SSN COLLEGE OF ENGINEERING, KALAVAKKAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 7

Name: Jayannthan P T *Dept: CSE 'A'* *Roll No.: 205001049*

Writing to and Reading from the SD Card

Ex. No:7

Title of the Program:

Develop an android application to read the text from the SD Card and Write into the SD Card. To perform this, create two TextViews one for writing the text and save the text into the SD Card once submit button is clicked and another one for Displaying the text that is retrieved from the SD Card.

Objective:

The objective of the FileReadWrite Android App project is to create an application that allows users to write content to a file and read the contents from the file. The app provides a simple user interface with text fields for entering file contents and file names. It also includes buttons to write to and read from files, along with a separate activity to display the contents read from the file.

Algorithm:

1. Create an Android app with two activities: MainActivity and ReadActivity.
2. MainActivity includes EditTexts for file contents and file name, along with buttons for writing to and reading from files.
3. Implement a method (writeToFile) to write user-provided contents to a file specified by the user.
4. Implement a method (readFromFile) to read contents from a file specified by the user.
5. Display a toast message upon successful writing to a file.
6. Transfer the file name to ReadActivity using an Intent.
7. Read the file contents in ReadActivity and display them in a TextView.

Features used:

1. EditText for user input of file contents and file name.
2. Buttons for triggering write and read operations.
3. File I/O operations for writing and reading files.
4. Toast message for user feedback after writing to a file.
5. Intent to pass data between MainActivity and ReadActivity.
6. TextView for displaying file contents in ReadActivity.

Source code:

- MainActivity.java

```
package com.example.exercise7;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        EditText fileContents = (EditText) findViewById(R.id.fileContents);
        EditText fileName = (EditText) findViewById(R.id.fileName);

        Button write = (Button) findViewById(R.id.write);
        write.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String writeFileContents = fileContents.getText().toString();
                String writeFileName = fileName.getText().toString();
                writeToFile(writeFileName, writeFileContents);
            }
        });

        Button read = (Button) findViewById(R.id.read);
        read.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String readFileName = fileName.getText().toString();
                Intent i = new Intent(MainActivity.this, ReadActivity.class);
                i.putExtra("fileName", readFileName);
                startActivity(i);
            }
        });
    }
```

```java
    private void writeToFile(String writeFileName, String writeFileContents) {
        File path = getApplicationContext().getFilesDir();

        try {
            FileOutputStream writer = new FileOutputStream(new File(path, writeFileName));
            writer.write(writeFileContents.getBytes());
            writer.close();
            Toast.makeText(getApplicationContext(), "Written to file: " + path + "/" +
writeFileName,
                    Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

- activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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" android:orientation="vertical"
tools:context=".MainActivity">

    <EditText android:id="@+id/fileContents" android:layout_width="300dp"
android:layout_height="50dp" android:layout_gravity="center"
android:layout_marginTop="150dp" android:text="File Contents" android:textSize="24sp"
android:typeface="monospace" android:textStyle="bold" android:textColor="#000000" />
    <EditText android:id="@+id/fileName" android:layout_width="300dp"
android:layout_height="50dp" android:layout_gravity="center"
android:layout_marginTop="50dp" android:text="File Name" android:textSize="24sp"
android:typeface="monospace" android:textStyle="bold" android:textColor="#000000" />

    <Button android:id="@+id/write" android:layout_width="250dp"
android:layout_height="50dp" android:layout_gravity="center"
android:layout_marginTop="75dp" android:text="Write to File" android:textSize="20sp"
android:typeface="monospace" android:textStyle="bold" />

    <Button android:id="@+id/read" android:layout_width="250dp"
android:layout_height="50dp" android:layout_gravity="center"
android:layout_marginTop="50dp" android:text="Read from File" android:textSize="20sp"
android:typeface="monospace" android:textStyle="bold" />

</LinearLayout>
```

- ReadActivity.java

```java
package com.example.exercise7;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

import java.io.File;
import java.io.FileInputStream;

public class ReadActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_read);

        TextView contentsFromFile = (TextView) findViewById(R.id.contentsFromFile);

        Intent i = getIntent();

        String readFileName = i.getStringExtra("fileName");
        String readFileContents = readFromFile(readFileName);
        contentsFromFile.setText(readFileContents);
    }

    private String readFromFile(String readFileName) {
        File path = getApplicationContext().getFilesDir();
        File readFrom = new File(path + "/" + readFileName);
        byte[] content = new byte[(int) readFrom.length()];

        try {
            FileInputStream reader = new FileInputStream(readFrom);
            reader.read(content);
            return new String(content);
        } catch (Exception e) {
            e.printStackTrace();
            return e.toString();
        }
    }
}
```

- activity_read_data.xml

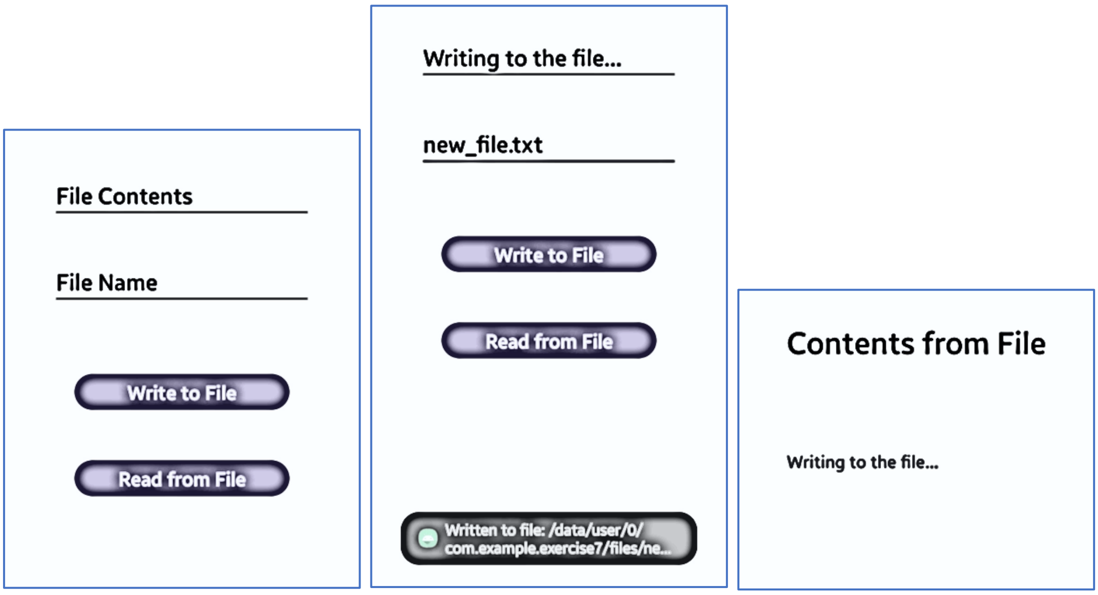
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" android:orientation="vertical"
tools:context=".ReadActivity">

    <TextView android:layout_width="400dp" android:layout_height="50dp"
android:layout_gravity="center" android:layout_marginTop="100dp" android:text="Contents
from File" android:gravity="center" android:textSize="32sp" android:typeface="monospace"
android:textStyle="bold" android:textColor="#000000" />
```

```
    <TextView android:id="@+id/contentsFromFile" android:layout_width="300dp"
android:layout_height="200dp" android:layout_gravity="center"
android:layout_marginTop="100dp" android:text="" android:textSize="18sp"
android:typeface="monospace" android:textStyle="bold" />

</LinearLayout>
```

Output:



Written to file: /data/user/0/com.example.exercise7/files/ne...

Writing to the file...

Result:

The mobile application was completed successfully

Best Practices:

1. Use meaningful variable names and comments for code clarity.
2. Implement error handling for file operations to address potential exceptions.
3. Provide user-friendly toast messages for successful or unsuccessful file operations.
4. Utilize separate activities for distinct functionalities to maintain a clean code structure.
5. Follow Android coding conventions and design guidelines for consistency.
6. Test the application on various devices and Android versions to ensure compatibility.
7. Consider implementing additional error handling and user prompts for improved robustness.

Learning Outcomes:

1. Understanding file read and write operations in Android.
2. Implementing activities, intents, and passing data between activities.
3. Handling user input through EditText and triggering actions with buttons.
4. Displaying information using TextView.
5. Gaining familiarity with Android layout design using LinearLayout.
6. Handling exceptions related to file I/O operations.
7. Implementing best practices for a clean and maintainable codebase.