OVERVIEW

A1

Conceptual Model:

Entities:

- **User Account:**
  - Attributes: Username, Password (hashed), Account Status
- **Error Message:**
  - Attributes: Error Code, Error Description

Boundary Classes:

- **Login Page**
  - Stereotype: Boundary
- **Error Message Display**
  - Stereotype: Boundary

Control Classes:

- **Login Controller:**
  - Stereotype: Control

A2

Conceptual Model:

Entities:

- **User Account:**
  - Attributes: Name, Email, Username, Password (hashed), Account Status

Boundary Classes:

- **Registration Page:**
  - Stereotype: Boundary
- **Error Message Display:**
  - Stereotype: Boundary
- **Confirmation Email Interface:**
  - Stereotype: Boundary

Control Classes:

- **Registration Controller:**
  - Stereotype: Control
- **Account Validation:**
  - Stereotype: Control
- **Email Confirmation Processor:**
  - Stereotype: Control

A3

Conceptual Model:

Entities:

- **User Profile:**
  - Attributes: Name, Email, Contact Information, Health Metrics, Personal Information

Boundary Classes:

- **Personal Profile Page:**
  - Stereotype: Boundary
- **Error Message Display:**
  - Stereotype: Boundary

Control Classes:

- **Profile Management Controller:**
  - Stereotype: Control
- **Data Validation Service:**
  - Stereotype: Control

A4

Conceptual Model:

Entities:

- **Fitness Goal:**
  - Attributes: Goal Type, Target Metrics, Progress Tracking

Boundary Classes:

- **Fitness Goal Management Page:**
  - Stereotype: Boundary
- **Error Message Display:**
  - Stereotype: Boundary

Control Classes:

- **Fitness Goal Controller:**
  - Stereotype: Control
- **Fitness Guidance Service:**
  - Stereotype: Control

A5

Conceptual Model:

Entities:

- **Workout Session:**
  - Attributes: Exercise Type, Duration, Intensity

Boundary Classes:

- **Workout Tracking Page:**
  - Stereotype: Boundary
- **Error and Feedback Display:**
  - Stereotype: Boundary

Control Classes:

- **Progress Tracking Controller:**
  - Stereotype: Control
- **Workout Analysis Tool:**
  - Stereotype: Control

A6

Conceptual Model:

Entities:

- **Calorie Intake:**
  - Attributes: Food Items, Calorie Values

Boundary Classes:

- **Calorie Tracking Page:**
  - Stereotype: Boundary
- **Food Item Input Interface:**
  - Stereotype: Boundary

Control Classes:

- **Calorie Tracking Controller:**
  - Stereotype: Control
- **Nutrition Calculator:**
  - Stereotype: Control

A7

Conceptual Model:

Entities:

- **Workout Plan:**
  - Attributes: Exercise Routines, Sets, Reps, Progression

Boundary Classes:

- **Personalized Workout Plan Page:**
  - Stereotype: Boundary
- **Workout Tracking Interface:**
  - Stereotype: Boundary

Control Classes:

- **Workout Plan Generator:**
  - Stereotype: Control
- **Workout Plan Controller:**
  - Stereotype: Control

A8

Conceptual Model:

Entities:

- **Recipe:**
  - Attributes: Ingredients, Cooking Instructions, Nutritional Information

Boundary Classes:

- **Recipe Book Page:**
  - Stereotype: Boundary
- **Recipe Details Interface:**
  - Stereotype: Boundary

Control Classes:

- **Recipe Recommendation Engine:**
  - Stereotype: Control
- **Recipe Management Controller:**
  - Stereotype: Control

A9

Conceptual Model:

Entities:

- **Gym Location:**
  - Attributes: Name, Address, Operating Hours

Boundary Classes:

- **Gym Location Page:**
  - Stereotype: Boundary
- **Location Input Interface:**
  - Stereotype: Boundary

Control Classes:

- **Gym Location Finder:**
  - Stereotype: Control
- **Real-time Capacity Checker:**
  - Stereotype: Control

DETAILED INFO

A1

Boundary Class

Login Page:

Responsibilities: Serves as the interface where users input their login credentials (username/email and password). It also presents the "Login" button that users click to submit their credentials and the "Forgot Password?" link for users who need to reset their password. Displays error messages received from the Login Controller in case of failed login attempts.

Interactions: Receives user input and interacts with the Login Controller to submit the credentials for verification. Receives error details from the Login Controller and presents them to the user. This includes errors from the main login flow as well as alternative flows, like account lockout notifications.

Error Message Display:

Responsibilities: Specifically designed to show error messages related to login attempts, such as incorrect credentials or account lockout messages after too many failed attempts. This interface is crucial for informing the user of the nature of the login issue and suggesting corrective actions.

Interactions: Receives error details from the Login Controller and presents them to the user. This includes errors from the main login flow as well as alternative flows, like account lockout notifications.

Control Class

Login Controller:

Responsibilities: Manages the login process, including receiving credentials from the Login Page, invoking the Verify Login use case to check credentials, managing session initiation upon successful login, and handling errors or failed login attempts.

Interactions: Interacts directly with the Login Page to receive user credentials and with the User Account entity to verify these credentials. Determines the outcome of the login attempt (success or failure) and communicates with the Error Message Display to inform the user of the result.

Entity Class

User Account:

Responsibilities: Stores user authentication and profile information, including usernames, hashed passwords, and account status (e.g., active, locked). It represents the user's identity within the system and is the primary entity involved in the login process.

Interactions: Accessed by the Login Controller to validate login credentials against stored information. The User Account's status may be updated based on the outcome of the login attempt (e.g., locked after too many failed attempts).

A2

Boundary Class

Registration Page:

Responsibilities: Collects user-provided information such as name, email address, desired username, and password. Handles user interactions with the terms of service and privacy policy agreements. Receives and displays validation error messages from the Registration Controller.

Interactions: Directly interacts with the user to gather input and with the Registration Controller to submit the collected data. May also receive and display validation error messages from the Registration Controller.

Error Message Display:

Responsibilities: Displays specific error messages to the user, including validation issues (e.g., missing required fields, password strength requirements not met) and conflicts (e.g., username already taken).

Interactions: Receives error information from the Registration Controller and presents it to the user in an understandable format.

Confirmation Email Interface:

Responsibilities: Represents the structure and content of the confirmation email sent to users, including the confirmation link that users must click to verify their email address.

Interactions: Managed by the Email Confirmation Processor, which populates the email content and sends it to the user's provided email address. The user interacts with this interface outside the system environment (e.g., in their email client).

Control Class

Registration Controller:

Responsibilities: Orchestrates the registration process. Validates user input, ensures uniqueness and compliance with requirements, creates the User Account entity upon successful validation, initiates sending the confirmation email, and manages error feedback.

Interactions: Receives data from the Registration Page, communicates with the Account Validation class for input validation, interacts with the database to create the User Account, and instructs the Email Confirmation Processor to send the confirmation email. Sends error messages back to the Registration Page for display.

Account Validation:

Responsibilities: Performs detailed checks on the registration input, such as verifying the uniqueness of usernames and email addresses against the system's database and ensuring all inputs meet the specified criteria (e.g., password strength, valid email format).

Interactions: Invoked by the Registration Controller to validate user input. Accesses the database to check for existing usernames and email addresses. Returns validation results to the Registration Controller.

Email Confirmation Processor:

Responsibilities: Handles the generation, sending, and processing of confirmation emails. Creates a unique confirmation link, sends the email to the user's provided address, and processes clicks on the confirmation link to verify the user's email address.

Interactions: Triggered by the Registration Controller after successful user account creation. Generates and sends the confirmation email and updates the User Account status to active upon the user's confirmation click. Interacts with the database to update the User Account entity.

Entity Class

User Account:

Responsibilities: Stores user authentication and profile information, including usernames, hashed passwords, and account status (e.g., active, locked). Represents the user's identity within the system and is involved in the registration process.

Interactions: Accessed by the Registration Controller to validate registration input against stored information. Created and updated based on successful registration.

A3

Boundary Class

Personal Profile Page:

Responsibilities: Serves as the user interface where the user views and edits their personal and health information. It includes forms for editing information, buttons for submitting changes, and areas for displaying current information.

Interactions: Receives user inputs for editing personal and health information and displays current information fetched from the database. It interacts with the **Profile Management Controller** to submit changes and display confirmation or error messages based on the outcome of update attempts.

Error Message Display:

Responsibilities: Provides feedback to the user in the event of errors encountered during the profile management process, such as validation failures or issues saving changes to the database.

Interactions: Receives error details from the **Profile Management Controller** and displays them to the user, guiding them to correct the issues or retry their actions.

Control Class

Profile Management Controller:

Responsibilities: Manages the logic for retrieving, displaying, editing, and saving the user's profile information. It validates user inputs, updates the **User Profile** entity with new information, and handles errors.

Interactions: Communicates with the **Personal Profile Page** to receive user inputs and send back confirmation of updates or error messages. It interacts with the database to fetch current profile information and to save updates. It also ensures that only authorized changes are made, respecting privacy and data protection regulations.

Data Validation Service:

Responsibilities: Ensures that the user's input meets specified criteria for personal and health information (e.g., format validation, data completeness).

Interactions**:** Invoked by the **Profile Management Controller** during the process of updating the user's profile. Provides validation results back to the controller, which then decides whether to proceed with saving the updates or to instruct the **Personal Profile Page** to display error messages.

Entity Class

User Profile:

Responsibilities: Represents the user's personal and health information stored in the system. It includes attributes such as name, contact information, health metrics (e.g., medical conditions, allergies), and any other personal data relevant to the user's profile.

Interactions: Updated by the **Profile Management Controller** when the user submits changes. It is the core entity that is read from and written to during the profile management process.

A4

Boundary Class

Fitness Goal Management Page:

Responsibilities**:** Acts as the user interface where users can view their current fitness goal, set new goals, modify existing goals, or cancel their goals. This page includes forms for inputting goal details, options for modifying or canceling goals, and buttons for submitting changes.

Interactions**:** Receives user inputs related to fitness goals and displays current goal information fetched from the database. It interacts with the **Fitness Goal Controller** to submit changes and display confirmation or error messages.

Error Message Display:

Responsibilities**:** Provides feedback to the user in the event of errors encountered during the fitness goal management process, such as validation failures or issues saving changes to the database.

Interactions**:** Receives error details from the **Fitness Goal Controller** and displays them to the user, guiding them to correct the issues or retry their actions.

Control Class

Fitness Goal Controller:

Responsibilities**:** Manages the logic for setting, modifying, and canceling fitness goals. It validates user inputs, updates the **Fitness Goal** entity with new information, and handles errors. It also provides guidance or recommendations for setting realistic and achievable fitness goals based on the user's inputs.

Interactions**:** Communicates with the **Fitness Goal Management Page** to receive user inputs and sends back confirmation of updates or error messages. It interacts with the database to fetch current goal information and to save updates. Ensures that only authorized changes are made, respecting privacy and data protection regulations.

Fitness Guidance Service:

Responsibilities**:** Provides recommendations or guidance to users setting new fitness goals, ensuring the goals are realistic and achievable based on user inputs and general fitness principles.

Interactions**:** Invoked by the **Fitness Goal Controller** during the process of setting a new fitness goal or modifying an existing one. Provides feedback to the controller, which then may suggest modifications to the user's proposed goal or offer advice directly through the **Fitness Goal Management Page**.

Entity Class

Fitness Goal:

Responsibilities**:** Represents the user's fitness goal details stored in the system. It includes attributes such as goal type (e.g., weight loss, muscle gain, endurance improvement), target metrics (e.g., target weight, distance, time), and progress tracking.

Interactions**:** Updated by the **Fitness Goal Controller** when the user submits changes. It is the core entity around which the fitness goal management process revolves.

Boundary Class

Workout Tracking Page:

Responsibilities**:** Serves as the user interface for inputting and reviewing workout details, including the type of exercise, duration, intensity, and other relevant parameters. It allows users to confirm their workout details and view their progress towards their fitness goals.

Interactions**:** Receives workout input from the user, interacts with the **Progress Tracking Controller** to submit and retrieve workout progress and goal achievement data, and displays this information to the user. It also displays error messages or incorrect input warnings when necessary.

Error and Feedback Display:

Responsibilities**:** Provides immediate feedback to the user in cases of incorrect input or system errors encountered during the workout tracking process. This includes displaying messages for incorrect input and highlighting fields that require correction.

Interactions**:** Triggered by validation processes in the **Progress Tracking Controller**, it alerts the user to correct inputs or informs them of system errors and possible actions to resolve them.

Control Class

Progress Tracking Controller:

Responsibilities**:** Manages the logic for recording workout sessions, analyzing workout progress in relation to fitness goals, and providing feedback on goal achievement. It validates workout inputs, updates the **Workout Session** entities, retrieves the **Fitness Goal** for comparison, and calculates progress metrics.

Interactions**:** Receives workout details from the **Workout Tracking Page**, interacts with the database to record session details and fetch fitness goals, analyzes progress, and returns feedback or progress updates to be displayed on the **Workout Tracking Page**.

Workout Analysis Tool:

Responsibilities**:** Provides detailed analysis of the user's workout progress over time and in relation to their set fitness goals. It might include calculating averages, trends, and projections based on workout data.

Interactions**:** Invoked by the **Progress Tracking Controller** to process workout data stored in **Workout Session** entities. It provides analytical insights that help the controller compare progress against **Fitness Goals** and generate meaningful feedback for the user.

Entity Class

Workout Session:

Responsibilities**:** Represents the details of an individual workout session input by the user, including exercise type, duration, intensity, and any other relevant workout metrics.

Interactions**:** Created and updated based on user inputs on the **Workout Tracking Page**. It is stored in the database and used for analyzing progress towards fitness goals.

Boundary Class

Calorie Tracking Page:

Responsibilities**:** Provides the interface for users to enter and view details of their daily food intake, including meals and snacks. It displays calculated maintenance calories, total daily intake, and recommended intake for weight management goals.

Interactions**:** Receives food intake details from the user, interacts with the **Calorie Tracking Controller** to submit food details, and displays calculated calorie data and feedback on reaching or exceeding nutritional goals.

Food Item Input Interface:

Responsibilities**:** Allows users to input specific food items and their quantities. This interface may include search functionality to select from a database of food items with pre-calculated calorie values.

Interactions**:** Collects detailed input from the user regarding their food consumption and conveys this information to the **Calorie Tracking Controller** for processing and calorie calculation.

Control Class

Calorie Tracking Controller:

Responsibilities**:** Manages the logic for calculating daily maintenance calories, total calorie intake, and recommended intake for weight management goals based on user input and profile information. It also provides feedback on nutritional goal achievement.

Interactions**:** Retrieves user profile information to calculate maintenance calories, processes user-inputted food items using data from the **Food Database**, calculates total daily calorie intake, and compares this with nutritional goals to provide feedback.

Nutrition Calculator:

Responsibilities**:** Performs detailed calculations to determine maintenance calories, total calorie intake, and recommended calorie intake for weight gain or loss, utilizing algorithms based on nutritional science and user profile data.

Interactions**:** Utilized by the **Calorie Tracking Controller** to apply complex nutritional formulas, ensuring that calorie tracking and recommendations are accurate and personalized to the user's goals and profile.

Entity Class

User Profile:

Responsibilities**:** Stores user-specific data necessary for calorie calculations, such as age, gender, weight, height, and activity level. This entity is crucial for determining maintenance calories and recommended intake levels.

Interactions**:** Read by the **Calorie Tracking Controller** to perform personalized calorie needs calculations based on the user's physical and lifestyle parameters.

Boundary Class

Personalised Workout Plan Page:

Responsibilities**:** Serves as the user interface for accessing and viewing the personalized workout plan. It includes detailed information about exercise routines, sets, reps, rest periods, and progression, as well as functionality for users to track their workouts and progress.

Interactions**:** Displays the personalized workout plan generated by the system based on the user's profile information. Allows users to input their workout performance data, which is then processed by the **Workout Plan Controller**.

Workout Tracking Interface:

Responsibilities**:** Allows users to record their performance for each workout session, including weights lifted, reps completed, and any personal notes. May also include a feature to adjust or customize the workout plan.

Interactions**:** Collects workout performance data from the user and sends it to the **Workout Plan Controller** for processing and feedback.

Control Class

Workout Plan Generator:

Responsibilities**:** Generates a personalized workout plan based on the user's profile information and fitness goals. Adjusts the plan based on user feedback, performance data, and progress towards goals.

Interactions**:** Uses **User Profile** data to create a tailored **Workout Plan**, which is then displayed to the user. It may adjust the plan in response to user progress reported through the **Workout Tracking Interface**.

Workout Plan Controller:

Responsibilities**:** Manages user interactions with their workout plan, including tracking workouts, recording performance, and providing feedback or suggestions for adjustments to the plan.

Interactions**:** Receives workout data from the **Workout Tracking Interface**, processes this data to track progress, and may adjust the user's **Workout Plan** in response to this data. Provides feedback to the user through the **Personalised Workout Plan Page**.

Entity Class

User Profile:

Responsibilities**:** Stores the user's personal and fitness-related information, such as age, gender, weight, height, fitness level, and goals. This data is essential for generating a personalized workout plan.

Interactions**:** Accessed by the **Workout Plan Generator** to tailor the workout plan to the user's individual needs and goals.

Boundary Class

Recipe Book Page:

Responsibilities**:** Serves as the user interface where users can browse, view, and interact with the recommended recipes. It includes functionality for displaying recipe lists, detailed recipe information, nutritional breakdowns, and options for saving favorites or printing recipes.

Interactions**:** Displays recipes recommended by the system based on the user's fitness goals and profile information. Allows users to select recipes to view more details and provides options for user actions such as saving or printing recipes.

Recipe Details Interface:

Responsibilities**:** Provides detailed information about a selected recipe, including ingredients, cooking instructions, nutritional information, and additional cooking tips or notes.

Interactions**:** Activated when a user selects a recipe from the **Recipe Book Page**, it retrieves and displays all relevant details for the recipe, facilitating an informative and user-friendly experience.

Control Class

Recipe Recommendation Engine:

Responsibilities**:** Analyzes the user's profile information and fitness goals to generate a list of recommended recipes that align with the user's dietary preferences and nutritional requirements.

Interactions**:** Uses **User Profile** data to select appropriate recipes from the **Recipe** entity, then provides these recommendations for display on the **Recipe Book Page**. It may also adjust recommendations based on user feedback or changes in user profile information.

Recipe Management Controller:

Responsibilities**:** Manages user interactions within the recipe book feature, including tracking user selections, processing requests for more detailed recipe information, and handling user actions such as saving favorites or printing recipes.

Interactions**:** Coordinates between the **Recipe Book Page**, **Recipe Details Interface**, and **Recipe Recommendation Engine** to ensure a seamless user experience, from browsing recommended recipes to accessing detailed cooking instructions.

Entity Class

User Profile:

Responsibilities**:** Stores the user's personal and fitness-related information, such as age, gender, weight, height, fitness level, dietary preferences, and fitness goals. This information is crucial for tailoring recipe recommendations.

Interactions**:** Accessed by the **Recipe Recommendation Engine** to determine the types of recipes that would best align with the user's dietary needs and fitness goals.

Recipe:

Responsibilities**:** Represents the data structure for recipes in the recipe book, including details like recipe name, ingredients list, cooking instructions, nutritional information, and any other relevant metadata.

Interactions**:** Managed and retrieved by the **Recipe Recommendation Engine** to be displayed to the user based on their profile information and fitness goals.

Boundary Class

Gym Location Page:

Responsibilities**:** Acts as the interface where users can search for and view nearby gym locations. It includes functionalities for displaying a list of gyms, detailed information about each gym, and options for users to enter or update their location information.

Interactions**:** Presents search results to the user based on their location and selected filters. Allows users to select a gym to view more details and possibly save it to a favourites list for easy access in the future.

Location Input Interface:

Responsibilities**:** Provides a mechanism for users to input or update their current location if the system cannot automatically retrieve it or if the user wishes to search for gyms in a different area.

Interactions**:** Collects location data from the user and sends it to the **Gym Location Finder** for processing and retrieving relevant gym information.

Control Class

Gym Location Finder:

Responsibilities**:** Manages the logic for querying the database for gym locations based on the user's location and preferences. It also processes user inputs for location changes and filters gym results accordingly.

Interactions**:** Retrieves user location from the **Location Input Interface**, uses this information to find nearby gyms from the **Gym Information** entity, and returns a list of suitable gyms to the **Gym Location Page**.

Real-time Capacity Checker:

Responsibilities**:** Provides up-to-date information on the current capacity of gyms to ensure users have access to real-time data, helping them make decisions based on gym occupancy levels.

Interactions**:** Continuously updates gym capacity information in the **Gym Information** entity, which is then displayed to users on the **Gym Location Page** when viewing gym details.

Entity Class

User Profile:

Responsibilities**:** Stores user-specific information, which may include preferences for gym facilities, equipment, or other criteria important for filtering gym search results.

Interactions**:** Accessed by the **Gym Location Finder** to tailor search results based on the user's preferences and location.

Gym Information:

Responsibilities**:** Represents detailed information about each gym, including name, address, operating hours, current capacity, amenities, available equipment, and special offers.

Interactions**:** Managed by the system to provide users with comprehensive details about gym locations, facilitating an informed decision-making process.