Hands-On TEST 02 (SEC 003)

DATE: ***12th Dec., 2024*** **Marks/Weight: 50/25%**
Time: ***2.30 pm to 04.25 pm***

Be sure to read the following general instructions carefully:

- This hands-on test **must be completed individually** by all the students. **DO NOT** give your solution to others, otherwise there will be serious consequences. You are not allowed to share, communicate, e-mail, text during the test. See College Academic Policy
- Read the project **naming and submission guidelines** as explained in the following sections
- You should **provide the runtime screenshots** of your solution and **submit the project and screenshots through the Test02 link on eCentennial**.

IDE: Android Studio – Koala Version and Kotlin Jetpack Compose

Android Project Naming rules:

1. You must name your **Android Studio project** according to the following rule:
   YourFullName_COMP304_003_Test02
   Example: NickAdam_COMP304-003_Test02
2. Use minSdkVersion **2**4 and targetSdkVersion **34 or higher**
3. **Package name** must be **com.yourfirstname.yourlastname**, for example: com.nick.adam
4. **Main activity** should be named: YourFirstNameActivity, for example: NickActivity.
5. The **second activity** should be named YourLastNameActivity, for example: AdamActivity
6. Must provide **student's name and number as a comment** at the top of each activity

Submission rules:

1. Archive your project in a **zip file** named according to the following rule:
   YourFullName_COMP304-003_Test02.zip
   Example: **NickAdam_COMP304-003_Test02.zip**

2. Upload the project through the Hands-On Test 02 link on **eCentennial**.

Exercise #1

Develop an Android application that retrieves stock information from a Room database and displays it in another activity. Your application should use the **MVVM-Repository architecture**.

Instructions:

Entity Class:

- If your first name starts with a letter from A-N inclusive:

- Create a new class named CompanyStock with the following private instance variables:

```
@Entity(tableName = "company_stock")
data class CompanyStock(
   @PrimaryKey val companyName: String,
   val openingPrice: Double,
   val closingPrice: Double
    val yearlyAveragePrice:Double
)
```

- If your name starts with a letter from O-Z inclusive:
  - Create a new class named StockInfo with the following private instance variables:

```
@Entity(tableName = "stock_info")
data class StockInfo(
   @PrimaryKey val stockSymbol: String,
   val companyName: String,
   val currentStockQuote: Double
   val yearlyAverageStockPrice:Double
)
```

Create the DAO Interface:

- Define methods for inserting, updating, deleting, and querying the entity.

Create the Room Database Class:

- Define the abstract class that extends RoomDatabase.

Create the Repository Class:

- Implement the data access logic using the DAO.

Create the ViewModel Class:

- Provide the data to the UI and handle configuration changes.

Create MyApp.kt

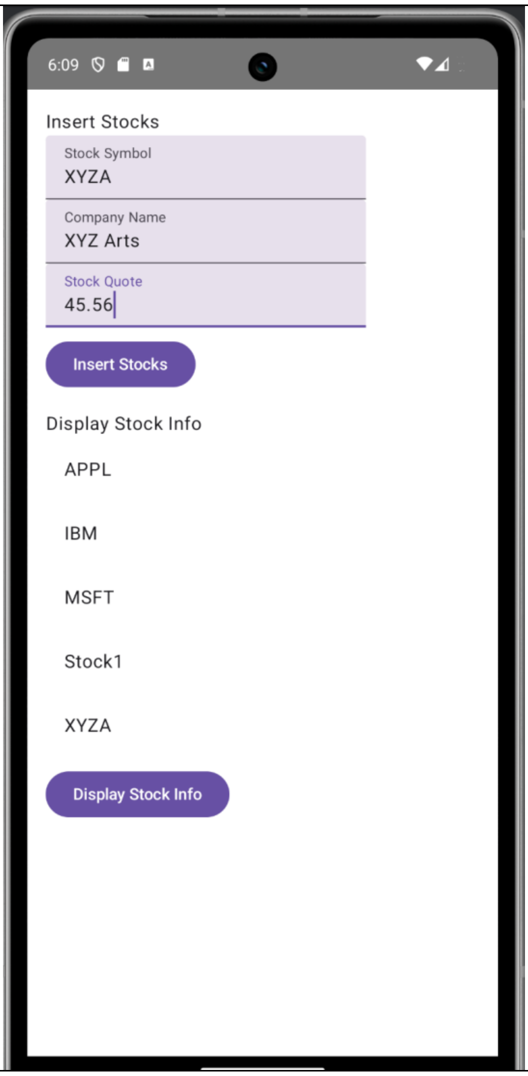
Create MyApplication class

Make sure to add this to Android manifest file: android:name=".MyApplication"

Create the UI Activities:

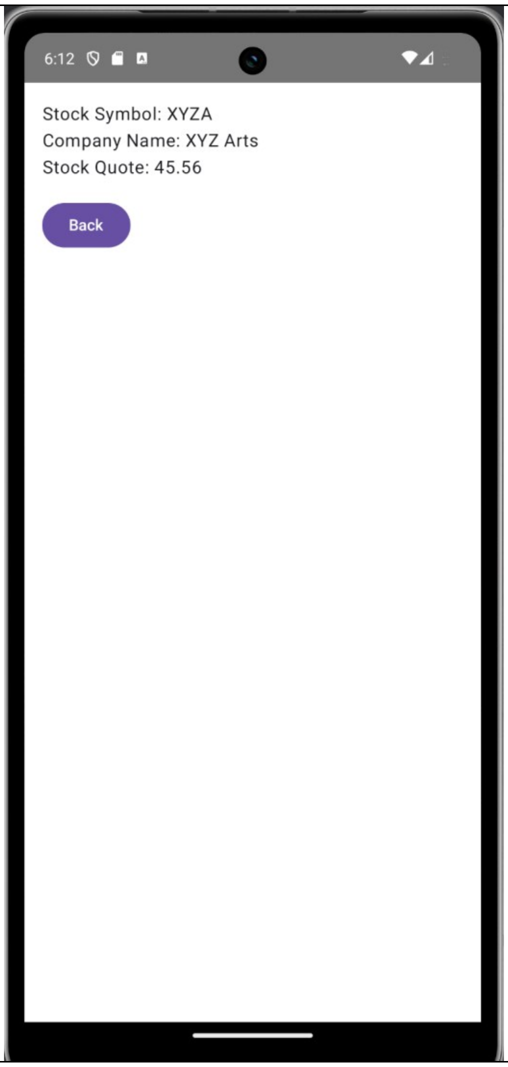
Main Activity:

- Allow the user to input stock information and insert it into the database.
- Display a list of stock symbols in the database using a LazyColumn.
- Provide a button to display the selected stock information in DisplayActivity.



Display Activity:

- Display the retrieved stock information in Text elements.
- Provide a "Back" button to return to MainActivity.



Stock Symbol: XYZA
Company Name: XYZ Arts
Stock Quote: 45.56

Evaluation Table:

Item	Percentage of Total Mark
Project Setup	**10%**
- Create a new Android project	5%
- Ensure Jetpack Compose is enabled	5%
Database Setup	**20%**
- Define StockInfo or CompanyStock entity class	5%
- Create StockDao interface	5%
- Create AppDatabase class	5%
- Create StockRepository class	5%

Item	**Percentage of Total Mark**
ViewModel Setup	**20%**
- Create StockViewModel class with StateFlow	10%
- Implement methods to insert stock information and fetch stock symbols	10%
MainActivity	**30%**
- Allow user to input stock information and insert it into the database	10%
- Display a list of stock symbols in the database using a LazyColumn.	10%
- Provide a button to display the selected stock information in DisplayActivity	10%
DisplayActivity	**10%**
- Display the stock information passed from DisplayActivity	5%
- Provide a "Back" button to return to MainActivity	5%
UI Components	**10%**
- Use Jetpack Compose to build the UI	5%
- Ensure the UI is responsive and user-friendly	5%
Total	**100%**