uOttawa

SEG2105 – Introduction to Software Engineering – Fall 2019

Android Project: Walk in Clinics Services App (20%)

The application addresses the need for people to know waiting times at nearby walk-in clinics without having to leave their home. It also allows users to know the services offered by nearby walk-in clinics and allow them to check-in/book appointments at the clinic of their choice.

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 clinic's employee functionality, **WILL NOT be accepted**. The team must produce a single application with all the required functionality.

The purpose of this project is 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 walk in clinics 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 walk-in clinics employees, and the patients. The administrator manages all the possible services that can be offered to patients by walk-in clinics. The walk-in clinic employee creates a profile for the clinic and selects the services offered by the walk-in clinic and the working hours. The patients can search for a walk-in clinic by address/type of service provided or by working hours.

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. Create services (at least 5) to be offered by walk-in clinics using the application.
2. Delete accounts of walk in clinics and users

The walk in clinic employee can:

1. Create an account for the walk in clinic
2. Select services provided and the rate for each service
3. Enter the working hours of the walk in clinic

The patient can:

1. Create an account
2. Search for a walk in clinic by address/type of service provided/working hours
3. Check in/book an appointment
4. Rate his/her experience at the walk in clinic

Your application should:

1. Show the expected waiting times for walk in clinics based on the number of patients waiting to be seen (assume a fixed time of 15 minutes per patient)
2. Show the rating that each walk in clinic has
3. Show the admin account all registered users to allow him/her 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 20
2 – Admin functionality (3%)	November 10
3 – Walk-in employee user functionality (3%)	November 20
4 – Patient and application functionality (9%)	December 4
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. **You can track and assign tasks with GIthub.**

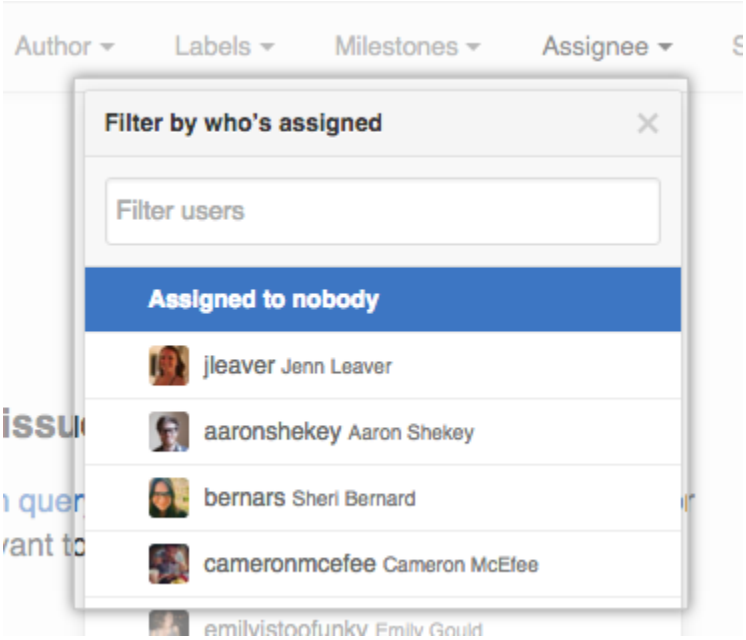
Your application must be written in Java and built using the Android Studio 3.1. You should compile your project against the earliest possible SDK version allowed by the API methods you are using (API 10, level 29). By the end of the semester, you must implement and submit a working application based 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 addition, you will need to add the user *SEG2105-uottawa* as contributor and enter in GitHub all tasks (you will need to break the deliverables into smaller tasks) and who is responsible for each task, as show in the image below.



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: 5T5ptQ), but it should be possible to create as many walk-in clinic employee accounts and patients 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*.

Ex. If a patient named Neymar (he injured his back while diving) logs in, he/she should receive the following message:

'Welcome *Neymar*! You are logged-in as *a patient*.

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 Github repository and the list of all members.
2. An APK of your app (with the functionality described in deliverable 1). The 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 (or pdf) document that contains an UML Class diagram of your domain model. This will only include the UML classes related to deliverable 1.

Deliverable 1 marking scheme

Feature or Task	% Weight (out of 100)
Github: Repository created in Github contains all members of the group, user *SEG2105-uottawa* and the tasks assigned.	**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 **walk in clinic employee account**	**10**
Can create a **patient account**	**10**
Can see the 'Welcome screen' after successful authentication. Can see the user role 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)	**10**

(-1 for each field in which the user input is not validated)	
OPTIONAL - Group uses a DB (e.g. Firebase or SQLITE, or other similar technology)	**+25**
Note: Passwords must be stored as hash values using SHA-256. See MessageDigest documentation.	**-10** **(if not implemented)**

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 that can be offered by walk-in clinics using the application.

In addition, the admin account must be able to delete walk-in clinic employee and patient accounts.

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:

- Provide in your README file the credentials needed to sign-in as admin (if a predefined admin account has been created).
- Categories and subcategories of services are optional. This can however help you organize your screens (imaging listing 50 services in a single screen).
- 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 (-2 for each missing class) (-2 for incorrect generalization) (-0.5 for each incorrect multiplicity) (-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**
Can **add services** to be offered by walk in clinics using the application. A service has a name and role of person performing the service (doctor/nurse/staff)	**15**
Can **remove services** that are no longer being offered	**15**
Can **edit services**	**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. (-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.	**+10**

DELIVERABLE 3

In this deliverable, you need to implement the walk in clinic employee related functionality. That is, the app should allow walk in clinic employees to complete their profiles and associate their clinic to the set of predefined and available services (that were created by the admin). The walk in clinic employee must also be able to set the working hours of the clinic.

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.

Notes:

- Provide in your README file the credentials needed to sign-in as a walk-in clinic employee (if a predefined walk in clinic account has been created). The TA will also attempt to create a service provider account 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 (-2 for each missing class) (-2 for incorrect generalization) (-0.5 for each incorrect multiplicity) (-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**. - Enter address and phone number (mandatory field) - Name of the clinic (mandatory field) - Insurance types and payment methods accepted (mandatory field). You can include any other fields your find necessary. The three fields above need to be included. 'Mandatory field' means that the user must specify a value.	**15**
Can **add services to their** profile (From the list of services added by the admin, the walk in clinic employee can select one or more services.)	**10**
Can **delete services from their profile**	**10**
Can **specify clinic working hours**. You are free to implement this feature as you want. That is, with a calendar or by selecting predefined timeslots, by week, by day, etc. Usability is key for this feature!	**20**
Can **see the list of (their own) working hours**.	**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. (-1 for each field in which the user input is not validated)	**10**
OPTIONAL – Can edit the working hours	**+10**

DELIVERABLE 4

In this deliverable, you need to implement the patient related functionality. That is, the app should allow patients to search for a walk in clinic by address/working hours/type of services provided. The application must display the list of available walk in clinics based on the user search and allow the user to check in/book an appointment. Finally, the application must show the expected waiting time based on the number of people waiting to be seen (15 minutes per person).

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 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

Notes:

- Provide in your README file the credentials needed to sign-in as a patient. The TA will also attempt to create a service provider account from scratch.
- Make sure your apk is correctly generated.
    - Go to Build -> Build Bundle(s) / APK(s) > Build APK(s)

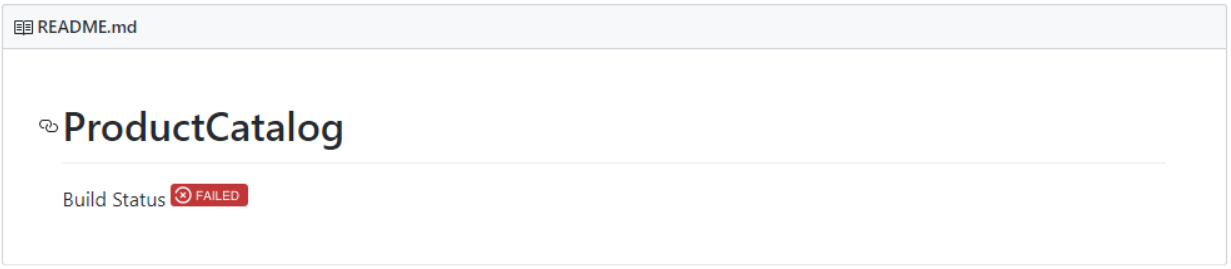
- Android Studio saves the APKs you build in project name/modulename/build/outputs/apk/
- **Use the _Debug one for submission.**
- For more info: https://developer.android.com/studio/run/

Deliverable 4 marking scheme

Feature or Task	% Weight (out of 100)
Updated 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)	**5**
APK submitted and **ALL** features working **AND SOURCE CODE** **Make sure you test your APK. An invalid APK will receive 0.**	**5**
Final report including: - A title page (2.5 points) - Short Introduction (2.5 points) - Update UML class diagram - Table with the roles in the team and contributions of team members for each deliverable. (10 points) - All the screenshots of your app. (10 points) - Lessons learned (5 points)	**30**
10 Unit test cases (simple local tests). No need to include instrumentation or Espresso Tests (UI). Test cases must be relevant to the features of deliverable 4	**10**
Can search for a walk in clinic by: - address - working hours - type of services provided	**10**
View approximate waiting time and check in/book an appointment	**15**
Can rate a clinic 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. For instance, you should not be able to select a date in the past when booking an appointment. This should be implemented along with valid error messages. (-1 for each field in which the user input is not validated)	**10**
BONUS FOR EXCELLENT UI DESIGN	**+10**

How to add a build button in your GitHub repo?



In your README.MD file, add the following line:

```
#Starts Here

Build Status

[![Build

Status](https://circleci.com/gh/SEG2105F18/ProductCatalog.png?branch=master)](https://circle

ci.com/gh/SEG2105F18/ProductCatalog)

#Ends Here
```

Elements in blue need to be modified to correspond with your username and project name.

Example: https://github.com/SEG2105F18/ProductCatalog/edit/master/README.md

Your build must be successful! **Green** Button instead of a red one.