Introduction to Mobile Application and Development

Mobile Phones

A **mobile phone** (also called **mobile**, **cellular telephone**, **cell phone**, or **hand phone** is an **electronic device** used to make

1. Calls across a wide geographic area.
2. Send Text SMS
3. Can see call records
4. Can capture image
5. Can run music
6. Can browse web etc.

Mobile Phone Applications: Development

1. Different Mobile phone vendor companies use different platforms for developing applications for their own platform.

1. We'll discuss about a open **"Legacy"** platform as an example which is **J2ME**

Java Platform, Micro Edition, or **Java ME:**

1. It is a **Java platform** designed for embedded systems (mobile Devices are one kind of such systems) .
2. Target devices range from industrial controls to mobile phones with Java (KVM support).

   Formerly known as **Java 2 Platform, Micro Edition** (**J2ME**).

Nokia, **Samsung**, **Sony Ericsson** and many other vendors supports J2ME for applications and games development

Development With J2ME

Limitations

1. **Memory Issues**
   1. Vendor Specific Implementation
   2. Insufficient Memory
   3. Inefficient Memory Management
2. **Device Compatibility Issues**
3. **Not enough Access to Core System of the Phone**
4. **Vendor Specific Implementation of KVM**

Smart Phones

1. A **smartphone** is a mobile phone that offers **more advanced computing ability, Features and connectivity** than a contemporary mobile phones.
2. Smartphones may be thought of as **handheld computers integrated with a mobile telephone.**
3. They also provide their own Operating System with Application Development Framework

Smart Phones Platforms

- Android
- iPhone
- BlackBerry
- Palm Pre

Why Android Platform?

- **Technical Advantages**
  - Android is built upon an **open-source platform**, and most of the Android code is released under the free software/open source Apache License.
  - **Rich and easy to integrate feature sets**.
  - Android applications are written in **the Java programming language**, which is a powerful, mature and very widely adopted language.
  - **Easy to learn** than Apple's Objective-C
  - **Inter-Application communication** is easier.
- **Business Advantages**
  - **Android Market** — is much more of an open marketplace than Apple's iTunes App Store.
  - **No delay for approval** of submitted application. So you can reach your users fast.

What is Android?

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

Google describes Android as:

The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation.

Features

- Application framework
- Dalvik virtual machine
- Integrated browser , Optimized graphics
- SQLite , Media support
- GSM Telephony

- Bluetooth, EDGE, 3G, and WiFi
- Camera, GPS, compass, and accelerometer
- Rich development environment

Ex. No: 1 APPLICATION THAT USES GUI COMPONENTS, FONT AND COLOURS

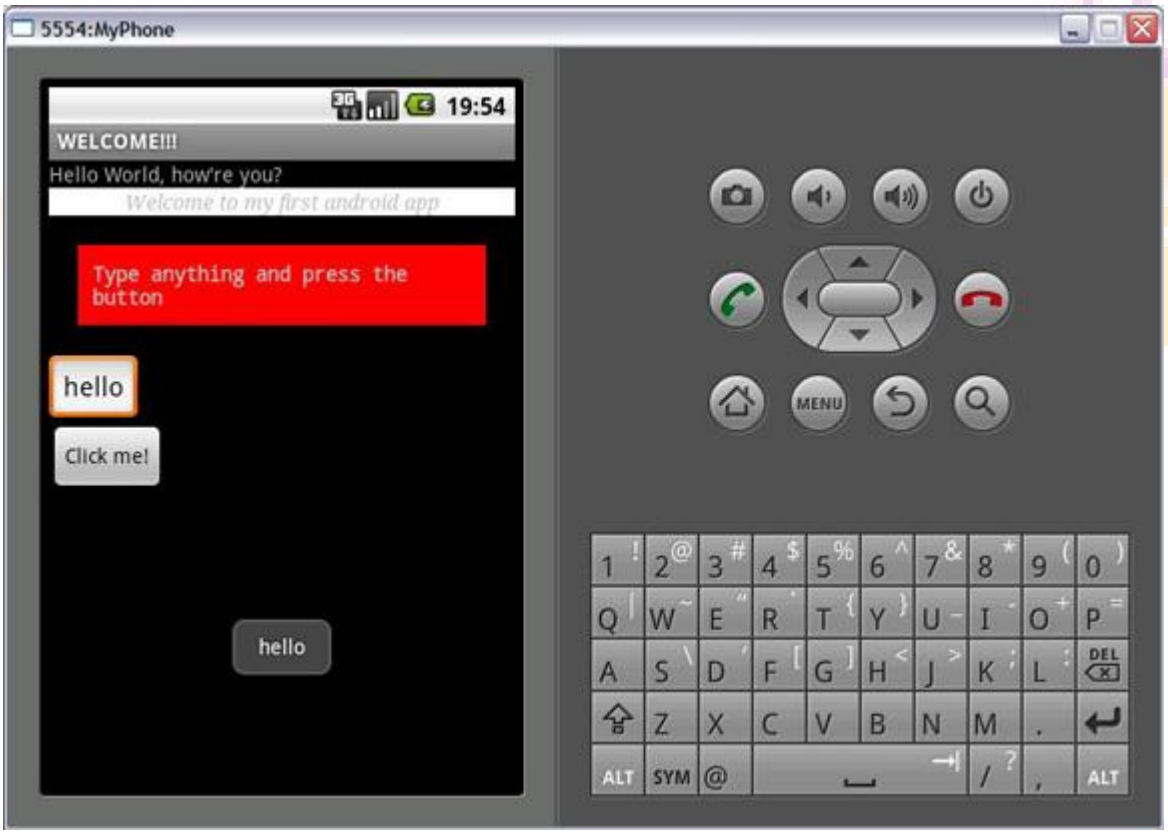
AIM:

Create a mobile application that uses GUI components, Fonts and Colours using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. In layout add necessary components (like edit box, label, button, etc.,) from toolbox.
4. Select the component and modify the colour, font and size from the properties window.
5. For tthe <Button> element, add the android:onClick attribute.
6. Within the MyActivity class, add the sendMessage() method.
7. Write the code for displaying text from edit box to label within sendMessage() method.
8. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:



RESULT:

Thus creation of a mobile application that used GUI components, Fonts and Colours has been developed successfully.

Ex. No: 2 APPLICATION THAT USES LAYOUT MANAGERS AND EVENT LISTENERS

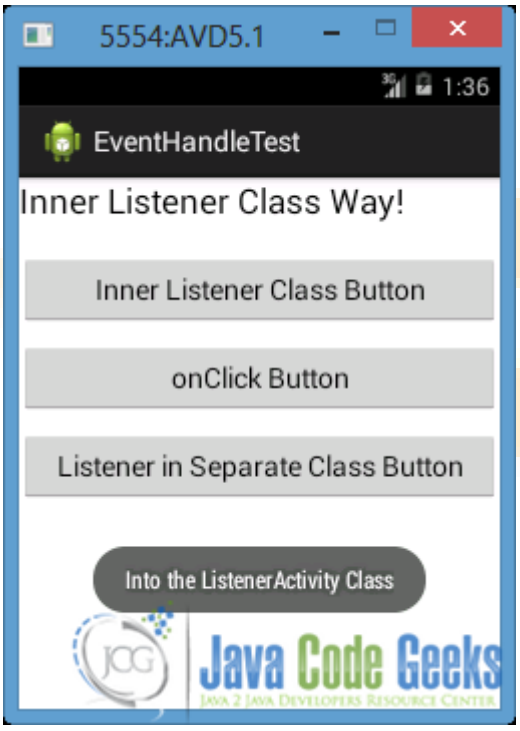
AIM:

Create a mobile application that uses Layout Managers and event listeners using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. In layout add necessary components (like edit box, label, more than 2 buttons, etc.,) from toolbox.
4. Implement the Event Listener Interface Activity in MainActivity class.
5. In onClickListener method, display toast message for clicked events.
6. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:



RESULT:

Thus creation of a mobile application that used Layout Managers and event listeners has been developed successfully.

Ex. No: 3 **NATIVE CALCULATOR APPLICATION**

AIM:

Create a mobile native calculator application using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. Declare a KeypadButton enum class to model the keypad buttons.
4. Create The Keypad Buttons.
5. Display the Keypad in layout.
6. Implement the Business Logic for calculator.
7. Call ProcessKeypadInput Method to process button events.
8. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:

RESULT:

Thus creation of a mobile native calculator application has been developed successfully.

Ex. No: 4 APPLICATION THAT DRAWS BASIC GRAPHICAL PRIMITIVES

AIM:

Create a mobile application that draws basic graphical primitives on the screen using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. Create the ImageView that will be used as a drawing board.
4. Write a Display class to get detail about screen.
5. Set OnTouchListener interface to draw screen line by touching ImageView object.
6. Call onTouch method when a touch event is dispatched to a view.
7. Call getAction() to return the kind of action being performed..
8. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:

RESULT:

Thus creation of a mobile application that draws basic graphical primitives on the screen has been developed successfully.

Ex. No: 5 APPLICATION THAT MAKES USE OF DATABASE

AIM:

Create a mobile application that makes use of database using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. In layout add necessary components (like edit box, label, button, etc.,) from toolbox.
4. Create an SQLite database and a table in the database using SQLiteDatabase class from the android.database.sqlite package and the Cursor class from the android.database package.
5. Use db.execSQL() function to execute any SQL command.
6. Implement the Event Listener Interface Activity.
7. Create user defined functions to add, edit and delete the records in database.
8. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:

RESULT:

Thus creation of a mobile application that makes use of database has been developed successfully.

Ex. No: 6 APPLICATION THAT MAKES USE OF RSS FEED

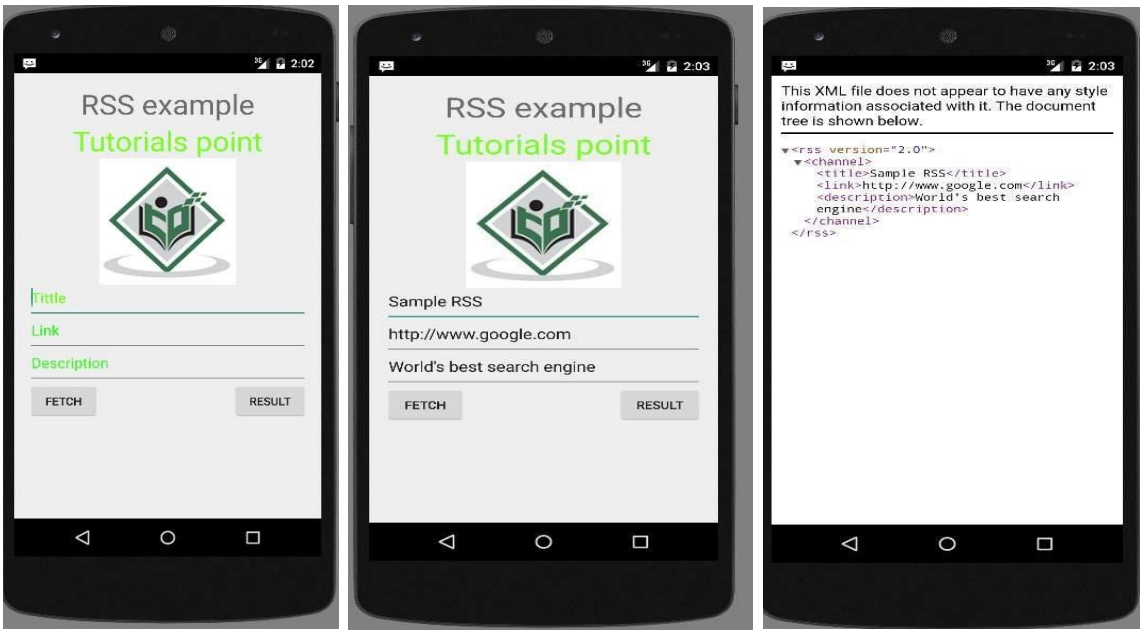
AIM:

Create a mobile application that makes use of RSS Feed using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. Modify src/MainActivity.java file to add necessary code.
4. Modify the res/layout/activity_main to add respective XML components.
5. Create a new java file under src/HandleXML.java to fetch and parse XML data.
6. Create a new java file under src/second.java to display result of XML.
7. Modify AndroidManifest.xml to add necessary internet permission.
8. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:



RESULT:

Thus creation of a mobile application that makes use of RSS Feed has been developed successfully.

Ex. No: 7 APPLICATION THAT IMPLEMENTS MULTI THREADING

AIM:

Create a mobile application that implements Multi threading using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. Modify src/MainActivity.java file to add necessary code.
4. Create an object named hand of Handler class which is to be used for Multithreading.
5. Call the run method of Runnable interface instance with reference run after 1000 milliseconds for the very 1st time.
6. Create updateTime method to decrease the timer text by 1 if the timer is not 0.
7. Modify AndroidManifest.xml to add necessary internet permission.
8. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:

RESULT:

Thus creation of a mobile application that implements Multi threading has been developed successfully.

Ex. No: 8 APPLICATION THAT USES GPS LOCATION INFORMATION

AIM:

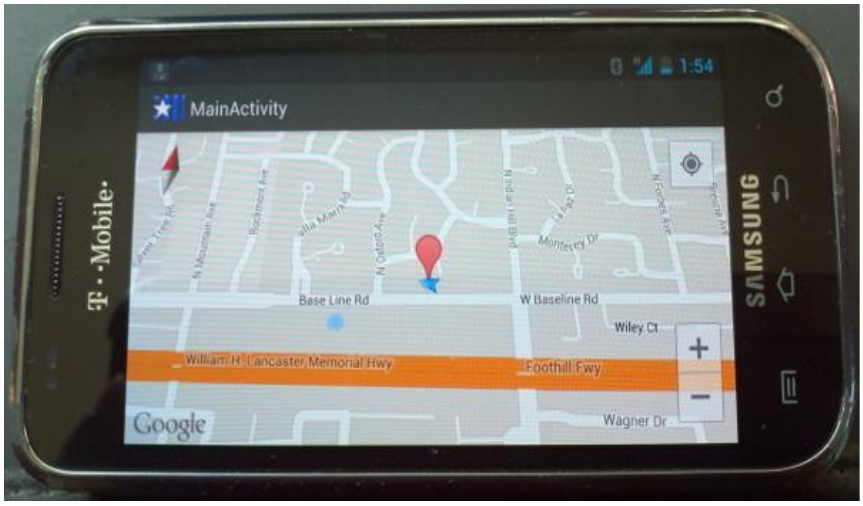
Create a mobile application that uses GPS location information using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. Modify src/MainActivity.java file to add necessary code.
4. Use Google Maps to display the map.
5. Get the object reference for the map we just displayed.
6. Tie up a location listener to this map that responds to location changed events from the location services.
7. Focus our location on the map.
8. Modify AndroidManifest.xml to add necessary internet permission.

9. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:



RESULT:

Thus creation of a mobile application that uses GPS location information has been developed successfully.

Ex. No: 9 APPLICATION THAT WRITES DATA TO THE SD CARD

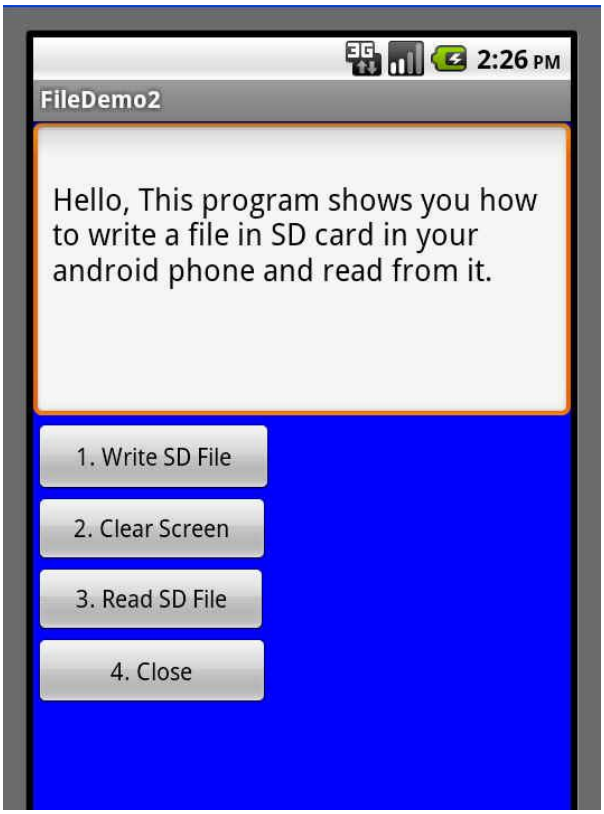
AIM:

Create a mobile application that writes data to the SD card using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. In layout add necessary components (like edit box, label, button, etc.,) from toolbox.
4. Write the code to read edit box and to save that contents to a file in SD card in save button event.
5. Write a code to read file from SD card in read button event.
6. Modify AndroidManifest.xml to add necessary permission.
7. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:



RESULT:

Thus creation of a mobile application that writes data to the SD card has been developed successfully.

Ex. No: 10 APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE

AIM:

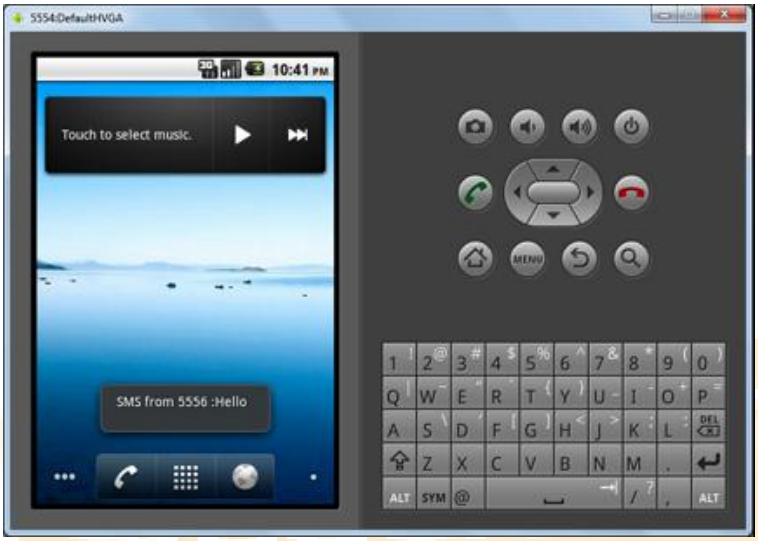
Create a mobile application that creates an alert upon receiving a message using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. In layout add necessary components (like edit box, label, button, etc.,) from toolbox.
4. Initialize the Activity for display.
5. Specify the Permission for Sending the SMS.
6. Create the SMS Receiver to receive the SMS.

7. Display a Toast When the Message is Sent or Recieved.
8. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:



RESULT:

Thus creation of a mobile application that creates an alert upon receiving a message has been developed successfully.

Ex. No: 11 APPLICATION THAT CREATES ALARM CLOCK

AIM:

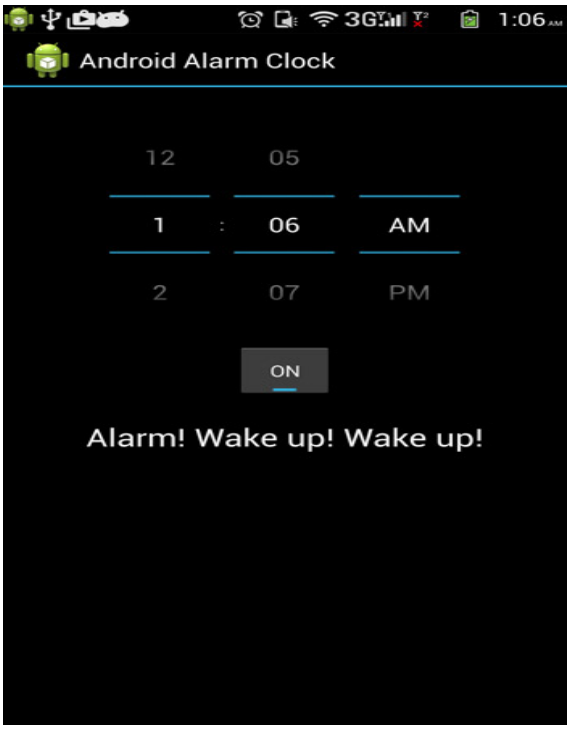
Create a mobile application that creates alarm clock using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. Give uses-permission for WAKE_LOCK in AndroidManifest.xml.
4. Create a TimePicker component followed by a ToggleButton in activity_my.xml.
5. Use the AlarmManager to set the alarm and send notification on alarm trigger.

6. Use AlarmReceiver(WakefulBroadcasReceiver) to receive the alarm trigger on set time.
7. Send a standard notification to the user.
8. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:



RESULT:

Thus creation of a mobile application that creates alarm clock has been developed successfully.

Ex. No: 12 **ANDROID FLASHLIGHT APPLICATION**

AIM:

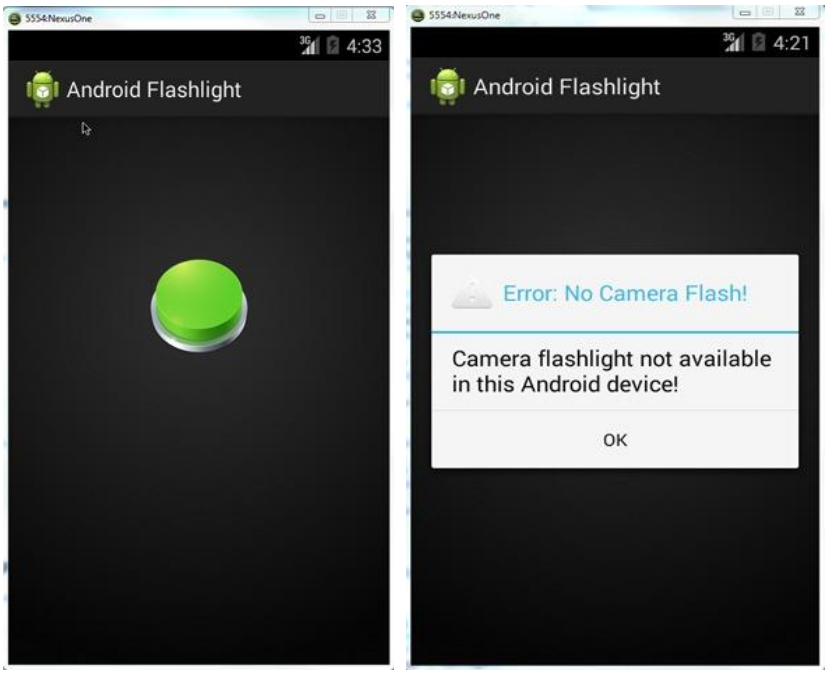
Create a mobile application that uses Flashlight using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.
2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. Create a Permission to use Camera.

4. Create Flashlight Control Layout.
5. Create an Activity to control Flashlight.
6. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:



RESULT:

Thus creation of a mobile application that uses Flashlight has been developed successfully.

Ex. No: 13 **ANDROID SPLASH SCREEN APPLICATION**

AIM:

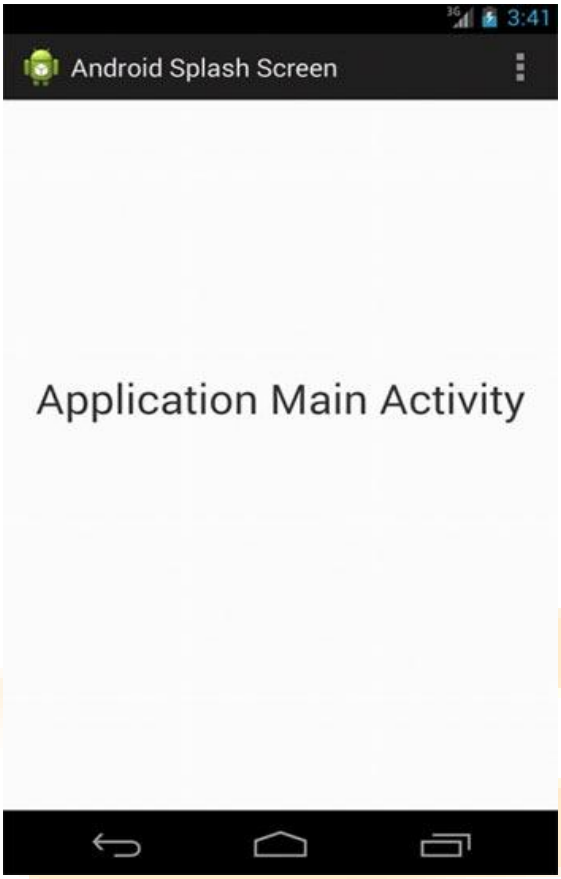
Create a mobile application that uses Splash Screen using Android Studio.

ALGORITHM:

1. Create a new project in Android Studio.

2. Configure your project like Application name, Company Domain, Package Name, and Project Location and select Minimum SDK.
3. Create a Splash screen activity to app's launcher activity.
4. Create image background repeat and logo for Splash screen.
5. Implement the Splash screen activity in Main Activity.
6. To run the project, click Run from tool bar and choose launch emulator from Choose Device Window.

OUTPUT:



RESULT:

Thus creation of a mobile application that uses Splash Screen has been developed successfully.