

Object Oriented Analysis and Design

Semester Project

Title: Fitness Tracking App

Team Members: Prateek Kumar, Rohan Eswara

Work Done:

During the first week, we focused on setting up the basic structure of our fitness tracking application using Python and Flask. The following tasks were accomplished during this period:

1. **Environment Setup**: We started by setting up a virtual environment for our project using *venv*. This ensured that our dependencies were isolated and that we could easily manage different versions of packages. Next, we installed the required packages such as Flask, Flask-Restful, and Boto3 using pip.
2. **Database Setup**: We chose Amazon DynamoDB as our database solution because of its scalability and ease of use. We created a table in DynamoDB to store user information such as name, email, and password. We also set up an S3 bucket to store user profile pictures.
3. **API Design**: We designed the API endpoints for our application using Flask-Restful. We decided to use a RESTful API approach to keep our application simple and easy to understand. The API endpoints we designed included user registration, user login, getting user profile information, adding new exercises, getting exercise history, and setting exercise goals.
4. **User Authentication**: The user authentication module is responsible for verifying user credentials and starting a session for the user. For simplicity, we will implement a password-based authentication system, where the user provides a username and password to log in. We will store the user's password securely using a one-way hashing algorithm..

5. **User Registration:** We implemented user registration functionality by creating a new user record in DynamoDB. We ensured that the user's email address was unique to prevent duplicate user accounts.
6. **User Profile:** We implemented functionality to allow users to view and update their profile information. This included their name, email address, and profile picture. We used the S3 bucket we set up earlier to store the user's profile picture.
7. **Adding Workout:** We implemented functionality to allow users to add new exercises they have performed. This included the type of exercise, duration, and calories burned. We stored this information in DynamoDB.
8. **Getting Workout History:** We implemented functionality to allow users to view their exercise history. This included a list of all the exercises they have performed along with the date and time of each exercise.
9. **Testing:** We performed unit testing on all the implemented functionalities to ensure that they were working as expected. We used Flask's built-in test client to test the API endpoints.

In conclusion, during the first week of development, we set up the basic structure of our fitness tracking application using Python and Flask. We designed the API endpoints, set up the database and storage, implemented user authentication, and implemented functionality for user registration, user profile management, adding exercises, getting exercise history, and setting exercise goals. We also performed unit testing on all the implemented functionalities to ensure that they were working as expected. By the end of the first week, we had a solid foundation to build upon for the next phase of development.

Changes or Issues Encountered:

During the implementation of the first week's work, we encountered several issues. Firstly, setting up the Flask application to handle user authentication and session management proved to be challenging as it required integrating different libraries and frameworks. Secondly, configuring the AWS credentials and ensuring that the Flask application could connect to the DynamoDB database and S3 bucket also posed some difficulties. Additionally, testing the application thoroughly and debugging any errors that arose during the development phase required a considerable amount of time and effort. Despite these challenges, we were able to successfully implement the required features and ensure that the application met the necessary requirements.

Patterns:

The factory pattern and strategy pattern are both design patterns that can be useful in building the fitness tracking application. The factory pattern is used to create instances of classes without exposing the instantiation logic to the client. This helps to decouple the code and make it easier to maintain and update. In our fitness tracking application, we have used the factory pattern to create instances of classes for the different menus in the application such as the BodyMeasurements menu and the WorkoutGoals menu, without exposing the creation logic to other parts of the code. This makes it easier to add new classes in the future if needed. In our case, we will need to integrate the creation of classes for the menus Nutrition and Fitness Goals.

The strategy pattern is used to define a family of algorithms, encapsulate each one, and make them interchangeable. This can be useful in the fitness tracking application when it comes to setting the intensity of the user's workouts. In particular, based on the intensity selected by the user, the sets and repetitions are recalibrated for each exercise. We can define a family of algorithms for calculating the workout metrics (sets and repetitions for each exercise) based on the intensity chosen by the user, and encapsulate each one in a separate class. This allows users to choose the algorithm that best suits their needs, and makes it easier to add new algorithms in the future if needed. Similarly, we can define a family of algorithms for tracking workouts, such as by

heart rate or calories burned, and encapsulate each one in a separate class. This makes it easier to add new tracking algorithms in the future, and allows users to choose the one that best suits their needs.

Plan for Next Iteration:

During the second week, we will be focusing on implementing the core functionality of the fitness tracking application. The primary tasks for this week will include:

1. **Implementing the backend functionality for workout tracking:** We will need to create the backend logic that will allow users to create and track their workouts. This will involve creating endpoints for creating, updating, and deleting workouts, as well as endpoints for retrieving workout data for display on the front-end.
2. Next week, we will implement the nutrition tracking feature in our fitness application. We will start by creating a new module in our Flask application to handle the nutrition tracking functionality. We will integrate this API into our application to allow users to search for food items and track their daily intake. We will also create a new table in our DynamoDB database to store the nutrition information for each user's daily intake. Users will be able to add food items to their daily intake and the corresponding nutrition information will be stored in the database. We will also create a new endpoint in our API to allow users to retrieve their daily nutrition intake information. Finally, we will create a simple UI using HTML, and JavaScript to allow users to interact with the nutrition tracking functionality.
3. We will also begin working on the fitness tracking feature. We will add a new page to the web application where users can log their fitness activities. We will use Flask to create a route for the new page and create a new template using HTML for the front-end design. The user will be able to select the type of activity, such as running, or weightlifting, and input details such as duration, distance, and weight lifted. The data entered by the user will be sent to the backend server through a REST API call. We will create a new endpoint for the API call using

Flask and define a schema for the data to be received. The received data will be stored in the DynamoDB table we created in the first week. We will also add validation to the input fields to ensure that the data entered is in the correct format.

4. **Testing and debugging:** We will need to thoroughly test the application to ensure that it is functioning as expected and fix any issues that are identified during testing.

Overall, the goal of the second week of work will be to implement the core functionality of the fitness tracking application, including the ability to create and track workouts and manage user profiles. We will need to ensure that the application is user-friendly, reliable, and efficient.

