

## COMP 3005 - Project Version 2 - Project Plan

Language: Java

3 different versions of the health and fitness application

- Member
- Trainer
- Administrative staff

Could be split into 3 programs

These functions will probably be split into several smaller functions

Member Functions

- User Registration
  - Create profile: (**Username, First Name, Last Name, Birthday, password, Profile\_Type**)
  - If the user is already registered, ask the user if they want to delete their old account and make a brand new one before proceeding.
- Profile Management
  - Change profile information (**First Name, Last Name, Password, Payment Options**) (with permission of old password maybe)
  - Update personal information (**Height, Weight, Diet Plan**)
  - Update fitness goals (**Goal weight, time to get to goal weight**)
- Dashboard Display
  - Display Exercise routines: (**Exercise Routines**)
  - Display Fitness Achievements: (**Weight Loss, Max Speed, Health Goals Achieved**)
  - Display health metrics (**BMI, Blood Pressure, Heart Rate, Cholesterol, Blood Sugar**)
- Schedule Management
  - Schedule personal/group training session
    - User chooses time
    - App displays list of available trainers
    - User choose trainer

Trainer Functions

- Schedule Management
  - Choose times of availability in the week
- Member Profile Viewing
  - Displays list of members
  - Search list of members by name (First or last)

Administrative Staff Functions

- Room Booking Management

- Update room information (**Room Status, Booking Begin, Booking End, Equipment Available**)
- Equipment Maintenance
  - Update equipment information (**Equipment Status, Type of Equipment**)
- Class Schedule Updating
  - Change the date of classes
- Billing and Payment Processing
  - Manage Bills (**Paid, Payee, date issued**)

### Legend:

**Bold:** Primary Key

**Red:** Foreign Key

**Blue:** Relations made for multi-value Variables

### Possible Relations:

Profiles: (**Username**, First\_Name, Last\_Name, Password, user\_type) DONE

Members: (**Username**, Credit\_Card, birthday, Height, Weight, Diet\_Plan, Goal\_Weight, Goal\_Speed, Goal\_Lift, Weight\_Deadline, Speed\_Deadline, Lift\_Deadline, BMI, systolic\_bp, diastolic\_bp, Heart\_Rate, Blood\_Sugar) DONE

**Exercise\_Routines:** (**Members**, Exercise) DONE

Weight\_Statistics(**Username**, Weight) DONE

Speed\_Statistics(**Username**, Speed) DONE

Lift\_Statistics(**Username**, Lift) DONE

Classes: (**Class\_ID**, **Trainer\_ID**, room\_id, Class\_Type, Class\_Description, Class\_Date, Time\_Of\_Day) DONE

**Dates\_Trainer\_Available:** (Schedule\_ID, **Trainer\_ID**, start\_trainer\_date, start\_time\_of\_day, end\_trainer\_date, end\_time\_of\_day) DONE

**Dates\_Trainer\_Unavailable:** (**Trainer\_ID**, Trainer\_Date, Time\_Of\_Day) DONE

**Class\_Members:** (**Class\_ID**, **Username**) DONE

Bookings: (Booking\_ID, Room\_ID, Class\_Date, Time\_Of\_Day, Event\_Info) DONE

**Equipment:** (**Equipment\_id**, Room\_ID, Equipment\_Status) DONE

Bills: (**Bill\_ID**, **Username**, Price, Date\_Issued) DONE

## Program:

1. **Main()** - start-up
  - Menu function that can take you to:
    - userRegistration()**
    - userLogin()**
    - userLogout()**
  - **Member Menu:**
    - profileInformation()**
    - memberScheduleManagement()**
  - **Trainer Menu:**
    - trainerScheduleManagement()**
    - viewMemberProfile()**
  - **Admin Menu:**
    - roomBookingManagement()**
    - manageEquipment()**
    - manageClassSchedule()**
    - viewBills()**
2. **userRegistration()** - (NONE)
  - Prompt user for username, password, first\_name, last\_name (if username already taken, try again)
  - Set user\_type to TYPE\_MEMBER
  - Set bmi, blood\_pressure, heart\_rate, cholesterol, blood\_sugar to random variables
3. **userLogin()** - (NONE)
  - Prompts user for username and password
  - If username and password non-existent, try again
4. **userLogout()** - (**MEMBER**, **TRAINER**, **ADMIN**)
  - Asks user if they're sure to log out
  - If yes, go back to **main()** as user\_type NONE
  - If no, go back to **main()** as user\_type **MEMBER**, **TRAINER**, or **ADMIN**
5. **profileInformation()** - (**MEMBER**)
  - Menu function that can take you to:
    - personallInformation()**
    - fitnessGoals()**

**healthMetrics()**  
**healthActions()**  
**healthStatistics()**

6. **personalInformation()** - (MEMBER)
  - Displays user's first\_name, last\_name, credit\_card, birthday, height, weight  
If No information found, go to **setPersonalInformation()**
  - Asks if the user wants to update information.  
If yes, go to **setPersonalInformation()**  
If no, go back to **profileInformation()**
7. **setPersonalInformation()** - (MEMBER)
  - Prompt user for first\_name, last\_name, credit\_card, birthday, height, weight
  - After confirming changes, go to **personalInformation()**
8. **fitnessGoals()** - (MEMBER)
  - Display's user's diet\_plan, goal\_weight, goal\_speed, goal\_lift, weight\_deadline, speed\_deadline, lift\_deadline
  - Asks if the user wants to update information.  
If yes, go to **setFitnessGoals()**  
If no, go back to **profileInformation()**
9. **setFitnessGoals()** - (MEMBER)
  - Ask user if they want to set values for diet\_plan, goal\_weight, goal\_speed, goal\_lift, weight\_deadline, speed\_deadline, lift\_deadline
    - Note: These values don't have to be filled out, so null values are allowed
  - If goal\_weight set, then set weight\_loss to random value
  - If goal\_speed set, then set max\_speed to random value
  - If goal\_lift set, then set max\_lift to random value
  - Grabs max value from one of either weight\_loss table,
  - After confirming changes, go to **fitnessGoals()**
10. **healthMetrics()** - (MEMBER)
  - Displays user's bmi, blood\_pressure, heart\_rate, blood\_sugar
11. **healthActions()** - (MEMBER)
  - Displays user's diet plan and exercises
  - Asks if the user wants to update information.  
If yes, go to **setHealthActions()**  
If no, go back to **profileInformation()**
12. **setHealthActions()** - (MEMBER)
  - Menu function that can take you to:  
**newRoutine()**

### **changeDietPlan()**

- Prompts user for diet plan and exercises
- After confirming changes, go to **healthActions()**

### 13. **newRoutine()** - (MEMBER)

- Prompts user for exercise
- Asks if the user wants to add another exercise
- If yes, go to **newRoutine()**
- If no, go to **healthActions()**

### 14. **changeDietPlan()** - (MEMBER)

- Prompts user for diet plan
- After confirming changes, go to **healthActions()**

### 15. **healthStatistics()** - (MEMBER)

- Displays user health statistics based off of aggregate queries on the Health\_Statistics table
- Truly random
- **So random statistics or proper statistics on random data? (IMPORTANT)**
- **Current implementation has max, min, and average on all data (lift, speed, weight) for any given member (based on who's logged in)**

### 16. **memberScheduleManagement()** - (MEMBER)

- Menu function that can take you to:  
  **personalTraining()**  
  **groupTraining()**

### 17. **personalTraining()** - (MEMBER)

- Prompts user for date and time\_of\_day for session
- Displays trainers available at that time of day
- User selects a trainer for the session
- Entry added to the Dates\_Trainer\_Unavailable
- Bill is added to the user for the personal session
- Asks user if they want to schedule another personal session
- If yes, repeat function
- If no, go back to **memberScheduleManagement()**

### 18. **groupTraining()** - (MEMBER)

- Displays classes available
- User selects a class they wish to participate in
- Bill is added to the user for the group session
- Member added to the Class\_Members table

- Asks user if they want to schedule another group session  
If yes, repeat function  
If no, go back to **memberScheduleManagement()**
19. **trainerScheduleManagement()** - (TRAINER)
    - Menu function that can take you to:  
**addSchedule()**  
**viewSchedule()**  
**changeSchedule()**
  20. **addSchedule()** - (TRAINER)
    - Asks user to enter start\_trainer\_date, end\_trainer\_date, start\_time\_of\_day, end\_time\_of\_day
      - Returns error if schedule conflicts with trainer's existing schedules
  21. **viewSchedule()** - (TRAINER)
    - Display's trainer's schedule\_id, start\_trainer\_date, end\_trainer\_date, start\_time\_of\_day, end\_time\_of\_day
  22. **changeSchedule()** - (TRAINER)
    - Prompts user for trainer\_id and schedule\_id of schedule to be changed
    - Menu function that can take you to:  
**changeStartDate()**  
**changeEndDate()**  
**changeStartTime()**  
**changeEndTime()**
  23. **changeStartDate()** - (TRAINER)
    - Asks user for new start date of schedule
  24. **changeEndDate()** - (TRAINER)
    - Asks user for new end date of schedule
  25. **changeStartTime()** - (TRAINER)
    - Asks user for new start time of schedule
  26. **changeEndTime()** - (TRAINER)
    - Asks user for new end time of schedule
  27. **viewMemberProfile()** - (TRAINER)
    - Asks for username of member and displays member profile
  28. **roomBookingManagement()** - (ADMIN)
    - Menu function that can take you to:  
**bookRoom()**  
**viewRoomBooking()**  
**cancelRoomBooking()**
  29. **bookRoom()** - (ADMIN)
    - Asks for Room ID of the room to be booked
      - Returns error if booking already exists for that room
    - Asks for Date, Time\_of\_day, and Event taking place in the room

- 30. `viewRoomBookings()` - (ADMIN)
- 31. `cancelRoomBooking()` - (ADMIN)
  - Asks for Room Id of the room booking to be canceled
  - Returns error if the booking does not exist for that room
  - NOTE: introduce `booking_id` to the bookings table for easier cancellation
- 32. `manageEquipment()` - (ADMIN)
  - Menu function that takes you
    - `viewEquipments()`**
    - `updateEquipmentStatus()`**
- 33. `viewEquipments()` - (ADMIN)
  - Displays all equipments, the room they're in, and their statuses
- 34. `updateEquipmentStatus()` - (ADMIN)
  - Asks for the `Equipment_id` of the equipment that the admin wants to update the status on
- 35. `manageClassSchedule()` - (ADMIN)
  - Menu function for:
    - `viewClassSchedule()`**
    - `bookClass()`**
    - `cancelClass()`**
- 36. `viewClassSchedule()` - (ADMIN)
  - Display all classes, `room_id` of the class, `class_type`, `date`, and `time_of_day`
- 37. `bookClass()` - (ADMIN)
  - Prompts user for `room_id`, `Class_Type`, `Date`, `Time_Of_Day`
    - Return error if room not available for that time of day on that date
    - Return error if `class_type` invalid
  - Displays trainers available at that time of day
  - User selects a trainer for the session
  - Entry added to the `Dates_Trainer_Unavailable`
  - Asks user if they want to schedule another class
  - If yes, repeat function
  - If no, go back to **`manageClassSchedule()`**
- 38. `cancelClass()` - (ADMIN)
  - Ask user for the `class_id` of the class they want to cancel
    - Return error if the class does not exist
- 39. `viewBills()` - (ADMIN)
  - View all bills and properties of the bills

**Documentation:**

1. ER Diagram
2. Assumptions made regarding cardinalities and participation types
3. Reduction to Relational Schemas
4. DDL File
5. DML File
6. Video Demonstration

Functionality