**Lab Assignment #4**

**Due Date**: ***Week 8 (11.59 pm, Sunday)*** **Marks/Weightage:** **30/10%**

**Purpose:** The purpose of this lab assignment is to:

1. Develop an Android App that manipulates a Room database
2. Use **Room persistence library**
3. Use **MVVM** architecture

**References:** Textbook, ppt slides and Android tutorials (<https://developer.android.com/training/data-storage/room/index.html>, <https://github.com/googlesamples/android-architecture-components/tree/master/BasicSample>, <https://developer.android.com/reference/android/content/SharedPreferences>).

This material provides the necessary information that you need to complete the exercises.

**Be sure to read the following general instructions carefully:**

- This assignment **MUST be completed in pairs** following **pair programming** rules: <http://www.extremeprogramming.org/rules/pair.html>. You should **use the full names of both team members**.

- You will have to upload the completed assignment on eCentennial through the **assignment link under Assessments**.

**Android Workspace/Project Naming rules:**

You must name your Android Studio workspace and project according to the following rule:

**yourfullname\_COMP304SectionNumber\_Labnumber\_ExerciseNumber**.

***Example***: **JohnSmith\_COMP304Sec001\_Lab4\_Ex1 and each subsequent exercise should be added as JohnSmith\_COMP304Sec001\_Lab4\_Ex2 and so on.**

**Submission rules:**

Submit your projects as **zip files** that are named according to the following rule:

**yourfullname\_COMP304SectionNumber\_Labnumber\_ExerciseNumber.zip**

***Example***: **JohnSmith\_COMP304Sec001\_Lab4\_Ex1.zip**

**Upload your zipped assignment using the assignment link in e-centennial.**

**Exercise #1**

In this exercise you will write an Android application that allows the professor to keep track of various classroom assignments in a college. Your application could be used by administrators in colleges. Use **Room persistence library and MVVM pattern** (as shown in class examples) to create and manipulate the application's database. Create the following **entities**:

|  |  |  |
| --- | --- | --- |
| **Student** | **Classroom** | **Professor** |
| studentId  firstname  lastname  department  professorId  classroomId | classroomId  studentId  professorId  floor  airconditioned | professorId  firstname  lastname  department  password |

Add more fields to Test entity if needed. Your application should contain the following activities:

1. The **main activity** that handles the navigation. **[4 marks]**
2. The **login activity** will allow the professor to **login.** Use professorId as user name for the professor. **[4 marks]**
3. The **student activity** will allow the professor to **enter/view student** information. **[4 marks]**
4. The **classroom activity** will allow a professor to **enter classroom** data for a student. **[4 marks]**
5. The **view classroom info** activity will allow the professors to **view classroom information** for a given student. **[4 marks]**
6. The **update info** activity will allow the professor to **update**/**display student information**. **[6 marks]**
7. Use **Shared Preferences** to store professorId after successful login. Provide a friendly and easy to navigate UI. Use images and image buttons. **[4 marks]**

**Evaluation/Rubric:**

|  |  |
| --- | --- |
| **Functionalities**:   * Correct implementation of the model (entity classes, Room database, Dao, and Repository classes) * Correct implementation of ViewModel class. * All activities working, proper naming of activities, variables, and methods. Provide comments. Provide explanation when asked during the demonstration of the app. | 30%  20%  25% |
| **UI friendliness** (proper layout, controls, styles, themes, images) | 15% |
| **Declaring resources** in proper resource files | 5% |
| **Innovative features/effective way of coding** | 5% |
| **Total** | **100%** |