uOttawa

SEG2105 - Introduction to Software Engineering - Fall 2018
Android Project: On-Demand Home Repair Services App (20%)

The app addresses homeowner's needs for on-demand home services. It allows users to connect with services providers such as plumbers, electricians, locksmiths, among other.

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 functionality.

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 an On-Demand Home Repair Services 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 service providers and the homeowners. The **administrator** adds services to be offered to **homeowners**. The **service provider** creates a profile and selects the category applying to the services he (she) offers. The homeowners can search for a specific service provider or see the list of all the service providers within a category.

The **administrator** can:

1. Create services (at least 10) to be offered:
   The list of services may include: Appliance install, carpet cleaning, moving, plumbing, appliance repair, Furniture assembly, Locksmith, painting, window cleaning, Electrical, Mould Remediation, Pest control, Junk Removal, Handyman services.
2. Specifies a rate per hour for the services created.

You may create categories and subcategories if you prefer (to better organize the screens).

The **Service Provider** can:

1. Create his/her profile, associate it with one or more services (that were created by the admin).
2. Enter his/her availabilities.

The **User** can:

1. Search for a service provider.
2. Book a service (based on the provider availabilities).
3. Rate a service (1 to 5 scale)

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 the Brightspace.

Deliverable	Due date
1. GitHub repository and **User accounts (4%)**	October 23
2. **Admin** functionality **(4%)**	November 10
3. **Service Provider** Functionality **(4%)**	November 20
4. **Home owner** functionality **(6%)**	December 6
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 if 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 the **Android Studio 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 toward 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 addition, you will need to add the user SEG2105F18** as contributor.

In this deliverable, you need to implement the user account management component. That is, the app needs to allow users to create, edit and delete 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 home owners and service providers as desired.

Once the user logs in, he should see a second screen with the following message, 'Welcome *firstname*! You are logged 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 Word 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.

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 services. A service must have an hourly rate.

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 Word 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. 5Unit test cases testing the functionality. You can include more than 5 test cases.

DELIVERABLE 3

In this deliverable, you need to implement the **Service Provider** related functionality. That is, the app should allow service providers to complete their profiles and associate their work to the set of predefined and available services (that were created by the admin) A service provider must also be able to enter his/her availabilities.

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 Word 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.

DELIVERABLE 4

In this deliverable, you need to implement the **Homeowners** related functionality. That is, the app should allow homeowners to search a service provider based on his/her needs. User may search by service provider name or by service name. He must be able to display the list of available service providers during a certain period of time.

What to submit: A zip file that will include the following:

- A readme file with the link for your repository and the list of all members.
- A Word 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 (entire android project).
    1. You can use git: **git archive -o latest.zip HEAD**
    2. Alternatively, right click on the folder and then ‘send to’ compressed via zip.