

Smart Grocery Coding Project Report

Group 2: Hrishikesh Badve, Sudhanshu Basuroy, Utsav Sharma, Mridvika Suresh

Project Description

The project is a platform designed to assist home cooks by suggesting recipes based on their available ingredients. It features a vast collection of recipes for users to explore and discover. It falls under the non-technical domain of food technology and the technical domain of mobile application development. It helps users discover new recipes and manage their grocery inventories. This project is based on the development project from Group 25 - CS440, Fall 2022. It implements functionality such as browsing through recipes, users adding and sharing their own recipes, and filtering recipes by dietary preferences.

Comparison with Original Project Design Document

Our prototype differs from the original document, which required the application to connect with nearby grocery stores and recommend recipes based on product availability. This functionality was not implemented due to the lack of easily accessible APIs for checking grocery availability. Instead, the team allowed users to manually enter grocery ingredients to receive recipe recommendations, deviating from the original plan. The original project design also included a barcode scanner feature. Implementing a barcode scanner would require specific libraries and a product information database, which were not included in the prototype. Despite these differences, our application can find nearby grocery stores, recommend recipes, and allow users to view recipes from other users.

Testing

We conducted 14 tests in total. They included: Registration functionality, successful login, invalid login, browse recipe filter, recommend recipe functionality, favorite button functionality, ratings functionality, add items to shopping list, cross out items in the shopping list, add recipe functionality, explore recipe functionality, explore recipe favorite button functionality, recipe search bar, nearby stores button.

Recommendations and Conclusions

2 of the 14 tests conducted failed. The following conclusions can be drawn:

- Ensure the addition of user feedback for all possible operations that the user may complete.
- In both cases, the functionalities are working from a technical perspective, but they are lacking in terms of user experience. Providing clear and immediate feedback to the user is a key aspect of user interface design and can greatly enhance the usability of our application.

Project and Issues

- **API Key Security Issue:** The Edamam API key is currently hardcoded in the application, making it vulnerable to security breaches. If the key is extracted, it could be misused to access the API.
- **Email Verification:** The application currently lacks email verification functionality. This leaves it susceptible to the creation of fake accounts, potentially leading to backend flooding.

Features for future releases

- **Login Persistence:** The application currently does not store the login state of users, requiring them to log in every time they reopen the application.
- **Share Recipes:** A feature for users to share recipes was planned but was moved to the backlog due to time constraints.
- **Moderation Backend:** A system for human moderators to review recipes (for user created recipes) is planned but not yet implemented.
- **Recipe Reviews:** Users can currently rate recipes on a 1-5 star scale. Future releases plan to allow users to provide detailed text reviews.
- **Community Forums:** Plans are in place to include a community forum for users to discuss recipes and food-related topics.
- **API Collaboration:** The application currently uses the free tier of the Edamam API, with plans to integrate an API from a food/recipe brand like Epicurious, Tasty, or Food Network in the future.
- **Nutritional Information:** While nutritional information is available from the Edamam API, a solution is needed to allow users to select recipes based on this information.
- **Integration with Grocery Store/Delivery Apps:** Integration of grocery delivery services like Instacart, Jewel-Osco, etc. that allow users to view availability of grocery items and have items from their shopping list delivered to their homes.

Project Retrospective

Flutter/Dart was chosen for application development due to its benefits in rapid prototyping and platform independence. This approach saved considerable time and effort compared to a native approach like Java/Kotlin for Android or Swift for iOS. We adopted a strategy of dividing work into independent modules and assigning them to individual members each release. This approach eliminated bottlenecks caused by dependencies and waiting for others to complete their tasks. However, it led to some work imbalance, with some team members having to put in more hours for features that required more setup and programming. In the future, our team aims to find a better work split balance, which depends on improving effort estimates. An architecture like Model-View-Controller would be beneficial for easier application updates. Using an incremental design approach resulted in significant rework for the UI and the backend, leading to a considerable amount of dead code. This was also an area for improvement in future projects.