PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE – 411043

**Department of Electronics & Telecommunication Engineering**

| |
|---|
| **BATCH AND ROLL NO: S6 42340** |
| **EXPERIMENT NO. 5** |
| **TITLE:** Design a mobile application to create different dialog boxes and menu ( popup, option ,context) |
| **DATE OF PERFORMANCE:** |
| **DATE OF SUBMISSION:** |

**Title:** Design a mobile application to create different dialog boxes and menu (popup, option, context)

**Requirements:**
1.Android Studio

**Theory:**

**Introduction**

In the ever-evolving field of mobile application development, the user interface plays a crucial role in shaping the user experience. Dialog boxes and menus are integral components that enhance user interactions within an application. This lab focuses on the design and implementation of a mobile application featuring different types of dialog boxes, including Popup Dialogs, Option Menus, and Context Menus.

**Objective of the Lab:** The primary goal of this lab is to guide you through the process of designing a mobile application with versatile user interaction components. Specifically, you will learn how to incorporate Popup Dialogs to display crucial information or prompt user actions, Option Menus for providing a set of actions within the app, and Context Menus to offer context-specific options based on user interactions.

**Components of the Application:**

1. **Popup Dialogs:**
    - Popup Dialogs are temporary overlay windows that appear on top of the current activity.
    - They are commonly used for alerts, confirmations, or presenting additional information without navigating to a new screen.
    - Popup Dialogs can be employed to offer contextual choices, providing users with quick access to specific actions.
2. **Option Menus:**
    - Option Menus provide a set of actions that users can access within the application.
    - They typically appear at the top of the screen and offer a range of options related to the current context.
    - Option Menus are ideal for presenting a concise list of actions that users may need at any point in the application.

3. **Context Menus:**
   - Context Menus are dynamic menus that appear when a user long-presses on a specific UI element, providing context-specific actions.
   - They are useful for offering relevant options based on the user's current interaction.

**Lab Prerequisites:**

- Basic understanding of mobile application development concepts.
- Familiarity with the chosen development environment (e.g., Android Studio).
- Prior knowledge of programming languages such as Java (for Android).

**Steps:**

**Step 1: Set Up Your Development Environment**

- Ensure that you have Android Studio installed and configured on your machine.

**Step 2: Create a New Project**

- Open Android Studio and create a new project.
- Choose an appropriate project template, such as "Empty Activity" or "Basic Activity."

**Step 3: Design the Main Activity Layout**

- Open the XML layout file associated with your main activity (e.g., activity_main.xml).
- Design the layout with relevant UI elements for triggering different types of dialog boxes and Popup Menus.

**Step 4: Implement the Java Code**

- Open the Java file associated with your main activity (e.g., MainActivity.java).
- Implement the logic for creating and showing Popup Dialogs, Option Menus, and Context Menus in response to user interactions.

**Step 5: Implement Popup Dialogs**

- Create methods for showing Popup Dialogs with different functionalities (e.g., alerts, confirmations).
- Utilize the AlertDialog.Builder class to build and display Popup Dialogs.

**Step 6: Implement Option Menus**

- Override the onCreateOptionsMenu method in your activity to create the Option Menu.
- Inflate the menu resource file with relevant menu items.
- Handle item selections in the onOptionsItemSelected method.

**Step 7: Implement Context Menus**

- Register the view or views for which you want to show the Context Menu using registerForContextMenu.
- Override the onCreateContextMenu method to define the items in the Context Menu.
- Handle item selections in the onContextItemSelected method.

**Step 8: Test Your Application**

- Run your application on an emulator or a physical device.
- Test the functionality of Popup Dialogs, Option Menus, and Context Menus by interacting with the UI elements triggering these components.

**XML Code:**

**activity_main.xml :-**

```
<!-- activity_main.xml -->

<RelativeLayout 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:padding="16dp"

    tools:context=".MainActivity">



    <Button

        android:id="@+id/btnPopup"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="Show Popup Dialog"

        android:layout_centerHorizontal="true"

        android:layout_marginTop="50dp"/>
    <Button
        android:id="@+id/btnOption"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Option Menu"
        android:layout_below="@id/btnPopup"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"/>

    <Button
        android:id="@+id/btnContext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Context Menu"
        android:layout_below="@id/btnOption"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"/>

</RelativeLayout>
```

**Popup_dialog.xml : -**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">


    <TextView
        android:id="@+id/textPopup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="This is a Popup Dialog"
        android:padding="16dp"
        android:textSize="18sp"
        android:textColor="@android:color/black"
        android:gravity="center"/>


    <Button
        android:id="@+id/btnClosePopup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Close"/>


</LinearLayout>
```

**option_menu :-**

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu_item_1"
        android:title="Option 1"/>
    <item

        android:id="@+id/menu_item_2"

        android:title="Option 2"/>

</menu>
```

**Context_menu :-**

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

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

    <item

        android:id="@+id/context_item_1"

        android:title="Context Item 1"/>

    <item

        android:id="@+id/context_item_2"

        android:title="Context Item 2"/>

</menu>
```

**Java Code:**

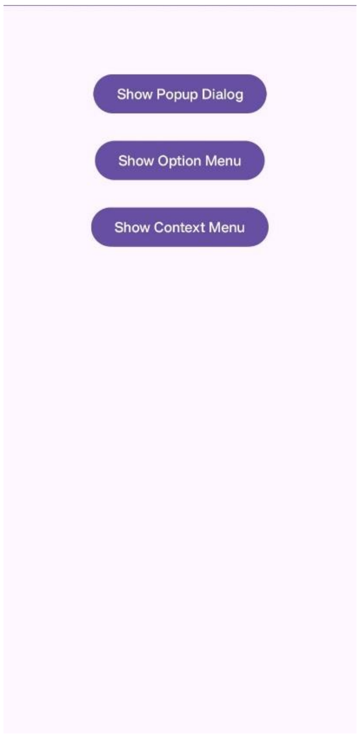
**MainActivity.java :-**

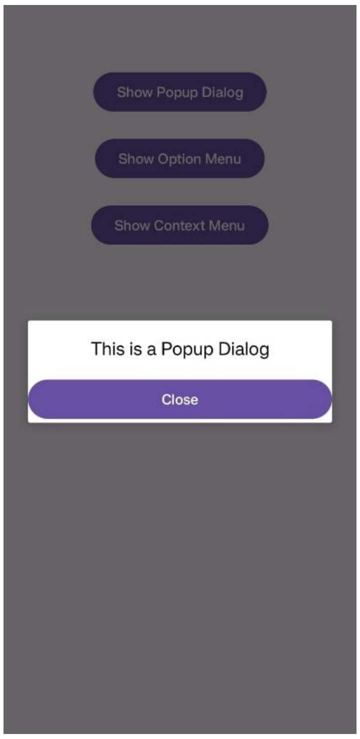
```
package com.example.ad_exp_5;

import android.os.Bundle;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.PopupMenu;
import android.widget.Toast;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private Button btnPopup, btnOption, btnContext;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnPopup = findViewById(R.id.btnPopup);
        btnOption = findViewById(R.id.btnOption);
        btnContext = findViewById(R.id.btnContext);

        btnPopup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showPopupDialog(v);
            }
        });

        btnOption.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showOptionMenu(v);
            }
        });

        registerForContextMenu(btnContext); // This line registers the button for context menu
    }

    private void showPopupDialog(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        View dialogView = getLayoutInflater().inflate(R.layout.popup_dialog, null);
        builder.setView(dialogView);
        AlertDialog alertDialog = builder.create();
        alertDialog.show();

        Button btnClosePopup = dialogView.findViewById(R.id.btnClosePopup);
        btnClosePopup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                alertDialog.dismiss();
            }
        });
    }

    private void showOptionMenu(View v) {
        PopupMenu popupMenu = new PopupMenu(this, v);
        popupMenu.getMenuInflater().inflate(R.menu.option_menu, popupMenu.getMenu());

        popupMenu.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem item) {
                if (item.getItemId() == R.id.menu_item_1) {
                    showToast("Option 1 selected");
                    return true;
                } else if (item.getItemId() == R.id.menu_item_2) {
                    showToast("Option 2 selected");
                    return true;
                }
                return false;
            }
        });

        popupMenu.show();
    }

    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        getMenuInflater().inflate(R.menu.context_menu, menu);
    }

    public boolean onContextItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.context_item_1) {
            showToast("Context item 1 selected");
            return true;
        } else if (item.getItemId() == R.id.context_item_2) {
            showToast("Context item 2 selected");
            return true;
        }
        return super.onContextItemSelected(item);
    }

    private void showToast(String message) {
        Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
    }
}
```

Conclusion:

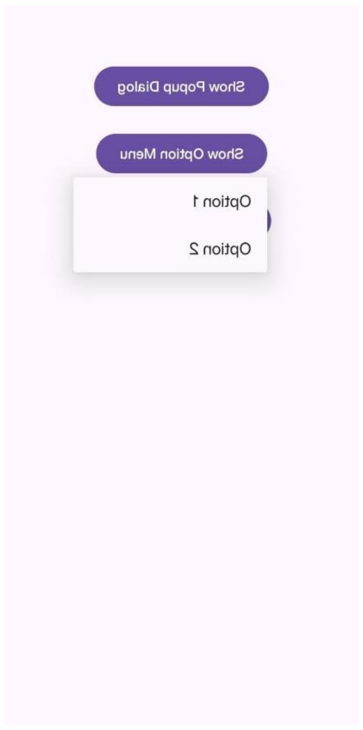
……………………………………………………………………………………………………
……………………………………………………………………………………………………
………………………………………………………………………………

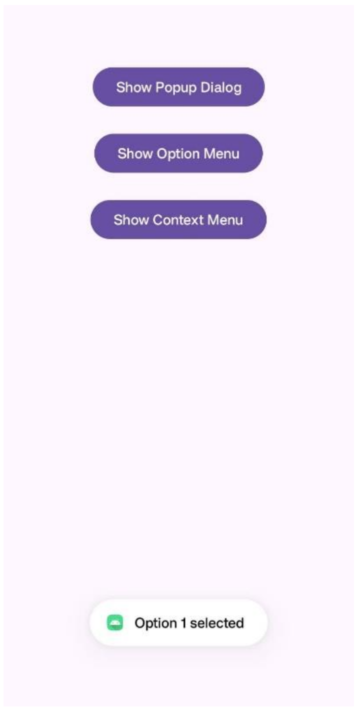
## Output:



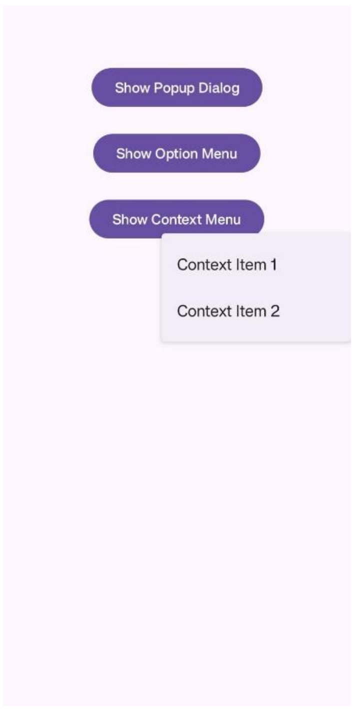


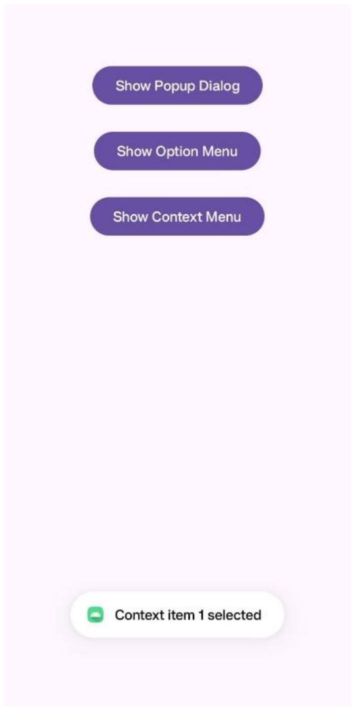
This is a Popup Dialog





Option 1 selected





Context item 1 selected