**10. DEVELOPING STAND-ALONE DEBUGGING**

**AIM:**

To developing stand-alone debugging using Mobile Application.

**ALGORITHM:**

**Step 1**: Create a new layout file named activity\_main.xml in the res/layout directory.

**Step 2**: Define a LinearLayout as the root layout with vertical orientation, allowing for

 stacked UI elements. Set the layout width and height to match the parent.

**Step 3**: Add a TextView at the top of the layout to serve as a title for the debugging tool. Set

 its text size, style, and center it within the layout.

**Step 4**: Include an EditText where users can input their debug messages. Set a hint to guide

 users and ensure it is wide enough to fill the available space.

**Step 5**: Add a Button below the EditText that users can click to send their input to the

 debugger. Center the button and provide margin for spacing.

**Step 6**: Insert another TextView below the button to display output messages, such as

 confirmation that the input was sent to the debugger. Set an appropriate text size for

 visibility.

**Step 7**: In MainActivity.java, initialize the UI components in the onCreate() method. Set the

 OnClickListener for the button to trigger a method (handleDebugMessage()) that

 retrieves the input message, logs it using Log.d(), updates the output TextView with a

 confirmation message, and clears the input field for new entries.

**Step 8**: Stop the Process.

**CODING:**

**activity\_main.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="match\_parent"

    android:orientation="vertical"

    android:padding="16dp">

    <TextView

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:text="Stand Alone Debugging Tool"

        android:textSize="24sp"

        android:textStyle="bold"

        android:layout\_gravity="center" />

    <EditText

        android:id="@+id/debugInputEditText"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:hint="Enter your debug message"

        android:layout\_marginTop="20dp" />

    <Button

        android:id="@+id/sendButton"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:text="Send to Debugger"

        android:layout\_gravity="center"

        android:layout\_marginTop="20dp"/>

    <TextView

        android:id="@+id/debugOutputTextView"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_marginTop="20dp"

        android:textSize="18sp"/>

</LinearLayout>

**AndroidManifest.xml :**

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

    package="com.example.standalonedebugging">

    <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.AppCompat.Light.DarkActionBar">

        <activity

            android:name=".MainActivity"

            android:exported="true"> <!-- Specify exported attribute -->

            <intent-filter>

                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />

            </intent-filter>

        </activity>

    </application>

</manifest>

**MainActivity.java :**

package com.example.standalonedebugging;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText debugInputEditText;

    private Button sendButton;

    private TextView debugOutputTextView;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        debugInputEditText = findViewById(R.id.debugInputEditText);

        sendButton = findViewById(R.id.sendButton);

        debugOutputTextView = findViewById(R.id.debugOutputTextView);

        sendButton.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                handleDebugMessage();

            }

        });

    }

    private void handleDebugMessage() {

        String debugMessage = debugInputEditText.getText().toString();

        if (!debugMessage.isEmpty()) {

            Log.d("DebugMessage", debugMessage); // Log the message for debugging

            debugOutputTextView.setText("Sent to debugger: " + debugMessage);

            debugInputEditText.setText(""); // Clear the input field

        }

    }

}

**OUTPUT :**

A screenshot of a phone

Description automatically generated

**RESULT:**

The above program has been successfully executed.