NELSON MANDELA UNIVERSITY

School of Information Technology

IT PROJECT 3 (ITPV302)

Bachelor of Information Technology (BIT)

27 September 2024

System Requirements, Specifications and Technical Design

Compiled by

Lecturer	Student Name	Student Number	Date Submitted
Prof Bertram Haskins	Nathan Rogers Zanele Mndaweni Max Naidoo	221412581 225097524 225227053	27 September 2024

Contents

Table of Figures

2. Introduction

This document aims to outline the system requirements, specifications, and technical design for the Thyme to cook recipe app. This document will provide a comprehensive overview of the proposed software solution and explain how it will be developed to address the objectives outlined in the business case, including the features, functionality, and constraints of the system.

This document contains 9 key sections. The plan to address objectives section which explains how the system will address the real-world issues identified in the business case. The project scope section which provides a high-level overview of the system's information, functional and communication requirements. The business requirements section which lays out the expectations and needs of the specific end-users that the system will provide. The hardware and software requirements section which covers the software and hardware tools that will be used during development of the app as well the system's expected hardware and software requirements for end users. The design constraints section where limitations related to security, interface design, data storage and system performance are discussed. The high-level use case diagram section which provides a visual representation showing how different user types will interact with the system. The UML class and relational database section will visually detail the core classes and tables that are involved in the system's architecture ultimately outlining how data is stored and managed in the application. The user interface design section presents possible designs of various key screens accompanied by a description of its functionality.

This document serves as blueprint to help plan and consider important aspects when developing the app to ensure the final product adheres to technical and design constraints.

2.1 Plan to address objectives

Plans to address the objectives stated in the business case document are discussed below:

- Offer personalized meal plan that cater to various dietary preferences

  To cater to users with varying dietary requirements (paleo, vegan, etc.), the app will implement a comprehensive user profile system. During the registration process, users will be prompted to specify their dietary preferences and any restrictions they may have. Based on these selections, the app will provide recipe recommendations using a simple filtering algorithm. Initially, the recommendations will focus on matching users with recipes that fit their chosen preferences, without complex dynamic updates.

- Provide a means to filter and search for recipes based on available ingredients

  Users often find it difficult to match their available ingredients with suitable recipes. The app will allow users to input their available ingredients via a simple input interface. The backend will utilize a search algorithm that compares the provided ingredients against the database of recipes, filtering results accordingly. This feature will ensure users can easily make use of what they already have, minimizing waste.

- Offer step-by-step instructions with integrated timers for efficient cooking

  To help users better manage their time in the kitchen, each recipe will be broken down into clearly defined steps. The app will offer integrated timers for any step that requires precise timing, such as baking or simmering. The Flutter framework will be used to design this real-time feature, allowing the app to handle multiple timers concurrently, enhancing the user experience.

- Enable offline saving of recipes and grocery list

  Many users may find themselves without internet access whether that’s while grocery shopping or cooking, the app will be designed to work offline. Recipes and grocery lists will be saved locally on the user’s device and will sync with the cloud whenever internet connectivity is restored. This ensures users can access important data in any situation while also keeping all devices up to date once they reconnect.

- Allow users to adjust recipes (such as changing measurement systems and serving sizes)

  To address the diverse cooking habits of users from different regions, the app will allow for real-time recipe adjustments. Users will be able to change serving sizes, and the ingredient quantities will update automatically to suit these changes. The app will also support conversions between metric and imperial units, giving users flexibility based on their preference or geographic location.

- Make planning meals easier by incorporating a meal planner

  Meal planning can be a daunting and time-consuming task. To simplify this process, the app will include a meal planner that allows users to schedule their meals for the week. Users will then be able to select recipes for specific days of the week. This feature will also integrate nicely with the grocery list generation tool to ensure users have everything they need for the week's meals.

- Generate grocery lists based on selected recipes

  To eliminate the hassle of manually creating grocery lists, the app will automatically generate a list of ingredients based on the user's selected recipe. This list will be customizable, allowing users to remove ingredients they already have and add any additional items they may need. The app will also enable users to sync their grocery lists across devices for seamless access while shopping.

- Allow users to add and share their own recipes

  Many home cooks have personal recipes that they may want to share or store digitally. The app will include a feature that allows users to share their own recipes with others. Additionally, users will be able to categorize their recipes with custom tags, making it easier for others to discover them through search filters.

- Support multi-platform access so the app is accessible across various devices

  To provide a seamless experience across different devices, the app will be developed using Flutter, a cross-platform development tool. This ensures that the app will be accessible on Android, iOS, and the web, allowing users to switch between devices without losing progress or saved data. Cloud synchronization will ensure that user data (grocery lists, meal plans, etc.) is consistently updated across all platforms.

2.2 Project Scope

2.1 Information Scope

- User information

  Basic profile information such as usernames, email addresses, users' dietary preferences, any past meal plan data. Also, user-generated data such as recipes saved, grocery lists generated/made and personalized meal plans will also be stored. Grocery lists and meal plans will be stored locally on the user's device as well as on the cloud so that data can be synced across all their devices. User login credentials will be encrypted both in transit (HTTPS) and at rest since they will be stored on the Firebase database. Personal information such as the user's dietary preferences (allergies, diet) and email will also be encrypted to ensure data privacy and protection from threats.

- Recipe information

  Recipe details such as the ingredients, step by step instructions, name of the recipe and the associated dietary tags will be stored. Since users will also be able to upload their own recipes. The names, photos and related information will also be stored for each recipe. This information will be stored in a cloud-based database Firestore, recipes that are accessed regularly will be cached locally on the user's device so they can access them offline. Recipes that are added by the user will first go through moderation by moderators before they are uploaded to the cloud globally to all users. Recipes that are public to all users will not be encrypted.

- Grocery List and Meal Planner information

  Grocery lists and meal plans created will be stored so users can reference them when needed. Information such as ingredients, quantities and recipes linked to the grocery list and plans will all be saved. This data will be stored locally on the users' device as well as on the cloud. Allowing users to then access their grocery lists and meal plans from any device and view them offline. Basic encryption will be applied to the data syncs with the cloud since it is not highly sensitive, to ensure the integrity of the data.

- Images

  Images linked to recipes; this includes user-uploaded photos will be stored. These images will mainly be stored on the cloud (Firestore). Recently viewed images will be cached locally so user experience is improved a bit.

- Offline Storage and Syncing

  Saved recipes, grocery lists, and meal plans selected by the user will be stored locally. This information will be stored on the user's device in a secure format. When an internet connection is established, the local data will be synced with the cloud data.

In summary the types of information that will be stored in the app will be user-data, recipe information, grocery list and meal planning information. These will be stored securely in the cloud with the use of Firestore. Recipes can be saved locally on the user's device to allow accessibility offline as well. User credentials will be encrypted and stored securely on the Firebase database. Local data that has been saved offline will be synced with the cloud once the user has gone online. This allows users to edit their meal plans or grocery lists while offline, which can then be synced across different devices. This includes their selected preferences as well as account information to check if they have been registered.

User data to be stored:

- Usernames
- Email address
- Dietary preferences
- Favourites and saved recipes
- User generated content
- Meal plans
- Grocery list items

Recipe data to be stored:

- Titles
- Images
- Ingredients
- Instructions
- Nutrition information
- Tag data

2.2 Functional Scope

- Searching and filtering recipes

  This feature will allow users to input their ingredients, diet, recipe name/or part of it and the system will return a list of recipes that match those parameters.

- Planning meals

  This feature will allow users to create customized meal plans tailored to their dietary restrictions and preferences. The app will provide an interface where users can select recipes and assign them to specific days of the week.

- Customizing recipes to suit preferences

  This feature will allow users to modify serving sizes, the app will automatically update the recipe by adjusting the ingredient quantities. The app will also have the option switch between metric and imperial measurement systems.

- Sharing of recipes

  This feature will allow users to add and share their own recipes, meal plans and grocery lists.

- Generating grocery lists

  This feature will allow users to automatically generate a grocery list containing all necessary ingredients based on a selected recipe. Users will also be able to customize this list by adding or removing items as they choose as well as adjusting the quantity of each ingredient in the list.

- Saving of recipes, grocery lists, and meal plans offline

  This feature allows for recipes, grocery lists, and meal plans to be saved offline. All data will automatically sync with the cloud once a connection is established.

- View recipes with step-by-step instructions and integrated timers

  Most recipes will include step by step instructions to guide users through the cooking process. This feature will offer timers for specific cooking stages, ensuring users can manage their cooking times effectively.

- Generating reports and recipes to moderate for administrators

  This feature will generate a report regarding which recipe was most searched, tried, liked, etc. This feature will send admins recipes to approve.

2.3 Communication Scope

Communication within the app will occur through these channels:

The app will send push notifications to alert users to new recipes or reminders for their meal plans. The app will notify users when their data has been successfully synced after they are back online. The app will function over both mobile data and Wi-fi, allowing users to access features regardless of their internet connection type. The app will also notify admins of any recipes that need to be moderated.

2.3 Business Requirements

For the project we aim to build a recipe app which home-cooks or any beginner will be able to follow. The app will be cross-platform which allows users to share and sync their recipes and preferences whether they make use of a mobile device or access the website from a desktop. The three key user types for this system are registered users, system administrators and unregistered user/guests, each with their own specific needs and requirements.

Registered users

These are the primary users of the app; they will require a user-friendly and dependable interface for meal planning and cooking. Their main requirements are:

- Offline accessibility: Users will be able to access their saved recipes, meal plans and grocery lists offline.
- Personalized recipe suggestions: New recipes will be available to users based on their preferences or diet. These personalized suggestions will be synced across multiple devices for the user.
- Meal plans and grocery lists: Users can plan meals and generate a corresponding grocery list with the necessary ingredients so that they can prepare for future recipes that they may wish to cook.

System administrators

System admins are responsible for moderating and maintaining the app to ensure that it runs smoothly, and everyone adheres to community standards. Their requirements are:

- Monitoring and moderation: Admins will be able to moderate user submitted recipes, ensuring only recipes that meet the standards are posted. The most liked and disliked recipes will be logged for the system admin to view and post.
- Manage user account: Admins will be able to manage user accounts by adjusting access levels or deleting accounts.
- Monitoring performance: Admins will have access to tools that will allow them to monitor the analytics of the database.

Unregistered users/guests

Unregistered users or guests will have minimal amount of access to the app's features. Their requirements are:

- Basic recipe access: They will still be able to view and browse through some recipes.
- No access to advanced features: They will not have access to features such as adding to a grocery list, meal planning or the user preference features.

2.4 Hardware and Software Requirements

This section covers the software and hardware tools that will be used during development of the app and the system's expected hardware and software requirements for end users.

2.4.1 Software Requirements

- The application will be developed using Visual Studio Code as the IDE
- Flutter will be used for the development of the app within VS Code.
- Firebase will be used to handle the login and registration as well as the security covering them.
- Firebase will manage both online storage and offline caching.
- The mobile application requires Android 5 (API level 21) or higher.
- The web app will run on any browser but will be optimized for Microsoft Edge.
- The database will be hosted on the cloud using Firebase Cloud Firestore.

2.4.2 Hardware Requirements

- The app will require a phone or tablet with at least 2GB of RAM and running android 5.0 or higher. Adequate storage is also needed to handle downloaded recipes and cached data.
- The web app can be accessed from any device with a web browser; however, users will need a stable internet connection.
- The database will be hosted on Firebase's cloud infrastructure.

2.5 Design Constraints

This section outlines the key constraints that may impact the app/website development and performance of the system, including security, interface, performance and data storage constraints.

2.5.1 Security Constraints

Firebase will manage user authentication which ensures all login and registration data are securely processed. It automatically handles password hashing and salting using bcrypt behind the scenes. Additional security measures, such as multi-factor authentication will also be implemented to enhance security.

When a user adds a recipe to the app or website all text must be verified as text and not code as this could create security vulnerabilities.

All sensitive data such as user credentials, preferences and recipe details must be encrypted both in transit and at rest. Firebase encrypts all data stored in Firestore at rest with AES-256 encryption and has built in security features during transmission between the client and server.

Having role-based access control will ensure different roles whether it be admin, registered users or guest users have varying levels of access to app features. This restricts users from having access to features they prohibited to access (based on their permissions) which safeguards both the system and user privacy.

2.5.2 Interface Constraints

The interface of the app needs to be easy to use and not feature many complex navigation options that cause confusion. The app must not be overly simplified to the point where features are sacrificed.

Users should be given the opportunity to be able select a diet preference and have a section that recommends recipes that conform to the preferences. Additionally, an area that allows users to explore new and trending recipes will enhance the user experience.

The interface should be designed to be responsive and adapt smoothly to various screen sizes such as those of tablets and smartphones. Ensuring consistency across devices is very important so users have a uniform experience and design even when switching platforms.

2.5.3 Performance Constraints

The performance of the app may differ across platforms. Since the app is cross-platform, it will be harder to maintain consistent performance across multiple devices. The app should be optimized to minimize CPU and memory usage so that it can perform smoothly even on low-end devices.

The responsiveness of the interface and UX design will need to be consistent so that the user’s do not have different experiences in terms of the performance of the system. Regardless of whether the app is accessed on an android or web browser should be irrelevant and the system should provide consistent load times, navigation speeds and responsiveness.

Considering mobile users will be able to access offline features such as saved recipes or meal plans, the app should be optimized to use minimal resources such as battery power and network data.

2.5.4 Data Storage Constraints

Firestore which will be used to store the recipe images on the cloud has a free-tier model which is limited and will be used for this app. The free tier has a limit of 1GB storage, as well as 1GB of outbound data per month. The database will continue to function normally, but additionally charges will be placed when exceeding this limit which could impact the system’s scalability and in the long term its sustainability.

Users that want to upload their own recipes with images will need to be constrained so that they do not take up the amount of cloud storage space. The maximum file sizes for the image that a user wants to upload should also be limited to ensure the system still operates efficiently within its data limits.

As the user saves the recipes too it should be considered if they have enough local storage space to save the recipe offline. The app will need to monitor available storage on the device so that when space limits are reached, the user is prevented from downloading the recipes. Users should also be informed of potential sync delays.

2.6 High-level use case diagram

Figure 2.1 represents the high-level use case diagram of the Thyme To Cook system.

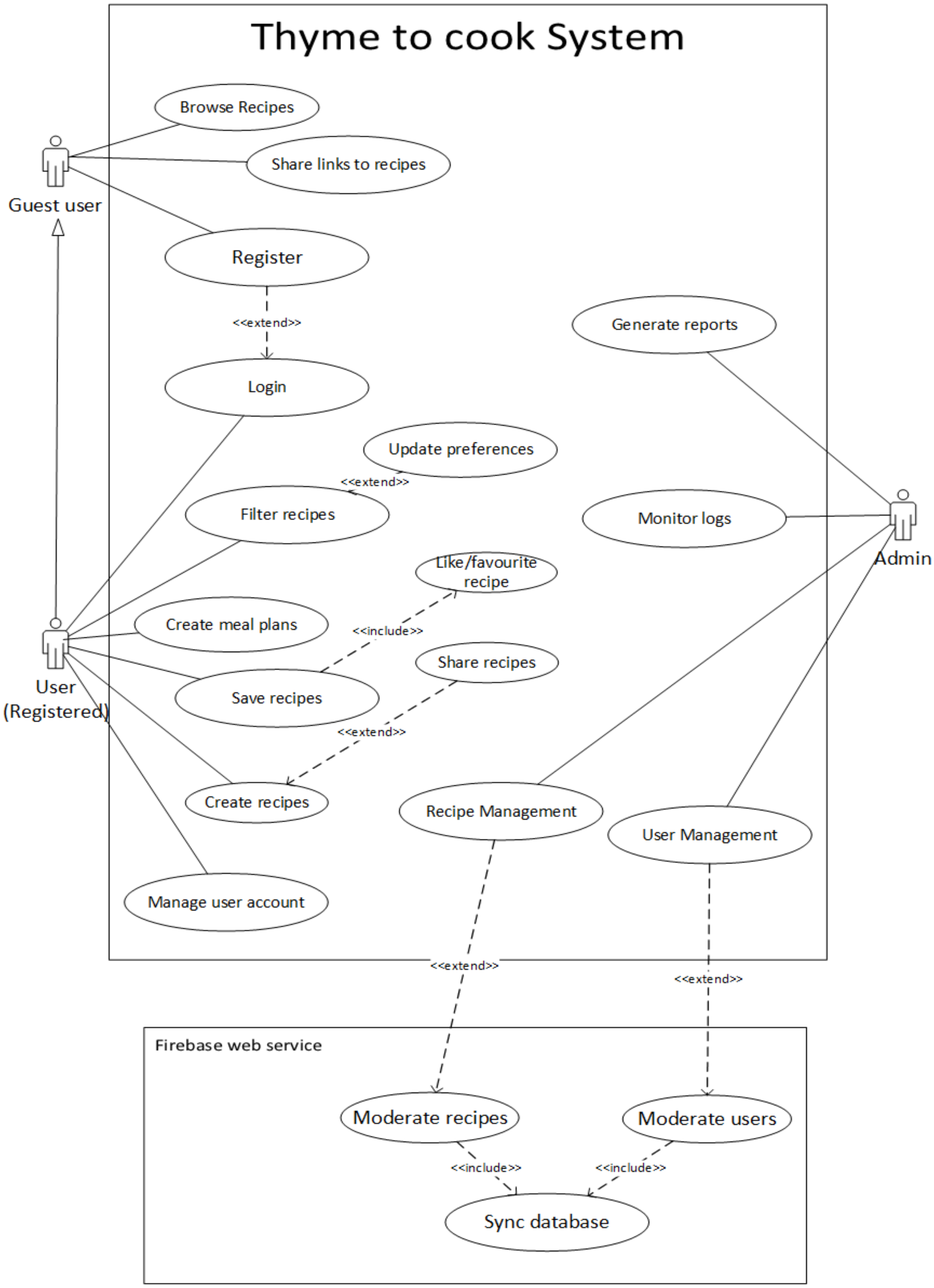


Figure 2. 1: High-Level Use Case Diagram

2.7 UML Class Diagram

Figure 2.2 represents the UML class diagram of the Thyme To Cook system

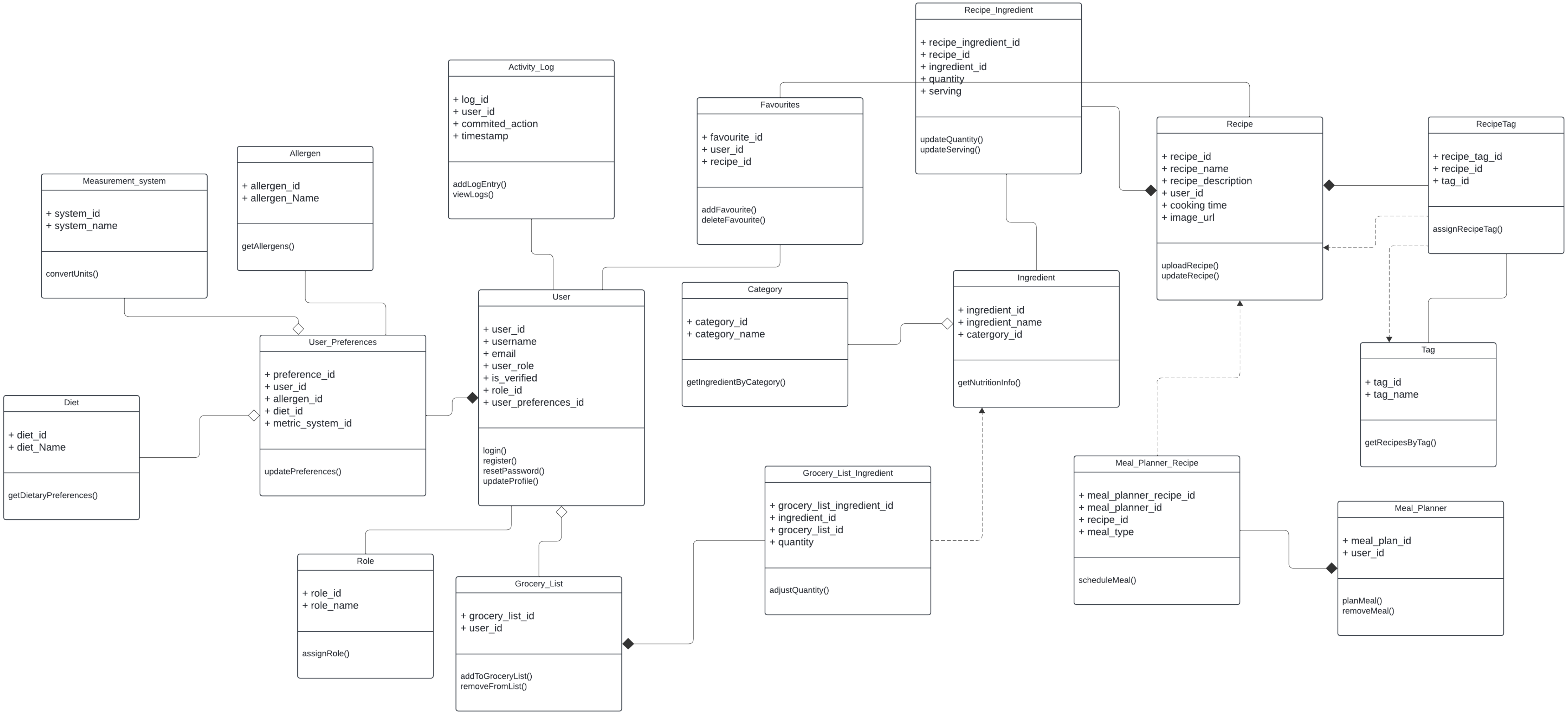
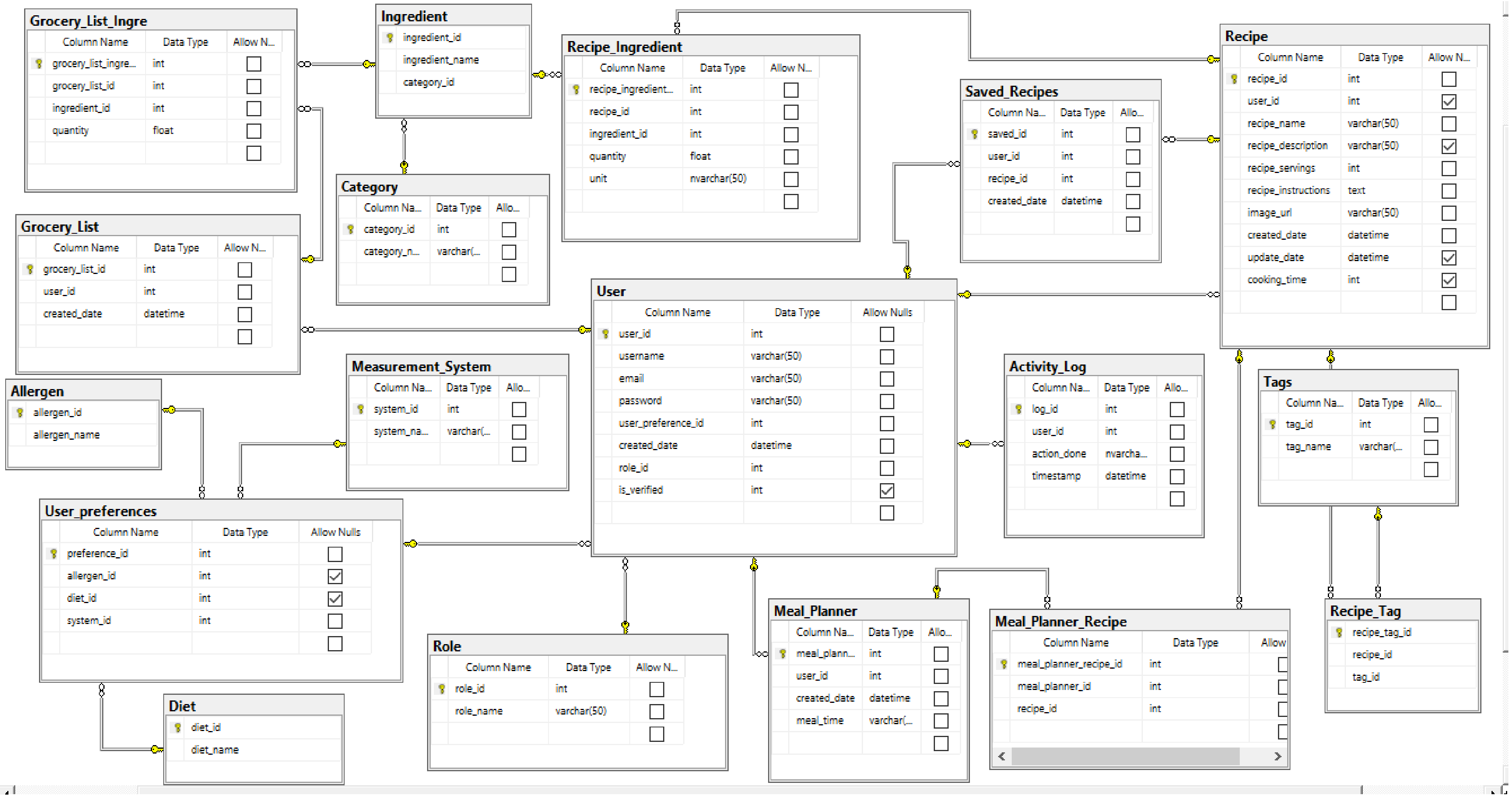


Figure 2. 2: UML Class Diagram

2.8 Relational database diagram

Figure 2.3 represents the relational database diagrams of the Thyme To Cook system



Grocery_List_Ingre

Column Name	Data Type	Allow N...
🔑 grocery_list_ingre...	int	□
grocery_list_id	int	□
ingredient_id	int	□
quantity	float	□
		□

Ingredient

🔑 ingredient_id
ingredient_name
category_id

Recipe_Ingredient

Column Name	Data Type	Allow N...
🔑 recipe_ingredient...	int	□
recipe_id	int	□
ingredient_id	int	□
quantity	float	□
unit	nvarchar(50)	□
		□

Recipe

Column Name	Data Type	Allow N...
🔑 recipe_id	int	□
user_id	int	☑
recipe_name	varchar(50)	□
recipe_description	varchar(50)	☑
recipe_servings	int	□
recipe_instructions	text	□
image_url	varchar(50)	□
created_date	datetime	□
update_date	datetime	☑
cooking_time	int	☑
		□

Saved_Recipes

Column Na...	Data Type	Allo...
🔑 saved_id	int	□
user_id	int	□
recipe_id	int	□
created_date	datetime	□
		□

Category

Column Na...	Data Type	Allo...
🔑 category_id	int	□
category_n...	varchar(...	□
		□

Grocery_List

Column Name	Data Type	Allow N...
🔑 grocery_list_id	int	□
user_id	int	□
created_date	datetime	□
		□

User

Column Name	Data Type	Allow Nulls
🔑 user_id	int	□
username	varchar(50)	□
email	varchar(50)	□
password	varchar(50)	□
user_preference_id	int	□
created_date	datetime	□
role_id	int	□
is_verified	int	☑
		□

Measurement_System

Column Na...	Data Type	Allo...
🔑 system_id	int	□
system_na...	varchar(...	□
		□

Allergen

🔑 allergen_id
allergen_name

Activity_Log

Column Na...	Data Type	Allo...
🔑 log_id	int	□
user_id	int	□
action_done	nvarcha...	□
timestamp	datetime	□
		□

Tags

Column Na...	Data Type	Allo...
🔑 tag_id	int	□
tag_name	varchar(...	□
		□

User_preferences

Column Name	Data Type	Allow Nulls
🔑 preference_id	int	□
allergen_id	int	☑
diet_id	int	☑
system_id	int	□
		□

Role

Column Name	Data Type	Allow N...
🔑 role_id	int	□
role_name	varchar(50)	□
		□

Meal_Planner

Column Na...	Data Type	Allo...
🔑 meal_plann...	int	□
user_id	int	□
created_date	datetime	□
meal_time	varchar(...	□
		□

Meal_Planner_Recipe

Column Name	Data Type	Allow
🔑 meal_planner_recipe_id	int	□
meal_planner_id	int	□
recipe_id	int	□
		□

Recipe_Tag

🔑 recipe_tag_id
recipe_id
tag_id

Diet

🔑 diet_id
diet_name

Figure 2. 3: Relational Database Diagram

2.9 User interface design

Figure 2.4 represents the login screen for the thyme to cook recipe app where users can enter their credentials to access the additional features of the app. Users can log in to the app using an email and password. A “Forgot Password?” option is also available for users who need to reset their password. Once users have successfully logged in, they will be navigated to their personalized home page.

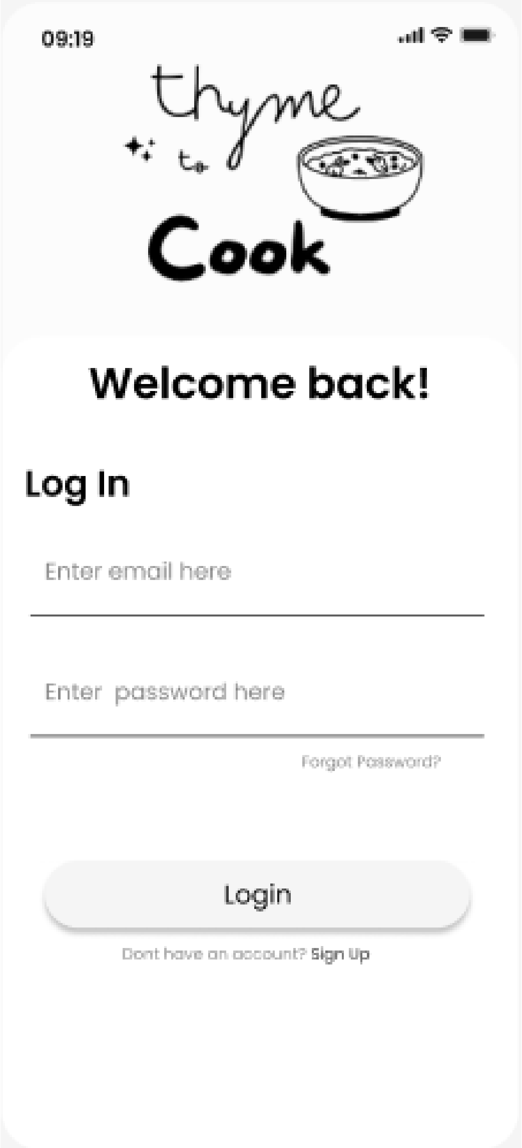


Figure 2. 4: Login screen

Figure 2.5 represents the sign-up screen for the Thyme to cook app where new users can create an account by providing a username, email address and password. This will also be the starting point to personalize their experience in the app.

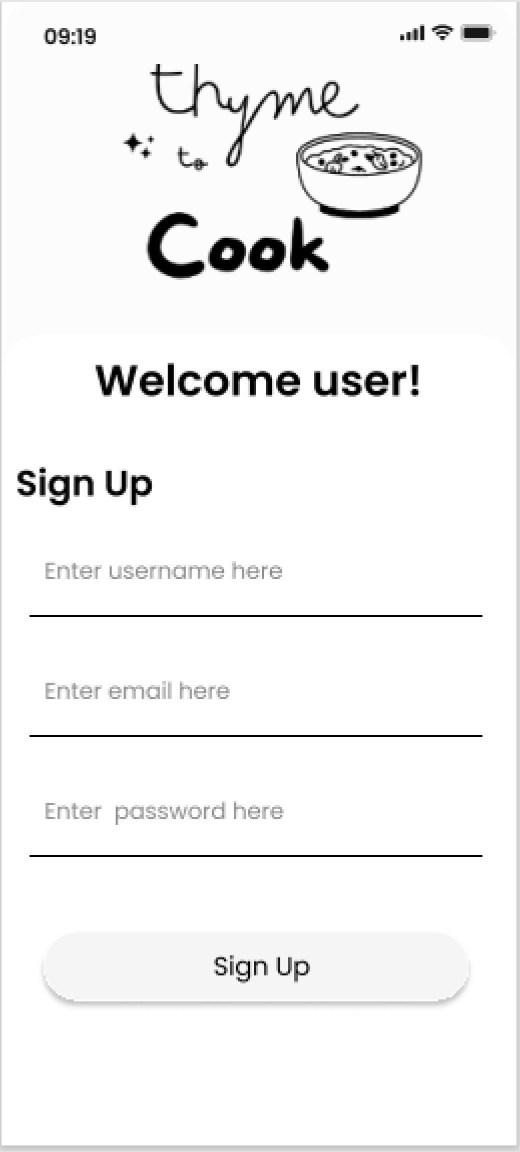


Figure 2. 5: Signup screen

Figure 2.6, Figure 2.7 and Figure 2.8 represent preference screens where users can choose their preferences and personalize their cooking experience. Figure 2.6 shows the ingredients to avoid screen where users can select ingredients they wish to avoid in recipes. Figure 2.7 shows the special diet selection screen where users can choose the specific diets they follow. Figure 2.8 shows the measurement system preference screen where users can choose their preferred measurement system (metric or imperial).

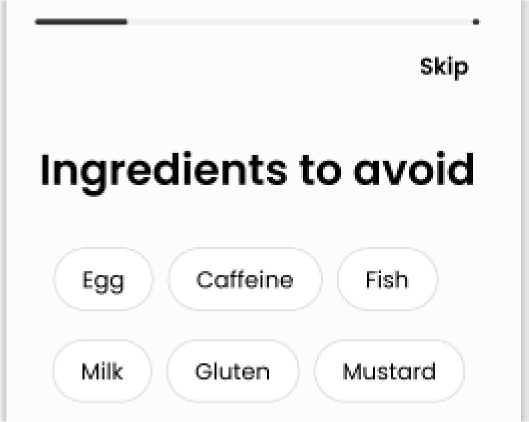


Figure 2. 6: Ingredients to ignore screen

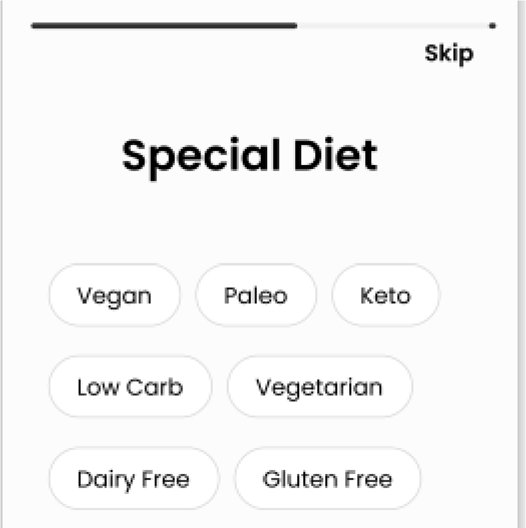
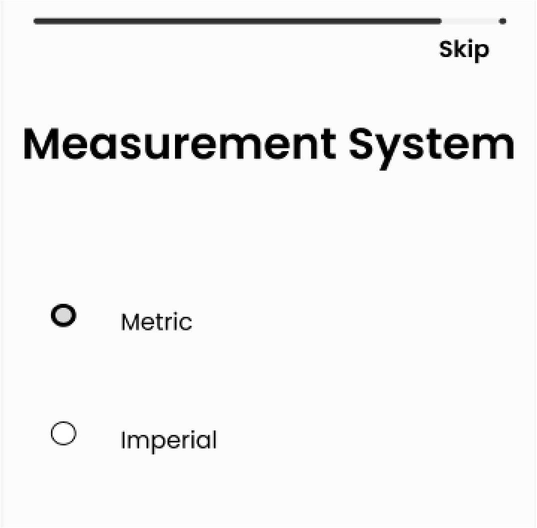


Figure 2. 7: Special diet screen



Measurement System

● Metric

○ Imperial

Figure 2. 8: Measurement system preference screen

Figure 2.9 represents the home screen which serves as the main dashboard where users can explore recipe recommendations based on their preferences, see recently viewed recipes and discover popular dishes. From here users can navigate to their profile as well.

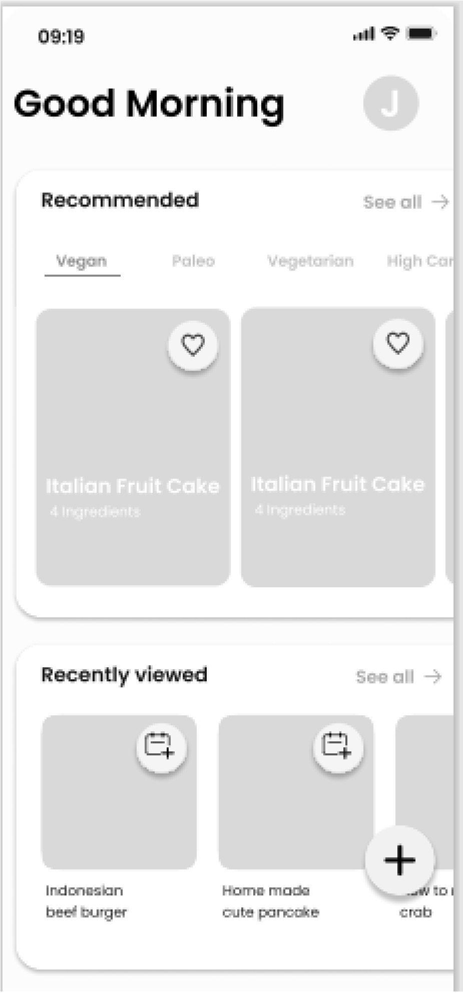


Figure 2. 9: Home screen

Figure 2.10 represents the profile screen where users can view and edit their account details, see their kitchen activity (what they've been cooking), view created recipes and manage their settings.

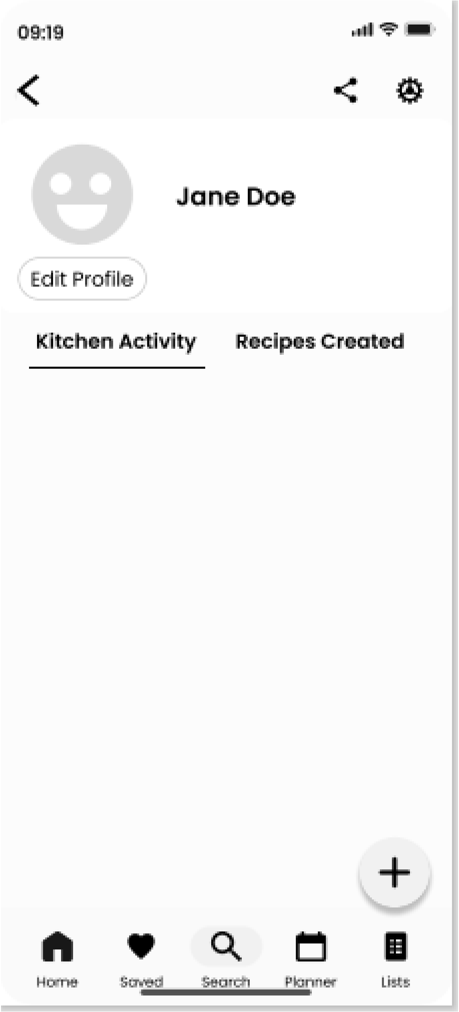


Figure 2. 10: Profile screen

Figure 2.11 represents the search page where users can search for recipes by keyword or filter recipes based on ingredients, diets or nutrition. This screen allows users easy access to a wide range of recipes tailored to their specific needs.

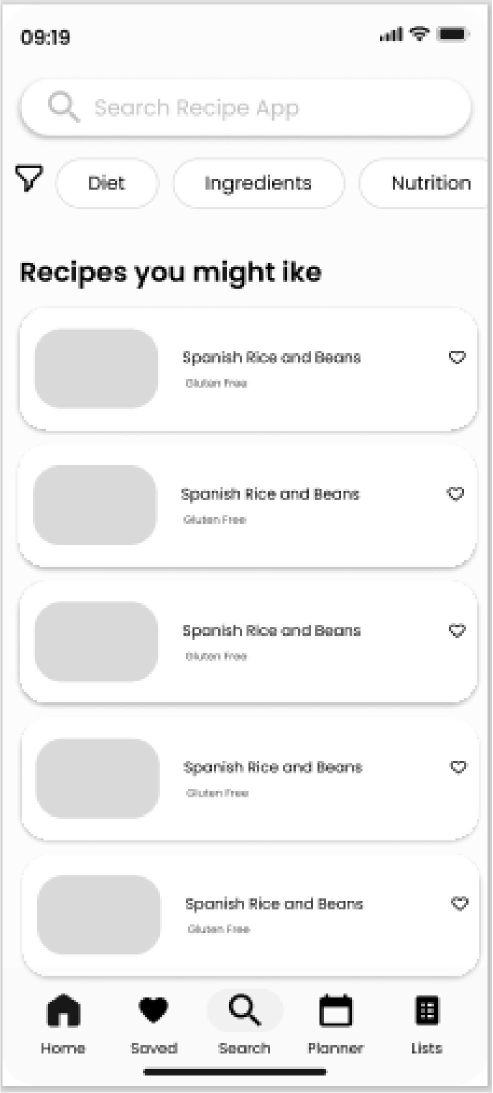


Figure 2. 11: Search screen

Figure 2.12 represents the saved recipes page where a user's recipe collections are kept. Users can then filter these recipes based on recently viewed or browse all saved recipes.

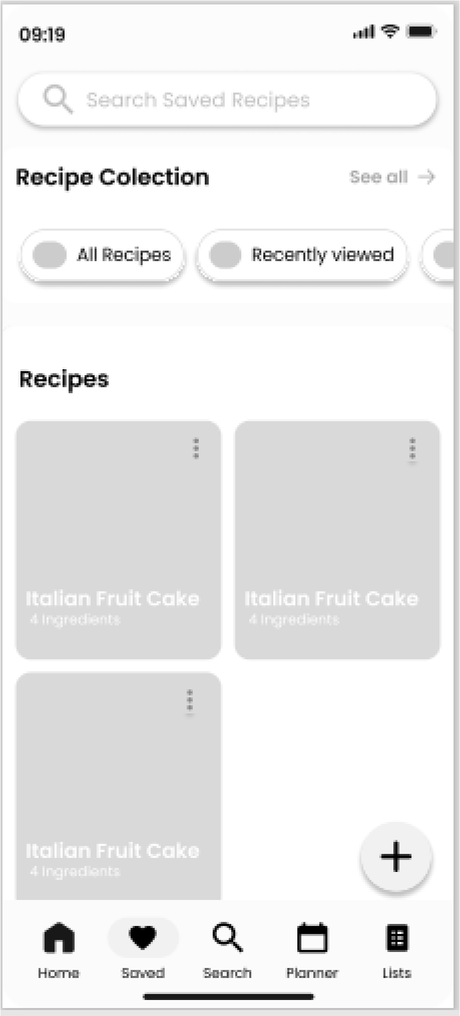


Figure 2. 12: Saved recipes screen

Figure 2.13 represents the meal planner screen where the user can plan meals for each day of the week, selecting recipes for breakfast, lunch, dinner and even snack time.

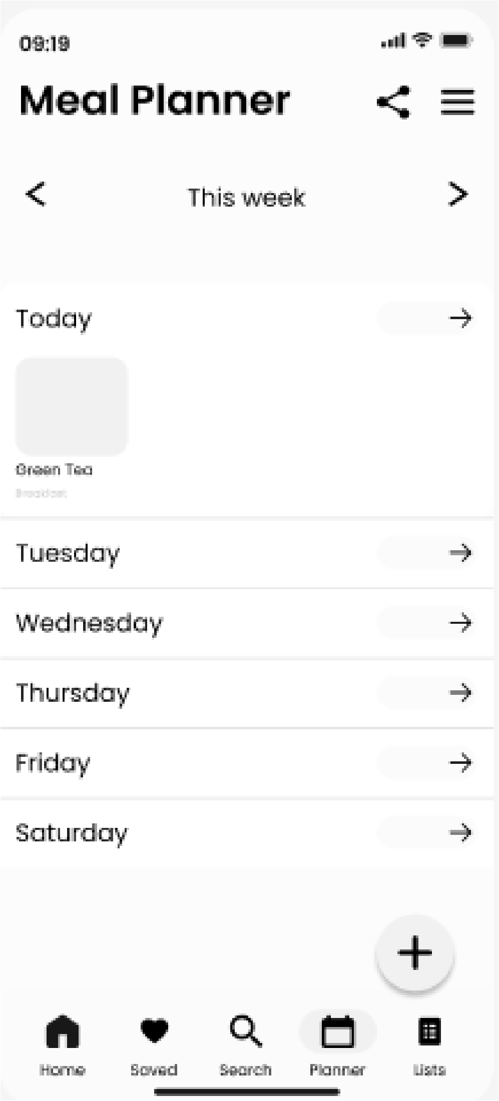


Figure 2. 13: Meal planner screen

Figure 2.14 represents the grocery list screen which helps users manage their grocery list based on the ingredients required for their selected recipes. Users can adjust ingredients and update their list as needed by adding or ticking off various ingredients. Figure 2.15 shows how users can then edit the quantity of the ingredients in their grocery list. These lists can then be shared with friends and family.

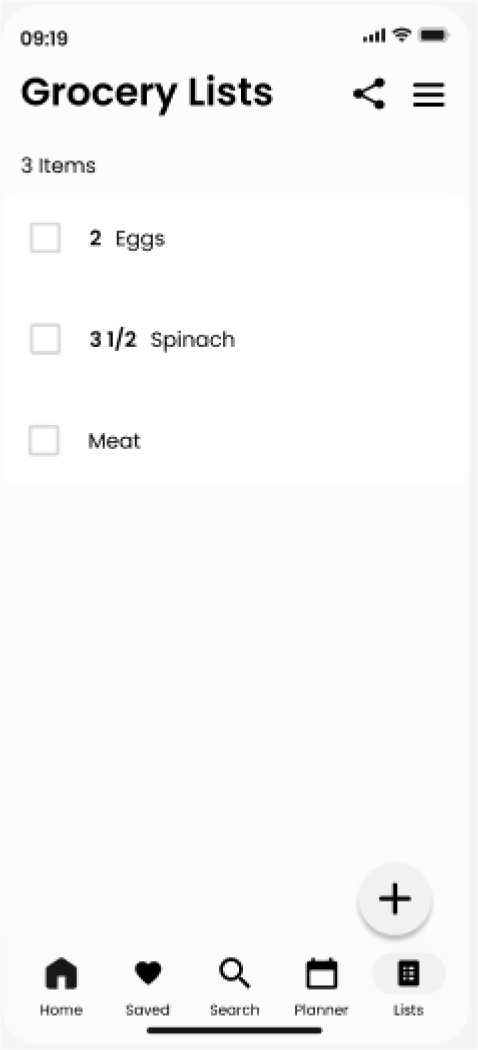
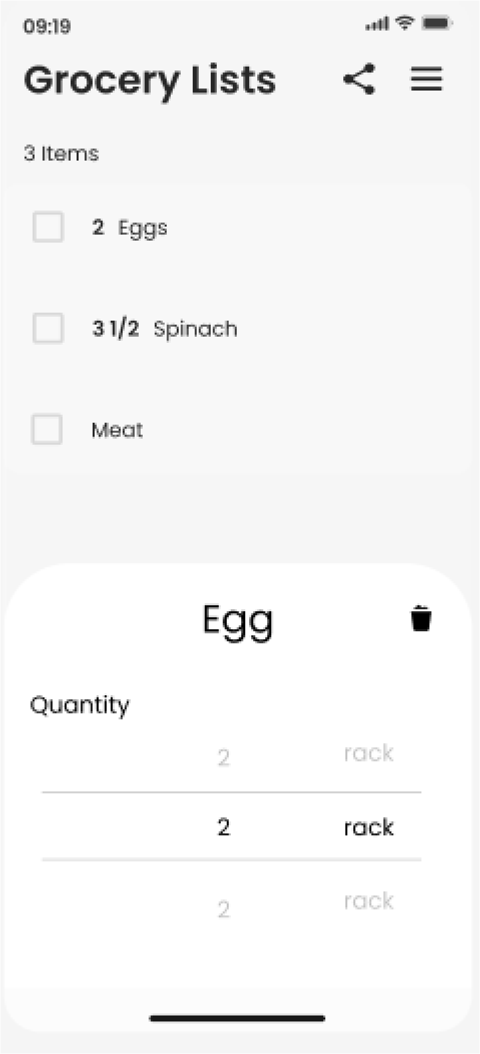


Figure 2. 14: Grocery list screen



☐ 2 Eggs

☐ 3 1/2 Spinach

☐ Meat

Figure 2. 15: Editing ingredient quantity

Figure 2.16 represent the recipe detail screen which is broken up into two tabs, the ingredients tab and the instructions tab. The ingredients tab which shows the recipe ingredients with options to adjust serving sizes and convert between metric and imperial systems. The instructions tab which provides the recipe instructions and navigation to the cook with me feature shown in figure 2.17 which has integrated timers for easier cooking. Recipes can also be saved and added to their meal planner and grocery list.

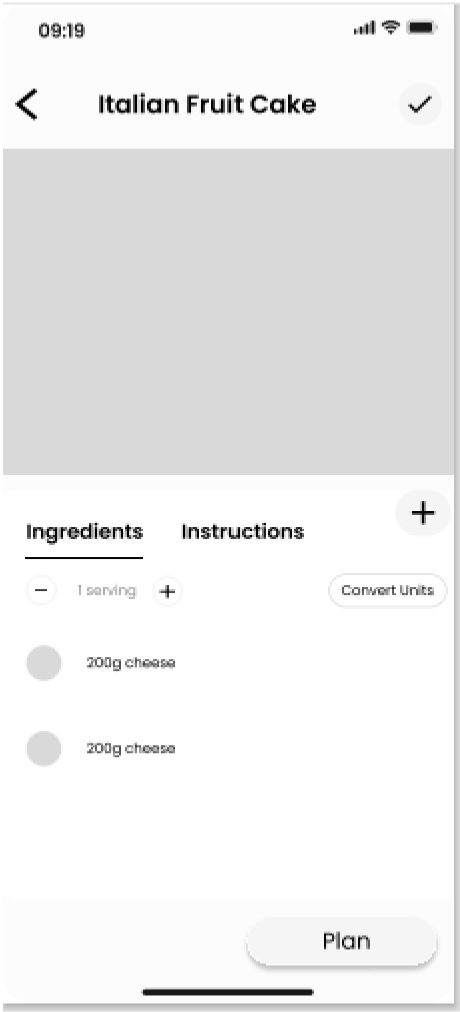
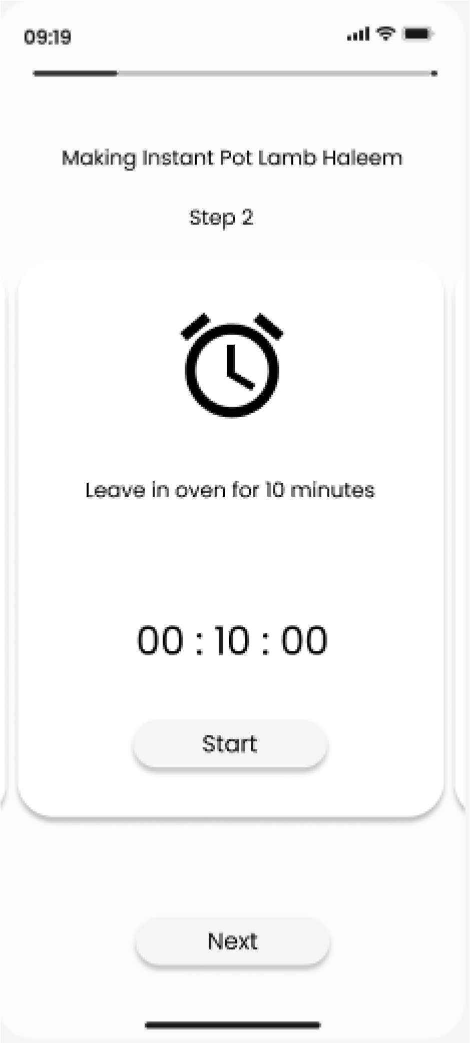


Figure 2. 16:Recipe detail screen



Making Instant Pot Lamb Haleem

Step 2

Leave in oven for 10 minutes

00 : 10 : 00

Figure 2. 17: Cook with me feature