Software Requirements Specification

for

<Nutri-Vision>

Version 1.0 approved

Prepared by

Tan Ke Yuan

Ng Yoon Yik

Pan HaoLun

Tee Jeeng Yee

Randy Tan Yu Hong

Muhammad Nahid Bin Nisar

NTU, SC2006 Seitrams

17/04/2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

In line with the Singapore government's Healthier SG initiative, the Seitrams team will develop a mobile application Nutri-Vision that enables the users to obtain the nutritional information of the food they consume in order to make healthier food choices.

1.2 Document Conventions

In this SRS, Headings will be written in **bolded Times New Roman**, **Heading 1: size 18, Heading 2: size 16, Heading 3: size 14**. All other sections will be written in Arial 12, except for tables which will be written in Arial 11.

1.3 Intended Audience

This document is intended for multiple audiences, including:

- Stakeholder: Health Promotion Board Singapore.
- Developers: To implement and verify the specified requirements.
- Project Managers: To oversee the project's progression.
- Users: To learn more about Nutri-Vision’s development background

1.4 Product Scope

Nutri-Vision is a comprehensive dietary tracking app that aligns with Singapore's goals to promote public health and wellness. The application aims to:

- Allow users to set goals and achieve them eventually through a healthy diet. By acknowledging the calorie and macro intake of every single meal.
- Encourage healthier eating habits that fit their health goals.

2. Overall Description

2.1 Product Perspective

Nutri-Vision is a mobile application designed to integrate with existing health and nutrition APIs. The app offers unique functionality by enabling users to scan and identify food items, track their calorie and macro counts, and share their meal plan within the community to promote a healthy eating lifestyle.

2.2 Product Functions

2.2.1 Core User Profile Management

- Manage Personal Information: Allow users to manage and edit their personal information including height, weight, age, gender.
- Health and Dietary Preferences: Enable users to input and update their health goals.

2.2.2 Dietary Tracking and Management

- Meal Logging: Permit users to log meals, including details like food items, portion sizes, and meal times, with options to edit entries before final submission.
- Nutritional Information: Automatically provide detailed nutritional information for scanned or inputted food items, allowing customization based on serving size.

2.2.3 Real-Time Food Recognition and Management

- Capture and Process Food Images: Allow users to upload food images from the gallery or capture in real-time, with tools to crop, rotate, and adjust images.
- Food Identification: Identify foods from images, and provide options to re-upload in case of errors like blurriness or incorrect file types.

2.2.4 Historical Data

- Meal History Access: Display a reverse chronological history of meal entries, allowing users to view and modify past entries.

2.3 User Classes and Characteristics

- Casual Users: Individuals looking to make more informed food choices without specific dietary needs.
- Health Enthusiasts: Users who frequently track their dietary habits and fitness regimes.
- Individuals with Dietary Restrictions: Users with specific dietary requirements based on health conditions or personal choices.
- Medical Professionals: Dieticians and healthcare providers who may recommend the app to patients.

2.4 Operating Environment

Operating Platform

- Development platform: Visual Studio Code.
- Operating system platform: Android 13.
- Device used: Pixel 8.

Backend

- Data is stored on Firebase.
- Authentications are authenticated through Firebase Authenticator.

External API

- CalorieNinja API for nutritional information retrieval.
- OpenAI API for image recognition.

2.5 Design and Implementation Constraints

2.5.1 Regulatory Compliance

The app must comply with health data protection regulations.

2.5.2 Technology Limitations

The accuracy of food identification and nutritional information retrieved will be affected by the capabilities of the integrated APIs.

2.6 Assumptions and Dependencies

2.6.1 Assumptions

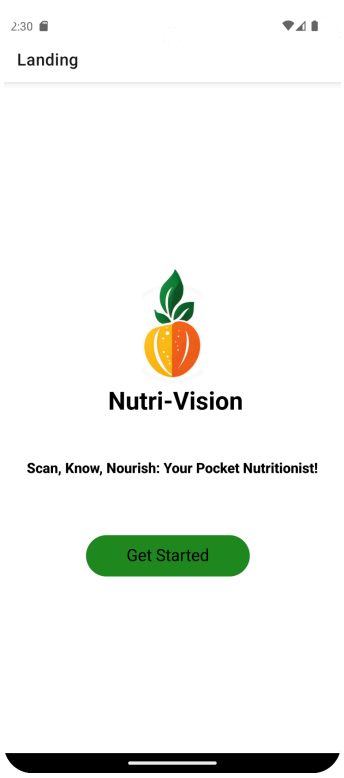
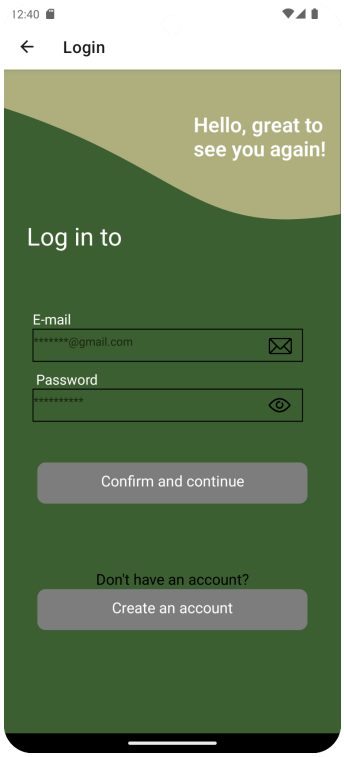
It is assumed that users have operational knowledge of smartphones and that the device's camera is working.

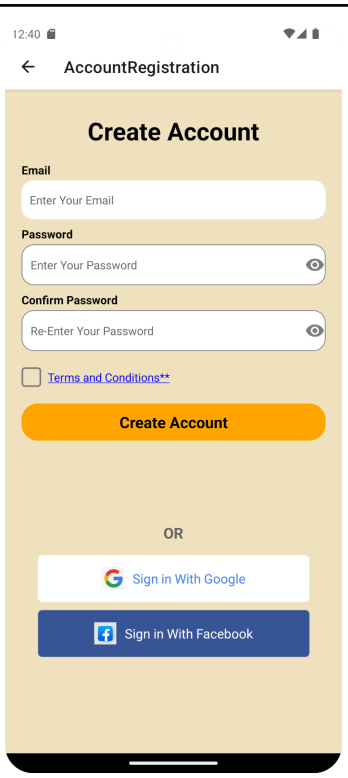
2.6.2 Dependencies

The app's performance is dependent on the reliability of the Food Recognition and Nutritional Information APIs.

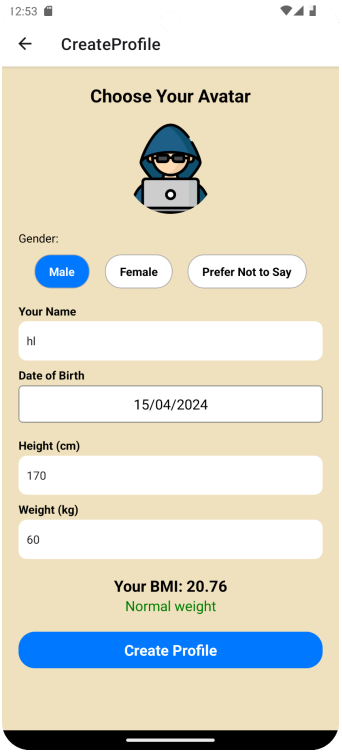
3. External Interface Requirements

3.1 User Interfaces

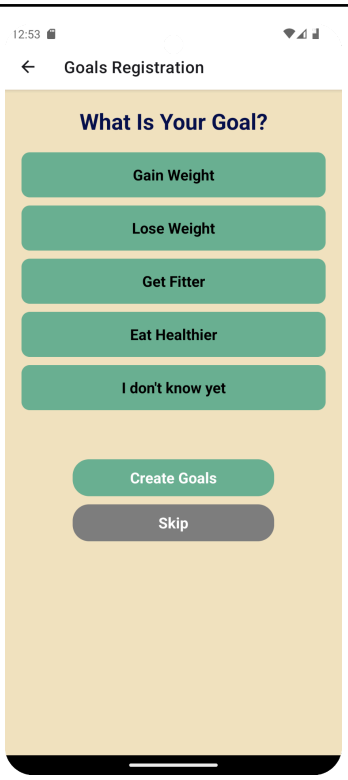
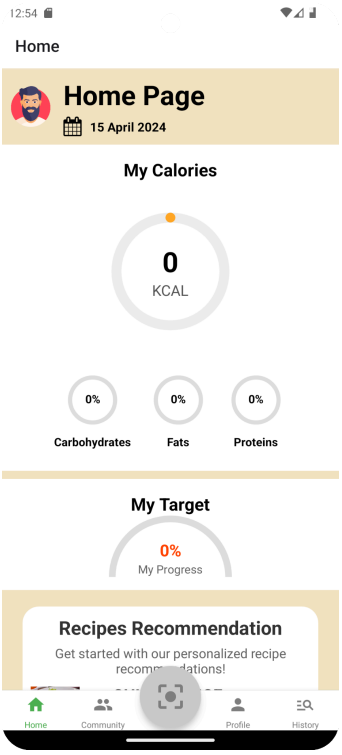
	This shows the first page that the user will see when the user clicks on the app from his or her phone screen. The “Get Started” button will navigate to “Login” page.
	This shows the login page for existing/registered users to login. After entering the correct email and password, the user will be logged in.

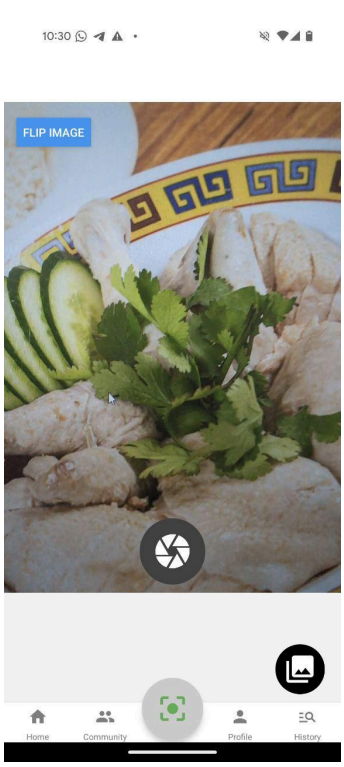


This shows the registration page for new users to register their user account. The user needs to enter a valid email, a strong password (Upper and lower case letter, numeric, special character) and the password again to confirm password.

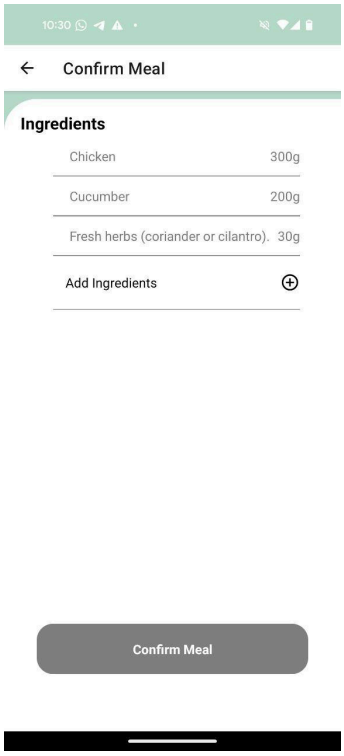


This shows the page for the new user to enter his/her personal information details

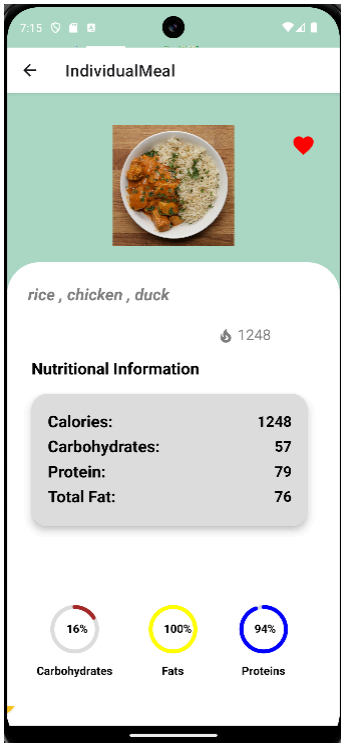
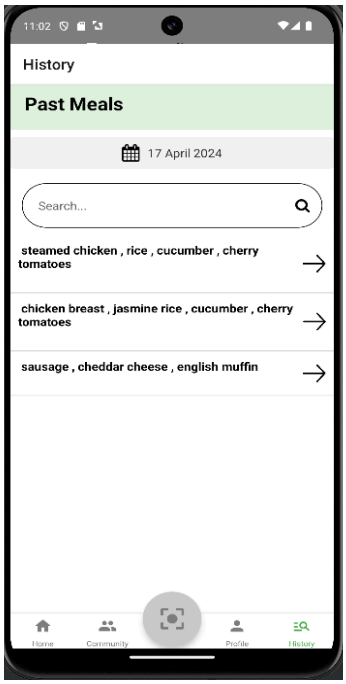
	This shows the goals registration page that will be navigated to after create profile page. This page allows the user to select a goal and the system will take this goal into consideration to give recommended daily calorie and macros intake, if the user is unsure of which goals to choose, the user is able to skip this page.
	This shows the user's daily calorie and macro limit with a tab below to navigate to other pages.



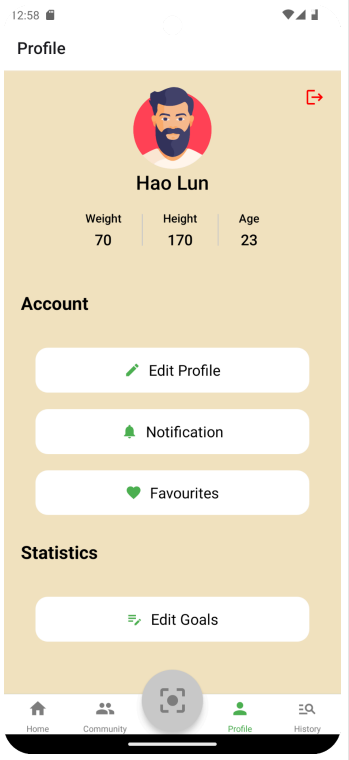
This page allows the user to take a picture of the meal.



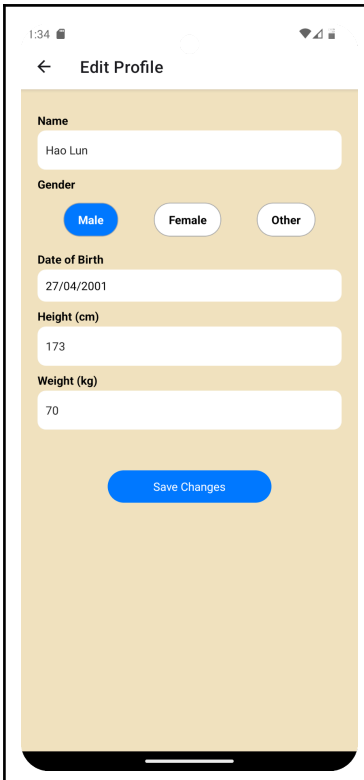
This page serves as a confirmation for the picture to be uploaded to the CalorieNinja API.

	This page will return the nutritional information from the CalorieNinja API and display it to the user.
	This shows the past meal entries filtered by dates.

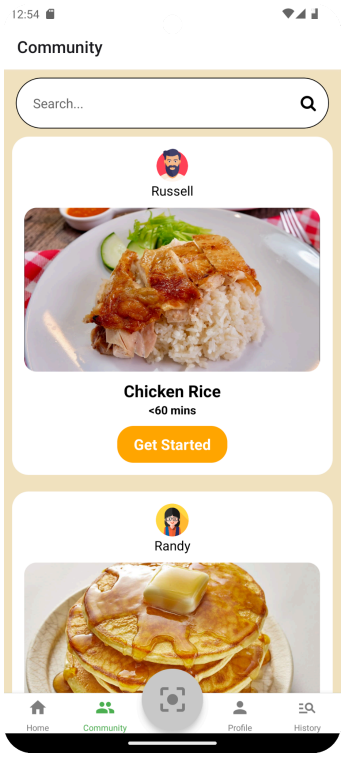
This page displays the meals that the user sets as favourite so it is easier for them to refer.



This page displays a brief description of the user's personal information, after clicking on various buttons, the user can edit his/her personal information; toggle on and off notifications; view their favourites page; or edit their goals



This page allows the user to edit/update his/her personal information.



This page allows users to share their daily meals and interact with other users.

3.2 Hardware Interfaces

The application will require an Android phone with a camera and internet connection to run. No other hardware devices are required.

3.3 Software Interfaces

3.3.1 Firebase Authentication

The user will be prompted to enter their account email and password on the login screen, which will be verified by Firebase Authentication. Users who do not have an existing account will have to register through the application to create a personal account for themselves, where the account details will be stored in the Firebase.

3.3.2 OpenAI API

After the user takes a picture of the meal, the image will be sent to OpenAI API to recognise what are the components in the meal and an estimate of the portions of the individual components. Then the API will send this information as text to the second API, CalorieNinja.

3.3.3 CalorieNinja API

Upon receiving the ingredients of the meal from OpenAI API, the CalorieNinja API will analyse the calorie and macro count of the meal and return this information to Nutri-vision UI for the user to see. At the same time, this information will also be stored in the Firebase database as Meal History so that the user can access it if needed.

3.4 Communications Interfaces

Nutri-Vision uses an internet connection with the internet protocol as follows:

- To connect to APIs to identify and retrieve nutritional information of meal scanned.
- To access the user's profile information like their email and password from the Firebase Database API.

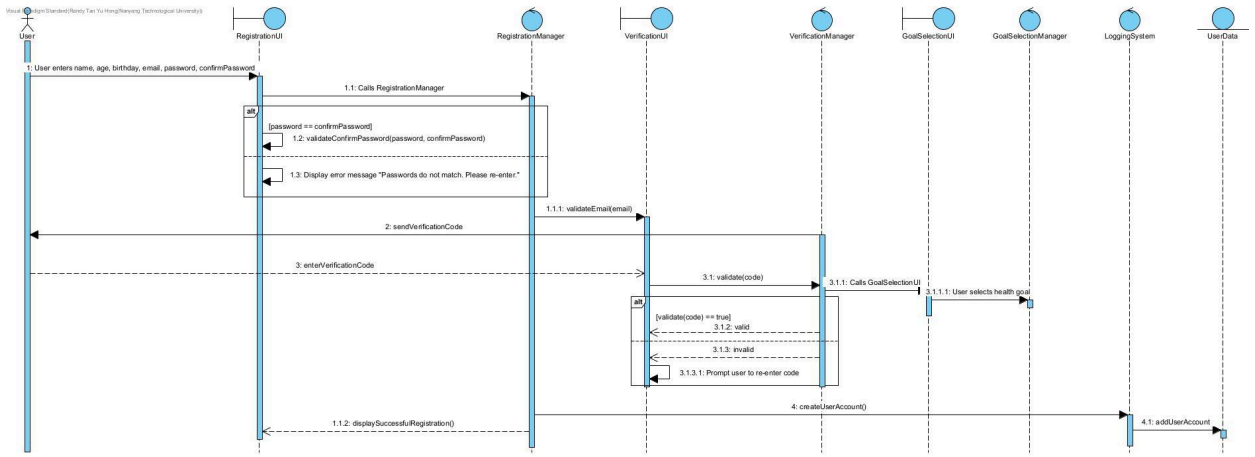
4. System Features

4.1 Registration

4.1.1 Description and Priority

Users will be able to register a new user account with Firebase. Priority for this feature is high as users must register an account in order to use the application.

4.1.2 Stimulus/Response Sequences



4.1.3 Functional Requirements

The app must have an interface for new users to create an account.

REQ - 1.1	The page must request for the new user's email and password. ● The page must request for the new user's email address. ● The page must request for the new user's password. ● The page must request for the new user to re-enter his password.
REQ - 1.2	There must be a check box to prompt users to accept the app's terms and conditions.
REQ - 1.3	There must be a "Register" button for users to click on after they have furnished all the details. The app must be able to verify whether the email address and password are valid before proceeding with the account creation

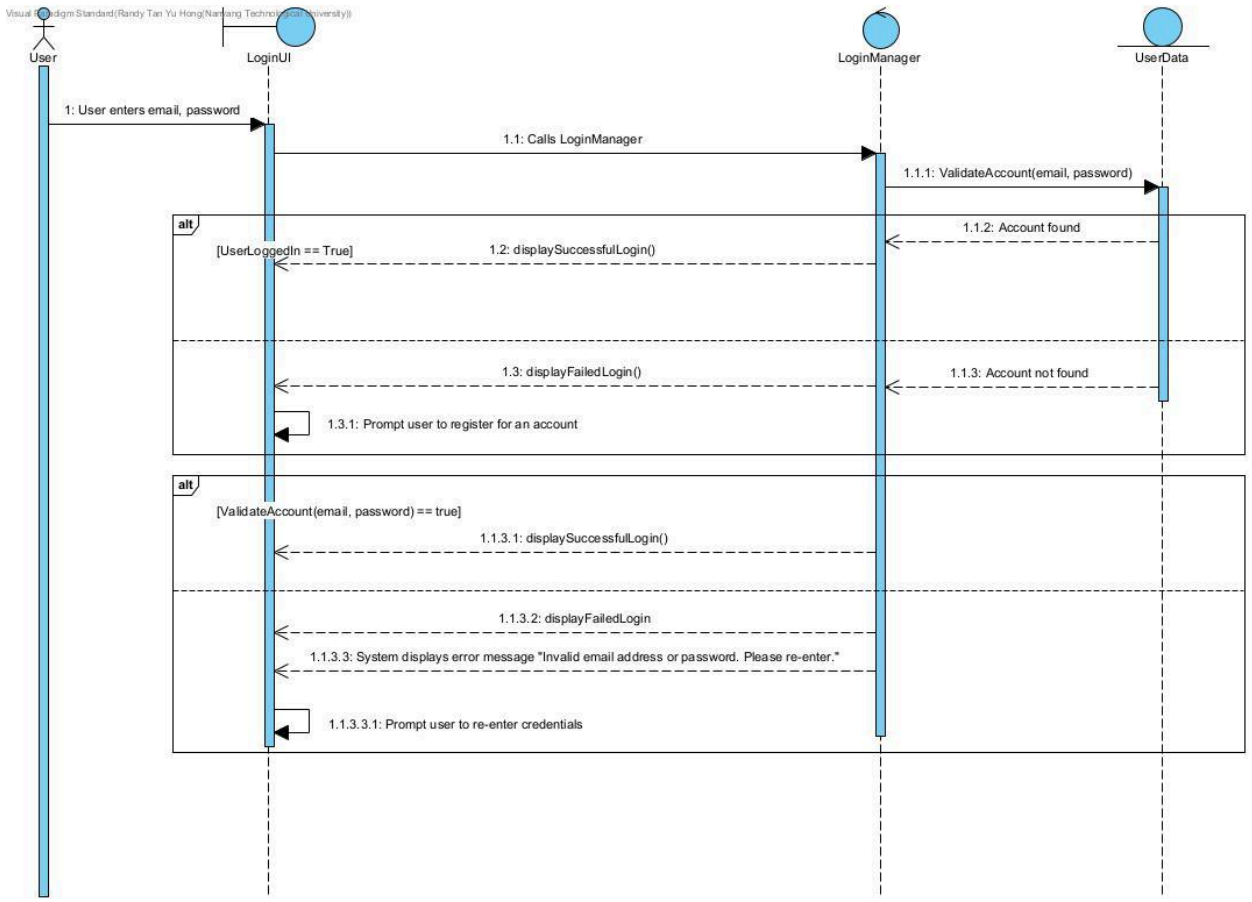
ERROR - 1.4 User has left out one of the required fields to complete.	User will not be allowed to proceed with account creation. User will be directed to the field left blank.
ERROR - 1.5 User has entered an invalid email address.	User will not be allowed to proceed with account creation. User will be prompted to input a valid email address.
ERROR - 1.6 User has entered a weak password.	User will not be allowed to proceed with account creation. User will be prompted to input a strong password.
ERROR - 1.7 User has entered a different password for confirm password.	User will not be allowed to proceed with account creation. User will be prompted to input a matching password.

4.2 Login

4.2.1 Description and Priority

User logs into his user account by entering his email address and password. Priority for this feature is high as every user has to login to the application to use any of the features in Nutri-Vision.

4.2.2 Stimulus/Response Sequences



4.2.3 Functional Requirements

The app must have an interface for users to login.

REQ - 2.1	The login screen must contain blanks for users to input their email address and password.
REQ - 2.2	There must be an option for a new user to create an account on the login screen.

ERROR - 2.3 Email address	Application rejects the user's login attempt; prompts user to create an account.

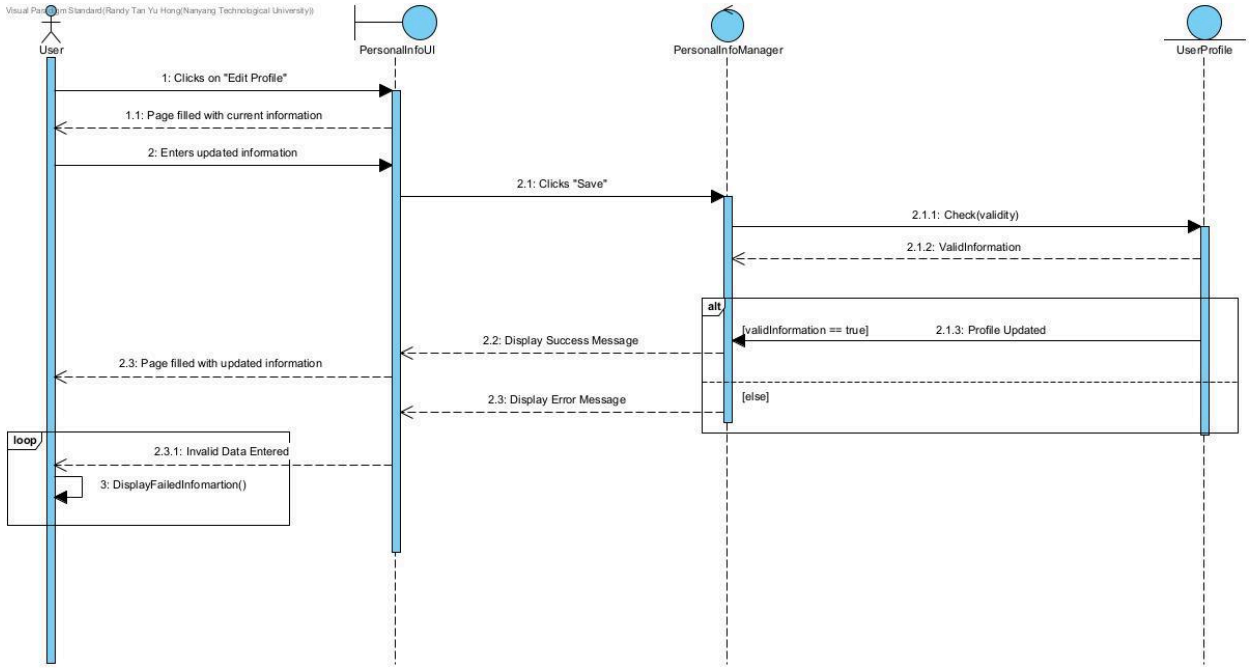
does not exist.	
ERROR - 2.4 Invalid password.	Application rejects the user's login attempt; prompts user to re-enter the correct password.

4.3 Personal Information

4.3.1 Description and Priority

User is able to input their personal information after creating their account and log in for the first time. Priority for this feature is medium.

4.3.2 Stimulus/Response Sequences



4.3.3 Functional Requirements

The app must have an interface for users to input their personal information.

REQ - 3.1	The login screen must contain blanks for users to input their personal information
REQ - 3.2	There must be a button for the user to save their personal information and navigate to the next page.

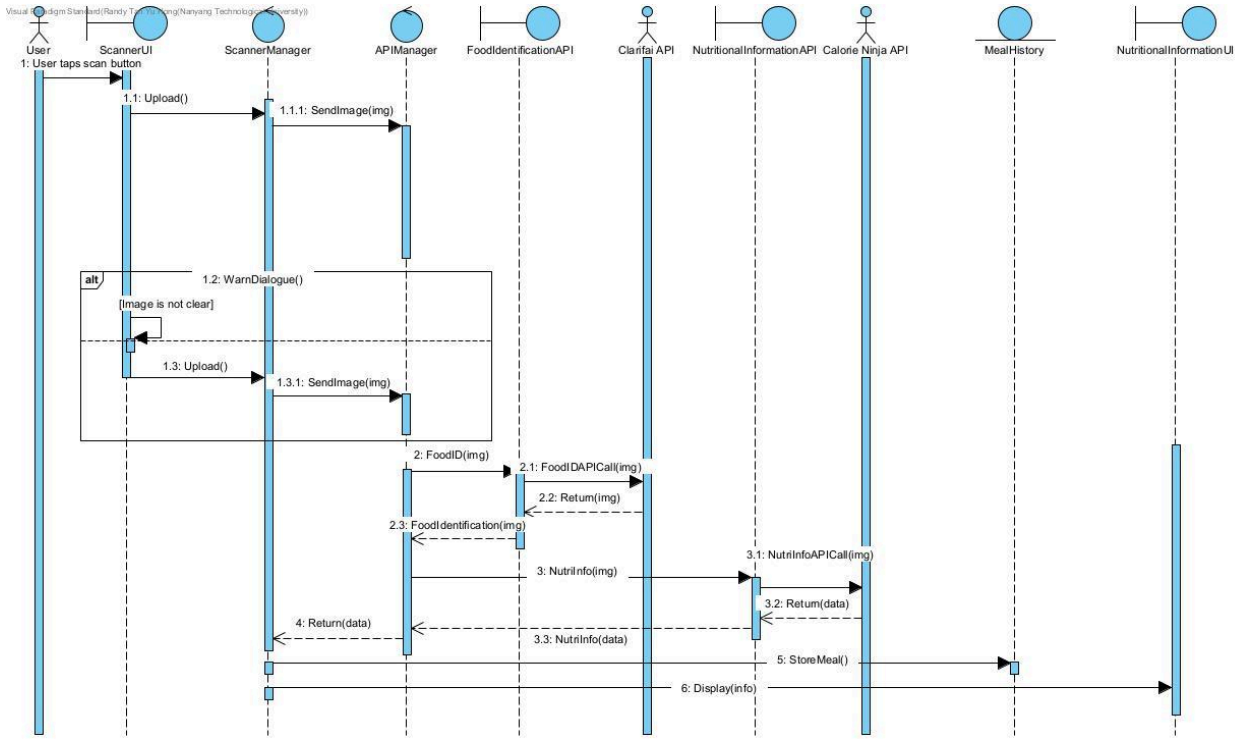
ERROR - 3.3 Empty fields.	App rejects the user's create profile attempt; prompts user to fill up the missing fields.
ERROR - 3.4 Invalid/Out of bound fields.	App rejects the user's create profile attempt; prompts user to re-enter the invalid fields.

4.4 Log Meal

4.4.1 Description and Priority

User is able to take or upload a picture of their meal and retrieve its nutritional information. Priority for this feature is high as it is the main feature of Nutri-vision.

4.4.2 Stimulus/Response Sequences



4.4.3 Functional Requirements

The app must have an interface for users to view which entry they want to see in detail.

REQ - 4.1	The scanner screen must have the function to take a photo and upload the photo to the API.
REQ - 4.2	The nutritional information page must display the retrieved nutritional information from the API.

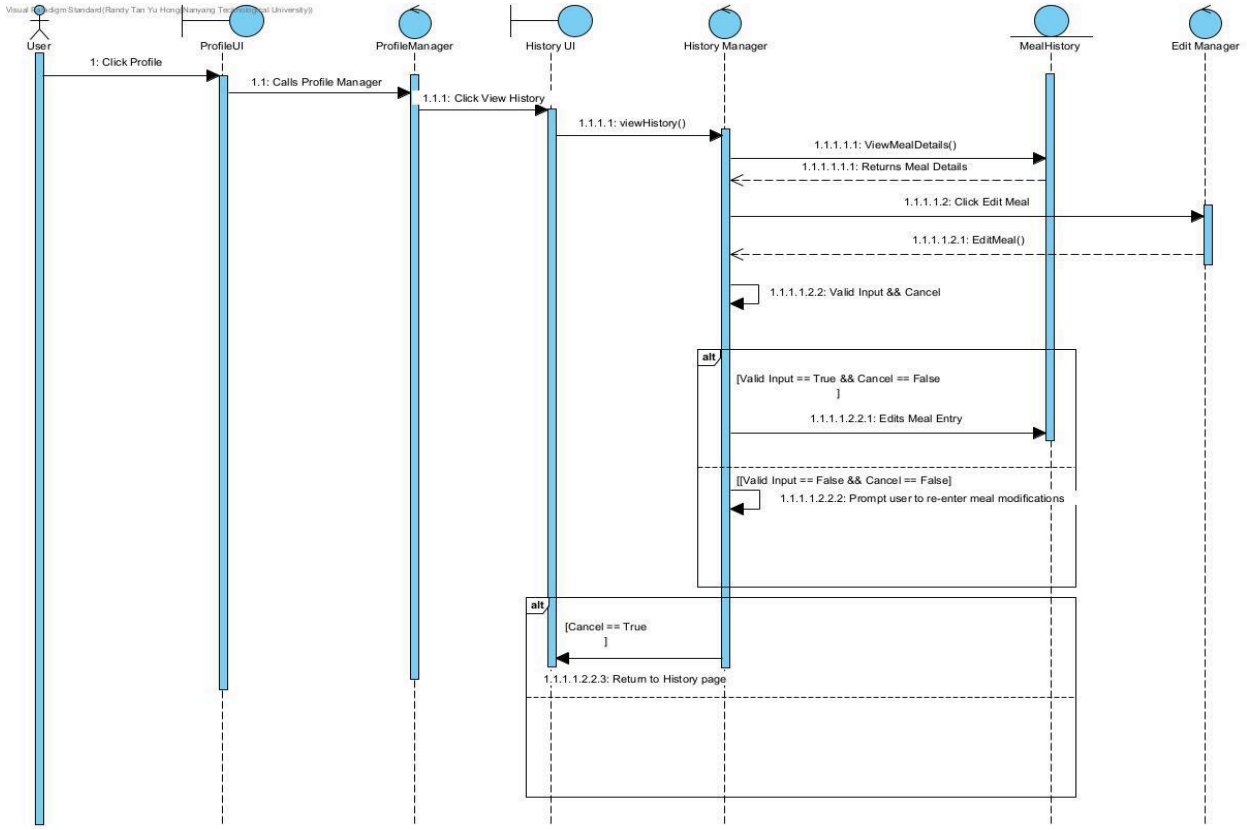
ERROR - 4.3 Blurry photo	App prompts the user to retake or re-upload a photo of the meal.

4.5 View History

4.5.1 Description and Priority

User is able to view past meal entires' nutritional information. Priority for this feature is medium.

4.5.2 Stimulus/Response Sequences



4.5.3 Functional Requirements

The app must have an interface for users to view their past meal entries.

REQ - 5.1	The meal history page will have different meal entries filtered by date.

ERROR - 5.2 History is empty	App shows “No meals logged yet”.
ERROR - 5.3 User chooses invalid date	App shows an error in selecting date, prompts user to select a valid date.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Performance

- Image Processing Speed: The app shall scan and process any uploaded images within 10 seconds.
- Response Time: The app shall respond within 5 milliseconds for each button click on the interface, ensuring a responsive user experience.
- User Interaction: Texts within the app must be clear and understandable.

5.1.2 Accuracy

- Nutritional Information: The nutritional content displayed by the app should be accurate to the nearest whole number. Additionally, users should have the capability to edit this nutritional information, allowing them to adjust it based on more specific or updated data they might have.

5.2 Safety Requirements

5.2.1 Data Handling

The app shall ensure that all user data is handled securely to prevent data leaks or breaches.

5.2.2 Operational Safety

The application shall not perform any operations that could harm the physical health of users, such as suggesting harmful dietary restrictions.

5.3 Security Requirements

The app prioritises the protection and privacy of user data. To enhance account security, a strong password is required to create an account. Additionally, password entries are masked with black dots, with an option for users to view their passwords if needed. Compliant with Singapore's Personal Data Protection Act (PDPA), Nutri-Vision is dedicated to using collected data solely to enhance user experience, strictly prohibiting the release of this data to external third parties.

5.4 Software Quality Attributes

5.4.1 Scalability

- The app must be able to accommodate large volumes of users and data over time.
- The app shall support up to 10,000 users simultaneously without crashing.
- The app shall support up to 10,000 API calls simultaneously without crashing.
- The app shall support up to 10,000 responses to users without any slowdown in performance.

5.4.2 Reliability

- The app shall provide a reliable user experience with minimal disruptions.
- The app shall handle up to 10,000 users simultaneously without any slowdown in performance.

5.4.3 Maintenance

- The app shall have a maximum downtime of 6 hours during maintenance periods.
- The app shall notify the users of any impending updates or maintenance.
- The app shall receive monthly maintenance to ensure its core functions are running well without failure.
- The app shall receive updates whenever the developers have tried and tested the updates amongst themselves to ensure quality and reliability before releasing them to the users.

5.4.4 Compatibility

- The app shall mainly run on the Android OS.
- The app should be compatible with a wide range of Android devices, including different screen sizes, resolutions, and versions of the Android OS.

5.5 Business Rules

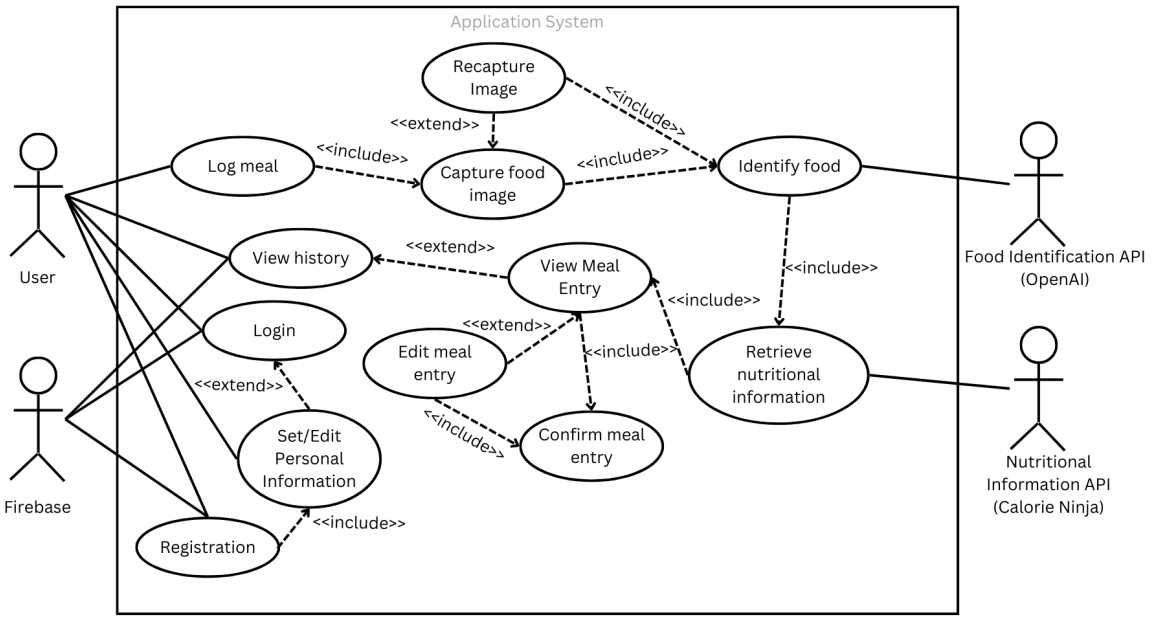
Our application does not have any business rules pertaining to any specific user group.

6. Other Requirements

6.1 Data Dictionary

Term	Definition
Action Plan	The user can choose the action plans, like nutritional information, recommendations on food, and recommendations on meal plans.
Activity Level	A descriptor representing how much physical activity the user does on a daily basis.
Calorie	It's a measure of how much energy your body can get from consuming a particular food or beverage.
Cuisine Preference	User’s preferred type of food that is cooked in a specific way based on a culture's ingredients, region, and traditions.
Diet	User-logged information regarding their preferred food choices including lifestyle choices, allergens and intolerances.
Dietary Plan	A possible option of a diet that fulfils the user’s medical needs, cuisine and dietary preferences.
Dietary Preference	User’s preference for certain foods for personal, medical, religious or cultural reasons.
Entry	Generated/Edited list of food items with nutritional information.
Health Goals	User’s target for weight loss/gain.
Healthy/Healthier	Options of food that adhere to: - Users’ dietary requirements and preferences; - Users’ basal metabolic rate.
History	Date, time, and nutritional information that was previously logged, sorted in reverse chronological order.
Macros	Macros mainly consist of Carbohydrates, Protein, Fats
Meal	Any image or text that contains any amount of food.
Medical Information	Information including user’s allergies and intolerances, and the level of severity.
Notifications	Push Notifications include reminders for meals, reminders to stick to goals and reminders to log the user’s meal.
Nutritional information	Values of different nutrients per serving, such as calories, carbohydrates, protein, and fats.
Personal information	Information including user’s height and weight, gender of user, age, and medical conditions.
Recommendations	List of food items curated to users' needs.
Report	A summary of macros, nutrition and weight gains/losses.
Satiety	The user’s feeling of fullness gratified beyond the point of satisfaction from food.

6.2 Use Case Model



Actor	Use Cases
Food Identification API (OpenAI) Nutritional Information API (Calorie Ninja) Firebase User	Capture Food Image Confirm Meal Entry Edit Meal Entry Identify Food Log Meal Login Recapture Image Registration Retrieve Nutritional Information Set/Edit Personal Information View History View Meal Entry

Term	Definition
Food Identification API	An API that uses machine learning and computer vision algorithms to analyse images or descriptions of food items and identify their names.
Nutritional Information API	An API that can retrieve detailed nutritional information about food products, such as the calorie content and macro information.
Firebase	A platform that provides backend services, such as data storage of meal history and user profile details, and authentication
User	An individual who downloads, installs and uses our mobile application, Nutri-Vision.

Capture Food Image	Uploading an image of a food item from the user's gallery; or taking a real-time image of a food item using the user's smartphone camera.
Confirm Meal Entry	The act of verifying and finalising the entry of a meal into the daily meal log.
Edit Meal Entry	The act of modifying the details of a meal before it is logged into the meal log; or modification of a previously entered meal, to correct any mistakes or to update information about the foods consumed.
Identify Food	The process of recognizing or determining the name of a food item using the Food Identification API.
Log Meal	The act of recording or entering details about a meal, such as the foods consumed and portion sizes.
Login	The process of accessing the mobile application by providing credentials that uniquely identify the user - the email and password.
Recapture Image	Reuploading an image of a food item from the user's smartphone gallery; or retaking a real-time image of a food item using the user's smartphone camera because of blurry/invalid images.
Registration	The process of signing up or creating an account on Nutri-Vision. The user has to provide an email and a desired password to create a unique account that allows access to Nutri-Vision's features and services.
Retrieve Nutritional Information	The process of retrieving detailed nutritional information about food products, such as the calorie content and macro information, using the Nutritional Information API.
Set/Edit Personal Information	The act of entering/modifying the user's personal details on Nutri-Vision, such as their name, age, gender, contact information, health goals, dietary preferences, and nutritional goals.
View History	The act of accessing a reverse chronological record of past meal entries by the user.
View Meal Entry	The act of accessing each individual meal entry in the user's history.

6.3 Use Case Descriptions

Use Case ID:	LM1		
Use Case Name:	Log meal		
Created By:	Haolun	Last Updated By:	Jeeng Yee
Date Created:	08/02/2024	Date Last Updated:	14/04/2024

Actor:	User
Description:	User wants to know about their meal's nutritional information.
Preconditions:	User must have the app installed on their device. User must have an account tied to the system.
Postconditions:	System brings the user to the Scanner UI.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User clicks on the scanner button from home page 2. User will be brought to the Scanner UI
Alternative Flows:	LM1-AF-S1 user decides not to log meal: 1. User selects back. 2. System brings the user to the home page.
Exceptions:	The database of the Calorie Ninja API does not have the nutritional values of the food input by the user
Includes:	Capture food image, Recapture Image, Identify food
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

Use Case ID:	CFI1		
Use Case Name:	Capture food image		
Created By:	Haolun	Last Updated By:	Jeeng Yee
Date Created:	08/02/2024	Date Last Updated:	14/04/2024

Actor:	User
Description:	User captures a food item image.
Preconditions:	1. User must have a working camera on their device. 2. User must have allowed camera and gallery access permission for the app.
Postconditions:	1. Nutritional information of the scanned food is displayed to the user. 2. Food item is logged in the user's meal history.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User selects the option to take a live photo of a food item or upload a photo from their gallery 2. The app processes the image, identifies the ingredients and retrieves the nutritional information from the integrated API. 3. User will be brought to the Confirm Meal Page UI 4. User can edit, delete or add ingredients to the list displayed from OpenAI API results 5. User presses the 'Confirm Meal' button 6. The app displays the nutritional information to the user. 7. The app writes the nutritional information to the Firebase and it is readily accessible at the History Page UI
Alternative Flows:	CF1-AF-S1 user decides to upload image instead of scan: 1. System processes the image, identifies the ingredients and retrieves the nutritional information from the integrated API. CF1-AF-S2 user decides not to capture Image: 1. User selects back. 2. System brings the user to the home page.
Exceptions:	If the app cannot recognize the food item, the user will be prompted to enter the food details manually.
Includes:	Recapture food image, Identify food
Special Requirements:	1. Scanning and food identification features must produce an output in less than 10 seconds, to ensure a smooth user experience.
Assumptions:	Nil
Notes and Issues:	Food might be covered/not seen in the photo, how to calculate the calories of those? Manually add to the ingredients list

Use Case ID:	RFI1		
Use Case Name:	Recapture food image		
Created By:	Haolun	Last Updated By:	Jeeng Yee
Date Created:	08/02/2024	Date Last Updated:	14/04/2024

Actor:	User
Description:	User recaptures the food item image as the image was not clear.
Preconditions:	1. User must have a working camera on their device. 2. Capture food image must happen first.
Postconditions:	1. Nutritional information of the scanned food is displayed to the user. 2. Food item is logged in the user's dietary history.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	1. The user will be prompted to take or upload another photo of the food item. 2. The app processes the image, identifies the ingredients and retrieves the nutritional information from the integrated API. 3. User will be brought to the Confirm Meal Page UI 4. User can edit, delete or add ingredients to the list displayed from OpenAI API results 5. User presses the 'Confirm Meal' button 6. The app displays the nutritional information to the user. 7. The app writes the nutritional information to the Firebase and it is readily accessible at the History Page UI
Alternative Flows:	RFI1-AF-S1 user's image is not clear again: 1. Repeat RFI1
Exceptions:	If the system cannot recognize the food item, the user will be prompted to enter the food details manually.
Includes:	Identify food
Special Requirements:	1. Scanning and food identification features must produce an output in less than 10 seconds, to ensure a smooth user experience.
Assumptions:	Nil
Notes and Issues:	Nil

Use Case ID:	VH1		
Use Case Name:	View History		
Created By:	Muhammad Nahid	Last Updated By:	Jeeng Yee
Date Created:	08/02/2024	Date Last Updated:	14/04/2024

Actor:	User
Description:	User gets to view his past logged meals and track his dietary progress.
Preconditions:	User must have logged at least one meal prior.
Postconditions:	System will direct the user to the Individual Meal Page should the user click on the specific meal entry to view more details.
Priority:	Medium
Frequency of Use:	High
Flow of Events:	1. User navigates to the History section on the app. 2. User selects the date to view entries logged on that specific date.
Alternative Flows:	VH1-AF-S1 if user has not logged any meals: 1. System shows “No meals logged yet”. VH1-AF-S2 if user selects back: 1. System brings user back to home page.
Exceptions:	Nil
Includes:	View Meal Entry.
Special Requirements:	Nil
Assumptions:	Users saved past meal logs accurately.
Notes and Issues:	Nil

Use Case ID:	VME1		
Use Case Name:	View Meal Entry		
Created By:	Muhammad Nahid	Last Updated By:	Jeeng Yee
Date Created:	08/02/2024	Date Last Updated:	14/04/2024

Actor:	User
Description:	User gets to view a particular meal's nutritional info.
Preconditions:	User must have scanned a food item and clicked on a specific meal entry from History Page.
Postconditions:	User will confirm the meal entry or edit meal entry.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User clicks on the specific meal entry to view more details. 2. System brings user to the Individual Meal Page of the particular meal entry selected. 3. Individual Meal Page displays the macro-nutrients values of the meal entry. 4. System brings user back to home page.
Alternative Flows:	VME1-AF-S1A if user decides to select back and was previously from History page: 1. User selects back. 2. User is brought to History Page. VME1-AF-S1B if user decides to select back and was previously from Scanner Page: 1. User selects back. 2. User is brought to Scanner Page.
Exceptions:	Nil
Includes:	Confirm Meal Entry.
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

Use Case ID:	EME1		
Use Case Name:	Edit Meal Entry		
Created By:	Muhammad Nahid	Last Updated By:	Jeeng Yee
Date Created:	08/02/2024	Date Last Updated:	14/04/2024

Actor:	User
Description:	User gets to edit his meal entry to improve the accuracy.
Preconditions:	User must have captured food image from camera or uploaded food image from gallery
Postconditions:	System will save meal entry upon confirmation, and write to Firebase.
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1. Confirm Meal Page displays a list of ingredients identified by OpenAI API after user captures or uploads an image 2. User can add new ingredients that were not captured by the API, edit or delete existing ingredients. 3. User presses ‘Confirm Meal’ Button to confirm.
Alternative Flows:	EME1-AF-S1 If user decides to select back: 1. User selects the “back” button. 2. User is brought to Scanner page.
Exceptions:	Nil
Includes:	Confirm Meal Entry.
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

Use Case ID:	CME1		
Use Case Name:	Confirm Meal Entry		
Created By:	Muhammad Nahid	Last Updated By:	Jeeng Yee
Date Created:	08/02/2024	Date Last Updated:	14/04/2024

Actor:	User
Description:	User confirms a meal entry after scanning or uploading image, which then gets written to the Firebase.
Preconditions:	User must have captured or uploaded an image prior.
Postconditions:	System will update the database on the user's meal in Firebase. System will update the History Page. System will direct the user back to the home page.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User clicks on the ‘Confrim Meal’ button. 2. System updates History Page with new meal entry. 3. If the meal logged fits the user’s target according to his chosen goal, a green button which states ‘It fits your target!’ is shown. 4. System updates the database. 5. User is brought to the home page.
Alternative Flows:	CME1-AF-S1 If user selects back: 1. User selects back. 2. User is brought to the Scanner Page. CME1-AF-S4 If meal entry logged does not fit target: 1. A red button which states ‘It does not fit your target!’ is shown. 2. System updates the database. 3. User is brought to the home page.
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

Use Case ID:	RG1		
Use Case Name:	Registration		
Created By:	Haolun	Last Updated By:	Haolun
Date Created:	08/02/2024	Date Last Updated:	27/02/2024

Actor:	User
Description:	User registers a unique account with email and password.
Preconditions:	1. User must have the app installed. 2. User account must not already exist in the database.
Postconditions:	User is able to log in with their unique email and password.
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. User will register on the login page. 2. User has to register a brand new email 3. User has to register a brand new password. 4. The email and password will be stored in the database. 5. User will be able to login with their email and password
Alternative Flows:	RG1-AF-S1 if user has already registered: 1. The user has registered and proceeds to login. RG1-AF-S2 if email already exists (After checking with the database): 1. User has to re-enter another email . 2. User has to register a brand new password. 3. The email and password will be stored in the database. 4. User will be able to login with their email and password. RG1-AF-S3 If the password does not meet the requirements: 1. System will prompt user about the requirements 2. User has to re-enter a new password. 3. The email and password will be stored in the database. 4. User will be able to login with their email and password
Exceptions:	Nil
Includes:	Set/Edit Personal Information
Special Requirements:	The password must have a minimum of 8 characters, 1 upper case, 1 lower case, 1 numeric character, 1 special symbol.
Assumptions:	Nil
Notes and Issues:	Nil

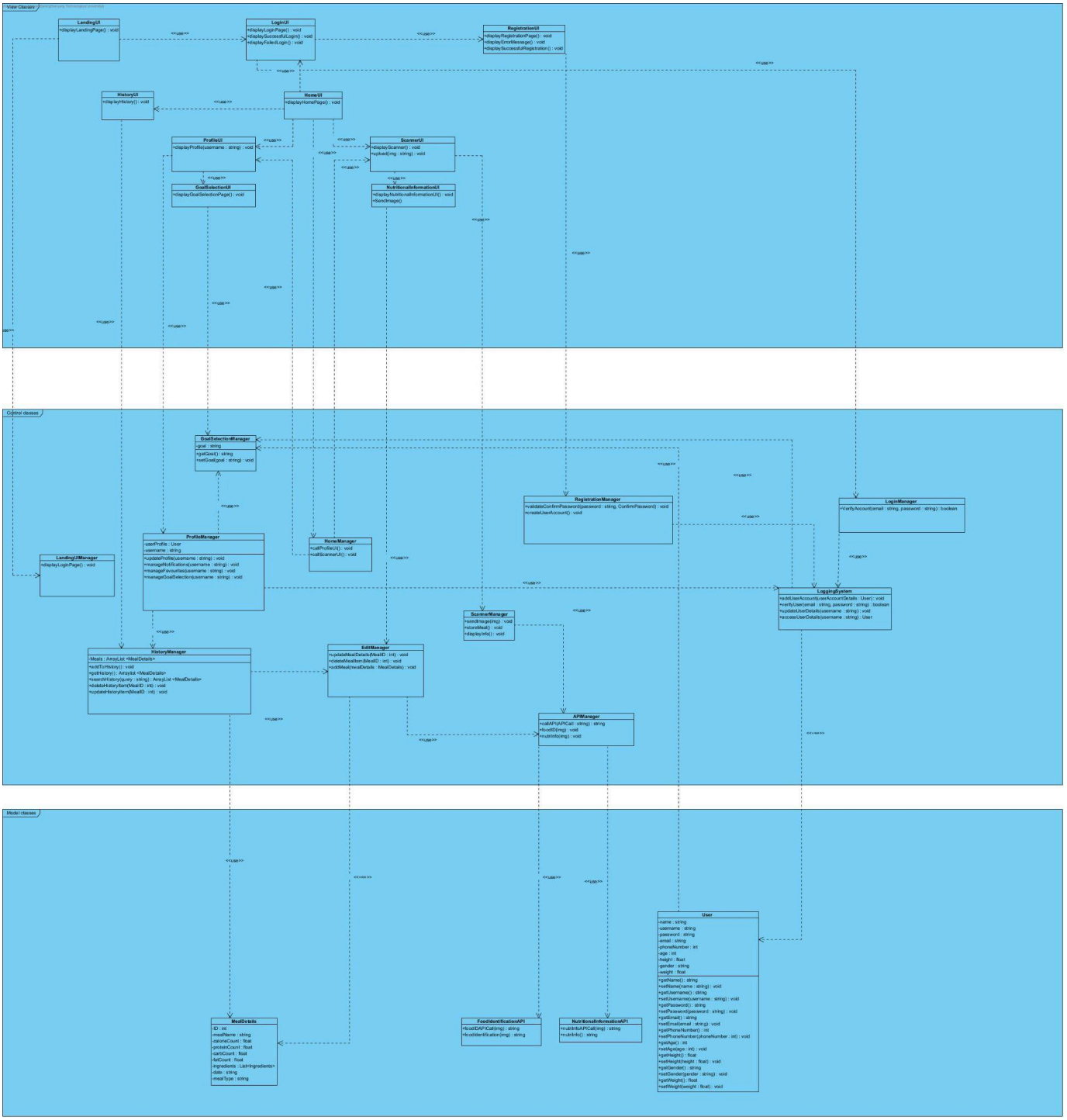
Use Case ID:	SEPI1		
Use Case Name:	Set/Edit Personal Information		
Created By:	Haolun	Last Updated By:	Haolun
Date Created:	08/02/2024	Date Last Updated:	27/02/2024

Actor:	User
Description:	User is able to set and edit their personal information.
Preconditions:	User must be registered.
Postconditions:	User's personal information will be saved in their profile
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	1. User will be prompted to enter their personal information. 2. User clicks on the edit profile button and input their personal information and clicks save after finishing 3. System will check for the validity of the information and save the user's personal information in the database.
Alternative Flows:	SEPI1-AF-S1 user decided not to enter personal information: 1. User click on the 'skip' button. 2. User will be brought to the homepage. SEPI1-AF-S2 user enters invalid personal information: 1. User will be prompted to correct the invalid information. 2. Repeat SEPI1-AF-S2 until all information are valid. 3. Valid information will be then saved in the database.
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

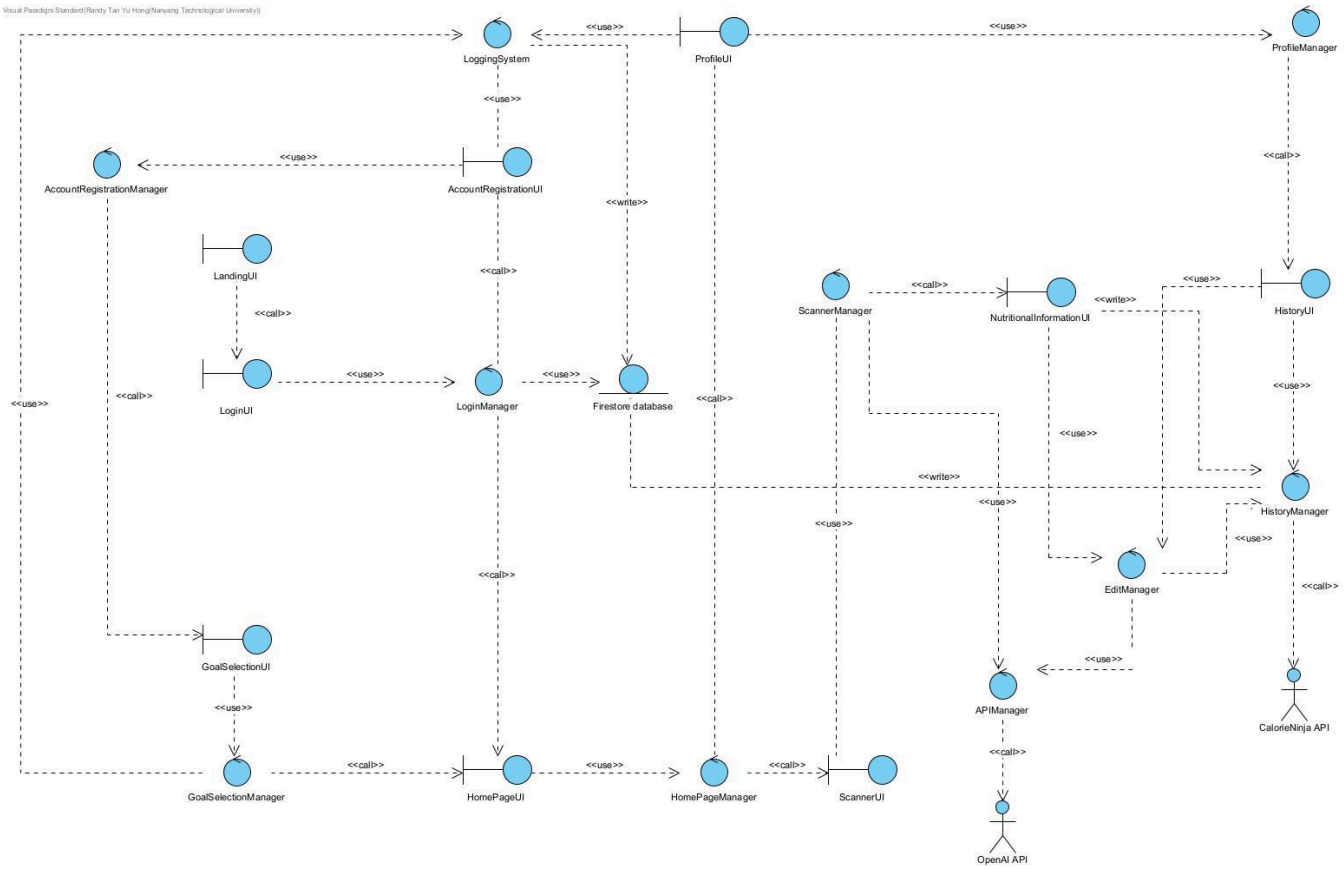
Use Case ID:	LG1		
Use Case Name:	Login		
Created By:	Hao Lun	Last Updated By:	Muhammad Nahid
Date Created:	08/02/2024	Date Last Updated:	09/02/2024

Actor:	User
Description:	User is able to login with their email and password
Preconditions:	User must be registered.
Postconditions:	User will have access to all the functionality of the app.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User will open the app 2. User will enter their email and password 3. System will check against the database if the password is correct. 4. User will be logged in and brought to the homepage.
Alternative Flows:	LG1-AF-S2 if user enters wrong email: System prompts user to re-enter email. 1. User will enter their password. 2. User will be brought to the home page. LG1-AF-S3 if user enters the wrong password: 1. System prompts user to re-enter password 2. User will be brought to homepage
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	User remember their email and password.
Notes and Issues:	Nil

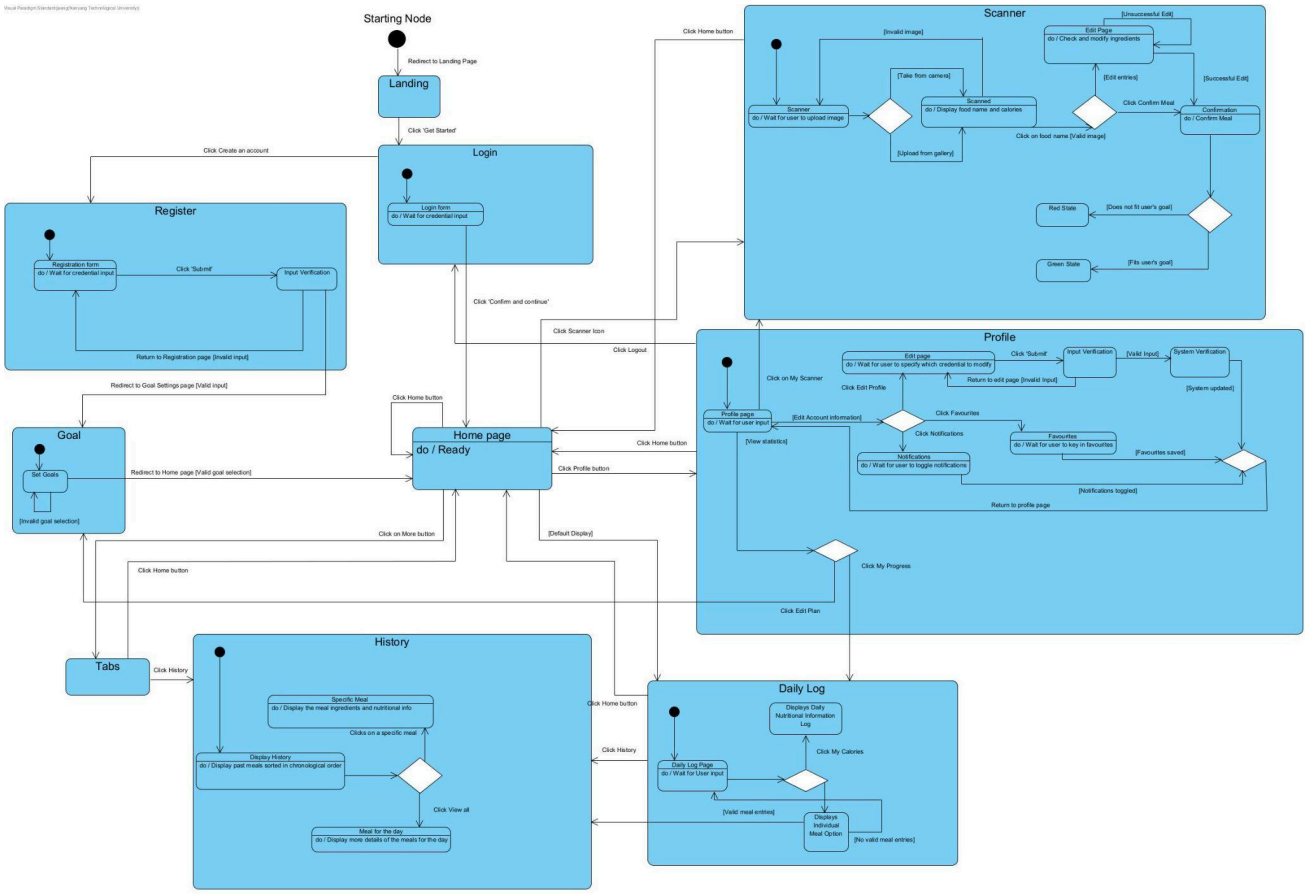
6.4 Class Diagram



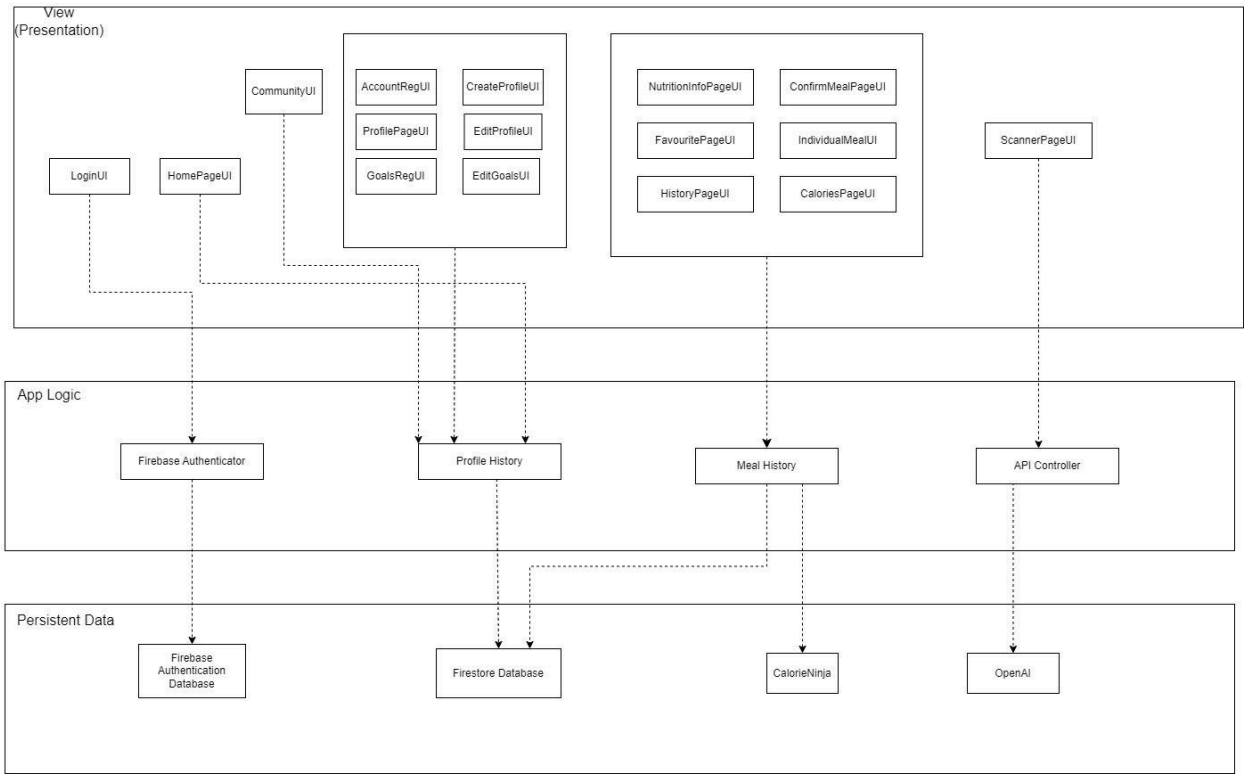
6.5 Conceptual Model



6.6 Dialog Map



6.7 System Architecture



7. Black-Box Testing

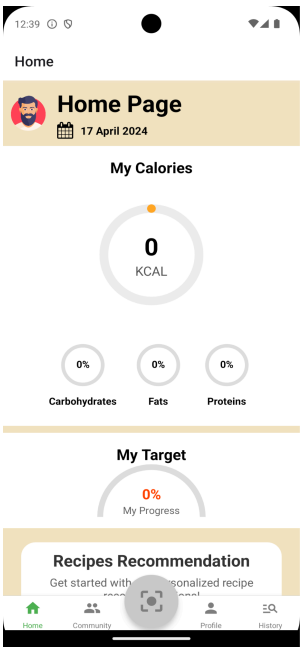
7.1 Login - Equivalence Class Testing

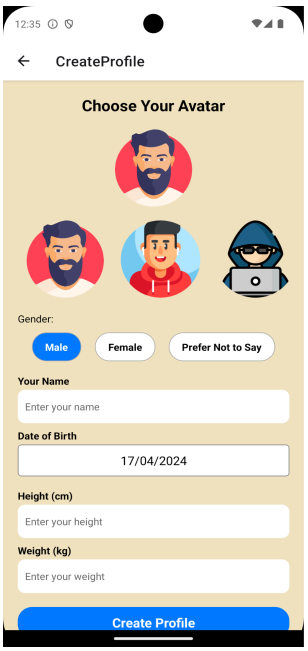
Generic Cases

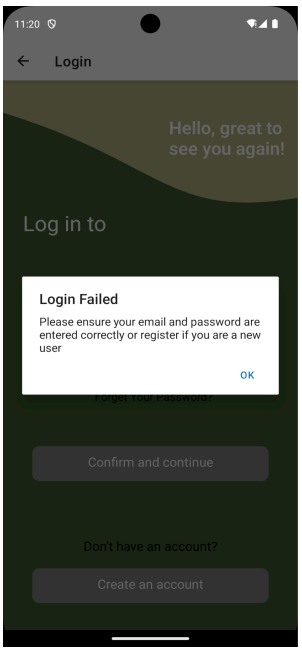
Test Id	Scenario	Expected Result	Actual Result
1	Login with valid email and password	The system displays the home page for user to continue the operation	The system displays the home page for user to continue the operation
2	Login for the first time with valid email and password	The system displays create profile page for user to continue the operations	The system displays create profile page for user to continue the operations
3	Login without valid credentials	The system prompts the user to enter the credentials again or to register if he/she is a new user	The system prompts the user to enter the credentials again
4	Login without filling up the required fields	The system prompts the user to fill up the required fields for logging in	The system prompts the user to fill up the required fields for logging in

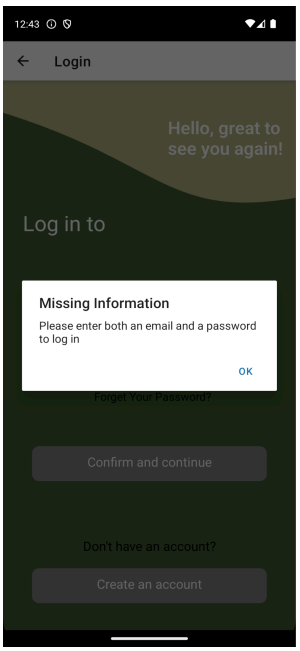
Specific Cases (Combination)

Email	Password	Expected Result	Actual Result
testuser	testpass	Successful login	Successful login
wronguser	testpass	Invalid email/password	Invalid email/password
testuser	wrongpass	Invalid email/password	Invalid email/password
Empty(“”)	testpass	Please fill in all required fields	Please fill in all required fields
testuser	Empty(“”)	Please fill in all required fields	Please fill in all required fields


12:39
Home
Home Page
17 April 2024
My Calories
0
KCAL
0%
Carbohydrates
0%
Fats
0%
Proteins
My Target
0%
My Progress
Recipes Recommendation
Home
Community
Profile
History


12:35
CreateProfile
Choose Your Avatar
Gender:
Male
Female
Prefer Not to Say
Your Name
Enter your name
Date of Birth
17/04/2024
Height (cm)
Enter your height
Weight (kg)
Enter your weight
Create Profile


11:20
Login
Hello, great to see you again!
Log in to
Login Failed
Please ensure your email and password are entered correctly or register if you are a new user
OK
Confirm and continue
Don't have an account?
Create an account


12:43
Login
Hello, great to see you again!
Log in to
Missing Information
Please enter both an email and a password to log in
OK
Confirm and continue
Don't have an account?
Create an account

7.2 Registration - Equivalence Class Testing

Generic Cases

Test Id	Scenario	Expected Result	Actual Result
1	Register with valid account email and password	The system displays the login page for user to continue the operation	The system displays the login page for user to continue the operation
2	Register with weak password	The system prompts the user to enter stronger password	The system prompts the user to enter stronger password
3	Register with incomplete fields	The system prompts the user to fill up the required fields for registration	The system prompts the user to fill up the required fields for registration
4	Register with password mismatch	The system prompts the user re-enter the password	The system prompts the user re-enter the password
5	Register without accepting terms and conditions	The system prompts the user to accept terms and conditions	The system prompts the user to accept terms and conditions

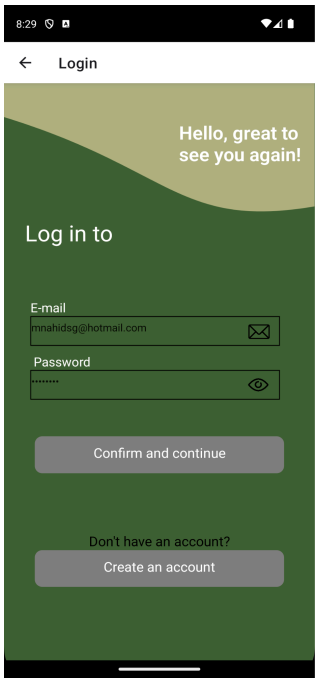
Specific Cases (Email address)

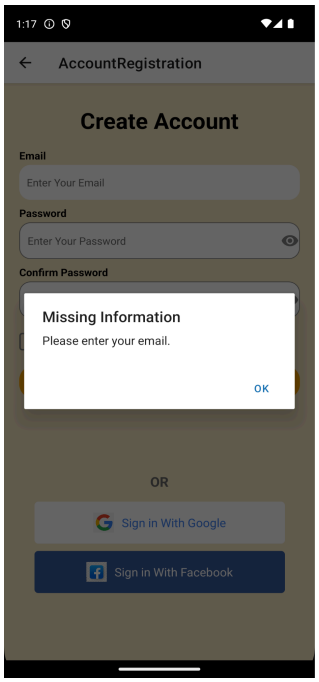
Test Id	Email Address	Expected Result	Actual Result
1	test@gmail.com	Approve	Approve
2	test	Reject	Reject
3	test@asghiuash	Reject	Reject

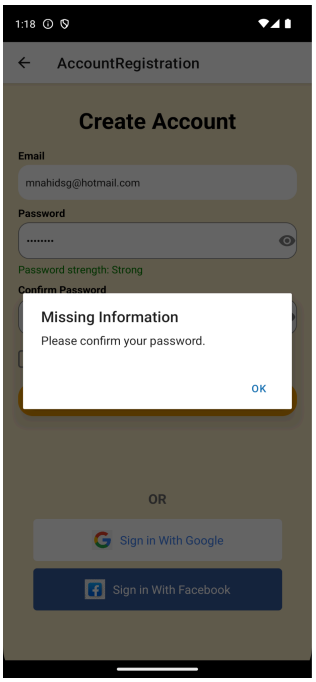
Specific Cases (Combination)

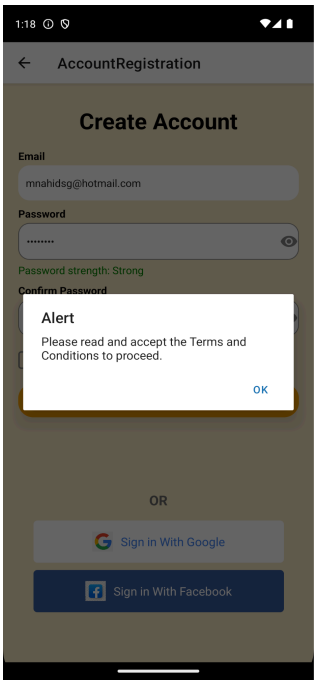
Email Address	Password	Confirm Password	Accept Terms and Conditions	Expected Result	Actual Result
user@gmail.com	Testpass@123	Testpass@123	Accept	Created new user	Created new user
Empty(“”)	Testpass@123	Testpass@123	Accept	Please fill in all required fields	Please fill in all required fields
user@gmail.com	Empty(“”)	Testpass@123	Accept	Please fill in all required fields	Please fill in all required fields

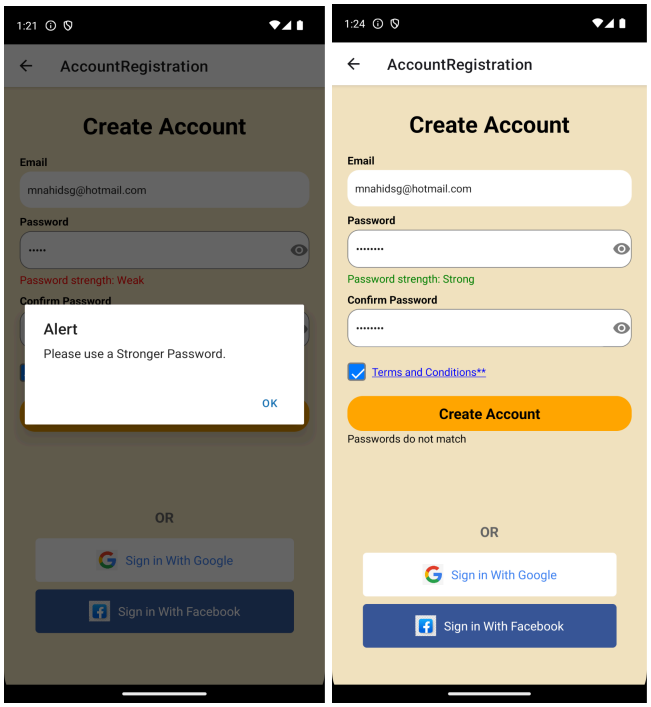
user@gmail.com	Testpass@123	Empty(“”)	Accept	Please fill in all required fields	Please fill in all required fields
existing@gmail.com	Testpass@123	Testpass@123	Accept	Email has been used	Email has been used
user@gmail.com	Testpass@123	Testpass@1234	Accept	Entered passwords do not match	Entered passwords do not match
user@gmail.com	Testpass@1234	Testpass@123	Accept	Entered passwords do not match	Entered passwords do not match
user@gmail.com	Testpass@123	Testpass@123	Did not accept	Please accept the terms and conditions	Please accept the terms and conditions










1:21
AccountRegistration
Create Account
Email
mnahidsg@hotmail.com
Password
Password strength: Weak
Confirm Password
Alert
Please use a Stronger Password.
OK
OR
Sign in With Google
Sign in With Facebook
1:24
AccountRegistration
Create Account
Email
mnahidsg@hotmail.com
Password
Password strength: Strong
Confirm Password
Terms and Conditions**
Create Account
Passwords do not match
OR
Sign in With Google
Sign in With Facebook

7.3 Profile Creation - Equivalence Class Testing & Boundary Value Testing

Generic Cases

Test Id	Scenario	Expected Result	Actual Result
1	Create profile with valid gender, name, date of birth (DOB), height and weight	The system displays the goals registration page for user to continue the operation	The system displays the goals registration page for user to continue the operation
2	Register with incomplete fields	The system prompts the user to fill up the required fields for registration	The system prompts the user to fill up the required fields for registration
3	User tries to register with out of bonds DOB	The system prevents the user from selecting an out of bounds DOB	The system prevents the user from selecting an out of bounds DOB
4	Register with invalid height	The system prompts the user re-enter the height	The system prompts the user re-enter the height
5	Register with invalid weight	The system prompts the user re-enter the weight	The system prompts the user re-enter the weight
6	Register without selecting gender	The system prompts the user to select a gender	The system prompts the user to select a gender

Specific Cases (Date of Birth)

Test Id	Test Input Date of Birth	Test Case	Expected Result	Actual Result
1	01/01/1901	Lower boundary	Approve	Approve
2	Today's date	Upper boundary	Approve	Approve
3	31/12/1899	Just below lower boundary	Reject	Reject
4	Tomorrow's date	Just above upper Boundary	Reject	Reject

Specific Cases (Height)

Test Id	Test Input Height (cm)	Test Case	Expected Result	Actual Result
1	1	Lower boundary	Approve	Approve

2	300	Upper boundary	Approve	Approve
3	0	Just below lower boundary	Reject	Reject
4	301	Just above upper Boundary	Reject	Reject

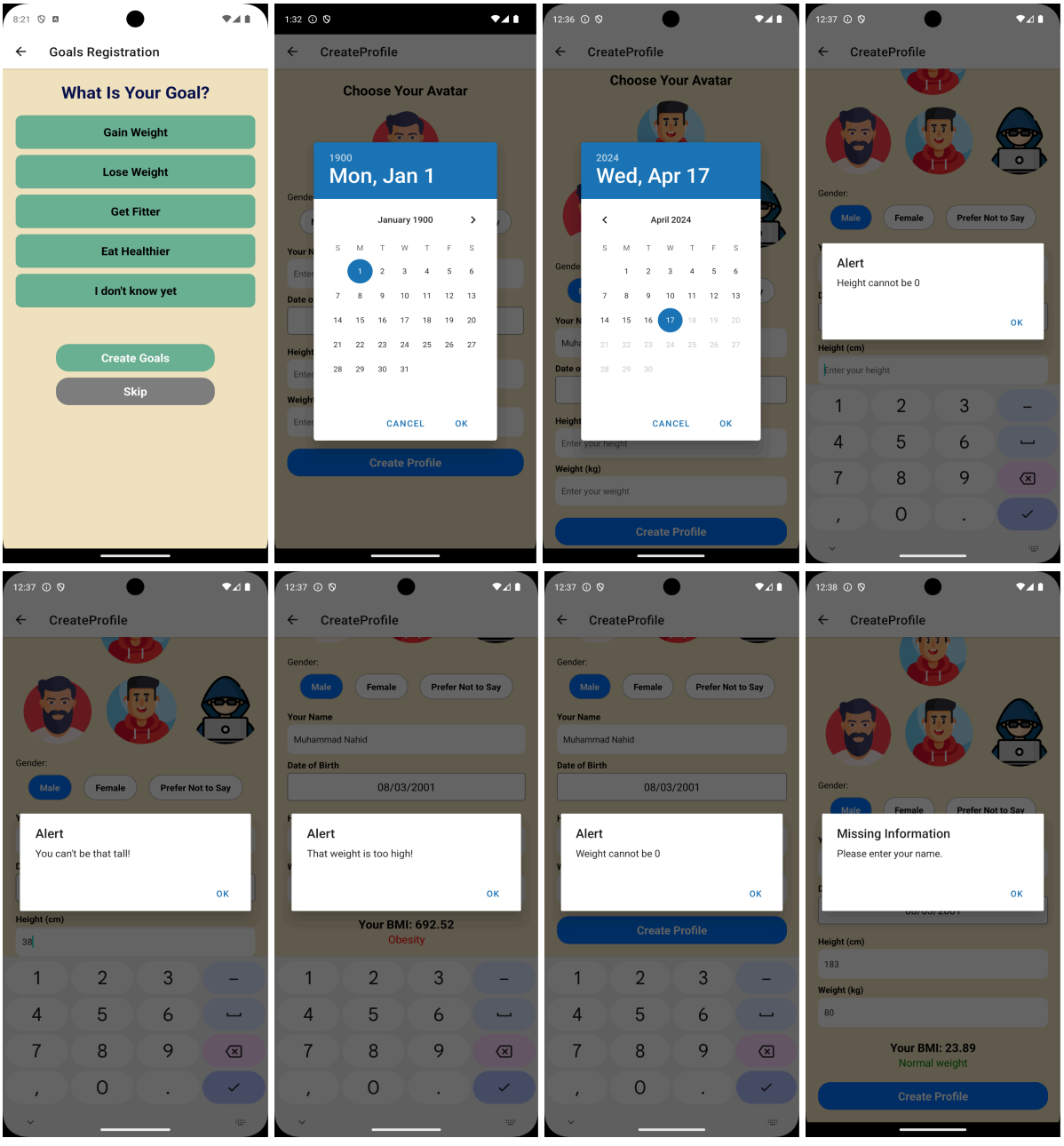
Specific Cases (Weight)

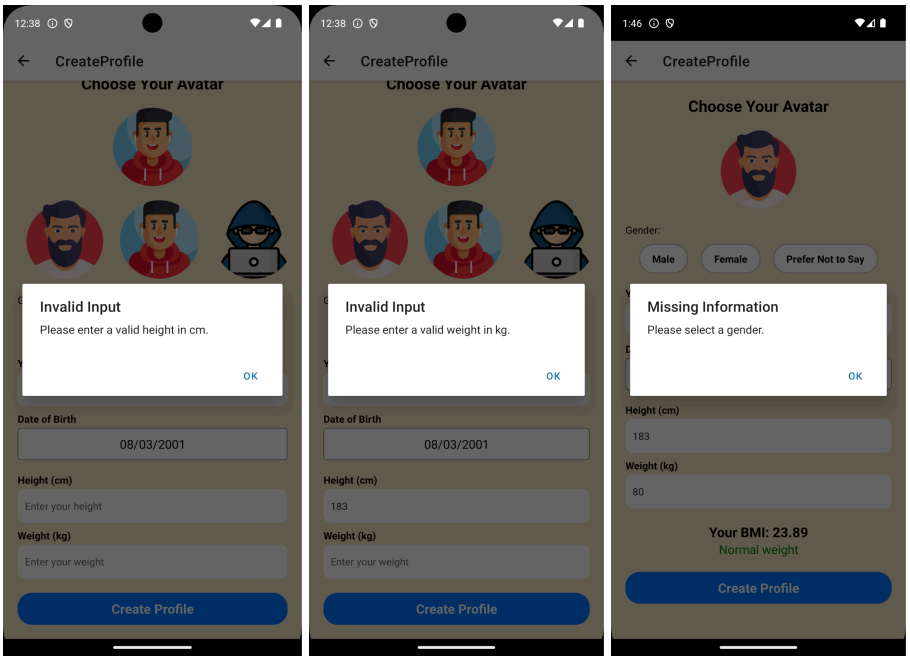
Test Id	Test Input Weight (kg)	Test Case	Expected Result	Actual Result
1	1	Lower boundary	Approve	Approve
2	1000	Upper boundary	Approve	Approve
3	0	Just below lower boundary	Reject	Reject
4	1001	Just above upper Boundary	Reject	Reject

Specific Cases (Combination)

Name	Date Of Birth	Height (cm)	Weight (kg)	Gender	Expected Result	Actual Result
John Doe	01/01/1998	183	80	Selected	Created new profile	Created new profile
Empty("")	01/01/1998	183	80	Selected	Please fill in all required fields	Please fill in all required fields
John Doe	Empty("")	183	80	Selected	Please fill in all required fields	Please fill in all required fields
John Doe	01/01/1998	Empty("")	80	Selected	Please fill in all required fields	Please fill in all required fields
John Doe	01/01/1998	183	Empty("")	Selected	Please fill in all required fields	Please fill in all required fields
John Doe	01/01/1998	183	80	Unselected	Please fill in all required fields	Please fill in all required fields

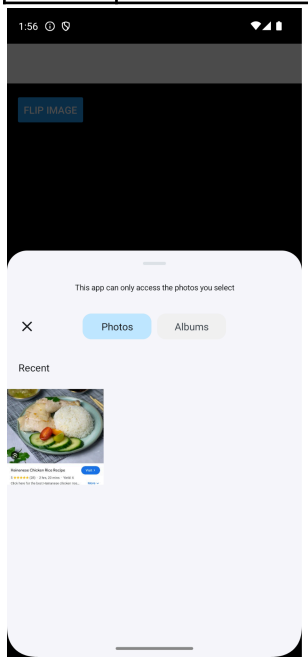
John Doe	31/31/1900	183	80	Selected	Date of birth is out of bounds	Date of birth is out of bounds
John Doe	01/01/1998	0	80	Selected	Please enter valid height	Please enter valid height
John Doe	01/01/1998	183	0	Selected	Please enter valid weight	Please enter valid weight

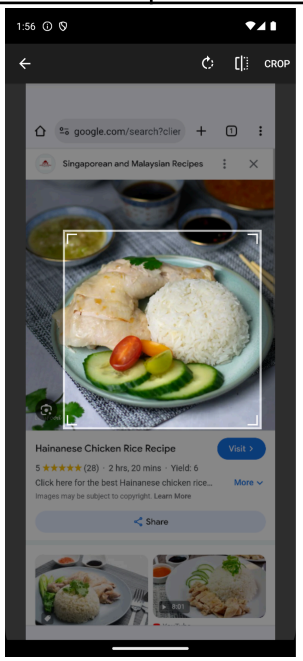


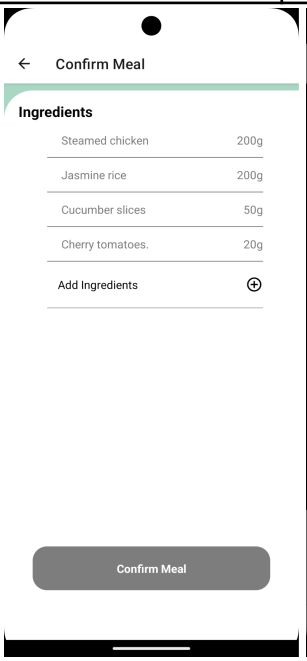

12:38
CreateProfile
Choose Your Avatar
Invalid Input
Please enter a valid height in cm.
OK
Date of Birth
08/03/2001
Height (cm)
Enter your height
Weight (kg)
Enter your weight
Create Profile
12:38
CreateProfile
Choose Your Avatar
Invalid Input
Please enter a valid weight in kg.
OK
Date of Birth
08/03/2001
Height (cm)
183
Weight (kg)
Enter your weight
Create Profile
1:46
CreateProfile
Choose Your Avatar
Gender:
Male
Female
Prefer Not to Say
Missing Information
Please select a gender.
OK
Height (cm)
183
Weight (kg)
80
Your BMI: 23.89
Normal weight
Create Profile

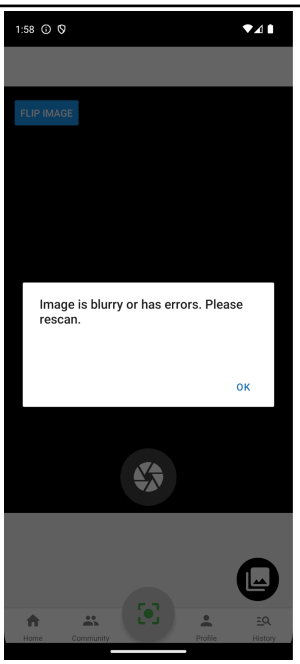
7.4 Image Detection - Equivalence Class Testing

Test Id	Scenario	Expected Result	Actual Result
1	Upload photo which image recognition API can detect	The system displays confirm meal page for user to continue the operations. The confirm meal page contains the ingredient list and their corresponding weights.	The system displays confirm meal page for user to continue the operations. The confirm meal page contains the ingredient list and their corresponding weights.
2	Upload photo which image recognition API cannot detect	The system prompts the user to retake or re upload photo	The system prompts the user to retake or re upload photo



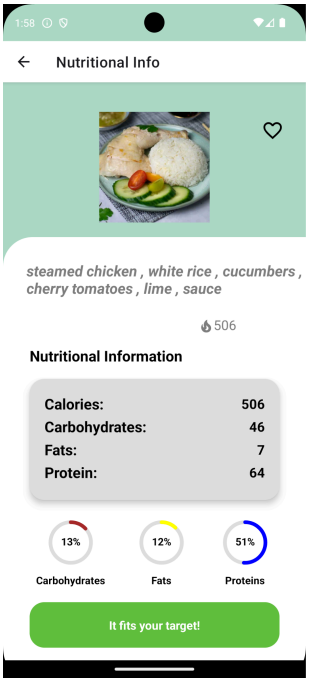






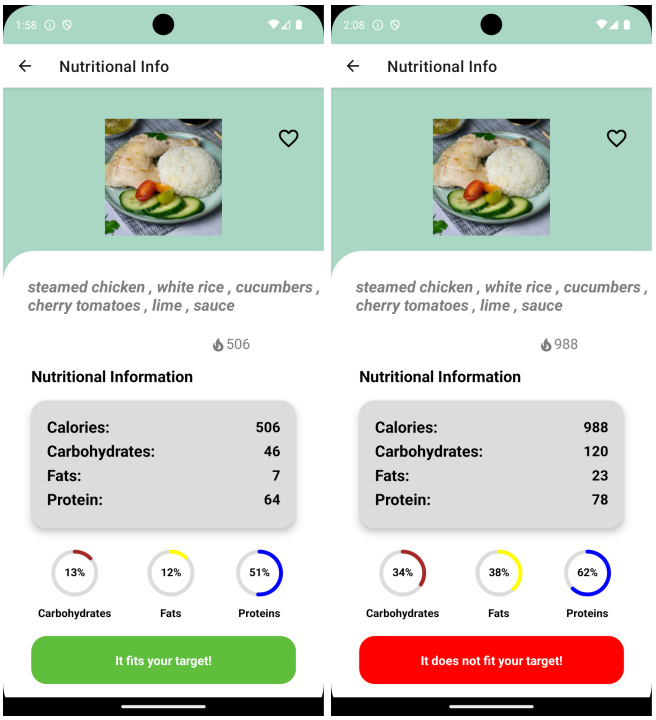
7.5 Ingredient Detection - Equivalence Class Testing

Test Id	Scenario	Expected Result	Actual Result
1	Successful API call of ingredients to Calorie Ninja	The system displays the nutritional info page for the user to continue the operations. The nutritional info page contains macros of each ingredient.	The system displays the nutritional info page for the user to continue the operations. The nutritional info page contains macros of each ingredient.
2	Unsuccessful API call of ingredients to Calorie Ninja	The system prompts the user to re enter the ingredients.	The system prompts the user to re enter the ingredients.



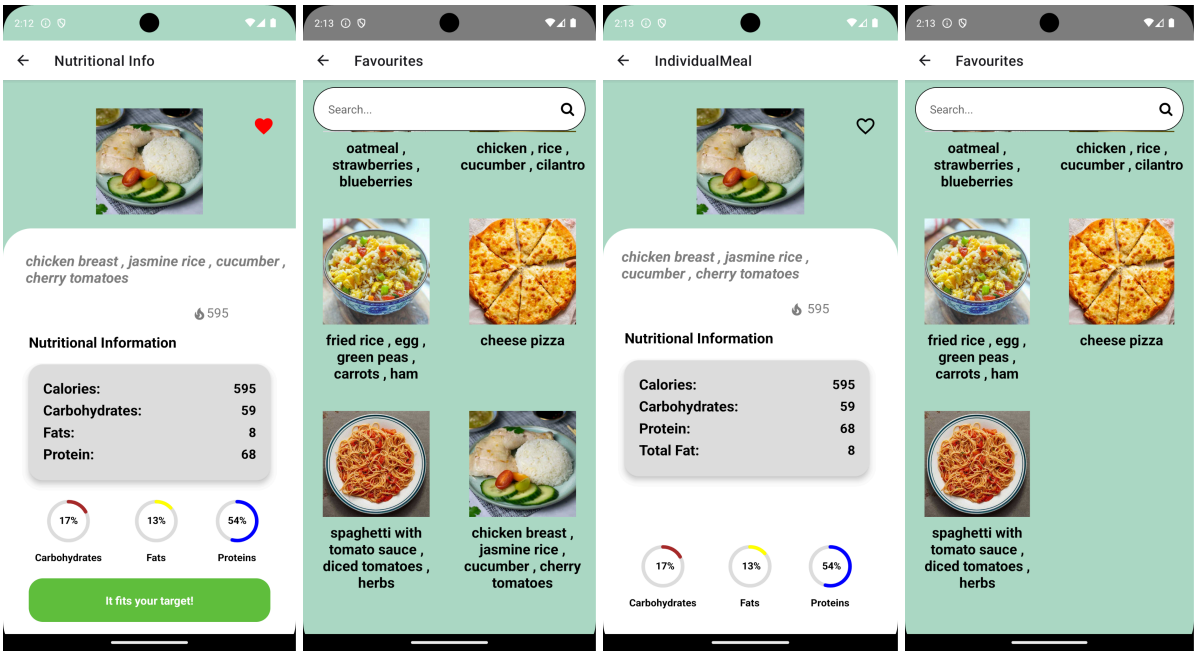
7.6 Meal Confirmation - Equivalence Class Testing

Test Id	Scenario	Expected Result	Actual Result
1	Calories fit dietary goals	The system displays a green button that says “It fits your target!”. Upon clicking, the system saves the meal into the database and brings the user to the scanner page.	The system displays a green button that says “It fits your target!”. Upon clicking, the system saves the meal into the database and brings the user to the scanner page.
2	Calories do not fit dietary goals	The system displays a red button that says “It does not fit your target!”. Upon clicking, the system saves the meal into the database and brings the user to the scanner page.	The system displays a red button that says “It does not fit your target!”. Upon clicking, the system saves the meal into the database and brings the user to the scanner page.



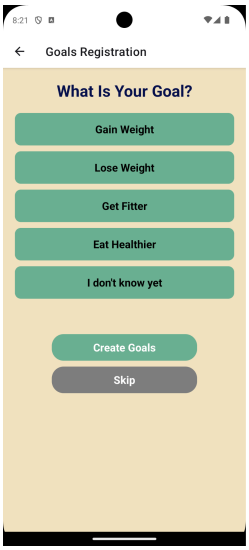
7.7 Favourites - Equivalence Class Testing

Test Id	Scenario	Expected Result	Actual Result
1	Selecting heart icon on an individual meal	The system changes the database attribute “favourite” to “True”. The meal gets displayed on the Favourites page.	The system changes the database attribute “favourite” to “True”. The meal gets displayed on the Favourites page.
2	Unselecting heart icon on an individual meal	The system changes the database attribute “favourite” to “False”. The meal does not get displayed on the Favourites page.	The system changes the database attribute “favourite” to “False”. The meal does not get displayed on the Favourites page.



7.8 Goals Creation - Equivalence Class Testing

Test Id	Scenario	Expected Result	Actual Result
1	Selected a goal: 'Gain Weight', 'Lose Weight', 'Get Fitter', 'Eat Healthier'	The system updates the database with user's dietary goal requirements.	The system updates the database with user's dietary goal requirements.
2	Selected 'skip' or 'I don't know yet'	The system updates the database with generic dietary goal requirements.	The system updates the database with generic dietary goal requirements.
3	Unsuccessful update of dietary goals in database	The system displays error message and prompts user to reselect a goal.	The system displays error message and prompts user to reselect a goal.



7.9 Edit Profile - Equivalence Class Testing & Boundary Value Testing

Generic Cases

Test Id	Scenario	Expected Result	Actual Result
1	Successful updating of profile.	The system displays the profile page with updated details.	The system displays the profile page with updated details.
2	Unsuccessful updating of profile.	The system displays an error message saying ‘profile update unsuccessful’.	The system displays an error message saying ‘profile update unsuccessful’.
3	Empty or Invalid update of profile details	The system displays an error message and prompts the user to reenter the details.	The system displays an error message and prompts the user to reenter the details.

Specific Cases (DOB)

Test Id	Test Input Date of Birth	Test Case	Expected Result	Actual Result
1	01/01/1998	Within Range	Approve	Approve
2	01/01/1900	Lower Boundary	Reject	Reject
3	01/01/2025	Upper Boundary	Reject	Reject

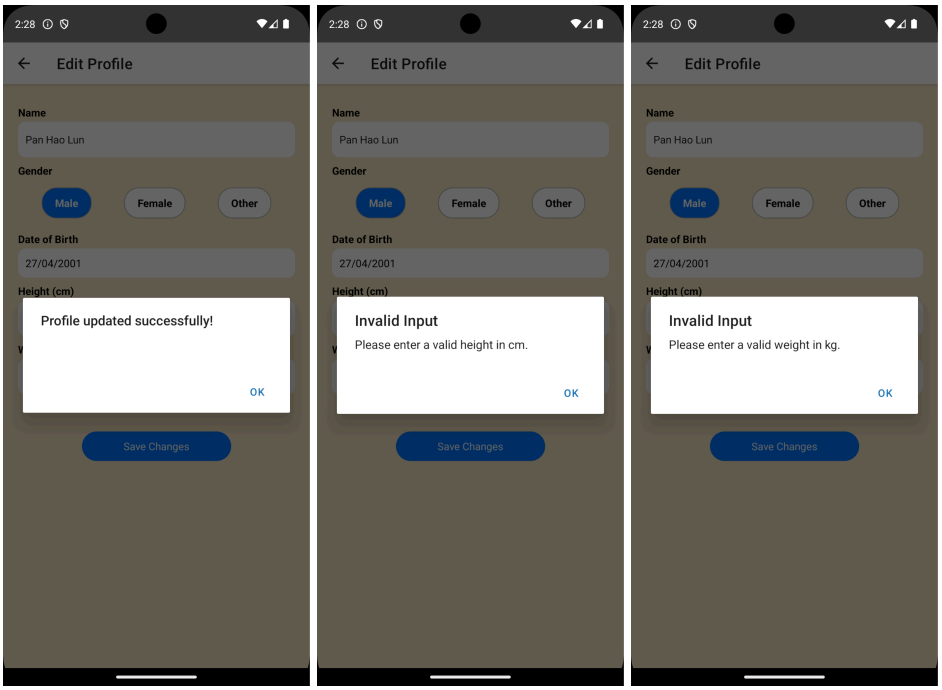
Specific Cases (Height)

Test Id	Test Input Height (cm)	Test Case	Expected Result	Actual Result
1	183	Within Range	Approve	Approve
2	300	Upper Boundary	Reject	Reject
3	0	Lower Boundary	Reject	Reject
4	-1	Lower Boundary	Reject	Reject

Specific Cases (Weight)

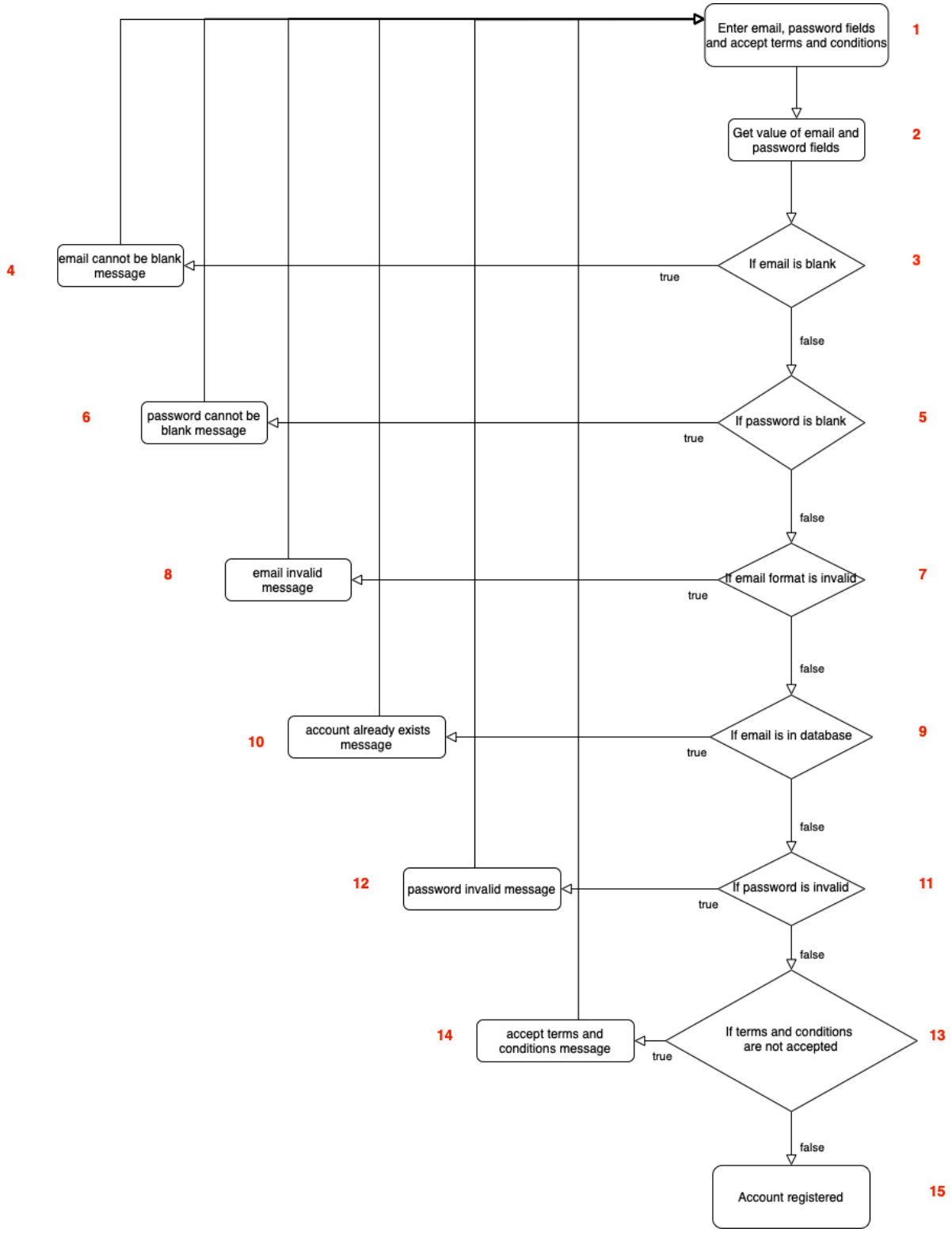
Test Id	Test Input Weight (kg)	Test Case	Expected Result	Actual Result
1	80	Within Range	Approve	Approve

2	1000	Upper Boundary	Reject	Reject
3	0	Lower Boundary	Reject	Reject
4	-1	Lower Boundary	Reject	Reject



8. White-Box Testing

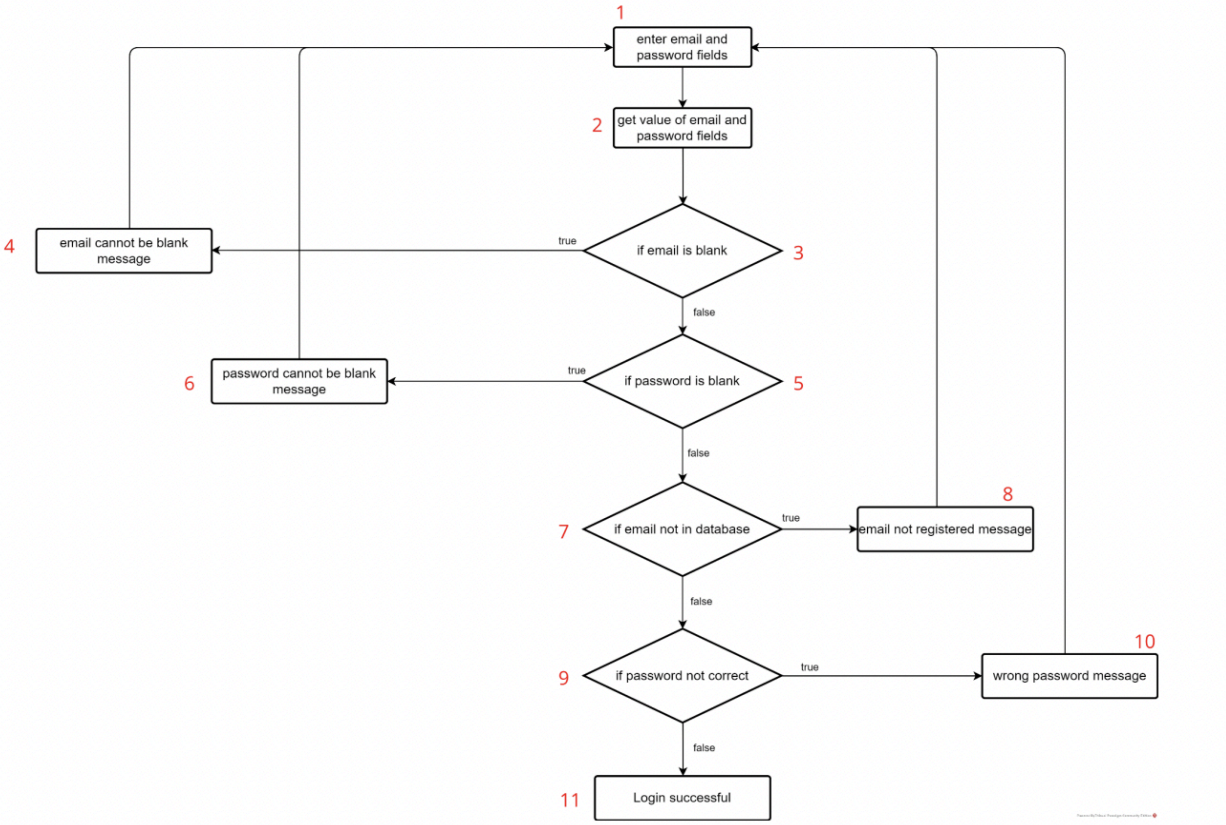
8.1 Registration



Cyclomatic Complexity = 6 (Binary Decision Points) + 1 = 7
We need 7 Basis Paths.

Path no.	Path	Email	Password	Email format invalid	Email registered	Password invalid	Terms and conditions not accepted
1 (baseline)	1,2,3,5,7,9,11,13,15	johndoe123@gmail.com	J0hn321!	No	No	No	No
2	1,2,3,4,1,2,3,5,7,9,11,13,15	(Blank)	–	–	–	–	–
3	1,2,3,5,6,1,2,3,5,7,9,11,13,15	johndoe123@gmail.com	(Blank)	–	–	–	–
4	1,2,3,5,7,8,1,2,3,5,7,9,11,13,15	johndoe123@gmail.com	J0hn321!	Yes	–	–	–
5	1,2,3,5,7,9,10,1,2,3,5,7,9,11,13,15	johndoe123@gmail.com	J0hn321!	No	Yes	–	–
6	1,2,3,5,7,9,11,12,1,2,3,5,7,9,11,13,15	johndoe123@gmail.com	J0hn321!	No	No	Yes	–
7	1,2,3,5,7,9,11,13,14,1,2,3,5,7,9,11,13,15	johndoe123@gmail.com	J0hn321!	No	No	No	Yes

8.2 Login

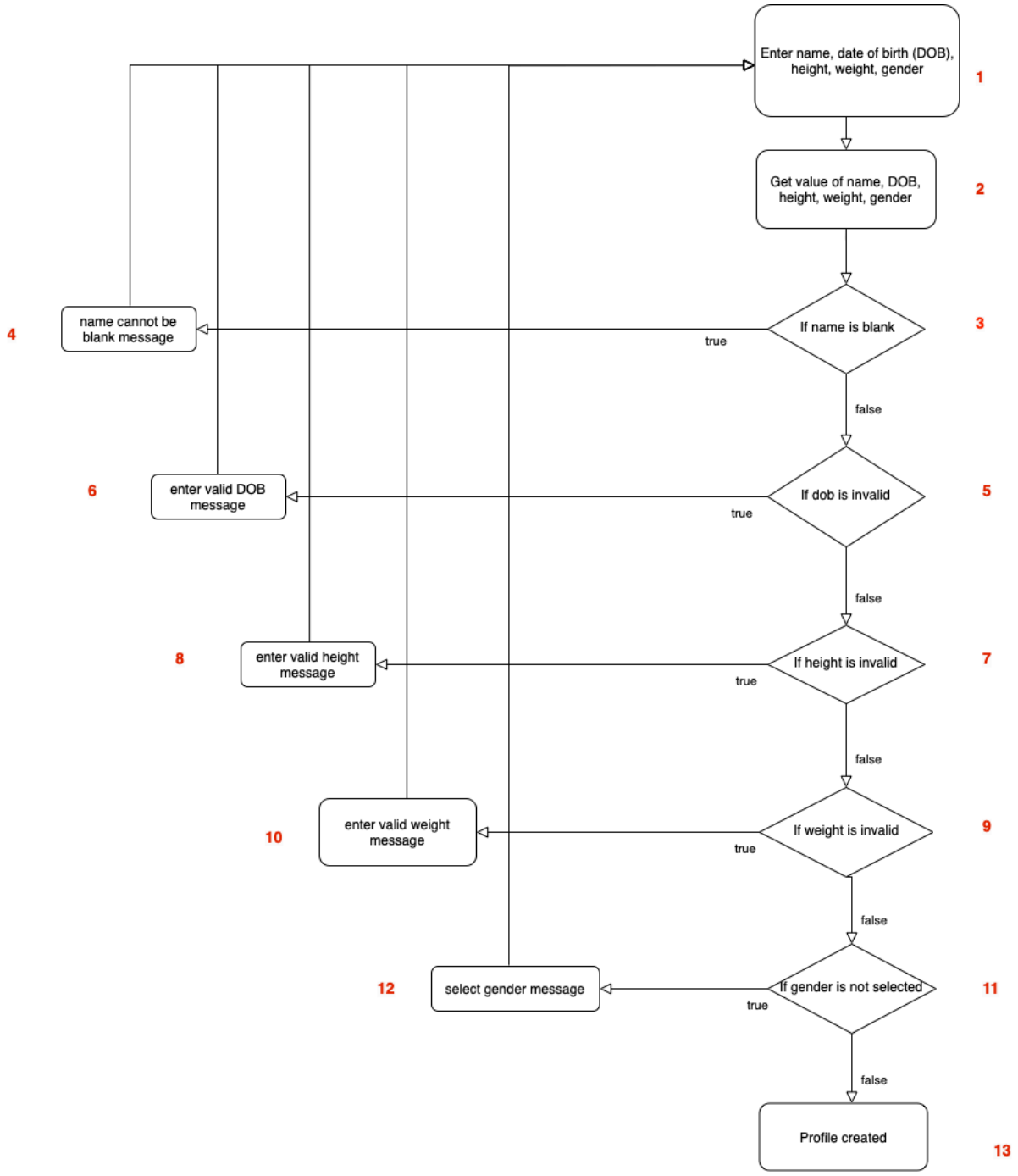


Cyclomatic Complexity = 4 (Binary Decision Points) + 1 = 5
We need 5 basis paths.

Path no.	Path	Email	Password	Email registered	Password match
1 (baseline)	1,2,3,5,7,9,11	johndoe123 @gmail.com	J0hn321!	Yes	Yes
2	1,2,3,4,1,2,3, 5,7,9,11	(Blank)	–	–	–
3	1,2,3,5,6,1,2, 3,5,7,9,11	johndoe123 @gmail.com	(Blank)	–	–
4	1,2,3,5,7,8,1, 2,3,5,7,9,11	johndoe123 @gmail.com	J0hn321!	No	–

5	1,2,3,5,7,9,10 ,1, 2,3,5,7,9,11	johndoe123@gmail.com	J0hn321!	No	No

8.3 Profile Creation

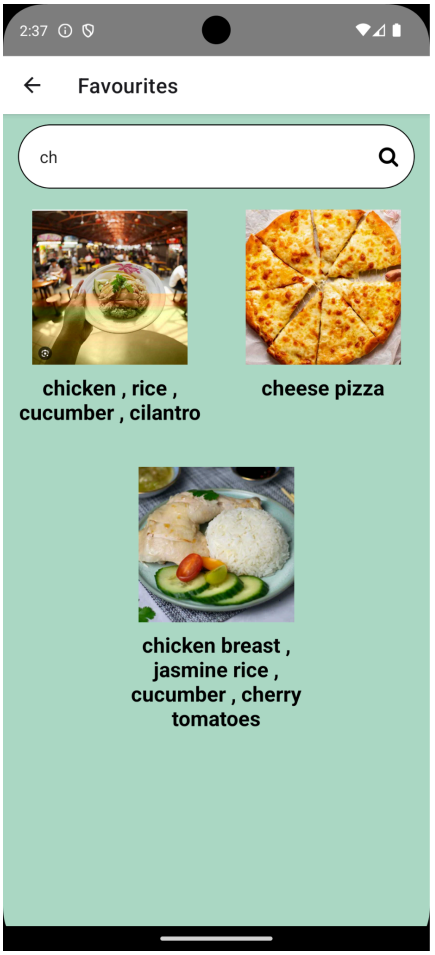


Cyclomatic Complexity = 5 (Binary Decision Points) + 1 = 6
We need 6 basis paths.

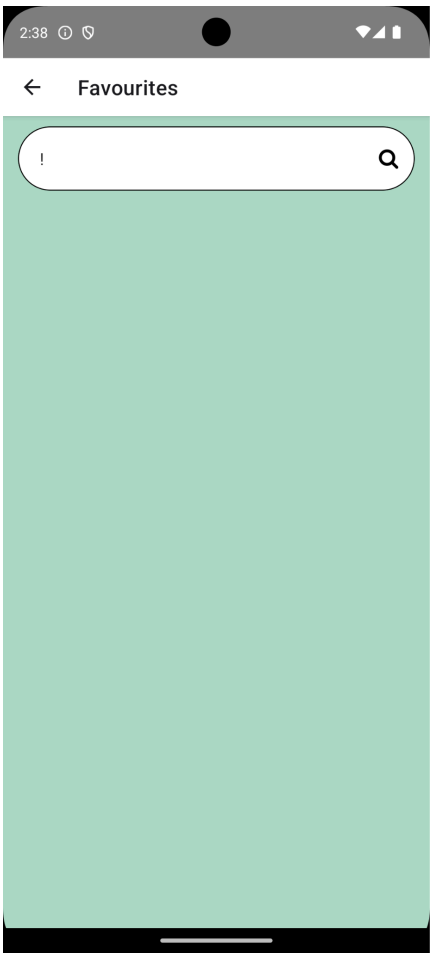
Path no.	Path	Name	Date of birth invalid	Height invalid	Weight invalid	Gender unselected
1 (baseline)	1,2,3,5,7, 9,11,13	John Doe	No	No	No	No
2	1,2,3,4,1, 2,3, 5,7,9,11, 13	(Blank)	–	–	–	–
3	1,2,3,5,6, 1,2, 3,5,7,9,1 1,13	John Doe	Yes	–	–	–
4	1,2,3,5,7, 8,1, 2,3,5,7,9, 11,13	John Doe	No	Yes	–	–
5	1,2,3,5,7, 9,10,1, 2,3,5,7,9, 11,13	John Doe	No	No	Yes	–
6	1,2,3,5,7, 9,11,12,1 2,3,5,7,9, 11,13	John Doe	No	No	No	Yes

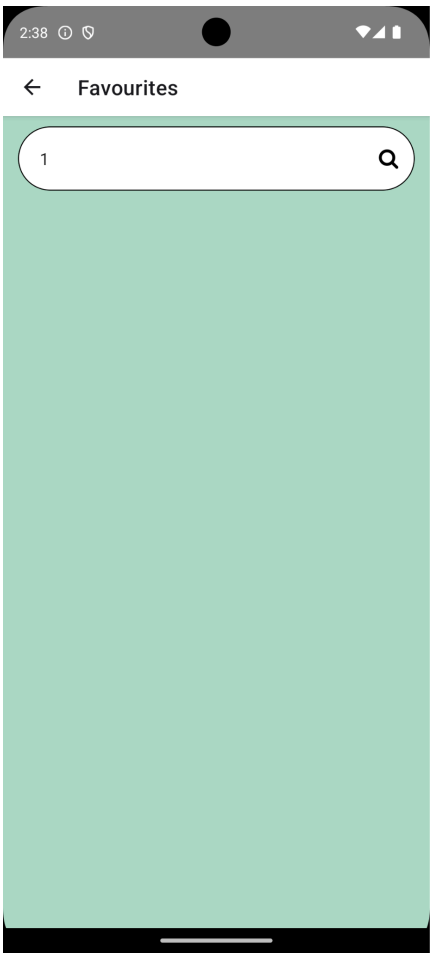
9. Other Testing

9.1 Search Test Cases

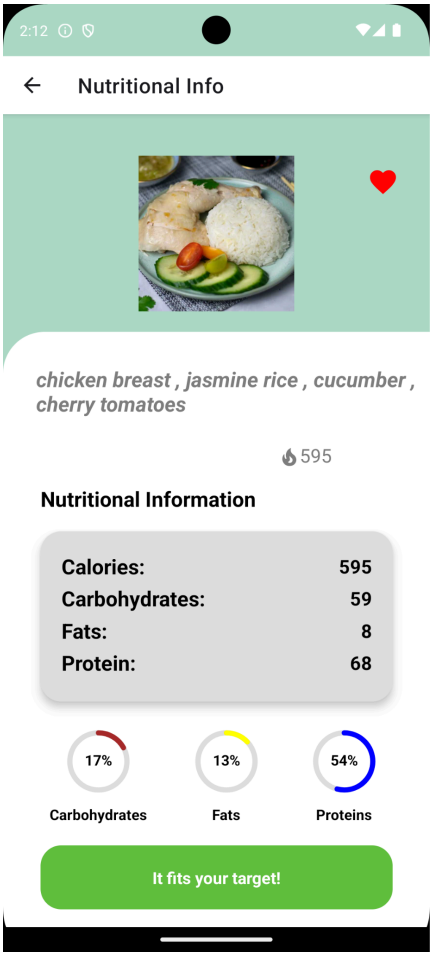
Test Input	Expected Output	Actual Output
User types “ch”	Only meals with names having the phrase “ch” will be shown	

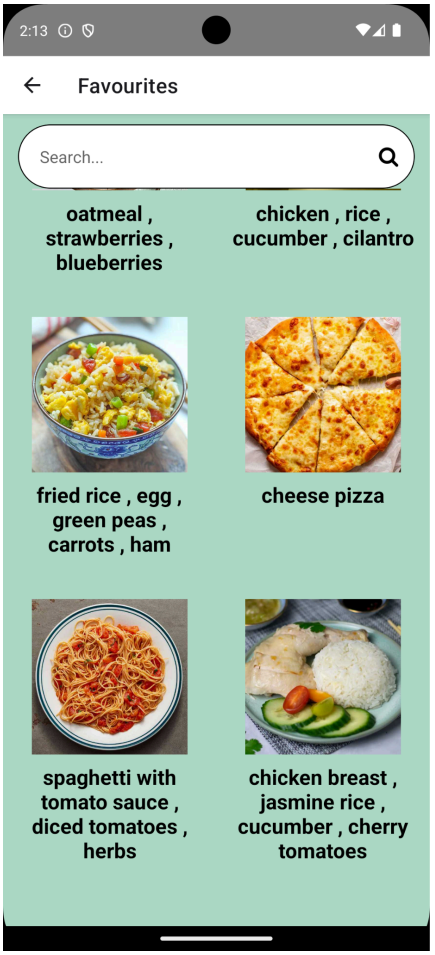
User types “chicken breast”	Only meals with names having the phrase “chicken breast” will be shown	

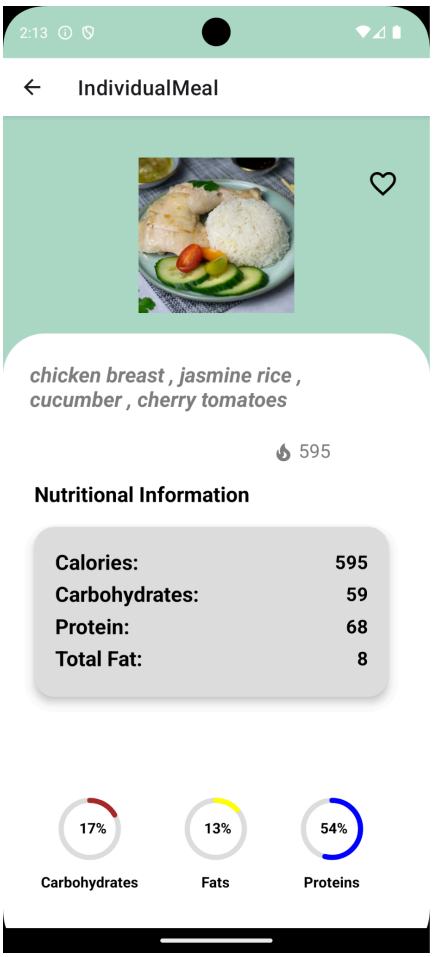
User types in special characters such as “!”	No meals will be shown to the user.	

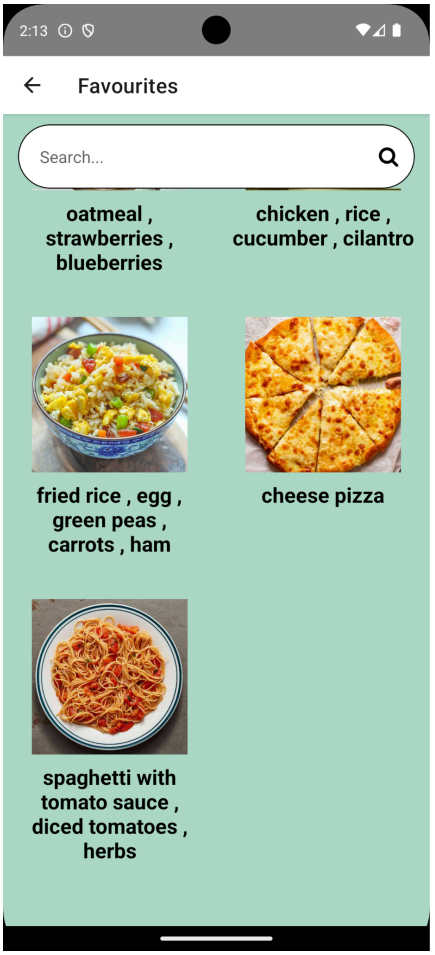
User types in numbers such as “1”	No meals will be shown to the user.	

9.2 Favourites Test Cases

Test Input	Expected Output	Actual Output
User ‘favourites’ a meal by clicking the ‘heart’ button.	Meal gets added to favourites list in Favourites page	

		 2:13 Favourites Search... oatmeal , strawberries , blueberries chicken , rice , cucumber , cilantro fried rice , egg , green peas , carrots , ham cheese pizza spaghetti with tomato sauce , diced tomatoes , herbs chicken breast , jasmine rice , cucumber , cherry tomatoes

User ‘un-favourites’ a meal by unclicking the red heart button	The meal gets removed from the favourites list in the Favourites page	

		 2:13 Favourites Search... oatmeal , strawberries , blueberries chicken , rice , cucumber , cilantro fried rice , egg , green peas , carrots , ham cheese pizza spaghetti with tomato sauce , diced tomatoes , herbs