

BSCS FINAL PROJECT

Requirements Specification

SugarSage – AI Companion For Diabetics



Project Advisor

Dr. Imran Arshad Choudhry

Presented by:

Group ID: F23CS014

Student Reg#

L1F20BSCS0204

L1F20BSCS0207

L1F20BSCS0466

L1F20BSCS0467

Student Name

Khizar Iqbal

Ahmed Ali

Hamail Shahbaz

Safoora Masood

Faculty of Information Technology

University of Central Punjab

Software Requirements Specification

Version 01

SugarSage – AI Companion For Diabetics

Advisor: Dr. Imran Arshad Choudhry

Group: F23CS014

| Member Name | Primary Responsibility |
|--------------------|-------------------------------|
| Khizar Iqbal | Introduction and Background |
| Hamail Shahbaz | Overall Description |
| Ahmed Ali | Functional Requirements |
| Safoora Masood | Nonfunctional Requirements |

Table of Contents

| | |
|--|-----------|
| Table of Contents | i |
| Revision History | ii |
| Abstract..... | 1 |
| 1. Introduction and Background | 2 |
| 1.1 Product (Problem Statement)..... | 2 |
| 1.2 Background..... | 3 |
| 1.3 Scope | 3 |
| 1.4 Objective(s)/Aim(s)/Target(s) | 3 |
| 1.5 Challenges | 4 |
| 1.6 Learning Outcomes | 4 |
| 1.7 Nature of End Product | 4 |
| 1.8 Completeness Criteria..... | 4 |
| 1.9 Business Goals..... | 4 |
| 1.10 Related Work/ Literature Survey/ Literature Review..... | 5 |
| 2. Overall Description..... | 8 |
| 2.1 Product Features | 8 |
| 2.2 User Classes and Characteristics | 8 |
| 2.3 Operating Environment | 9 |
| 2.4 Design and Implementation Constraints..... | 9 |
| 2.5 Assumptions and Dependencies | 10 |
| 3. Functional Requirements | 11 |
| 3.1 Patient Use Cases | 11 |
| 3.2 Admin Use Cases | 17 |
| 3.3 Requirements Analysis and Modeling..... | 28 |
| 3.3.1 Use Case Diagram..... | 28 |
| 3.3.2 ER Diagram..... | 30 |
| 3.3.3 Abstract Class Diagram..... | 31 |
| 3.3.4 Sequence Diagram | 33 |
| 4. Nonfunctional Requirements | 34 |
| 4.1 Performance Requirements..... | 34 |
| 4.2 Safety Requirements..... | 34 |
| 4.3 Security Requirements..... | 34 |
| 4.4 Additional Software Quality Attributes..... | 35 |
| 5. Revised Project Plan | 36 |
| 6. References..... | 36 |

Revision History

| Name | Date | Reason For Changes | Version |
|------|------|--------------------|---------|
| | | | |
| | | | |

Abstract

This project introduces SugarSage, an AI-based mobile application specifically designed to address the complex dietary management needs of diabetic patients in Pakistan. The significance of this problem is emphasized by Pakistan's high diabetes prevalence, with a staggering 31% of the population affected. SugarSage aims to fill the gap in available resources for diabetic individuals by offering personalized dietary recommendations that consider individual food preferences, sugar levels, energy requirements, and locally available food options.

The application leverages machine learning techniques to analyze various factors influencing dietary choices and suggests optimal diet plans. Additionally, it enables users to record insulin intake, track walking and sleeping patterns, and maintain a calorie count, thus providing a comprehensive solution for diabetes management. The project's scope includes not only dietary recommendations but also the development of a user-friendly interface and functionalities that allow for effective monitoring of diabetes-related activities.

The outcomes of this project are expected to empower diabetic patients in Pakistan to take control of their diabetes management process, thereby improving their overall well-being and quality of life. The project also aims to provide valuable insights into the application of machine learning algorithms in a real-life healthcare scenario, along with experience in mobile application development and backend management.

1. Introduction and Background

Managing the dietary needs of individuals with diabetes has always been a tricky and complex task, and it becomes even more challenging for diabetic patients in Pakistan who lack readily available resources to help them cope with their food-related issues. The food they consume plays a crucial role in maintaining their sugar levels and managing the severity of their diabetes, making it essential to have an effective dietary plan.

While doctors can recommend food options, their pre-made generic eating plans may not consider individual preferences and health-related information nor take into account all the possible effective and appealing diets for each individual. Moreover, doctors may not always be available to monitor and track patients' diabetes-related activities, leaving patients with limited support and guidance. With Pakistan having the world's highest diabetes ratio of 31%, it makes a dire need to develop a system that can manage the dietary requirements of diabetics, including keeping track of their everyday routines such as walking, sleeping, and insulin intake.

SugarSage is an AI-based mobile application that uses machine learning techniques to suggest optimal diet plans for people with diabetes while considering their food preferences, sugar levels, and energy requirements. The app takes a systematic approach to process all the possibilities and gives the most effective option for individual users.

In addition to providing personalized diet plans, SugarSage also allows users to record their insulin intake, track their walking and sleeping, and keep a calorie track. These features enable users to monitor their diabetes-related activities and make informed decisions about their health. The app's user-friendly interface and functionalities make managing dietary needs cost-efficient and convenient for diabetic patients in Pakistan.

SugarSage empowers diabetic patients in Pakistan to take control of their diabetes management process and overcome the challenges associated with managing their dietary needs. The app's comprehensive features and functionalities provide users with a personalized and effective solution to manage their diabetes, improving their overall well-being and quality of life.

1.1 Product (Problem Statement)

In Pakistan, 31% of the population is affected by diabetes, a chronic condition with severe repercussions when poorly managed. Poorly managed diabetes can lead to vision loss and limb amputations, according to the World Health Organization. Diet is one of the key factors in effective diabetes management. A well-balanced diet can significantly control blood sugar levels, reduce dependency on medication, and mitigate the risk of complications. However, medications remain the more commonly utilized approach to manage the condition. These medications can introduce additional health issues, such as hormonal imbalances and gastrointestinal disturbances.

Unfortunately, the currently available diet plans for diabetics lack personalization. Also it is impossible for a doctor to provide 24/7 monitoring. To fill this gap, we introduce SugarSage, an AI-driven mobile application that provides personalized dietary management solutions to diabetics in Pakistan. With SugarSage, users can access around-the-clock monitoring and receive locally relevant dietary advice tailored to their individual preferences and health requirements. The app integrates state-of-the-art machine learning algorithms to analyze dietary choices and suggest suitable meal plans. It also allows users to track their physical activities and insulin levels and maintain a healthy lifestyle. SugarSage is a complete diabetic care system that offers a user-friendly interface, making it a valuable resource for managing diabetes effectively and conveniently.

1.2 Background

Pakistan has the highest ratio of diabetics in the world. Despite being the most in need of awareness and facilities to help people manage diabetes efficiently, there is a lack of both. Living with diabetes requires managing it through a combination of medication and diet. While medication is important, it's equally important to keep an eye on what a diabetic person eats. Their diet can either contribute to the development of diabetes or help reduce it. In some cases, a balanced diet can even control diabetes more efficiently than medication because medications can cause complications such as stomach problems, kidney issues, hormonal imbalances, and more.

Although diabetes dietitians can provide diet plans for diabetics, they often lack enough customization based on the patient's taste preferences. While if asked, they may make some adjustments to a rigid plan, it may not be as effective as needed. However, if a system could access a large database of foods, consider the complexities of what is good and what is not for a particular diabetic person, and consider the patient's preferences, it could offer a tailored diet plan. Such a system would be a step forward in helping patients manage their diabetes with ease.

1.3 Scope

The main scope of this project lies in dietary recommendations for diabetics based on these factors:

1. Their sugar levels
2. Their food preferences
3. Their insulin intake
4. Their energy requirements
5. Locally available food options

The secondary scope lies in that the app will be able to track sugar (user input dependent), detect and count daily steps, detect, and measure sleep time. The app will provide users with access to blogs and news about diabetes and diet. The

This system does not aim to cure but control diabetes through diet. The system cannot assist people with severe diabetes complications that can no longer be controlled merely through diet. The system does not provide an alternative to a real-life diabetes doctor who can provide professional medical consultation.

It must be noted that any misuse of the system or failure to use it as intended falls outside the scope of this project. The system is not responsible for forcing users to comply or recover from misuse, as these are not part of the project's objectives.

1.4 Objective(s)/Aim(s)/Target(s)

- The main objective is to provide ease to Pakistani diabetes patients in managing their diet.
- Patients can easily manage diabetes with an all-in-one app for tracking sugar levels, calories, and activities.
- Providing patients with 24/7 accessibility for easy management of diabetes.

1.5 Challenges

- Acquiring datasets for diabetes patients
- Developing a comprehensive food database
- Collecting local foods and their nutrients
- Learning and making Mobile Application
- Learning and implementing Data Science and Machine Learning
- Learning and implementing efficient APIs
- Designing an easy and interactive UI

1.6 Learning Outcomes

- To better understand how machine learning algorithms work and how they can be implemented in a real-life scenario.
- To get hands-on experience of developing a mobile application from scratch.
- To gain practical experience in building and managing backends and databases for real-life scenarios.
- To understand how the APIs are built for effective communication between front-end and back-end.
- To understand the intricacies of designing a user interface that is user-friendly.

1.7 Nature of End Product

The end product of our project will be a mobile app written in React Native, called “SugarSage” with a ML Algorithm in the backend. The backend will be made in NodeJS and the ML Algorithm will be written in Python connected to NodeJS through APIs. The app will feature a user-friendly interface.

1.8 Completeness Criteria

| Sr. No. | Criteria | Weightage % |
|---------|---------------------------------|-------------|
| 1 | Datasets Gathering/Preparation | 5 |
| 2 | Algorithm Development | 30 |
| 3 | UI Designing | 10 |
| 4 | Mobile App Development | 20 |
| 5 | Server-Side Development | 25 |
| 6 | APIs Development and Connection | 10 |

1.9 Business Goals

The proposed system will be able to provide round-the-clock services to its users for diabetes management. One of the business goals include introducing a subscription model for premium features, including personalized support, and advanced tracking capabilities. That being said, implementing sustainable monetization strategies while ensuring accessibility for all income groups. Another business goal is establishing SugarSage as a sole and leading diabetes management app in Pakistan. We aim to continuously grow the user base by providing exceptional services and user experience. We also aim to prioritize user satisfaction through continuous improvement based on feedback and evolving needs.

1.10 Related Work/ Literature Survey/ Literature Review

1. Papers:

A. Intelligent Nutrition Diet Recommender System for Diabetic's Patients [1]

Objective:

This paper aimed to create an intelligent nutrition diet recommender system specifically tailored for diabetic patients. The focus was on offering individualized meal plans, considering various factors such as medical requirements, personal constraints, and socio-cultural differences.

Findings:

The research concluded that their expert agent system was effective in suggesting individualized food plans for diabetic patients. They employed fuzzy logic to enhance the accuracy of recommendations and incorporated adaptive learning techniques for ongoing improvement.

B. Pakistan's Recommendations for Optimal Management of diabetes from Primary to Tertiary care level [2]

Overview:

The document discusses the updated National Clinical Practice Guidelines for Type-2 diabetes management in Pakistan. It focuses on developing affordable and accessible treatment, establishing a referral system from primary to tertiary care, and forming multidisciplinary teams for comprehensive diabetes management. The guidelines include diagnostic criteria, referral criteria, lifestyle and pharmacological management strategies, and address complications like diabetic foot, obesity, hypertension, and dyslipidemia. These comprehensive guidelines aim to improve diabetes care across Pakistan.

C. Pakistan and diabetes-A country on the edge [3]

Overview:

The paper "Pakistan and diabetes-A country on the edge" focuses on the increasing burden of diabetes in Pakistan. It attributes this rise to lifestyle changes, including sedentary habits and poor diet, leading to obesity. The paper emphasizes the need for a cost-effective, population-based approach for diabetes screening and management, highlighting the results of the National Diabetes Survey of Pakistan.

D. Current management strategies to target the increasing incidence of diabetes within Pakistan [4]

Overview:

"Current management strategies to target the increasing incidence of diabetes within Pakistan" discusses the strategies for managing the growing diabetes incidence in Pakistan. It stresses the importance of developing multidisciplinary teams and standardized protocols for healthcare professionals in diabetes care.

E. Diabetes Prevalence Survey of Pakistan (DPS-PAK): prevalence of type 2 diabetes mellitus and prediabetes using HbA1c: a population-based survey from Pakistan [5]

Overview:

This study conducted a nationwide survey in Pakistan to estimate the prevalence of type 2 diabetes using glycated haemoglobin (HbA1c) as the screening tool. The cross-sectional study involved multistage stratified cluster sampling of individuals aged 20 years and above. It found that the prevalence of prediabetes was 10.91% and type 2 diabetes was 16.98%. The study identified several risk factors for diabetes, including older age, increased BMI, and central obesity. The findings suggest a much higher prevalence of diabetes and prediabetes in Pakistan than previously thought, underscoring the need for comprehensive strategies for screening, prevention, and treatment at the community level.

2. Apps:

Diabetes & Diet Tracker:

Description:

This is a diet tracker application designed for various types of diabetes. It tracks insulin and medication usage, counts carbs and calories, and offers insights into managing diabetes. It also provides tips and reveals factors affecting blood glucose levels.

- Similarities:

- Designed for diabetics.
- Tracks insulin and medication usage.
- Keeps a count of carbs and calories.

- Differences:

- Lacks a meal planner specifically for diabetics.
- Does not have an activity tracker.
- Does not offer a food catalog tailored to local foods.

Eat This Much:

Description:

"Eat This Much" is an app that provides personalized meal plans based on dietary preferences, nutritional targets, and budget. While not exclusively for diabetics, it offers a flexible approach to diet planning.

- Similarities:

- Offers personalized meal plans.

- Differences:

- Not specifically designed for diabetics.
- Broader focus on general nutrition rather than diabetic-specific needs.

DiabTrend:

Description:

"DiabTrend" is an app that helps diabetics monitor and predict blood sugar levels based on various factors, including meals. It provides insights into how different foods and activities impact sugar levels.

- Similarities:

Designed for diabetics.

Provides insights based on meals.

- Differences:

Focuses more on monitoring and predicting blood sugar levels rather than providing meal recommendations.

3. Datasets:

A. Food Exchange Lists:

Compiled By: American Dietetic Association and American Diabetes Association, 2008

Description:

The "Food Exchange List" document provides a comprehensive categorization of foods and their nutritional content, designed to facilitate dietary planning and tracking. It's segmented into groups like starches, fruits, vegetables, sweets, meats, fats, and dairy, with specified serving sizes and corresponding amounts of carbohydrates, protein, fat, and calories. The list enables users to manage dietary needs by detailing the nutritional exchange for various food items, which is particularly beneficial for individuals counting macronutrients or managing conditions like diabetes. The exchange lists are drawn from the American Dietetic Association and the American Diabetes Association guidelines.

B. Frida's Food Dataset:

Compiled By: Technical University of Denmark DTU

Description:

The dataset provides a detailed nutritional analysis of various foods, including energy content, macronutrients such as proteins, carbohydrates, and fats, and a broad spectrum of micronutrients like vitamins and minerals. It also includes the composition of fatty acids and the presence of organic acids, which are critical for dietary assessments and nutrition-focused research projects. This extensive dataset serves as an essential tool for health professionals, dietitians, and researchers to evaluate and plan nutritionally balanced diets and study the health impacts of specific food items.

2. Overall Description

2.1 Product Features

The "SugarSage - AI Companion for Diabetics" project aims to address the challenges of diabetes management in Pakistan considering user's local food preferences. It offers the following key features: Product Features:

1. **Personalized Diet Planning:** The system includes an AI algorithm capable of generating personalized diet plans for diabetics, taking into account individual health metrics, medication regimens, and activity levels.
2. **Local Food Database:** The system integrates a comprehensive food database that considers local Pakistani dietary data, ensuring that dietary recommendations align with cultural norms, making it more feasible for users to follow.
3. **Mobile Application:** The end product is a user-friendly mobile application that serves as a digital dietitian for diabetics. Users can access and adhere to their personalized dietary recommendations through this intuitive platform.
4. **User Experience:** The project focuses on designing a user-friendly interface that caters to a diverse user base, from tech-savvy individuals to those less familiar with mobile applications, ensuring ease of use and accessibility.
5. **Activity Tracking:** Integration manual entry of physical activity can help users monitor their exercise routines (Daily steps, sleep schedule). The app can adjust meal plans based on activity levels to help maintain blood sugar levels.
6. **Health Reports:** Users can generate health reports summarizing their progress and adherence to the dietary recommendations. These reports can be shared with healthcare providers for better-informed care.

2.2 User Classes and Characteristics

1. **Diabetic Individuals:** Diabetic individuals are the primary users of the product. They vary in terms of age, gender, and cultural backgrounds. They may have different levels of technical expertise, ranging from tech-savvy individuals to those less familiar with mobile applications.

Usage:

- Diabetic individuals use the mobile application to access personalized diet plans and manage their diabetes diet.
- They input their health metrics, medication information, and activity levels.
- They follow the recommended meal plans, track their progress, and receive guidance and feedback.
- They may log their meals, monitor their physical activity, and record blood glucose levels using the app.

2. Technical Support Team:

Technical support specialists with expertise in the product.

Usage:

- Users who encounter technical issues may contact the technical support team for assistance.
- The support team helps troubleshoot and resolve technical problems.

2.3 Operating Environment

1. Hardware Platform: The mobile application is designed to operate on a variety of hardware platforms, primarily targeting smartphones and tablets. This includes both Android and iOS devices.

2. Operating System and Versions:

- For Android devices, the application should be compatible with recent versions of the Android operating system.
- For iOS devices, compatibility should extend to recent versions of the iOS operating system.

3. The primary software components:

- **Mobile Application:** The "SugarSage" mobile application itself, which houses the user interface, AI algorithms, and data storage.
- **AI and ML Libraries:** The application relies on AI and ML libraries, which may be integrated into the app to provide personalized dietary recommendations.
- **Database Management System:** A database system is used to store user data, including health metrics, meal plans, and user profiles.
- **Backend Server:** If required, a backend server may be used to manage data synchronization and communication between the mobile application and the database.
- **Admin Panel:** An admin panel hosted on a web server to monitor and manage the application's performance and user data. The admin panel provides insights, analytics, and administrative controls for efficient oversight of the application.

2.4 Design and Implementation Constraints

- **Hardware Limitations:** Mobile devices may have limited processing power, memory, and storage capacity. The application must be designed to operate efficiently within these constraints to ensure optimal performance.

- **Compatibility with Mobile Operating Systems:**
The app must be compatible with various versions of Android and iOS. Ensuring compatibility and optimizing performance across different devices and OS versions can be a constraint.
- **Local Food Database Integration:**
Incorporating local dietary data may be challenging due to the availability and accuracy of such data. Efforts to obtain and maintain this data can be a constraint.
- **AI and ML Model Development:**
Developing and fine-tuning AI and ML algorithms to generate personalized diet plans may require access to large and diverse datasets, which could be a constraint if such data is limited or hard to acquire.

2.5 Assumptions and Dependencies

Assumptions:

1. **Data Availability:** It is assumed that relevant dietary and nutritional data, including local food databases, will be available or obtainable for integration into the application.
2. **User Adoption:** The project assumes that diabetic individuals in Pakistan will adopt and use the mobile application effectively for managing their condition.

Dependencies:

1. **Mobile Operating Systems:** The application is dependent on the stability and compatibility of Android and iOS operating systems, as updates or changes can impact the app's performance.
2. **AI and ML Models:** The development of AI and ML algorithms for personalized diet planning depends on access to suitable datasets and resources for model training and fine-tuning.
3. **User Acceptance:** The project's success is dependent on user acceptance and engagement. Ensuring that the app aligns with user needs and preferences is critical.

3. Functional Requirements

3.1 Patient Use Cases

3.1.1 Name of Use-Case 1: Signup

| | | |
|----------------------------|---|---|
| Identifier | UC-1 | |
| Purpose | User will be able to create an account. | |
| Priority | High | |
| Pre-conditions | <div>1. User should not have an account in the system.</div> <div>2. Internet connectivity required.</div> | |
| Post-conditions | <div>Minimal guarantees:<div>1. Registration failed due to invalid input.</div><div>2. User already exists.</div></div> <div>Success guarantees:<div>1. User successfully registered.</div></div> | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The user clicks on the "Sign up" option on the startup screen of the application. | The system will redirect to the signup form. |
| 2 | The user enters email, password, name, and medical information (BMI, blood sugar levels etc.) on the registration form and clicks the Signup button. | The system will authenticate the user credentials and redirect to the home screen of the application. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The user enters email, password, medical information, etc., and clicks the Signup button. | <div>The system detects that the user credentials are invalid and displays an error message.</div> <div>Example:<div>1. "Your Password doesn't contain any special characters."</div><div>2. "Your BMI input is invalid."</div><div>3. "Email doesn't match the format."</div></div> |

| | | |
|----------|--|--|
| 2 | User enters email, password, medical information etc. and clicks on the Signup button. | System detects that the user already exists in the system and displays the error message "User already exists." |
|----------|--|--|

Table 1: UC-1

3.1.2 Name of Use-Case 2: Login

| | | |
|----------------------------|---|---|
| Identifier | UC-2 | |
| Purpose | User will be able to access their account. | |
| Priority | High | |
| Pre-conditions | 1. User should be registered in the system. 2. Internet connectivity required. | |
| Post-conditions | Minimal guarantees: 1. The user account does not exist in the system. Success guarantee: 1. The user successfully logged in. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | User clicks on "Login" option on the startup screen of the application. | The system will redirect you to the login form. |
| 2 | The user enters the email and password on the login form and clicks the Login button. | The system will verify user credentials and redirect to the home screen of the application. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The user enters the email and password on the login form and clicks the Login button. | The system detects that the user doesn't exist in the database and displays the error message "User doesn't exist." |

Table 2: UC-2

3.1.3 Name of Use-Case 3: Get Meal Plan

| | | |
|----------------------------|---|---|
| Identifier | UC-3 | |
| Purpose | To generate personalized meal plans, delete meal plans, and share meal plans. | |
| Priority | High | |
| Pre-conditions | 1. User should be logged in on the system 2. User medical information should be known. 3. Internet connectivity required. | |
| Post-conditions | System will generate a meal plan according to user preferences | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The user clicks the "Get Meal Plan" button on the Meal Plan icon. | The system will redirect to the meal form. |
| 2 | User will enter their blood sugar levels, food preferences, food recommended by their doctor, and duration of meal plan on the meal form. | The system will use the user's food preferences and medical information to create a personalized meal plan. |
| 3 | User clicks on the "Edit Meal Plan" button and choose the particular item they want to substitute with a different one. | The system updates and displays a customized meal plan based on previous inputs and new modifications. |
| 4 | The user clicks on the "Delete Meal Plan" button. | The system deletes the previous meal plan and redirects to the Meal Plan tab. |
| 5 | The user clicks on the "Share Meal Plan" button. | The system displays a list of applications that can be used to share the meal plan image. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | N/A | N/A |

Table 3: UC-3

3.1.4 Name of Use-Case 4: User Profile

| | | |
|----------------------------|---|--|
| Identifier | UC-4 | |
| Purpose | Allow users to view and edit their personal information, as well as check their achievements and give feedback. | |
| Priority | Medium | |
| Pre-conditions | 1. User should be logged in on the system. 2. Internet connectivity required. | |
| Post-conditions | User profile is successfully viewed and edited. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The user clicks on the "User Profile" icon on the application. | The system will display personal information and a few achievements of the user and the option to log out. |
| 2 | | The system also shows options to edit personal information and view more achievements. |
| 3 | The user clicks on the edit icon on the Personal Information. | The system will allow the user to edit their personal information. |
| 4 | The user edits their personal information and saves it. | The system will verify and update the edited information on the database. |
| | | The system now displays the newly updated personal information. |
| 5 | The user clicks on "View More Achievements". | The system will redirect to another page and display the user's achievement in detail. |
| 6 | The user clicks on the "Give Feedback" option | The system will redirect to the feedback form page. |
| 7 | The user writes the feedback and click the submit button. | The system will store the feedback in the database and redirect to the User Profile tab. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The user edits their personal information and saves it. | The system detects that the edited information is incorrect and displays an error message. |

Table 4: UC-4

3.1.5 Name of Use-Case 5: Health Profile

| | | |
|----------------------------|--|--|
| Identifier | UC-5 | |
| Purpose | Users can update their health status. | |
| Priority | Medium | |
| Pre-conditions | <div>1. User should be logged in on the system</div> <div>2. Internet connectivity required.</div> <div>3. User gives permissions to access device sensors and locations etc.</div> | |
| Post-conditions | <div>Minimal guarantees:</div> <div>1. Invalid input values for Insulin levels, HbA1c score, and blood sugar levels.</div> <div>Success guarantee:</div> <div>1. Successfully updated Insulin levels, HbA1c score, and blood sugar levels.</div> <div>2. Successfully viewed health information.</div> | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The user clicks on "Health Profile" icon on the application. | The system will display health information like sleep tracking, physical activity (steps), HbA1c score, blood sugar, and insulin levels. |
| 2 | | The system shows option view sleep patterns, physical activity track, blood sugar, and insulins levels in detail. |
| 3 | | The system also shows the option to update HbA1c score, blood sugar, and insulin levels. |
| 4 | The user clicks on update icon and enters updates values for their HbA1c score, blood sugar, or insulin levels. | The system verifies the values and updates them in the database. |
| 5 | | The system now displays the newly updated information. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The user clicks on update icon and enters updates values for their HbA1c score, blood sugar, or insulin levels. | The system detects the incorrect values for HbA1c score, blood sugar, or insulins levels and displays an error message. |

Table 5: UC-5

3.1.6 Name of Use-Case 6: Logout

| | | |
|----------------------------|---|--|
| Identifier | UC-6 | |
| Purpose | Users can log out of the system | |
| Priority | Low | |
| Pre-conditions | 1. User should be logged in on the system 2. Internet connectivity required. | |
| Post-conditions | User has successfully logged out. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The user clicks on “User Profile” icon on the application. | The system will display other information along with the option to logout. |
| 2 | The user clicks on the logout option. | The system displays the dialogue box with message” Are you sure you want to log out from SugarSage?” |
| 3 | The user clicks on “Yes” button. | The system ends the session and redirects to the startup screen. |
| | | Go to UC-1 or UC-2 |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | N/A | N/A |

Table 6: UC-6

3.2 Admin Use Cases

3.2.1 Name of Use-Case 7: Sign up

| | | |
|----------------------------|---|---|
| Identifier | UC-7 | |
| Purpose | Admin will be able to create an account. | |
| Priority | High | |
| Pre-conditions | <div>1. User should not have an account in the system.</div> <div>2. Internet connectivity required.</div> | |
| Post-conditions | <div>Minimal guarantees:<div>1. Registration failed due to invalid input.</div><div>2. Admin already exists.</div></div> <div>Success guarantees:<div>1. Admin successfully registered.</div></div> | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on the "Sign up" option on the startup screen of the web portal. | The system will redirect to the registration form. |
| 2 | The admin enters email and password, name, phone number, etc. on the registration form and clicks the Signup button. | The system will verify the admin credentials and redirect to the dashboard page. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin enters email, password, name, phone number etc. and clicks the Signup button. | <div>The system detects that the admin credentials are invalid and displays an error message.</div> <div>Example:<div>1. "Your email address doesn't exist."</div><div>2. "Your Password doesn't contain any special characters."</div><div>3. "Your Phone Number is invalid."</div></div> |
| 2 | Admin enters email, password, name, phone number etc. and clicks on the Signup button. | <div>System detects that the admin already exists in the system and displays the error message</div> <div>"Admin already exists in the system."</div> |

Table 7: UC-7

3.2.2 Name of Use-Case 8: Admin Login

| | | |
|----------------------------|---|--|
| Identifier | UC-8 | |
| Purpose | Admin will be able to access their account. | |
| Priority | High | |
| Pre-conditions | 1. Admin should be registered in the system. 2. Internet connectivity required. | |
| Post-conditions | Minimal guarantees: 1. The admin account does not exist in the system. Success guarantee: 1. The admin successfully logged in. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on "Login" option on the startup screen of the web portal. | The system will open the login form. |
| 2 | The admin enters the email and password on the login form and clicks the Login button. | The system will authenticate the admin's email and password, and upon successful authentication, redirect to the dashboard page. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin enters the email and password on the login form and clicks the Login button. | The system detects that the admin does not exist in the database and displays the error message "Admin doesn't exist." |

Table 8: UC-8

3.2.3 Name of Use-Case 9: Dashboard

| | | |
|----------------------------|---|--|
| Identifier | UC-9 | |
| Purpose | Admin will be able to view, add, update, delete food from the database. | |
| Priority | High | |
| Pre-conditions | 1. Admin should be registered in the system. 2. Internet connectivity required. | |
| Post-conditions | Minimal guarantees: 1. The admin account does not exist in the system. Success guarantee: 2. The admin successfully logged in. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on "Login" option on the startup screen of the web portal. | The system will open the login form. |
| 2 | The admin enters the email and password on the login form and clicks the Login button. | The system will authenticate the admin's email and password, and upon successful authentication, redirect them to the home screen of the web portal. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin enters the email and password on the login form and clicks the Login button. | The system detects that the admin does not exist in the database and displays the error message "Admin doesn't exist." |

Table 9: UC-9

3.2.4 Name of Use-Case 10: Food Item Management

| | | |
|----------------------------|--|--|
| Identifier | UC-10 | |
| Purpose | Admin will be able to view, add, update, or delete food items. | |
| Priority | Medium | |
| Pre-conditions | 1. Admin should be logged in the system. 2. Internet connectivity required. | |
| Post-conditions | Minimal guarantees: 1. Invalid food input. Success guarantee: 1. Food Items displayed successfully. 2. Food Item updated successfully. 3. Food Item added successfully. 4. Food Item deleted successfully. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on "Foods" icon. | The system will redirect to the food page displaying all the foods along with the option to add, update, delete foods. |
| 2 | The admin clicks on "Add Food" button. | The system will display a form containing all the fields needed for a food to be added. |
| 3 | The admin fills all the fields (protein, carbs, glycemic index etc.) on the "Add Food" form and clicks the save button. | The system will verify the field values and add the food to the database and display the updated food list. |
| 4 | The admin clicks on the "Update Food" button. | The system will display a form containing all the fields of the food which can be edited. |
| 5 | The admin updates the fields of the food and clicks the save button. | The system will verify the field values and update the fields in the database and display the updated food list. |
| 6 | The admin clicks on the "Delete Food" button. | The system will ask the user "Are you sure you want to delete this Food Item?". |
| 7 | The admins click the "Yes, delete the Food Item". | The system will delete the Food Item from the database and display the updated food list. |
| 8 | The admins click the "No, don't delete the Food Item". | The system will redirect to the food list page. |
| Alternate Course of Action | | |

| S# | Actor Action | System Response |
|----|---|--|
| 1 | The admin fills all the fields on the "Add Food" form and clicks the save button. | The system detects that the provided food's fields are invalid and display an error message Example: 1. "Invalid Food macro-nutrients." 2. Invalid Food glycemic index. |
| 2 | The admin updates the fields of the food and clicks the save button. | The system detects that the edited food fields are invalid and displays an error message. |

Table 10: UC-10

3.2.4 Name of Use-Case 11: Dish Management

| | | |
|--------------------------|---|---|
| Identifier | UC-11 | |
| Purpose | Admin will be able to view, add, update, or delete dishes. | |
| Priority | Medium | |
| Pre-conditions | 1. Admin should be logged in the system. 2. Internet connectivity required. | |
| Post-conditions | Minimal guarantees: 1. Invalid dish input. Success guarantee: 1. Dishes displayed successfully. 2. Dishes updated successfully. 3. Dishes added successfully. 4. Dishes deleted successfully. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on "Dishes" icon. | The system will redirect to a page containing list of dishes along with the option to add, update, delete dishes. |
| 2 | The admin clicks on "Add Dish" button. | The system will display a form containing all the fields needed for a dish to be added. |
| 3 | The admin fills all the fields (ingredients, quantity etc.) on the "Add Dish" form and clicks | The system will verify the field values and add the dish to the database and display an |

| | | |
|-----------------------------------|---|---|
| | the save button. | updated list of dishes. |
| 4 | The admin clicks on the "Update Dish" button. | The system will display a form containing all the fields of the dish which can be edited. |
| 5 | The admin updates the fields of the dish and clicks the save button. | The system will verify the field values and update the fields in the database and display an updated list of dishes. |
| 6 | The admin clicks on the "Delete Dish" button. | The system will display a dialogue box with the message "Are you sure you want to delete this Dish?". |
| 7 | The admins click the "Yes, delete the selected Dish". | The system will delete the Food Item from the database and display an updated list of dishes. |
| 8 | The admins click the "No, don't delete the selected Dish". | The system will redirect to the dish list page. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin fills all the fields on the "Add Dish" form and clicks the save button. | The system detects that the provided Dish's fields are invalid and display an error message Example: 1. "Invalid Dish ingredients count." 2. "Invalid Dish ingredients." |
| 2 | The admin updates the fields of the dish and clicks the save button. | The system detects that the edited dish fields are invalid and displays an error message. |

Table 11: UC-11

3.2.5 Name of Use-Case 12: User Account Management

| | | |
|----------------------------|--|---|
| Identifier | UC-12 | |
| Purpose | Admin will be able to view, update, and delete user's accounts. | |
| Priority | Medium | |
| Pre-conditions | 1. Admin should be logged into the system. 2. Internet connectivity required. | |
| Post-conditions | 1. User list displayed successfully. 2. User deleted successfully. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on "User Accounts" icon. | The system will redirect to the user's account page and display all the users along with the option to update and delete users. |
| 2 | The admin clicks on "Update User" button. | The system will display a form containing all the fields of a user and allows the permission to change them. |
| 3 | The admin updates the user account details and clicks on the save button. | The system will verify the changes and update the corresponding fields in the database. |
| 4 | The admin clicks on "Delete User" button. | The system will display a dialogue box with the message "Are you sure you want to delete this user?" |
| 5 | The admin clicks on "Yes, delete the user" button. | The system will delete the user from the database and display an updated list of users. |
| 6 | The admin clicks on "No, don't delete the user" button. | The system will redirect to the user's account page. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin updates the user account details and clicks on the save button. | The system detects that the updated user account details are invalid and displays an error message. |

Table 12: UC-12

3.2.6 Name of Use-Case 13: Dashboard

| | | |
|----------------------------|--|---|
| Identifier | UC-13 | |
| Purpose | Admin will be able to view application analytics. | |
| Priority | Low | |
| Pre-conditions | 1. Admin should be logged into the system. 2. Internet connectivity required. | |
| Post-conditions | Display analytical graphs and charts according to the application usage. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on “Dashboard” icon. | The system will redirect to the dashboard page and display analytics regarding user traffic, user complaints etc. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | N/A | N/A |

Table 13: UC-13

3.2.6 Name of Use-Case 14: Feedback Management

| | | |
|--------------------------|--|--|
| Identifier | UC-14 | |
| Purpose | Admin will be able to view and delete the feedbacks. | |
| Priority | Medium | |
| Pre-conditions | 1. Admin should be logged into the system. 3. Internet connectivity required. | |
| Post-conditions | 1. Feedbacks displayed successfully. 2. Feedbacks deleted successfully. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on "Feedback" icon. | The system will redirect to a page displaying a list of feedbacks. |
| 2 | The admin clicks on "Delete Feedback" button. | The system will display a dialogue-box containing the message "Are you sure you want to delete this feedback?" |

| | | |
|-----------------------------------|---|--|
| 3 | The admin clicks on "Yes, delete this feedback." | The system will delete the corresponding feedback from the database and display the updated list of feedbacks. |
| 4 | The admin clicks on "No, don't delete this feedback." | The system will redirect to the page displaying a list of feedbacks. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | N/A | N/A |

Table 14: UC-14

3.2.6 Name of Use-Case 15: Dashboard

| | | |
|----------------------------|--|---|
| Identifier | UC-15 | |
| Purpose | Admin will be able to view application analytics. | |
| Priority | Medium | |
| Pre-conditions | 1. Admin should be logged into the system. 2. Internet connectivity required. | |
| Post-conditions | Display analytical graphs and charts according to the application usage. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on “Dashboard” icon. | The system will redirect to the dashboard page and display analytics regarding user traffic, user complaints etc. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | N/A | N/A |

Table 15: UC-15

3.2.7 Name of Use-Case 16: Admin Profile Management

| | | |
|----------------------------|--|--|
| Identifier | UC-16 | |
| Purpose | Admin will be able to view and update their personal information | |
| Priority | Low | |
| Pre-conditions | 1. Admin should be logged into the system. 2. Internet connectivity required. | |
| Post-conditions | Displays and updates admin’s information. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on “Profile” icon. | The system will display a drop down containing an option “View Profile”. |
| 2 | The admin clicks on “View Profile” button. | The system redirects to a page containing the admin’s personal information and an option for editing the information. |
| 3 | The admin clicks on “Edit Profile” button and updates their profile information. | The system will verify and then update the admin’s personal information in the database and display the updated information. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on “Edit Profile” button and updates their profile information. | The system will detect that there is an invalid value in a field and displays an error message. |

Table 16: UC-16

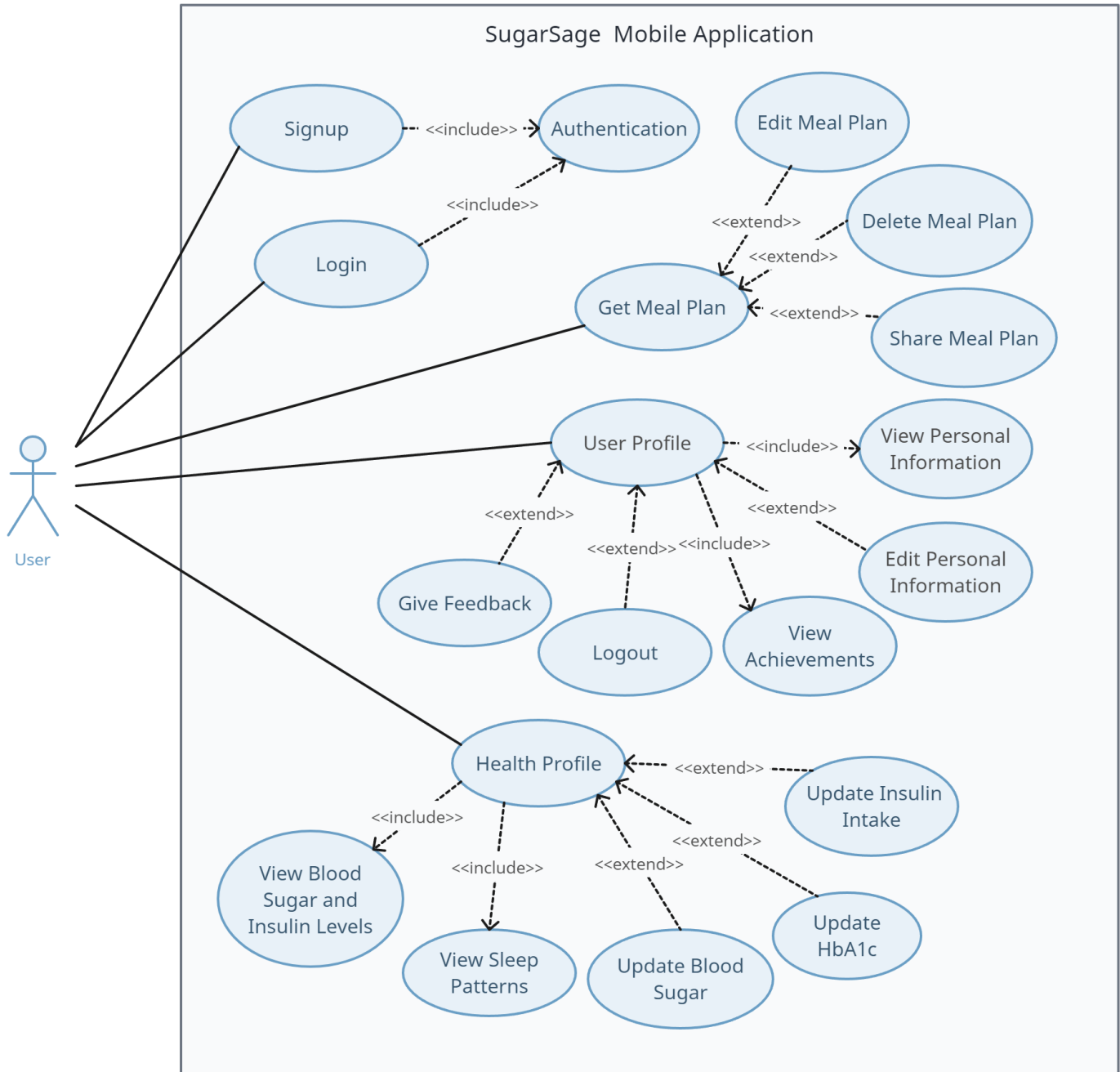
3.2.8 Name of Use-Case 17: Logout

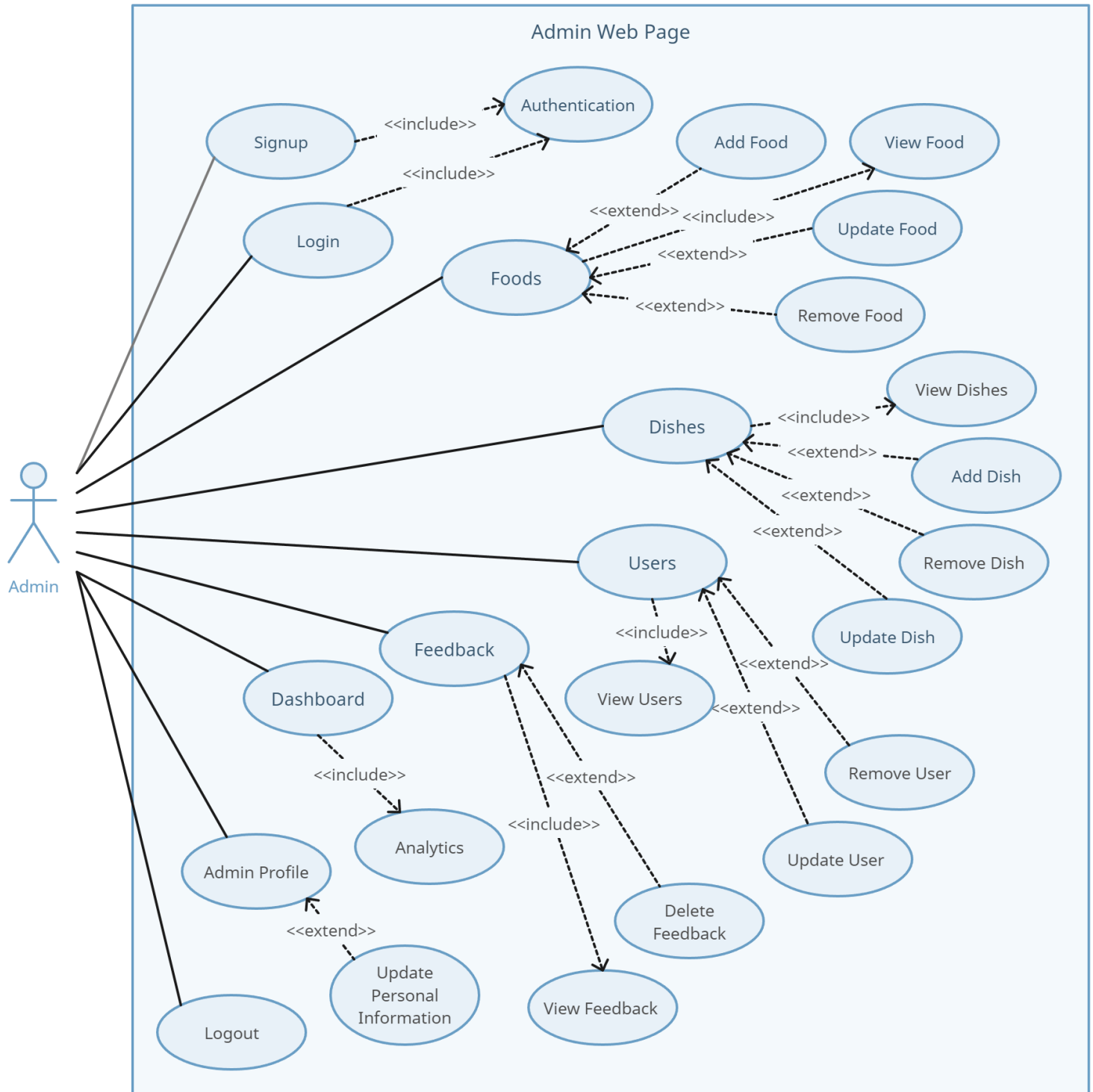
| | | |
|----------------------------|--|--|
| Identifier | UC-17 | |
| Purpose | Admin will be able to log out of the system. | |
| Priority | Low | |
| Pre-conditions | 1. Admin should be logged into the system. 2. Internet connectivity required. | |
| Post-conditions | Admin will be logged out of the system. | |
| Typical Course of Action | | |
| S# | Actor Action | System Response |
| 1 | The admin clicks on “Profile” icon on top right corner. | The system will display a drop down containing an option for logout. |
| 2 | The admin clicks on the “Logout” button. | The system will redirect to the startup screen containing the login and sign-up options. |
| Alternate Course of Action | | |
| S# | Actor Action | System Response |
| 1 | N/A | N/A |

Table 17: UC-17

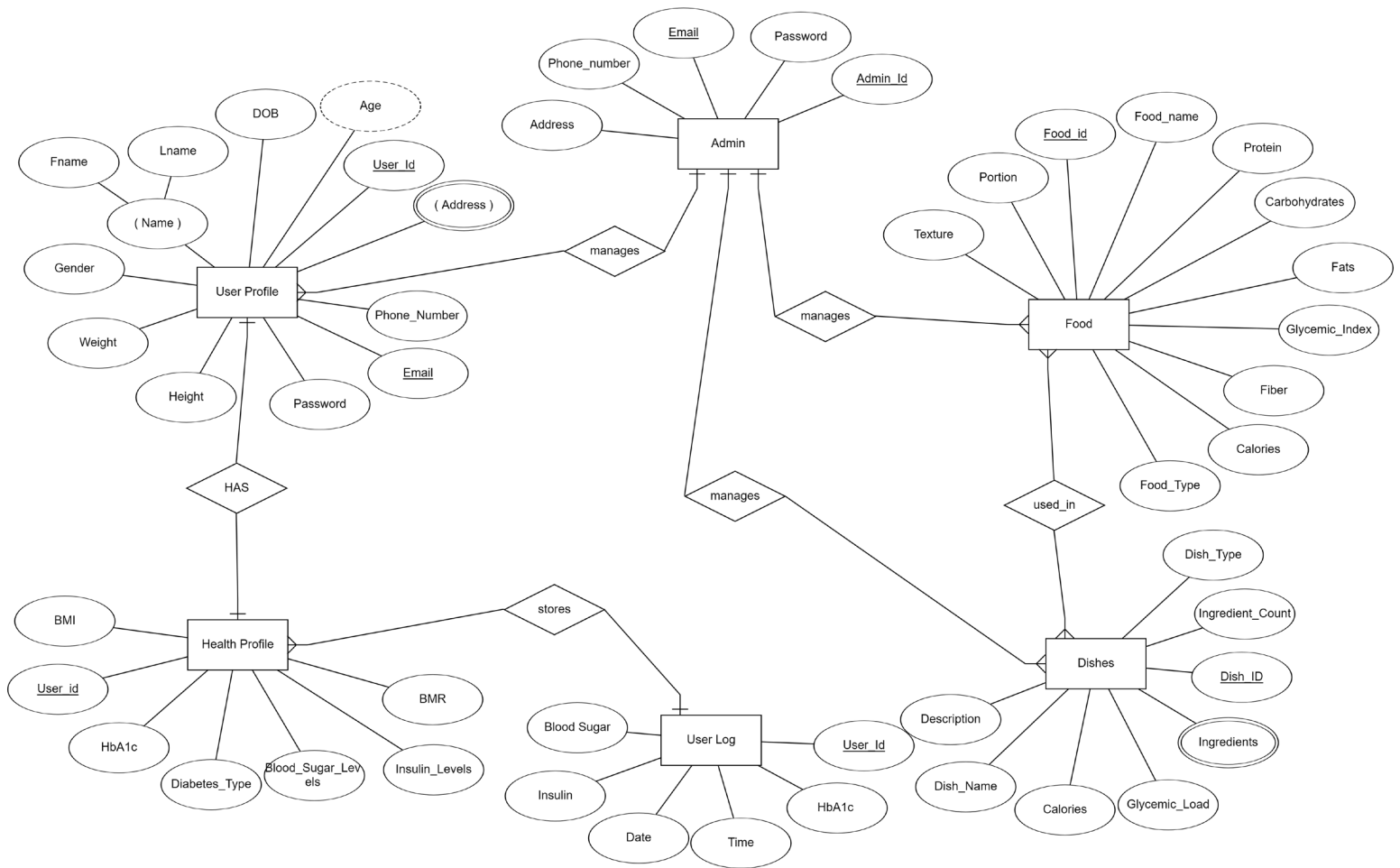
3.3 Requirements Analysis and Modeling

3.3.1 Use Case Diagram

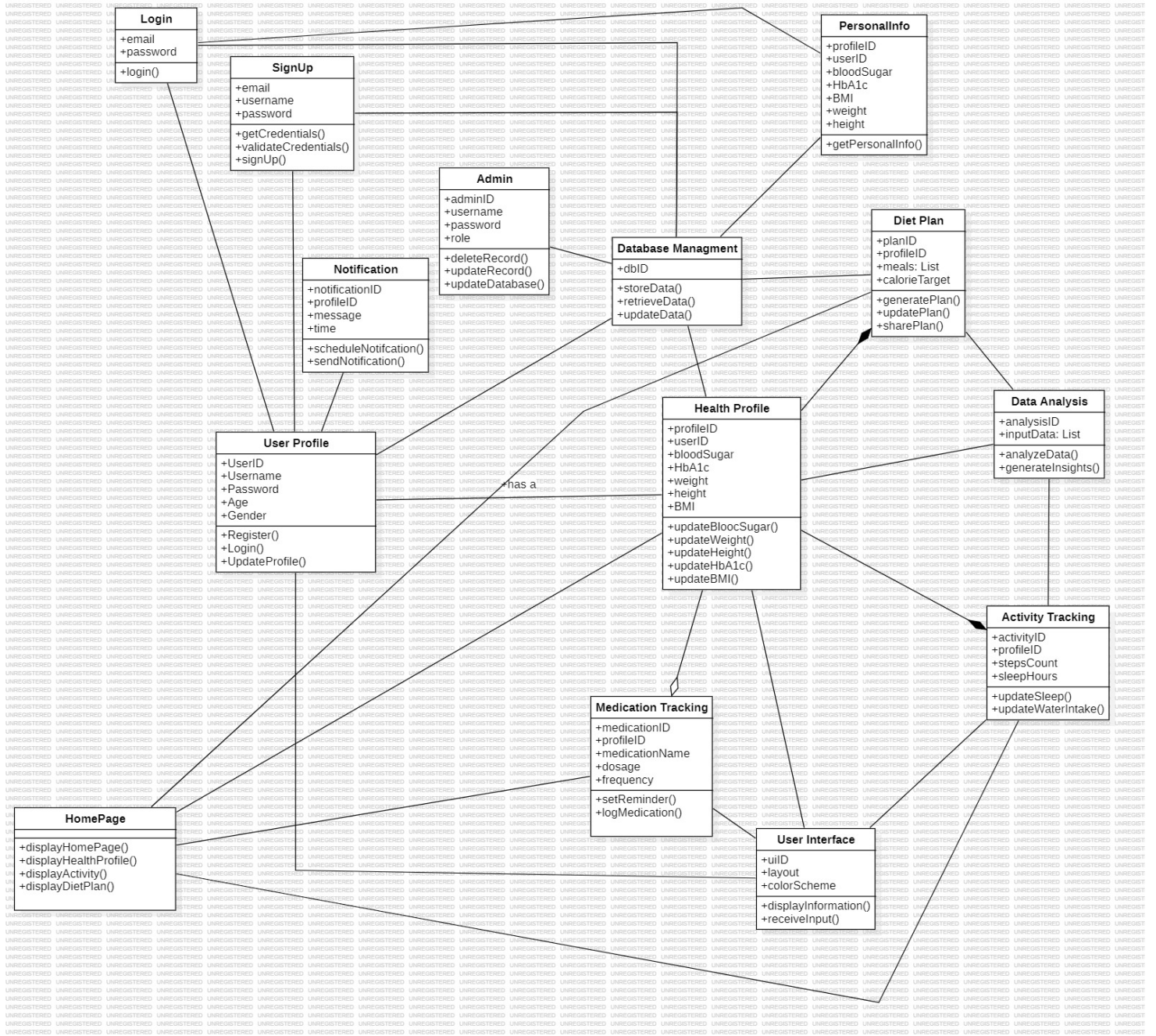




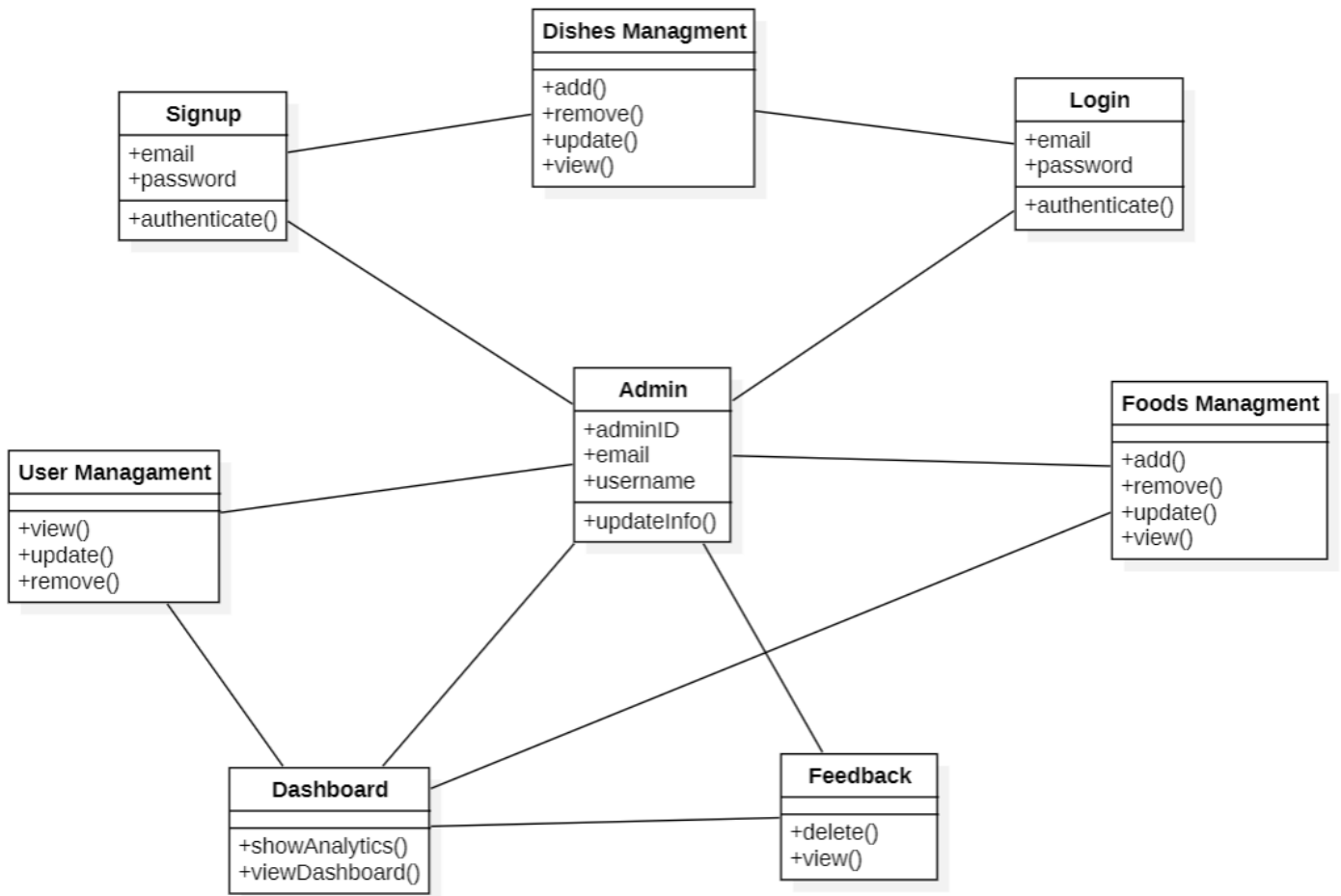
3.3.2 ER Diagram



3.3.3 Abstract Class Diagram

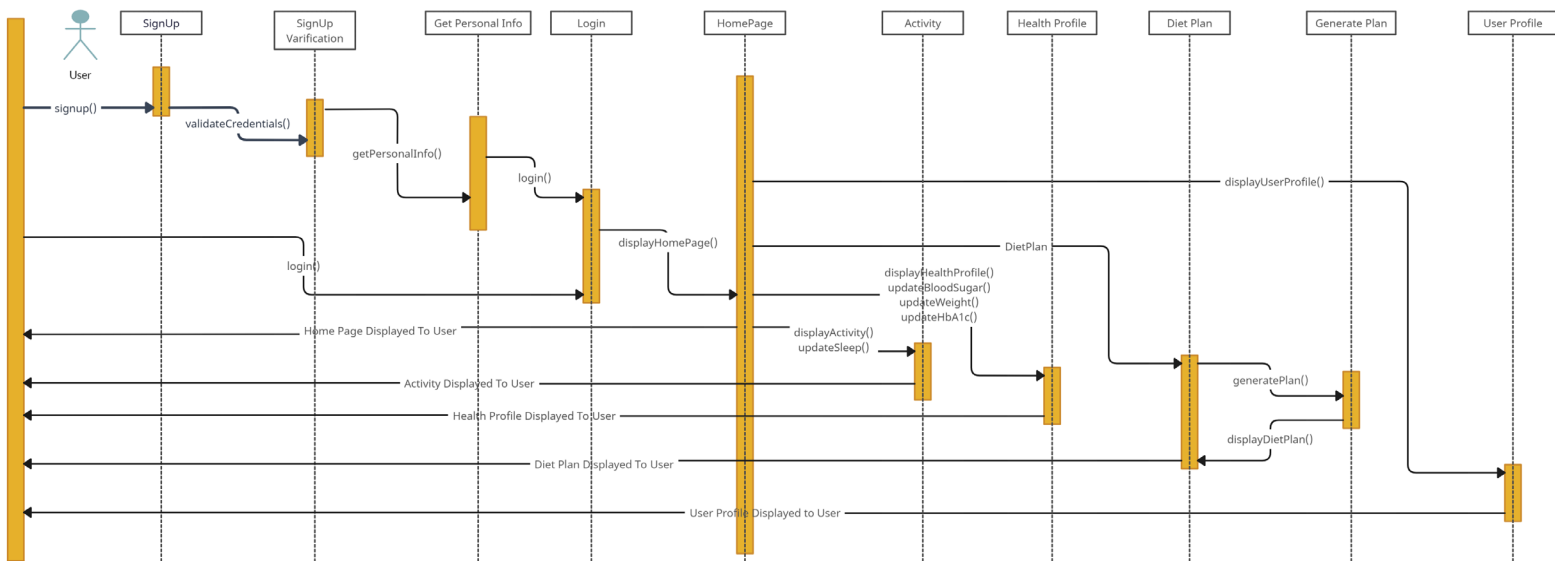


User Abstract Class Diagram

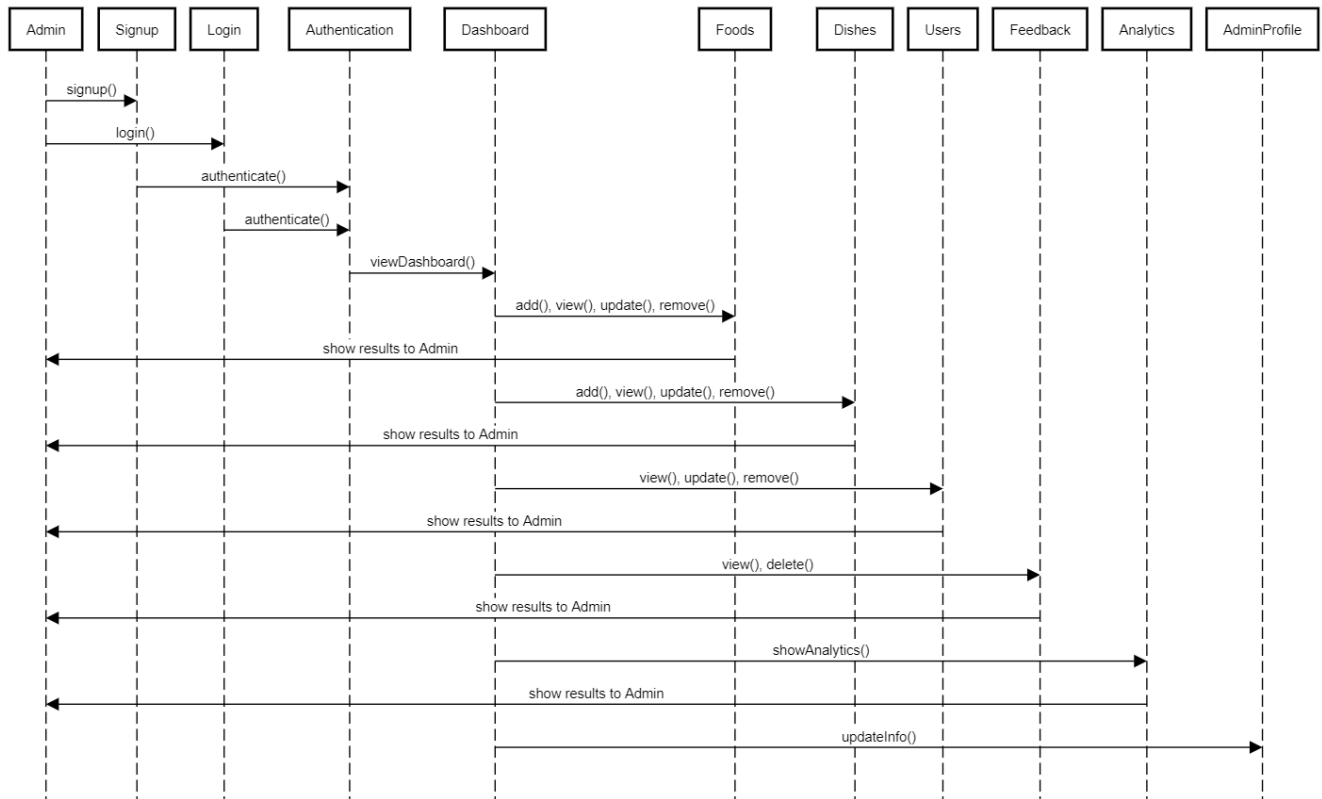


Admin Abstract Class Diagram

3.3.4 Sequence Diagram



User Sequence Diagram



Admin Sequence Diagram

4. Nonfunctional Requirements

4.1 Performance Requirements

- **Response Time:** The mobile application should provide near-instantaneous response times when generating personalized diet plans and navigating through the interface. Response time should be within 1-2 seconds for typical user interactions.
- **Scalability:** The system should be able to handle a large number of users simultaneously without significant degradation in performance. It should be capable of handling at least 10,000 concurrent users.
- **Data Processing Time:** The AI algorithm should process user data and generate personalized diet plans within a reasonable time frame.
- **Offline Capability:** The application should have the ability to provide basic functionality even in offline mode, allowing users to view previously generated diet plans and access educational resources.

4.2 Safety Requirements

- **Data Privacy:** The application must adhere to strict data privacy and protection standards. It should encrypt sensitive user information and comply with relevant data protection regulations.
- **Emergency Information:** The application should provide access to emergency contact information and guidelines for users experiencing critical health situations related to their diabetes.
- **Incorrect use of application:** In case of incorrect use of SugarSage, the application will provide warnings and disclaimers to warn the user that the application is not a doctor and should not be used for medical advice or medical assistance. These warnings will be presented as Terms and Conditions to the user at the time of sign in.

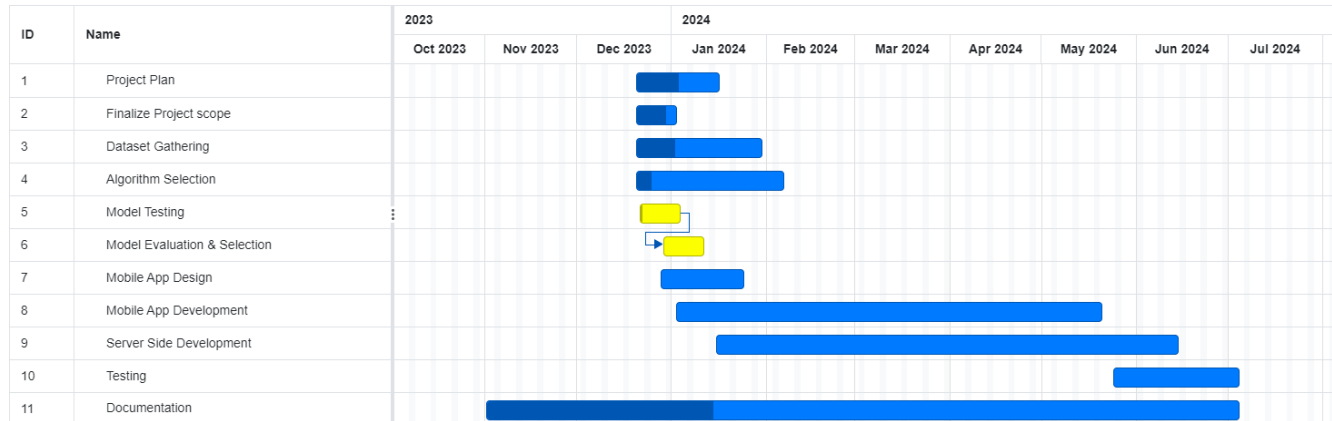
4.3 Security Requirements

- **User Authentication:** The application must implement robust user authentication mechanisms to ensure that only authorized users can access their personalized diet plans and health information.
- **Regular Security Audits:** The system should undergo periodic security audits and vulnerability assessments to identify and address potential security threats and vulnerabilities.

4.4 Additional Software Quality Attributes

- **Usability:** The application should have an intuitive and user-friendly interface, catering to users with varying levels of technological proficiency. It should be rated highly in user experience assessments.
- **Maintainability:** The codebase should be well-documented and organized, allowing for easy maintenance and updates by developers in the future.
- **Reliability:** The system should be highly reliable, with minimal downtime or disruptions in service. It should aim for an uptime of 99.9%.
- **Adaptability:** The application should be capable of adapting to evolving dietary guidelines, medical recommendations, and technological advancements in the field of diabetes management.
- **Portability:** The application should be designed for cross-platform compatibility, ensuring that it can run seamlessly on various devices and operating systems, including iOS and Android. It should offer a consistent user experience across different platforms, taking into consideration diverse screen sizes and resolutions.

5. Revised Project Plan



6. References

- [1] N. Tabassum, A. Rehman, M. Hamid, M. Saleem, S. Malik and Tahir Alyas, "Intelligent Nutrition Diet Recommender System for Diabetic's Patients," Intelligent Automation & Soft Computing, vol. 30, no. 1, p. 18,2021.
- [2] Shera AS, Basit A, PROMPT Team. Pakistan's Recommendations for Optimal Management of Diabetes from Primary to Tertiary care level (PROMPT). Pak J Med Sci. 2017;33(5):1279-1283. doi: <https://doi.org/10.12669/pjms.335.13665>
- [3] Basit A, Fawwad A, Siddiqui SA, Baqa K. "Pakistan and diabetes-A country on the edge." Diabetes Research and Clinical Practice. 2018; DOI: 10.1016/j.diabres.2018.11.001. [PubMed: 30423349]
- [4] Basit A, Fawwad A, Siddiqui SA, Baqa K. "Current management strategies to target the increasing incidence of diabetes within Pakistan." Diabetes Metab Syndr Obes. 2019;12:85-96. DOI: 10.2147/DMSO.S141356. [PubMed: 306434]
- [5] Aamir AH, Ul-Haq Z, Mahar SA, et al. Diabetes Prevalence Survey of Pakistan (DPS-PAK): prevalence of type 2 diabetes mellitus and prediabetes using HbA1c: a population-based survey from Pakistan. BMJ Open 2019;9:e025300. doi:10.1136/bmjopen-2018-025300