



BATCH AND ROLL NO: R08 42435

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:

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:title="Dialog & Menu Demo"
            android:layout_marginTop="30dp"
            app:titleTextColor="@android:color/white"/>

    </com.google.android.material.appbar.AppBarLayout>

    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">
```



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">

    <com.google.android.material.card.MaterialCardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        app:cardElevation="4dp"
        app:cardCornerRadius="8dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="16dp">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Dialog Boxes"
                android:textAppearance="?attr/textAppearanceHeadline6"
                android:layout_marginBottom="16dp"/>

            <com.google.android.material.button.MaterialButton
                android:id="@+id/btnSimpleDialog"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Show Simple Dialog"/>

            <com.google.android.material.button.MaterialButton
                android:id="@+id/btnCustomDialog"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Show Custom Dialog"/>

        </LinearLayout>

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

    <com.google.android.material.card.MaterialCardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        app:cardElevation="4dp"
        app:cardCornerRadius="8dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="16dp">
```



```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Long Press Items"
    android:textAppearance="?attr/textAppearanceHeadline6"
    android:layout_marginBottom="16dp"/>

<ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="200dp"/>

</LinearLayout>

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

</LinearLayout>

</androidx.core.widget.NestedScrollView>

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="16dp"
    android:src="@android:drawable/ic_dialog_info"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

dialog_custom.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"
    android:padding="16dp">

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your name">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/editTextName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textPersonName"/>

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

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your email"
        android:layout_marginTop="8dp">
```



```
<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/editTextEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"/>
```

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

```
</LinearLayout>
```

menu/contact_menu.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/action_edit"
        android:title="Edit"/>
    <item
        android:id="@+id/action_delete"
        android:title="Delete"/>
    <item
        android:id="@+id/action_share"
        android:title="Share"/>
</menu>
```

menu/main_menu.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/action_settings"
        android:title="Settings"
        android:icon="@android:drawable/ic_menu_manage"
        app:showAsAction="never"/>

    <item
        android:id="@+id/action_search"
        android:title="Search"
        android:icon="@android:drawable/ic_menu_search"
        app:showAsAction="ifRoom"/>

    <item
        android:id="@+id/action_about"
        android:title="About"
        app:showAsAction="never"/>
</menu>
```



AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.expt05_42441">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.DialogMenus">

        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Java Code:

ManiActivity.java:

```
package com.example.expt05_42435;

import android.os.Bundle;
import android.view.ContextMenu;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

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

import com.google.android.material.dialog.MaterialAlertDialogBuilder;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;

import java.util.Arrays;
import java.util.List;
```



```
public class MainActivity extends AppCompatActivity {

    private ListView listView;
    private List<String> items = Arrays.asList(
        "Item 1", "Item 2", "Item 3", "Item 4"
    );

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Setup toolbar
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        // Initialize views
        listView = findViewById(R.id.listView);
        FloatingActionButton fab = findViewById(R.id.fab);

        // Setup ListView
        ArrayAdapter<String> adapter = new ArrayAdapter<>(
            this,
            android.R.layout.simple_list_item_1,
            items
        );
        listView.setAdapter(adapter);
        registerForContextMenu(listView);

        // Setup click listeners
        findViewById(R.id.btnSimpleDialog).setOnClickListener(v -> showSimpleDialog());
        findViewById(R.id.btnCustomDialog).setOnClickListener(v -> showCustomDialog());

        fab.setOnClickListener(v -> {
            Snackbar.make(v, "This is a Snackbar message", Snackbar.LENGTH_LONG)
                .setAction("Action", null)
                .show();
        });
    }

    private void showSimpleDialog() {
        new MaterialAlertDialogBuilder(this)
            .setTitle("Simple Dialog")
            .setMessage("This is a simple dialog message")
            .setPositiveButton("OK", (dialog, which) -> {
                Toast.makeText(this, "OK clicked", Toast.LENGTH_SHORT).show();
            })
            .setNegativeButton("Cancel", null)
            .show();
    }

    private void showCustomDialog() {
        View dialogView = getLayoutInflater().inflate(R.layout.dialog_custom, null);
    }
}
```




```
new MaterialAlertDialogBuilder(this)
    .setTitle("Custom Dialog")
    .setView(dialogView)
    .setPositiveButton("Submit", (dialog, which) -> {
        Toast.makeText(this, "Form submitted", Toast.LENGTH_SHORT).show();
    })
    .setNegativeButton("Cancel", null)
    .show();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main_menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.action_settings) {
        Toast.makeText(this, "Settings selected", Toast.LENGTH_SHORT).show();
        return true;
    } else if (id == R.id.action_search) {
        Toast.makeText(this, "Search selected", Toast.LENGTH_SHORT).show();
        return true;
    } else if (id == R.id.action_about) {
        Toast.makeText(this, "About selected", Toast.LENGTH_SHORT).show();
        return true;
    }
    return super.onOptionsItemSelected(item);
}

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

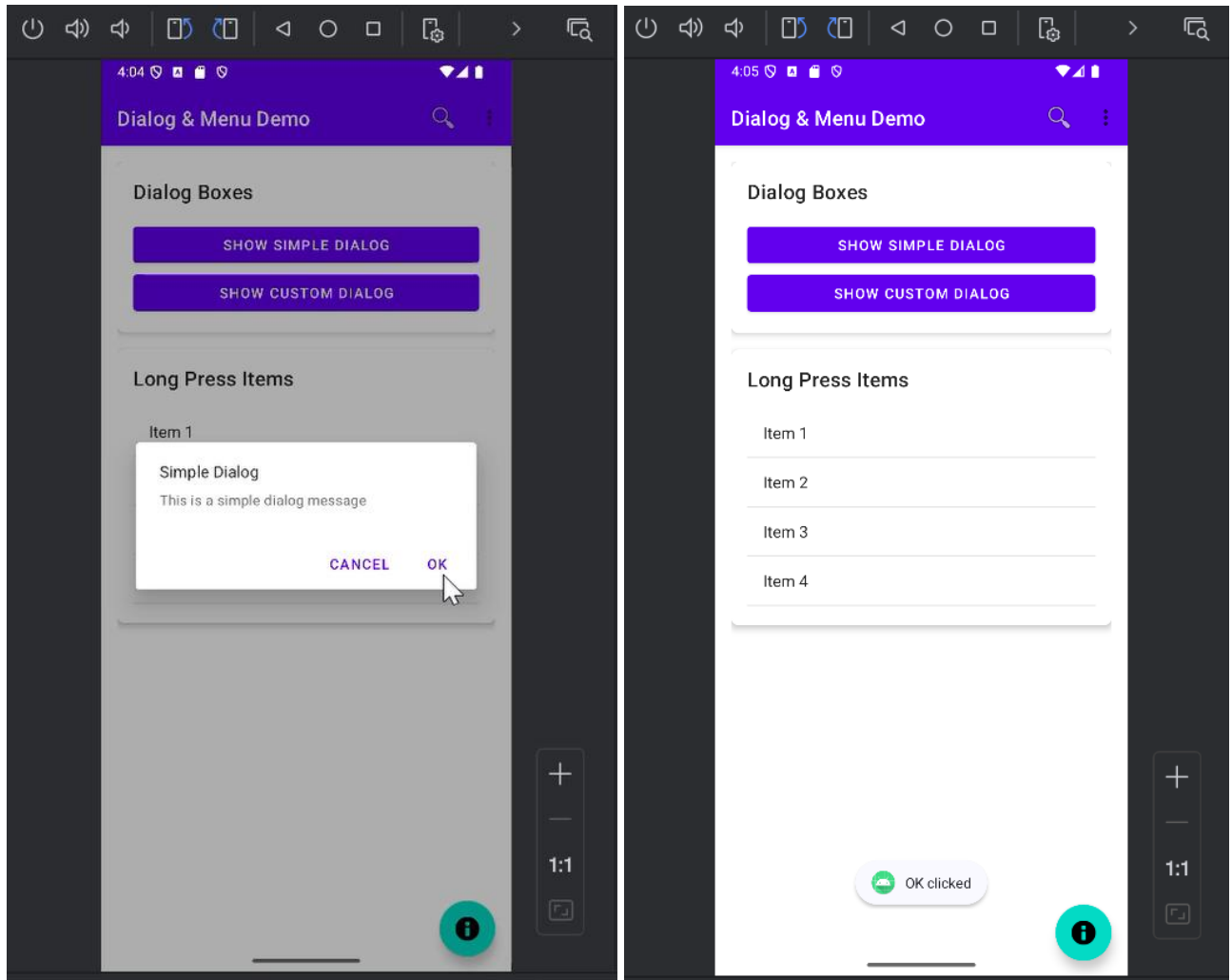
@Override
public boolean onContextItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.action_edit) {
        Toast.makeText(this, "Edit selected", Toast.LENGTH_SHORT).show();
        return true;
    } else if (id == R.id.action_delete) {
        Toast.makeText(this, "Delete selected", Toast.LENGTH_SHORT).show();
        return true;
    } else if (id == R.id.action_share) {
        Toast.makeText(this, "Share selected", Toast.LENGTH_SHORT).show();
        return true;
    }
    return super.onContextItemSelected(item);
}
}
```



Output:

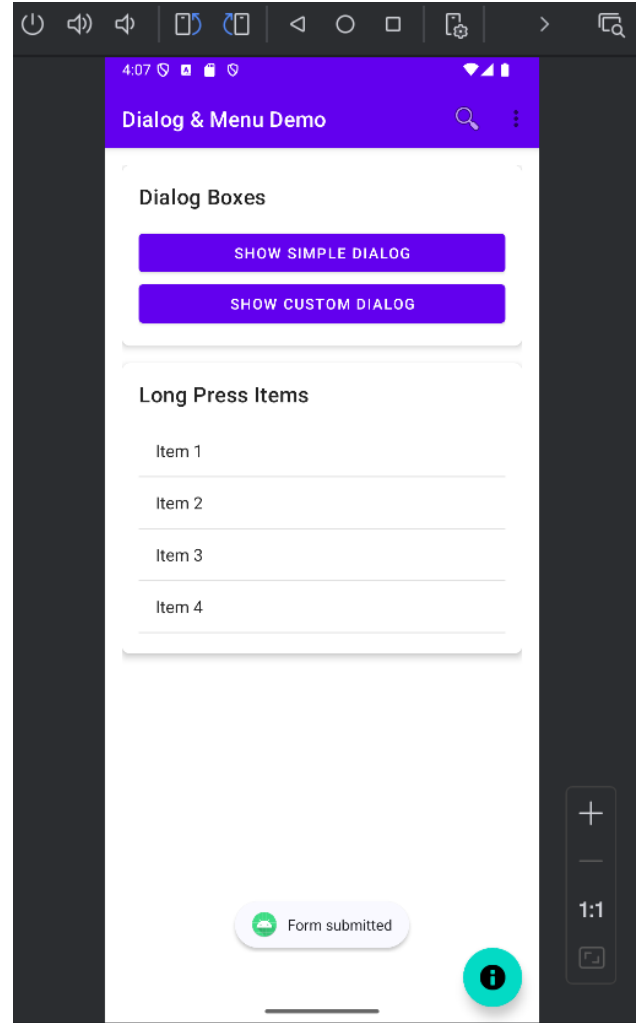
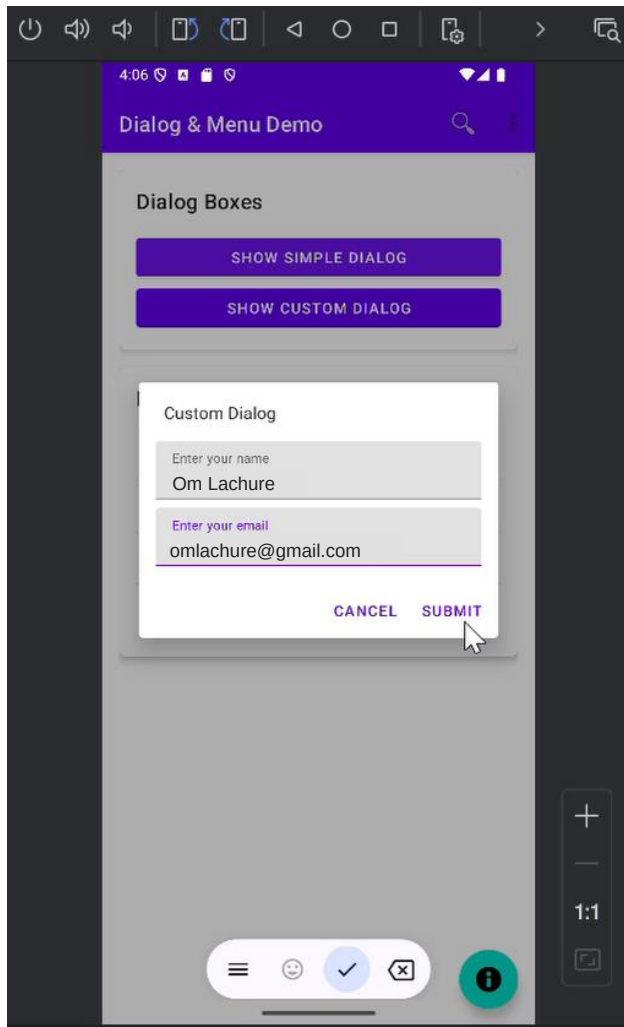
Simple Dialog:



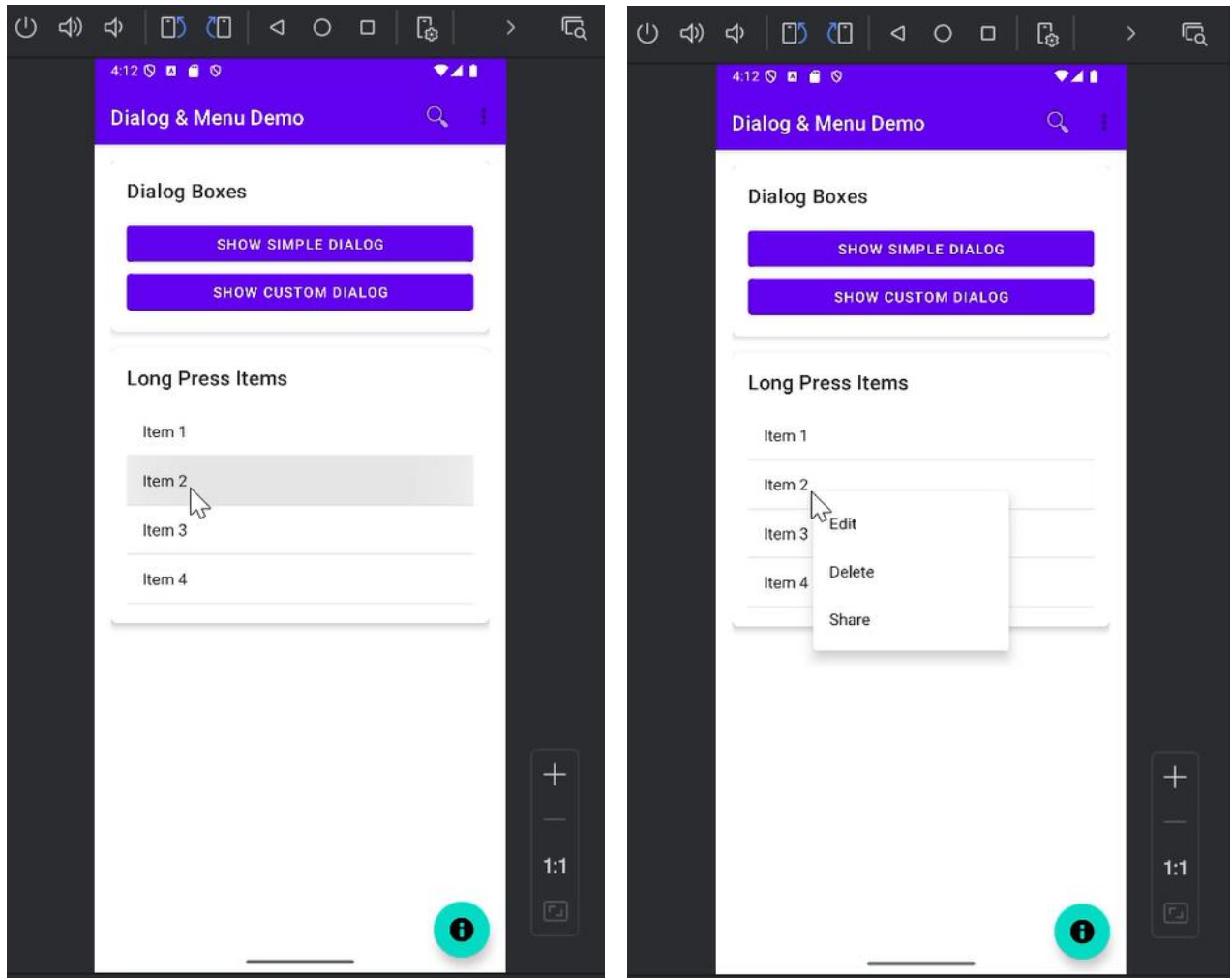


PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE – 411043
Department of Electronics & Telecommunication Engineering

Custom Dialog:



Context Menus:



Conclusion:

.....

.....

.....