

HELWAN UNIVERSITY
Faculty of Computers and Artificial Intelligence
Information Systems Department



Prepared by:

- [Omar Essam El Din Abbas (20180374)]
- [Amr Ramadan Zeinhom Sayed (20180400)]
- [Mahmoud Adel Hassan (20180564)]
- [Amina Ramadan Ahmed Ismael (20180137)]
- [MennatAllah Sroor Abd El-Aziz (20180622)]
- [Yasmeeen Ahmed Mohamed Ali (20180688)]

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in
Computers & Artificial Intelligence, at the **Information Systems** Department, the Faculty of
Computers & Artificial Intelligence, Helwan University

Supervised by:

- [Dr. Safaa Hegazi]

June 2022



H. Partner

- The best weight for you -

**Faculty of Computer and Artificial Intelligence
Helwan University.
Information System Department.**

Supervised By:

DR. Safaa Hegazi

Team Members:



Omar Essam



Amr Ramadan



Mahmoud Adel



Amina Ramadan



MennatAllah Sroor



Yasmeen Ahmed

Acknowledgement

First, we want to thank our supervisor **Dr. Safaa Hegazi** for his support, patience, kindness, unlimited help, and encouragement. He was always there for guiding us. He had done his best to support our project and our ideas, helped us more, and lifted our spirits.

Second, we want to thank our Parents for their patience and support. They tried so hard to be in our sides, encouraged us, and prayed for us.

Third, for our faculty for providing the suitable environment that leaded us to represent the best image that computer science graduates of Helwan University are supposed to represent.

Finally, we want to mention God's success that guides us to all of this.

Thank You All.

Table of Contents:

Chapter 1: Introduction.....	9
1.1 Project Overview.....	10
1.2 Objectives.....	10
1.3 Purpose.....	11
1.4 Scope.....	11
1.5 General Constraints.....	11
 Chapter 2: Project “Planning and Analysis”	13
2.1 Project Planning.....	14
2.1.1 Feasibility study.....	14
2.1.2 Estimated Cost.....	15
2.1.3 Gantt Chart	16
2.2 Analysis and Limitations of existing system.....	17
2.3 Need for the new system	19
2.4 Analysis of the new system.....	21
2.4.1 User Requirements	21
2.4.2 System Requirements	22
2.4.3 Domain Requirements	23
2.4.4 Functional Requirements	23
2.4.5 Non-Functional Requirements	25
2.5 Advantages of the new system	27
2.6 Risk and Risk Managements	28

Chapter 3: Software Design	29
3.1 Design of Database	30
3.2 Class Diagram	31
3.3 Use-case Diagram.....	32
3.4 Sequence Diagram.....	55
3.5 Activity Diagram.....	70
 Chapter 4: Implementation	 87
4.1 Software Architecture	88
4.2 Flowchart	89
 Chapter 5: Testing	 91
5.1 Unit Testing	92
5.2 Integrated Testing	92
5.3 Additional Testing	94
 Chapter 6: Results and Discussion	 101
6.1 Results	102
6.1.1 Expected Results	102
6.1.2 Actual Results.....	102
6.2 Discussion.....	103

Chapter 7: Conclusion104

Chapter 8: Future Work.....106

Bibliography.....108

List of Tables

- Table 1: Time.....	11
- Table 2: Test Cases.....	94

List of Figures

- Figure 1: Gantt Chart	16
- Figure 2: Design of Database	30
- Figure 3: Class Diagram.....	31
- Figure 4: Use-case Diagram	32-54
- Figure 5: Sequence Diagram.....	55-69
- Figure 6: Activity Diagram.....	70-86
- Figure 7: Admin Flowchart.....	89
- Figure 8: User Flowchart.....	90

Abstract

Our project is about healthy nutrition. It is a mobile application that tries to choose the best method of nutrition to achieve the goal that the user chooses and seeks to reach, whether that goal is weight gain for the ideal weight, weight loss or maintenance.

In fact, there are many people who suffer from the problem of proper nutrition and the healthy food they need and how to obtain it, so we thought to make this easier for people through our application

H.Partner, which helps the user to achieve his goal by knowing the user the number of calories he should eat and protein, fats and carbohydrates that must be eaten daily, and this is calculated by the Mifflin equation, and he can also buy the health products he needs.

Our application can help the user to know the nutrition information that is in his food by taking a picture of the food, so the application determines the number of calories, protein, fats, and carbohydrates in the food.

Our app help you to track your daily water intake to reach the healthy goal.

Chapter 1

Introduction

In this chapter, we will discuss the project's objectives, scope of the project, work needed to build the project, and overview of the whole document.

1.1 Overview

Nutrition plays a great role in our daily life. The food or liquids affect our body and health because each food or liquid contain particular nutrition which is very necessary for our physical and mental growth. A particular level of any particular nutrition is essential for our body. So, we should know that what food we have to take, how much and what type of nutrition contain a particular food. Whenever we take any food or nourishing liquids, our body digests and absorbs the simple but essential minerals, vitamins, fats, proteins, carbohydrates, fats and water from these food or nourishing liquids and converts it into the bloodstream and energy that help our body to grow and keep it healthy. The nutrition value is more important for any individual's health. The food or liquids whenever we take it affect our body and health as well both. So, it is very important that we should be more aware of the foods or liquids whatever we take in our daily life. Many diseases occur only due to wrong diet. Some certain diet may itself cause some disease or alter the course of a known disorder such as diabetes, heart or kidney disease.

1.2 Objectives

The user enters his age, gender, weight, height, activity, then the application calculates his daily calorie needs (fats, proteins, and carbs). The user can buy supplement products. There are many healthy recipes for users to make healthy food. The user can take photo of any food and app detects the image then give how many calories, fats, proteins, and carbs of meal.

1.3 Purpose

The main purpose of nutrition system, it helps people to protect their healthy and make good body building it helps people to know how many calories, fat, protein, carbohydrate, sugar, fiber, and certain vitamins and minerals their need to take it.

1.4 Scope

The user can register, login, add and update her information, doing nutrition system, buy and canceling product, add, update, add feedback, search in our system, user can take photo to any food too and the system can detect what the food in this photo and describe the protein, calories, fat, carbohydrate, inside the food. Admin can do login, add product, update product, delete product, add recipe, update recipe, delete recipe, receive feedback, delete users.

1.5 General Constraints

- **Time:**

Task	Time
Searching for ideas and choosing one	1 month
Designing Diagrams	1 months
Searching for technologies and studying them	3 months
Time of each sprint	2 weeks
Implementing the whole project	3 months
Editing diagrams and Testing	1 month

- **Assign Tasks:**

We assigned the tasks based on:

- ✓ The team members physical ability.
- ✓ The different skills of each member (Testing – Analysis – Programming).
- ✓ The ability of searching for ideas, technologies, and techniques.
- ✓ Knowledge and background about the project idea and the required techniques.
- ✓ Agile planning with divided sprints.

- **Gathering data:**

- ✓ The resources of the technology we want to use (Flutter).
- ✓ A tutorial that explains how can we use a firebase in flutter?
- ✓ How to build a system architecture diagram?

- **Manage behavior training:**

- ✓ Limitations in understanding and getting the information and points quickly.
- ✓ Limitations in selecting a meeting time appropriate to advisor.

Chapter 2

Project “Planning and analysis”

In this chapter we are going to discuss and go deeper in how we plan the project and show the steps and the instructions that we have followed to plan the application.

2.1 Project planning

2.1.1 Feasibility Study

A Feasibility study is used to determine if a business or a specific project is achievable, so for determining the achievability of our project we'll go deeper in the following points:-

1-Market Analysis: -

- We can make money from our app using ADS.
- This application helps people finding which nutrition plan suits them
- H.partner app makes users request orders in more quickly by just few clicks.
- H.partner mobile app isn't in just specific limited places.
- H.partner mobile app recommends recipe to users and show data for this food
- H.partner mobile app makes user take photo for eat and show its data

2-Operational Analysis: -

- In our project we provide some rules and services to our employees such as:
 1. Environment suitable for work.
 2. Commitment to the deadline for delivering tasks.
 3. Good salaries.
- We deliver our service by using social media, offline marketing, and advertisements
- We plan to support and retain our customers by providing feedback and promotions offers.

2.1.2 Estimated Cost

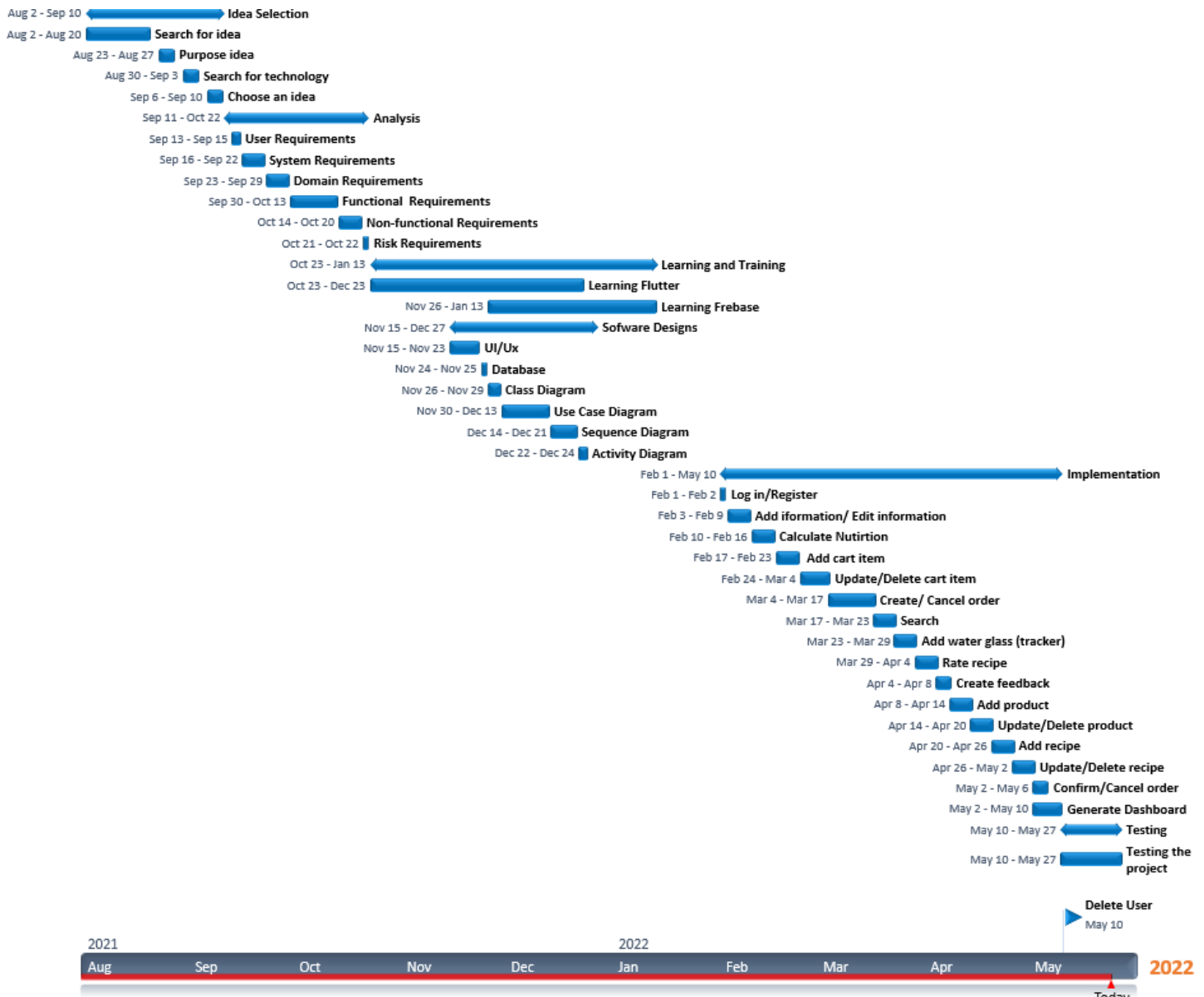
A cost estimate is approximation of the cost of a program, project or operation. The cost estimate is the product of the cost estimating process and our estimated cost for this project comes as following: -

- Meeting fortnightly a public workspace at the first three months:

$$(120 \text{ LE in one week} * 2 \text{ weeks in month} = 240 \text{ LE})$$
$$(240 * 3 = 720 \text{ LE})$$

- We spent a lot of time and effort in self-studying about our technologies and search for the best packages and service to be included in our app

2.1.3 Gantt Chart



2.2 Analysis and Limitations of existing system

1. ScanFood

The Calorie counter ScanFood app is designed specifically for those who monitor their food intake and strive to achieve their goals. Everyone's goal is different: some want to achieve weight loss, others, weight gain; others still wish to maintain their current body weight. With Calorie counter ScanFood, you will be able to scan barcodes of products sold in store, and the application will independently determine their caloric value, as well as the number of proteins, carbohydrates and fats contained in them. All you have to do is to indicate the approximate weight of the consumed food, and the application will independently calculate all the data.



2. FatSecret

Fatsecret helps users to keep track of their food, exercise and weight, using the world's highest quality food and nutrition database and connect with a global community of people looking to make a change for the better and start losing weight and achieve your goals the healthy way.



3. Keto diet & meal plans

Keto diet can be an integral part of your healthy life. It can accelerate body fat burn rather than carbs. Keto diet tracker consists of carefully picked healthy keto recipes that are high fat, adequate-protein, low carb keto recipes. It can monitor net carbs, sugar ketones, lazy keto, diabetic carbs. A healthy Mediterranean keto diet plus keto workout can bring tremendous change in your lifestyle. Keto recipes free is a wholesome pack that can help you practice a healthy ketogenic diet plan, check macros, blood sugar ketones, ketogenic carb counter etc. Low carb keto recipes free also provides access to awesome ketogenic diet plan even after 14 days.



4. Samsung Health

Samsung Health has various features to help users manage their health. As the app allows you to automatically record many activities, creating a healthy lifestyle is easier and simpler than ever.

Samsung Health helps users to check various health records on the Samsung Health home screen. Users easily add and edit the items that they want to manage such as daily steps, activity time, and body weight, simply by long pressing the screen. S Health doesn't have a barcode-scanning feature that allows users to scan an item's nutrition label. Instead, the app compares users' diets to nutritionist recommendations, and classifies them as either low, average, or high. The app significantly underestimates calcium, iron, and vitamin C levels in many foods, but gives accurate estimates for carbs and fat.

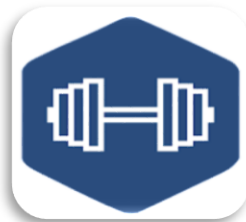


5. wger Workout Manager

Wger is developed a 100% free digital health and fitness tracker app for you, sized down to the most relevant features to make your life easier.

Wger is an Open Source project and all about:

- User's Body
- User's Workouts
- User's Progress
- User's Data



2.3 Need for the new system

The new system will help all people not some of them because the new system will include 3 goals (how to keep your weight, how to lose weight, how to gain weight). Other apps only serve people that want to lose their weight and some of them doesn't work well like Samsung Health, and Scanfood.

The weakness points of other apps are that many of apps helps people to lose their weight only and there are many of people that want to gain weight or keep that. Not all of application include fully healthy diet for free like FatSecret. There are some of apps that have many bugs like wger Workout Manager, ScanFood, and Samsung Health.

Our System based on:

Basal metabolic rate

Basal metabolic rate is the number of calories your body needs to accomplish its most basic (basal) life-sustaining functions.

How to estimate your BMR

One popular way to estimate BMR is through the Mifflin equation

Mifflin equation

The Mifflin-St Jeor equation is a widely used tool to determine the resting metabolic rate [RMR], which is defined as the number of calories burned while the body is in complete rest. RMR is also known as resting energy expenditure [REE]. The equation was developed by MD Mifflin and ST St Jeor and first introduced in a paper published in 1990.

Formula

- Females: $(10 \times \text{weight [kg]}) + (6.25 \times \text{height [cm]}) - (5 \times \text{age [years]}) - 161$
- Males: $(10 \times \text{weight [kg]}) + (6.25 \times \text{height [cm]}) - (5 \times \text{age [years]}) + 5$

Multiply by scale factor for activity level:

Sedentary *1.2

Lightly active *1.375

Moderately active *1.55

Active *1.725

Very active *1.9

2.4 Analysis of the new system

2.4.1 User Requirements

Mandatory requirements:

User should fill in personal data such as name, email, password, age, gender, weight, height and level of physical activity.

Desirable requirements:

1-The user needs to use camera phone to know about any food information and app detects the image then give how many calories, fats, proteins and carbs of meal.

2-The user uses the app with him at any place and any time so our app should be available in any place and any time.

3-When user finish his meal and press on the meal in the system, the system will deduct its calories form the main calories that are available to use today.

4-The user can shop in our system and search about supplement products

Optional requirement:

The user can see all recipes, select one of them, he can see the number of calories, carbs, protein, fat, ingredients, and directions.

2.4.2 System Requirements

1. Android

As the application works on the least versions of the android ice cream sandwich because we use a working framework which is responsive to the lowest versions of Android.

2. IOS

The application works on IOS 9 as we use Flutter Framework that support android and iOS and it is the minimum stable version working on it.

3. Camera

This app needs a mobile camera as it is very useful in taking pictures for listings to make it easy to the user to choose the suitable listing and it does not matter which version of camera you will use, but it will be awesome if you use a good one.

4. Internet Connections

This app uses a cloud server to store data, so you need to be connected to the internet to fetch and view it. A part of it considered a social media network so the whole data will be on the server [firebase] also you need it to enable Storage service to upload photos.

5. Storage space

To keep the user's data in addition to the application itself, we need a fine storage

2.4.3 Domain Requirements:

- Each user register in our system, the system ensures that their data won't be lost if they change their device or delete and then re-install our application.
- The system must ensure that do a full nutrition plan to any user based on his goal and his eating habit and the user can track calories, carbs, protein, and fat by user's dashboard.
- The system should push notifications deliver by information on progress toward the current goal and motivate users to keep moving toward their goals by reminding them.
- All products from shopping list must comply with food quality and safety standards.
- All due taxes must be collected during payment.
- It is necessary to make sure that the correctness of the food recipes.

2.4.4 Functional requirements

1. When a user registers in the system, the system shall create new account and record in the database his first name, last name, e-mail, password, gender, his weight, his height, his level of physical activity.

2. If a user already has an account and enters wrong inputs when he login, the system pushes error message and asks user entering correct inputs.
3. After user logs in the system, the system will show a list of goals (3 goals; lose weight, keep weight, gain weight).
4. The system shall insure that a user has eat foods' meals which the system introduces them for him by pressing on the current meal and the system will deduct the calories of this meal from the main available calories that should user eat.
5. The system also reminds the user to drink water should drink 8 glasses of water a day while the max he should drink 16 glasses of water per day.
6. If the user wants to take photo of his food, the system will help him to take this photo and detect what are the calories of it and the number of protein, carbohydrates, and fat in it.
7. The system allows the user to buy a supplement product or its ingredients or dietary supplements from the system's shopping list.
8. The system helps users to make their healthy food by themselves by the system's recipes lists, every recipe includes the photo of food, its ingredients, way to make, its calories, protein, carbohydrates, and fat in it.
9. Admin should manage the orders from users to buy products by insuring the data of users and the deadline of orders that user.

10. Admin can add or delete any product in shopping lists and update its price.
11. Admin can add or delete any recipe of recipes lists.
12. Admin should read the feedbacks from users.

2.4.5 Non-Functional Requirements

○ Look and feel

- 1- The product shall use three colors
- 2- The product should have many of pics
- 3-The product shall comply with corporate branding standards
- 4-The product should be simple
- 5-The product shall be attractive to an all audience
- 6-The product shall appear authoritative
- 7-The product shall be intriguing

○ Usability

1. Ease of learning. The system must be easy to learn for both novices and users with experience from similar systems.
2. Task efficiency. The system must be efficient for the frequent user.
3. Ease of remembering. The system must be easy to remember for the casual user.
4. Understandability: The user must understand what the system does.
5. Subjective satisfaction. The user must feel satisfied with the system.

- **performance**

- 1-The product shall load each page within 1 seconds
- 2-The product shall calculate the calorie in 3 sec
- 3-The product shall handle up many of users on the same time

- **Operational**

- 1-The product shall be operating when lighting is poor or at night
- 2-The product shall conserve battery life

- **Maintainability**

- 1-The product should be use on many platform
- 2-The product shall be translated into various foreign languages
- 3-The product should add many classes for users

- **Cultural**

- 1-The product shall be written in the United States
- 2-The product shall not use any terms or icons that might possibly offend anyone on the planet
- 3-The product shall not display religious symbols or words associated with mainstream religions

- **Security requirements:**
The nutrition system should guarantee the highest level of integrity and security of the data stored by the system. The system should only allow authorized persons to access the data through a secured page through which administration users or customers themselves access and enter their password and username to login and view users' page.
- **Reliability:** The system should be available to all user at any time. In a case of unavoidable circumstances that lead to system malfunctioning, it should be restored within 48 hours.
- **Robustness:** Time to restart after failure is 48 hours. Probability of data corruption on failure is little. Our application is able to handle error conditions gracefully, without failure. This includes a tolerance of invalid data, software defects, and unexpected operating conditions.
- **Portability:** Our application take advantage of portable to other systems, such as android or iOS.

2.5 Advantages of the new system:

1-our app has a push notification to reminding them about meals and others things and to motivate users to keep going.

2-system can calculate calorie in user's meals by scan photo and system make analysis on it.

3-system can calculate calorie for each user and how much takes for day by some info. like weight loss goal.

4- The system has shopping list to allow users to add groceries to a shopping list manually

5-system has Recipe book section to offer healthy recipes

6- The user can take photo of any food and app detects the image then give how many calories, fats, proteins and carbs of meal.

2.6 Risk and Risk Managements

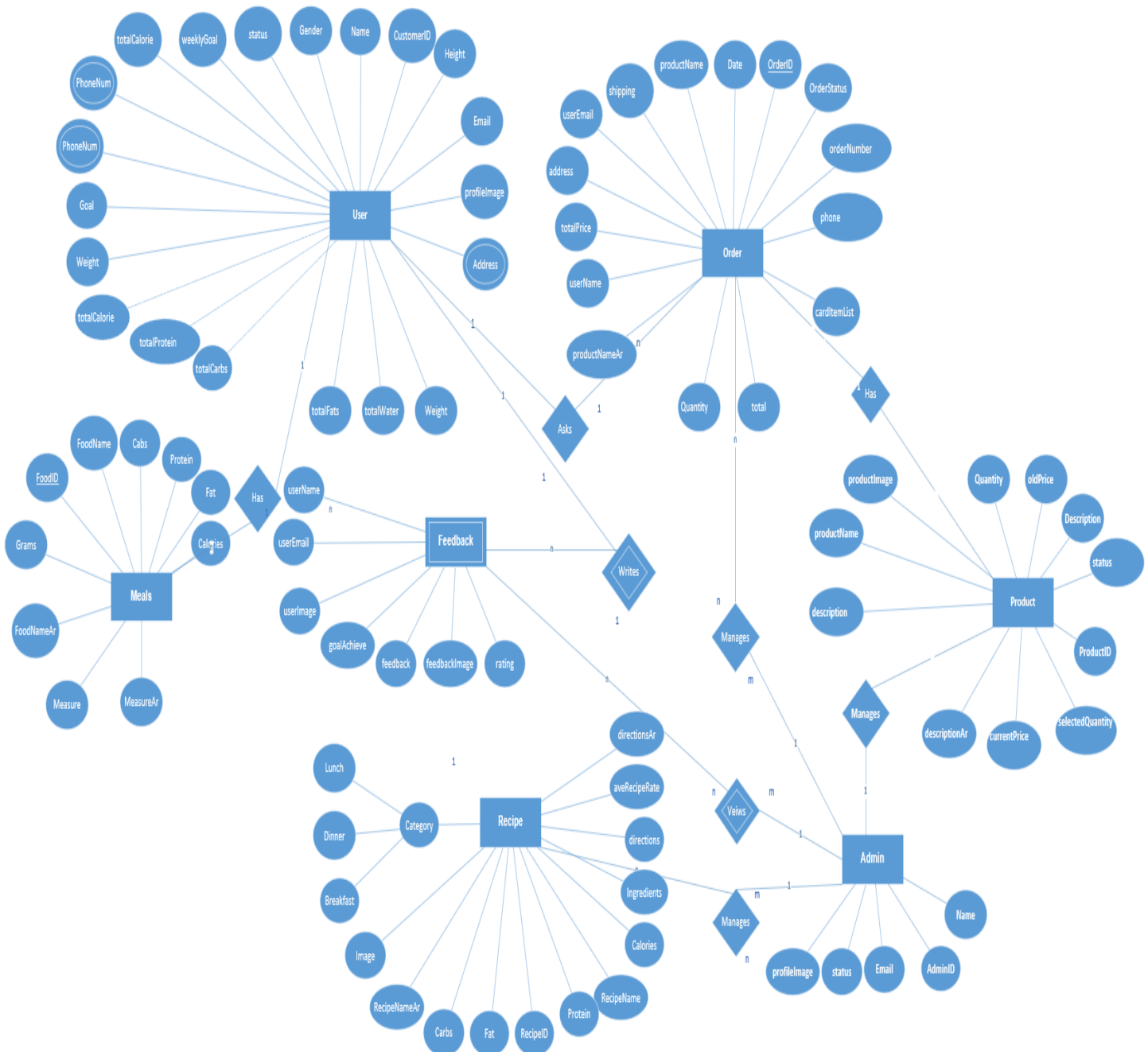
- After the user register in application the system doesn't store his data and doesn't create new user.
- When user login in application and the system doesn't identify the user.
- The system is hacked, and user's data is stolen.
- The system couldn't calculate calories for user and doesn't make user to reach his goal (big risk).
- The system couldn't recognize the photos and doesn't show accurate nutrition information.
- When user buy items from shopping list the system makes sure that the personal data of user don't be steal or lost.

Chapter 3

Software Design

In this chapter we are going to discuss and go deeper in Settle mobile application's design and present its diagrams and database.

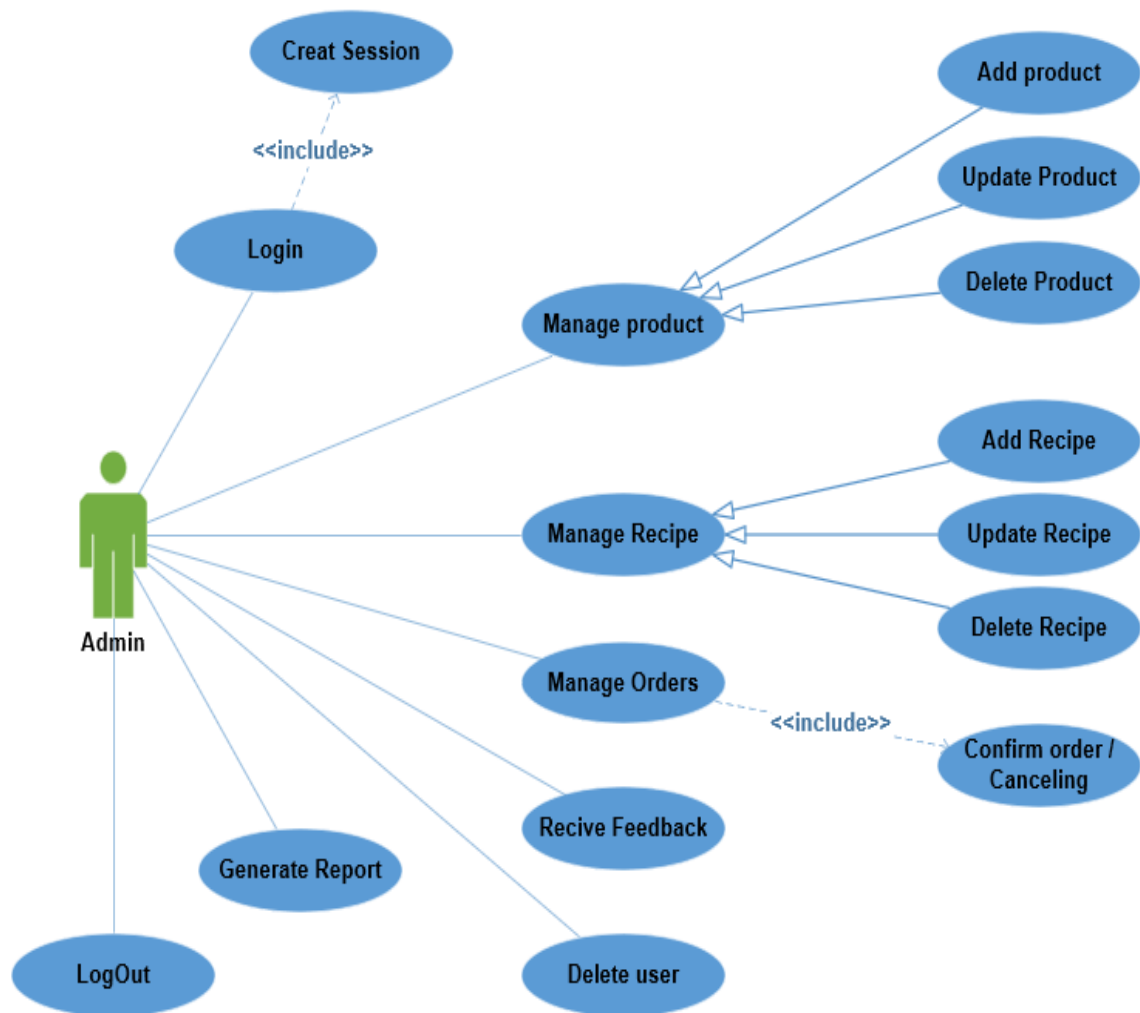
3.1.1 Design of database (ERD):



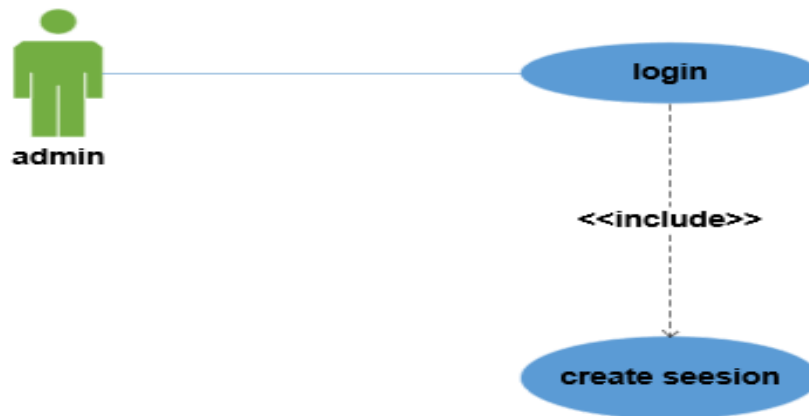
[illegible]

3.2 Use-case Diagram:

- Admin



1. Admin Login



Use-case Admin login:

Use case id: 1

Initiator: Admin

Goals: Admin login to administrator page

Main success scenario: Admin can login in the system without any problems and go to the dashboard.

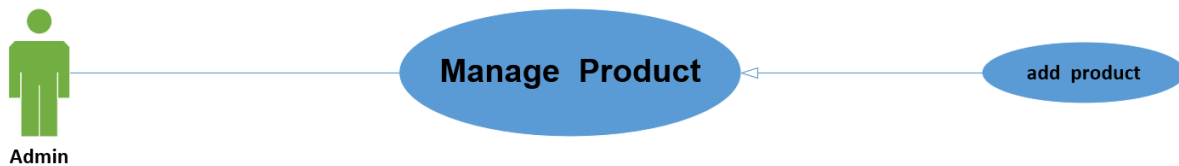
Unsuccessful scenario: The system can't identify admin and the admin can't login in the system

Precondition: Admin has already singed in database

Postcondition: admin login in the system successfully

2. Admin manage product

2.1 Add Product



Use-case Admin add Product:

Use case id: 2

Initiator: Admin

Goals: Admin add new product to the system

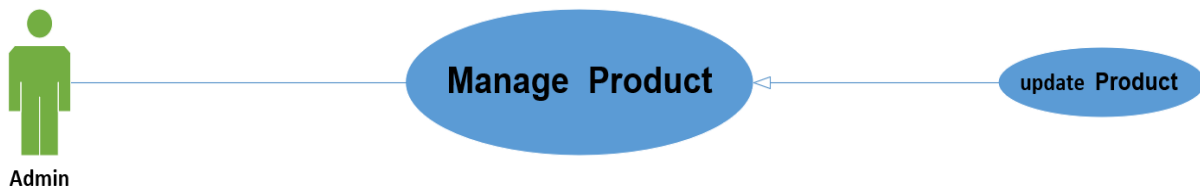
Main success scenario: Admin add new product in the system successfully

Unsuccessful scenario: The Admin has problem from the system and cannot add the product

Precondition: Admin add all properties of product

Postcondition: Admin add new product into the database and begin on the app

2.2 Update Product



Use-case Admin update Product:

Use case id: 3

Initiator: Admin

Goals: Admin update properties of product

Main success scenario: Admin can update the property or properties of any product successfully

Unsuccessful scenario: The Admin has problem from the system and cannot update any property of product

Precondition: Admin select the property or properties that want to edit it

Postcondition: Admin update property or properties of product

2.3 Delete product



Admin



Manage Product

delete product

Use-case Admin delete Product:

Use case id: 4

Initiator: Admin

Goals: Deleting product

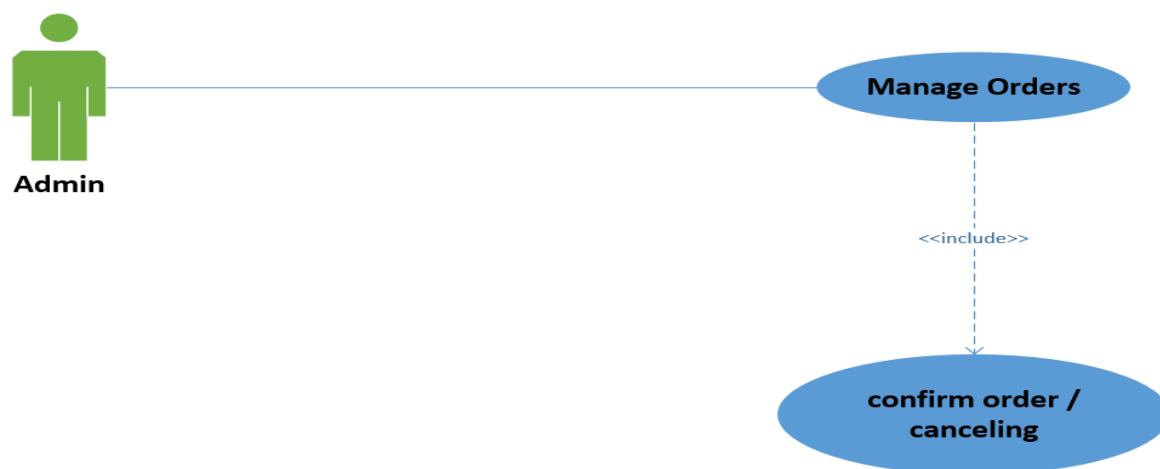
Main success scenario: Admin can delete any product successfully

Unsuccessful scenario: The Admin has problem from the system and can't delete the product

Precondition: Admin select the product that want to delete it

Postcondition: Admin delete the product from database

3. Admin manage orders



Use-case Admin Manage Orders:

Use case id: 5

Initiator: Admin

Goals: Admin confirm or cancel orders from users

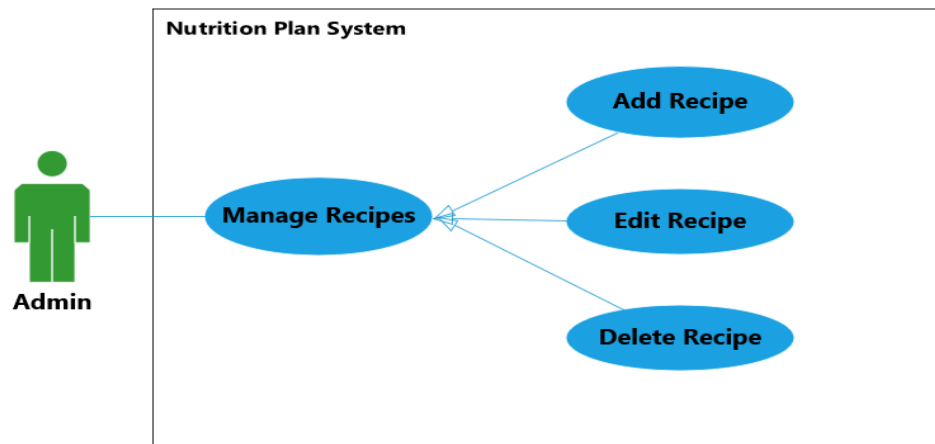
Main success scenario: Admin confirm or canceling order for users successfully

Unsuccessful scenario: Admin can't confirm order for user because the product not available, so he cancel the order

Precondition: Admin select the product that want to confirm or cancel it

Postcondition: Admin confirm or cancel the order for user

4. Admin manage recipe



Use-case Admin Manage recipe:

Use case id: 6

Initiator: Admin

Goals: Admin add, update, or delete recipe.

Main success scenario:

- 1-Admin can add ingredients and directions of recipes to some meals.
- 2-Admin can update the ingredients or directions recipe of the meals.
- 3-Admin can delete recipe of meals.

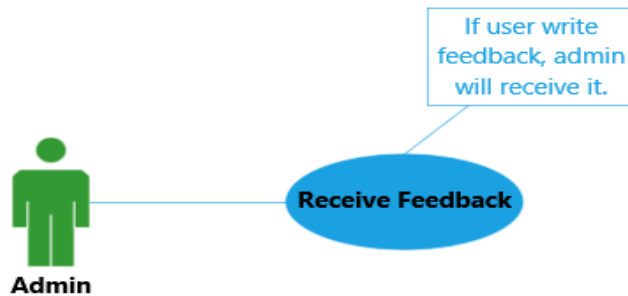
Unsuccessful scenario:

The Admin has problem from the system and can't add, update, or delete the recipe.

Precondition: Admin select the food that want to recipe it.

Postcondition: the system is ready to add, update, or delete the recipe of food.

5. Admin receive feedback



Use-case Admin Receive Feedback:

Use case id: 7

Initiator: Admin

Goals: Read feedback from users

Main success scenario:

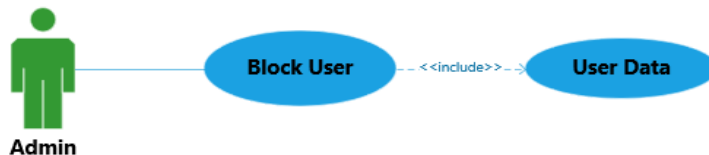
- 1-Open own page
- 2-Click on feedback link
- 3-Choose one of received feedback
- 4-Mark it as read

Unsuccessful scenario: If system don't send to admin the user feedback, admin can't to read that.

Precondition: Admin has already singed in database of the system and is logged successfully

Postcondition: The feedback is read by admin and deleted from new feedback list.

6. Admin delete user



Use-case Admin delete User:

Use case id: 8

Initiator: Admin

Goals: Delete user that breaks the system standards like write feedback that contain illegal words or cancelling order after its reach.

Main success scenario:

Open own page and select the user that do something wrong with system like fraud or buy something and don't receive it

Unsuccessful scenario: admin don't deduct the unwanted user and can't block him

Precondition: Admin is Logged successfully

Postcondition: The user is added to blacklist table in the database.

7. Admin generate report



Use-case Admin Generate Report:

Use case id: 9

Initiator: Admin

Goals: Generating report about sales in a month and about how many products in stock.

Main success scenario:

1-Open own page

2-In dashboard, show details about how many products in stock and not in stock, how many orders have been confirmed, average rate of the application and percentage of users who achieved their goals while using the application or didn't.

3-Click on see all in analytics section.

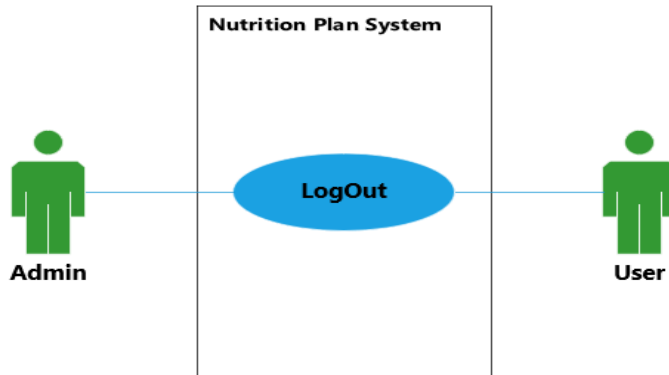
4-Show the visualization data about sales in a month and the percentage of quantity for every product in stock.

Unsuccessful scenario: If there is not enough order data, the report won't be accurate.

Precondition: Admin is Logged successfully

Postcondition: Admin knows how many users have brought a product in a month.

8. Log Out



Use-case log out:

Use case id: 10

Initiators: Admin, User

Goals: Logging out the system.

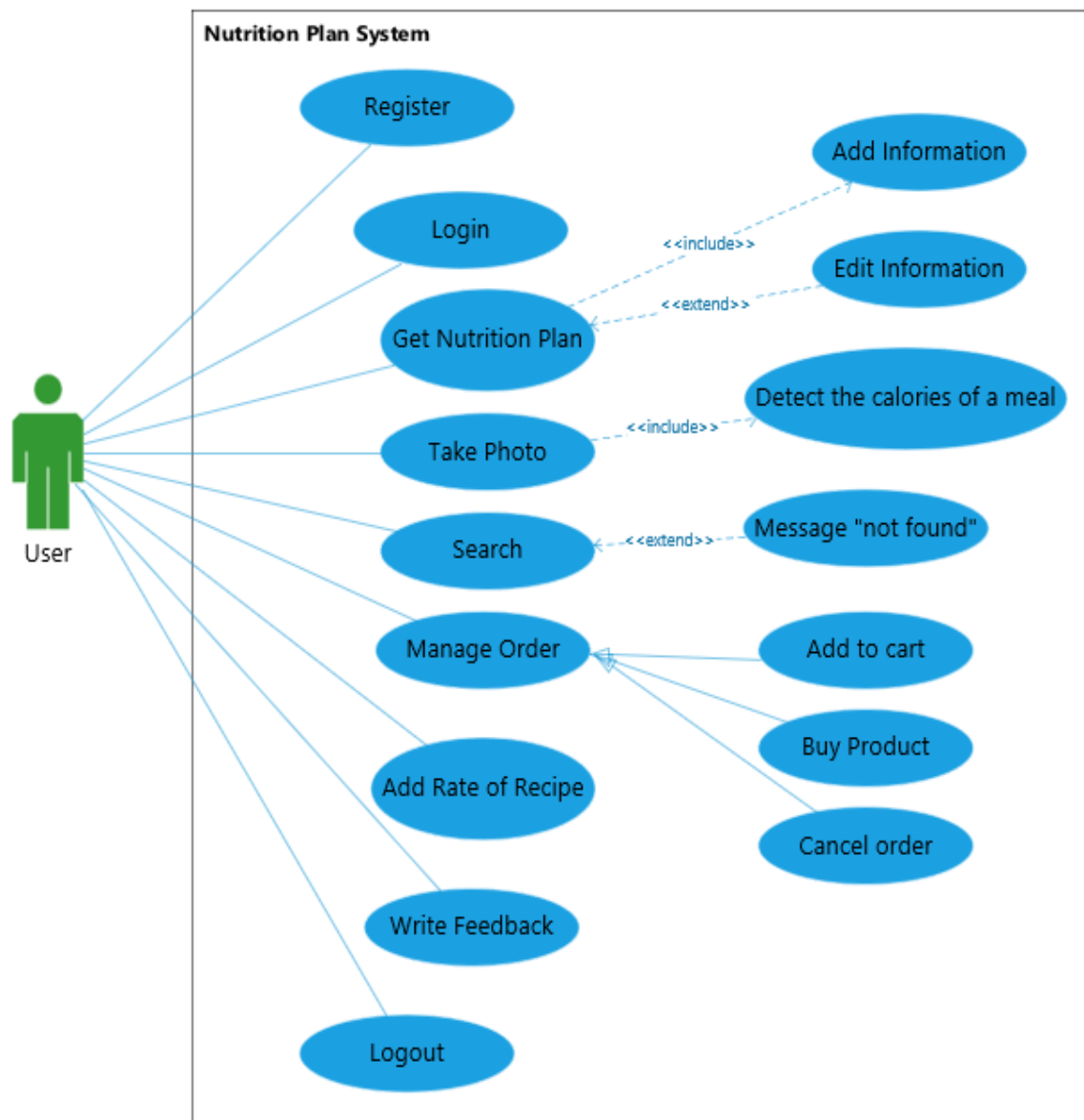
Main success scenario: -Click on log out button

Unsuccessful scenario: N/A.

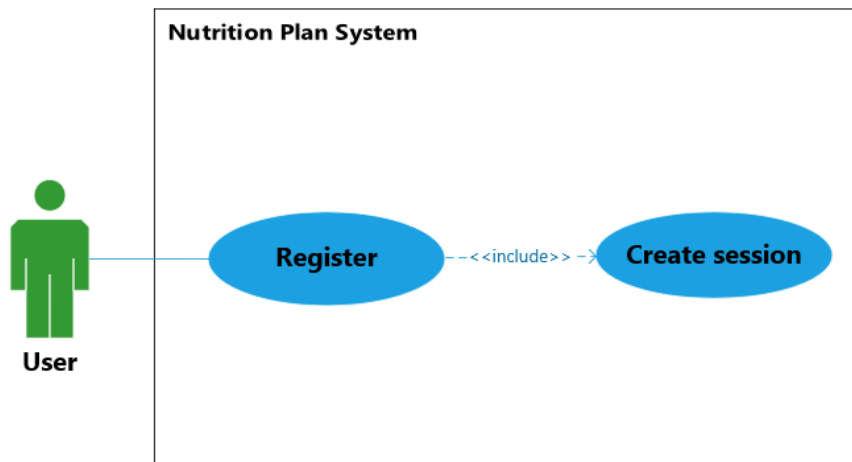
Precondition: Admin or User has accessed some processes

Postcondition: Admin or User is logged out.

- User



1. User registration



Use-case user registration:

Use case id: 11

Initiator: User

Goals: user register in application.

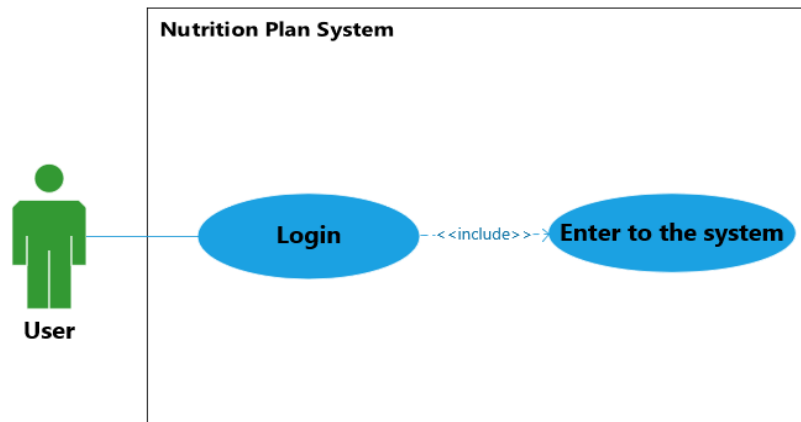
Main success scenario: user can register and can do anything in the app

Unsuccessful scenario: user can't register in app because he is blocked from the admin, another user uses the same name of registration, or the system has problem.

Precondition: the system is ready in registration screen.

Postcondition: the system create session to user in database and open the app to the user.

2. User login



Use-case user registration:

Use case id: 12

Initiator: User

Goals: user login in application.

Main success scenario: user login in the app

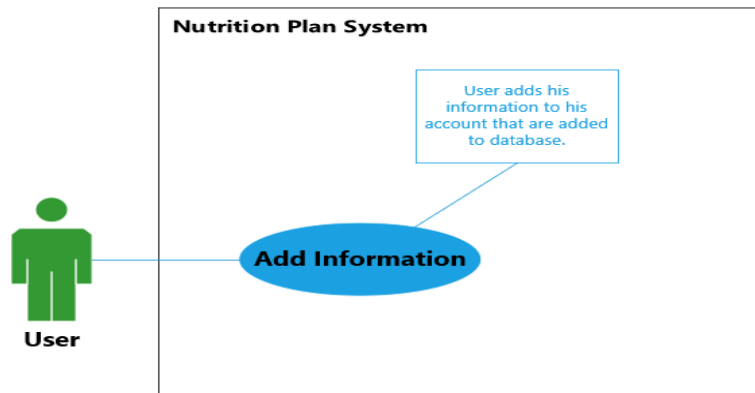
Unsuccessful scenario: user can't login in app because he is blocked from the admin, he writes e-mail or password wrong, or the system has problem

Precondition: the user register in application and have session in database.

Postcondition: the system is ready to use

3. User Get Nutrition Plan

3.1 User Add Information



Use-case User add information:

Use case id: 13

Initiator: User

Goals: User add information to his account.

Main success scenario:

1-open home page.

2-click on add Information link.

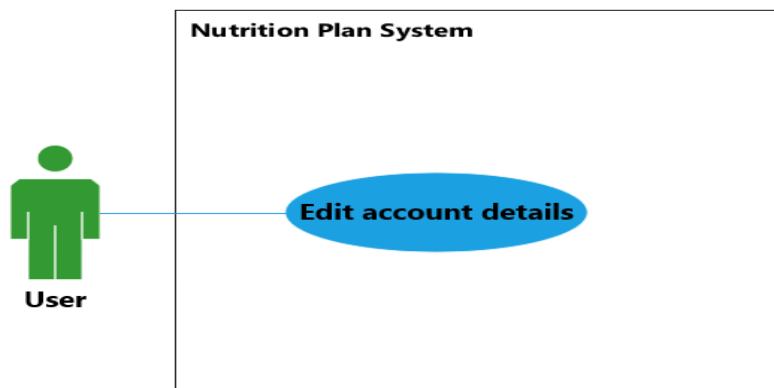
3-user enter add his height, his weight, his goal and his medical history and click on submit.

Unsuccessful scenario: user can't login in app because he is blocked from the admin, he writes e-mail or password wrong, or the system has problem

Precondition: User must login to system

Postcondition: User add his information in the system successfully

3.2 User Edit information



Use-case User edit information:

Use case id: 14

Initiator: User

Goals: User update his information.

Main success scenario:

1-open home page.

2-click on update Information link.

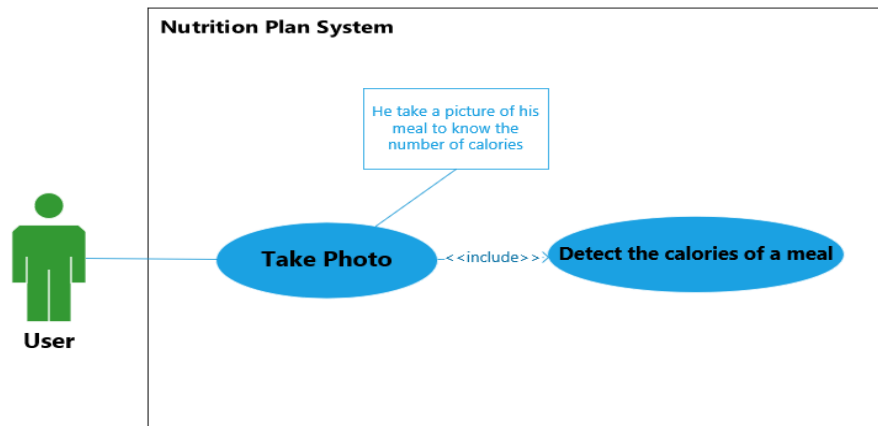
3-user update his data and click on submit.

Unsuccessful scenario: The User cannot update information to your account after his login

Precondition: User must login to the system

Postcondition: User update his information in the system successfully

4. User Take Photo



Use-case User take photo:

Use case id: 15

Initiator: User

Goals: User takes a picture of his meal to know the number of calories.

Main success scenario:

1-open home page

2-click on camera

3-take meal photo

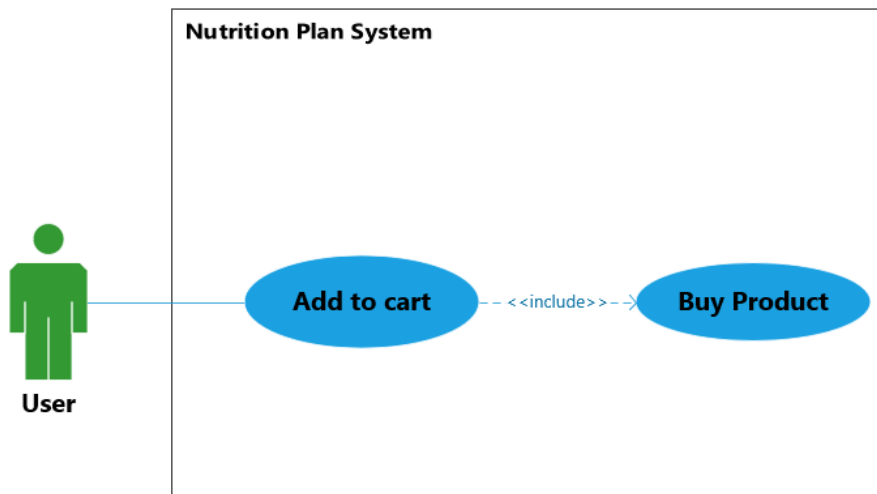
Unsuccessful scenario: The User has problem from the system and cannot takes a picture of his meal to know the number of calories

Precondition: User takes a picture of his meal and the system detect the calories in a meal

Postcondition: The user knows the number of calories

5. User Manage Order

5.1 User Add to Cart



Use-case add to cart:

Use case id: 16

Initiator: User

Goals: User can add a product he bought in a shopping list.

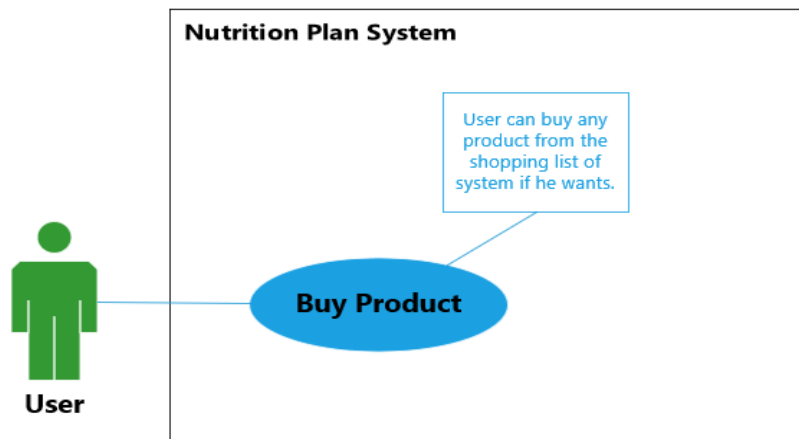
Main success scenario: User enter the app and buy a product or more and add them in shopping list.

Unsuccessful scenario: User can not add the product in shopping list because it something wrong with the payment method or the system has a problem or product not available.

Precondition: User login to app with his e-mail and buy a product.

Postcondition: The system reserved his products and send to hi confirmation message.

5.2 User Buy Product



Use-case buy product :

Use case id: 17

Initiator: User

Goals: User can buy any product.

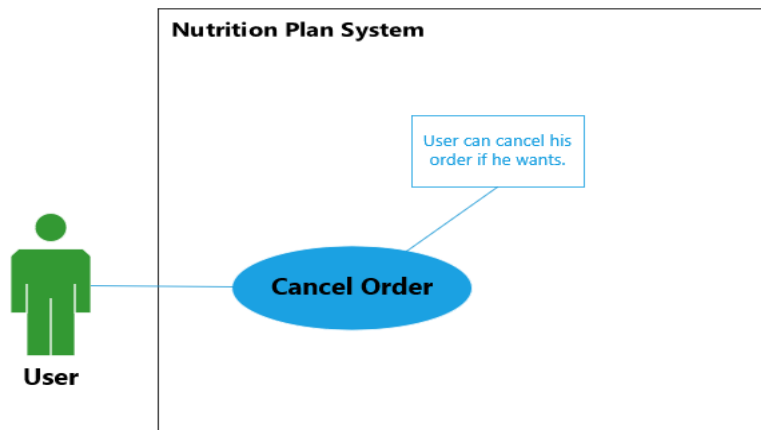
Main success scenario: User enter the app and buy a product.

Unsuccessful scenario: User cannot buy a product because no item of this product available or the system has a problem.

Precondition: User login to app with his e-mail and select the product he needs.

Postcondition: The system reserves the product and send to user confirmation message.

5.3 User Cancel Order



Use-case cancel order:

Use case id: 18

Initiator: User

Goals: User can cancel his order.

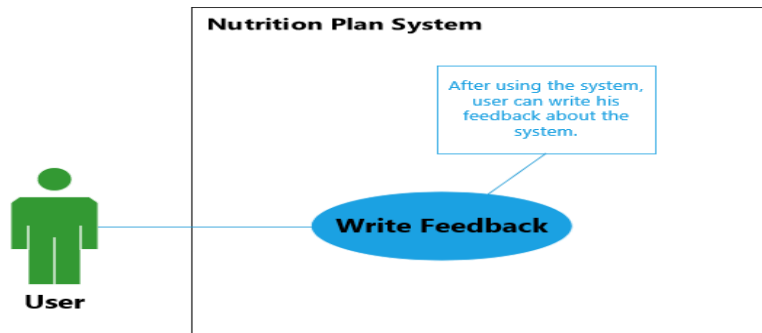
Main success scenario: User enter the app and cancel his order for buying a product.

Unsuccessful scenario: User can not cancel his order because it is 2days left to his order or the system has a problem.

Precondition: User login to app with his e-mail

Postcondition: The system canceled his order and get this product back and increase amount by 1.

6. User Write Feedback:



Use-case write feedback

Use case id: 19

Initiator: User

Goals: User can write his feedback about system accuracy and the quality of product.

Main success scenario:

1-User enter the app and write his feedback.

2-Enter rate of application.

3-Enter if he achieved his goal while using app or not.

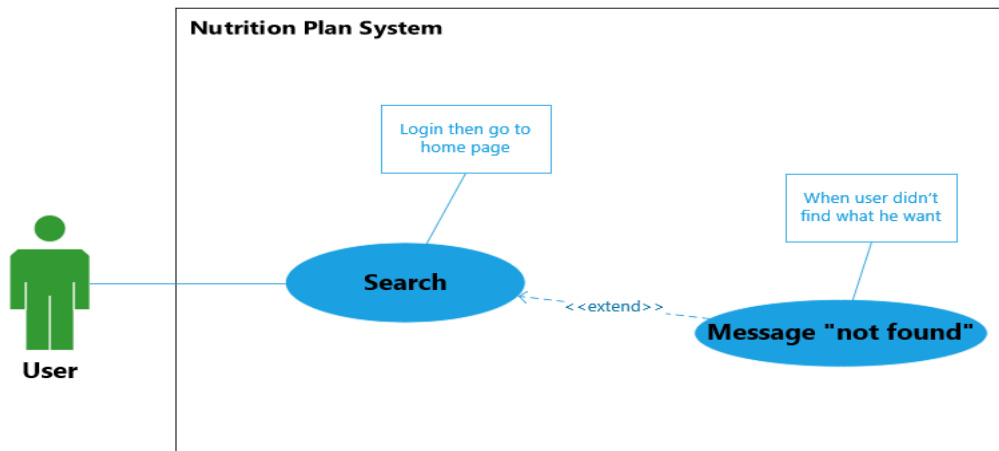
Unsuccessful scenario: User cannot write his feedback because his feedback has anon forbidden words or the system has a problem.

Precondition: User login to app with his e-mail, and use the app.

Postcondition: The system sends his feedback to the admin.

Unsuccessful scenario: he didn't write feedback before.

7. User Search



Use-case search:

Use case id: 20

Initiator: User

Goals: User can search about anything in system.

Precondition: User must login to system

Postcondition: User find the section or thing he want

Main success scenario:

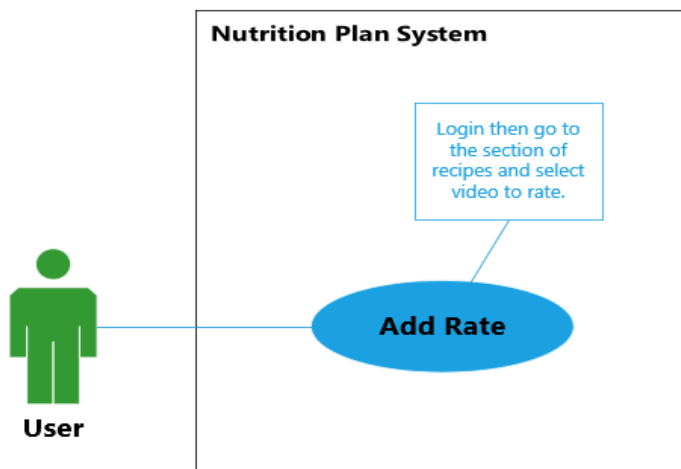
- 1- user login and into to the app
- 2- go to home
- 3-select icon of search
- 4-write what he wants on box
- 5-appear what he wants

Unsuccessful scenario: he wrote the word wrong, or the system don't found what the user search

Extension:

- 5.a he didn't find what he wants
- 5.a.1- send msg for user "this is not found"

8. User Add rate to Recipe



Use-case add rate of recipe:

Use case id: 21

Initiator: User

Goals: User can rate the video of recipe.

Precondition: User must login to system and the recipe that is want to rate it

Postcondition: User put his rate on a recipe

Main success scenario:

1- user login and into to the app

2- go to home

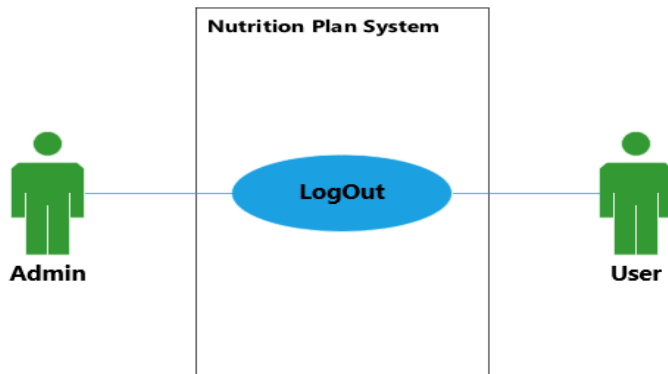
3-search on recipe section

4-watch the recipe from this section before rate

4-rate it

Unsuccessful scenario: The rate doesn't record in system

9. Log Out



Use-case log out:

Use case id: 22

Initiators: Admin, User

Goals: Logging out the system.

Main success scenario: -Click on log out button

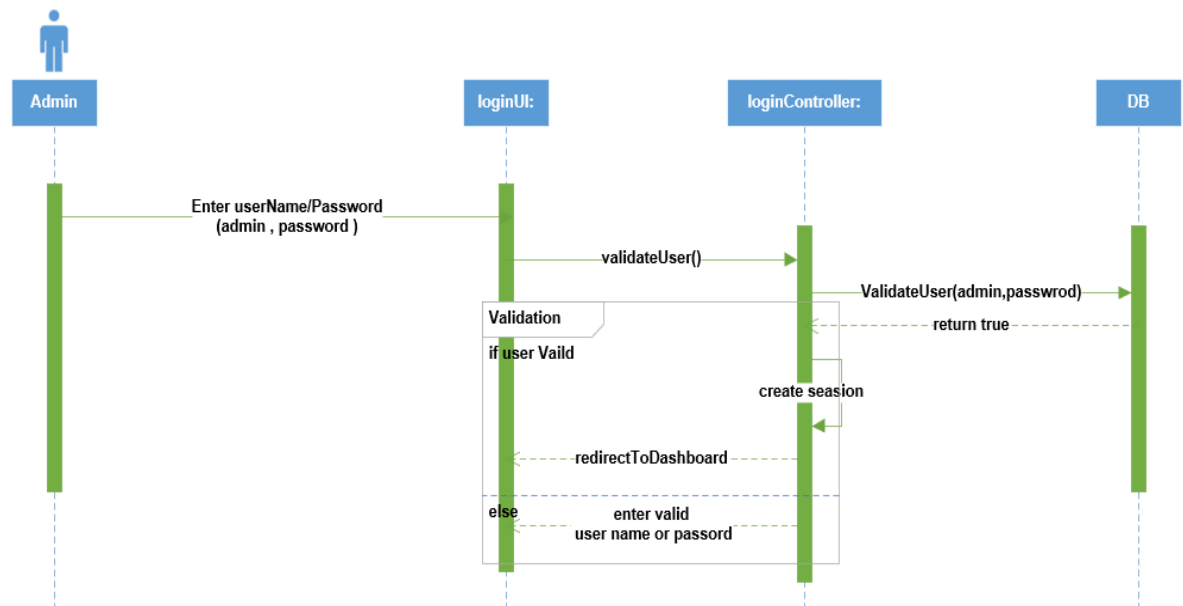
Unsuccessful scenario: N/A.

Precondition: Admin or User has accessed some processes

Postcondition: Admin or User is logged out.

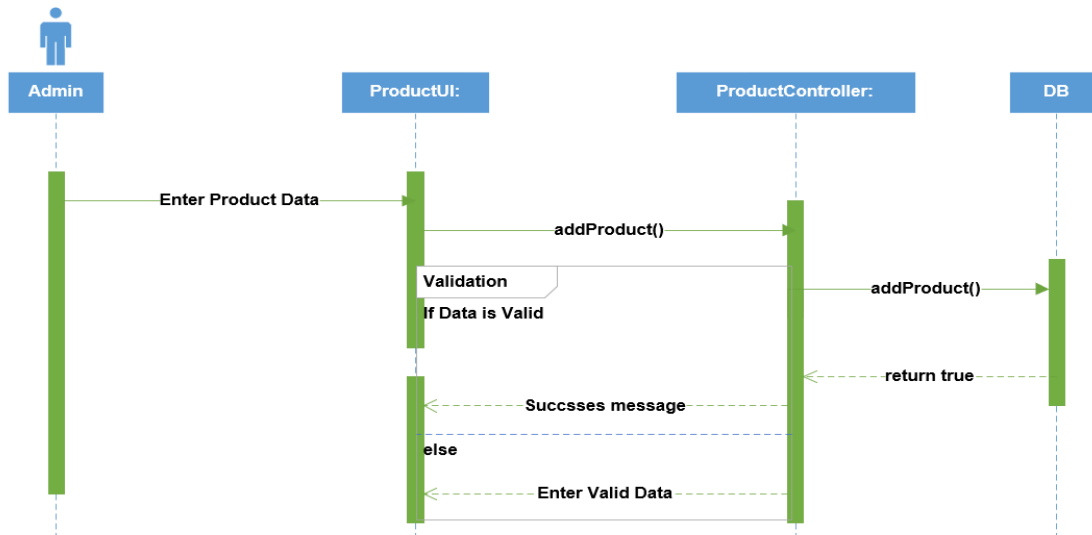
3.4 Sequence diagram

- Admin
 - Login

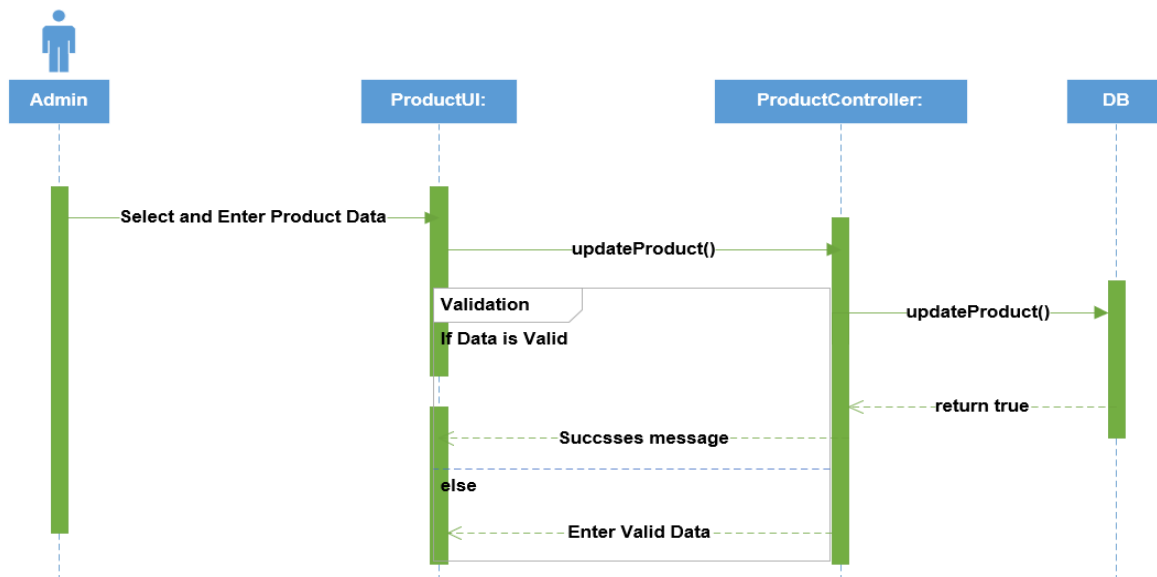


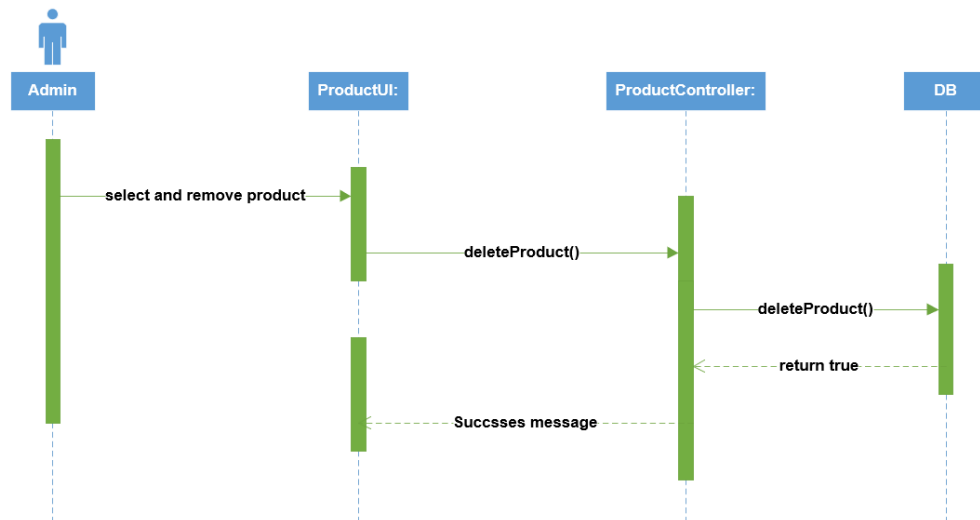
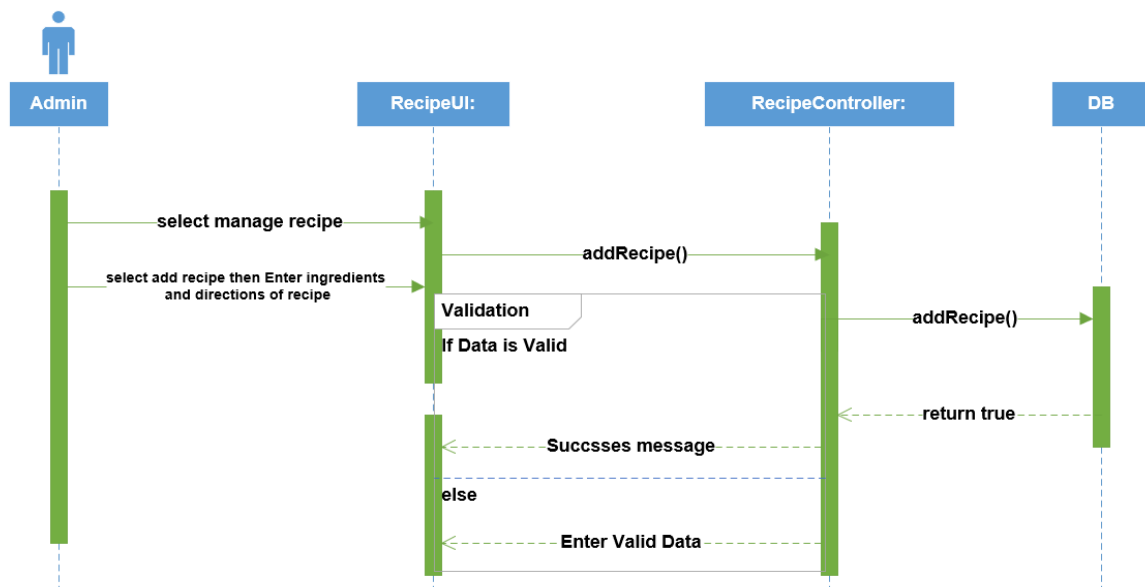
○ Manage Product

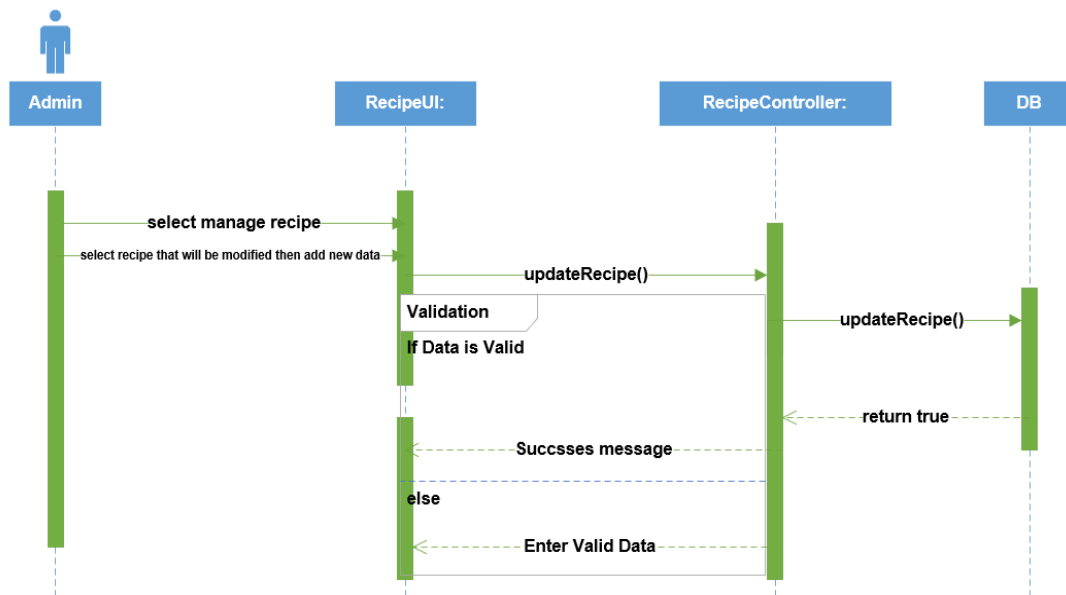
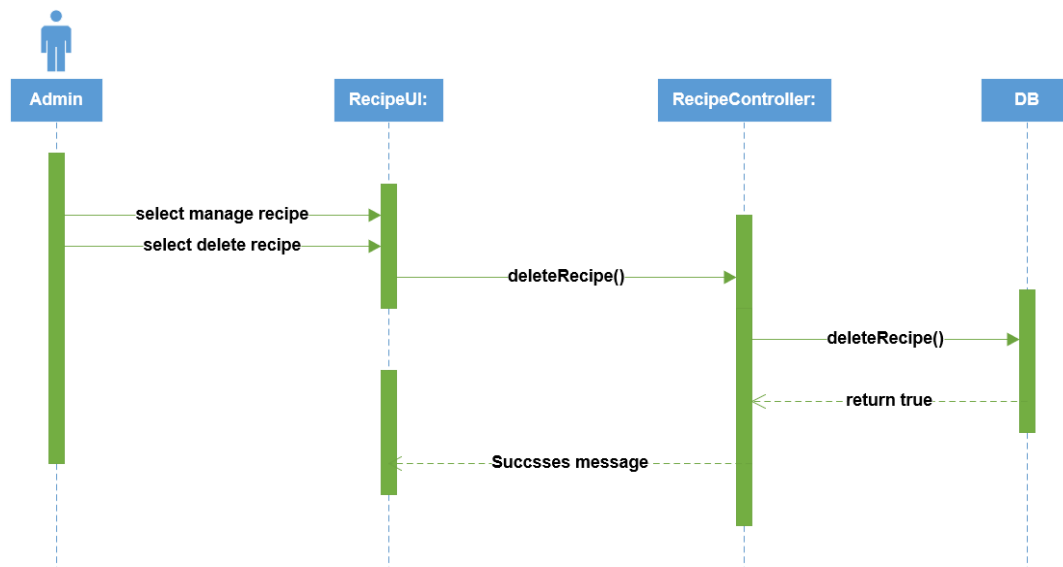
a. Add Product



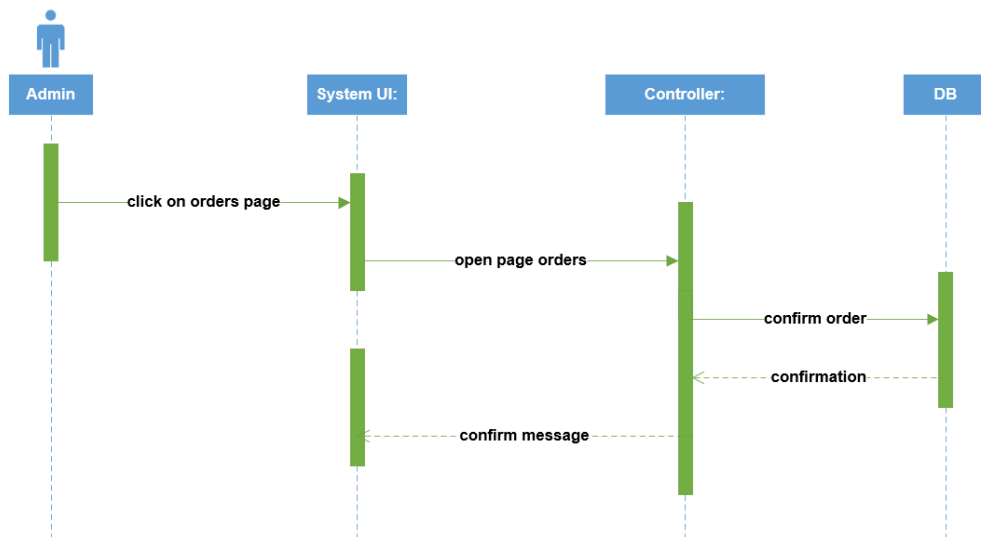
b. Update Product



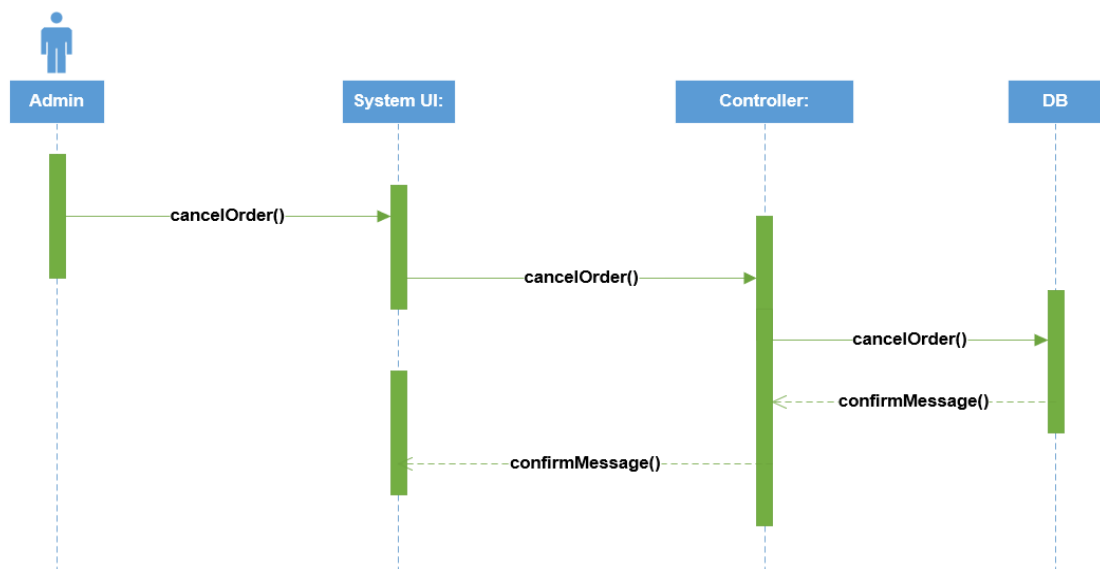
c. Delete Producto **Manage Recipe**a. Add Recipe

b. Update Recipec. Delete Recipe

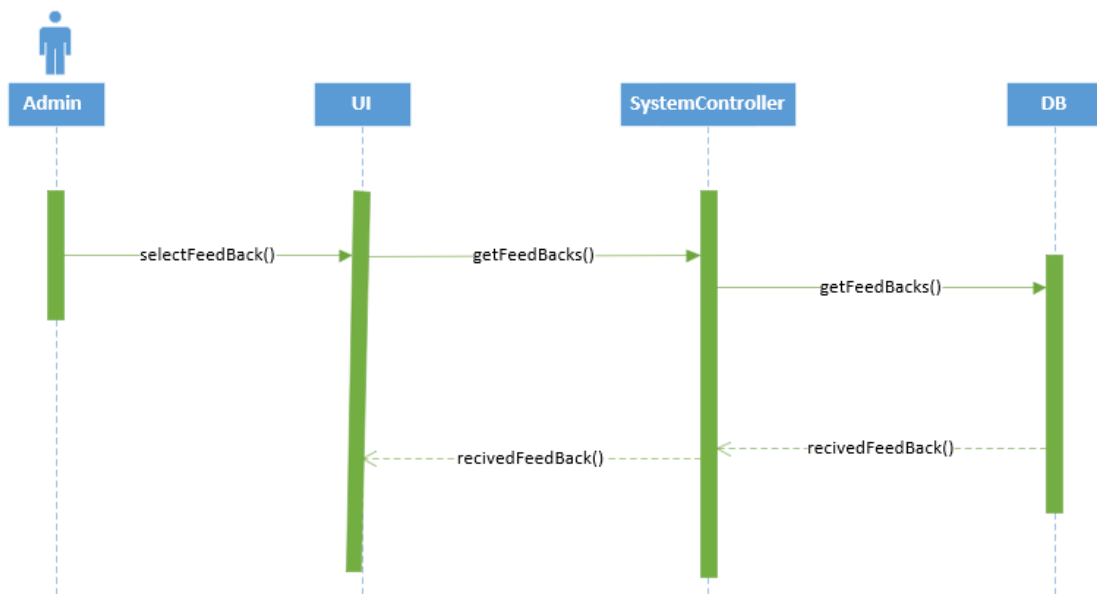
- **Manage Orders**
 - a. Confirm Order



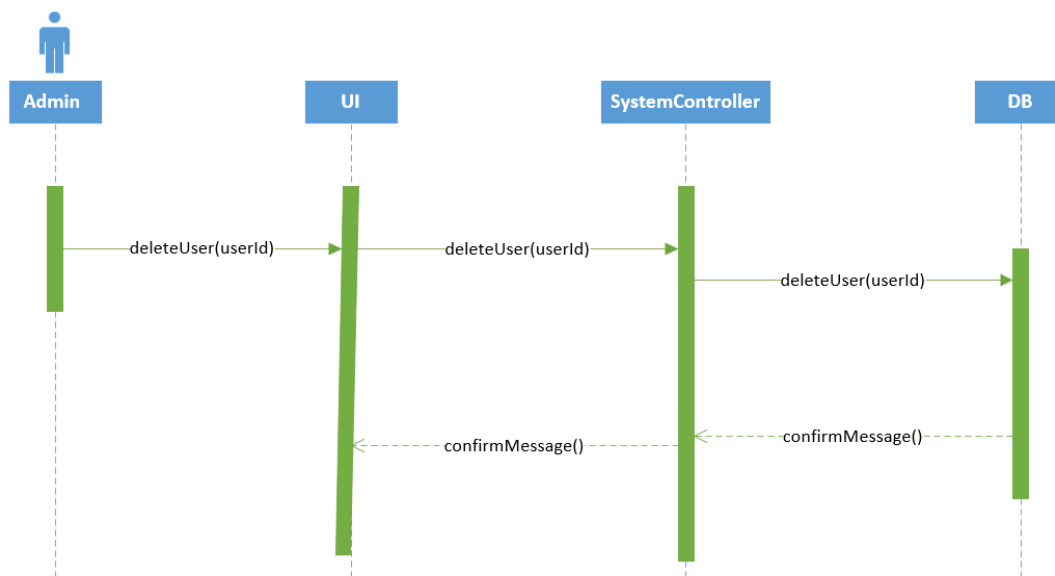
- b. Cancel Order



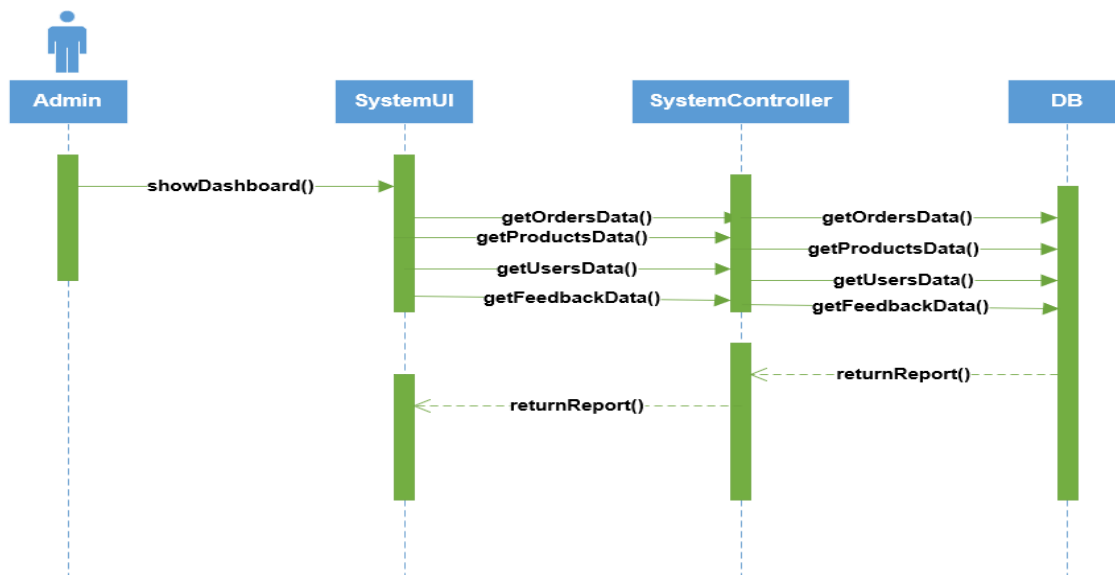
○ **Receive Feedback**



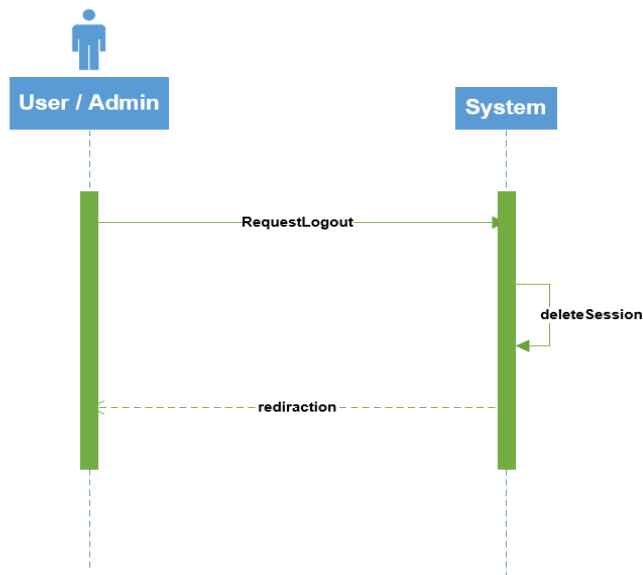
○ Delete User



○ Generate Report

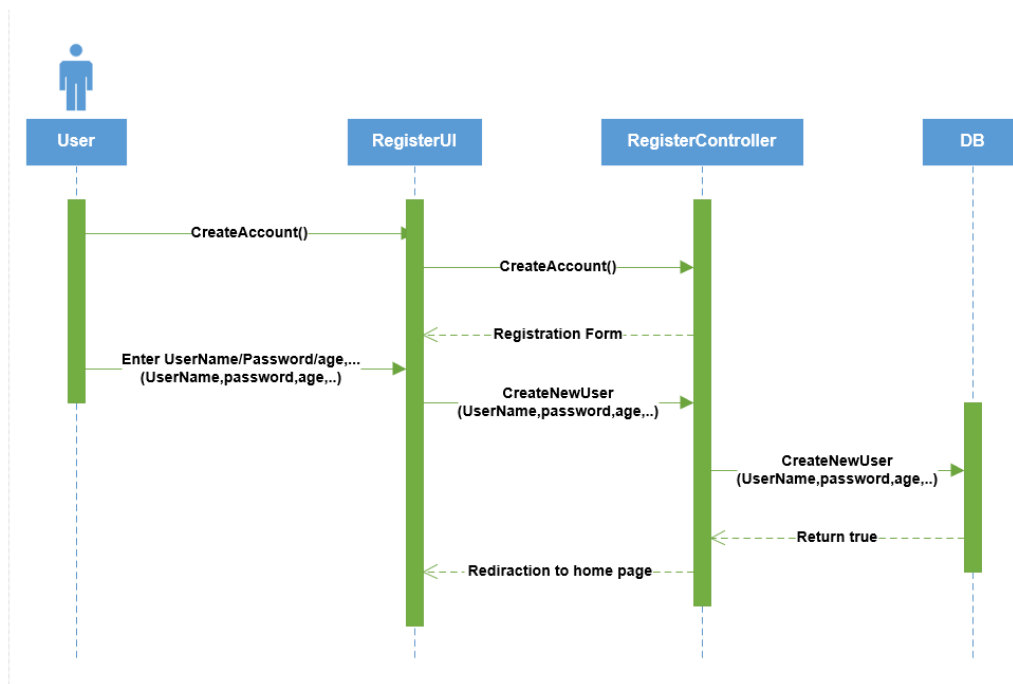


○ Log Out

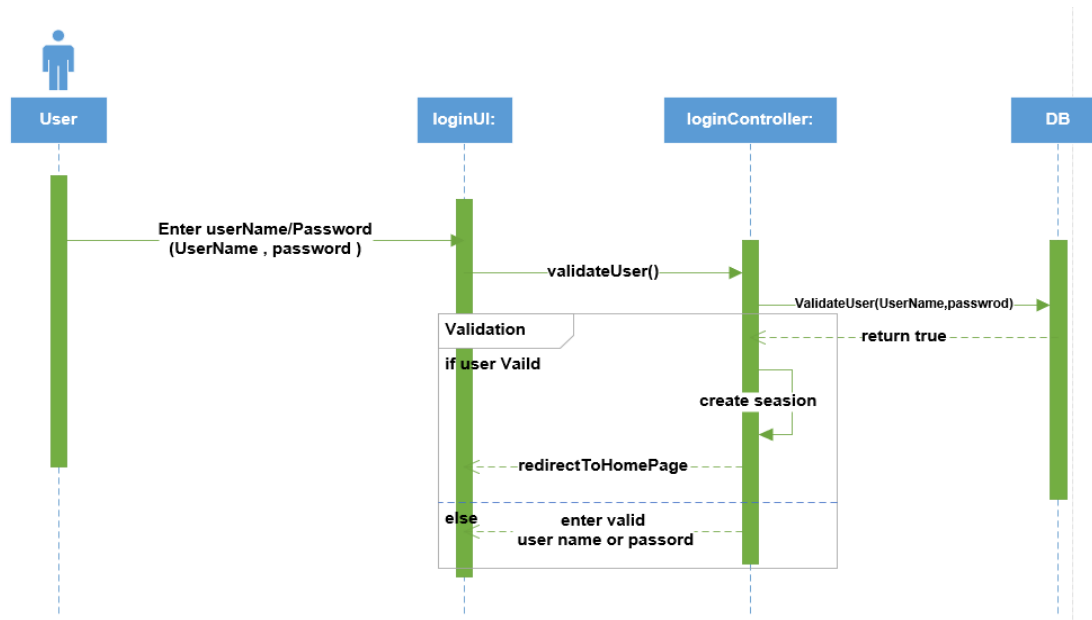


• User

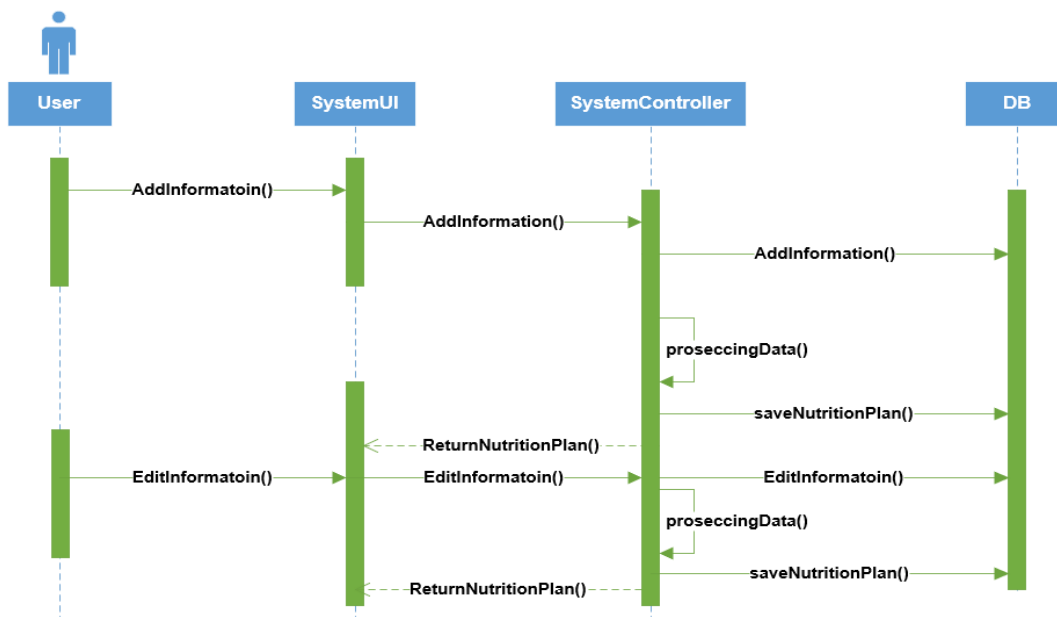
○ Register



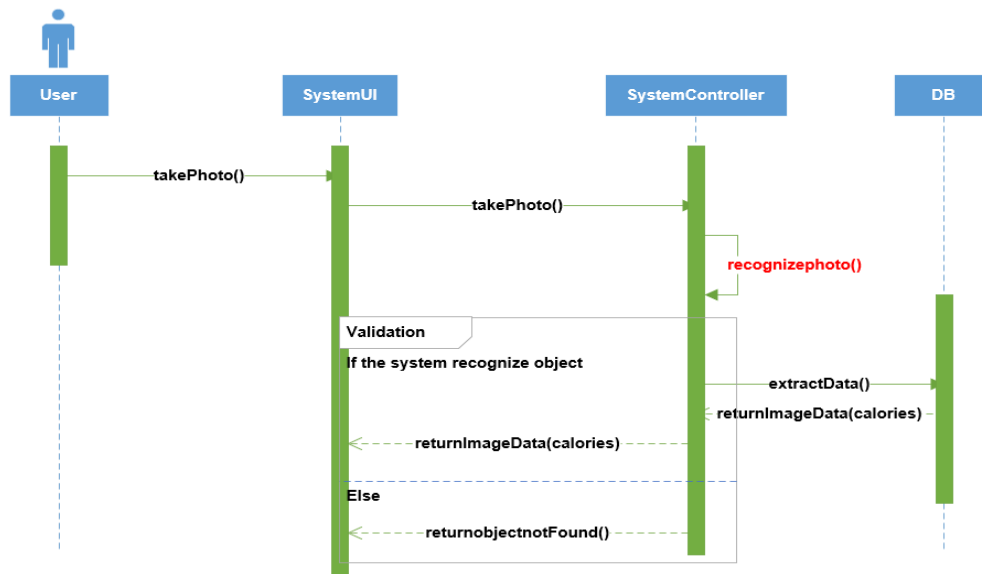
○ Login



○ Get Nutrition Plan



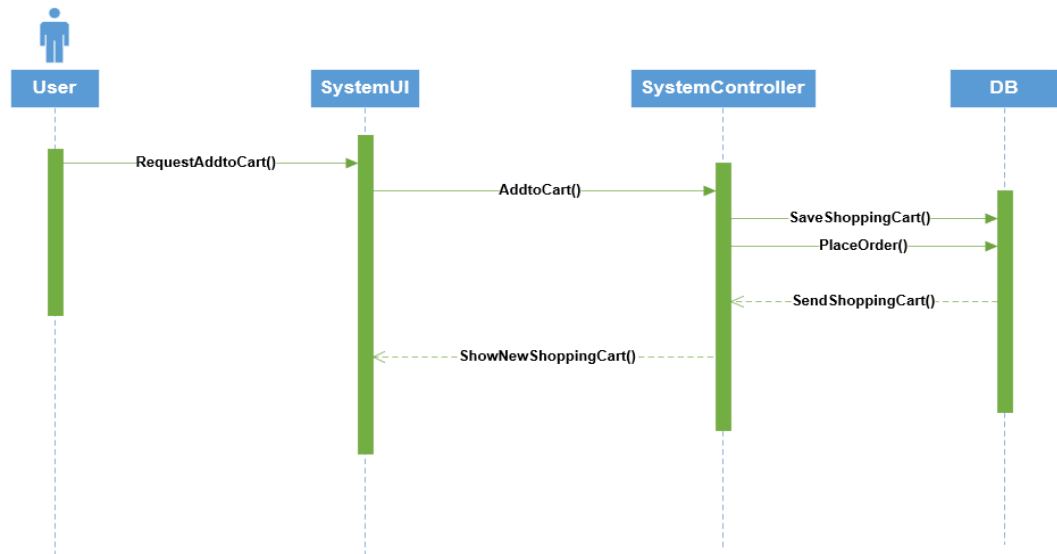
○ Take Photo



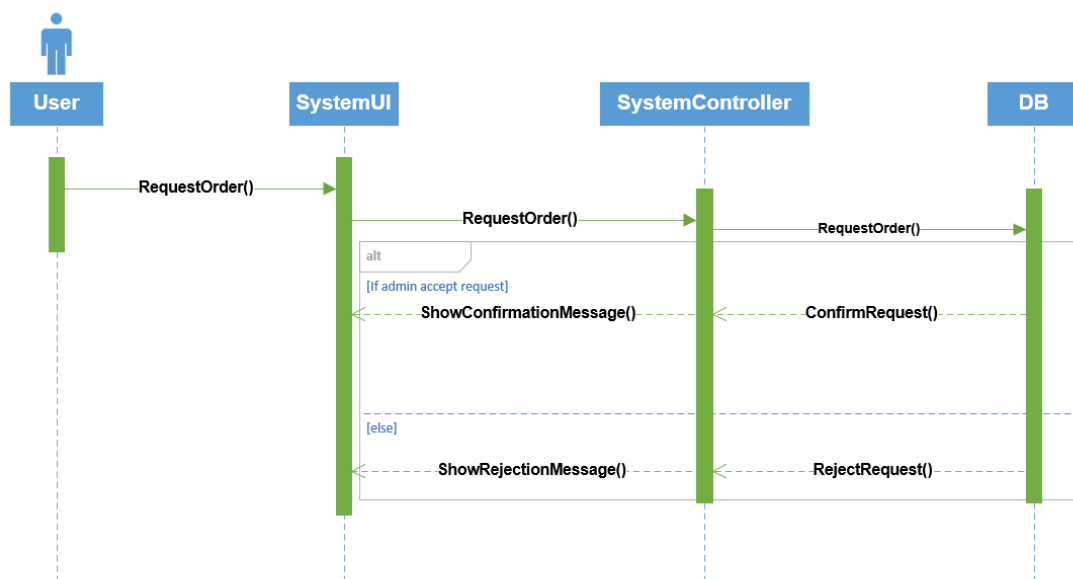
○ Search



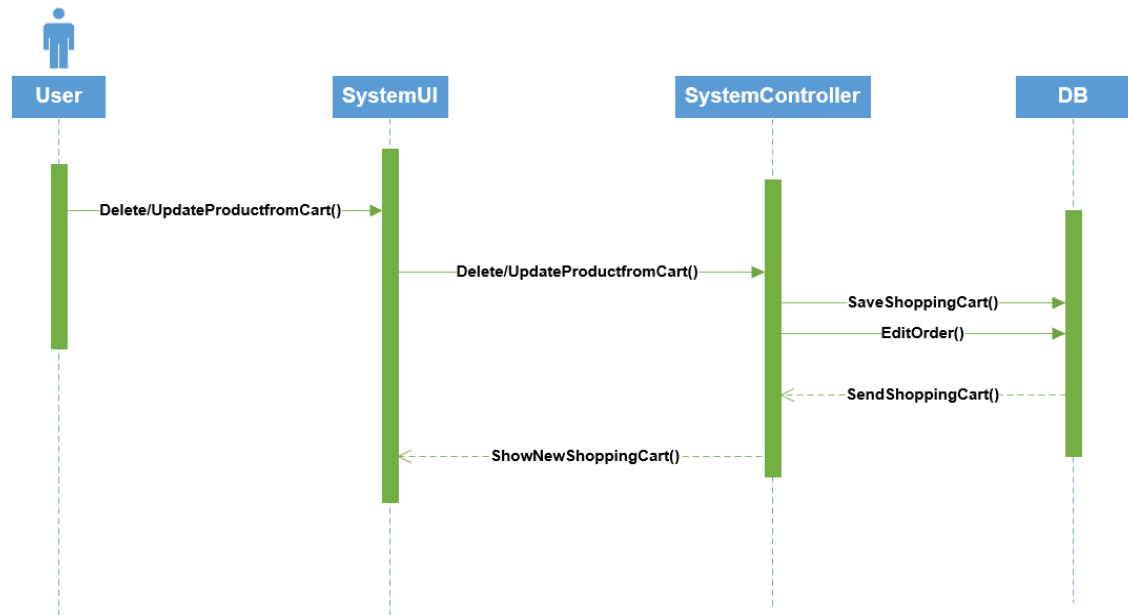
○ **Manage Order**
a. Add to car



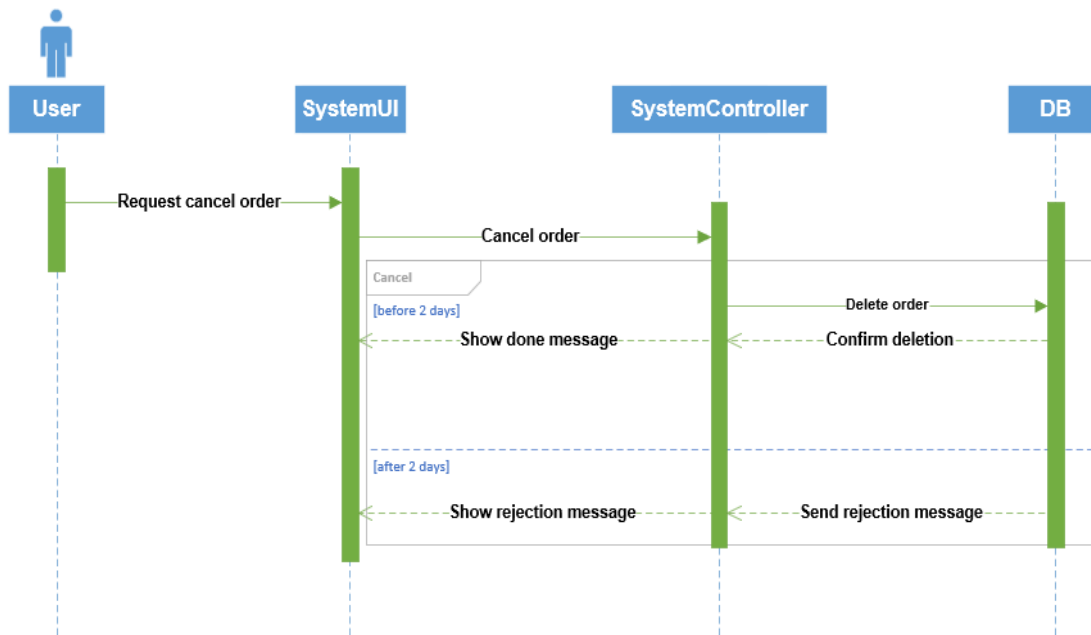
b. Buy product



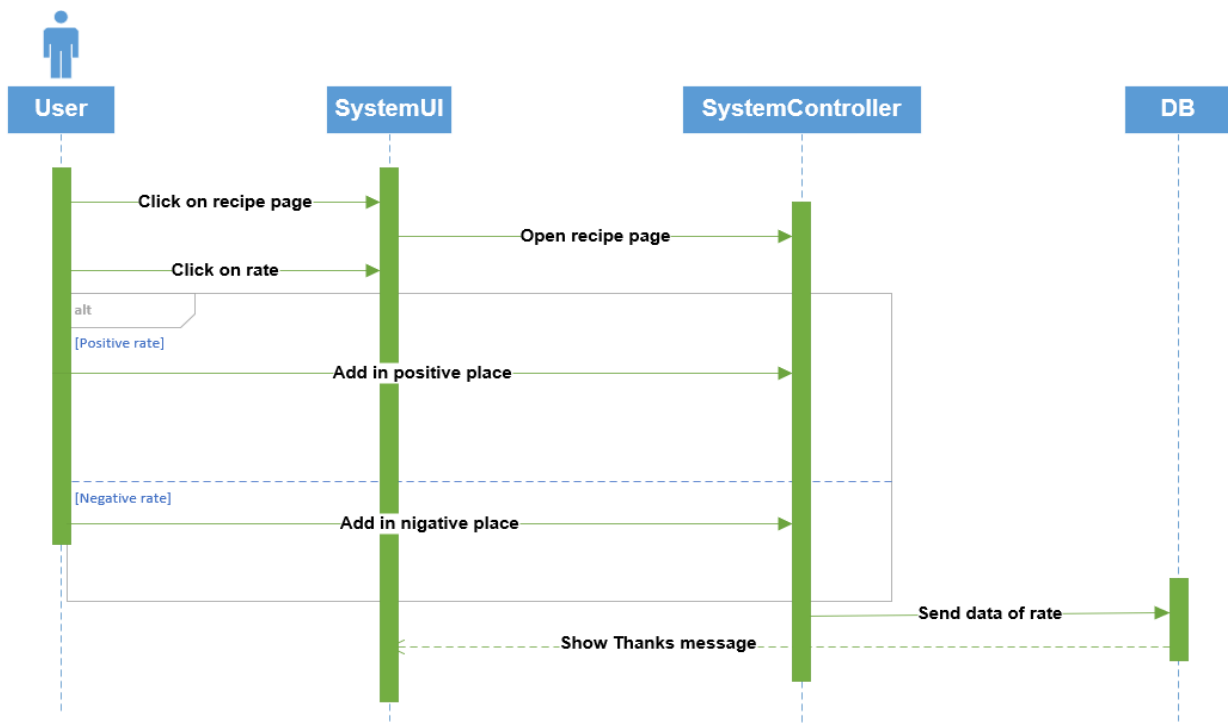
c. Delete/Update Product from cart



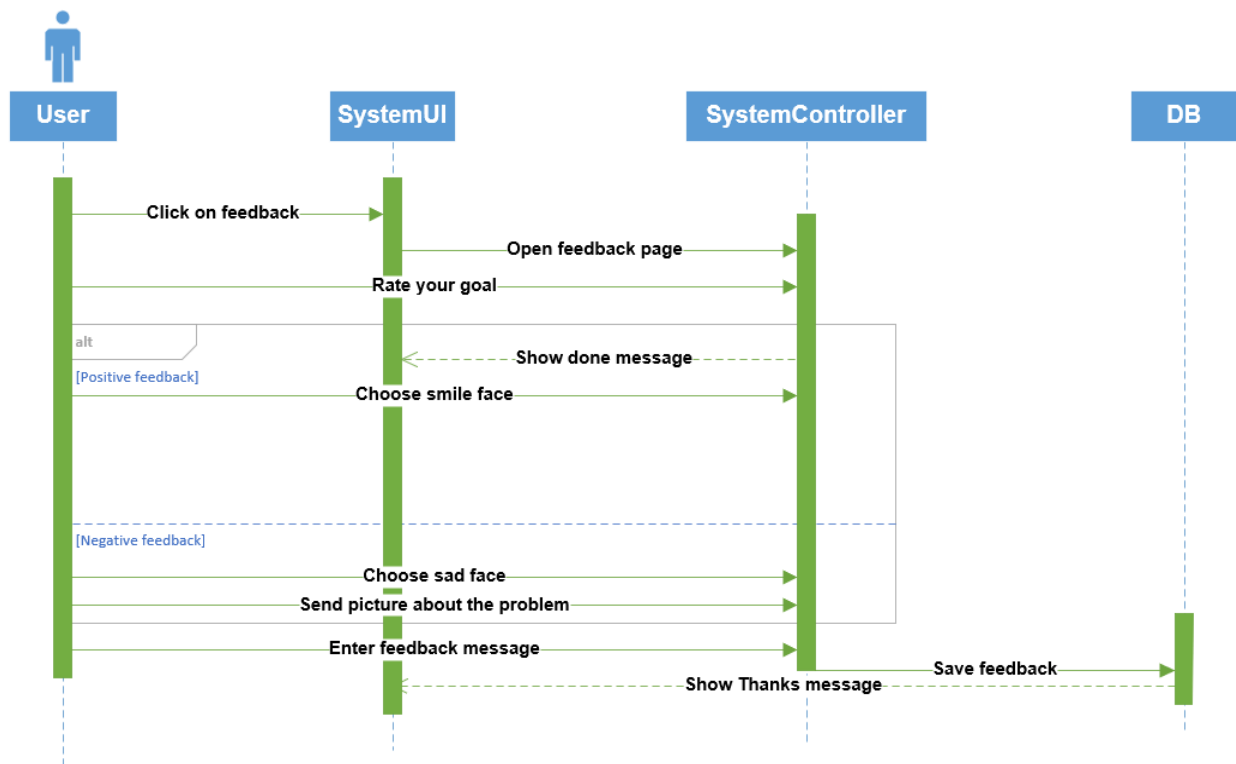
d. Cancel order



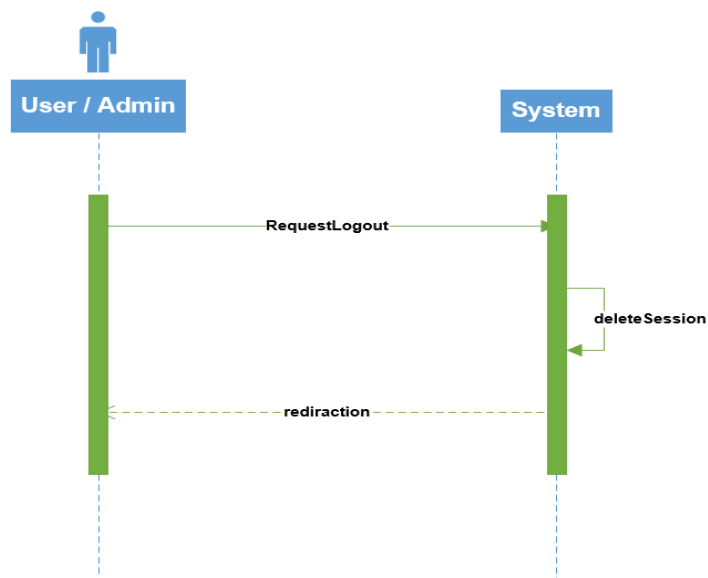
○ Add Rate of Recipe



○ Write Feedback

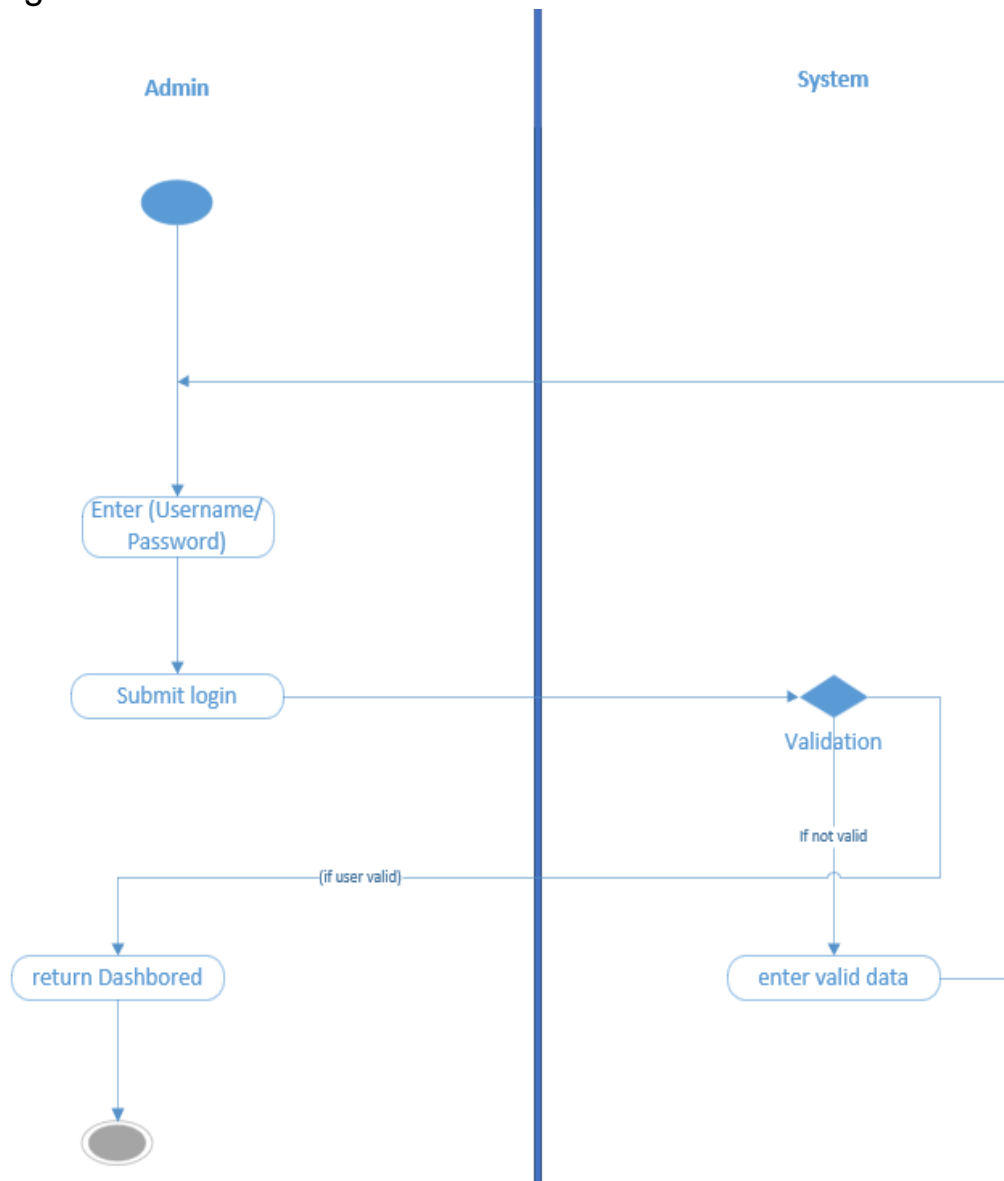


○ Logout

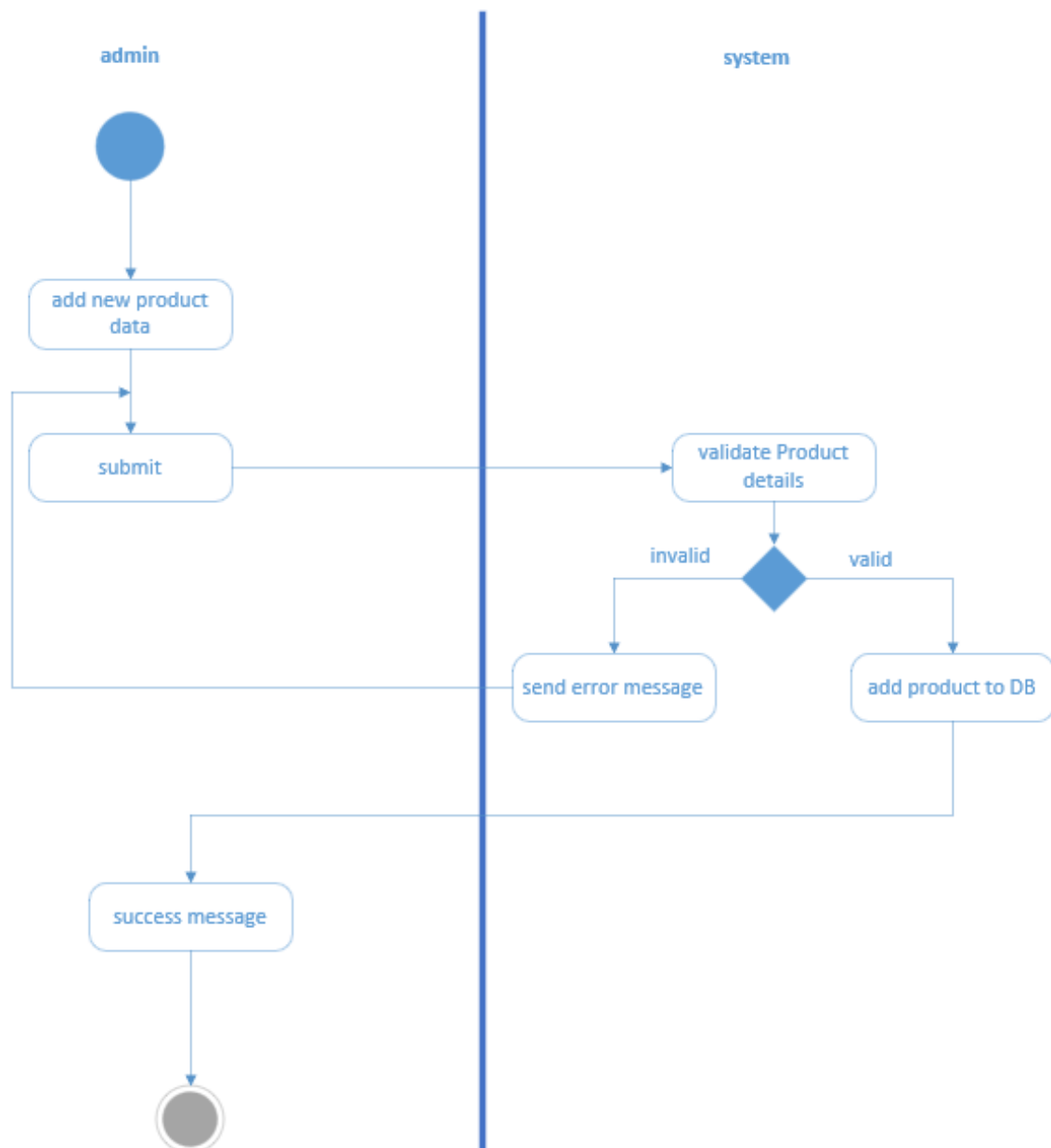


3.5 Activity Diagram

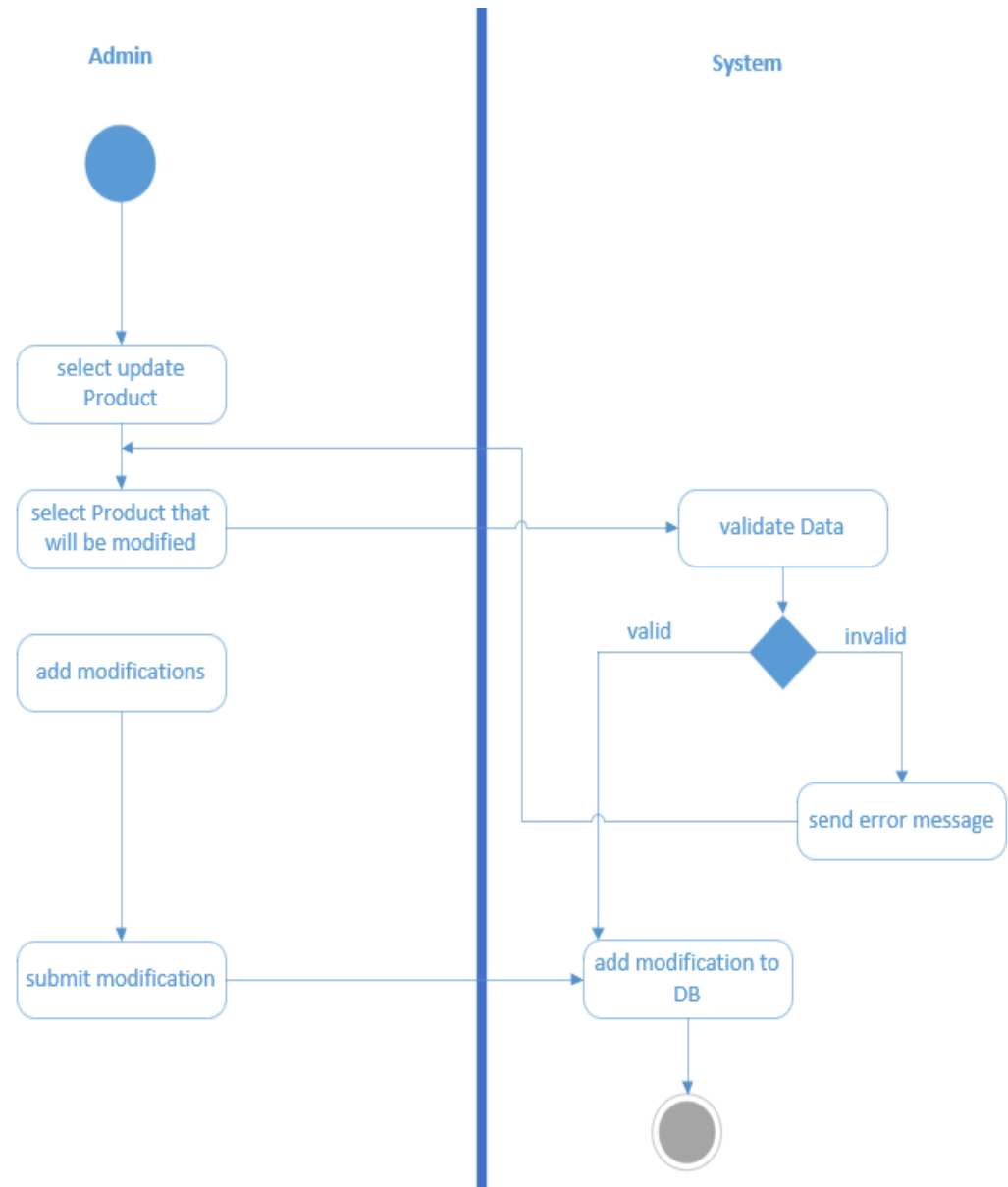
- Admin
 - Login



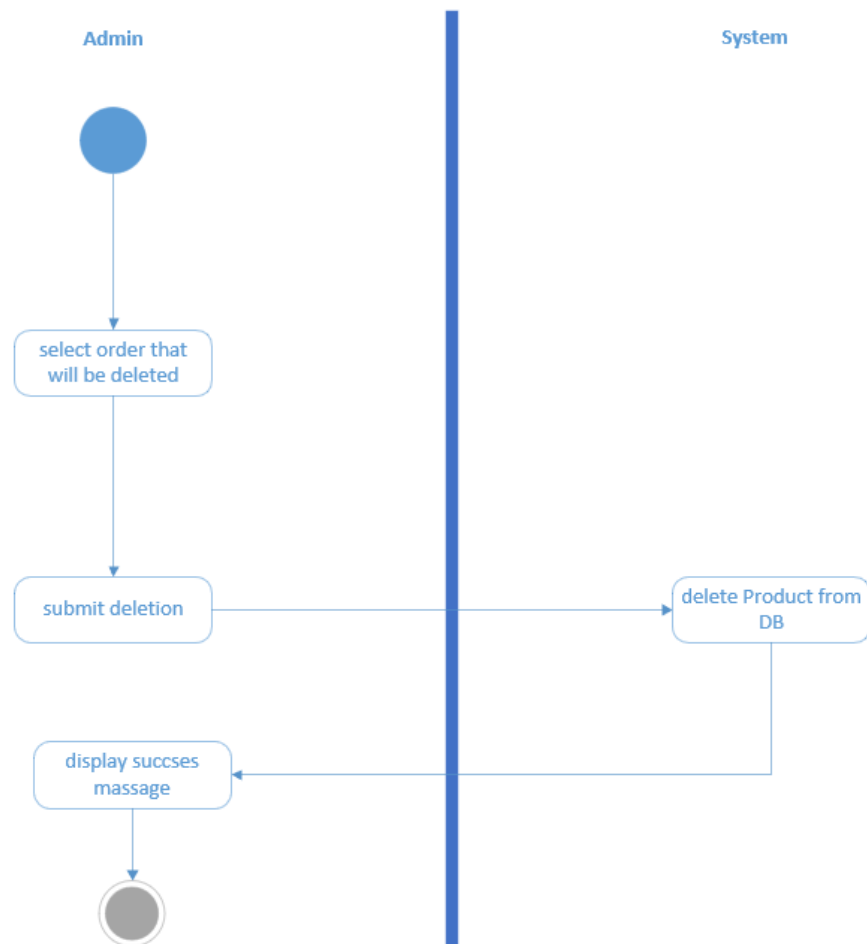
- **Manage Product**
 - a. Add Product



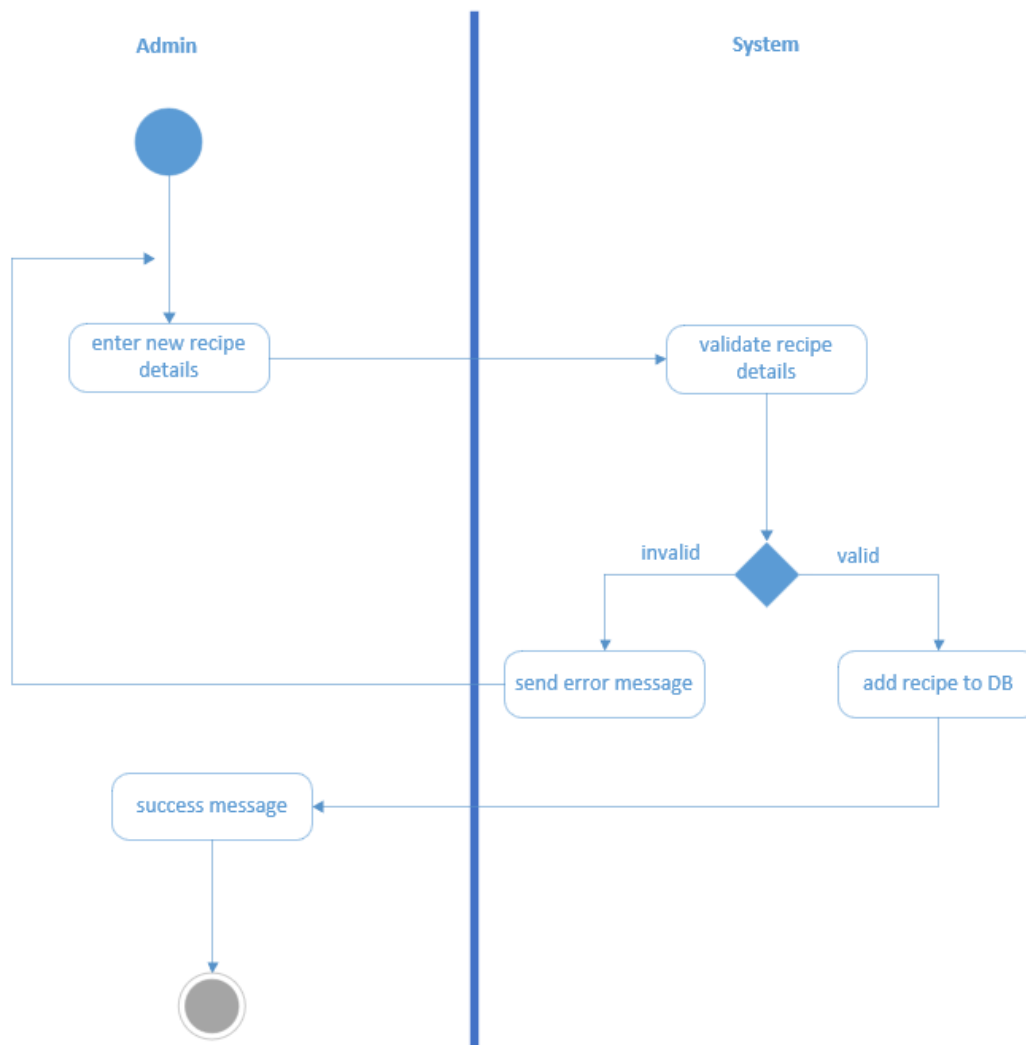
b. Update Product



c. Delete Product



- **Manage Recipe**
 - a. Add Recipe

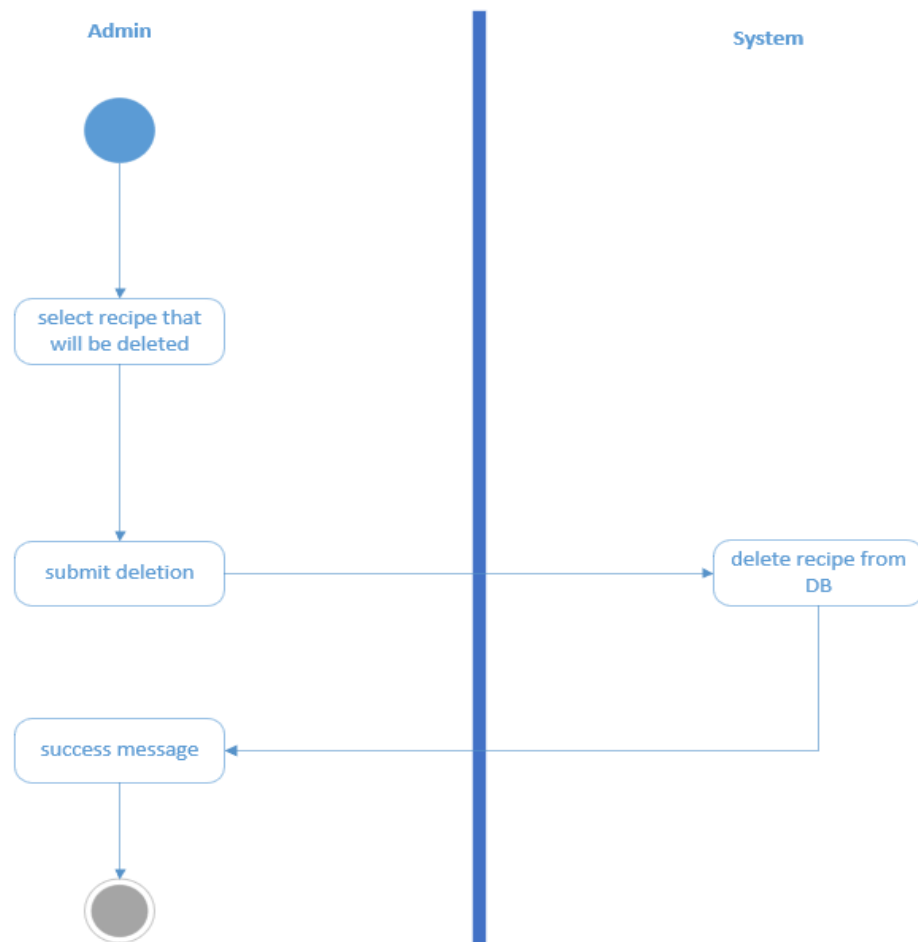


```

sequenceDiagram
    participant Admin
    participant System
    Admin->>System: select recipe that will be modified
    System->>System: add modifications
    System->>System: submit modification
    System->>System: send error message
    System->>System: add modification to DB
    System->>Admin: success message
    
```

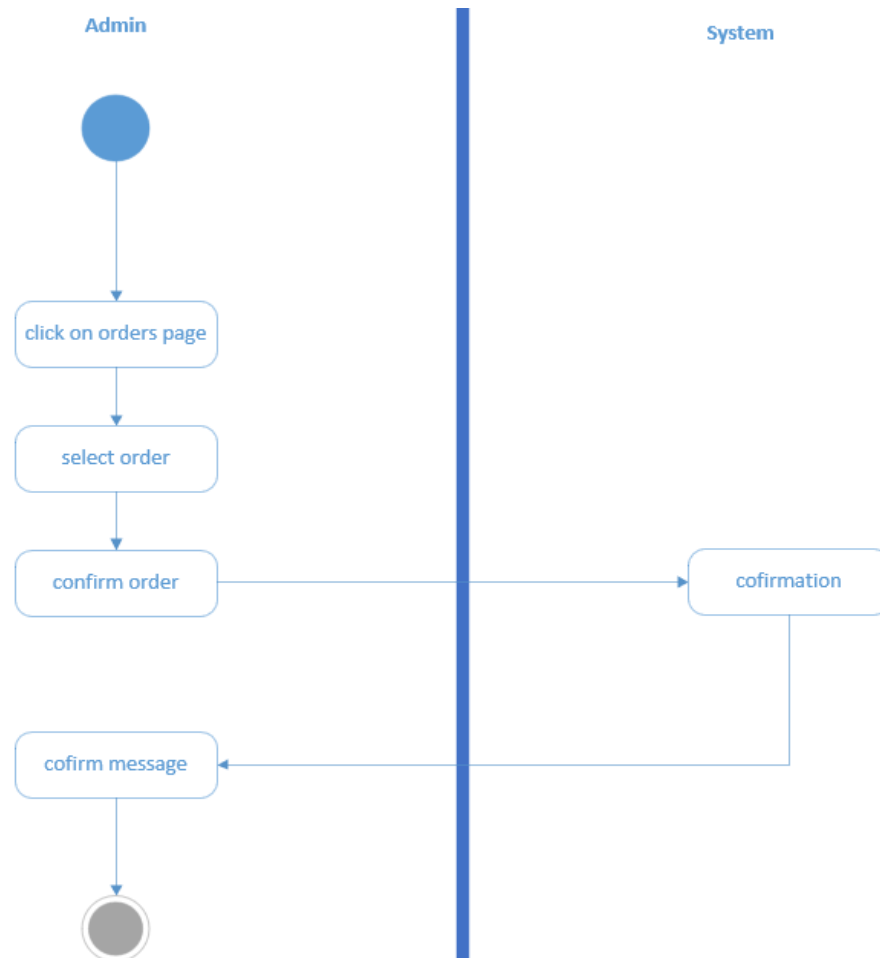
The diagram illustrates the process of modifying a recipe, divided into two swimlanes: Admin and System. The process begins with the Admin selecting a recipe to be modified. This action is then processed by the System, which adds modifications and submits the modification. A decision point follows, where the system checks if the modification is valid. If invalid, an error message is sent back to the Admin. If valid, the modification is added to the database, and a success message is sent back to the Admin, concluding the process.

c. Delete Recipe

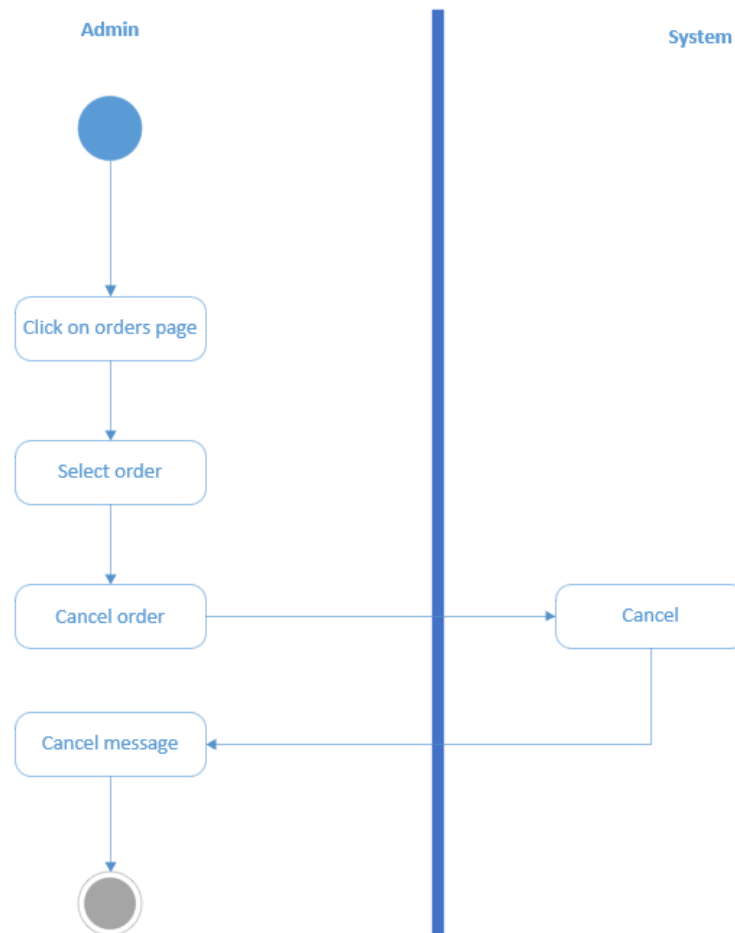


- **Manage Orders**

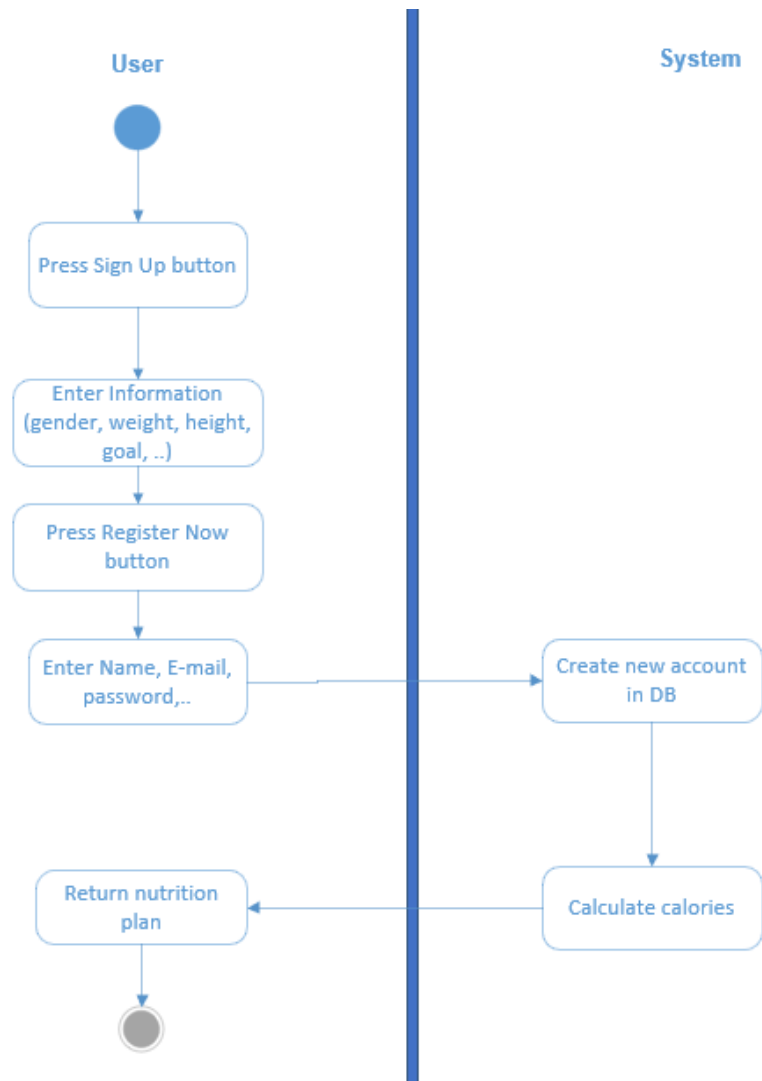
- a. Confirm Order



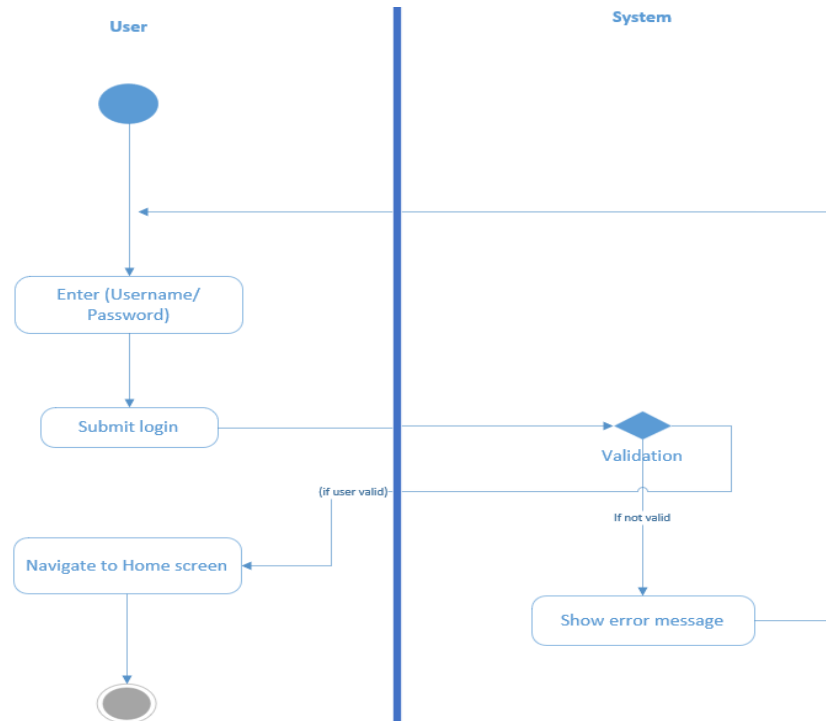
b. Cancel Order



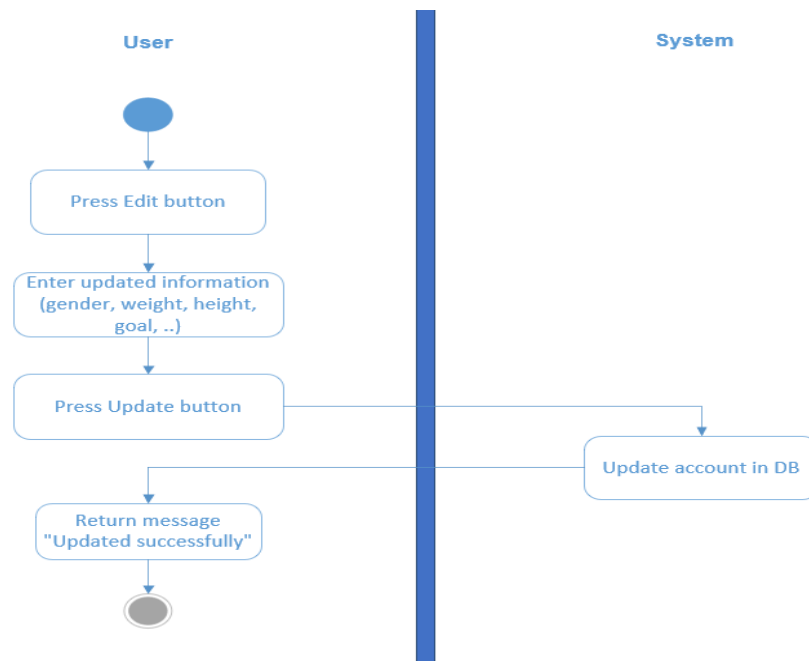
- User
 - Register and Add Information



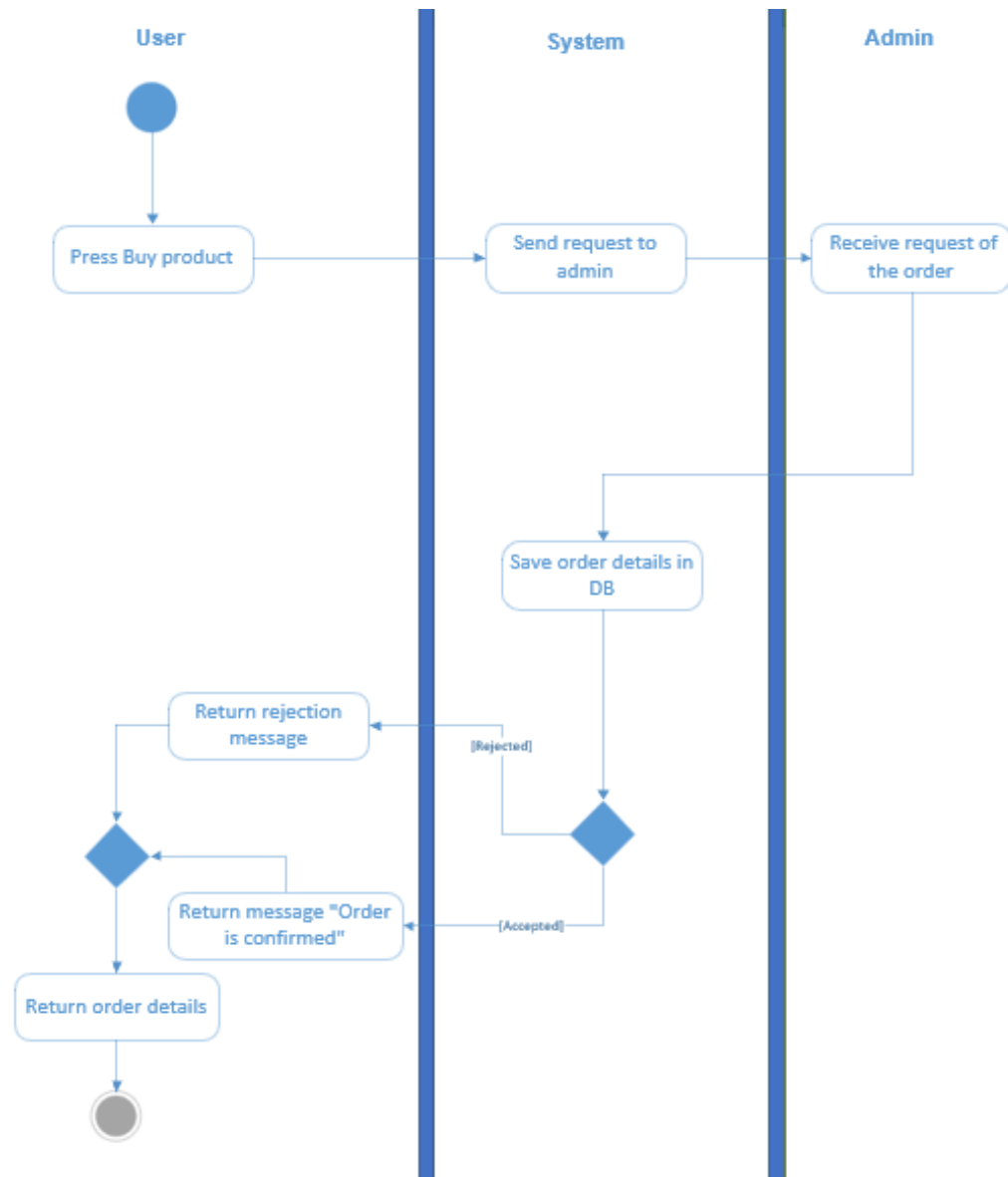
○ Login



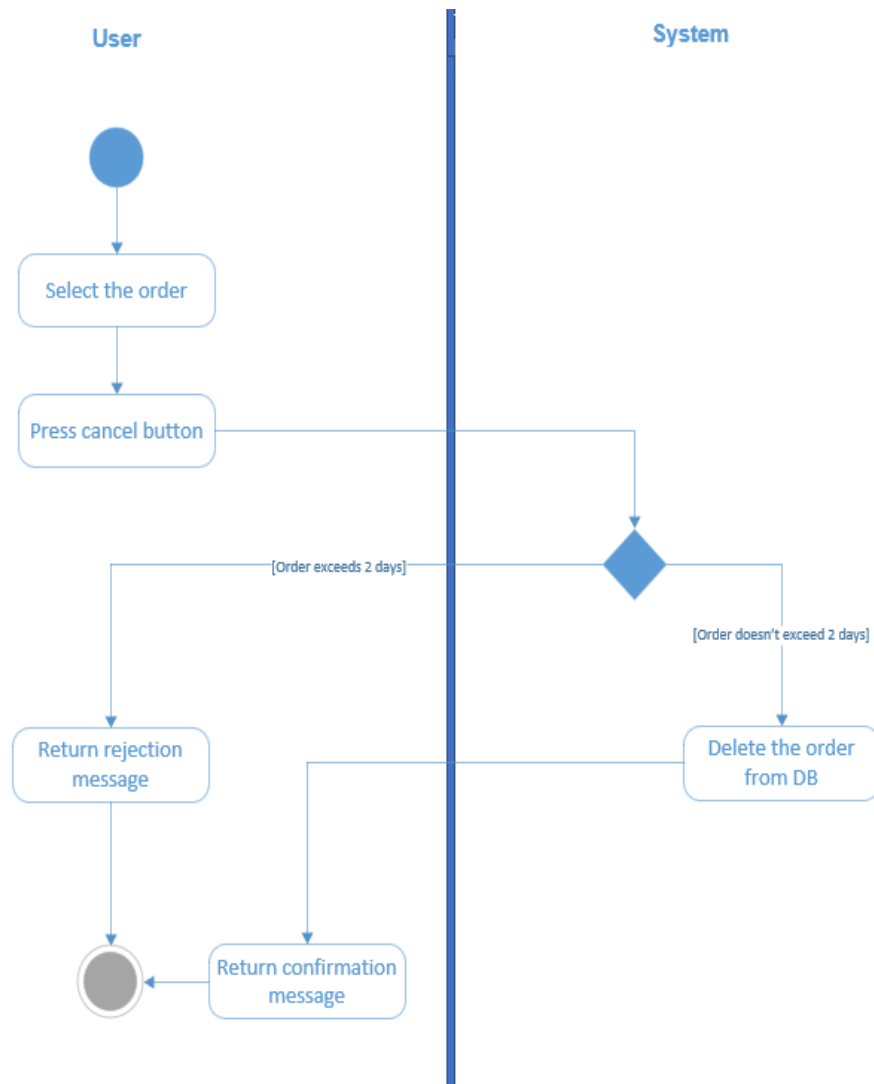
○ Edit Information



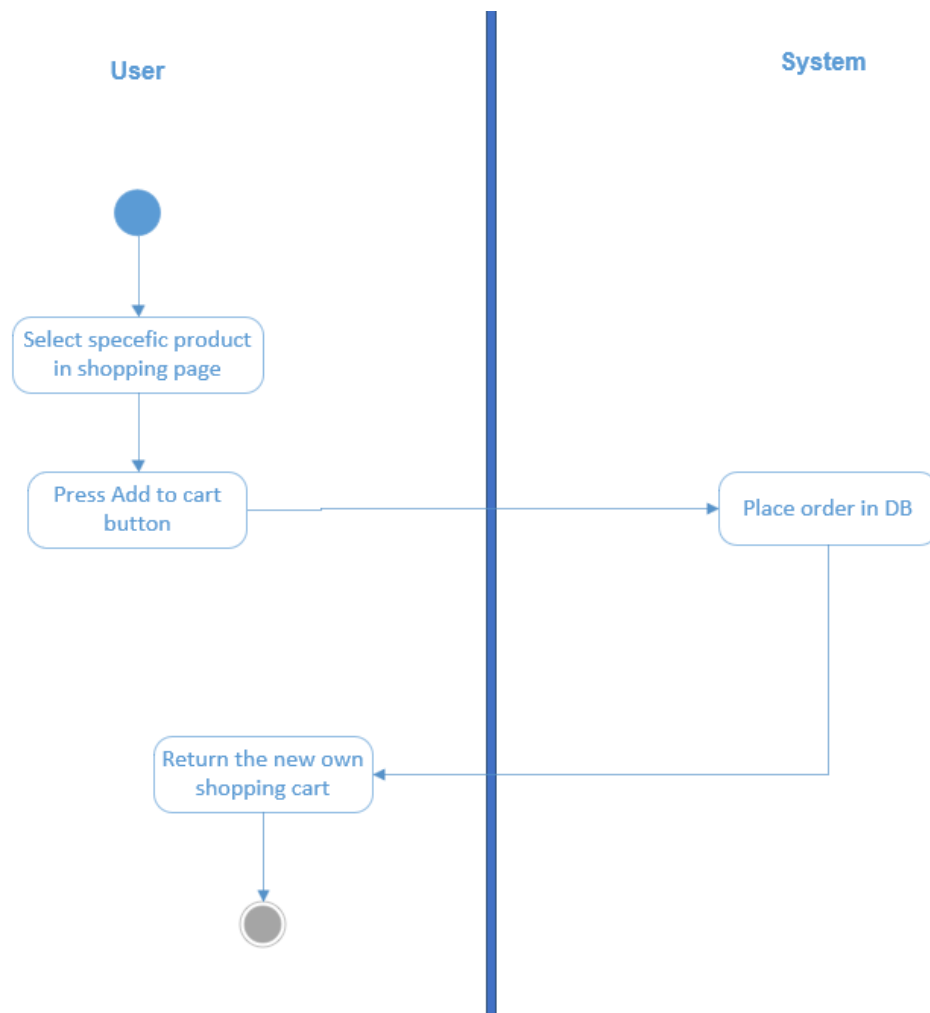
○ Buy Product



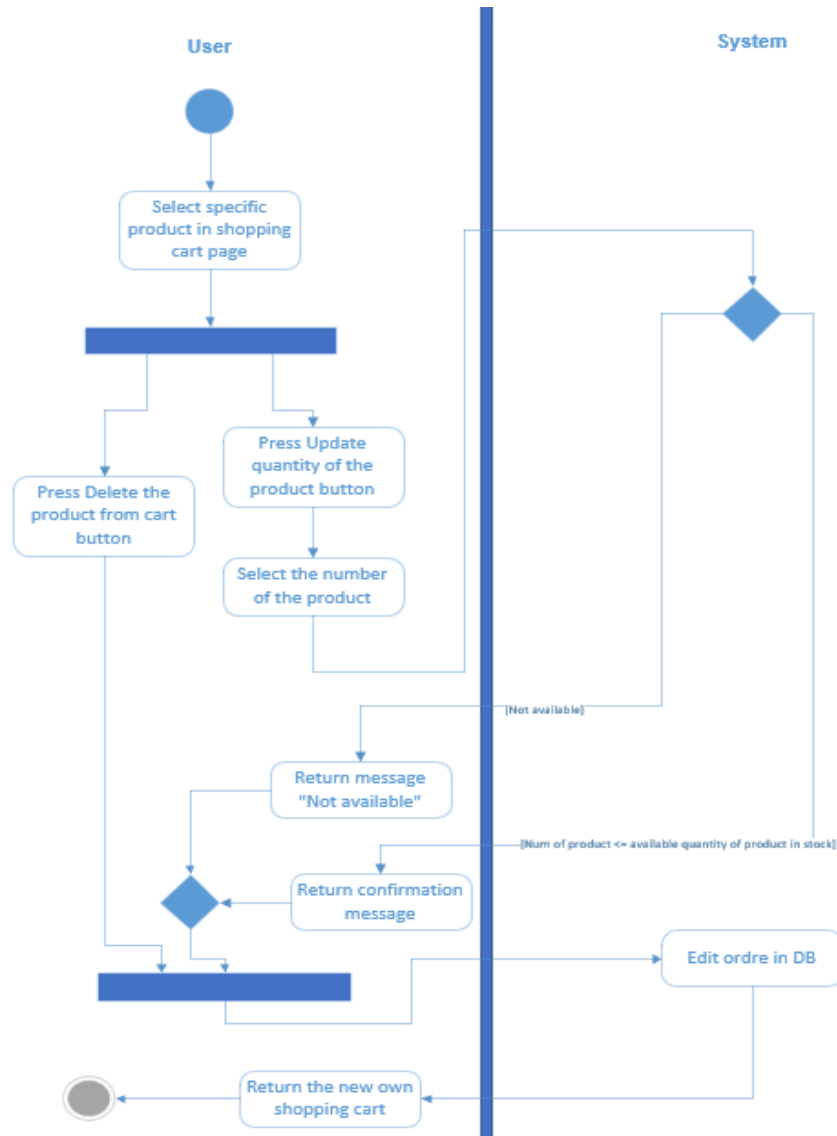
○ Cancel Order



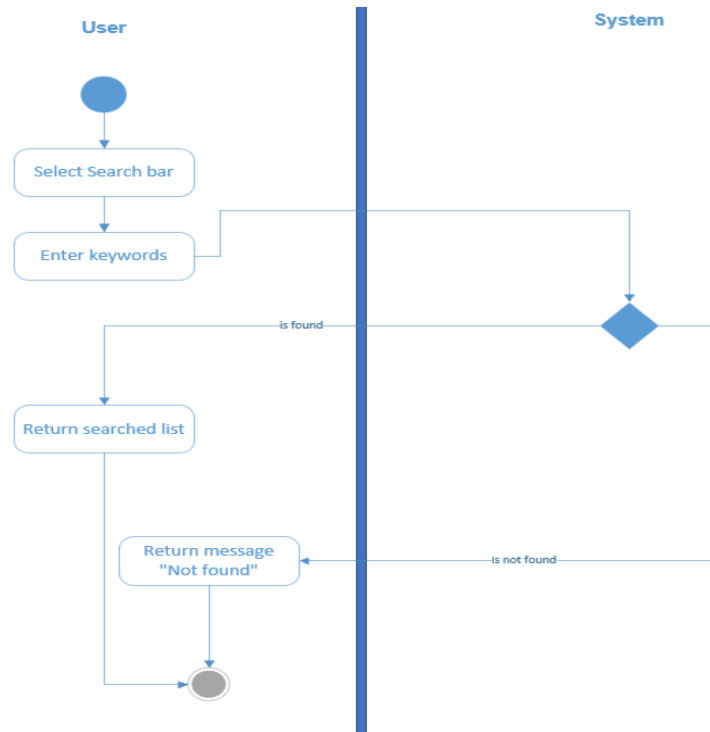
○ Add to Cart



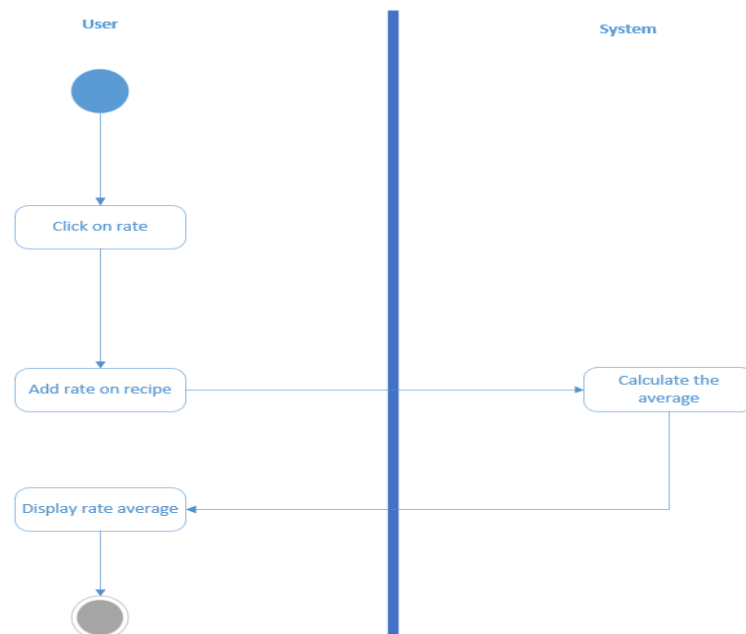
○ Delete /Update product from cart



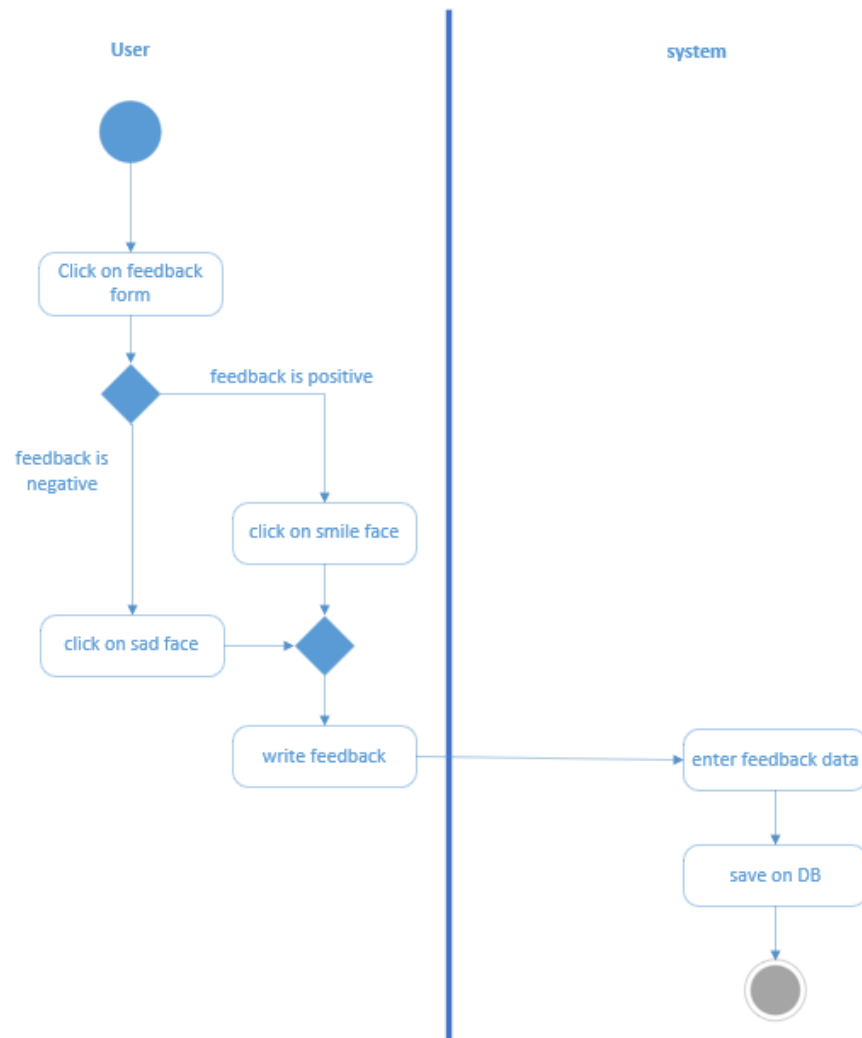
○ Search



○ Add Rate



○ Write Feedback

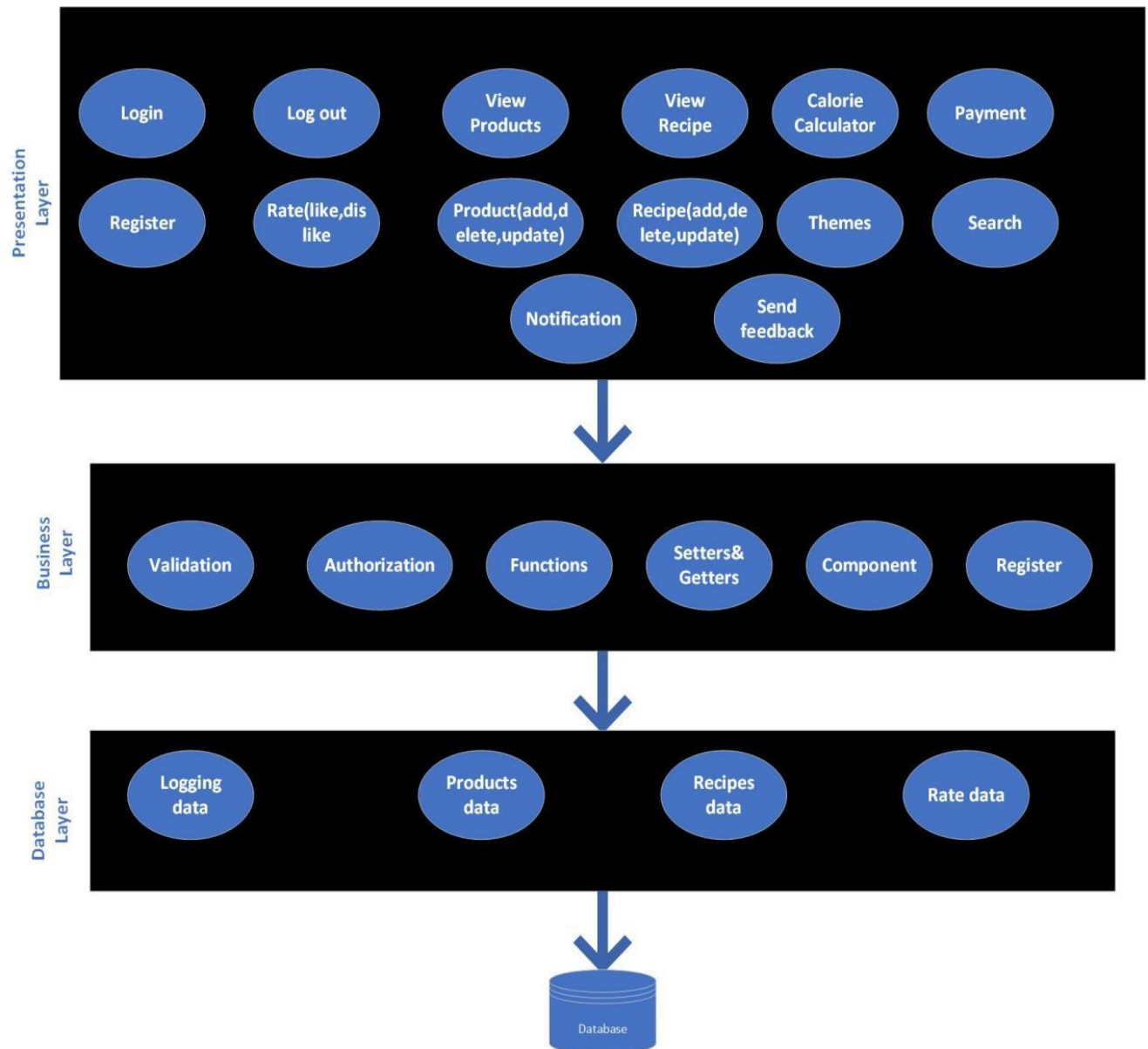


Chapter 4

Implementation

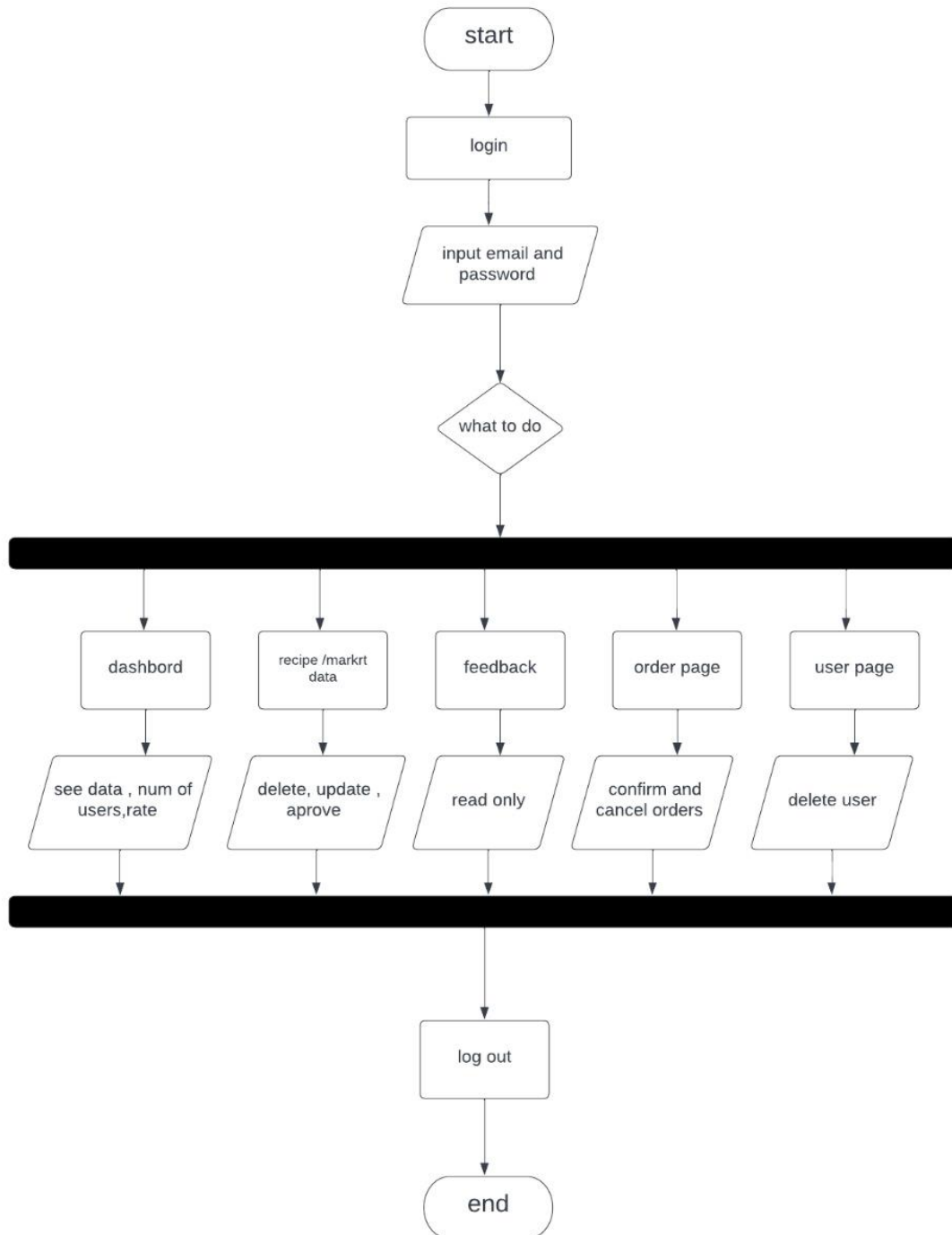
In this chapter we are going to discuss and go deeper in H. Partner mobile application's implementation and present its code and the algorithms used to build it.

4.1 Software Architecture

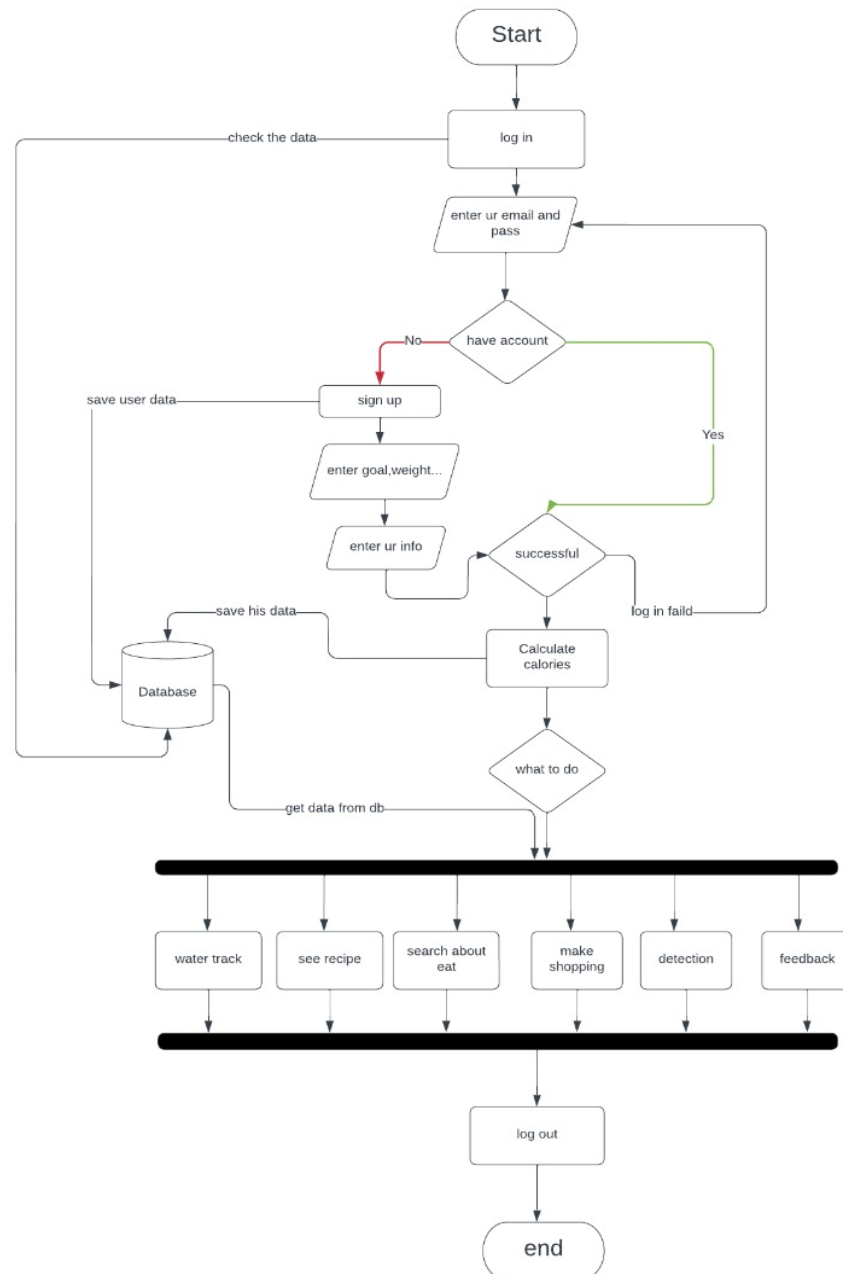


4.2 Flowchart

- Admin



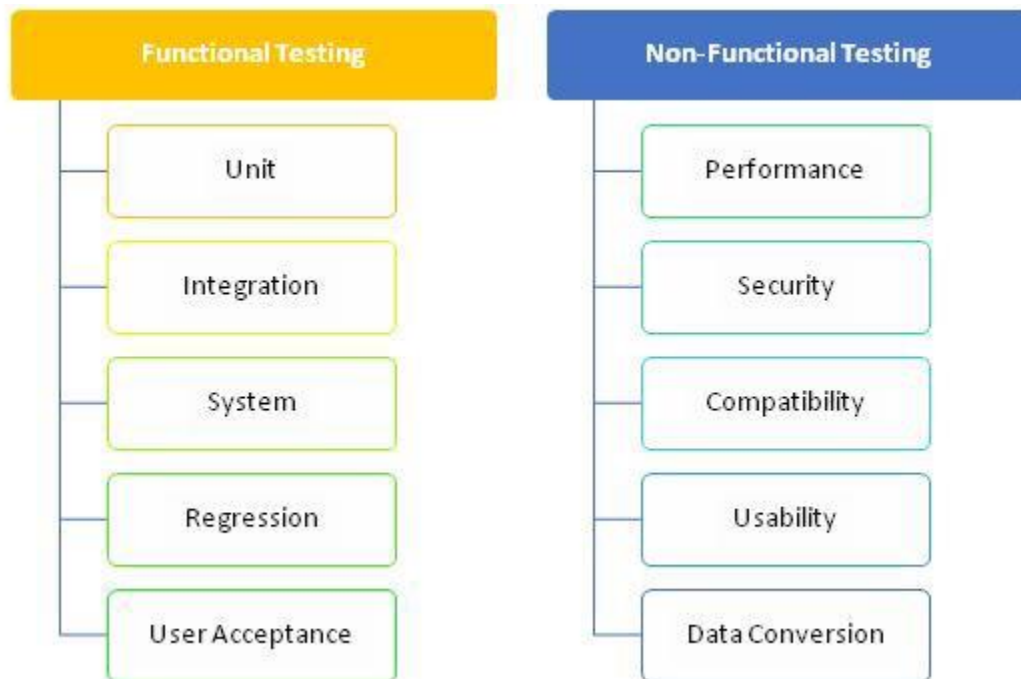
- User



Chapter 5

Testing

In this chapter we are going to discuss and go deeper in H. Partner mobile application's testing, and present the types of testing to be used and test cases we examined our application through.



Functional Testing:

5.1 Unit testing

Testing of individual items (e.g. modules, programs, objects, classes, etc.) usually as part of the coding phase, in isolation from other development item sand the system as a whole.

5.2 Integrated testing

Testing the interfaces between major (e.g. systems level application modules) and minor (e.g. individual programs or components) items with in an application which must interact with each other.

System testing

Testing a system behavior as a whole when development is finished, and the system can be tested as complete entity.

Regression testing

To check older functionality after integrating new functionality.

Acceptance testing

Testing to ensure that a development is ready to be deployed into the business, operational or production environment.

Non-Functional Testing:

Performance Testing

Accomplished a designated function regarding processing time and through put rate.

Load Testing

Measuring the behavior of within creasing load which can be handled by the component or system.

Stress Testing

Evaluate a system or component at or beyond the limits of its specified requirements.

Security Testing

Testing how well the system protects against unauthorized internal or external access.

5.3 Additional Testing

Test Cases:

Login test case:

Test Scenario Objective:

Verify log in successfully with correct email and correct password.

Assumptions/Dependencies:

Valid email: menna.olama2011@gmail.com

Valid password: 123456789

Step	Description	Expected Result	Actual Result	Error Type
1.	Open application	Application is opened		
2.	Navigate to log in screen.	User navigates to login page		
3.	Enter email: menna.olama2011@gmail.com	User can enter email		
4.	Press on text field called password	Show keyboard to type		
5.	Enter password: 123456	User can enter password		
6.	Click on login button	Log in will make successfully, home page will open	Passed	

Test Scenario Objective:

Make sure that message will appear to tell user that user should enter valid email when user enter email not in firebase.

Assumptions/Dependencies:

Invalid email: menna@gmail.com

Valid password: 123456789

Step	Description	Expected Result	Actual Result	Error Type
1.	Open application	Application is opened		
2.	Navigate to log in screen.	Show login page		
3.	Enter email: menna@gmail.com	User can enter email		
4.	Presse on text field called password	Show keyboard to type		
5.	Enter password: 123456789	User can enter password		
6.	Click on login button	Message will appear to tell you that this email is not valid, there is no user in firebase with this email, and still in log in page	Passed	

Test Scenario Objective:

Make sure that message will appear to tell you that you should enter Valid password when you enter wrong password.

Assumptions/Dependencies:

Valid email: menna.olama2011@gmail.com

Wrong password: 1234

Step	Description	Expected Result	Actual Result	Error Type
1.	Open application	Application is opened		
2.	Navigate to log in screen.	Show login page		
3.	Enter email: menna.olama2011@gmail.com	User can enter email		
4.	Presse on text field called password	Show keyboard to type		
5.	Enter password: 123456	User can enter password		
6.	Click on login button	Message will appear to tell you that this password is wrong, and still in log in page.	Passed	

Register test case:

Test Scenario Objective:

Make sign up successfully

Step	Description	Expected Result	Actual Result	Error Type
1.	Open application	Application is opened		
2.	Navigate to log in screen.	Show login page		
3.	Press on Sign up button	Navigate to add information screen		
4.	Choose gender	This choice will be colored blue		
5.	Choose goal	This radio box will be colored blue		
6.	Choose active level	This radio box will be colored blue		
7.	Enter your weight	The written number will show in text field		
8.	Enter your height	The written number will show in text field		
9.	Enter your goal weight	The written number will show in text field		
10.	Enter your age	The written number will show in text field		

11.	Press on Register Now button	Navigate to register screen		
12.	Enter Name	The written name will show in text field		
13.	Enter Email	The written email will show in text field		
14.	Enter password	The written pass will show invisible way		
15.	Upload photo if you want	Upload chosen photo		
16.	Press Register button	Sign up will make successfully, home page will open	Passed	

User's layout test case:

Test Scenario Objective:

Make an order from products page directly.

Step	Description	Expected Result	Actual Result	Error Type
1.	Navigate to products screen.	User navigate to products screen.		
2.	Choose a product you need it and tap on it	move to the product page of your choice		
3.	Click on "Buy now" button	The product has been ordered successfully	passed	

Test Scenario Objective:

Make an order from my cart page directly.

Step	Description	Expected Result	Actual Result	Error Type
1.	Navigate to products screen.	User navigate to products screen.		
2.	Choose a product you need it and tap on it	move to the product page of your choice		
3.	Click on “Add to cart” button	The product has been moved to your cart		
4.	Choose another product you need it and tap on it	move to the product page of your choice		
5.	Click on “Add to cart” button	The product has been moved to your cart		
6.	Navigate to my cart screen.	User navigate to my cart screen.		
7.	Click on “Buy now” button	The products have been ordered successfully	passed	

Admin's layout test case:

Test Scenario Objective:

Add new product/recipe, delete product/recipe and update/recipe product.

Step	Description	Expected Result	Actual Result	Error Type
1.	Navigate to products /recipes screen.	Admin navigate to products /recipes screen.		
2.	Click on "Add" button	Admin navigate to add product / recipe sheet.		
3.	Enter new product / recipe information	The admin entered all the text fields		
4.	Click on "Add" button	Add product will make successfully	passed	
5.	Choose a product / recipe you need to delete it and tap on it	move to the product/ recipe page of your choice		
6.	Click on "Delete" button	The product /recipe has been deleted	passed	
7.	Choose a product/ recipe you need to update it and tap on it	move to the product /recipe page of your choice		
8.	Click on "update" button	Admin navigate to update product /recipe sheet.		
9.	Enter edited product / recipe information	The admin entered all edited the text fields		
10.	Click on "Update" button	Update product / recipe will make successfully	passed	

Chapter 6

Result and Discussion

In this chapter we are going to find out the results of the project whether they are achieved or not and the differences between the desired results and the actual ones.

6.1 Results

6.1.1 Expected Results

- The user can register and login successfully.
- The app should calculate user calorie needs.
- The app should generate nutrition plan to the user.
- The app should display the ingredients for any type of food.
- The app should offer supplement products and user can buy it.
- The user can take a photo of any type of food and the app should detect what the food in this photo and describe the protein, calories, fat, carbohydrate, sugar, fiber, and certain vitamins and minerals inside the food.
- The app should generate reports.
- Admin should add product, update product , delete product.
- Admin should add recipe, update recipe, delete recipe.
- Admin should receive feedback, delete users.

6.1.2 Actual Results

- The user can register and login successfully.
- The app can calculate user calorie needs.
- The user generates his nutrition plan by himself, and the app helps him to know the minerals inside the food.
- The app displays the ingredients for any type of food.
- The app has market section to offer supplements products and the user can buy any of them.
- The user can take a photo of any type of food and the app should detect what the food in this photo and describe the protein, calories, fat, carbohydrate, sugar, fiber, and certain vitamins and minerals inside the food.
- The app can generate reports.
- Admin can add product, update product, and delete product.
- Admin can add recipe, update recipe, and delete recipe.

- Admin can receive feedback.
- Admin can delete users.

6.2 Discussion

- We managed to get the same expected result except that the app generate nutrition plan to user and the app helps him to know the minerals of any food to achieve his goal.
- We've added new features into our application such as: object detection that the user can take a photo of any type of food and the app should detect what the food in this photo and describe the protein, calories, fat, carbohydrate, sugar, fiber, and certain vitamins and minerals inside the food.

Chapter 7

Conclusion

The Conclusion chapter conclude the report by stating the task, the difficulties faced, experiences gained, results achieved and final thoughts on the project.

Finally, we did our best to get a perfect result. We also want to serve all of our society. We have chosen that we help all people to make life healthier and full of vitality and activity. This facilitates many tasks for people and reduces the percentage of diseases that may afflict a person if his diet is not healthy. Therefore, we seek to make life healthy for all. The state needs a society that is healthy, and sound, and this does not come unless the diet for everyone is healthy beginning from children to the elderly.

So, we hope that we can help all people to be healthy and their life to be better. We want to help people to do their tasks and duties without any problem.

Although we faced some difficulties such as:

- Learning new powerful technologies & tools such as machine learning.
- Studying a new technologies and techniques.
- Fixing errors and learning how to avoid repeating that mistake.
- Learning time management.
- Set deadlines and committed them.
- Get free resources.

We gained much experience like:

- Self-Study skills.
- Time management.
- Teamworking.
- Searching about a certain topic or a certain tutorial.
- Putting a plan.
- Dividing the tasks into sprints.
- How to apply a new technology.

Chapter 8

Future Work

In this chapter, we will talk about recommendations for future work and project enhancement should be included.

We are going to add more and more features to our application such as:

- 1- Add more data in foods dataset to be more useful and comprehensive for user.
- 2- Add barcode-scanning feature that allows users to scan an item's nutrition label.
- 3- Using recommendation for user to eat specific foods in every meal by daily way that depends on his goal, his medical history and if he is vegetarian or not.
- 4- Creating community that can help users to encourage each other to achieve their goals and communicate with each other about health life that they should live.
- 5- Add fitness section that helps users to do some exercises daily that make their life healthier.
- 6- Add nutrition experts to the application and make users can communicate with them if they need advice or something related to proper nutrition.

So, we will not miss any chance to keep working as a team on that project and enhancing it after graduation as we are looking forward to turning this application into a start-up.

Our dream started with this idea months ago, and we will insist on not letting go of it and not letting it be just a dream.

Bibliography

<https://firebase.flutter.dev/docs/overview/>

<https://flutter.dev/>

<https://api.flutter.dev/>

<http://rushrash.com/domain-requirement/>

https://en.wikipedia.org/wiki/User_requirements_document

https://en.m.wikipedia.org/wiki/Cost_estimate

**A
M
O
Y**

Advantageous

Main

Optimal

Yielding



H. Partner

- The best weight for you -

Thank You!