Exercise 7 – Writing To and Reading From the SD Card

Objective

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.

Algorithm

1. Initialize the Project:
   - Open Android Studio and create a new Android Project.

2. Design the UI:
   - Create a layout with two `TextView` widgets for displaying and entering text.
   - Add a `Button` for submitting the text.

3. Request External Storage Permission:
   - In the manifest file, request permission to read and write to external storage.

4. Read Text from SD Card:
   - Implement logic to read text from a file on the SD card.
   - Display the retrieved text in one of the `TextView` widgets.

5. Write Text to SD Card:
   - Implement logic to write the text entered by the user into a file on the SD card.
   - Trigger this logic when the user clicks the submit button.

6. Display Results:
   - Display the retrieved and written text in the respective `TextView` widgets.

Features used

Main Features:
- Permission handling for external storage access.
- UI components for entering, submitting, and displaying text.
- File I/O operations for reading and writing text to the SD card.

Source Code

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
xmlns:app="http://schemas.android.com/apk/res-auto"
tools:context=".MainActivity">
<TextView
android:id="@+id/title"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="SD Card App"
android:textAppearance="@style/TextAppearance.AppCompat.Display1"
android:textColor="#007565"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.497"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.064" />
<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="36dp"
android:layout_marginEnd="108dp"
```

```
android:text="Write contents into a file"
android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textColor="#00BFA5"

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/title" /> <TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="80dp"
android:layout_marginEnd="152dp"
android:text="File contents:"
android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textColor="#304FFE"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/title" /> <TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="180dp"
android:layout_marginEnd="256dp"
android:text="File Name:"
android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textColor="#304FFE"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/title" /> <EditText
android:id="@+id/content"
android:layout_width="326dp"
android:layout_height="44dp"
android:layout_marginTop="16dp"
android:layout_marginEnd="40dp"
android:ems="10"
android:gravity="start|top"
android:inputType="textMultiLine"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView2" /> <EditText
android:id="@+id/filename"
android:layout_width="163dp"
android:layout_height="38dp"
android:layout_marginTop="76dp"
android:layout_marginEnd="60dp"
android:ems="10"
```

```
android:gravity="start|top"
android:inputType="textMultiLine"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView2" /> <Button

android:id="@+id/writebtn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="32dp"
android:layout_marginEnd="156dp"
android:backgroundTint="#00BFA5"
android:text="Write File"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/filename" />
<Button
android:id="@+id/btnLoad"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="152dp"
android:layout_marginEnd="136dp"
android:backgroundTint="#00BFA5"
android:text="Load From File"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/filename" />
</androidx.constraintlayout.widget.ConstraintLayout>
//activity_read_file.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=".ReadFile">
<EditText
android:id="@+id/filename2" android:layout_width="163dp"
android:layout_height="38dp" android:layout_marginTop="28dp"
android:layout_marginEnd="52dp" android:ems="10" android:gravity="start|top"
android:inputType="textMultiLine" app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView6" />
<TextView
```

```
android:id="@+id/textView4" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_marginTop="40dp"
android:layout_marginEnd="152dp" android:text="File contents:"
android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textColor="#304FFE" app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/loadbtn" />
<TextView
android:id="@+id/title2" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_marginTop="44dp"

android:layout_marginEnd="108dp" android:text="SD Card App"
android:textAppearance="@style/TextAppearance.AppCompat.Display1"
android:textColor="#007565" app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toTopOf="parent" />
<TextView
android:id="@+id/textView5" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_marginStart="100dp"
android:layout_marginTop="28dp"
android:text="File Name:"
android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textColor="#304FFE" app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView6" />
<Button
android:id="@+id/loadbtn" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_marginTop="36dp"
android:layout_marginEnd="148dp" android:backgroundTint="#00BFA5"
android:text="LOAD FILE" app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/filename2" />
<TextView
android:id="@+id/textView6" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_marginTop="20dp"
android:layout_marginEnd="104dp" android:text="Read contents From File"
android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textColor="#00BFA5" app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/title2" />
<TextView
android:id="@+id/tvLoad" android:layout_width="283dp"
android:layout_height="116dp" android:layout_marginTop="32dp"
android:layout_marginEnd="52dp"
android:textAppearance="@style/TextAppearance.AppCompat.Body1"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView4" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.ex7;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;

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.FileReader;
import java.io.IOException;
import java.io.OutputStreamWriter;
public class MainActivity extends AppCompatActivity {
// Declare the View object references
Button btnSave, btnLoad;
EditText content, filenamev;
TextView tvLoad;
// Define some String variables, initialized with empty string String
filepath = "";
String filename = "";
String fileContent = "";
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
btnSave = findViewById(R.id.writebtn);
btnLoad = findViewById(R.id.btnLoad);
content = findViewById(R.id.content);
```

```
filenamev = findViewById(R.id.filename);
filepath = "NewDirectory";
if(!isExternalStorageAvailableForRW()){
btnSave.setEnabled(false);
}
btnSave.setOnClickListener(new View.OnClickListener() { @Override
public void onClick(View view) {
fileContent = content.getText().toString().trim();
filename = filenamev.getText().toString().trim();
// Check for Storage Permission
if(isStoragePermissionGranted()){
if(!fileContent.equals("")){
File myExternalFile = new
File(getExternalFilesDir(filepath), filename);
FileOutputStream fos = null;
try {
fos = new FileOutputStream(myExternalFile);
fos.write(fileContent.getBytes());

fos.close();
} catch (FileNotFoundException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
content.setText("");
filenamev.setText("");
// Show a Toast message to inform the user that the
operation has been successfully completed.
Toast.makeText(MainActivity.this, "File saved to SD card.",
Toast.LENGTH_SHORT).show();
} else{
// If the Text field is empty show corresponding Toast message
Toast.makeText(MainActivity.this, "Text field can not be
empty.", Toast.LENGTH_SHORT).show();
}
}
}
});
btnLoad.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
```

```
Intent myIntent = new Intent(MainActivity.this, ReadFile.class);
MainActivity.this.startActivity(myIntent);
}
});
}
public boolean isStoragePermissionGranted() {
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
if
(checkSelfPermission(android.Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
//Permission is granted
return true;
} else {
//Permission is revoked
ActivityCompat.requestPermissions(this, new
String[]{android.Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1); return
false;
}
}
else {
//permission is automatically granted on sdk<23 upon installation
//Permission is granted
return true;
}

}
private boolean isExternalStorageAvailableForRW() {
// Check if the external storage is available for read and write by calling
// MEDIA_M/O/UNETnEvDi,ronment.getExternalStorageState() method. If the
returned state is
// then you can read and write files. So, return true in that case,
otherwise, false.
String extStorageState = Environment.getExternalStorageState();
if(extStorageState.equals(Environment.MEDIA_MOUNTED)){
return true;
}
return false;
}
}
```

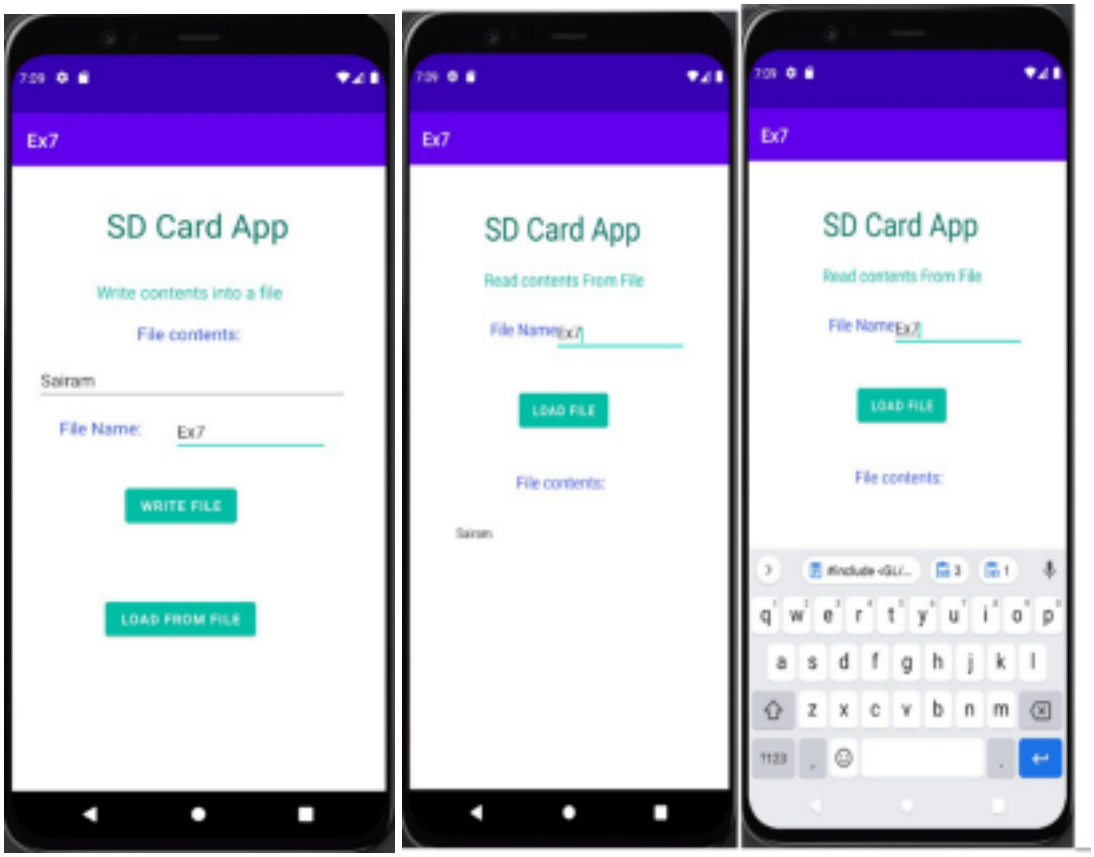
ReadFile.java

```
package com.example.ex7;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
public class ReadFile extends AppCompatActivity { Button btnLoad; TextView
tvLoad; EditText filenamev; String filename = "";
String filepath = "NewDirectory";
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_read_file);
btnLoad = findViewById(R.id.loadbtn); filenamev =
findViewById(R.id.filename2); tvLoad = findViewById(R.id.tvLoad);
btnLoad.setOnClickListener(new View.OnClickListener() { @Override public void
onClick(View view) {
filename = filenamev.getText().toString().trim();

FileReader fr = null;
File myExternalFile = new File(getExternalFilesDir(filepath),
filename);
StringBuilder stringBuilder = new StringBuilder();
try {
fr = new FileReader(myExternalFile); BufferedReader br = new
BufferedReader(fr); String line = br.readLine();
while(line != null){ stringBuilder.append(line).append('\n'); line =
br.readLine();
}
} catch (FileNotFoundException e) { e.printStackTrace();
} catch (IOException e) { e.printStackTrace();
} finally {
String fileContents = stringBuilder.toString();
```

```
tvLoad.setText(fileContents);
}
}
});
}
}
```

Output Screenshots



Result

Thus SD card read-write system was implemented

Best Practices

1. User friendly design
2. Readable layouts

3. Modularity
4. Used apt names for xml and java files.
5. Set padding and margins for dynamically added elements

Learning outcomes

● An android application to read and write from the SD card was implemented.
● Text is saved to and retrieved from the SD card.