Sample Assignment Document

This document details about the description of the application, design of the entities, REST  
services, endpoints, URI, HTTP methods,

\*

Group Members

1.

2.

3.

4.

Division of Work

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Marks** | **Responsibility** | **Tasks done** |
| Design: | 10% | All Group member |  |
| Backend: | 10% | Group Member 1 and 2 | Include <name> and task done |
| Frontend: | 10% | Group Member 3 and 4 | Include <name> and task done |
| Integration: | 10% | All Group members. |  |

# **Introduction**

Welcome to our fitness management system, designed to empower you in achieving your health and wellness goals effectively. Our platform provides a comprehensive suite of tools and features tailored to streamline your fitness journey, ensuring convenience and efficiency every step of the way.

From tracking your workouts to monitoring your progress, setting personalized goals, and managing your nutrition, our platform offers a seamless and intuitive user experience. Whether you're a seasoned fitness enthusiast or just beginning your fitness journey, our system is designed to cater to your needs and support you in reaching your aspirations.

**Key Features:**

Fitness Tracker application has the following components.

* **Entities**
  + Users
  + Workout Plans
  + Nutrition Items
  + Wearable Devices
* **Features**
  + Registration: Users can register by providing their name to generate a unique ID for future access.
  + GetID: Retrieve forgotten IDs by providing registered names.
  + Activity Logging: Log exercise types, notes (optional), sets, and more for comprehensive workout tracking.
  + Goal Setting: Set personalized fitness goals to track progress and achievements.
  + Progress Tracking: Monitor total weight lifted and other metrics to assess fitness journey.
  + Workout Plans: Access and manage personalized workout plans using unique UserIDs.
  + Wearable Integration: Connect with wearable devices for enhanced fitness tracking.
  + Nutrition Management: Add and track nutrition items (calories, protein, carbohydrates, fats) to support fitness goals.
  + View Nutrition: Monitor nutrition intake for informed dietary choices.

# **User Stories and Acceptance criteria**

## **EPIC 1: Registration and GetID**

* 1. **Registration:**

**User Story:** As a new user, I should be able to register as a customer to get ID.

**Acceptance Criteria:**

* Input data fields to enter - 1) Username and It will give you ID and that ID should be remembered.
* Usernames should be unique.
  1. **GetID:**

**User Story:** As a user, I should be able to add actitvites using ID. If I forgot the ID?. I can check here by giving my user name.

**Acceptance Criteria:**

* Input data fields to enter - 1) Username
* Indicate invalid usernames alerts to the user.
* After successful validation of all entered fields and on clicking Submit, show the respective ID to the user.

**Assumptions for EPIC 1[User Stories 1.1 and 1.2]**

Only the user interfaces and client side validation need to be implemented.

It can be assumed that the users are successfully created/logged in when the entered values are valid or Mock APIs can be used.

Any other assumptions made can be listed in the documentation

## **EPIC 2: Customer Activity Logging**

**User Story 2.1:** As a customer, I should be presented with a dashboard page.

**Description:** The dashboard page should contain navigation menu options to add Activity Logging, Progress Tracking, View Workout plans, Add Exercise types, Add Nutrition item and view nutrition items.

**Acceptance Criteria:**

If there are no Activities for the given userID, it should display a message “no activites”.

The status of Logged Activities with details to be displayed for the userID.

**User Story 2.2:** As a customer, I should be able to view the Logged Activities. Navigate to Work out plans page.

**Description:** On the user entering the userID, the list of logged activites should be displayed.

**Acceptance Criteria:**

The Logged Activities details to be displayed.

**User Story 2.3:** As a customer, I should be able to view the Progress. Navigate to Progress Tracking page.

**Description:** On the user entering the userID, the progress should be displayed.

**Acceptance Criteria:**

The Total Weight Lifted to be displayed.

# **Design and Architecture, System operations, and services identified.**

The Design and Architecture of the fitness tracking system is as follows:

1. **Frontend:**
   * The frontend of the application will be responsible for providing a user-friendly interface to users.
   * The frontend will communicate with the backend server through RESTful APIs to perform actions like searching users Logged activites, Users IDs, etc.
   * The UI should be responsive, allowing users to access the application from both desktop and mobile devices.
2. **Backend:**
   * The backend will handle all the business logic, process requests from the frontend, interact with the database, and coordinate with external services.
   * It can be designed using a ExpressJS which is Javascript framework for backend.
   * The backend will expose RESTful APIs to be consumed by the frontend and any other potential clients.
3. **Database:**
   * The application's data will be stored in a reliable and scalable database system.
   * For structured data, a relational database like PostgreSQL can be used to store information about customers, restaurants, menus, orders, etc.
4. **Menu Management and Activity Logging:**
   * Implement an efficient navbar functionality that allows users to browse options like Activity Logging, Adding Exercise, Workout plans, Adding Nutrition Items, Viewing Nutrition items.
   * Provide an easy-to-use interface for users to manage their activity details.
   * Users should be able to add activities to their account, review their activities.
5. **Progress Tracking and Work Out Plans:**
   * Implement a progress tracking system that displays the list of the Activities logged by the users.
   * It can tracked by using user ID in Progress Tracking and Workout plans.
6. **ARCHITECTURE:**

* Architecture Diagram

1. **REST API endpoints with URI and HTTP methods**

( Swagger Documentation can also be used)

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial No** | **Entity** | **Http Method** | **URI** |
| 1 | Users | GET | **/users/${id}** |
| POST | **/users** |
| PUT | **/users/${id}** |
| DELETE | **/users/${id}** |
| 2 | Workout Plans | GET | **/exercises/${id}** |
| POST | **/** **exercises** |
| PUT | **/exercises/${id}** |
| DELETE | **/exercises/${id}** |
| 3 | Nutrition Items | GET | **/nutrition** |
| POST | **/nutrition/${userId}** |

**Representation design in request and response**

Sample Request and Response header for the above-mentioned HTTP method:

1. **GET /customers**

* Request:
  + **Method: GET**
  + **URI: /users/${id}**
* Response:
  + **Status Code: 200 OK**
  + **Headers: Content-Type: application/json**
  + **Body:**

**{**

**"name":"chandu"**

**}**

1. **GET /customers/{customer\_id}**

* Request:
  + **Method: GET**
  + **URI: /customers/customer\_123**
* Response:
  + **Status Code: 200 OK**
  + **Headers: Content-Type: application/json**
  + **Body:**

**{**

**"customer\_id": "customer\_123",**

**"name": "John Doe",**

**"email": "john.doe@example.com",**

**"phone": "+1234567890",**

**"address": "123 Main Street, City, Country"**

1. **Frontend UI Design**
2. **Execution Instructions and Assumptions Made (If any)**
3. **GITHUB Repositories**

Frontend:

Backend:

1. **Demonstration Video**