SEG2105 - Introduction to Software Engineering - Fall 2021

Android Project (20%): Fitness Centre Booking App

This mobile application will implement a basic fitness class booking service for a particular fitness centre. Members will be able to enroll into drop-in classes, instructors will be able to schedule a class and edit details, and an administrator will manage all users and classes.

INSTRUCTIONS

1. The project will be done in teams of 5. This is the same team as your Labs.
2. You will commit code for each deliverable onto GitHub.
3. You will submit your application and UML for each deliverable via Brightspace.
4. Your team must complete a single version of the application with all functionality. For instance, you will NOT have one team member run one functionality on one phone, and another team member run a different functionality on a separate phone. The team must produce a single application.

The purpose of this project is to allow students to apply theoretical knowledge learned in class to a practical application. The project is designed to allow students to learn how to understand project descriptions, develop mobile applications, and test a product. Learning outcomes range from increased understanding of concepts related to software engineering, to overall knowledge of programming for Android, management, and team-relation skills.

The main outcome of the project is the implementation of a Fitness Class Booking application for Android devices. Students are to implement all components of the project, from their design specification, UML models, graphical assets, and source code. Students are encouraged to use the available tools in Android Studio but should refrain from copying whole blocks of code from the internet or peers to implement features. Should a group want to use a non-standard tool or API, they should request permission before doing so.

Note that this course does not focus on interface design, therefore, the project does not focus on usability aspects. Some design guidelines will be covered in tutorial sessions. Android Design Guidelines can be found at: Design for Android

USER TYPES

The app will be created with three different types of user in mind. The administrator, the instructors, and the gym members. The administrator manages all possible types of classes that can be offered to members at the fitness centre. The administrator also manages instructor and gym member accounts. Instructor users are able to select the type of class they would like to teach and then specify the class day, class time, class difficulty, and class capacity. The gym members are able to search for a drop-in class by class name and day of the week. They are then able to enroll and unenroll from classes.

The features that must be available for each user are outlined below. Note that these are the minimum required features. You are free to add more features to enrich your app.

The administrator can:

1. Login to an administrator account. The developer should pre-create such an account with these credentials:
    a. Username: *admin*
    b. Password: *admin123*
2. Create, edit, and delete types of classes to be offered by the fitness centre.
    a. To create a type of class, the administrator specifies the name of the class and a brief description of the class. No other course information is specified by the admin.
        i. Ex. The admin adds a *yoga* class, a *cycling* class, and a *kickboxing* class. Each class would have its own description.
    b. The administrator can edit the class name and description.
    c. The administrator can delete a type of class entirely.
3. Delete accounts of instructors and gym members.

An instructor can:

1. Create an instructor account and login to that account.
2. Choose a class to teach by selecting the type of class (pre-created by admin).
3. Add and edit the class day, time, difficulty, and capacity for a class they are teaching.
    a. Ex. *Cardio* on *Mondays* from 2:00-3:00*pm*, *beginner* difficulty, capacity 12.
4. Cancel a class, which deletes the class details.

A gym member can:

1. Create a member account and login to that account.
2. View all available classes and all classes they are enrolled in.
3. Search for a class by class name or by day of the week.
4. Enroll and unenroll from classes.

EXAMPLE USE CASE

Admin User:

- Creates 5 different types of classes: *yoga, cycling, kickboxing, aqua fitness, zumba.* Adds a description for each. Ex. kickboxing: “A high-energy workout that combines martial arts with cardio and strength training.”

Instructor User:

- Instructor A chooses to teach advanced yoga classes. They would select “yoga” as the type of class, select “advanced” as the difficulty level, choose Wednesdays at 7:30pm-8:30pm as the class time, and set the capacity to “35”.

Gym Member User:

- Member Z wants to enroll for a drop-in yoga class this week. They search for “yoga” and find an advanced level yoga class on Wednesday. Since the class is not at full capacity, the user is able to enroll into the class.

DETAILS

The project is to be carried out throughout the session and students are strongly encouraged to maintain a log of their project activities, as task allocation and project flow are components of the final report that you will submit at the end of the project. We suggest students keep track of duty assignments, with complexity of allocated tasks and completion dates.

Your application must be written in Java and built using Android Studio (you can use another IDE, but the TA will only provide support for Android Studio). You should compile your project against the earliest possible SDK version allowed by the API methods you are using. By the end of the semester, you must implement and present a working application based on the specifications. Firebase and SQLite will be presented in tutorials as options for storing and retrieving the application data. You must use Github for version control and your project repository must be shared with TA username: SEG2105-F2021.

ACADEMIC HONESTY

All work that you do towards the fulfillment of this course's expectations must be your own unless collaboration is explicitly allowed. Viewing or copying another individual's work (even if stored in a public directory) or lifting material from a book, magazine, website, or other source - even in part - and presenting it as your own constitutes academic dishonesty, as does showing or giving your work, even in part, to another student.

DELIVERABLE 1

In this deliverable, you must implement the user account management component and the admin functionality.

The app must allow users to create user accounts via a sign up page. Once the user logs in, they should see a second screen with the following message:

> *Welcome 'firstname/username'! You are logged in as 'role'.*

To simplify the development, there will be a single pre-created admin account. You can use the username *admin* and the password *admin123*. You should be able to create as many instructor and gym member accounts as desired.

The admin functionality includes creating, editing, and deleting types of classes. The admin must be able to create a type of class by entering a class name and a description. The admin can also edit the class name and description of existing types. These class types can later be seen by instructors and gym members in later deliverables. If desired, the admin should be able to delete a type of class entirely. The other functionality the admin should have is to be able to delete both instructor and gym member user accounts.

You can use Firebase or SQLite for database support. The deliverable must be uploaded by the date specified on Brightspace.

DELIVERABLE 1 MARKING SCHEME

Feature or Task	% Weight (Out of 100)
Github: Each member of the group has made at least ONE commit to the repository. Repo has been shared with TA Github username: SEG2105-F2021	10
UML class diagram for the description above, only pertaining to the functionality of deliverable 1. (-2 for each missing class) (-2 for incorrect generalization) (-0.5 for each incorrect multiplicity) (-0.5 for each missing attribute)	20
Can login as admin.	5
Can create and login as an instructor account.	10
Can create and login as a gym member account.	10

After successful authentication: - Can see a Welcome Screen after successful authentication. (1 point) - Can see the user role. (2 points) - Can see the name or username associated with the account. (2 points)	5
Admin can create a type of class by entering a name and description.	10
Admin can edit name and description of an existing class type.	10
Admin can delete a class type.	10
Admin can delete instructor and gym member accounts.	10
OPTIONAL - Fields are validated on all pages. Valid error messages. (ex. You cannot enter an invalid email, name, etc.) (ex. You cannot create a class with no name)	+5 bonus (+1 bonus for every validated field for a maximum of +5)

DELIVERABLE 2

In this deliverable, you must implement the instructor functionality. This includes viewing all scheduled fitness classes, scheduling a class that the instructor will teach, and editing their existing classes.

The instructor is able to view a list of all scheduled classes. They are also able to search by class name (ex. “Karate” or “Tai Chi”) and instructor names (including their own name).

An instructor user can choose to teach a class. To do so, the instructor selects the type of fitness they will teach (from the options that the admin created). Then the instructor selects the difficulty level (beginner, intermediate, or advanced), the day of the week, and the time interval that the class will be scheduled for. Finally, the instructor enters a maximum capacity for the class.

The instructor user can also cancel a class. This removes the class and all specified details (day, hours, difficulty level, and class capacity).

An instructor cannot schedule a class that is already scheduled by another instructor on the same day. (Ex. instructor cannot schedule a *HIIT* class on Thursday if there already exists a *HIIT* class on Thursday). Either include a valid error message upon saving, or change the UI to reflect this case.

DELIVERABLE 2 MARKING SCHEME

Feature or Task	% Weight (Out of 100)
Updated UML class diagram based on deliverable 2 description. (adding onto the UML diagram from deliverable 1). (-2 for each missing class) (-2 for incorrect generalization) (-0.5 for each incorrect multiplicity) (-0.5 for each missing attribute)	10
5 Unit test cases related to deliverable 1 or 2.	20
Instructor can view all scheduled classes.	5
Instructor can search for a class by: - Class name (5 points) - Instructor name (5 points)	10
Instructor can create a class by selecting the type of class (ex. *yoga*) and then choosing the class day, hours, difficulty level, and capacity limit.	10
Instructor can edit the class day, hours, difficulty level, and capacity for classes they are instructing.	10
Instructor can cancel a class. This deletes the existing class details (day, hours, difficulty, capacity)	10
Instructor cannot schedule a class that is already scheduled by another instructor on the same day. (Ex. instructor cannot schedule a *HIIT* class on Thursday if there already exists a *HIIT* class on Thursday). Either include a valid error message upon saving, or change the UI to reflect this case.	15
All fields are validated along with a valid error message. For instance, users should not be able to enter an invalid number for class capacity. (-1 for each field related to deliverable 2 where the user input is not validated)	10
OPTIONAL - Instructor sees the name of the other instructor when prevented from scheduling the same class on the same day.	+5 bonus

DELIVERABLE 3

In this deliverable, you must implement the gym member functionality as well as finalize the entire application.

The gym member user functionality includes viewing all fitness classes and searching for specific ones. The member must be able to view all classes that are offered as well as search using a class name and day of the week. For example: A member user wants to see all classes offered on Friday. Additionally, member users can view all the details of a class (description, day, hours, difficulty level, capacity).

Another functionality is that member users can enroll into a class. If a member user attempts to enroll into a class that has conflicting hours with an already enrolled class, then the app must show a valid error message. For example: A member user has already enrolled into *aqua fitness* on Tuesday at 10:00-11:30am. Then, when they try to enroll into *weightlifting*, which is on Tuesday at 11:00am-12:00pm, the app will display an error. Additionally, a gym member must not be able to enroll into a class that is at max. capacity.

Member users are able to see a list of all classes that they have enrolled in. Finally, they are able to unenroll from a class.

For the bonus, instructor users should be able to see a list of all member users who have enrolled into a class that they are instructing.

The final report is included with this deliverable and must be submitted on Brightspace.

DELIVERABLE 3 MARKING SCHEME

Feature or Task	% Weight (Out of 100)
Updated UML class diagram based on deliverable 3 description. (adding onto the UML diagram from deliverable 2). (-2 for each missing class) (-2 for incorrect generalization) (-0.5 for each incorrect multiplicity) (-0.5 for each missing attribute)	10
Final report including: - Title page. (1 point) - Short introduction about the project. (4 points) - Updated UML class diagram. - Table with the roles in the team and contributions of each team member for each deliverable. (10 points)	30

- All screenshots of your app. (10 points) - Lessons learned/challenges overcome. (5 points)	
5 Unit test cases relevant to this deliverable.	10
Member can view all scheduled fitness classes.	5
Member can search for a fitness class by: - Class name (5 points) - Day of the week (5 points)	10
Member can enroll into a fitness class.	5
Member can unenroll from a fitness class.	5
Member can see a list of all fitness classes they are enrolled for.	10
Member receives an error message when trying to enroll for a fitness class that conflicts with the time of a class they have already enrolled for.	10
Member cannot enroll into a class that has reached its maximum capacity. Either display an error or update the UI.	5
OPTIONAL - Instructor should be able to see a list of all enrolled gym members. (Only for a class they are instructing)	+5 bonus