EXP 7: SD Card File Transfer App

AIM

To develop an Android app that moves a selected file from internal storage to the SD Card using permissions. The app asks for permission at runtime and performs the move operation with proper feedback.

ALGORITHM

1. Start the app.
2. Ask for Read/Write permissions.
3. User selects a file using a file picker.
4. After selection, the app shows the file name.
5. User taps 'Move to SD Card'.
6. The app copies the file to /MySDCardFiles/ directory on the SD Card.
7. Displays a success message.
8. End

CODE

MainActivity.kt

```
package com.example.sdcardapp

import android.Manifest
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Bundle
import android.os.Environment
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import java.io.*

class MainActivity : AppCompatActivity() {

    private lateinit var selectFileBtn: Button
    private lateinit var moveFileBtn: Button
    private lateinit var statusText: TextView
```

```
    private var selectedFileUri: Uri? = null
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        selectFileBtn = findViewById(R.id.selectFile)
        moveFileBtn = findViewById(R.id.moveFile)
        statusText = findViewById(R.id.status)

        checkPermissions()

        selectFileBtn.setOnClickListener { pickFile() }
        moveFileBtn.setOnClickListener {
            if (selectedFileUri != null) {
                moveFileToSDCard(selectedFileUri!!)
            } else {
                Toast.makeText(this, "Please select a file first",
Toast.LENGTH_SHORT).show()
            }
        }
    }

    private fun checkPermissions() {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED ||
            ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(this,
                arrayOf(Manifest.permission.READ_EXTERNAL_STORAGE,
Manifest.permission.WRITE_EXTERNAL_STORAGE), 100)
        }
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions:
Array<out String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if (requestCode == 100) {
            if (grantResults.all { it == PackageManager.PERMISSION_GRANTED
}) {
                Toast.makeText(this, "Permissions granted",
Toast.LENGTH_SHORT).show()
            } else {
                Toast.makeText(this, "Permissions required",
```

```
Toast.LENGTH_SHORT).show()
            }
        }
    }

    private fun pickFile() {
        val intent = Intent(Intent.ACTION_GET_CONTENT)
        intent.type = "*/*"

        startActivityForResult(intent, 101)
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?)
{
        super.onActivityResult(requestCode, resultCode, data)
        if (requestCode == 101 && resultCode == Activity.RESULT_OK) {
            selectedFileUri = data?.data
            if (selectedFileUri != null) {
                statusText.text = "File selected: ${selectedFileUri!!.lastPathSegment}"
                moveFileBtn.isEnabled = true
            } else {
                statusText.text = "No file selected."
            }
        }
    }

    private fun moveFileToSDCard(sourceUri: Uri) {
        try {
            val inputStream = contentResolver.openInputStream(sourceUri)
            val fileName = sourceUri.path?.substringAfterLast("/") ?: "file.txt"

            val sdCardPath =
File(Environment.getExternalStorageDirectory().absolutePath +
"/MySDCardFiles")
            if (!sdCardPath.exists()) sdCardPath.mkdirs()

            val outputFile = File(sdCardPath, fileName)
            val outputStream = FileOutputStream(outputFile)

            inputStream?.copyTo(outputStream)

            inputStream?.close()
            outputStream.close()

            statusText.text = "File moved to: ${outputFile.absolutePath}"
            Toast.makeText(this, "File moved to SD Card!",
Toast.LENGTH_SHORT).show()
```

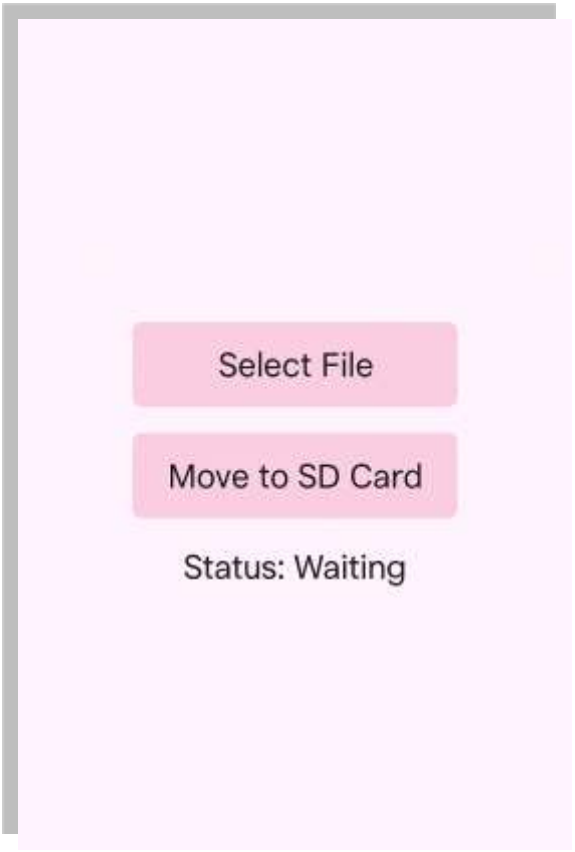
```
            } catch (e: Exception) {
                e.printStackTrace()
                statusText.text = "Error moving file: ${e.message}"
                Toast.makeText(this, "Error moving file",
        Toast.LENGTH_SHORT).show()
            }
        }
    }
```

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:background="#FFF1F8"
 android:padding="24dp"
 android:gravity="center">

 <TextView
  android:text="SD Card File Mover "
  android:textSize="28sp"
  android:textColor="#E91E63"
  android:layout_marginBottom="16dp"
  android:textStyle="bold"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"/>

 <Button
  android:id="@+id/selectFile"
  android:text="Select File"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:backgroundTint="#F8BBD0"
  android:layout_marginBottom="16dp"/>

 <Button
  android:id="@+id/moveFile"
  android:text="Move to SD Card"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:backgroundTint="#F8BBD0"
  android:layout_marginBottom="16dp"
  android:enabled="false"/>
```

```
<TextView
  android:id="@+id/status"
  android:text="Status: Waiting"
  android:textSize="18sp"
  android:textColor="#880E4F"
  android:layout_marginTop="24dp"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"/>

</LinearLayout>
```

OUTPUT:



RESULT

Thus the sd card permission and file transfer app runs perfectly without any issues.