BSCS FINAL PROJECT
Design and Test Specification

SugarSage – AI Companion For Diabetics

Project Advisor

Dr. Imran Arshad Choudhry

Presented by:
Group ID: F23CS014

Student Reg#	**Student Name**
L1F20BSCS0204	Khizar Iqbal
L1F20BSCS0207	Ahmed Ali
L1F20BSCS0466	Hamail Shahbaz
L1F20BSCS0467	Safoora Masood

Faculty of Information Technology

University of Central Punjab

University of Central Punjab

Design and Test Specification

SDP Phase III

SugarSage – AI Companion For Diabetics

Advisor: Dr. Imran Arshad Choudhry

Team F23CS014

Member Name	Primary Responsibility
Khizar Iqbal	Introduction, References
Ahmed Ali	Technical Architecture, Test Specifications
Hamail Shahbaz	Component Design
Safoora Masood	Screenshots/Prototype, Project Plan

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Abstract

This project introduces SugarSage, a mobile application powered by AI, specifically designed to address the complex dietary management needs of diabetic patients in Pakistan. The problem of diabetes is significant in Pakistan, 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 utilizes machine learning techniques to analyze various factors influencing dietary choices and suggests optimal diet plans. Additionally, it allows 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

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 consider 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

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

- The main objective is to provide a better solution to Pakistani diabetes patients for 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.

1.4 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.

1.5 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.

1.6 Document Conventions

1. **Section Numbering:**
   The document uses a hierarchical numbering system for sections, sub-sections, and sub-sub-sections (e.g., 1., 1.1, 1.1.1).
2. **Font Styles:**
   - Italics: Italics are used for emphasis, titles of books, and when introducing new terms or concepts for the first time.
   - Bold: Bold text is used for headings, subheadings, and to highlight important points.
3. **Lists:**
   - Bullet Points: Bullet points are used for listing items within a section.
   - Numbered Lists: Numbered lists are used when presenting a sequence of steps or items.
4. **Tables:**
   Tables are employed for organizing and presenting structured data, such as use case descriptions and functional requirements.
5. **Quotes:**
   Block quotes are used for excerpts from external sources, such as references, to clearly distinguish them from the main text.

6. **Hyperlinks:** Hyperlinks are used for references, citations, and external resources to provide easy access to additional information.

7. **Acronyms and Abbreviations:** Acronyms and abbreviations are spelled out upon first use, followed by the abbreviation in parentheses. The abbreviation is then used consistently throughout the document.

2. Technical Architecture

2.1 Application and Data Architecture

The SugarSage system is a personalized diabetes management and monitoring application that will be custom-built for users. It will consist of several application components, including the User Profile, Health Tracker, Meal Planner, Feedback, and Blog Reading modules. The application will primarily collect and manage data related to user health metrics, dietary intake, user feedback, and educational content. The critical data components will be User Data, Health Metrics, Meal Plans, Feedback, and Blogs.

The application will be developed using a layered architecture, utilizing a client/server model. The client will be a mobile application or web interface, while the server will provide API endpoints for data processing and storage. We will use programming languages suitable for mobile and web development, such as Flutter for mobile and JavaScript (Node.js or React) for the web client, with a server backend possibly in Python or Java.

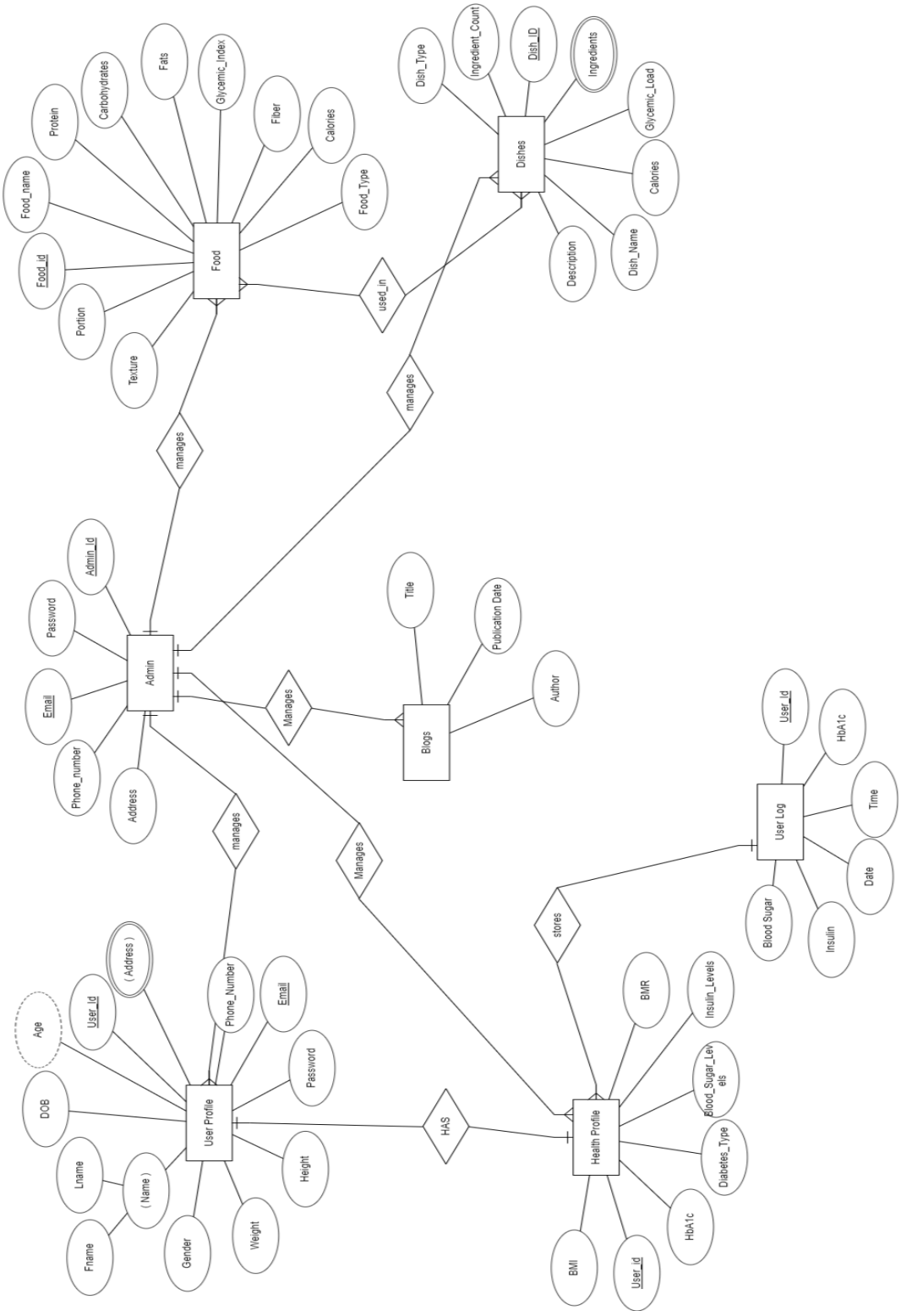
The hardware platform will not be device-specific, as the system is designed to be compatible with various smartphones and personal computers. Our backend infrastructure will be cloud-based, allowing for scalability and reliability. Depending on the data needs, we will use a relational database management system (RDBMS) like PostgreSQL or MySQL or a NoSQL database like MongoDB.

Our system will have a user-friendly interface accessible via mobile devices and web browsers. It is designed to be accessible online, ensuring users can manage their diabetes care anytime, anywhere.

The system will be hosted on a cloud platform (e.g., AWS, Azure) that provides the necessary computing resources, storage, and networking capabilities to support the application and data architecture.

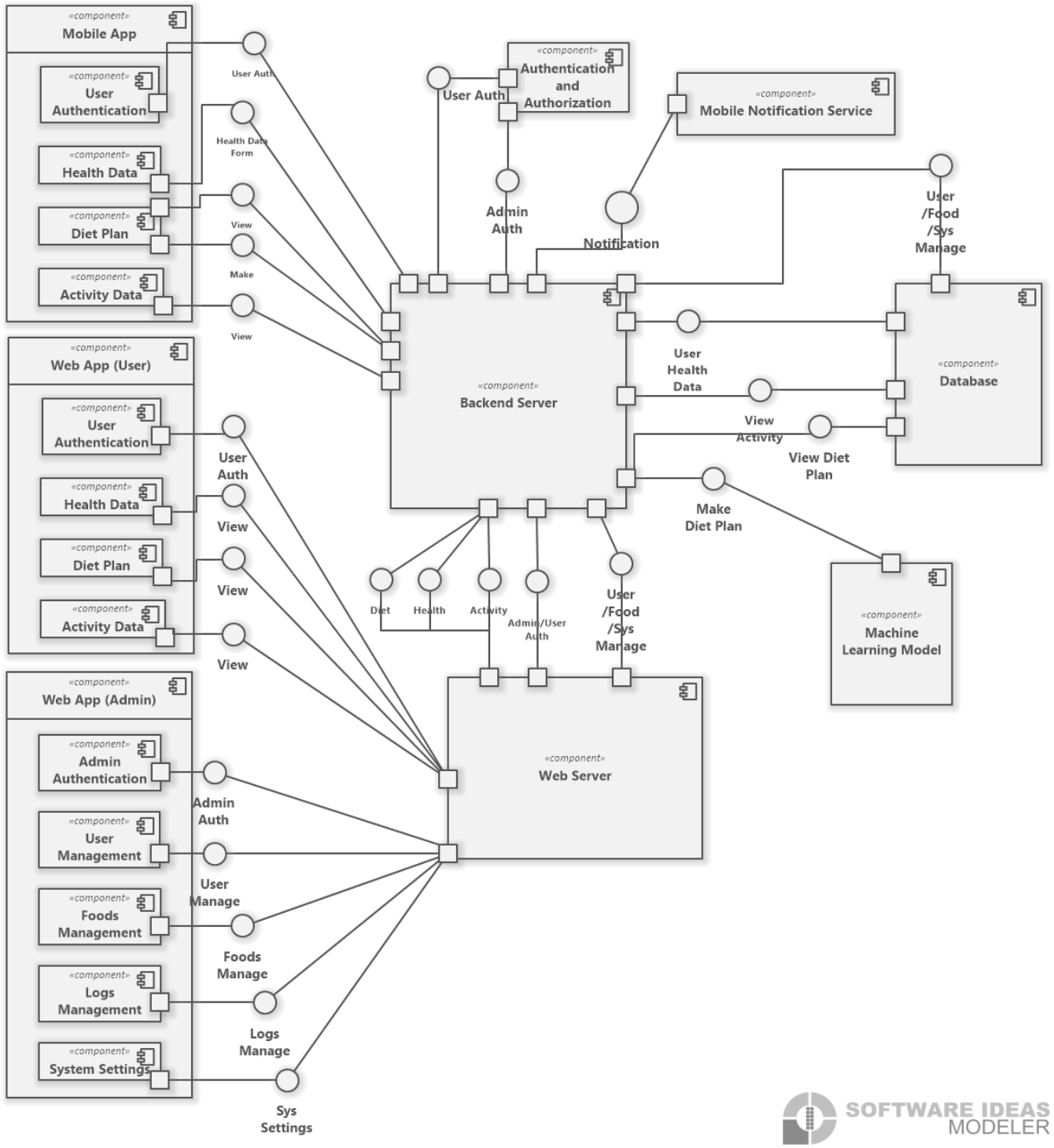
Diagrams and Descriptions

1. ER Diagram:



The Entity-Relationship (ER) Diagram depicts the data model of the SugarSage system, showing the entities (tables) like User Profiles, Admin, Blogs, Food, Dishes, and their relationships. It includes attributes of each entity and the types of relationships (one-to-many, many-to-many) between them.

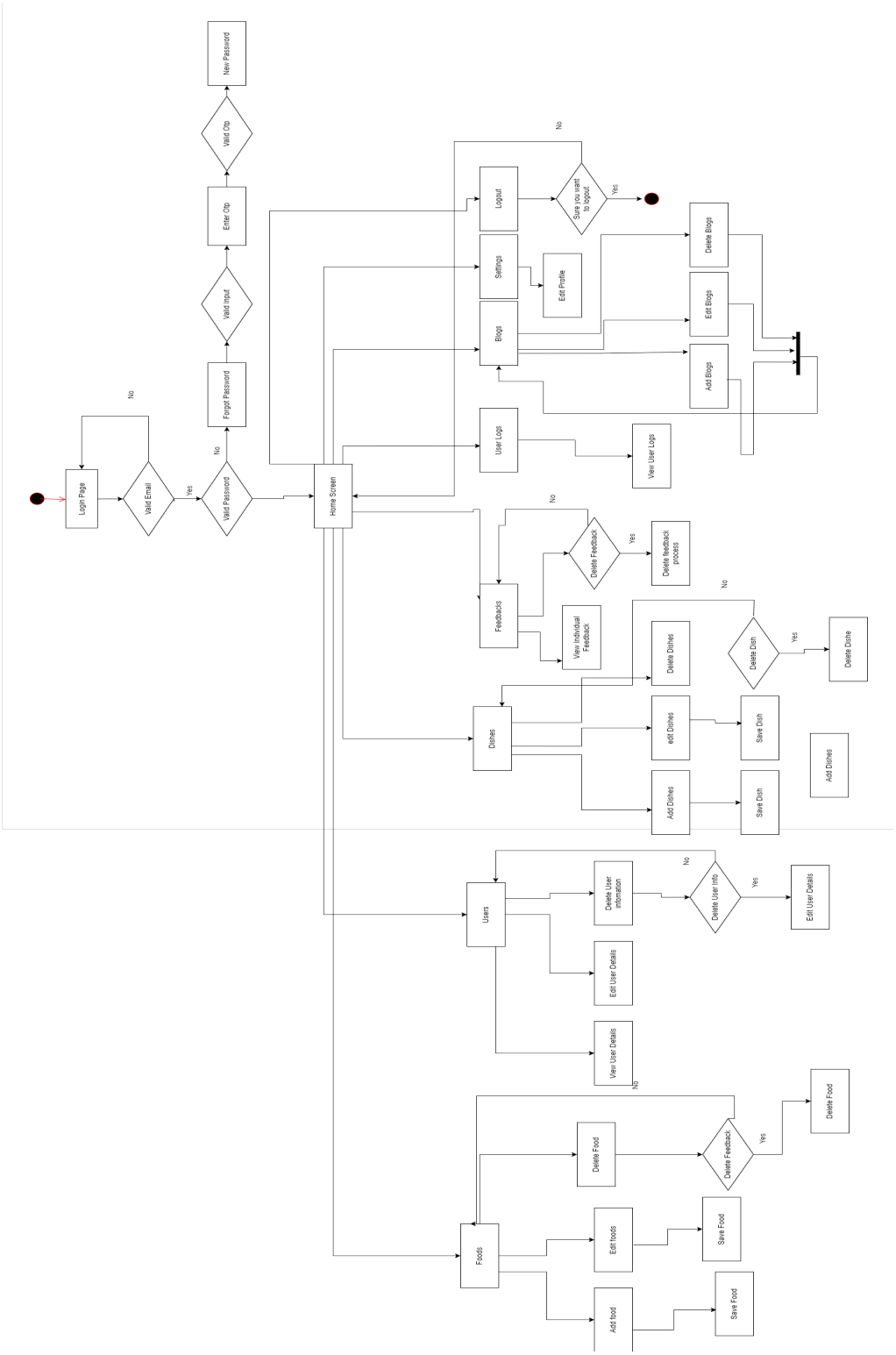
2. Component Diagram:



The Component Diagram illustrates the high-level architecture of the SugarSage system, showing the main components like the mobile app (for both users and admins), backend server, database, and machine learning model. It highlights the interaction between these components and subsystems, such as authentication, health data, diet plan, activity data, and notification service.

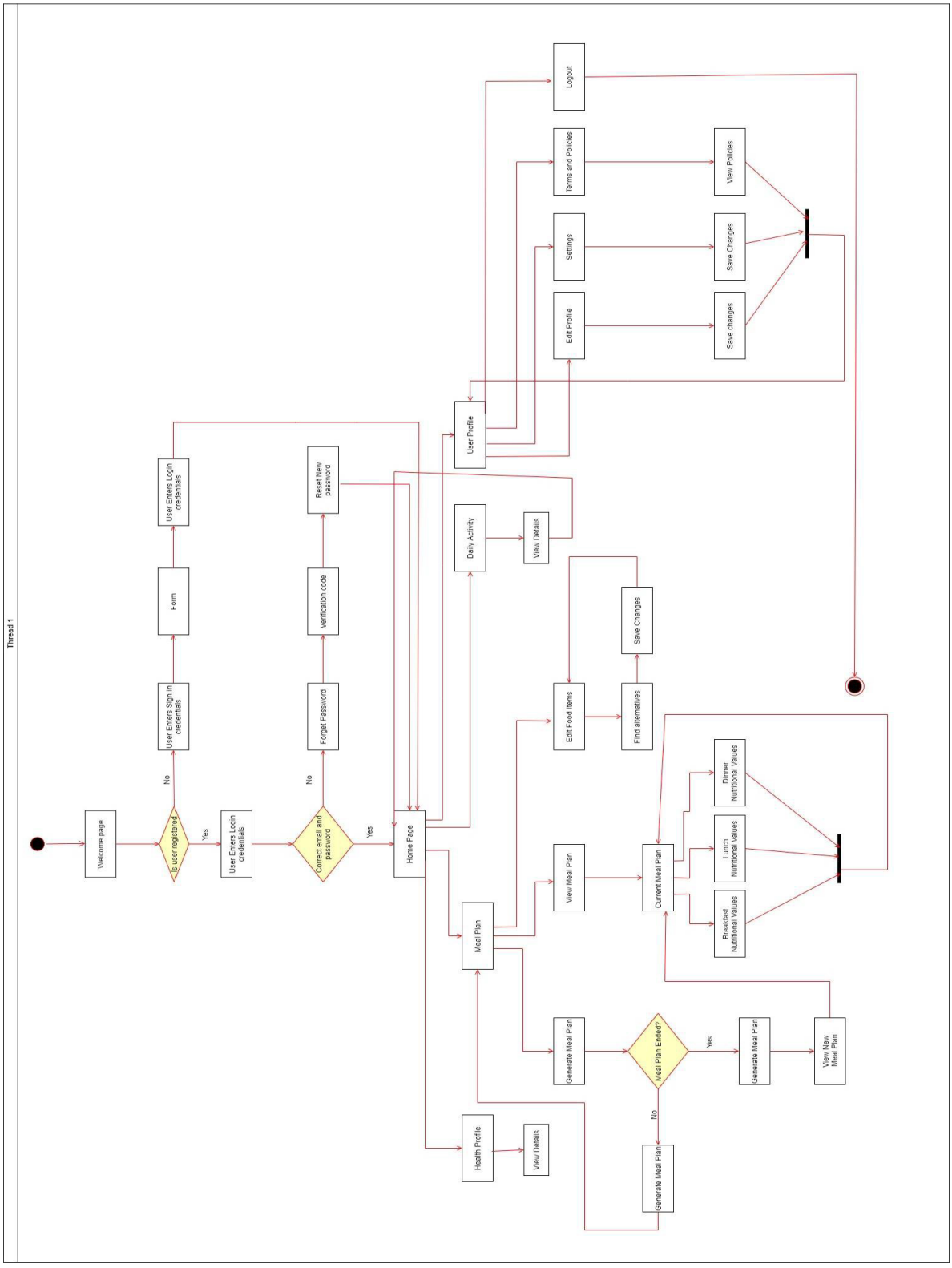
3. Activity Diagram:

- **Admin Activity Diagram:**



The Admin Activity Diagram is a flowchart that details the various actions an admin can perform within the SugarSage system. It covers processes such as logging in, managing user information, editing foods, handling feedback, and managing blog content, providing a visual guide to the admin's workflow and decision points.

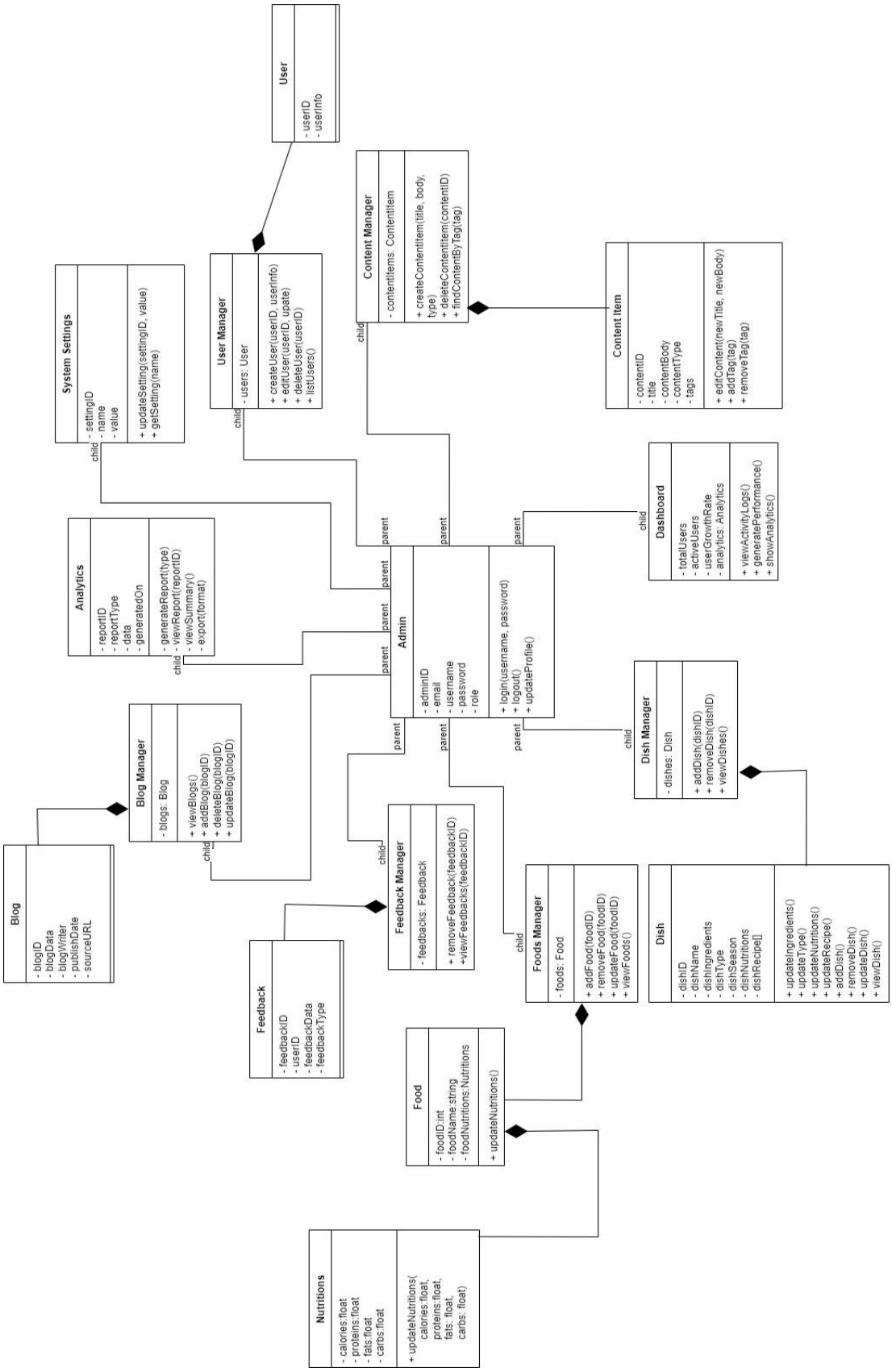
- **User Activity Diagram:**



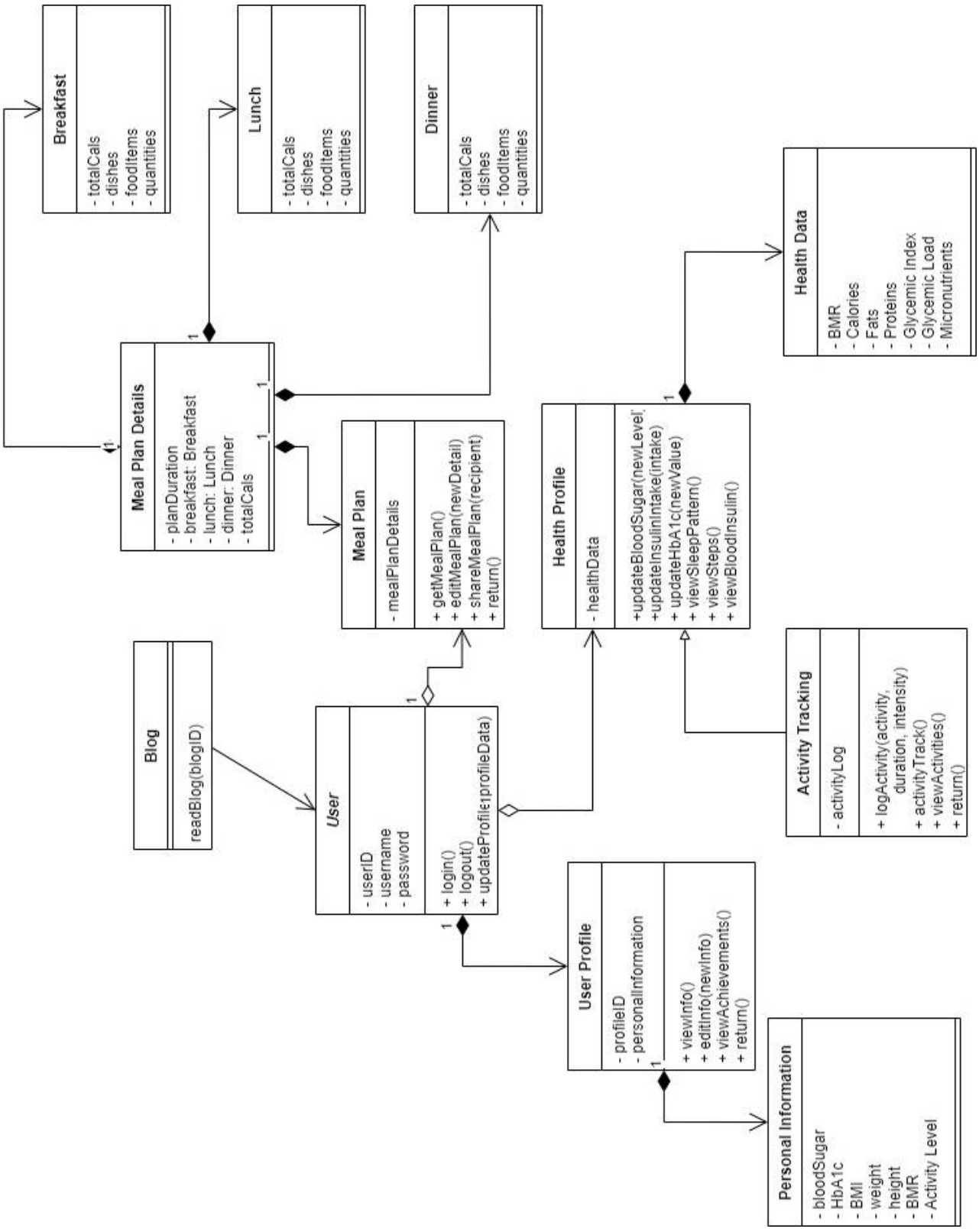
This diagram shows how users navigate the SugarSage system, from the welcome page to managing their health profile, meal plans, daily activities, and profile settings.

3. Class Diagram:

- **Admin Class Diagram:**



- **User Class Diagram:**



2.2 Component Interactions and Collaborations

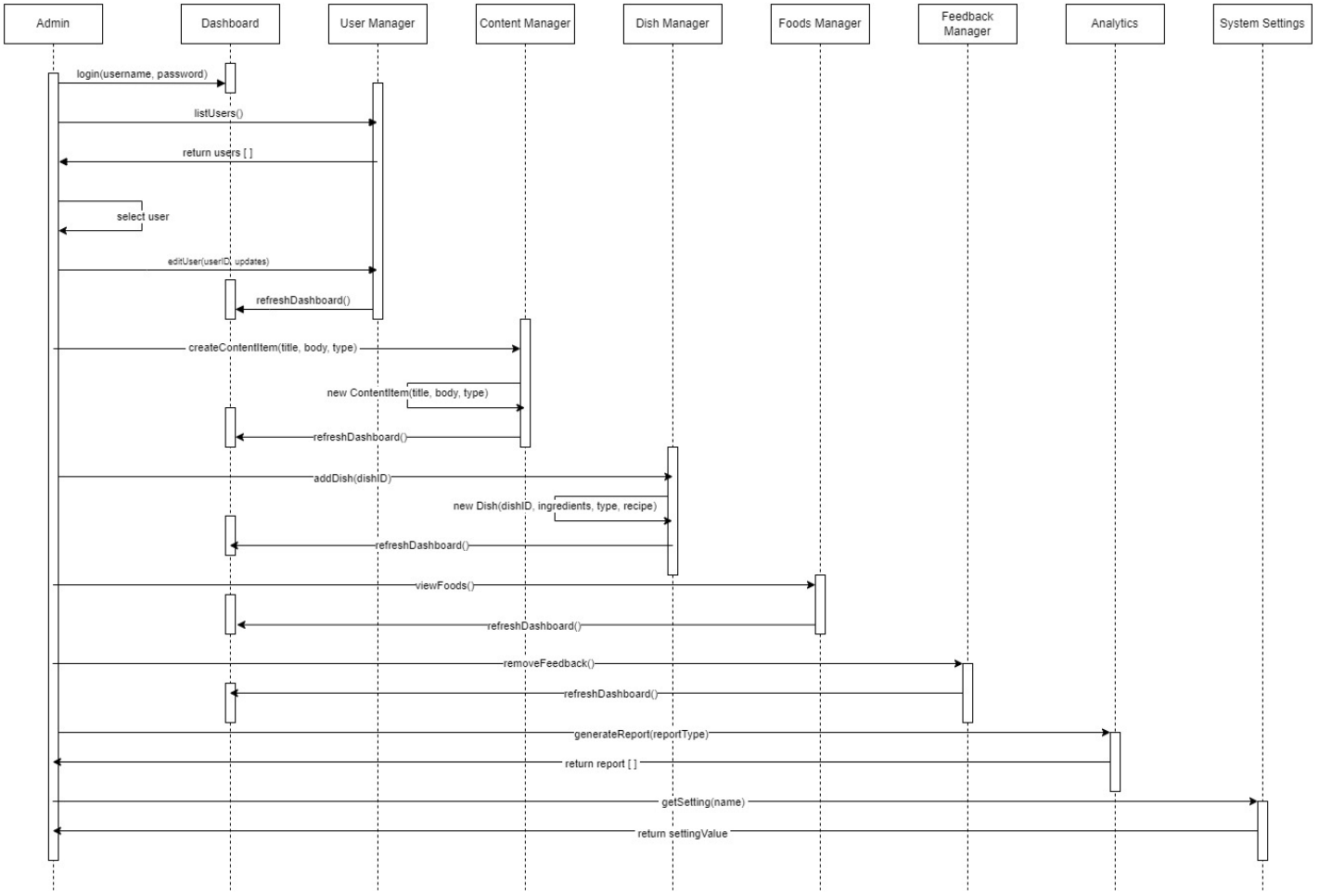
The components interact through well-defined API calls and responses. The User Profile component communicates with the Health Tracker to update and retrieve health metrics. The Meal Planner interacts with the User Profile to tailor diet plans based on user health data. The Feedback module lets users communicate their experience directly to the system administrators. Blog Reading enables users to access educational content provided by the system.

Design Level Sequence Diagrams, Collaboration Diagrams, and Event Trace diagrams will illustrate these components' detailed interactions and collaborations. These diagrams will show more details than in Phase 1, with explicit message calls, return values, and sequence flows.

Diagrams and Descriptions

1. **Sequence Diagram:**

- **Admin Sequence Diagram:**

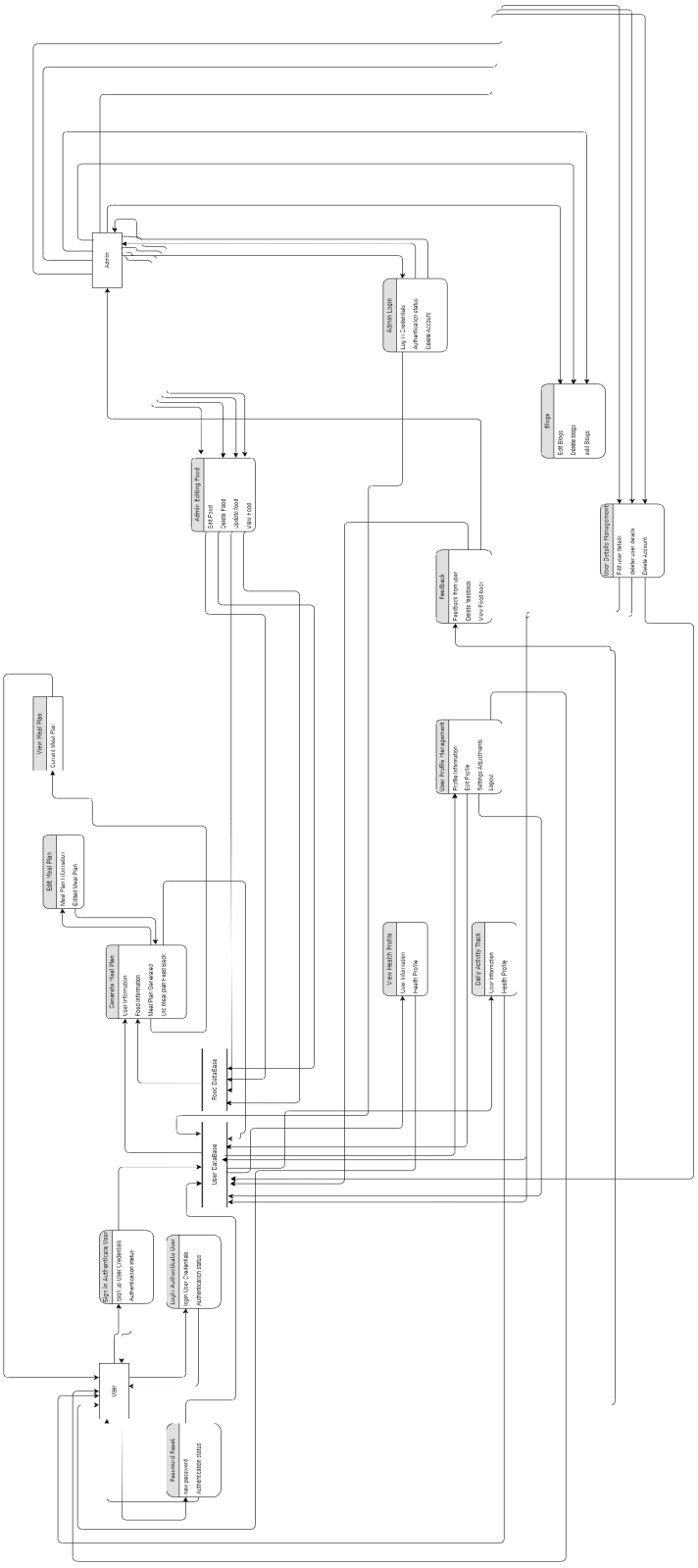


This sequence diagram provides a detailed interaction flow for administrative actions within the SugarSage system. It outlines the sequence of method calls between components such as the Admin, Dashboard, User Manager, Content Manager, Dish Manager, Foods Manager, Feedback Manager, Analytics, and System Settings when an admin performs tasks like user management or content creation.

- **User Sequence Diagram:**

This diagram displays the sequence of interactions from a user's perspective as they navigate the SugarSage system. It includes method calls among components like User, Home Page, Meal Plan, Activity Tracking, User Profile, Blog, Feedback, and Health Profile. It shows how users might log in, manage their meal plans, track activities, and read blogs.

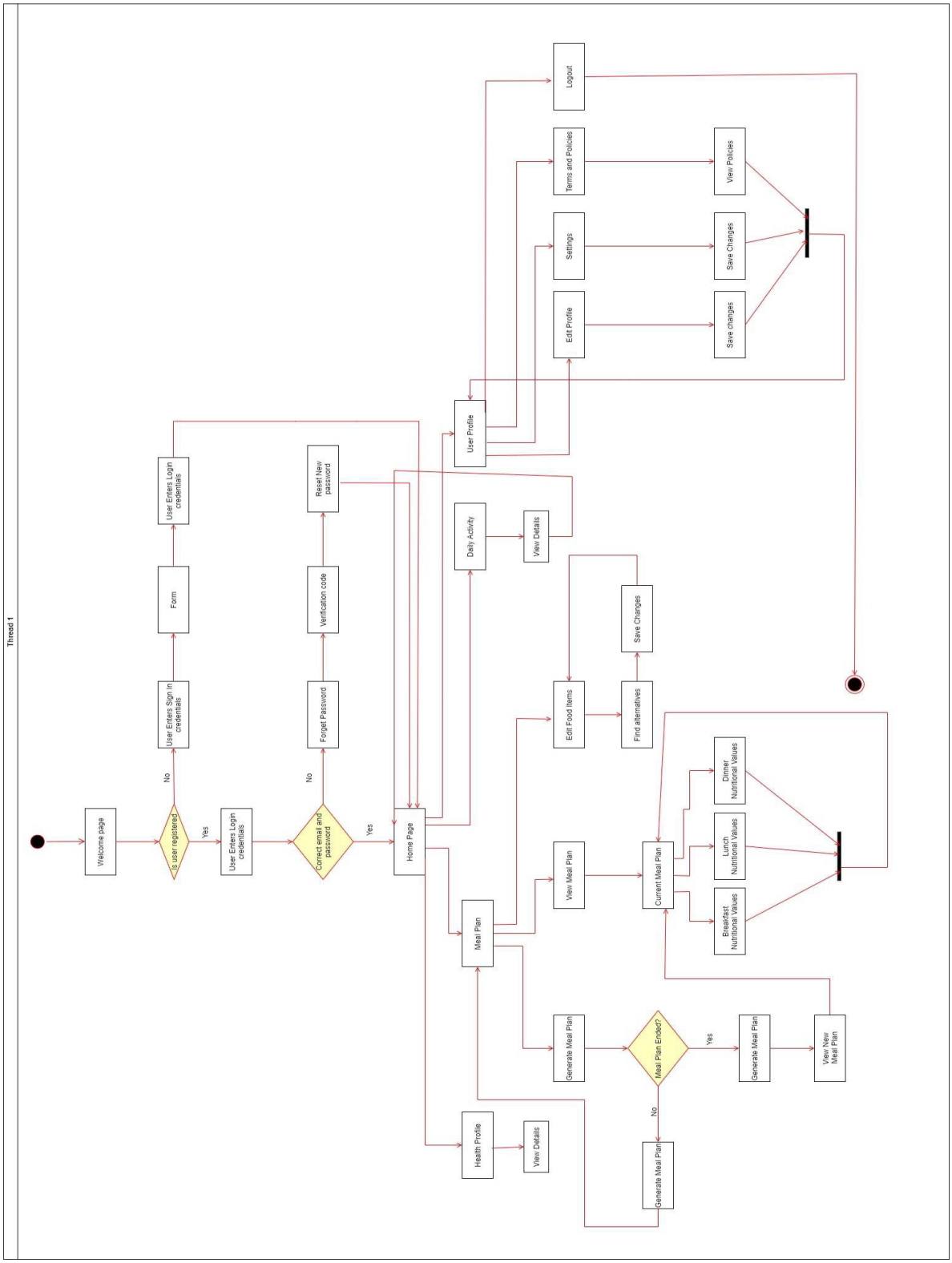
2. Data Flow Diagram:



The Data Flow Diagram visually represents data flow within the SugarSage system. It maps out how data is input, processed, and output by different entities in the system, including user and admin interactions, information storage, and the processing of health data, meal plans, and feedback.

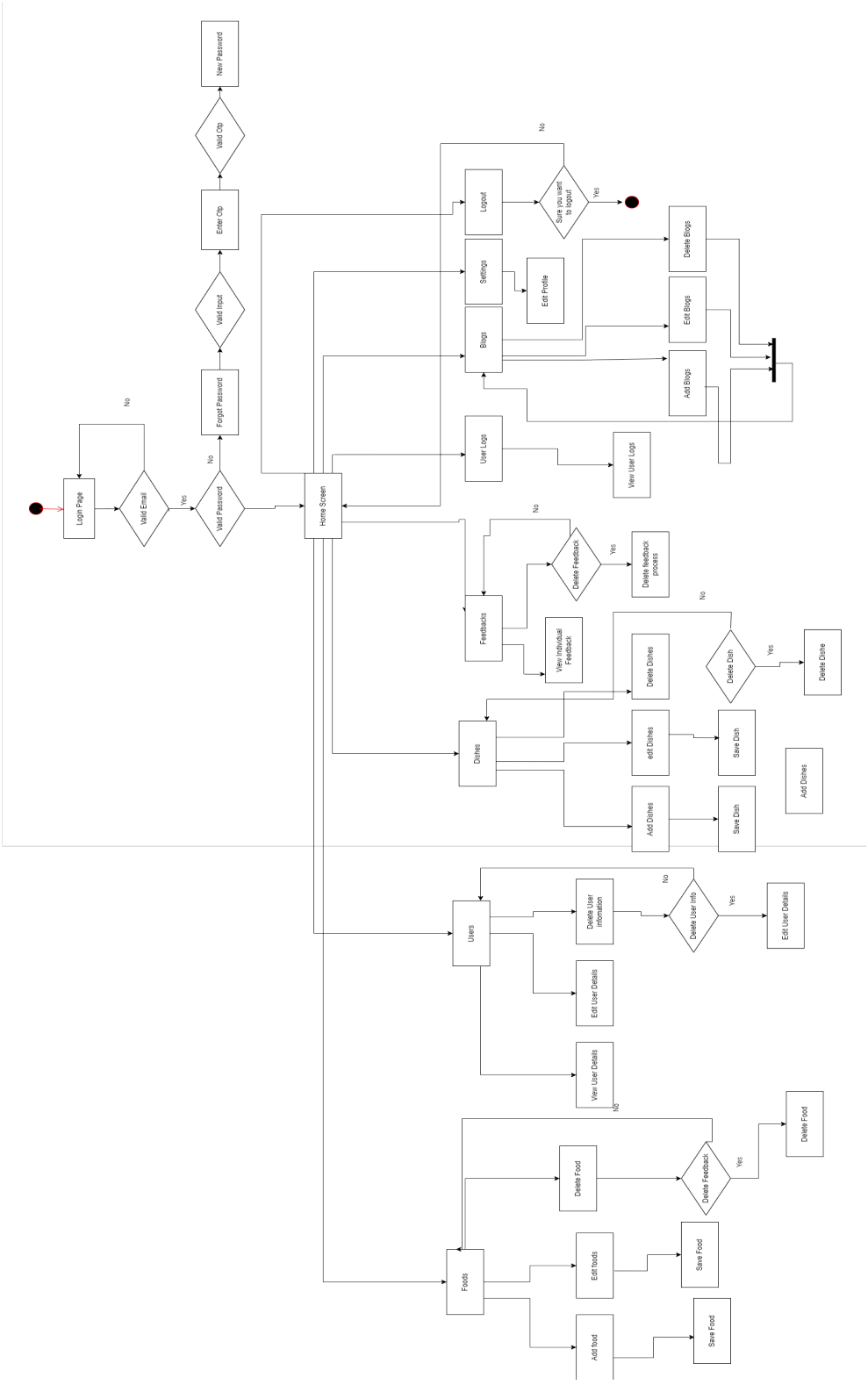
3. Activity Diagram:

- **User**



This diagram shows how users navigate the SugarSage system, from the welcome page to managing their health profile, meal plans, daily activities, and profile settings.

- **Admin Activity Diagram:**



The Admin Activity Diagram is a flowchart that details the various actions an admin can perform within the SugarSage system. It covers processes such as logging in, managing user information, editing foods, handling feedback, and managing blog content, providing a visual guide to the admin's workflow and decision points.

2.3 Design Reuse and Design Patterns

In developing our SugarSage system, we will incorporate strategies for design reuse and apply design patterns to ensure efficiency, maintainability, and scalability.

Design Reuse:

- Common Libraries:
  We will abstract functions and classes that are common across different parts of the system into common libraries. This will include components for user authentication, data validation routines, and error handling. Reusing these libraries will decrease development time and increase code consistency.

- UI Component Library:
  A set of reusable user interface components will be created to ensure a consistent user experience across the mobile and web applications. These components will range from form elements to complex data visualization tools.

- Service Layer:
  Our system will feature a service layer that abstracts business logic from the controllers, promoting reuse for different parts of the application that require similar business rules processing.

Design Patterns:

- Model-View-Controller (MVC):
  We will employ the MVC pattern to separate concerns, making our system easier to maintain and extend. The Model will manage the data and business logic; the View will handle the presentation of information to the user; the Controller will process user input and update the Model and View accordingly.

- Observer Pattern:
  We will utilize the Observer pattern to enable a publish-subscribe model, allowing components to listen and react to events or changes in other components. This pattern will be particularly useful in the SugarSage system for updating user interfaces in response to changes in health data or for dispatching notifications.

- Singleton Pattern:
  We will ensure that classes such as the database connection manager or the configuration manager are instantiated only once throughout the application to manage shared resources effectively.

2.4 Technology Architecture

Our anticipated infrastructure for the SugarSage system is designed to be robust and adaptable to the dynamic needs of users and system administrators. The core of our infrastructure will rely on a cloud-based environment, which provides several key advantages:

Cloud-Based Environment:

- Auto-Scaling:
  Our system will leverage the auto-scaling capabilities of the cloud to adjust resources dynamically based on the current load. This ensures the system remains responsive during peak usage times without incurring unnecessary costs during off-peak times.

- High Availability:
  The cloud infrastructure is designed for high availability, with redundancy and failover mechanisms to minimize downtime and ensure users have continuous access to the system.

Connectivity Requirements:

- Internet Access:
  Reliable internet access is a foundational requirement for SugarSage, enabling users to interact with the system from anywhere. Our system will utilize HTTPS protocols to secure all client and server communications.

Modes of Operation:

- Synchronous Online Transactions:
  Our system will support synchronous online transactions, essential for immediate and interactive user experiences. This includes actions like logging meals, tracking blood sugar levels, and receiving real-time feedback.

- Asynchronous Batch Processing:
  In addition to real-time interactions, our system will also handle asynchronous batch processing tasks. These tasks are critical for data analysis, health report generation, and other processes that do not require immediate user interaction.

Technology Stack:

- **Mobile Application Development:**

  Flutter: Chosen for its ability to compile into native code, which enhances performance and ensures a smooth user experience on both iOS and Android platforms. Flutter's extensive widget library facilitates the creation of visually appealing and responsive UIs, critical for engaging diabetic patients who require frequent interaction with the app.

- **Web Application Development:**

  ReactJs: Selected for its efficiency in updating and rendering components, which is essential for web applications requiring real-time data updates. React's component-based architecture makes it scalable, aiding in developing complex features such as diet tracking and progress monitoring. Its strong community support ensures access to a wide range of libraries and tools, enhancing development speed and capabilities.

- **Backend Development:**

  Node.js: Utilized for its non-blocking, event-driven architecture, which is excellent for asynchronous requests and handles multiple connections simultaneously. This is crucial for handling the intensive load of user interactions and data processing requests from both mobile and web platforms.

  Python: Integrated specifically for its robustness in scientific computing and machine learning. Python is used to handle the computational aspects of our machine learning models, which suggest personalized diet plans based on user health data. Python's extensive ecosystem of libraries, such as TensorFlow and Scikit-learn, facilitates efficient model development and integration.

- **Database Management:**

  SQL Database: Employed to manage structured data storage, offering robust transaction support and reliability. SQL databases ensure data integrity and security, which are paramount when dealing with sensitive medical information of users. They also provide powerful query capabilities that enable quick retrieval of user records and insights into app usage patterns.

System Hosting and Platform:

- **Cloud Platforms:**
  Platforms such as AWS, Azure or Google Cloud will be used to host our system, offering a range of services from compute instances to managed databases, which are essential for building a scalable and secure system.

2.5 Architecture Evaluation

2.5.1 Cloud-Based Infrastructure (AWS, Azure, Google Cloud):

1. **Pros:**

- Scalability: Cloud platforms provide the ability to scale resources dynamically. This flexibility is crucial for handling fluctuations in system demand without needing physical infrastructure expansion.
- Cost-Effectiveness: Using cloud services allows for a pay-as-you-go pricing model, which means the system incurs costs only based on actual usage, reducing unnecessary expenditure.
- High Availability and Reliability: Cloud platforms offer built-in redundancy and disaster recovery capabilities, ensuring the system remains operational and accessible without significant downtime.

2. **Cons:**

- Vendor Lock-in: Depending on proprietary services offered by specific cloud providers can make it challenging to migrate to other platforms in the future.
- Complexity in Management: Managing a cloud-based infrastructure requires specialized skills and understanding of cloud services, which can introduce deployment and ongoing management complexity.
- Reason for Selection: The decision to use cloud platforms like AWS, Azure, or Google Cloud was based on their robustness in handling large-scale deployments and their extensive suite of services that cater to all aspects of the system's needs, from computing power to database management.

2.5.2 Programming Languages and Frameworks:

1. **Flutter for Mobile Development:**

- **Pros**: Provides a single codebase for both iOS and Android platforms, which simplifies development and maintenance. It also supports native performance, which is critical for ensuring a smooth user experience.
- **Cons**: Flutter is relatively new, and while it has a growing community, it lacks some of the extensive resources and third-party libraries available to more established platforms like React Native.
- **Reason for Selection**: Chosen for its rapid development capabilities and excellent performance across both major mobile platforms, which is essential for a healthcare application requiring high responsiveness and reliability.

2. **ReactJs for Web Development:**

- **Pros**: Known for its virtual DOM, which makes it highly efficient for dynamic content updates—crucial for real-time web applications.
- **Cons**: React's JSX syntax and architectural depth can present a steep learning curve for new developers.
- **Reason for Selection**: React's component-based architecture allows for efficient development of complex user interfaces, and its widespread adoption ensures strong community support and continual updates.

2.5.3 Node.js and Python in Backend Development:

- **Pros**: Node.js offers non-blocking I/O operations that efficiently handle concurrent requests, making it ideal for high-performance backends. Python is renowned for its simplicity and robust library support, particularly for scientific computing and machine learning tasks.
- **Cons**: Python can be slower than some alternatives like Java or C++ for certain backend operations. Node.js, being single-threaded, might require additional considerations for CPU-bound tasks.
- **Reason for Selection:** Node.js was selected for its efficiency with I/O-bound tasks and Python for its superiority in data analytics and machine learning, both critical for a data-driven application like SugarSage.

2.5.4 SQL Database:

- **Pros**: Offers strong consistency, transactions, and robustness for complex queries, which are indispensable for medical data handling.
- **Cons**: SQL databases can be less flexible regarding schema modifications compared to NoSQL options.
- **Reason for Selection**: The relational nature of medical data, such as user records and health metrics, makes SQL an ideal choice due to its strong data integrity and complex querying capabilities.

3. Detailed Component Design

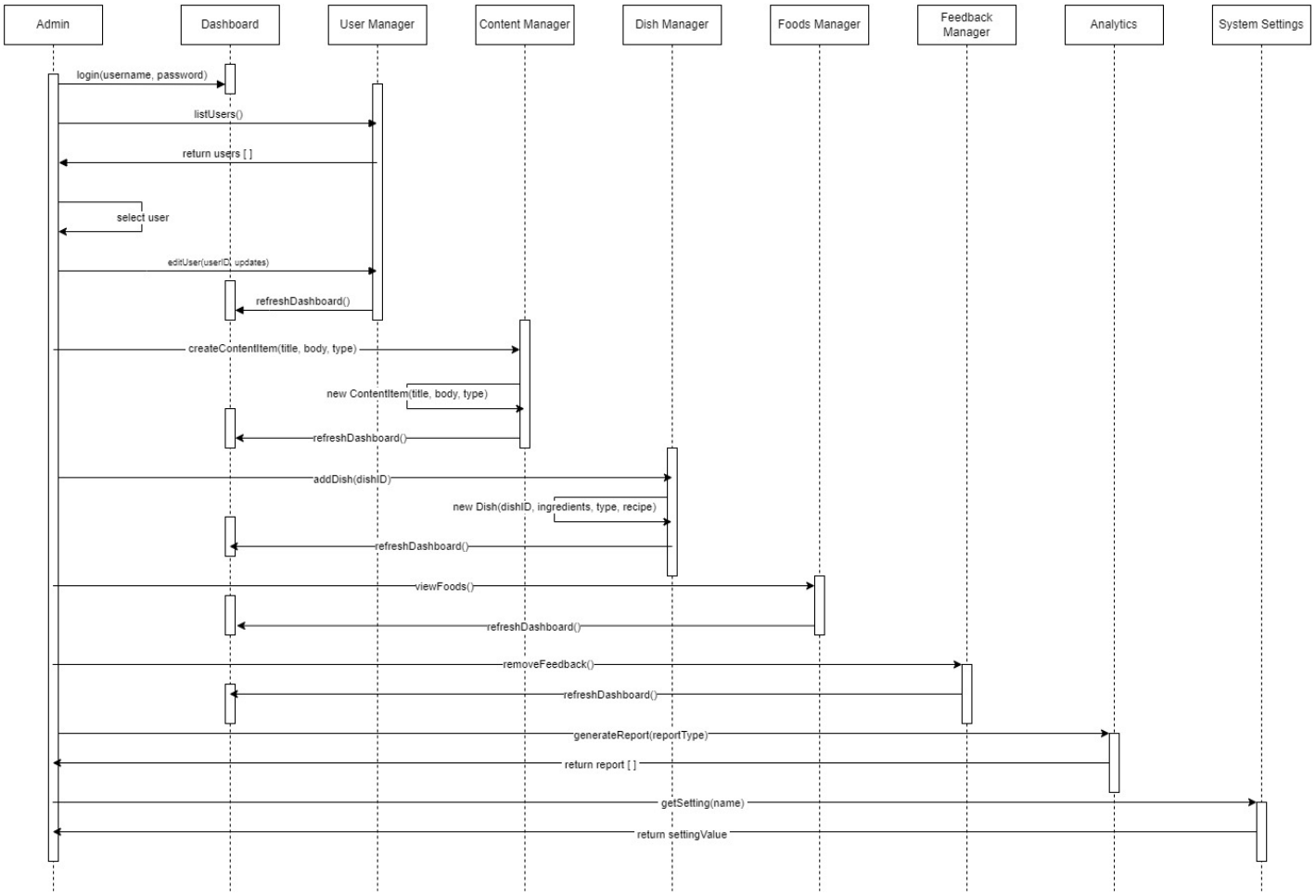
3.1 Component-Component Interface

- **User Profile and Health Tracker:**
  The User Profile component sends user demographic and health status data to the Health Tracker component, which uses this information to monitor and analyze health metrics over time.

- **Meal Planner and User Profile:**
  The Meal Planner component retrieves user-specific health data from the User Profile component to create personalized meal plans based on dietary needs and health objectives.

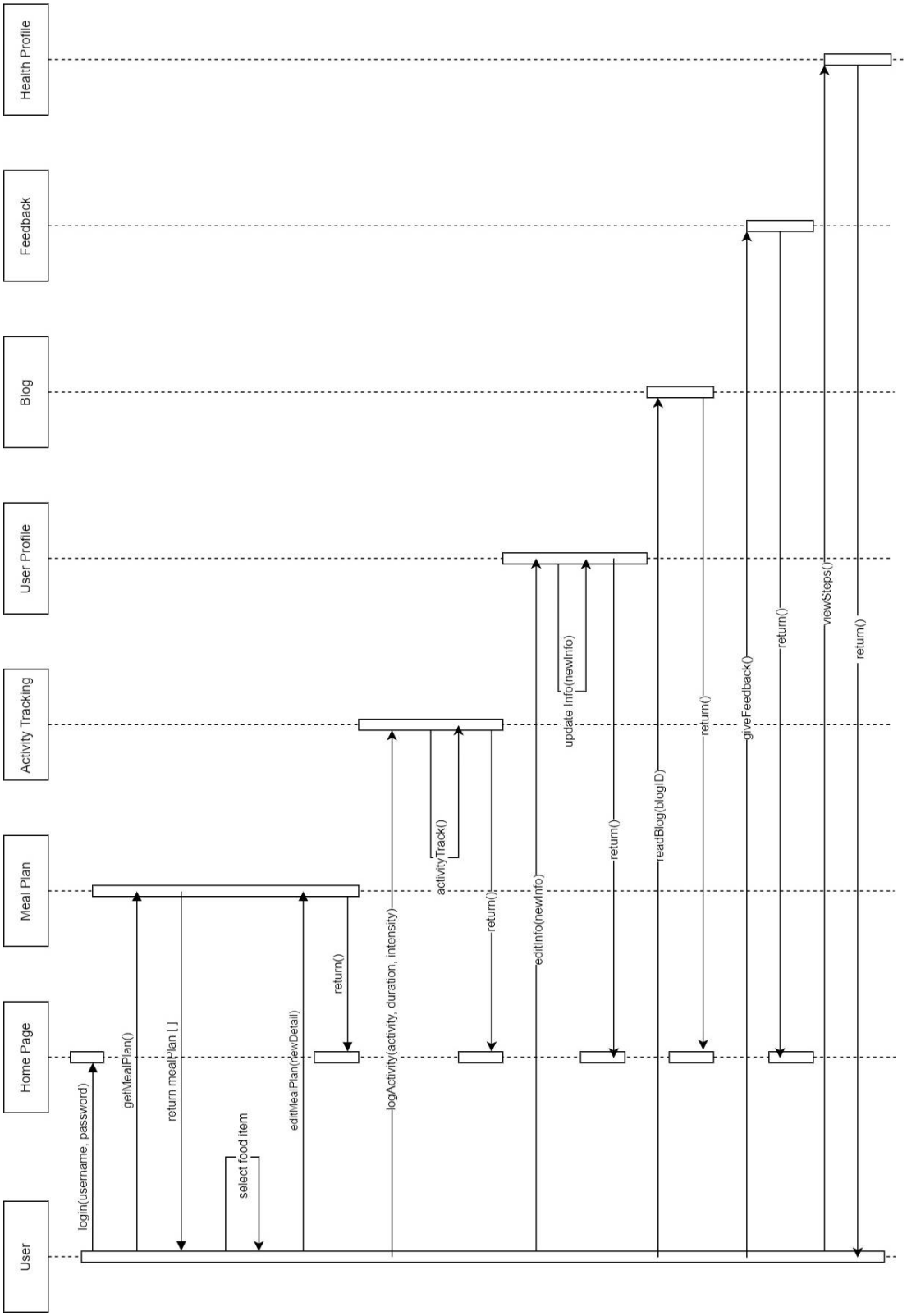
Sequence Diagram:

- **Admin Sequence Diagram:**



This sequence diagram provides a detailed interaction flow for administrative actions within the SugarSage system. It outlines the sequence of method calls between components such as the Admin, Dashboard, User Manager, Content Manager, Dish Manager, Foods Manager, Feedback Manager, Analytics, and System Settings when an admin performs tasks like user management or content creation.

- **User Sequence Diagram:**



This diagram displays the sequence of interactions from a user's perspective as they navigate the SugarSage system. It includes method calls among components like User, Home Page, Meal Plan, Activity Tracking, User Profile, Blog, Feedback, and Health Profile. It shows how users might log in, manage their meal plans, track activities, and read blogs.

3.2 Component-External Entities Interface

There are no External Entities used in the project. Everything we used is within libraries.

3.3 Component-Human Interface

1. **Screens and Interaction Points:**

   - **Login/Signup Screens**: Users receive forms to input their credentials or register a new account. Error messages are displayed if inputs are invalid or if credentials do not match existing records.

   - **Dashboard**: After login, users view their health dashboard which displays recent health metrics, suggested meal plans, and recent activities.

   - **Meal Plan Creation**: Users input their dietary preferences, health information, and other relevant data to generate customized meal plans.

   - **Health Tracking**: Users enter or sync health data such as daily physical activity, blood glucose levels, and meal intake. The system provides visual feedback on their health trends over time.

   - **Feedback Submission**: Users can submit feedback through a dedicated interface, which directly communicates with the backend to store and process these inputs for system evaluation and response.

2. **HCI Principles Applied**:

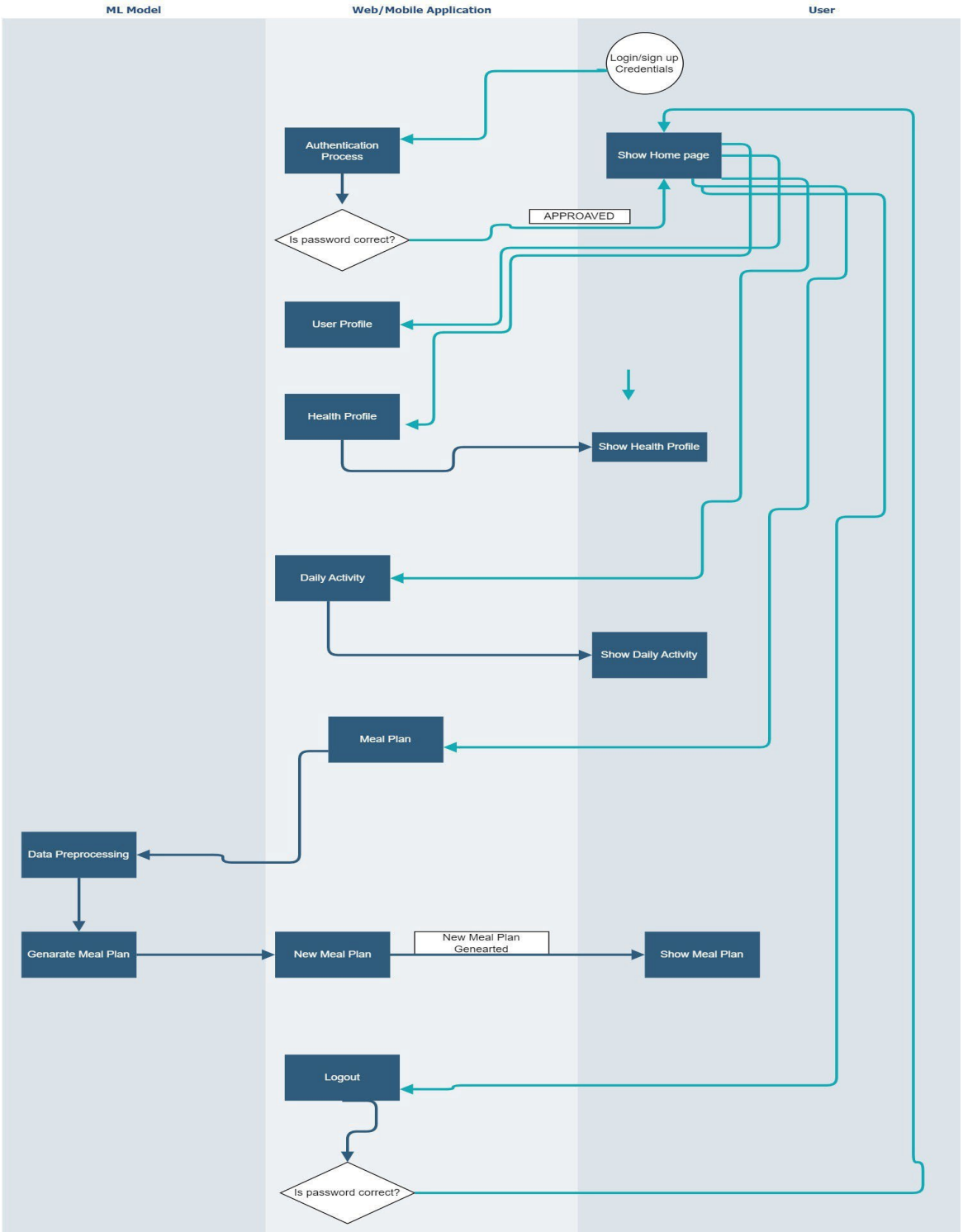
   - **Consistency**: The interface maintains consistent navigation and interaction patterns across different screens to reduce the learning curve and enhance user experience.

   - **Error Prevention and Recovery**: Forms and inputs are designed to prevent errors by using dropdowns for specific inputs and providing clear error messages to assist users in correcting entries before submission.

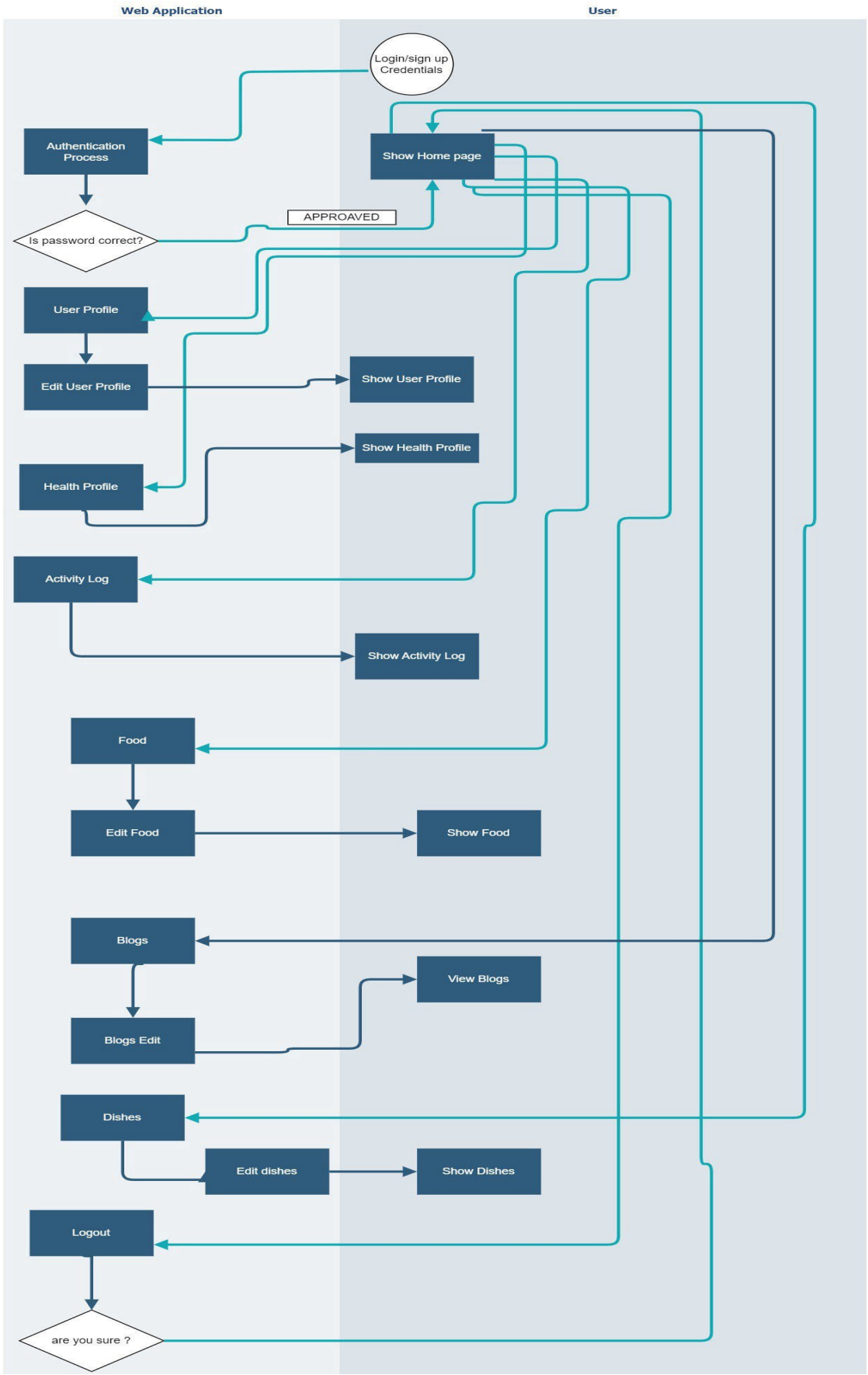
4. Screenshots/Prototype

4.1 Workflow

1. User Work Flow:

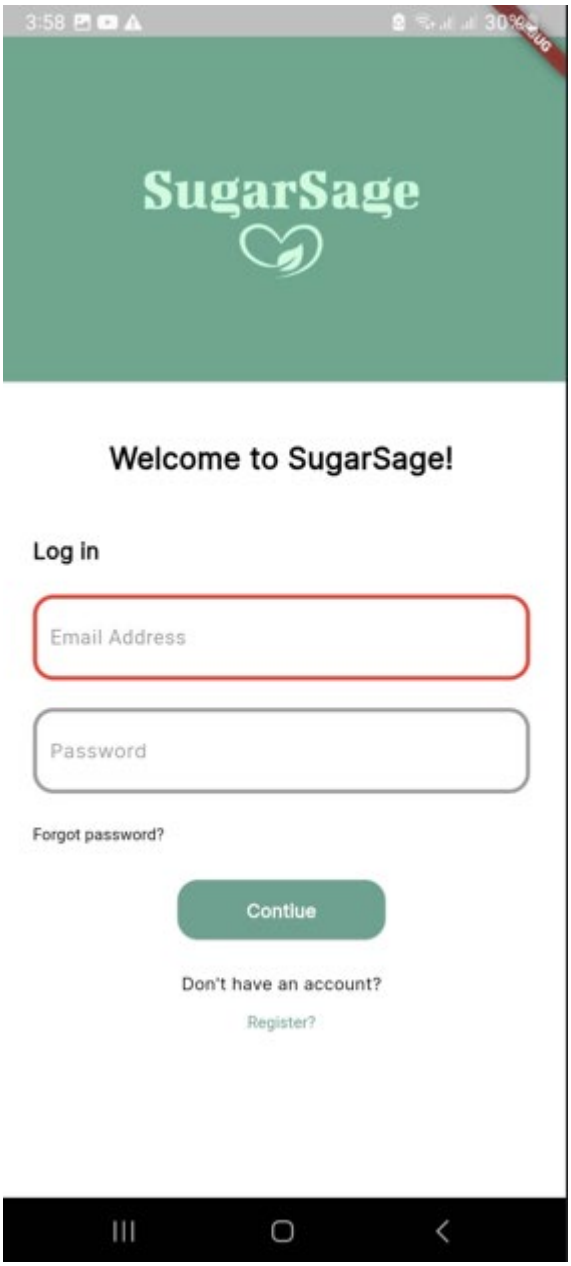


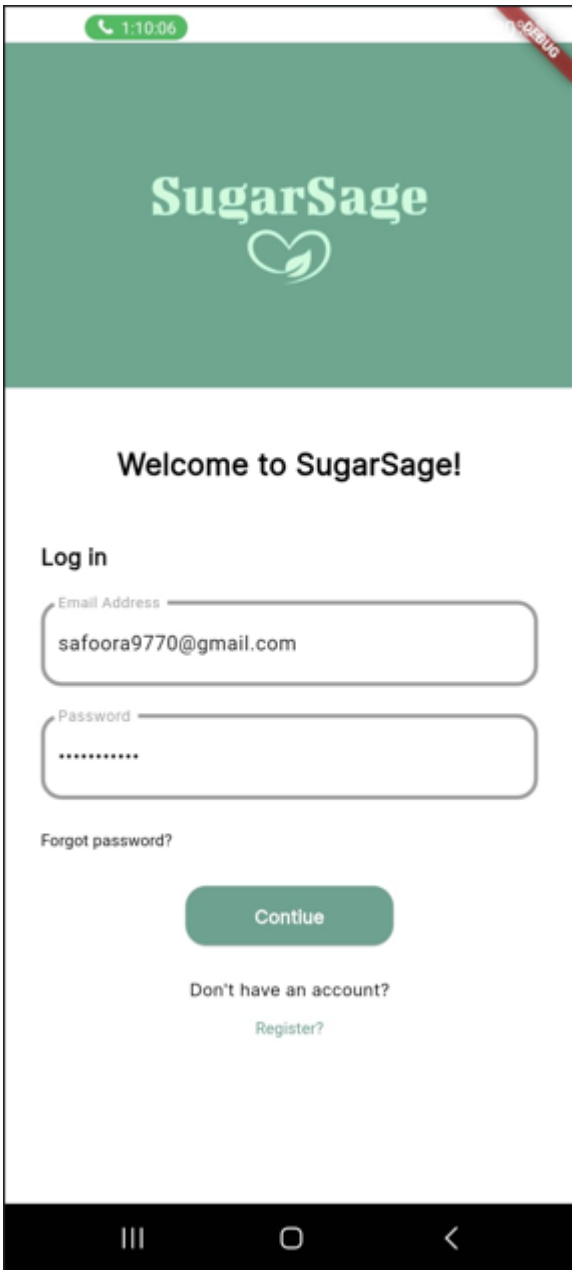
2. Admin Work Flow:

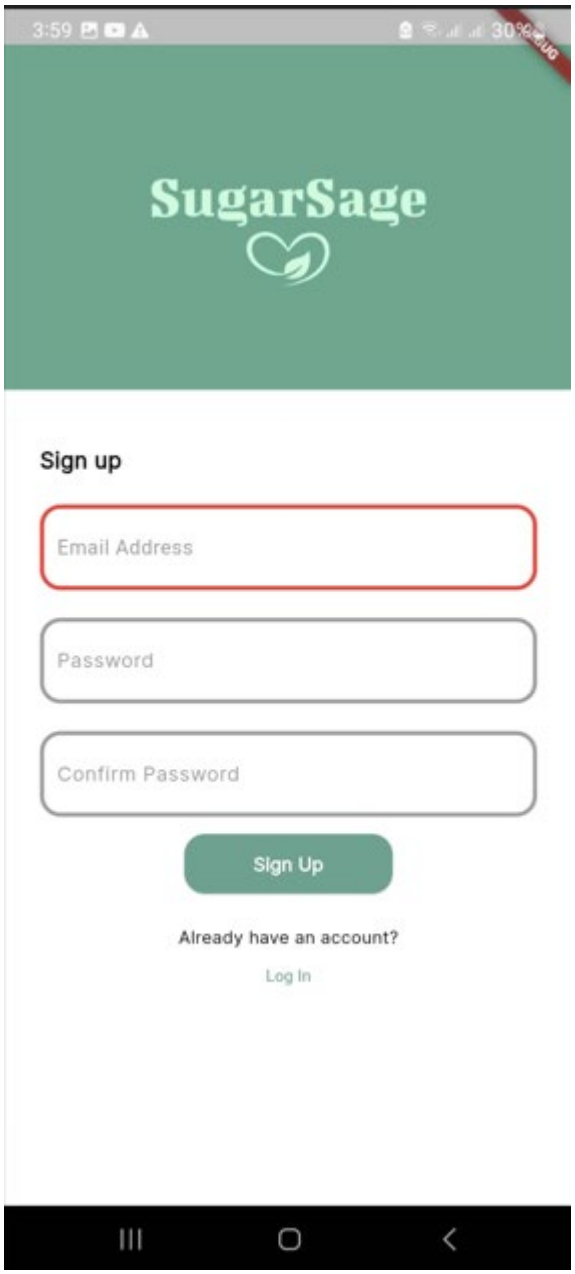


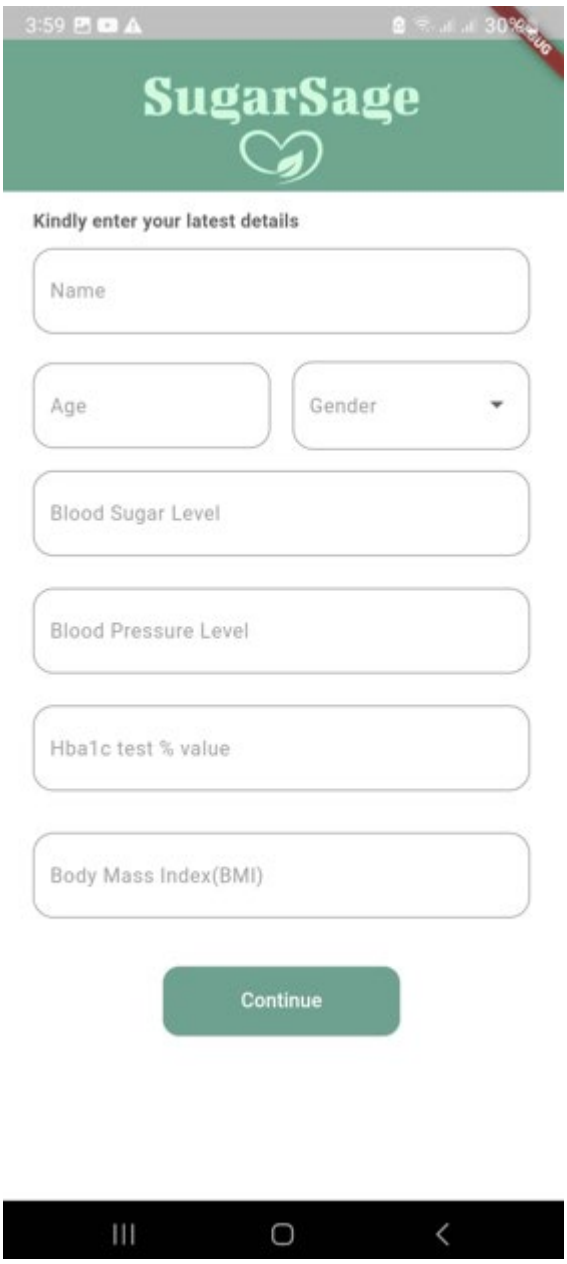
4.2 Screens

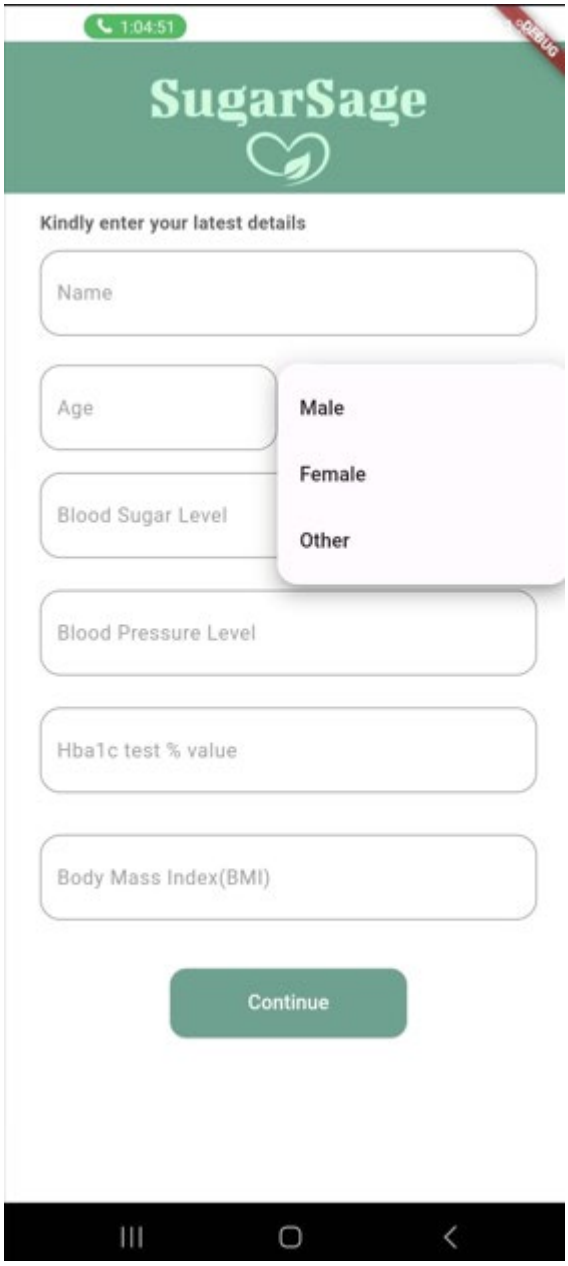
4.2.1 Mobile App Screen:

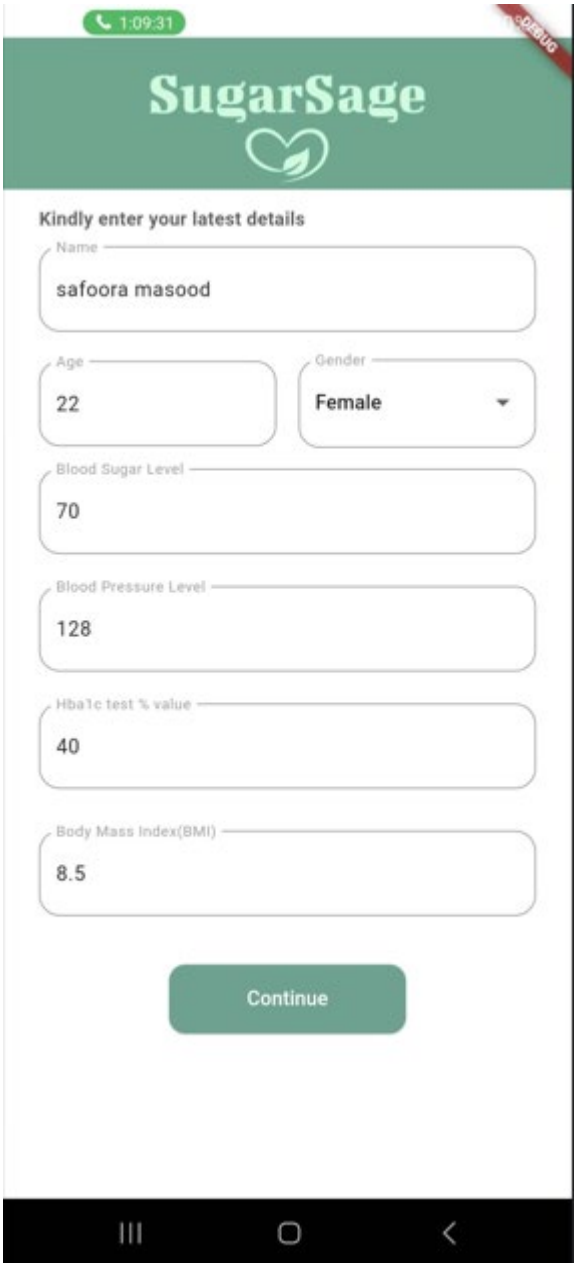


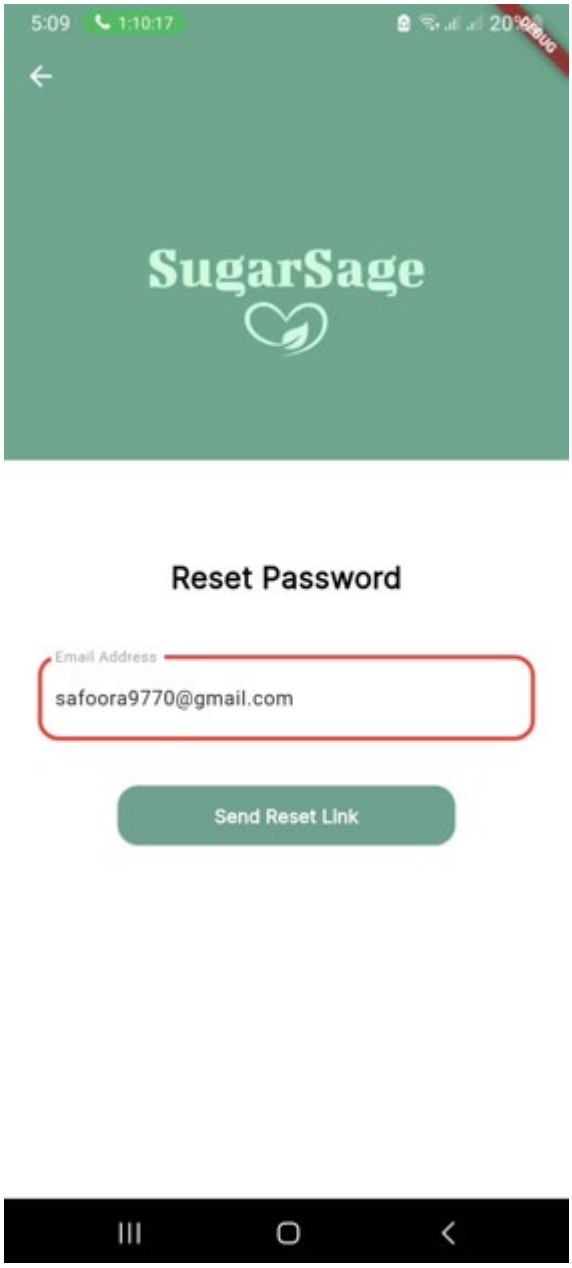

1:10:06
SugarSage
Welcome to SugarSage!
Log in
Email Address
safoora9770@gmail.com
Password
Forgot password?
Contiue
Don't have an account?
Register?

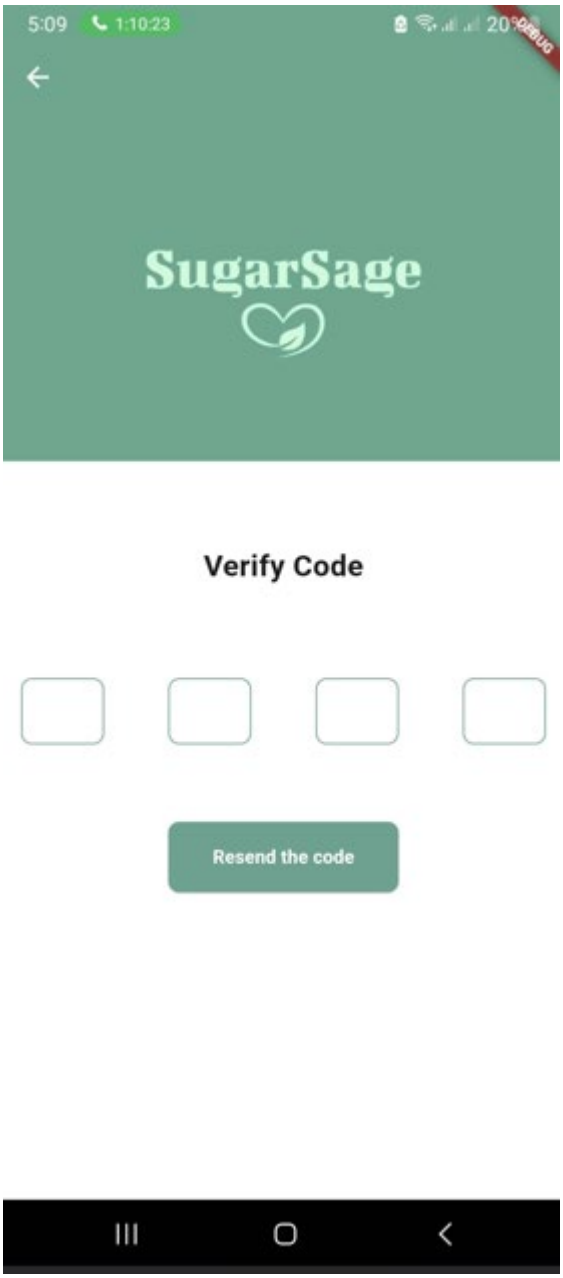

3:59
SugarSage
Sign up
Email Address
Password
Confirm Password
Sign Up
Already have an account?
Log In

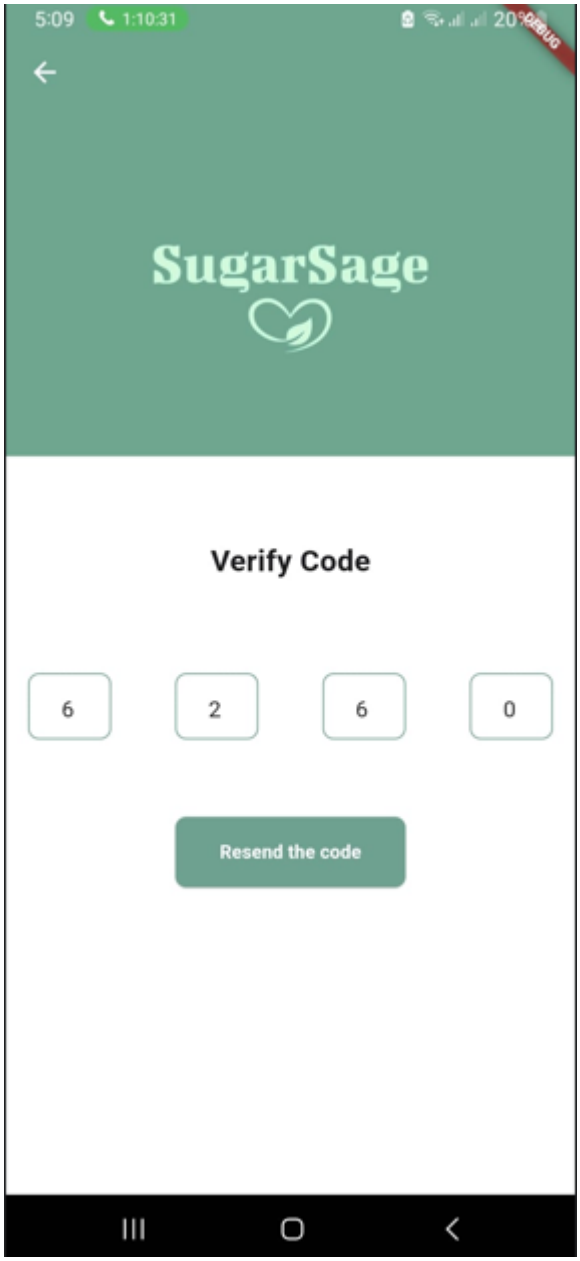

SugarSage
Kindly enter your latest details
Name
Age
Gender
Blood Sugar Level
Blood Pressure Level
Hba1c test % value
Body Mass Index(BMI)
Continue

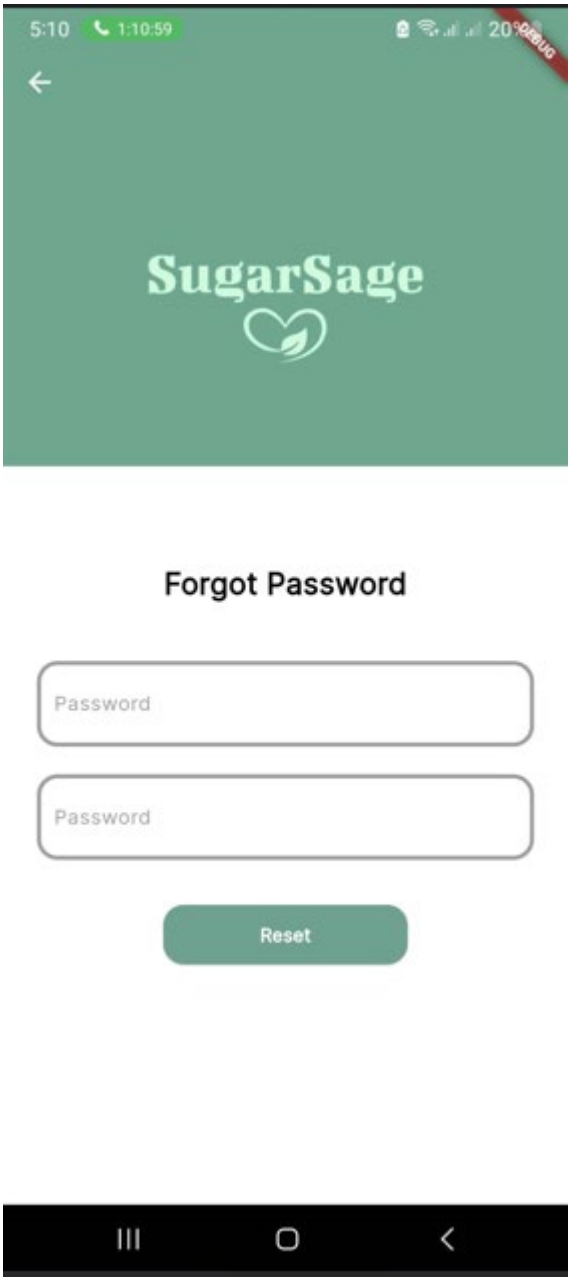

SugarSage
Kindly enter your latest details
Name
Age
Male
Female
Other
Blood Sugar Level
Blood Pressure Level
Hba1c test % value
Body Mass Index(BMI)
Continue


1:09:31
SugarSage
Kindly enter your latest details
Name
safoora masood
Age
22
Gender
Female
Blood Sugar Level
70
Blood Pressure Level
128
Hba1c test % value
40
Body Mass Index(BMI)
8.5
Continue

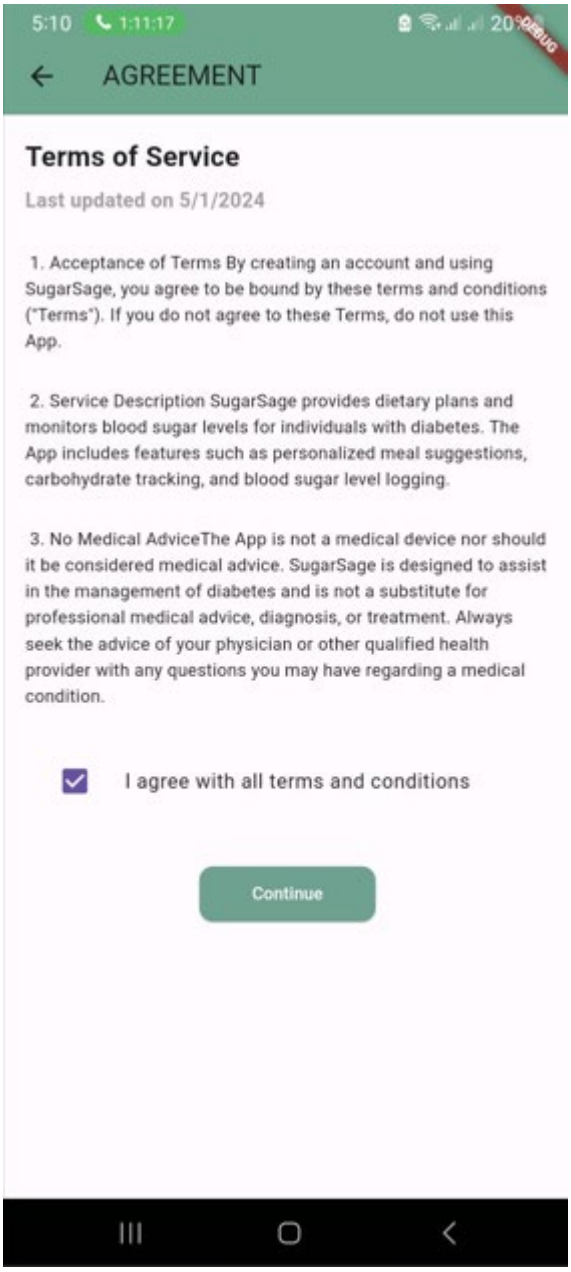

5:09
1:10:17
SugarSage
Reset Password
Email Address
safoora9770@gmail.com
Send Reset Link

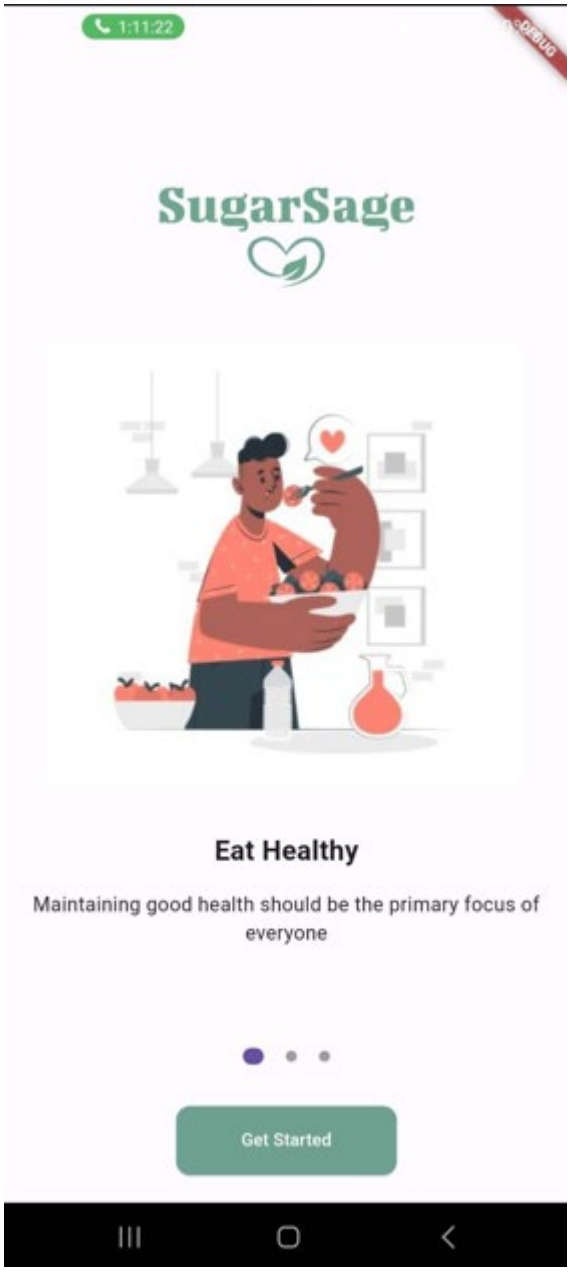

5:09
1:10:23
SugarSage
Verify Code
Resend the code


5:09
1:10:31
SugarSage
Verify Code
6
2
6
0
Resend the code

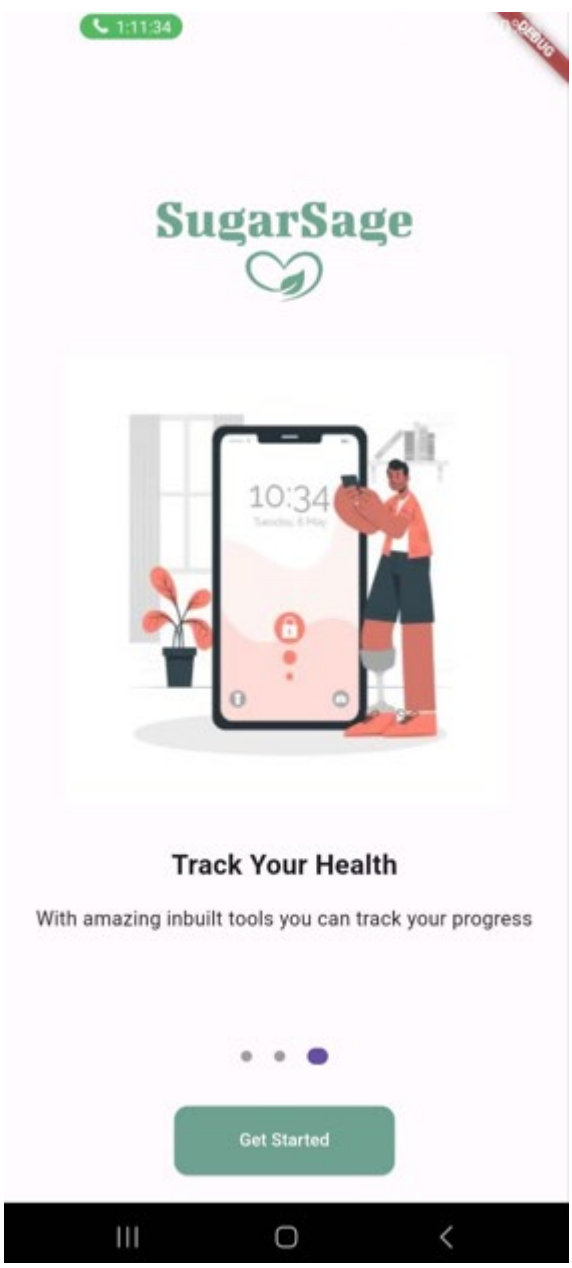

5:10
1:10:59
SugarSage
Forgot Password
Password
Password
Reset

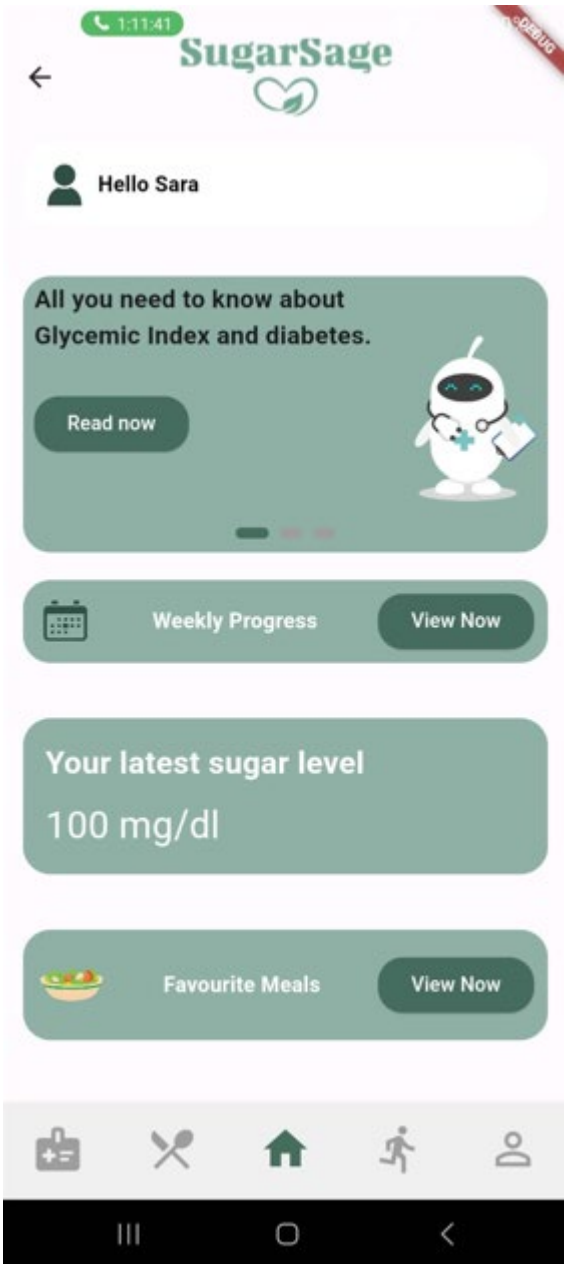

5:10
1:10:49
SugarSage
Forgot Password
Password
Password
Password
Reset

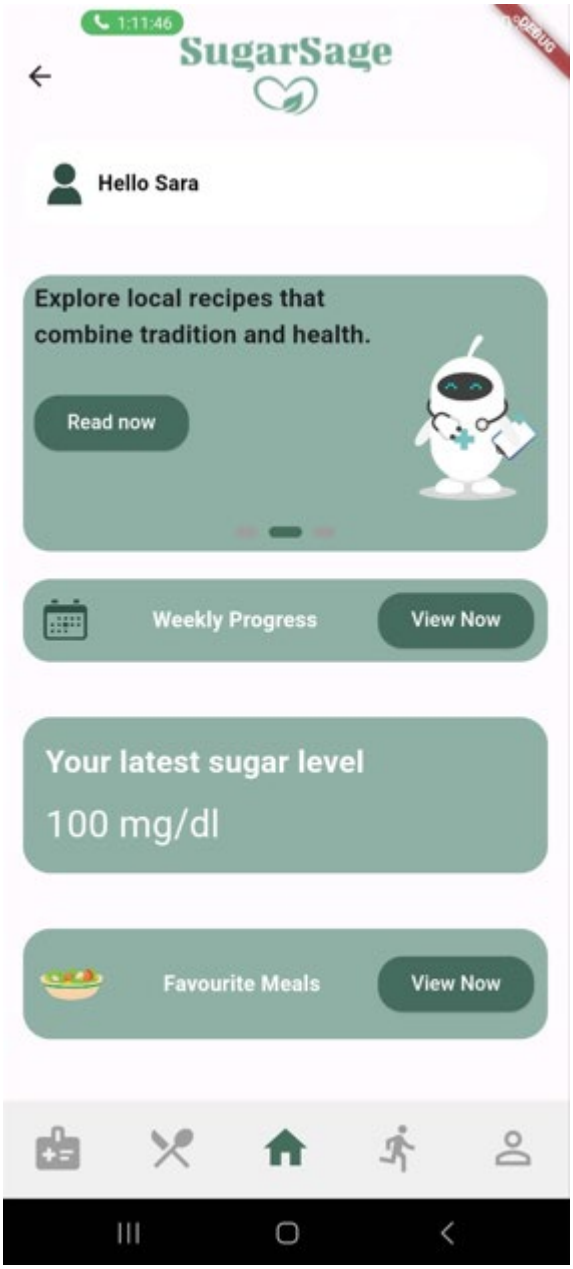

5:10
1:11:17
AGREEMENT
Terms of Service
Last updated on 5/1/2024
1. Acceptance of Terms By creating an account and using SugarSage, you agree to be bound by these terms and conditions ("Terms"). If you do not agree to these Terms, do not use this App.
2. Service Description SugarSage provides dietary plans and monitors blood sugar levels for individuals with diabetes. The App includes features such as personalized meal suggestions, carbohydrate tracking, and blood sugar level logging.
3. No Medical AdviceThe App is not a medical device nor should it be considered medical advice. SugarSage is designed to assist in the management of diabetes and is not a substitute for professional medical advice, diagnosis, or treatment. Always seek the advice of your physician or other qualified health provider with any questions you may have regarding a medical condition.
I agree with all terms and conditions
Continue


1:11:22
SugarSage
Eat Healthy
Maintaining good health should be the primary focus of everyone
Get Started

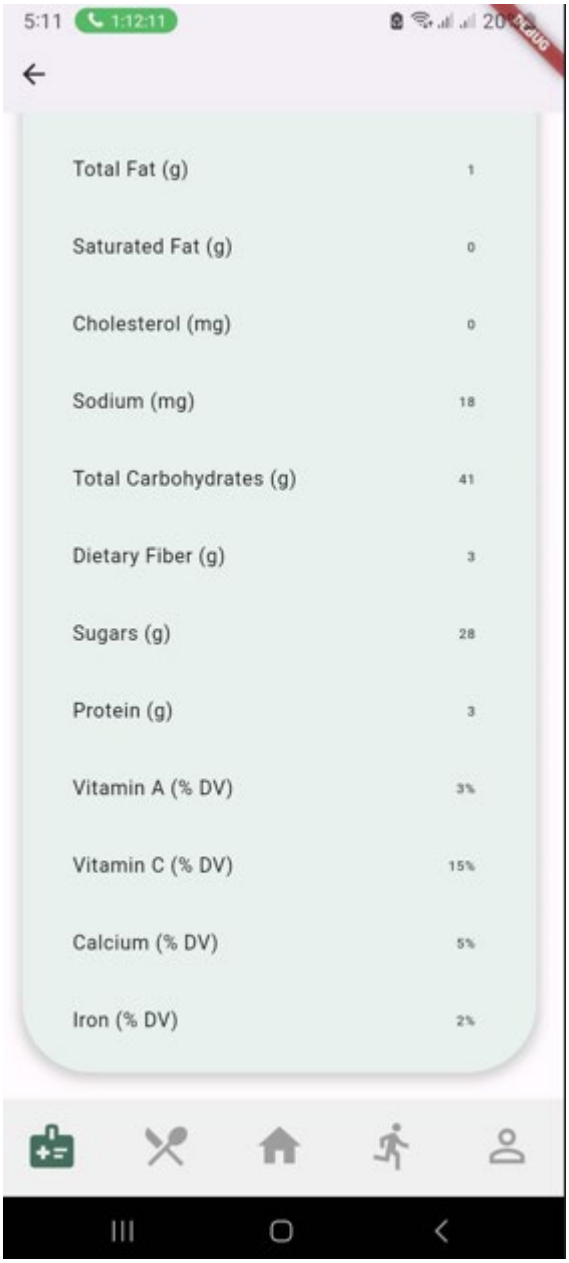

1:11:28
SugarSage
Healthy and Local Recipes
Have access to healthy and local recipes
Get Started

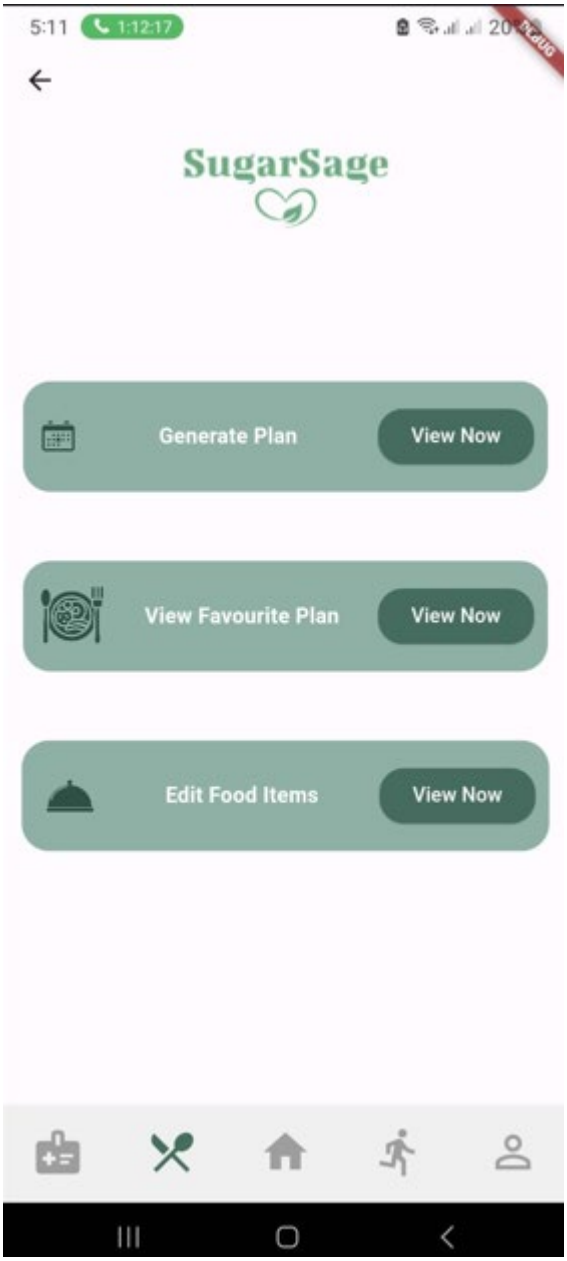

1:11:34
SugarSage
10:34
Track Your Health
With amazing inbuilt tools you can track your progress
Get Started


1:11:41
SugarSage
Hello Sara
All you need to know about Glycemic Index and diabetes.
Read now
Weekly Progress
View Now
Your latest sugar level
100 mg/dl
Favourite Meals
View Now


1:11:46
SugarSage
Hello Sara
Explore local recipes that combine tradition and health.
Read now
Weekly Progress
View Now
Your latest sugar level
100 mg/dl
Favourite Meals
View Now


5:11
1:11:55
SugarSage
Health Profile
Calories (kcal) 135
Calories from Fat (g) 8
Total Fat (g) 1
Saturated Fat (g) 0
Cholesterol (mg) 0
Sodium (mg) 18
Total Carbohydrates (g) 41
Dietary Fiber (g) 3
Sugars (g) 28

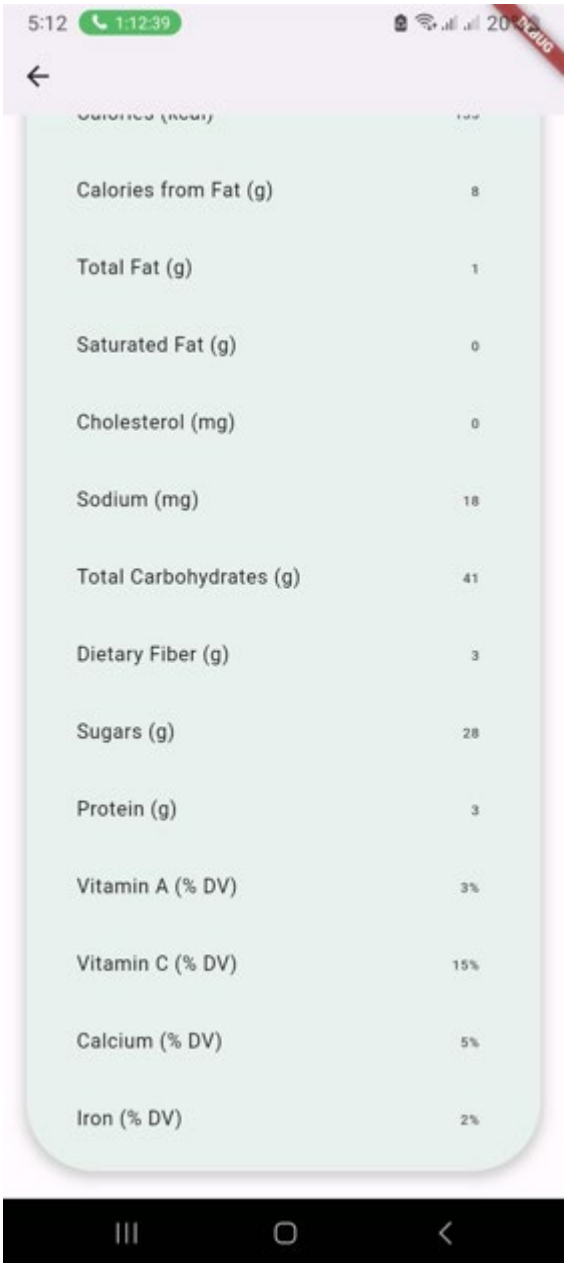

5:11
1:12:11
Total Fat (g) 1
Saturated Fat (g) 0
Cholesterol (mg) 0
Sodium (mg) 18
Total Carbohydrates (g) 41
Dietary Fiber (g) 3
Sugars (g) 28
Protein (g) 3
Vitamin A (% DV) 3%
Vitamin C (% DV) 15%
Calcium (% DV) 5%
Iron (% DV) 2%

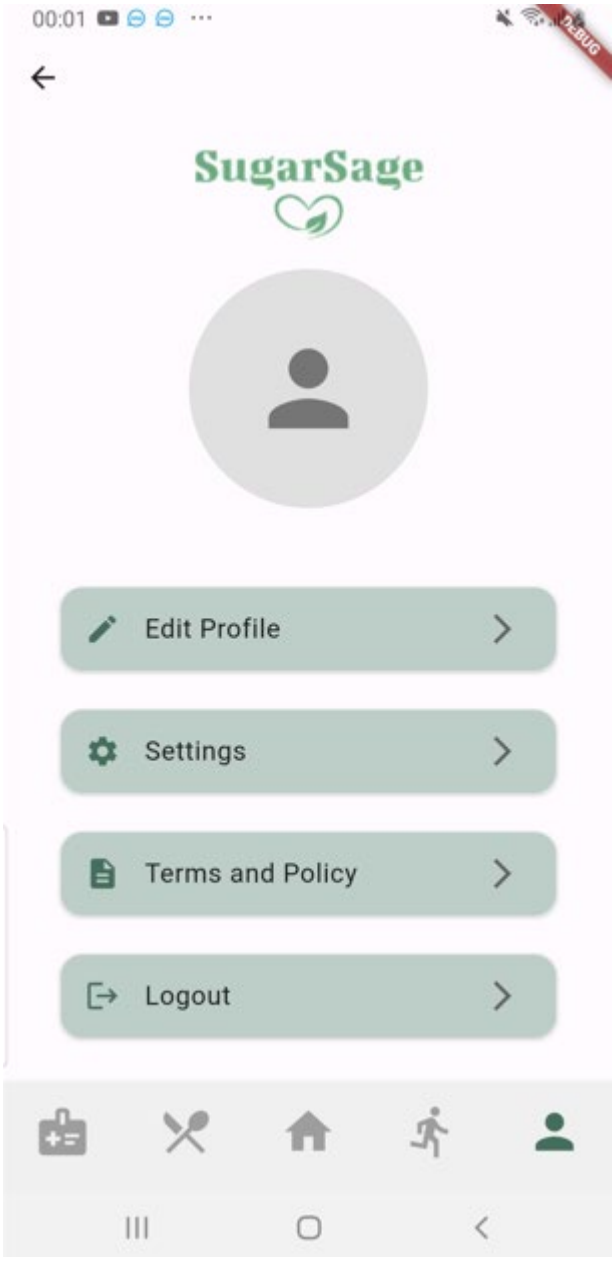

SugarSage
Generate Plan
View Now
View Favourite Plan
View Now
Edit Food Items
View Now

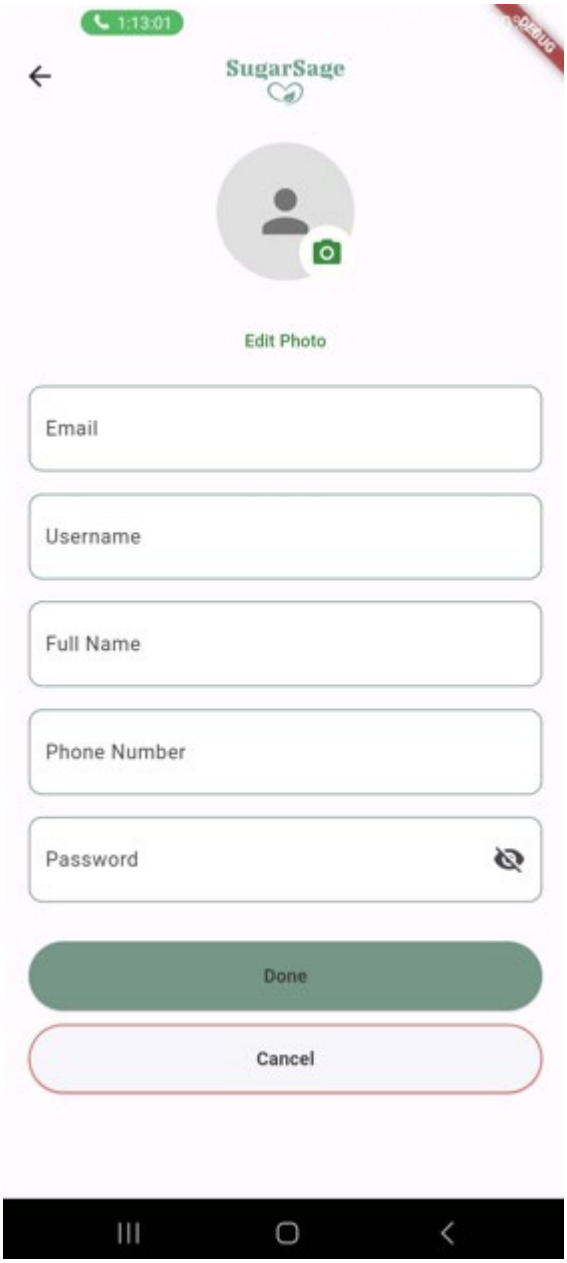

SugarSage
Breakfast
Banana Smoothie
1 medium banana
2 cups milk
1 tbsp peanut butter
Toast
2 slices of bran bread
1 tsp of salted butter
Nutritional Values
Lunch
Grilled Chicken
150g chicken breast
1 tbsp olive oil
Salad
1 cup mixed greens

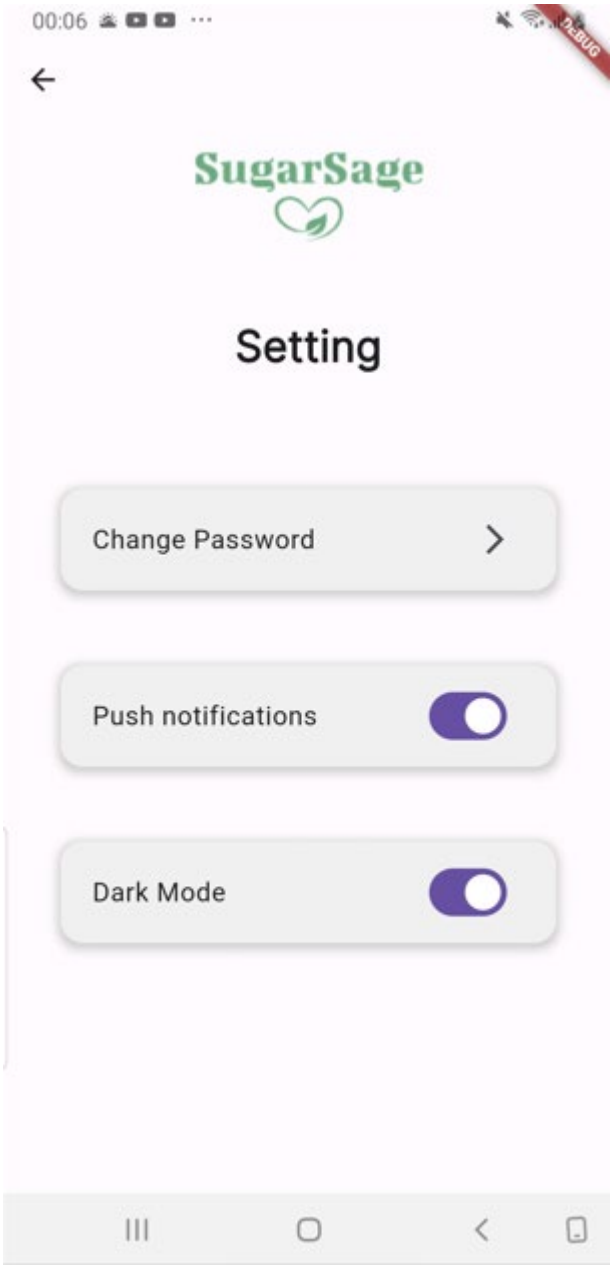

00:02
Lunch
Grilled Chicken
150g chicken breast
1 tbsp olive oil
Salad
1 cup mixed greens
1 tbsp vinaigrette
Nutritional Values
Dinner
Steak
200g sirloin steak
Mashed Potatoes
2 potatoes
1 tbsp butter
Steamed Vegetables
1 cup broccoli
1 tbsp olive oil
Nutritional Values

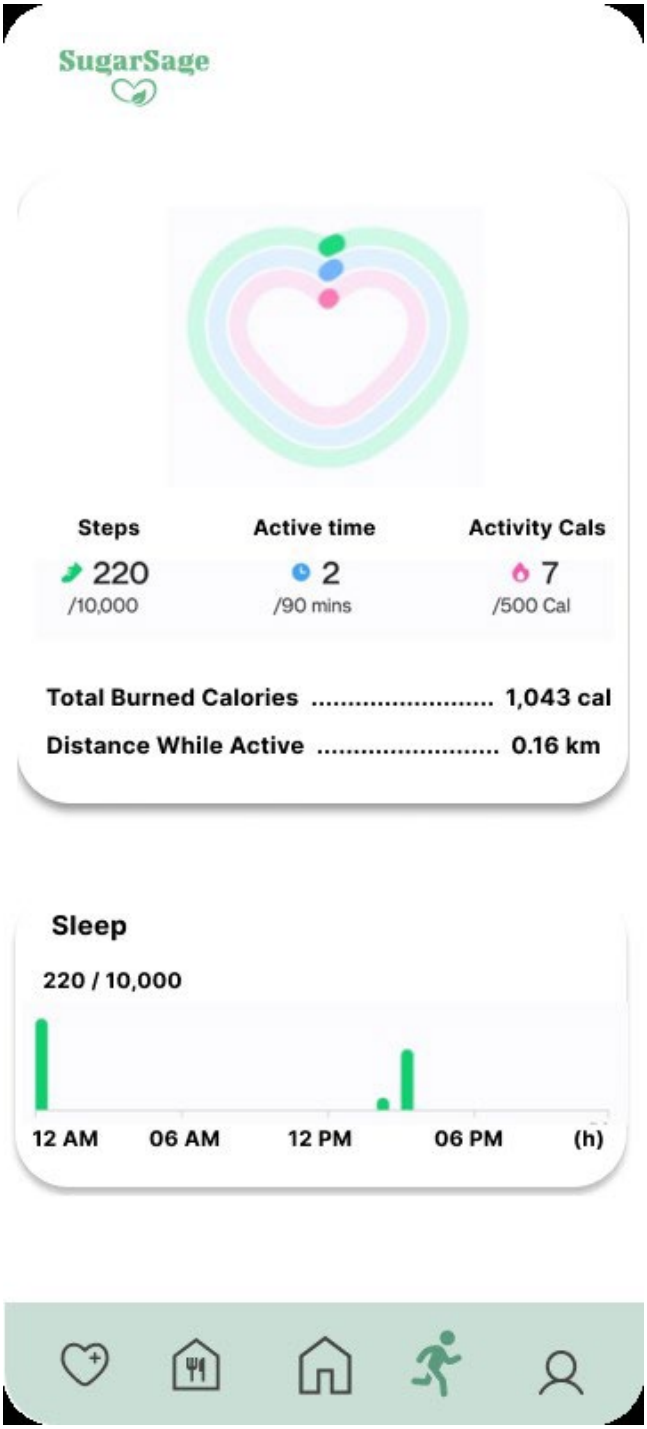

5:12
1:12:34
SugarSage
1. Banana Smoothie
Calories (kcal) 135
Calories from Fat (g) 8
Total Fat (g) 1
Saturated Fat (g) 0
Cholesterol (mg) 0
Sodium (mg) 18
Total Carbohydrates (g) 41
Dietary Fiber (g) 3
Sugars (g) 28
Protein (g) 3

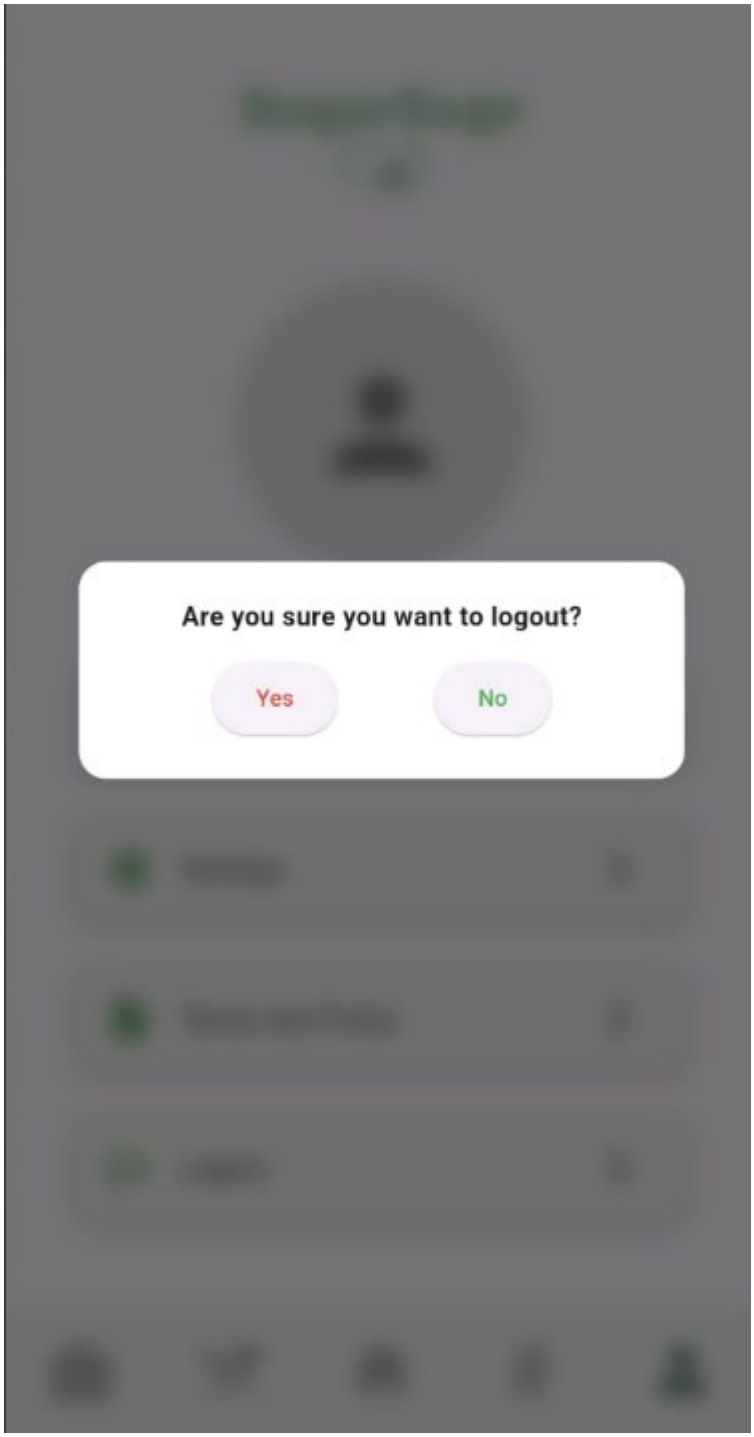

5:12
1:12:39
Calories from Fat (g) 8
Total Fat (g) 1
Saturated Fat (g) 0
Cholesterol (mg) 0
Sodium (mg) 18
Total Carbohydrates (g) 41
Dietary Fiber (g) 3
Sugars (g) 28
Protein (g) 3
Vitamin A (% DV) 3%
Vitamin C (% DV) 15%
Calcium (% DV) 5%
Iron (% DV) 2%


00:01
SugarSage
Edit Profile
Settings
Terms and Policy
Logout

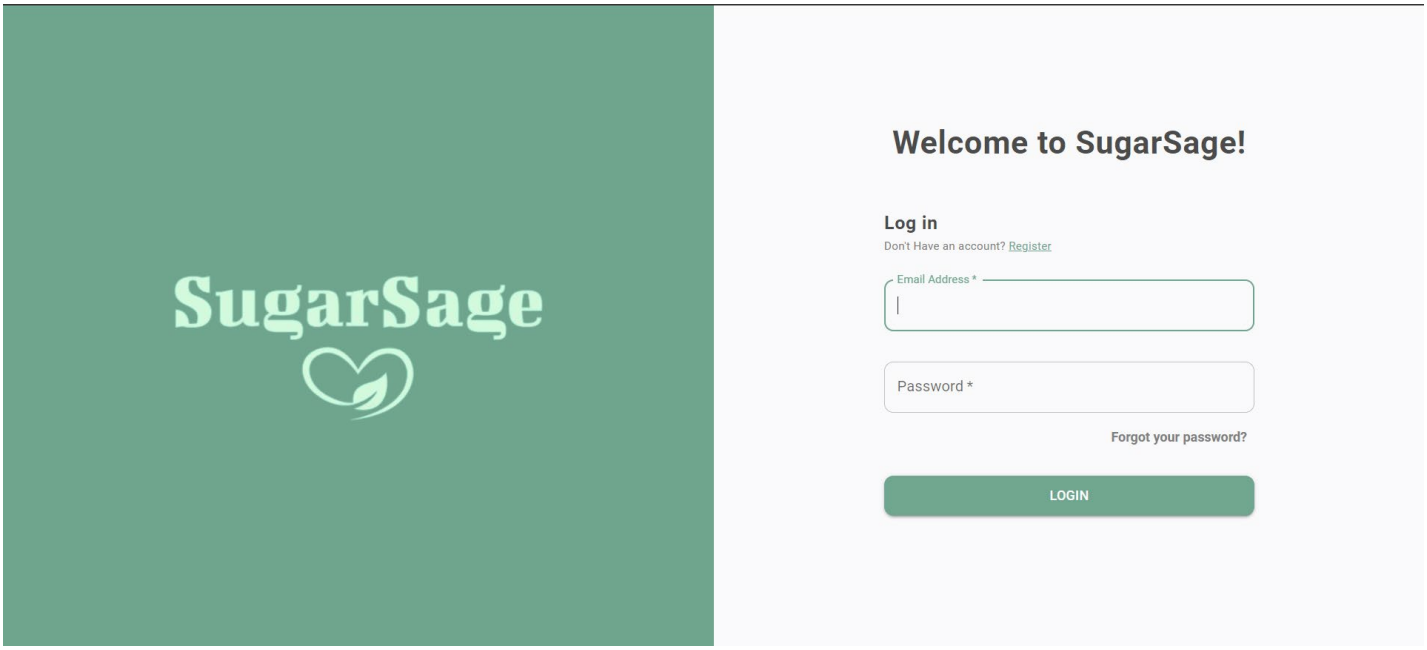

1:13:01
SugarSage
Edit Photo
Email
Username
Full Name
Phone Number
Password
Done
Cancel

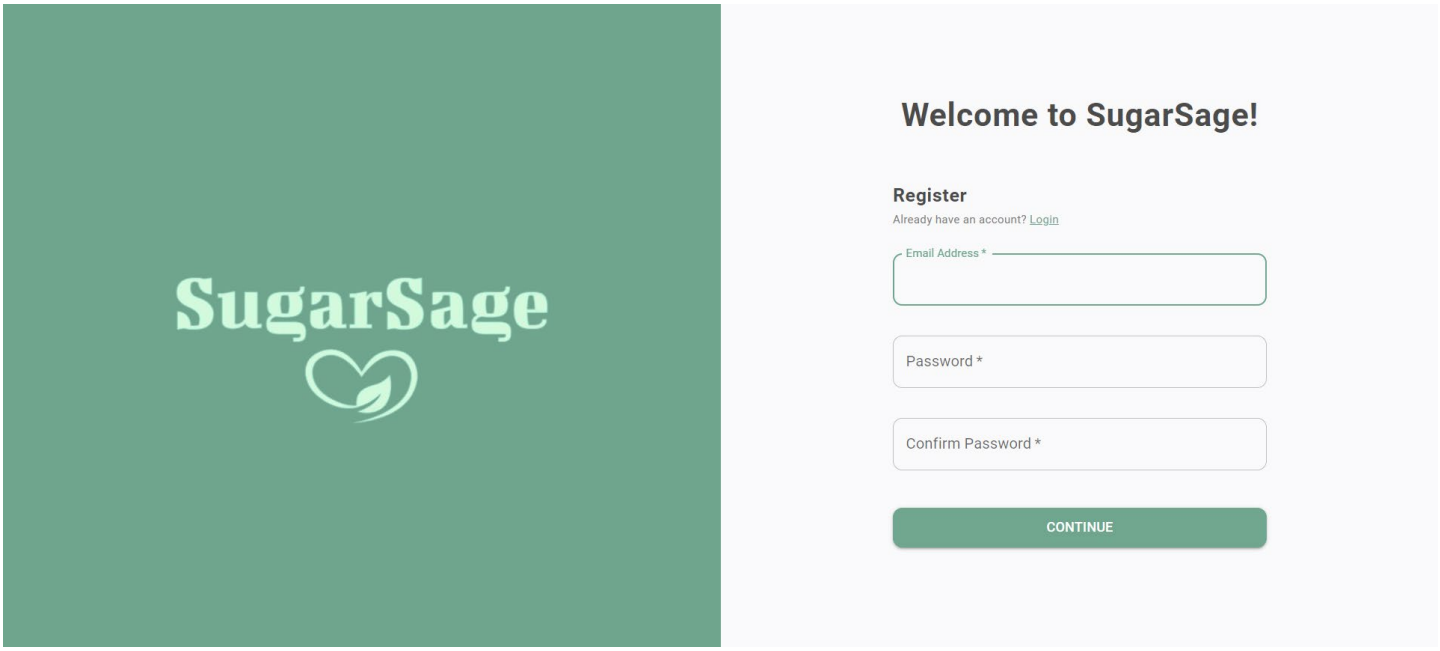

00:06
SugarSage
Setting
Change Password
Push notifications
Dark Mode


SugarSage
Steps
Active time
Activity Cals
220
/10,000
2
/90 mins
7
/500 Cal
Total Burned Calories ..... 1,043 cal
Distance While Active ..... 0.16 km
Sleep
220 / 10,000
12 AM
06 AM
12 PM
06 PM
(h)

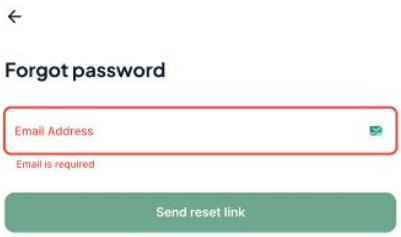

Are you sure you want to logout?
Yes
No

4.2.2 Web App Screens:

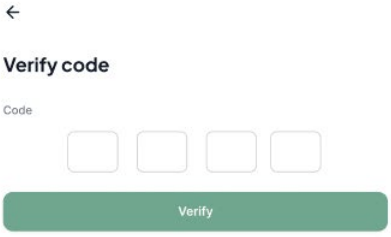




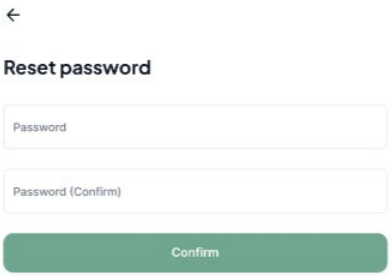

SugarSage


Forgot password
Email Address
Email is required
Send reset link

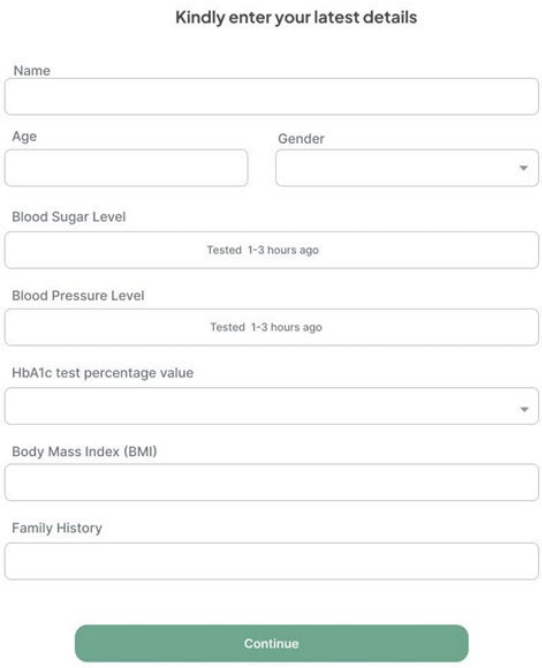

SugarSage

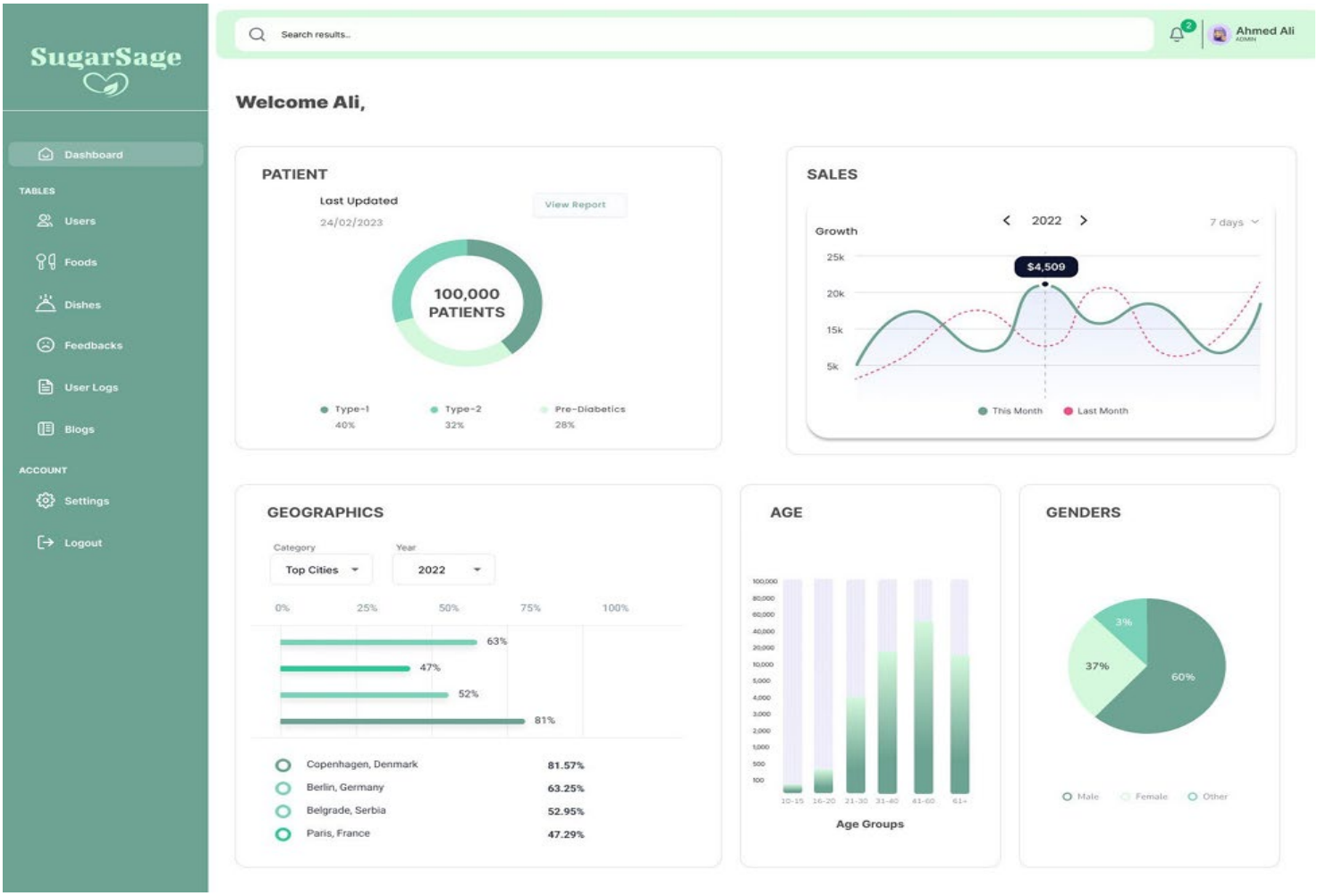

Verify code
Code
Verify

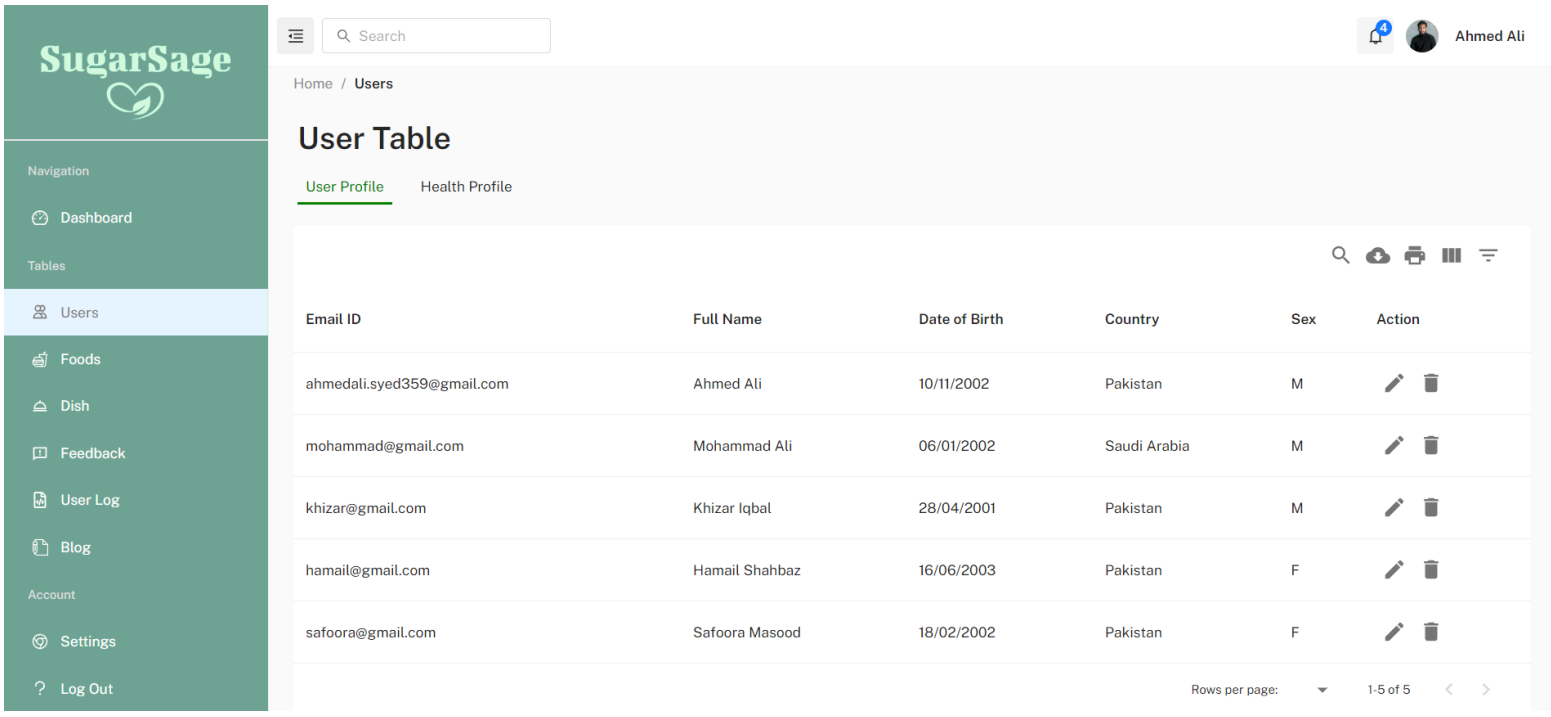

SugarSage

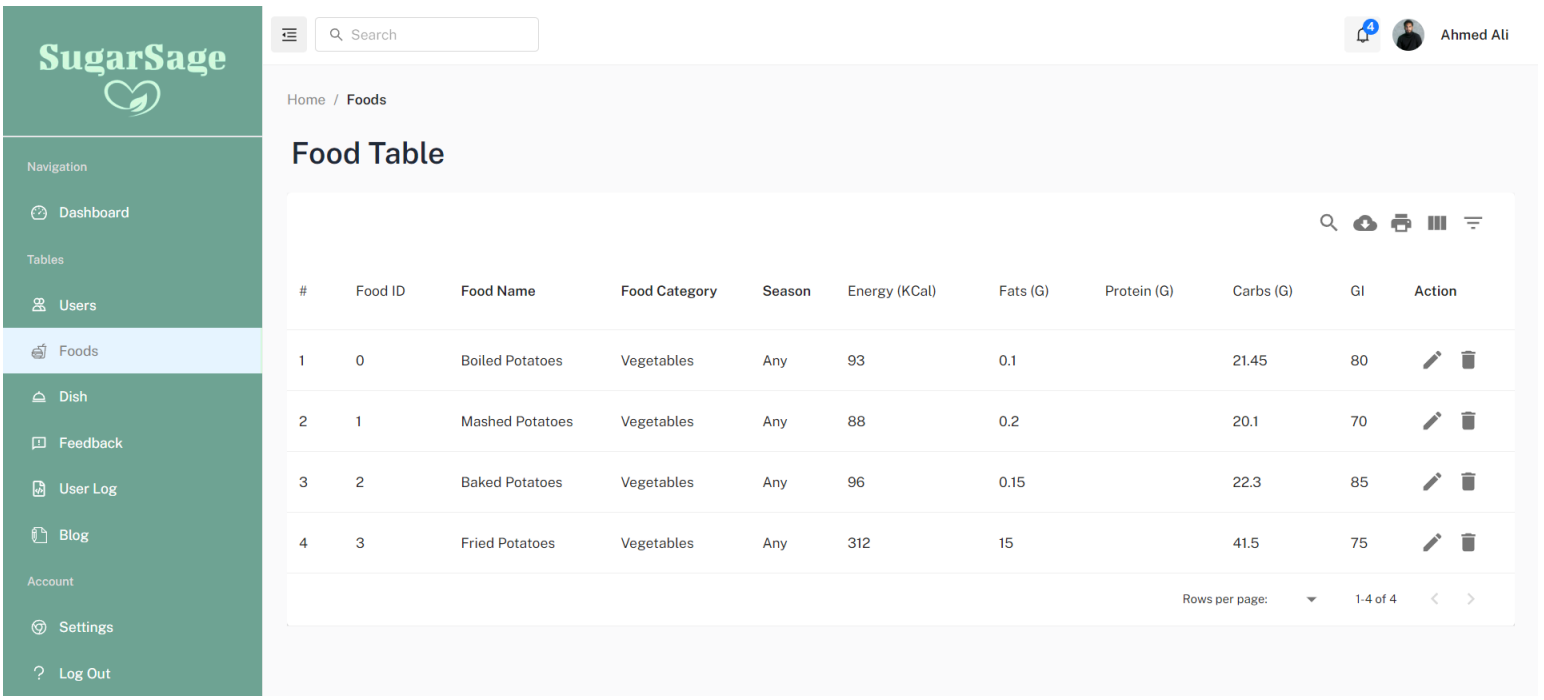

Reset password
Password
Password (Confirm)
Confirm

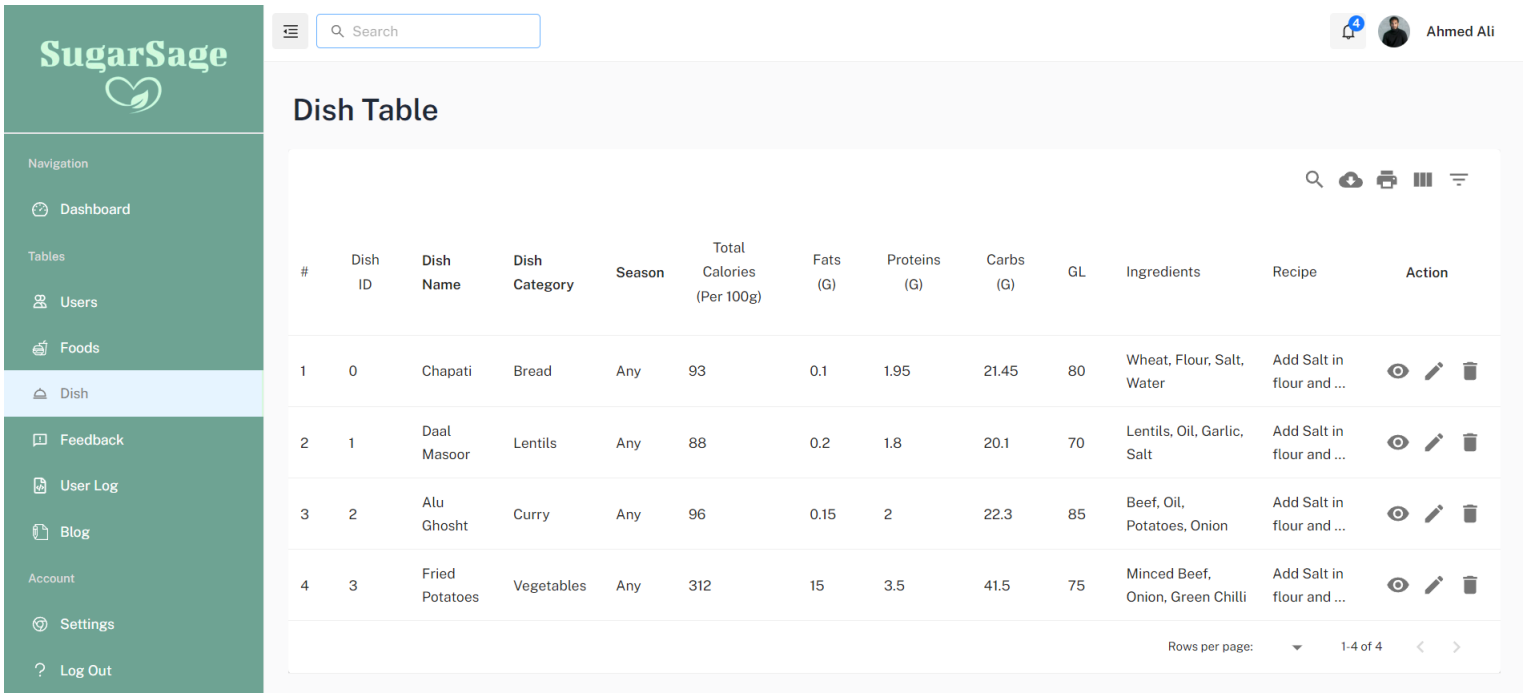

SugarSage

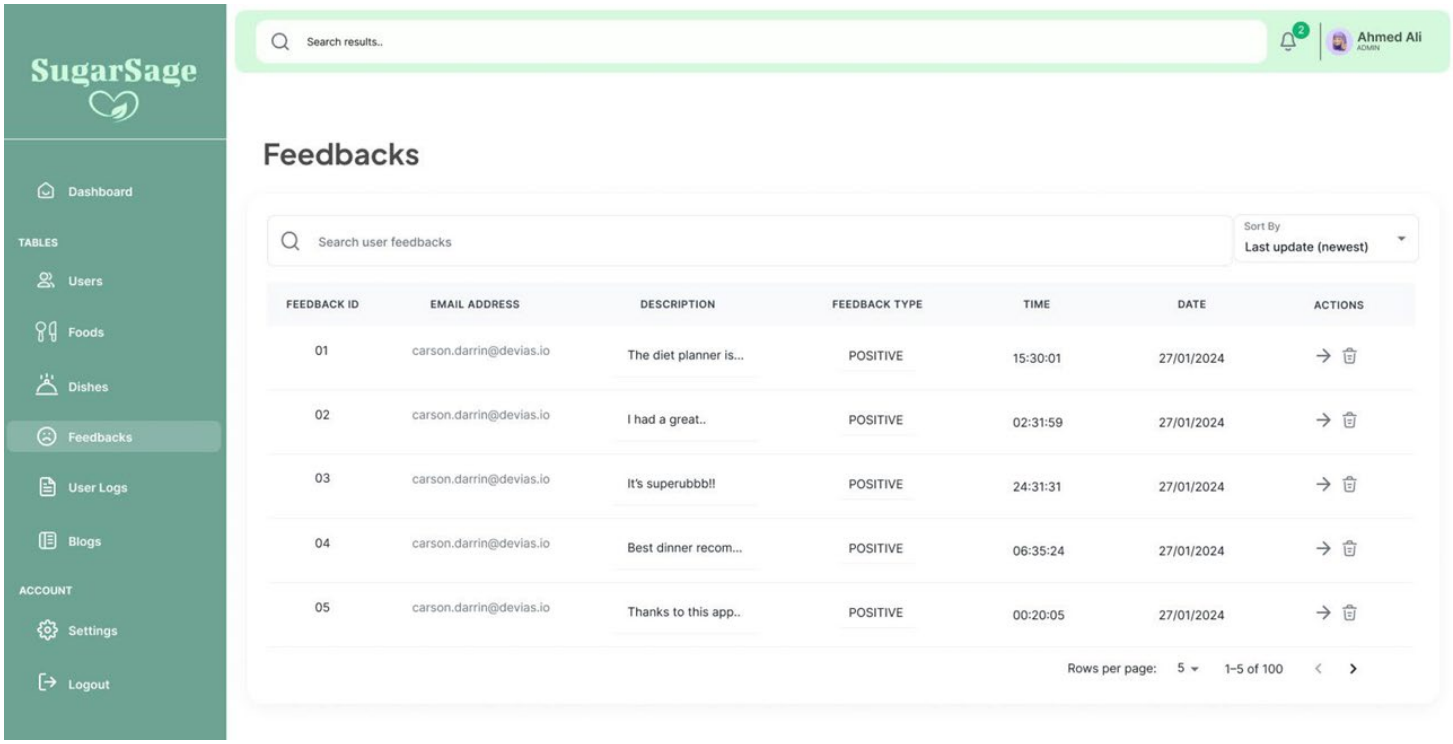

Kindly enter your latest details
Name
Age
Gender
Blood Sugar Level
Tested 1-3 hours ago
Blood Pressure Level
Tested 1-3 hours ago
HbA1c test percentage value
Body Mass Index (BMI)
Family History
Continue

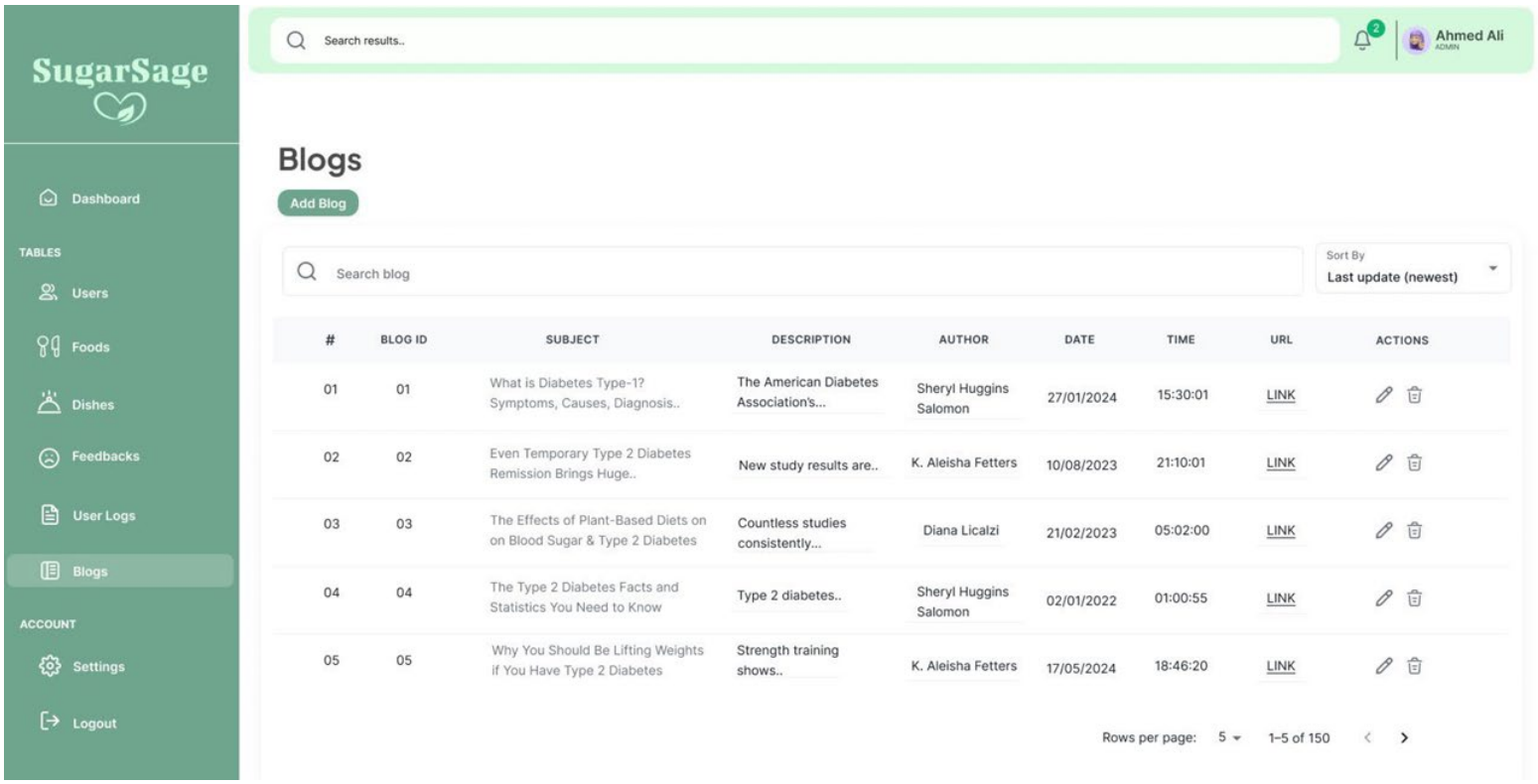

SugarSage
Search results..
Ahmed Ali
ADMIN
Welcome Ali,
Dashboard
TABLES
Users
Foods
Dishes
Feedbacks
User Logs
Blogs
ACCOUNT
Settings
Logout
PATIENT
Last Updated
24/02/2023
View Report
100,000
PATIENTS
Type-1
40%
Type-2
32%
Pre-Diabetics
28%
SALES
Growth
2022
7 days
25k
20k
15k
5k
$4,509
This Month
Last Month
GEOGRAPHICS
Category
Top Cities
Year
2022
0%
25%
50%
75%
100%
63%
47%
52%
81%
Copenhagen, Denmark
81.57%
Berlin, Germany
63.25%
Belgrade, Serbia
52.95%
Paris, France
47.29%
AGE
Age Groups
GENDERS
3%
37%
60%
Male
Female
Other

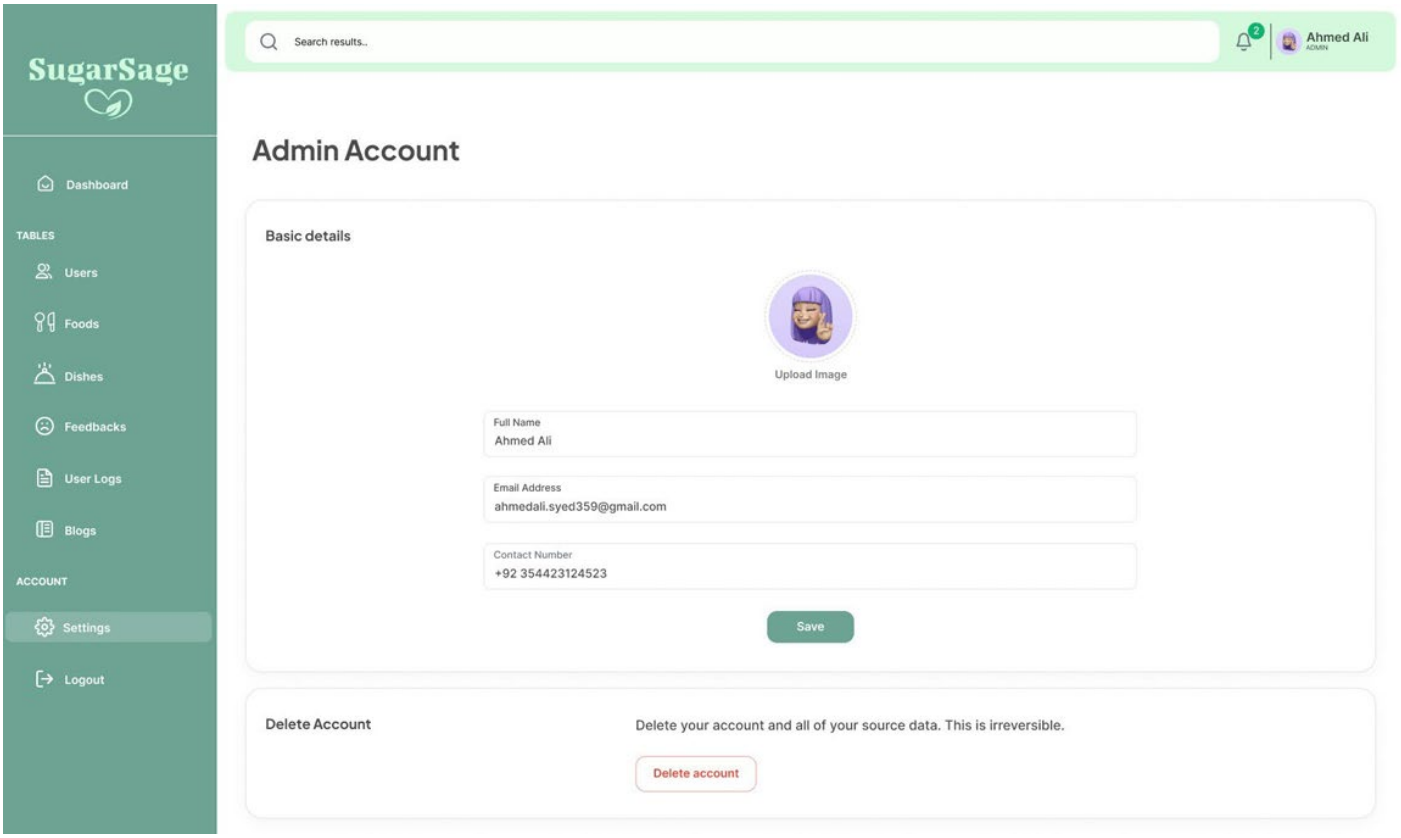

SugarSage
Search
Ahmed Ali
Home / Users
User Table
User Profile
Health Profile
Navigation
Dashboard
Tables
Users
Foods
Dish
Feedback
User Log
Blog
Account
Settings
Log Out
Email ID
Full Name
Date of Birth
Country
Sex
Action
ahmedali.syed359@gmail.com
Ahmed Ali
10/11/2002
Pakistan
M
mohammad@gmail.com
Mohammad Ali
06/01/2002
Saudi Arabia
M
khizar@gmail.com
Khizar Iqbal
28/04/2001
Pakistan
M
hamail@gmail.com
Hamail Shahbaz
16/06/2003
Pakistan
F
safoora@gmail.com
Safoora Masood
18/02/2002
Pakistan
F
Rows per page:
1-5 of 5

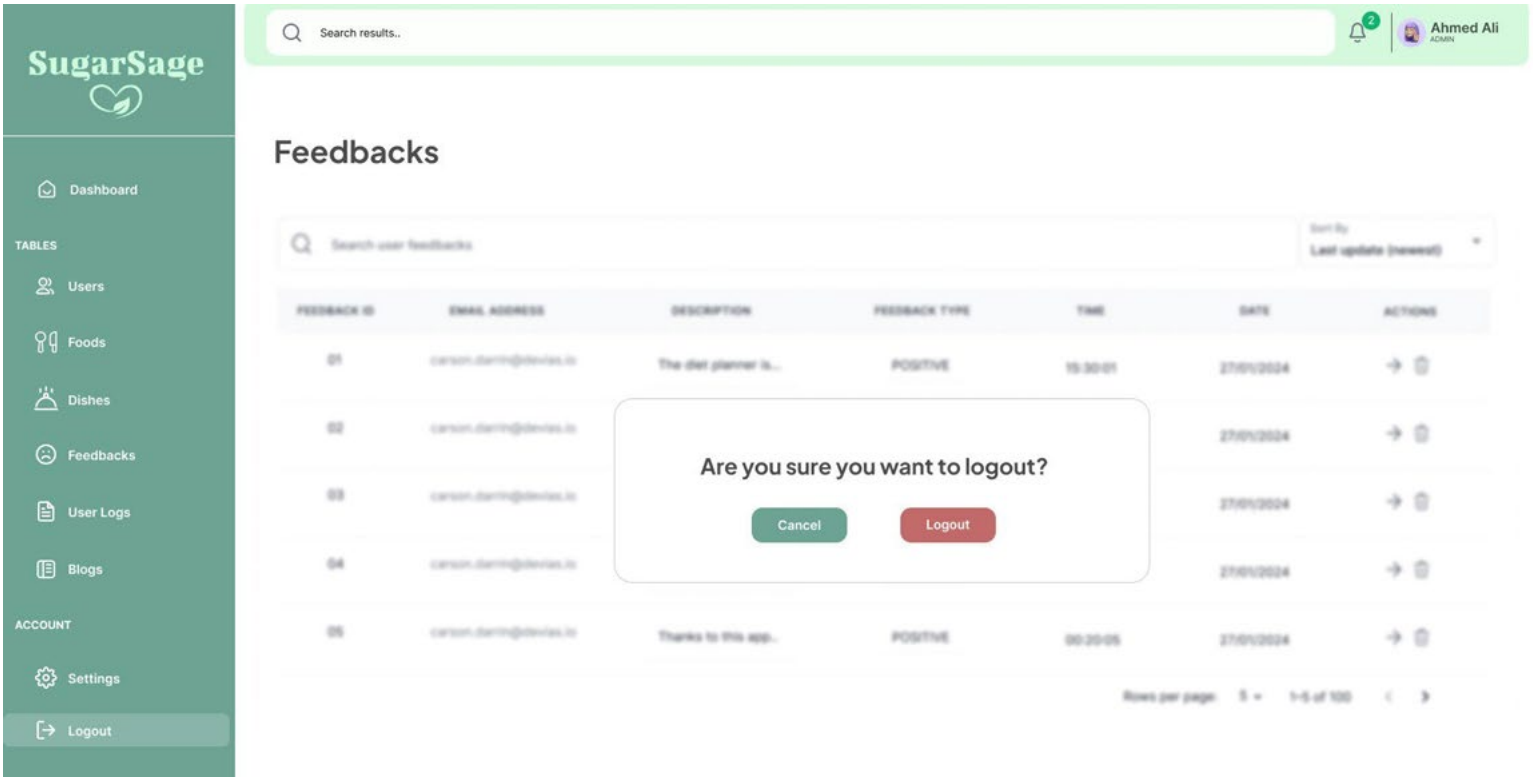

SugarSage
Search
Ahmed Ali
Navigation
Dashboard
Tables
Users
Foods
Dish
Feedback
User Log
Blog
Account
Settings
Log Out
Home / Foods
Food Table
Food ID Food Name Food Category Season Energy (KCal) Fats (G) Protein (G) Carbs (G) GI Action
1 0 Boiled Potatoes Vegetables Any 93 0.1 21.45 80
2 1 Mashed Potatoes Vegetables Any 88 0.2 20.1 70
3 2 Baked Potatoes Vegetables Any 96 0.15 22.3 85
4 3 Fried Potatoes Vegetables Any 312 15 41.5 75
Rows per page:
1-4 of 4

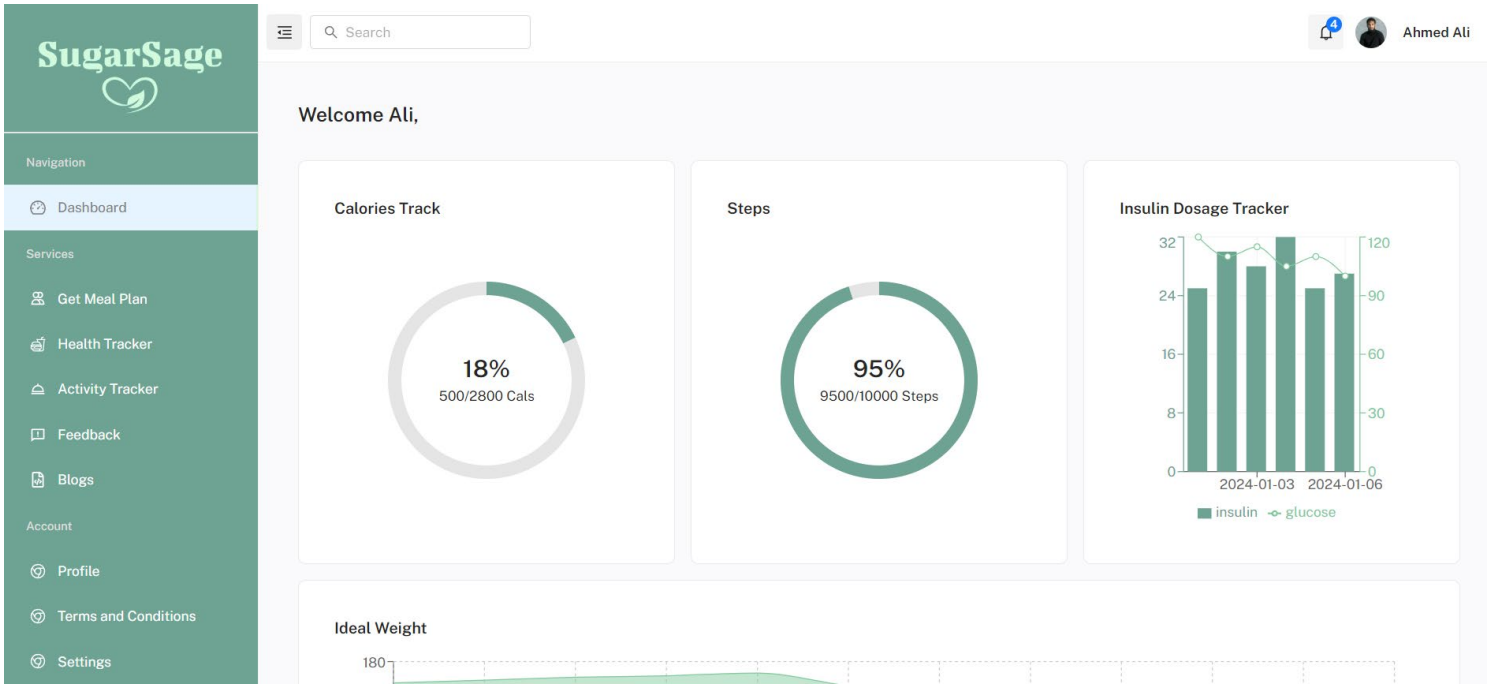

SugarSage
Search
Ahmed Ali
Navigation
Dashboard
Tables
Users
Foods
Dish
Feedback
User Log
Blog
Account
Settings
Log Out
Dish Table
Dish ID Dish Name Dish Category Season Total Calories (Per 100g) Fats (G) Proteins (G) Carbs (G) GL Ingredients Recipe Action
1 0 Chapati Bread Any 93 0.1 1.95 21.45 80 Wheat, Flour, Salt, Water Add Salt in flour and ...
2 1 Daal Masoor Lentils Any 88 0.2 1.8 20.1 70 Lentils, Oil, Garlic, Salt Add Salt in flour and ...
3 2 Alu Ghosht Curry Any 96 0.15 2 22.3 85 Beef, Oil, Potatoes, Onion Add Salt in flour and ...
4 3 Fried Potatoes Vegetables Any 312 15 3.5 41.5 75 Minced Beef, Onion, Green Chilli Add Salt in flour and ...
Rows per page:
1-4 of 4

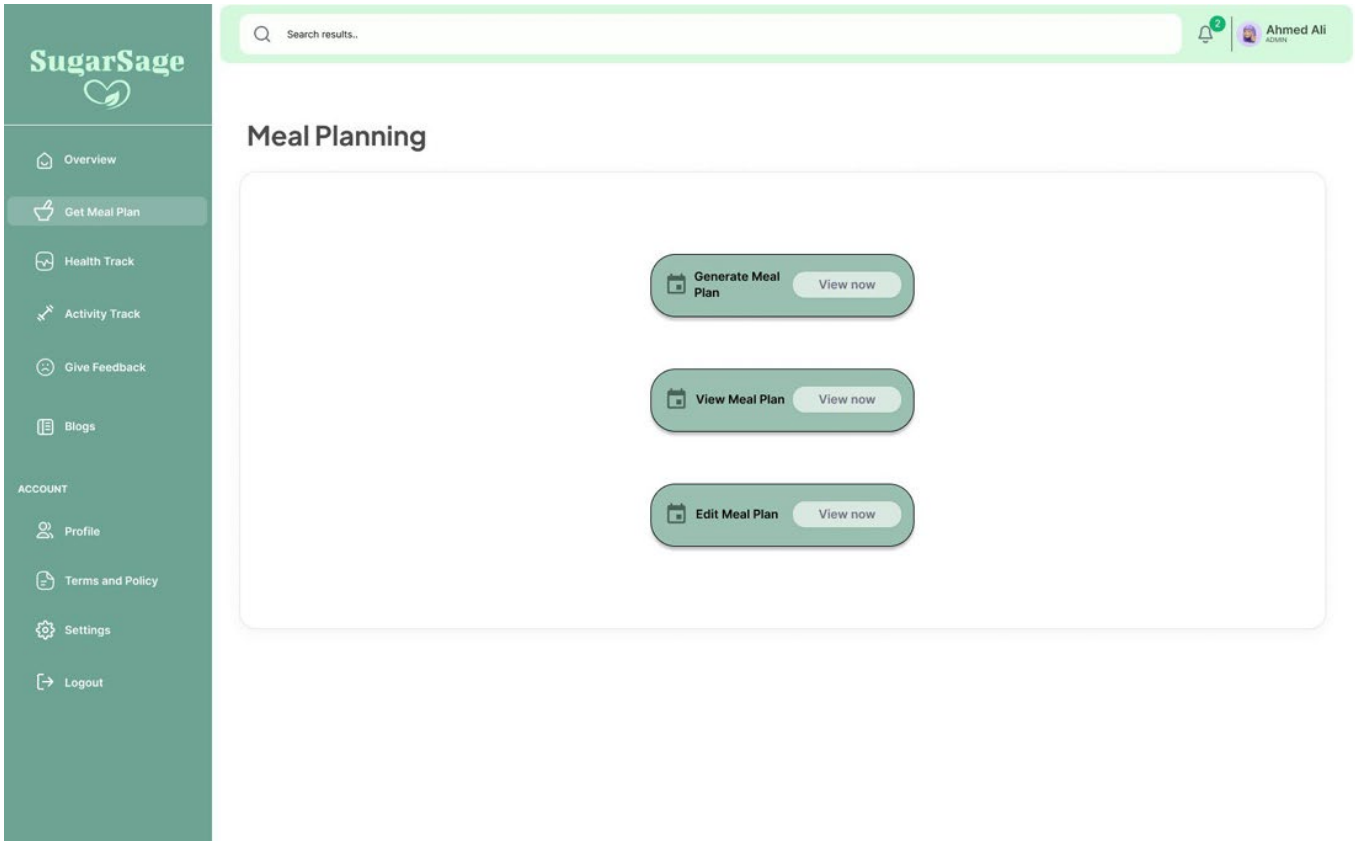

SugarSage
Search results..
Ahmed Ali
ADMIN
Dashboard
TABLES
Users
Foods
Dishes
Feedbacks
User Logs
Blogs
ACCOUNT
Settings
Logout
Feedbacks
Search user feedbacks
Sort By
Last update (newest)
FEEDBACK ID
EMAIL ADDRESS
DESCRIPTION
FEEDBACK TYPE
TIME
DATE
ACTIONS
01 carson.darrin@devias.io The diet planner is... POSITIVE 15:30:01 27/01/2024
02 carson.darrin@devias.io I had a great.. POSITIVE 02:31:59 27/01/2024
03 carson.darrin@devias.io It's superubbb!! POSITIVE 24:31:31 27/01/2024
04 carson.darrin@devias.io Best dinner recom... POSITIVE 06:35:24 27/01/2024
05 carson.darrin@devias.io Thanks to this app.. POSITIVE 00:20:05 27/01/2024
Rows per page: 5
1–5 of 100

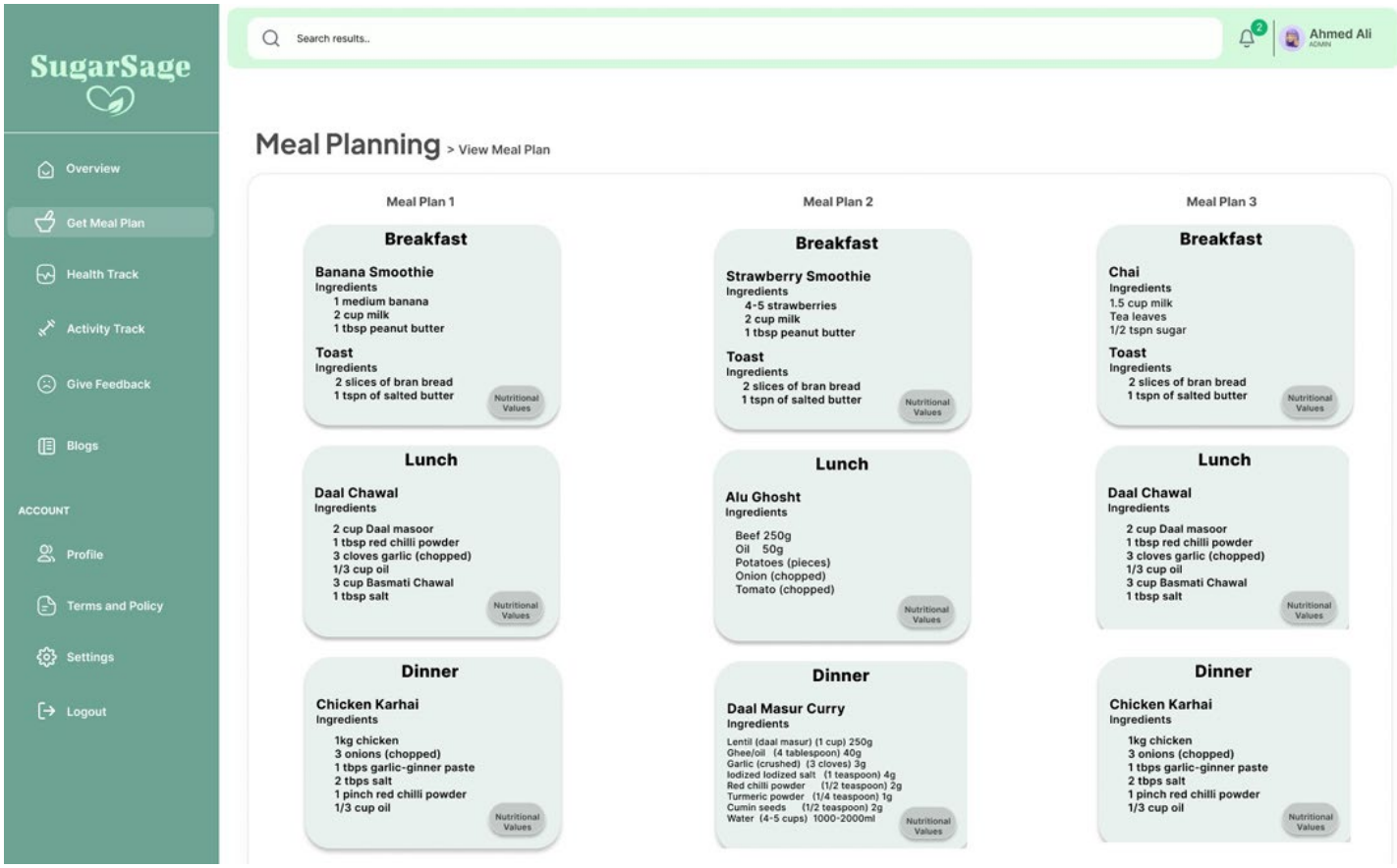

SugarSage
Search results..
Ahmed Ali
ADMIN
Dashboard
TABLES
Users
Foods
Dishes
Feedbacks
User Logs
Blogs
ACCOUNT
Settings
Logout
Blogs
Add Blog
Search blog
Sort By
Last update (newest)
BLOG ID SUBJECT DESCRIPTION AUTHOR DATE TIME URL ACTIONS
01 01 What is Diabetes Type-1? Symptoms, Causes, Diagnosis.. The American Diabetes Association's... Sheryl Huggins Salomon 27/01/2024 15:30:01 LINK
02 02 Even Temporary Type 2 Diabetes Remission Brings Huge.. New study results are.. K. Aleisha Fetters 10/08/2023 21:10:01 LINK
03 03 The Effects of Plant-Based Diets on on Blood Sugar & Type 2 Diabetes Countless studies consistently... Diana Licalzi 21/02/2023 05:02:00 LINK
04 04 The Type 2 Diabetes Facts and Statistics You Need to Know Type 2 diabetes.. Sheryl Huggins Salomon 02/01/2022 01:00:55 LINK
05 05 Why You Should Be Lifting Weights if You Have Type 2 Diabetes Strength training shows.. K. Aleisha Fetters 17/05/2024 18:46:20 LINK
Rows per page: 5
1–5 of 150

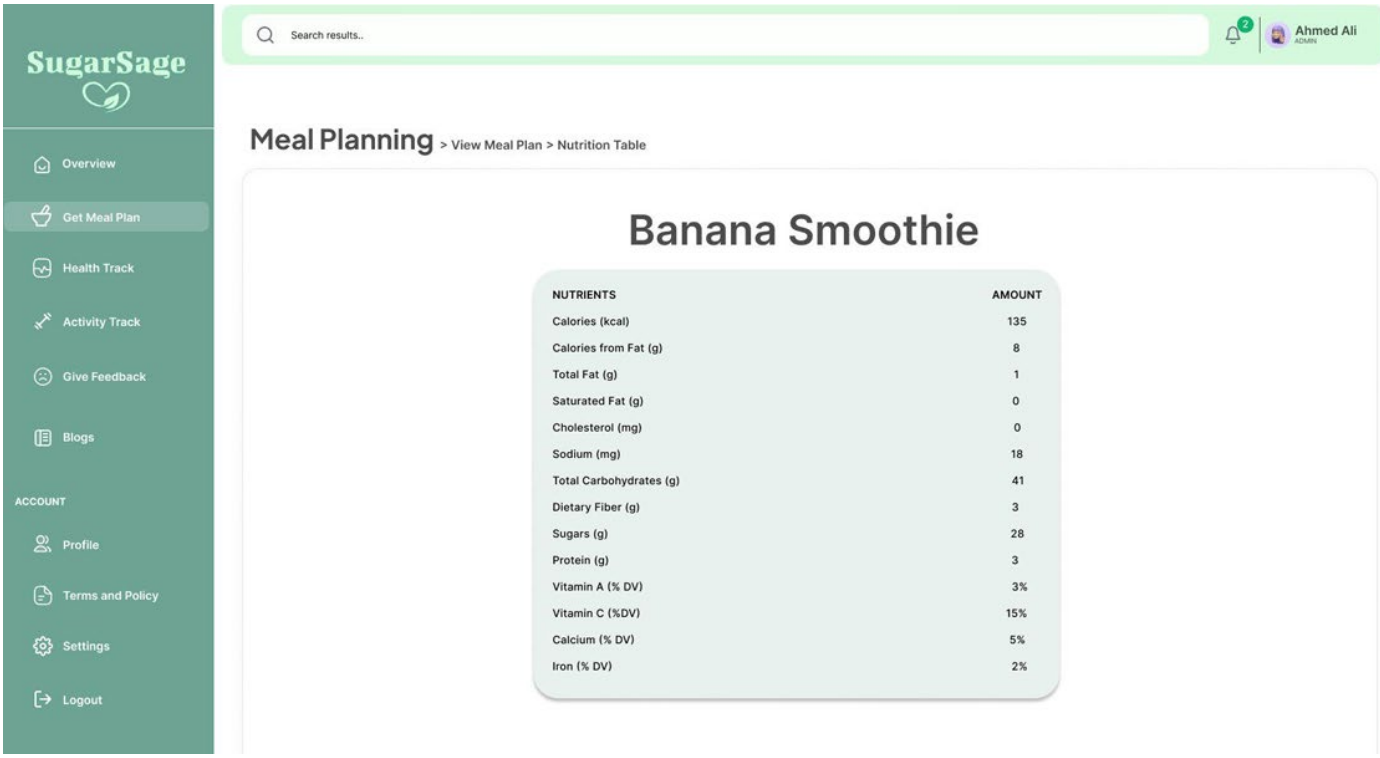

SugarSage
Search results..
Ahmed Ali
ADMIN
Dashboard
TABLES
Users
Foods
Dishes
Feedbacks
User Logs
Blogs
ACCOUNT
Settings
Logout
Admin Account
Basic details
Upload Image
Full Name
Ahmed Ali
Email Address
ahmedali.syed359@gmail.com
Contact Number
+92 354423124523
Save
Delete Account
Delete your account and all of your source data. This is irreversible.
Delete account

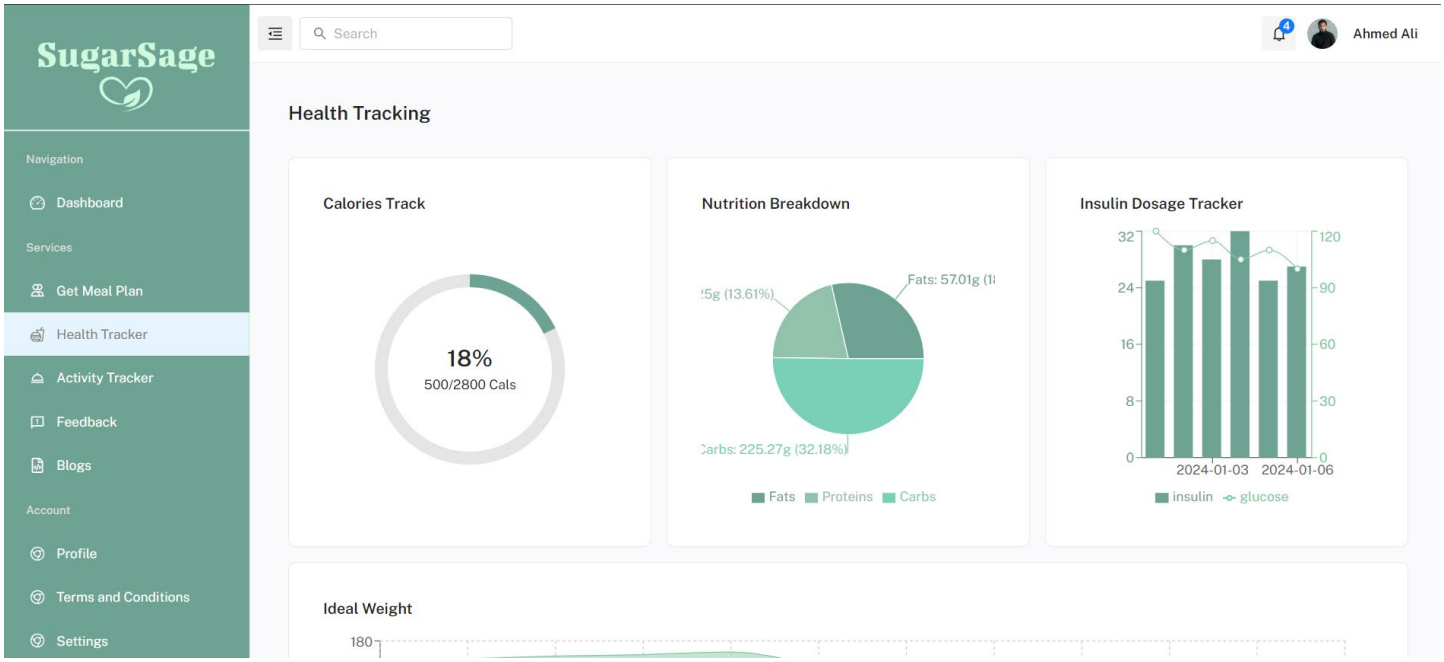

SugarSage
Search results..
Ahmed Ali
ADMIN
Dashboard
TABLES
Users
Foods
Dishes
Feedbacks
User Logs
Blogs
ACCOUNT
Settings
Logout
Feedbacks
Are you sure you want to logout?
Cancel
Logout

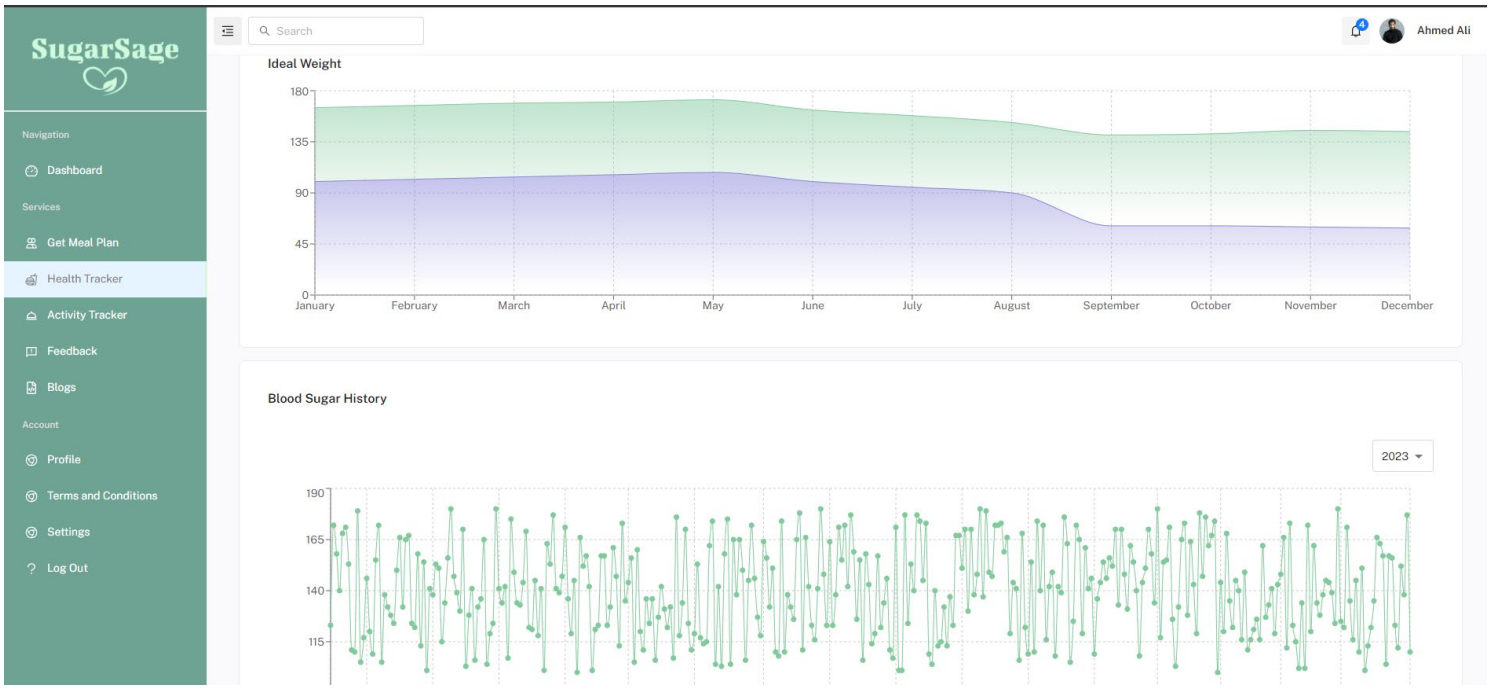

SugarSage
Search
Ahmed Ali
Welcome Ali,
Navigation
Dashboard
Services
Get Meal Plan
Health Tracker
Activity Tracker
Feedback
Blogs
Account
Profile
Terms and Conditions
Settings
Calories Track
18%
500/2800 Cals
Steps
95%
9500/10000 Steps
Insulin Dosage Tracker
32
24
16
8
0
120
90
60
30
0
2024-01-03
2024-01-06
insulin
glucose
Ideal Weight
180


SugarSage
Search results..
Ahmed Ali
Meal Planning
Overview
Get Meal Plan
Health Track
Activity Track
Give Feedback
Blogs
ACCOUNT
Profile
Terms and Policy
Settings
Logout
Generate Meal Plan
View now
View Meal Plan
View now
Edit Meal Plan
View now

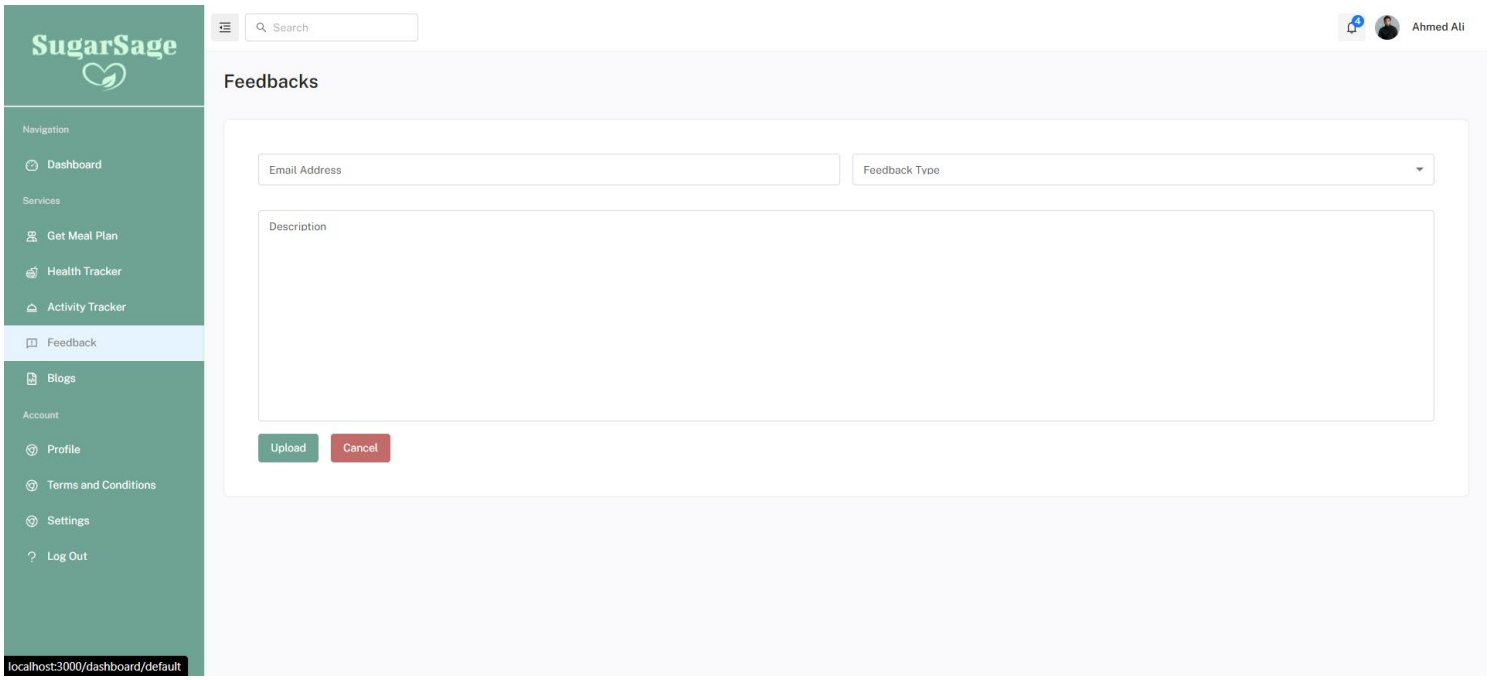

SugarSage
Search results..
Ahmed Ali
ADMIN
Overview
Get Meal Plan
Health Track
Activity Track
Give Feedback
Blogs
ACCOUNT
Profile
Terms and Policy
Settings
Logout
Meal Planning > View Meal Plan
Meal Plan 1
Breakfast
Banana Smoothie
Ingredients
1 medium banana
2 cup milk
1 tbsp peanut butter
Toast
Ingredients
2 slices of bran bread
1 tspn of salted butter
Nutritional Values
Lunch
Daal Chawal
Ingredients
2 cup Daal masoor
1 tbsp red chilli powder
3 cloves garlic (chopped)
1/3 cup oil
3 cup Basmati Chawal
1 tbsp salt
Nutritional Values
Dinner
Chicken Karhai
Ingredients
1kg chicken
3 onions (chopped)
1 tbps garlic-ginner paste
2 tbps salt
1 pinch red chilli powder
1/3 cup oil
Nutritional Values
Meal Plan 2
Breakfast
Strawberry Smoothie
Ingredients
4-5 strawberries
2 cup milk
1 tbsp peanut butter
Toast
Ingredients
2 slices of bran bread
1 tspn of salted butter
Nutritional Values
Lunch
Alu Ghosht
Ingredients
Beef 250g
Oil 50g
Potatoes (pieces)
Onion (chopped)
Tomato (chopped)
Nutritional Values
Dinner
Daal Masur Curry
Ingredients
Lentil (daal masur) (1 cup) 250g
Ghee/oil (4 tablespoon) 40g
Garlic (crushed) (3 cloves) 3g
Iodized Iodized salt (1 teaspoon) 4g
Red chilli powder (1/2 teaspoon) 2g
Turmeric powder (1/4 teaspoon) 1g
Cumin seeds (1/2 teaspoon) 2g
Water (4-5 cups) 1000-2000ml
Nutritional Values
Meal Plan 3
Breakfast
Chai
Ingredients
1.5 cup milk
Tea leaves
1/2 tspn sugar
Toast
Ingredients
2 slices of bran bread
1 tspn of salted butter
Nutritional Values
Lunch
Daal Chawal
Ingredients
2 cup Daal masoor
1 tbsp red chilli powder
3 cloves garlic (chopped)
1/3 cup oil
3 cup Basmati Chawal
1 tbsp salt
Nutritional Values
Dinner
Chicken Karhai
Ingredients
1kg chicken
3 onions (chopped)
1 tbps garlic-ginner paste
2 tbps salt
1 pinch red chilli powder
1/3 cup oil
Nutritional Values

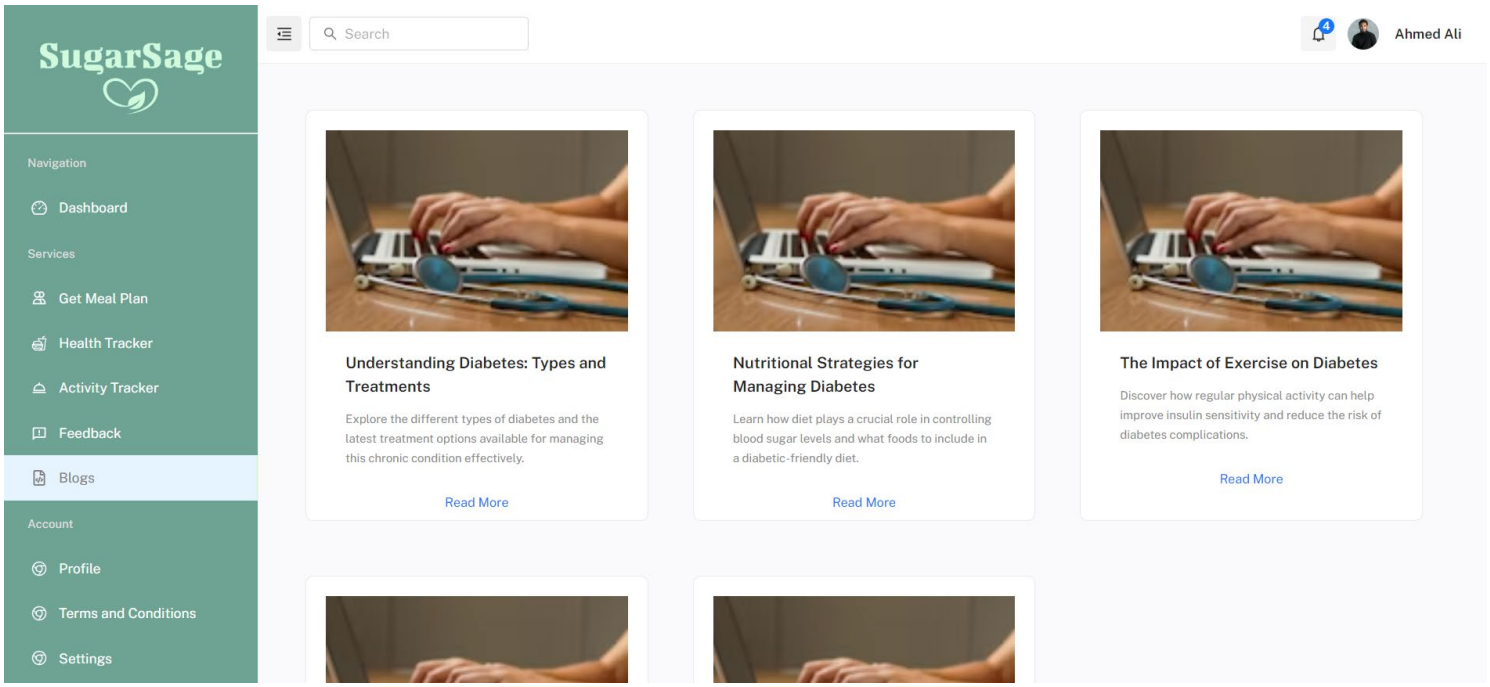

SugarSage
Search results..
Ahmed Ali
ADMIN
Overview
Get Meal Plan
Health Track
Activity Track
Give Feedback
Blogs
ACCOUNT
Profile
Terms and Policy
Settings
Logout
Meal Planning > View Meal Plan > Nutrition Table
Banana Smoothie
NUTRIENTS AMOUNT
Calories (kcal) 135
Calories from Fat (g) 8
Total Fat (g) 1
Saturated Fat (g) 0
Cholesterol (mg) 0
Sodium (mg) 18
Total Carbohydrates (g) 41
Dietary Fiber (g) 3
Sugars (g) 28
Protein (g) 3
Vitamin A (% DV) 3%
Vitamin C (%DV) 15%
Calcium (% DV) 5%
Iron (% DV) 2%

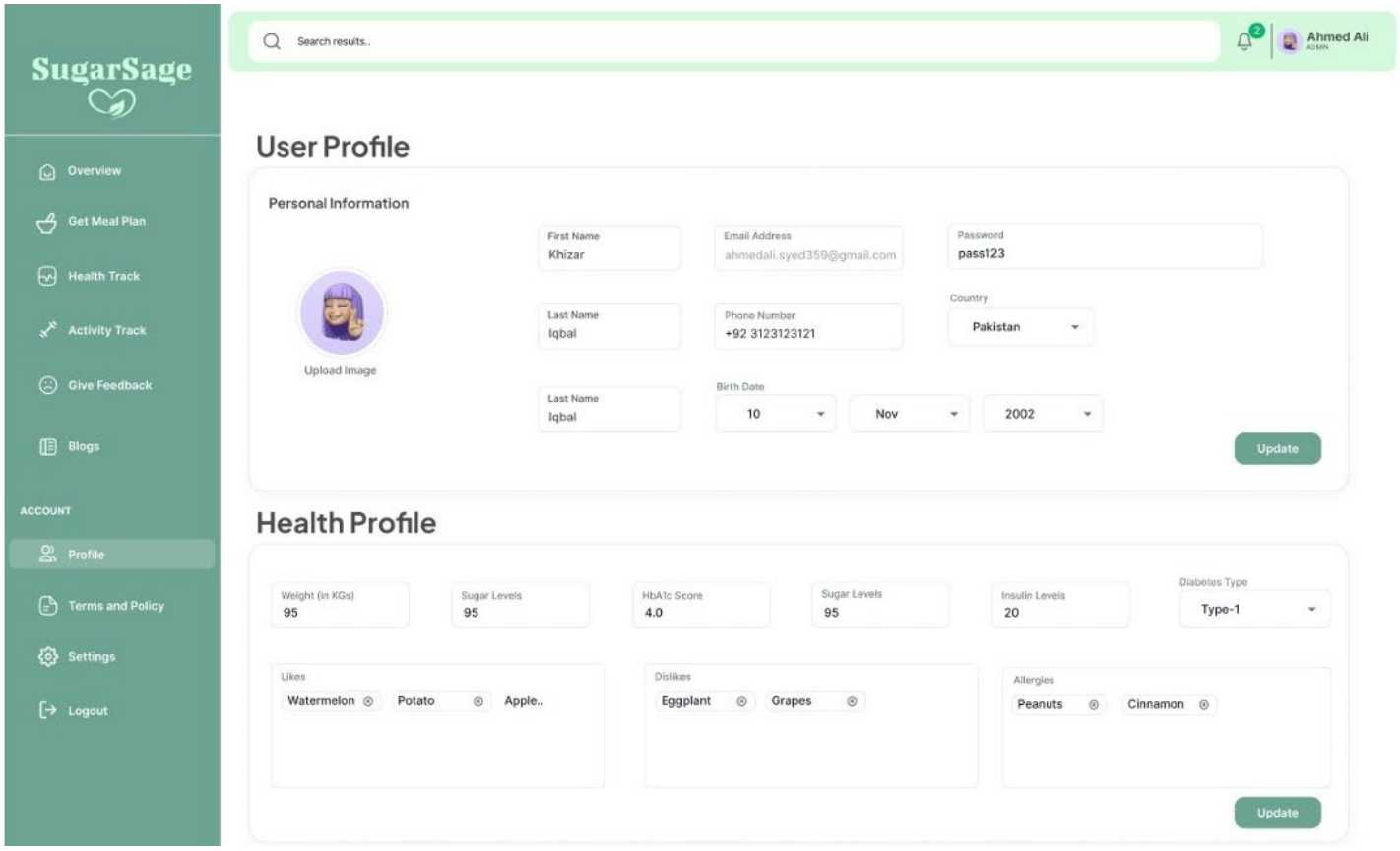

SugarSage
Search
Ahmed Ali
Navigation
Dashboard
Services
Get Meal Plan
Health Tracker
Activity Tracker
Feedback
Blogs
Account
Profile
Terms and Conditions
Settings
Health Tracking
Calories Track
18%
500/2800 Cals
Nutrition Breakdown
Fats: 57.01g (1
:5g (13.61%)
Carbs: 225.27g (32.18%)
Fats
Proteins
Carbs
Insulin Dosage Tracker
32
24
16
8
0
120
90
60
30
0
2024-01-03
2024-01-06
insulin
glucose
Ideal Weight
180


SugarSage
Search
Ahmed Ali
Navigation
Dashboard
Services
Get Meal Plan
Health Tracker
Activity Tracker
Feedback
Blogs
Account
Profile
Terms and Conditions
Settings
Log Out
Ideal Weight
180
135
90
45
0
January
February
March
April
May
June
July
August
September
October
November
December
Blood Sugar History
2023
190
165
140
115


SugarSage
Search
Ahmed Ali
Navigation
Dashboard
Services
Get Meal Plan
Health Tracker
Activity Tracker
Feedback
Blogs
Account
Profile
Terms and Conditions
Settings
Log Out
Activity
600
450
300
150
0
00:00
06:00
12:00
18:00
23:59
Calories
Sleep Stats
35%
6%
47%
12%
Steps
95%
9500/10000 Steps
Sleep Patterns
12
9
6
3
0
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday


SugarSage
Search
Ahmed Ali
Navigation
Dashboard
Services
Get Meal Plan
Health Tracker
Activity Tracker
Feedback
Blogs
Account
Profile
Terms and Conditions
Settings
Log Out
Feedbacks
Email Address
Feedback Type
Description
Upload
Cancel
localhost:3000/dashboard/default


SugarSage
Search
Ahmed Ali
Navigation
Dashboard
Services
Get Meal Plan
Health Tracker
Activity Tracker
Feedback
Blogs
Account
Profile
Terms and Conditions
Settings
Understanding Diabetes: Types and Treatments
Explore the different types of diabetes and the latest treatment options available for managing this chronic condition effectively.
Read More
Nutritional Strategies for Managing Diabetes
Learn how diet plays a crucial role in controlling blood sugar levels and what foods to include in a diabetic-friendly diet.
Read More
The Impact of Exercise on Diabetes
Discover how regular physical activity can help improve insulin sensitivity and reduce the risk of diabetes complications.
Read More


SugarSage
Search results..
Ahmed Ali
Overview
Get Meal Plan
Health Track
Activity Track
Give Feedback
Blogs
ACCOUNT
Profile
Terms and Policy
Settings
Logout
User Profile
Personal Information
Upload Image
First Name
Khizar
Email Address
ahmedali.syed359@gmail.com
Password
pass123
Last Name
Iqbal
Phone Number
+92 3123123121
Country
Pakistan
Last Name
Iqbal
Birth Date
10
Nov
2002
Update
Health Profile
Weight (in KGs)
95
Sugar Levels
95
HbA1c Score
4.0
Sugar Levels
95
Insulin Levels
20
Diabetes Type
Type-1
Likes
Watermelon
Potato
Apple..
Dislikes
Eggplant
Grapes
Allergies
Peanuts
Cinnamon
Update

5. Test Specification and Results

5.1 Test Case Specification for User

Table 5.1.1: TC-1 Successful Login

Identifier	TC-1
Related requirements(s)	UC-2 User Login
Short description	Verify that the system permits login with correct credentials.
Pre-condition(s)	1. User must have an active account. 2. Internet connectivity required.
Input data	Email Address and Password.
Detailed steps	1. User launches the mobile application. 2. Input valid email and password in the relevant fields. 3. Tap the Login button.
Expected result(s)	The system checks credentials and redirects to the Home Page upon success.
Post-condition(s)	1. The user's session is active. 2. User can access personal data and app features.
Actual result(s)	User is logged in and able to access data and features without any issues.
Test Case Result	Pass

Table 5.1.2: TC-2 Unsuccessful Login with Incorrect Email or Password

Identifier	TC-2
Related requirements(s)	UC-2 User Login
Short description	Ensure the system rejects login attempts with an incorrect password.
Pre-condition(s)	1. The user must have an active account registered in the system. 2. Internet connectivity required.
Input data	Provide either an incorrect email, password, or both, but never both correct.
Detailed steps	1. User launches the mobile application. 2. Input an email (valid or invalid) and password (valid or invalid) in the relevant fields. One of them should be invalid for this test case. 3. Tap the Login button.
Expected result(s)	The system checks credentials and identifies the credentials are mismatched and displays an error message "Incorrect credentials. Please try again.".
Post-condition(s)	1. User session not initiated. 2. User remains logged out. 3. User is prompted to retry entering credentials.
Actual result(s)	An error message about incorrect credentials was displayed correctly. The user did not gain access to the system.
Test Case Result	Pass

Table 5.1.3: TC-3 Successful Signup

Identifier	TC-3
Related requirements(s)	UC-1 User Signup
Short description	To ensure a new user can successfully create an account by providing all required information.
Pre-condition(s)	1. The user has no prior registration with the system. 2. Internet connectivity required.
Input data	Valid email, password, and additional information.
Detailed steps	1. User launches the mobile application. 2. Tap on the 'Register' button besides the "Don't have an account?" message. 3. Input all mandatory fields with correct formats in the signup form. 4. Tap on the 'Signup' button.
Expected result(s)	1. The system performs validation checks to confirm that all inputs meet specified criteria without errors. 2. A confirmation alert or page confirms "Registration Successful." 3. The system redirects the user to the login interface to enable secure access.
Post-condition(s)	1. User credentials are successfully entered within the database. 2. User must be able to Login using their registered credentials.
Actual result(s)	The signup was completed successfully, with a correct redirection to the login interface and user can login with the registered credentials.
Test Case Result	Pass

Table 5.1.4: TC-4 Registration with Existing Email

Identifier	TC-4
Related requirements(s)	UC-1 User Signup
Short description	To ensure the system correctly prevents registration with an email address that is already registered.
Pre-condition(s)	1. The user must have a registered account in the system. 2. Internet connectivity required.
Input data	Already registered email address.
Detailed steps	1. User launches the mobile application. 2. Tap on the 'Register' button besides the "Don't have an account?" message. 3. Input an already registered email and other information within relevant fields. 4. Tap on the 'Signup' button.
Expected result(s)	1. The system detects the email is already registered to an account. 2. An error message is displayed, "Email address already in use. Please log in or use a different email address."
Post-condition(s)	1. No new user account is created. 2. The system remains on the registration page, prompting the user to change the input data.
Actual result(s)	Error message displayed about email already in use.
Test Case Result	Pass

Table 5.1.5: TC-5 Password Reset Email Sent

Identifier	TC-5
Related requirements(s)	UC for Password Reset
Short description	To ensure the system's ability to send a password reset email when a user initiates the 'Forgot Password' process.
Pre-condition(s)	1. User must already be registered in the system. 2. Internet connectivity required.
Input data	Registered email address.
Detailed steps	1. User must be on the Login page. 2. User clicks on the 'Forgot Password' link. 3. Input the registered email address. 4. Tap the 'Send Reset Link' button.
Expected result(s)	1. The system validates that the email is registered. 2. A password reset email is generated and sent to the provided email address. his email contains a code that the user must enter on the Account Verification page to proceed with resetting their password.
Post-condition(s)	1. No password changes occur until the user successfully completes the password reset process. 2. User receives a password reset email, which includes a unique code necessary for verifying their identity on the Account Verification page. 3. The user is now prepared to proceed with the password reset process by entering the provided code on the specified page.
Actual result(s)	The system correctly validated the email address but failed to generate or send the password reset email to the provided email address.
Test Case Result	Fail

Table 5.1.6: TC-6 Password Reset with Invalid Email

Identifier	TC-6
Related requirements(s)	Use-case for password reset
Short description	To ensure that the password reset process is aborted when an invalid or unregistered email address is entered.
Pre-condition(s)	1. User is at the 'Forgot Password' page. 3. Internet connectivity required.
Input data	Invalid email address
Detailed steps	1. Go to the 'Forgot Password' page. 2. Input an invalid or unregistered email address in the input field. 3. Tap the 'Send Reset Link' button.
Expected result(s)	The system attempts to validate the email address and upon failure, an error message is displayed, "Email address not found. Please try again or register."
Post-condition(s)	1. No password reset link is sent to the email address. 2. The user remains on the 'Forgot Password' page, prompted to enter a different email or to register.
Actual result(s)	An appropriate error message about email not existing.
Test Case Result	Pass

Table 5.1.7: TC-7 View Profile

Identifier	TC-7
Related requirements(s)	UC-4 User Profile
Short description	To ensure that the user can successfully view their profile information within the application.
Pre-condition(s)	1. User must be logged in.
Input data	None
Detailed steps	1. Navigate to Profile page. 2. Profile information will be displayed by the system.
Expected result(s)	User Profile information is displayed correctly by the system.
Post-condition(s)	Users can view their correct profile information.
Actual result(s)	The user's profile was accessible and displayed all details as expected.
Test Case Result	Pass

Table 5.1.8: TC-8 Edit Profile

Identifier	TC-8
Related requirements(s)	UC-4 User Profile
Short description	To ensure the system's functionality for editing and updating user profile details.
Pre-condition(s)	1. User must be logged in. 2. Internet connectivity required. 3. User must be on the Profile page.
Input data	Updated profile information (e.g., new phone number, name, dob).
Detailed steps	1. Navigate to user's profile page. 2. Click on the 'Edit Profile' button. 3. Click on 'Save changes' button.
Expected result(s)	The profile is immediately updated with the new information and the changes are visibly confirmed on the profile page upon saving.
Post-condition(s)	Updated information is successfully stored in the database.
Actual result(s)	The profile update process completed successfully, and the updated information is visible on the profile page.
Test Case Result	Pass

Table 5.1.9: TC-9 Logout Functionality

Identifier	TC-9
Related requirements(s)	UC-8 User Logout
Short description	To ensure that the user can log out successfully and their session is terminated.
Pre-condition(s)	User should be logged in.
Input data	None.
Detailed steps	Click on the logout button.
Expected result(s)	User is logged out and redirected to the login page.
Post-condition(s)	Session is terminated.
Actual result(s)	User is logged out and redirected to the login page, confirming session termination.
Test Case Result	Pass

Table 5.1.10: TC-10 Update Password

Identifier	TC-10
Related requirements(s)	Use-case for password update
Short description	Test updating the password successfully from the settings.
Pre-condition(s)	1. User must be logged in. 2. Internet connectivity required.
Input data	Current valid password and new password.
Detailed steps	1. Navigate to the Settings page. 2. Select the 'Change Password' option. 3. Enter current and new passwords in the respective fields. 4. Click on the 'Save' button.
Expected result(s)	1. The system verifies the current password. 2. The new password overwrites the old password successfully. 3. A confirmation message is displayed indicating the password has been updated.
Post-condition(s)	User needs to use new password for subsequent logins and previous passwords are invalid.
Actual result(s)	User is required to use the new password for subsequent logins.
Test Case Result	Pass

Table 5.1.11: TC-11 View Health Profile

Identifier	TC-11
Related requirements(s)	UC-5 Health Profile
Short description	Test the functionality that allows users to access and view their health profile information.
Pre-condition(s)	1. User must be logged in. 2. There must be prior recorded health data of a user.
Input data	None
Detailed steps	1. Navigate to Health Profile page. 2. Observe the display of health information, including health metrics and health status.
Expected result(s)	The application displays accurate and complete health data without any delay or inconsistently.
Post-condition(s)	The health profile data is displayed that remains unchanged unless edited in a separate operation.
Actual result(s)	User's health details were correctly displayed and matched the data previously entered/recorded.
Test Case Result	Pass

Table 5.1.12: TC-12 Update Health Profile

Identifier	TC-12
Related requirements(s)	UC-5 Health Profile
Short description	To ensure the application allows users to update and save changes to their health profile data successfully.
Pre-condition(s)	1. The user has successfully logged into their account. 2. The user has existing health profile data available for editing.
Input data	Updated health details (e.g. blood sugar, HbA1c, weight, height)
Detailed steps	1. Navigate to health profile 2. Modify the necessary health details in the appropriate fields. 3. Confirm the changes by clicking the 'Save' or 'Update' button.
Expected result(s)	A confirmation message verifies successful profile updates, and the new details are instantly visible on the health profile page.
Post-condition(s)	Updated data is stored in the database.
Actual result(s)	Health profile is updated and displayed accurately.
Test Case Result	Pass

Table 5.1.13: TC-13 Activity Track

Identifier	TC-13
Related requirements(s)	UC for Activity Track
Short description	To ensure the ability of the system to display saved daily activity track.
Pre-condition(s)	User must be logged in.
Input data	None
Detailed steps	1. Navigate to Activity Track Page. 2. Observe the graphical representation of activities (e.g. daily step counts, sleep patterns).
Expected result(s)	The system displays the activity profile with all recorded daily physical activities.
Post-condition(s)	The user can view their activity tracking details without system errors or inconsistently.
Actual result(s)	The activity profile was successfully accessed and displayed the user's daily physical activities as expected.
Test Case Result	Pass

Table 5.1.14: TC-14 View Meal Plan

Identifier	TC-14
Related requirements(s)	UC-3 Meal Planning
Short description	To ensure that user can access and view their previously created meal plans.
Pre-condition(s)	1. User must be logged in. 2. User has requested a meal plan previously.
Input data	None
Detailed steps	1. Navigate to Meal Plan page 2. Select the "View Meal Plan" option to view the current meal plan.
Expected result(s)	The application displays the meal plan clearly, showing all details such as meal types, scheduled times, and nutritional information.
Post-condition(s)	The Meal plan remains viewable and can be followed by the user.
Actual result(s)	The meal plan was successfully retrieved and displayed as expected, with all details clearly visible.
Test Case Result	Pass

Table 5.1.15: TC-15 Generate Meal Plan

Identifier	TC-15
Related requirements(s)	UC-3 Meal Planning
Short description	Ensure the application can generate a personalized meal plan tailored to the user's entered health data.
Pre-condition(s)	1. User must be logged in. 2. User must have entered relevant health information and preferences in their profile.
Input data	Health metrics and preferences influence meal planning.
Detailed steps	1. Navigate to Meal Plan page. 2. Select the "Generate New Meal Plan" option to get a new meal plan.
Expected result(s)	A new meal plan, customized according to the provided health data, is created and displayed to the user.
Post-condition(s)	Meal plan is viewable and can be followed by the user.
Actual result(s)	The system successfully generated a meal plan, but it contained inconsistencies or errors relative to the user's health data inputs.
Test Case Result	Fail

Table 5.1.16: TC-16 View Blogs

Identifier	TC-16
Related requirements(s)	UC for Blogs
Short description	To ensure that users can view blogs and navigate to full articles.
Pre-condition(s)	User must be logged in.
Input data	None
Detailed steps	1. Navigate to Blog Page. 2. Click on a blog summary to read it. 3. Use the provided link or button to access the full article source.
Expected result(s)	The blog summaries are displayed which a link that redirects to the full article.
Post-condition(s)	User has viewed the blog summary and has the option to read the full article via an external link.
Actual result(s)	The blog summaries are accessible, accurately displayed, and the redirection link to the full articles functions correctly.
Test Case Result	Pass

5.2 Test Case Specification for Admin

Table 5.2.1: TC-17 Successful Login

Identifier	TC-17
Related requirements(s)	UC-7 Admin Login
Short description	To ensure that the system permits login with correct credentials.
Pre-condition(s)	1. Admin must have an active account in the system. 2. Internet connectivity required.
Input data	Email Address and Password.
Detailed steps	1. Admin launches the web application on their browser. 2. Input valid email and password in the relevant fields. 3. Tap the Login button.
Expected result(s)	The system checks credentials and redirects to the Dashboard Page upon success.
Post-condition(s)	1. The admin's session is active. 2. Admin can access personal data and app features.
Actual result(s)	Admin is logged in and able to access data and features without any issues.
Test Case Result	Pass

Table 5.2.2: TC-18 Unsuccessful Login with Incorrect Email or Password

Identifier	TC-18
Related requirements(s)	UC-7 Admin Login
Short description	Ensure the system rejects login attempts with an incorrect password.
Pre-condition(s)	1. The admin must have an active account in the system. 2. Internet connectivity required.
Input data	Provide either an incorrect email, password, or both, but never both correct.
Detailed steps	1. Admin launches the web application on their browser. 2. Input an email (valid or invalid) and password (valid or invalid) in the relevant fields. One of them should be invalid for this test case. 3. Tap the Login button.
Expected result(s)	The system checks credentials and identifies the credentials are mismatched and displays an error message "Incorrect credentials. Please try again.".
Post-condition(s)	1. Admin session not initiated. 2. Admin remains logged out. 3. Admin is prompted to retry entering credentials.
Actual result(s)	An error message about incorrect credentials was displayed correctly. The admin did not gain access to the system.
Test Case Result	Pass

Table 5.2.3: TC-19 View Profile

Identifier	TC-19
Related requirements(s)	UC-15 Admin Profile
Short description	To ensure that the admin can successfully view their profile information within the application.
Pre-condition(s)	1. User must be logged in. 2. Internet connectivity required.
Input data	None
Detailed steps	1. Navigate to Profile page. 2. Profile information will be displayed by the system.
Expected result(s)	Admin Profile information is displayed correctly by the system.
Post-condition(s)	Admin can view his profile information.
Actual result(s)	The admin profile was accessible and displayed all details as expected.
Test Case Result	Pass

Table 5.2.4: TC-20 Edit Profile

Identifier	TC-20
Related requirements(s)	UC-15 Admin Profile
Short description	To ensure the system's functionality for editing and updating admin profile details.
Pre-condition(s)	1. Admin must be logged in. 2. Internet connectivity required. 3. Admin must be on the Profile page.
Input data	Updated profile information (e.g., new phone number, name,).
Detailed steps	1. Navigate to admin's profile page. 2. Click on the 'Edit Profile' button. 3. Click on 'Save changes' button.
Expected result(s)	The profile is immediately updated with the new information and the changes are visibly confirmed on the profile page upon saving.
Post-condition(s)	Updated information is successfully stored in the database.
Actual result(s)	The profile update process completed successfully, and the updated information is visible on the profile page.
Test Case Result	Pass

Table 5.2.5: TC-21 Logout Functionality

Identifier	TC-21
Related requirements(s)	UC-27 Admin Logout
Short description	To ensure that admin can log out successfully and his session is terminated.
Pre-condition(s)	Admin should be logged in.
Input data	None.
Detailed steps	Click on the logout button.
Expected result(s)	Admin is logged out and redirected to the login page.
Post-condition(s)	Session is terminated.
Actual result(s)	Admin is logged out and redirected to the login page, confirming session termination.
Test Case Result	Pass

Table 5.2.6: TC-22 Admin Dashboard

Identifier	TC-22
Related requirements(s)	UC-8 Admin Dashboard
Short description	To evaluate the system's ability to accurately display essential system statistics and user activity.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. If user logins, they are redirected to the dashboard by default. Otherwise, they can navigate to Dashboard page themselves. 2. Observe the graphical representation of system activity (e.g. total users, geographics of users, age distributions, genders).
Expected result(s)	The dashboard displays up-to-date statistics without any delays or errors.
Post-condition(s)	The administrator retains access to the dashboard, which consistently displays system metrics.
Actual result(s)	The dashboard was accessible, and all visual data representations were correct and current as per the latest system data.
Test Case Result	Pass

Table 5.2.7: TC-23 View Users

Identifier	TC-23
Related requirements(s)	UC-9 User Management
Short description	To ensure the user management section accurately displays latest user data.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Users Table page. 2. View the personal and health information of each user in a table format (e.g. name, email, dob, country, gender)
Expected result(s)	All user information displayed in the 'User Table' is accurate and up to date, reflecting any recent changes.
Post-condition(s)	The admin continues to have access to the updated 'User Table' with the correct details after any changes are made.
Actual result(s)	The 'User Table' displayed the latest information for all users accurately during the test.
Test Case Result	Pass

Table 5.2.8: TC-24 Edit User

Identifier	TC-24
Related requirements(s)	UC-9 User Management
Short description	To test the functionality that allows an admin to edit user details.
Pre-condition(s)	1. Admin must be logged in. 2. Admin must be on the Users Table page.
Input data	None
Detailed steps	1. Click on the 'Edit' icon next to the user's row you want to modify. 2. Update the desired fields. 3. Save the changes by clicking the 'Save' button.
Expected result(s)	The system allows modifications to the selected user's details and updates the information in the user table upon saving.
Post-condition(s)	Changes are accurately saved and displayed, and the admin retains editing capabilities.
Actual result(s)	User details were successfully updated and accurately reflected in the user table.
Test Case Result	Pass

Table 5.2.9: TC-25 Delete User

Identifier	TC-25
Related requirements(s)	UC-9 User Management
Short description	To test the admin's ability to delete user profiles.
Pre-condition(s)	1. Admin must be logged in. 2. Admin must be on the Users Table page.
Input data	None
Detailed steps	1. Click the 'Delete' icon next to the user's row you intend to remove 2. Confirm the deletion by clicking on the Confirm' button.
Expected result(s)	The user profile is removed from the system immediately after confirmation. The user may no longer able to login into the system.
Post-condition(s)	The user is no longer listed in the user table, and the admin retains the ability to manage remaining profiles.
Actual result(s)	The selected user was successfully deleted from the system, and the user table was updated accordingly.
Test Case Result	Pass

Table 5.2.10: TC-26 View Foods

Identifier	TC-26
Related requirements(s)	UC-10 Food Management
Short description	To test the admin's ability to access and review the list of food items.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Foods Table page. 2. Can view the list of all available food items.
Expected result(s)	The food table loads successfully, displaying a comprehensive and accurate list of food items.
Post-condition(s)	Admin retains access to the system, and the displayed list of food items remains up to date.
Actual result(s)	The food list was accessed successfully, with all items correctly displayed.
Test Case Result	Pass

Table 5.2.11: TC-27 Add Food

Identifier	TC-27
Related requirements(s)	UC-10 Food Management
Short description	Verify that the admin can add new food items into the database.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Foods Table page. 2. Click on the 'Add Food' button. 3. Enter the necessary food details in the input fields 4. Submit the data by clicking on the 'Add' button.
Expected result(s)	The system should accept the input, save the new food item to the database, and display it in the Foods Table.
Post-condition(s)	The Foods Table includes the new food item, which is now available for further actions like edit or delete by the admin.
Actual result(s)	The new food item was successfully added and displayed in the Foods Table.
Test Case Result	Pass

Table 5.2.12: TC-28 Edit Food

Identifier	TC-28
Related requirements(s)	UC-10 Food Management
Short description	To test the admin's ability to edit existing food items in the database.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Foods Table page. 2. Select an existing food item by clicking the 'Edit' icon in its corresponding row. 3. Modify the required fields in the input form. 4. Confirm the changes by clicking the 'Save' button.
Expected result(s)	The system updates and saves changes to food items quickly and shows these changes in the Foods Table immediately.
Post-condition(s)	The Foods Table shows the updated details, and the system is ready for more edits.
Actual result(s)	The food item was edited successfully and the changes were correctly displayed in the Foods Table.
Test Case Result	Pass

Table 5.2.13: TC-29 Delete Food

Identifier	TC-29
Related requirements(s)	UC-10 Food Management
Short description	To test the admin's ability to delete food items from the database.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Foods Table page. 2. Select a food item by clicking the 'Delete' icon next to it. 3. Confirm the deletion of a food item by clicking the 'Save' button.
Expected result(s)	The system should delete the selected food item, remove it from the Foods Table, and not display it anymore.
Post-condition(s)	The Foods Table no longer lists the deleted item, ensuring the database is updated correctly.
Actual result(s)	The food item was successfully deleted from the database and no longer appears in the Foods Table.
Test Case Result	Pass

Table 5.2.14: TC-30 View Dish

Identifier	TC-30
Related requirements(s)	UC-11 Dish Management
Short description	To test the admin's ability to access and review the list of dish items.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Dishes Table page. 2. Can view the list of all available dish items.
Expected result(s)	The dish table loads successfully, displaying a comprehensive and accurate list of dish items.
Post-condition(s)	Admin retains access to the system, and the displayed list of dish items remains up to date.
Actual result(s)	The dish list was accessed successfully, with all items correctly displayed.
Test Case Result	Pass

Table 5.2.15: TC-31 Add Dish

Identifier	TC-31
Related requirements(s)	UC-10 Dish Management
Short description	Verify that the admin can add new dish items into the database.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Dishes Table page. 2. Click on the 'Add Dish' button. 3. Enter the necessary dish details in the input fields 4. Submit the data by clicking on the 'Add' button.
Expected result(s)	The system should accept the input, save the new dish item to the database, and display it in the Dishes Table.
Post-condition(s)	The Dishes Table includes the new dish item, which is now available for further actions like edit or delete by the admin.
Actual result(s)	The new dish item was successfully added and displayed in the Dishes Table.
Test Case Result	Pass

Table 5.2.16: TC-32 Edit Dish

Identifier	TC-32
Related requirements(s)	UC-10 Food Management
Short description	To test the admin's ability to edit existing dish items in the database.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Dishes Table page. 2. Selects an existing dish item by clicking the 'Edit' icon in its corresponding row. 3. Modify the required fields in the input form. 4. Confirms the changes by clicking the 'Save' button.
Expected result(s)	The system updates and saves changes to dish items quickly and shows these changes in the Dishes Table immediately.
Post-condition(s)	The Dishes Table shows the updated details, and the system is ready for more edits.
Actual result(s)	The dish item was edited successfully, and the changes were correctly displayed in the Dishes Table.
Test Case Result	Pass

Table 5.2.17: TC-33 Delete Dish

Identifier	TC-33
Related requirements(s)	UC-10 Food Management
Short description	To test the admin's ability to delete dish items from the database.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Dishes Table page. 2. Select a dish item by clicking the 'Delete' icon next to it. 3. Confirm the deletion of dish item by clicking the 'Save' button.
Expected result(s)	The system should delete the selected dish item, remove it from the Dishes Table, and not display it anymore.
Post-condition(s)	The Dishes Table no longer lists the deleted item, ensuring the database is updated correctly.
Actual result(s)	The dish item was successfully deleted from the database and no longer appears in the Dishes Table.
Test Case Result	Pass

Table 5.2.18: TC-34 View Feedback

Identifier	TC-34
Related requirements(s)	UC-12 Feedback Management
Short description	To test the system's ability to display feedback submitted by users.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to the Feedback Table page. 2. Review the list of feedback entries.
Expected result(s)	The system should display all feedback entries in an organized manner, allowing the admin to read each piece of feedback clearly.
Post-condition(s)	The feedback remains accessible and unchanged, with the admin able to continue viewing additional feedback or exit the section.
Actual result(s)	All feedback was successfully displayed, with entries being clear and accessible to the admin.
Test Case Result	Pass

Table 5.2.19: TC-35 Delete Feedback

Identifier	TC-35
Related requirements(s)	UC-12 Feedback Management
Short description	To test the admin's ability to delete user's feedback from the database.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to Feedback Table page. 2. Select certain feedback by clicking the 'Delete' icon next to it. 3. Confirm the deletion of a feedback by clicking the 'Confirm' button.
Expected result(s)	The system should successfully delete the selected feedback, removing it from the list and ensuring it is no longer accessible.
Post-condition(s)	The deleted feedback should not appear in the list, confirming it has been permanently removed from the database.
Actual result(s)	The feedback entry was successfully deleted, with the list updated to no longer show the deleted item.
Test Case Result	Pass

Table 5.2.20: TC-36 Add Blog

Identifier	TC-36
Related requirements(s)	UC-14 Blog Management
Short description	To test the ability of an admin to add new blog posts within the system.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to Blog Table page. 2. Click the 'Add Blog' button. 3. Enter the blog title, body content, link and upload any images if necessary. 4. Click the 'Submit' button to add the blog to the system.
Expected result(s)	The system should accept the new post, adding it to the blog section without issues and making it viewable to user's immediately.
Post-condition(s)	The blog section updates to include the new post, displaying all details as entered by the admin, who can add more posts or edit existing ones.
Actual result(s)	The new blog post was successfully created and displayed correctly in the blog section.
Test Case Result	Pass

Table 5.2.21: TC-37 Edit Blog

Identifier	TC-37
Related requirements(s)	UC-14 Blog Management
Short description	To test the admin's ability to edit and update blog content.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to Blog Table page. 2. Select a post to edit, make any necessary changes to the text or images. 3. Click the 'Update' button to submit the updates made on the blog.
Expected result(s)	The system should process and save the changes, immediately reflecting the updates in the blog post.
Post-condition(s)	The blog post shows all recent edits correctly, and the admin remains able to make further modifications.
Actual result(s)	The blog post was successfully edited, with all changes accurately shown.
Test Case Result	Pass

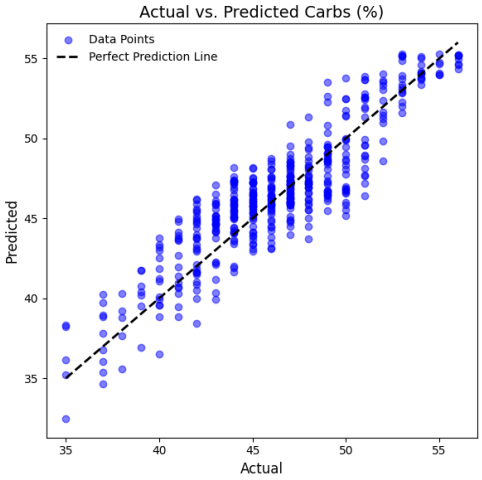
Table 5.2.22: TC-38 Delete Blog

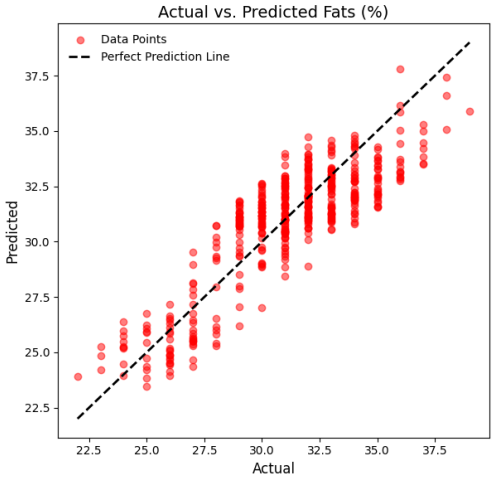
Identifier	TC-38
Related requirements(s)	UC-14 Blog Management
Short description	To test the admin's ability to delete blog posts within the system.
Pre-condition(s)	Admin must be logged in.
Input data	None
Detailed steps	1. Navigate to Blog Table page. 2. Select a post to edit, make any necessary changes to the text or images. 3. Click the 'Update' button to submit the updates made on the blog.
Expected result(s)	The system should permanently remove the selected blog post, ensuring it no longer appears in the blog section.
Post-condition(s)	The blog section no longer lists the deleted post, and the admin can continue to manage other posts.
Actual result(s)	The blog post was successfully deleted and is no longer visible in the blog section.
Test Case Result	Pass

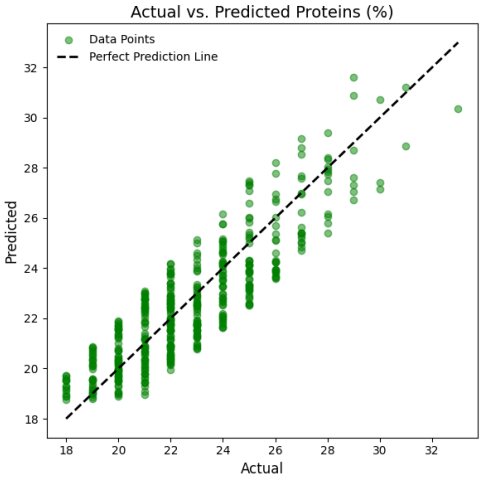
Table 5.2.23: Table For Model Testings

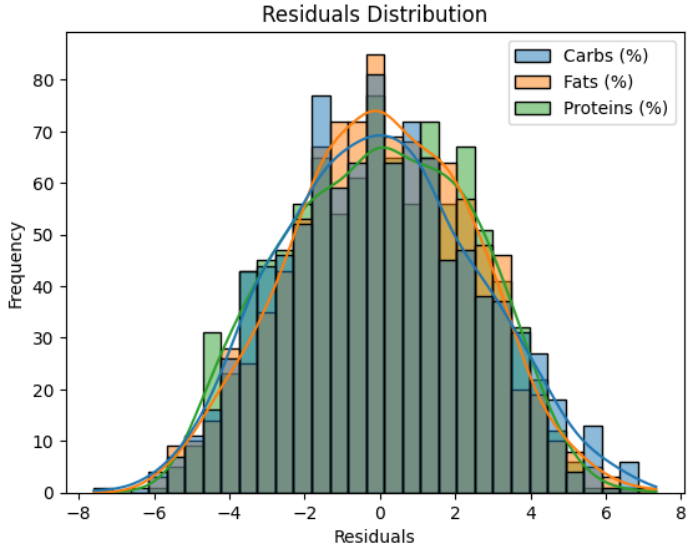
Model Name	Mean Squared Error (MSE)	Root Mean Squared Error (RMSE)	Mean Absolute Percentage Error (MAPE)
1. Linear Regression	6.93	2.63	7.23%
2. Ridge Regression	7.12	2.67	7.31%
3. Lasso Regression	6.92	2.63	7.23%
4. Elastic Net Regression	7.05	2.66	7.28%
5. Polynomial Regression	6.93	2.63	7.23%
6. Decision Tree Regression	7.06	2.66	7.31%
7. Random Forest Regression	6.22	2.49	6.96%
9. AdaBoost Regression	6.27	2.5	6.98%
10. XGBoost Regression	**6.0**	**2.45**	**6.78%**
11. K-Nearest Neighbors Regression	11.02	3.32	8.69%

XGBoost Regressor Results (Best Model):







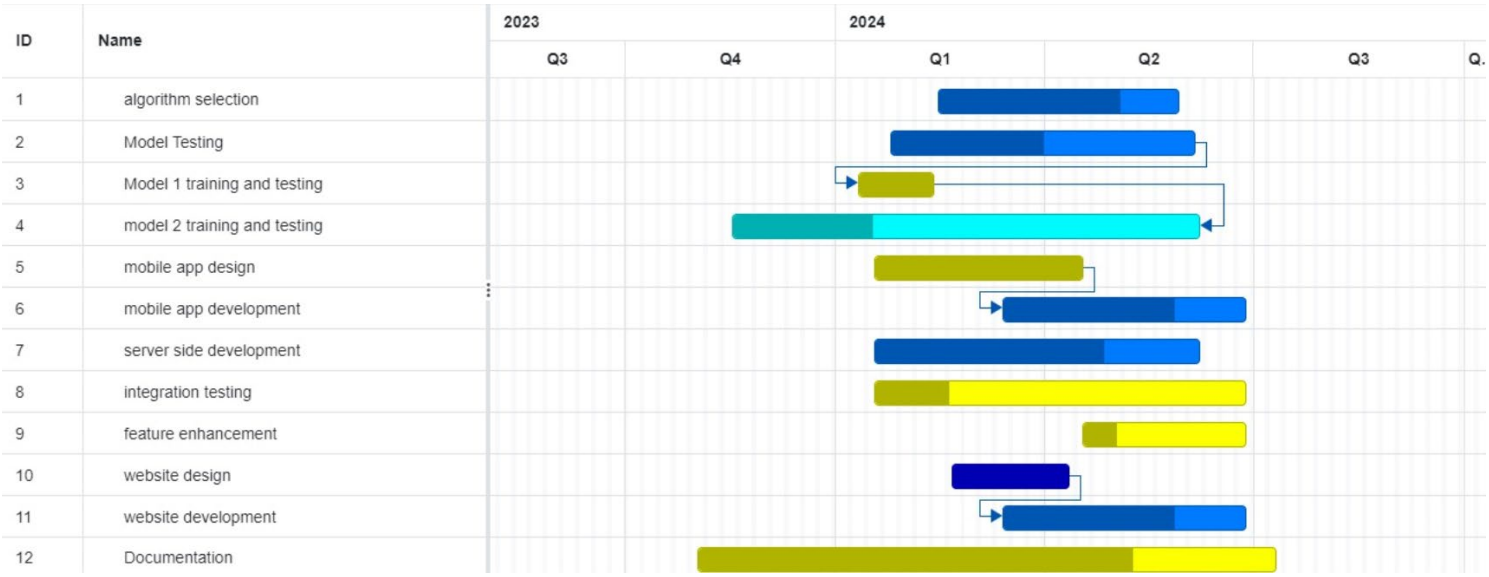


6.2 Summary of Test Results

Table 6.2: Summary of Test Results

Module Name	Test cases run	Number of defects found	Number of defects corrected so far	Number of defects still need to be corrected
User Authentication	TC1, TC2, TC3, TC4, TC5, TC6, TC9	1	0	1
User Profile Management	TC7, TC8, TC10, TC11, TC12	0	0	0
User Activity	TC13, TC14, TC15, TC16	1	0	1
Admin Authentication	TC17, TC18, TC21	0	0	0
Admin Profile Management	TC19, TC20	0	0	0
Users Management	TC22, TC23, TC24, TC25	0	0	0
Foods Management	TC26, TC27, TC28, TC29	0	0	0
Dishes Management	TC30, TC31, TC32, TC33	0	0	0
Feedback Management	TC34, TC35	0	0	0
Blogs Management	TC36, TC37, TC38	0	0	0
Complete System	38	2	0	2

6. Revised Project Plan



The Gantt chart outlines the project plan for various tasks spanning from the third quarter of 2023 through the third quarter of 2024. Here's a breakdown of the timeline and activities:

1. Algorithm Selection - This task began in late Q3 2023 and finished by early Q4 2023.
2. Model Testing - Starting mid-Q4 2023 and extending into early Q1 2024.
3. Model 1 Training and Testing - This task commenced early in Q1 2024 and is set to conclude towards the end of Q1 2024.
4. Model 2 Training and Testing - Initiated parallel to Model 1 but extended a bit longer into early Q2 2024.
5. Mobile App Design - Beginning in early Q1 2024 and concluding by mid-Q2 2024.
6. Mobile App Development - This longer phase started towards the end of Q1 2024 and will continue until mid-Q3 2024.
7. Server-Side Development - Scheduled to start at the beginning of Q2 2024 and end in early Q3 2024.
8. Integration Testing - Planned to begin in late Q2 2024 and expected to be completed by the start of Q3 2024.
9. Feature Enhancement - A shorter phase in mid-Q2 2024.
10. Website Design - Set to begin in early Q2 2024 and finish by the end of Q2 2024.
11. Website Development - Following right after the design phase and extending through mid Q3 2024.
12. Documentation - This critical phase overlaps several other tasks, starting early in Q2 2024 and finishing at the end of Q3 2024.

Each task is visually represented with a distinct color bar, indicating the start and end points, and some tasks show overlap or parallel progression, indicating simultaneous developments in different areas of the project. The chart effectively uses colors and lengths of bars to provide a clear visual timeline for project milestones and dependencies.

Table 6.2: Project Completion Status

Module Name	**Status** (Complete, Partially Implemented, Not Implemented)
Login	Completed
Authentication/Authorization	Partially Implemented
Dashboard/Graphs	Completed
CRUD Operations to manage users	Partially Implemented
CRUD operations to manage health data	Partially Implemented
Dataset Generation and Validation for Nutrition Distribution	Completed
Model Training for Nutrition Distribution	Completed
Model Testing for Nutrition Distribution	Completed
Dataset Generation for Food Selection	Partially Implemented
Model Training for Food Selection	Partially Implemented

7. References

1. Khan, M. A., Mebrahtu, S., Kyaw-Myint, T., & UNICEF Islamabad. (2001). Food Composition Table for Pakistan (Revised 2001). NWFP Agricultural University, Peshawar; Ministry of Planning and Development, Government of Pakistan; Department of Agricultural Chemistry.

Appendix A: Glossary

1. AI (Artificial Intelligence): Technology that enables machines to simulate intelligent human behavior. In the context of SugarSage, it refers to the algorithm used for generating personalized diet plans.

2. SRS (Software Requirements Specification): A detailed description of the software to be developed, including its functional and non-functional requirements.

3. Personalized Diet Planning: A feature within SugarSage that uses AI to create customized diet plans for diabetics based on their health metrics, medication regimens, and activity levels.

4. Local Food Database: A database within SugarSage that includes various local Pakistani foods and their nutritional information, allowing the system to align dietary recommendations with local dietary habits and preferences.

5. Mobile Application: The end-user platform of SugarSage, designed to provide a user-friendly interface for diabetics to manage their diet and monitor their health.

6. Singleton Pattern: A design pattern that ensures a class has only one instance and provides a global point of access to it. For SugarSage, it will be used for classes like the database connection manager or the configuration manager to manage shared resources effectively.

7. Cloud-Based Environment: A technology infrastructure that utilizes cloud computing to provide scalable and reliable service hosting, including capabilities such as auto-scaling to adjust resources based on demand.

8. Auto-Scaling: A feature of cloud services that automatically adjusts the number of computational resources according to the system's current load, ensuring responsiveness and cost-efficiency.

9. Admin Activity Diagram: A flowchart that illustrates the various actions an administrator can perform within the SugarSage system, including user management, content editing, and feedback processing.

10. User Activity Diagram: A diagram that outlines the flow of user interactions within the SugarSage system, from the welcome page to health profile management and meal plan adjustment.

11. API (Application Programming Interface): A set of protocols for building and interacting with software applications. SugarSage components use APIs for communication and data exchange.

12. User Profile Component: A part of the SugarSage system that stores and manages user-specific information such as health metrics and preferences.

13. Health Tracker: A component of SugarSage that monitors and records health-related data for users.

14. Meal Planner: A feature in SugarSage that generates personalized diet plans based on user health data and dietary preferences.

15. Feedback Module: A system within SugarSage that allows users to communicate feedback directly to system administrators.

16. Blog Reading: A functionality in SugarSage that provides users with access to educational content on diabetes management
17. GI (Glycemic Index) - A number representing the relative ability of a carbohydrate food to increase the level of glucose in the blood.
18. UI (User Interface) - The space where interactions between humans and machines occur.
19. ML (Machine Learning) - A subset of AI that allows systems to automatically learn and improve from experience.

Appendix B: IV & V Report

(Independent verification & validation)
IV & V Resource

Name Signature

S#	Defect Description	Origin Stage	Status	Fix Time	
				Hours	Minutes
1					
2					
3					
…					

Table 1: List of non-trivial defects

This document has been adapted from the following:

- Previous project templates at UCP
- High-level Technical Design, Centers for Medicare & Medicaid Services. (www.cms.gov)