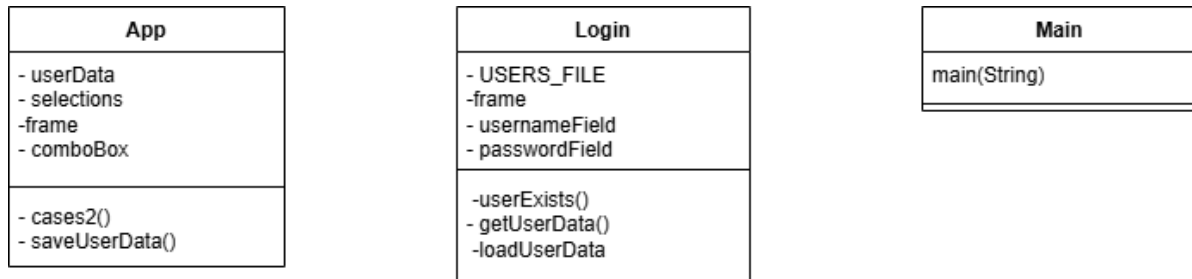


## 1. Class Diagram



## 2. Sequence Diagrams

- **User Registration**

**Steps:**

1. The user opens the Login GUI and enters a username and password.
2. The system checks if the username already exists.
3. If the username is new, the Login class opens the App GUI for user input.
4. The user enters their fitness data, which is saved to users.txt.

**Sequence Diagram:**

**User -> Login : Opens Login GUI**

**User -> Login : Enters Username & Password**

**Login -> File : Checks if user exists**

**alt User exists**

**Login -> User : Shows "User already exists" message**

**else**

**Login -> App : Opens App GUI for registration**

**User -> App : Enters fitness data**

**App -> File : Saves user data to users.txt**

- **User Login**

**Steps:**

1. The user opens the Login GUI and enters credentials.
2. The Login class verifies the credentials using users.txt.
3. If the credentials are valid, the App GUI is launched, pre-filled with user data.
4. The user can modify or view their fitness recommendations.

**Sequence Diagram :**

**User -> Login : Opens Login GUI**

**User -> Login : Enters Username & Password**

**Login -> File : Checks credentials in users.txt**

**If Credentials valid**

**Login -> App : Launch App GUI with user data**

**else**

**Login -> User : Shows "Invalid username or password" message**

**Sequence Diagram for cases2() Function**

**This function determines the recommendation case based on user data.**

**Steps:**

1. The user enters data in the App GUI.
2. The App class calls cases2() to evaluate the user data.
3. The function checks the gender, weight, and goals and maps them to a specific case.
4. The result is returned to generate a recommendation.

**User -> App : Enters fitness data**

**App -> cases2 : Evaluates gender, weight, goals**

**cases2 -> App : Returns case number**

**App -> Recommendation : Fetches corresponding recommendation**

### 3. Description of Unit Tests

- **testAppLaunch()**

Test if the app launch properly .

- **testSaveUserData()**

**Purpose:** Verify that user registration data is correctly saved to the users.txt file.

**Method:**

Call saveUserData() with a mock username, password, and fitness data.

Check if the data is written correctly in the file.

**Expected Result:** The users.txt file should contain the correct user data.

- **testValidateRecommendations()**

To verify that the `cases2()` method correctly evaluates user data and returns the appropriate recommendation case.

**Setup:**

Creates an instance of the App class.

Mocks user data (userData) and user selections (selections) with specific values:

Name: John , Age: 28 , Weight: 85 kg , Height: 180 cm ,Gender: Male ,Goal: Losing weight and Activity Level: Active

**Execution:**

Calls the cases2() method, which determines the recommendation case based on the mock data.

**Validation:**

Asserts that the returned case number matches the expected value (case 9 in this example).

