CM2601 Object Oriented Development

Case Study: "Personalized Fitness Tracker with AI Recommendation System"

Aim:

This exercise emphasizes the use of **OOP concepts**, **solution design using UML**, **exception handling**, **database connection**, and **multithreading**.

Task:

Develop a **Java-based Fitness Tracker** application that records user activities, stores the data in a database, and provides personalized fitness recommendations using a simple Machine Learning model.

Solution Features:

1. **User Registration and Login**:
    - ✓ Allow users to register with their details (name, age, weight, height).
    - ✓ Securely store user data in a database.
2. **Activity Logging**:
    - ✓ Users can log daily activities (e.g., steps, distance, calories burned).
    - ✓ Data is stored in the database.
3. **AI-Based Recommendation**:
    - ✓ Implement a basic ML model (e.g., Linear Regression or KNN).
    - ✓ Predict the number of steps a user should aim for based on their historical data.
4. **Data Visualization**:
    - ✓ Display weekly activity summaries (e.g., total steps, calories burned).
5. **Multithreading**:
    - ✓ Use threads to handle simultaneous user operations (e.g., activity logging while retrieving recommendations).
6. **Exception Handling**:
    - ✓ Handle errors gracefully (e.g., database connection errors, invalid input, or ML model failures).

Expectations:

1. OOP Design:

- Design Use cases (with descriptions), Activity, Class and Sequence diagrams.
- Classes may include User, Activity, FitnessTrackerApp, DBManager, RecommendationEngine.
- Use inheritance (e.g., specialized activities like Running, Walking).
- Identify abstract classes/ methods where necessary in the design and apply appropriate types of polymorphism.
- Apply composition for classes (e.g. Activity is part of the User).
- Use interfaces for ML-related functionality.

2. Database Connection:

- Use **JDBC** to connect to MySQL database (or you may use any other suitable DB connection).
- Tables: Users, Activities.

3. Simple ML Component:

- Train a basic model using stored data (e.g., predicting the steps goal for the next week).
- Serialize the trained model for reusability (additional task).

4. Multithreading:

- Implement threads for:
    - Activity logging and database saving.
    - Asynchronous recommendation generation (additional task).
    - Data visualization updates (additional task).

5. Version Controlling:

- Use of GitHub to maintain application/solution versions and proper logging of all activities through it (additional task).