Team Deliverable 4

Members:

- Vigan Ejupi
- Avnish Ozarkar
- Chris Smith
- Landon Ripper
- Lina Urrego
- Brayden Snyder

November 19th, 2024

ITIS 3300-092

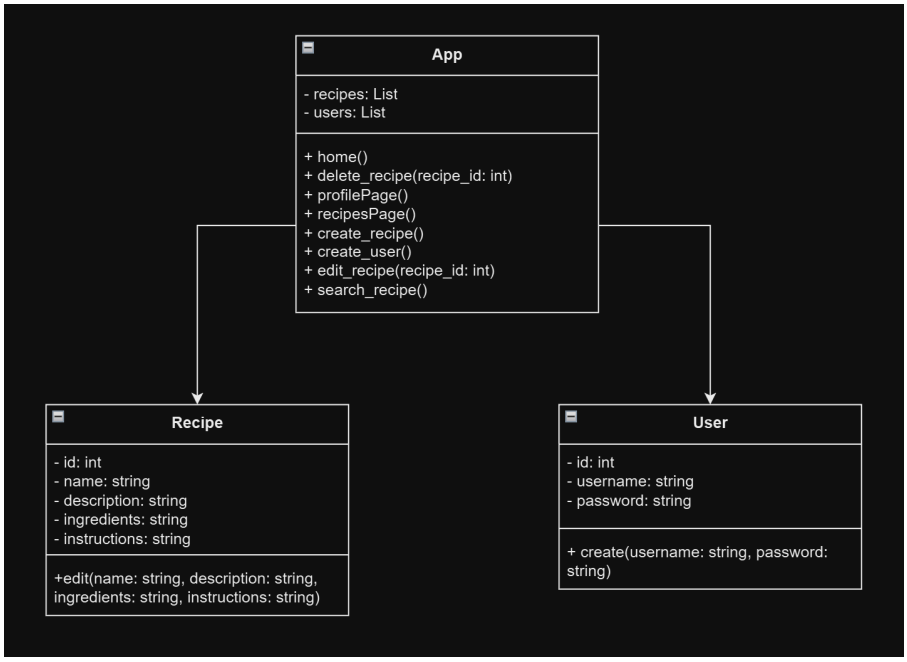
Report:

a. **Requirements:**
Our list of requirements have stayed the same from our 2nd Team Deliverable.

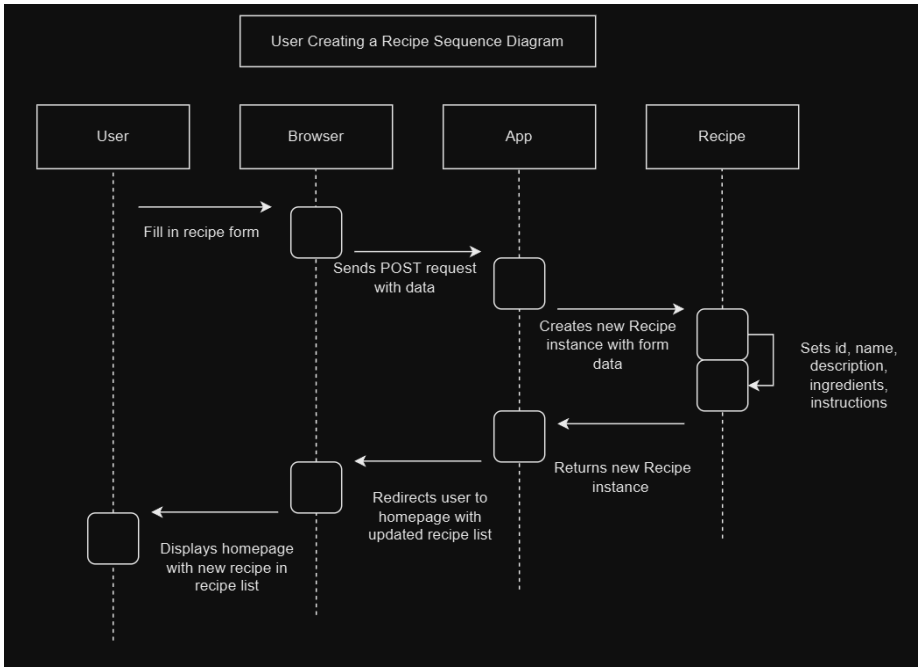
- We plan on focusing on developing the database and the API for our application. By this point, we are going to come up with the attributes that we plan on using for the data that we want to use for the app so actually creating the database for the app should hopefully be pretty simple.
- For the API, we plan to create various endpoints that will allow the application to get the data that is needed from the database and will allow users to make additions and edits to the data by using the application.
- We also plan on keeping user security in mind when working on this part of the project and will ensure that user data is only accessible by the users that own that data. The only data that should be publicly accessible on the application is the recipe data.
- The creation of the database and the API are both necessary in getting the application fully functional so they will be the most critical tasks for this phase of the project. Security is a very important aspect of the project as well, but that cannot be fully implemented until we are able to be finished with the API and database.

b. UML design for phase 2:

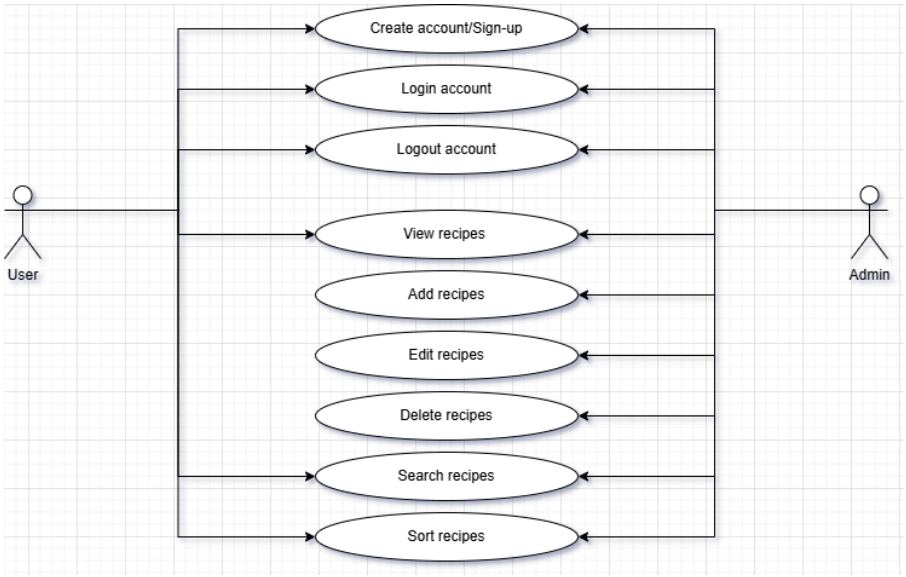
Here is our Class Diagram:



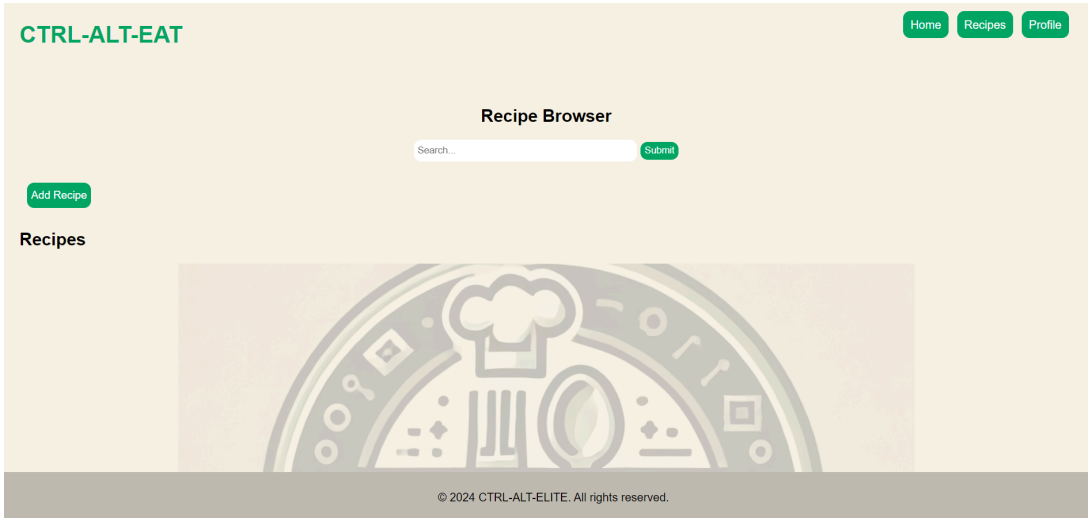
Here is our Sequence Diagram:



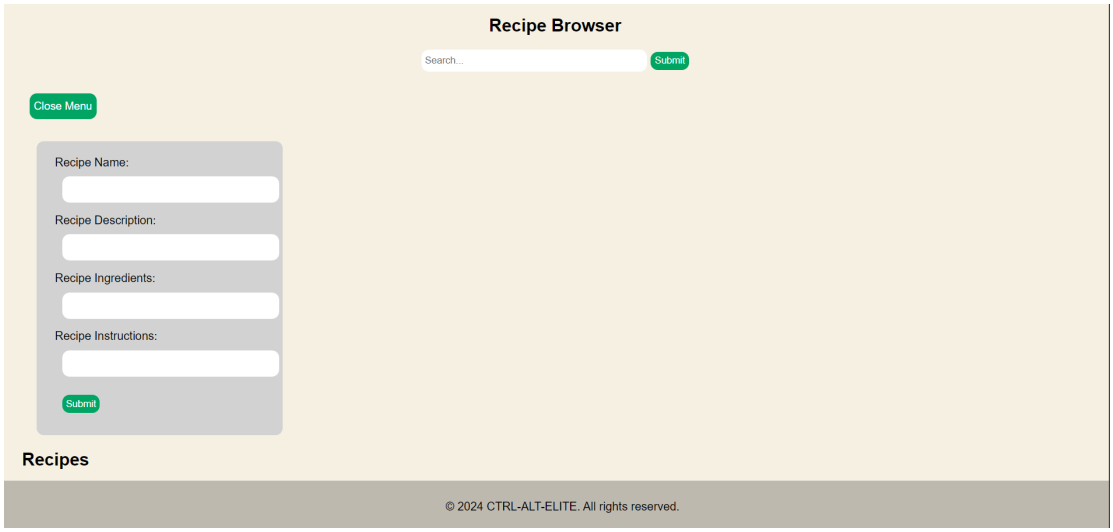
Here is our use case diagram:



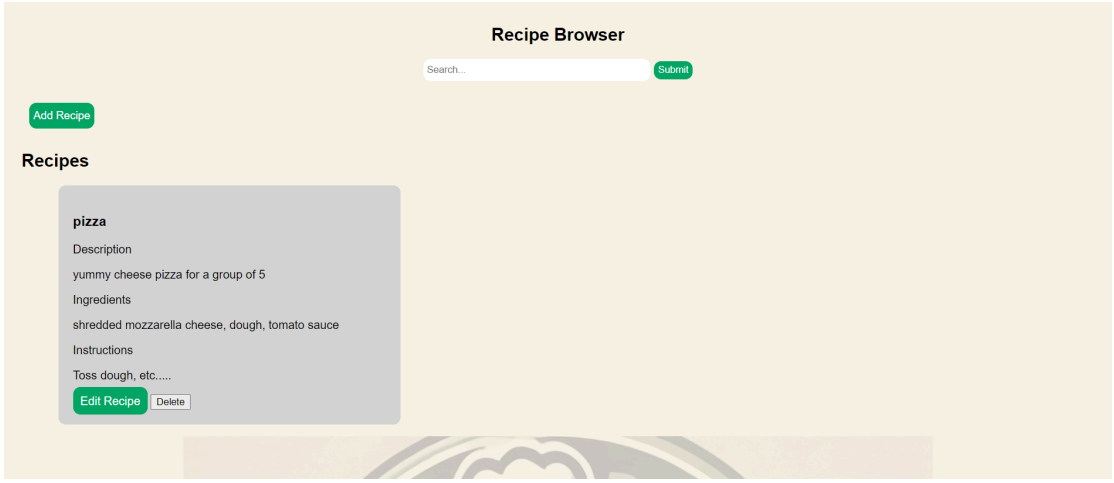
c. User manual on how to use program (can include screenshots):



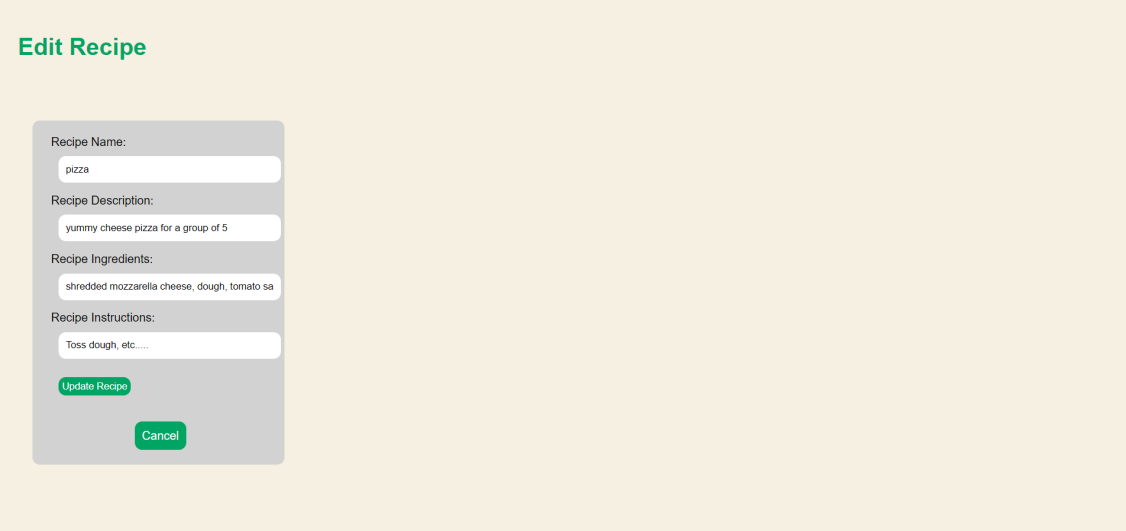
This is the landing page for the user. They will start off by selecting the green “Add Recipe” button on the left. Once the user selects it, there will be a menu that pops up. This is where they can enter their recipe name, recipe description, recipe ingredients, and their step by step instructions.



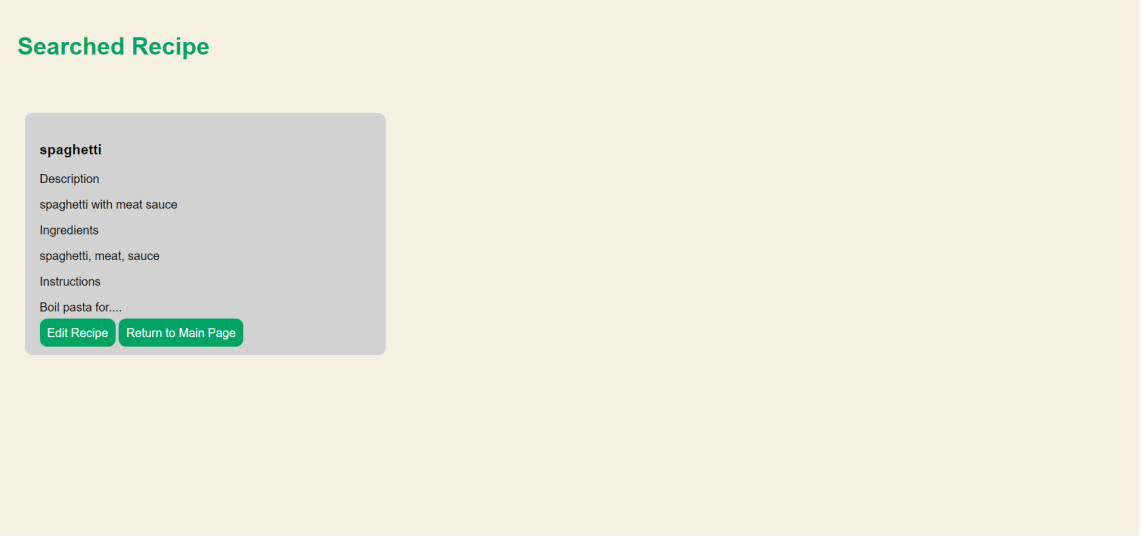
It will look like this.



Once the user selects the “Submit” button, all of their information for the recipe they created will be displayed which will then appear on the main list. The user then has the option to delete the recipe or edit the contents within the created recipe.



If the user clicks on “edit recipe”, the edit menu will prompt up and the user can then update or make the necessary changes to the recipe. Once finished they can select the “Update Recipe” button or choose to cancel.



If a user has multiple recipes created, they have the option to search up any recipe they choose to view. They will enter the name of the recipe in the search bar using any keywords and then select the “search” button. The results will then be displayed and the user can choose to edit the recipe they’ve searched or just view it and return to the main page.

d. Instructions on how to run program and test cases:

1. Make sure python is installed
2. Install Flask and you can do that by using the command: **pip install flask**
3. Download, navigate, or git clone the Project Folder from https://github.com/linalu12/itis-3300-CTRL-ALT-ELITE-Recipe-App.git

4. open a terminal (preferably Command Prompt) and type in the command: cd path_to_project/src
5. Then start the Flask Application by typing: flask run
6. In the output it'll look something like http://127.0.0.1:5000, CTRL + click or COMMAND + click in order to access the application on the browser.

The application should now display the home page. See part C for testing case scenario.

e. Test Cases (Unit Tests):

1. Test Case 1 - Add a new recipe
    a. Functionality tested - Recipe Management
    b. Input - Recipe name, Ingredients, Instructions
    c. Expected Output - Recipe shows in the recipe section with appropriate dynamic formatting.
    d. Pass / Fail
2. Test Case 2 - Editing an existing recipe
    a. Functionality tested - Recipe Management
    b. Input - Updated recipe details
    c. Expected Output - Recipe changes shown in the recipe section.
    d. Pass / Fail
3. Test Case 3 - Delete a recipe
    a. Functionality tested - Recipe Management
    b. Input - Recipe ID
    c. Expected Output - Recipe from the recipe section no longer appears
    d. Pass / Fail
4. Test Case - 4 Search recipe by exact name (Refined searching not yet implemented)
    a. Functionality tested - Recipe Search
    b. Input - Full recipe name
    c. Expected Output - The webpage shows search results based on imputed recipe name.
    d. Pass / Fail
5. Test Case 5 - User interface navigation
    a. Functionality tested - User Interface
    b. Input - Clicking navigation menu
    c. Expected Output - Moves to respective page name
    d. Pass / Fail
6. Test Case 6 - Page responsiveness
    a. Functionality tested - User Interface
    b. Input - Different webpage resolutions
    c. Expected Output - page adjusts layout dynamically according to resolution

    d. Pass / Fail
7. Test Case 7 - Search recipe that does not exist
    a. Functionality tested - Recipe Management
    b. Input - Recipe name
    c. Expected Output - Blank recipe search page with no results
    d. Pass / Fail
8. Test Case 8 - Search recipe that was previously deleted
    a. Functionality tested - Recipe Management
    b. Input - Recipe name
    c. Expected Output - Blank recipe search page with no results
    d. Pass / Fail

f. Feedback received during coding inspection and actions taken based on feedback:

- Implement and utilize currently unused SQL that was planned for use later
- Create more comments that are primarily focused on why the code is there instead of what it is
- Divide up our main Python API
    - Make a file for each API functionality
- Delete any unimportant commits that do not contribute critically to the project

g. Reflection:

So far, we have implemented prototypes of the functionality regarding recipe management that we plan to fully add when we add the database implementation in the final product. Using the website, users are currently able to add, edit, delete, and search recipes that they create as well as simulate account creation. Overall, this initial deliverable gave us a good view on what the final product will look like when we implement the database, but it is not good enough to be called a finished product on its own. However, for the objectives that we laid out for the first deliverable of this project, we have been able to achieve what we wanted to achieve. We plan to update the current base functionalities that we have created in order to make the website as a whole more user friendly and fit better with the vision that we have for this website. These plans include allowing the user to enter in keywords to search for a group of recipes rather than just searching for one at a time based on the whole title, allowing users to view the recipes of other viewers and save them to their recipe library, create a working login system and allow users to navigate the website using their own accounts, and allowing the users to input ingredient measurements when they are creating their own recipes.

h. Member contributions:

Member name	Contribution description	Overall contribution	Note
Vigan Ejupi	Requirements, and	20%	

	feedback received during coding inspection		
Avnish Ozarkar	reread the whole report for errors	20%	
Chris Smith	Reflection	20%	
Landon Ripper	Test Cases	20%	
Lina Urrego	GitHub files	20%	
Brayden Snyder	none	0%	We have not heard from Brayden in a long time. We have reached out to him.