****

**Cairo University**

**Faculty of Computers and Artificial Intelligence**

**Department of Computer Science**

**Nutrients Tracker App**

****

**Supervised by**

**Pros. Khaled Tawfik Wasif**

**TA. Randa Elbehery**

**Implemented by:**

|  |  |
| --- | --- |
| **ID** | **Name** |
| **20170058** | **Alaa Farouk Abd El-Majied** |
| **20170370** | **Nada Mohamed Mahmoud** |
| **20170326** | **Naira Magdy Mohamed** |
| **20170359** | **Aya Amr Mohamed** |

**Graduation Project**

**Academic Year 2020-2021**

**Final Documentation**

**Table of Contents**

[**Chapter 1: Introduction** 7](#_Toc78095587)

[**1.1. Motivation** 7](#_Toc78095588)

[**1.2. Problem definition** 7](#_Toc78095589)

[**1.3. Project objective** 8](#_Toc78095590)

[**1.4. Gantt chart** 8](#_Toc78095591)

[ Phase one 8](#_Toc78095592)

[ Phase two 9](#_Toc78095593)

[**1.5. Project development methodology** 9](#_Toc78095594)

[**1.6. The used tools in the project** 11](#_Toc78095595)

[**1.7 report organization** 12](#_Toc78095596)

[**Chapter 2: Related work** 13](#_Toc78095597)

[**2.1 MyFitnessPal** 13](#_Toc78095598)

[**2.2 Lose It! – Calorie Counter** 14](#_Toc78095599)

[**Chapter 3: System Analysis** 15](#_Toc78095600)

[**3.1 Project Specification** 15](#_Toc78095601)

[3.1.1 Functional Requirements 15](#_Toc78095602)

[3.1.2 Non-Functional Requirements 19](#_Toc78095603)

[**3.2 Use Case diagrams** 20](#_Toc78095604)

[3.2.1 Use case diagram 20](#_Toc78095605)

[3.2.2 Sample Use-Cases 21](#_Toc78095606)

[**Chapter 4: System Design** 29](#_Toc78095607)

[**4.1 system component diagram** 29](#_Toc78095608)

[**4.2 Class Diagram** 30](#_Toc78095609)

[**4.3 Sequence Diagram** 31](#_Toc78095610)

[4.3.1. Signup 31](#_Toc78095611)

[4.3.3 Add Recipe 32](#_Toc78095612)

[4.3.5 Macro tracker 34](#_Toc78095613)

[4.3.6 Recipe nutrients 34](#_Toc78095614)

[4.3.7 Search user 35](#_Toc78095615)

[**4.5 System GUI** 37](#_Toc78095616)

[ Cover page 37](#_Toc78095617)

[ Signup-login 37](#_Toc78095618)

[ Access camera 38](#_Toc78095619)

[ My profile-Edit profile 38](#_Toc78095620)

[ Users’ profile 39](#_Toc78095621)

[ Search user 39](#_Toc78095622)

[ Detail Recipe – Favorite list 40](#_Toc78095623)

[ Add recipe 40](#_Toc78095624)

[ Nutrition facts 41](#_Toc78095625)

[ Macro Tracker – Tips 41](#_Toc78095626)

[ Scan Camera-Ingredients-Nutrition Facts 42](#_Toc78095627)

[ Search Meals-Ingredients-Nutrition Facts 42](#_Toc78095628)

[**Chapter 5: Implementation and Testing** 43](#_Toc78095629)

[**5.1 Implementation** 43](#_Toc78095630)

[5.1.1 Database Management System 43](#_Toc78095631)

[5.1.2 Session manager 45](#_Toc78095632)

[5.1.3 Request to upload image 46](#_Toc78095633)

[5.1.3 Food Recognizer Model 48](#_Toc78095634)

[**5.2 Testing** 50](#_Toc78095636)

[Test case 1: 50](#_Toc78095637)

[Test case 2: 51](#_Toc78095638)

[Test case 3: 52](#_Toc78095639)

[Test case 4: 53](#_Toc78095640)

[Test case 5: 55](#_Toc78095641)

[Test case 6: 56](#_Toc78095642)

[**References** 58](#_Toc78095643)

**List of figures**

[Figure 1:Gantt chart first chase 7](#_Toc78098798)

[Figure 2: Gantt chart second phase 8](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098799)

[Figure 3: waterfall sequence of events 9](#_Toc78098800)

[Figure 4: Use case diagram 20](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098801)

[Figure 5: componentt diagram 29](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098802)

[Figure 6: Class diagram 30](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098803)

[Figure 7: Signup sequence diagram 31](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098804)

[Figure 8: Login sequence diagram 32](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098805)

[Figure 9: Add Recipe sequence diagram 32](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098806)

[Figure 10: Edit profile sequence diagram 33](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098807)

[Figure 11: Macro tracker sequence diagram 34](#_Toc78098808)

[Figure 12: Recipe nutrients sequence diagram 34](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098809)

[Figure 13: Search user Sequence diagram 35](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098810)

[Figure 14: Search recipe Sequence diagram 35](file:///C:\Users\dell\Downloads\FinalIsa%20(4).docx#_Toc78098811)

[Figure 15: Entity Relationship Diagram 36](#_Toc78098812)

**List of tables**

[Table 1: nonfunctional requirements 19](#_Toc78091888)

[Table 2: Sign up use case table 21](#_Toc78091889)

[Table 3: Log in use case table 22](#_Toc78091890)

[Table 4: Search user use case table 23](#_Toc78091891)

[Table 5 : Edit profile use case table 24](#_Toc78091892)

[Table 6: Scan food use case table 26](#_Toc78091893)

[Table 7: Add recipe use case table 27](#_Toc78091894)

[Table 8: Add to favorite list use case table 28](#_Toc78091895)

# 

# **Chapter 1: Introduction**

This is Food Image Recognition App: a smart camera app that uses deep learning to track nutrition from food images. This App is a promising tool for the real-time identification of food and for providing nutrition guidance. The app must provide with important nutrition-related information (e.g., calories) for each food component detected by the CNN-based model.

# **1.1. Motivation**

Calorie monitoring helps many people lose weight and maintain a healthy weight in the long term. By using an easy and simple Calorie counter app, we can make logging food and activity quick and easy. It also can give an insight into lifestyle habits, nutrient quality and how often we are meeting our daily goals. Moreover, this app can replace the old-school paper and pencil method of logging calories, which can be difficult to maintain.

Downloading a nutrition app is easy, but finding one that accurately track calories by recognizing scanned food images, tracking its nutrients, saving its recipe and giving nutritional tips is a tougher task. That’s why we decided to make our own app.

## **1.2. Problem definition**

With the summer months approaching, there can be external pressures to get fit. Tracking your food and calories helps you to lose more weight and keep the weight off in the long run. It can also help people who are trying to gain weight or can help in making a specific change to a certain diet, such as eating more protein or fewer carbs. Using an app to track your calories can be a healthy eating pattern depending on the individual and their needs and intentions; it can also be a behavior or symptom of those who suffer from eating disorders, obsessive compulsive disorder, or even anxiety.

## **1.3. Project objective**

Providing an app that makes it easier for the user to recognize food ingredients that may cause him discomfort or have any possible negative side effects. Also, by supporting the user with written recipes and allowing him to create, share and like recipes. The search profile capabilities will also help the user to search for his friends, view and like their recipes so encouraging them to follow the healthy steps their friends take and try new and different recipes that will smooth and facilitate the healthy pattern he wants to take. The app can also motivate you by giving you healthier tips depending on the tracked food.

## **1.4. Gantt chart**

### Phase one



Figure 1:Gantt chart first chase

### Phase two

Figure : Gantt chart second phase

## **1.5. Project development methodology**

Stating the system modelling methodology clearly to be used for analysis and design is an important point when developing any system. Therefore, we decided to use waterfall which is a linear approach to software development. In this methodology, we followed a sequence of events which is:

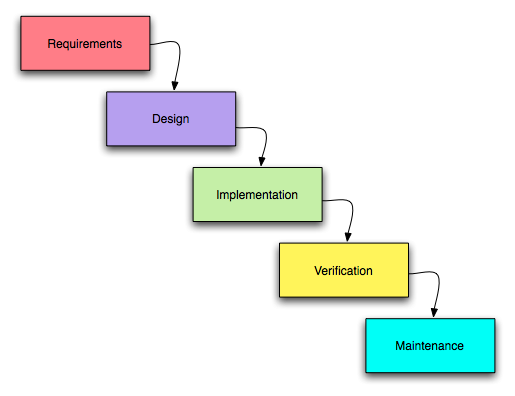


Figure 3: waterfall sequence of events

**Requirements:** The key aspect of waterfall is that all the requirements are gathered at the beginning of the project, allowing every other phase to be planned without further customer correspondence until the product is complete. It is assumed that all requirements can be gathered at this phase.

**Design:** The design phase is best broken up into two sub phases: logical design and physical design. The logical design sub phase is, when possible, solutions are brainstormed and theorized. The physical design sub phase is when those theoretical ideas and schemas are made into concrete specifications.

**Implementation:** The implementation phase is when programmers assimilate the requirements and specifications from the previous phases and produce actual code.

**Verification:** This phase is when the customer reviews the product to make sure that it meets the requirements laid out at the beginning of the project. This is done by releasing the completed product to the customer.

**Maintenance:** The customer is regularly using the product during the maintenance phase, discovering bugs, inadequate features and other errors that occurred during production. The production team applies these fixes as necessary until the customer is satisfied.

So that having many benefits such as:

* On what will be delivered early in the development lifecycle. This makes planning and designing more straightforward.
* Progress is more easily measured, as the full scope of the work is known in advance.
* Throughout the development effort, it’s possible for various members of the team to be involved or to continue with other work, depending on the active phase of the project.
* Because design is completed early in the development lifecycle, this approach lends itself to projects where multiple software components must be designed (sometimes in parallel) for integration with external systems.
* Finally, the software can be designed completely and more carefully, based upon a more complete understanding of all software deliverables.

## 

## **1.6. The used tools in the project**

Java - Android: we used android studio for purpose of built and to accelerate the app development and help build the highest-quality app. the language used in it is java.



Python –Machine learning: first we searched about a trained tflite model (ssd\_mobilenet.tflite) that contains some food to make our object detection.

SSD-MobileNet V2 Trained on MS-COCO Data and this model is a single-stage object detection model that goes straight from image pixels to bounding box coordinates and class probabilities.

And the training set information

MS-COCO is a data set for image recognition consisting of more than 300,000 images and 80 object classes.

Then we linked it with android studio by importing Open CV Module

.

## 

## **1.7 report organization**

In chapter 2 we will discuss some of the similar apps to our nutrient tracker app and what characterize each one of them mentioning their pros and cons.

In chapter 3 we will analyze the system using functional requirements that describe the functions of the app that must be performed. It represents a set of standards used to judge the specific operation of a system. Then the nonfunctionalrequirements that ensure the software app follow legal and compliance rules. Then finally the use case diagrams and tables.

In chapter 4 we explain the system design using diagrams to illustrate the system classes, components and architecture.

In chapter 5 we discuss the steps of the implementation then we give many test cases for the android part and the ml part provided by figures to illustrate it.

# **Chapter 2: Related work**

## **2.1 MyFitnessPal**

MyFitnessPal is one of the most popular calorie counters right now. It tracks your weight and calculates a recommended daily calorie intake. It also contains a well-designed food diary and an exercise log. The homepage provides a clear picture of how many calories you have consumed during the day.



In addition, it shows also the number of calories you have burned by exercising. The app also saves your favorite meals for convenient logging.

Pros:  
 MyFitnessPal has the largest database available in a diet tracker and includes many restaurant foods.  
It can download recipes from the internet and calculate the calorie content of each serving.  
You can “quick add” calories if you don’t have the time to add details about a certain meal.  
  
Cons:  
Since most foods are uploaded by other users, the calorie count may not be entirely accurate. Multiple entries may exist for the same product.  
Serving sizes in the database may be hard to edit, creating difficulties if your serving was smaller or larger than the one listed.

The Main difference between MyFitnessPal and nutrients tracker app is that Nutrients Tracker App provides a list with your favorite recipes and another list for you to created and added recipes. Also, it allows you to search for a friend and see his added recipes.so this its strength points despite not having the largest database nor the ability to download recipes.

## 

## **2.2 Lose It! – Calorie Counter**

Based on your weight, height, age and goals, Lose It! provides a personalized recommendation for calorie intake. It then tracks your calories on the home page.

Additionally, the Lose It! app has a barcode scanner for packaged foods, and common foods are saved for quick entry later on.

Lose It! presents weight changes on a graph, provides access to an active chat community and keeps a daily and weekly total. Its tab called “challenges” allows you to participate in dietary challenges or make your own.



Pros:  
Lose It! has a food database complete with popular restaurants, grocery stores and brand-name foods, all of which are verified by their team of experts.  
The app lets you set reminders to log your meals and snacks.  
  
Cons:  
It’s hard to log home-cooked meals or calculate their nutritional value.  
The app can be tricky to navigate.  
Lose It! doesn’t track macro nutrients.

The Main difference between lose it and nutrients tracker app is that Nutrients Tracker App doesn't have a barcode scanner for packaged foods also the challenges tab provided by lose it but it tracks the macros

# **Chapter 3: System Analysis**

## **3.1 Project Specification**

### 3.1.1 Functional Requirements

#### 3.1.1.1. Signup

User should be able to register into the system to be able to use its functionalities.

* Registration process needs user’s info (Email, User name, Password, Name, Gender, and Phone Number).
* The system checks whether the username and email had been used before, if it is not, it will create new account. Otherwise, the system will ask user to enter a new email/username.
* After a successful registration, a new account will be saved in the database and redirects him / her to the main page.

#### 3.1.1.2. Login

* User should be able to login into the system to be able use its functionalities.
* The System will allow the users who have account before to login into the system with registered email or username and password.
* The system checks whether the login information is correct. If yes, it creates a session for the user and redirects the user to the main page. Otherwise, the system redirects the user to the login page to try again.

#### 3.1.1.3. Search users

Logged user should be able to search for any user in the app by the username to be to view users’ profile.

#### 3.1.1.4. View Profile

Logged user should be able to view his profile and other users' profiles that Allows the user to do some functionalities.

##### **3.1.1.4.1. View My profile**

* + - **3.1.1.4.1.1. Edit profile**
      * Logged user should be able to edit his registered information after view his profile and choosing this option.
* The system will update the new information in the database then redirects the user to his profile page.
  + - **3.1.1.4.1.2. View my favorite list**
* Logged user should be able to view his Favorites list in his profile.
  + - **3.1.1.4.1.3 Delete from favorite list**
      * User should be able unfavourite a recipe from his favorite list after viewing recipe details.
    - **3.1.1.4.1.4 delete recipe** 
      * User should be able to remove a recipe from his recipe list after viewing recipe list

##### **3.1.1.4.2. View Users’ profile**

* **3.3.4.2.1. Add to favorite list**
* Logged user should be able to add other users' recipes in his Favorite List.

##### **3.1.1.4.3. View Recipe list**

Logged user should be able to view his own recipes in his profile and also can view users' recipes in a users' profile.

* **3.1.1.4.3.1. View recipe details**
* Logged user should be able to view all details of a specific recipe and its nutrients by click on the recipe in the recipe list.

#### 3.1.1.5. Access Camera

The System will request access to the user’s camera and gallery when the user request to upload or capture an image for the first time.

#### 3.1.1.6. Scan Food

* + The system checks if the user gives the system the permission to open his phone camera before open it. If it is not, the system ask user to allow camera access.
  + Logged user should be able to open the mobile camera and point it to a plate containing his meal to take a photo.
  + The system will use the photo to track the nutrients for each ingredient.

##### **3.1.1.6.1. Recognize Food**

The System will Recognize food name and detect its ingredients after the user

Takes the photo of his meal.

#### 3.1.1.7. Add Recipe

* Logged user should be able to create his food recipe and track the recipe nutrients.
* The system will save it in the database to view it in his profile.

#### 3.1.1.8. Search Recipe

Logged user should be able to search for specific foods created by another users to track its nutrition facts.

#### 3.1.1.9. Calculate Nutrients

The System will display table of all the nutritional facts as (Carbs, Fat, Protein, Sugar,

Fiber, cholesterol, vitamins ... etc.).

#### 3.1.1.10. Macro Tracker

* Logged user should be able to request to reduce/increase the Marcos (carbs, fats, protein) in his recipe or other users' recipes.
* The System will suggest ideas and some tips based on his request.

#### 3.1.1.11. Log out

* Logged user should be able to choose to log out from the system. By choosing this option in his profile.
* The system closes the open logged user session and redirects the user to the Start page.

### 

### 3.1.2 Non-Functional Requirements

|  |  |
| --- | --- |
| Performance | All operations that need system response should be done in a few Seconds. |
| Usability | The apps interface will be user friendly that the users will become familiar with it after logging in and using it few times. |
| Availability | The application will not be down for no more than one percent a day. |
| Security | The system must be highly secured to raise the reliability level and to let the users feel safe when using the system by using Password facility to ensure validity of user (The system validates each user by username and password combination). |
| Maintainability | It will be easy to find bugs and fixes them. |
| Reliability | The probability of a system failure should be minimized as much as possible. In other words, the system should be failure safe. |

Table 1: nonfunctional requirements

## **3.2 Use Case diagrams**

### 3.2.1 Use case diagram

Figure : Use case diagram

###### 

### 

### 3.2.2 Sample Use-Cases

#### 3.2.2.1. Signup

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3.1.1.1 | |
| Use Case Name: | Signup | |
| Actors: | User | |
| Pre-conditions: | User initiates the registration process. | |
| Post-conditions: | User Successfully registered into the system and accessed his mobile phone’s camera. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enters required information: Username, Gender, Phone-Number, Name, Email, and Password. |  |
|  | 2- System Verify that all fields are completed. |
|  | 3- System Create a profile to the user. |
|  | 4-System will ask user to access his mobile phone’s camera. |
| 5-User enters allow button. |  |
|  | 6-System allows user to take a photo. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enters Invalid Information doesn’t match the requirements. |  |
|  | 2-System Prompts the user to re-enter the information. |
| 1-User doesn’t access his mobile phone’s camera |  |
|  | 2-System will ask user to access his mobile phone’s camera. |
| Includes: |  | |
| Notes and Issues: |  | |

Table 2: Sign up use case table

#### 3.2.2.2. Login

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3.1.1.2 | |
| Use Case Name: | login | |
| Actors: | User | |
| Pre-conditions: | User already signed up and have an account. | |
| Post-conditions: | User Successfully logged. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enters his Email and Password. |  |
|  | 2- System Verify the entered email and password. |
|  | 3-System displays login successfully message. |
| 3-User is logged in. |  |
|  | 4-System relocate user to home page. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enters Invalid Email and password. |  |
|  | 2-System Prompts the user to re-enter the email and the password. |
|  | 3-System suggests to the User to signup if he does not have an account. |
| Includes: |  | |
| Notes and Issues: |  | |

Table 3: Log in use case table

#### 3.2.2.1. Search Users

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3.1.1.3 | |
| Use Case Name: | Search Users | |
| Actors: | User | |
| Pre-conditions: | User logged into the system. | |
| Post-conditions: | User Successfully Searched a User account. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enters the Username to find and Click on Search button. |  |
|  | 2- System search for the Username entered. |
|  |  | 3-System returns the Username profile if it is found. |
| Exceptions: | **User Action** | **System Action** |
| 1-User enters invalid or not existed Username |  |
|  | 2-System will not reach a User and gives no information. |
| Includes: | View user’s profile – view recipes. | |
| Notes and Issues: |  | |

Table 4: Search user use case table

#### 3.2.2.3. Edit profile

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3.1.1.4.1.1 | |
| Use Case Name: | Edit profile | |
| Actors: | User | |
| Pre-conditions: | User logged into the system and viewed his profile. | |
| Post-conditions: | User Successfully edit his profile. | |
| Flow of events: | **User Action** | **System Action** |
| 1-User Clicks on the edit profile icon. |  |
|  | 2-System will ask user to enter new information to be saved. |
| 3-User Enters the surrogate information for Email, Username, Password, phone, and Name. |  |
| 4-User clicks on Save Button. |  |
|  | 5- System will update the User’s information. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

Table 5 : Edit profile use case table

#### 3.2.2.2. Scan Food

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3.1.1.6 | |
| Use Case Name: | Scan Food | |
| Actors: | User | |
| Pre-conditions: | User logged into the system and accessing his phone’ camera. | |
| Post-conditions: | User Successfully taking a photo of food and detecting food’s name and ingredients. | |
| Flow of events: | **User Action** | **System Action** |
| 1-User Clicks on Scan Photo button |  |
|  | 2-System opens camera to allow user to take photo. |
| 3-User Takes a photo a plate of food. |  |
|  | 4- System saves the photo and sends it to the recognition Model to return Food name and its ingredients. |
|  | 5- System will use the ingredients to calculate the nutrients. |
| Exceptions: | **User Action** | **System Action** |
| 1- User doesn’t access his phone camera. |  |
|  | 2-System will ask him to access his phone camera. |
| 1-User takes a photo doesn’t containing food. |  |
|  | 2-System will ask the user to take another photo. |
| 1-User takes a photo containing food doesn’t match with database. |  |
|  | 2-System will match food with the most other foods and ask user if there is in or not. |
| Includes: | Recognize Food - Calculate Nutrients - Access Camera | |
| Notes and Issues: |  | |

Table 6: Scan food use case table

#### 3.2.2.3. Add Recipe

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3.1.1.7 | |
| Use Case Name: | Add Recipe | |
| Actors: | User | |
| Pre-conditions: | User logged into the system. | |
| Post-conditions: | User Successfully Add his Recipe. | |
| Flow of events: | **User Action** | **System Action** |
| 1-User Clicks on add recipe button. |  |
|  | 2-System will ask user to enter some information. |
| 3-User Enters the required information (Recipe Name, Ingredients, and Steps) for the recipe. |  |
| 4-User clicks on Save Button. |  |
|  | 5- System will Verify that all fields are completed. |
|  | 6-System will use the ingredients to calculate the nutrients. |
|  | 7- System saves the recipe in the database. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter invalid information doesn’t match the requirements. |  |
|  | 2- System Prompts the user to re-enter the information. |
| Includes: | Calculate Nutrients | |
| Notes and Issues: |  | |

Table 7: Add recipe use case table

#### 3.2.2.3. Add to favorite list

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5.3.4.2.1 | |
| Use Case Name: | Add to favorite list | |
| Actors: | User | |
| Pre-conditions: | User logged into the system and view another User’s profile. | |
| Post-conditions: | User Successfully Add Recipe to his favorite list. | |
| Flow of events: | **User Action** | **System Action** |
| 1-User Clicks on add to Favorite button. |  |
|  | 2-System will add the recipe to User’s favorite list. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

Table 8: Add to favorite list use case table

# **Chapter 4: System Design**

## **4.1 system component diagram**

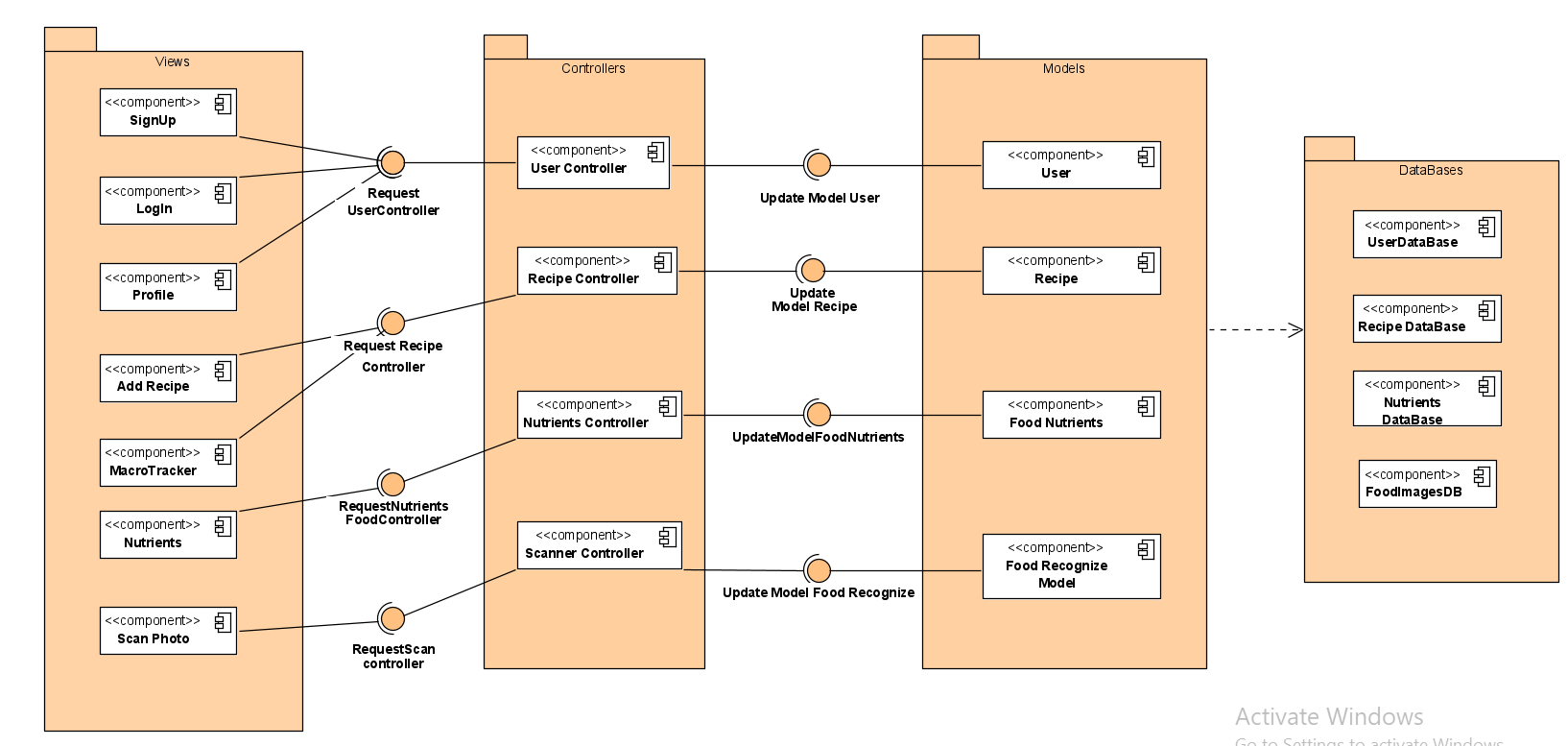


Figure : componentt diagram

## 

## **4.2 Class Diagram**

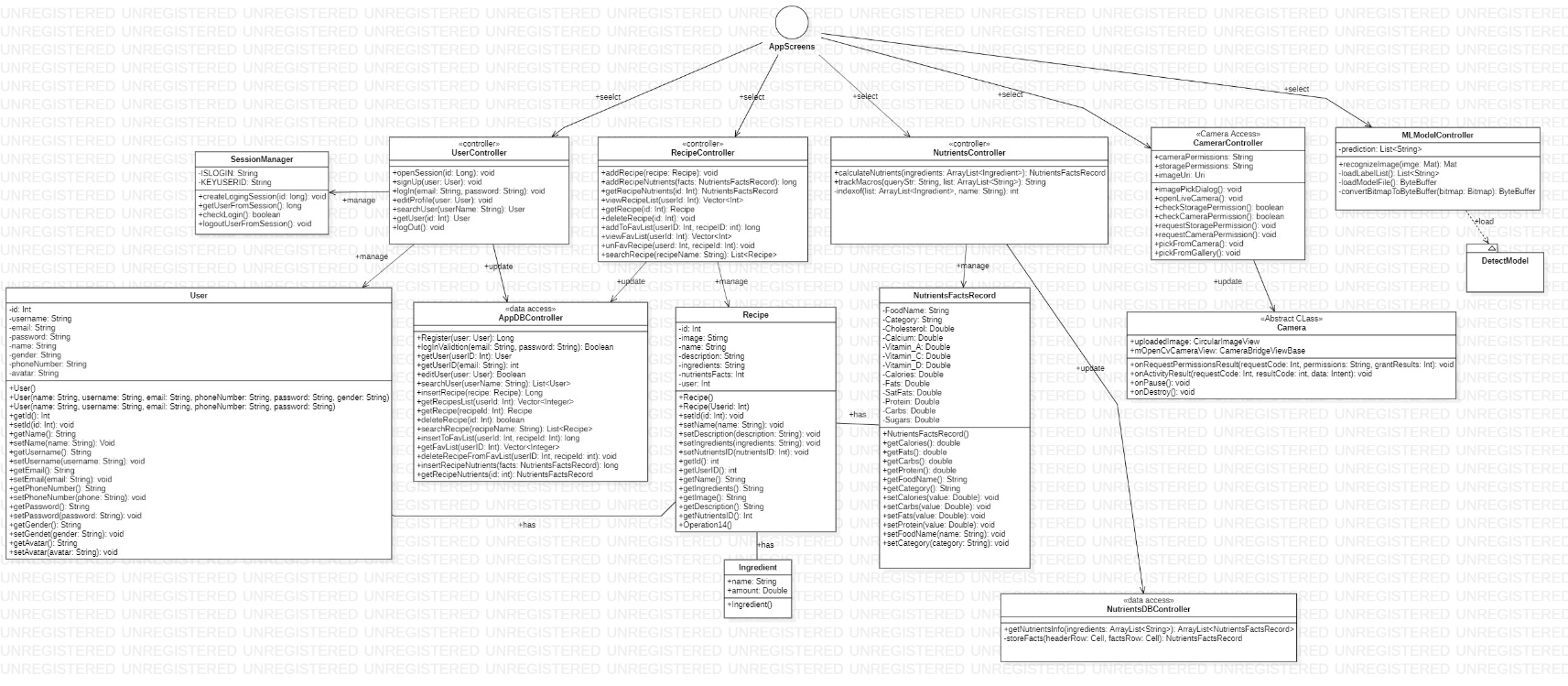


Figure : Class diagram

## 

## 

## **4.3 Sequence Diagram**

## 4.3.1. Signup



Figure : Signup sequence diagram

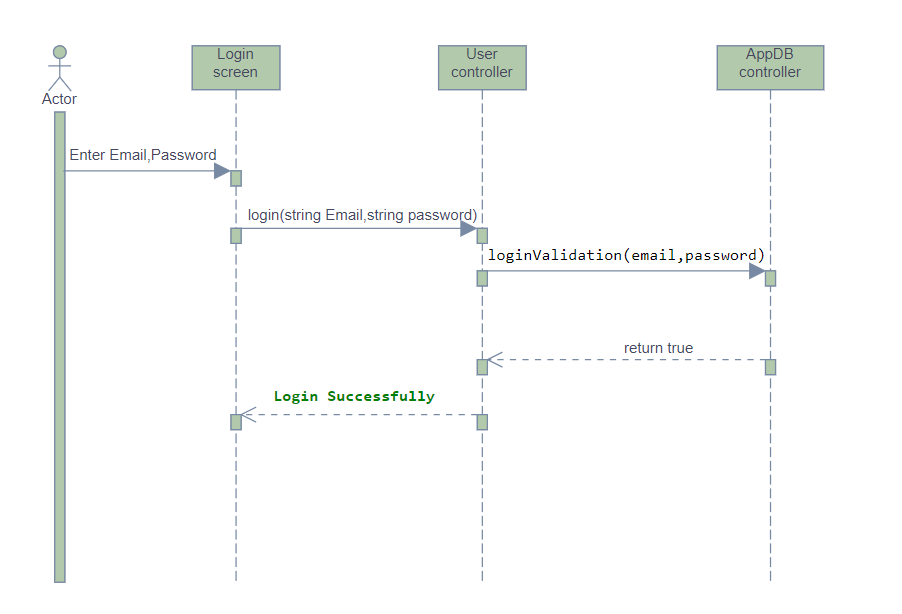
4.3.2. Login

Figure : Login sequence diagram

### 4.3.3 Add Recipe

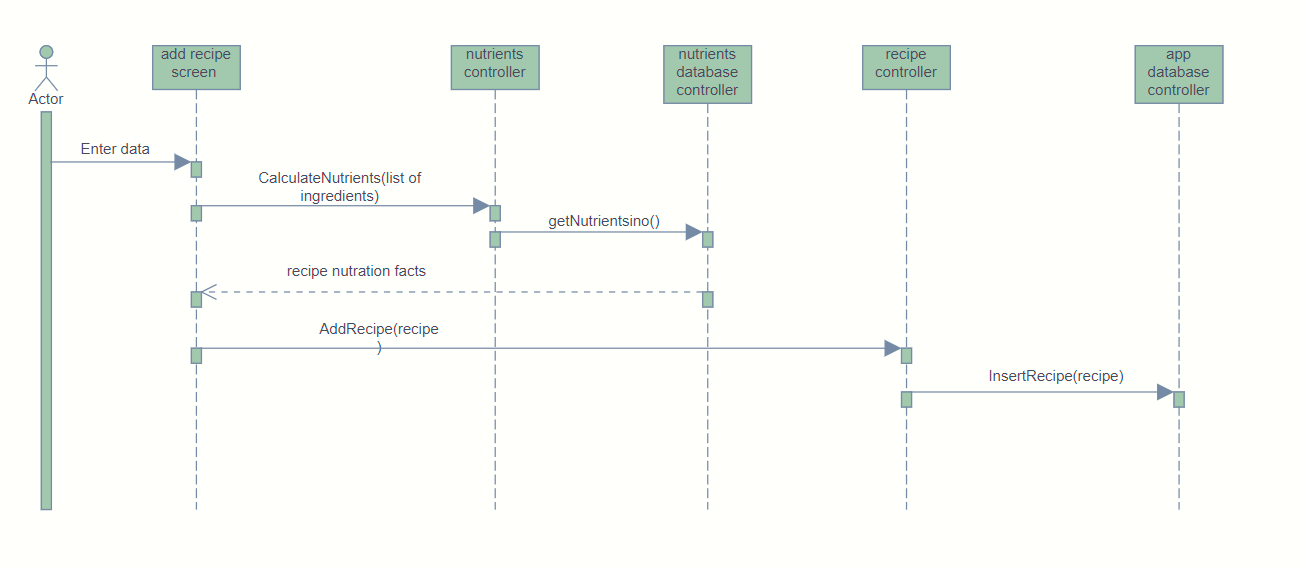
****

Figure : Add Recipe sequence diagram

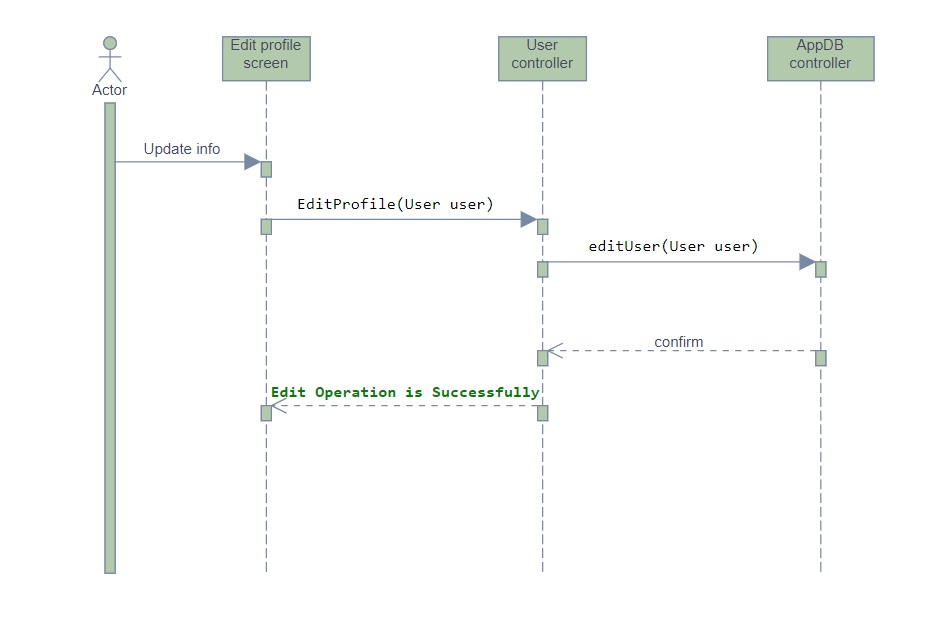
4.3.4. Edit profile

Figure : Edit profile sequence diagram

### 

### 4.3.5 Macro tracker

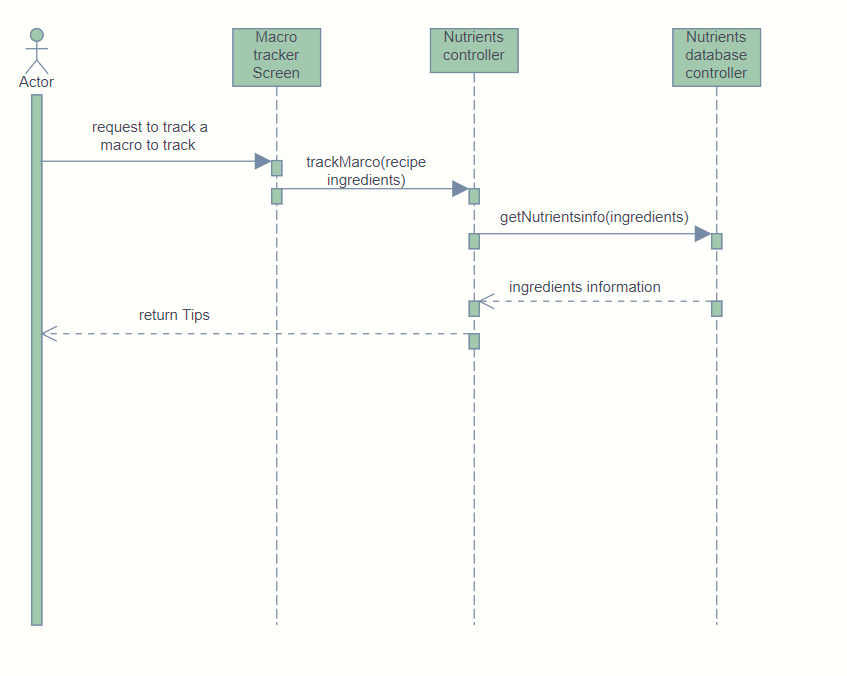
****

Figure 11: Macro tracker sequence diagram

4.3.6 Recipe nutrients

****

Figure : Recipe nutrients sequence diagram

### 4.3.7 Search user

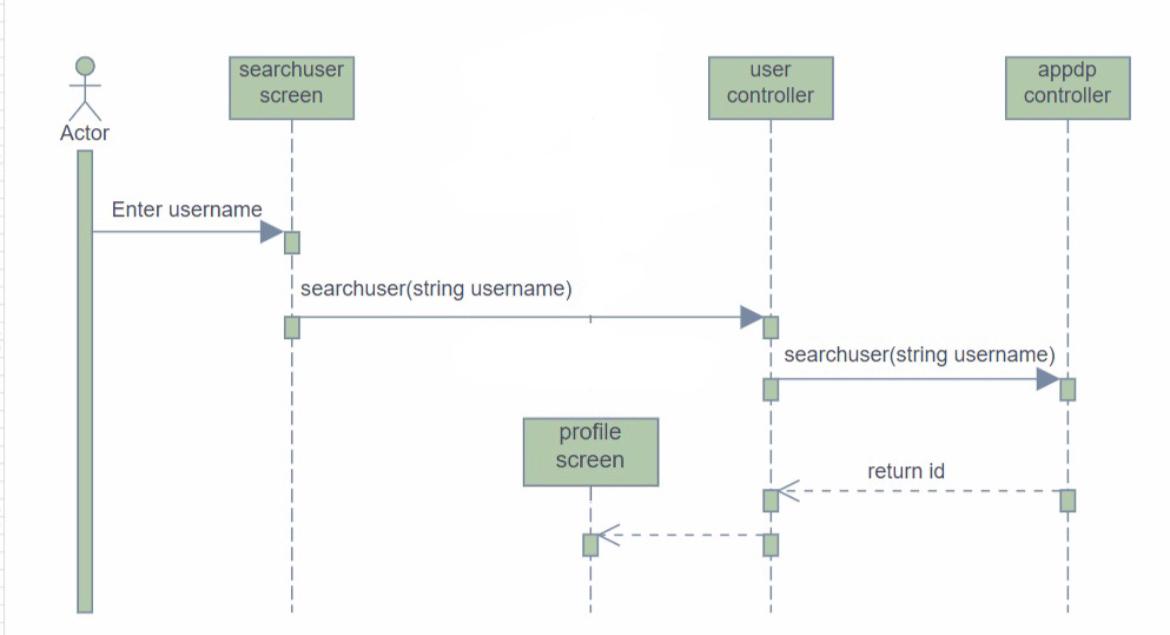


Figure : Search user Sequence diagram

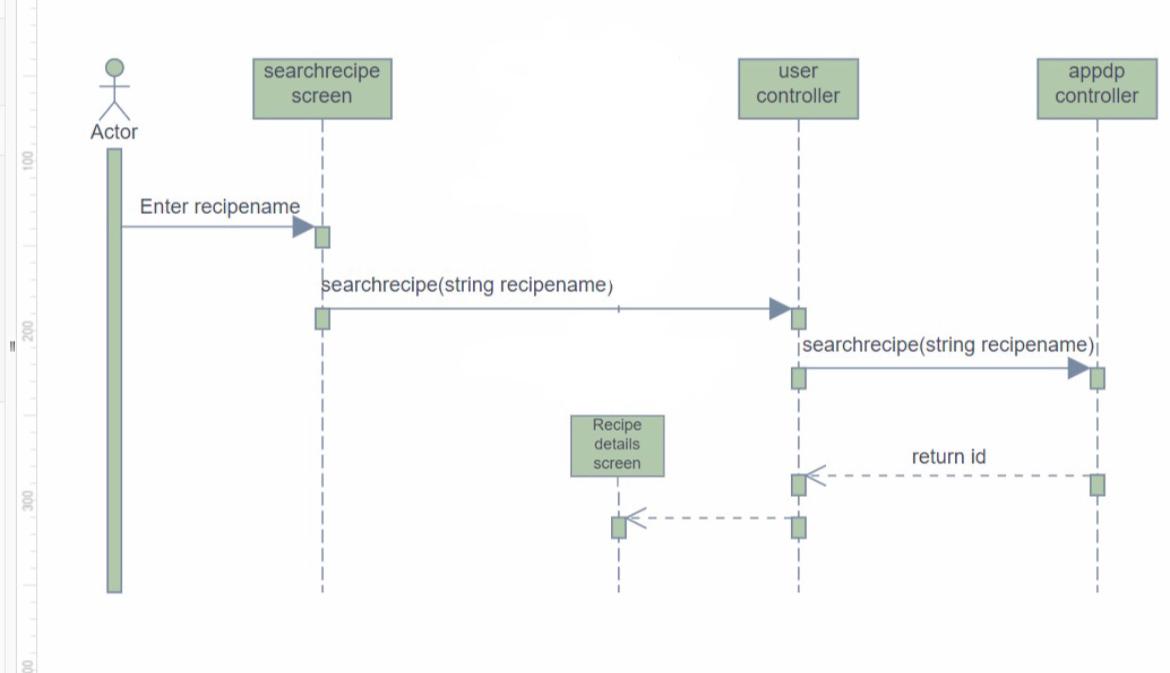
**4.3.8 Search recipe**

Figure : Search recipe Sequence diagram

**4.4 Entity Relationship Diagram (ERD)**

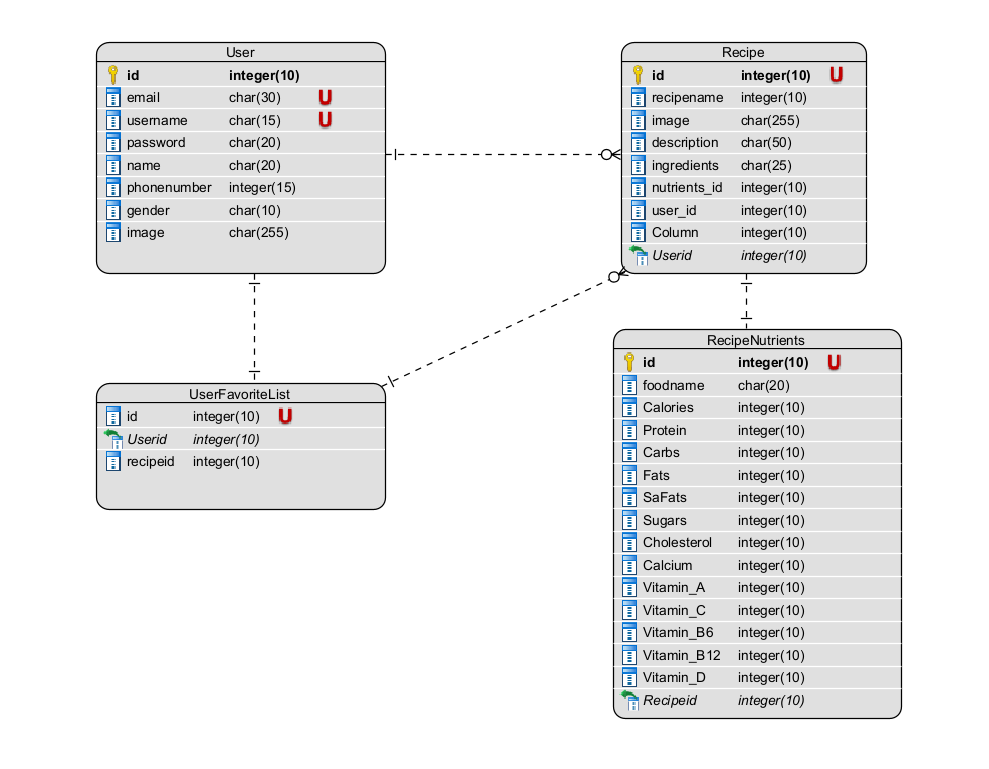
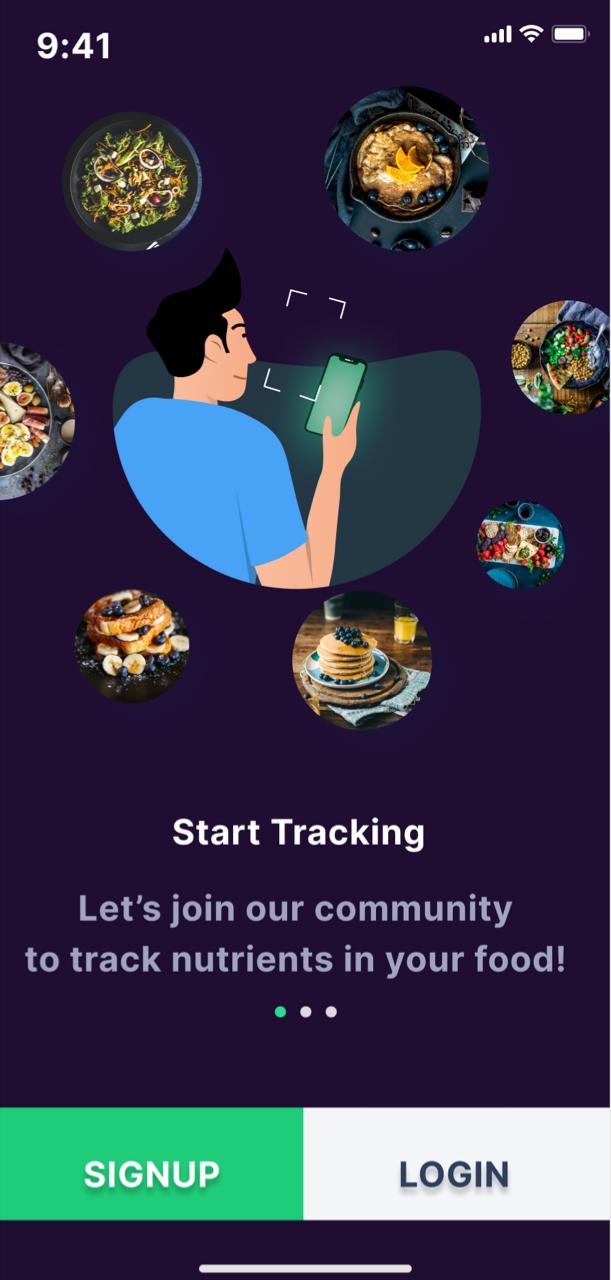


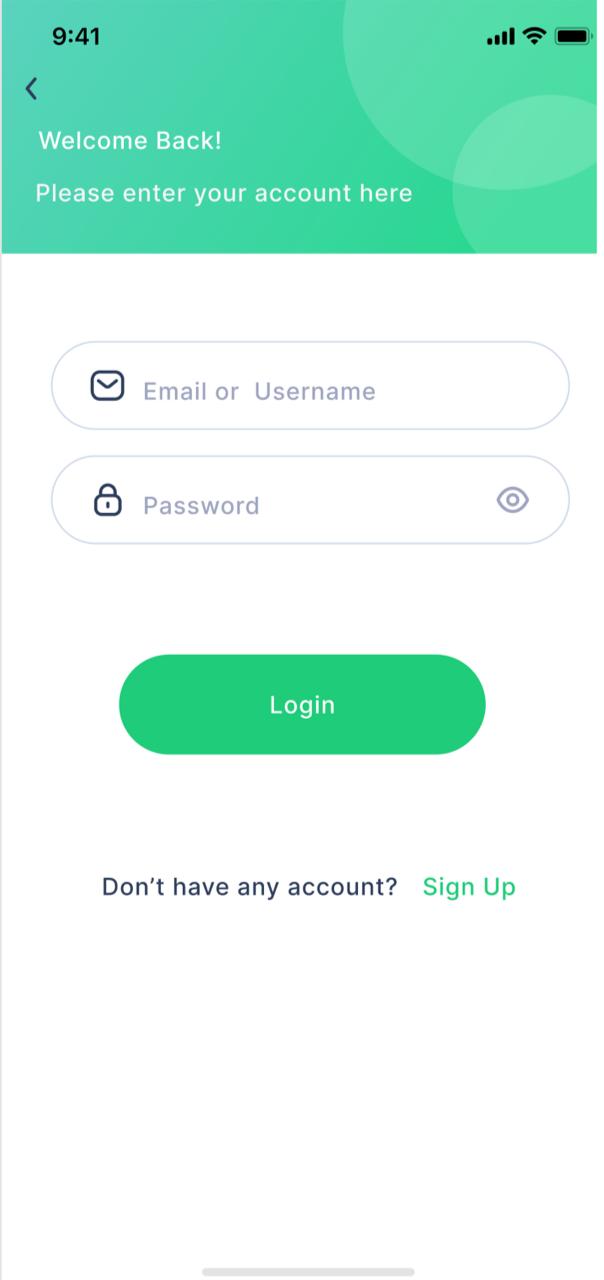
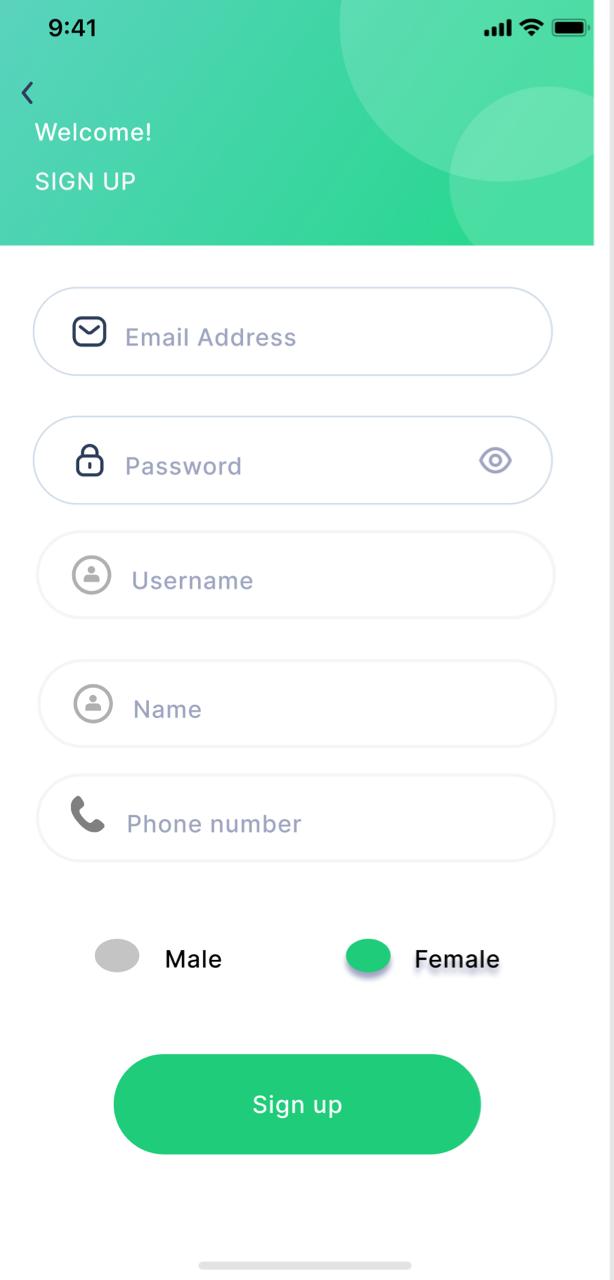
Figure 15: Entity Relationship Diagram

## **4.5 System GUI**

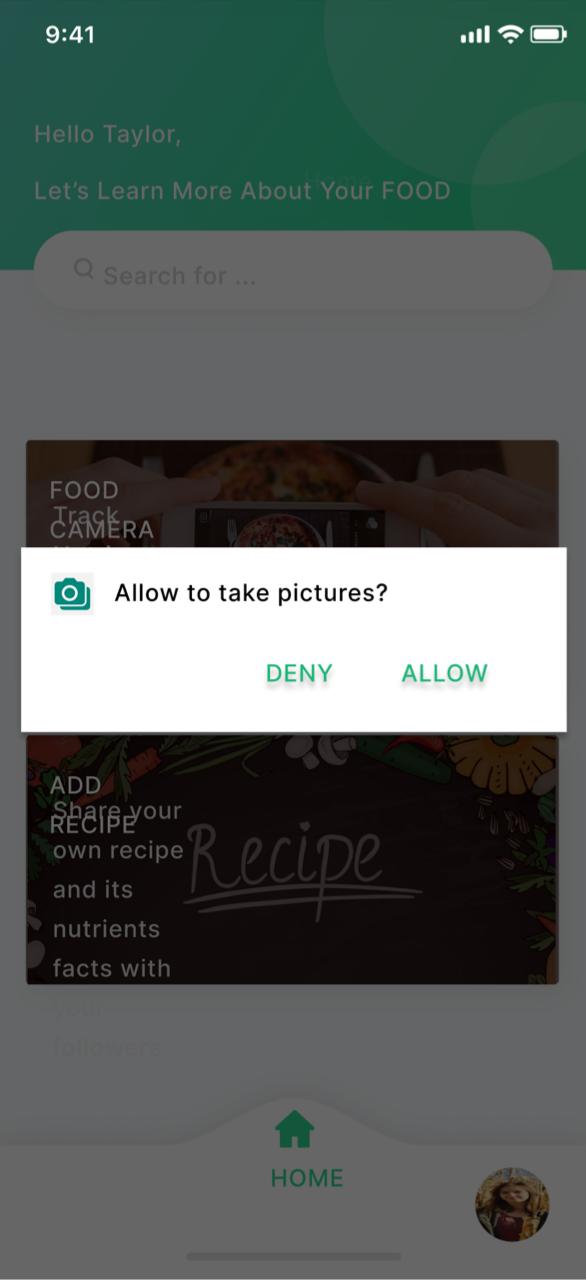
### Cover page



### Signup-login

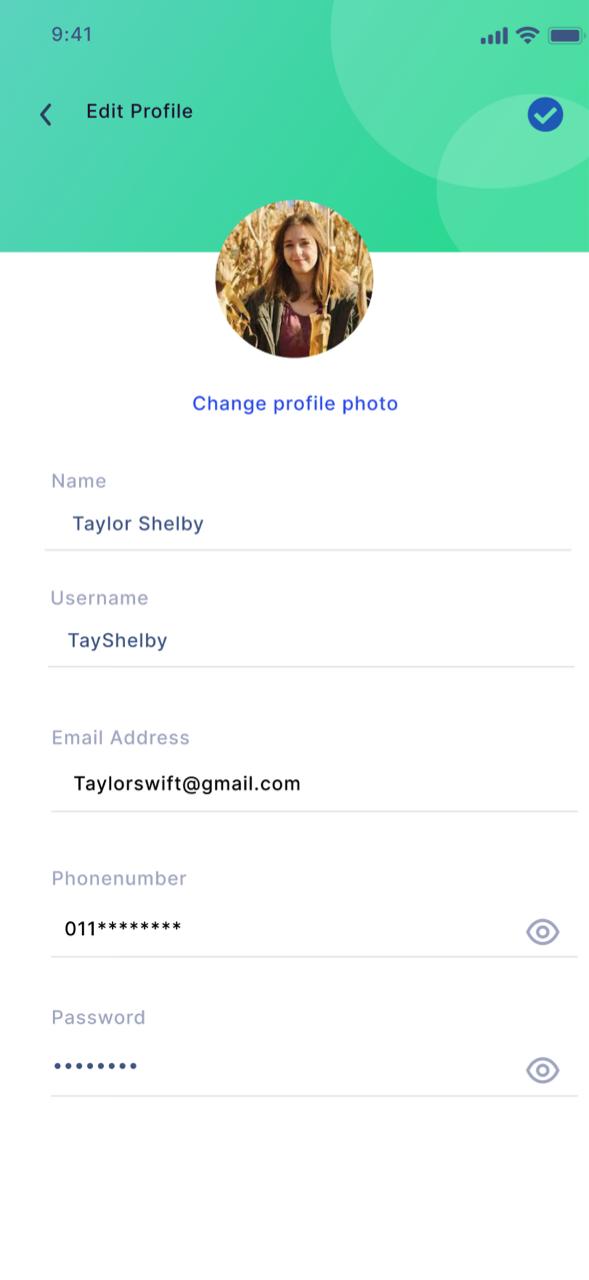
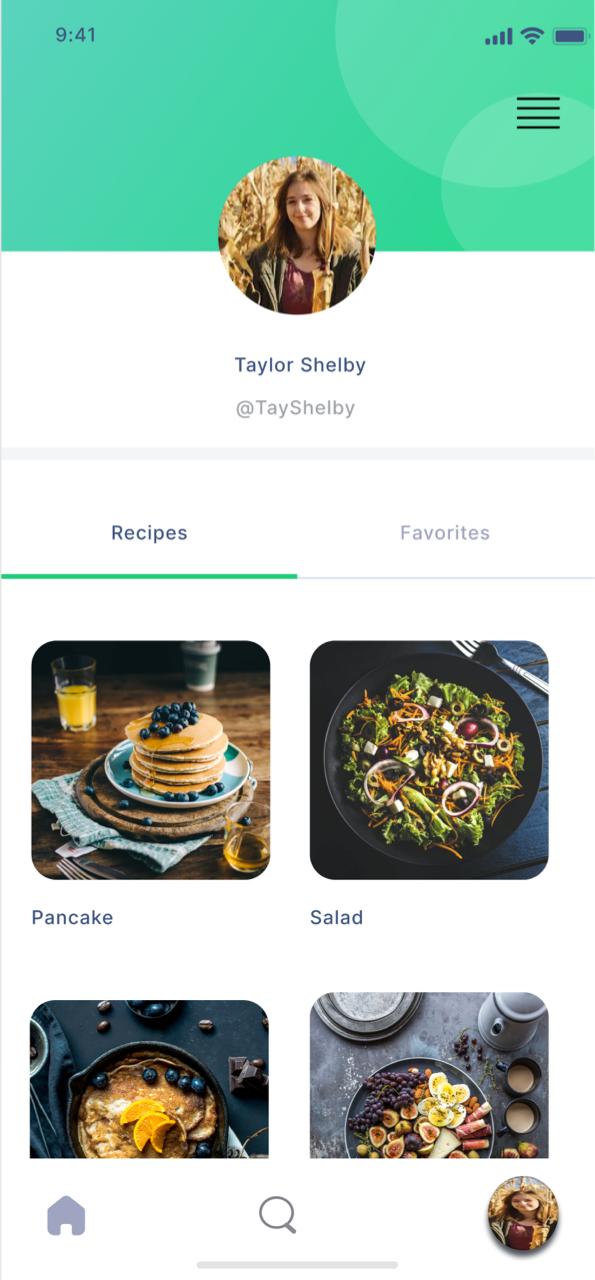


### Access camera

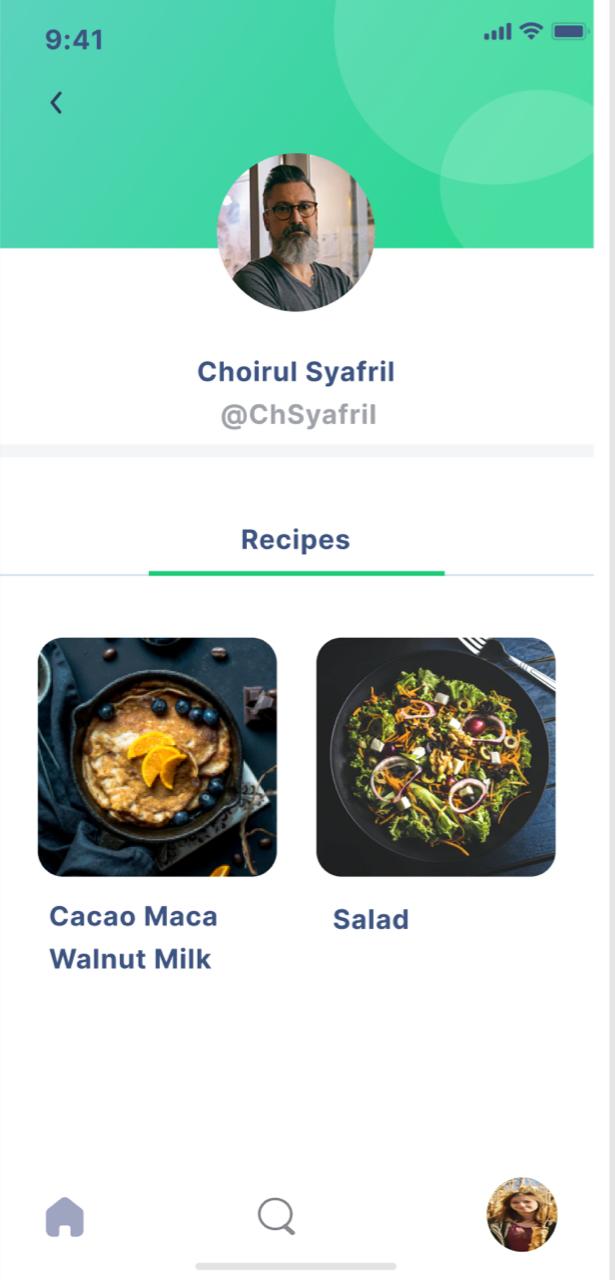


### My profile-Edit profile

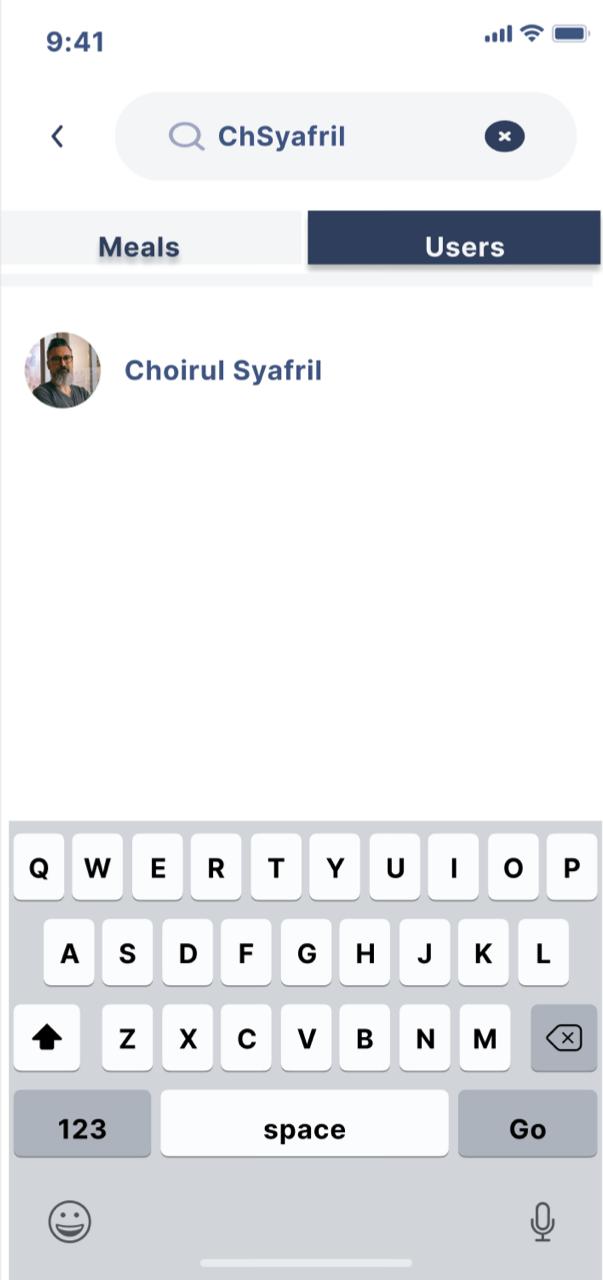
### 



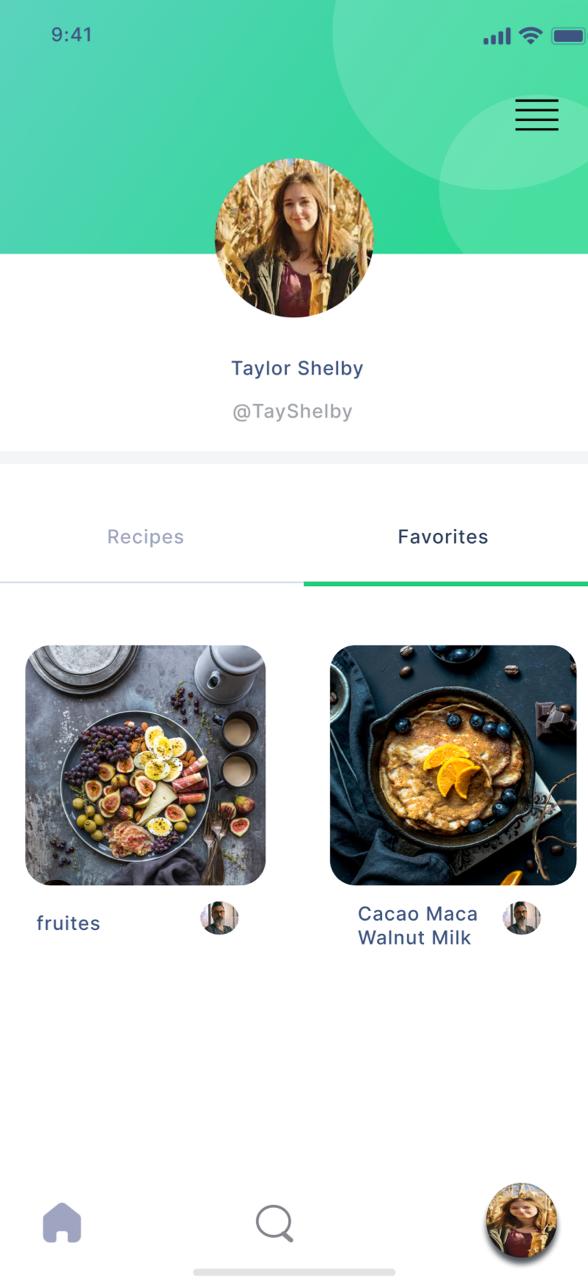
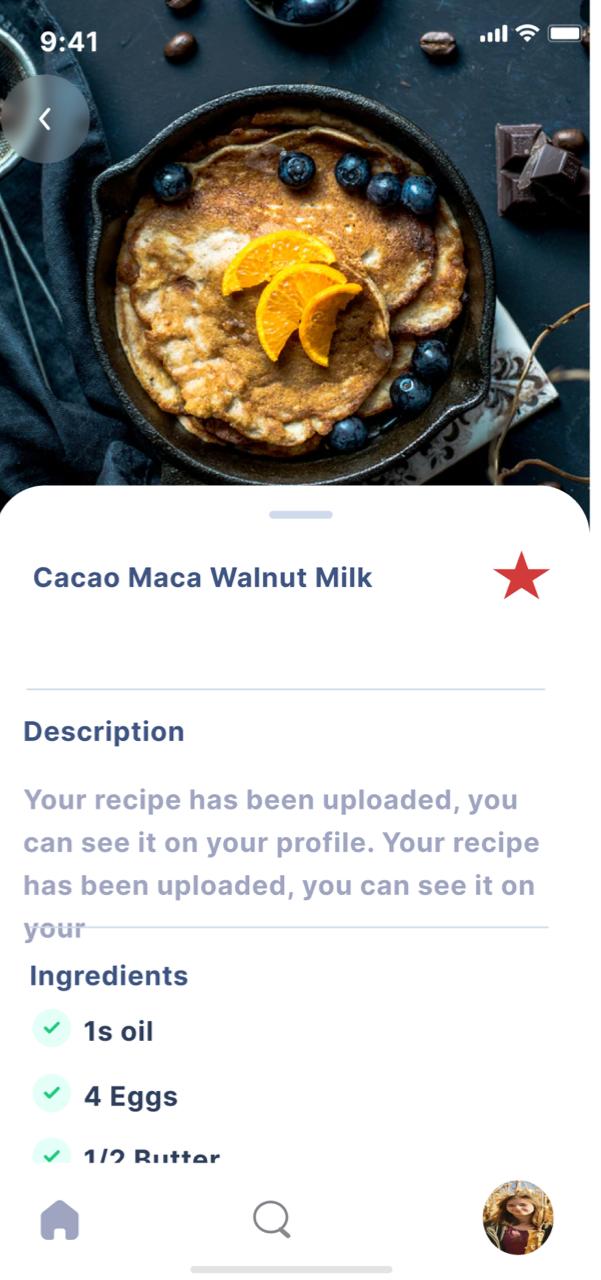
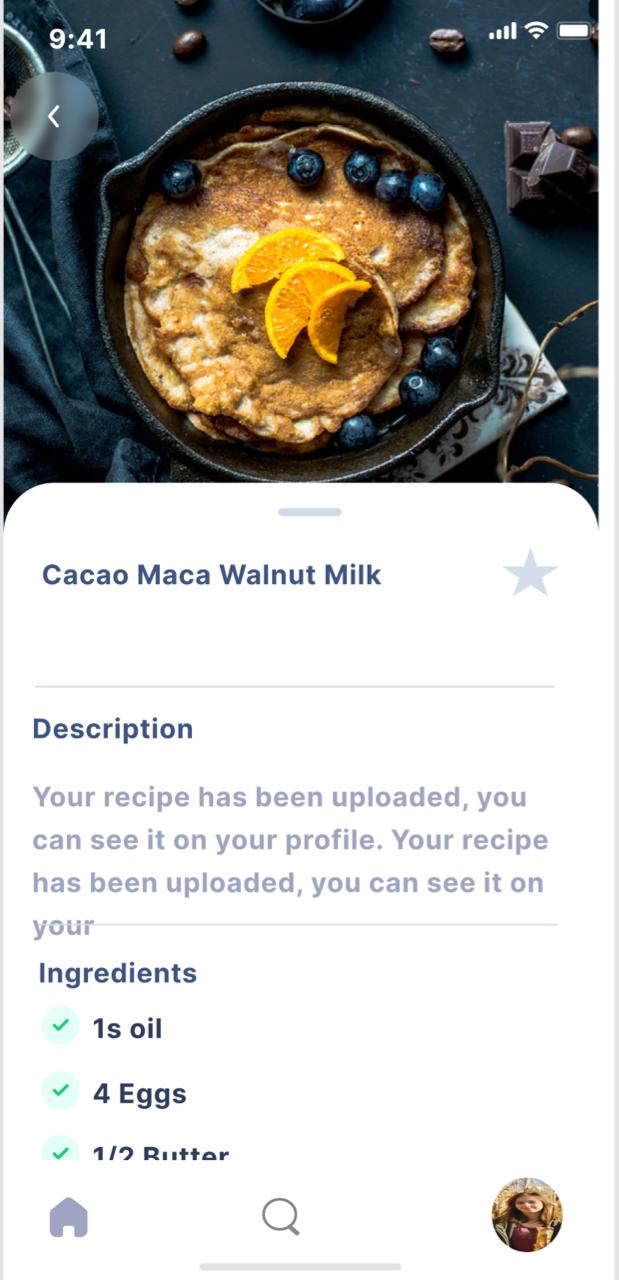
### Users’ profile



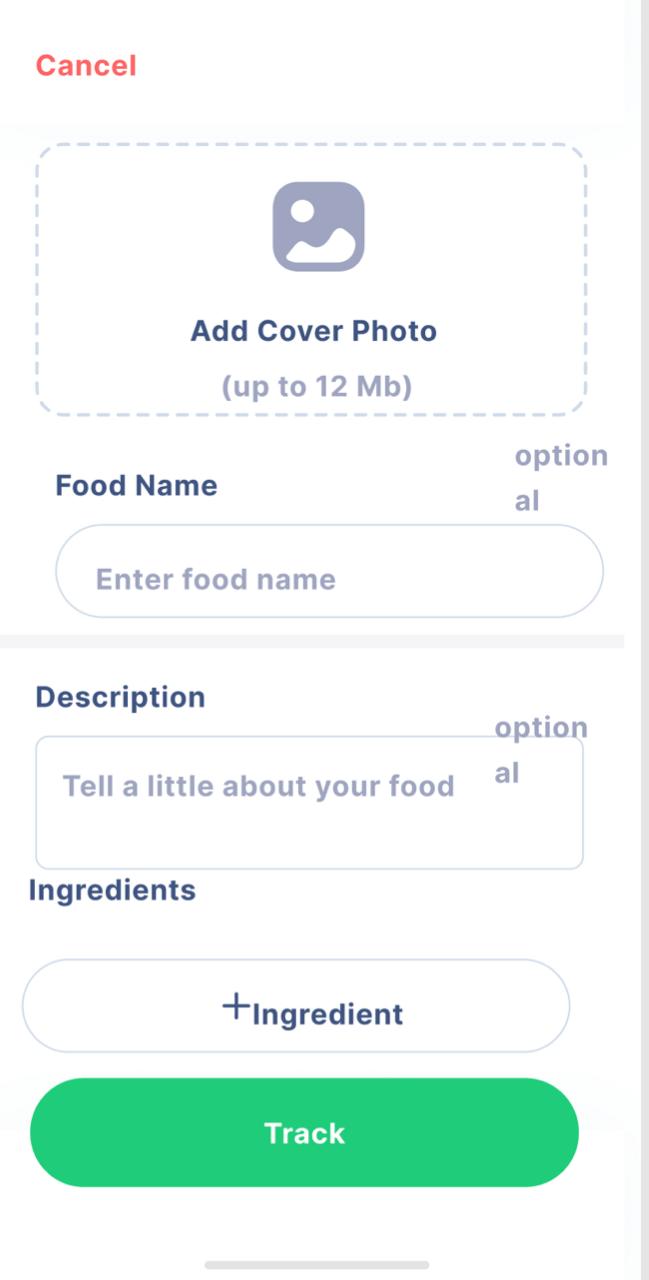
### Search user



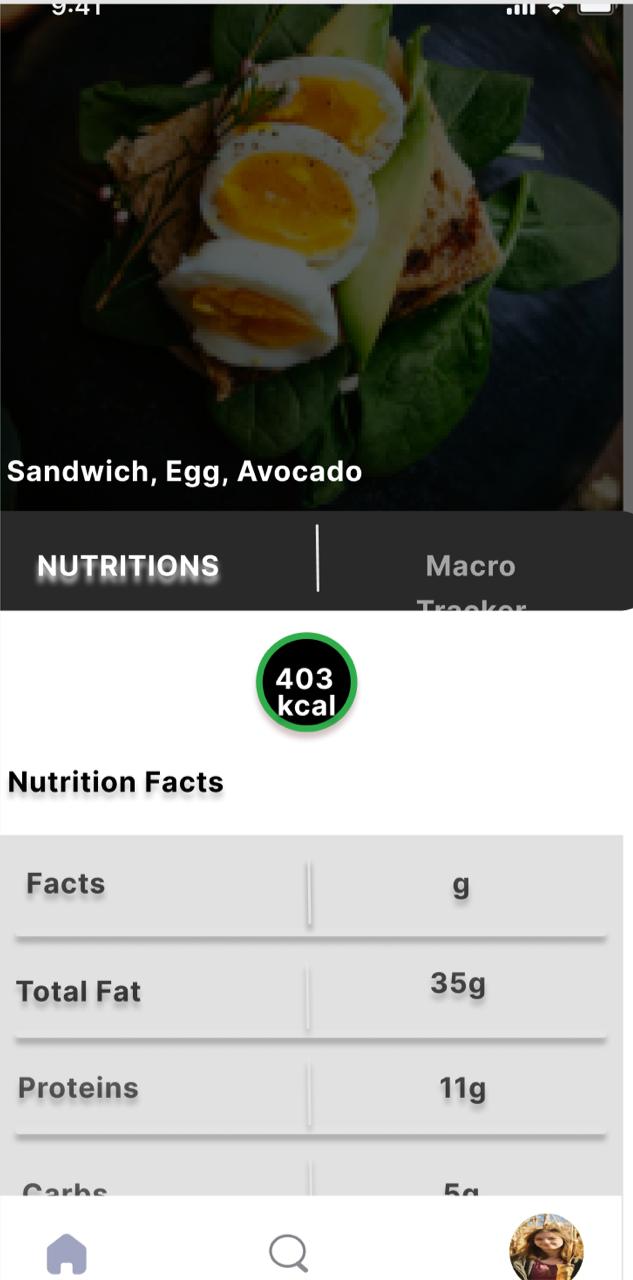
### Detail Recipe – Favorite list



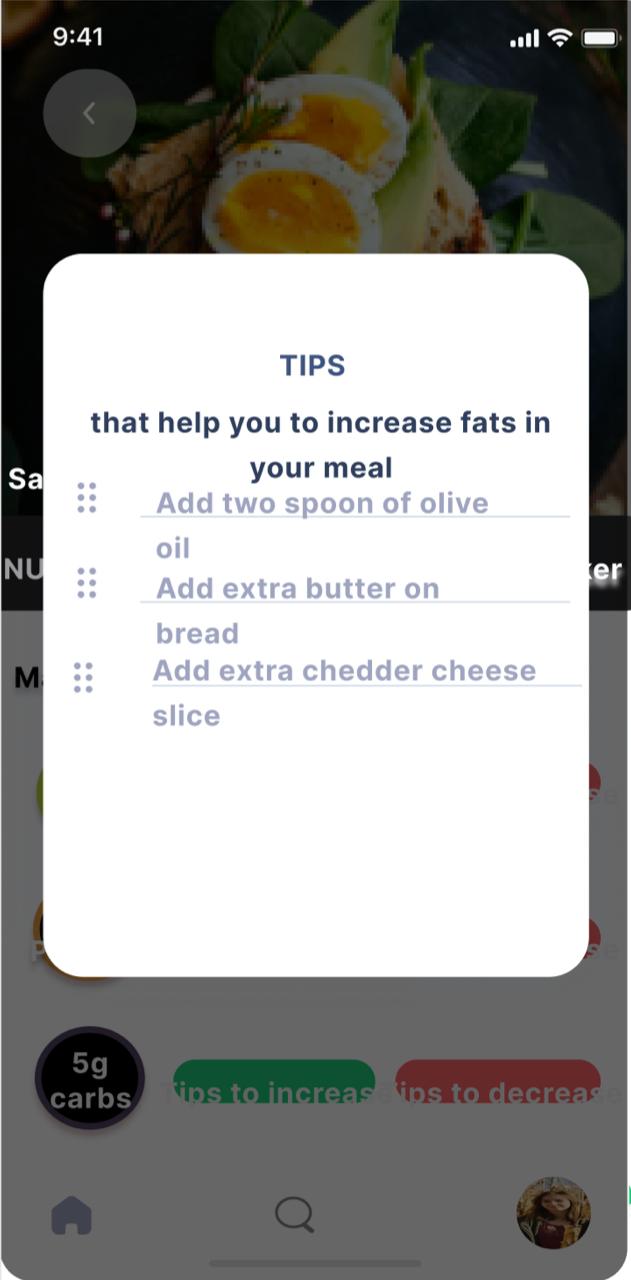
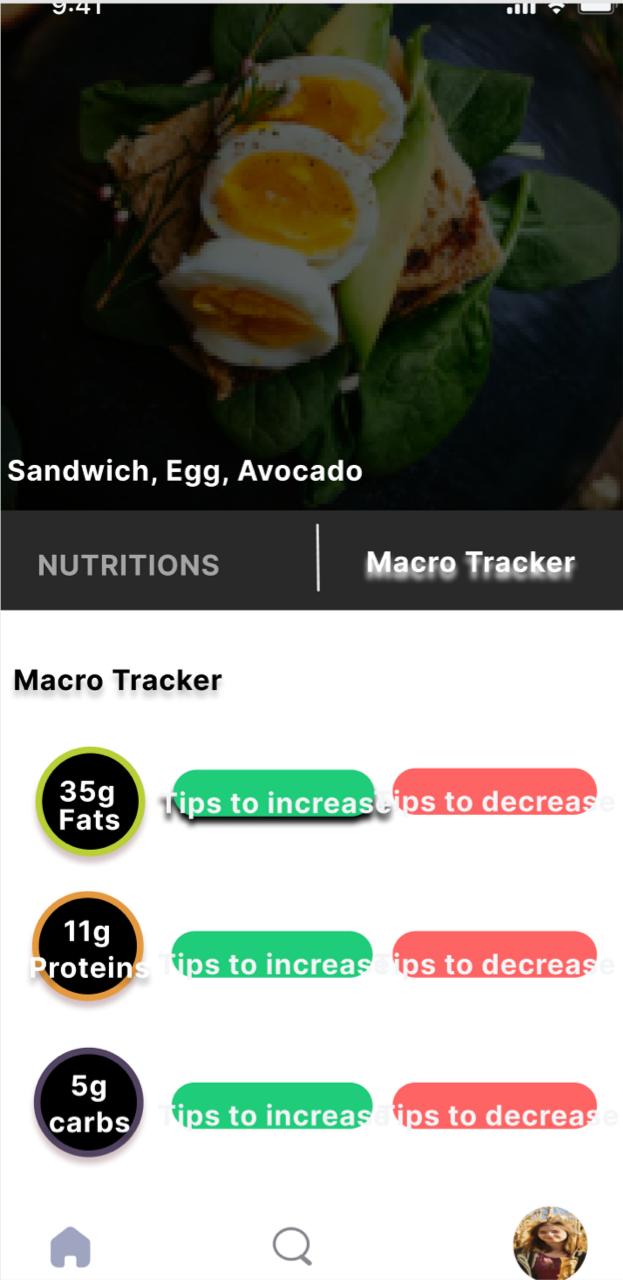
### Add recipe



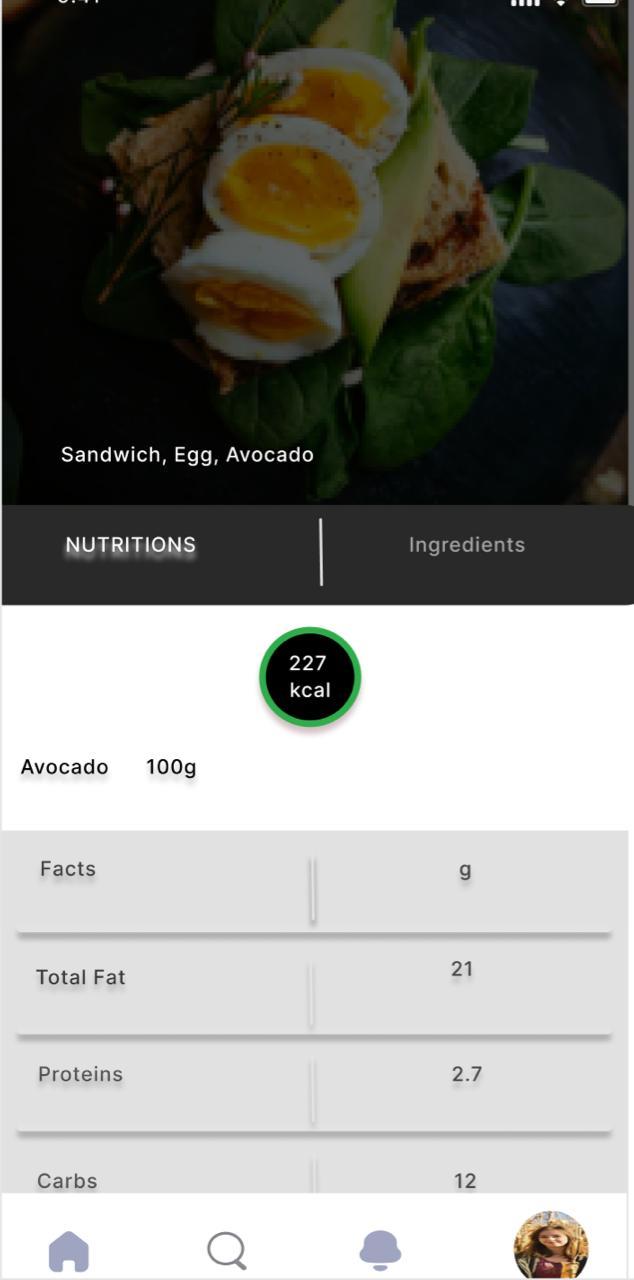
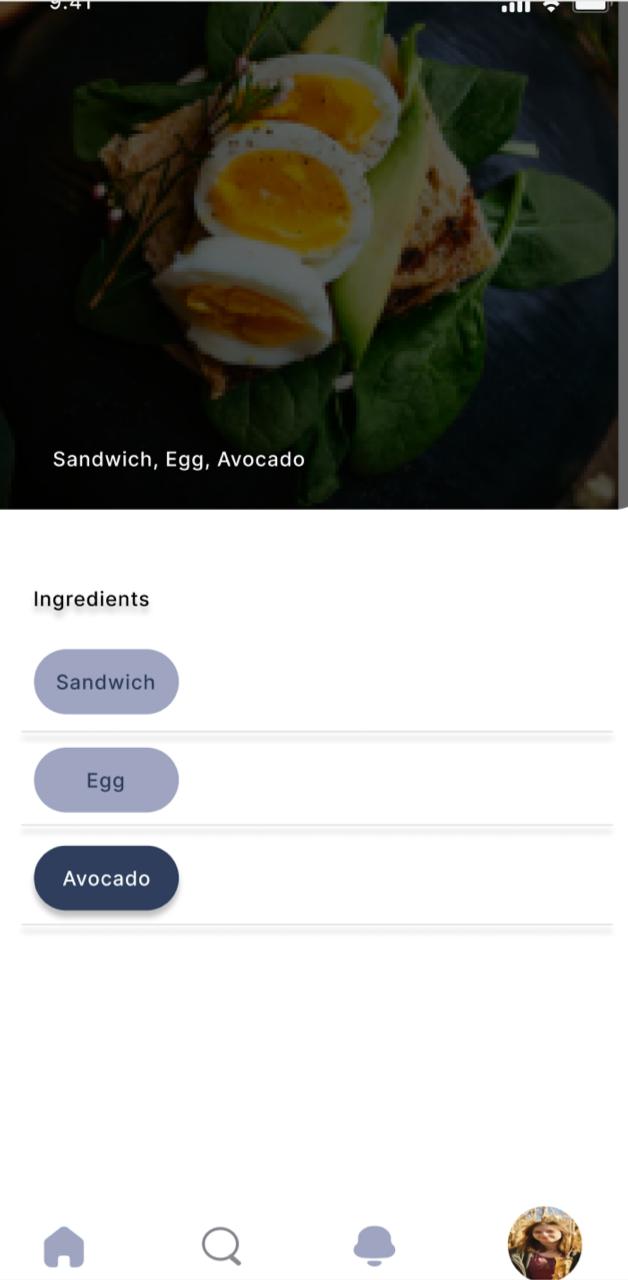
### Nutrition facts



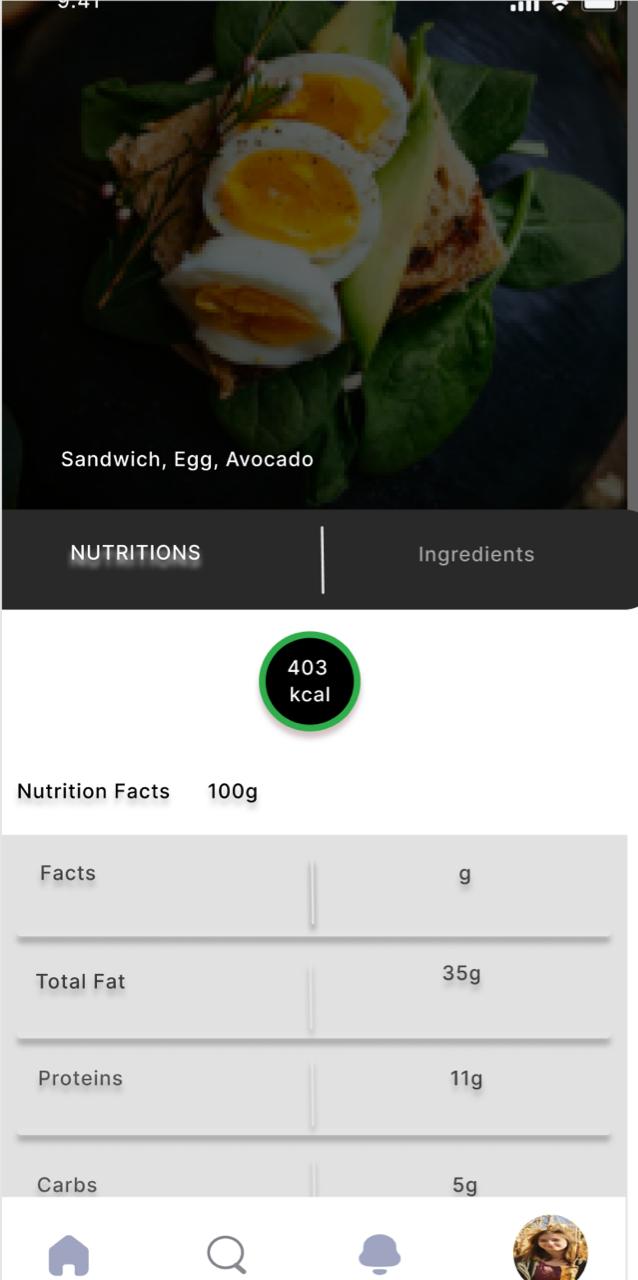
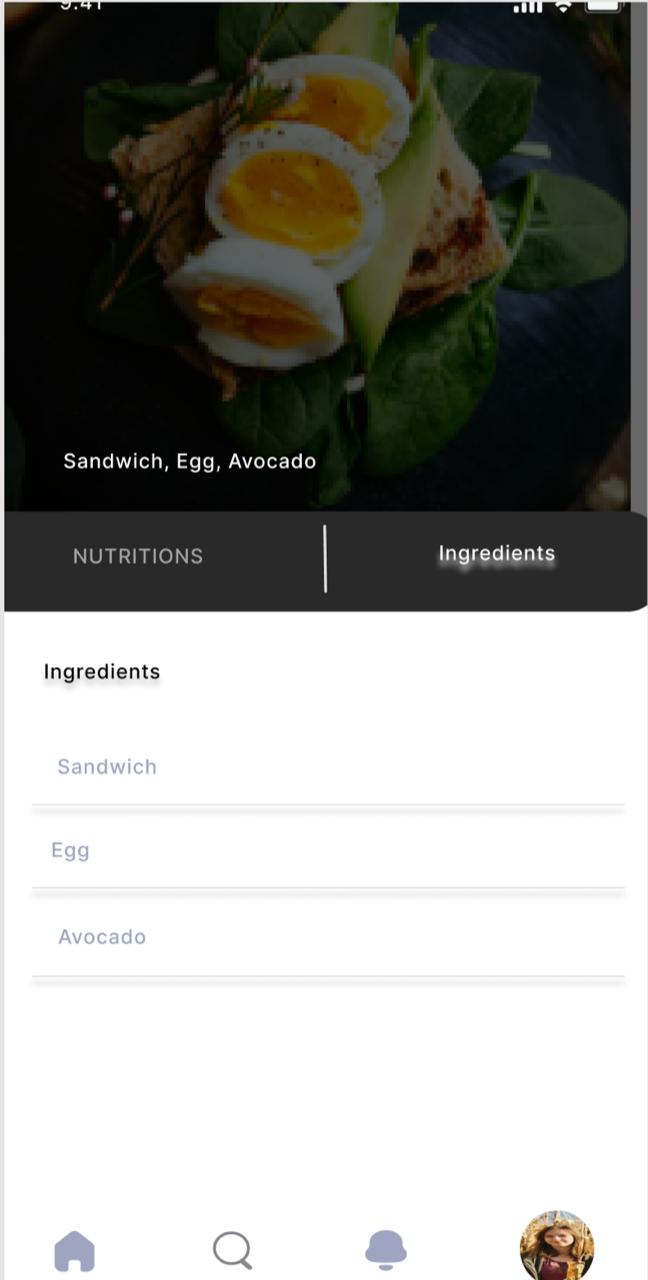
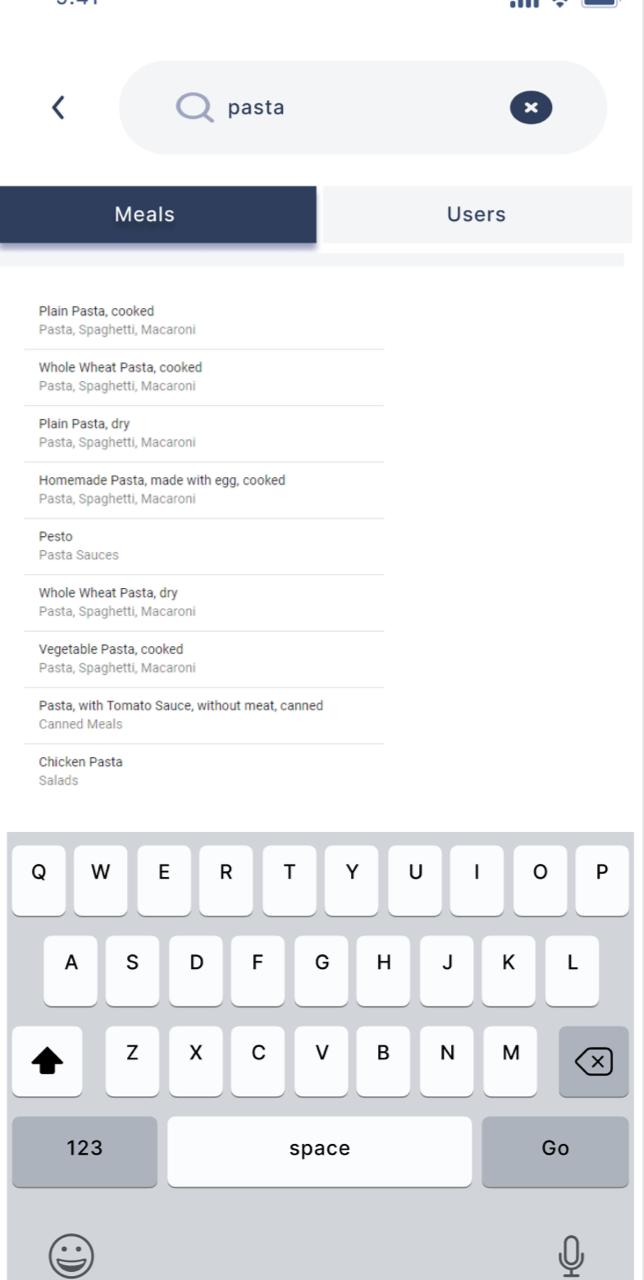
### Macro Tracker – Tips



### Scan Camera-Ingredients-Nutrition Facts



### Search Meals-Ingredients-Nutrition Facts



# **Chapter 5: Implementation and Testing**

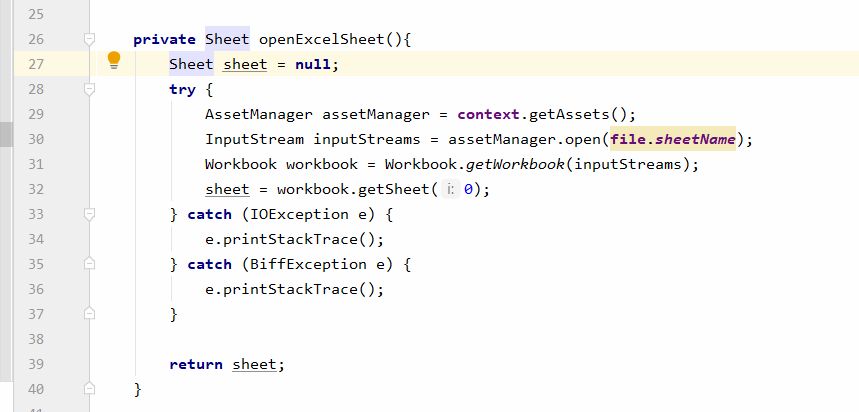
## **5.1 Implementation**

### 5.1.1 Database Management System

1-We created the database tables according to the ERD diagram using SQLite DBMS which allows database operations on android devices such as storing, manipulating or retrieving persistent data from the database.

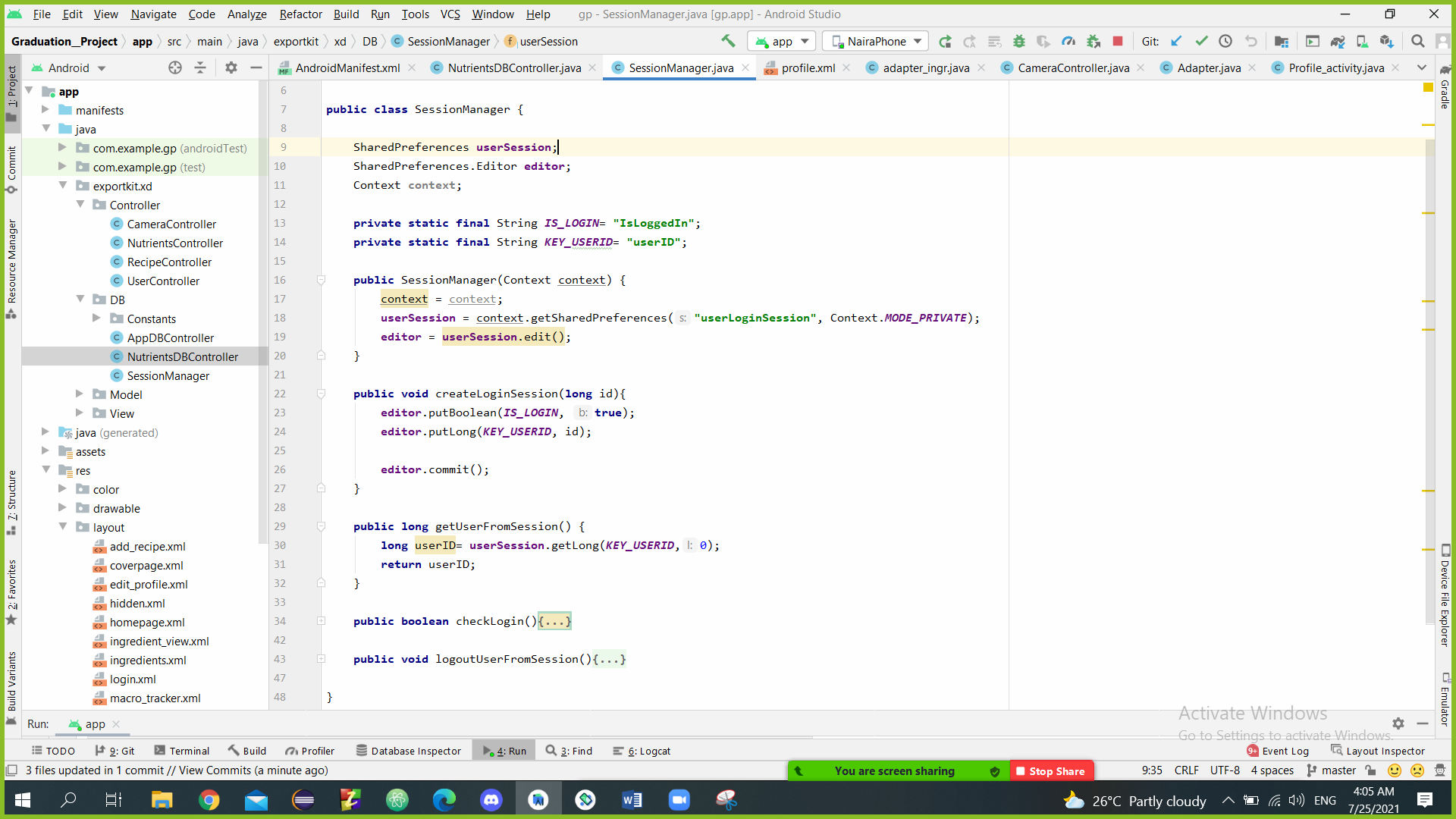


2- We collected a data set for nutrition facts as an excel sheet (.xls) that we read programmatically from assets folder.

****

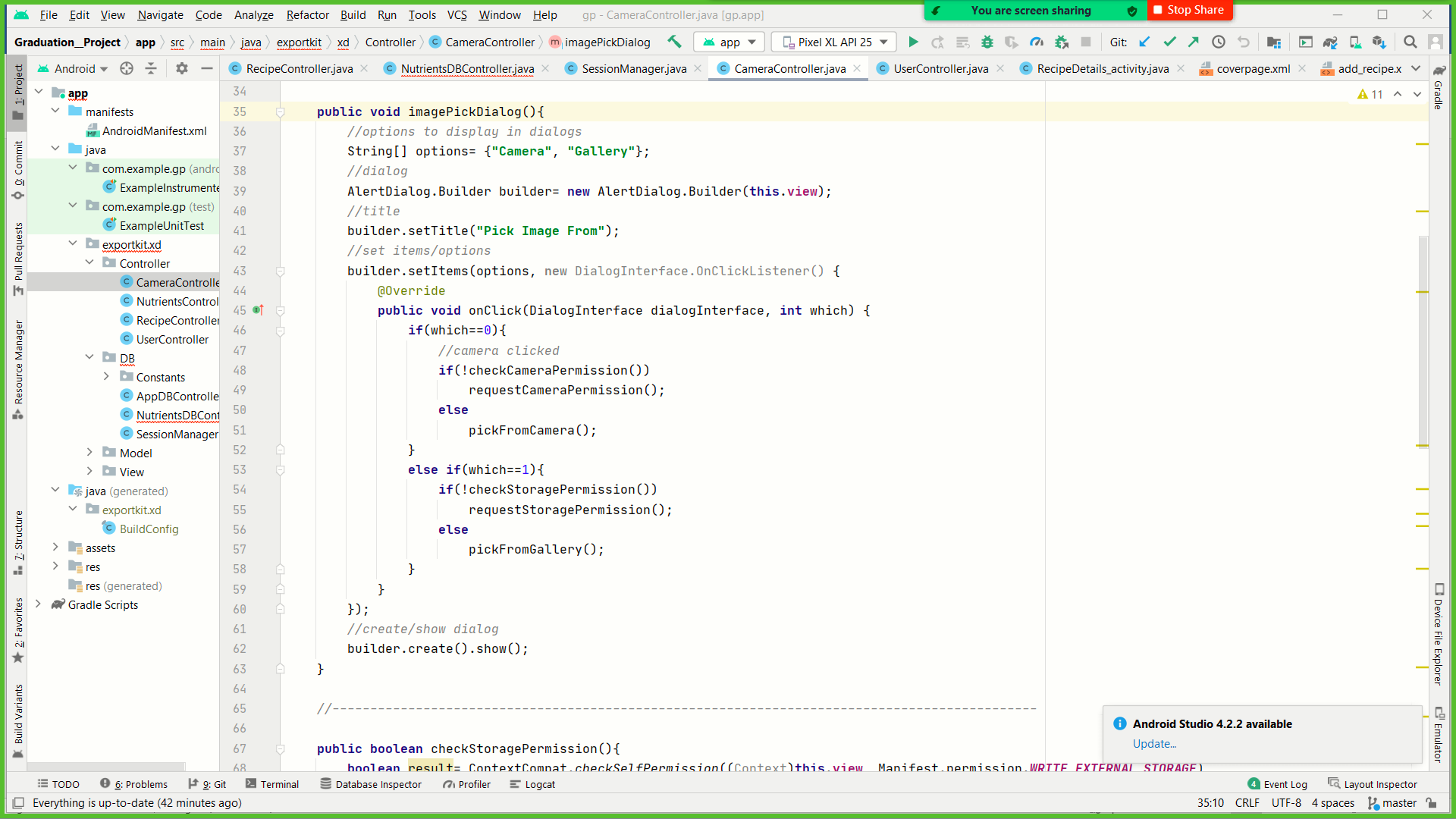
### 5.1.2 Session manager

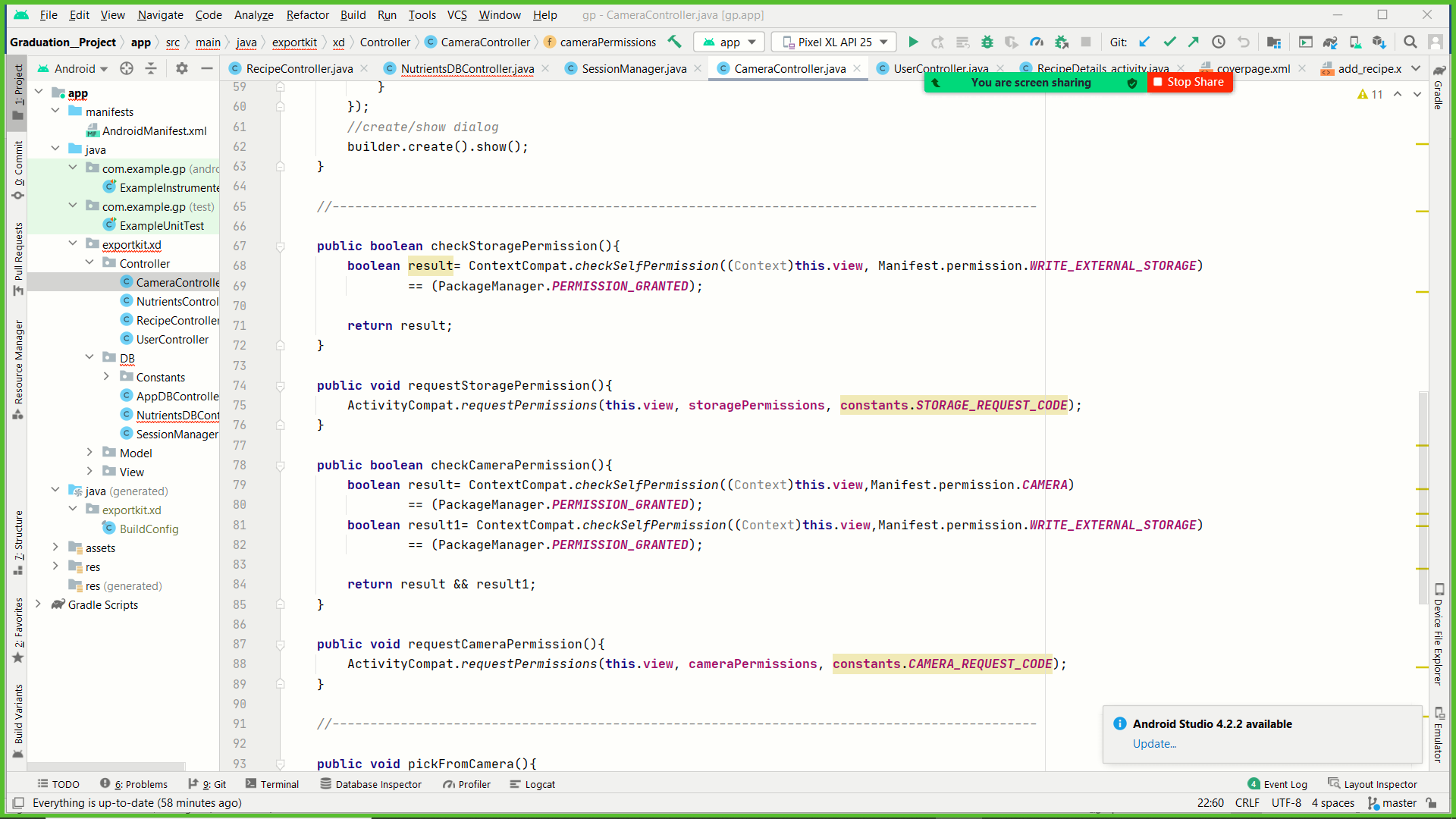
- We can manage the logged in user in session by storing its ID in shared preferences. Those will be persisted even if the application closed by the user (in case of logging out) and the user reopens it then he should redirect to our Home screen, rather than opening a cover page

****

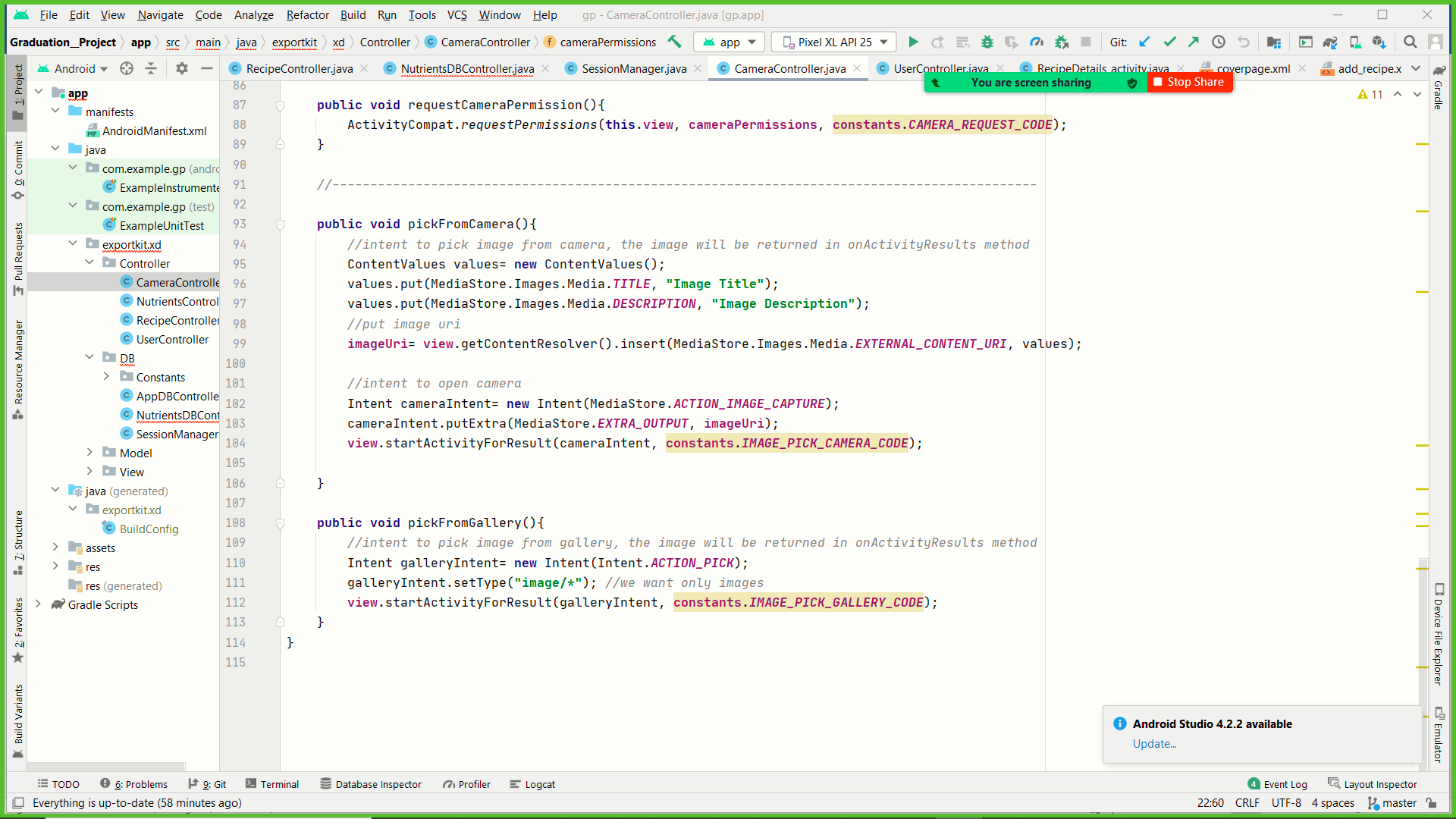
### 5.1.3 Request to upload image

User can upload image in two cases first when user uploads profile photo in edit profile screen and upload a recipe photo in add recipe screen.

* In case, the user choose to capture image by mobile camera or gallery.
* System will check the permission of access. The system will request an access if needed.



* After having the permission the system will capture the image.



### 5.1.3 Food Recognizer Model

We used this class to load ssd\_mobilenet.tflite Model, load the lables.txt and recognize the food

## Graphical user interface, application Description automatically generated

Text

Description automatically generated

## **5.2 Testing**

### Test case 1:

**Test case name:** Sign up

**Pre-condition:** sign up interface should be opened

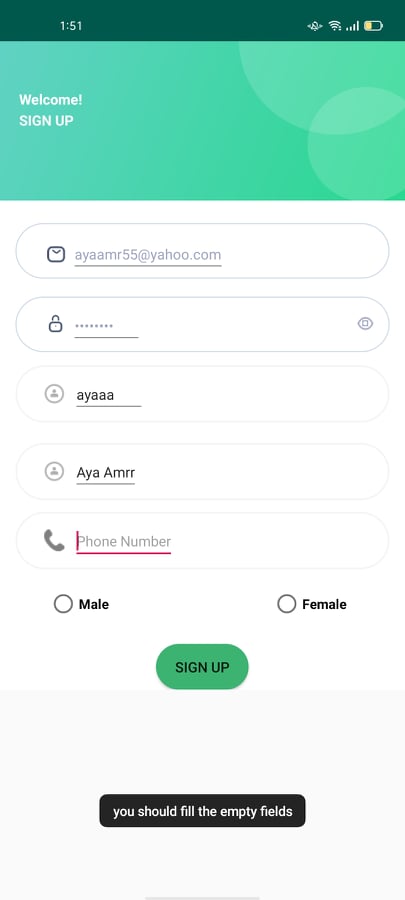
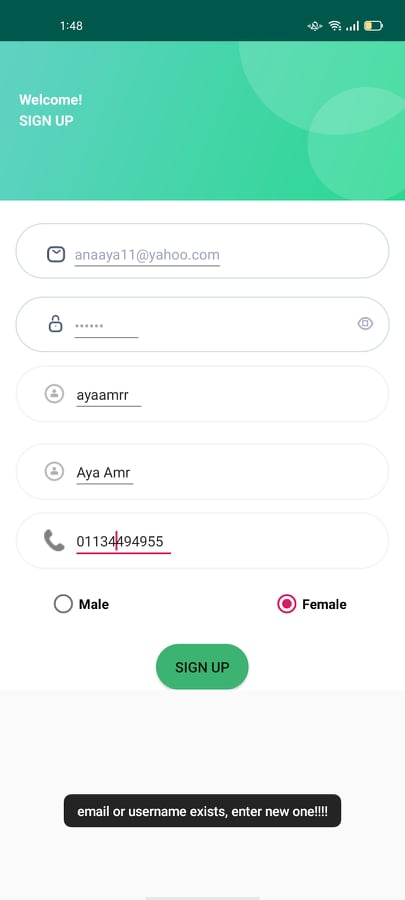
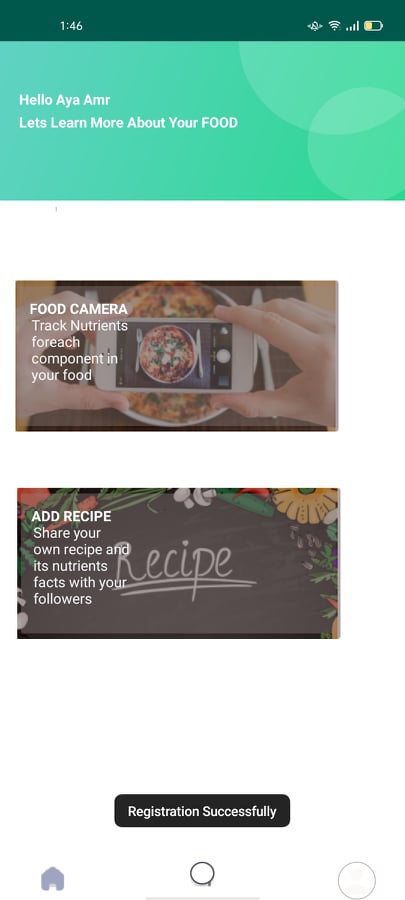
**Test case steps:**

-user will enter all the required information (Email, password, username, name, phone number, gender).

-user will click on sign up button to confirm.

**Expected results:**

The sign up is authenticated and the system will return a success message then redirect the user to the home page, if the entered email or username is used before the system will display an error message, if one or more of the required input cells is empty the system will display an error message asking user to fill the empty cells.



### Test case 2:

**Test case name**: login

**Pre-condition:** log in interface should be opened

**Test case steps:**

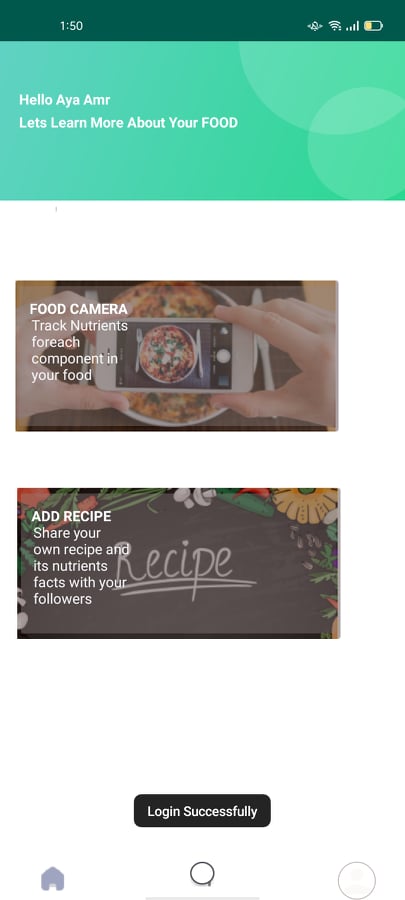
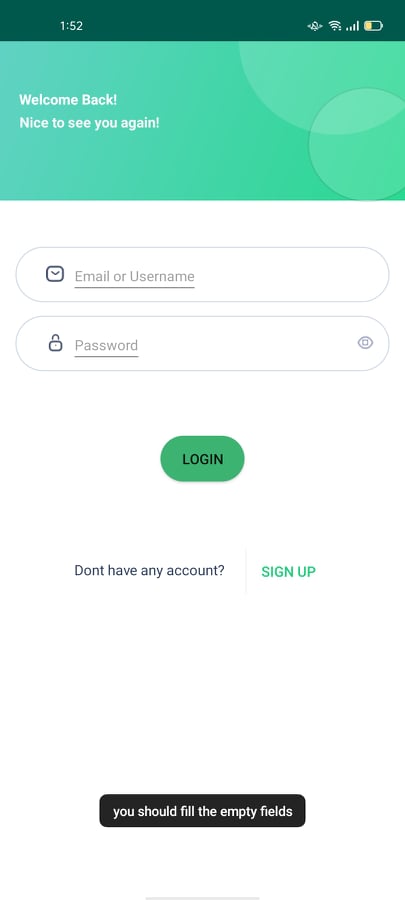
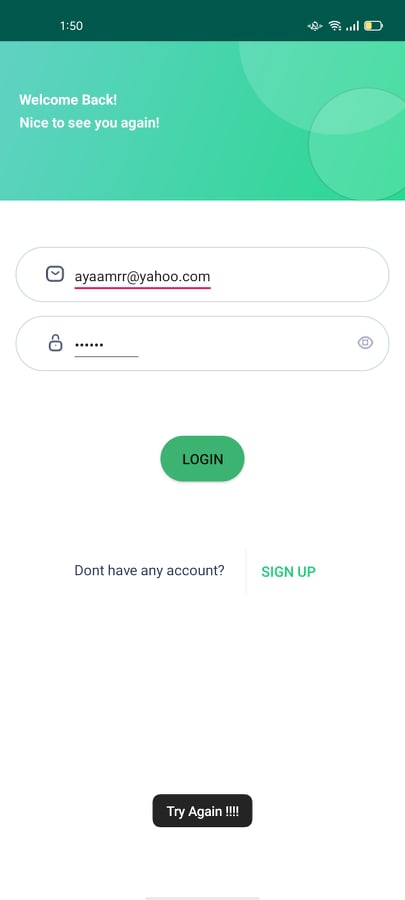
-user will enter his email and password

-user will click on log in button to confirm.

-user will click on sign up button if he does not have an account.

**Expected results:**

The log in is verified and the system will return a success message then redirect the user to the home page, if the entered email or password is wrong the system will display a try again message, if one or more of the required input cells is empty the system will display an error message asking user to fill the empty cells.

### 

### Test case 3:

**Test case name: f**irst time to open camera

**Pre-condition:** request to upload recipe photo on add recipe interface or upload profile picture on edit profile interface or scan image to detect ingredients.

**Test case steps:**

**-**user will click on camera

-user will click on to allow accessing camera every time using the app.

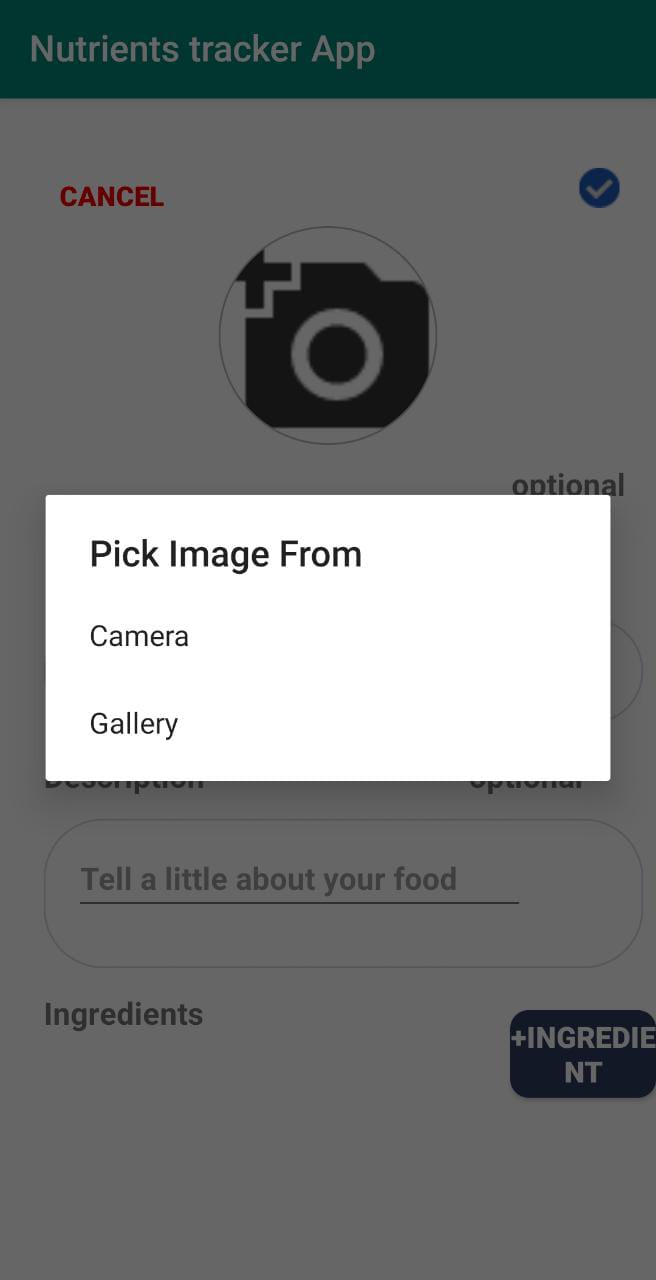
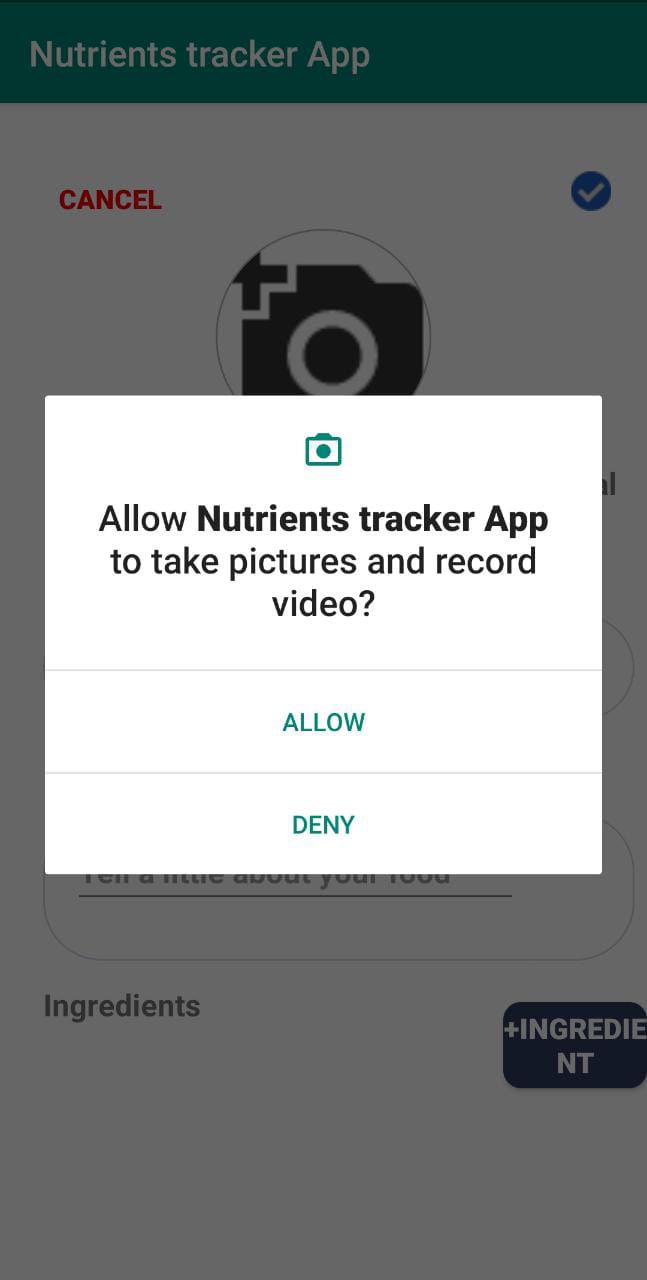
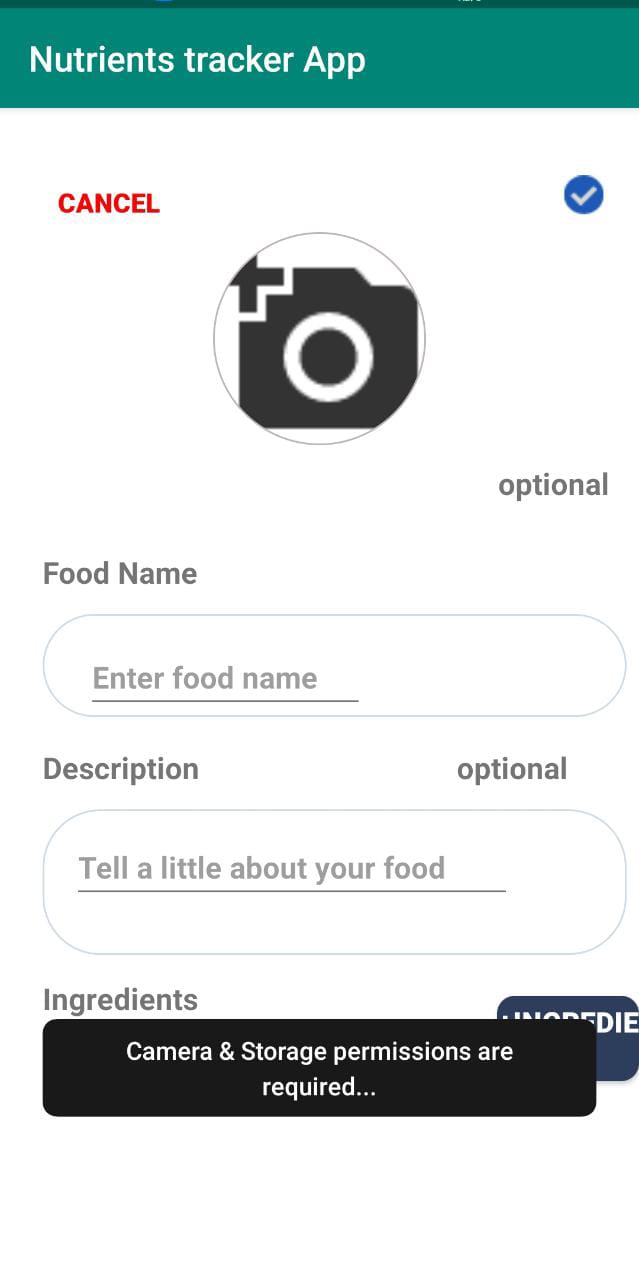
-user will click on (deny) bar to decline accessing camera.

**Expected results:**

The System will open the camera and capture a photo if the user allowed the access and choose camera.

The system will open the gallery and upload the photo if the user allowed the access and choose gallery.

The system will send an error message and fail to upload any photo if the user rejected the access.

### Test case 4:

**Test case name:** add recipe

**Pre-condition:**  add recipe interface should be opened

**Test case steps:**

-user will enter food name

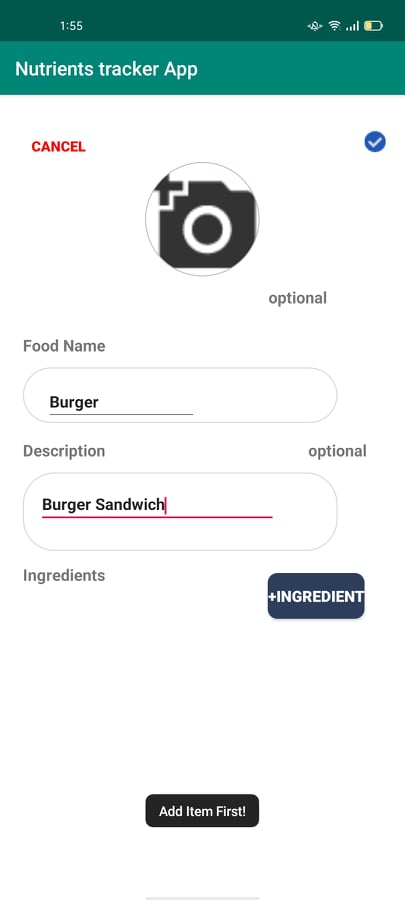
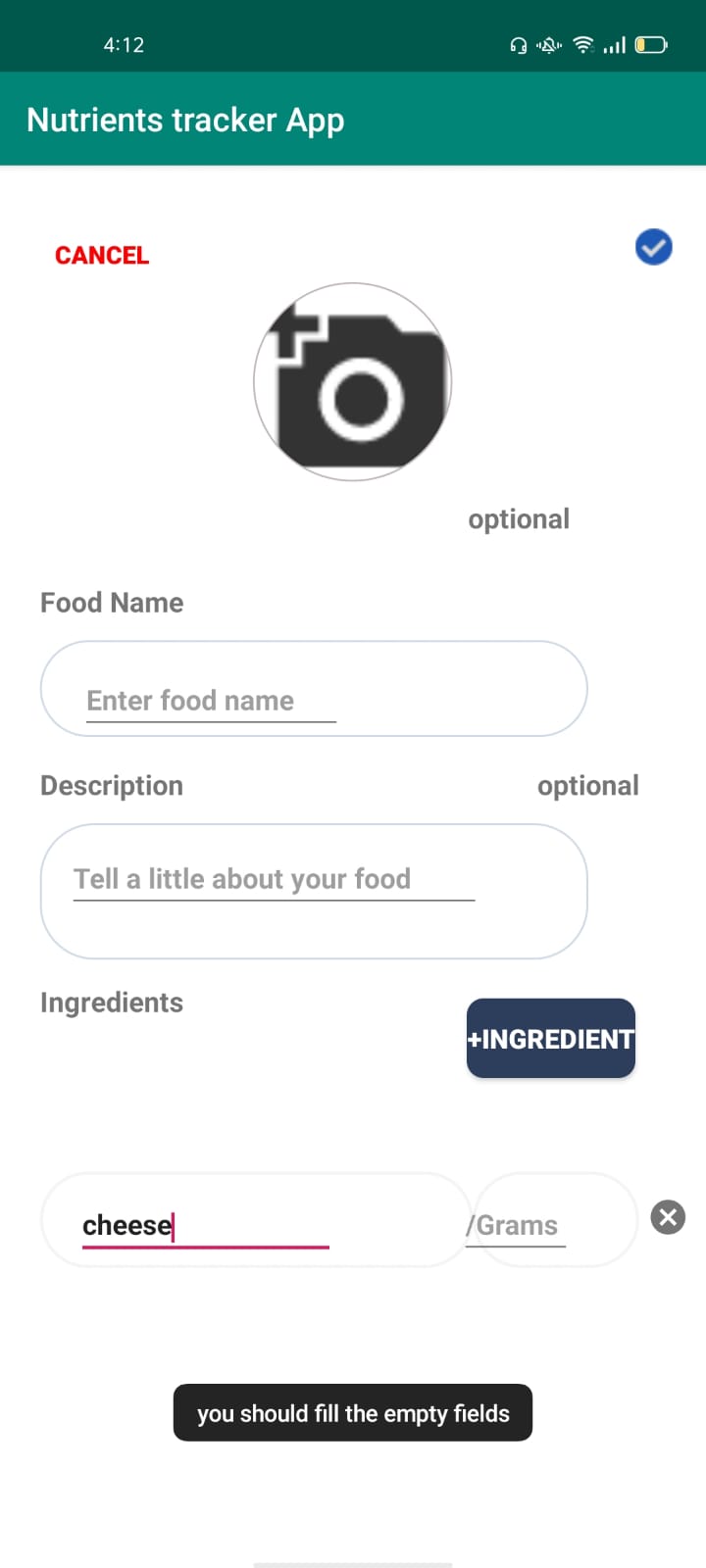
-user will enter at least one ingredient and click on add ingredient button to add an additional ingredient.

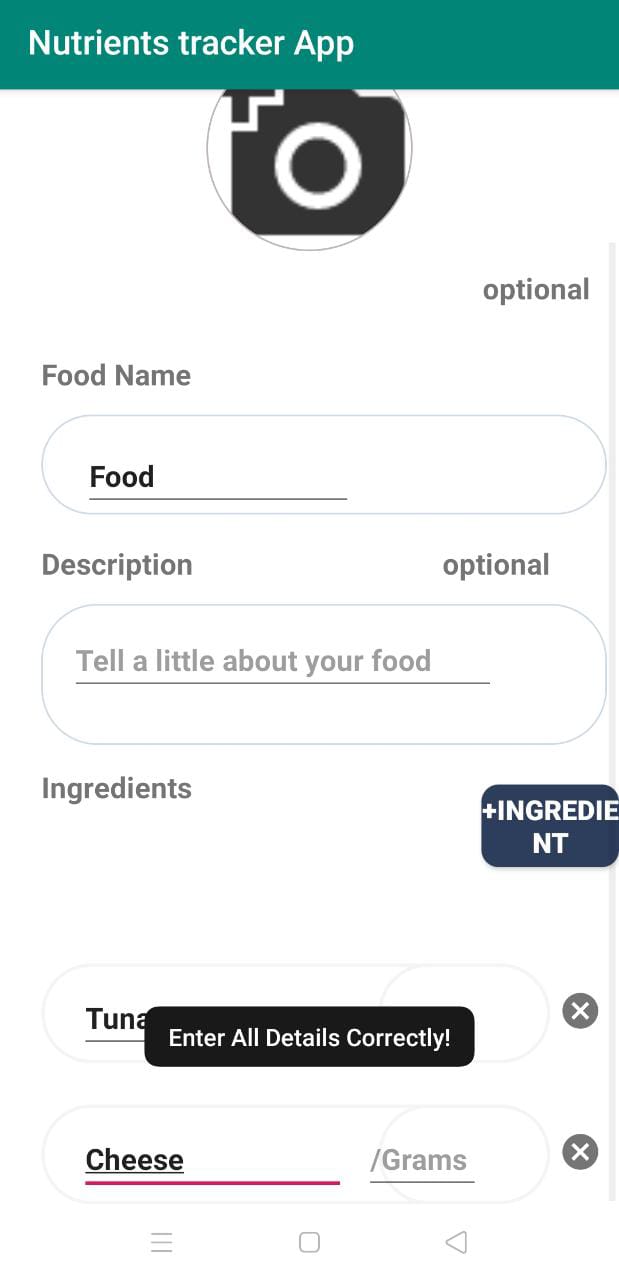
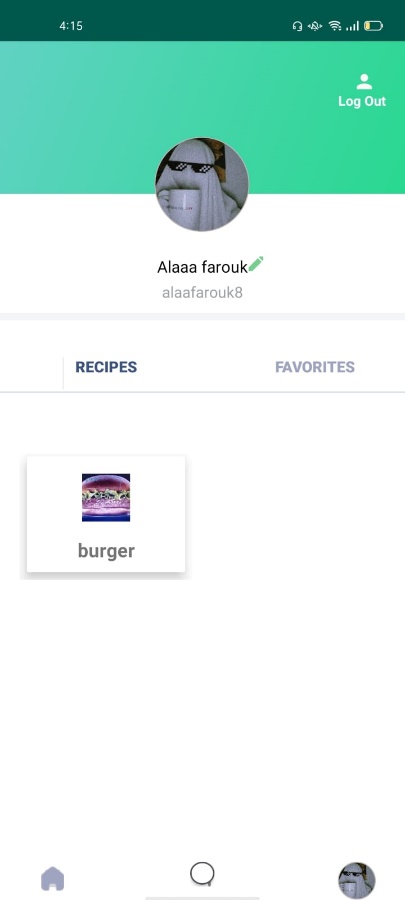
-user will enter food description.

-user will click on the check mark icon to confirm and cancel button to cancel the process.

**Expected results:**

The recipe is created and added to the user profile in the recipes list, if the user did not enter food name, the system will display an error message to fill the empty fields, if the user did not enter at least one ingredient, the system will display an error message to add item first.





### Test case 5:

**Test case name:** search user

**Pre-condition:**  search interface is opened and user button is clicked

**Test case steps:**

-user will enter the username in the search tab

**Expected results:**

If the username is correct and exist the system displays user’s profile, if the username is wrong or does not exist. The system displays a message that the user doesn't exist



### Test case 6:

**Test case name:** Scan food

**Pre-condition:** scan food interface is opened

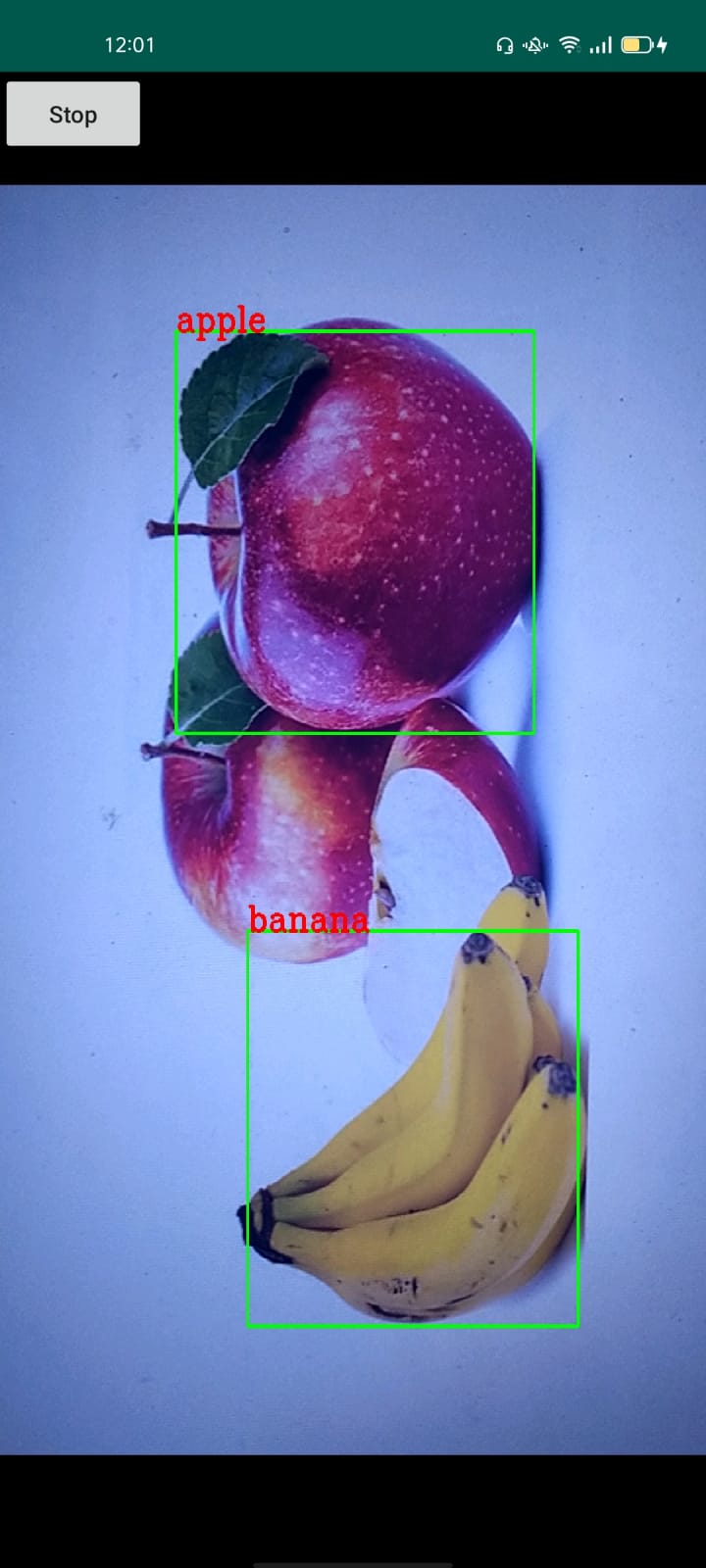
**Test case steps:**

-User will open camera live for detect ingredients

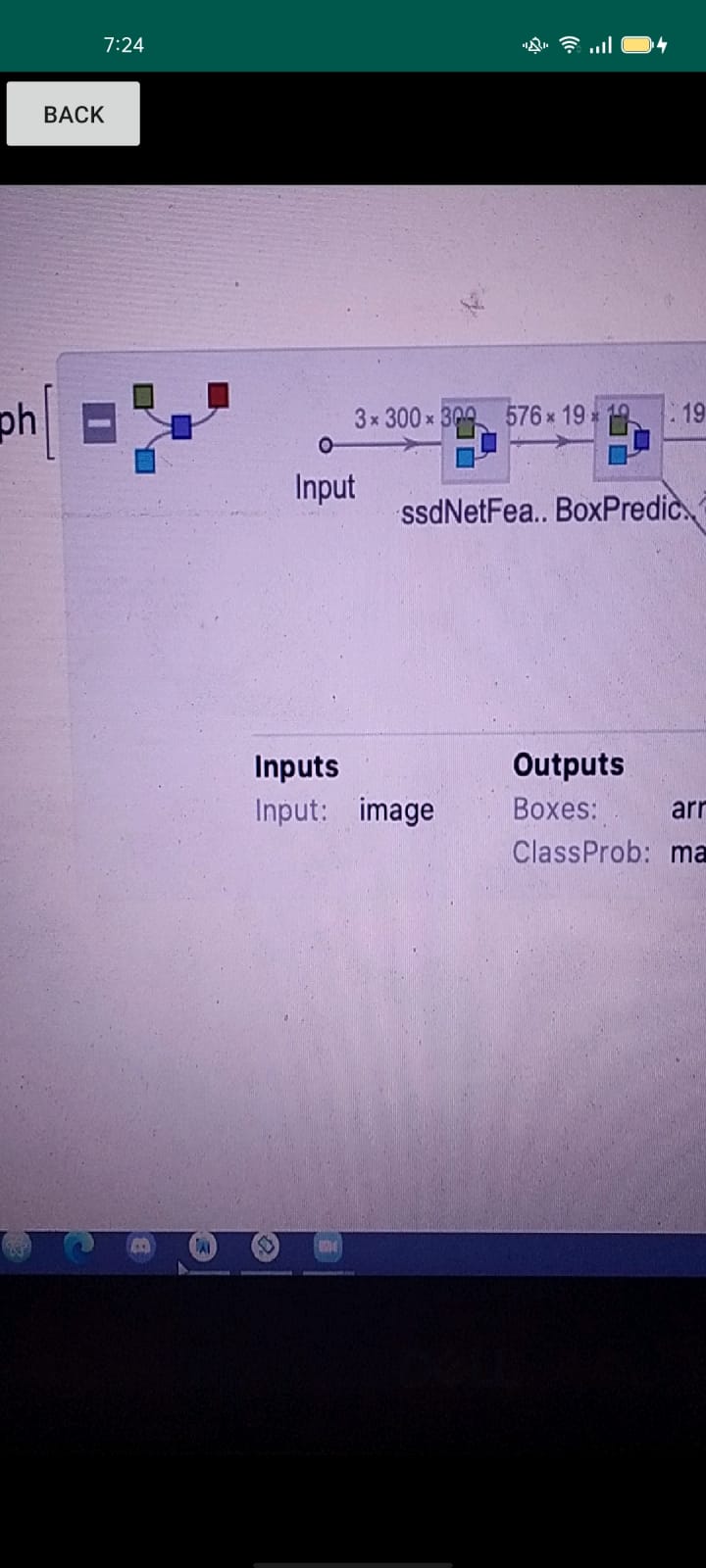
-User will click on stop button

**Expected results:**

The system will recognize the ingredients in the image if it is included in the dataset trained as in the following figure



If the image doesn’t contain food (wrong image to detect) or the ingredients is not included yet in the trained model the system will fail to recognize as in the following figure



### Test case 7:

**Test case name:** delete from favorite list

**Pre-condition:** the recipe to be deleted is in the favorite list.

**Test case steps:**

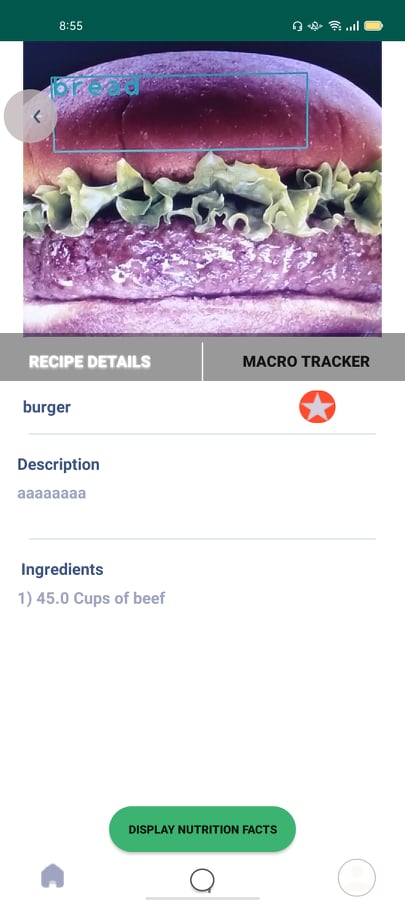
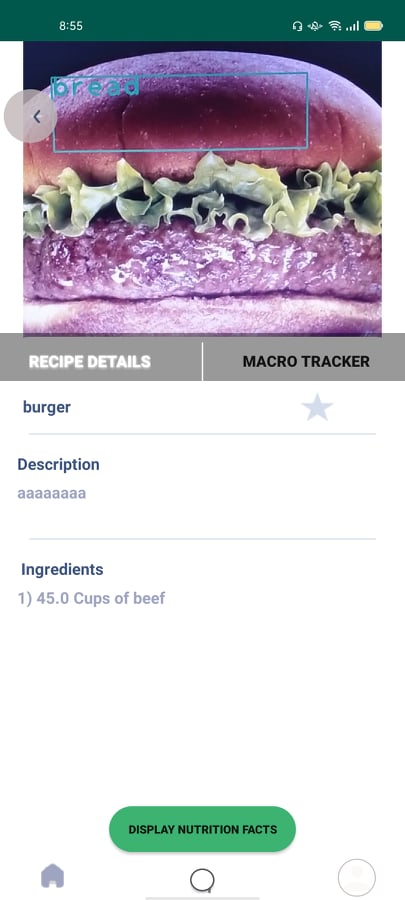
-user will open his recipe list

-user will select a recipe then open the recipe details

-user will click on unfavorite icon.

**Expected results:**

The system will remove the selected recipe from favorite list.

****

# 

# **References**

<https://github.com/NSTiwari/YOLOv3-to-TensorFlow-Lite-Conversion/blob/main/YOLOv3%20to%20TensorFlow%20Lite%20Conversion.pdf>

<https://blog.roboflow.com/tips-for-how-to-label-images/>

<https://developer.android.com/training/data-storage/sqlite>

<https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase>

<https://www.healthline.com/nutrition/5-best-calorie-counters#TOC_TITLE_HDR_1>

<https://github.com/jakkcoder/training_yolo_custom_object_detection_files>

<https://www.kdnuggets.com/2018/09/object-detection-image-classification-yolo.html>

<https://pysource.com/2020/04/02/train-yolo-to-detect-a-custom-object-online-with-free-gpu/>

[Waterfall vs. Agile: Which Methodology is Right for Your Project? (seguetech.com)](https://www.seguetech.com/waterfall-vs-agile-methodology/)

<https://github.com/pjreddie/darknet>

<https://github.com/tzutalin/labelImg>

<https://www.journaldev.com/9438/android-sqlite-database-example-tutorial>

<https://developer.android.com/reference/android/widget/GridLayout>

<https://developer.android.com/guide/topics/ui/layout/recyclerview>

<https://smallacademy.co/display-recyclerview-in-gridview/>

<https://abhiandroid.com/ui/tablelayout>

<https://stackoverflow.com/questions/5810084/android-alertdialog-single-button>

<https://stackoverflow.com/questions/24095223/android-linearlayout-add-border-with-shadow-around-a-linearlayout>