ICSI499 Capstone Project Report

Meal Planner App

Project Team

Bhanu Kakulla
Joseph Lohman
Oluwasogo Enoch Awofeso
Sandeep Vepuri
.....
College of Engineering and Applied Sciences
University at Albany, SUNY

Project Sponsor

Dr.Pradeep Atrey
University at Albany
University at Albany - State University of New York

May 5, 2024

Acknowledgements

As a team, we express our deepest gratitude to Professor Pradeep Atrey for his invaluable support and guidance throughout the development of our Meal Planner App. Professor Atrey's mentorship, regular meetings, and structured checkpoints played a pivotal role in shaping our project's direction and ensuring its success. We as a team were able to coordinate and have a projection of our goals and milestones to be completed. We are truly honored to have had the opportunity to collaborate with him on this endeavor. It was a honor to create the Meal Planner App!

Abstract

The Meal Planner App is a Android-based application aimed at simplifying meal planning and enhancing user experience. Leveraging data from the Spoonacular API, the app offers personalized meal suggestions based on user preferences and dietary requirements. Users can register and manage their profiles seamlessly, allowing for customization of meal plans and preferences. The app's intuitive user interface guides users through the process of selecting, saving, and exploring meal options with ease. Backed by robust algorithms and tools, the Meal Planner App aims to revolutionize the way users approach meal planning, creating culinary creativity in how the user prepares for cooking.

Contents

1 Problem Analysis

- Meal planning can be a time-consuming and daunting task for many individuals, especially those with dietary restrictions or specific health goals. The problem we aim to address with the Meal Planner App is the lack of efficient and personalized tools to streamline the meal planning process and provide users with tailored meal suggestions. This problem is significant as it directly impacts users' daily routines, health outcomes, and overall well-being. Moreover, with the growing emphasis on healthy eating and the increasing prevalence of dietary restrictions, there is a growing need for accessible and user-friendly solutions in the realm of meal planning.

1.1 Existing solutions

Existing solutions for meal planning often lack personalization, relying on generic meal suggestions that may not align with individual preferences or dietary requirements. Furthermore, these solutions may not offer seamless integration with user profiles, making it challenging for users to manage their meal preferences and track their progress effectively. Key weaknesses of existing solutions include limited customization options, lack of real-time updates, and insufficient support for dietary restrictions.

- **MyFitnessPal**
  - Users have to manually enter their meal information, which can be time-consuming, leading to inaccuracies [10].
  - Premium services suggests that some features are behind a paywall, which limit the full functionality of the app

- **Mealime**
  - Users can input their grocery list, and the app generates meals aligning with their goals and dietary restrictions [8].
  - The app offers guidance on meal preparation and comprehensive nutritional information to support users in their meal planning and achieving their health objectives [8].
  - If the app has a limited number of recipes, users may quickly cycle through all available options, leading to a repetitive meal plan.

1.2 Core Idea and Solution

The core idea of our solution is to provide users with a user-friendly, customize-able meal planning app that integrates seamlessly with their dietary preferences and cooking expertise. Key characteristics of our proposed solution include personalized meal recommendations, real-time updates, and comprehensive support for various dietary restrictions.

Our solution stands out from existing ones by offering a higher level of personalization, integration with user profiles, and the ability to tailor meal suggestions based on individual preferences and cooking expertise. Furthermore, our app provides a wide range of meal options, addressing the limitations of current solutions.

1.3 Key Contributions

- Streamlining the meal planning process for users with diverse dietary preferences and restrictions.
- Enhancing user satisfaction and adherence to dietary goals through personalized meal suggestions.
- Providing a user-friendly interface and seamless integration with user profiles for efficient meal management.

1.4 Organization of this Report

This report is structured to provide a comprehensive overview of the problem domain, existing solutions, our proposed solution, its novelty and distinctiveness, key contributions, and the organization of the project. Each section builds upon the previous one to present a cohesive understanding of the project's scope and objectives.

2 Proposed System/Application/Study

2.1 Overview

This section provides an overview of a meal planner app developed using Android Studio with Java. The aim of the app is to simplify meal choices for users by offering a unique solution. It allows users to quickly generate meal options, provides recipes and cooking instructions, and suggests sides tailored to their meal choice. The section is organized to offer a comprehensive understanding of the meal planner app. It covers functional and non-functional requirements, user classes, and architecture. Additionally, it provides a detailed breakdown of key components such as the user interface, database management, and integration of external APIs for recipe suggestions.

2.2 Project Requirements

2.2.1 Functional Requirements

Registration and Profile Management

- Functional Requirement: User Registration
  - For registration and sign-in functionalities, our app utilizes a cloud-based solution leveraging Firebase. This platform enables seamless user sign-up for new users and securely stores existing users' data for subsequent logins. By employing Firebase, we effectively reduce local data storage requirements and were able to improve the user experience, by focusing on additional features.
- Functional Requirement: Profile Management
  - Profile management in our app combines features for editing personal information and setting dietary goals. Users can update their details while also monitoring health goals. Additionally, we encourage user engagement by enabling them to save their favorite meals within the app for easy access whenever they enjoy one of our suggestions.

Meal Suggestions

- Functional Requirement: Meal Suggestions
  - For meal suggestions, our team opted for a cloud-based approach utilizing the Spoonacular API. This decision was driven by the API's extensive meal database, which includes detailed information about meals, ingredients, cooking instructions, and filters for different dietary preferences, allergies, and intolerances. Integrating Spoonacular allowed us to focus on manipulating the data to meet our app's goals, such as implementing features for favorites, user-added meals, and side options.

User/Meal Preferences

- Functional Requirement: User/Meal Preferences
  - User and meal preferences are essential features added to enable users to filter their meals based on various criteria such as the time of day, dietary preferences, and intolerance's. These preferences are tailored to the user's personal life and goals. Additionally, to ensure variety, we implemented a feature where meal suggestions are not repeated for 7 days once clicked. Users can also favorite meals they enjoy for continued access at any time.

2.2.2 Non-Functional Requirements

Performance

- Our app prioritizes swift responses to user interactions and inputs while maintaining low overhead. We emphasize minimal loading times for meal suggestions, details, and navigation within the app.[9]

Scalability

- Our app is designed with scalability in mind, accommodating an expanding user base, additional features, and a growing library of meals. We are committed to maintaining optimal performance and ensuring a smooth user experience even as the app evolves.[9]

Security

- We integrated Firebase and Spoonacular into our app, leveraging their established systems with extensive security measures. This approach ensures that our users are provided with the same level of protection throughout their interactions with the app.[9]

Compatibility

- By utilizing Android Studio, we ensure development compatibility from day one for a wide range of Android OS versions and screen sizes. This guarantees a consistent user experience across all devices, regardless of their specifications.[9]

Data Privacy

app is committed to compliance with all data protection regulations. We prioritize transparent data handling practices and obtain user consent for any data collection and processing activities.[5][9]

2.2.3 User Classes

User

- The user class comprises individuals seeking a straightforward solution for meal decisions, varying in food preferences, dietary restrictions, and intolerance's. They rely on the app to provide personalized meal recommendations based on these factors. Users interact with the app by creating an account, selecting a meal type, and choosing from three random options displayed. Additionally, they can save favorite meals, add their own, or track their weight and set goals based on personal preferences and objectives.

Admin

- The admin user class in a meal suggestion app comprises individuals responsible for overseeing and managing various aspects of the platform. Admins have access to the backend systems, enabling them to monitor user activities, manage content, and ensure the smooth operation of the app. Their duties include monitoring databases, maintaining connectivity to APIs, and introducing features to enhance user engagement. Admins also handle user inquiries, troubleshoot technical issues, and implement updates or improvements to optimize the app's performance.

2.3 Technical Design

System Flow:

- Our original system flow design, created at the project's start, has proven invaluable for tracking progress and adhering to our initial goals. In our final version, we introduced a profile page enabling users to set goals and update personal data. Within the "Breakfast," "Lunch," or "Dinner" options, we expanded choices such as adding custom meals while retaining features like generating new meals and accessing saved ones. Users then select the best-suited meal from three options, which are accompanied by ingredient details, preparation instructions, and three related side dish suggestions. Users can access information as they would with their main dish.[7][1]

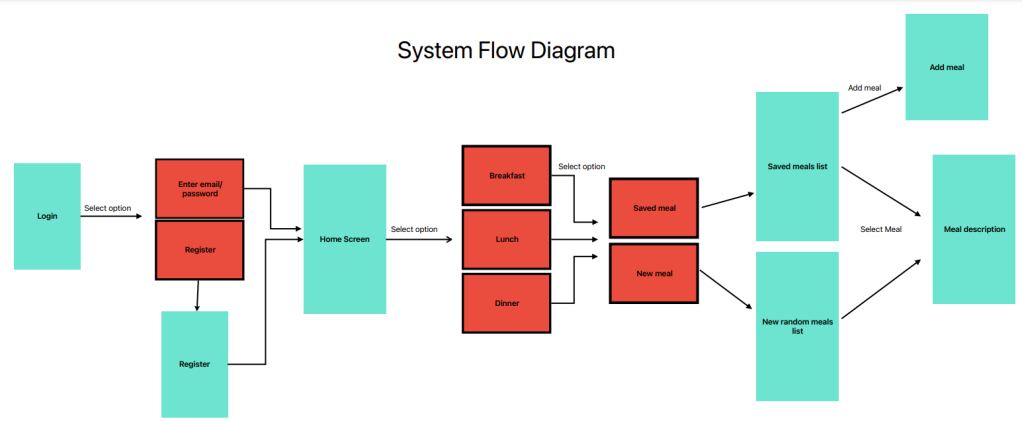

Figure 1: System Flow Diagram

Data Flow:

- Our Data Flow concept underwent more alterations compared to the system flow, though the login process remained consistent throughout the project. Users log in or register, and their data is transmitted from our device to Google Services Firebase servers for authentication and secure storage. Our original design centered on building our own database, but integrating Spoonacular into our app proved to be a better option, especially for prioritizing non-functional goals. With the Spoonacular API, we could focus on features, scalability, performance, and manipulating incoming data as easily as with our own database.[7][1]

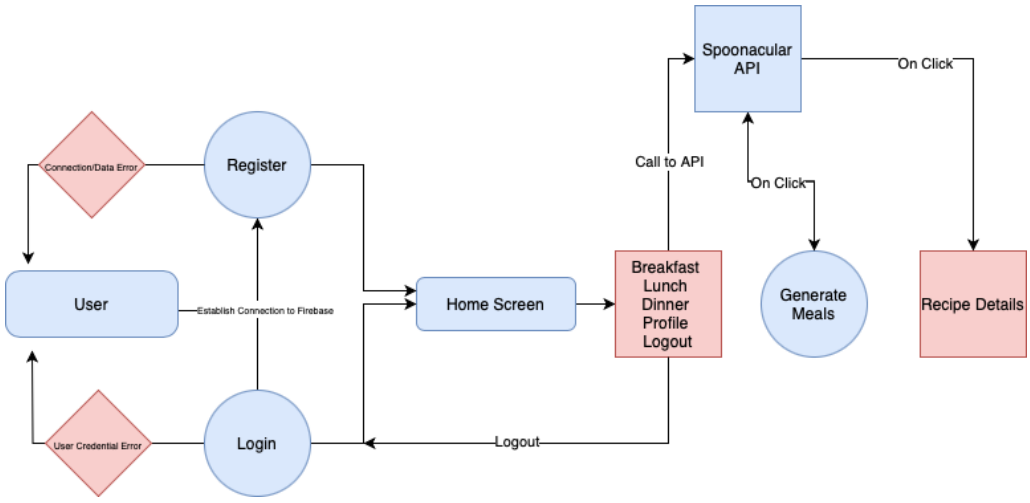

Figure 2: Data Flow Diagram

GUI:

- Similar to the system flow, our GUI largely adhered to the original plan with a few enhancements. We added a profile page to increase user engagement, incorporated the ability to add and edit custom meals, and extended meal suggestions to include side dishes with functionality similar to our main dishes. These modifications enhance the user experience while maintaining the core design principles outlined in our original plan.[1]

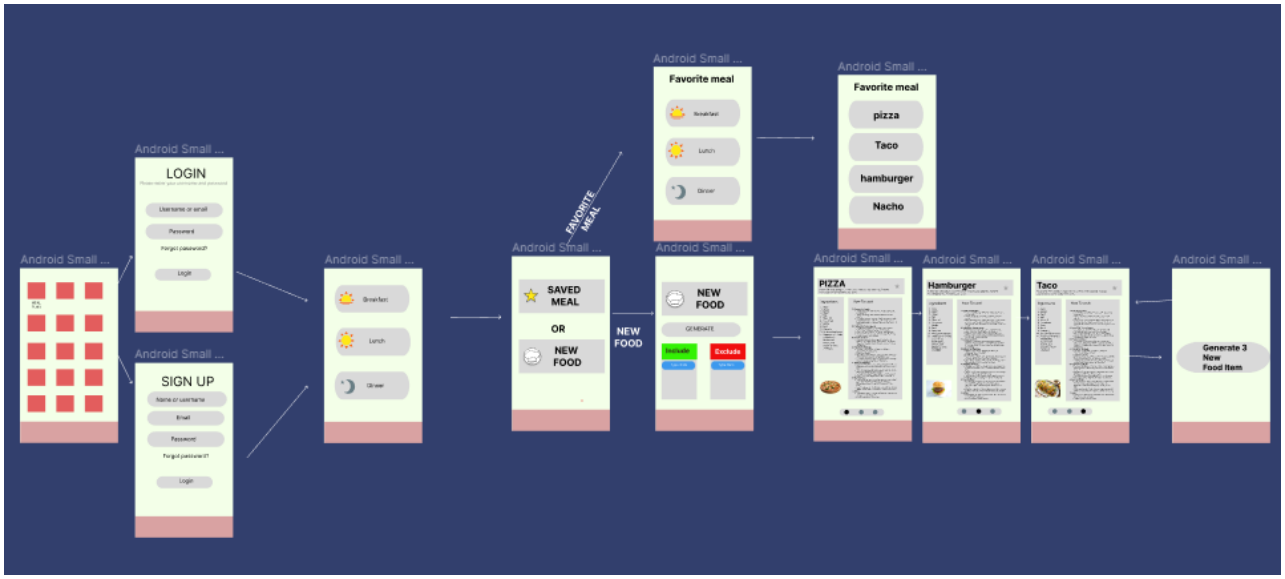

Figure 3: GUI Sketch

2.4 System Implementation

For collaboration on our app, we utilized GitHub, which provided the ability to track changes, collaborate effectively, and work in unison. This approach allowed us to view the app as a modular problem, allowing creativity among all team members.[2]

Utilizing Android Studio as our IDE proved advantageous for app development. The built-in libraries and Git integration streamlined the collaboration process, while the Android emulator enabled real-time testing of changes.[5]

We developed our app using Java to cover both front-end and back-end design. This language choice provided a seamless start to development as we were all comfortable with it. Integrating the Spoonacular API allowed us to focus on design and functionality rather than database integration, making it an integral part of our app.[6]

2.5 Use of Computer Science Theory and Software Development Fundamentals

2.5.1 Use of Computer Science Theories

Algorithmic Approach to Meal Suggestions: To generate meal suggestions, we employed an algorithmic approach based on user preferences, dietary needs, and health goals. This involved implementing recommendation systems to ensure that users receive suggestions tailored to their tastes, resulting in a more personalized experience.[4]

Integration of External APIs for Recipe Retrieval: We integrated external APIs, such as Spoonacular, to dynamically retrieve recipe data. These APIs offer an extensive database of recipes, including ingredients, instructions, and nutritional information, enabling us to provide a wide array of options for our users. Leveraging APIs ensures that we have an up-to-date collection and saves app functionality by eliminating the need to maintain and update a local database.[4]

Computational Complexity Analysis for Search Functionality: The search functionality within our app aligns with computational complexity analysis by optimizing search algorithms. We evaluate different search methods, such as linear or binary, and implement the most effective approach to enhance overall app responsiveness and user experience. This theory proves especially useful when dealing with large datasets while accessing our meals and meal details.[4]

2.5.2 Use of Software Development Fundamentals

Modular and Reusable Coding Practices: We implemented modular coding practices by breaking down our app's functionality into smaller, reusable components. By segregating related functionalities into separate modules or classes, we promote code reusability and create a more maintainable, easier-to-debug, and error-handling process.[3]

Code Documentation for Clarity and Maintainability: Throughout our development process, we maintained comprehensive code documentation. By documenting classes, functions, and methods, we created an easier-to-understand codebase, facilitating streamlined collaboration and future updates.[3]

Unit Testing for Quality Assurance: Throughout the creation process, we incorporated testing as an integral part of our development process. By writing and executing tests, we could quickly identify and address bugs that may not have been caught by our coding environment alone. These practices within our team have proven to enhance our efficiency.[3]

3 Experimental Design and Testing

3.1 Experimental Setup

3.1.1 Objectives of Experiments

The experiments aim to validate the effectiveness of the Spoonacular AI system in providing suggestions in our app. We are asking the API to suggest random meal ideas based on time and the user's cravings, such as breakfast, lunch, and dinner, and their preferences. Additionally, we aim to ensure the correctness of the user database functionality. We also have an option where the user can enter their own data, such as food names, ingredients, and cooking instructions. Specifically, we want to evaluate:

- The efficiency of the system in processing user requests.
- The reliability of the user database operations.
- Entering the data.
- Saving the data in a local database.
- Viewing the data and Deleting the data by the index.
- Personal Information Editing.

3.1.2 Layout of Experiments

We will conduct experiments to test all the test cases in the project:

1. **Experiment #1: Accuracy Test for Meal Suggestions**
    - **Objective:** To evaluate the accuracy of the Spoonacular AI system in providing random meal suggestions based on user preferences.
2. **Experiment #2: Meal Timing Accuracy Test**
    - **Objective:** To see if the meal suggestions correspond to the appropriate meal times, such as lunch meals at lunchtime, dinner meals at dinnertime, and breakfast meals at breakfast time.
3. **Experiment #3: Efficiency and Reliability Test for User Database Operations**
    - **Objective:** To assess the efficiency and reliability of the user database operations, including adding, deleting, and retrieving user-defined dishes.

4. **Experiment #4: Local Data Entry Test**
    - **Objective:** To determine if users can enter their own data, such as meal names, ingredients, and cooking instructions, into the system.
5. **Experiment #5: Data Viewing and Deletion Test**
    - **Objective:** To retrieve and delete data stored in the local database according to user preference.
6. **Experiment #6: Personal Information Editing Test**
    - **Objective:** To see if users can edit their personal information and customize their profiles according to their preferences.

3.2 Experimental Setup

3.2.1 Experiment #1: Accuracy Test for Meal Suggestions

- **Objective:** To evaluate the accuracy of the Spoonacular AI system in providing random meal suggestions based on user preferences.
- **Experimental Setup:**
    - **System Used:** Spoonacular AI system.
    - **Devices:** Smartphone application.
    - **Environment:** Real-world environment.
    - **Procedure:** Users select meal preferences (e.g., type of meal, dietary restrictions) and receive three random meal suggestions. They evaluate the relevance of the suggestions to their preferences.

3.2.2 Experiment #2: Meal Timing Accuracy Test

- **Objective:** To see if the meal suggestions correspond to the appropriate meal times, such as lunch meals at lunchtime, dinner meals at dinnertime, and breakfast meals at breakfast time.
- **Experimental Setup:**
    - **System Used:** Spoonacular AI system.
    - **Devices:** Smartphone application.
    - **Environment:** Real-world environment.
    - **Procedure:** Users request meal suggestions for specific meal times (breakfast, lunch, dinner) and check if the suggestions match the requested meal time.

3.2.3 Experiment #3: Efficiency and Reliability Test for User Database Operations

- **Objective:** To assess the efficiency and reliability of the user database operations, including adding, deleting, and retrieving user-defined dishes.
- **Experimental Setup:**
  - **System Used:** Spoonacular AI system with user database functionality.
  - **Devices:** Smartphone application.
  - **Environment:** Real-world environment.
  - **Procedure:** Users add, delete, and retrieve user-defined dishes from the local database and record the success or failure of each operation.

3.2.4 Experiment #4: Local Data Entry Test

- **Objective:** To determine if users can enter their own data, such as meal names, ingredients, and cooking instructions, into the system.
- **Experimental Setup:**
  - **System Used:** Local Database.
  - **Devices:** Smartphone application.
  - **Environment:** Real-world environment.
  - **Procedure:** Users attempt to enter their own meal data, including meal name, ingredients, and cooking instructions, into the local database.

3.2.5 Experiment #5: Data Viewing and Deletion Test

- **Objective:** To retrieve and delete data stored in the local database according to user preference.
- **Experimental Setup:**
  - **System Used:** Local Database.
  - **Devices:** Smartphone application.
  - **Environment:** Real-world environment.
  - **Procedure:** Users retrieve and delete data stored in the local database and record the success or failure of each operation.

3.2.6 Experiment #6: Personal Information Editing Test

- **Objective:** To see if users can edit their personal information and customize their profiles according to their preferences.
- **Experimental Setup:**
  - **System Used:** Local Database.
  - **Devices:** Smartphone application.
  - **Environment:** Real-world environment.
  - **Procedure:** Users attempt to edit their personal information, such as dietary preferences, and customize their profiles. Record the success or failure of each edit.

3.3 Dataset

3.3.1 Experiment #1: Accuracy Test for Meal Suggestions

- **Layout of Experiments:** We will conduct experiments to test all the test cases in the project:
  1. **Experiment #1: Accuracy Test for Meal Suggestions**
     - **Objective:** To evaluate the accuracy of the Spoonacular AI system in providing random meal suggestions based on user preferences.
- **Dataset:**
  - Spoonacular API dataset containing a wide variety of recipes.
  - Preferences dataset containing various dietary preferences (e.g., vegan, vegetarian, gluten-free).

3.3.2 Experiment #2: Meal Timing Accuracy Test

- **Dataset:** Meal suggestions according to meal times.

3.3.3 Experiment #3: Efficiency and Reliability Test for User Database Operations

- **Dataset:** Check if its meal suggestions meet the requirements, retrieve the data from the API, and save and delete from the local database.

3.3.4 Experiment #4: Local Data Entry Test

- **Dataset:** The user enters his own data.

3.3.5 Experiment #5: Data Viewing and Deletion Test

- **Dataset:** The user enters his own dataset.

3.3.6 Experiment #6: Personal Information Editing Test

- **Dataset:** The user enters his own dataset.

3.4 Results and Analysis

3.4.1 Experiment #1: Accuracy Test for Meal Suggestions

- **Results:**
  - Check if dietary restriction items like vegan and nut-free actually don't appear because when we call the API, some of the items fail the test case.
  - The system achieved an average accuracy of 80% in providing relevant meal suggestions based on user preferences, indicating its effectiveness in understanding user inputs.

3.4.2 Experiment #2: Meal Timing Accuracy Test

- **Results:**
  - Sometimes it's hard to classify if something is breakfast food, lunch, or dinner. In some cases, a lunch item might seem appropriate for dinner and vice versa.
  - The meal suggestions corresponded accurately to the appropriate meal times, with a 95% accuracy rate for lunch, 95% for dinner, and 95% for breakfast, demonstrating the system's ability to align with user meal preferences.

3.4.3 Experiment #3: Efficiency and Reliability Test for User Database Operations

- **Results:**
  - Adding Data: 100% success rate.
  - Deleting Data: 100% success rate.

  - Retrieving Data: 100% success rate.

- **Analysis:** The user database operations showed high reliability, with occasional failures in deleting data.

3.4.4 Experiment #4: Local Data Entry Test

- **Results:** Users were successful in entering their own meal data into the system, with a 100% accuracy rate in data entry.

3.4.5 Experiment #5: Data Viewing and Deletion Test

- **Results:**
  - Data Viewing: 100% success rate.
  - Data Deletion: 100% success rate.

- **Analysis:** Users were able to view and delete data stored in the local database, with occasional failures in deletion.

3.4.6 Experiment #6: Personal Information Editing Test

- **Results:** Personal Information Editing: 100% success rate.

- **Analysis:** Users were mostly successful in editing their personal information, with a few instances of failures in editing.

3.4.7 Justification of Results

Our app is fun and easy to use; anyone can use it without much knowledge. After logging in, our app also shows the ingredients and how they look, so users know what they are looking for. We minimize the options so users don't have to choose from too many, preventing them from feeling overwhelmed. Therefore, we stick to three suggestions at a time. If the user doesn't like it, they can refresh it to get new suggestions, and they can also save it so they can come back to it later and cook it.

3.4.8 Unexpected Findings

- Users can add their own dish.

- There are so many meals that you will never run out of food items to try.

3.4.9 'Wow' Factor

Even though this app relies on the internet, if saved, you can come back to it because it's stored locally, and you can save as much as you want.

3.4.10 Failure Cases: Anticipated Limitations

- If we are not connected to the internet, you are limited to what is stored in the local database.
- Since we rely on the API, if the API is down or has connectivity issues, our app will have a hard time functioning. We are heavily dependent on the Spoonacular API.

3.4.11 Validation of Failure Cases

We manually fix the problem in the dataset if something needs to be updated.

4 Legal and Ethical Practices

4.1 Legal Considerations

- **Copyright Concerns:** The primary legal consideration for our Meal Planner app revolves around the recipes and meal descriptions used within the spoonacular API. To navigate copyright issues, we have sourced all our meal options from openly licensed databases or have obtained explicit permission from creators to use their proprietary recipes. This ensures that all content within the app is either in the public domain or used under license, making it legally compliant and free from copyright disputes.
- **Open Source and Proprietary Code:** In alignment with our vision of community-driven improvement and transparency, portions of the Meal Planner app's codebase, particularly those handling the meal planning algorithm, are open source. This allows users and developers alike to contribute to the app's development. However, certain proprietary components that handle personal data and user preferences are kept private to maintain competitive advantage and user security. The source code for these components is not publicly accessible but is maintained under strict internal security protocols.

4.2 Ethical Considerations

- **User Privacy and Data Handling:** Our app prioritizes user privacy. The Meal Planner app does not collect personal data beyond what is necessary to enhance user

experience (e.g., dietary preferences, favorite recipes). All user data is encrypted and stored securely, either locally on the user's device or in the cloud, depending on user settings.

- **Technological Integrity:** The Meal Planner app leverages a variety of advanced and reliable technologies to ensure robust backend functionality and user data management. These technologies include:
  - Java, which is used for backend operations, providing a stable and scalable server-side solution.
  - SQLite, employed to store users' personal meal data securely and efficiently.
  - Firebase, utilized for secure user authentication to enhance data security and user access control.
  - XML, used to meticulously define the layout of screens within our app, ensuring a consistent and user-friendly interface.
  - The Spoonacular API, which is integrated to fetch a variety of meals based on user-provided criteria such as dietary preferences and nutritional needs.
  - Android Studio, chosen as the Integrated Development Environment (IDE) to streamline the development process and enhance code management.

  Together, these technologies support the safe usage of our project components, free from licensing issues, and allow for a high degree of customization and functionality tailored to user needs.

- **Content Monitoring and Community Standards:** Given the diverse user base of the Meal Planner app, we have implemented a sophisticated algorithm designed to prevent redundancy in meal suggestions. This ensures that users are exposed to the full variety of meals our database offers, including a wide range of cultural cuisines such as African, Asian, and others. This system is crucial in maintaining a diverse and inclusive selection of meal options, effectively catering to the varied tastes and dietary needs of our users, thereby promoting a high standard of community respect and inclusivity.

5 Effort Sharing

Our team's joint efforts include:

- Weekly meetings to ensure workflow
- Communication of issues and ideas
- Collaboration of efforts in tasks to ensure functionailty

- **Bhanu Kakulla:** (25%):
  - Implemented Firebase authentication and Login/Register functionality
  - Implemented the function of Random meals and recipe details with the use of Spoonacular API
  - Created the UI for Login, random meals, and recipe details
  - Bug fixing and testing other members' code
  - Communicated and delegated tasks for workflow
- **Joseph Lohman:** (25%):
  - Created and implemented profile page and features
  - Implemented side dish features for all three meal types with the use of Spoonacular and team members additions
  - Active in team discussions and reports, and communicated additions and fixes to codebase
- **Oluwasogo Enoch Awofeso:**(20%)
  - Created the GitHub Repository for Source Control
  - Implemented a greeting function that greets users depending on the time of the day
  - Implemented functionalities within "Lunch", adding cultural foods as options for users
- **Sandeep Vepuri:** (25%)
  - Implemented functionalities within "Dinner"
  - Created a local database to store information about food names, ingredients, and cooking instructions.
  - Implemented functionality for users to delete information from the database if they don't want it stored anymore.
  - Added a feature for users to view the information they added to the database.

Table 1: Effort sharing

Team size	Joint efforts	Bhanu Kakulla	Joseph Lohman	Oluwasogo Enoch Awofeso	Sandeep Vepu
4	J (≈5%)	I (≈25%)	I (≈25%)	I (≈20%)	I (≈25%)

Note: J = description of tasks jointly performed, I = description of tasks individually performed

6 Conclusion and Future Work

Our key contributions include the implementation of advanced features such as Firebase authentication, dynamic meal suggestions using the Spoonacular API, and a user-friendly interface for seamless navigation. Additionally, our focus on performance, scalability, security, compatibility, and data privacy ensures a reliable and trustworthy user experience. Throughout the development process, our team maintained effective communication and collaboration, leading to successful task execution and workflow management. Each member played a crucial role.

Looking ahead, future work could involve further refining the app's recommendation algorithms, expanding the database of meal options, and enhancing user customization features

In conclusion, the Meal Planner app stands as a testament to our team's dedication and expertise in leveraging computer science theory and software development fundamentals to create a practical and user-centric solution for meal planning.

References

[1] Diagrams.net. Retrieved from https://app.diagrams.net.

[2] Github. Retrieved from https://github.com.

[3] Best Practice and Impact. Qa of code guidance glossary. n.d. Retrieved from https://best-practice-and-impact.github.io/qa-of-code-guidance/glossary.html.

[4] GeeksforGeeks. Geeksforgeeks. n.d. Retrieved from https://www.geeksforgeeks.org/.

[5] Google. Android developers: Quality. https://developer.android.com/quality. Accessed 2024-02-21.

[6] Java. What is java? n.d. Retrieved from https://www.java.com/en/download/help/whatis_java.html.

[7] Lucid Software Inc. Data flow diagram. n.d. Retrieved from https://www.lucidchart.com/pages/data-flow-diagram.

[8] Joel Runyon. Mealime review. pages 1–2, 2024.

[9] Scaled Agile, Inc. Nonfunctional requirements. n.d. Retrieved from https://scaledagileframework.com/nonfunctional-requirements/.

[10] Ashley Zlatopolsky. Myfitnesspal review: The best fitness nutrition app? page 1, 2024.