Codes:

Boot complete file

* for static registration package com.example.april12;

import android.content.BroadcastReceiver; import android.content.Context; import android.content.Intent;

public class BootCompleteReceiver extends BroadcastReceiver { @Override

public void onReceive(Context context, Intent intent) {

if(Intent.*ACTION\_BOOT\_COMPLETED*.equals(intent.getAction()))

{

Intent serviceIntent=new Intent(context, BackgroundService.class);

context.startService(serviceIntent);

}

}

}

Background service .java

* for static registration package com.example.april12;

import android.app.Service;

import android.content.Intent;

import android.os.IBinder;

import android.widget.Toast;

import androidx.annotation.Nullable;

public class BackgroundService extends Service { @Nullable

@Override

public IBinder onBind(Intent intent) { return null;

}

@Override

public int onStartCommand(Intent intent, int flags, int

startId) {

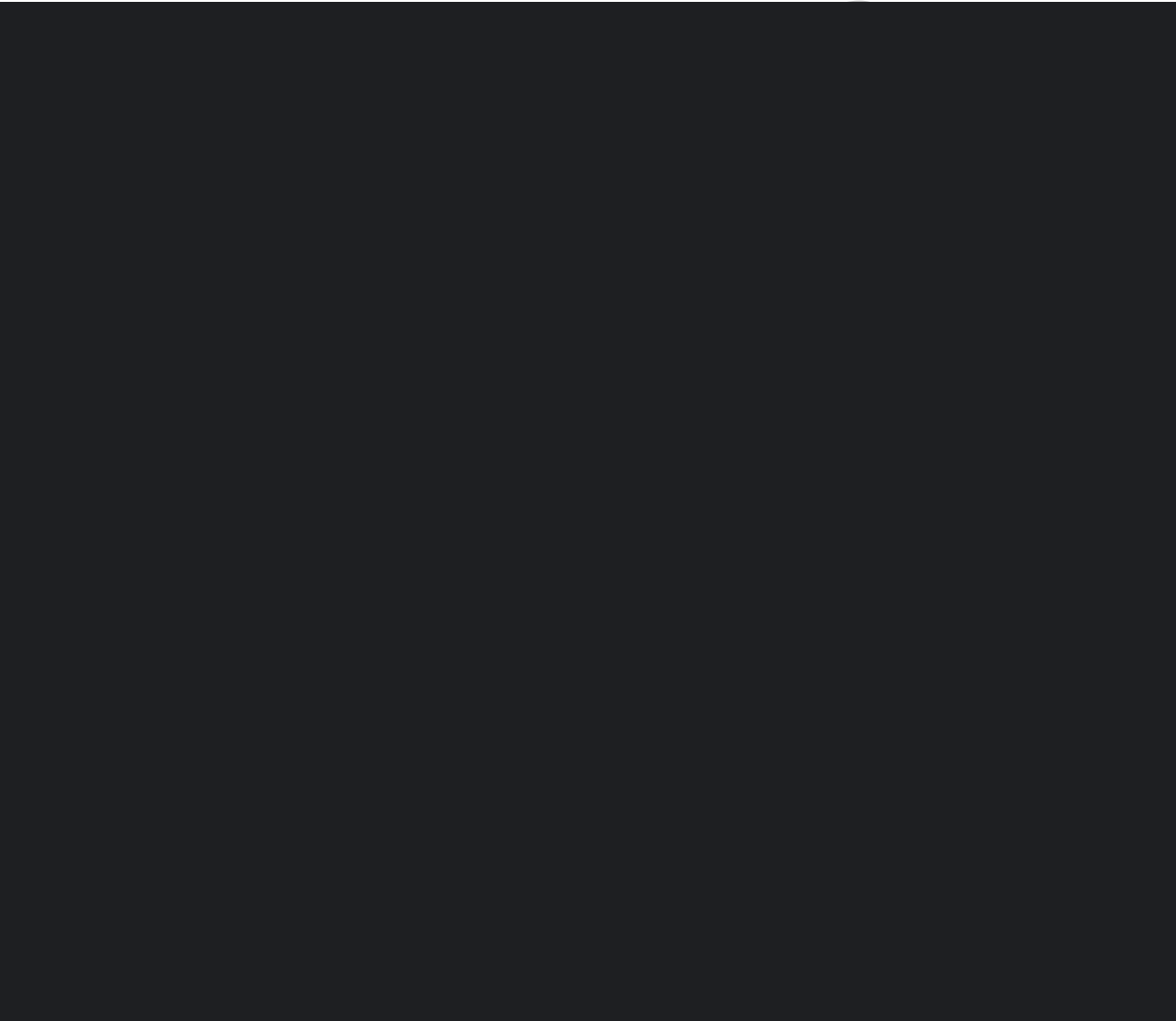
Toast.*makeText*(this,"Service started on boot!",Toast.*LENGTH\_SHORT*).show();

return *START\_STICKY*;

}

}

Main activity .java



package com.example.april12;

import android.content.BroadcastReceiver; import android.content.Context;

import android.content.Intent; import android.content.IntentFilter; import android.os.Bundle;

import android.provider.Settings; import android.widget.Toast;

import androidx.activity.EdgeToEdge;

import androidx.appcompat.app.AppCompatActivity; import androidx.core.graphics.Insets; import androidx.core.view.ViewCompat;

import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity { private BroadcastReceiver airplaneModeReceiver; @Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); EdgeToEdge.*enable*(this); setContentView(R.layout.*activity\_main*);

airplaneModeReceiver=new BroadcastReceiver(){

@Override

public void onReceive(Context context, Intent

intent) {

boolean isAirplaneMode= Settings.System.*getInt*(

context.getContentResolver(),Settings.Global.*AIRPLANE\_MODE\_ON*,0 )!=0;

Toast.*makeText*(context,"Airplane Mode:

"+isAirplaneMode,Toast.*LENGTH\_SHORT*).show();

}

};

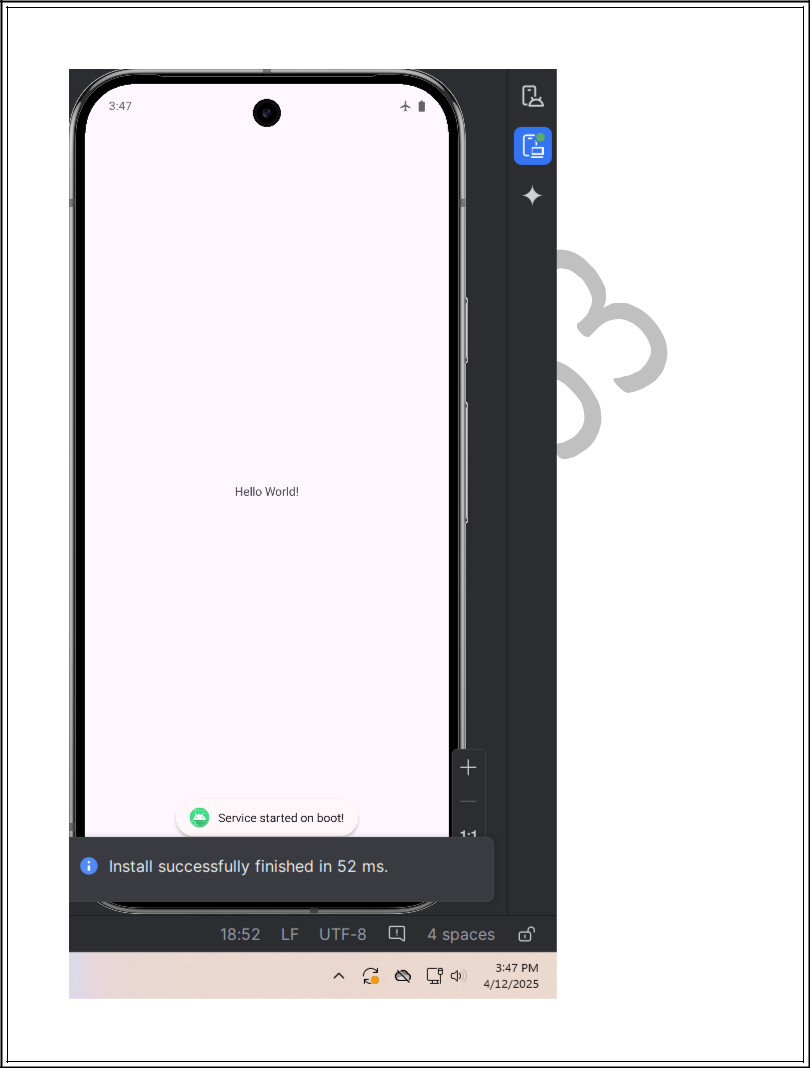
IntentFilter filter =new

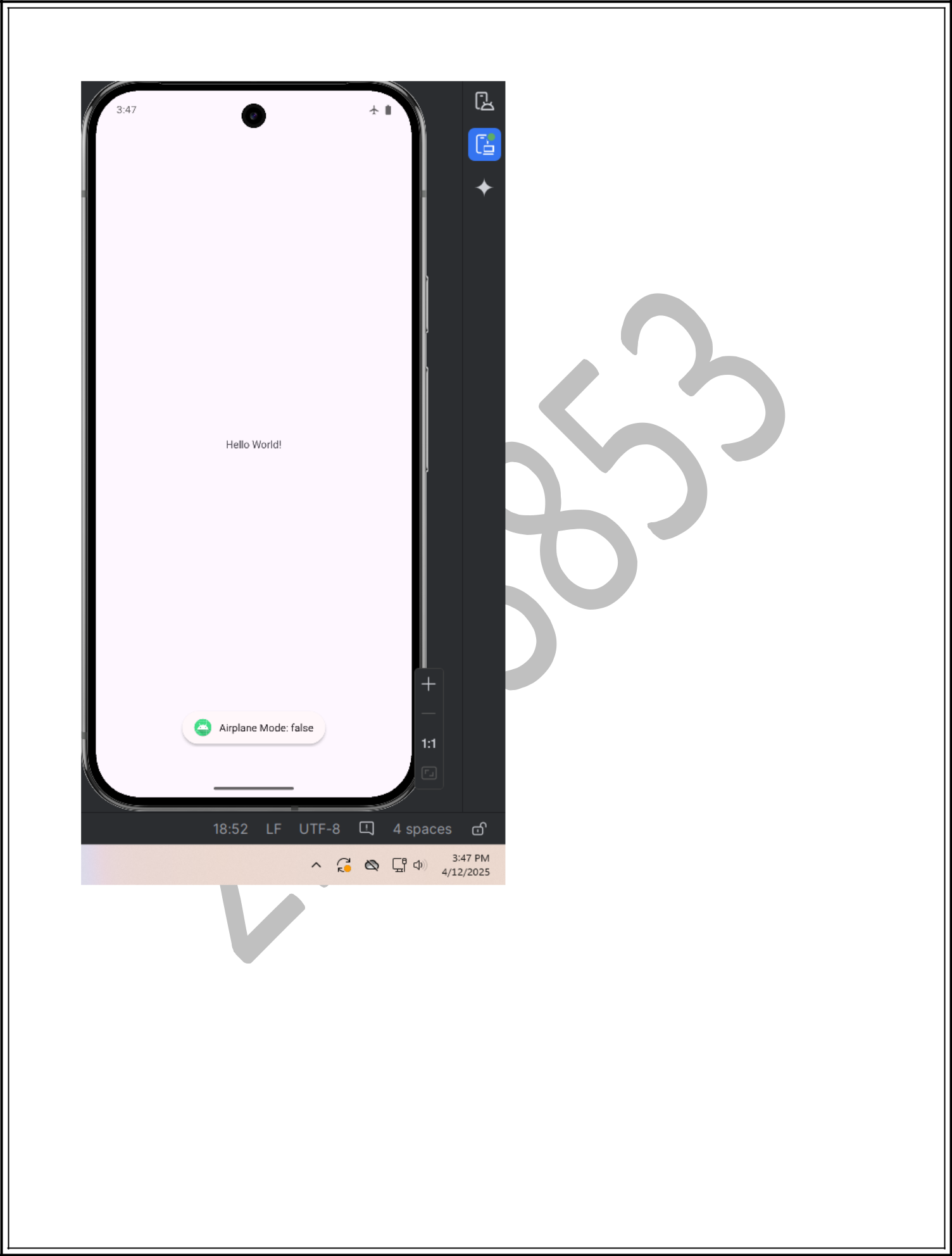
IntentFilter(Intent.*ACTION\_AIRPLANE\_MODE\_CHANGED*);

registerReceiver(airplaneModeReceiver,filter);

}

}



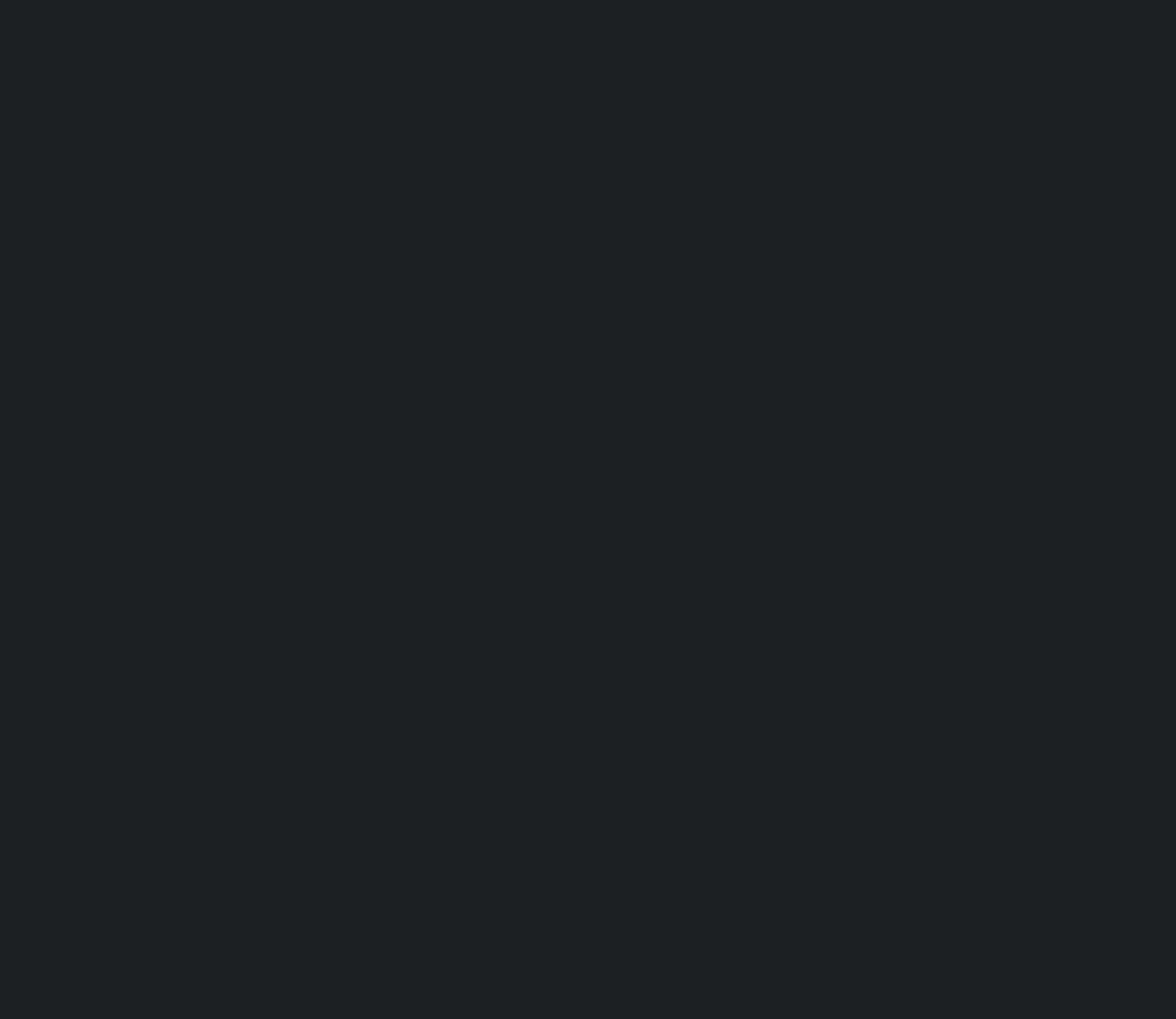


package com.example.april12;

import android.content.BroadcastReceiver; import android.content.Context; import android.content.Intent;

import android.telephony.PhoneStateListener; import android.telephony.TelephonyManager; import android.widget.Toast;

public class RecieverCall extends BroadcastReceiver {



Context context;

@Override

public void onReceive(Context context, Intent intent) {

this.context=context;

Toast.*makeText*(context,"call

recieved",Toast.*LENGTH\_SHORT*).show();

TelephonyManager telephonyManager=(TelephonyManager) context.getSystemService(Context.*TELEPHONY\_SERVICE*);

}

private class MyPhoneStateListener extends PhoneStateListener{

@Override

public void onCallStateChanged(int state, String phoneNumber) {

if(state==1)

{

String msg="New Phone call Event.Incoming Number: "+phoneNumber;

int duration=Toast.*LENGTH\_LONG*;

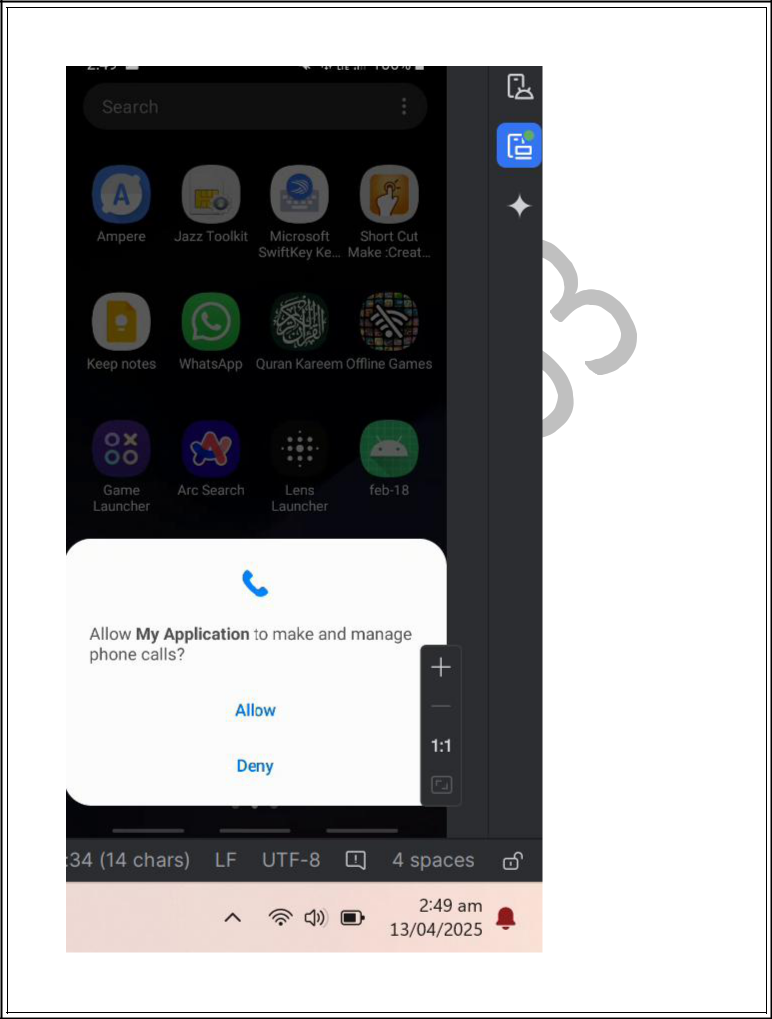
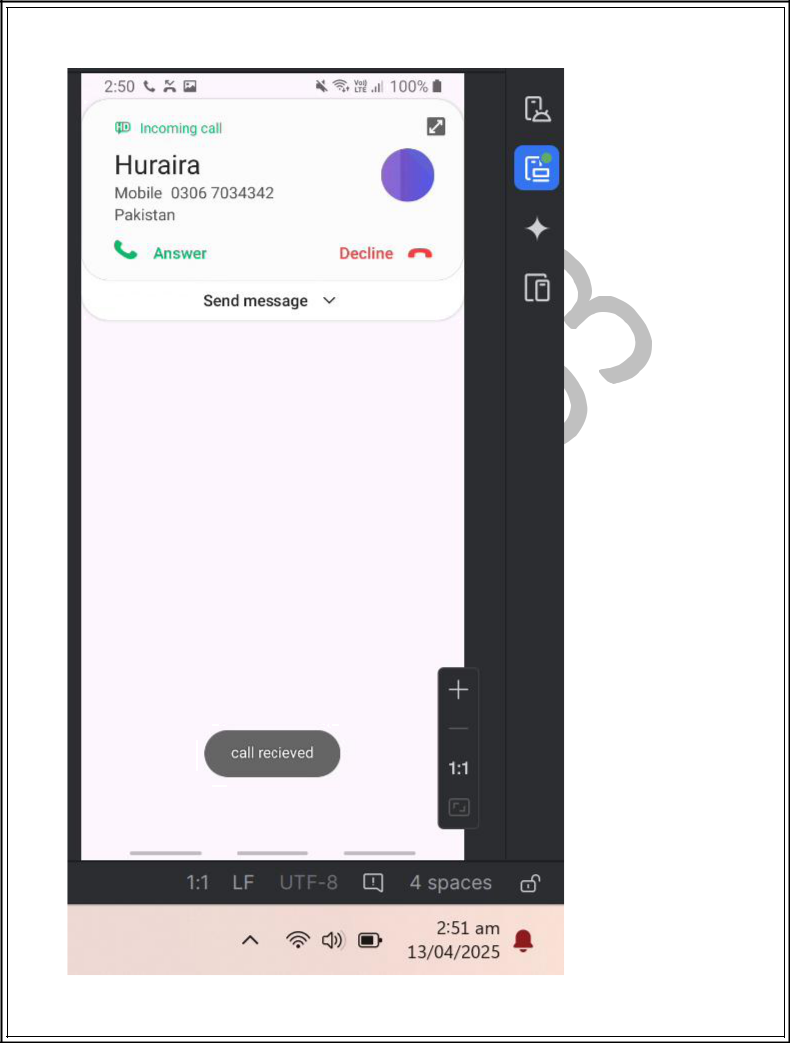
Toast

toast=Toast.*makeText*(context,msg,duration); toast.show();

}

}

}



Home task

package com.example.myapplication;

import android.Manifest;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.content.IntentFilter;

import android.content.pm.PackageManager;

import android.os.BatteryManager;

import android.os.Bundle;

import android.widget.Toast;

import androidx.activity.EdgeToEdge;

import androidx.appcompat.app.AppCompatActivity; import androidx.core.app.ActivityCompat; import androidx.core.content.ContextCompat;

public class MainActivity2 extends AppCompatActivity {

private ReceiverCall receiverCall;

private BatteryStateReceiver batteryStateReceiver;

private static final int *REQUEST\_PHONE\_STATE\_PERMISSION* = 1;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

EdgeToEdge.*enable*(this);

setContentView(R.layout.*activity\_main2*);

receiverCall = new ReceiverCall();

* Initialize the BatteryStateReceiver instance batteryStateReceiver = new BatteryStateReceiver();

checkAndRequestPermission();

}

//Method to check and request READ\_PHONE\_STATE permission

private void checkAndRequestPermission() {

if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_PHONE\_STATE*) != PackageManager.*PERMISSION\_GRANTED*) {

* + Permission is not granted, request it ActivityCompat.*requestPermissions*(

this,

new String[]{Manifest.permission.*READ\_PHONE\_STATE*},

*REQUEST\_PHONE\_STATE\_PERMISSION*

);

} else {

registerPhoneStateReceiver();

}

}

// Method to register the ReceiverCall for PHONE\_STATE changes

private void registerPhoneStateReceiver() {

IntentFilter filter = new IntentFilter("android.intent.action.PHONE\_STATE"); registerReceiver(receiverCall, filter);

}

@Override

public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) { super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if (requestCode == *REQUEST\_PHONE\_STATE\_PERMISSION*) {

if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {

registerPhoneStateReceiver();

} else {

System.*out*.println("READ\_PHONE\_STATE permission denied.");

}

}

}

@Override

protected void onStart() {

super.onStart();

if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_PHONE\_STATE*)

* PackageManager.*PERMISSION\_GRANTED*) { registerPhoneStateReceiver();

}

* Register the battery state receiver IntentFilter batteryFilter = new IntentFilter(); batteryFilter.addAction(Intent.*ACTION\_BATTERY\_LOW*); batteryFilter.addAction(Intent.*ACTION\_BATTERY\_OKAY*); batteryFilter.addAction(Intent.*ACTION\_BATTERY\_CHANGED*); registerReceiver(batteryStateReceiver, batteryFilter);

}

@Override

protected void onPause() {

super.onPause();

* Unregister the phone state receiver to avoid memory leaks unregisterReceiver(receiverCall);
* Unregister the battery state receiver unregisterReceiver(batteryStateReceiver);

}

// Inner class to handle battery state changes

private class BatteryStateReceiver extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) { String action = intent.getAction();

if (action != null) {

switch (action) {

case Intent.*ACTION\_BATTERY\_LOW*:

// Show a Toast when the battery is critically low

Toast.*makeText*(context, "Battery is critically low!", Toast.*LENGTH\_LONG*).show(); break;

case Intent.*ACTION\_BATTERY\_OKAY*:

// Show a Toast when the battery recovers from being low

Toast.*makeText*(context, "Battery is okay now.", Toast.*LENGTH\_LONG*).show(); break;

case Intent.*ACTION\_BATTERY\_CHANGED*:

// Extract battery level and charging status

int level = intent.getIntExtra(BatteryManager.*EXTRA\_LEVEL*, -1); int scale = intent.getIntExtra(BatteryManager.*EXTRA\_SCALE*, -1); float batteryPct = level / (float) scale \* 100;

boolean isCharging = intent.getIntExtra(BatteryManager.*EXTRA\_STATUS*, -1)

* + BatteryManager.*BATTERY\_STATUS\_CHARGING*;
* Show a Toast with the current battery percentage and charging status String chargingStatus = isCharging ? " (Charging)" : " (Not Charging)"; Toast.*makeText*(context, "Battery: " + (int) batteryPct + "%" + chargingStatus,

Toast.*LENGTH\_SHORT*).show();

break;

}

}

}

}

}

A screenshot of a phone

AI-generated content may be incorrect.