

MCWC(2170701)

Assignment 2 (Android App)

Name: Yogesh Bavishi

Division: E Sem: 7th

Enrollment No.: 170200107003

*MainActivity.java

```
package com.example.texteditor;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.net.URI;
import java.util.regex.*;
import android.Manifest;
import android.annotation.SuppressLint;
import android.app.ActionBar;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.DocumentsProvider;
import android.provider.MediaStore;
import android.provider.OpenableColumns;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {

    public static int PICK_FILE = 1;
    private EditText mainEditor;
    private Intent openFileIntent;
    private static final int STORAGE_PERMISSION_CODE = 101;
```

```

//    public static final String ACTION_OPEN_DOCUMENT = "open_file"
private static Uri file_path;
private String filePath;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //check permissions
    checkPermissionStatus(new String[] {Manifest.permission.READ_EXTERNAL_STORAGE,
Manifest.permission.WRITE_EXTERNAL_STORAGE});
    //get ids
    mainEditor = findViewById(R.id.editText1);

}

private void checkPermissionStatus(String[] permission) {

    if (ContextCompat.checkSelfPermission(MainActivity.this, permission[1] )
        == PackageManager.PERMISSION_DENIED){
        Toast.makeText(MainActivity.this, "Without Storage Permission you won't be
able to SAVE files!!!", Toast.LENGTH_SHORT).show();

        ActivityCompat.requestPermissions(MainActivity.this,permission,STORAGE_PERMISSION_CODE);
    }
    else {
        Toast.makeText(MainActivity.this, "Permission already granted",
        Toast.LENGTH_SHORT).show();
    }
}

//menu items creation
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.main_menu, menu);
    return true;
}

//menu item selected
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.open_file:
            //newGame();
            openFile();
            return true;
        case R.id.save_file:
            saveFile();

```

```

        return true;
    case R.id.close_file:
        AlertDialog.Builder alert_close = new
AlertDialog.Builder(MainActivity.this);
        alert_close.setTitle("TextEditor:");
        alert_close.setMessage("Unsaved work will be lost. Are you sure to
exit?");

        alert_close.setNegativeButton("No", null);
        alert_close.setPositiveButton("Yes", new
DialogInterface.OnClickListener(){
            @Override
            public void onClick(DialogInterface diag1,int id){
                finish();
            }
        });
        alert_close.setNeutralButton("Save File",new
DialogInterface.OnClickListener(){
            @Override
            public void onClick(DialogInterface diag1,int id){
                Toast.makeText( MainActivity.this,"Saving file before
exit!!!",Toast.LENGTH_SHORT).show();
                saveFile();
            }
        });
        alert_close.show();
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }

}

//file open option
private void openFile() {
    openFileIntent = new Intent(Intent.ACTION_GET_CONTENT);
    openFileIntent.setType("text/plain");
    startActivityForResult(openFileIntent,PICK_FILE);
    setFileName();

}

//FileName set for ActionBar when opened/saved
private void setFileName() {
//    ActionBar act = getActionBar();
//    act.setTitle(file_path.getLastPathSegment().toString());

}

//openfile subfunction

```

```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent
data) {
    if(requestCode==PICK_FILE){
        if(resultCode==RESULT_OK){

            filePath=data.getData().getPath();
            file_path=data.getData();
            String str=data.getData().getLastPathSegment();
            Toast.makeText(MainActivity.this,filePath+" Opened!
",Toast.LENGTH_SHORT).show();
            String fileContent = readTextFile(file_path);
            mainEditor.setText(fileContent);

        }
    }
}

//openfile subfuntion2
private String readTextFile(Uri file_path1) {
    BufferedReader reader = null;
    StringBuilder builder = new StringBuilder();

    try {
        reader = new BufferedReader(new
InputStreamReader(getContentResolver().openInputStream(file_path1)));
        String line = "";
        while ((line = reader.readLine()) != null)
        {
            builder.append(line+"\n");
        }
        reader.close();
    }
    catch (IOException e) { e.printStackTrace(); }
    return builder.toString();
}

//permission request
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions,@NonNull int[] grantResults){
    super.onRequestPermissionsResult(requestCode,permissions,grantResults);
    if (requestCode == STORAGE_PERMISSION_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(MainActivity.this,"Storage Permission
Granted",Toast.LENGTH_SHORT).show();
        }
        else {
            Toast.makeText(MainActivity.this,"Storage Permission
Denied",Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

    }
}

//file save option
private void saveFile(){
    final TextView messageSaveFile = (TextView)findViewById(R.id.textView_save);
    final AlertDialog.Builder alert = new AlertDialog.Builder(MainActivity.this);
    View mView = getLayoutInflater().inflate(R.layout.custom_dialog,null);
    final EditText txt_inputText = (EditText) mView.findViewById(R.id.editText_save);
    Button btn_cancel = (Button)mView.findViewById(R.id.btnCancel_save);
    Button btn_okay = (Button)mView.findViewById(R.id.btnSave_save);
    alert.setView(mView);
    final AlertDialog alertDialog = alert.create();
    alertDialog.setCanceledOnTouchOutside(false);
    btn_cancel.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            alertDialog.dismiss();
        }
    });
    btn_okay.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            filePath=txt_inputText.getText().toString();
            try {
                writeFileToStorage(filePath);
            } catch (IOException e) {
                e.printStackTrace();
            }
            alertDialog.dismiss();
        }
    });
    alertDialog.show();
    setFileName();
}

private void writeFileToStorage(String path) throws IOException {
    String state = Environment.getExternalStorageState();

    if(Environment.MEDIA_MOUNTED.equals(state)){
        File root = Environment.getExternalStorageDirectory();
        File folderPath = new File(root.getAbsolutePath()+"/TextEditor");
        //to be devided into folder path and file name using pattern in future
release
        //also directory chooser to be added
        if(!folderPath.exists()){
            folderPath.mkdir();
            Toast.makeText(getApplicationContext(), "FOLDER
CREATED", Toast.LENGTH_SHORT).show();

```

```

    }
    File file = new File(folderPath, filePath);
    if(!file.exists()){
        file.createNewFile();
        Toast.makeText(getApplicationContext(),filePath+" file
saved",Toast.LENGTH_SHORT).show();
    }
    String textData = mainEditor.getText().toString();
    try {
        FileOutputStream fos = new FileOutputStream(file);
        fos.write(textData.getBytes());
        Toast.makeText(getApplicationContext(),"data to file written and
saved",Toast.LENGTH_SHORT).show();
        fos.close();
    } catch (Exception e) {
        mainEditor.setText(mainEditor.getText() + " \n\n" +e.toString());
        Toast.makeText(getApplicationContext(),"Toast of
exception",Toast.LENGTH_SHORT).show();
    }

}
else{
    Toast.makeText(getApplicationContext(),"File not saving
error!!!",Toast.LENGTH_LONG).show();
}

}

}

```

*activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

```

```

<EditText
    android:id="@+id/editText1"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:ems="10"
    android:gravity="start|top"
    android:inputType="textMultiLine"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.466"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.409" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

*AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.texteditor">
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <application
        android:requestLegacyExternalStorage="true"
        android:allowBackup="true"
        android:icon="@drawable/ic_texteditor"
        android:label="@string/app_name"
        android:roundIcon="@drawable/ic_texteditor"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

```



```
</manifest>
```

*main_menu.xml

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

    <item
        android:id="@+id/close_file"
        android:icon="@drawable/ic_close"
        android:title="@string/close_file"
        android:tooltipText="@string/close_file"
        app:showAsAction="always"/>
    <item
        android:id="@+id/save_file"
        android:icon="@drawable/ic_save"
        android:title="@string/save_file"
        android:tooltipText="@string/save_file"
        app:showAsAction="always" />
    <item
        android:id="@+id/open_file"
        android:icon="@drawable/ic_open"
        app:showAsAction="always"
        android:title="@string/open_file"
        android:tooltipText="@string/open_file"
        />

</menu>
```

*custom_dialog.xml (for save menu)

```
<?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">

        <TextView
            android:id="@+id/textView_save"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/save_path_sdcard"
            android:textAlignment="center"
            android:textSize="22sp" />

        <EditText
            android:id="@+id/editText_save"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:hint="@string/enter_path"
            android:importantForAutofill="no"
            android:inputType="text"
            android:textSize="20sp" />
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp">
            <Button
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:backgroundTint="@color/colorPrimary"
                android:text="Cancel"
                android:textColor="#ffffff"
                android:textSize="18sp"
                android:layout_weight="1"
                android:id="@+id/btnCancel_save"/>
            <Button
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:backgroundTint="@color/colorPrimary"
                android:text="Save"
                android:textColor="#ffffff"
                android:textSize="18sp"
                android:layout_weight="1"
                android:id="@+id/btnSave_save"/>
        </LinearLayout>
    </LinearLayout>

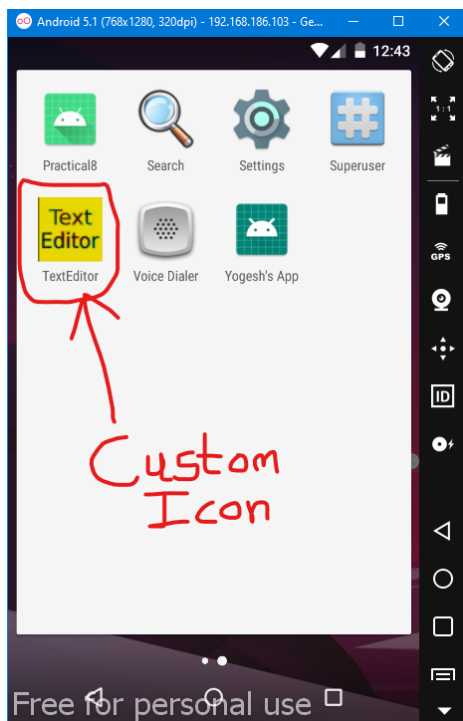
```

*strings.xml

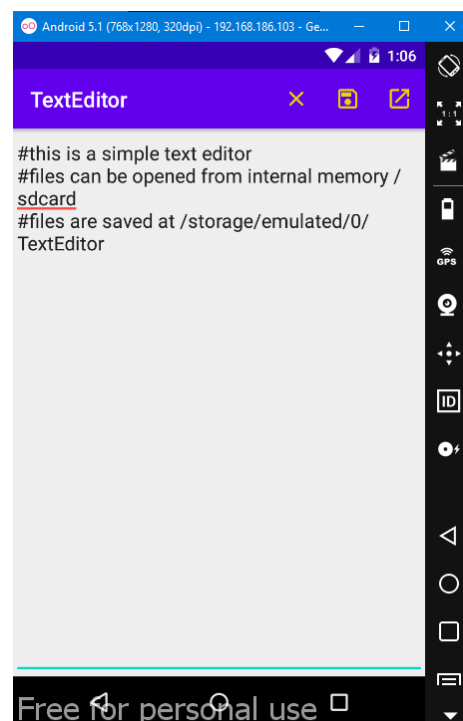
```
<resources>
    <string name="app_name">TextEditor</string>
    <string name="save_file">Save</string>
    <string name="open_file">Open</string>
    <string name="close_file">Close</string>
    <string name="save_path_sdcard">Save Path sdcard/:</string>
    <string name="enter_path">Enter path</string>
</resources>
```

*Output:

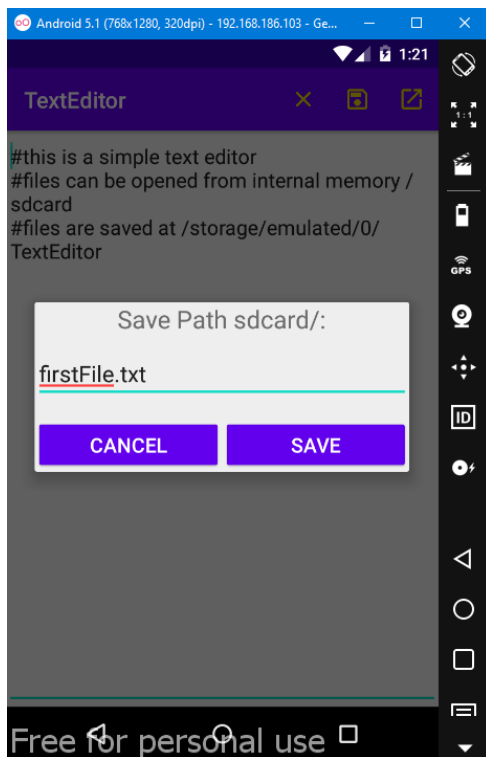
-Custom icon



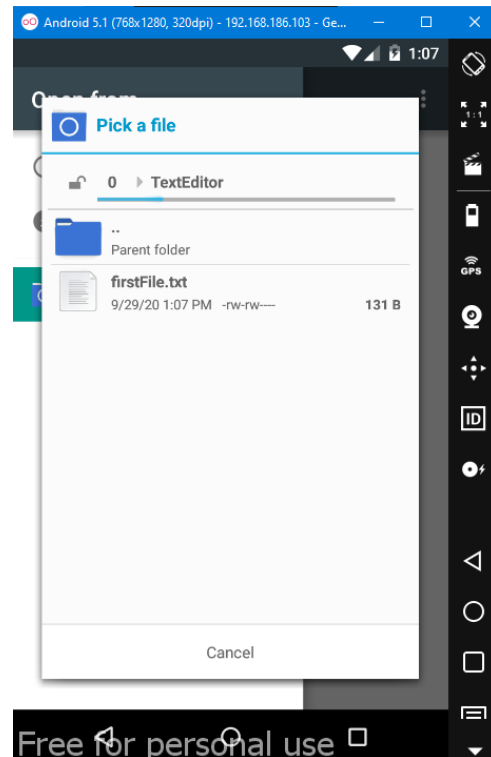
-Editing Text



-Saving File



-Opening File



-Closing File

