uOttawa

# SEG2105 – Introduction to Software Engineering – Fall 2023

## Android Project: GCC Cycling Mobile Application (20%)

The Grimpeurs Cycling Club (GCC) App is a specialized mobile application that serves as a hub for all things cycling-related, including event coordination, registration, route tracking, and community engagement.

### Instructions

1. Project will be done in teams of 3-5 people.
2. Only one team member needs to submit the deliverables via Brightspace, but make sure all team members are identified (name and student number) on your cover page or README file.
3. The team must present only one version of the application. For instance, one student having one screen with the search functionality and other one having another different screen (running on a different phone) with the services provider functionality, **WILL NOT be accepted**. The team must produce a single application with all the required functionalities.

The purpose of this project to expand on the theoretical work, allowing students to gain practical experience implementing the concepts learned in class. This project is also designed to allow students to learn how to work with their colleagues and develop mobile applications. Learning outcomes range from increased understanding of concepts relating 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 GCC application for android devices. Students are to implement all components of the project, from their design specification, UML and additional documentation, graphical assets and source code. Students are encouraged to use the available toolset in android studio but should refrain from copying whole blocks of code from the internet to implement features. Should a group want to use a non-standard tool/API they should request permission before doing so.

---

The app will be conceived with three different types of users in mind. The administrator, the event organizer, and the participants. The administrator manages all the possible events that can

be offered to cycling participants. The event organizer creates a profile for his/her Cycling club and selects the type of events offered by that club (the club offers a set of events from the complete set of events created and maintained by the administrator. The participants should be able to search for a cycling club by name or type of event.

The features that should be available to each type of user are given below. Note that those are the minimum required features, and you are free to add more features you think might enrich your application.

**The administrator can:**

1. **Event Management:** Create type of events (**at least 3** as elaborated below, but the more the merrier) that could be offered by different cycling clubs and required information that needs to be given by a participant wanting to join an event:
    a. **Time Trial:** Time trials, often referred to as "TTs," are individual races against the clock. Cyclists start at intervals and race alone to complete a set course as quickly as possible. It's a test of a rider's ability to maintain a consistent pace and maximize speed.
    b. **Hill Climb:** Hill climbing events challenge cyclists to ascend steep inclines or mountains as quickly as possible. Participants showcase their climbing abilities, endurance, and strength during these challenging races.
    c. **Road Stage Race:** Road stage races are multi-day events composed of multiple stages, each with its own route and terrain. Cyclists compete over several days, and the overall winner is determined by the lowest cumulative time across all stages. Events like the Tour de France are classic examples.
    d. **Road Race:** Road races are competitive cycling events held on paved roads. Cyclists race in a group, and the winner is typically determined by the first rider to cross the finish line. Road races vary in distance and can be one-day events or part of a stage race.
    e. **Group Rides:** Group rides are non-competitive, social rides where cyclists come together to enjoy riding as a community. They offer an opportunity for cyclists of various skill levels to ride together, share experiences, and explore new routes while emphasizing camaraderie and fun.
2. **Account Management:** Delete accounts of cycling clubs and participants.
3. **Content Moderation:** Monitor and moderate event-related content to ensure accurate and safe information.

**The Cycling Club (single person representing the club) can:**

1. **Event Creation:**
    a. Create and organize cycling events within the region, specifying event type, difficulty level, and route details.
    b. Set registration fees and participant limits.

2. **Route Planning:**
    a. Design cycling routes, including distance, elevation, and landmarks.
3. **Registration Management:**
    a. Accept participant registrations.
    b. Communicate event details and updates to participants.
4. **Results and Awards:**
    a. Record and publish event results.
    b. Distribute awards and recognition to top performers.

**The participant can:**

1. **Account Creation:**
    - Register as a participant to access event details and registration.
2. **Event Discovery:**
    - Browse and search for upcoming cycling events within the region based on location, date, and type.
3. **Event Registration:**
    - Register for cycling events and receive event confirmations.
4. **Route Tracking (optional):**
    - Use GPS tracking to record cycling routes and performance during events.

**Your application should:**

1. Show a summary of all events created by a cycling club.
2. Show to the admin, all registered users to allow them to delete user accounts.

Note: This course does not focus on interface design; hence, we do not focus on usability aspects. However, students are welcome to "beautify" their projects, should they be comfortable with user interface design. Consider the Android Design Guidelines when designing your application. This topic will be covered in a tutorial session and detailed information is available at: https://developer.android.com/design/index.html

## DELIVERABLES

The project is divided into 4 incremental deliverables. Students are required to submit each deliverable by the posted deadline online using Brightspace.

| Deliverable | Due date |
|---|---|
| **1 – GitHub repository and user accounts (3%)** | October 10 |
| **2 – Admin functionality (3%)** | November 6 |
| **3 – Club owner functionality (3%)** | November 28 |
| **4 – Participant and application functionality (9%)** | December 6 |
| **5 – Demo (2%)** | Last week of classes |

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 a component of the final document provided alongside the android application. We suggest students keep track of duty assignment, with complexity of allocated tasks and completion dates.

Your application must be written in Java and built using Android Studio 2022.3.1. 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 submit a working application based on the specifications. Firebase or SQLite can be used for storing and retrieving the application data.

## ACADEMIC HONESTY

**All work that you do towards the fulfillment of this course's expectations must be your own unless collaboration is explicitly allowed (e.g., by some problem set or the final project). Viewing or copying another individual's work (even if left by a printer or 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

You need to create a GitHub repository in which you will commit all your code. All members need to be added as contributors.

In this deliverable, you need to implement the user account management component. That is, the app needs to allow users to create user accounts.

To simplify the development, there will be a single admin account (**username: admin, pwd: admin**), but it should be possible to create as many clubs and participants as desired.

Once the user logs in, they should see a second screen with the following message, 'Welcome firstname! You are logged in as "role". No additional functionality needs to be implemented at this point.

You can use Firebase or SQLite as DB support.

**What to submit:**

A zip file that will include the following:

1. A readme file with the link for your repository and the list of all members.
2. An APK of your app (with the functionality described in deliverable 1). APK generated (debug one). APK can be found in <yourAndroidProject>/app/build/outputs/apk/app-debug.apk. Name the APK after the name of your team "group1_debug_apk".
3. A PDF document that contains an UML Class diagram of your domain model. This will only include the UML classes related to deliverable 1
4. 3 Unit test cases testing the creation of the accounts. You can include more than 3 test cases. We expect at this point very simple test cases.

Deliverable 1 marking scheme

| Feature or Task | % Weight (out of 100) |
|---|---|
| **Github:** Repository created in Github contains all members of the group | **10** |
| **Github:** Each member of the group has made at least ONE commit to the repository. | **20** |
| **UML Class diagram** of your domain model (-2 for each missing class) (-2 for incorrect generalization) (-0.5 for each incorrect multiplicity) (-0.5 for each missing attribute) | **30** |
| **APK** submitted | **5** |
| Can create a **cycling club account** | **10** |
| Can create a **participant account** | **10** |
| Can see the 'Welcome screen' after successful authentication.<br>Can see the user role<br>Can see the name or username associated to the account | **5** |
| **Fields are validated** in all screens (e.g., you can't enter an invalid email, name, etc)<br>(-1 for each field in which the user input is not validated) | **10** |
| **OPTIONAL** - Group uses a DB (e.g. Firebase or SQLITE, or other similar technology) | **+5** |

# DELIVERABLE 2

In this deliverable, you need to implement the Admin related functionality. That is, the app should allow admin users to add, edit and delete event types that can be offered by cycling clubs using the application. For each event, event details and registration requirements need to be specified (age, pace, level, etc).

In addition, the admin account must be able to delete cycling club accounts and participant accounts. A cycling club is managed by a single owner.

What to submit:

A zip file that will include the following:

1. A readme file with the link for your repository and the list of all members.
2. An APK of your app with the functionality described in deliverable 2.
3. A PDF document that contains an updated UML Class diagram of your domain model. This will only include the classes related to deliverable 1 and deliverable 2.
4. At least 5 unit test cases testing the functionality.

Notes:

- I need to be able to login using the credentials:
    - username: admin
    - password admin
- Make sure your apk is correctly generated.
    - Go to Build -> Build Bundle(s) / APK(s) > Build APK(s)
    - Android Studio saves the APKs you build in project name/modulename/build/outputs/apk/
    - **Use the _Debug one for submission.**
    - For more info: https://developer.android.com/studio/run/

Deliverable 2 marking scheme

| Feature or Task | % Weight (out of 100) |
|---|---|
| **Updated UML Class diagram** of your domain model<br>(-2 for each missing class)<br>(-2 for incorrect generalization)<br>(-0.5 for each incorrect multiplicity)<br>(-0.5 for each missing attribute) | **10** |
| **APK** submitted and **ALL** features working (see notes below) | **5** |
| **5 Unit test cases** (simple local tests). No need to include instrumentation or Espresso Tests (UI) | **30** |
| At least 3 type of events (check page 2) that can be offered by cycling clubs using the application implemented. An event has a name and list of required information (check page 2) | **15** |
| Can **remove event types** that are no longer being organized | **15** |
| Can **edit event types** | **15** |
| **All fields are validated**. For instance, you should not be able to create a service with no name. This should be implemented along with valid error messages.<br>(-1 for each field in which the user input is not validated) | **10** |
| **OPTIONAL** – Integration with CircleCI to see the automated builds and automated testing. This will become mandatory for deliverable 4. | **+5** |

# DELIVERABLE 3

In this deliverable, you need to implement the cycling club related functionality. That is, the app should allow club owners to complete the club profile and associate their club to the set of predefined and available type of events (that were created by the admin).

What to submit:

A zip file that will include the following:

1. A readme file with the link for your repository and the list of all members.
2. An APK of your app (with the functionality described in deliverable 3).
3. A PDF document that contains an updated UML Class diagram of your domain model. This will only include the UML classes related to deliverables 1-3.
4. ~~2 Unit test cases testing the functionality. You can include more than 2 test cases.~~

Notes:

- **Please have the application contain at least one cycling club account already implemented with the following credentials:**
  - **username: gccadmin**
  - **password: GCCRocks!**
- The TA will also attempt to create a cycling club profile from scratch.
- Make sure your apk is correctly generated.
  - Go to Build -> Build Bundle(s) / APK(s) > Build APK(s)
  - Android Studio saves the APKs you build in project name/modulename/build/outputs/apk/
  - **Use the _Debug one for submission.**
  - For more info: https://developer.android.com/studio/run/

Deliverable 3 marking scheme

| Feature or Task | % Weight (out of 100) |
|---|---|
| **Updated UML Class diagram** of your domain model<br>(-2 for each missing class)<br>(-2 for incorrect generalization)<br>(-0.5 for each incorrect multiplicity)<br>(-0.5 for each missing attribute) | **10** |
| **APK** submitted and **ALL** features working (see notes below) | **5** |
| ~~**2 Unit test cases** (simple local tests). No need to include instrumentation or Espresso Tests (UI)~~ | ~~**10**~~ |
| Can **complete the profile information**.<br>- Enter Instagram link or other social media link (mandatory field)<br>- Enter name of main contact<br>- Enter phone number (mandatory field)<br>You can include any other fields your find necessary.<br>The two fields above need to be included.<br>'Mandatory field' means that the user must specify a value. | **15** |
| Can create events of certain types (From the list of event types added by the admin, the club owner can select one event type)<br>Ex. Club owner can create a Group ride, Sunday Dec 2, 2023, for 20 participants (assuming Group ride was defined as one of the event types by the admin). | **20** |
| Can delete events from their profile when they are no longer being organized | **20** |
| Can edit events (change the values of any of the fields) | **10** |
| **All fields are validated**. For instance, you should not be able to enter an invalid phone number or address. This should be implemented along with valid error messages.<br>(-1 for each field in which the user input is not validated) | **10** |
| **OPTIONAL** – Can add a club logo | **+5** |

# DELIVERABLE 4

In this deliverable, you need to implement the participant related functionality. That is, the app should allow participants to search for a cycling club, type of event or specific event. The application must display the list of cycling clubs organizing such event.

What to submit: A zip file that includes the following:

- A readme file with the link for your repository and the list of all members.
- A PDF document that contains an updated UML Class diagram of your domain model. This will only all UML classes.
    1. The document must include the lessons learned (and suggestions) and a table stating the roles and contributions of team members for each deliverable. You must add explanations in those cases where you find that the contributions were not fair.
    2. All the screenshots of your app.
- 10 Unit test cases testing the functionality. You can include more than 10 test cases.
- APK generated (debug one). APK can be found in <yourAndroidProject>/app/build/outputs/apk/app-debug.apk
- Source code

Notes:

- **Please have the application contain at least one customer account already implemented with the following credentials:**
    - **username: cyclingaddict**
    - **password: cyclingIsLife!**
- Make sure your apk is correctly generated.
    - Go to Build -> Build Bundle(s) / APK(s) > Build APK(s)
    - Android Studio saves the APKs you build in project name/modulename/build/outputs/apk/
    - **Use the _Debug one for submission.**
    - For more info: https://developer.android.com/studio/run/

Deliverable 4 marking scheme

| Feature or Task | % Weight (out of 100) |
|---|---|
| **Updated UML Class diagram** of your domain model<br>(-2 for each missing class)<br>(-2 for incorrect generalization)<br>(-0.5 for each incorrect multiplicity)<br>(-0.5 for each missing attribute) | **5** |
| **APK** submitted and **ALL** features working **AND SOURCE CODE**<br>**Make sure you test your APK. An invalid APK will receive 0.** | **5** |
| Final report including:<br>- A title page (2.5 points)<br>- Short Introduction (2.5 points)<br>- Update UML class diagram<br>- Table with the roles in the team and contributions of team members for each deliverable. (10 points)<br>- All the screenshots of your app. (10 points)<br>- Lessons learned (5 points) | **30** |
| 10 Unit test cases (simple local tests). No need to include instrumentation or Espresso Tests (UI).<br>Test cases must be relevant to the features of deliverable 4 | **10** |
| Can search for a cycling club<br>- Type of event<br>- Event name<br>- Club name | **10** |
| Fill in the required information and join the desired event | **15** |
| Can rate a cycling club by providing a comment and a rating from 1 to 5 | **5** |
| ~~CircleCI - Build button needs to appear in the Github (see instructions below).~~ | ~~**10**~~ |

| | |
|---|---|
| **All fields are validated**. (-1 for each field in which the user input is not validated) | **10** |