Tutorial 3

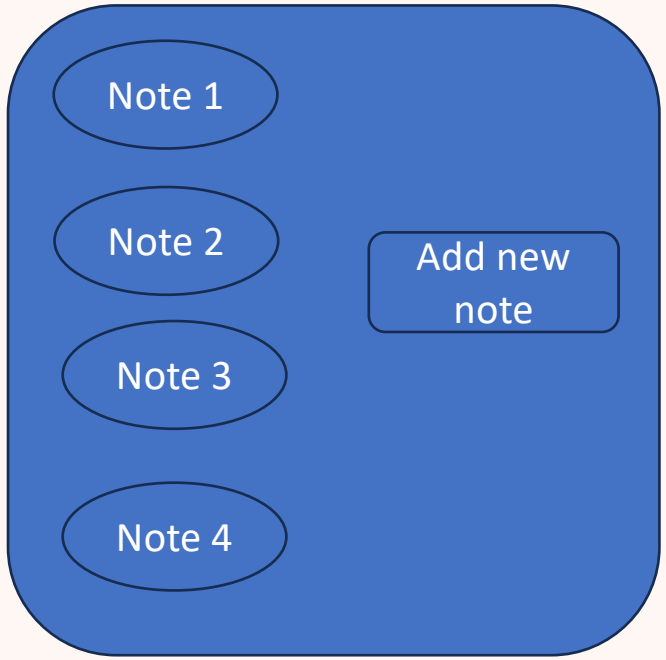
Simple Note-taking Application

Note-taking application

- You will develop a note-taking app.
- You will create a main activity where users can initiate the process of adding a new note. Upon clicking the "Add Note" button, a new activity will be started for note input. The entered note will then be returned to the main activity.
- The main activity saves the note into a list.
- Limitation 1: All notes will be saved in the memory. It means if the app is restarted, they will be gone!!
- Limitation 2: We can only save four notes.
- We will remove these limitations in later tutorials.

A dummy sketch of the MainActivity

- You are open to creating your own layout. This is just a dummy



- The note x buttons will only be visible when a note is saved. So initially all of them will be invisible. When we add the first note, then note 1 button will be visible, and so on

How to make a button invisible/visible

```
Button button1 = findViewById(R.id.button1);

// To make the button invisible
button1.setVisibility(View.INVISIBLE);

// To make the button gone (not taking up space)
button1.setVisibility(View.GONE);

// To make the button visible again
button1.setVisibility(View.VISIBLE);
```

MainActivity functionality

- Implement a click listener for the "Add Note" button. When clicked, this button should initiate the process of adding a new note by starting the note input activity using the ActivityResultLauncher.

```
detailButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent( packageContext: MainActivity.this, DetailActivity.class);
        detailActivityLauncher.launch(intent);
    }
});
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Initialize the ActivityResultLauncher
    someActivityResultLauncher = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        result -> {
            if (result.getResultCode() == RESULT_OK) {
                // Handle the result data here
                Intent data = result.getData();
            }
        }
    );

    // ... rest of your code
}

// Somewhere in your code, you can use the launcher
public void startOtherActivity() {
    Intent intent = new Intent(this, OtherActivity.class);
    someActivityResultLauncher.launch(intent);
}
```

MainActivity functionality (Cont..)

- Implement a click listener for the "Note X" button if it is visible. When clicked, this button should initiate the process of editing the existing note by starting the note-taking activity using the ActivityResultLauncher.

Note Input Activity – new note

- Design a note input activity layout that includes an EditText for users to input their notes.
- Upon opening this activity, display a user-friendly interface allowing efficient note input.
- Implement a save button in this activity to confirm and save the entered note.
- When the save button is clicked, package the entered note into an intent, set the result using setResult(RESULT_OK, intent), and finish the activity.

Note Input Activity – edit note

- If the note input activity is called from the saved note buttons, it will be used to edit the note.
- The save button will be renamed to “update”. The rest of the functionality will be the same.

MainActivity – handling the result

- Use the result code and getString from the intent. Example from lecture 3 source codes. An example from lecture 3:

```
ActivityResultLauncher<Intent> detailActivityLauncher = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        result -> {
            if(result.getResultCode()==RESULT_OK){
                Intent intent = result.getData();
                name = intent.getStringExtra( name: "NAME");
                detailActivityChecked =1;
                printButton.setAlpha(1.0f);
                printButton.setEnabled(true);
            }
        }
);
```

- **Now add the string to a list. If you have three strings/text in the list, make three note buttons visible.**

Marking Guideline

- App functionality – 15 marks. Partial marks will be given based on your implementation level.
- Adaptive layout – 5 marks. It should properly scale in phone, tablet, portrait and landscape modes.
- **Due 27 August Sunday 23:59 AWST, 20 Marks**
- **It is recommended to submit early and get marked. Do not wait for the due date. Tutorial 1,2,3,4 have the same due date, i.e., 27 August Sunday, 23:59 AWST**