**Final-TERM EXAM [TEST-02]**

**Due Date: Week 14 (Midnight, Friday)**

**Marks/Weightage:** **100/25%**

Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Read the following general instructions carefully:***

* This lab test **must be completed individually** by all the students.
* Read the project **naming and submission guidelines** carefully.
* After completion and testing, **submit the project through the Final-Term Exam link on e-Centennial**.
* **Only the .zip file is accepted**
* **Only one zip file and one upload is allowed.**

**Android project naming rules:**

You **must** name your Android Studio project at the start of the Android Studio, according to the following rule:

**Yourfullname\_CourseCode\_SectionNumber \_Finalterm**

Example: **JohnSmith\_COMP304\_Sec001\_Finalterm**

**Submission rules: [Only the .zip file is accepted and only one zip file and only one upload is allowed]**

Archive your project in a **zip file** named according to the following rule:

**Yourfullname\_CourseCode\_SectionNumber \_Finalterm.zip**

Example: **JohnSmith\_COMP304\_Sec001\_Finalterm.zip (***e.g. if your section number is 001. Ignore if it does not apply)*

Upload the project through the **Final-Term Exam link** on **eCentennial**.

**Integrated Development Environment (IDE):**

1. Android Studio Version should be – 4.x.x
2. Android Studio gradle plug-in should be – 4.0.0
3. Gradle Scripts version should be – 6.1.1 or higher
4. Programming Language – Java SE7 onwards

**Exercise #1**  **[100 marks]**

Develop an Android application that retrieves stocks information from a Room database and sends it to an Android service or a phone device using SMS. **Use an Android Service if your first name starts with a letter from A-J, otherwise use SMS**. Your application should use **MVVM-Repository** architecture.

Start a new Android Studio project. Name it **Yourfullname\_COMP304\_001\_Finalterm.**

Create a new class name ***StockInfo*** that declares the following private instance variables:

**private** String **stockSymbol**; //primary key  
**private** String **companyName**;  
**private double stockQuote**;

Generate a default constructor, a constructor with three arguments, and getter/setter methods.

Apply the necessary annotations to make this an **entity class**. Create the **DAO interface**, **Room database** class, **Repository** class, **ViewModel** class and the **UI Activity**.

In the main activity:

1. Populate table *StockInfo* table with two rows (hardcode this). Put this code in the *onClick* event of Insert Stocks button. [30 marks]
2. When the user clicks on Display Stock Info button:
   1. Select the *stock symbol* from the RadioButton controls. [10 marks]
   2. Retrieve the *company name* and *stock quote* from *StockInfo* table. [20 marks]
   3. Display the Stock Info in a text view [10 marks]
   4. If you have to use an **Android Service**: [30 marks]
      1. Start an Android Service and send the *company name* and *stock quote* to the service. Pass the stock information retrieved from the table to the service using the Intent method *putExtra*. Display the stock information in *onStartCommand* method of the service using a Toast as shown in the following picture.
   5. If you have to use **SMS**:
      1. Send an SMS message with the *company name* and *stock quote* to a phone device (it could be the same device).Make sure the information is displayed on its screen.

|  |
| --- |
|  |

**Evaluation:**

|  |  |
| --- | --- |
| **Functionality**: |  |
| Main Activity (UI and Event Handling) | 30% |
| Using Room Database API (MVVM – Repository) | 45% |
| Service API or SMS API | 15% |
| **UI friendliness and defining resources in XML files**: | 10% |
| **Total**: | 100% |