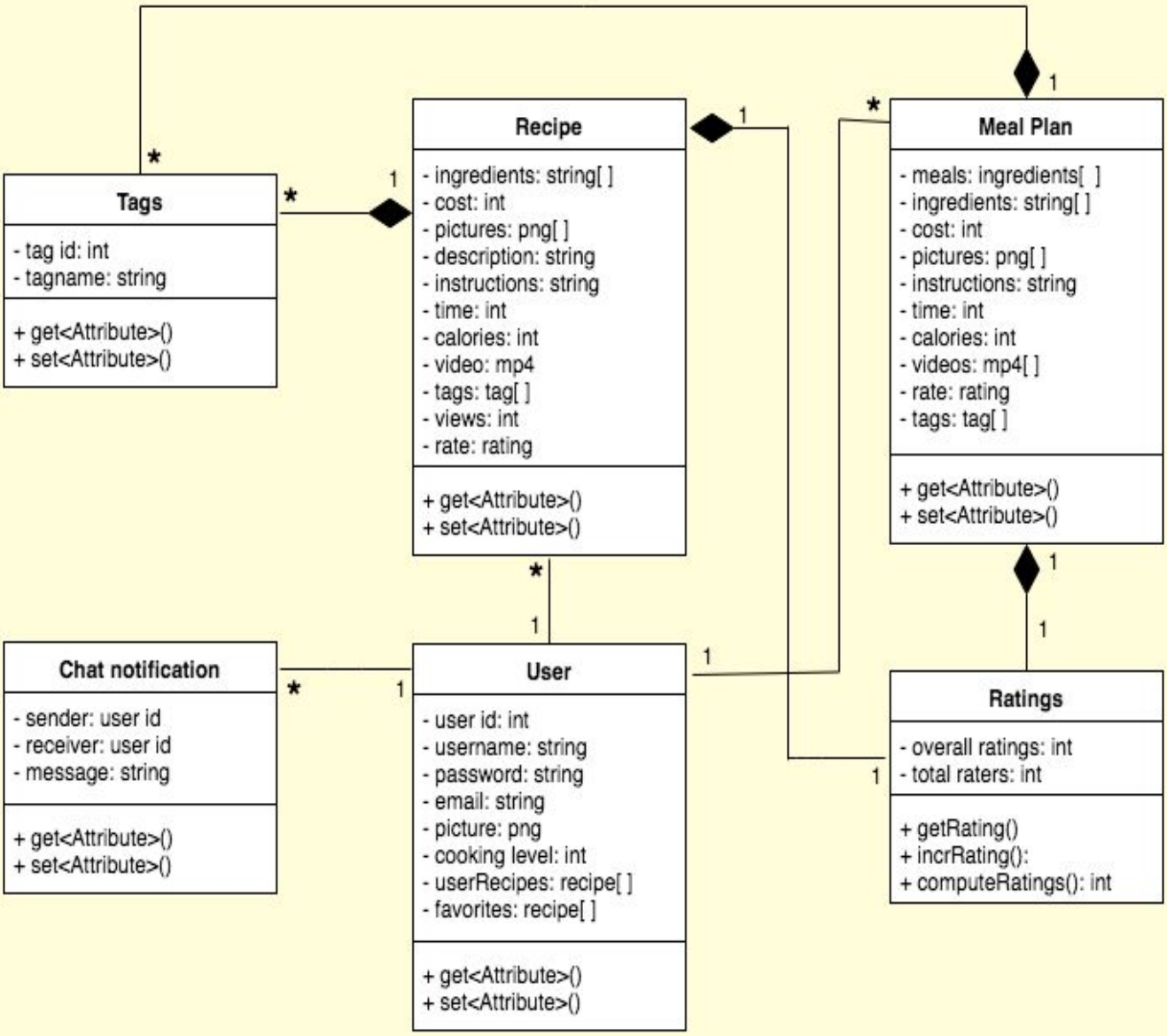
   Choice: HTML, CSS, and Javascript

   Justification: Based on our choice to use Bootstrap as our front-end framework, we have subsequently decided on HTML,CSS, and Javascript as our front-end languages. This is thanks to Bootstrap's native support for these languages. Using HTML and CSS will allow us to build our webpage and bolster it with an attractive design. Additionally, Bootstrap provides extra Javascript components which allow implementation of different interface elements which could be used, if needed.

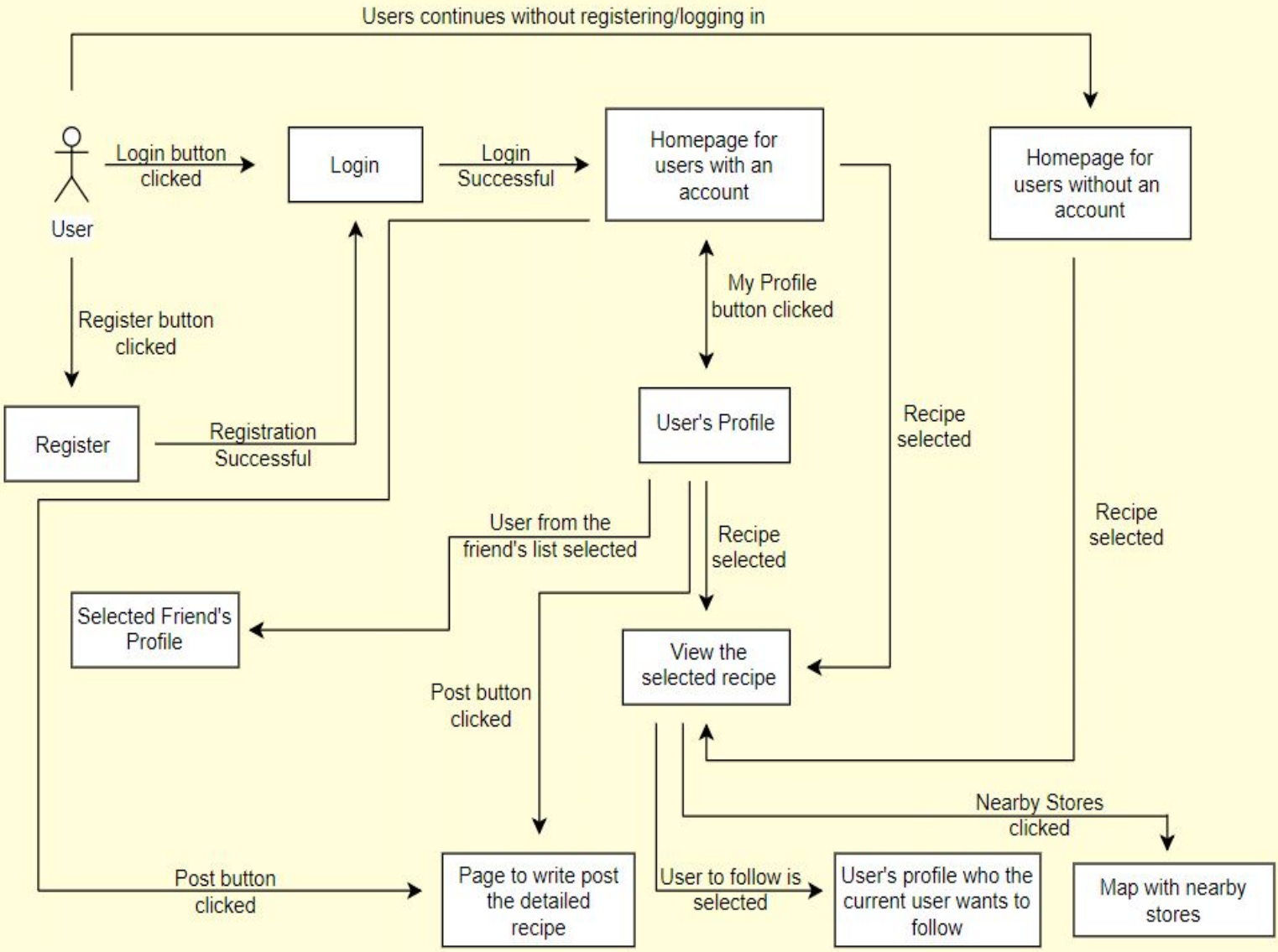
Design Details



Class Design



Navigation Flow Map



Descriptions of Classes and Interaction between Classes

User:

- Object created when account is made
- Each user has a unique user id
- Each user must have their own username, email and password
- Each user has a profile picture
- Each object must contain a user rating, starting at a value of 100
- Each user must have a “chef level” which indicates the level of their culinary ability and knowledge.
- Each user must have a list of favorite recipes, populated when a user presses the favorite button on a recipe
- Each user must have a list of their own created recipes

Recipe:

- Object is created when a user creates a new recipe
- Each recipe must contain all the tags a user assigns to it
- Each recipe must have a list of ingredients
- Each recipe must have a short description
- Each recipe must have a list of detailed instructions
- Each recipe must contain the time it takes to create a recipe
- Each recipe should be able to contain a video of the recipe being made
- Each recipe must have a value for number of views
- Each recipe will have one or more pictures

Meal Plan:

- Object is created when a user creates a new meal plan
- Each meal plan must have a list of recipes

Chat notification:

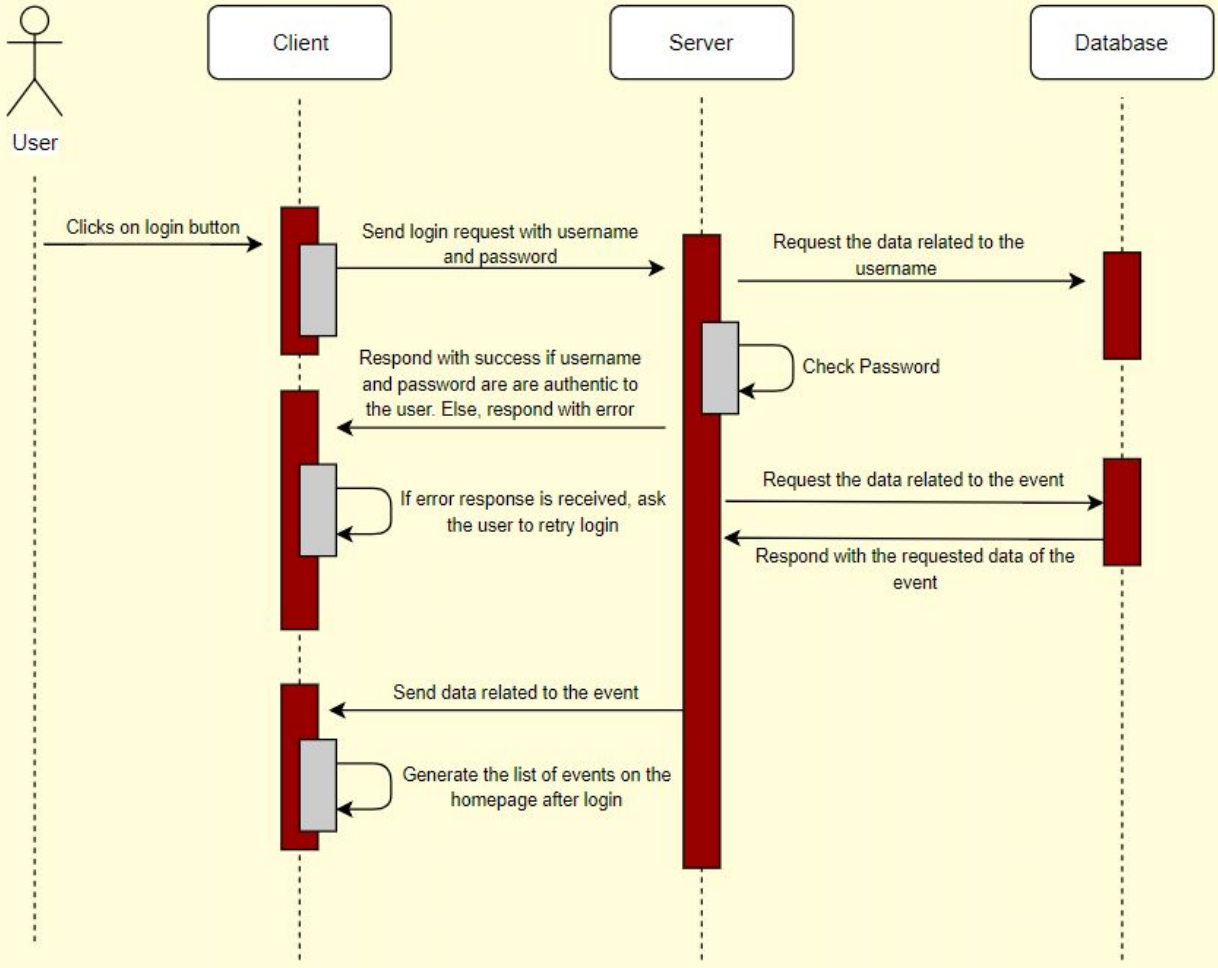
- Object is created when a user receives a message from another user
- Each notification must contain a sender and receiver id
- Each notification must contain a message

Ratings:

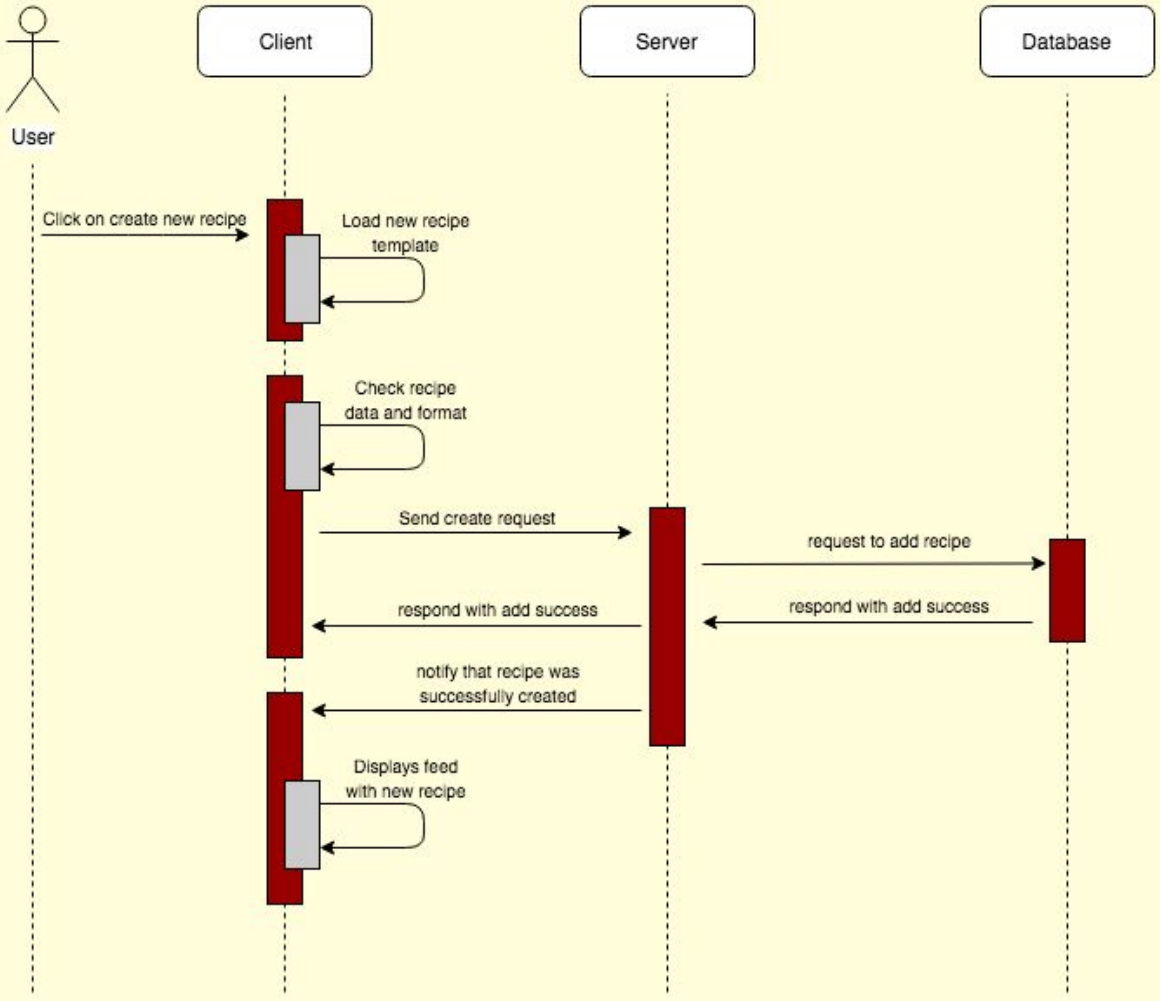
- Object is created when a user creates a recipe/meal plan
- Each rating must have overall rating and total number of raters

Sequence Diagram

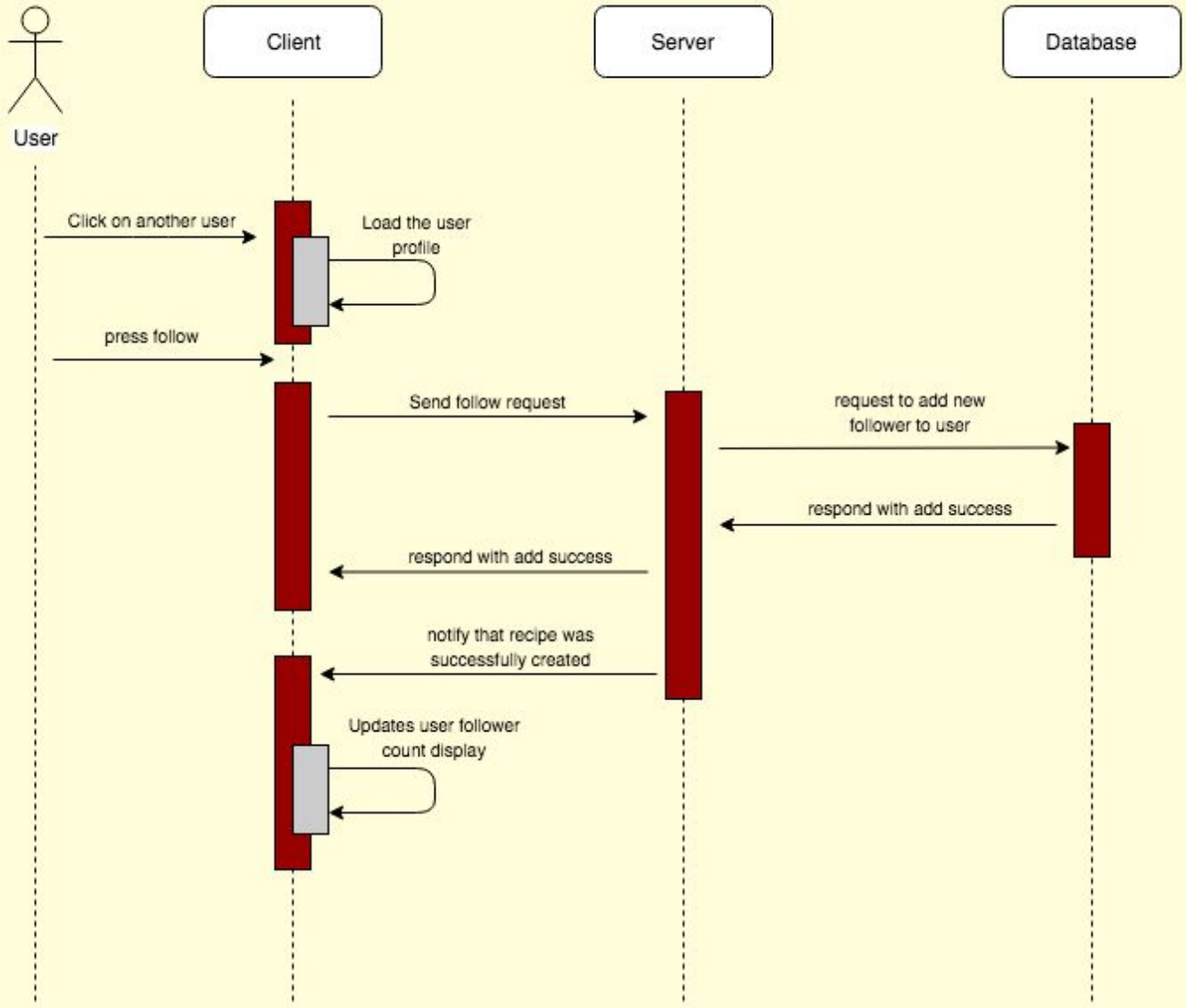
1. **Sequence of events when user logs in**
   When the user clicks on the login button, the client will display the template for the user to enter his/her username and password. The client sends the user's credentials to the server which further requests data related to the user from the user database. The server also checks if the password entered is linked to the username. If the username and password match, the server responds with success, otherwise, the server responds with an error. If the error response is sent from the server, the client asks the user to try logging in again. Unless the username and password do not match, the server will request data from the database and the database will respond with the requested user data. If the login is successful, the data is further sent to the client and the client displays the user's homepage template.



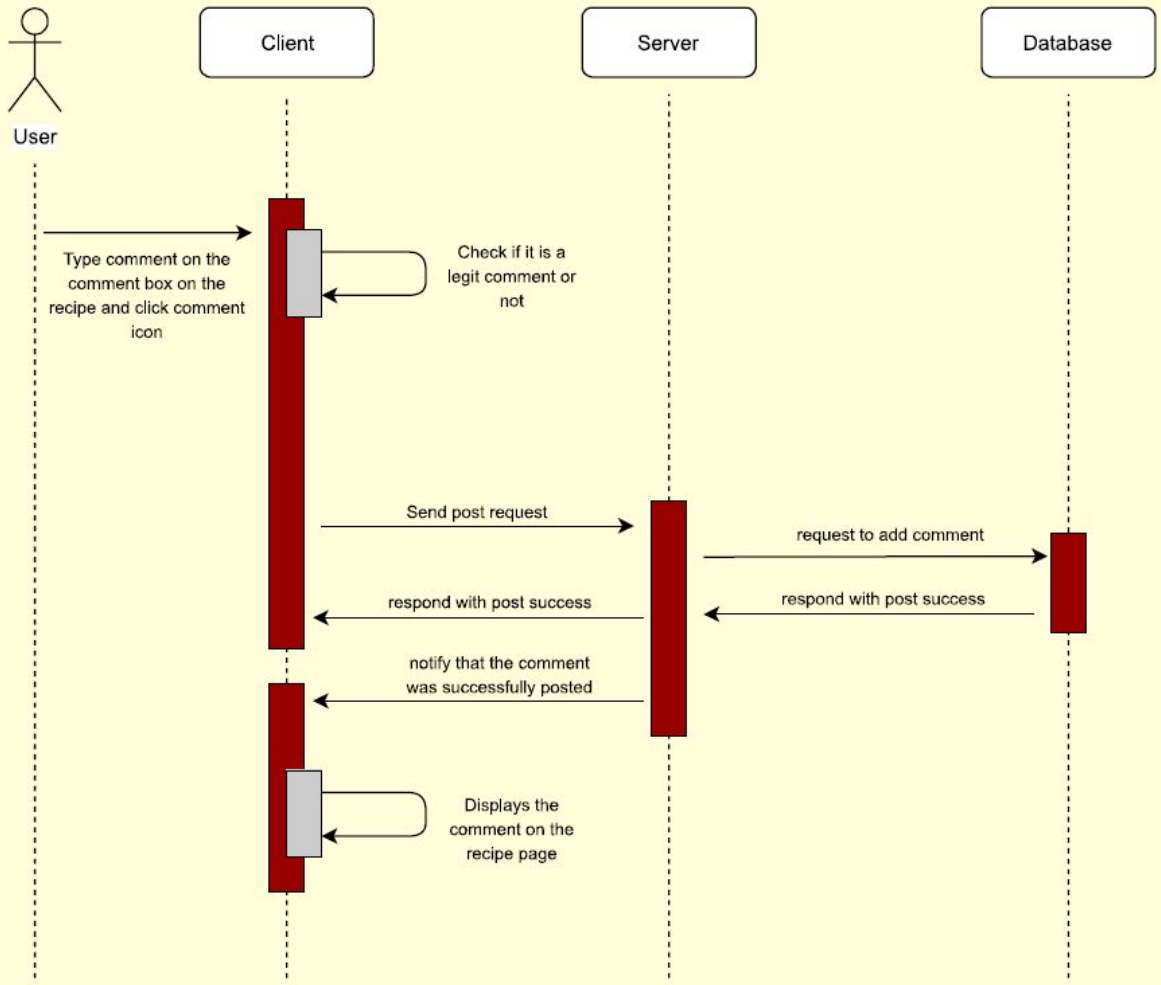
2. **Sequence of events when user creates a new recipe to post**
   When the user clicks on create recipe the client will display the template for the user to input the recipe parameters. Once the data has been introduced by the user the client will check the format before sending it to the server. After the server receives the create request it will pass it on to the database where the new recipe will be created and stored. The database then responds to the server if the operation was successful. The server will respond, informing the client if the operation was successful or not, if successful it will display the new recipe on the users page.



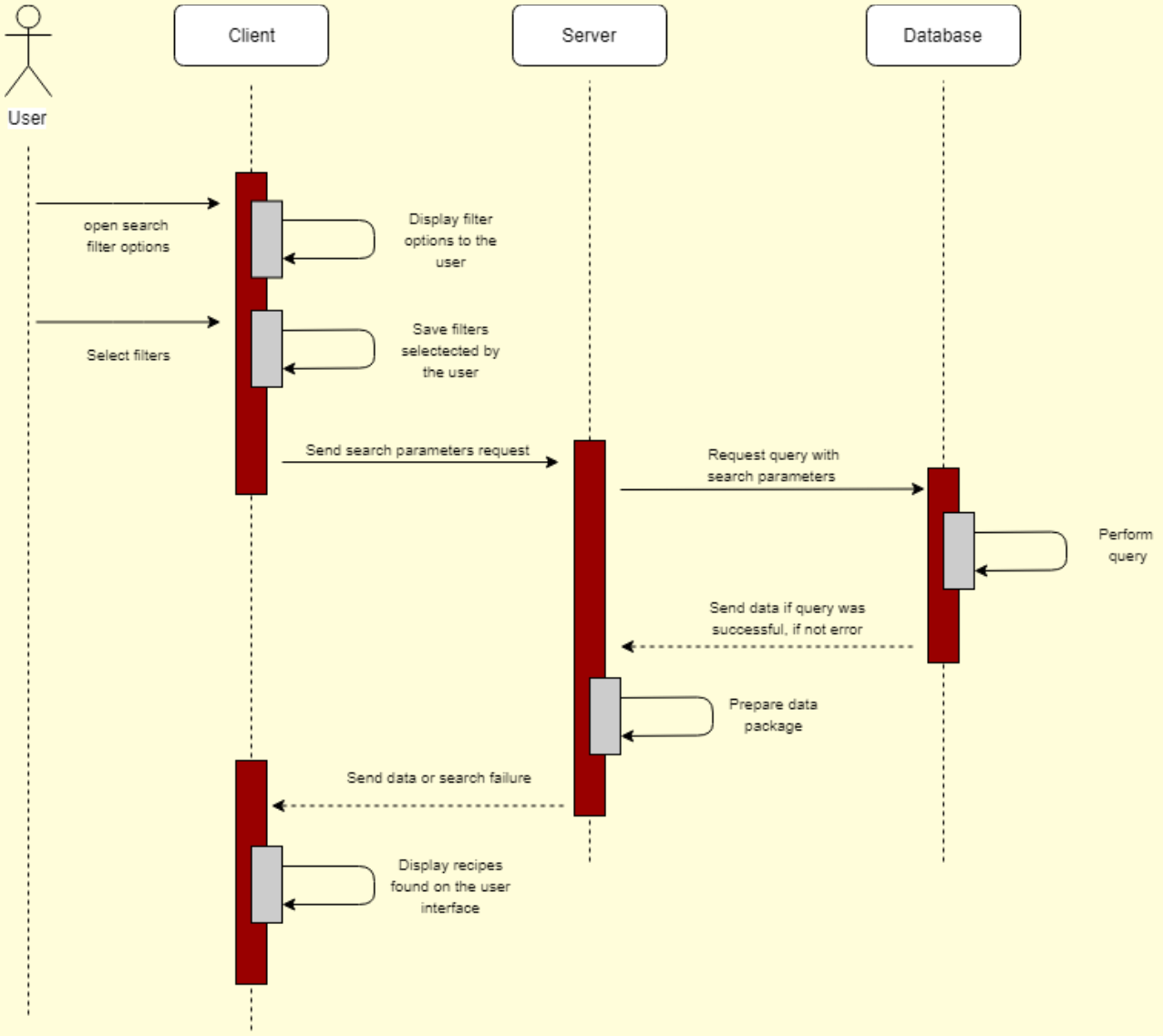
3. **Sequence of events when user follows another user**
   To follow someone, their profile has to be loaded up. Next, the user has to press the follow button. The follow request is then sent to the server by the client. This new follower's information has to be updated on the database.The database then informs if the operation was successful. The server will then send a response to the client which will update the user follower count display.



4. **Sequence of events when user comment on a recipe**
   The user types his/her comments in the comment box at the bottom of the recipe page and clicks on the comment icon to submit the comment. Once the user clicks on the icon, the client will check if the information that the user typed is legitimate to prevent possible cross-site scripting. If the client thinks it is legitimate, it will send the comment server side to connect to the database to store in the backend. Once the comment is successfully stored, the database will give a signal to the server which also tells the client that the comment has been successfully stored. Then the client will display the comment on the recipe page.

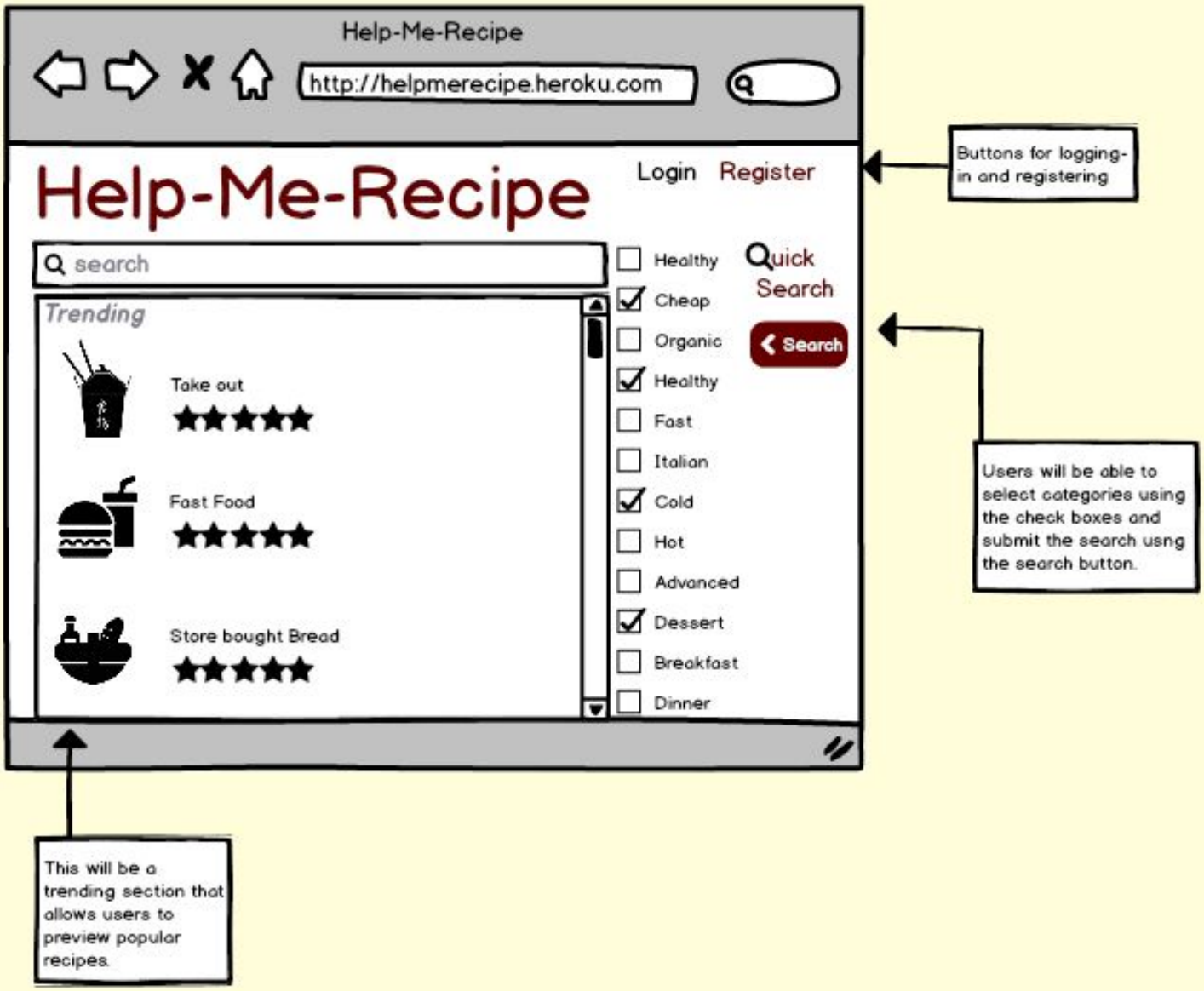


5. **Sequence of events when user searches for recipes using filters**
   When a user wants to search for recipes he/she will have to open the filter option and select the desired search parameters. The client will have to show these options to the user and once they have been selected it will send a request to the server. The server will then take the parameters sent by the client and perform a query request on the database. The database will perform the query and send the result to the server. The server will process this data and send it to the client to be displayed to the user.

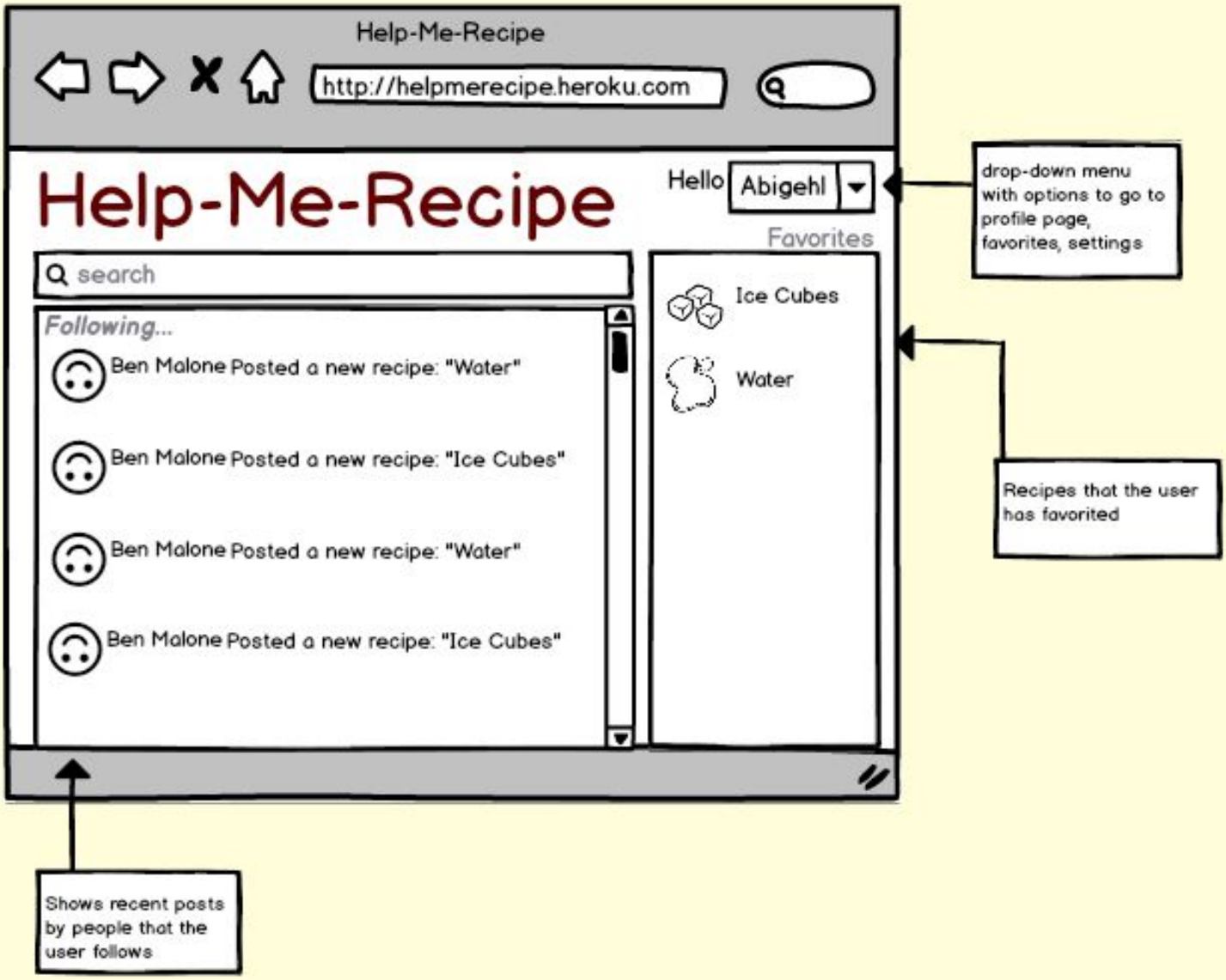


UI Mockup

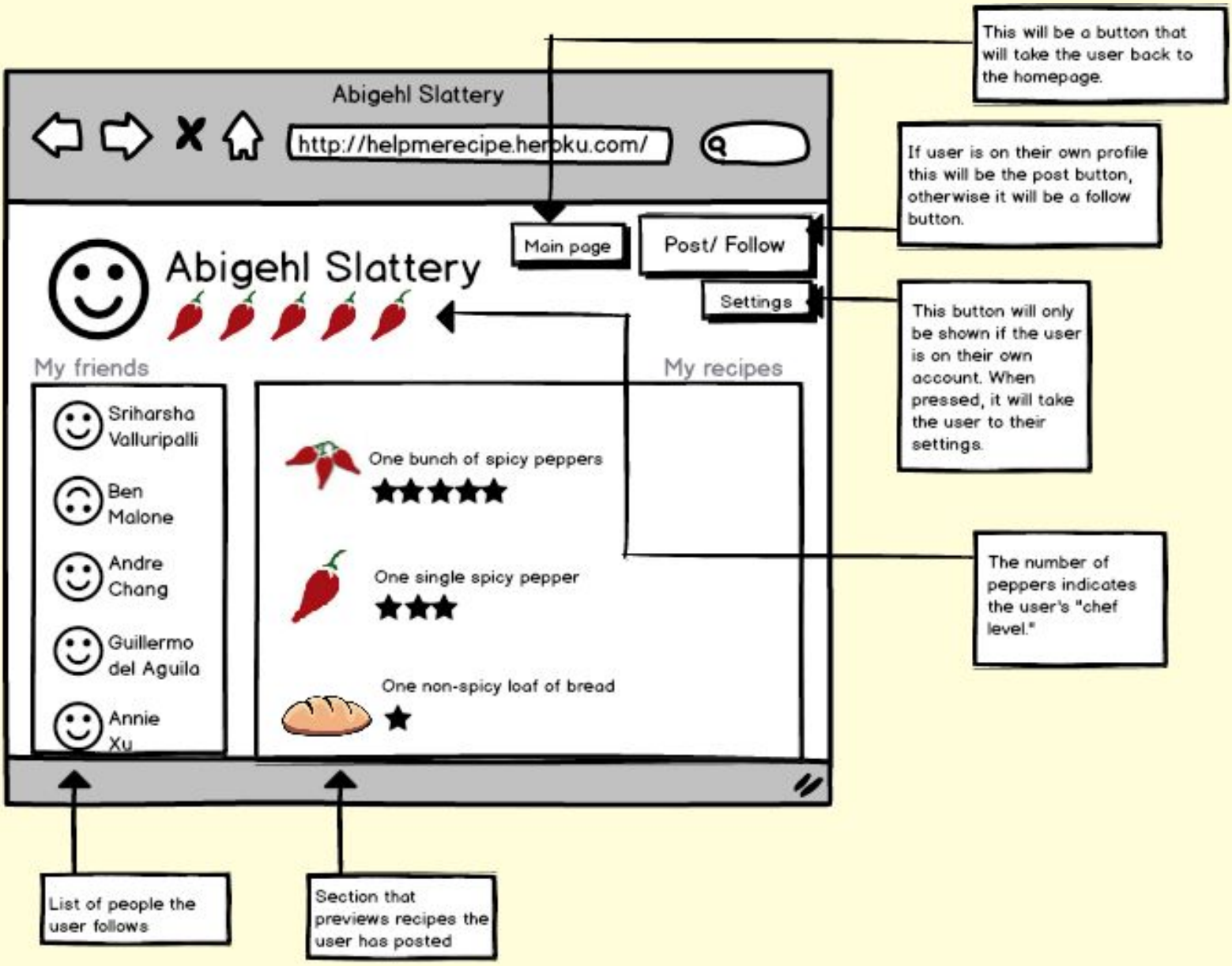
Home when not logged in



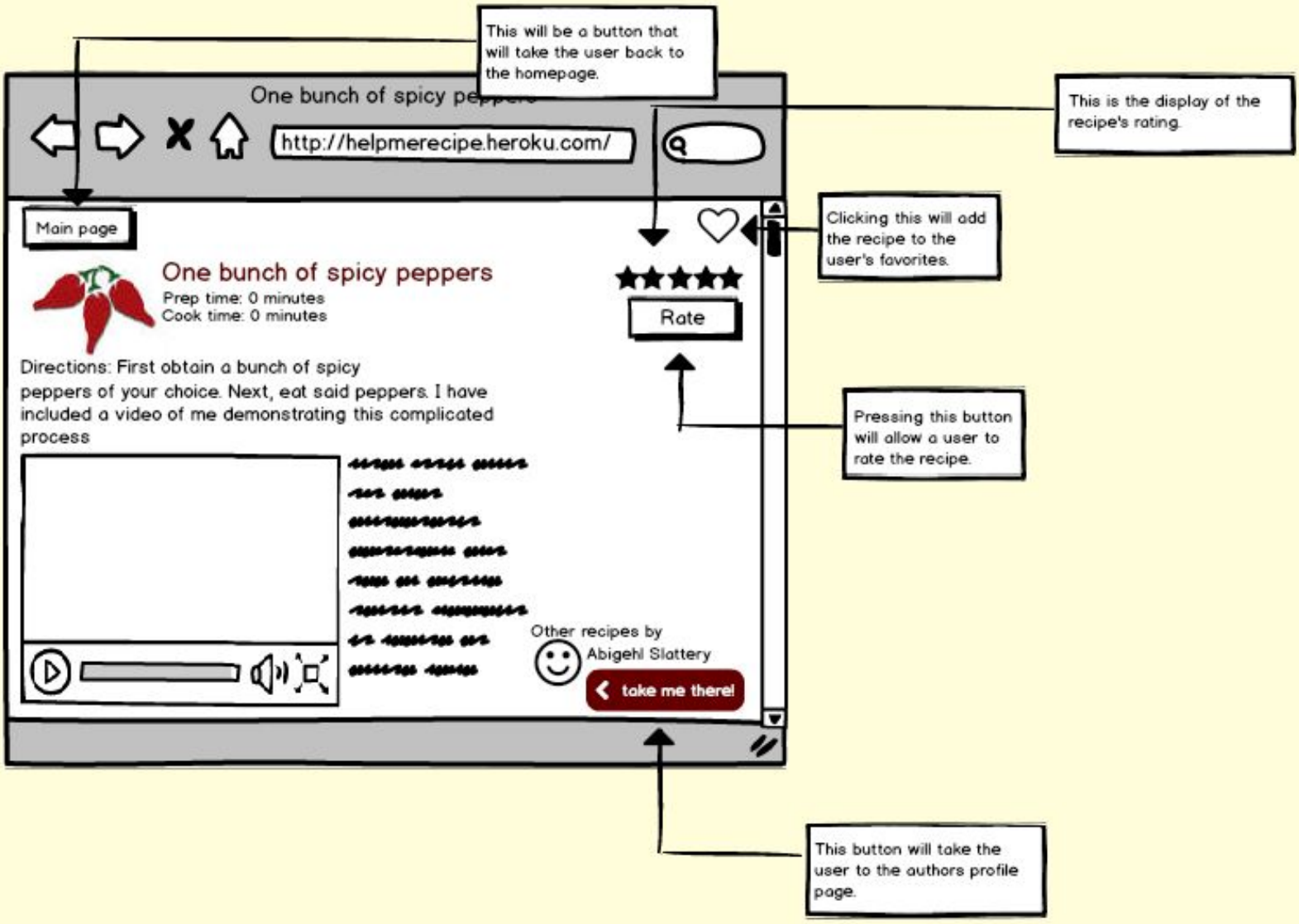
Home when logged in



Profile Page (self or other's)



Recipe Page


This will be a button that will take the user back to the homepage.
This is the display of the recipe's rating.
http://helpmerecipe.heroku.com/
Main page
Clicking this will add the recipe to the user's favorites.
One bunch of spicy peppers
Prep time: 0 minutes
Cook time: 0 minutes
Rate
Directions: First obtain a bunch of spicy peppers of your choice. Next, eat said peppers. I have included a video of me demonstrating this complicated process
Pressing this button will allow a user to rate the recipe.
Other recipes by
Abigehl Slattery
take me there!
This button will take the user to the authors profile page.

Weekly Meal Plan Page