|  |
| --- |
| **BATCH AND ROLL NO:** |
| **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:**

**Java Code:**

**Output:**

#### Conclusion:

#### ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………