# Data Management & Architecture Components

## *DataStore, Room, LiveData & ViewModel*

**CSC2007 & ICT2105 Mobile Application Development Spring 2022**

## Overview

This lab provides guidelines to create a simple app to gain familiarity with Android Data Management and Architecture Components, namely DataStore, Room, LiveData & ViewModel.

## Outcomes

Upon completion of the session, you should be able to:

- Create a simple interactive interface
- Save app settings using Preferences DataStore
- Implement Android app architecture using LiveData & ViewModel
- Create and populate a simple SQLite database using the Room architecture component

## Tutorials & Codelabs

The following are references to help you learn about the concepts in this exercise:

Preferences DataStore:
https://developer.android.com/topic/libraries/architecture/datastore
https://developer.android.com/codelabs/android-preferences-datastore#0

Room:
https://developer.android.com/training/data-storage/room
https://developer.android.com/codelabs/android-room-with-a-view-kotlin#0

LiveData
https://developer.android.com/topic/libraries/architecture/livedata

ViewModel
https://developer.android.com/topic/libraries/architecture/viewmodel

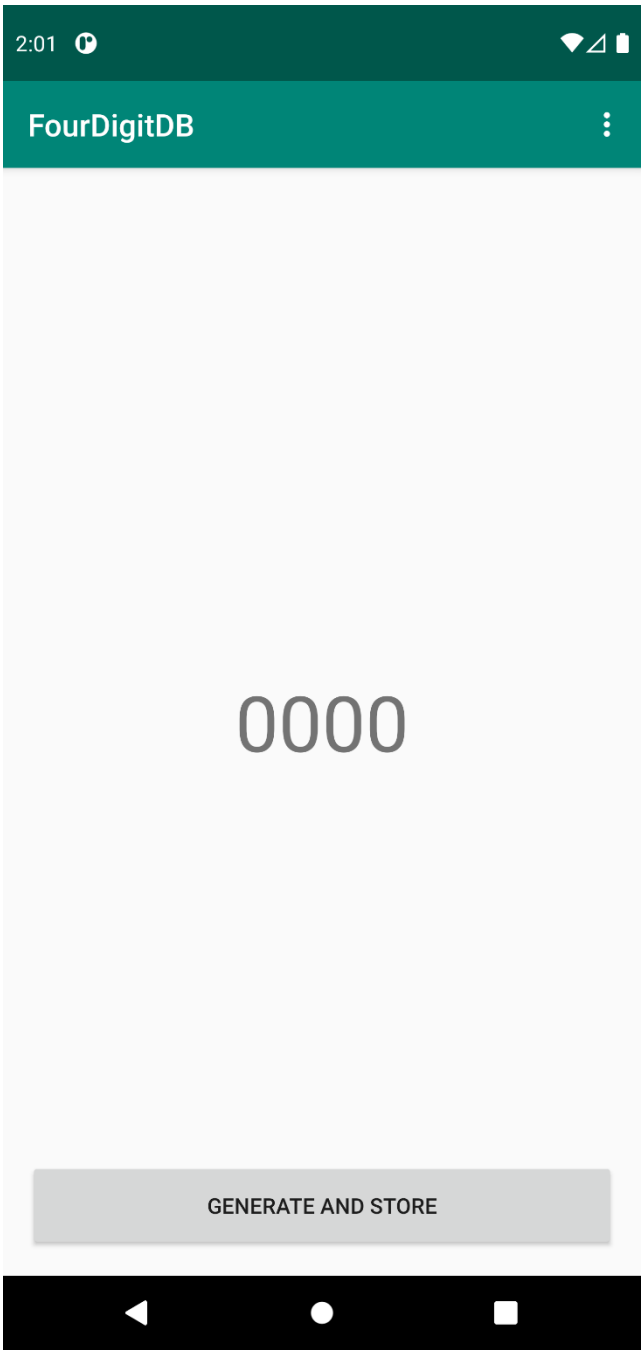
# The random four-digit generator application

This section provides an exercise to build a simple four-digit generator application (this has nothing to do with a certain popular 4D lottery game in Singapore). It consists of a user interface to randomly generate a four-digit number and save it to a SQLite relational database, using the Room architecture component.

The application includes a menu option to display a list of all the four-digit numbers that were saved to the database. This list should be displayed on a separate screen, using a RecyclerView.

Fork the repo **csc2007-lab04-2022** or **ict2105-lab04-2022** to get started. Design the layout of the screen for the `MainActivity` to be similar to the following screenshot.

Implement the logic to **randomly** generate a four-digit number between 1000 and 9999. The random integer generator should be implemented in `generateRandomFourDigitNumber(): Int`




**MainActivity**

IMPORTANT: Ensure that within the layout XML linked to MainActivity:
- TextView id displaying the number is named **textViewFourDigitNum**
- “GENERATE AND STORE” button id is named **generateButton**

## Saving four-digit numbers to a Room (SQLite) database

Next, implement architecture components to save every generated four-digit value to a Room database (backed by SQLIte), together with a ViewModel and LiveData, such that the user interface will be updated for every new saved value.

Hitting the “GENERATE AND STORE” button will create a new random four-digit number and insert it into the Room database and the ViewModel is updated. When the screen is rotated, the last generated four-digit value is maintained on the screen.

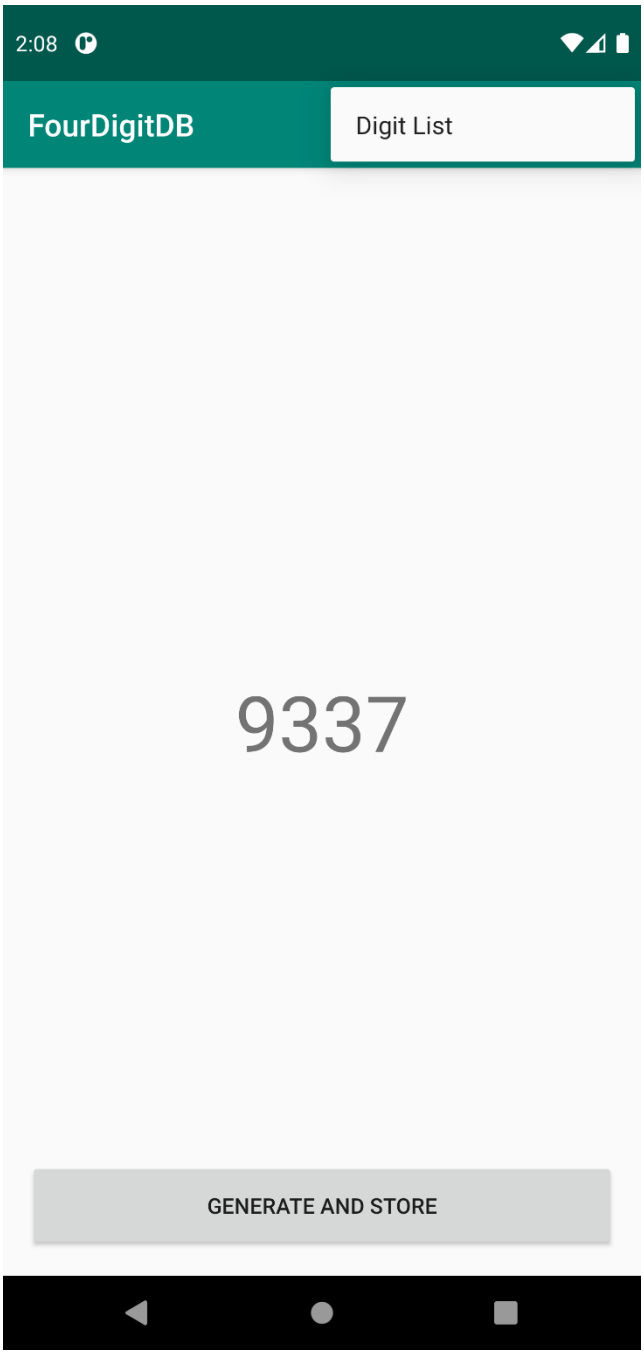
There are two columns for the SQLite database, the ID, or primary key for each record in the table (this is auto incremented and auto generated), and the value of the four-digit number itself.

Hint: The following classes and interfaces are most probably needed: **FourDigit, FourDigitDao, FourDigitRepository, FourDigitRoomDatabase, FourDigitViewModel.**

## Create an action bar menu item

Implement an action bar menu item that opens an option on the top right corner called “Digit List”. This menu option will display a second activity called `ListActivity`.

In the `AndroidManifest.xml` file for the app, add the correct elements so that this application can use the `ListActivity` class. This class will be used later to display a RecyclerView that contains the list of generated four digits.




**Action Bar Menu**

IMPORTANT: Ensure that within the menu XML linked to MainActivity:
- “Digit List” menu item id displaying the number is named **actionDigitList**

# Displaying a list of four-digit numbers

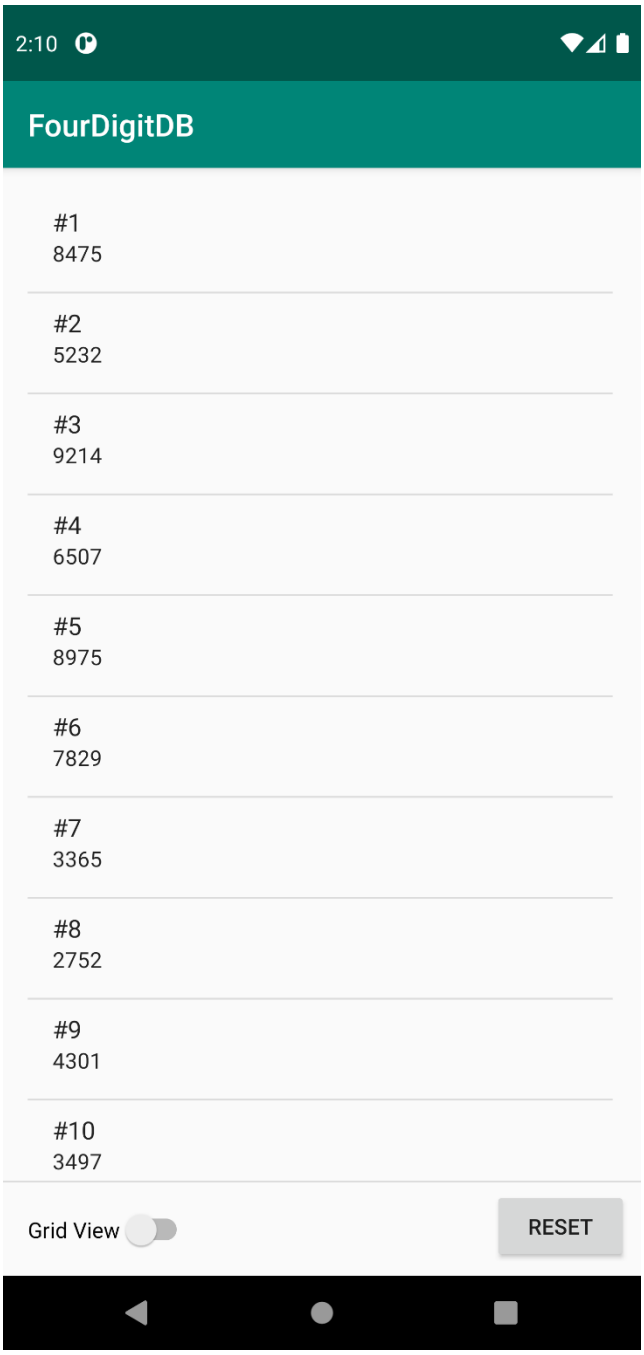
The ListActivity should display the list of randomly generated four-digit numbers that were stored in the database, and the autogenerated database index number.

When the screen with the RecyclerView is rotated, the list of saved four-digit values are maintained on the screen. Hitting the back button will return to the main screen with the large four-digit number.

The following classes are most probably needed: **FourDigitListAdapter and FourDigitViewModel.**

Additionally, you are to add a *Switch* button at the bottom of the ListActivity for switching between normal (LinearLayoutManager) and grid (GridLayoutManager) views (off = normal, on = grid view). *This setting must be saved using Preferences DataStore*, such that the setting is persisted between instances (i.e., the switch setting should remain as it was set by the user, even if the device is restarted).

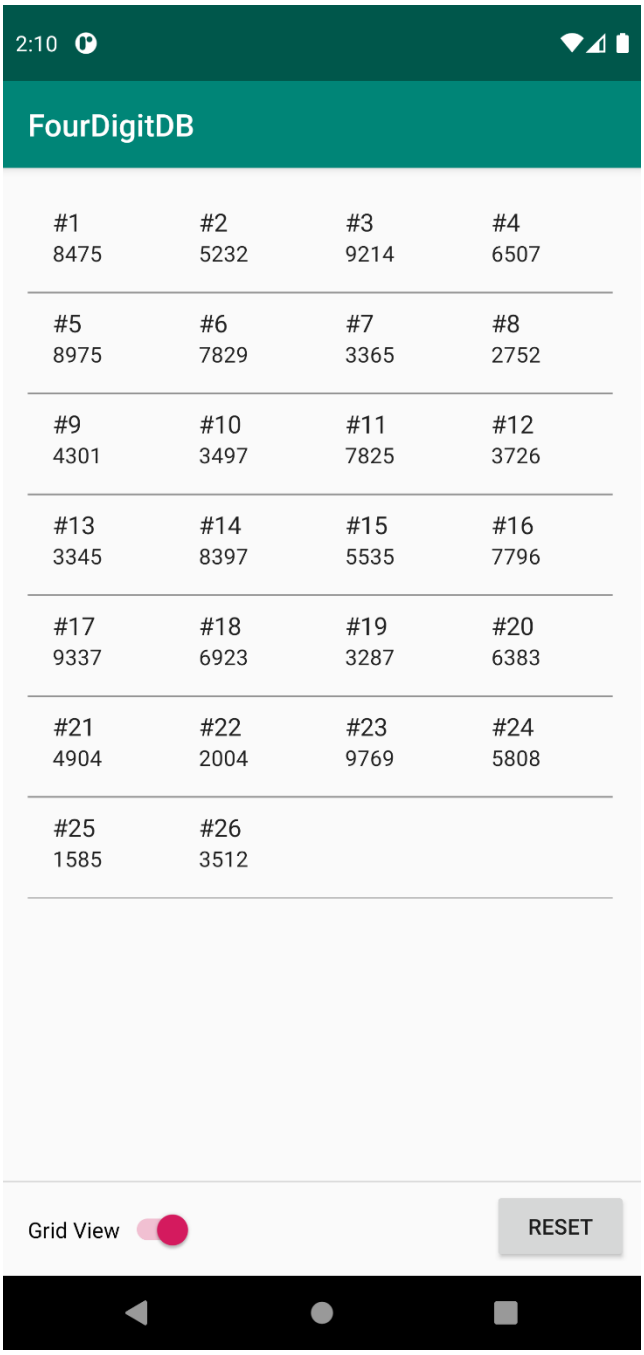
Finally, add a button called “RESET” at the bottom that clears all data from the database.




**ListActivity in Normal View**

IMPORTANT: Ensure that within the layout XML linked to ListActivity:

- RecyclerView id is named **listViewFourDigits**
- Switch id is named **switchGrid**
- “RESET” button id is named **resetDataButton**




**ListActivity in Grid View**

# Lab Exercise 4

**Due Date:**

**Wed Feb 22, 2022 2359 hrs (CSC2007)**
**Thu Feb 25, 2022 2359 hrs (ICT2105)**

1. Fork the repo **csc2007-lab04-2022** or **ict2105-lab04-2022.**
2. Design the layouts of the screen similar to the given screenshots.
3. Implement the logic on MainActivity screen, to generate a random four-digit number each time the “GENERATE AND STORE” button is clicked.
4. Implement the logic to create a simple database table for storing the four-digit values. The rows should consist of the id (auto-generated/incremented), and the four-digit number itself.
5. Implement the logic to add a new random number to the database each time the “GENERATE AND STORE” button is clicked.
6. Implement the app architecture using Room, LiveData & ViewModel
7. Implement the missing tags within AndroidManifest.xml to launch the correct activities.
8. Implement an additional menu option named “Digit List” to launch the ListActivity from the MainActivity.
9. In the ListActivity, all the random numbers in the database are displayed and updated in the RecyclerView.
10. In the ListActivity, add a *Switch* button for toggling the grid view on or off. This setting should be saved using Preferences DataStore, such that the setting is persisted between instances.
11. In the ListActivity, add a button, “RESET”, that clears all four-digit numbers from the database, resetting it back to empty.
12. Commit and push all changes to your forked repository **csc2007-lab04-2022** or **ict2105-lab04-2022.**

**END OF DOCUMENT**