AI Smart Diet Consultant

Project ID: Winter-2022-4

Session: BSCS Fall 2018 to 2022

Project Advisor: Sir Awais Ali

Submitted By

Muhammad Ali	70067089
Usama Shahzad	70067080

Department of Computer Science & IT
The University of Lahore
Lahore, Pakistan

Declaration

We have read the project guidelines and we understand the meaning of academic dishonesty, in particular plagiarism and collusion. We hereby declare that the work we submitted for our final year project, entitled **AI Smart Diet Consultant** is original work and has not been printed, published or submitted before as final year project, research work, publication or any other documentation.

Group Member 1 Name: **Muhammad Ali**
SAP No: **70067089**

Signature: **…………………………**

Group Member 2 Name: **Usama Shahzad**
SAP No: **70067080**

Signature: **…………………………**

Statement of Submission

This is to certify that **Muhammad Ali** Roll No. 70067089**, Usama Shahzad** Roll No. 70067080 have successfully submitted the final project named as: **AI Smart Diet Consultant**, at Computer Science & IT Department, The University of Lahore, Lahore Pakistan, to fulfill the partial requirement of the degree of **BS in Computer Science**.

Supervisor Name: Awais Ali

Signature: …………………………

Date: ………………………

Dedication

This project is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

Acknowledgement

. We are very thankful to Allah Almighty who gave us the courage and zeal to perform this work successfully. We would like to express our gratitude to the many people who saw us through this documentation, to all those who provided support, talked things over, read, wrote, offered comments, allowed us to quote their remarks, and assisted in the editing, proofreading, and design.

A study or a project of this volume can never be the outcome of a single person or just a mere group of dedicated Suppliers. We are grateful to our Supervisor Sir Awais Ali, Assistant Professor of University of Lahore for being the essence of guidance during the entire project. Without his help, this project would not have been completed. We want to thank the rest of our family, who supported and encouraged us despite all the time it took them to help us complete this degree program. It was a long and difficult journey for them.

We would like to thank our team for helping in the process of selection and editing. Last and not least: We beg forgiveness of all those who have been with us over the years and whose names we have failed to mention.

Date:

June 27, 2022

Abstract

This project is based on Android Artificial Intelligence smart diet consultant app. Which is basically based on user body type, weight, height, age, sex and physically activity. It calculates BMI an BMR and tells us how much calories should user to intake per day. This application is for any type of body person, it also suits for any range of weight people. It will ideally inform how should we cut down the weight using target program such as 1LB per week gain/loose. This system also provides the diet plan according to the information that send by the user. The system asks all his data from the user and processes it to provide the diet plan to the user. The user does not need to visit any dietitian which also saves time and the user can get the required diet plan in just one click. The system will give more accurate results as it accepts the data entered by the user and processes it depending on some metrics already known to the application on the basis of which a diet plan is generated and ask the user if the user accepts the diet plan. If not accepted the system may also give and alternative diet plan.

Area of the Project

Android Application, Artificial Intelligence

Technologies used

Visual studio with React Native, PHP(Laravel), Python, SQL

List of Figures

List of Tables

Table of Content

Chapter 1: Introduction to the Problem

1.1 Introduction

This project is based on Android Artificial Intelligence smart diet consultant app. Which is basically based on user body type, weight, height, age, sex and physically activity. It calculates BMI an BMR and tells us how much calories should user to intake per day. This application is for any type of body person, it also suits for any range of weight people. It will ideally inform how should we cut down the weight using target program such as 1LB per week gain/loose. This system also provides the diet plan according to the information that send by the user. The system asks all his data from the user and processes it to provide the diet plan to the user. The user does not need to visit any dietitian which also saves time and the user can get the required diet plan in just one click. The system will give more accurate results as it accepts the data entered by the user and processes it depending on some metrics already known to the application on the basis of which a diet plan is generated and ask the user if the user accepts the diet plan. If not accepted the system may also give and alternative diet plan[1].

Artificial Intelligence (AI) has gained more popularity among computational tools. The application relies on Artificial Intelligence to produce custom data related to calorie intake and accordingly make food suggestions. It is a smart calorie counter powered by AI. Their artificial intelligence learns an individual's tastes, preferences, and body type and then tells exactly what to eat. The app's machine-learning algorithm gathers data about individuals from their health tracking devices and food logging to personalize food suggestions using a bot to deliver answers throughout the day. Artificial Intelligence and its subsets have been leveraged by these platforms to identify the calorie intake and then make food recommendations for a healthy diet[2].

1.2 Purpose

The technique of artificial intelligence, you will get access to all the facilities via this application, which is actually provided by a human dietitian. The main advantage of using this standalone AI Diet Consultant application is that the time required by the people to travel to the dietitian will be reduced and also it reduces the cost of hiring dietitians for some particular purpose. Also, this AI Diet Consultant application offers more than one diet plan also, for some particular kind of functionalities of human bodies[3].

1.3 Objective

- This project is based on Rete algorithm it gives the prediction/suggestion of the food according to the attributes entered by the user[4].
- The project helps users to check their Body Mass Ratio and also helps to choose their own choice of food items according to their body type along with calorie calculation.
- They can choose the food items of their choice and be healthy by keeping track of calories of the food items added to the list.
- This project is successfully able to suggest various combination of food items and even if the user needs a custom diet according to their choices they can select and implement.

1.4 Existing Solution

There are a lot of application like Smart Diet Planner and Healthy Diet app but we use artificial intelligence in it then user can more accurately diet plan in our application and also if user want to add more diet plan in the table list, then user can change the diet plan according to their choice.

1.5 Proposed Solution

We use Artificial Intelligence to make user diet plan more accurate in it. If user want more diet plan in the diet chart then user send the request to the dietitian for more diet plan.

Chapter 2: Software Requirement Specification

2.1 Introduction

2.1.1 Purpose

This subsection should

1. **Delineate the purpose of the SRS**

The Purpose of the Software Requirements Specification document is to maintain all the functions and the specifications of 'AI SMART DIET CONSULTANT APP'. Besides it contains detailed descriptions of all the requirements specified.

2. **Specify the intended audience for the SRS.**

This intended audience of the document includes all stakeholders of AISDCA project who are supposed to review and sign-off this document. The primary stakeholders of the system include registered users, guest users.

2.1.2 Scope

This subsection should

This project is based on Android Artificial Intelligence smart diet consultant app. Which is basically based on user body type, weight, height, age, sex and physically activity.

This system also provides the diet plan according to the information that send by the user. The system asks all his data from the user and processes it to provide the diet plan to the user. The user does not need to visit any dietitian which also saves time and the user can get the required diet plan in just one click[5].

- We plan to focus on improving the overall performance of the system. Also, interaction between guider and dietitian through video calling and secure prescription will be focused upon.
- This application can be improved with the help of an expert nutritionist who can help us creating different types of programs for different classification of users
- This app is used for diet consultant. The user does not need to visit any dietitian or any nutrition.
- This app will be the automation of the dietitian plan. App will maintain the dietitian planed also have the option to give suggest their respective need.
- The project helps users to check their Body Mass Ratio and also helps to choose
their own choice of food items according to their body type along with calorie
calculation. This project is successfully able to suggest various combination of food items and even if the user needs a custom diet according to their choices they can select and implement[1].

2.1.3 Definitions, acronyms, and abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents[6].

So here what are definitions, acronyms, and abbreviations.

Definitions: is define the exact functionality of the app through which a user can easily understand the meaning of app.

Abbreviations: AISDC acronyms of AI Smart Diet Consultant.

2.2 Overall description

2.2.1 Product perspective

This project is intended to maintain the dietitian plan of the user enter your weight height. This application is customizable and can be used by smart diet consultancy. Although we will not describe every requirement in detail, this section will describe the factors that affect the final product[7].

This subsection is sub-divided into the following

- **System interfaces**
  The user shall be able to access the app user from app store like Play store

- **Hardware interfaces**
  On the Spot App user shall let users log in to their personal profiles. When accessing the app user, the user will be asked to either sign-in with their username and password, sign up as a user for the app user[8]

- **Software interfaces**
  This app is used for only android user only.

- **Communications interfaces**
  The system needs internet connection. Without the internet, it would not be able to login this application[9].

- **Memory**
  Hard Drive – up to 2 GB of free space to store Dietitian and User records.
  RAM: it requires a minimum of 2GB ram to run its features

- **Operations**
  A backup of the database will be created at the end of the day, and the previous backup will be deleted to overcome space issues. If the database is corrupted, the system will be able to recover it from backed-up databases[10]

- **Site adaptation requirements**

  Smart Diet Consultant is a Android application, so it does not require any adaptation.

2.2.2 Product functions

- System would provide different interfaces and views for admin and end-users.
- The system shall provide the diet on the basis of the body type and fat percentage of the user.
- The system can send alert message before maintenance.
- The system can send update notification
- The system shall allow the user to view different food items and their diet plan.
- The system shall allow the users to post their reviews.
- The system shall allow the user to call our dietitian when its need.
- The system can suggest the dietitian or nutrition.
- The user will be able to login using username and password
- The user can change his account password using the 'change my password.' Option
- The user can update our profile
- The user can send custom request to dietitian for add more diet plan
- Admin will be able to login into the system using username and password
- Admin would be able to add any dietitian
- Admin can remove any dietitian
- Admin can remove any user.
- Dietitian will be able to login using username and password
- Dietitian would be able to add different food item about user diet plan.
- Dietitian would be able to update food item
- Dietitian can recommend diet plan for user need.
- Dietitian would be able to update the user diet chart for add more diet which user send custom request
- Dietitian can check user’s reviews

2.2.2.1 User

2.2.2.1.1 Create Account

ID:	FR_01			
Name:	Create Account			
Description	Input	Output	Requirements	Basic Work Flow
Enter details to create account	Name, Phone number, Email, Password, Confirm Password etc.	Account created	Internet Connectivity required	Add all the required details, and then click on submit button. The system will save the record in the database

Table 1 Functional Requirement Create Account for User

2.2.2.1.2 Login Account

ID:	FR_02			
Name:	Login Account for User			
Description	Input	Output	Requirements	Basic Work Flow
Enter details to login account	Email, Password	Logged in into system as user.	Already registered in the system	Enter the email and password and then click on the login button to use the system as a user.

Table 2 Functional Requirement Create Account for User

2.2.2.1.3 Change Password

ID:	FR_03			
Name:	Change password			
Description	Input	Output	Requirements	Basic Work Flow
Enter details to change password	Old password, new password	Password changed successfully	User should enter old password to change password	Enter the old password and new password and hit change password button; the password will be updated in the database.

Table 3 Functional Requirement Change Password

2.2.2.1.4 Create Profile

ID:	FR_04			
Name:	Create Profile			
Description	Input	Output	Requirements	Basic Work Flow
Enter details to update profile	Age, Height (in cms), Weight, Phone number etc.	Profile changed successfully.	User should select the input (name, age, weight etc) to create the personal profile	Enter the details of all information in the Create Diet Plan. System saved the record in database

Table 4 Functional Requirement Create Profile

2.2.2.1.5 Update Profile

ID:	FR_05			
Name:	Update Profile			
Description	Input	Output	Requirements	Basic Work Flow
Enter details to update profile	Age, Height (in cms), Weight, Phone Number etc.	Profile changed successfully.	User should select the input (name, age, weight etc) Which user wants to update.	If user wants to upgrade the profile with the previous one then user click on the profile to update. System updated the record in database

Table 5 Functional Requirement Update Profile

2.2.2.1.6 Feedback

ID:	FR_06			
Name:	Feedback for User			
Description	Input	Output	Requirements	Basic Work Flow
Enter the feedback.	Feedback	Feedback submitted successfully.	Admin daily see the feedback.	User enter the feedback about any type of confusion or working about app then admin can see the user feedback and to solve the problems. System saved the feedback in the database.

Table 6 Functional Requirement Feedback for User

2.2.2.1.7 Custom Request

ID:	FR_07			
Name:	Custom Request for User			
Description	Input	Output	Requirements	Basic Work Flow
Enter the request to add more diet plan.	Request for more diet plan	Request submitted successfully.	Dietitian daily see the User request.	If User want more food in the table food lists then User click on the custom request button to request the dietician for some more food in the diet plan. System saved the request in the database.

Table 7 Functional Requirement Custom Request for User

2.2.2.2 Admin

2.2.2.2.1 Login Account

ID:	FR_08			
Name:	Login Account for Admin			
Description	Input	Output	Requirements	Basic Work Flow
Enter details to login account	Email, Password	Logged in system as an admin.	Already registered in the system	Enter the email and password and then click on the login button to use the system as an admin

Table 8 Functional Requirement Login account for Admin

2.2.2.2.2 Add Dietitian

ID:	FR_09			
Name:	Add Dietitian for Admin			
Description	Input	Output	Requirements	Basic Work Flow
Enter details of all input field of dietitian	Name, Age, Gender, Email, Password	Data submitted successfully	Enter all the input field of dietitian and click on the submit button	Only Admin can add the dietitian by entering the input detail of dietitian and click on the submit button. System saved the data of dietitian in the database

Table 9 Functional Requirement Add Dietitian for Admin

2.2.2.2.3 Remove Dietitian

ID:	FR_10			
Name:	Remove Dietitian for Admin			
Description	Input	Output	Requirements	Basic Work Flow
Remove a Dietitian from the database	Dietitian Id	Data deleted successfully	Only admin can remove the dietitian. Dietitian should already be registered	Enter the id of the dietitian and click on the delete record button. The system will delete the dietitian from the database

Table 10 Functional Requirement Remove Dietitian for Admin

2.2.2.2.4 Remove User

ID:	FR_11			
Name:	Remove User for Admin			
Description	Input	Output	Requirements	Basic Work Flow
Remove a User from the database	User Id	Data deleted successfully	User should already be registered	Enter the id of the user and click on the delete record button. The system will delete the user from the database

Table 11 Functional Requirement Remove User for Admin

2.2.2.3 Dietitian

2.2.2.3.1 Login Account

ID:	FR_12			
Name:	login for Dietitian			
Description	Input	Output	Requirements	Basic Work Flow
Enter details in login account	Email, Password	Logged in system as a Dietitian.	Already registered in the system	Enter the email and password and then click on the login button to use the system as a Dietitian

Table 12 Functional Requirement Login Account for Dietitian

2.2.2.3.2 Add Food Item

ID:	FR_13			
Name:	Add Food Item for Dietitian			
Description	Input	Output	Requirements	Basic Work Flow
Enter details in the food Item	Food Name, carbohydrates, Fats, Proteins Consumed by (Underweight, Healthy, Over Weight) Consumed by (Breakfast, Lunch, Dinner)	Data Submitted successfully	Admin should select all input field to add the Food.	Enter the detail of all information in the food item and click on the submit button. System save the data in the database.

Table 13 Functional Requirement Add Food Item for Dietitian

2.2.2.3.3 Update Food Item

ID:	FR_14			
Name:	Update Food Item for Dietitian			
Description	Input	Output	Requirements	Basic Work Flow
Enter details the update the food item	Food Name, carbohydrates, Fats, Proteins Consumed by (Underweight, Healthy, Over Weight) Consumed by (Breakfast, Lunch, Dinner)	Data Updated successfully	Admin should select the input field which admin wants to update the Food.	Admin click on the update button which admin want to update the food item. System updated the record in database

Table 14 Functional Requirement Add Food Item for Dietitian

2.2.2.3.4 Update Diet Chart

ID:	FR_15			
Name:	Update Diet Chart for Dietitian			
Description	Input	Output	Requirements	Basic Work Flow
After submit the request by user for more diet plan then Dietitian update the diet chart of user	Custom Request	User Chat updated successfully	User must be login and request dietitian using custom request	User can request the dietitian to update diet chart according to his choice then dietitian update the diet chart according to user custom request.

Table 15 Functional Requirement Update Diet Chart for Dietitian

2.2.2.3.5 View Feedback

ID:	FR_16			
Name:	View Feedback for Dietitian			
Description	Input	Output	Requirements	Basic Work Flow
Dietitian can see user feedback in it	Click on View Feedback Button	View Feedback	Dietitian must be login in	User feedback saves in database. When user click view feedback button the system shows feedback about user that's saved in database.

Table 16 Functional Requirement View Feedback for Dietitian

2.2.3 User characteristics

Admin: Admin must have necessary skills and some data entry skills.

End-Users: They are not required to have any technical knowledge or are not bounded by educational level or technical expertise. Anyone would be able to use this system easily.

2.2.4 Constraints

2.2.4.1 Hardware Limitations

The system will be able to run on Android devices.

2.2.4.2 Interface

This system will provide application interface

2.2.4.3 Database

The system will be designed using SQLite Database

2.2.4.4 Control function

Admin will have all the controls over the system to update or maintain database.

2.2.5 Assumptions and dependencies

- The device must have a working internet connection
- Android devices meet the minimum storage requirement to install this application
- The system has already created the administrator

2.2.6 Apportioning of requirements

This subsection of the SRS should identify requirements that may be delayed until future versions of the system.

2.3 Specific requirements

2.3.1 Functional Requirement

- System would provide different interfaces and views for admin and end-users.
- The system shall provide the diet on the basis of the body type and fat percentage of the user.
- The system can send alert message before maintenance.
- The system can send update notification
- The system shall allow the user to view different food items and their diet plan.
- The system shall allow the users to post their reviews.
- The system shall allow the user to call our dietitian when its need.
- The system can suggest the dietitian or nutrition.
- The user will be able to login using username and password
- The user can change his account password using the 'change my password.' Option
- The user can update our profile
- The user can send custom request to dietitian for add more diet plan
- Admin will be able to login into the system using username and password
- Admin would be able to add any dietitian

- Admin can remove any dietitian
- Admin can remove any user.
- Dietitian will be able to login using username and password
- Dietitian would be able to add different food item about user diet plan.
- Dietitian would be able to update food item
- Dietitian can recommend diet plan for user need.
- Dietitian would be able to update the user diet chart for add more diet which user send custom request
- Dietitian can check user's reviews

2.3.2 Non-functional Requirements

- **Usability:** We would implement a user-friendly, interactive and user can easily understand and use it.
- **Performance:** The application would be made quick and responsive to run smoothly on any Android devices The system should be able to respond on time. The system shall allow the users to post their review.
- **Maintainability:** The application would make it easier to maintain. For instance, we have divided our application into Admin and users, further divided into smaller modules. Before maintenance the system sent alert message for our user about maintenance.
- **Reliability:** The application would be made on software standards and will be error-free. Users can rely on our system.
- **Security:** Administrators of the system will have access modify access to database and no other will be allowed to alter database. The database would be secure as there will be no external access to database This app can meet all security features to safe user record.

Chapter 3: Use Case Analysis

3.1 Use Case Diagram for User

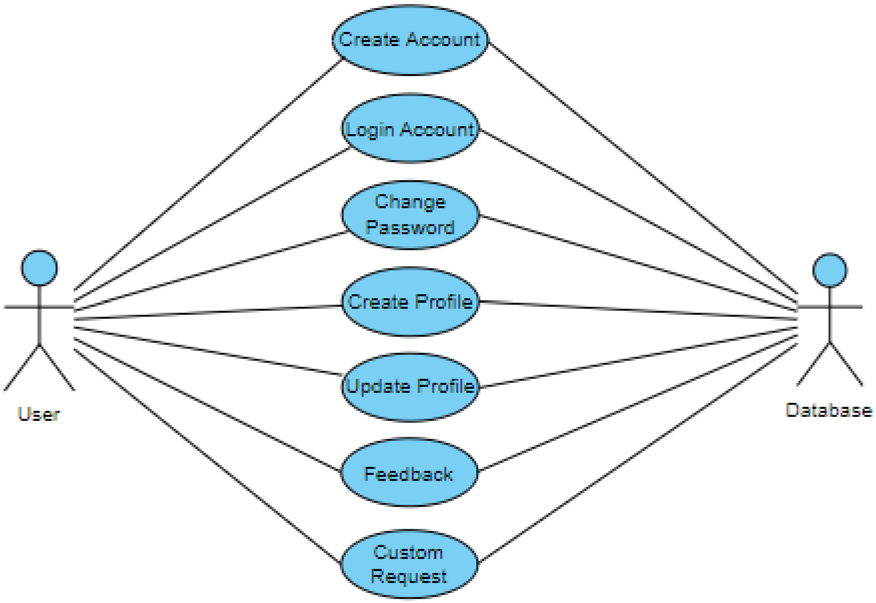


Figure 1 Use Case Diagram for User

3.2 Use Case Diagram for Admin

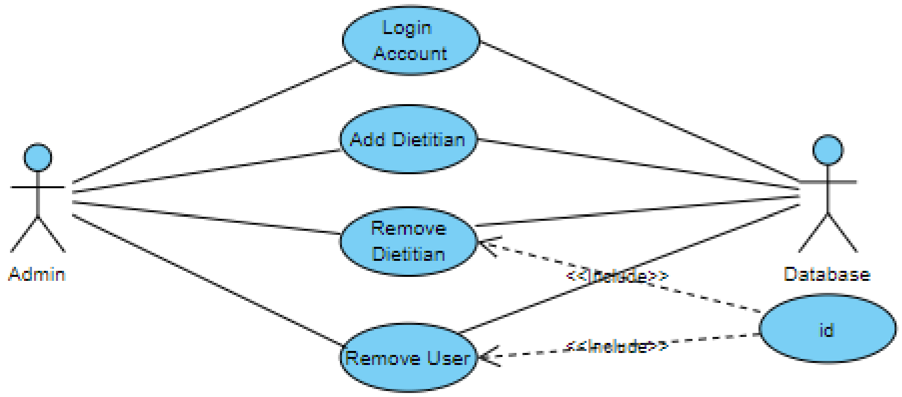


Figure 2 Use Case Diagram for Admin

3.3 Use Case Diagram for Dietitian

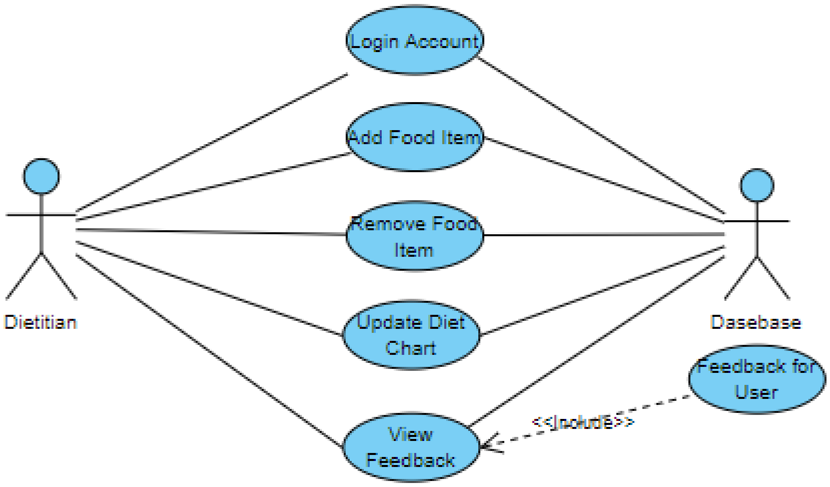


Figure 3 Use Case Diagram for Dietitian

3.1 User

3.1.1 Create Account

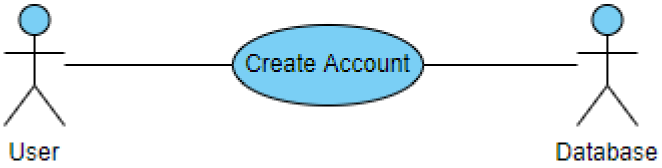


Figure 4 Use Case Diagram Create Account for User

Use Case ID	UC_01	
Use Case Name	Create Account for User	
Description	It will add the User registration detail to the database	
Primary Actor	User	
Secondary Actor	Database	
Pre-Condition	Logged in as Internet required	
Post-Condition	Record submitted successfully in database	
Basic Flow	Actor Action	System Action
	Click on click here button Enter Name, Phone number, Email, Password, Confirm Password	The system will store the details in the database and displays a message "Record submitted successfully" on the screen
Alternate Flow	Some required fields are empty, the record not saved	

Table 17 Use case Create Account for User

3.1.2 Login Account

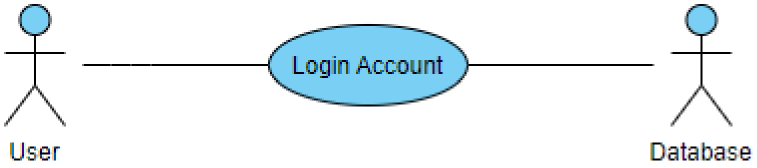


Figure 5 Use Case Diagram Login Account for User

Use Case ID	UC_02	
Use Case Name	Login Account for User	
Description	It will allow the user to login	
Primary Actor	User	
Secondary Actor	Database	
Pre-Condition	The user has an Email and password	
Post-Condition	Logged in into system as User	
Basic Flow	Actor Action	System Action
	Click on the login as admin button Enter Email and password Click on the login button	The system would check the credentials and will log in the user account
Alternate Flow	Email or password is incorrect, again on the login screen	

Table 18 Use case Login Account for User

3.1.3 Change Password

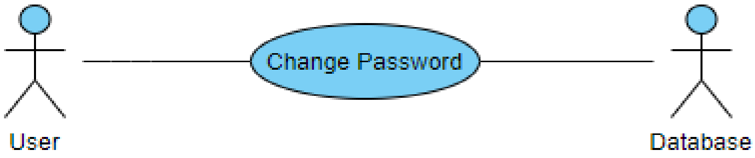


Figure 6 Use Case Diagram Change Password for User

Use Case ID	UC_03	
Use Case Name	Change Password for User	
Description	It will change the user password	
Primary Actor	User	
Secondary Actor	Database	
Pre-Condition	Logged in as user	
Post-Condition	user password has been updated	
Basic Flow	Actor Action	System Action
	Click on change password button Enter the old password and new password Click on the change password button	The system will match the entered password with the old password, and if it is correct, then the system will update it with the new password
Alternate Flow	The old password doesn't match, again on the change password screen	

Table 19 Use case Change Password for User

3.1.4 Create Profile

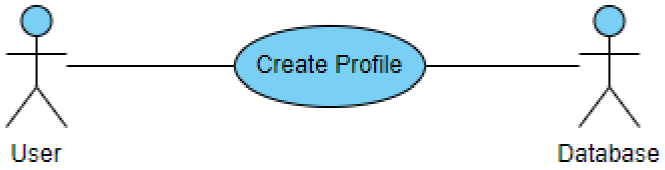


Figure 7 Use Case Diagram Create Profile for User

Use Case ID	UC_04	
Use Case Name	Create Profile for User	
Description	It will add the registration detail to the database	
Primary Actor	User	
Secondary Actor	Database	
Pre-Condition	Logged in as Internet required	
Post-Condition	Record submitted successfully in database	
Basic Flow	Actor Action	System Action
	Click on create profile button Enter Age, Height, Weight, Gender	The system will store the details in the database and displays a message "Record submitted successfully" on the screen
Alternate Flow	Some required fields are empty, the record not saved	

Table 20 Use case Create Profile for User

3.1.5 Update Profile

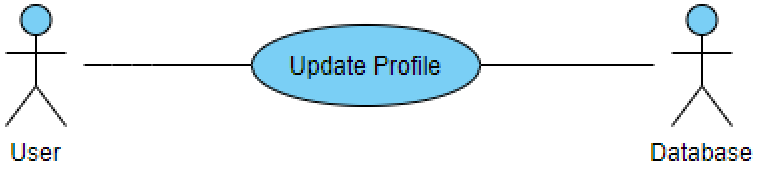


Figure 8 Use Case Diagram Update Profile for User

Use Case ID	UC_05	
Use Case Name	Update Profile for User	
Description	It will update the profile detail in the database	
Primary Actor	User	
Secondary Actor	Database	
Pre-Condition	Logged in as User Internet required	
Post-Condition	Record updated successfully in database	
Basic Flow	Actor Action	System Action
	Click on update student button Change required Name, Age, Height (in cms), Weight, Phone number	The system will update the details in the database and displays a message "Record updated successfully" on the screen
Alternate Flow	Some required fields are empty	

Table 21 Use case Update Profile for User

3.1.6 Feedback

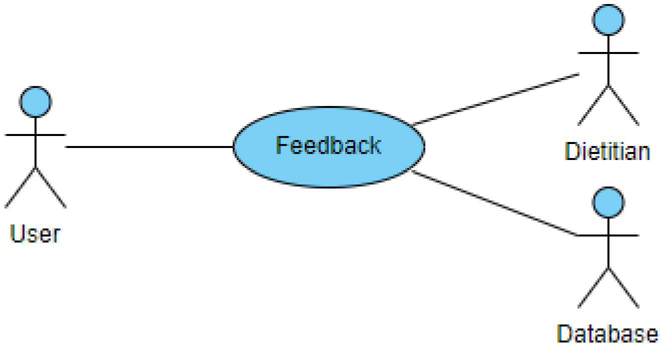


Figure 9 Use Case Diagram Feedback for User

Use Case ID	UC_06	
Use Case Name	Feedback for User	
Description	It will submit the feedback in the database	
Primary Actor	User	
Secondary Actor	Database	
Pre-Condition	Logged in as User Internet required	
Post-Condition	Feedback updated successfully in database	
Basic Flow	Actor Action	System Action
	Click on the feedback button Enter the request or any problem and click on the submit button	The system will submit the feedback in the database and displays a message "Feedback submitted successfully" on the screen
Alternate Flow	If required field is empty record will be not saved	

Table 22 Use case Feedback for User

3.1.7 Custom Request

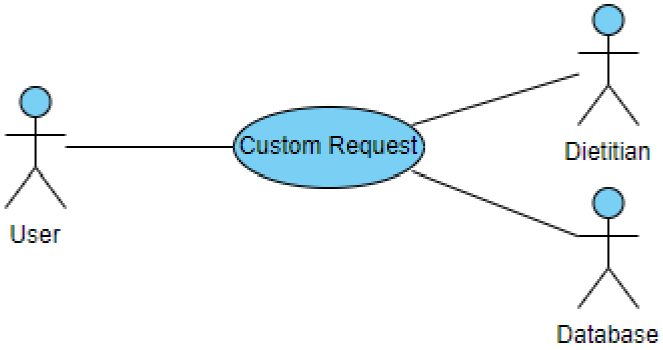


Figure 10 Use Case Diagram Custom Request for User

Use Case ID	UC_07	
Use Case Name	Custom Request for User	
Description	It will submit the request to add more diet plan to dietitian in the database	
Primary Actor	User	
Secondary Actor	Database	
Pre-Condition	Logged in as User Internet required	
Post-Condition	Request submitted successfully in database	
Basic Flow	Actor Action	System Action
	User click on the custom request button to request the dietician for some more food in the diet plan. System saved the request in the database.	The system will submit the Custom Request in the database and displays a message "Request submitted successfully" on the screen
Alternate Flow	If required field is empty record will be not saved	

Table 23 Use case Custom Request for User

3.2 Admin

3.2.1 Login Account

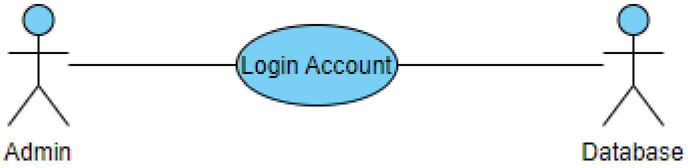


Figure 11 Use Case Diagram Login Account for Admin

Use Case ID	UC_08	
Use Case Name	Login Account for Admin	
Description	It will allow the admin to login	
Primary Actor	Admin	
Secondary Actor	Database	
Pre-Condition	The admin has an Email and password	
Post-Condition	Logged in into system as an admin	
Basic Flow	Actor Action	System Action
	Click on the login as admin Enter Email and password Click on the login button	The system would check the credentials and will log in the admin account
Alternate Flow	Email or password is incorrect, again on the login screen	

Table 24 Use case Login Account for Admin

3.2.2 Add Dietitian

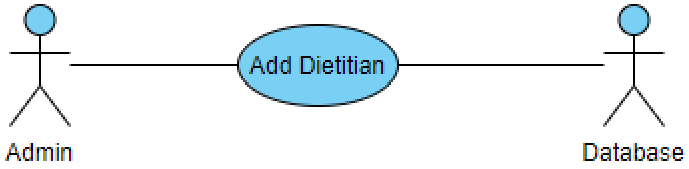


Figure 12 Use Case Diagram Add Dietitian for Admin

Use Case ID	UC_09	
Use Case Name	Add Dietitian for Admin	
Description	It will add the registration detail to the database	
Primary Actor	Admin	
Secondary Actor	Database	
Pre-Condition	Logged in as Internet required	
Post-Condition	Record submitted successfully in database	
Basic Flow	Actor Action	System Action
	Click on Add Dietitian button Enter Name, Age, Gender, Email, Password	Only admin can add the dietitian. The system will store the details in the database and displays a message "Record submitted successfully" on the screen
Alternate Flow	Some required fields are empty, the record not saved	

Table 25 Use case Add Dietitian for Admin

3.2.3 Remove Dietitian

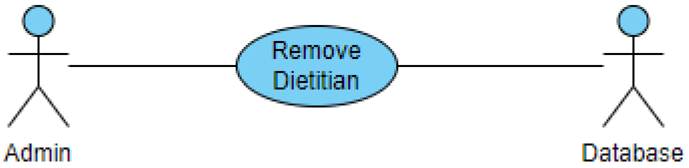


Figure 13 Use Case Diagram Remove Dietitian for Admin

Use Case ID	UC_10	
Use Case Name	Remove Dietitian for Admin	
Description	It will remove the dietitian in the database	
Primary Actor	Admin	
Secondary Actor	Database	
Pre-Condition	Logged in as admin Internet required	
Post-Condition	Record deleted successfully in database	
Basic Flow	Actor Action	System Action
	Enter the id of the dietitian and click on the delete record button.	The system will delete the details in the database and displays a message "Record deleted successfully" on the screen
Alternate Flow	Record not found against that id, again on the delete record screen	

Table 26 Use case Remove Dietitian for Admin

3.2.4 Remove User

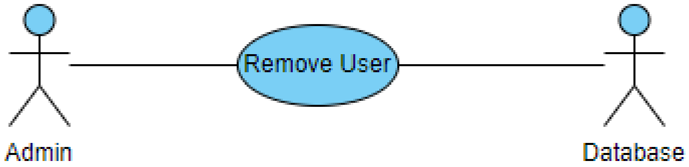


Figure 14 Use Case Diagram Remove User for Admin

Use Case ID	UC_11	
Use Case Name	Remove User for Admin	
Description	It will remove the user in the database	
Primary Actor	Admin	
Secondary Actor	Database	
Pre-Condition	Logged in as admin Internet required	
Post-Condition	Record deleted successfully in database	
Basic Flow	Actor Action	System Action
	Enter the id of the user and click on the delete record button.	The system will delete the details in the database and displays a message "Record deleted successfully" on the screen
Alternate Flow	Record not found against that id, again on the delete record screen	

Table 27 Use case Remove User for Admin

3.3 Dietitian

3.3.1 Login Account

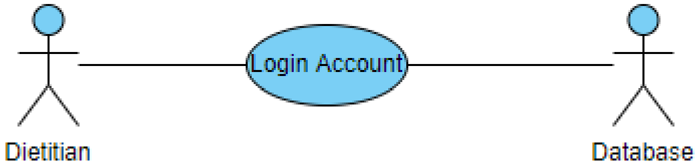


Figure 15 Use Case Diagram Login Account for Dietitian

Use Case ID	UC_01	
Use Case Name	Login Account for Dietitian	
Description	It will allow the dietitian to login	
Primary Actor	Dietitian	
Secondary Actor	Database	
Pre-Condition	The dietitian has an Email and password	
Post-Condition	Logged in into system as a dietitian	
Basic Flow	Actor Action	System Action
	Click on the login as dietitian Enter Email and password Click on the login button	The system would check the credentials and will log in the dietitian account
Alternate Flow	Email or password is incorrect, again on the login screen	

Table 28 Use case Login Account for Dietitian

3.3.2 Add Food Item

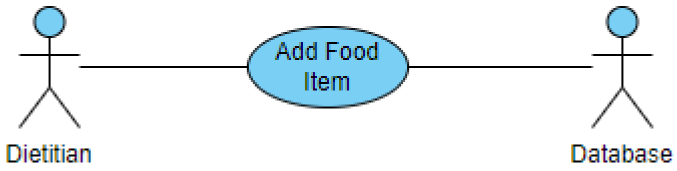


Figure 16 Use Case Diagram Add Food Item for Dietitian

Use Case ID	UC_03	
Use Case Name	Add Food Item for Dietitian	
Description	It will add the registration detail to the database	
Primary Actor	Dietitian	
Secondary Actor	Database	
Pre-Condition	Logged in as dietitian Internet required	
Post-Condition	Record submitted successfully in database	
Basic Flow	Actor Action	System Action
	Click on the add food item sign (+) then enter the Food Name, carbohydrates, Fats, Proteins Consumed by (Underweight, Healthy, Over Weight) Consumed by (Breakfast, Lunch, Dinner)	The system will store the details in the database and displays a message "Record submitted successfully" on the screen
Alternate Flow	Some required fields are empty, the record will be saved	

Table 29 Use case Add Food Item for Dietitian

3.3.3 Update Food Item

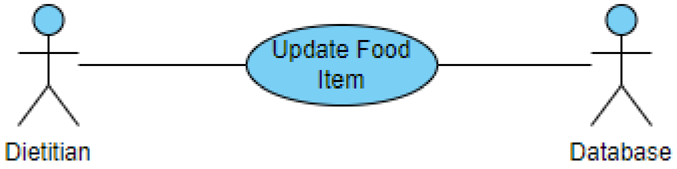


Figure 17 Use Case Diagram Update Food Item for Dietitian

Use Case ID	UC_04	
Use Case Name	Update Food Item for Dietitian	
Description	It will update the food item in the database	
Primary Actor	Dietitian	
Secondary Actor	Database	
Pre-Condition	Logged in as Dietitian Internet required	
Post-Condition	Record updated successfully in database	
Basic Flow	Actor Action	System Action
	Click on food item in the food list. Change required Food Name, carbohydrates, Fats, Proteins Consumed by (Underweight, Healthy, Over Weight) Optional Consumed by (Breakfast, Lunch, Dinner) Optional	The system will update the details in the database and displays a message "Record updated successfully" on the screen
Alternate Flow	Some required fields are empty	

Table 30 Use case Update Food Item for Dietitian

3.3.4 Update Diet Chart

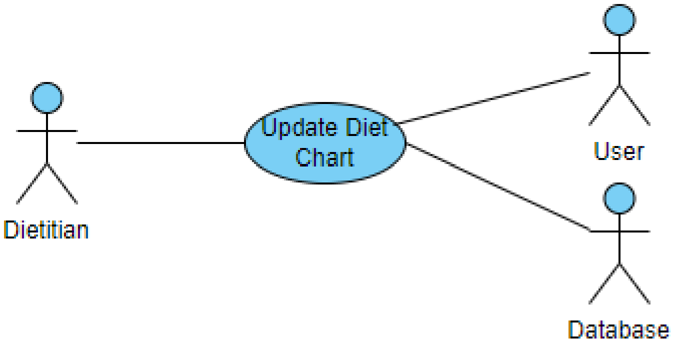


Figure 18 Use Case Diagram Update Diet Chart for Dietitian

Use Case ID	UC_04	
Use Case Name	Update Diet Chart for Dietitian	
Description	It will update the diet chart of user in the database	
Primary Actor	Dietitian	
Secondary Actor	Database	
Pre-Condition	Logged in as dietitian Internet required	
Post-Condition	Record updated successfully in database	
Basic Flow	Actor Action	System Action
	User can request the dietitian to update diet chart according to his choice then dietitian update the diet chart according to user custom request.	The system will update the user diet chart in the database and displays a message "Record updated successfully" on the screen
Alternate Flow	User must be login and request dietitian using custom request	

Table 31 Use case Update Diet Chart for Dietitian

3.3.5 View Feedback

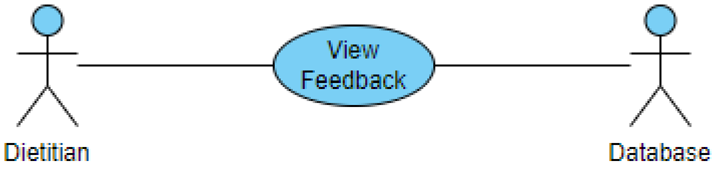


Figure 19 Use Case Diagram View Feedback for Dietitian

Use Case ID	UC_06	
Use Case Name	View Feedback for Dietitian	
Description	Dietitian can view user feedback	
Primary Actor	Dietitian	
Secondary Actor		
Pre-Condition	Logged in as Dietitian into the system Internet required	
Post-Condition	Dietitian has viewed his/her feedback.	
Basic Flow	Actor Action	System Action
	Dietitian selects view user details option	System displays user feedback on the screen
Alternate Flow	NA	

Table 32 Use case View Feedback for Dietitian

Chapter 4: Design

4.1 Architecture Diagram

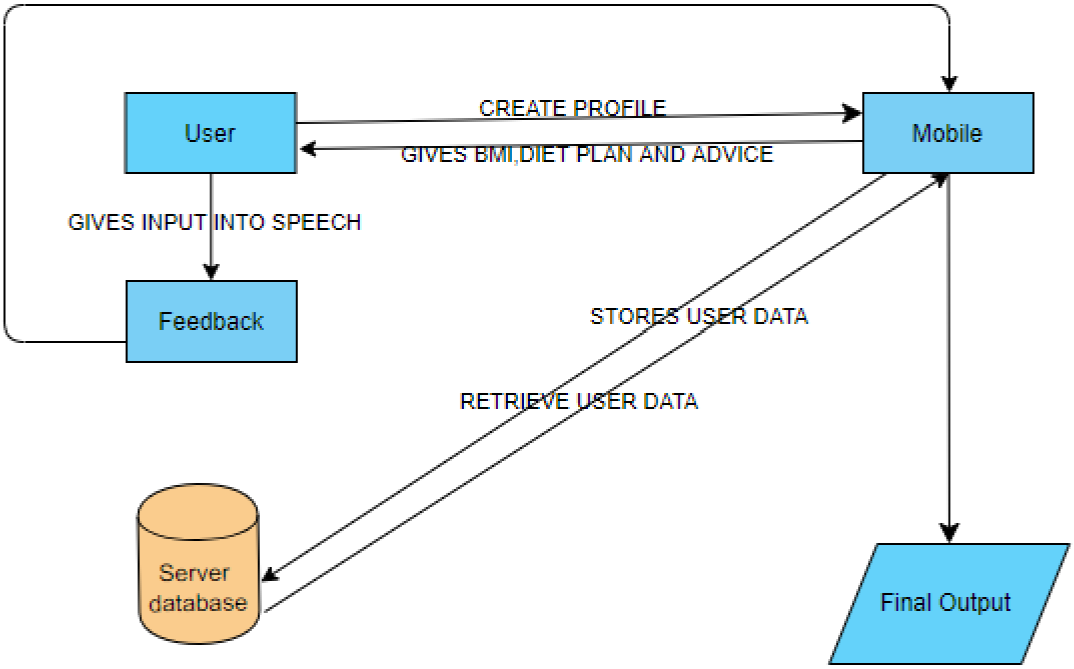


Figure 20 Architecture Diagram

4.2 ERD with data dictionary

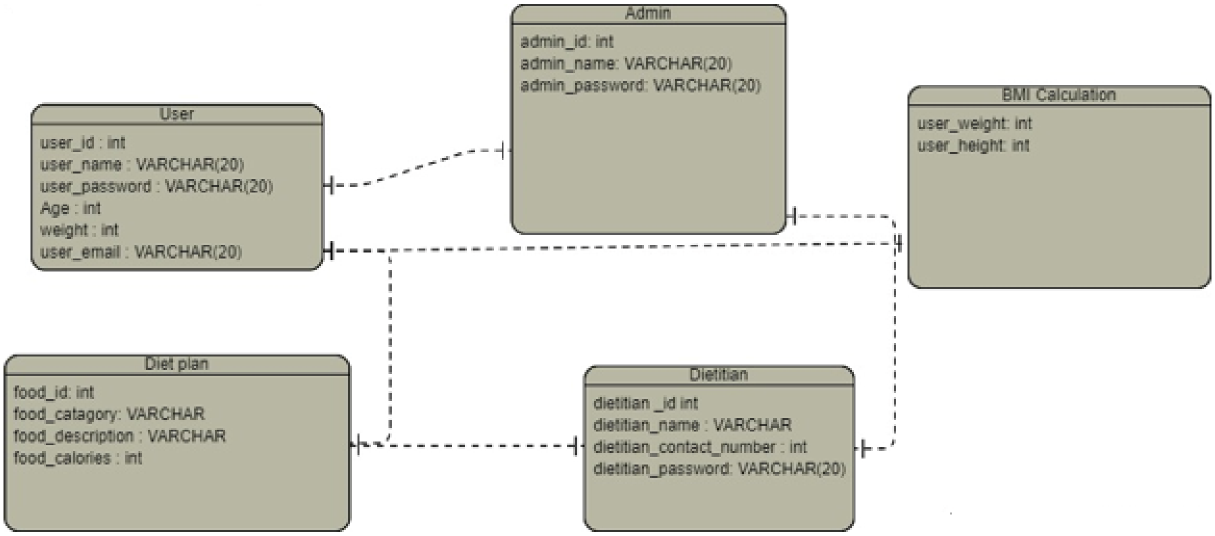


Figure 21 ERD

4.3 Data Flow diagram

Data flow diagram includes two levels

4.3.1 The level 0

The flow of information inside the system is defined in this level

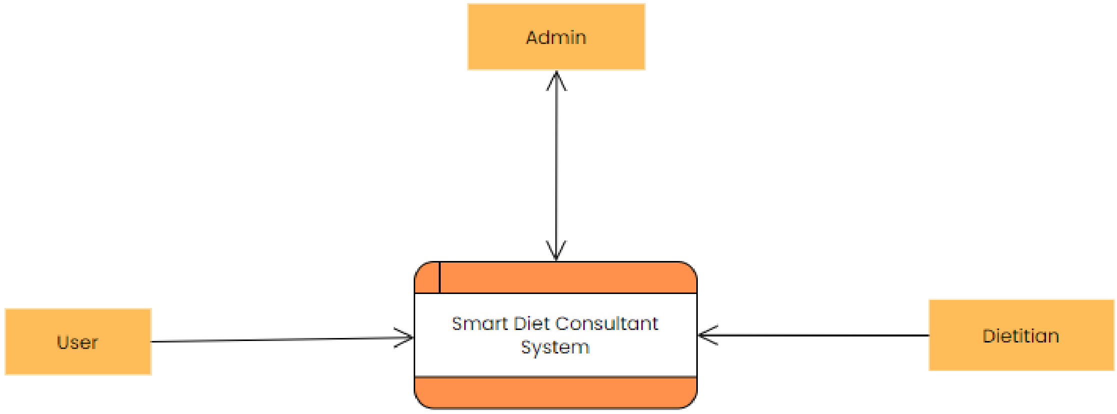


Figure 22 Level 0 DFD

4.3.2 The level 1

The flow of information outside the system is defined in this level

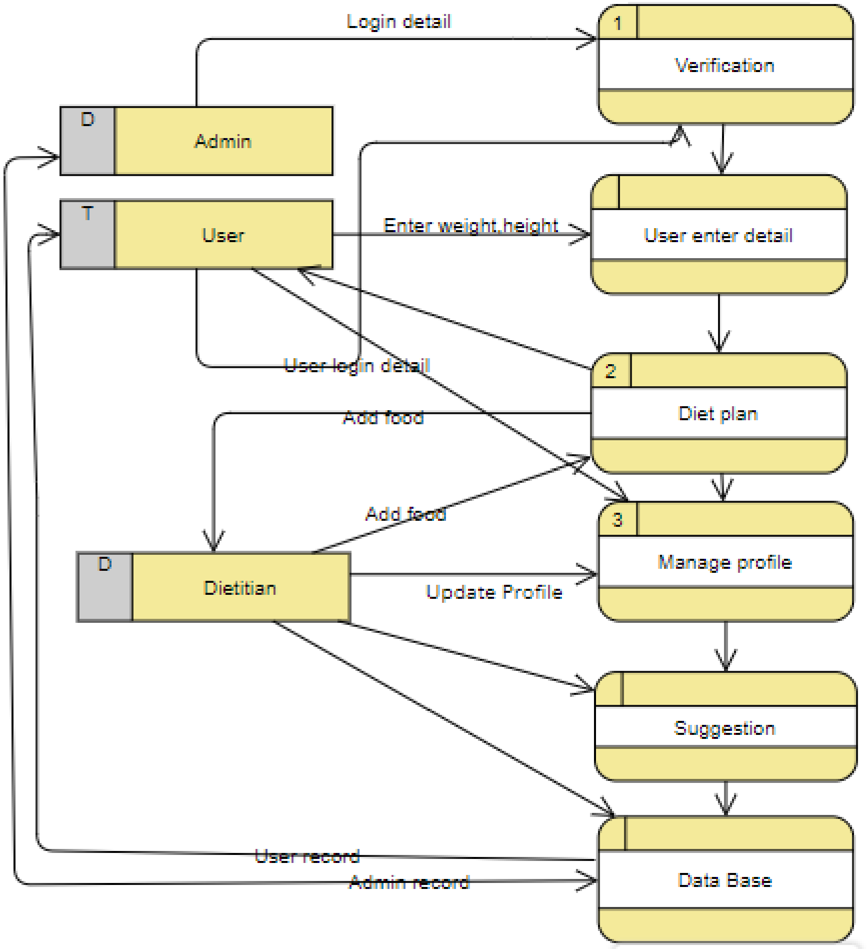


Figure 23 Level 1 DFD

4.4 Class Diagram

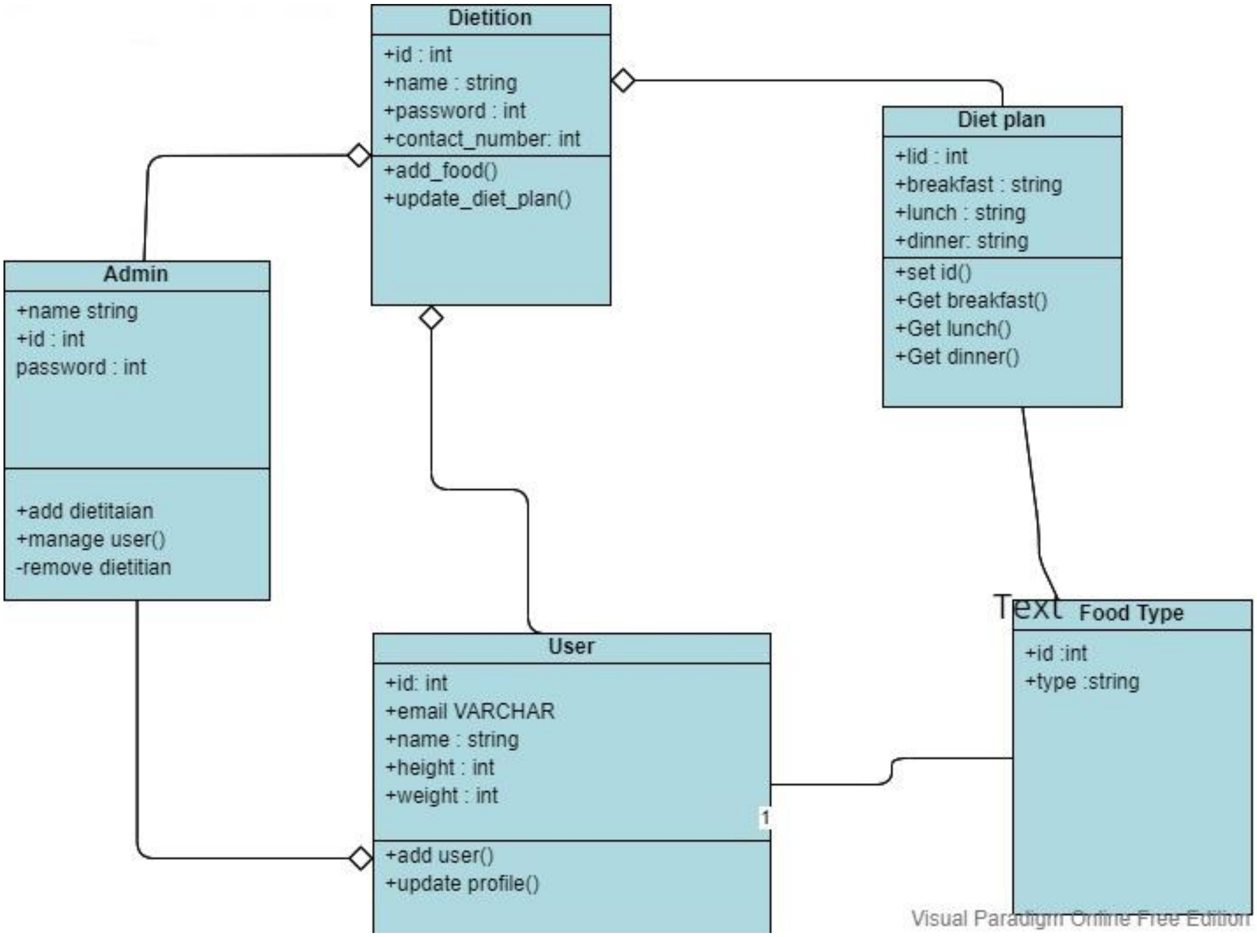


Figure 24 Class Diagram

4.5 Activity Diagram

This Section Includes all the activity diagrams

4.5.1 User

4.5.1.1 Create Account

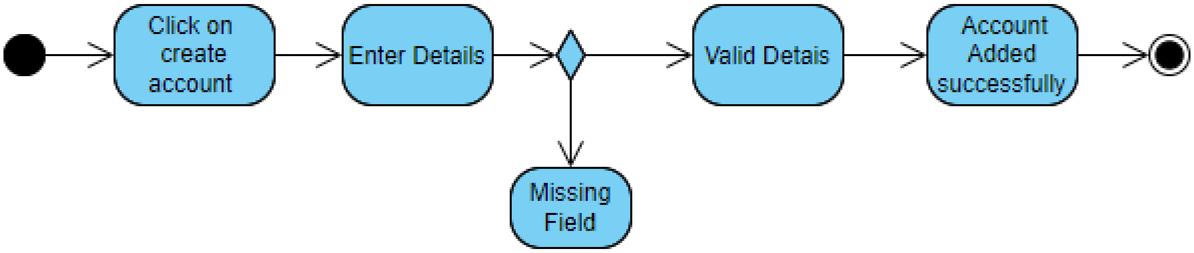


Figure 25 Activity Diagram Create Account for User

4.5.1.2 Login Account

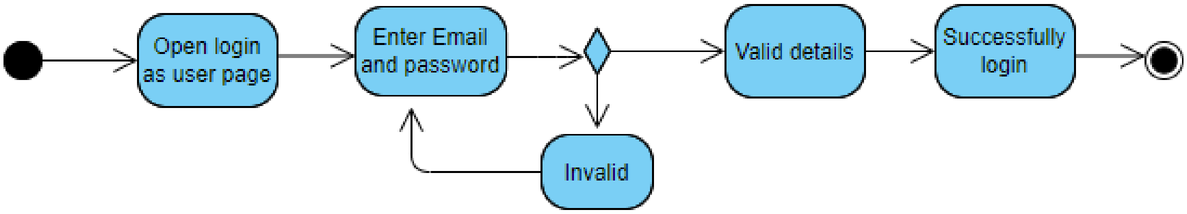


Figure 26 Activity Diagram Login Account for User

4.5.1.3 Change Password

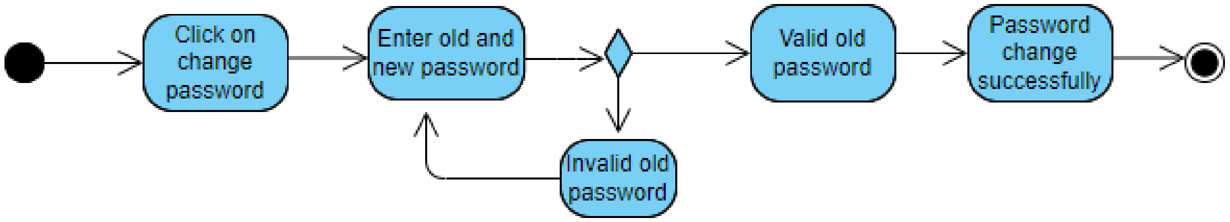


Figure 27 Activity Diagram Change Password for User

4.5.1.4 Create Profile

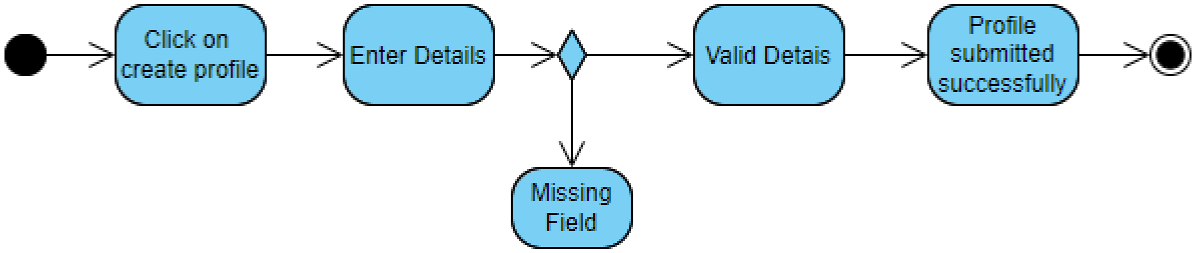


Figure 28 Activity Diagram Create Profile for User

4.5.1.5 Update Profile

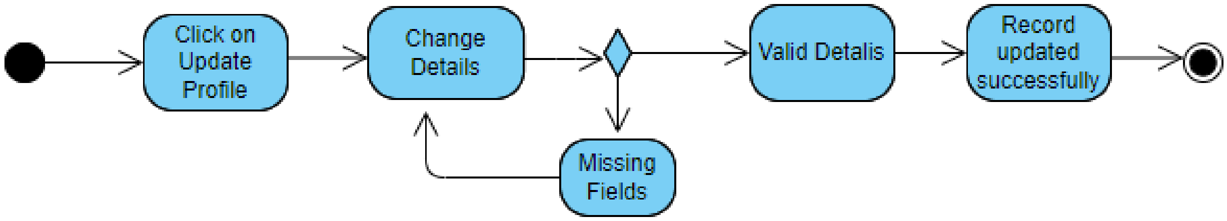


Figure 29 Activity Diagram Update Profile for User

4.5.1.6 Feedback

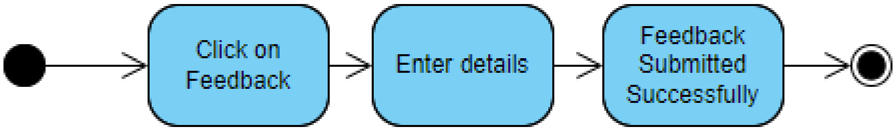


Figure 30 Activity Diagram Feedback for User

4.5.1.7 Custom Request

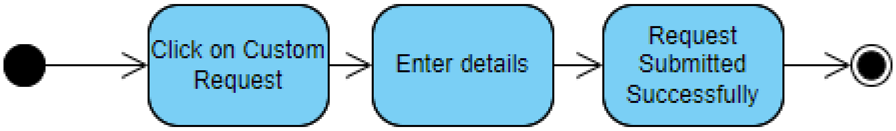


Figure 31 Activity Diagram Custom Request for User

4.5.2 Admin

4.5.2.1 Login Account

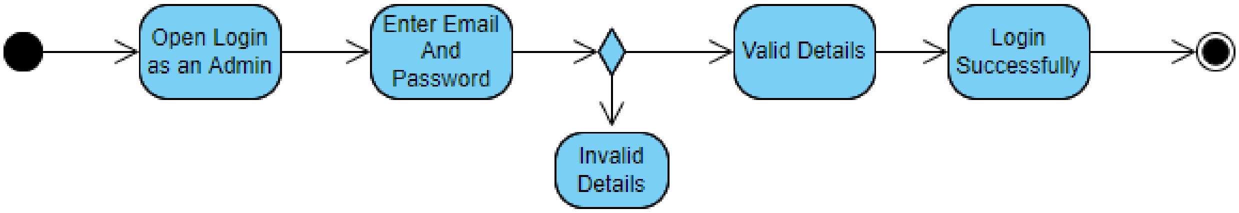


Figure 32 Activity Diagram Login Account for Admin

4.5.2.2 Add Dietitian

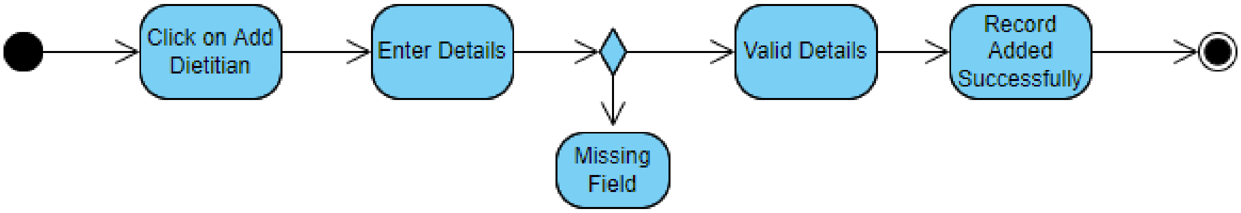


Figure 33 Activity Diagram Add Dietitian for Admin

4.5.2.3 Remove Dietitian

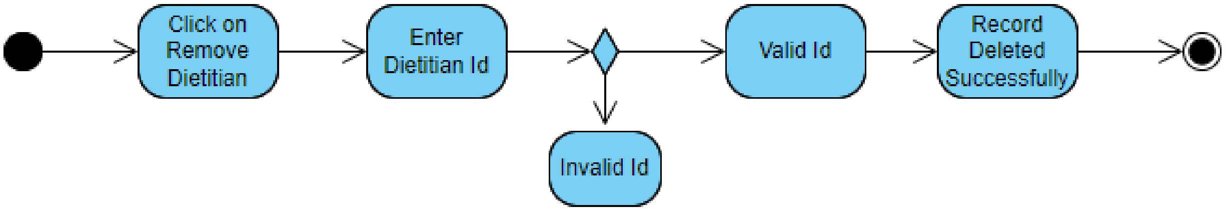


Figure 34 Activity Diagram Remove Dietitian for Admin

4.5.2.4 Remove User

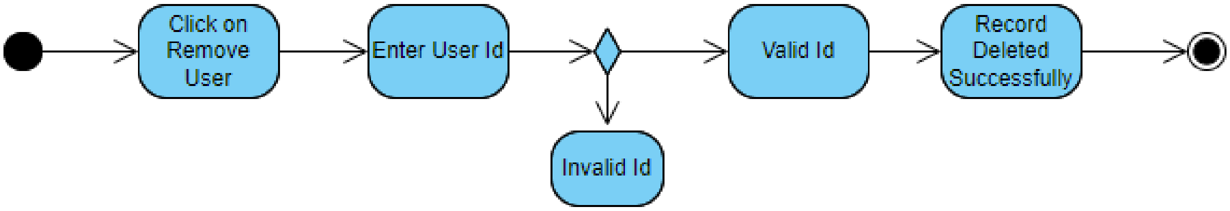


Figure 35 Activity Diagram Remove User for Admin

4.5.3 Dietitian

4.5.3.1 Login Account

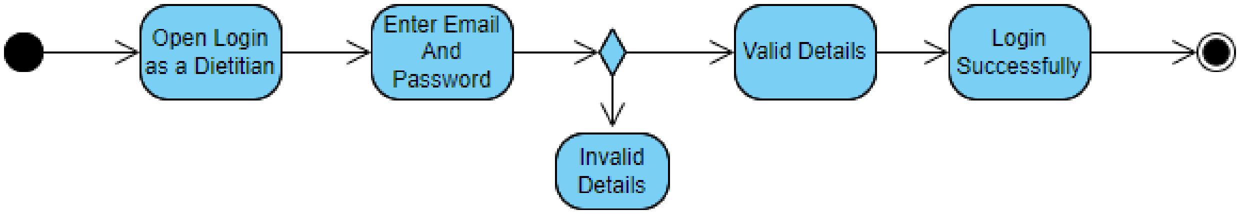


Figure 36 Activity Diagram Login Account for Dietitian

4.5.3.2 Add Food Item

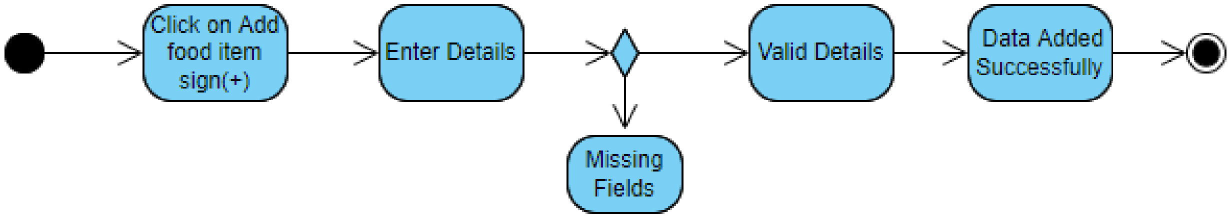


Figure 37 Activity Diagram Add Food Item for Dietitian

4.5.3.3 Update Food Item

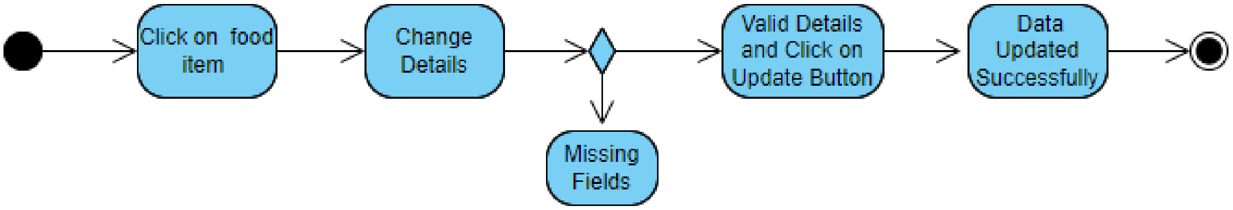

Click on food item
Change Details
Valid Details and Click on Update Button
Data Updated Successfully
Missing Fields

Figure 38 Activity Diagram Update Food Item for Dietitian

4.5.3.4 Update Diet Chart

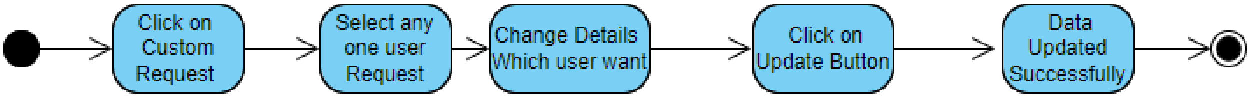

Click on Custom Request
Select any one user Request
Change Details Which user want
Click on Update Button
Data Updated Successfully

Figure 39 Activity Diagram Update Diet Chart for Dietitian

4.5.3.5 View Feedback

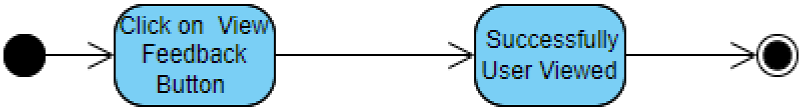


Figure 40 Activity Diagram View Feedback for Dietitian

4.6 Sequence Diagram

4.6.1 User

4.6.1.1 Create Account

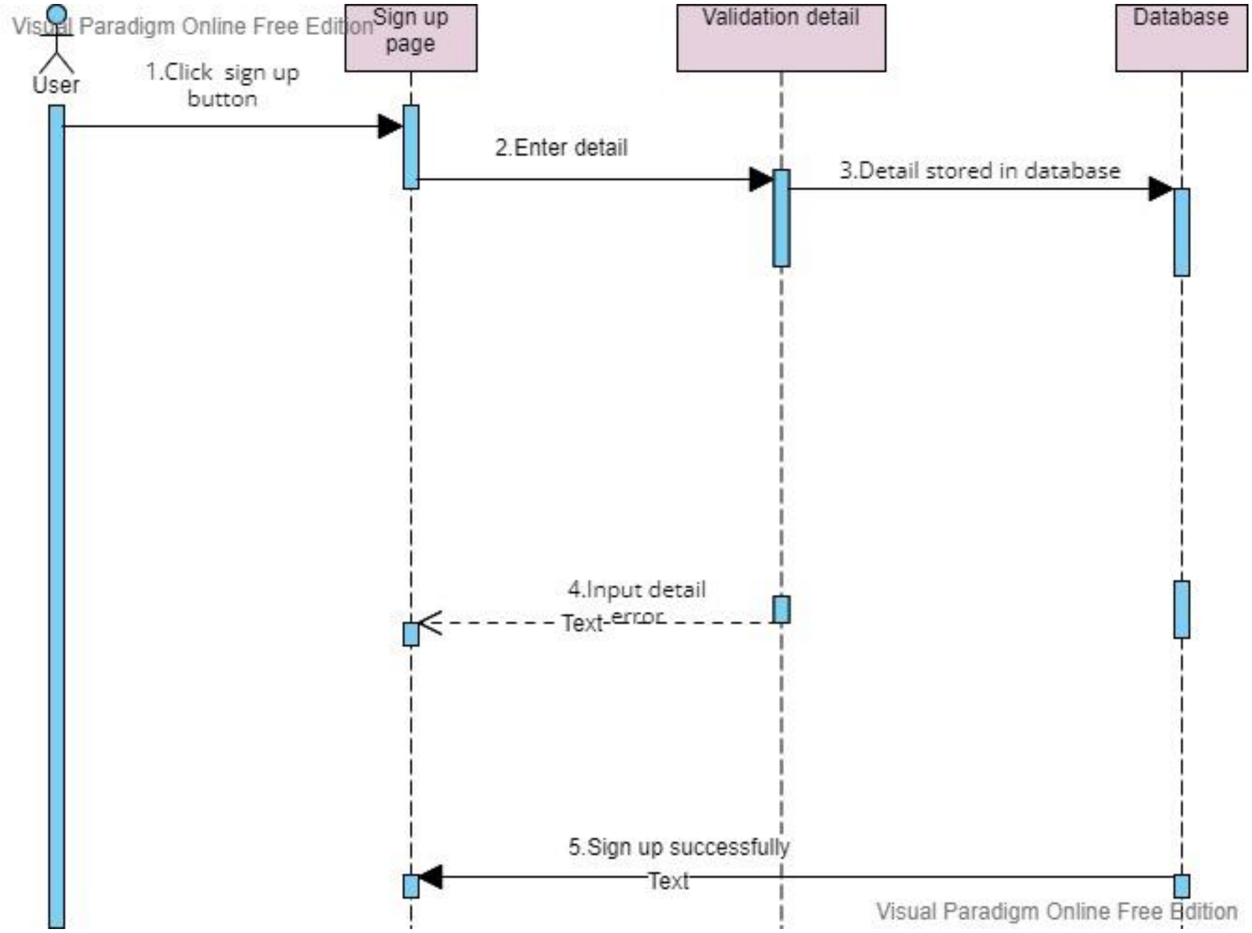


Figure 41 Sequence Diagram Create Account for User

4.6.1.2 Login Account

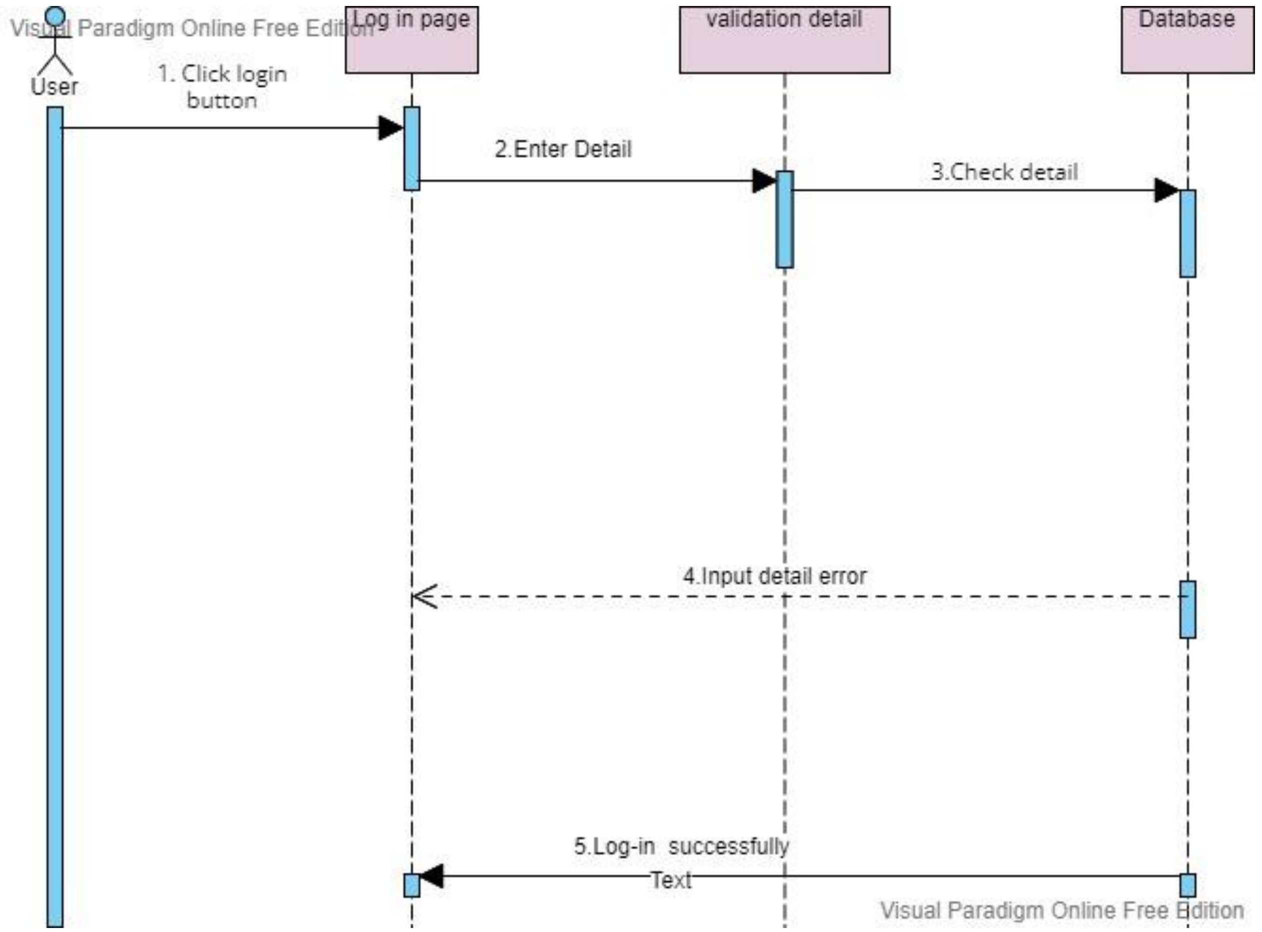


Figure 42 Sequence Diagram Login Account for User

4.6.1.3 Change Password

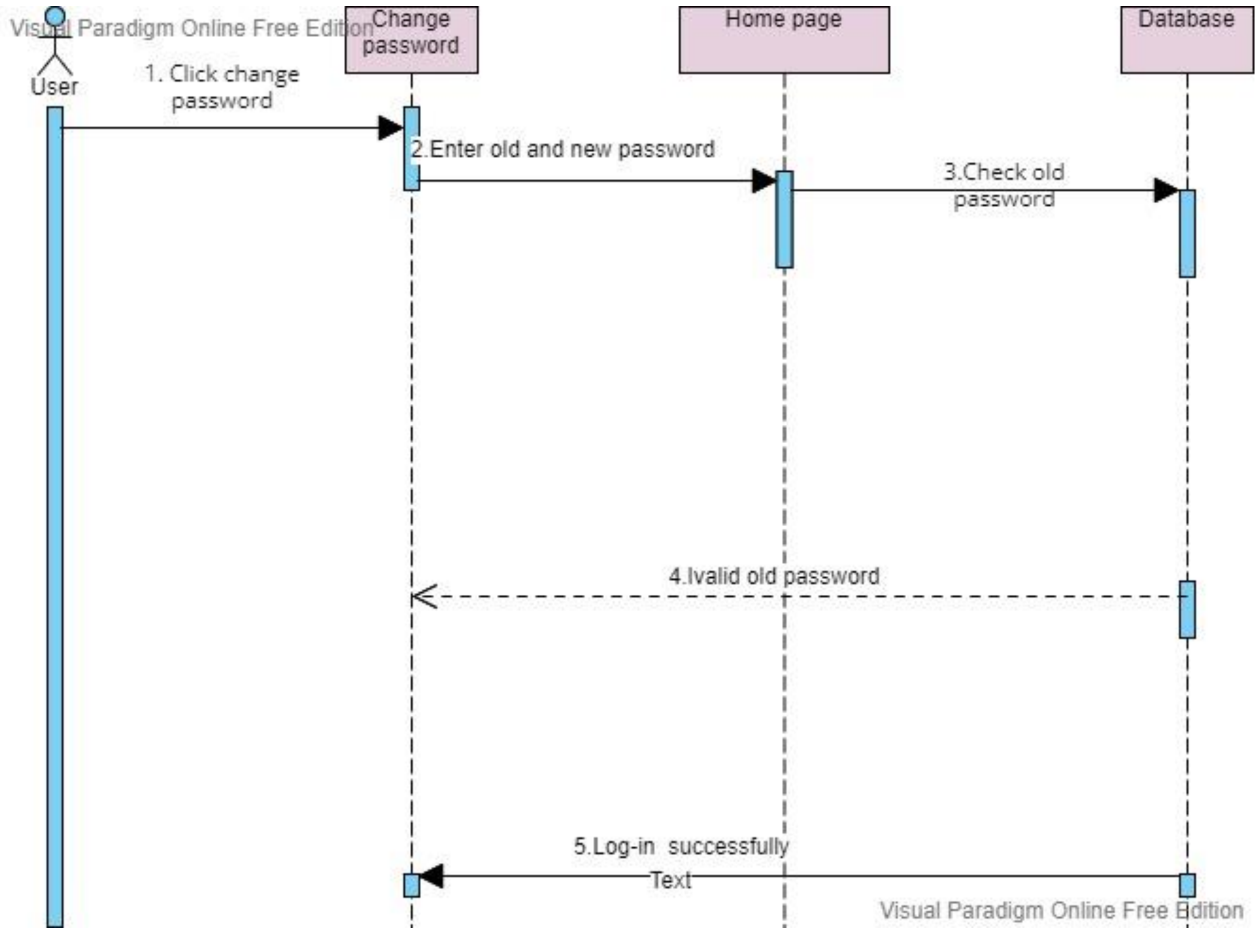


Figure 43 Sequence Diagram Change Password for User

4.6.1.4 Create Profile

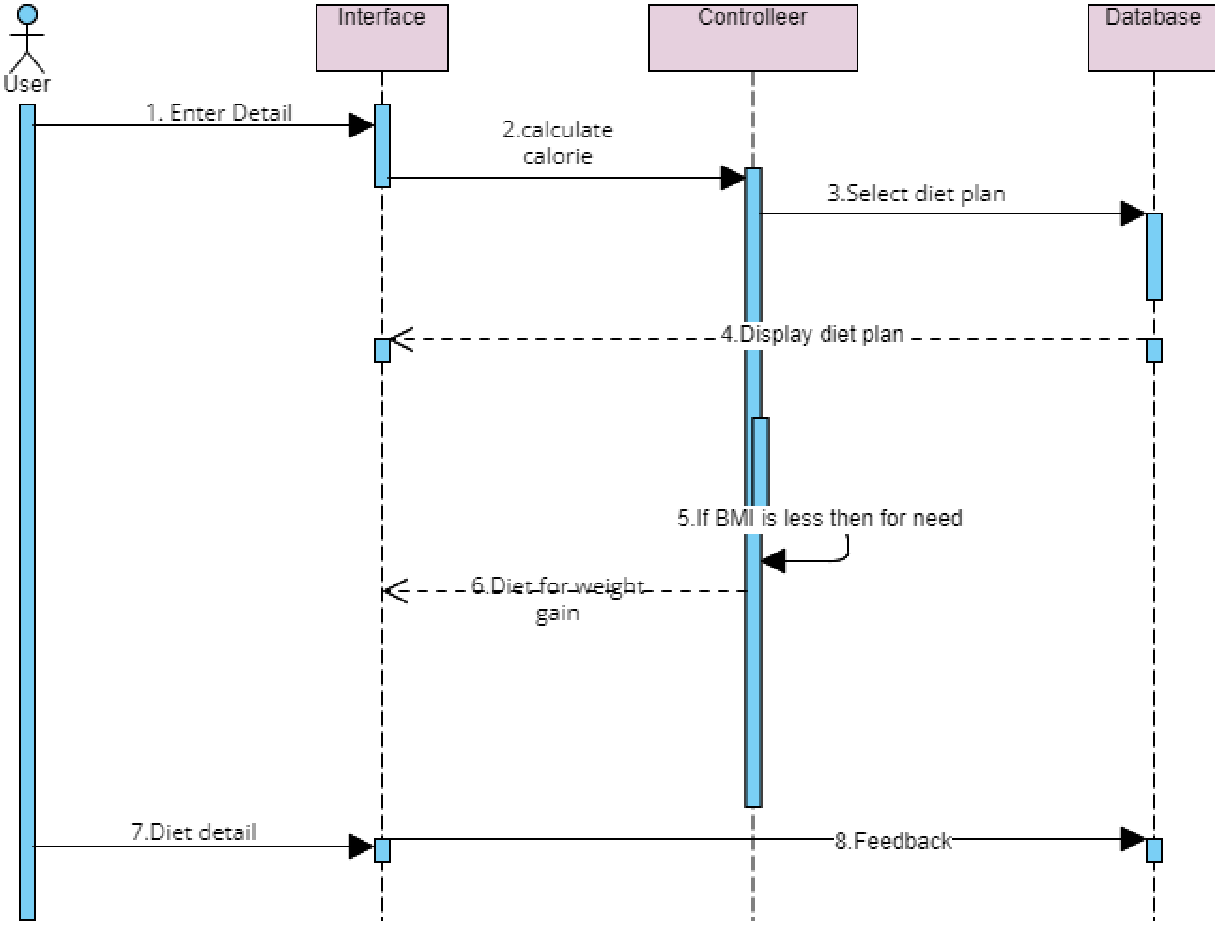


Figure 44 Sequence Diagram Create Profile for User

4.6.1.5 Update Profile

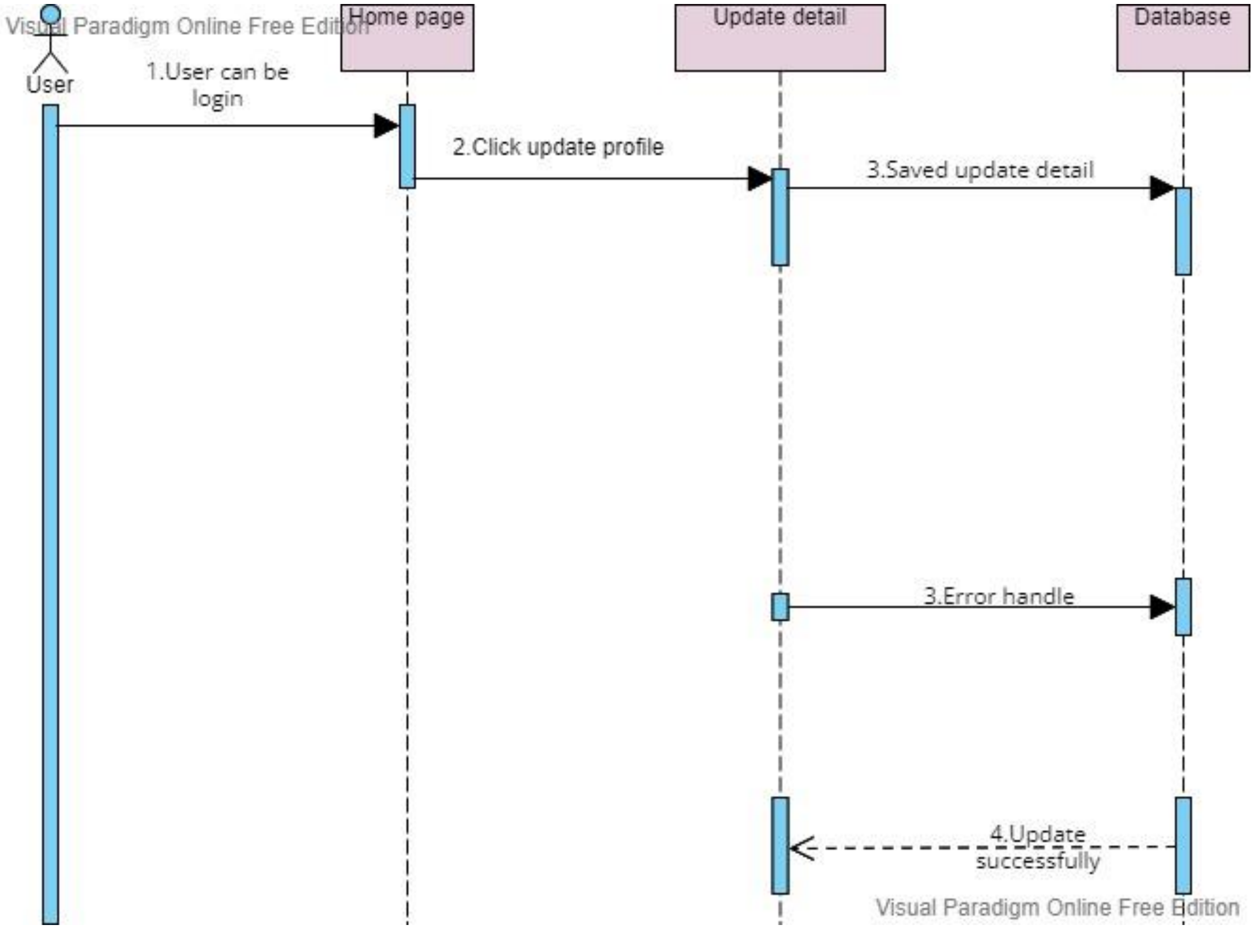


Figure 45 Sequence Diagram Update Profile for User

4.6.2 Admin

4.6.2.1 Login Account

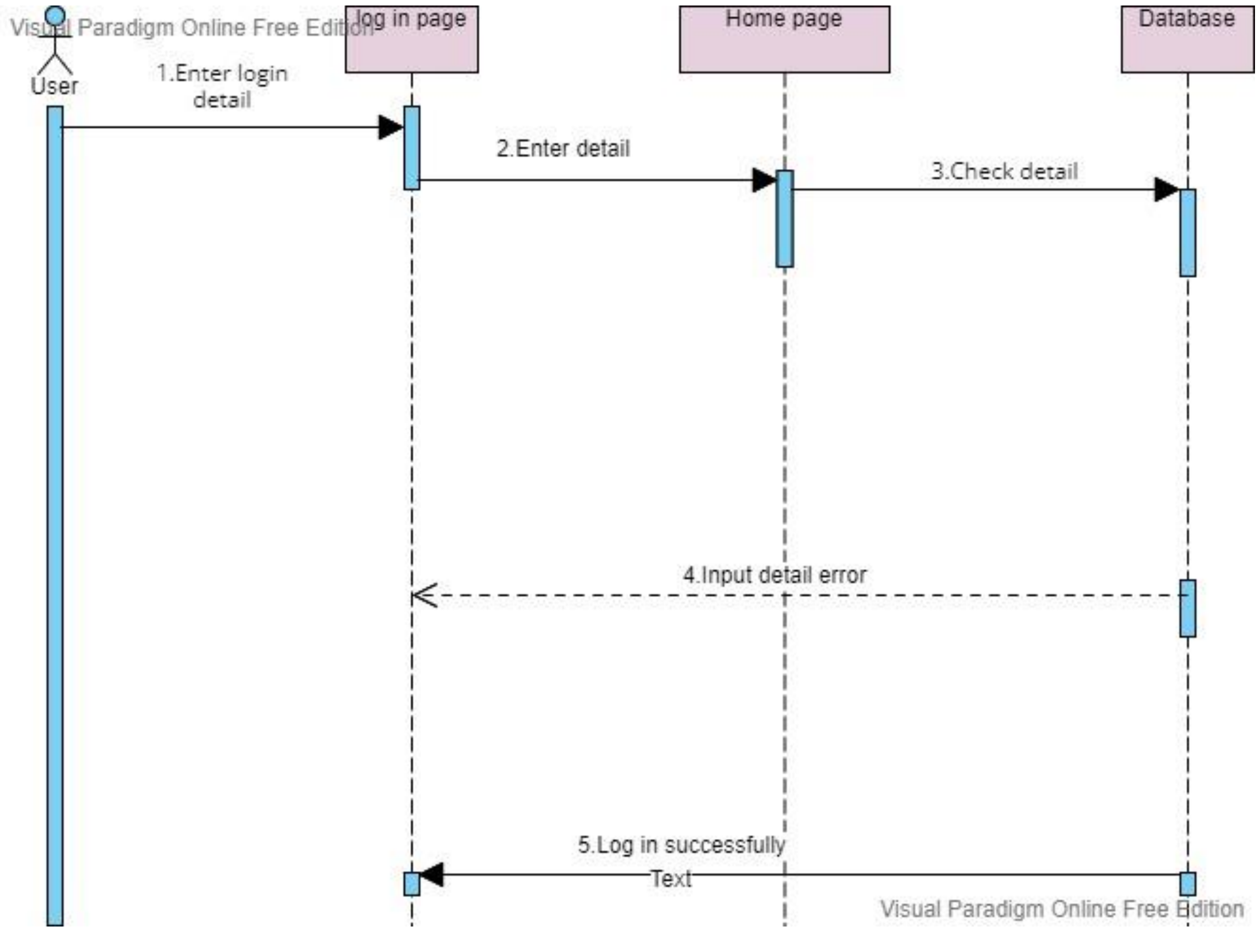


Figure 46 Sequence Diagram Login Account for Admin

4.6.2.2 Add Dietitian

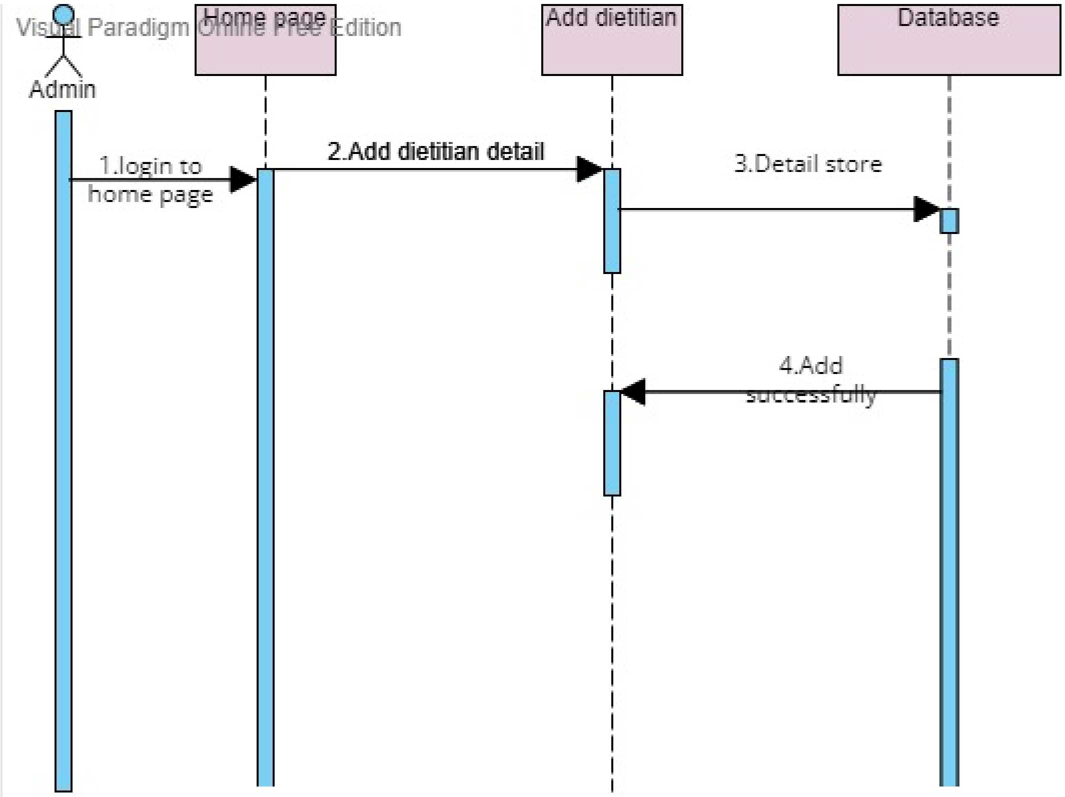


Figure 47 Sequence Diagram Add Dietitian for Admin

4.6.3 Dietitian

4.6.3.1 Login Account

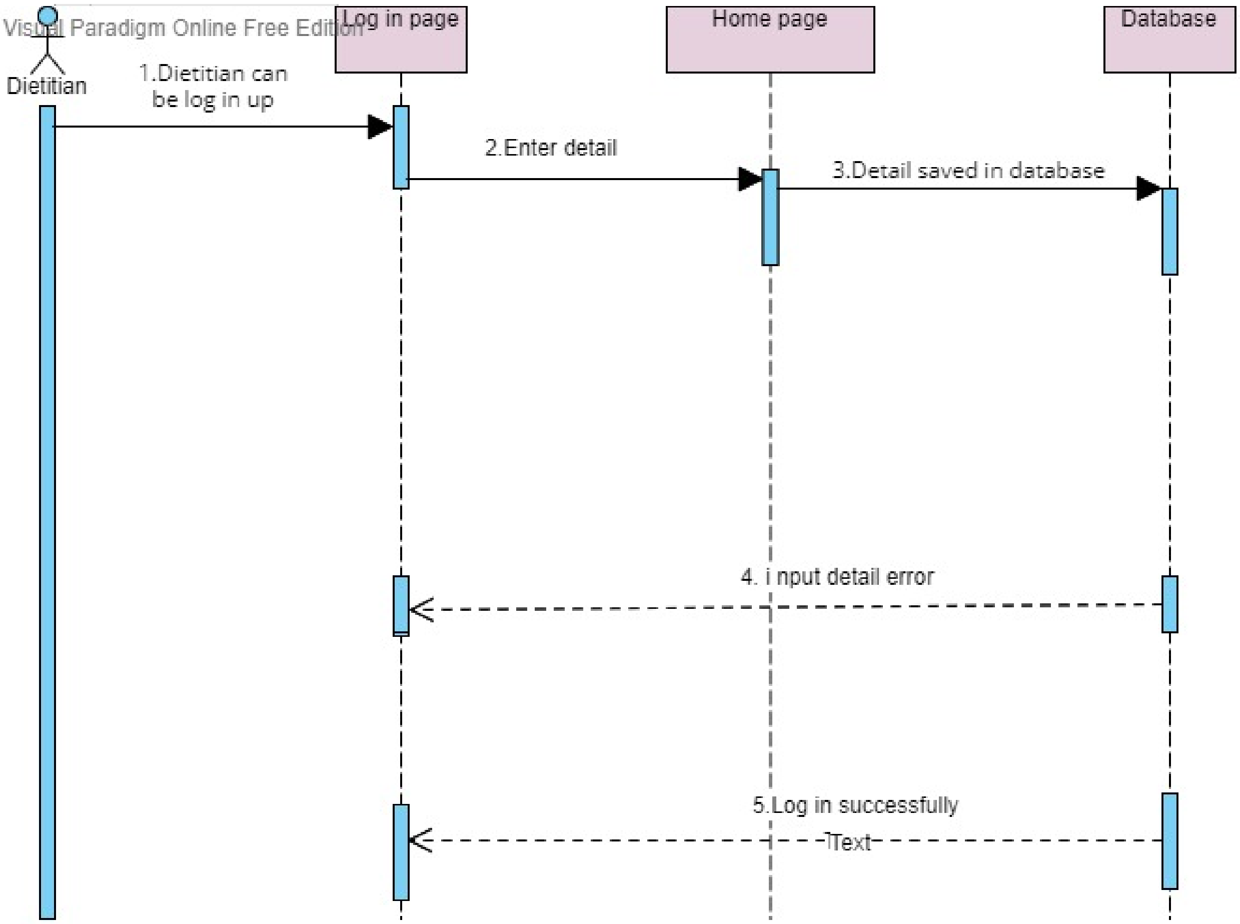


Figure 48 Sequence Diagram Login Account for Dietitian

4.6.3.2 Add Food Item

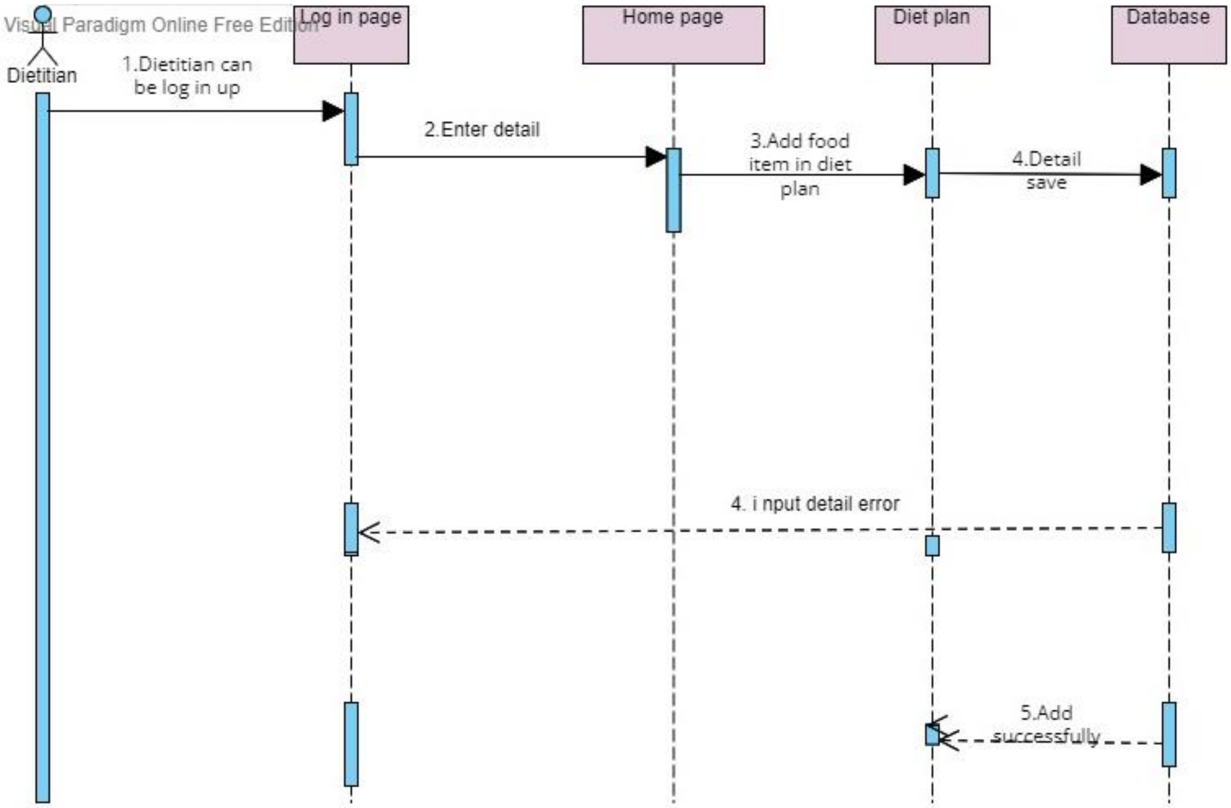


Figure 49 Sequence Diagram Add Food Item for Dietitian

4.6.3.3 Remove Food Item

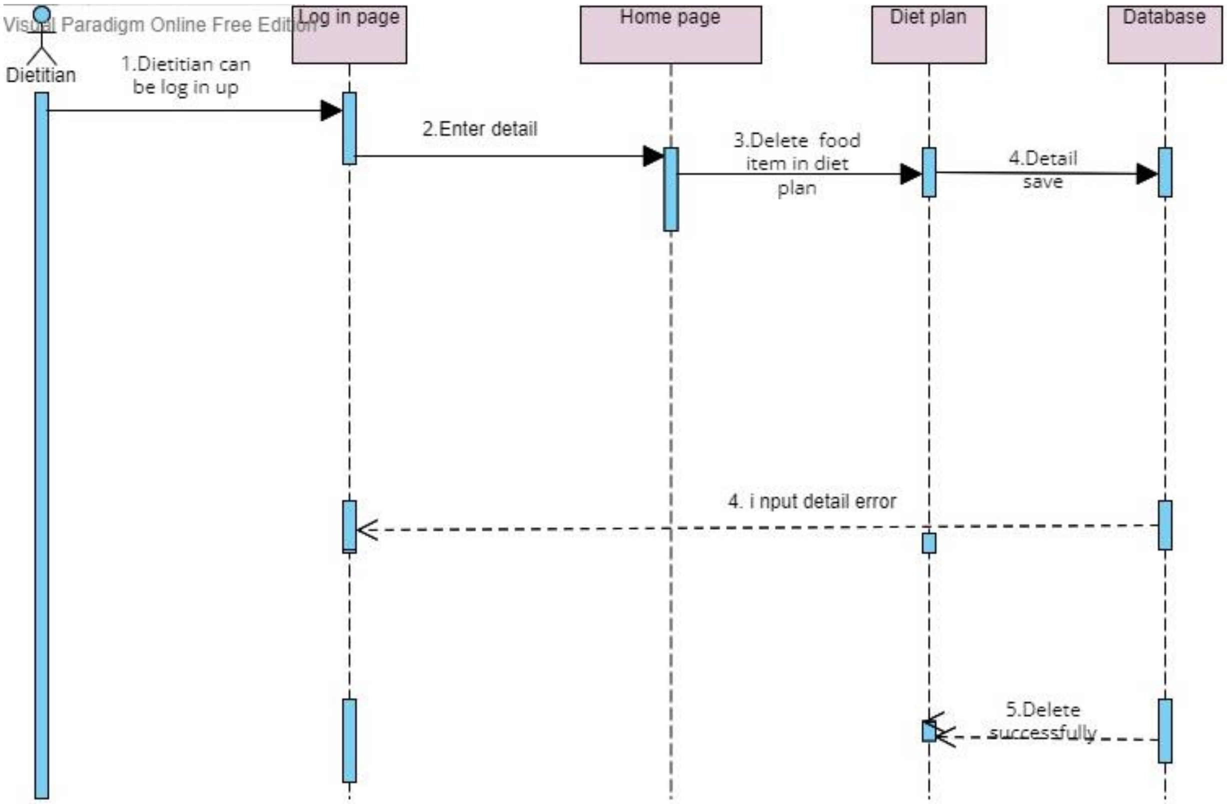


Figure 50 Sequence Diagram Remove Food Item for Dietitian

4.7 Collaboration Diagram

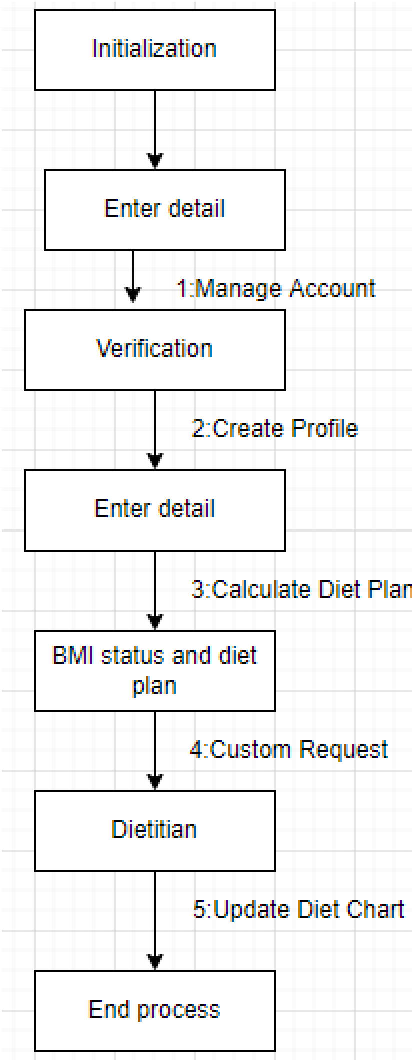


Figure 51 Collaboration Diagram

4.8 State Transition Diagram

State Transition diagram is used to describe the states of different objects in its life cycle. So, the emphasis is given on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately

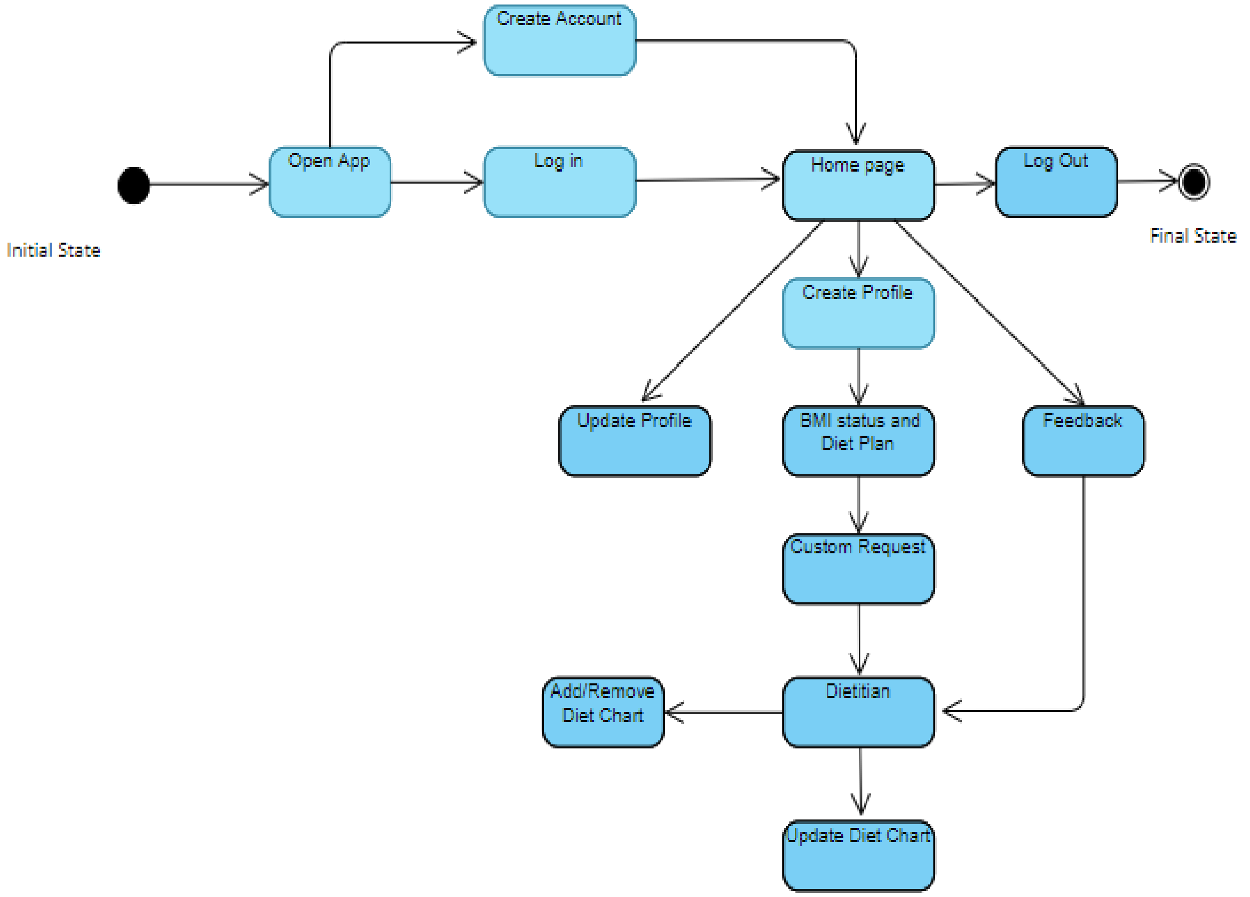


Figure 52 State Transition Diagram

4.9 Component Diagram

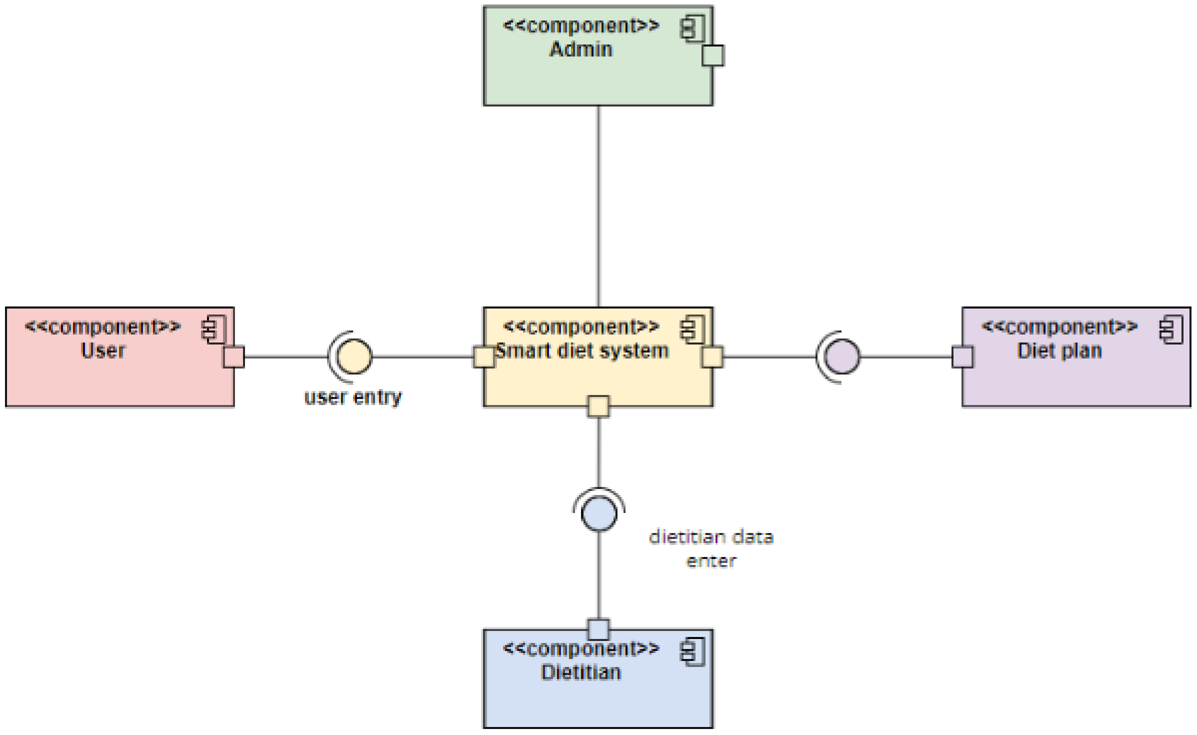


Figure 53 Component Diagram

4.10 Deployment Diagram

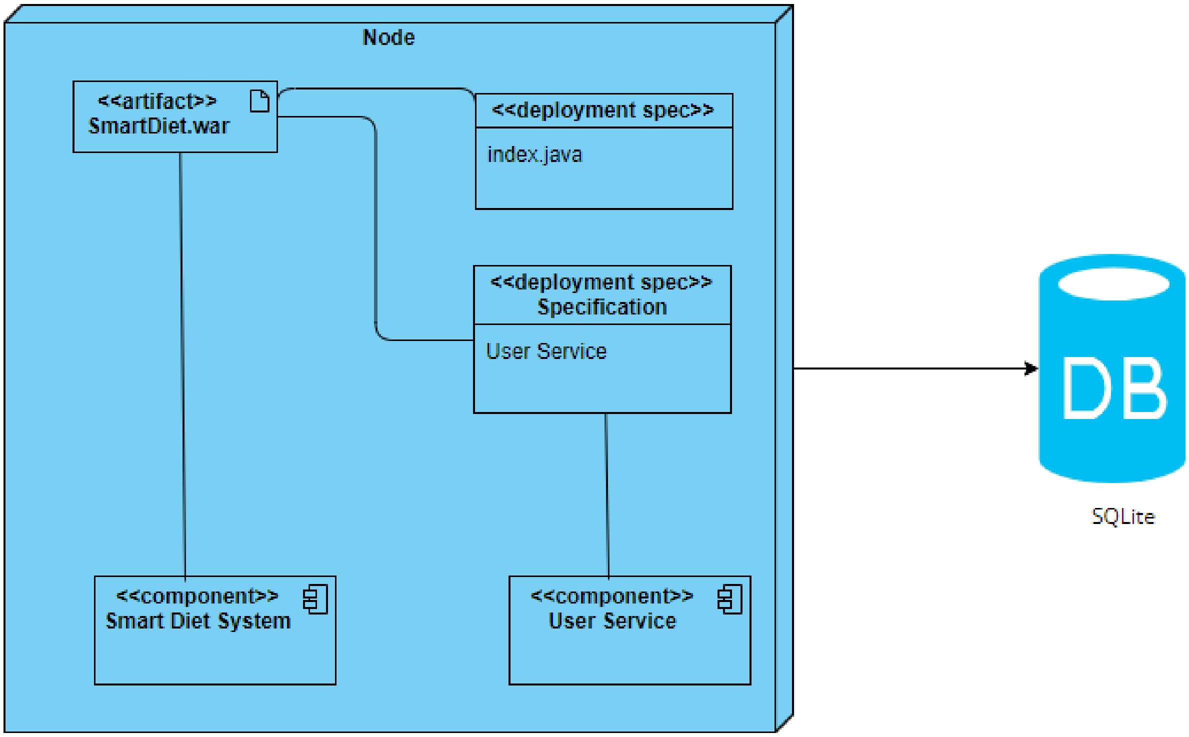


Figure 54 Deployment Diagram

Chapter 5: Testing

5.1 Test Case Specifications

This Testing phase includes all the Test Cases of the functional requirements of your project

5.1.1 Test Case for login account

Positive Test Case	
ID	TC_01
Priority	High
Description	To verify user authentication to system.
Reference	Functional Requirement reference
Users	User, Admin, Dietitian
Pre-requisites	A. System is online. B. User must have active login credentials provided by system administrator. C. User has internet access.
Steps	A. Install the app on your mobile for User and web link to system for admin and Dietitian B. Open Login page C. Enter login id D. Enter Password. E. Press Login.
Input	Login id and password
Expected result	Successfully enters the system and admin dashboard opens.
Status	Tested, passed.

Table 33 TC_01 Login

5.1.2 Test Case for Login Account

Negative Test Case	
ID	TC_02
Priority	High
Description	To verify user authentication to system.
Reference	Functional Requirement reference
Users	User, Admin, Dietitian
Pre-requisites	A. System is online. B. User must have active login credentials provided by system administrator. C. User has internet access.
Steps	A. Install the app on your mobile for User and web link to system for admin and Dietitian B. Open Login page C. Enter incorrect login id. D. Enter Incorrect Password. E. Press Login.
Input	Incorrect Login id or password or deactivated credentials.
Expected result	Does not allow access to system features and notifies the error.
Status	Tested, passed.

Table 34 TC_02 Login

5.1.3 Test Case for Login Account

Negative Test Case	
ID	TC_03
Priority	High
Description	Check the required fields by not filling any data.
Reference	Functional Requirement reference
Users	User, Admin, Dietitian
Pre-requisites	A. System is online. B. User has internet access C. User must have active login credentials provided by system administrator.
Steps	A. Install the app on your mobile for User and web link to system for admin and Dietitian B. Open Login Page C. Blank Email field D. Blank Password field E. Press Login.
Input	NA
Expected result	User should not log in and should show proper error message
Status	Tested, passed.

Table 35 TC_03 Login

5.1.4 Test Case for Login Account

Negative Test Case	
ID	TC_04
Priority	High
Description	Check When passing a correct username and invalid password
Reference	Functional Requirement reference
Users	User, Admin, Dietitian
Pre-requisites	A. System is online. B. User must have active login credentials provided by system administrator. C. User has internet access.
Steps	A. Install the app on your mobile for User and web link to system for admin and Dietitian. B. Enter correct login id. C. Enter Invalid Password. D. Press Login.
Input	Enter Correct Login Id and Invalid Password
Expected result	Does not allow access to system features and notifies the error.
Status	Tested, passed.

Table 36 TC_04 Login

5.1.5 Test Case for Login Account

Negative Test Case	
ID	TC_05
Priority	High
Description	Verify when passing incorrect login id and correct password
Reference	Functional Requirement reference
Users	User, Admin, Dietitian
Pre-requisites	A. System is online. B. User must have active login credentials provided by system administrator. C. User has internet access.
Steps	A. Install the app on your mobile for User and web link to system for admin and Dietitian B. Enter incorrect login id. C. Enter correct Password. D. Press Login.
Input	Enter Incorrect Login Id and correct Password
Expected result	Does not allow access to system features and notifies the error.
Status	Tested, passed.

Table 37 TC_05 Login

5.1.6 Test Case for Login Account

Negative Test Case	
ID	TC_06
Priority	High
Description	Verify when passing login id and blank password field
Reference	Functional Requirement reference
Users	User, Admin, Dietitian
Pre-requisites	A. System is online. B. User must have active login credentials provided by system administrator. C. User has internet access.
Steps	A. Install the app on your mobile for User and web link to system for admin and Dietitian B. Open Login Page C. Enter login id. D. Blank Password field. E. Press Login.
Input	Enter Login Id and blank Password field
Expected result	Does not allow access to system features and notifies the error.
Status	Tested, passed.

Table 38 TC_06 Login

5.1.7 Test Case for Login Account

Negative Test Case	
ID	TC_07
Priority	High
Description	Verify when passing blank login id and correct password field
Reference	Functional Requirement reference
Users	User, Admin, Dietitian
Pre-requisites	A. System is online. B. User must have active login credentials provided by system administrator. C. User has internet access.
Steps	A. Install the app on your mobile for User and web link to system for admin and Dietitian B. Open Login page C. Blank login id field. D. Enter Password. E. Press Login.
Input	Blank Login Id and correct Password field
Expected result	Does not allow access to system features and notifies the error.
Status	Tested, passed.

Table 39 TC_07 Login

5.1.8 Test Case for Sign up

Positive Test Case	
ID	TC_08
Priority	High
Description	Check user should Register by filling all the required fields
Reference	Functional Requirement reference
Users	Users
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile. B. Open signup page C. Enter valid values in the required fields. D. Press Register
Input	NA
Expected result	Users should be registered successfully.
Status	Tested, passed.

Table 400 TC_08 Signup

5.1.9 Test Case for Sign up

Negative Test Case	
ID	TC_09
Priority	High
Description	Check the required fields by not filling any data
Reference	Functional Requirement reference
Users	Users
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Do not enter any value in the field. D. Press Register
Input	NA
Expected result	It should show error message on mandatory fields.
Status	Tested, passed.

Table 411 TC_09 Signup

5.1.10 Test Case for Sign up

Negative Test Case	
ID	TC_10
Priority	High
Description	Check the Email text field that has an Email address without @ symbol.
Reference	Functional Requirement reference
Users	Users.
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Enter Invalid Emails D. Press Register
Input	Email address without @ symbol
Expected result	It should show the validation message for valid email.
Status	Tested, passed.

Table 422 TC_10 Signup

5.1.11 Test Case for Sign up

Negative Test Case	
ID	TC_11
Priority	High
Description	Check all the valid emails.
Reference	Functional Requirement reference
Users	Users.
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Enter valid Emails D. Click on the Register Button.
Input	Valid Email
Expected result	It should not show any validation message.
Status	Tested, passed.

Table 433 TC_11 Signup

5.1.12 Test Case for Sign up

Positive Test Case	
ID	TC_12
Priority	High
Description	Check the password when passing valid data
Reference	Functional Requirement reference
Users	Users.
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Enter value in numeric which is in between 8-32 D. Click on the Register Button.
Input	Valid Password
Expected result	It should not show the validation message
Status	Tested, passed.

Table 444 TC_12 Signup

5.1.13 Test Case for Sign up

Negative Test Case	
ID	TC_13
Priority	High
Description	Check the password limit when enter value less than min
Reference	Functional Requirement reference
Users	Users.
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Enter value which is alphanumeric but less than 8. D. Click on the Register Button.
Input	Password
Expected result	It should show the validation message
Status	Tested, passed.

Table 455 TC_13 Signup

5.1.14 Test Case for Sign up

Negative Test Case	
ID	TC_14
Priority	High
Description	Verify if blank spaces are passed in required fields.
Reference	Functional Requirement reference
Users	Users.
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Passed blank spaces in required fields. D. Click on the Register Button.
Input	NA
Expected result	Those Blank spaces should trim and Validation error message for required fields should visible.
Status	Tested, passed.

Table 466 TC_14 Signup

5.1.15 Test Case for Sign up

Negative Test Case	
ID	TC_15
Priority	High
Description	Verify if we have filled all required fields except the name field
Reference	Functional Requirement reference
Users	Users.
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Passed all field with valid data in required fields except name field D. Click on the Register Button.
Input	NA
Expected result	It should show the validation message on name field.
Status	Tested, passed.

Table 477 TC_15 Signup

5.1.16 Test Case for Sign up

Negative Test Case	
ID	TC_16
Priority	High
Description	Verify if we have filled all required fields except the email field
Reference	Functional Requirement reference
Users	Users.
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Passed all field with valid data in required fields except email field D. Click on the Register Button.
Input	NA
Expected result	It should show the validation message on email field.
Status	Tested, passed.

Table 488 TC_16 Signup

5.1.17 Test Case for Sign up

Negative Test Case	
ID	TC_17
Priority	High
Description	Verify if we have filled all required fields except the Password field
Reference	Functional Requirement reference
Users	Users.
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Passed all field with valid data in required fields except Password field D. Click on the Register Button.
Input	NA
Expected result	It should show the validation message on password field.
Status	Tested, passed.

Table 499 TC_17 Signup

5.1.18 Test Case for Sign up

Positive Test Case	
ID	TC_18
Priority	High
Description	Check if user receives signup successfully email after signup
Reference	Functional Requirement reference
Users	Users.
Pre-requisites	A. System is online. B. User has internet access.
Steps	A. Install the app on your mobile B. Open signup page C. Enter value in numeric which is in between 8-32 D. Click on the Register Button.
Input	NA
Expected result	User Should receive a sign-up successfully email.
Status	Tested, passed.

Table 500 TC_18 Signup

5.1.19 Test Case for Feedback

Positive Test Case	
ID	TC_19
Priority	High
Description	To add feedback for admin.
Reference	Functional Requirement reference
Users	Users
Pre-requisites	A. System is online. B. User must have active login credentials C. User has internet access.
Steps	A. Install the app on your mobile. B. Login C. Click on the feedback button D. Enter the Valid details E. Press submit button
Input	NA
Expected result	Successfully, your feedback will be submitted.
Status	Tested, passed.

Table 55 TC_19 Feedback

5.1.20 Test Case for Feedback

Negative Test Case	
ID	TC_20
Priority	High
Description	Check the required fields by not filling any data
Reference	Functional Requirement reference
Users	Users
Pre-requisites	A. System is online. B. User has internet access. C. User must have active login credentials
Steps	A. Install the app on your mobile. B. Login C. Click on the feedback button. D. Passed blank spaces in required fields. E. Click on the submit Button.
Input	NA
Expected result	Validation error message for required fields should visible.
Status	Tested, passed.

Table 56 TC_20 Feedback

5.1.21 Test Case for Custom Request

Positive Test Case	
ID	TC_21
Priority	High
Description	To add custom request for nutritionist to add about more diet plan or any issue.
Reference	Functional Requirement reference
Users	Users
Pre-requisites	A. System is online. B. User must have active login credentials C. User has internet access.
Steps	A. Install the app on your mobile. B. Login C. Click on the diet plan button D. Click on custom request button E. Enter the Valid details F. Press submit button
Input	NA
Expected result	Successfully, your request will be submitted.
Status	Tested, passed.

Table 57 TC_21 Custom Request

5.1.22 Test Case for Custom Request

Negative Test Case	
ID	TC_22
Priority	High
Description	Check the required fields by not filling any data
Reference	Functional Requirement reference
Users	Users
Pre-requisites	A. System is online. B. User has internet access. C. User must have active login credentials
Steps	A. Install the app on your mobile. B. Login C. Click on the diet plan button D. Click on custom request button E. Passed blank spaces in required fields. E. Click on the submit Button.
Input	NA
Expected result	Validation error message for required fields should visible.
Status	Tested, passed.

Table 58 TC_22 Custom Request

5.1.23 Test Case for Add Dietitian

Positive Test Case	
ID	TC_23
Priority	High
Description	For Add new Nutritionist for diet plan
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin has internet access. C. Admin must have active login credentials
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Filled the required filled with valid data F. Click on the Add button
Input	NA
Expected result	Successfully Nutritionist will be added
Status	Tested, passed.

Table 59 TC_23 Add Dietitian

5.1.24 Test Case for Add Dietitian

Negative Test Case	
ID	TC_24
Priority	High
Description	Verify if blank spaces are passed in required fields
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin has internet access. C. Admin must have active login credentials
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed blank spaces in required fields F. Click on the Add button
Input	NA
Expected result	Validation error message for required fields should visible
Status	Tested, passed.

Table 60 TC_24 Add Dietitian

5.1.25 Test Case for Add Dietitian

Negative Test Case	
ID	TC_25
Priority	High
Description	Verify if invalid data are passed in required fields.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin has internet access. C. Admin must have active login credentials
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed invalid data in required fields. F. Click on the Add button
Input	NA
Expected result	Error message should be displayed.
Status	Tested, passed.

Table 61 TC_25 Add Dietitian

5.1.26 Test Case for Add Dietitian

Negative Test Case	
ID	TC_26
Priority	High
Description	Verify if we have filled all required fields but not filled the name field.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin has internet access. C. Admin must have active login credentials
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed data to all required field except name field. F. Click on the Add button
Input	NA
Expected result	The error message should be displayed on the name field.
Status	Tested, passed.

Table 62 TC_26 Add Dietitian

5.1.27 Test Case for Add Dietitian

Negative Test Case	
ID	TC_27
Priority	High
Description	Verify if we have filled name field but not filled the all required field
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin has internet access. C. Admin must have active login credentials
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed data to name filed and not filled all required field. F. Click on the Add button
Input	NA
Expected result	The error message should be displayed on the all required field.
Status	Tested, passed.

Table 510 TC_27 Add Dietitian

5.1.28 Test Case for Add Dietitian

Negative Test Case	
ID	TC_28
Priority	High
Description	Verify if we have filled all required fields but not filled the email field.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin has internet access. C. Admin must have active login credentials
Steps	C. Open the web link to system. D. Login C. Click on the Add Nutritionist D. Passed data to all required field except email field F. Click on the Add button
Input	NA
Expected result	The error message should be displayed on email field.
Status	Tested, passed.

Table 521 TC_28 Add Dietitian

5.1.29 Test Case for Add Dietitian

Negative Test Case	
ID	TC_29
Priority	High
Description	Verify if we have filled email field but not filled all the required field.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin has internet access. C. Admin must have active login credentials
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed data to all required field except email field. F. Click on the Add button
Input	NA
Expected result	The error message should be displayed on the all required field.
Status	Tested, passed.

Table 532 TC_29 Add Dietitian

5.1.30 Test Case for Add Dietitian

Negative Test Case	
ID	TC_30
Priority	High
Description	Verify if we have filled password field but not filled all the required field.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin has internet access. C. Admin must have active login credentials
Steps	C. Open the web link to system. D. Login C. Click on the Add Nutritionist D. Passed data to password filed and not filled all required field. F. Click on the Add button
Input	NA
Expected result	The error message should be displayed on the all required field.
Status	Tested, passed.

Table 543 TC_ 30 Add Dietitian

5.1.31 Test Case for Add Dietitian

Negative Test Case	
ID	TC_31
Priority	High
Description	Verify if we have filled all required fields but not filled the password field.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin has internet access. C. Admin must have active login credentials
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist. D. Passed data to all required field except email field. E. Click on the Add Button.
Input	NA
Expected result	The error message should be displayed on the password field.
Status	Tested, passed.

Table 554 TC_31 Add Dietitian

5.1.32 Test Case for Add Dietitian

Negative Test Case	
ID	TC_32
Priority	High
Description	Verify if we passed email without @.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin must have active login credentials C. Admin has internet access.
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed email without @. E. Click on the Add button.
Input	NA
Expected result	The error message should be displayed on the email field.
Status	Tested, passed.

Table 565 TC_38 Add Dietitian

5.1.33 Test Case for Add Dietitian

Negative Test Case	
ID	TC_33
Priority	High
Description	Verify if we passed repeated email.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin must have active login credentials C. Admin has internet access.
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed repeated email in email field. E. Click on the Add Button.
Input	NA
Expected result	The error message should be displayed on the email field.
Status	Tested, passed.

Table 576 TC_33 Add Dietitian

5.1.34 Test Case for Add Dietitian

Negative Test Case	
ID	TC_34
Priority	High
Description	Verify if we passed a password whose length is less than 8 characters.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin must have active login credentials C. Admin has internet access.
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed password whose length is less than 8 characters E. Click on the Add Button.
Input	NA
Expected result	The error message should be displayed on the password field.
Status	Tested, passed.

Table 587 TC_34 Add Dietitian

5.1.35 Test Case for Add Dietitian

Negative Test Case	
ID	TC_35
Priority	High
Description	Verify if we passed a password whose length is more than 16 characters.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin must have active login credentials C. Admin has internet access.
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed password whose length is less than 16 characters E. Click on the Add Button.
Input	NA
Expected result	The error message should be displayed on the password field.
Status	Tested, passed.

Table 598 TC_35 Add Dietitian

5.1.36 Test Case for Add Dietitian

Negative Test Case	
ID	TC_36
Priority	High
Description	Verify if we have filled email field but not filled the all required field.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Click on the Add Nutritionist. C. Admin has internet access.
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed data email filed and not filled all required field. E. Click on the Add Button.
Input	NA
Expected result	The error message should be displayed on the all required field.
Status	Tested, passed.

Table 609 TC_36 Add Dietitian

5.1.37 Test Case for Add Dietitian

Negative Test Case	
ID	TC_37
Priority	High
Description	Verify if passed digits data in user text field.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin must have active login credentials C. admin has internet access.
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist D. Passed digits in user text field. E. Click on the Add Button.
Input	1234, etc.
Expected result	The error message should be displayed on the name text field.
Status	Tested, passed.

Table 610 TC_37 Add Dietitian

5.1.38 Test Case for Add Dietitian

Negative Test Case	
ID	TC_38
Priority	High
Description	Verify if passed digits in email field.
Reference	Functional Requirement reference
Users	Admin
Pre-requisites	A. System is online. B. Admin must have active login credentials C. Admin has internet access.
Steps	A. Open the web link to system. B. Login C. Click on the Add Nutritionist. D. Passed digits in email field. E. Click on the Add Button.
Input	2456, etc.
Expected result	The error message should be displayed on the all required field.
Status	Tested, passed.

Table 621 TC_38 Add Nutritionist

5.1.39 Test Case for Add Food item

Positive Test Case	
ID	TC_39
Priority	High
Description	Add food item for user diet plan with valid data
Reference	Functional Requirement reference
Users	Dietitian
Pre-requisites	A. System is online. B. Dietitian must have active login credentials C. Dietitian has internet access.
Steps	A. Open the web link to system. B. Login C. Click on the Add food items. D. Filled the required filled with valid data E. Click on the Add Button.
Input	NA
Expected result	Successfully new Food item will be added
Status	Tested, passed.

Table 632 TC_39 Add Food item

5.1.40 Test Case for Add Food item

Negative Test Case	
ID	TC_40
Priority	High
Description	Verify if blank spaces are passed in required fields when Add food item.
Reference	Functional Requirement reference
Users	Dietitian
Pre-requisites	A. System is online. B. Dietitian must have active login credentials C. Dietitian has internet access.
Steps	A. Open the web link to system. B. Login C. Open food item page D. Passed blank spaces in required fields. E. Click on the Add button.
Input	NA
Expected result	Validation error message for required fields should visible.
Status	Tested, passed.

Table 643 TC_40 Add Food Item

5.1.41 Test Case for Add Food item

Negative Test Case	
ID	TC_41
Priority	High
Description	Verify if invalid data are passed in required fields when Add food item.
Reference	Functional Requirement reference
Users	Dietitian
Pre-requisites	A. System is online. B. Dietitian must have active login credentials C. Dietitian has internet access.
Steps	A. Open the web link to system. B. Login C. Open food item page D. Passed blank spaces in required fields. E. Click on the Add button.
Input	NA
Expected result	Error message should be displayed.
Status	Tested, passed.

Table 654 TC_41 Add Food item

5.1.42 Test Case for Add Food item

Negative Test Case	
ID	TC_42
Priority	High
Description	Verify if we filled all required field except food name.
Reference	Functional Requirement reference
Users	Dietitian
Pre-requisites	A. System is online. B. Dietitian must have active login credentials C. Dietitian has internet access.
Steps	A. Open the web link to system. B. Login C. Open food item page D. Filled all field except name field. E. Click on the Add button.
Input	NA
Expected result	Error message should be displayed on name text field.
Status	Tested, passed.

Table 665 TC_42 Add Food item

5.1.43 Test Case for Add Food item

Negative Test Case	
ID	TC_43
Priority	High
Description	Verify if we filled numeric data in food name text field.
Reference	Functional Requirement reference
Users	Dietitian
Pre-requisites	A. System is online. B. Dietitian must have active login credentials C. Dietitian has internet access.
Steps	A. Open the web link to system. B. Login C. Open food item page D. Passed blank spaces in required fields. E. Click on the Add button.
Input	NA
Expected result	Validation error message for required fields should visible.
Status	Tested, passed.

Table 676 TC_43 Add food item

5.1.44 Test Case for Add Food item

Negative Test Case	
ID	TC_44
Priority	High
Description	Check the required fields by not filling any data.
Reference	Functional Requirement reference
Users	Dietitian
Pre-requisites	D. System is online. E. Dietitian must have active login credentials F. Dietitian has internet access.
Steps	F. Open the web link to system. G. Login H. Open food item page I. Passed blank spaces in required fields. J. Click on the Add button.
Input	NA
Expected result	Validation error message for required fields should visible.
Status	Tested, passed.

Table 687 TC_44 Add Food item

5.2 Black Box Test Cases

5.2.1 Equivalence Partitions (EP)

5.2.1.1 Login

Variables	**Valid Classes**	**Invalid Classes**
email	1. unique 2. Case sensitive. 3. Compulsory field.	1. Email without @ 2. Empty Field.
Password	1. Length should be equal or greater than 8 characters. 2. May contain symbols, alphabets [a-z A-Z] and digits [0-9].	1. Length less than 8 characters. 2. Empty field.

Table 698 *EP Login*

5.2.1.2 Sign up

Variables	Valid Classes	Invalid Classes
name	1. A valid name should start with letter (a-z) (A-Z)	1. Empty Field. 2.. Numerical value
Age	1. A valid number should have only number	1. Empty field 2. Numeric value
Weight	1. A valid number should only in kg	1. Alphabet. 2. Empty field.
Height	1. Height in cm.	1. Only Alphabet 2. Empty Field
Gender	1. choose one item	
Number	1. A number should only Number	1. empty Field
Email	1 unique 2. Case sensitive. 3. Compulsory field	1. Email without @ 2.. Empty Field. Same two emails
Password	1. Length should be 2. equal or greater than 8 characters. 3. May contain symbols, alphabets [a-z A-Z] and digits [0-9].	1. Length less than 8 characters. 2. Empty field.

Table 709 EP Sign up

5.2.1.3 Add food Item

Variables	Valid Classes	Invalid Classes
Food Name	1. Case in-sensitive. 2. Compulsory field.	1. All Numeric data 4. Empty Field.
Food Detail	1. A valid should start with a letter (a-z)/(A-Z)	3. Empty field 4. Numeric value

Table 710 EP Add Food Item

5.2.1.4 Add food Plan

Variables	Valid Classes	Invalid Classes
Food Name	3. Case in-sensitive. 4. Compulsory field.	1. All Numeric data 4. Empty Field.
Food Detail	1. A valid should start with a letter (a-z)/(A-Z)	3. Empty field 4. Numeric value

Table 721 EP Add Food Plan

5.2.1.5 Add Dietitian

Variables	Valid Classes	Invalid Classes
Nutritionist Name	1. A valid should start with a letter (a-z)/(A-Z) 2.Compulsory field.	1. All Numeric data 4. Empty Field.
Nutritionist Detail	1. Compulsory field	3. Empty field 4. Numeric value

Table 732 EP Add Dietitian

5.2.2 Boundary Value Analysis

5.2.2.1 Email

MINIMUM VALUES			MAXIMUM VALUES		
Invalid Partition	**Valid Partition**	**Valid Partition (min + 1)**	**Valid Partition**	**Valid partition**	**Invalid Partition (max + 1)**
Empty	5 letters is valid input.	Greater than 6 letters.	Less than 29 letters.	30 letters.	Greater than 31 letters.

Table 743 BVA Email

5.2.2.2 Password

MINIMUM VALUES			MAXIMUM VALUES		
Invalid Partition	**Valid Partition**	**Valid Partition (min + 1)**	**Valid Partition**	**Valid partition**	**Invalid Partition (max + 1)**
Empty	8 letters is valid input.	Greater than 9 letters.	Less than 29 letters.	30 letters.	Greater than 31 letters.

Table 754 BVA Password

5.2.2.3 Name

MINIMUM VALUES			MAXIMUM VALUES		
Invalid Partition	**Valid Partition**	**Valid Partition (min + 1)**	**Valid Partition**	**Valid partition**	**Invalid Partition (max + 1)**
Empty	3 letters are valid input.	Greater than 4 letters.	Less than 49 letters.	50 letters.	Greater than 51 letters.

Table 765 BVA Name

5.2.2.4 Age

MINIMUM VALUES			MAXIMUM VALUES		
Invalid Partition	**Valid Partition**	**Valid Partition (min + 1)**	**Valid Partition**	**Valid partition**	**Invalid Partition (max + 1)**
Empty	Only Number is valid input.	Greater than 1 digit.	Less than 3-digit number	2 digits.	Greater than 1 digit

Table 776 BVA Age

5.2.2.5 Height

MINIMUM VALUES			MAXIMUM VALUES		
Invalid Partition	**Valid Partition**	**Valid Partition (min + 1)**	**Valid Partition**	**Valid partition**	**Invalid Partition (max + 1)**
Empty	3 letters is valid input.	Greater than 1 digit	Less than 3 digits.	Only cm.	Greater than 2 digits.

Table 787 BVA Height

5.2.2.6 Weight

MINIMUM VALUES			MAXIMUM VALUES		
Invalid Partition	**Valid Partition**	**Valid Partition (min + 1)**	**Valid Partition**	**Valid partition**	**Invalid Partition (max + 1)**
Empty	3 digits is valid input.	Greater than 1 digit.	Less than 3 digits	3 digits.	Greater than 2 digits

Table 798 BVA Weight

5.2.2.7 Contact Number

MINIMUM VALUES			MAXIMUM VALUES		
Invalid Partition	**Valid Partition**	**Valid Partition (min + 1)**	**Valid Partition**	**Valid partition**	**Invalid Partition (max + 1)**
Empty	11 digits is valid input.	Greater than 12 digits.	Less than 12 digits.	14 digits.	Greater than 14 digits.

Table 809 BVA Contact Number

5.2.2.8 Gender

MINIMUM VALUES			MAXIMUM VALUES		
Invalid Partition	**Valid Partition**	**Valid Partition (min + 1)**	**Valid Partition**	**Valid partition**	**Invalid Partition (max + 1)**
Empty	Select one item.	Only one item can select	One item can be select.	One item.	One item.

Table 810 BVA Gender

5.2.3 Decision Value Analysis

5.2.3.1 Admin Login

Condition 1: Admin should enter valid email

Condition 2: Admin should enter valid password.

Result 1: Admin login successfully.

Result 2: Admin haven't entered email, password or wrong password

CONDITIONS	VALUES	COMBINATIONS			
Condition 1	T, F	T	T	F	F
Condition 2	T, F	T	F	T	F
RESULTS					
Result 1		T	F	F	F
Result 2		F	T	T	T

Table 821 Decision table for Admin Login

5.2.3.2 Admin Logout

Condition 1: Admin should login.
Condition 2: Admin press logout button.

Result 1: Logout successfully.
Result 2: Admin not logout.

CONDITIONS	VALUES	COMBINATIONS			
Condition 1	T, F	T	T	F	F
Condition 2	T, F	T	F	T	F
RESULTS					
Result 1		T	F	F	F
Result 2		F	T	T	T

Table 92 Decision table for Admin Logout

5.2.3.3 Dietitian Sign up

Condition 1: Dietitian should enter valid full name.

Condition 2: Dietitian should enter valid email address.

Condition 3: Dietitian should enter valid password.

Condition 4: Dietitian should repeat password.

Result 1: Dietitian signup successfully.

Result 2: Dietitian haven't entered full name, password, repeat password, email address

CONDITIONS	VALUES	COMBINATIONS															
Condition 1	T, F	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
Condition 2	T, F	T	T	T	T	F	F	F	F	T	T	T	T	F	F	F	F
Condition 3	T, F	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F	F
Condition 4	T, F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
Results																	
Result 1		T	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Result 2		F	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

Table 833 Decision table for Dietitian Sign Up

5.2.3.4 Dietitian Login

Condition 1: Admin should enter valid email

Condition 2: Admin should enter valid password.

Result 1: Dietitian login successfully.

Result 2: Dietitian haven't entered email, password or wrong password

CONDITIONS	VALUES	COMBINATIONS			
Condition 1	T, F	T	T	F	F
Condition 2	T, F	T	F	T	F
RESULTS					
Result 1		T	F	F	F
Result 2		F	T	T	T

Table 844 Decision table for Dietitian Login

5.2.3.5 Dietitian Logout

Condition 1: Dietitian should login.
Condition 2: Dietitian press logout button.

Result 1: Logout successfully.
Result 2: Dietitian not logout.

CONDITIONS	VALUES	COMBINATIONS			
Condition 1	T, F	T	T	F	F
Condition 2	T, F	T	F	T	F
RESULTS					
Result 1		T	F	F	F
Result 2		F	T	T	T

Table 855 Decision table for Dietitian Logout

5.2.3.6 User Sign Up

Condition 1: User should enter valid name.

Condition2: User should enter valid Age.

Condition3: User should enter valid Height.

Condition4: User should enter valid Weight.

Condition4: User should select Gender

Condition5: User should enter valid phone no.

Condition 6: User should enter valid email address.

Condition 7: User should enter valid password.

Result 1: User signup successfully.

Result 2 User haven't entered name, password, repeat password, email address or phone no.

CONDITIONS	VALUES	COMBINATIONS																															
Condition 1	T, F	T	T	T	T	T	F	F	F	F	F	T	T	T	T	T	T	F	F	F	T	T	T	F	F	F	F	T	F	F	F	T	F
Condition 2	T, F	T	T	T	T	F	T	F	T	T	T	F	F	F	T	T	T	F	T	T	F	F	T	F	F	F	T	F	F	T	T	F	T
Condition 3	T,F	T	T	T	F	T	T	T	F	T	T	F	T	T	F	F	T	F	F	T	F	T	F	F	F	T	F	F	F	F	T	T	T
Condition 4	T,F	T	T	F	T	T	T	T	T	F	T	T	F	T	F	T	F	T	F	F	F	F	F	F	T	F	F	F	F	T	F	T	T
Condition 5:	T, F	T	F	T	F	T	T	T	T	T	F	T	T	F	T	F	F	T	T	F	T	F	F	T	F	F	F	F	F	F	T	F	T
Condition 6:	T, F	T	F	T	F	T	T	T	T	T	F	T	T	F	T	F	F	T	T	F	T	F	F	T	F	F	F	F	F	F	T	F	T
Condition 7:	T, F	T	F	T	F	T	T	T	T	T	F	T	T	F	T	F	F	T	T	F	T	F	F	T	F	F	F	F	F	F	T	F	T
Condition 8:	T, F	T	F	T	F	T	T	T	T	T	F	T	T	F	T	F	F	T	T	F	T	F	F	T	F	F	F	F	F	F	T	F	T
RESULTS																																	
Result 1		T	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Result 2		F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	T

Table 866 Decision table for User Sign Up

5.2.3.7 User Login

Condition 1: User should enter valid email

Condition 2: Customer should enter valid password.

Result 1: User login successfully.

Result 2: User haven't entered email, password or wrong password

CONDITIONS	VALUES	COMBINATIONS			
Condition 1	T, F	T	T	F	F
Condition 2	T, F	T	F	T	F
RESULTS					
Result 1		T	F	F	F
Result 2		F	T	T	T

Table 877 Decision table for User Login

5.2.3.8 User Update Profile

Condition 1: User should enter valid name.

Condition2: User should enter valid Age.

Condition3: User should enter valid Height.

Condition4: User should enter valid Weight.

Condition4: User should select Gender

Condition5: User should enter valid phone no.

Condition 6: User should enter valid email address.

Condition 7: User should enter valid password.

Result 1: User signup successfully.

Result 2 User haven't entered name, password, repeat password, email address or phone no.

CONDITIONS	**VALUES**	**COMBINATIONS**																													
Condition 1	T, F	T	T	T	T	T	F	F	F	F	F	T	T	T	T	T	T	F	F	F	T	T	T	F	F	F	F	T	F	F	F	T	F
Condition 2	T, F	T	T	T	T	F	T	F	T	T	T	F	F	F	T	T	T	F	T	T	F	F	T	F	F	F	T	F	F	T	T	F	T
Condition 3	T,F	T	T	T	F	T	T	T	F	T	T	F	T	T	F	F	T	F	F	T	F	T	F	F	F	T	F	F	F	F	T	T	T
Condition 4	T,F	T	T	F	T	T	T	T	T	F	T	T	F	T	F	T	F	T	F	F	F	F	F	F	T	F	F	F	F	T	F	T	T
Condition 5:	T, F	T	F	T	F	T	T	T	T	T	F	T	T	F	T	F	F	T	T	F	T	F	F	T	F	F	F	F	F	F	T	F	T
Condition 6:	T, F	T	F	T	F	T	T	T	T	T	F	T	T	F	T	F	F	T	T	F	T	F	F	T	F	F	F	F	F	F	T	F	T
Condition 7:	T, F	T	F	T	F	T	T	T	T	T	F	T	T	F	T	F	F	T	T	F	T	F	F	T	F	F	F	F	F	F	T	F	T
Condition 8:	T, F	T	F	T	F	T	T	T	T	T	F	T	T	F	T	F	F	T	T	F	T	F	F	T	F	F	F	F	F	F	T	F	T
RESULTS																																	
Result 1		T	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
Result 2		F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	T

Table 888 Decision table for User Update Profile

5.2.4 State transition Testing

5.2.4.1 Login

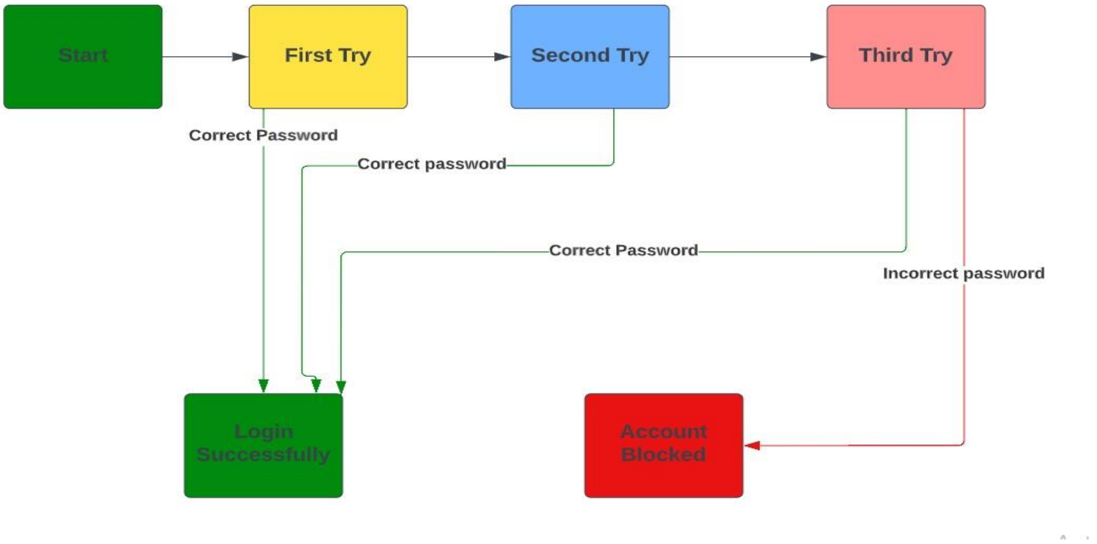


Figure 635 State transition diagram for Login

State Transition Table

	Correct Password	Incorrect Password
S1) Start	S5	S2
S2) First Try	S5	S3
S3) Second Try	S5	S4
S4) Third Try	S5	S6
S5) Login Successfully	-	-
S6) Account Blocked	-	-

Table 899 State transition table for login

5.2.4.2 Signup

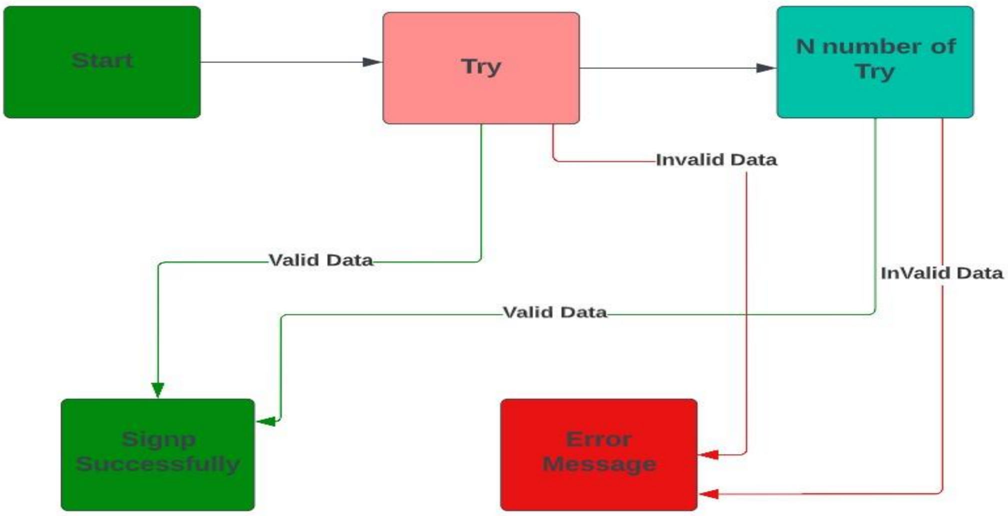


Figure 646 State transition diagram for Sign Up

State Transition Table

	Valid Data	Invalid Data
S1) Start	**S4**	**S5**
S2) Try	**S4**	**S5**
S3) Nth Try	**S4**	**S5**
S4) Signup Successfully	**-**	**-**
S5) Error Message	**-**	**-**

Table 900 State transition diagram for Login

5.2.5 Use Case Testing

	Steps	**Description**
Main Scenario **U: User** **S: System**	**1**	U: Signup
	a	S: Show Signup Page
		U: Enter Details
	b	S: Check Valid Details
	2	U: Login
	a	S: Show Login Page
		U: Enter Email and Password
	b	S: Check for valid Email and password
		U: Check the Home page
	3	U: Check the Diet plan
	a	S: Show Diet plan
		U: Create a Custom request
	b	S: Save the request

Table 911 Use case testing

5.3 White Box Test Cases

5.3.1 Cyclometric complexity

Login

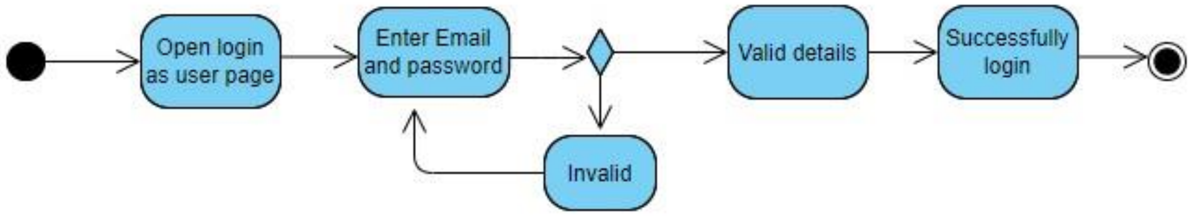


Figure 657 White box test for Login

E=6s N=8

CC=E-N+2

CC=4

Path Coverage

Start State-Open App-Enter Email and Password—Click Login-Check Validation Valid Details-Redirect in the home page

Start State-Open Website-Enter Email and Password—Click Login-Check Validation Invalid Details-Redirect to login page

Statement Coverage

The statement coverage is 1 because all nodes are coverage

Sign Up

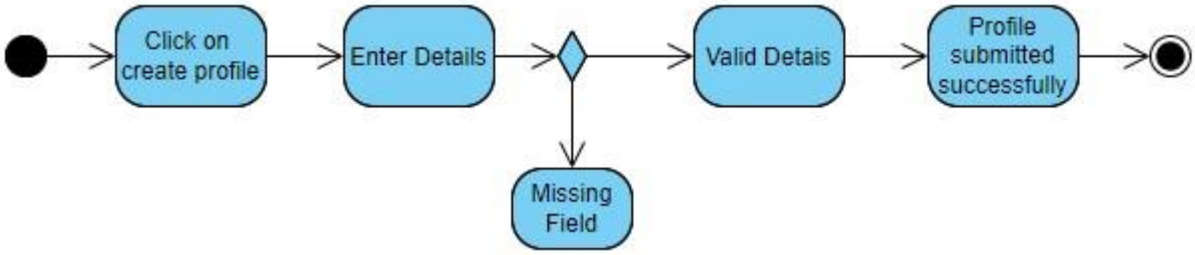


Figure 668 White box test for Sign Up

E=6s N=4

CC=E-N+2

CC=4

Path Coverage

There are two paths

Start State-Create Profile-Enter Detail-Click Save-Check Validation-Valid Details-Create Successfully

Start State-Enter Details-Click Save-Check Validation-Invalid Details-Redirect to create profile

Statement Coverage

The statement coverage is 1 because all nodes are coverage

Update Profile

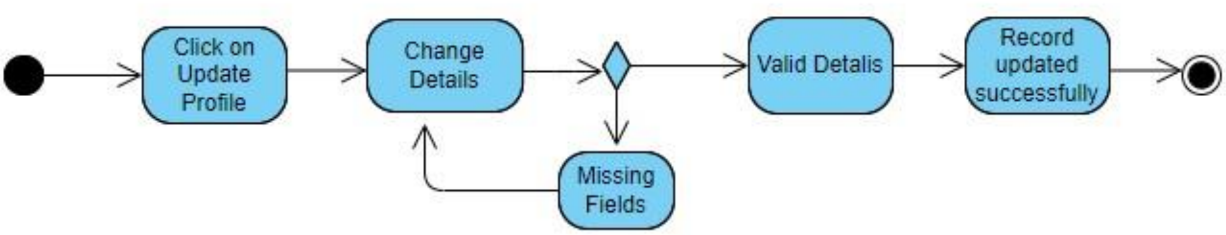


Figure 679 White box test for Update Profile

E=8s N=8
CC=E-N+2
CC=2

Path Coverage

There are two paths

Start State Click update profile-Enter Details-Click Save-Check Validation-Valid Details-Updated

Start State-Click update profile-Enter Details-Click Save-Check Validation-Invalid Details-
Redirect to update profile

Statement Coverage

The statement coverage is 1 because all nodes are coverage

Custom Request

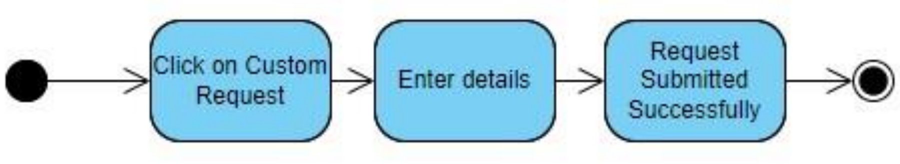


Figure 680 White box test for Custom Request

E=4s N=5
CC=E-N+2
CC=1

Path Coverage

There are one paths

Start State Click custom request-Enter Details-Click Submit-Check Validation-successfully send

Statement Coverage

The statement coverage is 1 because all nodes are coverage

Add Dietition

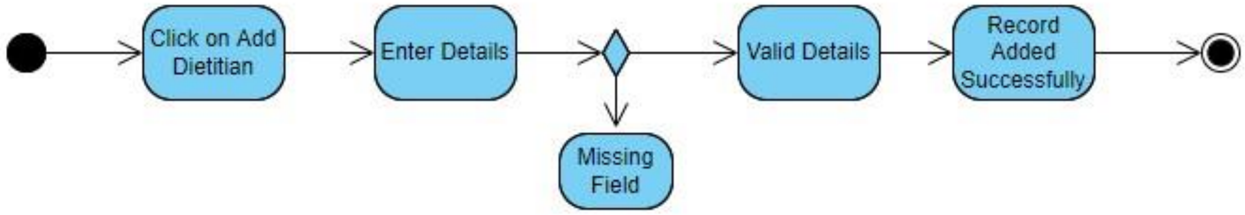


Figure 691 White box test for Add Dietitian

E=7s N=8
CC=E-N+2
CC=3

Path Coverage

There are two paths

Start State Click Add Dietitian-Enter Details-Click Save-Check Validation-Valid Details-Redirect to home Screen

Start State Click Add Dietitian-Enter Details-Click Save-Check Validation-Valid Details-submit-Redirect to add dietitian page

Statement Coverage

The statement coverage is 1 because all nodes are coverage

Add Food Item

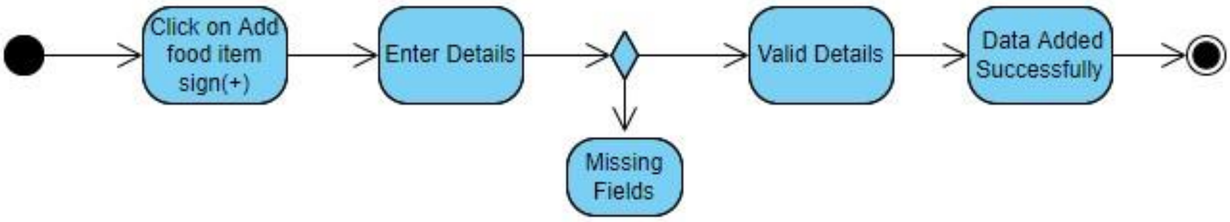


Figure 702 White box test for Add Food Item

E=8s N=8

CC=E-N+2

CC=2

Path Coverage

There are one paths

Start State Click Add Food Item-Enter Details-Click Save-Check Validation-Valid Details-Add successfully

Statement Coverage

The statement coverage is 1 because all nodes are coverage

Update Food Plan

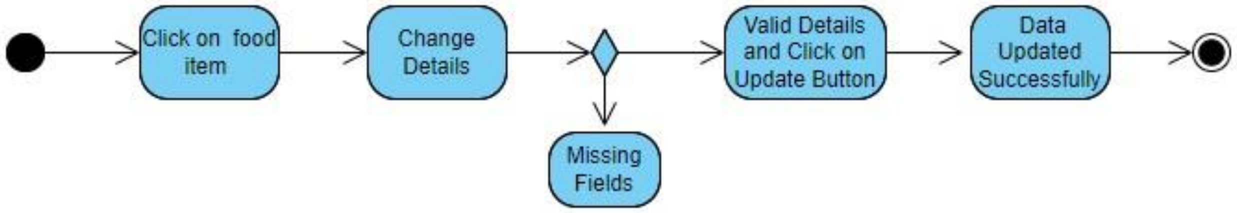

Click on food item
Change Details
Valid Details and Click on Update Button
Data Updated Successfully
Missing Fields

Figure 713 White box test for Update Food Plan

E=8s N=8
CC=E-N+2
CC=2

Path Coverage

There are two paths

Start State Click Add Food Item-Enter Details-Click Save-Check Validation-Valid Detail-Add successfully

Start State-Enter Details-Click Save-Check Validation-Invalid Details-Redirect to update food plan

Statement Coverage

The statement coverage is 1 because all nodes are coverage

5.4 Performance testing

The focus of Performance Testing is checking a software program

Speed

In this testing, we determined that App service responds quickly

Scalability

In this service we determine the maximum load can be handled it.

Stability

In this testing, we determined that Smart Diet App is stable under varying loads

Performance Test Cases

The app be able to handle the proposed load

The help of performance metrics checks for performance bottlenecks.

The app as well as server response quickly.

Check the app performance concerning various network strength irregularities.

Response time of the application under low, normal, moderate and heavy load conditions is good.
The app can handle maximum load.

5.5 Stress Testing

Stress Testing is used to verified the application stability and reliability; it is used to test the robustness of the system in extremely heavy conditions. During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress.

1. Many users can use this application at one time.
2. System can bear the load of many users and performance of each user at on single time

5.6 System Testing

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective. System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing. In Smart Diet A system testing results shows:

1.Each function performs its function according to functional requirements.

2.Every function is behaving smoothly and fully

3. Smart Diet App is performed the same behavior in a different environment.

5.7 Regression Testing

Regression testing a black box testing technique that consists of re-executing those tests that are impacted by the code changes. These tests should be executed as often as possible throughout the software development life cycle. Types of Regression Tests:

Final Regression Tests

A "final regression testing" is performed to validate the build that hasn't changed for a period of time. This build is deployed or shipped to customers.

Regression Tests

A normal regression testing is performed to verify if the build has NOT broken any other parts of the application by the recent code changes for defect fixing or for enhancement.

5.7.1 Selecting Regression Tests

- Requires knowledge about the system and how it affects by the existing functionalities.
- Tests are selected based on the area of frequent defects.
- Tests are selected to include the area, which has undergone code changes many a times.
- Tests are selected based on the criticality of the features.

5.7.2 Regression Testing Steps

Regression tests are the ideal cases of automation which results in better **R**eturn **on** **I**nvestment (ROI).

- Select the Tests for Regression.
- Choose the apt tool and automate the Regression Tests
- Verify applications with Checkpoints
- Manage Regression Tests/update when required
- Schedule the tests
- Integrate with the builds
- Analyze the results

Chapter 6: Tools and Techniques

Languages

6.1.1) Frontend

1. CSS: CSS has been used for all the designing part

2. React Native: We have used react native for fronted in this project. React native is a JavaScript library

3. Apache2: Project will be run over the Apache2 server

6.1.2) Backend

1. PHP(Laravel)

2. Pickle (Python library) for loading our trained model

Applications and tools

1. Visual studio code: We have used a visual code editor for this project

Libraries and Extensions if any

Chapter 7: Summary and Conclusion

This chapter include the summary and the conclusion

Summary

Smart Diet App is based on Android Artificial Intelligence smart diet consultant app. Which is basically based on user body type, weight, height, age, sex and physically activity. It calculates BMI and tells us how much calories should user to intake per day. This application is for any type of body person, it also suits for any range of weight people. It will ideally inform how should we cut down the weight using target program such as 1LB per week gain/loose. This system also provides the diet plan according to the information that send by the user. The system asks all his data from the user and processes it to provide the diet plan to the user. The user does not need to visit any dietitian which also saves time and the user can get the required diet plan in just one click. The system will give more accurate results as it accepts the data entered by the user and processes it depending on some metrics already known to the application on the basis of which a diet plan is generated and ask the user if the user accepts the diet plan.

The app’s machine-learning algorithm gathers data about individuals from their health tracking devices and food logging to personalize food suggestions using a bot to deliver answers throughout the day. Artificial Intelligence and its subsets have been leveraged by these platforms to identify the calorie intake and then make food recommendations for a healthy diet.

Conclusion

The world is rapidly evolving and heading toward technical expertise. The technique of artificial intelligence, you will get access to all the facilities via this application, which is actually provided by a human dietitian. The main advantage of using this standalone AI Diet Consultant application is that the time required by the people to travel to the dietitian will be reduced and also it reduces the cost of hiring dietitians for some particular purpose. Also, this AI Diet Consultant application offers more than one diet plan.

Chapter 8: User Manual

Login

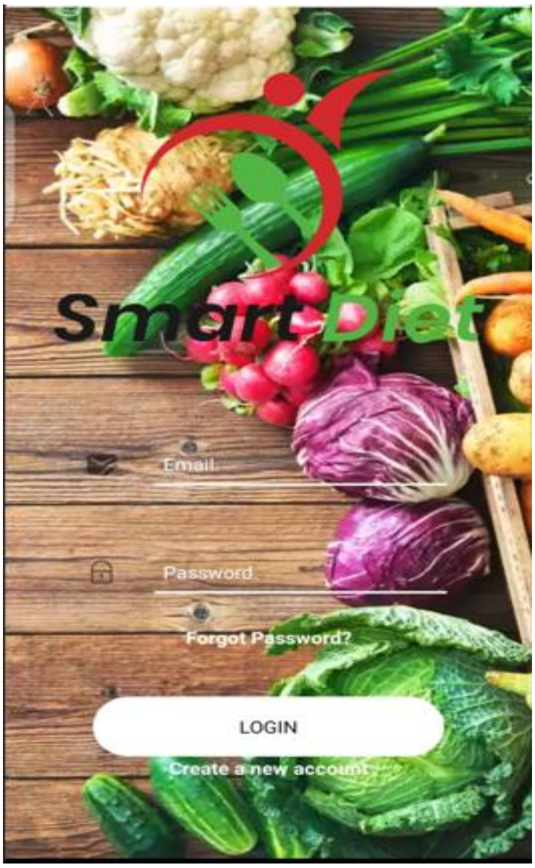


Figure 724 Login Prototype Page

User must have

1. Active username and password.
2. Mobile app

Step
:

1. Open login app page
2. Enter login details.
3. Click Login.
4. Done.

Sign Up

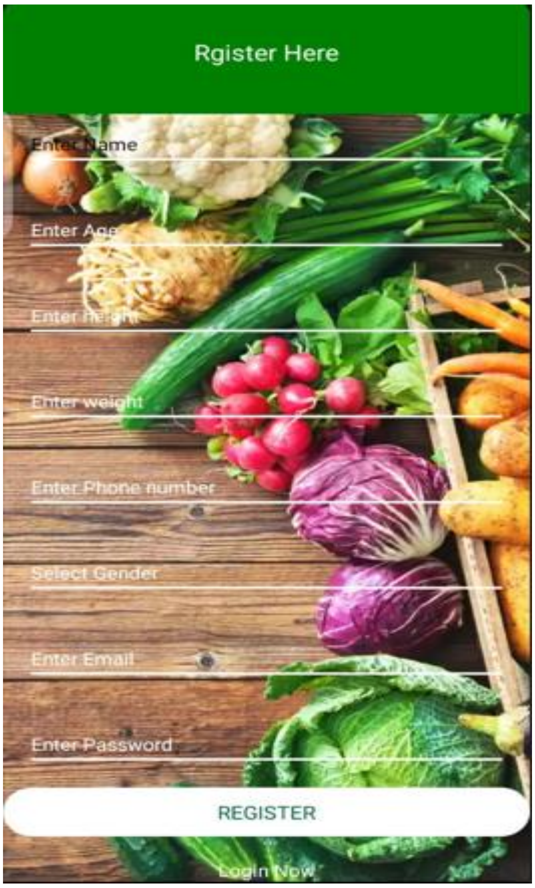


Figure 735 Sign up Prototype Page

User must have

3. app signup address.

Steps:

5. Open signup app page
6. Enter signup details.
7. Click signup.
8. Done.

User Home Page

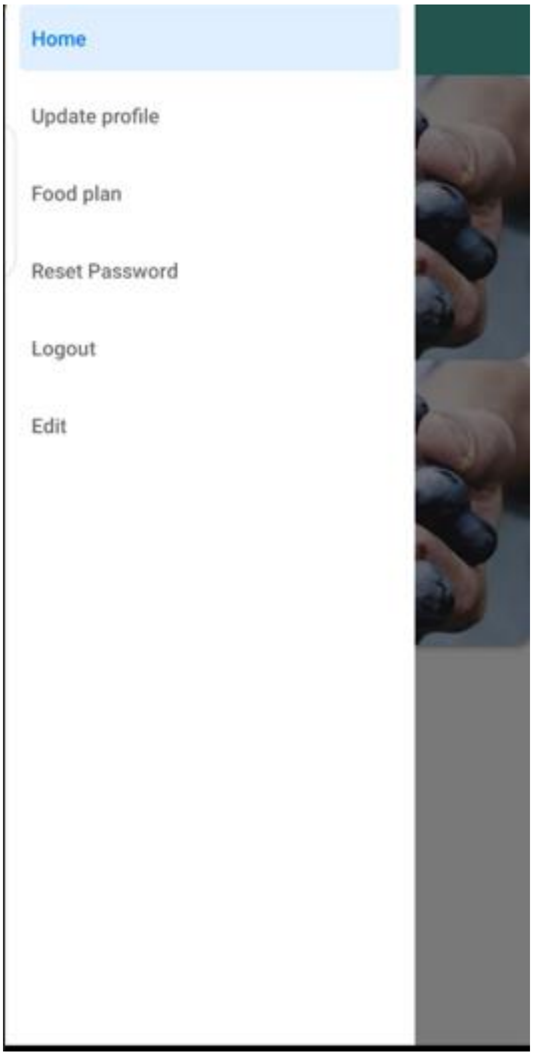


Figure 746 Home Prototype Page

User must have

3. App Home page.

Steps:

5. Open Login app page
6. Enter Login details.
7. Click on login Button.
8. Home page will be open
9. Done.

Admin Login Page

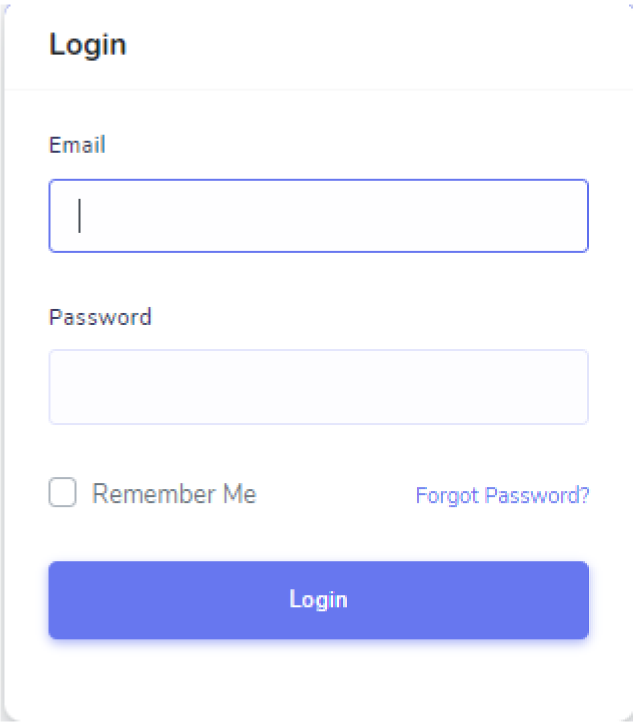


Figure 757 Login Prototype Page

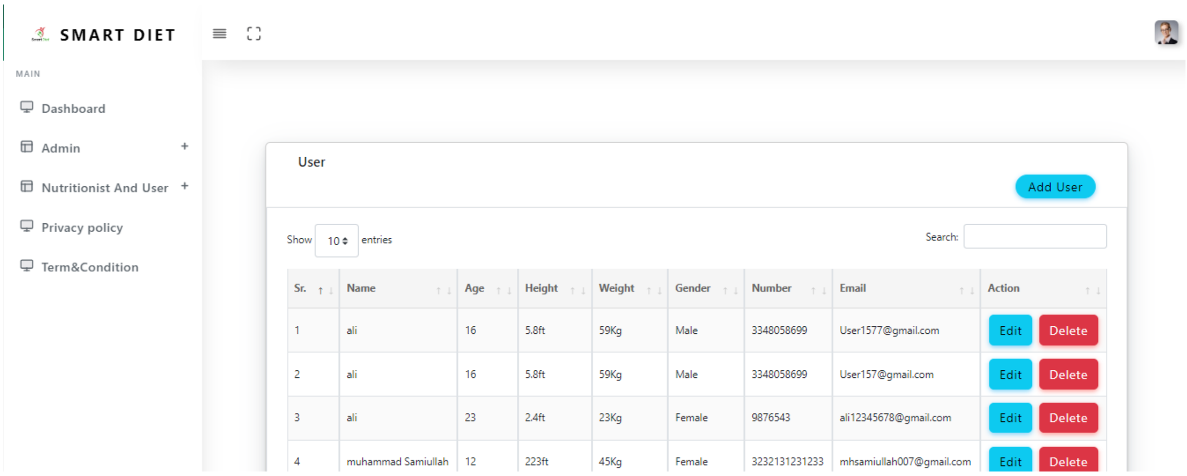
User must have

3. Active username and password.
4. Web Login Address

Steps:

5. Open login webpage
6. Enter login details.
7. Click Login.
8. Done.

Admin Home Page



Sr.	Name	Age	Height	Weight	Gender	Number	Email	Action
1	ali	16	5.8ft	59Kg	Male	3348058699	User1577@gmail.com	Edit Delete
2	ali	16	5.8ft	59Kg	Male	3348058699	User157@gmail.com	Edit Delete
3	ali	23	2.4ft	23Kg	Female	9876543	ali12345678@gmail.com	Edit Delete
4	muhammad Samiullah	12	223ft	45Kg	Female	3232131231233	mhsamiullah007@gmail.com	Edit Delete

Figure 768 Home Prototype Page

User must have

3. Web Page Address.

Steps:

5. Open Login Web page
6. Enter Login details.
7. Click on login Button.
8. Home page will be open
9. Done.

References

1. Maduwanthi, K., *Rule Based Diet Planning Mobile Application*. 2021.
2. Kulkarni, A. and S. Sarode, *Fitness Amigo: A Smart Diet Planner.*
3. Kahraman, A. and H.A. Seven. *Healthy daily meal planner*. in *Proceedings of the 7th annual workshop on genetic and evolutionary computation*. 2005.
4. West, J.H., et al., *Controlling your "app" etite: how diet and nutrition-related mobile apps lead to behavior change.* JMIR mHealth and uHealth, 2017. **5**(7): p. e7410.
5. Lee, W., et al., *Evaluation of a mobile phone-based diet game for weight control.* Journal of telemedicine and telecare, 2010. **16**(5): p. 270-275.
6. Oh, S.-J., *Design of Sensor-based Healthy Diet App for Smartphones.* The Journal of the Institute of Internet, Broadcasting and Communication, 2014. **14**(4): p. 141-147.
7. Farrokhi, A., et al., *Application of Internet of Things and artificial intelligence for smart fitness: A survey.* Computer Networks, 2021. **189**: p. 107859.
8. Pathipati, M.P., et al., *Digital health for functional gastrointestinal disorders.* Neurogastroenterology & Motility, 2021: p. e14296.
9. Quelly, S.B., A.E. Norris, and J.L. DiPietro, *Impact of mobile apps to combat obesity in children and adolescents: a systematic literature review.* Journal for Specialists in Pediatric Nursing, 2016. **21**(1): p. 5-17.
10. Sandborg, J., et al., *Effectiveness of a smartphone app to promote healthy weight gain, diet, and physical activity during pregnancy (HealthyMoms): Randomized controlled trial.* JMIR mHealth and uHealth, 2021. **9**(3): p. e26091.