

COURSE CODE : DFP50293	COURSE NAME : MOBILE APPLICATION AND DEVELOPMENT
NAME :	REGISTRATION NO:
CLASS :	DATE :
CLO1 : Construct the elements of GUI from java package that integrates database for an interactive GUI application (P4, PLO 3)	
DURATION : 10 Hours	
MODE : Pair / Group (Maximum 4 Students)	



Procedure: Follow the instructions below. (CLO1 : P3)

Question 1

Scenario:

Develop and demonstrate an Android application named “*MyStudentNote App*” by using Android Studio (Java). The application should integrate multiple UI components and a database to allow users to register, create, view, update, and delete notes efficiently.

Application Requirements :

A. APP CONTENT

i. Register and Login Page

- Students must **develop** a registration and login interface that enables new users to create an account by entering their **name, email, and password**.
- The page should include **input validation** to ensure data accuracy (e.g., valid email format, password length) and a **simple authentication logic** that verifies user credentials during login.
- Upon successful login, users should be directed to the **Main Menu Page**.

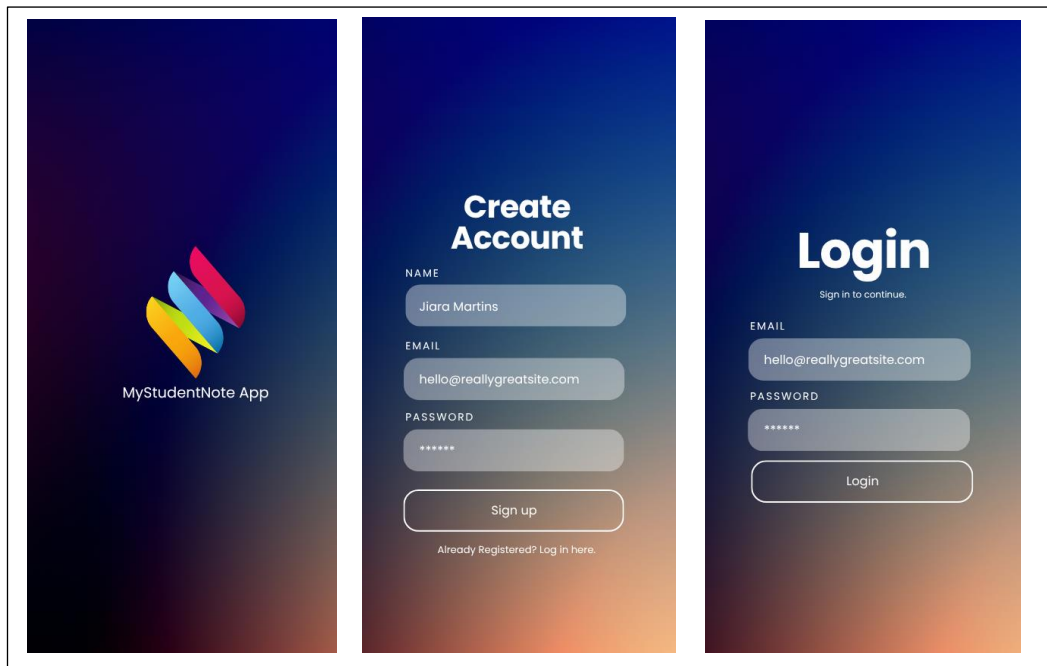


Figure 1 : Register and Login Page

ii. Main Menu Page

- Construct a **navigation hub** that displays four main options: **Add New Note, View Notes, Profile and Logout**
- Each menu item should be represented using icons or buttons with clear labels.
- The menu should demonstrate proper **navigation between Activities or Fragments**, ensuring that users can seamlessly access every feature of the app.

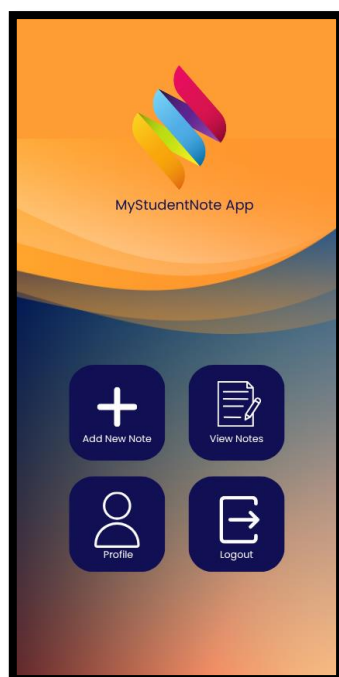
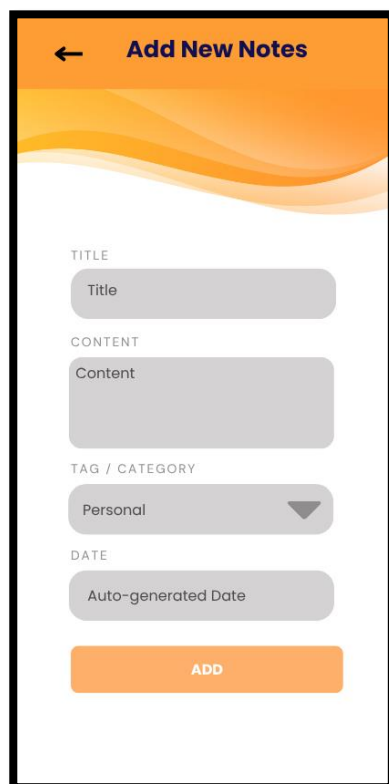


Figure 2 : Main Menu Page

iii. Add Note Page

- Design a form-like page that allows users to **create new notes**. This page should contain the following **input fields**:
 - i. **Title (Text)** – required field.
 - ii. **Content (Multi-line text)** – to record details or ideas.
 - iii. **Tag/Category (Spinner)** – options such as *Personal, Study, Work*.
 - iv. **Date** – automatically generated when a note is created.
- The page must validate the input fields before saving and display confirmation (e.g., Toast message: *"Note saved successfully."*).
- All notes should be stored in the local SQLite database.



The image shows a mobile application interface for adding a new note. At the top, there is a header bar with a back arrow and the title "Add New Notes". Below the header, there is a decorative orange and yellow wavy graphic. The form consists of four input fields, each with a label above it: "TITLE" (a single-line text field), "CONTENT" (a multi-line text field), "TAG / CATEGORY" (a spinner menu currently showing "Personal"), and "DATE" (a field showing "Auto-generated Date"). At the bottom of the form is an orange button labeled "ADD".

Figure 2 : Main Menu Page

iv. View Notes Page

- Develop a **RecyclerView-based page** that lists all saved notes from the database in a structured format.
- Each note item must show its **Title, Tag, and Date**.
- Users should be able to **tap on a note** to either **view, update, or delete** it.
- This page should also demonstrate basic CRUD operations effectively, with proper data refreshing and visual feedback (e.g., *"Note deleted"* or *"Note updated"*).

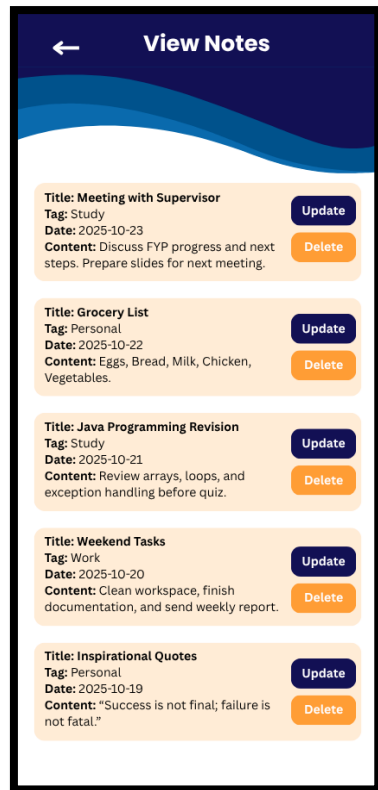


Figure 2 : Main Menu Page

v. Profile Page

- Construct a **Profile page** that displays the current user's information, including **name and email**.
- Users should be able to **edit and update their name** through this page. Implement simple input validation to ensure updated data is correctly saved in the database.
- The profile layout should maintain visual consistency with the rest of the application.

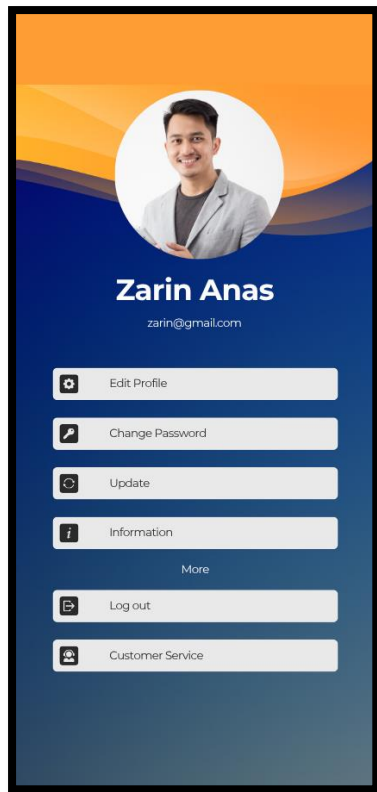


Figure 2 : Main Menu Page

vi. Logout Function

- Implement a **Logout feature** that securely ends the user session and redirects them to the **Login Page**.
- This demonstrates the student's ability to manage user states and data sessions within Android activities.

B. APP DESIGN

Students may customise the application design according to suitability and creativity. However, all data items and required functions must still be retained.

- Implement proper navigation between all pages using Activities and/or Fragments.
- Apply at least two (2) layout types, such as *LinearLayout* and *ConstraintLayout*, appropriately based on page design.
- Students may modify colours, button styles, icons, and typography to suit their design preference while maintaining overall usability and clarity.
- Ensure the application runs smoothly without errors or crashes during navigation or data processing.

C. DATABASE INTEGRATION

- Use **SQLite** as the main database to store all user and note information. Students must create a **database helper class** to manage CRUD operations with the following tables:

Table 1 : Users Table

Field	Type	Description
id	INTEGER (PK)	User ID (Auto Increment)
name	TEXT	User's full name
email	TEXT	User's email address
password	TEXT	Login password

Table 2 : Notes Table

Field	Type	Description
id	INTEGER (PK)	Note ID (Auto Increment)
title	TEXT	Note title
content	TEXT	Note content
tag	TEXT	Category of the note
date	TEXT	Date created
user_id	INTEGER (FK)	Reference to user ID

D. REPORT

Students are required to prepare a **mini report** comprising the following sections:

- ii. **Cover Page** – Name, matric number, section.
- iii. **Introduction** – Explain the project purpose, objectives, and intended learning outcomes.
- iv. **Content** – Include **screenshots of each interface** (Register/Login, Menu, Add Note, View Notes, Profile) with **brief explanations** of features and functions.
- v. **Conclusion** – Reflect on the **skills acquired**, challenges faced, and personal learning experiences from this mini project.

E. SUBMISSION

- i. Compress the entire Android Studio project folder into .zip or .rar format.
- ii. Upload to the designated submission platform before the deadline.
- iii. **Late submission will not be entertained.**

INFORMATION AND COMMUNICATION TECHNOLOGY DEPARTMENT									
MINI PROJECT SESSION : 1 2025/2026									
COURSE CODE		DFP50293			COURSE NAME		MOBILE APPLICATION DEVELOPMENT		
CLO1 : Construct the elements of GUI from java package that integrates database for an interactive GUI application (P4, PLO 3))					LECTURER'S NAME				
STUDENT'S NAME		1			REGISTRATION NUMBER		1		
		2					2		
		3					3		
		4					4		
CRITERIA	4	3	2	1	WEIGHT	STUDENT'S SCORE			
						1	2	3	4
1. Ability to Create New Android Activities	Can independently create and configure multiple activities with correct intent linking.	Creates multiple activities but minor linking/naming issues.	Limited activity creation; navigation partially set.	Unable to create or run activity correctly.	×1				
2. Ability to Run Project Using Emulator / Device	Successfully builds and runs smoothly on emulator/device without crashing.	Runs with minor warnings or UI lag.	Runs but occasionally crashes.	Fails to run or crashes repeatedly.	×1				
3. Login / Register Function	Fully functional registration and login with validation, authentication logic, and seamless navigation to main menu.	Works with minor validation or navigation issues.	Basic login/register works but lacks validation.	Incomplete or non-functional login/register.	×3				
4. Ability to Collect User-Entered Data (Add Note Page)	All input fields (Title, Content, Tag, Date) work correctly; validated and stored in DB with success feedback.	Works well but minor validation or saving issue.	Collects data but fails validation or partial save.	Form incomplete or cannot save data.	×3				

5. View Notes Function (RecyclerView / CRUD)	Fully functional view, update, and delete; proper data refresh and confirmation messages.	CRUD works but with small UI or logic issues.	Partial CRUD; only displays or deletes.	View page incomplete or fails to show data.	×4				
6. Profile Page Function	Displays and updates user info correctly with DB linkage and validation.	Displays info; update partially working.	Basic layout only; update not linked to DB.	Not functional.	×2				
7. Logout Function	Securely clears session and returns to login page.	Works but no session management.	Redirect only; session not reset.	Missing or broken logout.	×1				
8. Database Integration (SQLite)	Both Users & Notes tables implemented with full CRUD accuracy and relational linkage.	CRUD mostly works with small sync issues.	Partial database functionality.	No proper database linkage or frequent errors.	×4				
9. App Design, Layout & Creativity	Professional, appealing UI with consistent colour, icons, typography, and layout use (LinearLayout/ConstraintLayout).	Good consistency and creativity.	Acceptable layout but lacks visual balance.	Poor design or unorganised layout.	×3				
10. Task Completion & Following Instructions	Completed all tasks within time, followed instructions accurately, and well-tested.	Completed most with small deviations.	Partial completion or minor missing parts.	Did not follow given instructions or incomplete.	×2				
11. Originality of Work	100% original coding and layout; strong individual/group effort.	Mostly original with slight reference to samples.	Some copied code without understanding.	Mostly copied or template reused.	×2				
TOTAL									