Software Requirements and Design Document

for

Dietary Management Application

Prepared by

Mishal Ali 22i-1291

Ayaan Mughal 22i-0861

Ayaan Khan 22i-0832

Tism Tech

Monday, 25th Nov 2024

Table of Contents

Introduction

1.1. Purpose

This document aims to compile, examine, and define the high-level requirements and features for the Dietary Management System application. It focuses on the capabilities needed by stakeholders and target users, highlighting functionalities that will support effective dietary monitoring and wellness management. The system is designed to streamline meal planning, meal logging, exercise tracking, and nutritional analysis. By incorporating these features, it facilitates healthy lifestyle habits, personalized dietary adjustments, and comprehensive recordkeeping to enhance users' overall health and fitness. The use cases -elaborated below- provide detailed insights into how the application meets user needs, ensuring a user-friendly and efficient dietary management experience.

1.2. Product Scope

This Vision Document outlines the scope of the Dietary Management System, an application designed to promote healthier lifestyles through effective dietary monitoring and wellness management. The system will provide a robust set of features, including personalized meal planning, meal logging, exercise tracking, and nutritional analysis. It aims to cater to a diverse range of users, including teenagers, adults, and health-conscious individuals, across various demographic, social, and age groups, helping them enhance their overall fitness and well-being.

The application is designed for deployment on Windows-based PCs, ensuring compatibility with widely used desktop platforms. It will be accessible via a secure, user-friendly interface and provide real-time updates, ensuring the accuracy and reliability of logged data. The application will also leverage a Java-based architecture, utilizing JavaFX for the user interface, a relational database for data storage, and well-structured design principles such as GRASP and GoF for scalability and maintainability.

1.3. Title

Dietary Management System: Promoting Health and Fitness Through Personalized Meal Planning and Lifestyle Tracking

1.4. Objective

Develop a comprehensive software solution that empowers users to monitor their meals and relative nutritional content effectively. The application will support users in achieving their dietary goals, whether focused on weight management, muscle building, or overall improving health. By

providing personalized insights, detailed tracking capabilities, and data-driven recommendations, the software will serve as a reliable tool for fostering healthier lifestyle choices and enabling users to reach their fitness objectives with precision and consistency.

1.5.Problem Statement

Despite the rapid evolution of technology and automation, many individuals still rely on manual methods, such as journals or basic notes apps, to track their diets and nutritional intake. This approach is time-consuming, prone to error, and lacks the efficiency needed to maintain consistency in dietary monitoring.

Our solution addresses this challenge by automating and streamlining the process through a software application that eliminates the need for manual tracking. The application will provide users with a seamless experience, automatically logging meals and nutritional content while offering scalability for future enhancements such as personalized diet recommendations and integration with other fitness apps. While the initial version of the project will focus on core functionalities, its design will support future growth to deliver a more comprehensive and user-centric dietary management experience.

The problem of	**Manual tracking of diets and nutritional intake using journals or notes apps.**
Affects	Individuals aiming to monitor their diet and achieve fitness goals.
The impact of which is	Time-consuming, prone to errors, and inconsistent dietary monitoring.
Leading to the solution	To develop a software application that automates meal tracking, streamlines nutritional monitoring, and eliminates the need for manual recordkeeping while providing scalability for future features like diet recommendations and fitness app integration.

Overall Description

1.6.Product Perspective

A brand-new stand-alone software program called the Dietary Management System was created to help users control their eating patterns and reach their fitness objectives. It is intended to serve as a foundational tool in the larger field of fitness and health management applications, but it is independent of any current systems.

1.7.Product Functions

The Dietary Management System will provide the following major functionalities:

- ***Meal Logging**: Users can log their meals, including details such as ingredients, portion sizes.*
- ***Nutritional Analysis**: Automatically calculates and displays the nutritional content (calories, macronutrients, etc.) of logged meals.*
- ***Generating Diet Plans:** Users can customize their own diets or the system can generate one on the basis of the user's health information and goals.*
- ***Progress Tracking**: Tracks users' progress toward dietary goals, such as weight loss, weight gain, or muscle building.*
- ***Personalized Goal Setting**: Allows users to define specific dietary objectives and monitors their adherence to these goals.*
- ***Cross-Platform Accessibility**: Accessible via mobile and web applications, ensuring seamless usage across devices.*
- ***Scalability for Future Features**: Designed to integrate additional features, such as personalized diet recommendations, integration with fitness wearables, and syncing with other health and fitness apps.*

1.8. List of Use Cases

As per the use case diagram provided in *Section 2.5* ,following is the list of use cases implemented in application:

1.8.1. **Manage Food Catalogue**
Admin manages the database of food items available, using the *CRUD* operations

1.8.2. **Display Nutritional Information**
The system displays detailed nutritional content of food items to users.

1.8.3. **Provide Food Lists**
Users are provided with categorized food lists for easier selection and tracking.

1.8.4. **Track Progress**
Users can monitor their progress toward dietary goals, such as calorie consumption or fitness objectives.

1.8.5. **Log In/Register**
Users and admins can create accounts or log in to access the system.

1.8.6. **Generate Diet Plan**
The system helps generate personalized diet plans based on user goals and preferences.

1.8.7. **Review Diet Plans**
Users and dietitians review diet plans for customization and approval.

1.8.8. **Add New Meal**
Users can add meals to their diet logs, specifying nutritional content.

1.8.9. **View Existing Diet Plans**
Users can view their previously created or recommended diet plans.

1.8.10. **Log Meals**
Users record details of consumed meals into the system.

1.8.11. **Search Diet Plans**
Users search for specific diet plans based on their preferences or health goals.

1.8.12. **Add Health Information**
Users add health-related data such as weight, age, or medical conditions to customize recommendations.

1.8.13. **Record Exercise**
Users log physical activities to balance calorie intake and expenditure.

1.8.14. **Search Food Items**
Users search the database for specific food items and their nutritional values.

1.9.Extended Use Cases

1.9.1. **Manage Food Catalogue**

Use case ID	UC01

Use case name	Manage Food Catalogue
Scope	Dietary Management System
Level	Sub-function
Primary Actor	Admin
Stakeholders and Interests	· Admin: maintains an up-to-date list of food items, including their nutritional information · User: accesses this up-to-date catalogue when logging meals or searching for food items · Dietitian: accesses the food catalogue to help make personalized diet plans for the users · System: ensures integrity of the food database and provides reliable information to the users
Preconditions	· Admin should be registered and logged into the system · Admin is authenticated to access the management functions of the system · The food database should be accessible and existing
Success Guarantee (Postconditions)	· Food catalogue is updated successfully · Users/Dietitians can view the updated food catalogue while searching for logging meals

Main Success Scenario	Actor Action 1. Admins logs into the system. 3. Admin navigates to the food catalogue section. 5. Admin chooses to add new food item, edit an existing item, or remove an outdated item. 6.Admin inputs the required information (e.g. name, calories, nutrients). 7. Admin submit the changes.	System Response 2.System checks Admin's credentials. 4. System displays the food items in the catalogue and CRUD options. 8. System validates the input and updates the catalogue. 9. Confirms the update to the admin. 10. Displays the updated catalogue.
Extensions	· 5a. If admin attempts to remove a food item being referenced by an active meal log, the system displays a warning and prevents removal. · 6a. If the admin inputs invalid or incomplete data (e.g., missing nutritional values), the system prompts the admin to correct the information before proceeding.	
Special Requirements	· System must ensure data integrity and prevent redundancy · Nutritional information should be in standardized units (e.g. grams, kcal) · User interface must be user friendly and easy to use.	
Technology & Data	Mobile Phone, internet connections and the app for using the features	

variations list	
Frequency of Occurrence	Varies on how frequently the food database needs updating.

1.9.2. **Display Nutritional Information**

Use case ID	UC02
Use case name	Display Nutritional Information
Scope	Dietary Management System
Level	Sub-function
Primary Actor	System
Stakeholders and Interests	· **Admin:** makes sure that the nutritional information is up-to date and accurate · **User:** wants to view nutritional information for food items to make informed dietary decisions · **Dietitian:** accesses the nutritional information when personalizing users diet plans · **System:** responsible for fetching and displaying the correct nutritional data

<table>
<tr><td>Preconditions</td><td colspan="2">· User should be registered and logged into the system to access the display
· System should be correctly connected to the database
· Food catalogue should be populated with nutritional information for each food item</td></tr>
<tr><td>Success Guarantee (Postconditions)</td><td colspan="2">· User successfully views the detailed nutritional information for the selected food item
· System displays the nutritional data clearly</td></tr>
<tr><td>Main Success Scenario</td><td>Actor Action

1. User logs into the system.

2. User searches the specific food item or browses the food catalogue

3. User selects the required food item for which they want to view the nutritional information.

6. User views the information and decides whether to add the food to</td><td>System Response

2. System checks User’s credentials.

4. System retrieves the nutritional data for the selected food item from the database

5. System displays the detailed retrieved nutritional data including calories, macronutrients and micronutrients,</td></tr>
</table>

	their meal as per their goal requirement.	
Extensions	· 2a. If the user cannot find the desired food item, the system displays a message suggesting similar food items or prompts the user to add the new food item · 4a. If the system fails to retrieve the nutritional data (e.g. database error), it displays an error message	
Special Requirements	· Nutritional information must be displayed in standardized units (e.g. grams, kcal) · System should allow users to see portion-specific nutritional data (e.g. per 100 grams, per serving) · Data should be displayed in a user-friendly format, with the option to view detailed breakdowns	
Technology & Data variations list	Mobile Phone, internet connections and the app for using the features	
Frequency of Occurrence	Frequently, depending on user's meal logging	

1.9.3.Provide Food Lists

Use case ID	UC03
Use case name	Provide Food Lists
Scope	Dietary Management System
Level	Sub-function
Primary Actor	System
Stakeholders and Interests	· Admin: ensures that the food list is comprehensive, well-organized, and easily accessible · User: wants to browse for food items to either log meals or plan their diet · Dietitian: accesses the nutritional information when personalizing users diet plans · System: responsible for fetching and displaying the list of available food items
Preconditions	· User should be registered and logged into the system to access the food lists · System should be correctly connected to the database · Food catalogue should be populated with wide variety of food items

<table>
<tr><td>Success Guarantee (Postconditions)</td><td colspan="2">· User successfully views the food lists, that can be filtered based on preferences
· User can successfully select and add the food items from the list to their meals or diet plan</td></tr>
<tr><td>Main Success Scenario</td><td>Actor Action

1. User logs into the system.

1. User navigates to the “List of Foods” section or searched food items

6. User views the food list and scrolls through it.

7. User selects one or more food items to view their nutritional information or to log into their meal</td><td>System Response

2. System checks User’s credentials.

4. System retrieves the full list of available food items from the food catalogue or directly from the database

5. System displays the list of foods, categorized (e.g. fruits, vegetables, proteins etc.) and sorted alphabetically</td></tr>
<tr><td>Extensions</td><td colspan="2">· 5a. If the food list is large, the system allows the user to filter the list by categories (e.g. by food group, calorie count, macronutrients)
· 3a. If the user searches for a food item that is not available, the system suggests similar items or requests the user to add the food item
· 4a. If the system fails to retrieve the list due to an error (e.g. database connection issue), it displays an error message and tells the user to try again later</td></tr>
</table>

Special Requirements	· Food list should be filterable and searchable on the basis of user's preferences · System should ensure that the food list is up-to date and correctly matches user's preferences · Data should be displayed in a user-friendly format, with the option to view detailed nutritional information for each food item in the list
Technology & Data variations list	Mobile Phone, internet connections and the app for using the features
Frequency of Occurrence	Frequently, users may access it multiple times a day when logging in meal or planning a diet plan

1.9.4. Track Progress

Use case ID	UC04
Use case name	Track Progress
Scope	Dietary Management System
Level	User goal
Primary Actor	User, Dietitian

Stakeholders and Interests	· **Admin:** ensures that the system accurately collects and presents user data · **User:** Wants to monitor their health, fitness, or dietary progress over time (e.g. weight, calorie intake, exercise etc.) to stay on track of their goal · **Dietitian:** assesses user's progress by reviewing their logged data (e.g. meals, exercise, weight) to provide personalized feedback and make necessary adjustments to the user's diet plan · **System:** Responsible for collecting, storing, and displaying progress data in an understandable format
Preconditions	· User is successfully logged into the system and has availed the Dietitian feedback feature · User logs meals, exercises and other health related data regularly · System should be correctly connected to the database, and should be capable of tracking and analyzing data over time · Dietitian has access to user's data (selectively) to view and analyze user's progress
Success Guarantee (Postconditions)	· User successfully maintains his daily logs and is able to view his fitness, health progress via graphs, trends or reports · Dietitian reviews the user's progress and offers personalized recommendations

	Actor Action	System Response
Main Success Scenario	1a. User logs into the system.	2a. System checks User's credentials.
		3a. System retrieves the user's logged data
	4a. User navigates to the "Track Progress" section	
	5a. User selects a specific time frame to view their progress within that period	6a. System displays the user's progress using graphs, charts or reports (e.g. calories, steps, weight changes etc.) over that period
	1b. Dietitian logs into the system.	2b. System checks Dietitian's credentials
	3b. Dietitian goes to his list of user customers' availing his consultation and selects the specific user.	4b. System retrieves the selected user's specific logged data
	5b. Dietitian navigates to the user's Track Progress section	6b. System displays the user's progress via graphs, charts, reports over a certain period of time
	7b. Dietitian reviews and analyzes the progress of the user and provides feedback or recommendations	8b. System sends the Dietitian's feedback to the user's account.

Extensions	· 2a. If user hasn't logged in enough data to show progress, the system notifies the user and dietitian that there isn't sufficient data to track progress · 5b. If the dietitian does not have the required permissions to view the user's data, the system restricts access and prompts the dietitian to request authorization from the user · 7b. If the user's progress shows regression (e.g., weight gain when the goal was weight loss), the dietitian might send an urgent message, suggesting a consultation or changes in the plan
Special Requirements	· The system should allow users and dietitians to set custom goals and track them over time. · Both the user and dietitian should be able to communicate seamlessly through the system to discuss the progress and provide feedback. · The system should present progress data in an easy-to-understand format (e.g., graphs, trends, and reports)
Technology & Data variations list	Mobile Phone, internet connections and the app for using the features
Frequency of Occurrence	· Daily or weekly, depending on how frequently the user logs meals and exercises. · The dietitian may review progress periodically

1.9.5. **Log In/Register**

Use case ID	UC05

Use case name	Log in/ Register
Scope	Dietary Management System
Level	User goal
Primary Actor	User, Dietitian
Stakeholders and Interests	· **Admin:** logs in to make changes in the food catalogue and ensure up-to date information and capture necessary user information · **User:** creates a new account or logs in to gain access to the system's features (e.g. logging meals, viewing diet plans). · **Dietitian:** creates new account or logs in to communicate and provide consultation to the users (e.g. personalized meals, exercise routines) · **System:** responsible for securely capturing and storing user information, validating inputs, and creating a new user profile.
Preconditions	· User or Dietitian should not have an existing account when registering · System should be correctly connected to the database to confirm credentials when logging in · Registration and log in system should be functional and accessible
Success Guarantee (Postconditions)	· New user's or dietitian's account is successfully created and entered in the database · User/ Dietitian successfully logs into his account and their history is maintained in the account · New user can access the features of the system after registration

Main Success Scenario	**Actor Action** 1.User/ Dietitian navigates to the registration page 3.User/ Dietitian enters the relevant information 1. User/Dietitian submits the registration/login details 6.User/Dietitian enters the code sent to the email for verification	**System Response** 2. System prompts the user/dietitian to enter relevant information (e.g. name, password, email, role-user/ dietitian) 5. System validates the input and verifies the email, by sending a code to the email 7. System after verifying lets the user/dietitian log into the system and enters them into the database in case of registration. 7. Personalized features are made available to the user/dietitian by the system
Extensions	● 5a. If email is already registered in case of registration the system prompts the forgot password window ● 5b. If the user/dietitian provides invalid data (e.g., weak password, incorrect email format), the system prompts them to correct the information ● 4a. If the user/dietitian has forgotten their password during loging in, the system provides a "Forgot Password" option, allowing them to reset their password via email.	
Special Requirements	● The system must use encrypted communication (e.g., HTTPS) to ensure the security of personal data and login credentials. ● Passwords must be securely hashed and stored in the system. ● The system should offer a two-factor authentication (2FA) option to enhance security.	

	● Email verification should be mandatory before granting access to the system
Technology & Data variations list	· Mobile Phone, internet connections and the app for using the features · Users or dietitians may log in using third-party authentication (e.g., Google, Facebook) if the system supports social logins. · The system could allow guest browsing for limited features (e.g., viewing diet plans or recipes) without requiring registration, but full functionality is restricted to registered users and dietitians.
Frequency of Occurrence	Very frequently

1.9.6. **Generate Diet Plan**

Use case ID	UC06
Use case name	**Generate Diet Plan**
Scope	Dietary Management System
Level	**Sub-function**

Primary Actor	**System**	
Stakeholders and Interests	● **User**: Receives diet plans to follow based on goals. ● **Dietician**: Uses system-generated plans as a starting point for personalized recommendations. ● **System**: Generates accurate diet plans based on user goals, preferences, and health data.	
Preconditions	● User must have provided health data (e.g., weight, goals, dietary preferences). ● System must have access to meal and nutritional data.	
Success Guarantee (Postconditions)	● Diet plan is generated based on user's health and dietary preferences. ● User can access the plan and use it to track meals.	
Main Success Scenario	Actor Action 1. User provides health data and goals. 3. User submits data for plan generation. 5. User reviews the generated plan. 7. Users approve or request modifications to the plan. 9. User logs out after reviewing the plan.	System Response 2. System analyzes user data and preferences. 4. System generates a personalized diet plan. 6. System displays the plan with daily meals and nutritional targets. 8. System updates the plan based on user input. 10. System saves the session and logs the user out.

Extensions	• **7a**. If user preferences conflict with health goals, the system suggests alternatives or warns of potential issues. • **3a**. If the user has incomplete data, the system prompts them to provide the missing information.
Special Requirements	• System must use accurate health and nutritional data. • Diet plans should be customizable based on user preferences and goals.
Technology & Data variations list	None
Frequency of Occurrence	• Varies based on user activity and health goals.

1.9.7. **Review Diet Plans**

Use case ID	UC07
Use case name	**Review Diet Plans**
Scope	Dietary Management System

Level	**User Goal**
Primary Actor	**Admin, Dietician**
Stakeholders and Interests	• **Admin** Reviews and ensures all diet plans are appropriate and meet health standards. • **Dietician** Reviews diet plans to personalize them for users based on health data. • **User** receive accurate and effective diet plans. • **System** Provides access to diet plans and ensures they are up-to-date.
Preconditions	• Admin/Dietician must be logged into the system. • Diet plans must exist in the database. • The system must be connected to relevant health and nutritional data.
Success Guarantee (Postconditions)	• Admin/Dietician reviews the diet plan. • Plans are either approved, modified, or flagged for review based on user requirements.

<table>
<tr><td>Main Success Scenario</td><td>Actor Action

1. Admin or Dietician logs into the system.

3. Admin/Dietician selects "Review Diet Plans."

5. Admin/Dietician selects a specific diet plan to review.

7. Admin/Dietician reviews each meal in the diet plan.

9. Admin/Dietician modifies the plan if necessary.

11. Admin/Dietician approves or flags the plan for further review.

13. Admin/Dietician logs out.</td><td>System Response

2. System verifies the credentials and displays the dashboard.

4. System retrieves all available diet plans for review.

6. System displays the details of the selected plan (meals, nutritional breakdowns, user goals).

8. System highlights any potential issues (e.g., caloric surplus/deficit).

10. System saves the modifications and updates the plan accordingly.

12. System confirms the plan's status and notifies the user of updates.

14. System saves the session and updates logs.</td></tr>
<tr><td>Extensions</td><td colspan="2">● 7a. If the diet plan is incomplete, the system prompts the Admin/Dietician to complete it before proceeding.
● 9a. If a plan is flagged as unsuitable for a user’s goals, the system suggests modifications based on health data.</td></tr>
<tr><td>Special Requirements</td><td colspan="2">● The system must present diet plans in an easily reviewable format.
● The review process should support detailed nutritional breakdowns and health checks.</td></tr>
</table>

Technology & Data variations list	None
Frequency of Occurrence	• As often as new plans are created or modified.

1.9.8. **Add New Meal**

Use case ID	UC08
Use case name	**Add New Meal**
Scope	Dietary Management System
Level	**Sub-function**
Primary Actor	**User, Admin**

<table>
<tr><td>Stakeholders and Interests</td><td colspan="2">User: Can create personalized meals if not available in the system.
Admin: Ensures that the food database is regularly updated with new meals.
System: Manages and stores meal information efficiently.</td></tr>
<tr><td>Preconditions</td><td colspan="2">User or admin must be logged into the system.
The food database should be active and modifiable.</td></tr>
<tr><td>Success Guarantee (Postconditions)</td><td colspan="2">New meal is successfully added to the system’s database.
Users can view the meal when logging it in the future.</td></tr>
<tr><td>Main Success Scenario</td><td>Actor Action

1. User or admin selects “Add new meal” from the menu.

3. User/admin inputs the meal name and ingredients.

5. User/admin verifies and confirms the meal details.

7. User/admin submits the new meal.

9. User or admin reviews the updated list of meals.

11. Admin verifies the new meal for system-wide access.</td><td>System Response

2. System prompts for meal information input.

4. System retrieves and displays nutritional information for the ingredients.

6. System calculates the total nutritional values for the meal.

8. System saves the new meal to the database.

10. System confirms the meal has been added successfully.

12. System makes the new meal available for all users.</td></tr>
</table>

Extensions	• **3a**. If some ingredients are missing in the system, the user/admin is prompted to manually input nutritional values. • **7a**. If the user tries to input an incomplete meal, the system prompts them to add the missing information.
Special Requirements	• Nutritional data must be accurate and standardized. • The system must support adding new food types as needed.
Technology & Data variations list	None
Frequency of Occurrence	• Occasionally, when new meals are not available in the system.

1.9.9. **View Existing Diet Plans**

Use case ID	UC09
Use case name	**Add New Meal**
Scope	Dietary Management System

Level	**Sub-function**	
Primary Actor	**User, Admin**	
Stakeholders and Interests	● **User**: Can create personalized meals if not available in the system. ● **Admin**: Ensures that the food database is regularly updated with new meals. ● **System**: Manages and stores meal information efficiently.	
Preconditions	● User or admin must be logged into the system. ● The food database should be active and modifiable.	
Success Guarantee (Postconditions)	● New meal is successfully added to the system's database. ● Users can view the meal when logging it in the future.	
Main Success Scenario	Actor Action 1. User or admin selects "Add new meal" from the menu. 3. User/admin inputs the meal name and ingredients. 5. User/admin verifies and confirms the meal details. 7. User/admin submits the new meal. 9. User or admin reviews the updated list of meals.	System Response 2. System prompts for meal information input. 4. System retrieves and displays nutritional information for the ingredients. 6. System calculates the total nutritional values for the meal. 8. System saves the new meal to the database. 10. System confirms the meal has been added successfully.

	11. Admin verifies the new meal for system-wide access.	12. System makes the new meal available for all users.
Extensions	● **3a**. If some ingredients are missing in the system, the user/admin is prompted to manually input nutritional values. ● **7a**. If the user tries to input an incomplete meal, the system prompts them to add the missing information.	
Special Requirements	● Nutritional data must be accurate and standardized. ● The system must support adding new food types as needed.	
Technology & Data variations list	None	
Frequency of Occurrence	● Occasionally, when new meals are not available in the system.	

1.9.10. **Log Meals**

Use case ID	UC10
Use case name	**Log Meals**

Scope	Dietary Management System
Level	**User Goal**
Primary Actor	**User**
Stakeholders and Interests	● **User**: Logs their meals to track daily caloric and nutritional intake. ● **System**: Stores and retrieves logged meal data and ensures accuracy.
Preconditions	● The user must be registered and logged into the system. ● The food database should be available with valid nutritional data. ● The user should have access to a device with an internet connection.
Success Guarantee (Postconditions)	● User's meal is successfully logged into the system. ● Nutritional intake for the day is updated and viewable by the user.

Main Success Scenario	Actor Action 1. User selects the option to log a meal. 3. User chooses food items or searches for specific food. 5. User inputs serving size and meal details. 7. User submits the meal log. 9. User reviews the updated daily nutritional summary. 11. User logs out after finishing the meal log.	System Response 2. System displays the available food items and meal input options. 4. System retrieves nutritional information for the selected items. 6. System calculates the total nutritional values for the meal. 8. System saves the meal data and updates the user's daily nutritional intake. 10. System confirms successful meal logging and displays the updated values. 12. System saves the session and logs the user out.
Extensions	● **3a**. If the user cannot find a food item, the system suggests similar items. ● **7a**. If the internet connection is lost during submission, the system prompts the user to retry logging the meal once the connection is restored.	
Special Requirements	● The meal logging interface must be simple and quick to use. ● The system should store food items' nutritional data securely and efficiently.	
Technology & Data variations list	None	

Frequency of Occurrence	• Several times a day, depending on user activity.

1.9.11. **Search Diet Plans**

1.9.12. **Add Health Information**

Use Case ID	UC12	
Use Case Name	Add Health Information	
Scope	Dietary Management System	
Level	User Goal	
Primary Actor	User	
Stakeholders and Interests	**None**	
Preconditions	· The user must be logged in.	
Postconditions	The user successfully enters and submits their health information.	
Main Success Scenario	**Actor Action**	**System Response**
	2. Accepts the prompt.	1. Prompts user to fill in their health information.
	3. Fills out their health information e.g. weight, height, related health issues and allergies.	4. Saves the information in users' database for future reference.

Extensions	2a. User denies to fill the information. 4a. Possible crash in system. Information not saved.
Special Requirements	**None**
Technology and Data **Variation List**	**None**
Frequency of Occurrence	Once, either during registration or when the user chooses to.

1.9.13. **Record Exercise**

Use Case ID	UC13
Use Case Name	Record Exercise
Scope	Dietary Management System
Level	Sub-Function
Primary Actor	User
Stakeholders and **Interests**	**None**
Preconditions	· The user must be logged in. · The system should have a database of types of exercises and their general results/outcomes.
Postconditions	The exercise is logged on to the system and daily nutrition is updated.

Main Success Scenario	**Actor Action**	**System Response**
	1. Selects ‘log exercise’ option.	2. Provides options for exercises the user wants to do.
	3. Chooses which exercise they want to do.	4. Starts timer for chosen exercise.
	5. Stops the timer when theyre done.	6. Calculates calories burnt and other relevant information and updates users’ daily nutrition
Extensions	3a. The exercise does not exist. 4a. System fails to start timer. 6a. Incorrect calculations are made.	
Special Requirements	· Existing database of exercises and their results.	
Technology and Data **Variation List**	**None**	
Frequency of Occurrence	Every time the user exercises.	

1.9.14. **Search Food Items**

Use Case ID	UC14
Use Case Name	Search Food Items
Scope	Dietary Management System
Level	Sub-Function

<table>
<tr><td>Primary Actor</td><td colspan="2">User</td></tr>
<tr><td>Stakeholders and Interests</td><td colspan="2">None</td></tr>
<tr><td>Preconditions</td><td colspan="2">· A database of food items should already exists.
· User is logged in.
· System should have nutritional information about the food</td></tr>
<tr><td>Postconditions</td><td colspan="2">The food item user is looking for is chosen the logged on to their meal.</td></tr>
<tr><td>Main Success Scenario</td><td>Actor Action
1. Selects ‘Search for food’.

2. Searches a food name.

5. Selects the food theyre looking for.

8. User can choose to log the food or quit search.</td><td>System Response
3. Filters through database to find the food user is looking for.

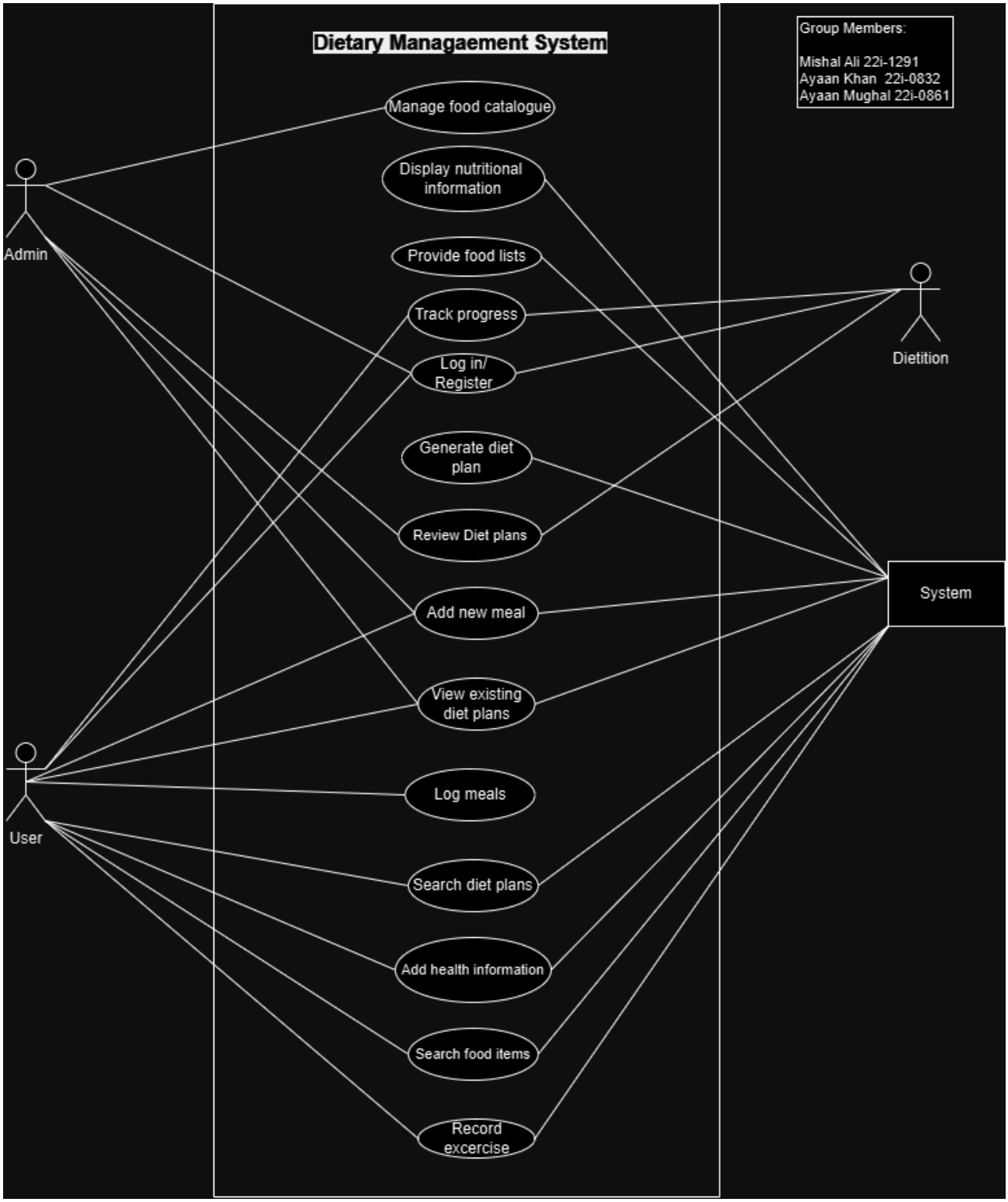
4. Displays the options of food through the filters.

6. Displays nutritional information.

7. Asks user if they want to log the food as a meal or snack.</td></tr>
<tr><td>Extensions</td><td colspan="2">3a. Food does not exist.
3b. User entered wrong information (spelling mistake, not a food etc.)
8a. (User chose to log the food) Food is added to their meal, system updates the daily meal data and nutritional intake.
8b. (User chose to quit) System navigates back to the main page.</td></tr>
<tr><td>Special Requirements</td><td colspan="2">· The system should display the food items in an easy to navigate manner.</td></tr>
</table>

	· There should be an extensive detail about the food item for the user to review.
Technology and Data **Variation List**	**None**
Frequency of Occurrence	Every time a user logs in a food item for their meal logging.

1.10. Use Case Diagram



Other Nonfunctional Requirements

1.11. Performance Requirements

The Dietary Management System must meet the following performance requirements for smooth functionality and user satisfaction:

1. ***Response Time***
    - *The system must respond to user actions (e.g., logging meals, generating diet plans, searching food items) within **2-3 seconds** under normal operating conditions.*
2. ***Data Retrieval***
    - *Retrieval of diet plans, food information, or user progress history must be completed in **> 3 seconds** from the database.*
3. ***Data Processing***
    - *Nutritional calculations for logged meals and exercise tracking should process data in **real-time**, ensuring that the information is immediately available for progress tracking.*
4. ***Mobile and Web Accessibility***
    - *Both mobile and web platforms should load the dashboard or home screen within **3-5 seconds***
5. ***Security and Data Encryption***
    - *Authentication and data encryption processes must not add more than **2 second** to login and data retrieval times.*
6. ***Data Storage***
    - *The system must efficiently handle large datasets (e.g., food catalogues, user logs) without exceeding a query processing time of **3 seconds** for the largest expected dataset size.*

1.12. Safety Requirements

The Dietary Management System has been designed with safety as a priority to ensure the security of user data and the accurate operation of the system. Below are the safety requirements that must be met to prevent loss, damage, or harm resulting from the use of the product:

3.2.1. Accuracy of Recommendations

The system must use validated algorithms and accurate nutritional databases to ensure reliable dietary and wellness recommendations. Misleading or incorrect data entry by users (e.g., erroneous calorie values) should trigger warnings or validations to minimize errors.

3.2.2. Integration Safeguards

When integrating with fitness wearables or health monitoring systems, only verified and compatible devices should be supported to prevent inaccurate data synchronization. Real-time data synchronization must include error-checking mechanisms to avoid propagating incorrect information.

3.2.3. User Safety and Well-being

Warnings on Extreme Data Inputs: The system must flag potentially harmful entries, such as overly restrictive calorie targets or excessive exercise durations, which could lead to unhealthy behaviors.

Disclaimer on Health Advice: Users must be informed that the application is a tool for tracking and planning and not a replacement for professional medical or nutritional advice. A disclaimer must be displayed prominently.

3.2.4. Preventive Actions

Prevent unauthorized data modifications by restricting certain operations to administrative users. Prohibit actions that could lead to system vulnerabilities, such as exposing the application to untrusted third-party extensions or scripts.

3.2.5. Safety Certifications

The system should adhere to international standards for information security management. Any nutritional guidelines or recommendations should align with recognized standards, such as those from WHO (World Health Organization) or USDA (United States Department of Agriculture), to ensure credibility and safety.

1.13. Security Requirements

The Dietary Management System prioritizes the security and privacy of user data to ensure safe and reliable use. Below are the specific security requirements to address potential vulnerabilities and safeguard sensitive information:

3.3.1. Data Protection

Encryption of Data: All sensitive data, including personal information, dietary logs, and health metrics, must be encrypted both in transit and at rest (

Data Anonymization: Where applicable, user data should be anonymized to protect individual identities, especially in analytics or reporting features.

3.3.2. User Authentication

Multi-Factor Authentication (MFA): Users must authenticate themselves using multiple factors, such as a combination of passwords and one-time codes sent via email or SMS.

Role-Based Access Control (RBAC): Access to specific features and data must be granted based on user roles (e.g., administrator, standard user).

Password Requirements: Enforce strong password policies, including a minimum length, complexity requirements, and periodic updates.

3.3.3. Data Breach Prevention

Intrusion Detection and Prevention: Implement monitoring tools to detect and block unauthorized access attempts or data breaches.

Audit Logs: Maintain detailed logs of user activities, including login attempts, data modifications, and administrative actions, to enable forensic investigations in case of security incidents.

3.3.4. Secure Integration

API Security: All external integrations (e.g., with fitness wearables or third-party health monitoring systems) must use secure APIs with token-based authentication.

Sandboxing: Integrations should be sandboxed to prevent malicious or unintended actions from affecting the main system.

3.3.5. Privacy Requirements

User Consent: Explicit consent must be obtained from users before collecting, processing, or sharing personal data.

Data Minimization: Only collect data necessary for the application's functionality to reduce exposure to security risks.

1.14. Software Quality Attributes

The Dietary Management System is designed to adhere to high-quality standards that ensure user satisfaction, efficient development, and maintainable performance. Below are the key software quality attributes prioritized for this project, with specific, measurable objectives where applicable:

1. Adaptability

Requirement: The system must accommodate future enhancements, such as integrating additional fitness wearables, new dietary guidelines, or user-requested features.

Metric: New integrations should be implemented with minimal changes to existing modules, ideally in under two weeks.

2. Availability

Requirement: The system must be available for use 99.9% of the time during operational hours.

Metric: Downtime should not exceed 8 hours per year, with proactive monitoring for server health.

3. Correctness

Requirement: The system must accurately calculate total calories, nutritional breakdowns, and adherence to dietary plans.

Metric: Discrepancies in calculations should occur in less than 1 in 10,000 cases, verified through automated testing.

4. Flexibility

Requirement: Users must have the ability to customize meal plans, log exercises, and adjust preferences without requiring technical assistance.

Metric: 90% of customization tasks should be completed in under 5 clicks.

5. Interoperability

Requirement: The application should integrate seamlessly with fitness wearables (e.g., Fitbit, Apple Watch) and third-party health platforms.

Metric: Integration should use standardized APIs and successfully sync data within 5 seconds.

6. Maintainability

Requirement: The system should have a modular architecture, enabling quick updates or fixes to individual components without disrupting other functionalities.

Metric: 80% of bug fixes or enhancements should be deployable within 3 days.

7. Portability

Requirement: The application must run seamlessly on Windows PCs (target platform) and be compatible with major browsers (e.g., Chrome, Firefox, Edge).

Metric: Testing should confirm full compatibility across these platforms, with no critical UI or functional issues.

8. Reliability

Requirement: The system must handle up to 1,000 concurrent users without performance degradation.

Metric: The response time for core functionalities (e.g., meal logging, retrieving diet plans) should not exceed 1 second under maximum load.

9. Reusability

Requirement: Components like the database connection manager, authentication module, and API handlers should be reusable across other projects.

Metric: Reused components must require less than 10% modification for integration into new systems.

10. Robustness

Requirement: The system must handle unexpected inputs or errors gracefully without crashing.

Metric: Automated testing should confirm 100% exception handling coverage for all user inputs.

11. Testability

Requirement: The system should include automated test cases covering at least 90% of the codebase.

Metric: Testing tools (e.g., JUnit) must verify all key functionalities during development and before deployment.

12. Usability

Requirement: The user interface must be intuitive, requiring minimal learning effort.

Metric: A usability study should confirm that 80% of new users can complete core tasks (e.g., adding a meal plan) within 5 minutes of first use.

1.15. Business Rules

***User Roles and Permissions**:*

- ***Admin**:*
  - *Can manage all aspects of the system, including adding, editing, and deleting users, diet plans, and food items.*
  - *Can approve or reject user-generated diet plans.*
  - *Has access to all user data, including meal logs and health information.*
- ***Dietitian**:*
  - *Can create and modify diet plans based on user goals and health information.*
  - *Can provide personalized dietary advice to users.*
  - *Cannot manage users or delete items from the system.*
- ***User (Regular User)**:*
  - *Can create their own meal logs, track their diet, and access their personalized diet plans.*
  - *Cannot create or modify diet plans for others.*
  - *Cannot access the admin or dietitian functionalities.*
  - *Can consult with a dietitian for personalized advice.*

***User Authentication**:*

- *Users, dietitians, and admins must log in to access their respective features.*

- *A user must register before logging in, providing a valid email and password.*
- *If the user forgets their password, they must be able to request a password reset through their registered email.*
- *A user is limited to one account, and no two users can share the same email address.*

***Diet Plan Management**:*

- *Users can create their own diet plans by selecting from a catalog of food items, but they must be approved by an admin before they can be shared with others.*
- *Diet plans created by users or dietitians must be aligned with the user's goals (e.g., weight loss, muscle gain).*
- *Admins have the final approval to make any diet plan public and visible to all users.*

***Food Item Management**:*

- *Food items are added and categorized by the admin or dietitians. Only the admin can delete food items from the system.*
- *Users can search for food items by name, view nutritional information, and add them to their meal logs or diet plans.*
- *Admins can add or edit nutritional information for food items.*

***Meal Logging**:*

- *Users must log meals for each day they wish to track, including the food items they consumed.*
- *Each meal log must include the date, food items, and quantities consumed.*
- *Users can only log meals for the current day or past days, but not future dates.*

***Health Information**:*

- *Health information (e.g., weight, height, age) must be provided during user registration or can be updated at any time by the user.*
- *Users must enter health information correctly to ensure personalized diet recommendations are accurate.*
- *Dietitians can access health information to provide tailored dietary advice but cannot modify the health details of a user.*

***Data Integrity and Access Control**:*

- *Only the user or an admin can modify a user’s account details (e.g., name, email, password).*
- *Users can only access their own meal logs, diet plans, and health information. Admins can access all users' data.*
- *Data backups should be done periodically to ensure that user and meal logs are preserved in case of system failure.*

***Diet Plan Search and Filtering**:*

- *Users and dietitians can search for diet plans based on criteria like name, goal type (e.g., weight loss, muscle gain), and calorie content.*
- *Diet plans that are not approved by an admin cannot be shared with other users or searched for in the system.*
- *All diet plans visible to users must comply with the dietary goals defined by the system and be tailored to different user preferences (e.g., vegetarian, gluten-free).*

1.16. Operating Environment

Hardware Platform:

Minimum Requirements:
Processor: Intel Core i3 or equivalent
RAM: 4 GB (8 GB recommended)
Storage: 500 MB of free disk space for application installation and database files
Display: 1280x720 resolution or higher
Operating System:

Supported Platforms:

Windows: Windows 10 or later (64-bit preferred)

Software Components:

Java Runtime Environment (JRE):
Version 17 or later (JDK required for development purposes)

Database:
PostgreSQL Server

JavaFX:
JavaFX 17 or later for UI components

Development IDE (optional for developers):
Eclipse
IntelliJ IDEA Community/Ultimate Edition

Networking Requirements:

Local database access for standalone installations without internet.

Compatibility Requirements:

Should coexist peacefully with antivirus software such as Windows Defender or third-party tools.
No conflicts with popular web browsers or standard firewall configurations.

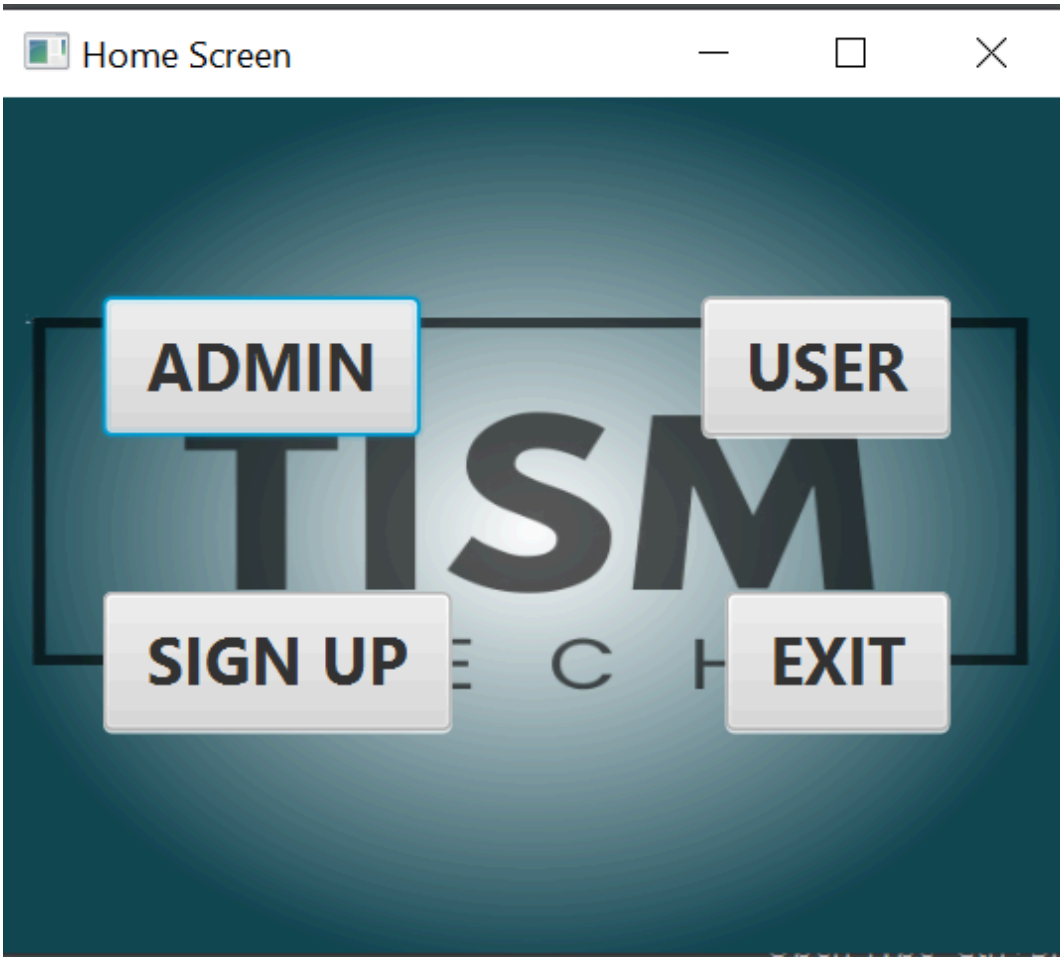
Security and Middleware:

Windows Authentication for secure access to SQL Server in production.

1.17. User Interfaces

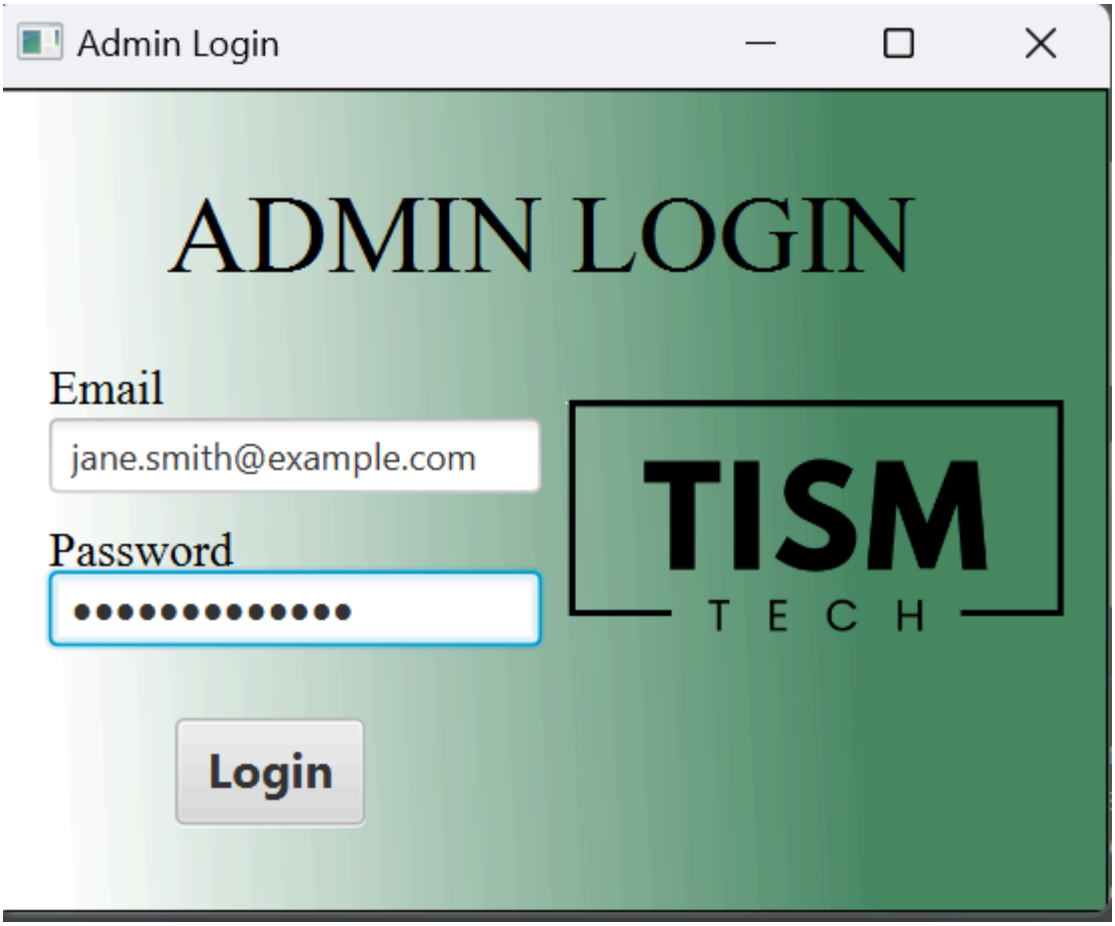
The user interface for the Dietary Management System (DMS) is designed to be intuitive and user-friendly, ensuring that users can easily monitor their diet and track their meals, exercise, and nutritional intake. It adheres to modern GUI standards, prioritizing accessibility and responsiveness across devices (e.g., smartphones, tablets, desktops).

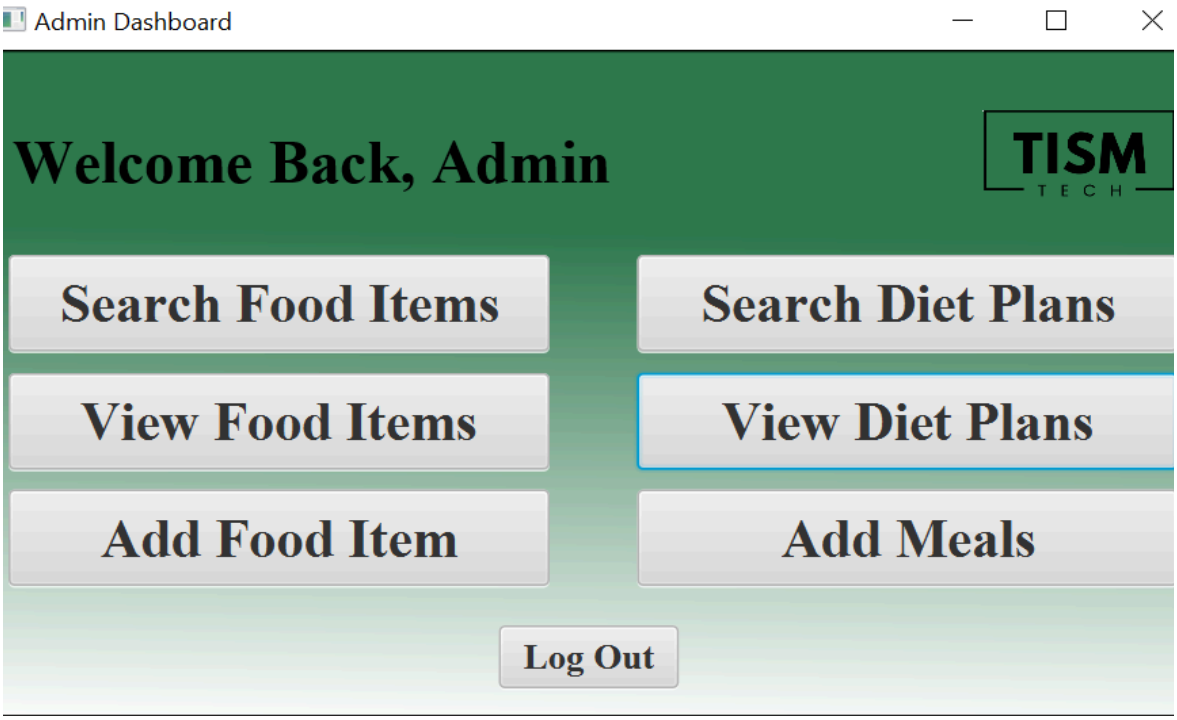
1. **Homescreen:**
   The home screen is easy to use and understand with the basics of the application and displays the company name as the background. The font is kept large enough to easily read.



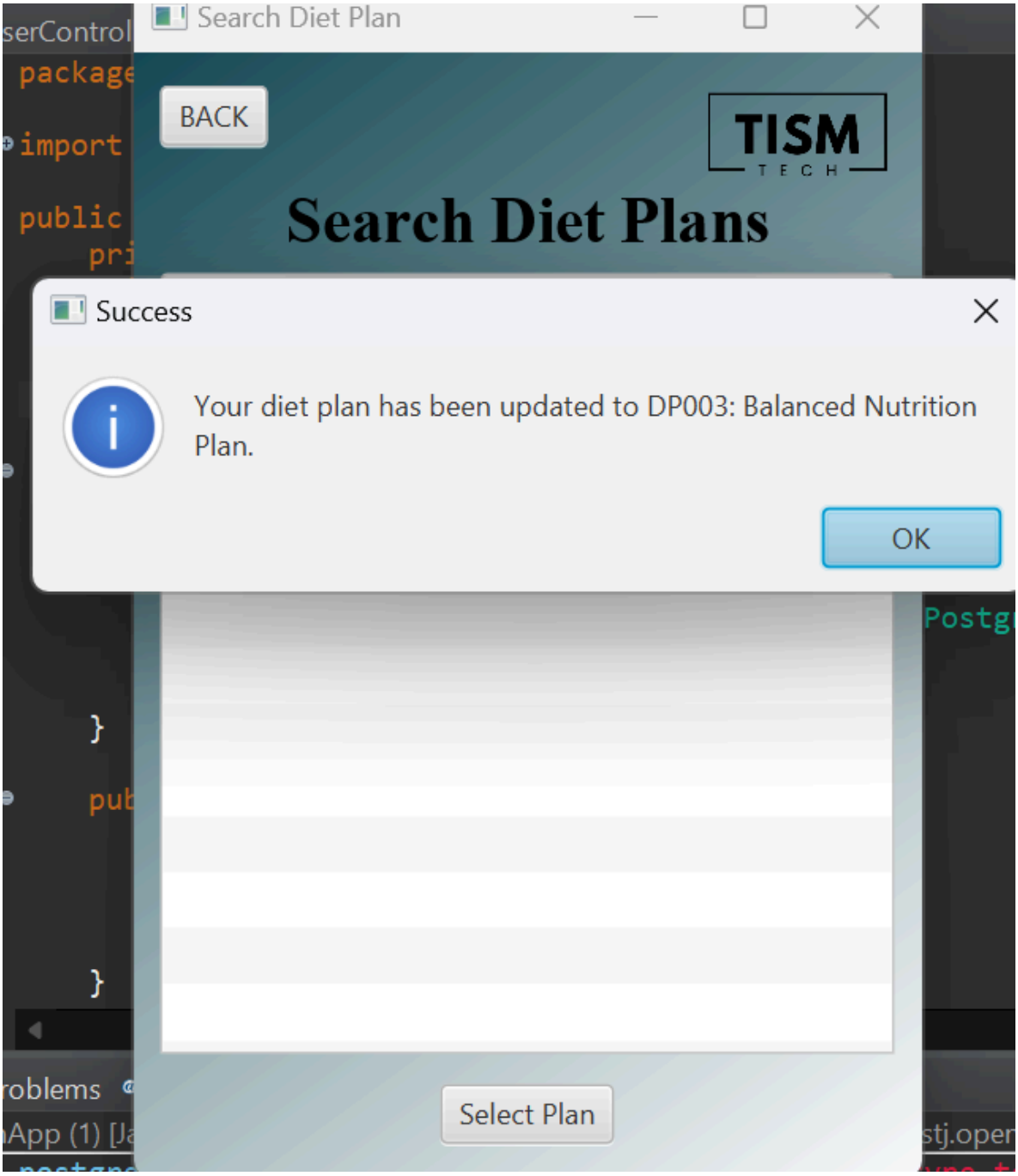
2. Admin User interfaces:

The admin User interfaces follow a consistent background color to differentiate. The login button verifies the admin's credentials from the database and directs to the admin dashboard.



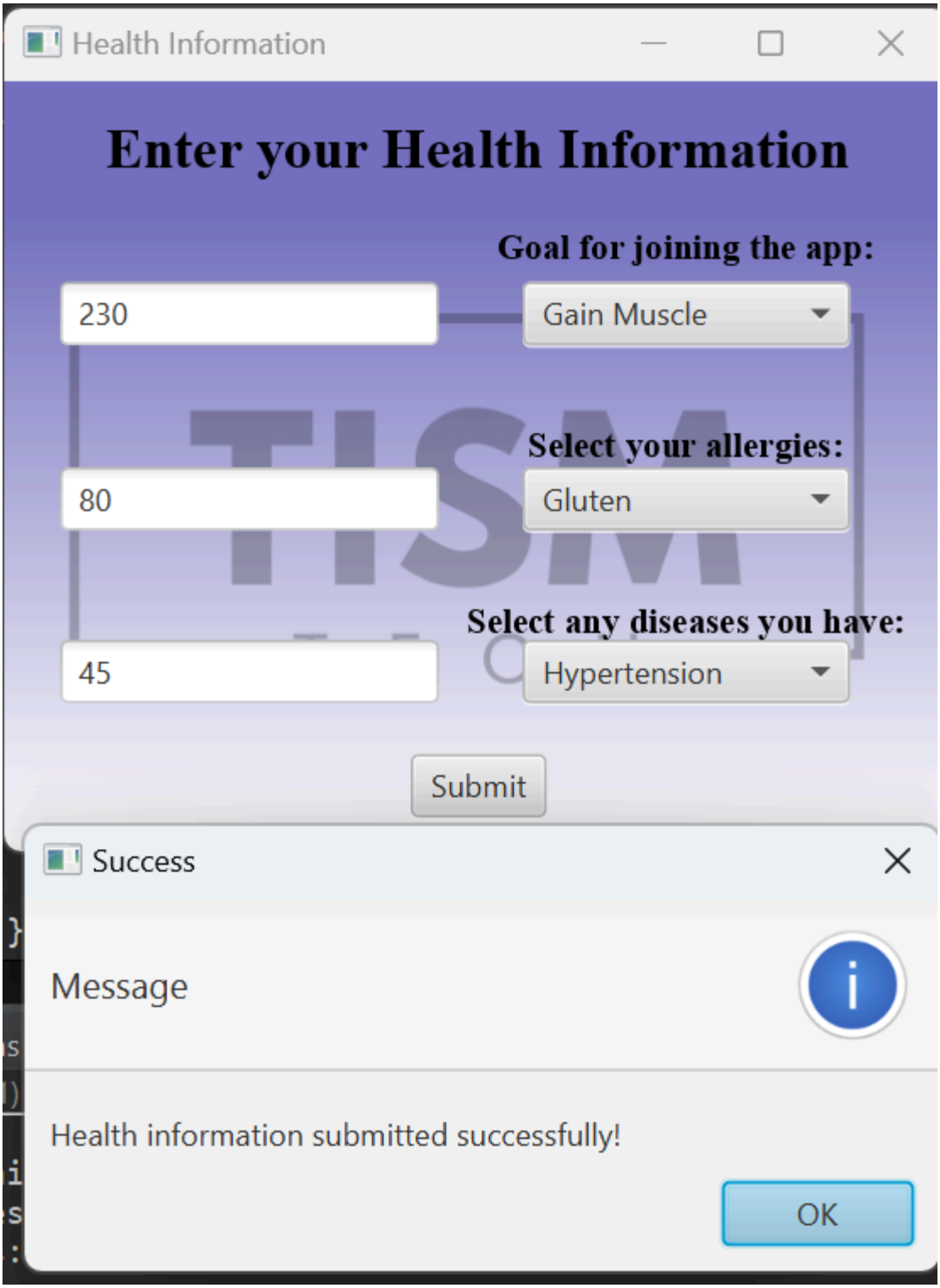


For more convenient search the search diet plan searches alphabet to alphabet as you type . Admin can perform the CRUD operations on the diet plans, hence the pop up shows the confirmation of the updation of a diet plan in the database.



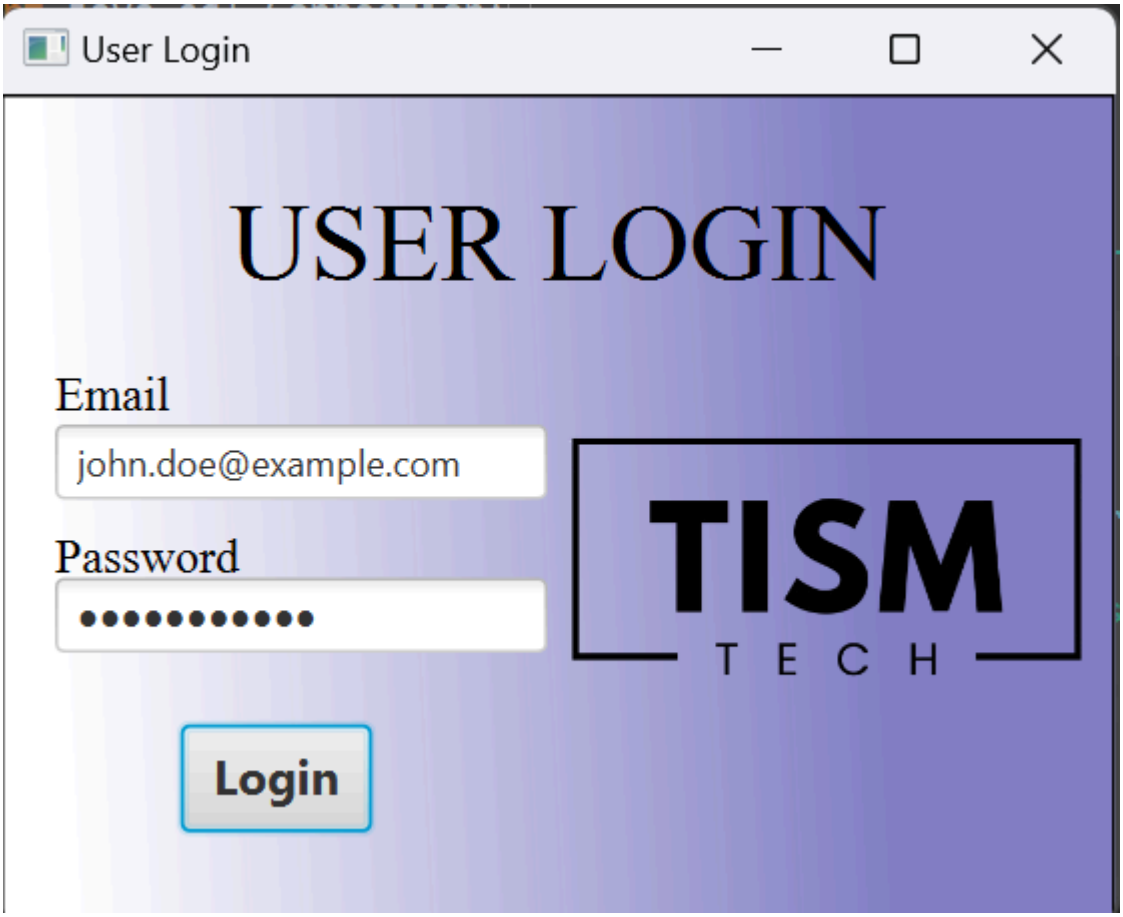
3. **Signup Interfaces:**
   Only users can sign up for the app as the admins will already be given their account details by the company. New users details are added in the Users table in the database.

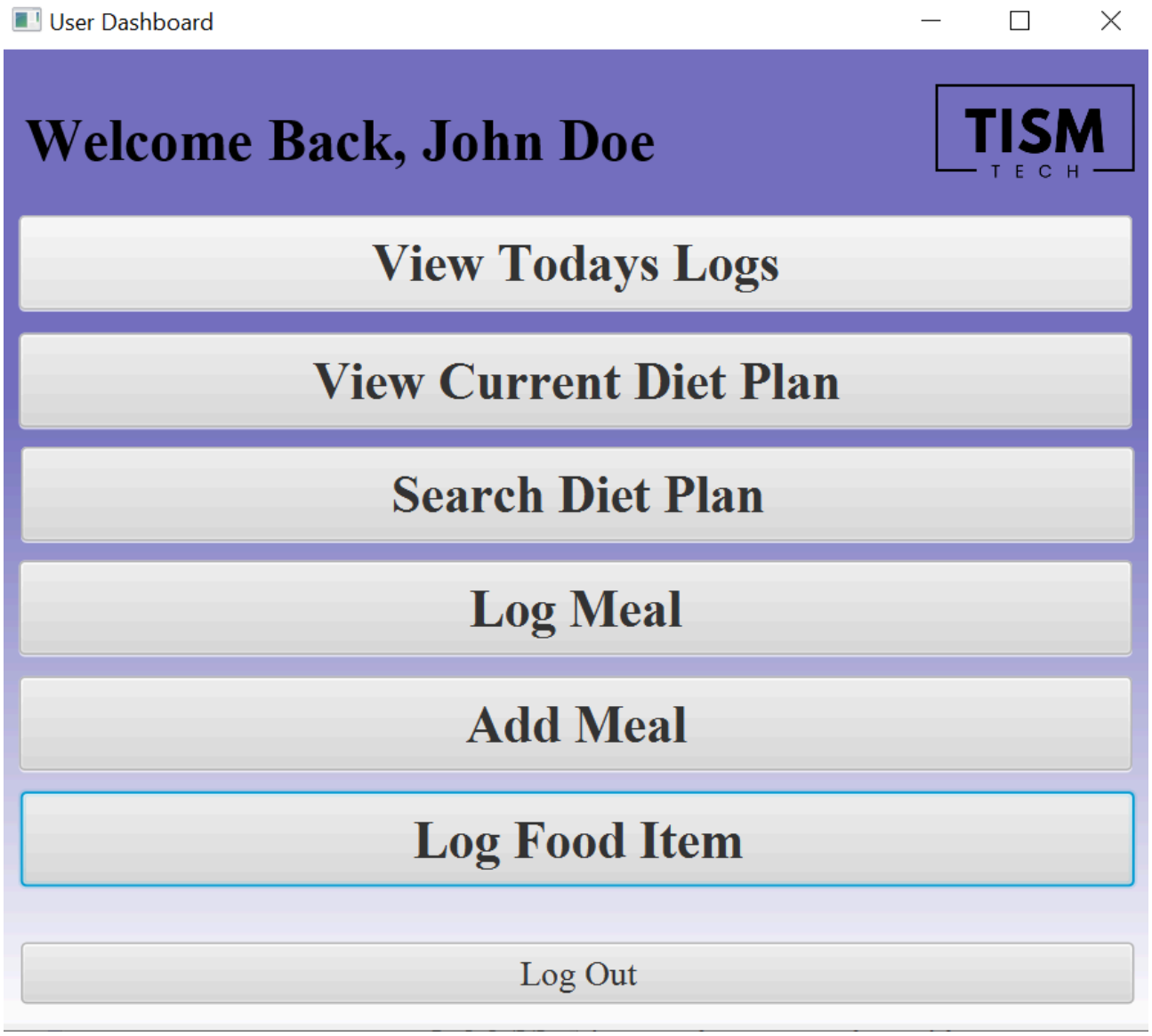
When a new user signs up , he is required to enter his health information such as allergies, weight and goals to customize his experience. This health info helps filter out meals and diet plans while searching.



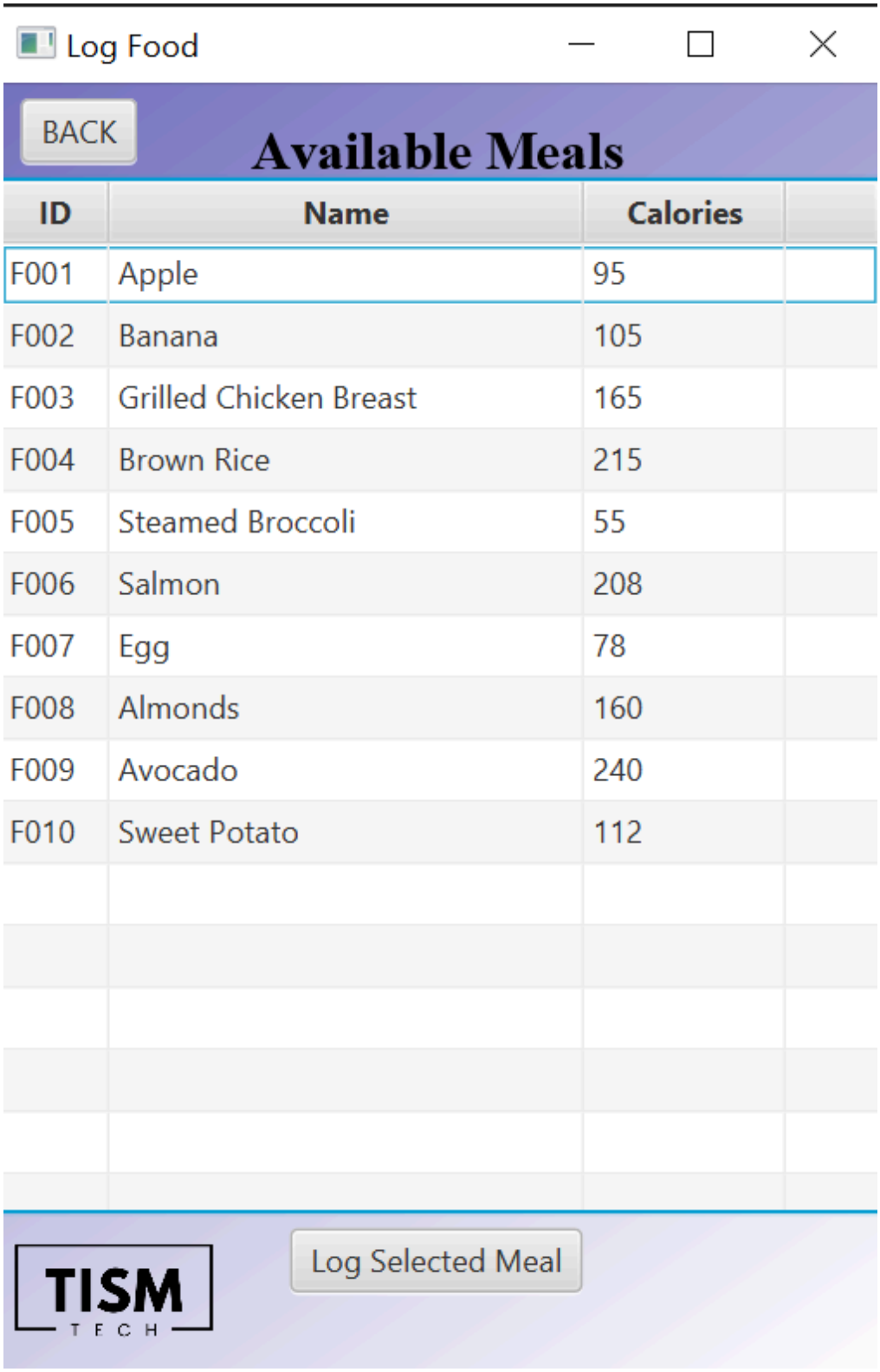
4. User Interfaces:

The users' interfaces follow a consistent background color to differentiate. The login button verifies the user's credentials from the database and directs to the user dashboard.

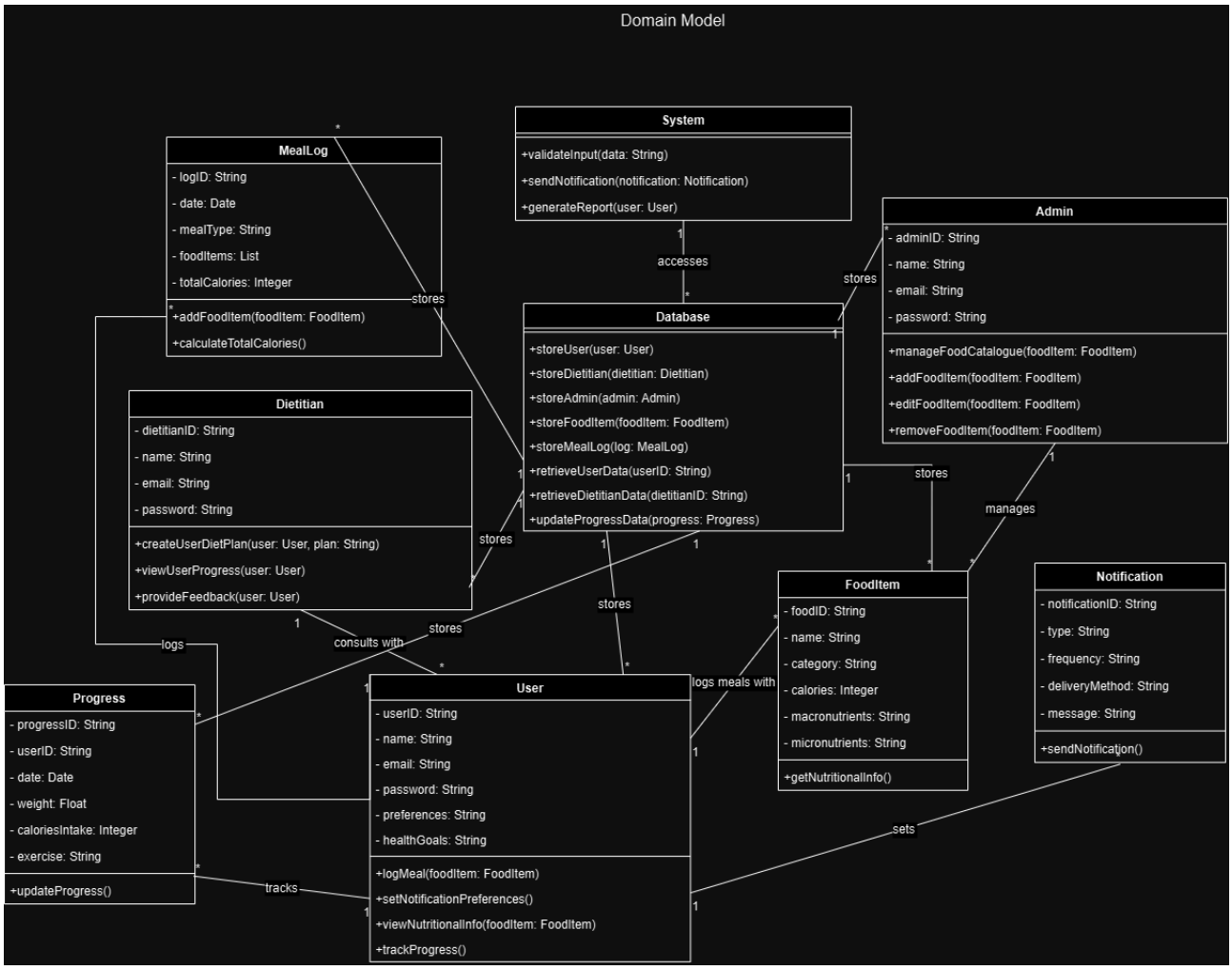

User Login
USER LOGIN
Email
john.doe@example.com
Password
TISM
TECH
Login



The user dashboard gives different options for the user such as logging his/her meal, viewing his/her diet plan or tracking his/her progress through the Today Logs button.
Similarly the back to dashboard and back buttons provide easy navigation between different pages/ screens for user conveiniece.

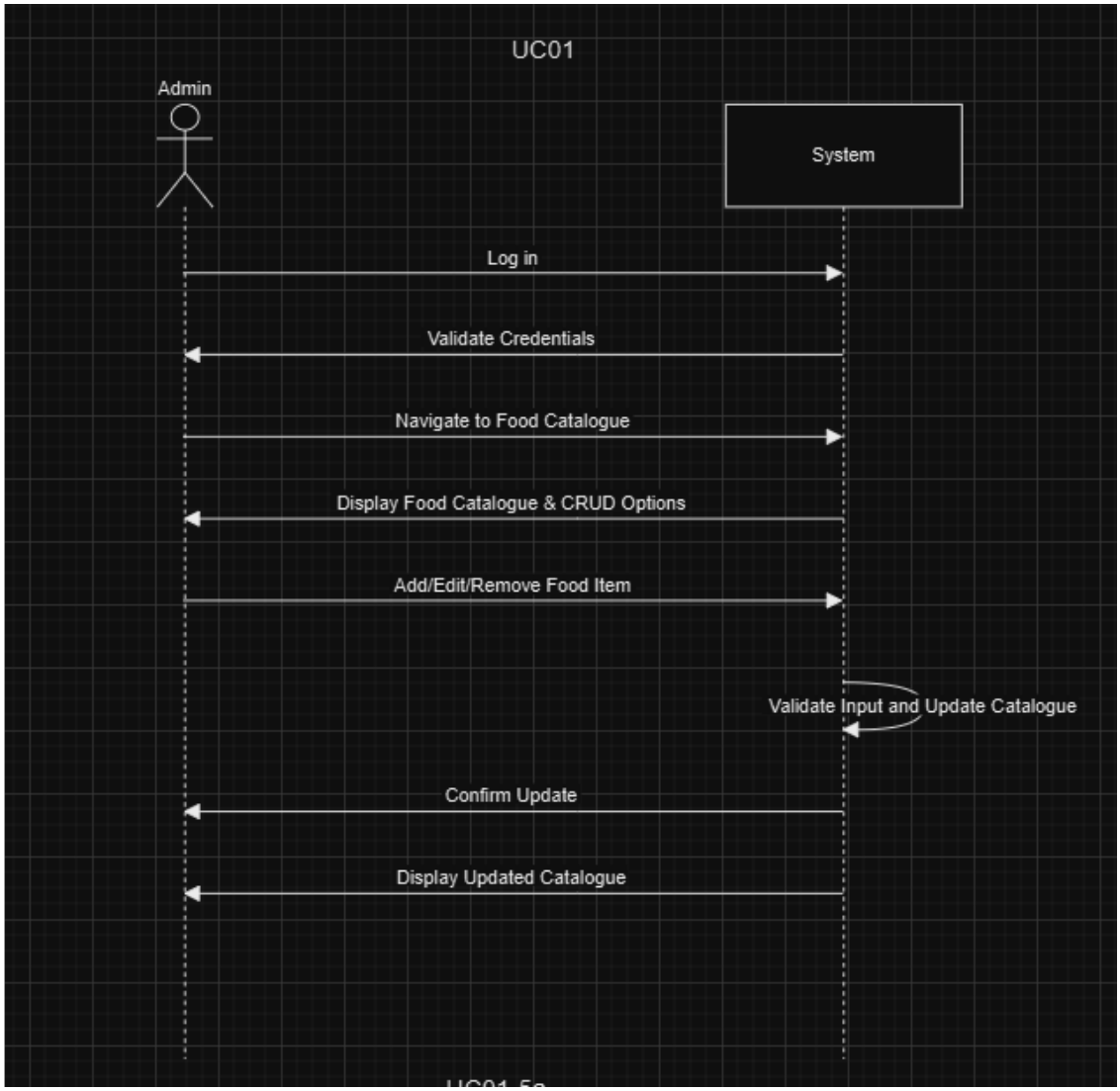

Log Food
BACK
Available Meals
ID
Name
Calories
F001
Apple
95
F002
Banana
105
F003
Grilled Chicken Breast
165
F004
Brown Rice
215
F005
Steamed Broccoli
55
F006
Salmon
208
F007
Egg
78
F008
Almonds
160
F009
Avocado
240
F010
Sweet Potato
112
TISM
TECH
Log Selected Meal

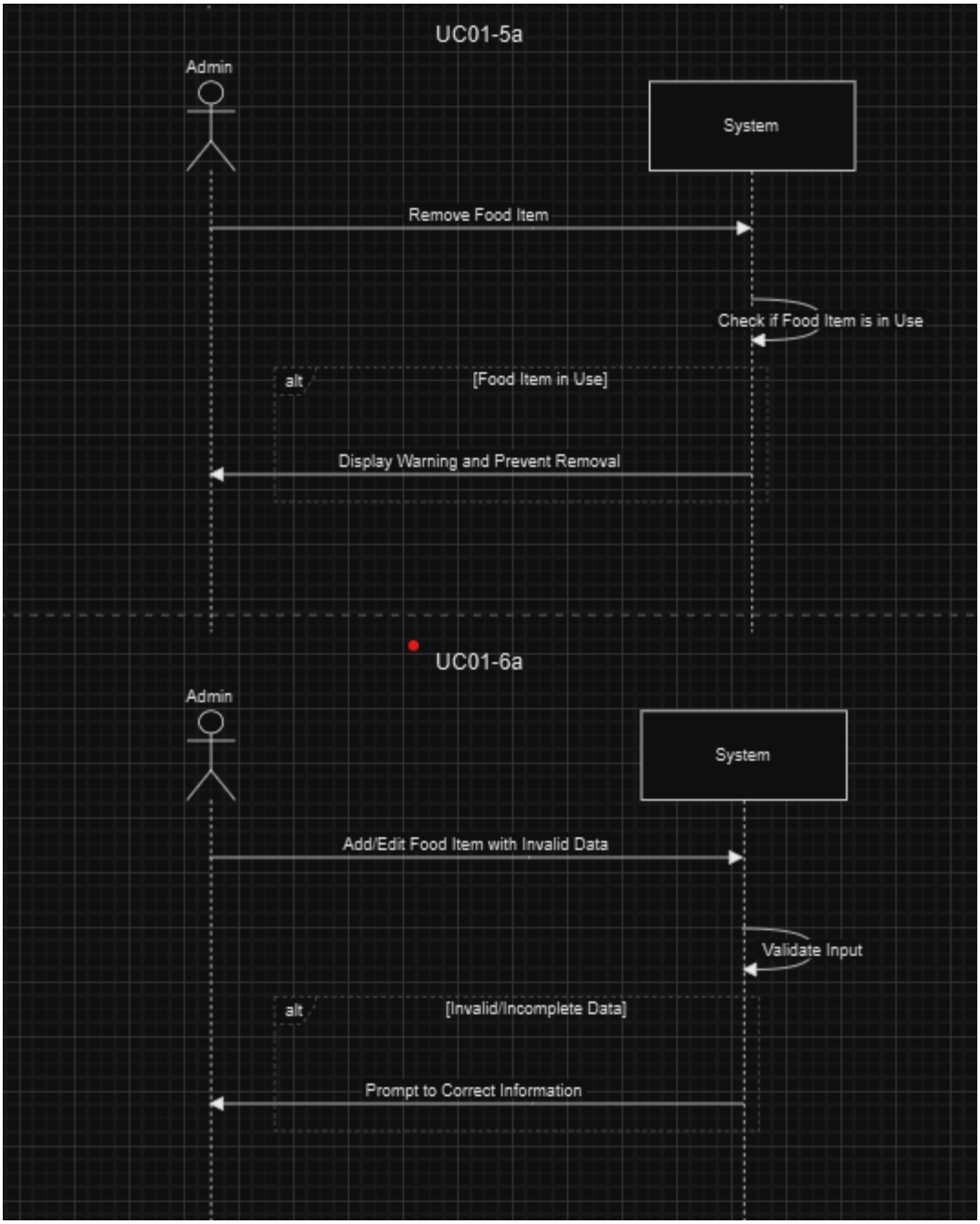
Domain Model



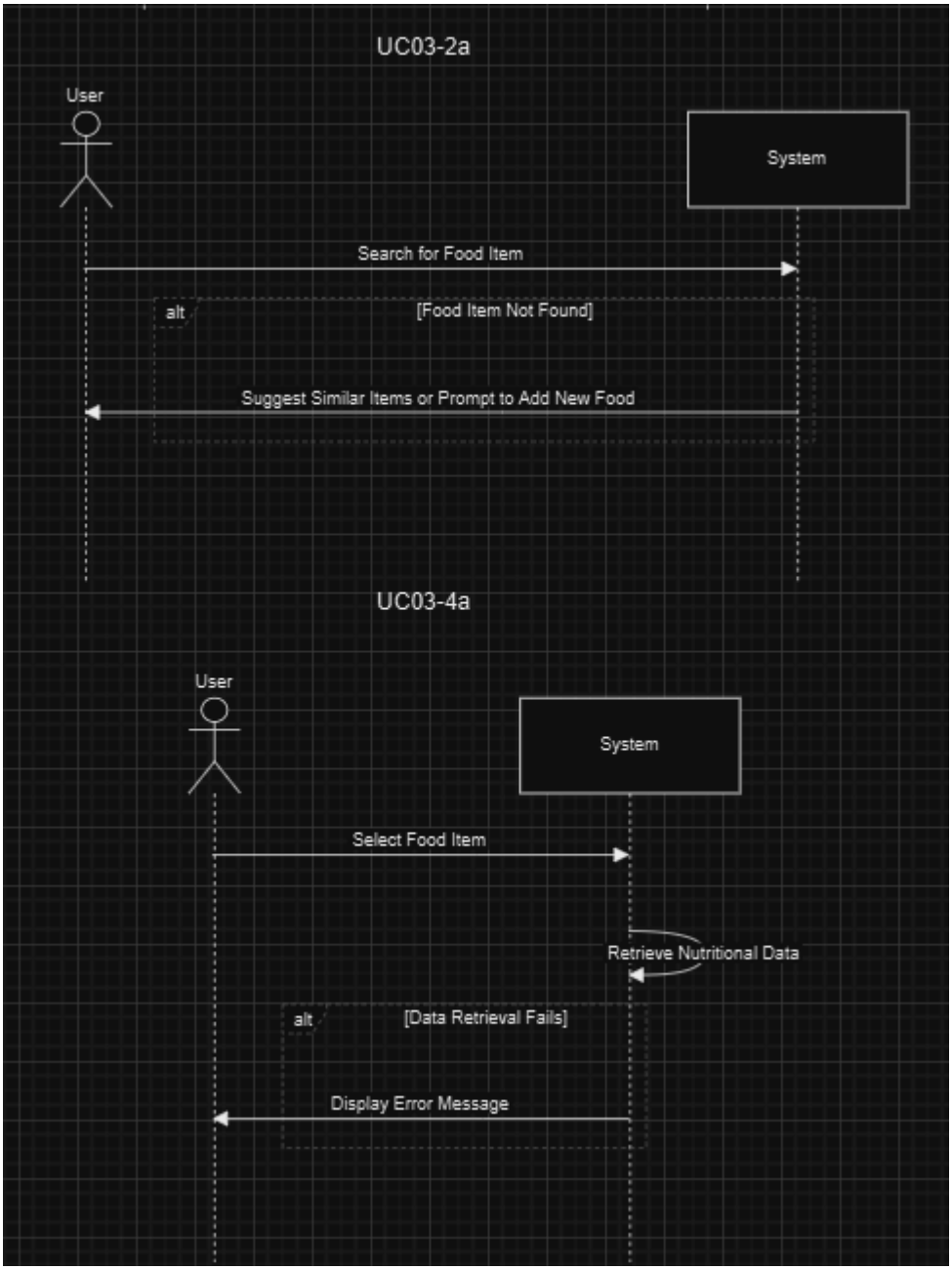
System Sequence Diagram

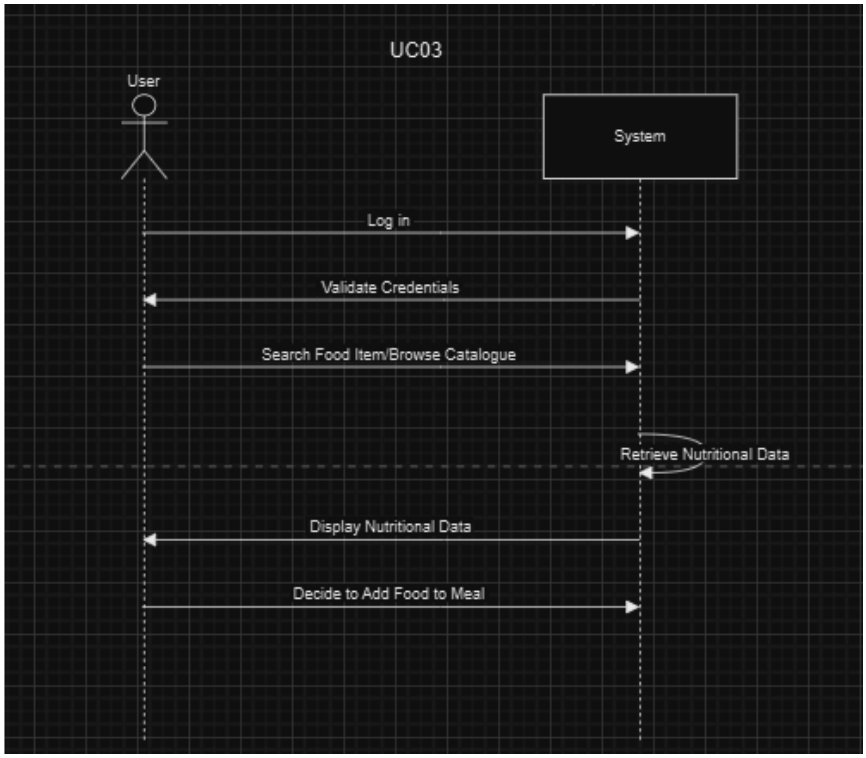
1. **Manage Food Catalogue**


UC01
Admin
System
Log in
Validate Credentials
Navigate to Food Catalogue
Display Food Catalogue & CRUD Options
Add/Edit/Remove Food Item
Validate Input and Update Catalogue
Confirm Update
Display Updated Catalogue

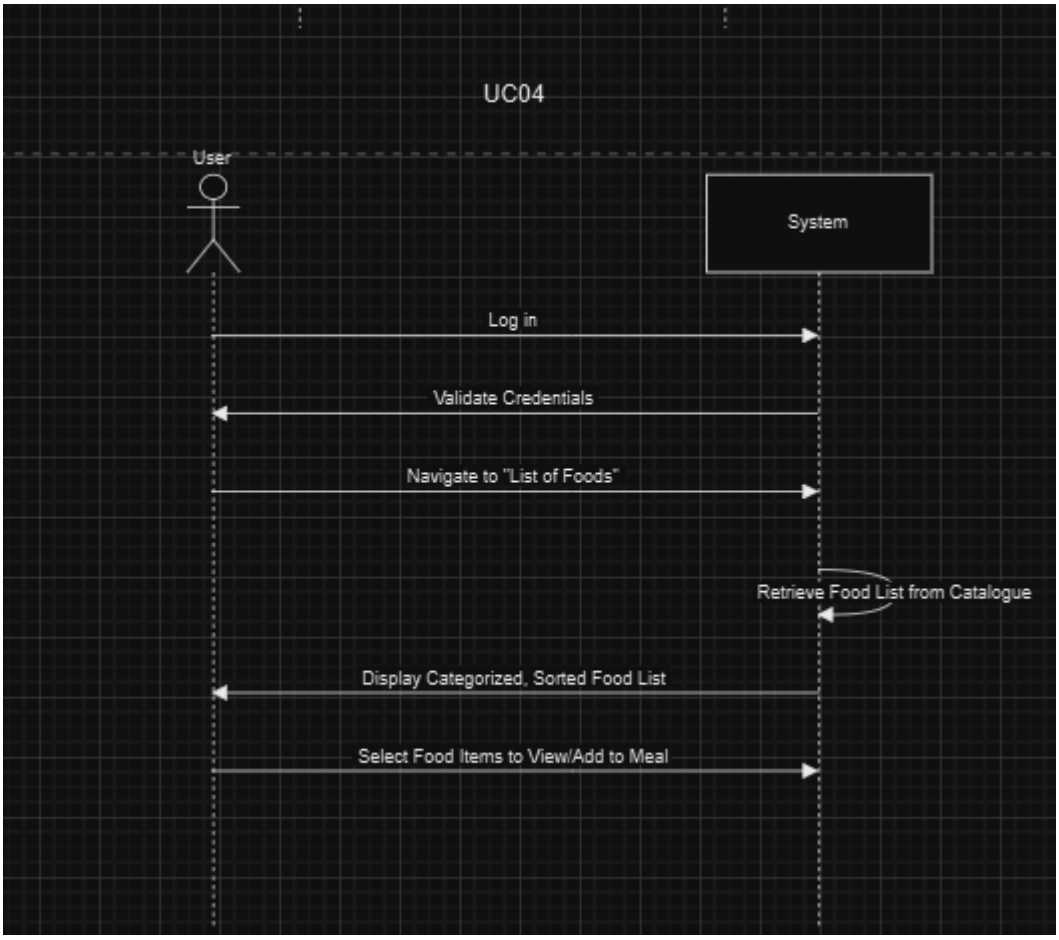

UC01-5a
Admin
System
Remove Food Item
Check if Food Item is in Use
alt
[Food Item in Use]
Display Warning and Prevent Removal
UC01-6a
Admin
System
Add/Edit Food Item with Invalid Data
Validate Input
alt
[Invalid/Incomplete Data]
Prompt to Correct Information

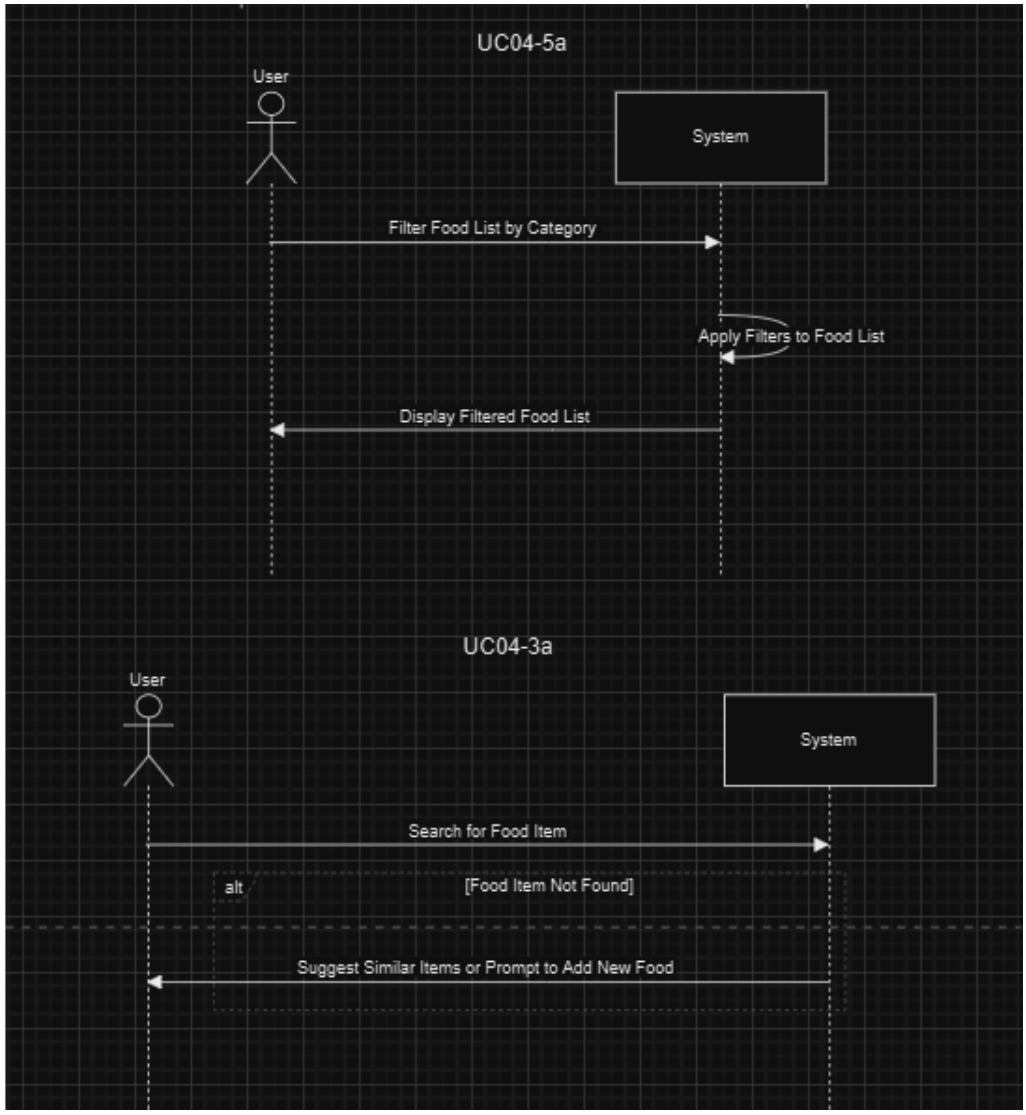
2. **Display Nutritional Information**



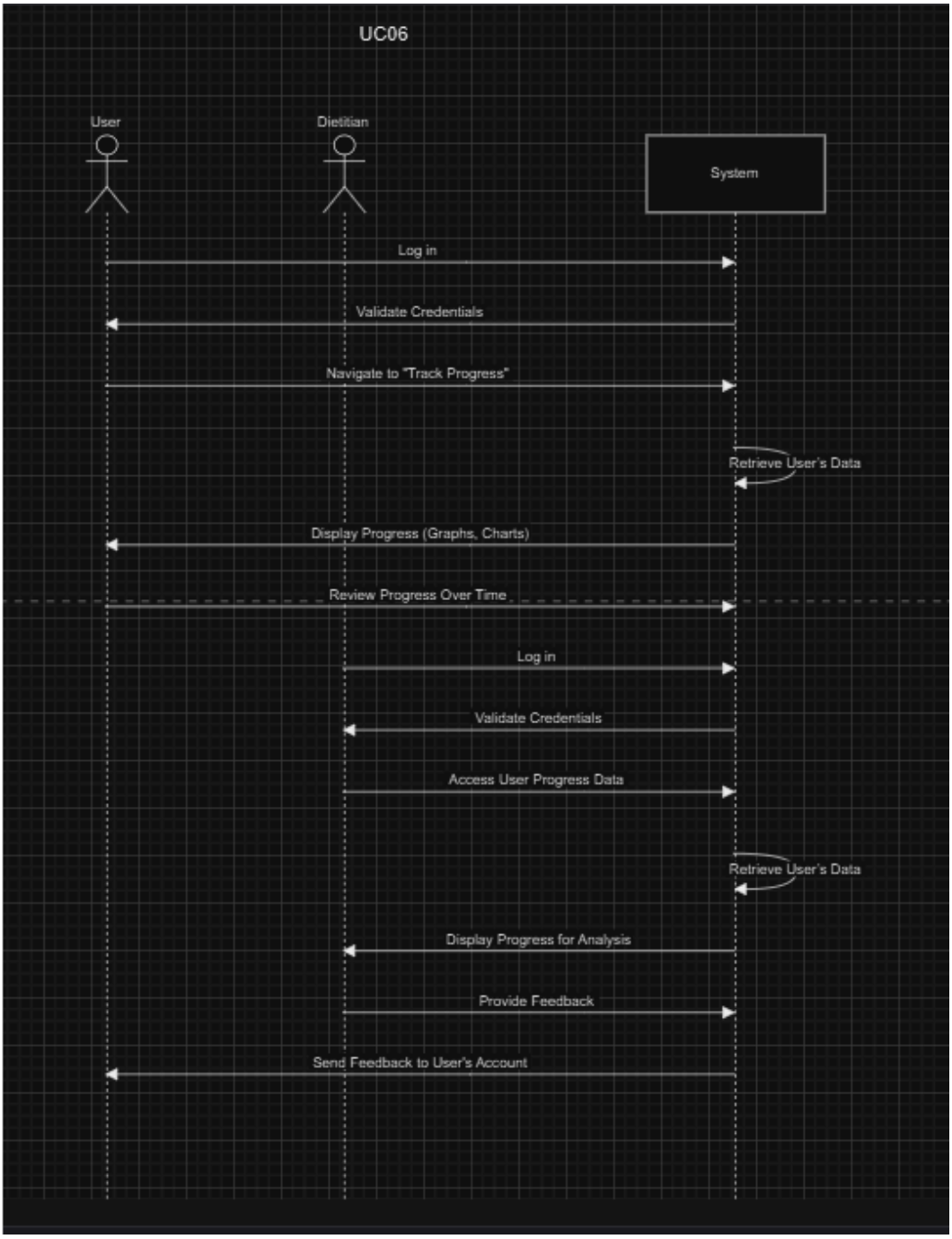

UC03
User
System
Log in
Validate Credentials
Search Food Item/Browse Catalogue
Retrieve Nutritional Data
Display Nutritional Data
Decide to Add Food to Meal

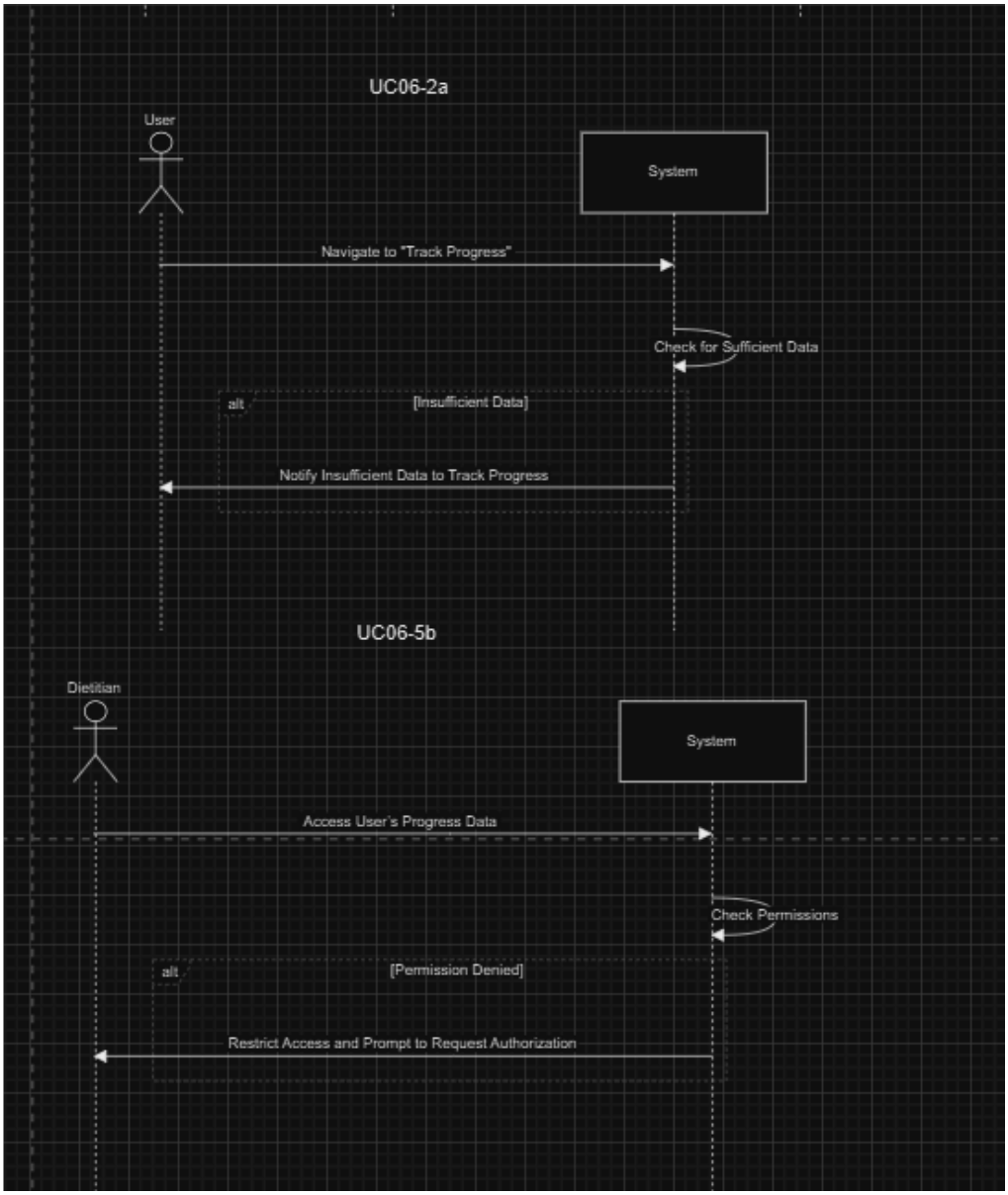
3. **Provide Food Lists**



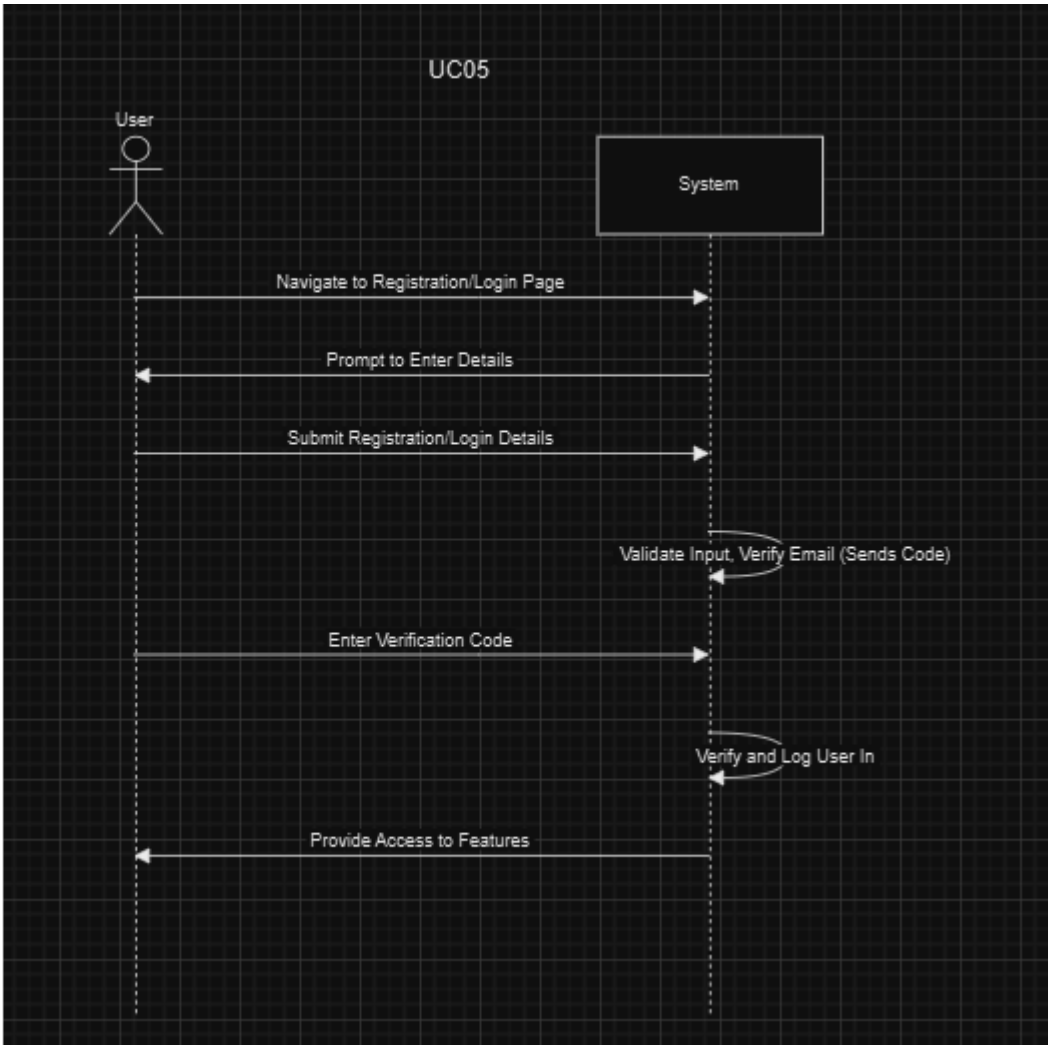


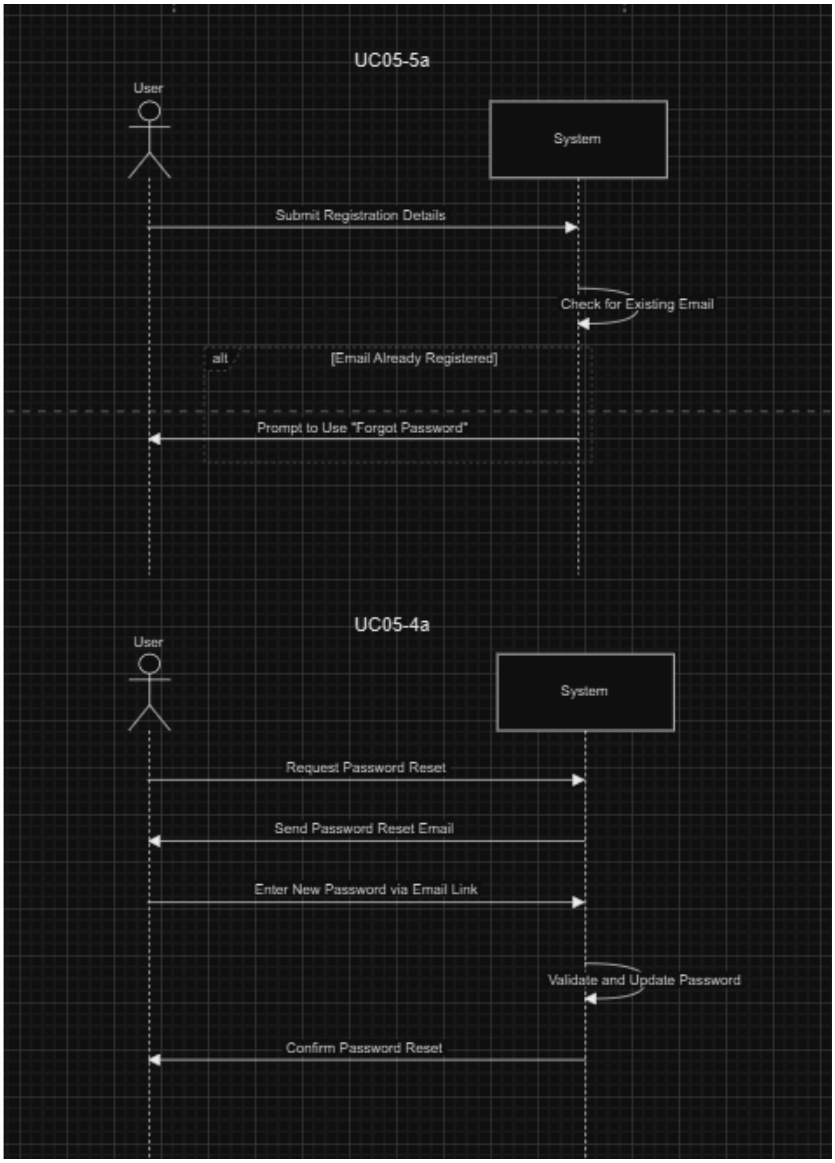
4. **Track Progress**


UC06
User
Dietitian
System
Log in
Validate Credentials
Navigate to "Track Progress"
Retrieve User's Data
Display Progress (Graphs, Charts)
Review Progress Over Time
Log in
Validate Credentials
Access User Progress Data
Retrieve User's Data
Display Progress for Analysis
Provide Feedback
Send Feedback to User's Account

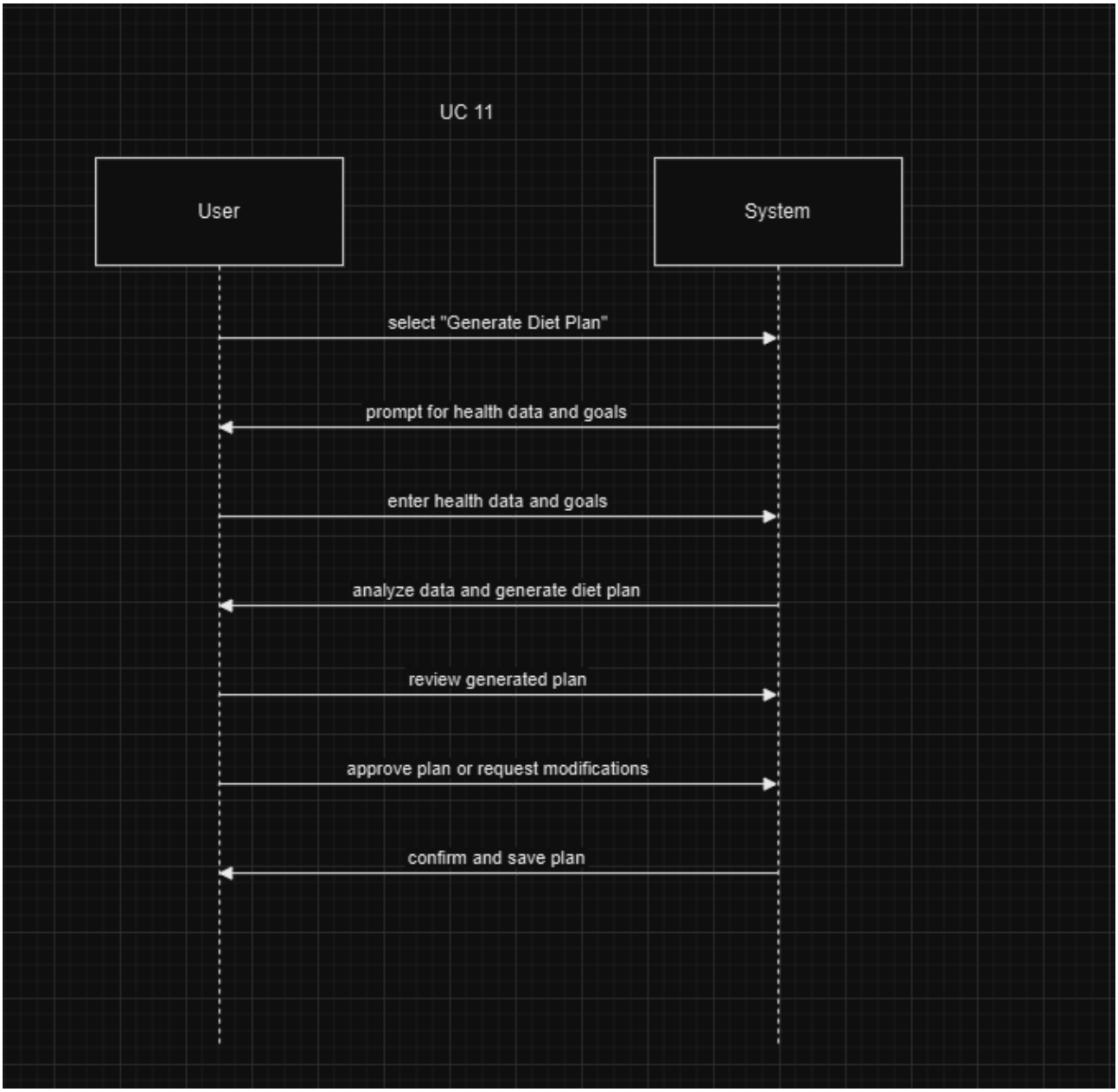


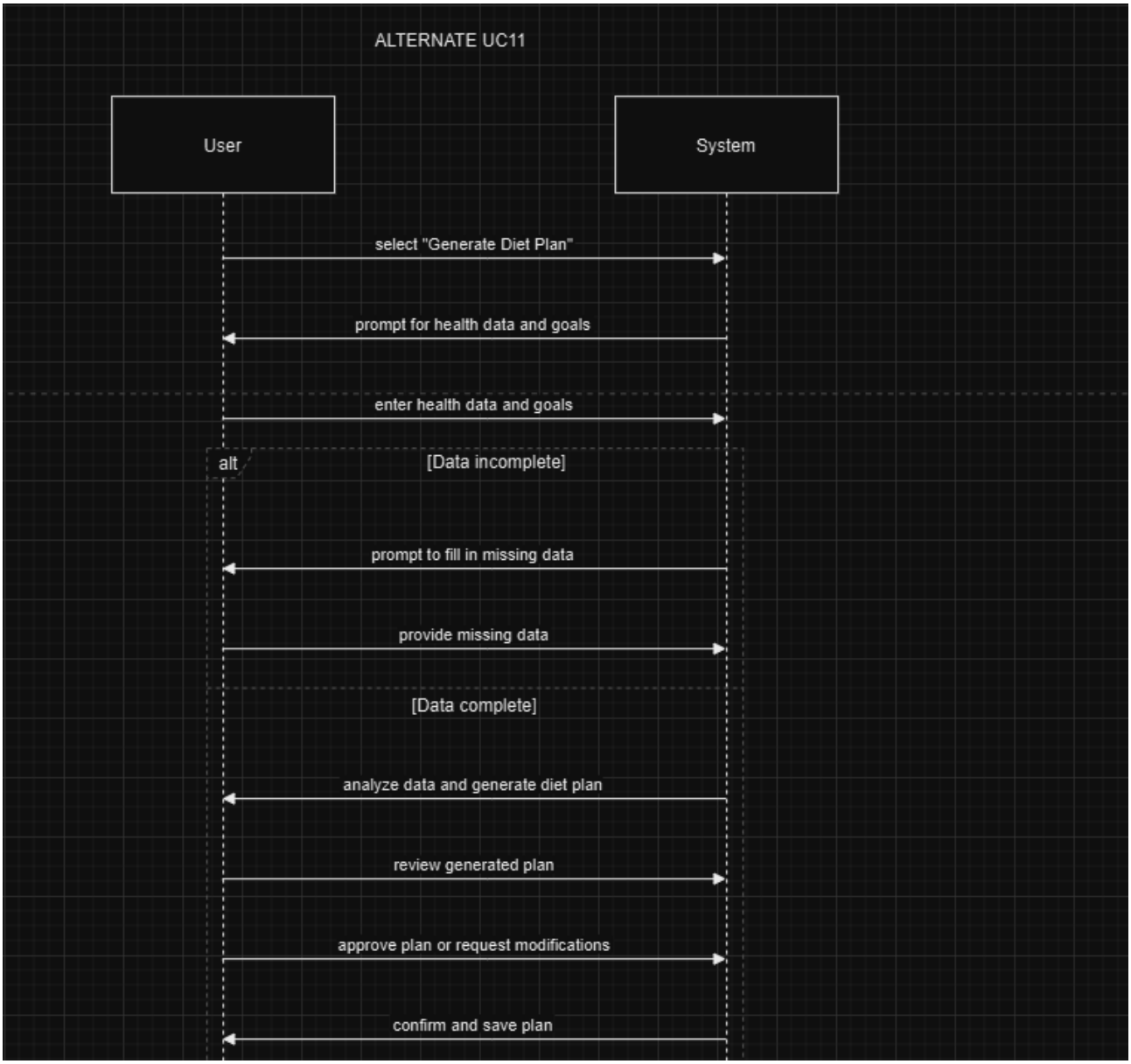
5. **Log In/Register**


UC05
User
System
Navigate to Registration/Login Page
Prompt to Enter Details
Submit Registration/Login Details
Validate Input, Verify Email (Sends Code)
Enter Verification Code
Verify and Log User In
Provide Access to Features

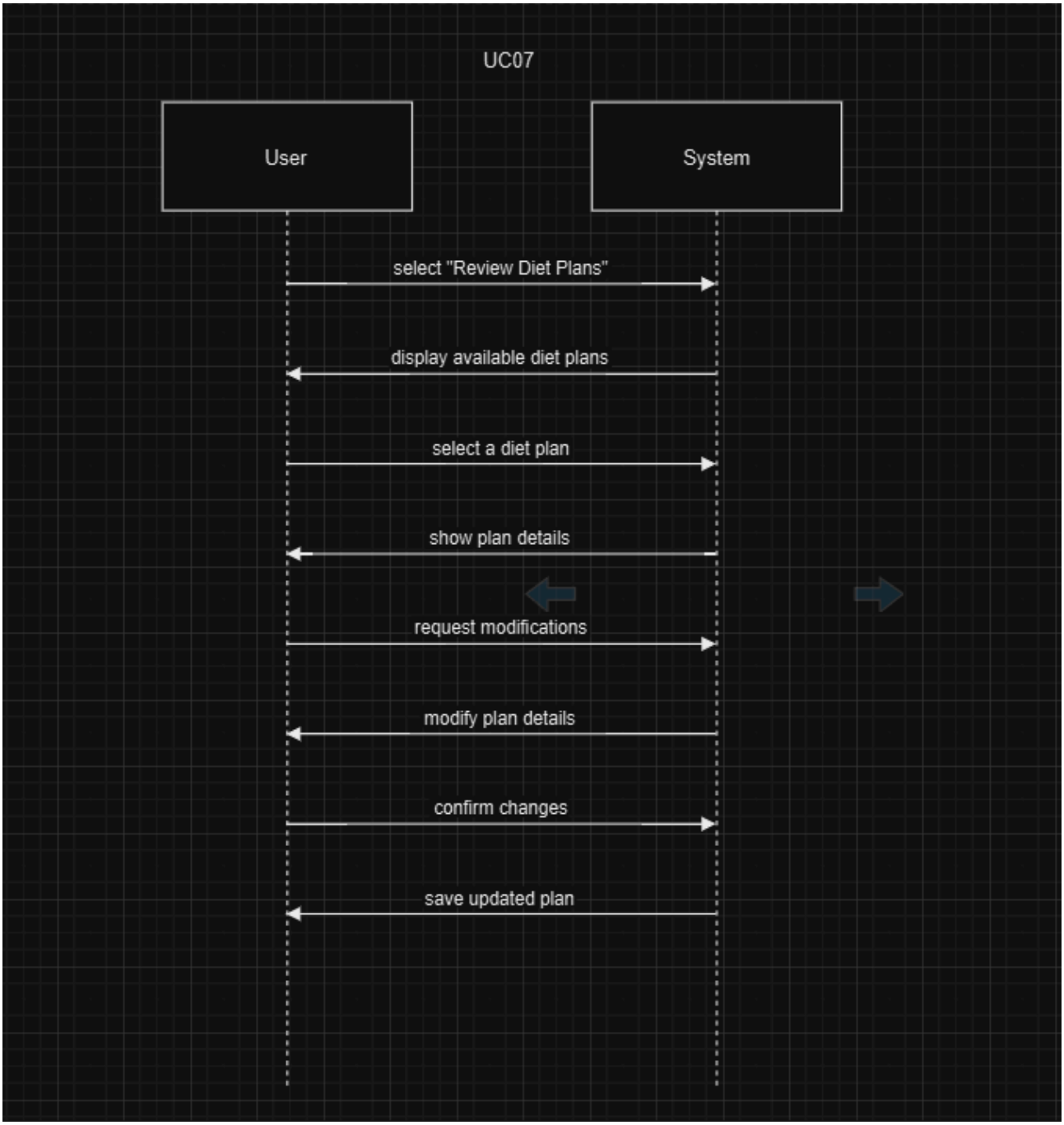

UC05-5a
User
System
Submit Registration Details
Check for Existing Email
alt
[Email Already Registered]
Prompt to Use "Forgot Password"
UC05-4a
User
System
Request Password Reset
Send Password Reset Email
Enter New Password via Email Link
Validate and Update Password
Confirm Password Reset

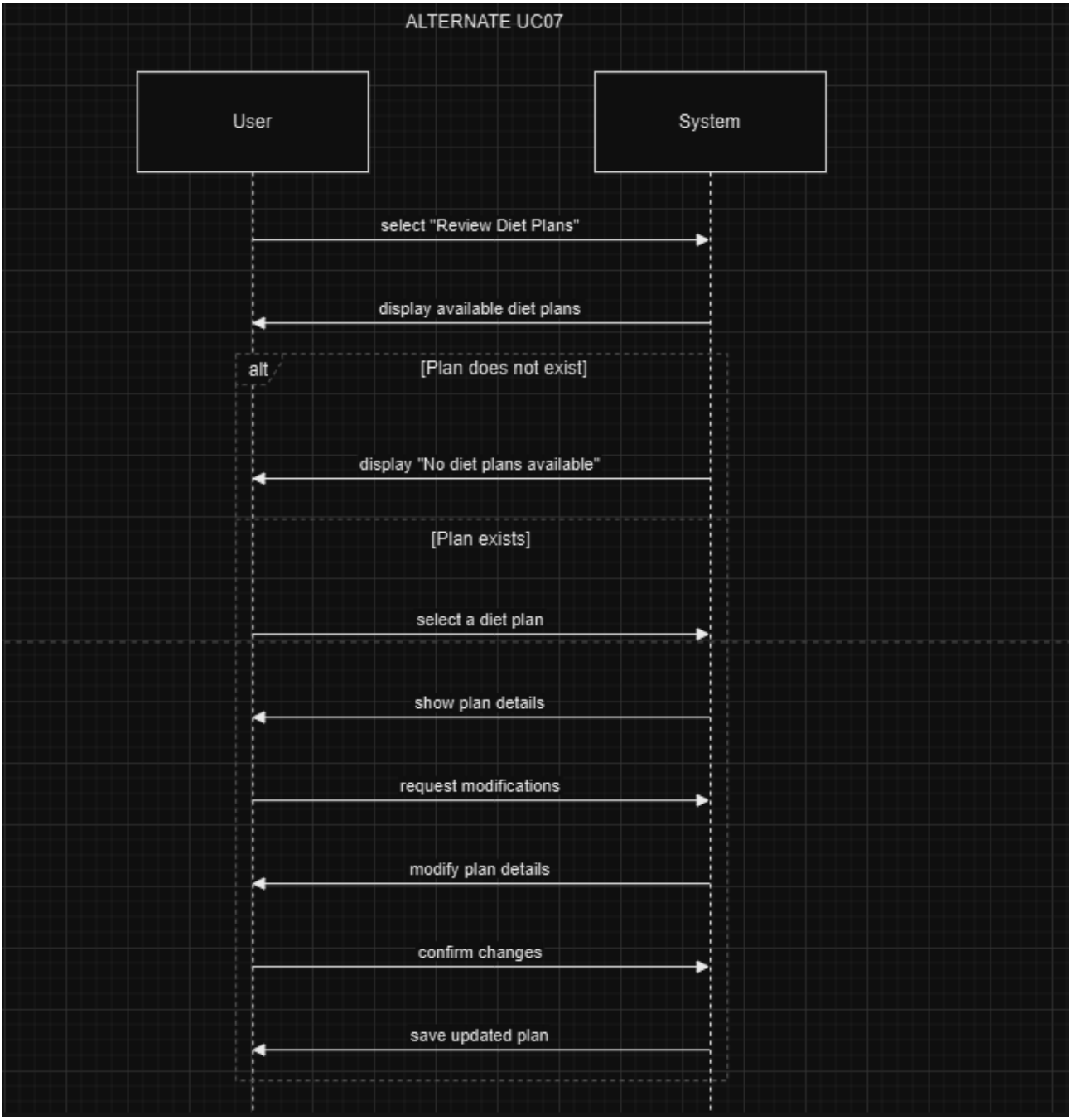
6. **Generate Diet Plan**



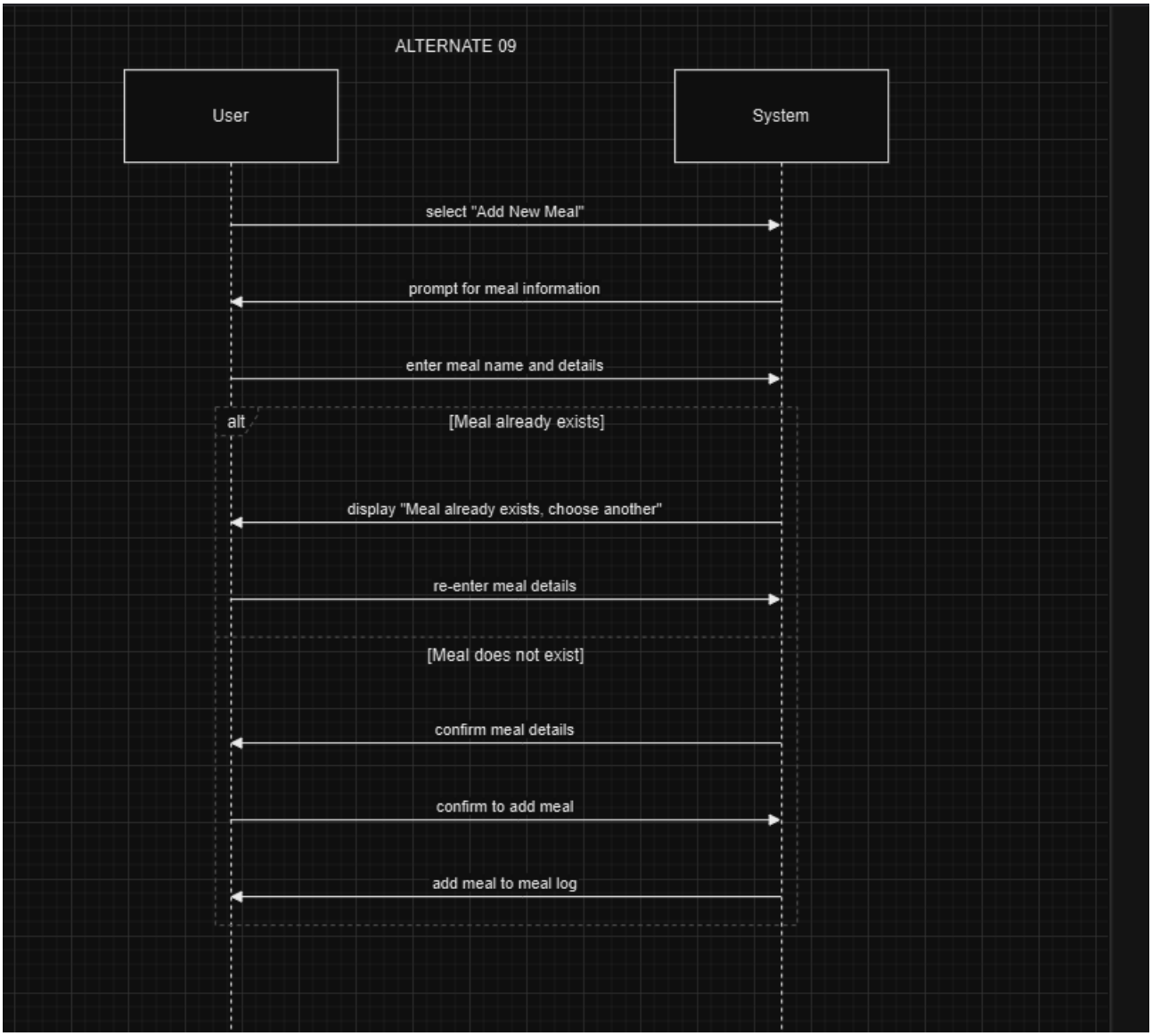

ALTERNATE UC11
User
System
select "Generate Diet Plan"
prompt for health data and goals
enter health data and goals
alt
[Data incomplete]
prompt to fill in missing data
provide missing data
[Data complete]
analyze data and generate diet plan
review generated plan
approve plan or request modifications
confirm and save plan

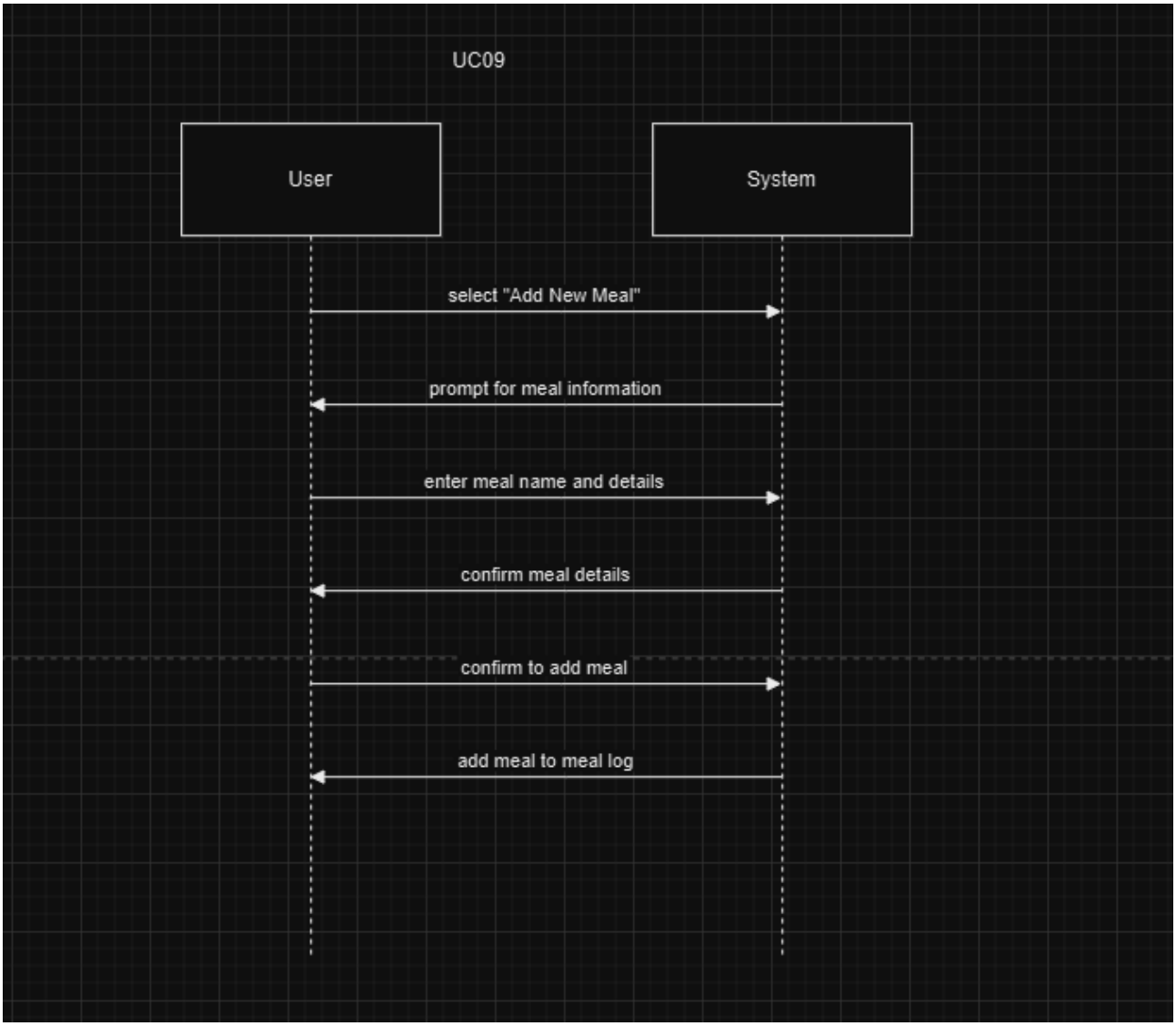
7. **Review Diet Plans**



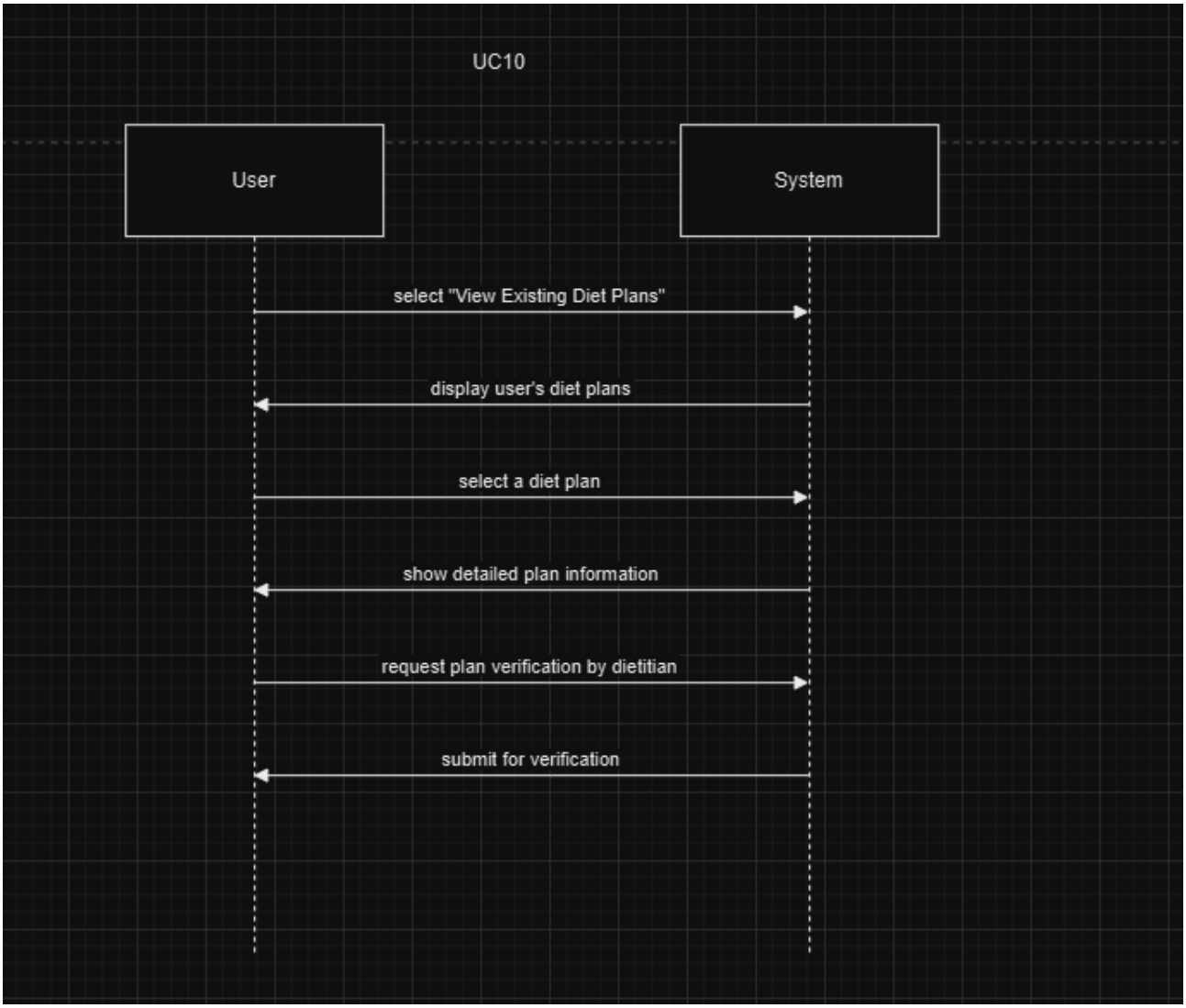

ALTERNATE UC07
User
System
select "Review Diet Plans"
display available diet plans
alt
[Plan does not exist]
display "No diet plans available"
[Plan exists]
select a diet plan
show plan details
request modifications
modify plan details
confirm changes
save updated plan

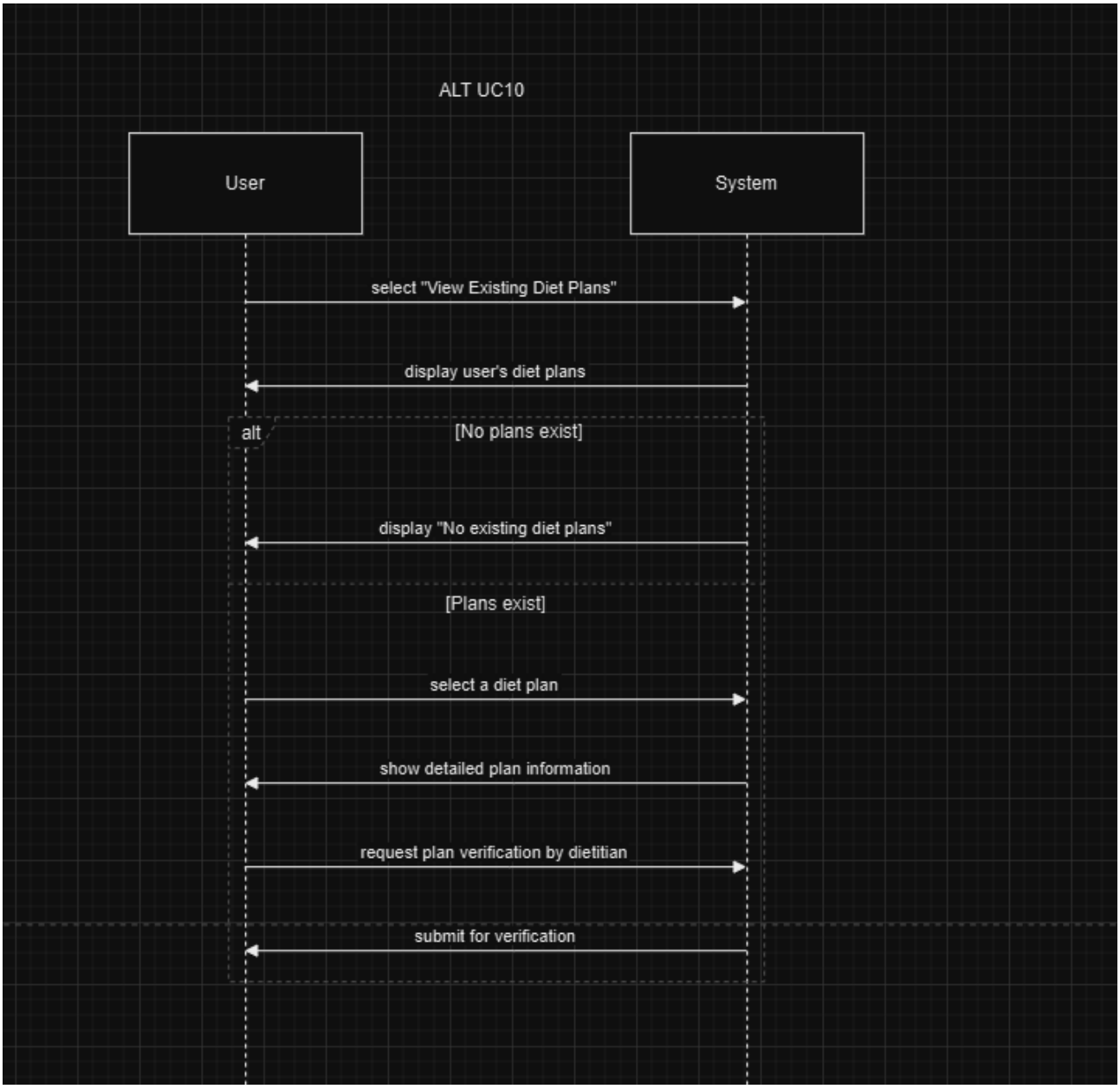
8. **Add New Meal**



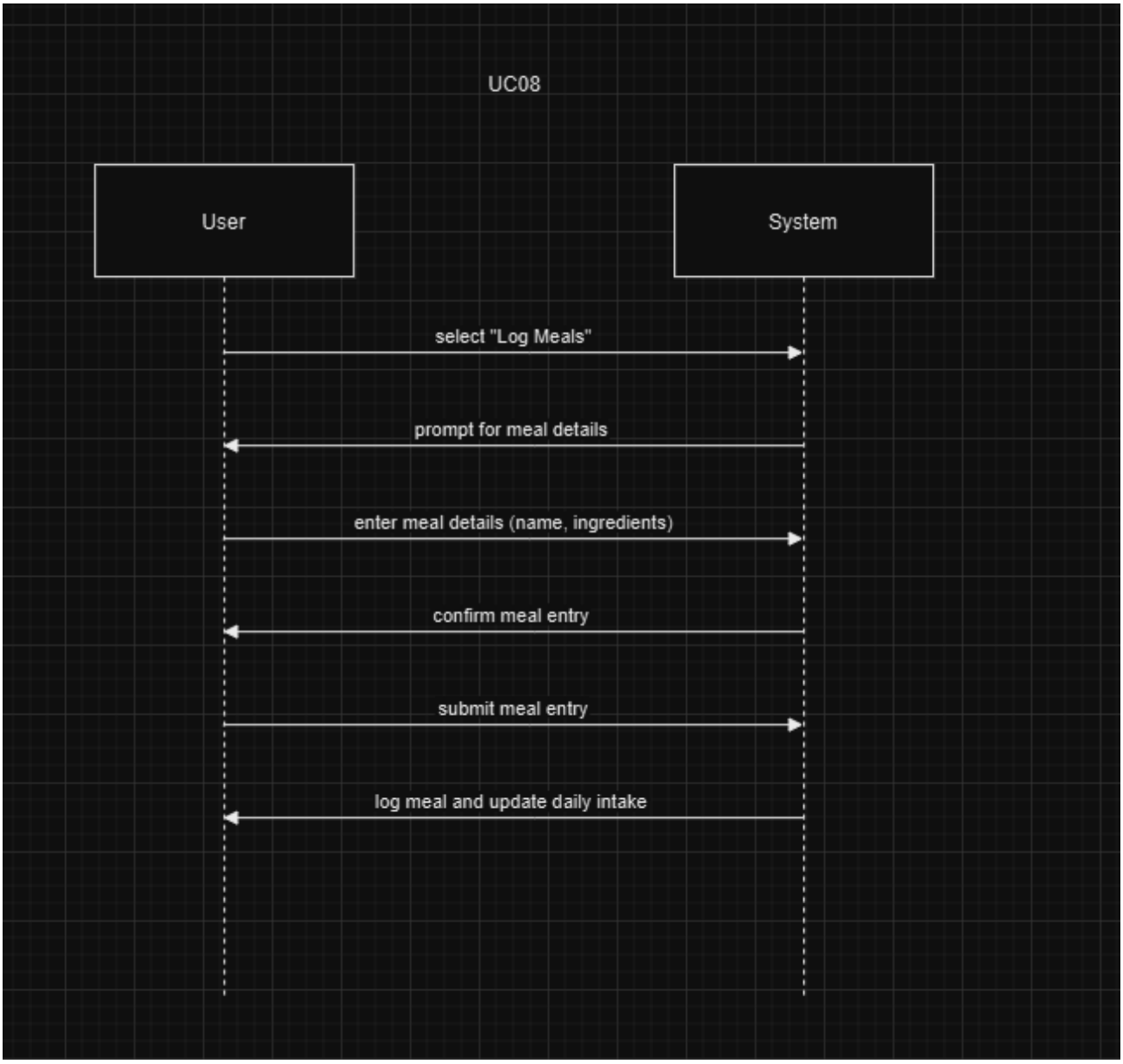

UC09
User
System
select "Add New Meal"
prompt for meal information
enter meal name and details
confirm meal details
confirm to add meal
add meal to meal log

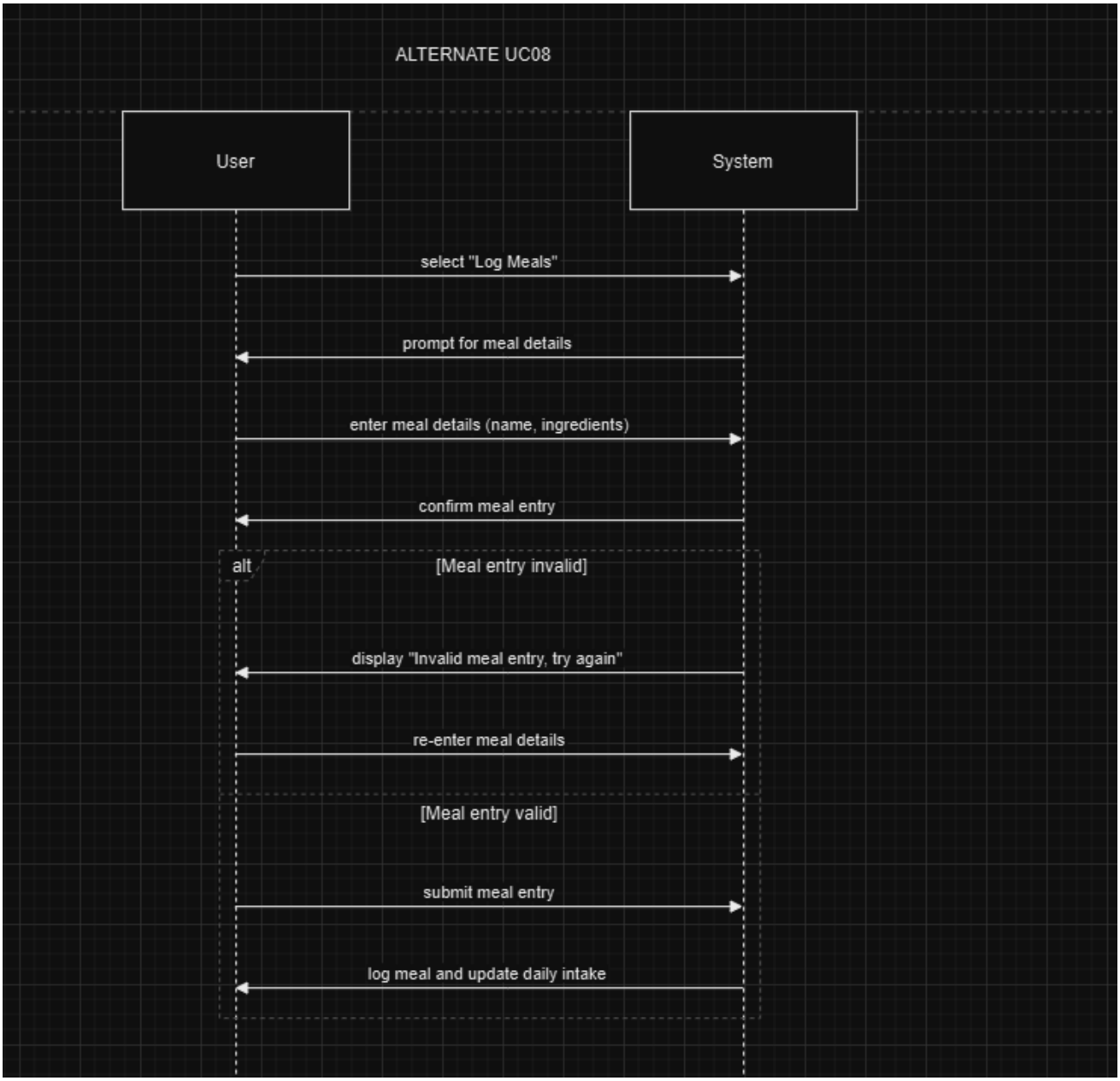
9. **View Existing Diet Plans**



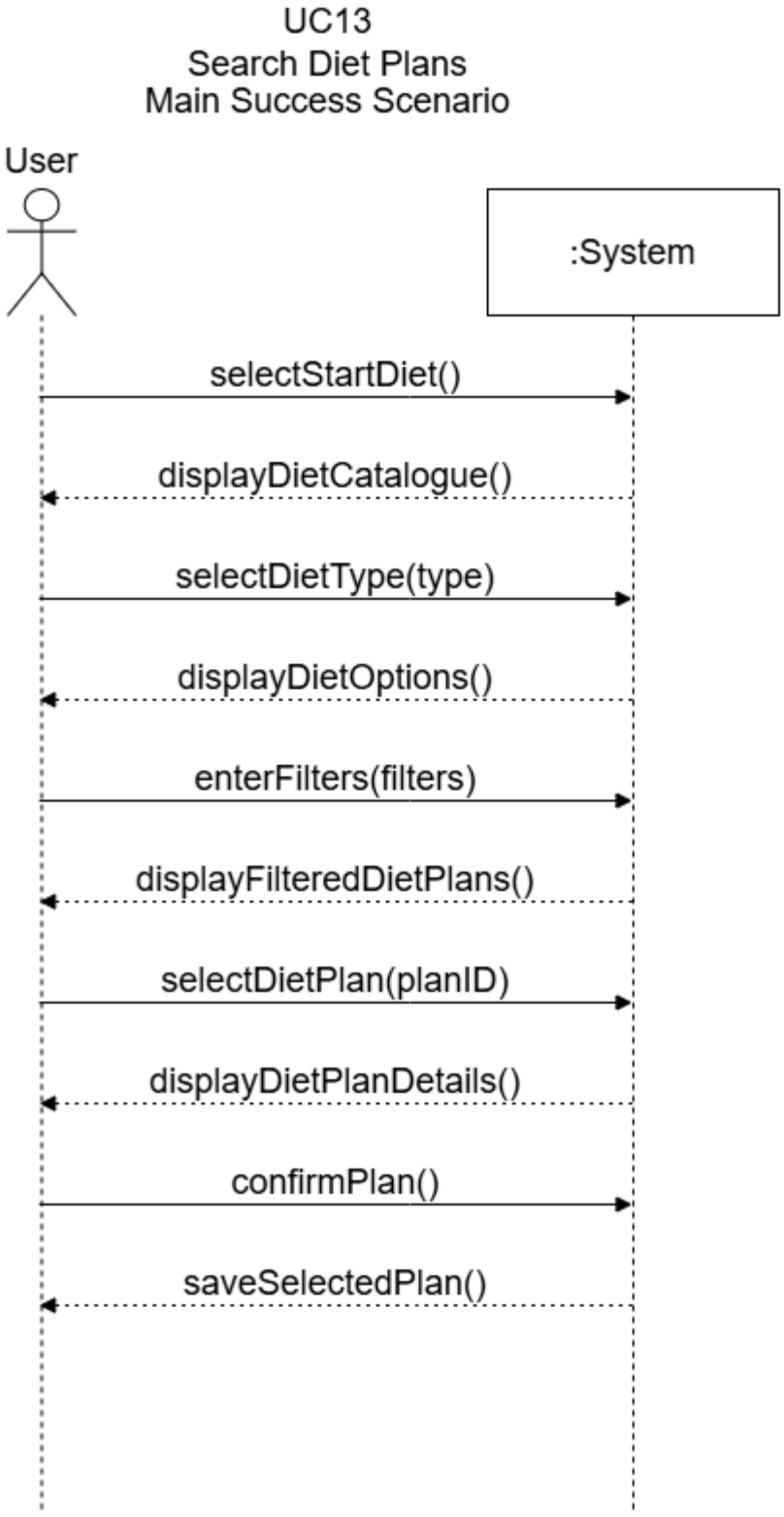

ALT UC10
User
System
select "View Existing Diet Plans"
display user's diet plans
alt
[No plans exist]
display "No existing diet plans"
[Plans exist]
select a diet plan
show detailed plan information
request plan verification by dietitian
submit for verification

10. **Log Meals**

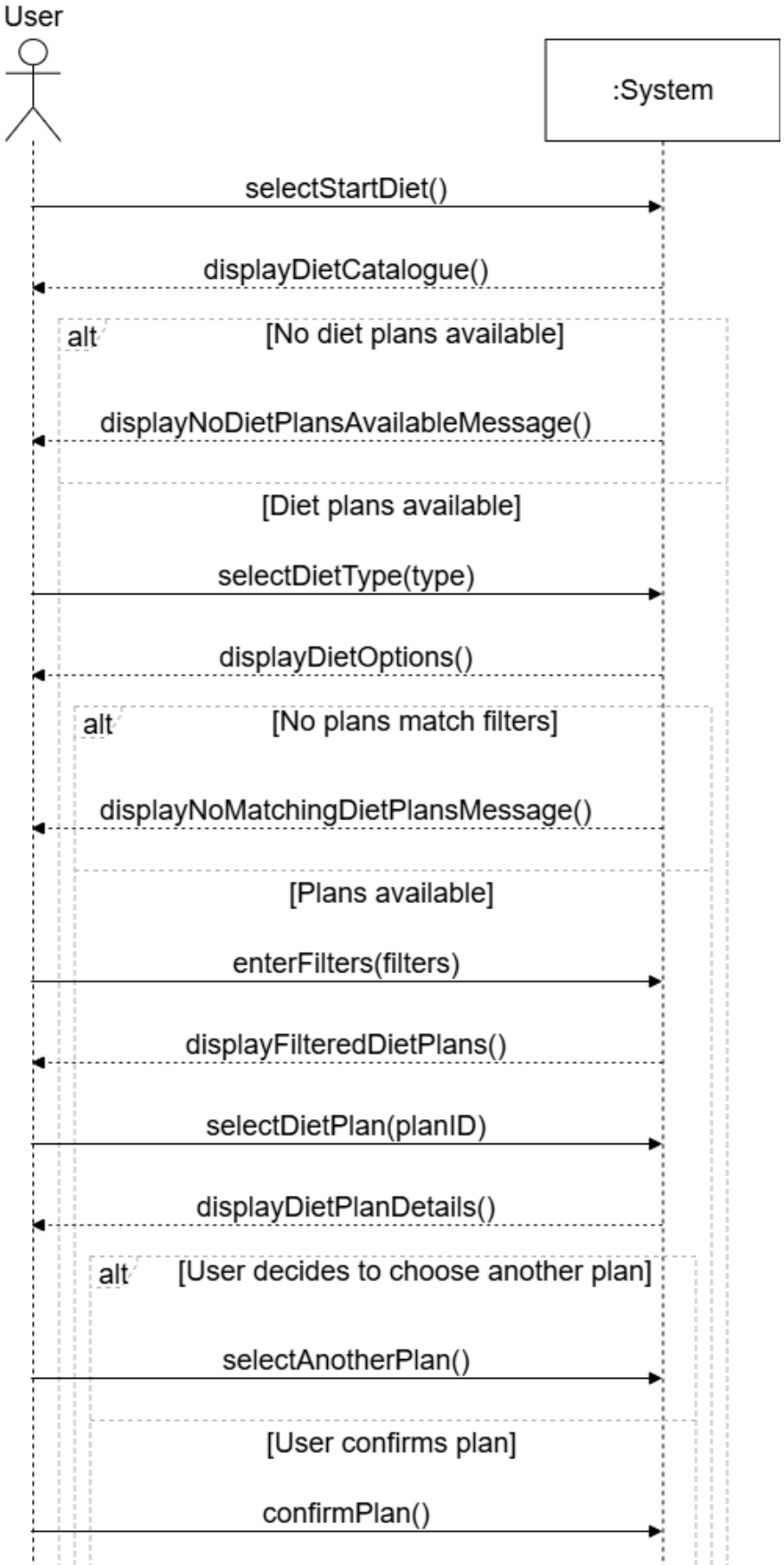

UC08
User
System
select "Log Meals"
prompt for meal details
enter meal details (name, ingredients)
confirm meal entry
submit meal entry
log meal and update daily intake



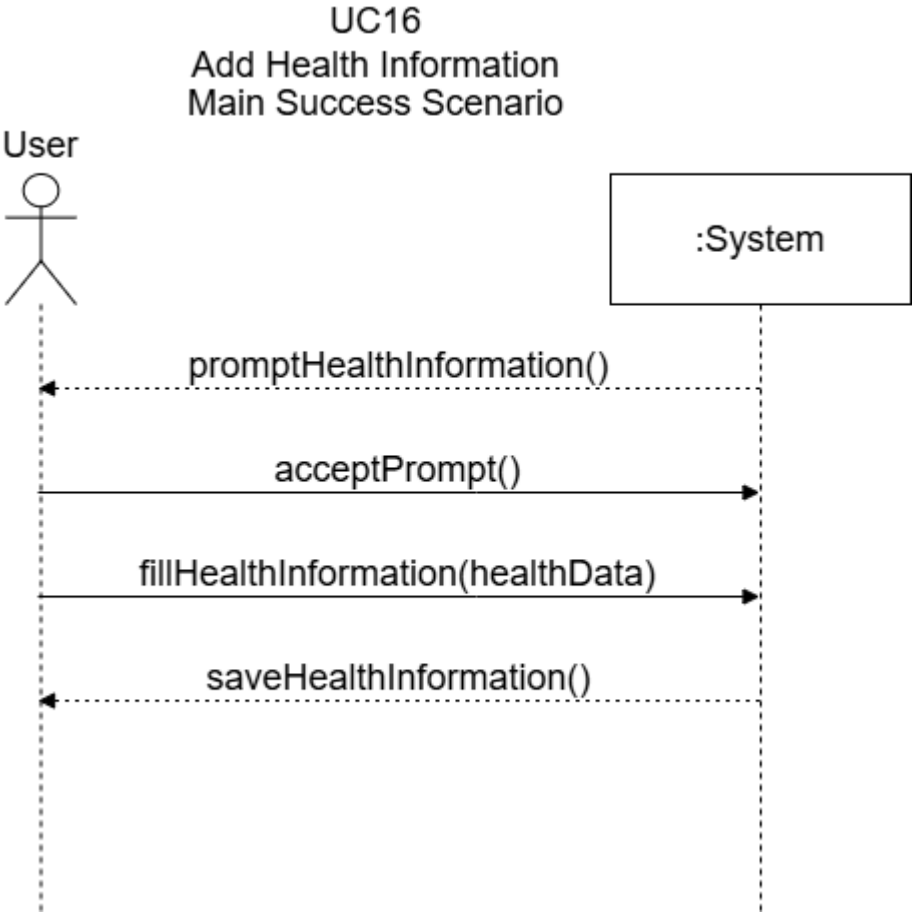
11. **Search Diet Plans**

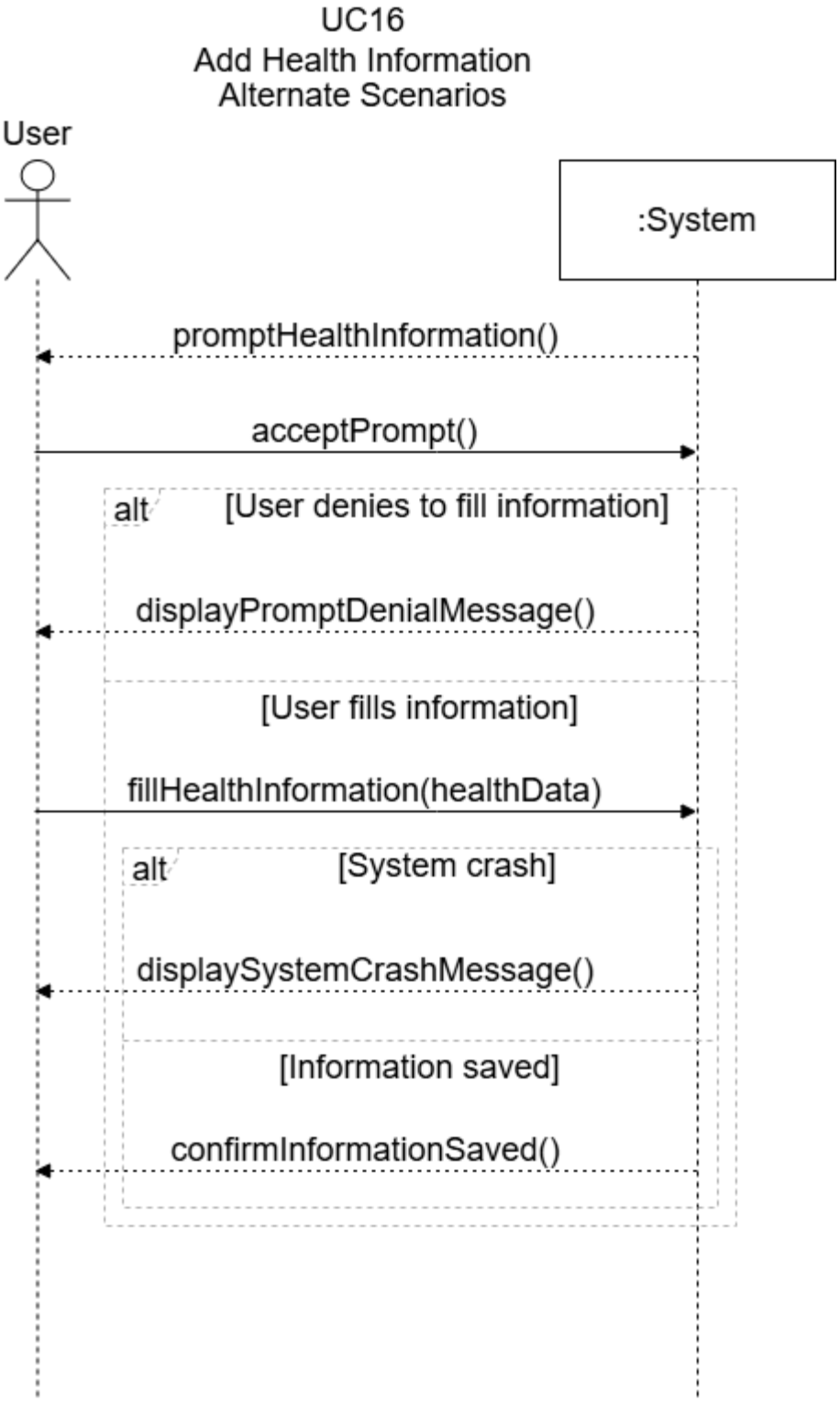

UC13
Search Diet Plans
Main Success Scenario
User
:System
selectStartDiet()
displayDietCatalogue()
selectDietType(type)
displayDietOptions()
enterFilters(filters)
displayFilteredDietPlans()
selectDietPlan(planID)
displayDietPlanDetails()
confirmPlan()
saveSelectedPlan()

UC13
Search Diet Plans
Alternate Scenarios

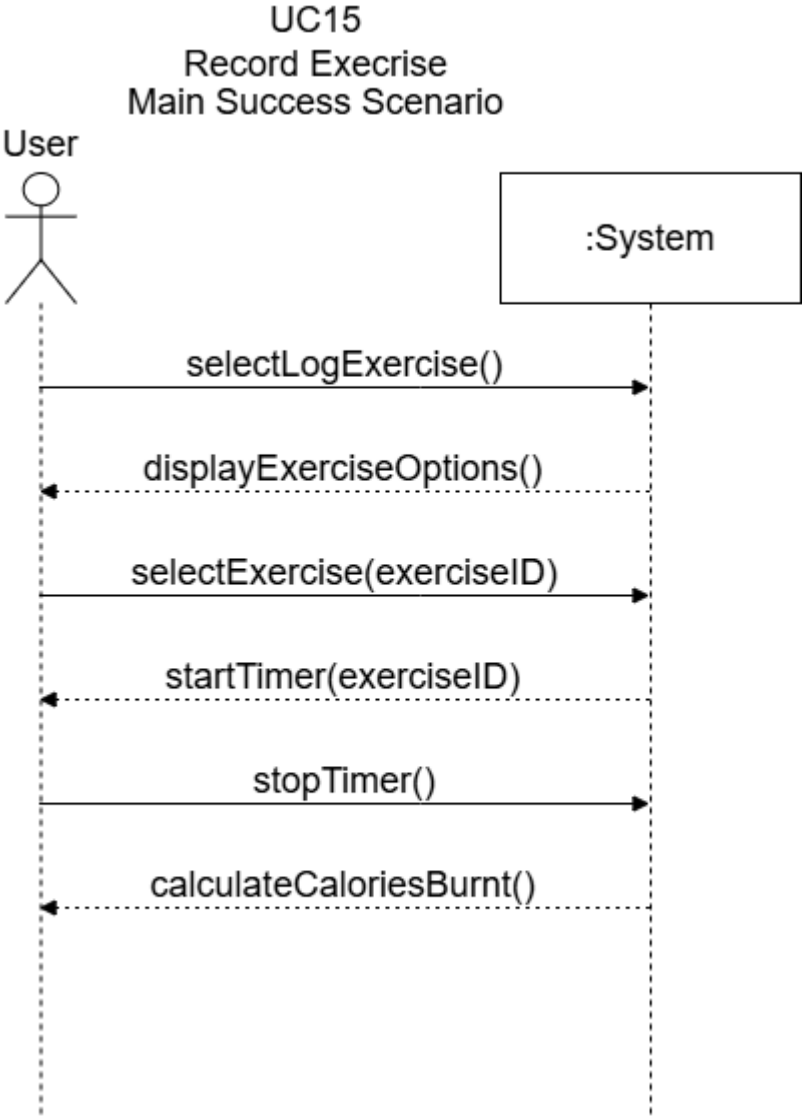


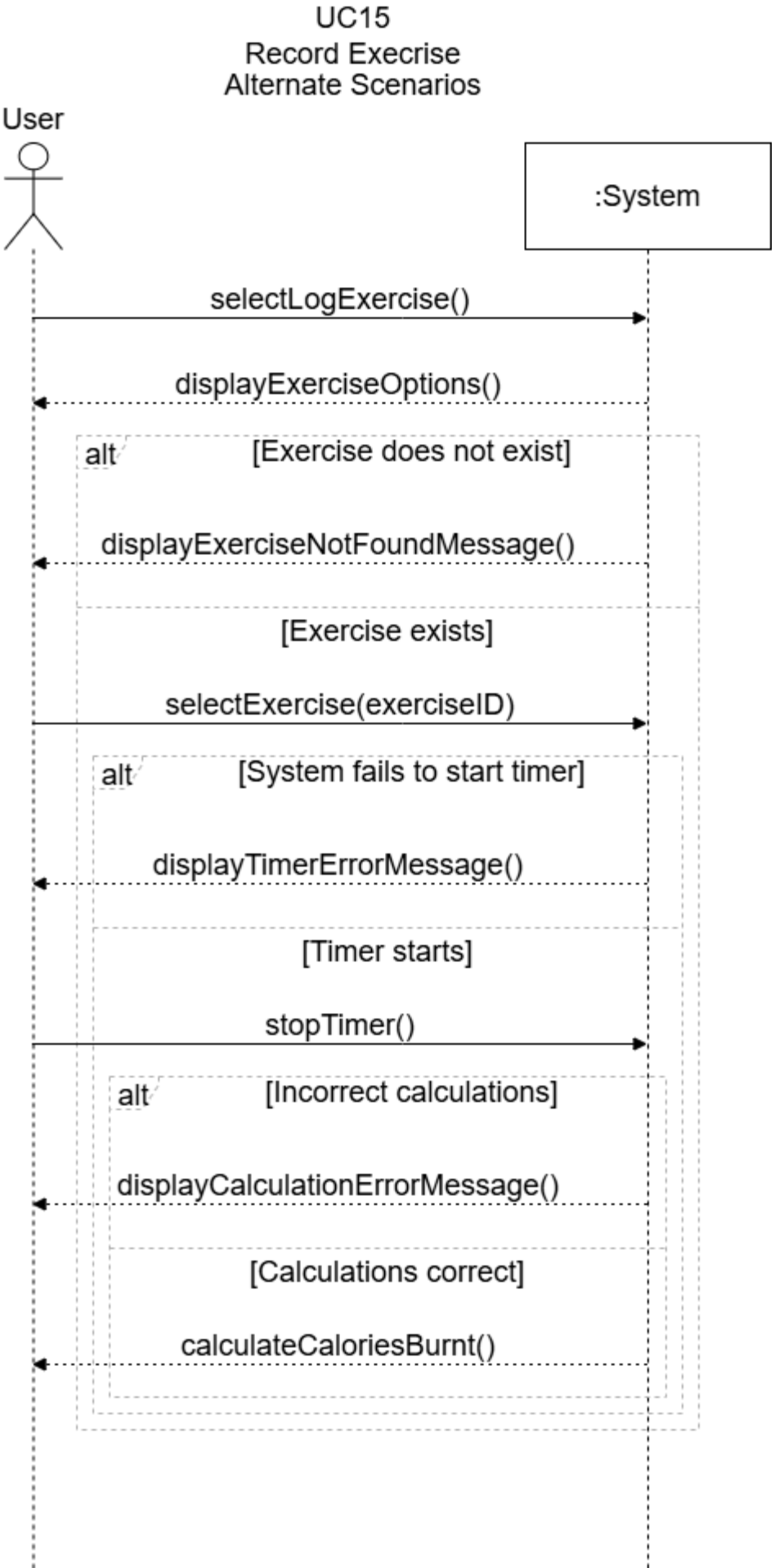
12. **Add Health Information**



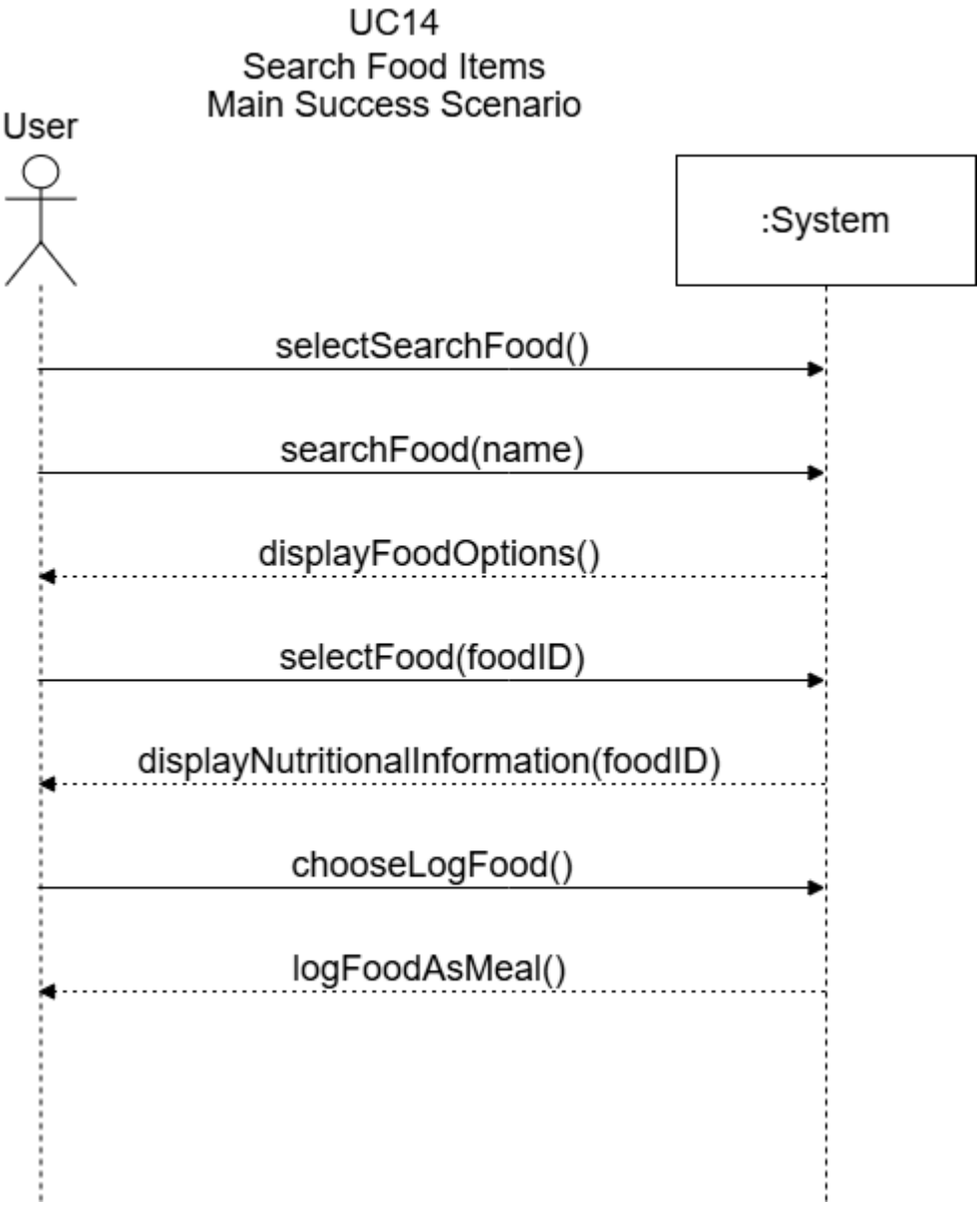

UC16
Add Health Information
Alternate Scenarios
User
:System
promptHealthInformation()
acceptPrompt()
alt
[User denies to fill information]
displayPromptDenialMessage()
[User fills information]
fillHealthInformation(healthData)
alt
[System crash]
displaySystemCrashMessage()
[Information saved]
confirmInformationSaved()

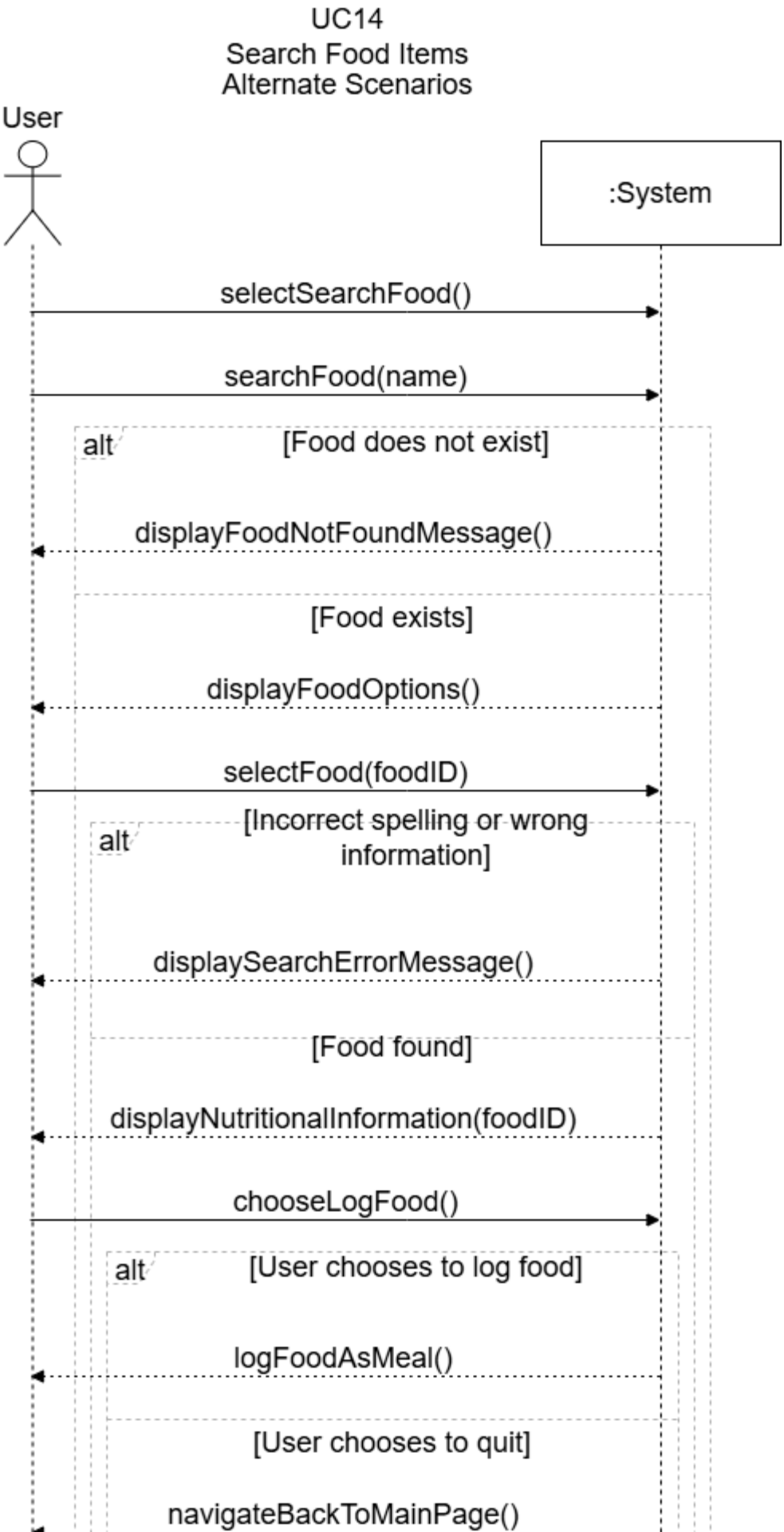
13. **Record Exercise**




UC15
Record Execrise
Alternate Scenarios
User
:System
selectLogExercise()
displayExerciseOptions()
alt
[Exercise does not exist]
displayExerciseNotFoundMessage()
[Exercise exists]
selectExercise(exerciseID)
alt
[System fails to start timer]
displayTimerErrorMessage()
[Timer starts]
stopTimer()
alt
[Incorrect calculations]
displayCalculationErrorMessage()
[Calculations correct]
calculateCaloriesBurnt()

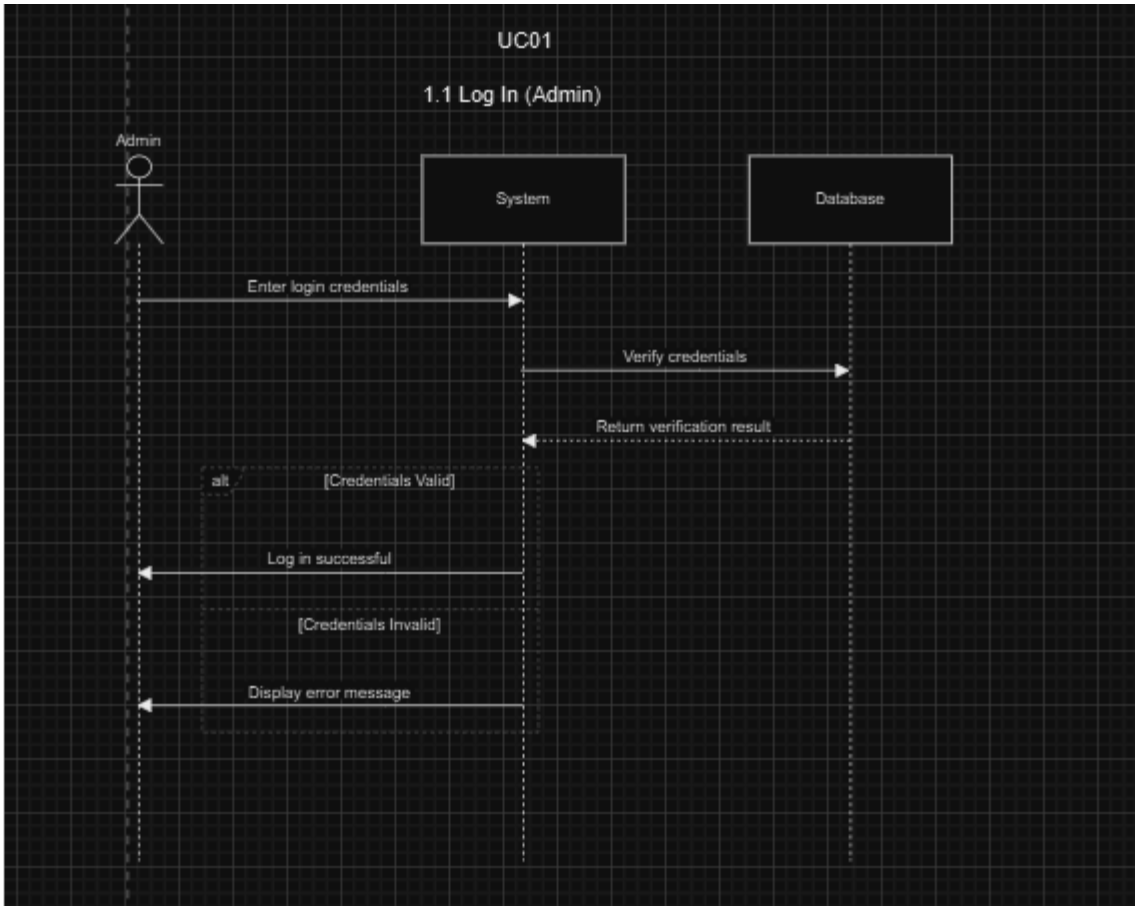
14. **Search Food Items**

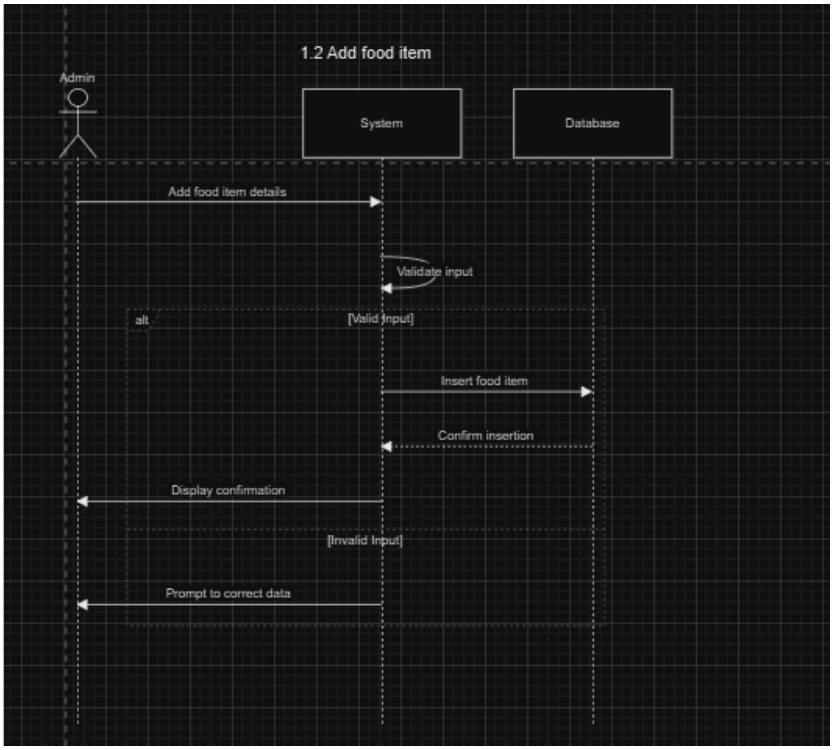


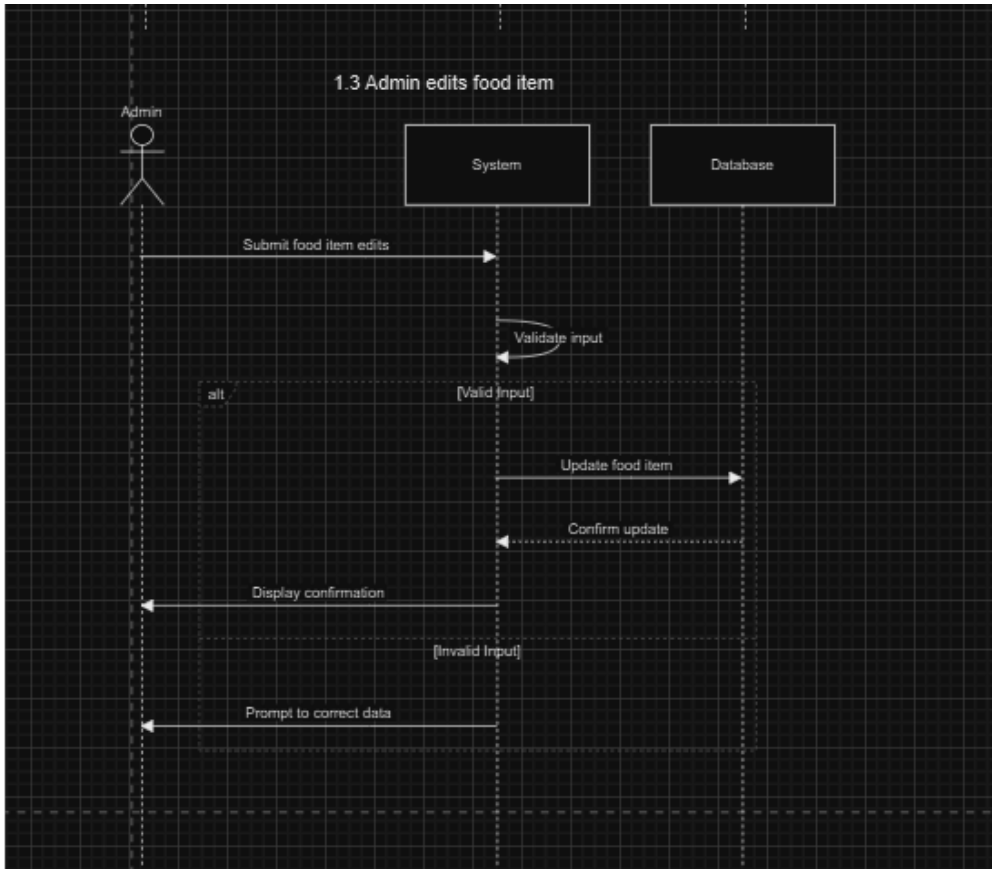

UC14
Search Food Items
Alternate Scenarios
User
:System
selectSearchFood()
searchFood(name)
alt
[Food does not exist]
displayFoodNotFoundMessage()
[Food exists]
displayFoodOptions()
selectFood(foodID)
alt
[Incorrect spelling or wrong information]
displaySearchErrorMessage()
[Food found]
displayNutritionalInformation(foodID)
chooseLogFood()
alt
[User chooses to log food]
logFoodAsMeal()
[User chooses to quit]
navigateBackToMainPage()

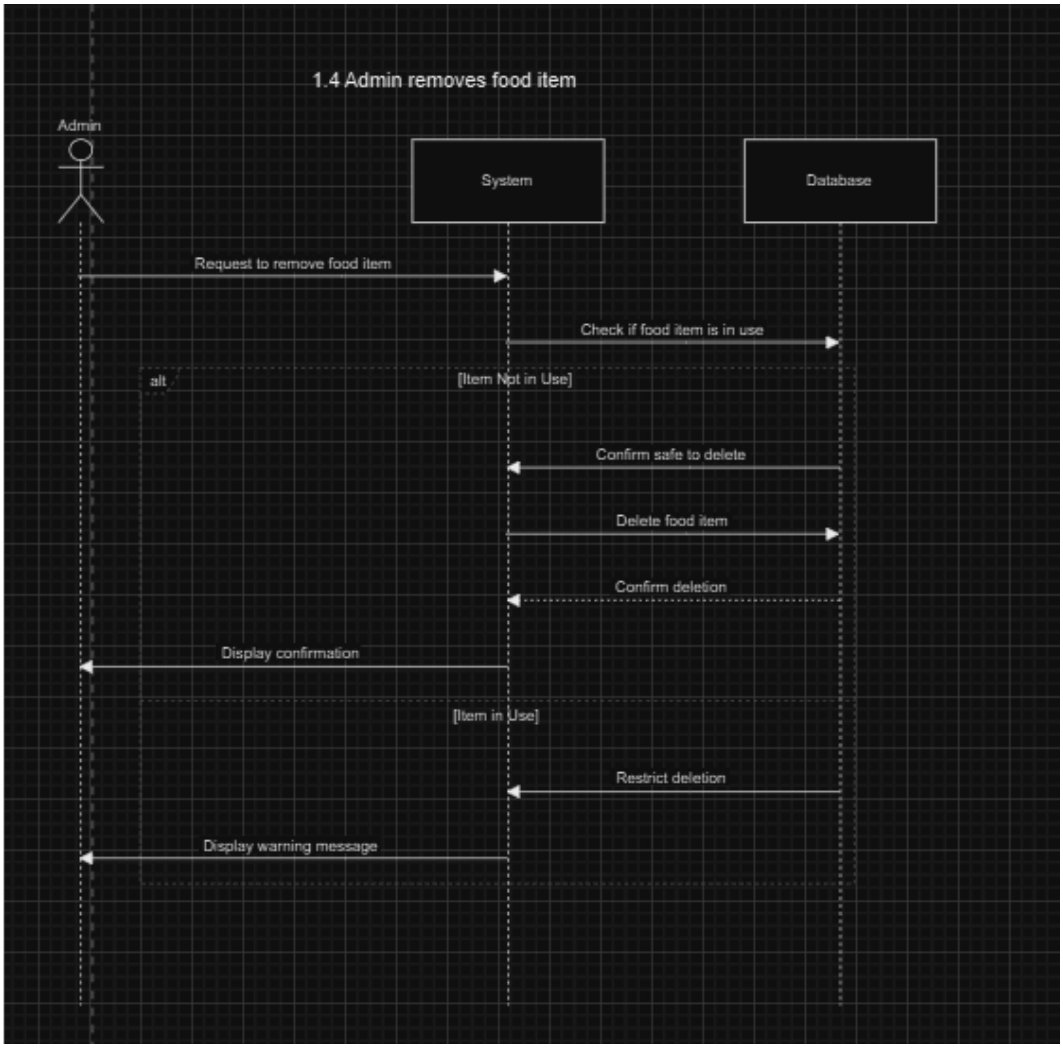
Sequence Diagram

1. Manage Food Catalogue

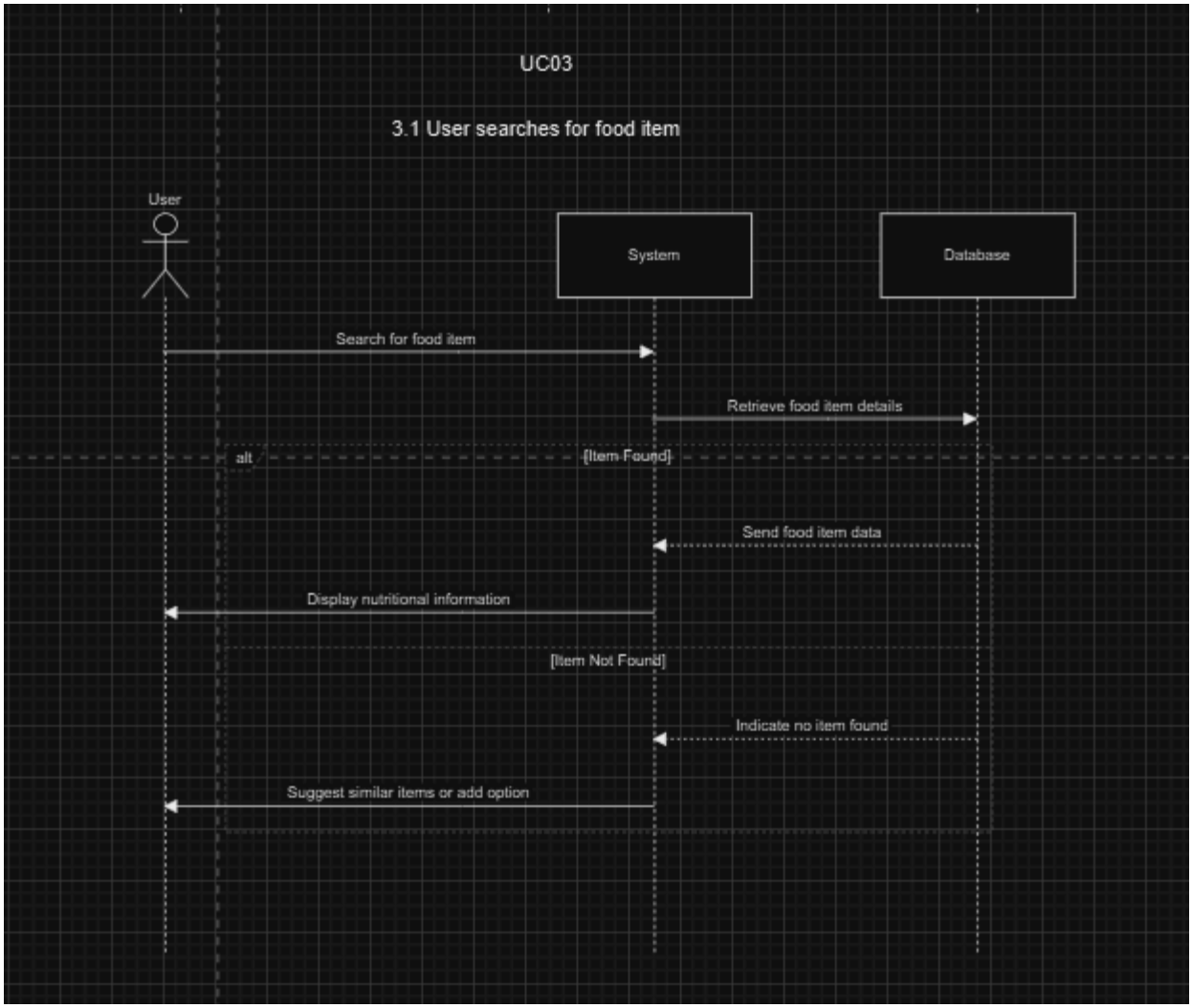



1.2 Add food item
Admin
System
Database
Add food item details
Validate input
alt
[Valid Input]
Insert food item
Confirm insertion
Display confirmation
[Invalid Input]
Prompt to correct data

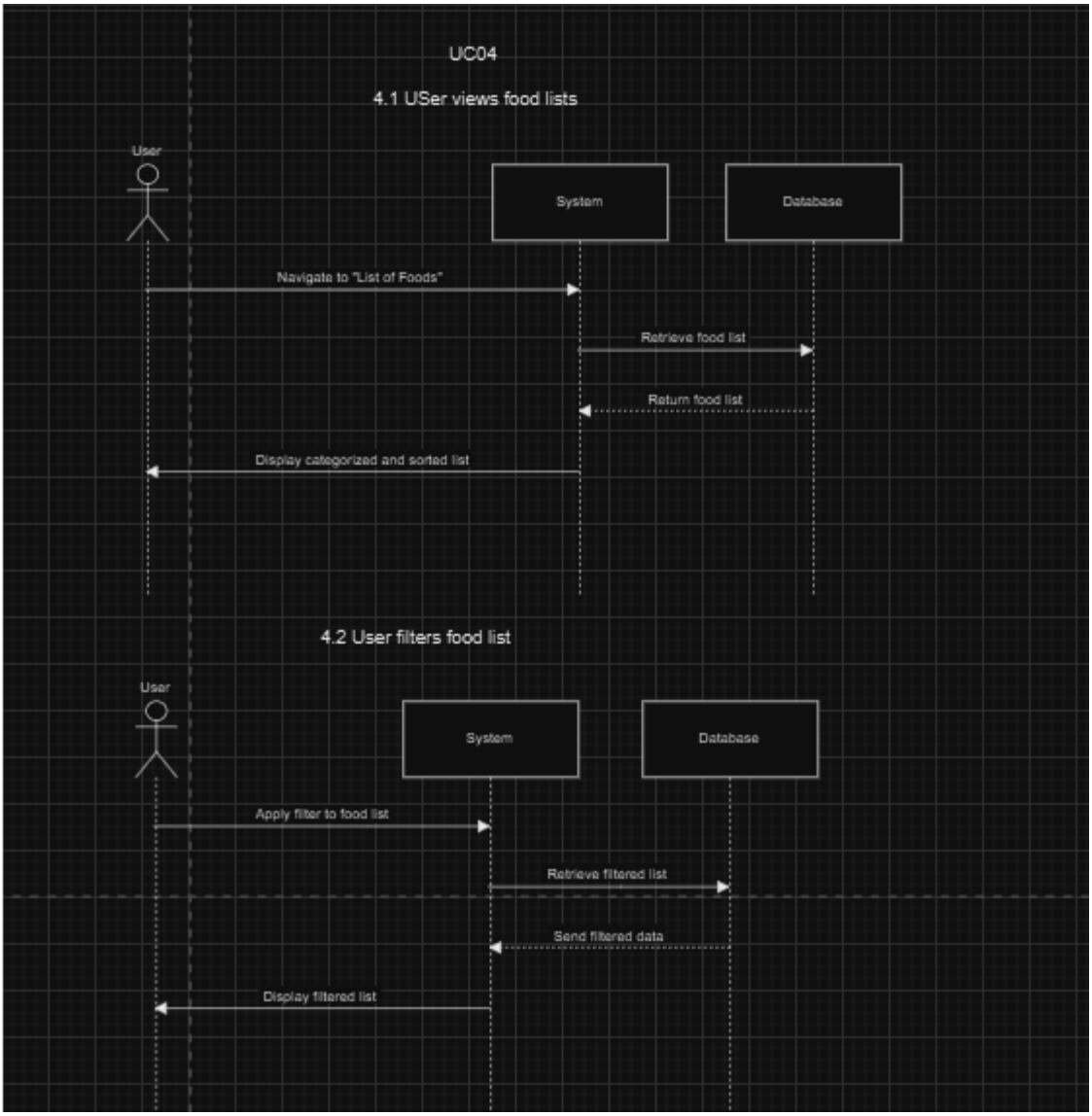

1.3 Admin edits food item
Admin
System
Database
Submit food item edits
Validate input
alt
[Valid Input]
Update food item
Confirm update
Display confirmation
[Invalid Input]
Prompt to correct data



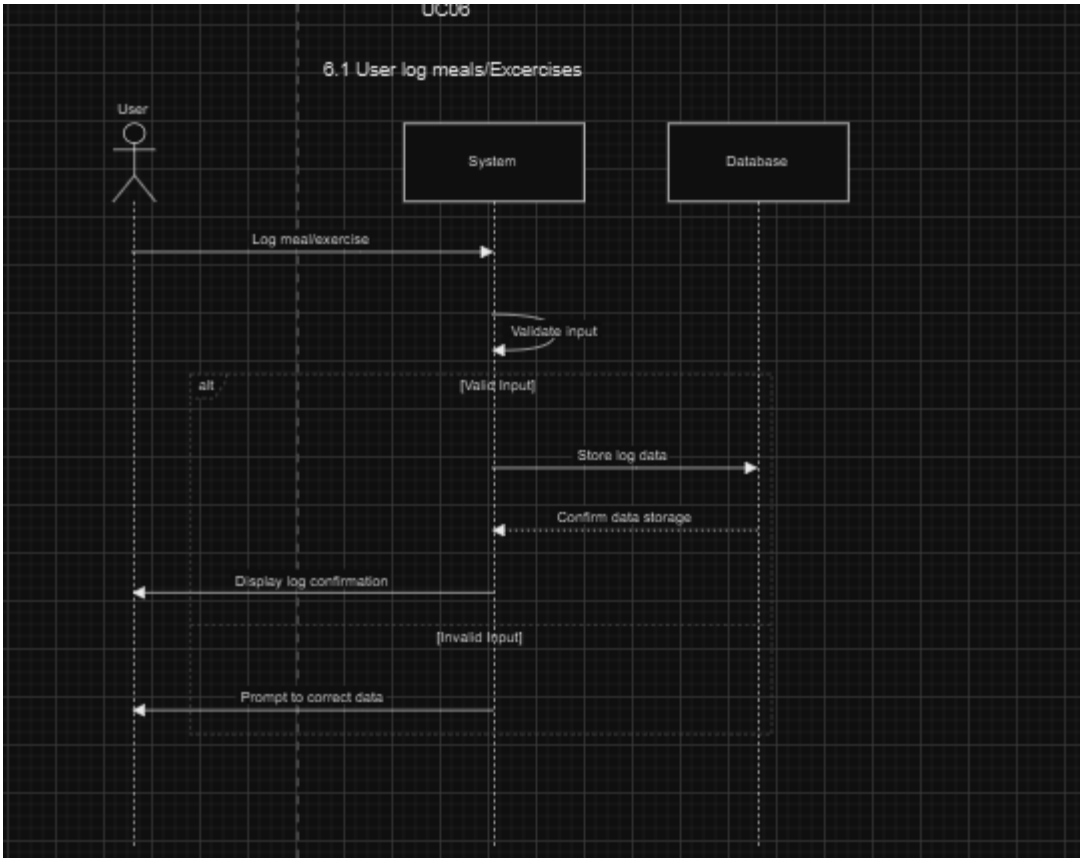
2. Display Nutritional Information

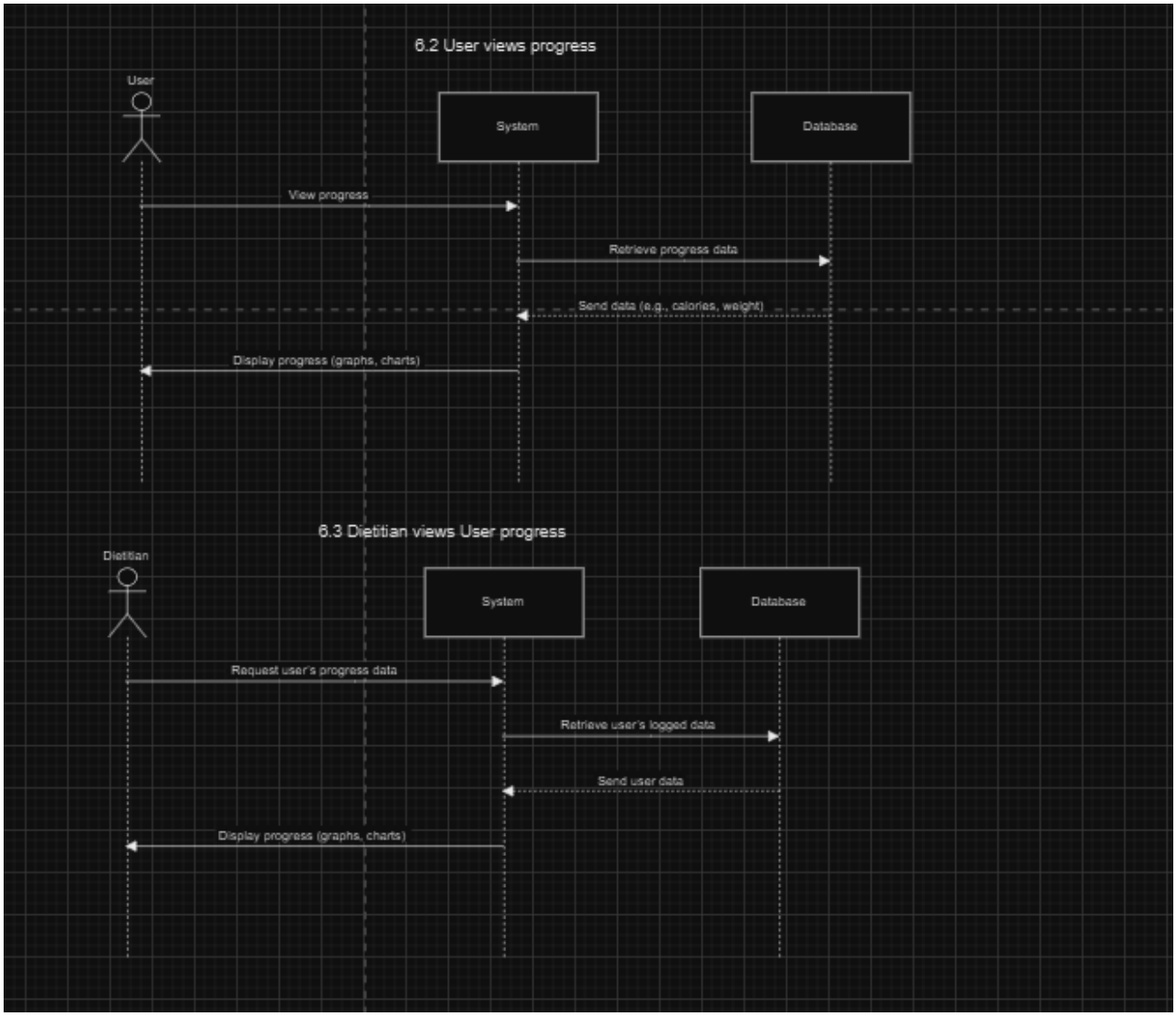


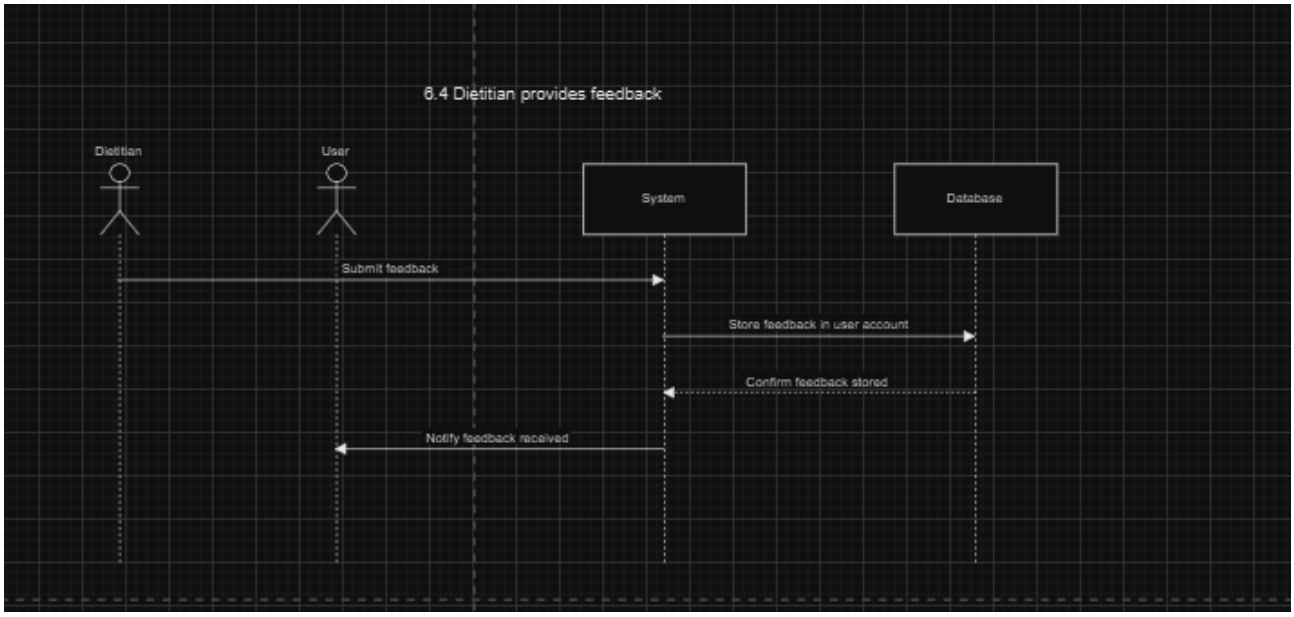
3. Provide Food Lists



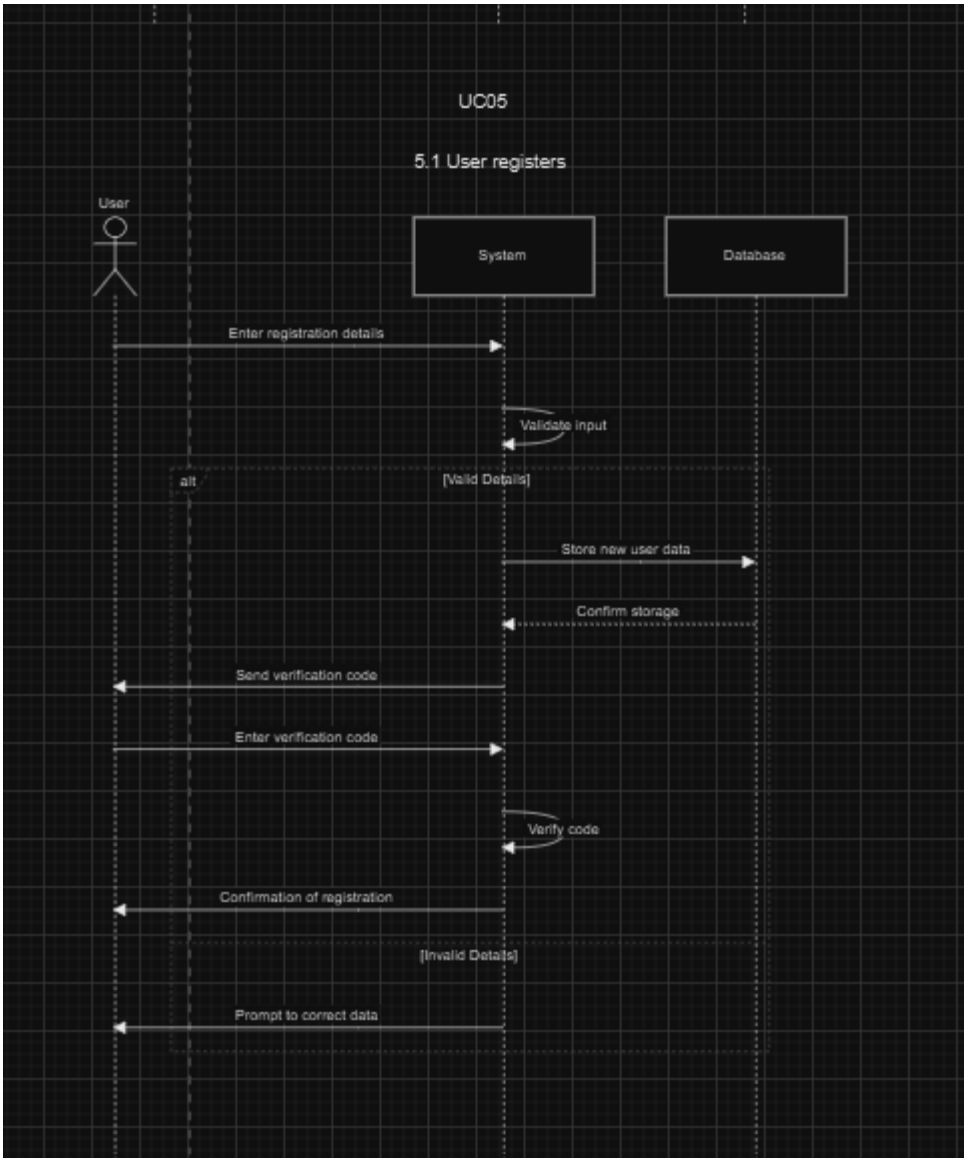
4. Track Progress

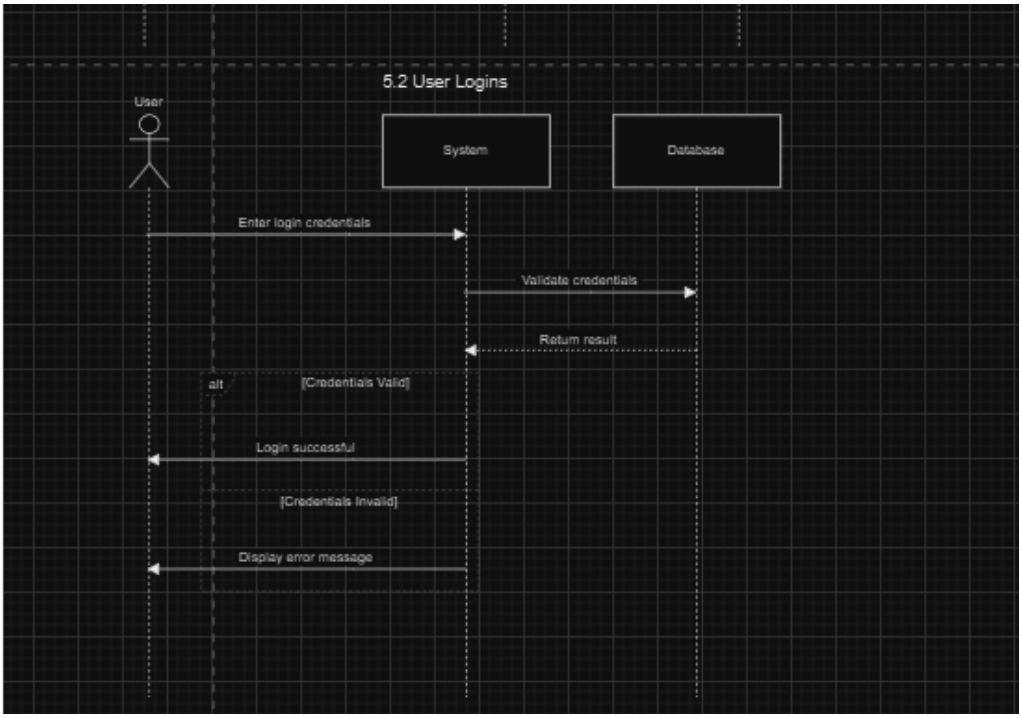

UC06
6.1 User log meals/Excercises
User
System
Database
Log meal/exercise
Validate input
alt
[Valid input]
Store log data
Confirm data storage
Display log confirmation
[Invalid input]
Prompt to correct data

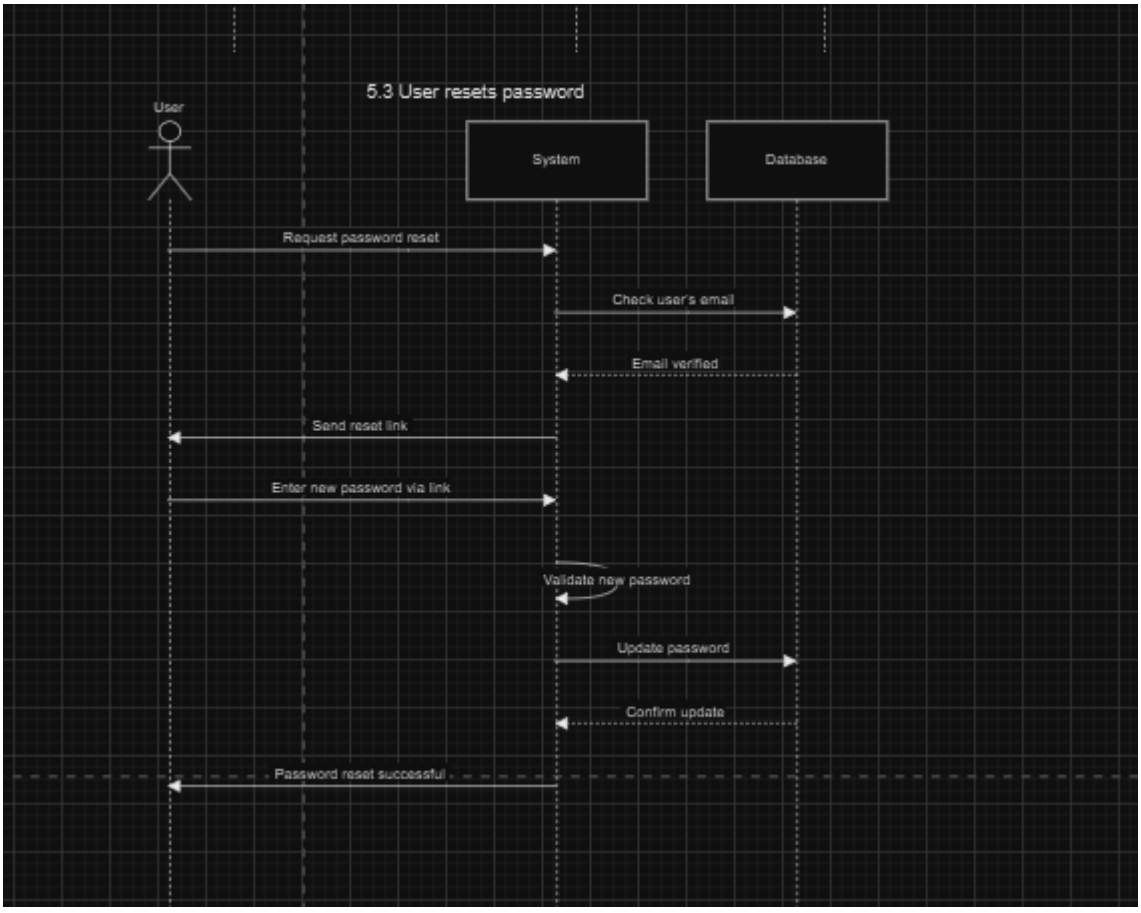

6.2 User views progress
User
System
Database
View progress
Retrieve progress data
Send data (e.g., calories, weight)
Display progress (graphs, charts)
6.3 Dietitian views User progress
Dietitian
System
Database
Request user's progress data
Retrieve user's logged data
Send user data
Display progress (graphs, charts)



5. Log In/Register

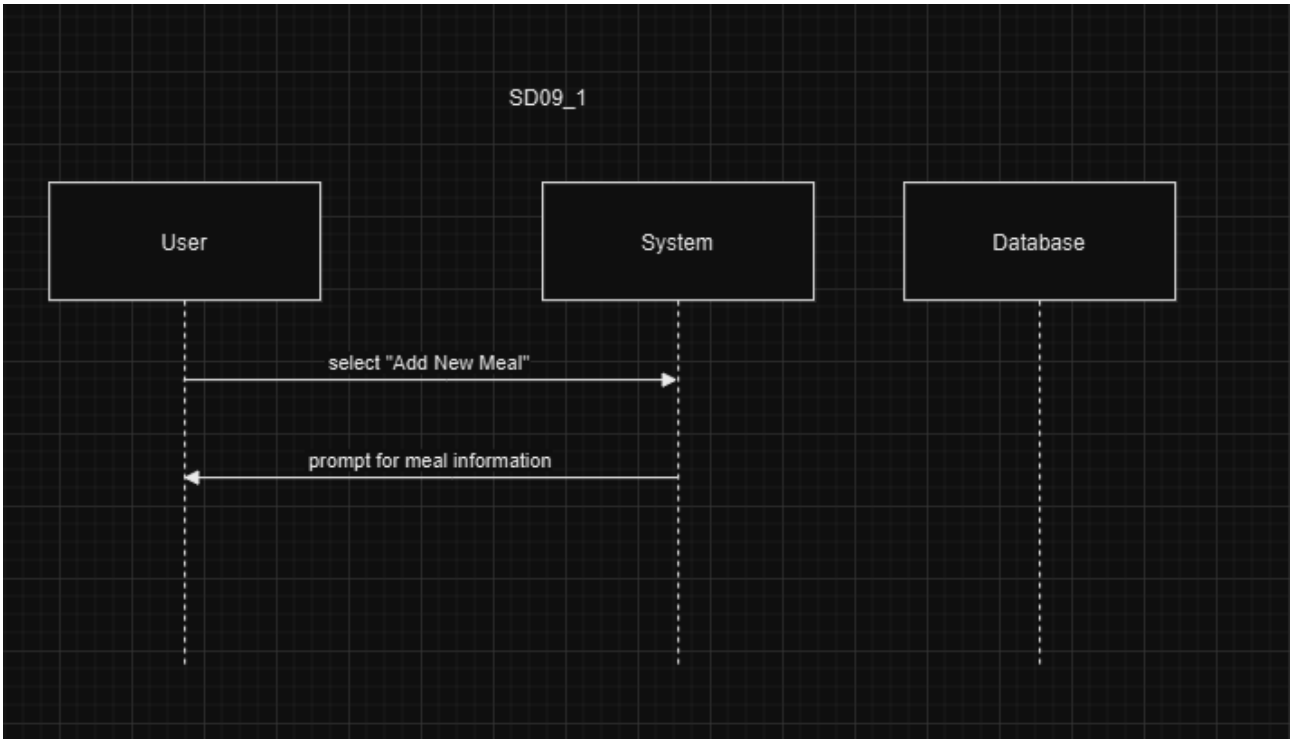

UC05
5.1 User registers
User
System
Database
Enter registration details
Validate input
alt
[Valid Details]
Store new user data
Confirm storage
Send verification code
Enter verification code
Verify code
Confirmation of registration
[Invalid Details]
Prompt to correct data

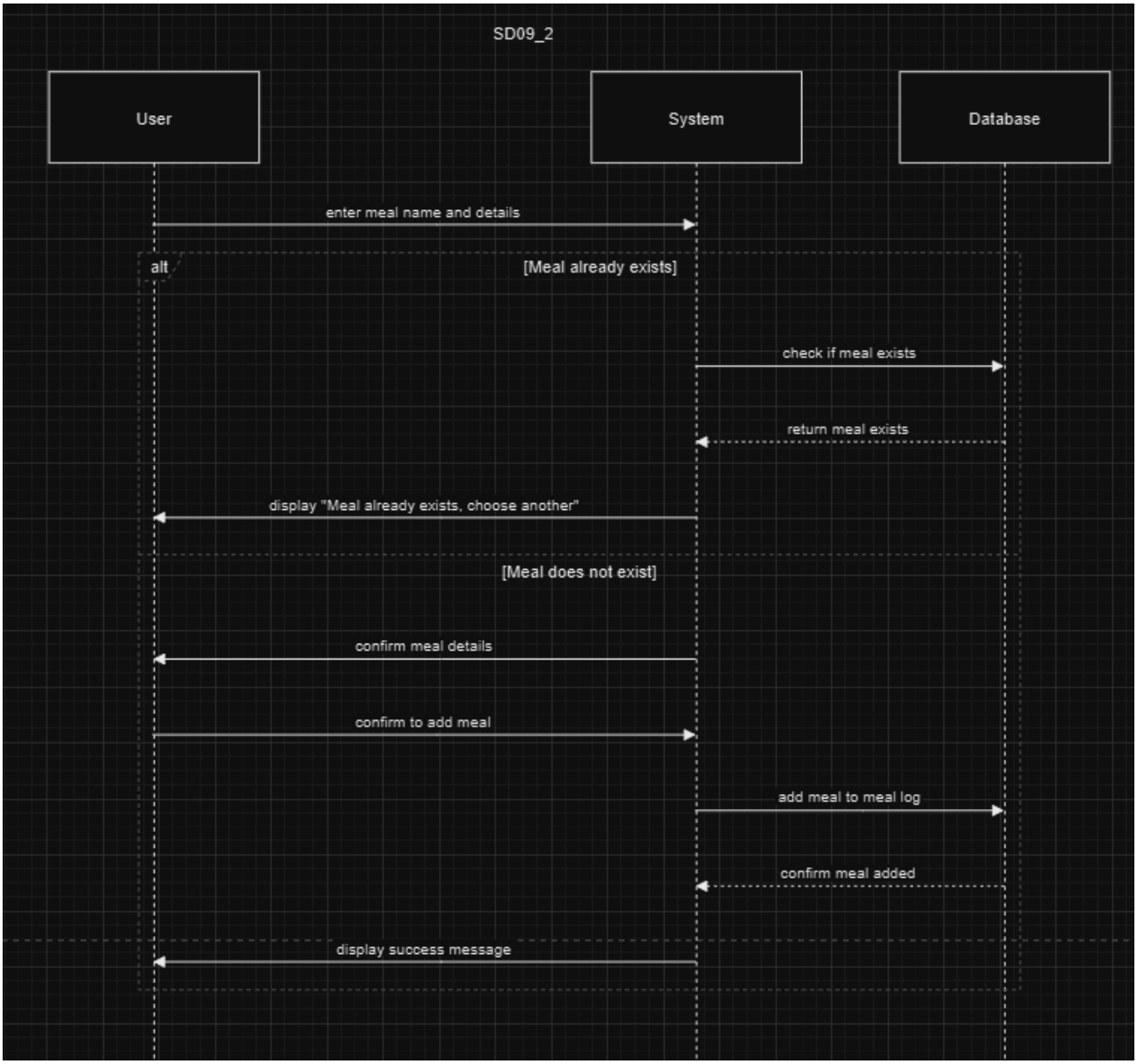




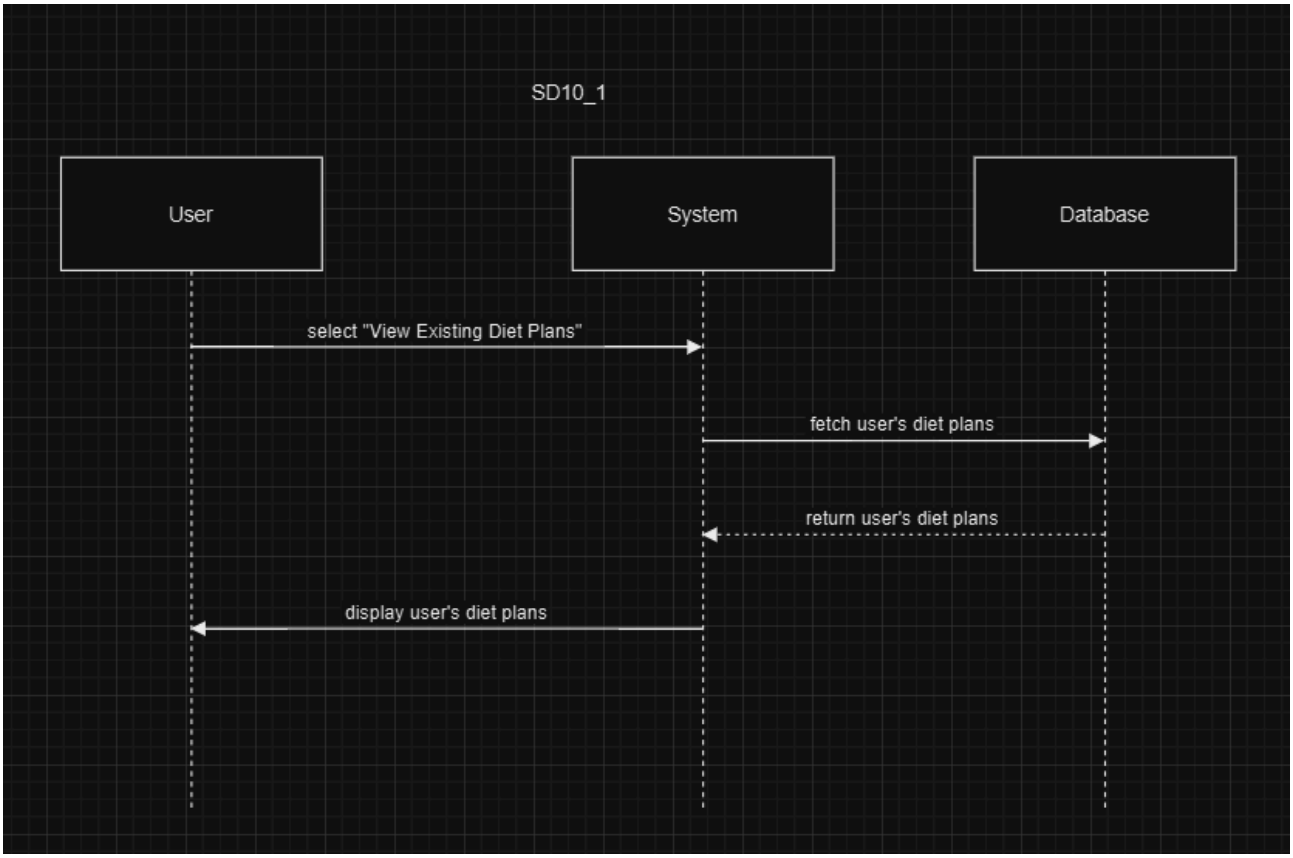
6. **Generate Diet Plan**
7. **Review Diet Plans**

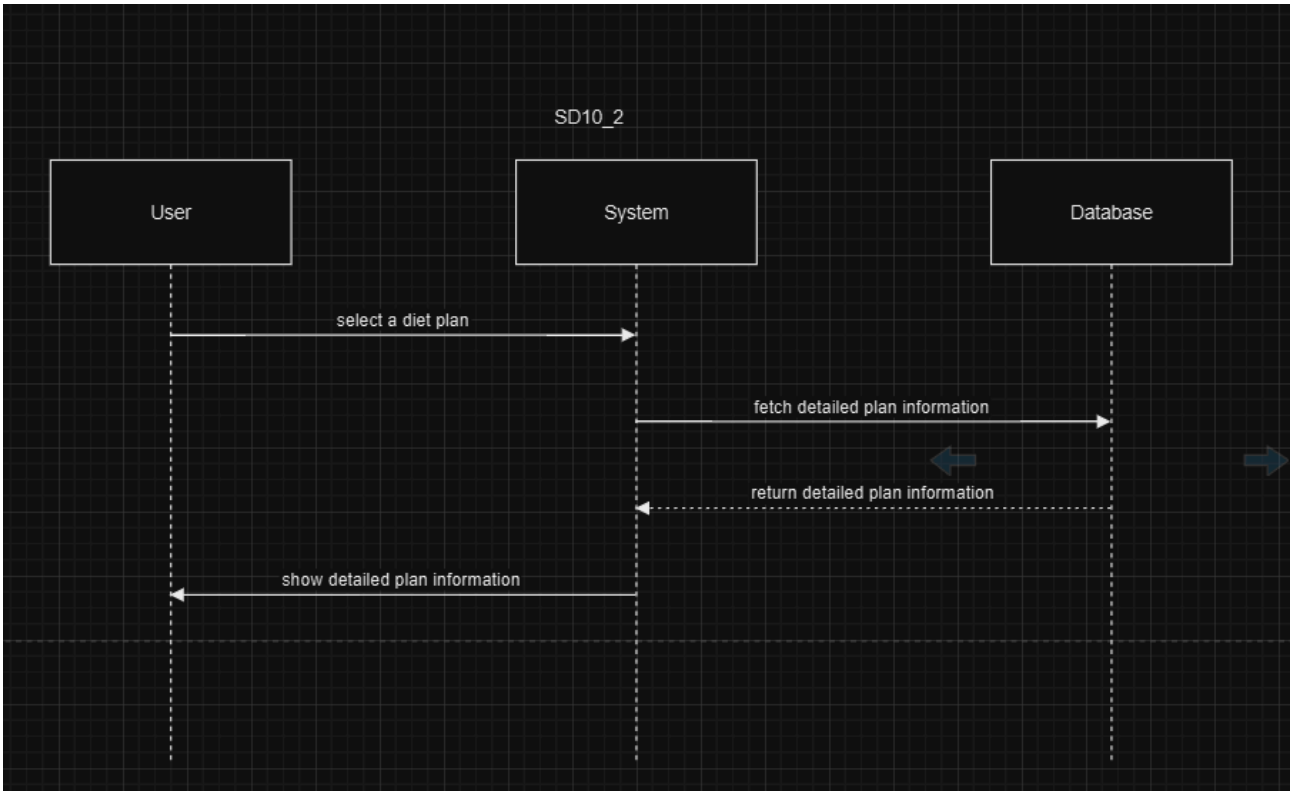
8. Add New Meal

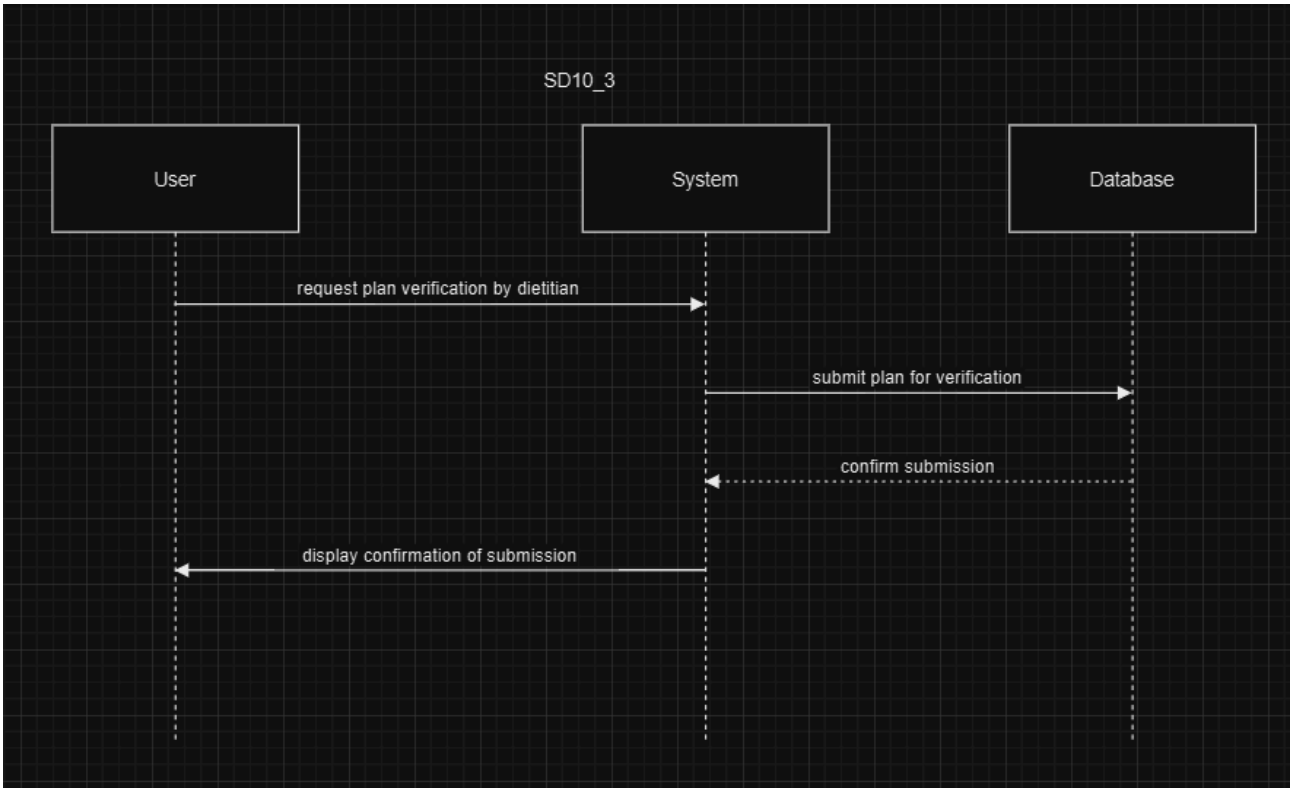




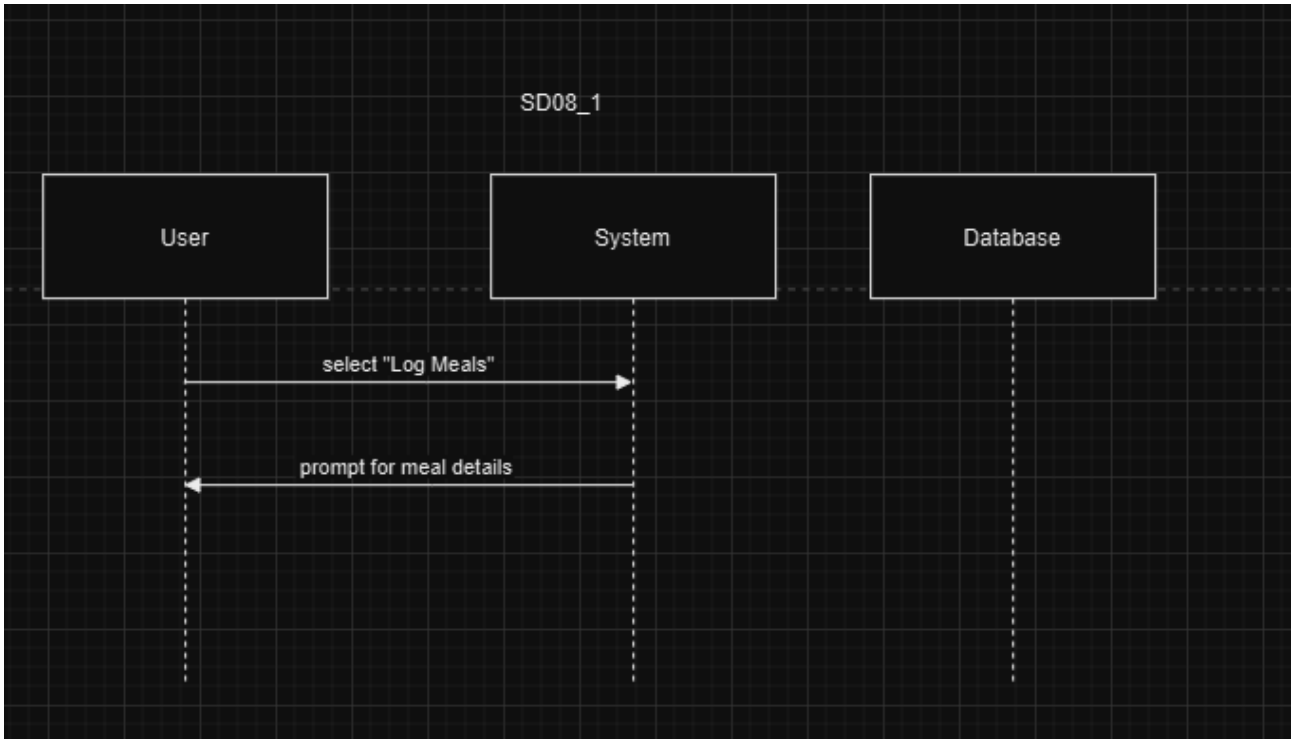
9. View Existing Diet Plans

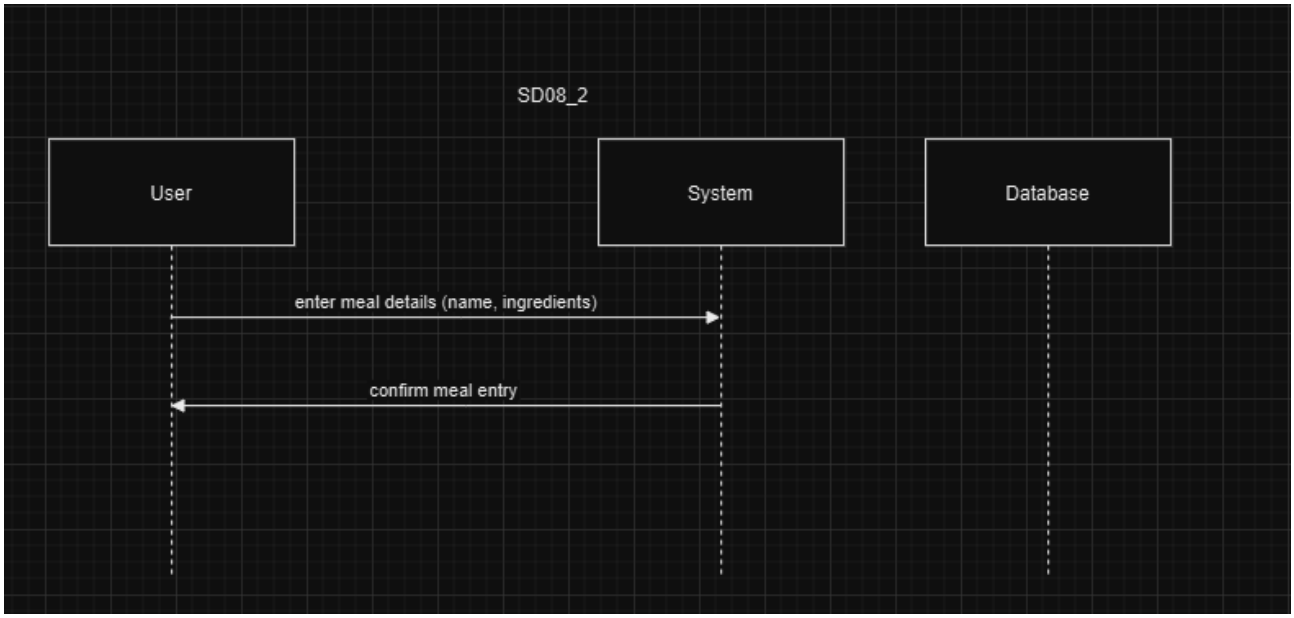

SD10_1
User
System
Database
select "View Existing Diet Plans"
fetch user's diet plans
return user's diet plans
display user's diet plans

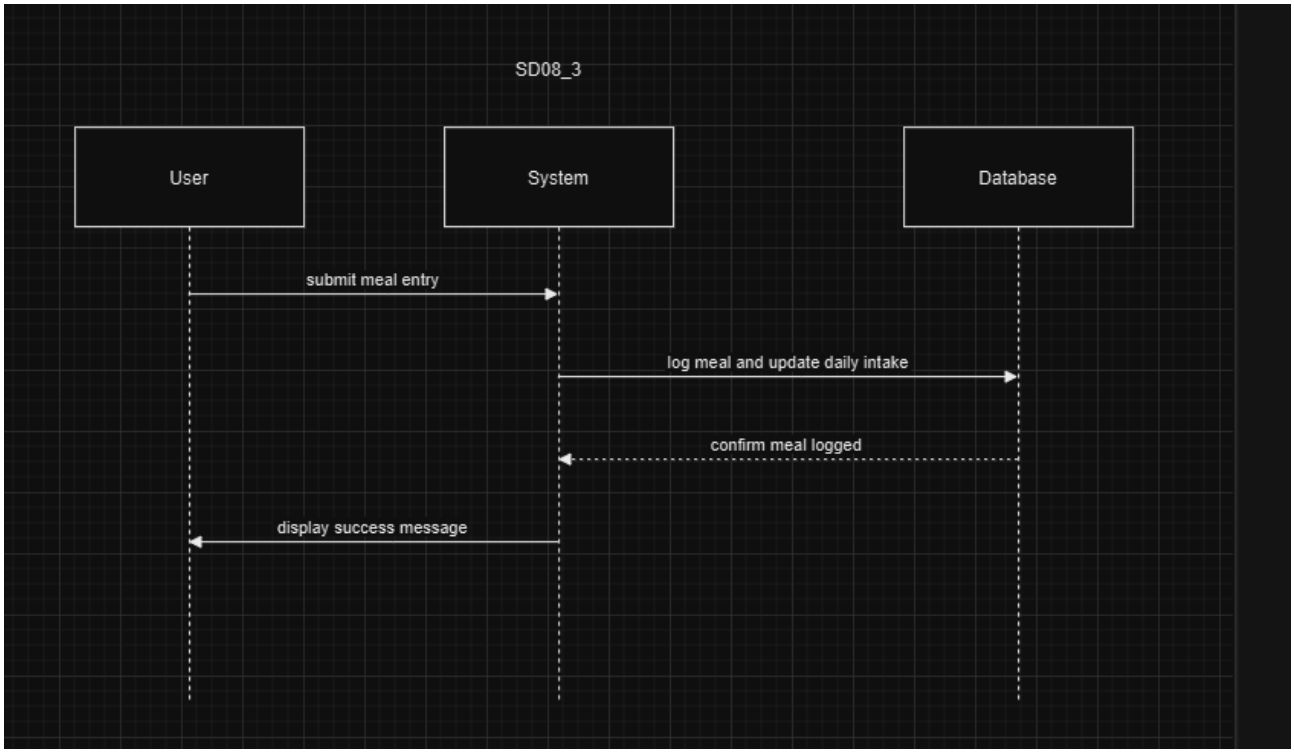

SD10_2
User
System
Database
select a diet plan
fetch detailed plan information
return detailed plan information
show detailed plan information


SD10_3
User
System
Database
request plan verification by dietitian
submit plan for verification
confirm submission
display confirmation of submission

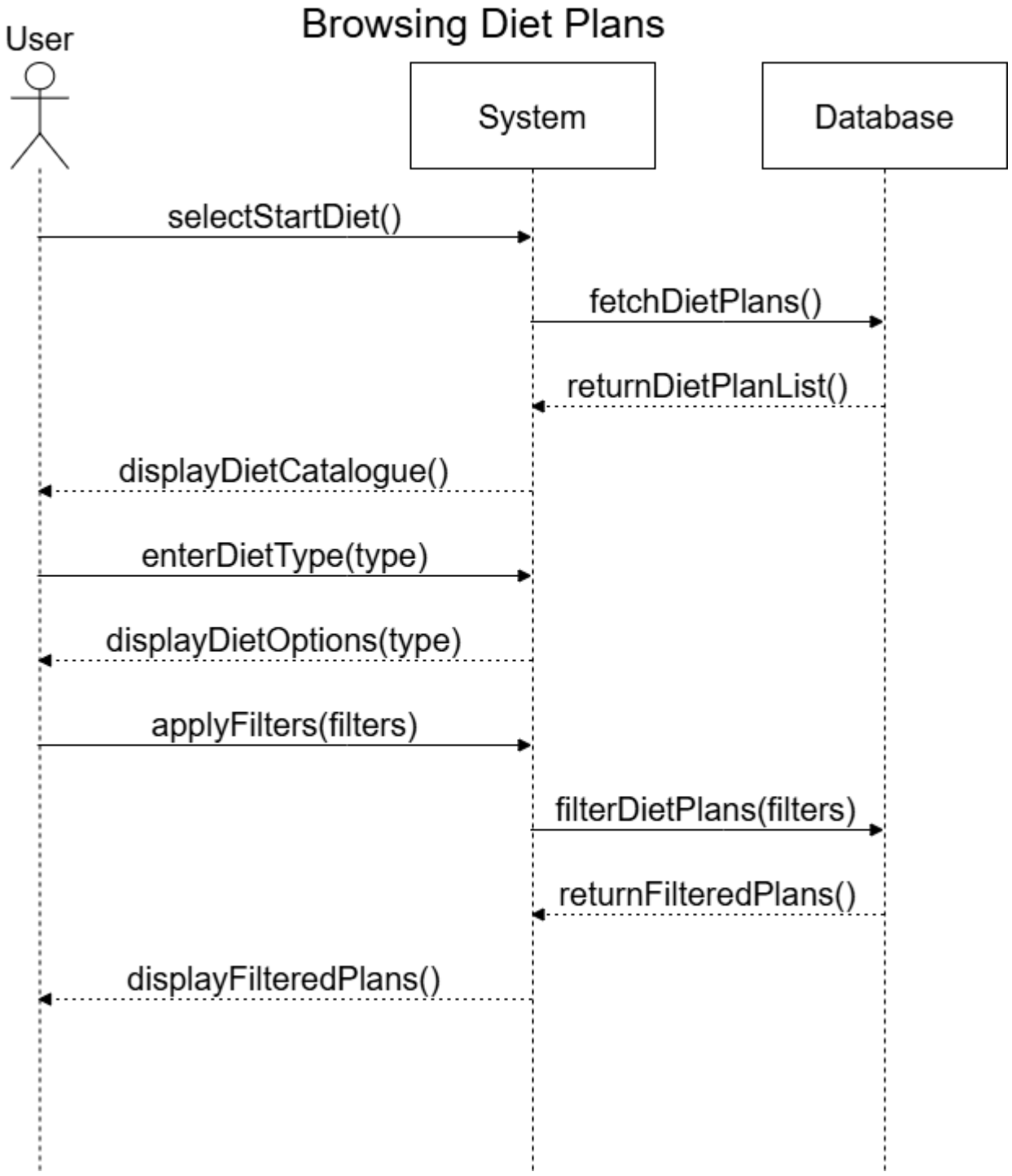
10. Log Meals

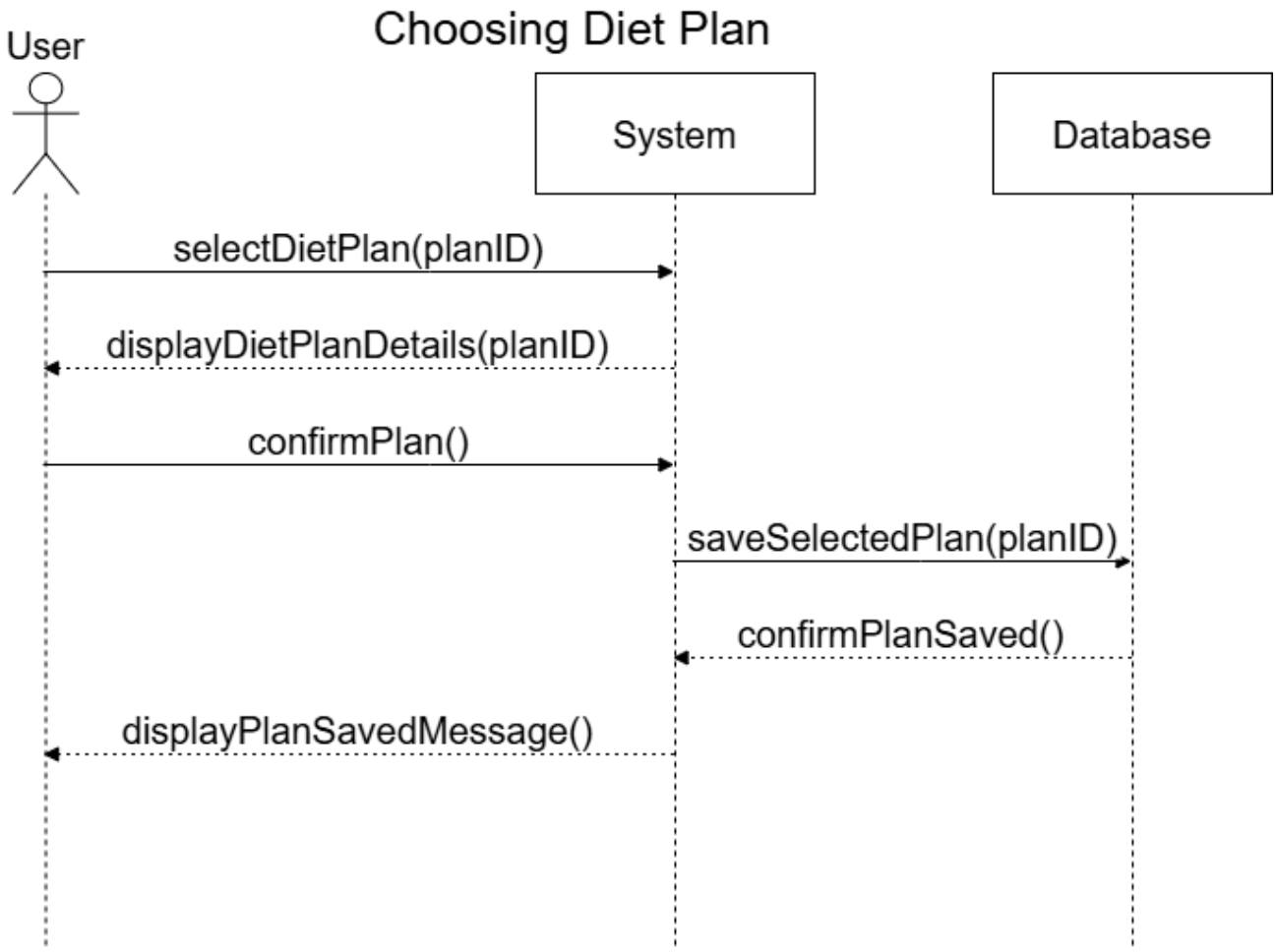

SD08_1
User
System
Database
select "Log Meals"
prompt for meal details



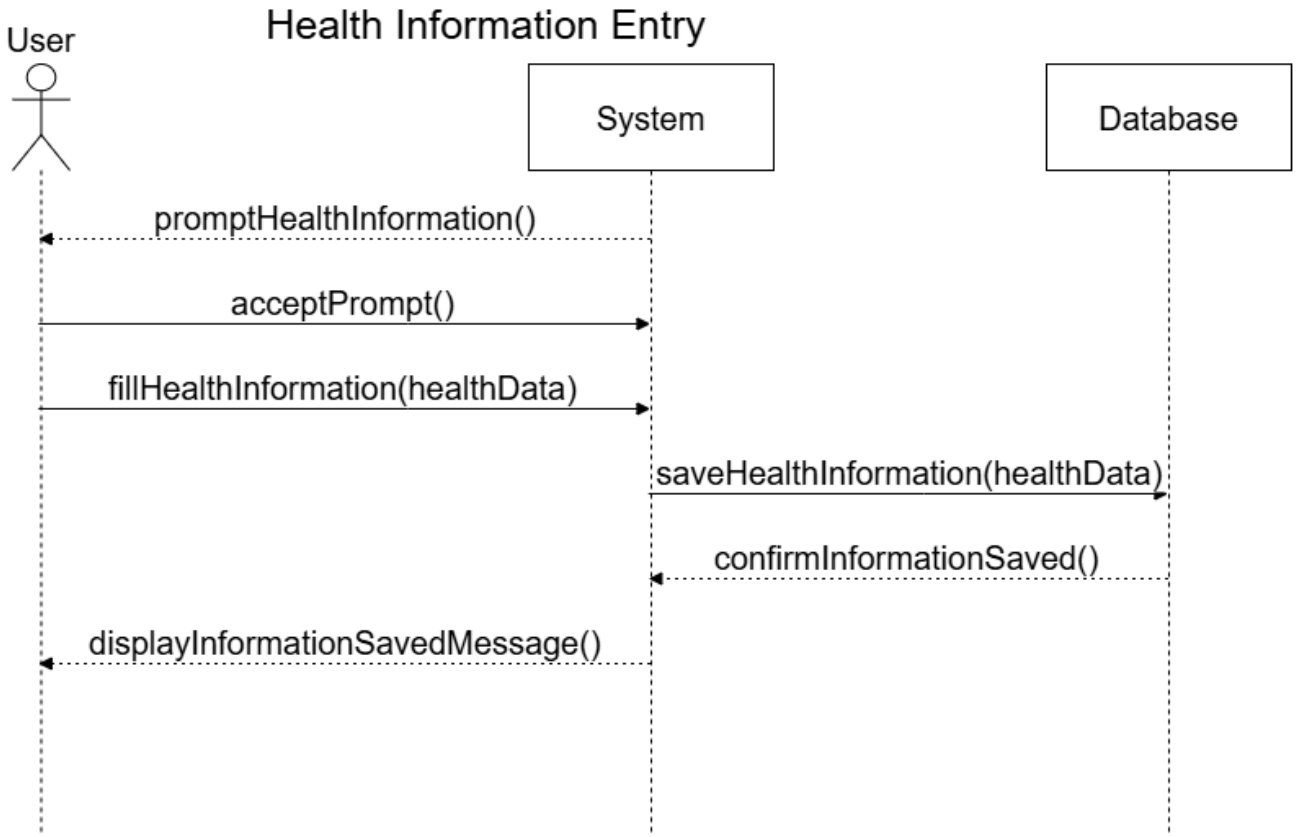


11. Search Diet Plans

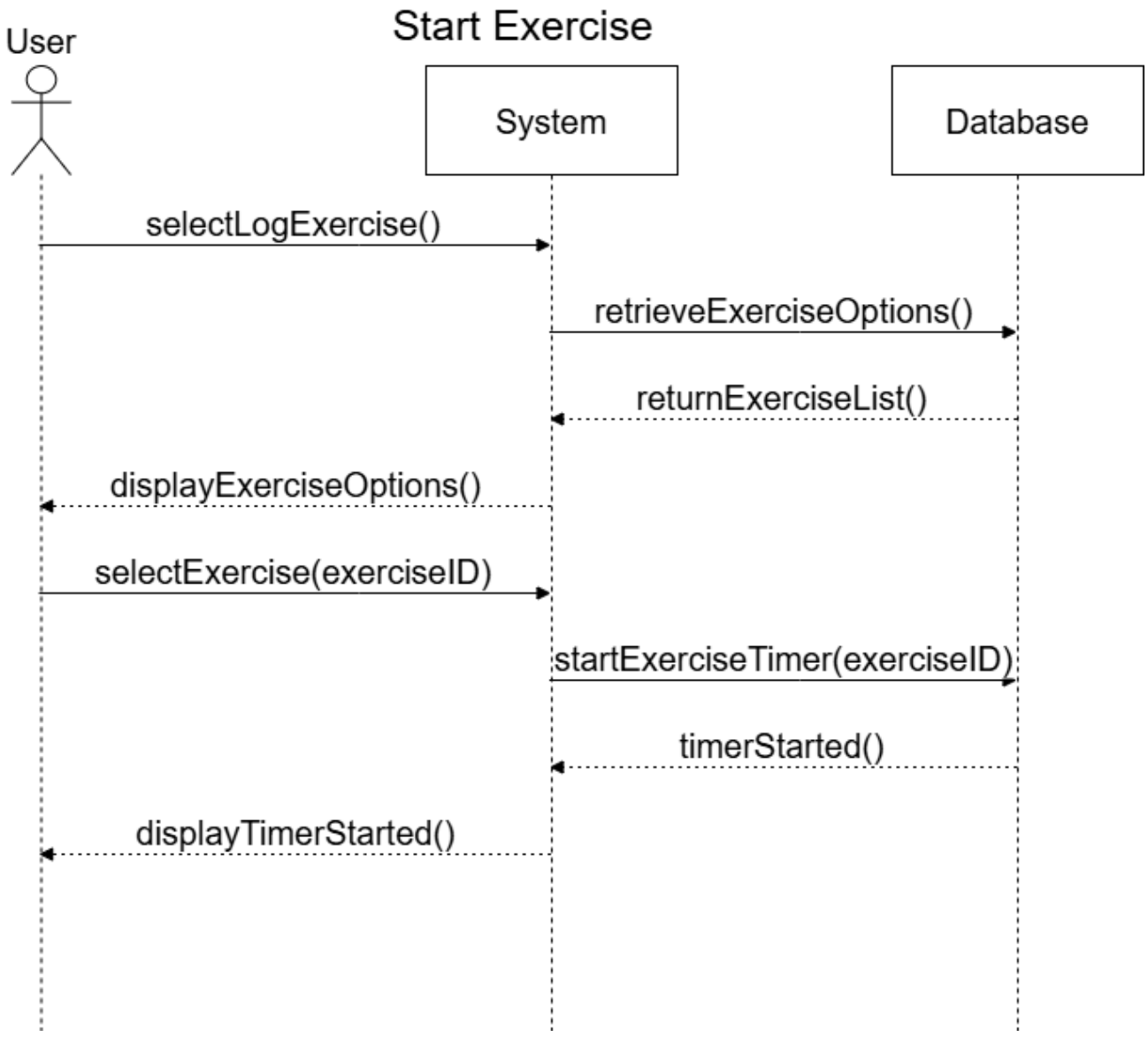

Browsing Diet Plans
User
System
Database
selectStartDiet()
fetchDietPlans()
returnDietPlanList()
displayDietCatalogue()
enterDietType(type)
displayDietOptions(type)
applyFilters(filters)
filterDietPlans(filters)
returnFilteredPlans()
displayFilteredPlans()

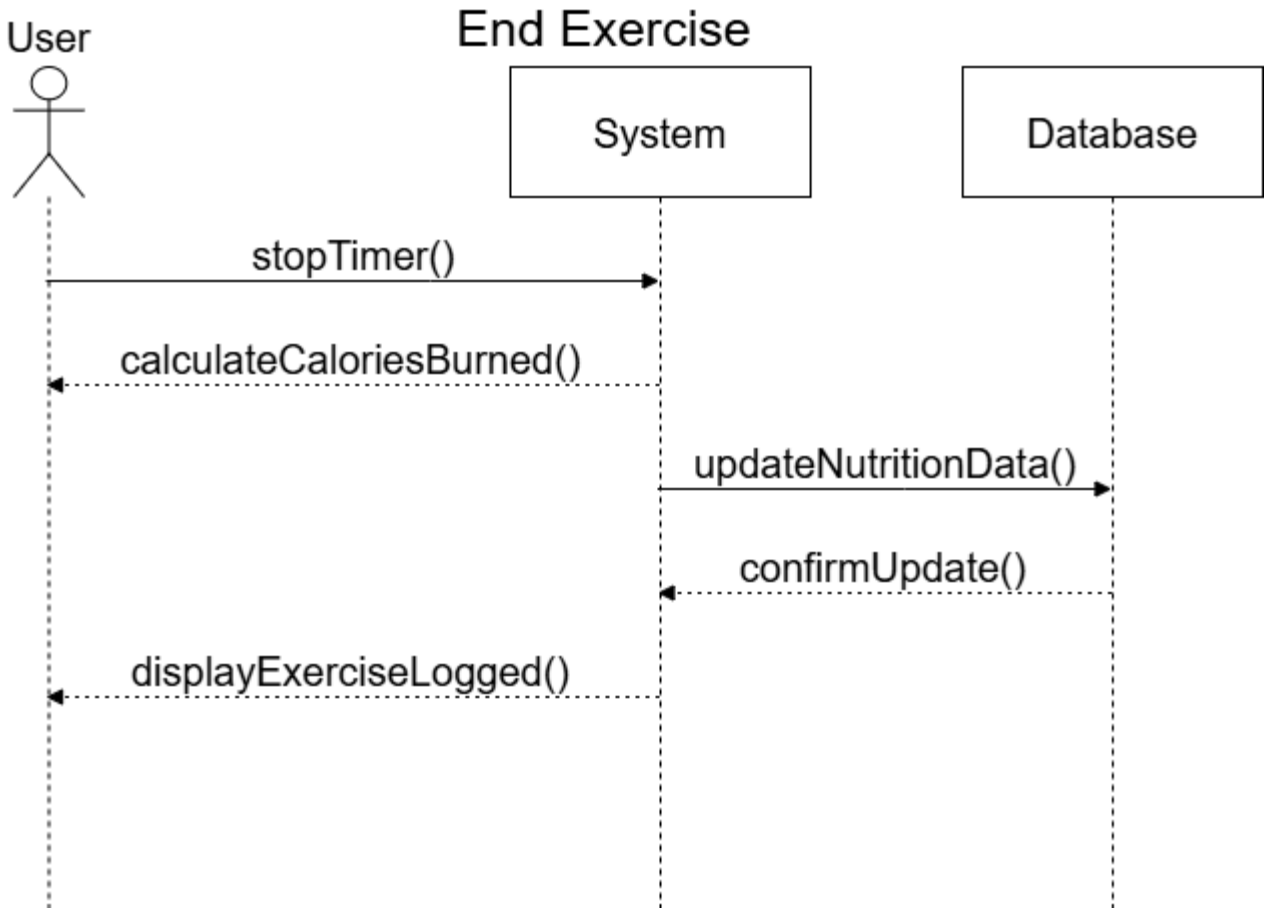


12. Add Health Information

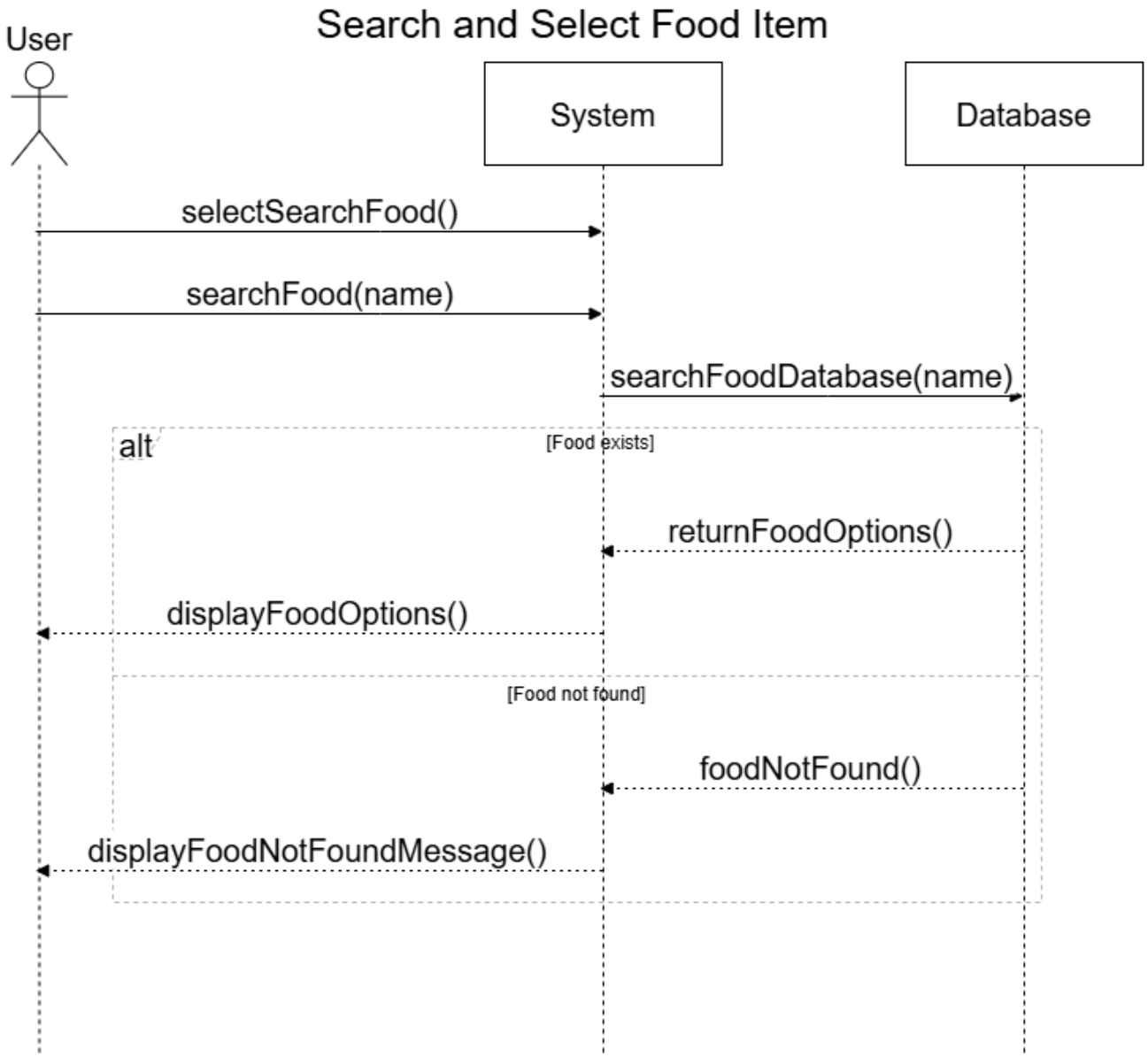


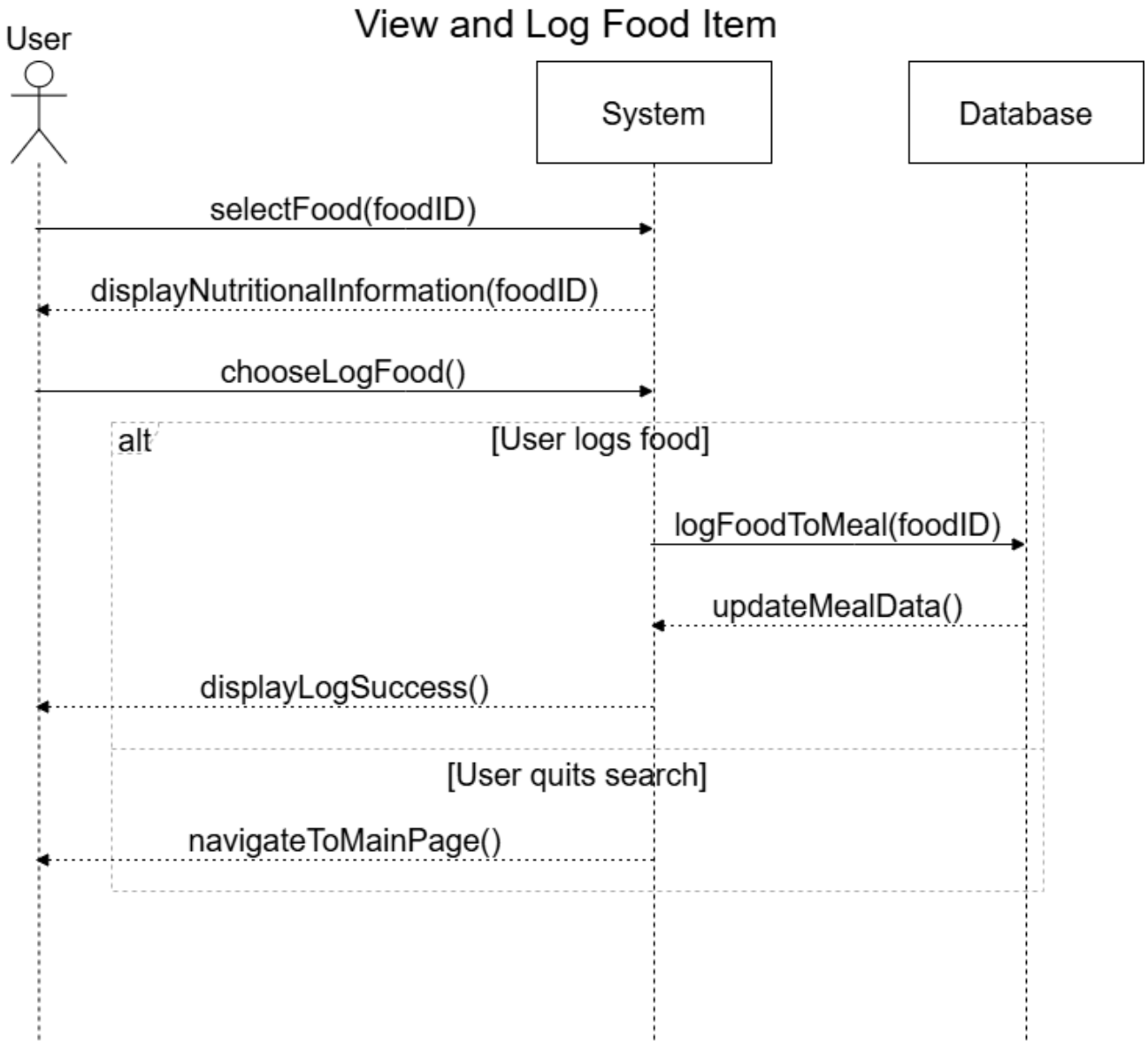
13. Record Exercise


Start Exercise
User
System
Database
selectLogExercise()
retrieveExerciseOptions()
returnExerciseList()
displayExerciseOptions()
selectExercise(exerciseID)
startExerciseTimer(exerciseID)
timerStarted()
displayTimerStarted()

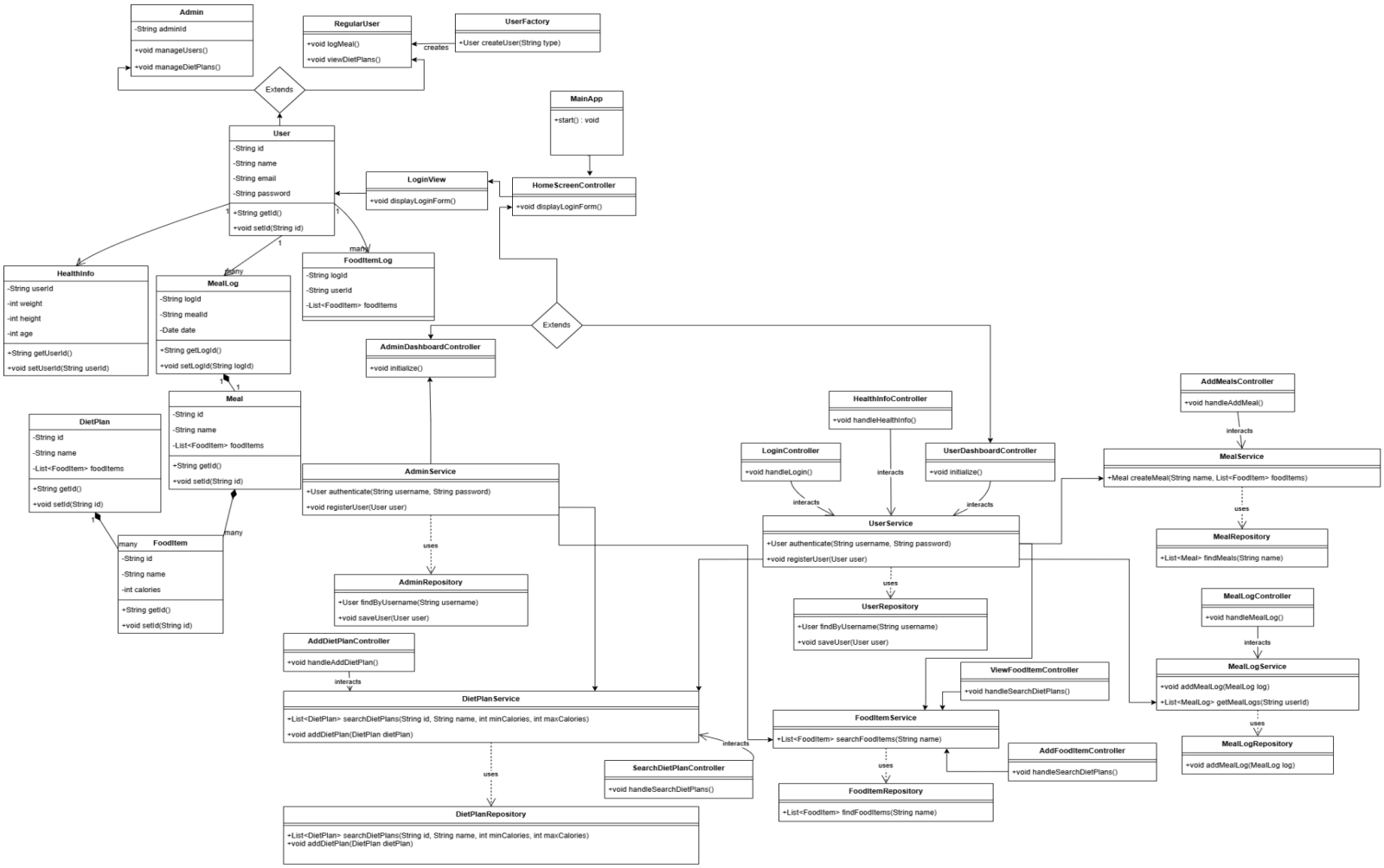


14. Search Food Items

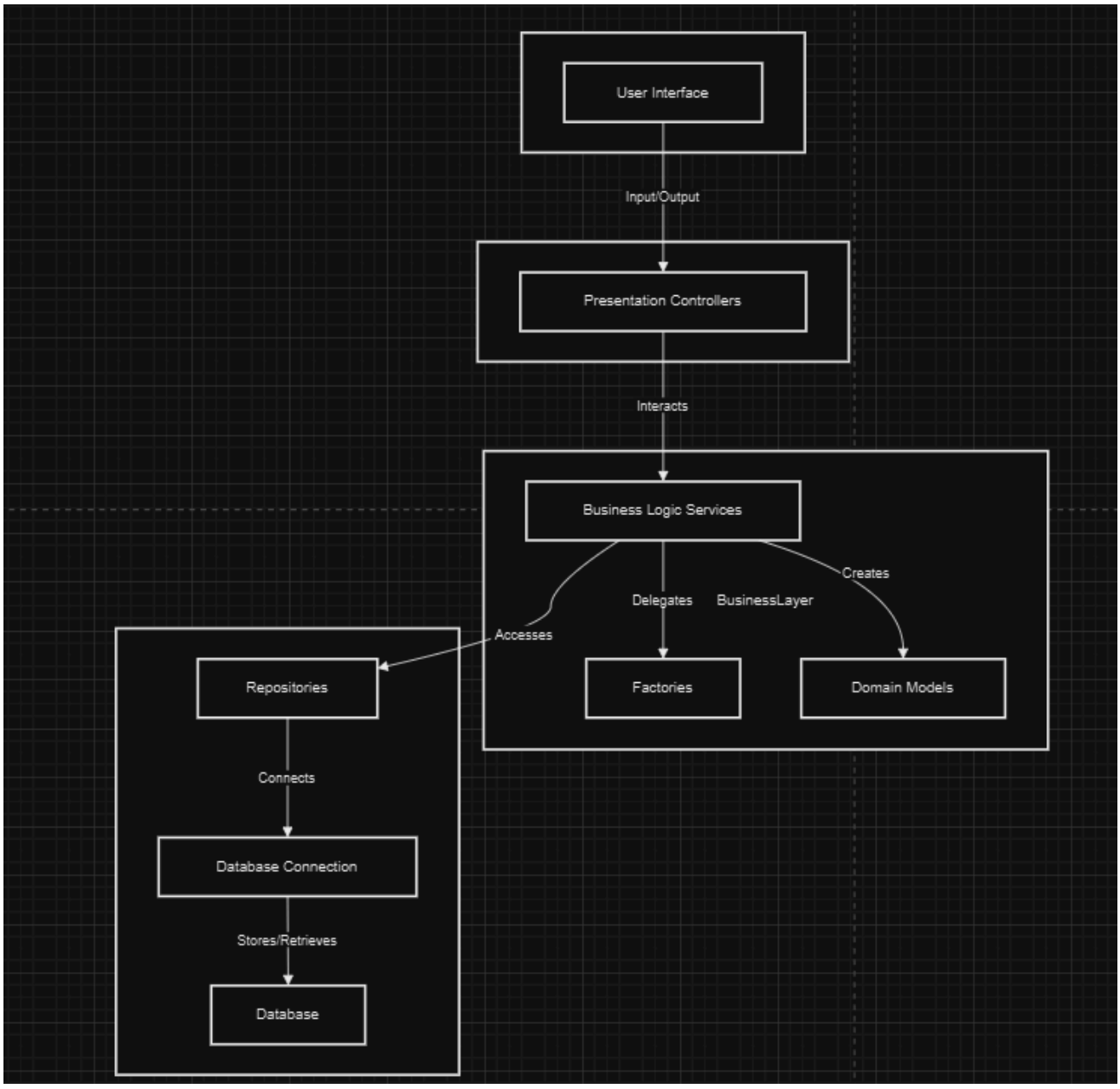

Search and Select Food Item
User
System
Database
selectSearchFood()
searchFood(name)
searchFoodDatabase(name)
alt
[Food exists]
returnFoodOptions()
displayFoodOptions()
[Food not found]
foodNotFound()
displayFoodNotFoundMessage()


View and Log Food Item
User
System
Database
selectFood(foodID)
displayNutritionalInformation(foodID)
chooseLogFood()
alt
[User logs food]
logFoodToMeal(foodID)
updateMealData()
displayLogSuccess()
[User quits search]
navigateToMainPage()

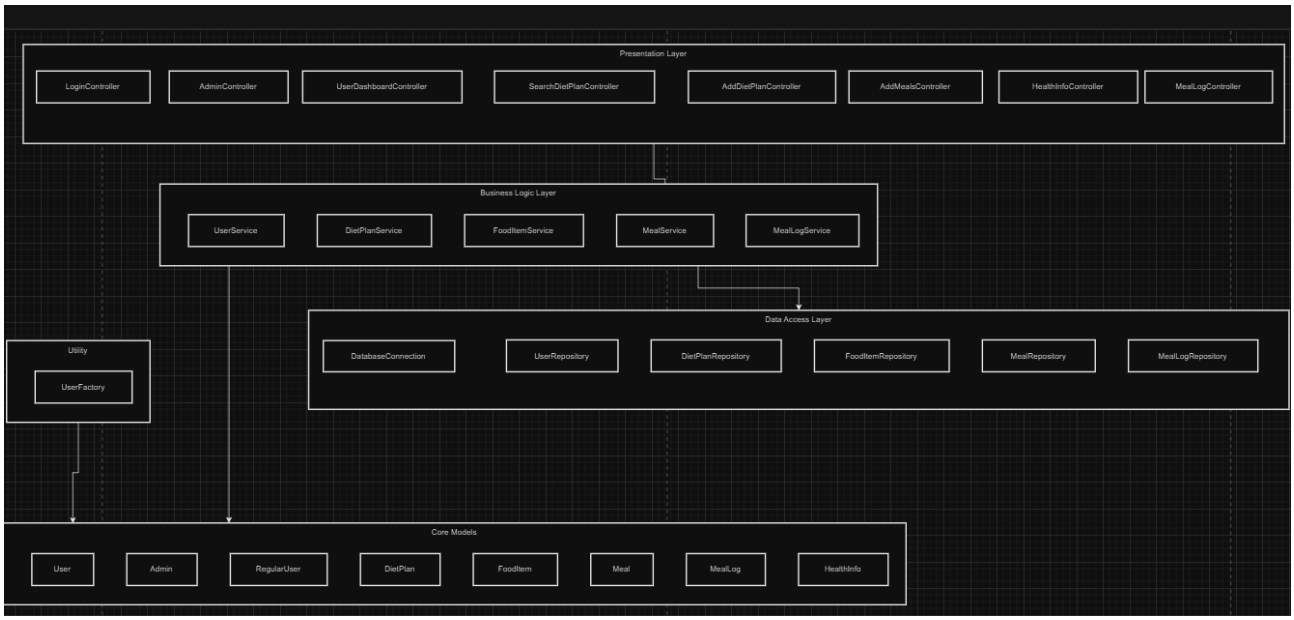
Class Diagram



Component Diagram



Package Diagram



Deployment Diagram