CMPS 312 Mobile Application Development
Lab 4- User Interface (UI) development: Components and Layouts

Objective

The objective of the lab is to learn how to design user interfaces using Compose. Compose is a modern Android UI toolkit that makes it easy to create beautiful and user-friendly interfaces. It is based on the Kotlin programming language and uses a declarative syntax to describe the UI.

In this lab, you will learn the following:

- The basic concepts of Compose, such as composable functions, modifiers, and layouts.
- How to use Compose to create simple and complex user interfaces.
- How to customize the look and feel of your UI using modifiers and themes.
- How to reuse composable functions to simplify your code.
- How to test your user interfaces on different devices and screen sizes.
- How to use Material 3 Color Theming, Typography, Shapes

Overview

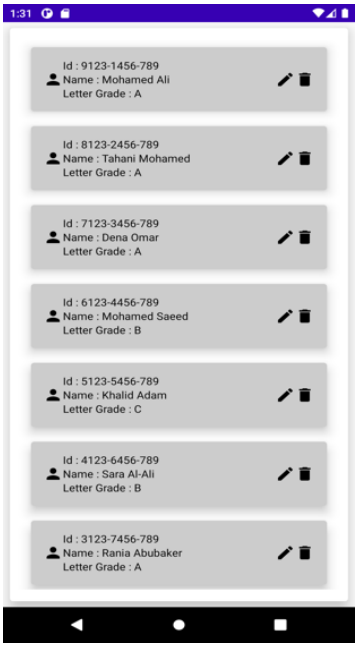
- **Part A :** Jetpack compose basics codelab tutorial
- **Part B :** You will create a simple android app using the concepts that you learned in the code lab

Part A – Jetpack Compose Basics

In this part of the lab, you will follow the **Google Codelabs** for Jetpack Compose basics. These codelabs will teach you all the fundamentals of Compose, such as composable functions, modifiers, state hoisting, and layouts.

You can visit this link to start the tutorial ➔ Jetpack Compose Basics

Part B – Create the following App



Using the concepts you practiced in part A, develop the following android applications.

Exercise 1

StudentList App: You are given the "students.json" file. So, your first task is to implement the first Screen, which lists all the students on the Screen. Then, you should use LazyColumn to display the students. Also, you need each row to be inside a card and have the necessary buttons such as edit and delete.

Implementation Details

1. Create a new project and name it **Student Management App**
2. Create assets folder and add the **students.json** file
3. Add the necessary dependencies that allow you to serialize JSON files [Refer to previous lab 3]
4. Create a data class called Student that is serializable inside a package called **model**

5. Create a class called **StudentRepository** inside a new package named repository
6. Implement the **StudentRepository** class. This class should have one single method called **getStudents().** This method should return the list of students inside the **students.json** file.
7. Create a file named **StudentList** that contains **StudentList composable and StudentCard composable. This composable should** get the students list and display them in a LazyColumn, as shown in Figure1.
8. Make sure you add **edit** and **delete** icons to each student row card.
9. Test the **StudentList** composable separately by creating a preview composable

Exercise 2

In this exercise, you will create a **Task Manager App** that allows users to manage their tasks and to-do lists efficiently. The app should have a visually appealing interface, making it user-friendly and engaging. Users should be able to perform the following actions:

1. **Add Tasks:** Users can add new tasks to their to-do list. Each task should have a title, a description, a due date, and a priority level.
2. **Edit Tasks:** Users can edit the details of existing tasks. This includes modifying the title, description, due date, and priority level.
3. **Delete Tasks:** Users can remove tasks from their list when they are completed or are no longer relevant.
4. **Task List:** Display the list of tasks in a dynamic and flexible manner using Jetpack Compose. Ensure that the tasks are sorted by priority and due date. Users should be able to easily see which tasks are most urgent.
5. **Visual Appeal:** Design the app with an aesthetically pleasing interface. You are free to use any colors, fonts, and UI elements that you believe will enhance the user experience, but ensure that the overall design is intuitive and cohesive.