APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 1025-1044
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

The Development of HomeChef: A Recipe Finder Mobile App

Ahmad Syukri Sahartang[1], Ruhaya Ab. Aziz[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: ruhaya@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.054

Article Info

Received: 29 July 2024
Accepted: 21 Oct., 24
Available online: 15 Dec 2024

Keywords

HomeChef, Recipe Finder, Mobile Application, Prototyping Model, Flutter, Cloud Firestore Database

Abstract

Traditional approaches often lead to uninspiring choices, inefficient ingredient use, and time-consuming recipe searches. In today's fast-paced world, these problems can make home cooking feel overwhelming and impractical. Motivated by these issues, HomeChef, a groundbreaking Recipe Finder Mobile Application, addresses the prevalent challenge of meal planning and preparation faced by individuals. The HomeChef app adopts the Prototyping Model to streamline these processes. With modules like ingredient management, recipe discovery, and meal planning, HomeChef optimizes food utilization and simplifies meal planning. Incorporating advanced Cloud Firestore Indexing System enhances recipe import and modification. User acceptance testing confirms its effectiveness, making home cooking accessible and sustainable. Software such as Cloud Firestore Database and Flutter are used to develop the system. This study unveils an innovative solution to culinary challenges, promoting healthier and eco-conscious meal practices.

1. Introduction

Nutrition is a fundamental aspect of human well-being, with recipes serving as the blueprints for creating culinary delights [1]. Recipes, akin to individual signatures, ensure consistency and predictability in food preparation, contributing to efficient resource utilization. In response to common challenges faced in meal planning and preparation, the HomeChef: A Recipe Finder Mobile Application is developed. This application addresses issues such as culinary monotony, inefficient ingredient utilization, and time-consuming meal planning. Aimed at simplifying these processes, HomeChef integrates a cloud firestore indexing system. With modules covering user registration, ingredient management, recipe discovery, meal planning, social interaction, user profile management, and personal recipe management, the app strives to offer a comprehensive solution to these challenges. The subsequent sections delve into the problem statement, objective, scope and report organization of the HomeChef app, highlighting its potential to revolutionize home cooking practices for a diverse user base.

The conventional home cooking process faces challenges such as a lack of culinary creativity, leading to repetitive dishes, food wastage due to ineffective ingredient utilization, and the time-consuming nature of manual recipe searches. This issue not only results in financial losses but also contributes to food waste, which has negative environmental implications, including increased landfill waste and resource depletion. Along the food supply chain, private households represent the largest food-waste faction [2]. Scouring through cookbooks or online sources for suitable recipes based on available ingredients can be tedious, especially for individuals

with busy lifestyles. Time pressure is one of the most critical external cues affecting consumer decisions [3]. Addressing these issues, the "HomeChef" mobile app serves as a modern solution. By leveraging technology, it provides a curated recipe selection based on available ingredients, encouraging culinary variety. The app offers guidance to minimize food wastage and streamlines the recipe search process, saving users time and promoting enjoyable, resourceful meal preparation. There are three main project objectives that need to be achieved, which are:

1. To analyze and design the Recipe Finder Mobile Application by using Object-Oriented Approach.
2. To develop a Recipe Finder Mobile Application.
3. To test the recipe finder mobile Application by using Functional Testing and User Acceptance Testing.

These modules are designed to provide a comprehensive and user-friendly experience within the "HomeChef" mobile application, ensuring users can easily find, save, and prepare delicious meals using the ingredients they have in their kitchen. There are eight modules involved in this application as tabulated in Table 1.

Table 1: The modules involved in the application

Application Module	Description
1. User registration and login	This module facilitates the creation of user accounts by enabling users to register using email and a chosen password. Registered users can subsequently log in with their credentials to access the application's features and functionalities.
2. Ingredient management	Users can input and manage the ingredients based on user's ingredient that are available in the kitchen. This module helps users keep track of the available ingredients.
3. Recipe discovery	This module assists users in discovering recipes based on the ingredients the user has input. It provides a variety of recipes tailored to available ingredients.
4. Meal planning	Users can plan the meals by selecting recipes, scheduling them, and creating a meal calendar. This module simplifies the meal planning process.
5. Social interaction	Users can interact with other HomeChef app users. They can share recipes, rate recipes, and leave comments and feedback.
6. User profile management	This module enables users to update their profile information, including their name, profile picture, and dietary preferences.
7. Personal recipe management	Users can add, edit, or delete their own recipes. This module also provides options for organizing and categorizing recipe collections. Every user has the authority to make modifications to users' recipes.

The "HomeChef: A Recipe Finder Mobile Application," which aims to simplify meal preparation and planning, is the project's final product. This mobile application offers an intuitive user experience with eight key modules covering ingredient and recipe management, interaction with others, user registration, and more. With the ability to create, modify and organize recipes, users may solve issues with time-consuming searches, ingredient efficiency, and culinary creativity. The HomeChef app creates a community where users exchange recipes and experiences, providing a practical and effective solution. Meal planning is revolutionized for a variety of culinary demands thanks to its promotion of less food waste, more culinary innovation, and an all- around pleasurable cooking experience.

2. Related Works

This section will discuss on literature review that had been done for the system and the existing system to manage recipes.

2.1 Recipe Finder Mobile App

The presence of food related content on the web has become prominent in recent years. Along with the overwhelming volume of images on social media, the use of online sites as a source for recipes and culinary ideas is expanding [4]. In essence, a recipe finder mobile app serves as a dynamic culinary companion, leveraging digital technology to empower users in their cooking endeavours. By allowing users to input ingredients and dietary preferences, the tool goes beyond conventional recipe repositories, offering a personalized and tailored cooking experience. This proposed system ensuring that recommendations align with users' specific dietary considerations. The comprehensive features, including the ability to save favourites and explore new cooking ideas, transform the recipe finder into a versatile and indispensable tool for home cooks. It encapsulates the essence of culinary exploration, making it accessible, enjoyable, and efficient for users to navigate the realm of cooking possibilities.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1025-1044

2.2 Cloud Firestore Indexing System

Cloud Firestore's Indexing System is designed to enhance query performance and scalability by automatically creating single-field indexes for every field in a document. This allows Firestore to efficiently filter and sort data. For more complex queries, such as those involving multiple fields or array contents, Firestore employs composite indexes. This enables rapid retrieval of documents based on array content, as seen in queries using the whereArrayContainsAny method. By leveraging this feature, developers can efficiently retrieve documents that contain specific values within their array fields, enhancing the flexibility and performance of database operations [5]. Overall, Firestore's indexing system combines automated and user-defined indexing with distributed query processing to provide fast, scalable, and efficient data access across large datasets.

2.3 Comparison with the Existing Systems

In comparing the features of existing recipe finder systems which are Supercook, Mypantry, and My Fridge Food with the proposed HomeChef system, a comprehensive analysis was conducted to highlight the strengths and unique aspects of each. The summarized and comparison features between the similar existing applications and the proposed system are presented in Table 2.

Table 2: Comparison of Features in Existing and Proposed Systems

Features/System	Supercook	MyPantry	My Fridge Food	Proposed System
User Registration and Login	Yes	Yes	Yes	Yes
Ingredient Management	Yes	Yes	Yes	Yes
Recipe Discovery	Yes	Yes	Yes	Yes
Meal Planning	No	No	No	Yes
Social Interaction	No	No	No	Yes
User Profile Management	Yes	Yes	No	Yes
Personal Recipe Management	No	No	No	Yes

Based on Table 2, the existing systems, such as Supercook, MyPantry, and My Fridge Food, offer basic functionalities like user registration, ingredient management, and recipe discovery. However, they lack robust meal planning and social interaction features. The proposed system addresses these gaps by adding comprehensive modules for meal planning and social interaction, enhancing the overall user experience. The table shows how the proposed system not only includes existing functionalities but also extends capabilities, providing a more holistic and user-friendly recipe finder mobile application.

3. Methodology

This section explains the use of prototyping model in this project and the activities that had been carried out in each phase. The Prototyping Model is a software development approach focused on constructing, testing, and refining prototypes until an acceptable version is achieved [6]. This model serves as scaled-down representations of the final software or product. Unlike traditional waterfall models, the Prototyping Model prioritizes flexibility and adaptability, emphasizing user feedback for continuous refinement throughout the development life cycle. The model is chosen for its suitability in projects benefiting from active user involvementand feedback, such as the Recipe Finder Mobile Application. By involving users early with tangible prototypes, the model ensures their preferences are considered and integrated, contributing to the development of a more user-friendly and effective application. The Prototyping Model's iterative nature facilitates early identification and correction of design flaws, aligning with the dynamic and user-centric nature of the project. The model encompasses key phases, starting with Planning and concluding with the final System phase, incorporating user input for the development of the complete system. This Prototyping Model is illustrated on Gantt Chart.

There are total of six phases from the prototyping model. As shown in Table 3, each phase has its own assignment and output that need to produce during the entire project development. Besides that, the output had been completed within the specific days that have been given.

Table 3: Software development activities and their task

Phase	Task	Output
Planning	Proposed the project. Determine the project schedule, activities and output.	Project Proposal. Develop Gantt Chart.
Analysis	Study of similar systems. Analyse and identify the features of proposed application based on the comparison analysis. Figure out requirements for the system. Interview and survey with culinary expert.	Functional Requirements and non-functional requirements of the system. Use Case Diagram. Class Diagram. Sequence Diagram. **(Appendix A)** Activity Diagram. **(Appendix B)** Literature Review. Google Form survey results.
Design	Design potential user-interface of the proposed system. Design the database for guidance.	Data Dictionary. Wireframe of the system. Database Design. User-interface design
Implementation	Develop the system based on requirement. Perform user acceptance testing. Conduct testing.	Program Code. Test Cases. Testing response through Microsoft Form.
System Prototype	Find bugs and errors in the system. Figure out which part of system is unsatisfactory. If prototype is unsatisfying, return to design phase.	Prototyping Model.
System	Deploy system to the public as a demo. Prep for presentation for the panel.	Complete System. Final Report.

4. Analysis and Design

This section explains the analysis and design that had been done for the system to manage recipes.

4.1 System Requirement Analysis

The primary objective of requirements analysis for the Recipe Finder Mobile Application is to ensure that the system is designed to meet user needs effectively and to understand the user interaction process within the application's defined flow. Additionally, the analysis and design phase for the mobile application includes considerations for database design and user interface design.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1025-1044

4.1.1 Use Case Diagram

Use cases can also be described at various levels of detail. The use cases can be factored and described in terms of other, simpler use cases. A use case is implemented as a collaboration in the interaction view [7]. Figure 1 shows the interaction between the actors and its functionalities for the proposed system "HomeChef: A Recipe Finder Mobile App". There are Sequence Diagram (Appendix A) and Activity Diagram (Appendix B) serve to offer comprehensive insights into the functioning of each module within the system.

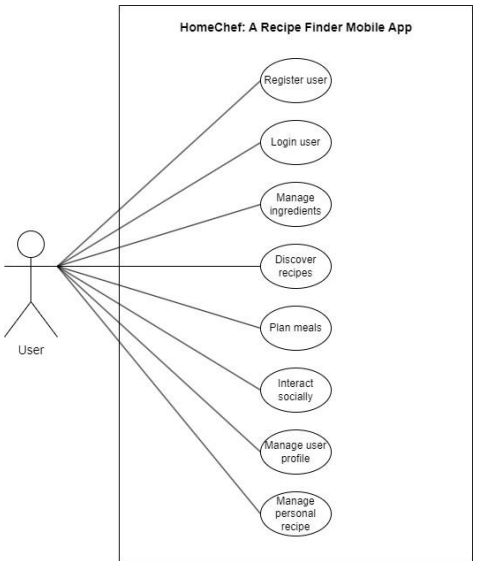


Fig. 1: Use Case Diagram

4.1.2 Functional and Non-Functional Requirements

a. Functional Requirements

The proposed system outlines seven specific functional requirements, as detailed in Table 4, which illustrates the modules and functionalities incorporated into the system.

Table 4: Functional Requirements of the proposed system

No.	Modules	Functionalities
1	User registration and login	• The system shall allow users to create accounts by registering with their email and a chosen password. • Registered users shall have the ability to log in using their credentials. • The system shall include password recovery mechanisms for users who forget their passwords. • Users shall have the option to customize their profiles after registration. • The system shall provide secure authentication to protect user accounts.
2	Ingredient management	• Users shall be able to select and manage a variety of ingredients. • The system shall support the selection of ingredients. • Users shall have the ability to categorize ingredients for better organization.
3	Recipe discovery	• The system shall analyse user-input ingredients to recommend suitable recipes. • Users shall be able to browse a diverse range of recipes based on their available ingredients. • Users shall receive personalized recipe suggestions based on their cooking history.
4	Meal planning	• Users shall have the ability to create and customize meal plans. • The system shall provide a seven-day plan view for users to schedule meals. • Users shall be able to make modifications on which day to cook any recipe saved in the meal plan.
5	Social interaction	• Users shall be able to connect with others in the HomeChef community. • The system shall include features for recipe liking and commenting. • Users shall receive notifications for social interactions, such as comments or likes on their shared recipes.

Table 4: (cont)

6	User profile management	• The system shall allow users to update personal information, including their name, username, number phone and profile picture. • The system shall provide a button to allow users give feedback or improvements through email.
7	Manage personal recipe	• Users shall have the ability to add, edit, or delete their own recipes. • The system shall provide list view to ease the users to view the recipe collections and interactions from other users to the recipe. • Every user shall have the authority to make modifications to users' recipes.

b. Non-Functional Requirements

Table 5 provides a detailed description of the requirements and specifications for the proposed system.

Table 5: Non-Functional Requirements of the proposed system

No.	Requirements	Descriptions
1	Performance	The system shall respond to user interactions (e.g., recipe searches, ingredient input) within 3 seconds to ensure a responsive and efficient user experience.
2	Availability	The application shall have an uptime of at least 99%, ensuring accessibility to users at any time without significant downtime for maintenance or updates.
3	Operational	The system shall be compatible with major mobile platforms (Android) and regularly updated to adapt to new platform versions and address emerging compatibility issues.
4	Usability	The user interface shall be intuitive and user-friendly, promoting easy navigation and minimizing the learning curve for users.
5	Integrity	Data integrity shall be maintained, ensuring that user-inputted information, such as ingredients and recipes, is accurately stored and retrieved without corruption.
6	Maintainability	The system shall be designed with modular and well-documented code, facilitating ease of maintenance, updates, and future enhancements by developers.
7	Security	User data, including personal information and dietary preferences, shall be securely stored and encrypted to prevent unauthorized access and protect user privacy.

4.1.3 User Requirement Analysis

Table 6 outlines the user requirements for the proposed system.

Table 6: User Requirements of the proposed system

No.	User Requirements
1	Users shall be able to register for a new account using their email and password.
2	Users shall have the ability to log in to the application using their registered credentials.
3	Users shall be able to input and manage ingredients based on what is available in their kitchen.
4	Users shall have the capability to discover recipes based on the ingredients they have input.
5	Users shall be able to plan their meals by selecting recipes, scheduling them, and creating a meal calendar.
6	Users shall have the option to interact with other HomeChef app users, including liking on recipes and leaving comments and feedback.
7	Users shall be able to update their profile information, including their name, profile picture, username and number phone.
8	Users shall have access to support resources and the ability to contact the app's support team for assistance, inquiries, or reporting issues.
9	Users shall be able to add, edit, or delete the user own recipes, with options for organizing and categorizing recipe collections.
10	Every user shall have the authority to make modifications to users' recipes.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1025-1044

4.1.4 Domain Class Diagram

A class diagram is a graphic presentation of the static view that shows a collection of declaratives (static) model elements, such as classes, types and their contents and relationships [7]. In Figure 2, 6 classes are identified in the proposed system which are User, Personal_Recipes, Ingredients, Recipes, Meal_Calendar and Interactions. Each class has its own operations and features for this system.

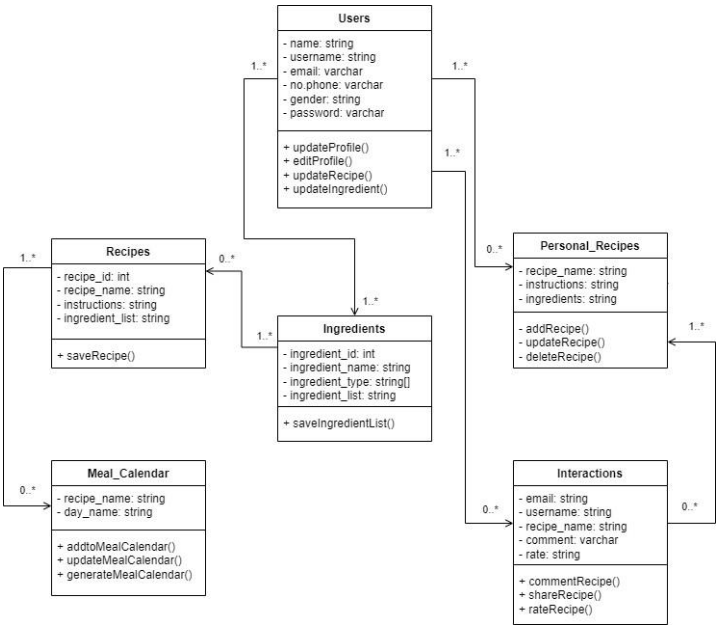


Fig. 2: Class Diagram

4.2 Design

Once all the user requirements have been thoroughly analysed, the project will advance to the design phase. During this stage, both the interface and the database are designed to provide a visual representation of the system before proceeding with the coding process.

4.2.1 System Architecture

The Recipe Finder Mobile Application follows a client-server architecture for streamlined functionality as shown in Figure 3. Users, as clients, interact with a centralized server housing core logic and a relational database. Microservices enable modular development, ensuring independence for functions like user management and recipe discovery.

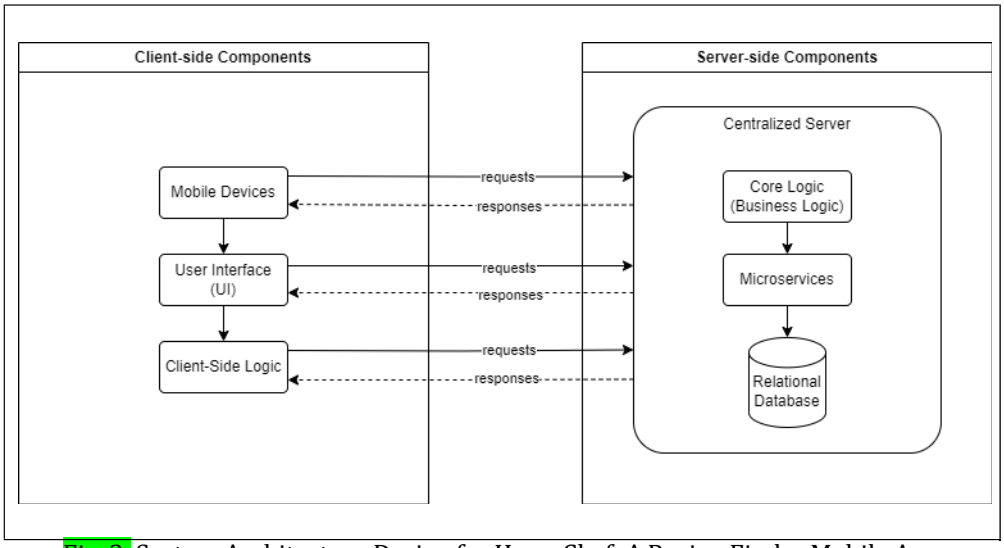


Fig. 3: System Architecture Design for HomeChef: A Recipe Finder Mobile App

4.2.2 Hardware and Software Requirement Analysis

Hardware and software requirements are shown in Tables 7 and 8.

Table 7: Hardware Requirements

Requirement	Specification
Processor	Dual-core processor, 2.0 GHz or higher
RAM	8 GB or higher
Storage	Minimum 256 GB SSD
Display	Minimum 13-inch display with 1920x1080 resolution
Graphics Card	Integrated or dedicated graphics with support for OpenGL
Network	Ethernet and/or Wi-Fi capability

Table 8: Software Requirements

Requirement	Specification
Operating System	Windows 10 or macOS Big Sur
Development Environment	Flutter SDK with Dart
IDE	Android Studio
Database	Cloud Firestore Database
Version Control	Git
Dependency Management	Flutter Package Manager (pub)
Build Tools	Flutter CLI

4.2.3 User Interface Design

Figma software was employed to design the interface of the system. Figure 4 and Figure 5 displays the essential interface design of the proposed system. Other interfaces refer to **Appendix C**.

Fig. 4: Login interface for users

Fig. 5: Social Interaction interface for users

4.2.4 Database

The subsequent list enumerates the database schema derived from the class diagram.

i. users(name, display_name, email, phone_number, gender, password, photo_url, list_of_ingredients)
ii. personalRecipes(pr_ingredient_list, pr_instructions, pr_liked_by, pr_photo_recipe, pr_recipe_name)
iii. mealPlans(mealDay, prPhotoRecipe, prRecipeName, recipeTitle, recipeName)
iv. recipe_list(ingredient_list, instructions, photo_recipe, recipe_name)
v. comments(comment_text, comment_user, created time, post_type)

The database schema is created in advance to identify the necessary elements and the links between them.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1025-1044

5. Result and Discussion

5.1 Implementation

The HomeChef mobile app is developed using Flutter and Firebase. The programming language used is Dart Language both for the front-end and back-end.

To access the application, a user must register an account by providing a name, username, email, phone number, password, and gender. It is important to remember the password and email, as these will be needed for logging in. The User Registration Interface is illustrated in Figure 6, and the User Registration Code Segment is shown in Figure 7.

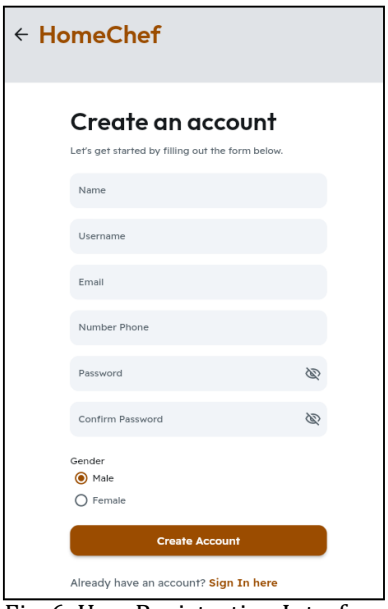


Fig. 6: User Registration Interface

```
final user = await authManager
    .createAccountWithEmail(
  context,
  _model.emailAddressTextController.text,
  _model.passwordTextController.text,
);
if (user == null) {
  return;
}

await UsersRecord.collection
    .doc(user.uid)
    .update(createUsersRecordData(
      email: _model
          .emailAddressTextController
          .text,
      name:
          _model.nameTextController.text,
      gender: _model.genderOptionValue,
      displayName: _model
          .usernameTextController.text,
      phoneNumber: _model
          .numberPhoneTextController.text,
    ));

ScaffoldMessenger.of(context)
    .showSnackBar(
  SnackBar(
    content: Text(
      'Your account has been registered. Welcome aboard!',
      style: GoogleFonts.getFont(
        'Merriweather',
        color:
            FlutterFlowTheme.of(context)
                .primaryText,
        fontWeight: FontWeight.w500,
        fontSize: 15,
      ),
    ),
    duration:
        Duration(milliseconds: 4000),
    backgroundColor:
        FlutterFlowTheme.of(context)
            .secondaryBackground,
  ),
);
```

Fig. 7: User Registration Code Segment

In the User Login module, all users must enter an email and password to access the application. Users can also log in using a Google account by clicking the "Continue with Google" button. This login module serves as the start page of the application. If a user has not yet registered, they can click on the "Sign Up Here" text. The User Login Interface is illustrated in Figure 8, and the User Login Code Segment is shown in Figure 9.

Fig. 8: User Login Interface

```
336                 final user =
337                     await authManager.signInWithEmail(
338                   context,
339                   _model.emailAddressTextController.text,
340                   _model.passwordTextController.text,
341                 );
342                 if (user == null) {
343                   return;
344                 }
345
346                 context.goNamedAuth(
347                     'pantryPage', context.mounted);
348               },
394                 final user = await authManager
395                     .signInWithGoogle(context);
396                 if (user == null) {
397                   return;
398                 }
399
400                 context.goNamedAuth(
401                     'pantryPage', context.mounted);
402               },
```

Fig. 9: User Login Code Segment

After logging into the application, the Ingredient Management Interface will appear. This module allows users to manage and select the available ingredients they currently have. Once the users have finished managing and selecting the ingredients, they must click the "Save Pantry" button to save the changes. The Ingredient Management Interface is illustrated in Figure 10, and the Ingredient Management Code Segment is shown in Figure 11.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1025-1044

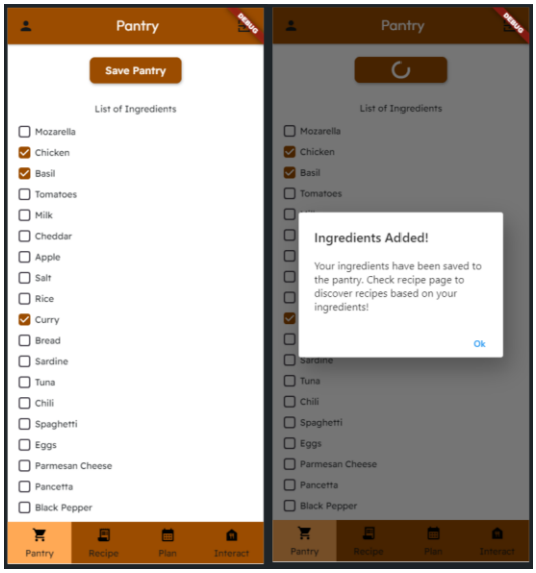


Fig. 10: Ingredient Management Interfaces

```
child: FFButtonWidget(
  onPressed: () async {
    if (_model.listIngredientsCBValues !=
            null &&
        (_model.listIngredientsCBValues)!
            .isNotEmpty) {
      await currentUserReference!.update({
        ...mapToFirestore(
          {
            'list_of_ingredients': _model
                .listIngredientsCBValues,
          },
        ),
      });
      await showDialog(
        context: context,
        builder: (alertDialogContext) {
          return AlertDialog(
            title: Text('Ingredients Added!'),
            content: Text(
                'Your ingredients have been saved to the pantry. Check recipe page to discover recipes based on your ingredients!'),
            actions: [
              TextButton(
                onPressed: () =>
                    Navigator.pop(
                        alertDialogContext),
                child: Text('Ok'),
              ),
            ],
          );
        },
```

Fig. 11: Ingredient Management Code Segment

After managing and selecting available ingredients, the next step is to navigate to the "Recipe" page. The Recipe Discovery module provides results based on the selected ingredients. This module displays all recipes that include any ingredients chosen from the "Pantry" page. Users can search for specific recipe names to save time and check the number of missing ingredients for each recipe. If a user is interested in a particular recipe, it can be saved and added to the meal plan. The Recipe Discovery Interface is illustrated in Figure 12, and the Recipe Discovery Code Segment is shown in Figure 13.

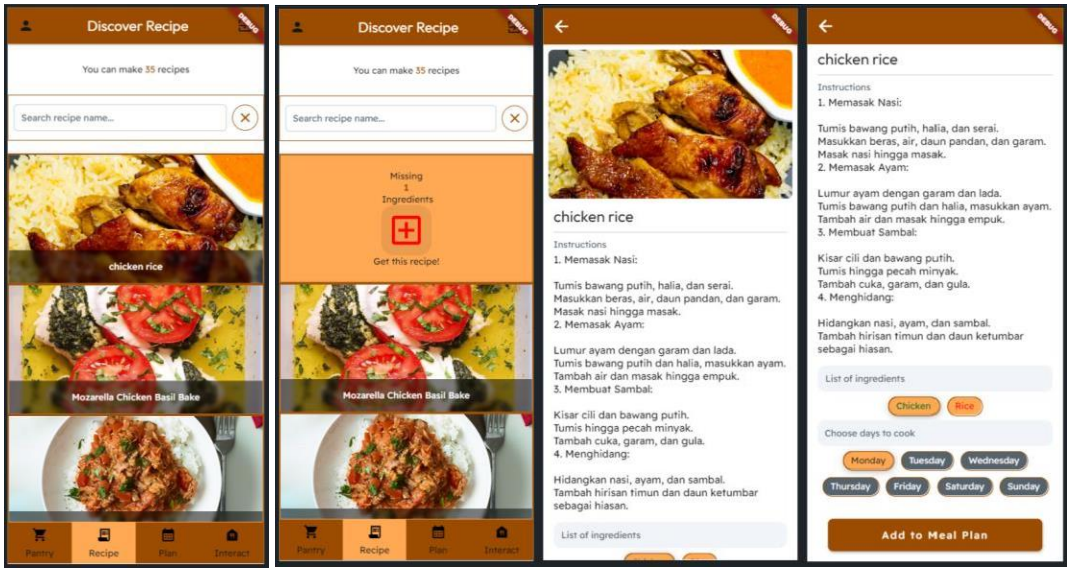


Fig. 12: Recipe Discovery Interfaces

```
children: [
  Expanded(
    child: AuthUserStreamWidget(
      builder: (context) => StreamBuilder<List<RecipeListRecord>>(
        stream: queryRecipeListRecord(
          queryBuilder: (recipeListRecord) =>
              recipeListRecord.whereArrayContainsAny(
                  'ingredient_list',
                  (currentUserDocument?.listOfIngredients
                              ?.toList() ??
                          []) !=
                      ''
                  ? (currentUserDocument?.listOfIngredients
                          ?.toList() ??
                      [])
                  : null),
        ),
```

Fig. 13: Recipe Discovery Code Segment

After saving a recipe from the Recipe Discovery module, a user can check and view the recipe for any selected day. The user can manage the recipe by editing the day or deleting the recipe for that day. The Meal Planning Interface is illustrated in Figure 14, and the Meal Planning Code Segment is shown in Figure 15.

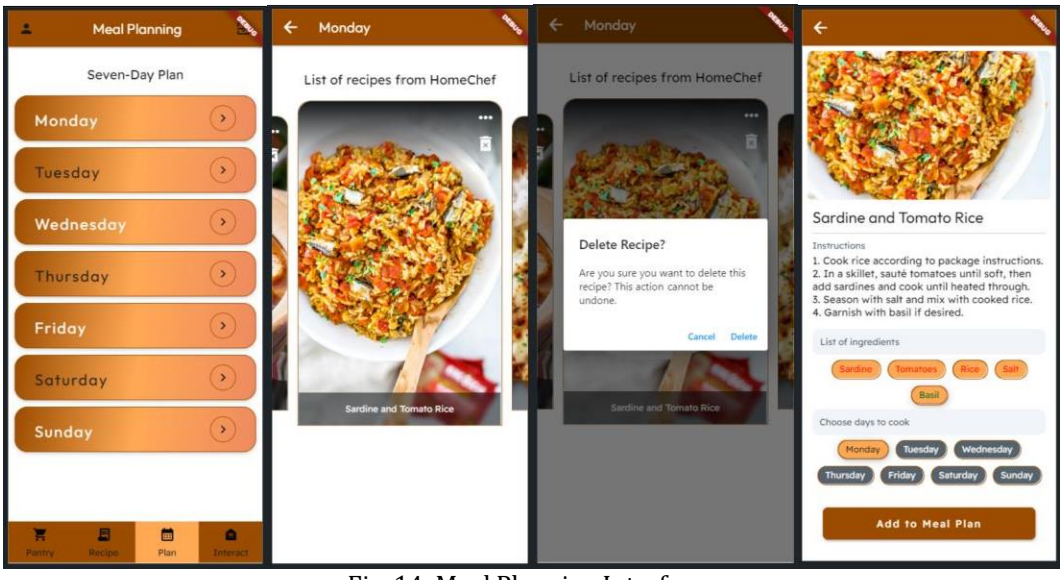


Fig. 14: Meal Planning Interfaces

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1025-1044

```
children: [
  Expanded(
    child: AuthUserStreamWidget(
      builder: (context) => StreamBuilder<List<RecipeListRecord>>(
        stream: queryRecipeListRecord(
          queryBuilder: (recipeListRecord) =>
              recipeListRecord.whereArrayContainsAny(
                  'ingredient_list',
                  (currentUserDocument?.listOfIngredients
                              ?.toList() ??
                          [])
                      .isNotEmpty
                      ? (currentUserDocument?.listOfIngredients
                              ?.toList() ??
                          [])
                      : null),
        ),
```

Fig. 15: Meal Planning Code Segment

The Social Interaction module displays all recipes from users registered to the application, based on the selected ingredients from the "Pantry" page, similar to the "Recipe" page. Each user has the right to create and publish recipes to the application, which can then be viewed by others in the Social Interaction module. Users can interact with recipes from others by liking and commenting on them. Additionally, it is possible to save any recipe from this module to a meal plan. The Social Interaction Interface is illustrated in Figure 16, and the Social Interaction Code Segment is shown in Figure 17.

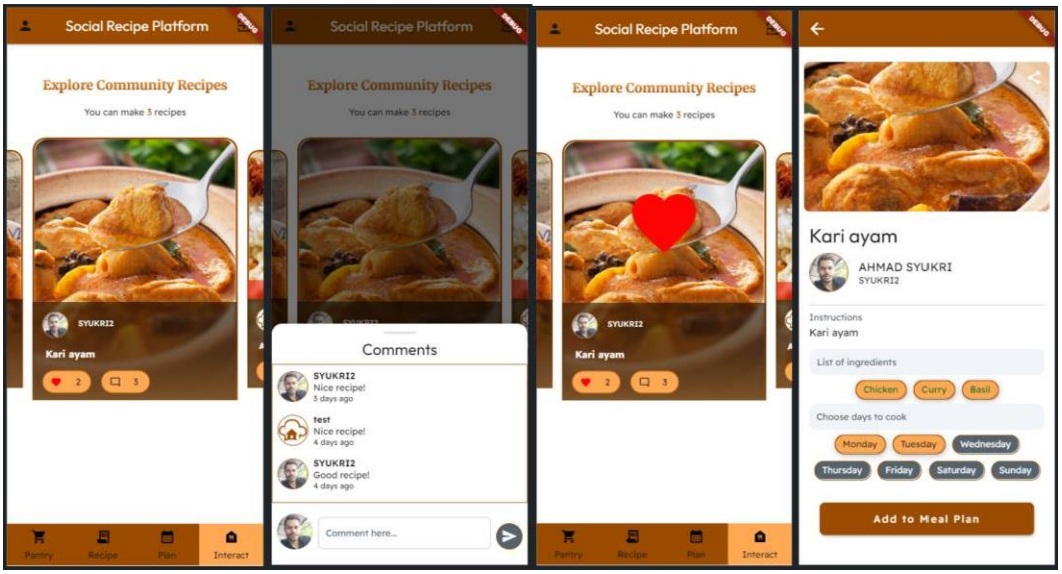


Fig. 16: Social Interaction Interfaces

```
child: AuthUserStreamWidget(
  builder: (context) =>
      StreamBuilder<List<PersonalRecipesRecord>>(
    stream: queryPersonalRecipesRecord(
      queryBuilder: (personalRecipesRecord) =>
          personalRecipesRecord.whereArrayContainsAny(
              'pr_ingredient_list',
              (currentUserDocument?.listOfIngredients
                          ?.toList() ??
                      [])),
    ),
```

Fig. 17: Social Interaction Code Segment

Every user can manage and update information such as profile photos, names, usernames, and phone numbers on this User Profile Management module. If any user has suggestions or improvements for this application, clicking "Improve This App" will send it to the HomeChef email. The User Profile Management Interfaces are illustrated in Figure 18, and the User Profile Management Code Segment is shown in Figure 19.

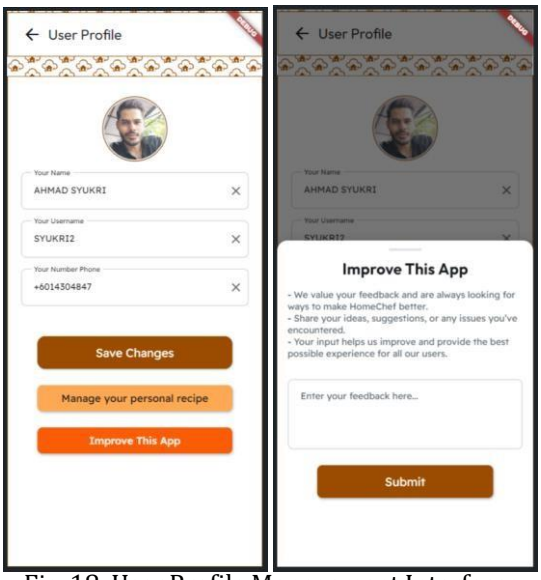


Fig. 18: User Profile Management Interfaces

```
38                if (_model.uploadedFileUrl != null &&
39                    _model.uploadedFileUrl != '') {
40                  await currentUserReference!.update(createUsersRecordData(
41                    name: _model.yourNameTextController.text,
42                    displayName: _model.yourUsernameTextController.text,
43                    phoneNumber: _model.yourNumberPhoneTextController.text,
44                    photoUrl: _model.uploadedFileUrl,
45                  ));
46                  setState(() {});
47                } else {
48                  await currentUserReference!.update(createUsersRecordData(
49                    name: _model.yourNameTextController.text,
50                    displayName: _model.yourUsernameTextController.text,
51                    phoneNumber: _model.yourNumberPhoneTextController.text,
52                  ));
53                }
54                ScaffoldMessenger.of(context).showSnackBar(
55                  SnackBar(
56                    content: Text(
57                      'Your profile information has been updated successfully.',
58                      style: TextStyle(
59                        color: FlutterFlowTheme.of(context).primaryText,
60                      ),
61                    ),
62                    duration: Duration(milliseconds: 2000),
63                    backgroundColor:
64                        FlutterFlowTheme.of(context).secondaryBackground,
65                  ),
66                );
67              }
68            },
69        text: 'Save Changes',
```

Fig. 19: User Profile Management Segment

Every user possesses the right to create and publish the user own recipe to the application, which can be viewed on the 'Interact' page. This Personal Recipe Management module enables users to add, edit, and delete the recipes that the users have created by providing a recipe photo, recipe name, recipe instructions, and recipe ingredients. Users can also view the number of likes and comments for each recipe created here. The Personal Recipe Management Interfaces are illustrated in Figure 20, and the Personal Recipe Management Code Segment is shown in Figure 21.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1025-1044

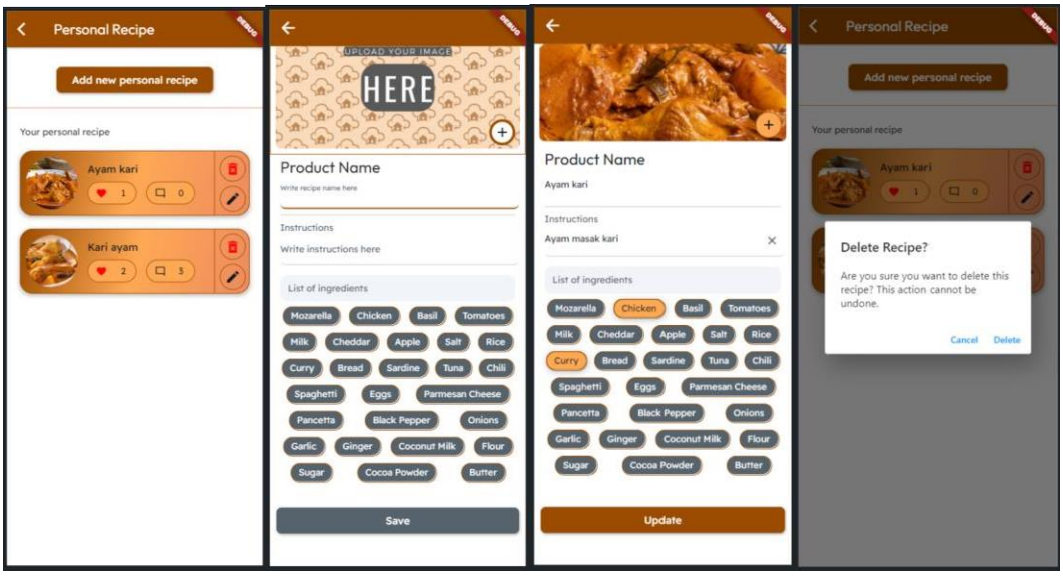


Fig. 20: Personal Recipe Management Interfaces

```
List<PersonalRecipesRecord>
    listViewPersonalRecipesRecordList =
    snapshot.data!;
return ListView.builder(
  padding: EdgeInsets.zero,
  shrinkWrap: true,
  scrollDirection: Axis.vertical,
  itemCount:
      listViewPersonalRecipesRecordList.length,
  itemBuilder: (context, listViewIndex) {
    final listViewPersonalRecipesRecord =
        listViewPersonalRecipesRecordList[
            listViewIndex];
    return Padding(
      padding: EdgeInsetsDirectional.fromSTEB(
          20, 0, 20, 20),
      child: Container(
        width: double.infinity,
        height: 100,
        decoration: BoxDecoration(
          boxShadow: [
            BoxShadow(
              blurRadius: 4,
              color: Color(0x33000000),
              offset: Offset(
                0,
                2,
              ),
              spreadRadius: 2,
            )
          ],
```

Fig. 21: Personal Recipe Management Segment

5.2 Testing

The mobile app undergoes testing to verify whether it fulfills the requirements. Two types of testing are conducted, including functional requirement testing and user acceptance testing.

The overall test results for the functionality testing are shown in Table 9. The total test cases for each module and the number of passed test results are concluded in the table.

Table 9: Overall Test Case Result

Test Case ID	Total Test Cases	Total Passed
TEST_100	5	5
TEST_200	4	4
TEST_300	7	7
TEST_400	4	4
TEST_500	5	5
TEST_600	7	7
TEST_700	6	6
TEST_800	6	6
	44	44

Next, the user acceptance testing was conducted among adults in Malaysia, including students, chefs, teachers, and others. A total of fifteen users participated in the test. From a scale 1 to 5 representing very unsatisfied to very satisfied, the satisfaction of the users on these aspects are recorded. Table 10 presents the results from user involvement in the HomeChef Mobile App.

Table 10: Overall Test Case Result

No.	Question	Scale					Total
		1	2	3	4	5	
1	How satisfied are you with the overall layout and design of the app?	0	0	0	1	14	15
2	How easy is it to enter and manage your available ingredients in the app?	0	0	0	2	13	15
3	How intuitive is the process of finding recipes based on your available ingredients?	0	0	0	0	15	15
4	How satisfied are you with the speed and performance of the app?	0	0	0	3	12	15
5	How well does the app help you discover new recipes you might not have thought of?	0	0	0	3	12	15
6	How easy is it to navigate between different sections of the app?	0	0	0	0	15	15
7	How satisfied are you with the overall user experience of the app?	0	0	0	3	12	15
8	How satisfied are you with the visual appeal of the recipe photos?	0	0	0	3	12	15
9	How easy is it to understand and use the app's main features?	0	0	0	2	13	15
10	How easy is it to understand the app's icons?	0	0	0	4	11	15

6. Conclusion

In conclusion, this study has delved into the intricate landscape of Recipe Finder Mobile Applications, highlighting their pivotal role in simplifying meal planning and promoting sustainable cooking practices. The identified challenges of culinary monotony, food wastage, and time-consuming recipe searches underscore the significance of innovative solutions such as the "HomeChef" app. By employing a cloud firestore indexing system,this app uniquely addresses these challenges, offering users a comprehensive and user-friendly platform for discovering, managing, and organizing recipes. The Prototyping Model emerges as a suitable development approach, fostering user involvement and feedback throughout the iterative process. The study emphasizes the transformative potential of the "HomeChef" app in fostering culinary creativity, reducing food waste, and enhancing the overall cooking experience. To further enhance this research, future studies could explore user perceptions and experiences with Recipe Finder Mobile Applications, offering valuable insights for continuous refinement and development. Additionally, investigating the impact of such apps on cooking behaviors and sustainability practices could contribute to the broader discourse on technology's role in shaping contemporary culinary practices.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1025-1044

References

[1] L. Z. Bloom, Recipe. Bloomsbury Publishing USA, 2022. Accessed: Dec. 26, 2023. [Online]. Available: https://books.google.com.my/books?hl=en&lr=&id=VSxlEAAAQBAJ&oi=fnd&pg=PR5&dq=Nutrition+is+an+essential+requirement+for+human+sustenance.+Each+delectable+culinary+creation

[2] K. Schanes, K. Dobernig, and B. Gözet, "Food waste matters - A systematic review of household food waste practices and their policy implications," Journal of Cleaner Production, vol. 182, pp. 978–991, May 2018, doi: https://doi.org/10.1016/j.jclepro.2018.02.030.

[3] S. Huseynov and M. A. Palma, "Food decision-making under time pressure," Food Quality and Preference, vol. 88, p. 104072, Mar. 2021, doi: https://doi.org/10.1016/j.foodqual.2020.104072.

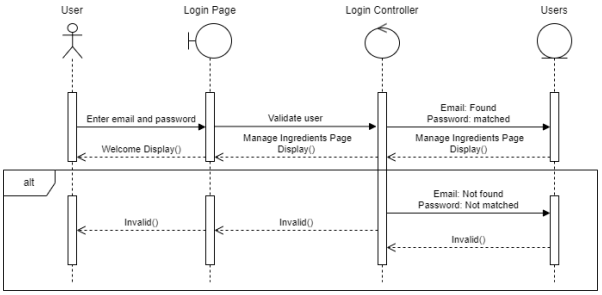
[4] M. Chen, X. Jia, E. Gorbonos, C. T. Hoang, X. Yu, and Y. Liu, "Eating healthier: Exploring nutrition information for healthier recipe recommendation," Information Processing & Management, vol. 57, no. 6, p. 102051, Nov. 2020, doi: https://doi.org/10.1016/j.ipm.2019.05.012.

[5] V. Padme, "ENHANCING ANDROID UI/UX WITH FIRESTORE," 1482. Available: https://www.irjmets.com/uploadedfiles/paper/issue_1_january_2024/48154/final/fin_irjmets1705334661.pdf

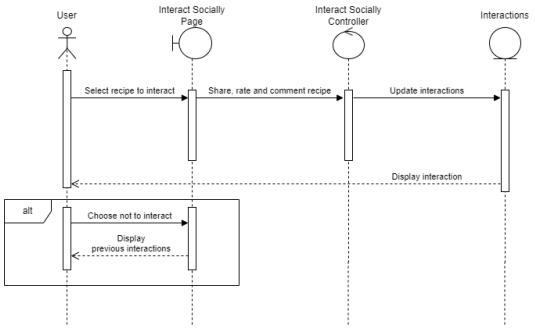
[6] M. Martin, "Prototyping Model in Software Engineering: Methodology, Process, Approach," Guru99.com, Oct. 24, 2019. https://www.guru99.com/software-engineering-prototyping-model.html

[7] J. Rumbaugh, I. Jacobson, and G. Booch, "The Unified Modeling Language Reference Manual," 2021.

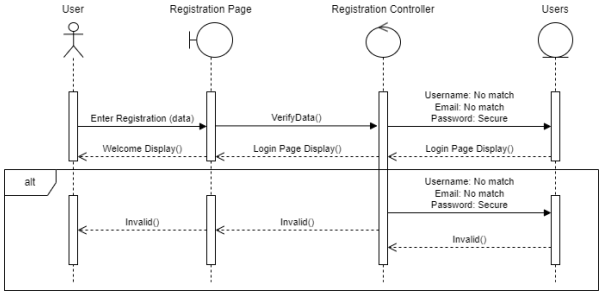
Appendix A: Sequence Diagram



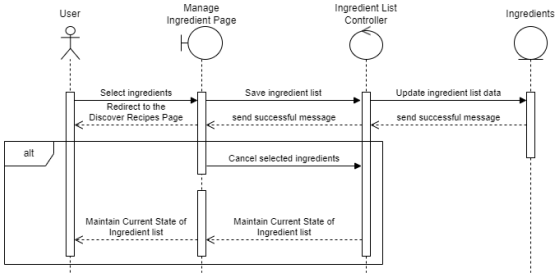
Login



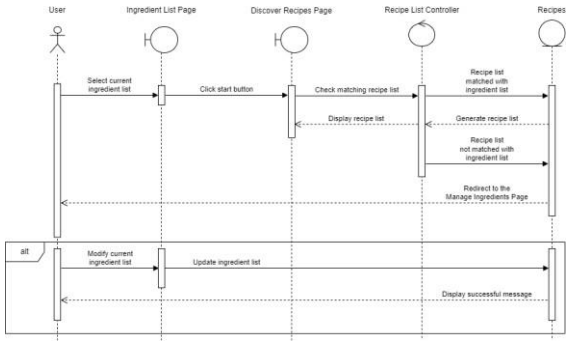
Interact Socially



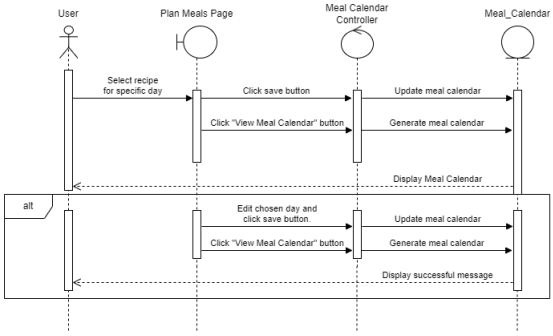
Register



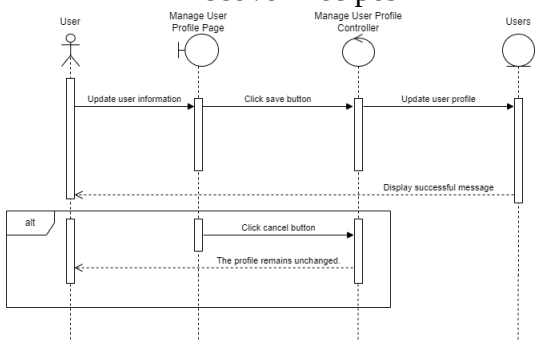
Manage Ingredients



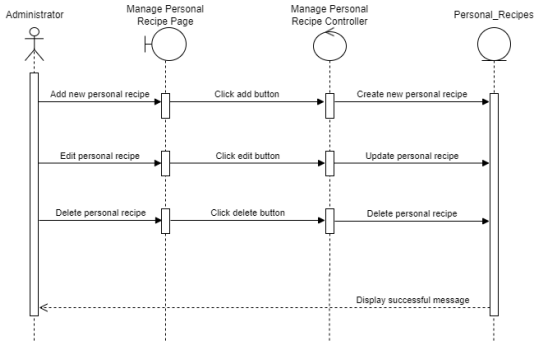
Discover Recipes



Plan Meals



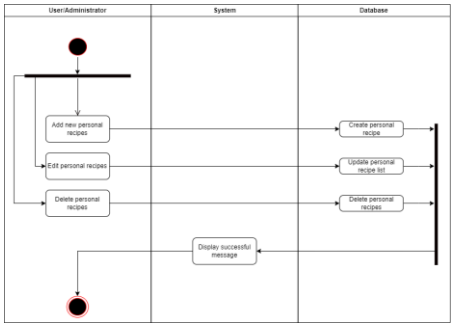
Manage User Profile



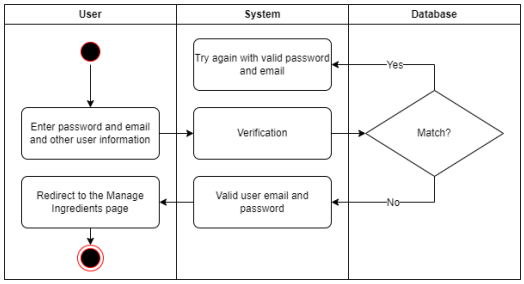
Manage Personal Recipe

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1025-1044

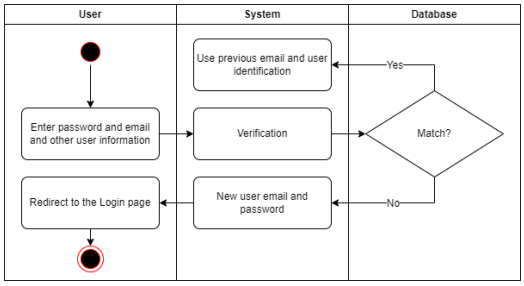
Appendix B: Activity Diagram



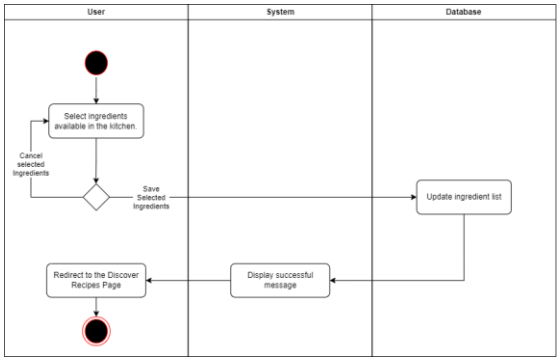
Manage Personal Recipe



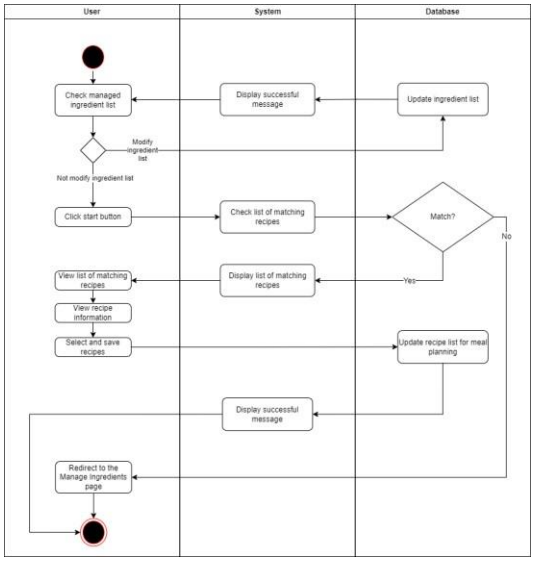
Login



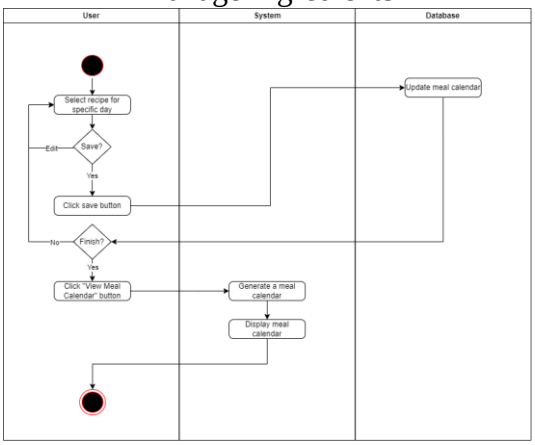
Register



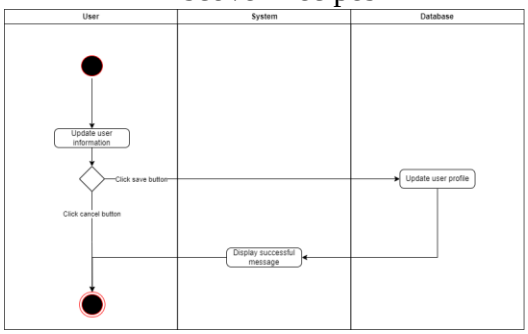
Manage Ingredients



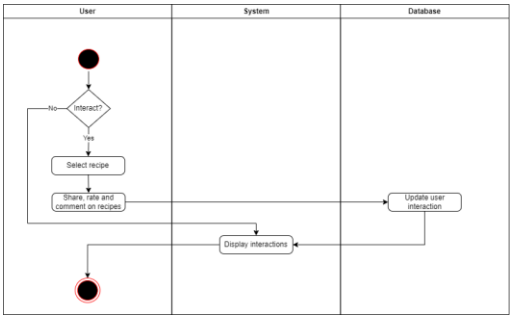
Discover Recipes



Plan Meals



Manage User Profile



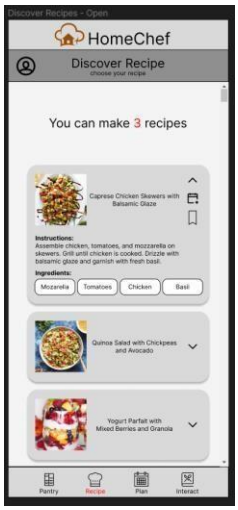
Interact Socially

Appendix C: HomeChef interfaces by using FIGMA

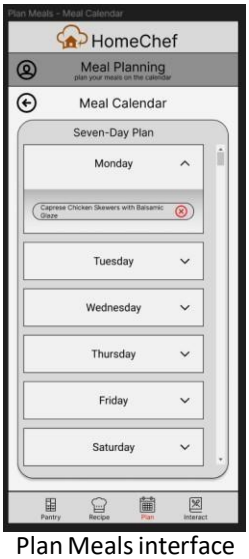
Login interface

Social Interaction interface

Manage Ingredients interface



Discover Recipes interface



Plan Meals interface

Social Interaction interface