**ASSIGNMENT:1**

**Task Manager App:**

**CODES:**

* **Activity\_main.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/tasksRecyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:listitem="@layout/item\_task" />  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/fabAddTask"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom|end"  
 android:layout\_margin="16dp"  
 android:contentDescription="Add new task"  
 app:srcCompat="@drawable/ic\_add" />  
</androidx.coordinatorlayout.widget.CoordinatorLayout>

* **Activity\_task\_creation.xml:**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp"

tools:context=".TaskCreationActivity">

<com.google.android.material.textfield.TextInputLayout

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/editTextTaskTitle"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Task Title"

android:inputType="textCapWords" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/editTextTaskDescription"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="top"

android:hint="Task Description"

android:inputType="textMultiLine"

android:minLines="3" />

</com.google.android.material.textfield.TextInputLayout>

<Button

android:id="@+id/buttonSaveTask"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="24dp"

android:text="Save Task" />

</LinearLayout>

* **Item\_task.xml**

<?xml version="1.0" encoding="utf-8"?>

<com.google.android.material.card.MaterialCardView xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="8dp"

android:layout\_marginVertical="4dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/textViewTaskTitle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAppearance="?attr/textAppearanceHeadline6"

tools:text="Task Title" />

<TextView

android:id="@+id/textViewTaskDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="4dp"

android:textAppearance="?attr/textAppearanceBody2"

tools:text="This is a description of the task." />

</LinearLayout>

</com.google.android.material.card.MaterialCardView>

* **MainActivity.java**

package com.example.taskmanagerapp;

import android.content.Intent;

import android.os.Bundle;

import androidx.activity.result.ActivityResultLauncher;

import androidx.activity.result.contract.ActivityResultContracts;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.google.android.material.floatingactionbutton.FloatingActionButton;

import java.util.ArrayList;

import java.util.List;

public class MainActivity extends AppCompatActivity {

private TaskAdapter taskAdapter;

private List<Task> taskList;

private ActivityResultLauncher<Intent> addTaskResultLauncher = registerForActivityResult(

new ActivityResultContracts.StartActivityForResult(),

result -> {

if (result.getResultCode() == RESULT\_OK) {

Intent data = result.getData();

if (data != null && data.hasExtra("NEW\_TASK")) {

Task newTask = data.getParcelableExtra("NEW\_TASK");

if (newTask != null) {

taskAdapter.addTask(newTask);

}

}

}

});

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

taskList = new ArrayList<>();

taskList.add(new Task(1, "Buy Groceries", "Milk, Bread, Eggs, Cheese"));

taskList.add(new Task(2, "Finish Assignment 1", "Complete Task Manager App."));

RecyclerView tasksRecyclerView = findViewById(R.id.tasksRecyclerView);

taskAdapter = new TaskAdapter(taskList);

tasksRecyclerView.setLayoutManager(new LinearLayoutManager(this));

tasksRecyclerView.setAdapter(taskAdapter);

FloatingActionButton fab = findViewById(R.id.fabAddTask);

fab.setOnClickListener(view -> {

Intent intent = new Intent(MainActivity.this, TaskCreationActivity.class);

addTaskResultLauncher.launch(intent);

});

}

}

* **TaskAdapter.java**

public void onBindViewHolder(@NonNull TaskViewHolder holder, int position) {

Task task = taskList.get(position);

holder.titleTextView.setText(task.getTitle());

holder.descriptionTextView.setText(task.getDescription());

}

@Override

public int getItemCount() {

return taskList.size();

}

public void addTask(Task task) {

taskList.add(task);

notifyItemInserted(taskList.size() - 1);

}

public static class TaskViewHolder extends RecyclerView.ViewHolder {

TextView titleTextView;

TextView descriptionTextView;

public TaskViewHolder(@NonNull View itemView) {

super(itemView);

titleTextView = itemView.findViewById(R.id.textViewTaskTitle);

descriptionTextView = itemView.findViewById(R.id.textViewTaskDescription);

}

}

}

* **TaskCreationActivty.Java**

package com.example.taskmanagerapp;

import android.content.Intent;

import android.os.Bundle;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class TaskCreationActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_task\_creation);

EditText titleEditText = findViewById(R.id.editTextTaskTitle);

EditText descriptionEditText = findViewById(R.id.editTextTaskDescription);

Button saveButton = findViewById(R.id.buttonSaveTask);

saveButton.setOnClickListener(v -> {

String title = titleEditText.getText().toString();

String description = descriptionEditText.getText().toString();

if (title.trim().isEmpty()) {

Toast.makeText(this, "Title cannot be empty", Toast.LENGTH\_SHORT).show();

return;

}

Task newTask = new Task(System.currentTimeMillis(), title, description);

Intent resultIntent = new Intent();

resultIntent.putExtra("NEW\_TASK", newTask);

setResult(RESULT\_OK, resultIntent);

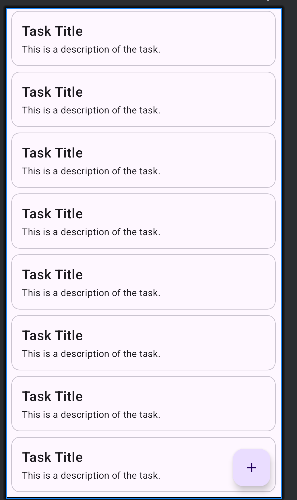
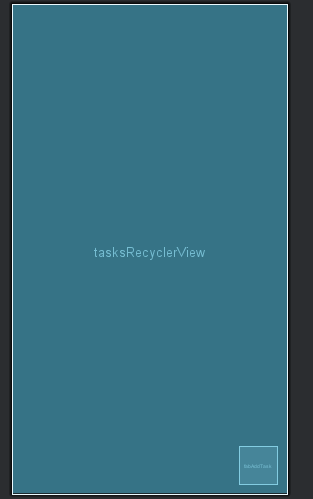
finish();

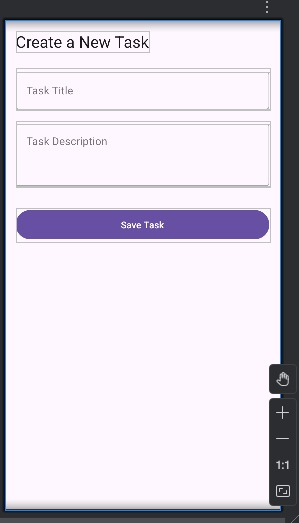
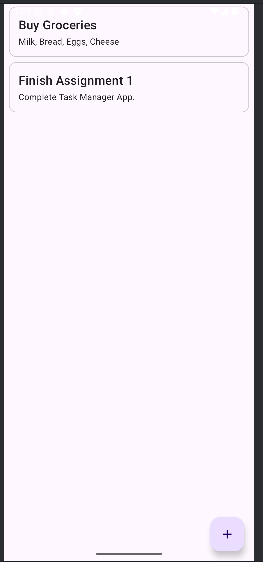
});

}

}

**OUTPUT**

****

** **

**Project Report: Task Manager App**

**1. Introduction:**

This report describes the design and implementation of a Task Manager application for the Android platform. The primary objective of this project was to create a functional and visually appealing mobile app using modern development tools and principles. The app allows users to create, view, and manage their daily tasks in a simple list format.

The project was developed using Android Studio, with Java as the programming language and XML for designing the user interface. A key focus was placed on implementing Material Design principles to ensure an intuitive and user-friendly experience.

**2. UI Design Choices:**

The user interface was designed to be clean, simple, and easy to navigate. The following Material Design components were used to achieve this:

* RecyclerView: The main screen uses a RecyclerView to display the list of tasks. This component was chosen because it is highly efficient for displaying long, scrollable lists of data. It recycles views as the user scrolls, which saves memory and ensures smooth performance, even with a large number of tasks.
* CardView: Each task in the list is presented within a CardView. This was done to visually separate each task, giving it a clean, modern look with distinct boundaries and a subtle shadow (elevation). It helps in organizing information and making the list easy to read.
* FloatingActionButton (FAB): A FloatingActionButton (the plus '+' button) is placed at the bottom-right of the main screen. This is a standard Material Design pattern for the primary action on a screen. In this app, its purpose is to allow the user to quickly navigate to the task creation screen, making the "add task" functionality highly discoverable.
* TextInputLayout: On the task creation screen, TextInputLayout was used for the input fields. This component enhances user experience by providing floating labels, which remain visible even after the user starts typing, and a clear, outlined-box style.

[Insert a Screenshot of Your App Here]

*(Run your app on an emulator, take a screenshot, and paste it into the report here.)*

**3. Navigation Flow:**

The application's navigation is designed to be straightforward and consists of two main screens:

1. Main Screen (MainActivity): The app opens to this screen, displaying the list of all existing tasks.
2. Task Creation Screen (TaskCreationActivity): The user clicks the FloatingActionButton (+) on the main screen.
3. Intents: An Intent is used to start the TaskCreationActivity. The MainActivity uses an ActivityResultLauncher to wait for a result back from this new screen.
4. Data Input: In TaskCreationActivity, the user enters the task title and description and clicks the "Save" button.
5. Passing Data Back: A new Task object is created. This object is made Parcelable, allowing it to be attached to a result Intent. The setResult() method is called with RESULT\_OK and the data.
6. Updating the List: Control returns to MainActivity. The ActivityResultLauncher's callback is triggered. The new Task object is extracted from the result Intent, and it is added to the RecyclerView's adapter, which automatically updates the list on the screen.

**4. Challenges Faced:**

During the development of this application, a few challenges were encountered:

* RecyclerView Setup: Initially, setting up the RecyclerView and its Adapter was a key challenge. It required a clear understanding of how onCreateViewHolder, onBindViewHolder, and getItemCount work together to display data correctly.
* Data Passing Between Activities: Implementing the modern ActivityResultLauncher API to pass data back from TaskCreationActivity to MainActivity was a learning experience. Unlike the older onActivityResult method, the new API is more type-safe and better structured, but it required understanding the setup of launchers and callbacks.
* Data Persistence: A limitation of the current app is that the tasks are not saved permanently. If the app is closed, the list of tasks is lost. A future improvement would be to integrate a local database like Room or use SharedPreferences to store the tasks, so they persist between app sessions.

**5. Conclusion:**

This project was a successful exercise in building a fundamental Android application. It provided hands-on experience with core Android components like Activities, Intents, RecyclerView, and Material Design principles. The final app is a functional task manager that meets all the requirements of the assignment.