Fitness Tracker Application Project Report

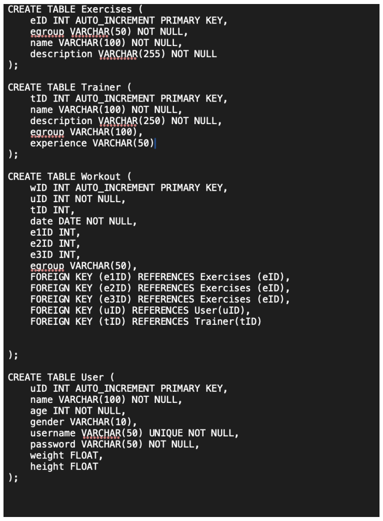
1. Introduction

The Fitness Tracker Application is designed to help users manage their fitness routines by tracking exercises, users, trainers, and workouts. This application provides a graphical user interface (GUI) developed using JavaFX, with data persistence managed through a MySQL database. Initially, the application was built as a console application to facilitate early testing, allowing for refinement of core functionalities before integrating the GUI.

2. Application Overview

2.1 Features

- **User Authentication**: Users can sign up, log in, and log out securely.
- **Manage Exercises**: Users can view and sort for certain exercises
- **Manage Users**: Users can view their account information and create new users
- **Manage Trainers**: Users can view trainer profiles and sort for certain trainers
- **Manage Workouts**: Users can add, view and sort their tracked workouts accordingly
- **Generate Random Workouts**: Users can automatically generate a planned 3-exercise workout in which the user can change what they want



```sql
CREATE TABLE Exercises (
    eID INT AUTO_INCREMENT PRIMARY KEY,
    egroup VARCHAR(50) NOT NULL,
    name VARCHAR(100) NOT NULL,
    description VARCHAR(255) NOT NULL
);

CREATE TABLE Trainer (
    tID INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description VARCHAR(250) NOT NULL,
    egroup VARCHAR(100),
    experience VARCHAR(50)
);

CREATE TABLE Workout (
    wID INT AUTO_INCREMENT PRIMARY KEY,
    uID INT NOT NULL,
    tID INT,
    date DATE NOT NULL,
    e1ID INT,
    e2ID INT,
    e3ID INT,
    egroup VARCHAR(50),
    FOREIGN KEY (e1ID) REFERENCES Exercises (eID),
    FOREIGN KEY (e2ID) REFERENCES Exercises (eID),
    FOREIGN KEY (e3ID) REFERENCES Exercises (eID),
    FOREIGN KEY (uID) REFERENCES User(uID),
    FOREIGN KEY (tID) REFERENCES Trainer(tID)


);

CREATE TABLE User (
    uID INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    age INT NOT NULL,
    gender VARCHAR(10),
    username VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(50) NOT NULL,
    weight FLOAT,
    height FLOAT
);
```

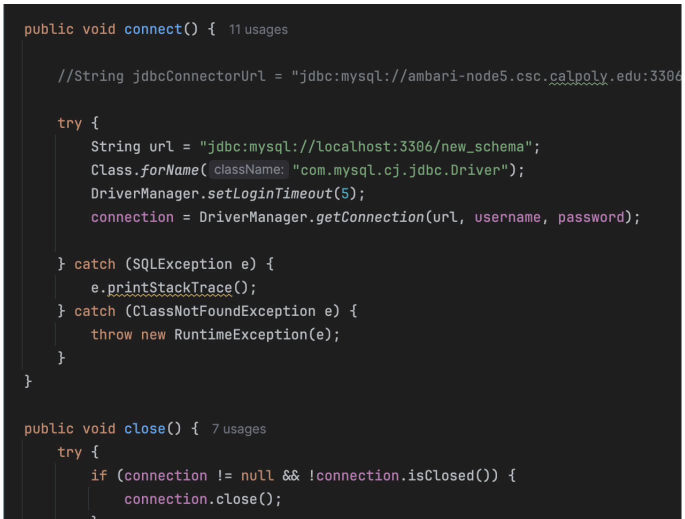
2.2 Technology Stack

- **Programming Language**: Java
- **GUI Framework**: JavaFX
- **Database**: MySQL
- **Build Tool**: Maven

3. System Design

3.1 Class Structure

- **Account**: Manages user account information and authentication.



```java
public void connect() {  11 usages

    //String jdbcConnectorUrl = "jdbc:mysql://ambari-node5.csc.calpoly.edu:3306

    try {
        String url = "jdbc:mysql://localhost:3306/new_schema";
        Class.forName( className: "com.mysql.cj.jdbc.Driver");
        DriverManager.setLoginTimeout(5);
        connection = DriverManager.getConnection(url, username, password);

    } catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        throw new RuntimeException(e);
    }
}

public void close() {  7 usages
    try {
        if (connection != null && !connection.isClosed()) {
            connection.close();
```

- **DatabaseConnector**: Manages the connection to the MySQL database.
- **Exercise**: Represents an exercise with attributes like ID, group, name, and description.
- **Login**: Handles the login functionality.
- **Main**: The main application class that launches the JavaFX application.
- **NewUser**: Handles new user registration.
- **Trainer**: Represents a trainer with attributes like ID, name, specialty, and experience.
- **Workout**: Represents a workout with attributes like ID, user ID, trainer ID, date, and list of exercise IDs.

3.2 FXML Files and Their Controllers

- **afterLogin.fxml**: Defines the layout for the user's home page after logging in.
- **hello-view.fxml**: Defines the layout for the application's main login view.
- **NewUser.fxml**: Defines the layout for the new user registration view.
- **Trainer.fxml**: Defines the layout for viewing trainer information.
- **Workout.fxml**: Defines the layout for managing workouts.
- **Account.fxml**: Defines the layout for user account information.
- **Exercises.fxml**: Defines the layout for managing exercises.

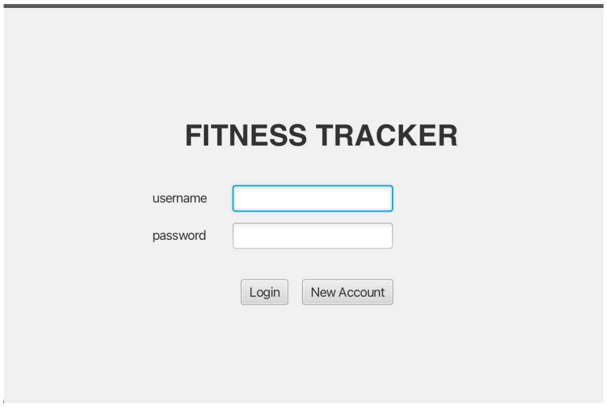
3.3 Key Components

- **Account.java**: Handles user account operations including authentication.
- **DatabaseConnector.java**: Manages the database connection and CRUD operations.
- **Exercise.java**: Represents the exercise entity and its attributes.
- **Login.java**: Handles the login logic and authentication.
- **Main.java**: The entry point of the application, initializing the JavaFX application.
- **NewUser.java**: Handles the registration of new users.
- **Trainer.java**: Represents the trainer entity and its attributes.
- **Workout.java**: Represents the workout entity and manages workout-related operations.

4. User Interface

4.1 Main Application Window

The main application window features a menu bar with options to exercises, account, trainers, and workouts. It also offers the option to logout on all windows, which successfully logs out the user.



4.2 Exercise Management, Trainer

The exercise and trainer management interface allows users to view and sort exercises or trainers based on a certain category. The exercise table has a short name and a short length description about the exercise. The trainer table displays the name, description, and the experience of the trainer.

4.3 User

The user can create new profiles , log in, and view their information once they log in. The application also integrates password and username authentication. Additionally, users can log out securely. New users are able to log in which will display that specific user's information privately.

4.4 Workout Management

Users can generate three random exercises according to a certain group of exercises. Additionally, they can change the Trainer in order to create an optimal workout. The user also inputs the group of workout and date of the specific workout.

5. Implementation Details

5.1 Connecting to MySQL

The application uses the DatabaseConnector class to establish and manage the connection to the MySQL database. SQL commands are executed to perform CRUD operations on the database tables.

5.2 Integrating JavaFX

JavaFX is used to create simple, effective, and interactive user interfaces. The main application window and various management interfaces are defined using FXML files, with corresponding controllers to handle user interactions.

6. Testing

Manual testing was used for ensuring the application's functionalities. Test cases are written for each manager class to ensure that operations are correctly performed. Tests also cover the generation of random workouts.

7. Conclusion

The Fitness Tracker Application provides a comprehensive solution for managing fitness-related data. By combining Java, JavaFX, and MySQL, the application offers a robust and user-friendly interface for tracking and organizing workouts. The initial console-based implementation allowed for thorough testing and validation of core functionalities, ensuring a stable foundation for the GUI integration.