Lab 8
Implementation of Storage in Android

Shared Preferences

In android, **Shared Preferences** are used to save and retrieve the primitive data types (integer, float, boolean, string, long) data in the form of key-value pairs from a file within an apps file structure.

Generally, the **Shared Preferences** object will point to a file that contains key-value pairs and provides a simple read and write methods to save and retrieve the key-value pairs from a file.

The Shared Preferences file is managed by an android framework and it can be accessed anywhere within the app to read or write data into the file, but it's not possible to access the file from any other app so it's secured.

Shared Preferences Example

Following is the example of storing and retrieving the primitive data type values from shared preferences file using **SharedPreferences**.

Create a new android application using android studio and give names as **SharedPreferencesExample**.

Once we create an application, open **activity_main.xml** file from **\res\layout** folder path and write the code like as shown below.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical" android:layout_width="match_parent"
   android:layout_height="match_parent">
   <TextView
      android:id="@+id/fstTxt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:layout_marginTop="150dp"
      android:text="UserName" />
   <EditText
      android:id="@+id/txtName"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10"/>
    <TextView
        android:id="@+id/secTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password"
        android:layout_marginLeft="100dp" />
    <EditText
        android:id="@+id/txtPwd"
        android:inputType="textPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10" />
    <Button
        android:id="@+id/btnLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:text="Login" />
</LinearLayout>
```

Now we will create another layout resource file **details.xml** in **\res\layout** path to get the first activity (**activity_main.xml**) details in second activity file for that right click on your **layout** folder → Go to **New** → select **Layout Resource File** and give name as **details.xml**.

Once we create a new layout resource file **details.xml**, open it and write the code like as shown below

details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/resultView"
        android:layout_gravity="center"
        android:layout_marginTop="170dp"
        android:textSize="20dp"/>
    <Button
        android:id="@+id/btnLogOut"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"
        android:text="Log Out" />
</LinearLayout>
```

Now open your main activity file **MainActivity.java**

MainActivity.java

```
import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    EditText uname, pwd;
    Button loginBtn;
    SharedPreferences pref;
    Intent intent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        uname = (EditText)findViewById(R.id.txtName);
        pwd = (EditText)findViewById(R.id.txtPwd);
        loginBtn = (Button)findViewById(R.id.btnLogin);
        pref = getSharedPreferences("user_details",MODE_PRIVATE);
        intent = new Intent(MainActivity.this,DetailsActivity.class);
        if(pref.contains("username") && pref.contains("password")){
            startActivity(intent);
        }
        loginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String username = uname.getText().toString();
                String password = pwd.getText().toString();
                if(username.equals("suresh") && password.equals("dasari")){
                    SharedPreferences.Editor editor = pref.edit();
                    editor.putString("username",username);
                    editor.putString("password",password);
                    editor.commit();
                    Toast.makeText(getApplicationContext(), "Login
Successful",Toast.LENGTH_SHORT).show();
```

```
                startActivity(intent);
            }
            else
            {
                Toast.makeText(getApplicationContext(),"Credentials are not
valid",Toast.LENGTH_SHORT).show();
            }
        }
    });
  }
}
```

If you observe above code, we are checking whether the entered username and password details matching or not based on that we are saving the details in shared preferences file and redirecting the user to another activity.

Now we will create another activity file **DetailsActivity.java** in **\java**

Once we create a new activity file **DetailsActivity.java**, open it and write the code like as shown below

DetailsActivity.java

```
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;



public class DetailsActivity extends AppCompatActivity {
  SharedPreferences prf;
  Intent intent;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.details);
    TextView result = (TextView)findViewById(R.id.resultView);
    Button btnLogOut = (Button)findViewById(R.id.btnLogOut);
    prf = getSharedPreferences("user_details",MODE_PRIVATE);
    intent = new Intent(DetailsActivity.this,MainActivity.class);
    result.setText("Hello, "+prf.getString("username",null));
    btnLogOut.setOnClickListener(new View.OnClickListener() {
      @Override
```

```
        public void onClick(View v) {
            SharedPreferences.Editor editor = prf.edit();
            editor.clear();
            editor.commit();
            startActivity(intent);
        }
    });
  }
}
```

Now we need to add this newly created activity in **AndroidManifest.xml** file in like as shown below.

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.tutlane.sharedpreferencesexample">
  <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/AppTheme">
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name=".DetailsActivity" android:label="Shared Preferences -
Details"></activity>
  </application>
</manifest>
```

Internal Storage

In android, **Internal Storage** is useful to store the data files locally on the device's internal memory using a **FileOutputStream** object. After storing the data files in device internal storage, we can read the data file from the device using a **FileInputStream** object.

The data files saved in the internal are managed by an android framework and it can be accessed anywhere within the app to read or write data into the file, but it's not possible to access the file from any other app so it's secured. When the user uninstalls the app, automatically these data files will be removed from the device internal storage.

Internal Storage Example

Following is the example of storing and retrieving the data files from the device's internal memory by using **FileOutputStream** and **FileInputStream** objects.

Create a new android application using android studio and give names as **InternalStorageExample**.

Once we create an application, open **activity_main.xml** file from **\res\layout** folder path and write the code like as shown below.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/fstTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="150dp"
        android:text="UserName" />
    <EditText
        android:id="@+id/txtName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10"/>
    <TextView
        android:id="@+id/secTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password"
        android:layout_marginLeft="100dp" />
    <EditText
        android:id="@+id/txtPwd"
        android:inputType="textPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10" />
    <Button
```

```
        android:id="@+id/btnSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:text="Save" />
</LinearLayout>
```

Now we will create another layout resource file **details.xml** in **\res\layout** path to get the first activity (**activity_main.xml**) details in second activity file for that right click on your **layout** folder → Go to **New** → select **Layout Resource File** and give name as **details.xml**.

Once we create a new layout resource file **details.xml**, open it and write the code like as shown below

details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/resultView"
        android:layout_gravity="center"
        android:layout_marginTop="170dp"
        android:textSize="20dp"/>
    <Button
        android:id="@+id/btnBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"
        android:text="Back" />
</LinearLayout>
```

Now open your main activity file **MainActivity.java** from **\java\ com.tutlane.internalstorageexample** path and write the code like as shown below

MainActivity.java

```
import android.content.Context;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.io.FileNotFoundException;
```

```
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {
    EditText uname, pwd;
    Button saveBtn;
    FileOutputStream fstream;
    Intent intent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        uname = (EditText)findViewById(R.id.txtName);
        pwd = (EditText)findViewById(R.id.txtPwd);
        saveBtn = (Button)findViewById(R.id.btnSave);
        saveBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String username = uname.getText().toString()+"\n";
                String password = pwd.getText().toString();
                try {
                    fstream = openFileOutput("user_details", Context.MODE_PRIVATE);
                    fstream.write(username.getBytes());
                    fstream.write(password.getBytes());
                    fstream.close();
                    Toast.makeText(getApplicationContext(), "Details Saved
Successfully",Toast.LENGTH_SHORT).show();
                    intent = new Intent(MainActivity.this,DetailsActivity.class);
                    startActivity(intent);
                } catch (FileNotFoundException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

If you observe above code, we are taking entered username and password details and saving it in device local file and redirecting the user to another activity.

Now we will create another activity file **DetailsActivity.java**

Once we create a new activity file **DetailsActivity.java**, open it and write the code like as shown below

DetailsActivity.java

```
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;



public class DetailsActivity extends AppCompatActivity {
    FileInputStream fstream;
    Intent intent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.details);
        TextView result = (TextView)findViewById(R.id.resultView);
        Button back = (Button)findViewById(R.id.btnBack);
        try {
            fstream = openFileInput("user_details");
            StringBuffer sbuffer = new StringBuffer();
            int i;
            while ((i = fstream.read())!= -1){
                sbuffer.append((char)i);
            }
            fstream.close();
            String details[] = sbuffer.toString().split("\n");
            result.setText("Name: "+ details[0]+"\nPassword: "+details[1]);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                intent = new Intent(DetailsActivity.this,MainActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

Now we need to add this newly created activity in **AndroidManifest.xml** file in like as shown below.

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.tutlane.internalstorageexample">
    <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/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".DetailsActivity" android:label="Internal Storage - Details"></activity>
    </application>
</manifest>
```

External Storage

In android, **External Storage** is useful to store the data files publically on the shared external storage using the **FileOutputStream** object. After storing the data files on external storage, we can read the data file from external storage media using a **FileInputStream** object.

The data files saved in external storage are word-readable and can be modified by the user when they enable USB mass storage to transfer files on a computer.

External Storage Example

Following is the example of storing and retrieving the data files from external memory by using **FileOutputStream** and **FileInputStream** objects.

Create a new android application using android studio and give names as **ExternalStorageExample**

Once we create an application, open **activity_main.xml** file from **\res\layout** folder path and write the code like as shown below.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
   android:orientation="vertical"
   android:padding="10dp">
   <EditText
      android:id="@+id/edit_text"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"/>
   <Button
      android:id="@+id/btnWrite"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginVertical="10dp"
      android:text="Write Data"/>
   <Button
      android:id="@+id/btnRead"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginVertical="10dp"
      android:text="Read Data"/>
   <Button
      android:id="@+id/btnClear"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginVertical="10dp"
      android:text="Clear Data"/>
</LinearLayout>
```

MainActivity.java

```
package com.example.externalstoragedemo1;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.View;
import android.widget.*;
```

```
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);
        final EditText editText = findViewById(R.id.edit_text);
        Button btnWrite = findViewById(R.id.btnWrite);
        Button btnRead = findViewById(R.id.btnRead);
        Button btnClear = findViewById(R.id.btnClear);
        if(checkSelfPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE)==
PackageManager.PERMISSION_GRANTED){
          // writedata();
        }else
        {
            requestPermissions(new String[] {Manifest.permission.WRITE_EXTERNAL_STORAGE},200 );
        }

        if (isExternalStorageAvailable() || isExternalStorageReadable()) {
            btnWrite.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    try {
                        FileOutputStream fileOutputStream = new FileOutputStream(getStorageDir("demo.txt"), true);

                        fileOutputStream.write(editText.getText().toString().getBytes());
                        fileOutputStream.write("\n".toString().getBytes());
                        fileOutputStream.close();
                        Toast.makeText(getApplicationContext(), "Data Written in the SDCARD!",
Toast.LENGTH_SHORT).show();
                        editText.setText("");
                    } catch (FileNotFoundException e) {
                        e.printStackTrace();
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }
            });
            btnRead.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    try {
                        FileInputStream fileInputStream = new FileInputStream(getStorageDir("demo.txt"));
                        StringBuffer str = new StringBuffer();
                        int c;
                        while ((c = fileInputStream.read()) != -1) {
                            str.append((char) c);
                        }
                        editText.setText(str);
                    } catch (FileNotFoundException e) {
                        e.printStackTrace();
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }
            });
            btnClear.setOnClickListener(new View.OnClickListener() {
                @Override
```

```
            public void onClick(View v) {
                try {
                    FileOutputStream fileOutputStream = new FileOutputStream(getStorageDir("demo.txt"));
                    fileOutputStream.flush();
                    fileOutputStream.close();
                } catch (FileNotFoundException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                Toast.makeText(getApplicationContext(), "File Clear!", Toast.LENGTH_SHORT).show();
            }
        });
    } else {
        Toast.makeText(getApplicationContext(), "SDCARD is not available!", Toast.LENGTH_SHORT).show();
    }
}

// check if external storage is available
public boolean isExternalStorageAvailable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}

//checks if external storage is available for read
public boolean isExternalStorageReadable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
        return true;
    }
    return false;
}

//get file path
public String getStorageDir(String fileName) {
    File file = new File(Environment.getExternalStorageDirectory(), "Demo");
    if (!file.mkdirs()) {
        file.mkdirs();
    }
    String filePath = file.getAbsolutePath() + File.separator + fileName;
    return filePath;
}
@Override
public void onRequestPermissionsResult(int requestCode,  String[] permissions,  int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if(requestCode==200)
    {
    //writedata();
    }
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.externalstoragedemo1">
```

```
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <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.ExternalStorageDemo1">
        <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>
```

Exercise

1. Create an android application to save data in a text file (internal storage). Then load file from memory and show in the view.

2. Create an android application for storing and retrieving data file from external memory.