Shared Preference

Shared Preference in Android are used to save data based on key-value pair. If we go deep into understanding of word: shared means to distribute data within and preference means something important or preferable, so SharedPreferences data is shared and preferred data.

Shared Preference can be used to save only primitive data type: string, long, int, float and Boolean.

What Is Preference File?

Before we begin explaining shared preference, it is important to understand preference file.

A preference file is actually a xml file saved in internal memory of device. Every application have some data stored in memory in a directory data/data/application package name i.e data/data/com.tmu.harryapp so whenever getSharedPreferences(String name,int mode) function get called it validates the file that if it exists or it doesn't then a new xml is created with passed name.

Two Ways To Save Data Through Shared Preference:

There are two different ways to save data in Android through Shared Preferences – One is using Activity based preferences and other is creating custom preferences.

Activity Preferences:

For activity preferences developer have to call function getPreferences (int mode) available in Activity class
Use only when one preference file is needed in Activity
It doesn't require name as it will be the only preference file for this activity

Developer doesn't usually prefer using this even if they need only one preference file in Activity. They prefer using custom getSharedPreferences(String name,int mode).

To use activity preferences developer have to call function getPreferences (int mode) available in Activity class. The function getPreferences(int mode) call the other function used to create custom preferences i.e getSharedPreferences(String name,int mode). Just because Activity contains only one preference file so getPreferences(int mode) function simply pass the name of Activity class to create a preference file.

Important Note: Mode are discussed in Custom preferences.

Custom Preferences:

Developer needs to use getSharedPreferences(String name,int mode) for custom preferences

Used in cases when more than one preference file required in Activity

Name of the preference file is passed in first parameter

Custom Preferences can be created by calling function getSharedPreferences(String name,int mode), it can be called from anywhere in the application with reference of Context. Here name is any preferred name for example: User,Book etc. and mode is used to set kind of privacy for file.

Mode And Its Type In Shared Preference:

To create activity based preferences or custom preferences developer have to pass mode to let the system know about privacy of preference file.

Before we begin a brief discussion on Context

Context is an abstract class used to get global information in Android Application resources like strings, values, drawables, assets etc. Here modes are declared and static final constant in Context class.

There are three types of Mode in Shared Preference:

Context.MODE_PRIVATE – default value (Not accessible outside of your application)

Context.MODE_WORLD_READABLE – readable to other apps

Context.MODE_WORLD_WRITEABLE – read/write to other apps

MODE_PRIVATE – It is a default mode. MODE_PRIVATE means that when any preference file is created with private mode then it will not be accessible outside of your application. This is the most common mode which is used.

MODE_WORLD_READABLE – If developer creates a shared preference file using mode world readable then it can be read by anyone who knows it's name, so any other outside application can easily read data of your app. This mode is very rarely used in App.

MODE_WORLD_WRITEABLE – It's similar to mode world readable but with both kind of accesses i.e read and write. This mode is never used in App by Developer.

Important Google Recommend Naming Convention For Preference File:Please make sure to follow recommendation by Google while giving preference file name. Google recommends to give name as application package name + preferred name.Refer Code for example

Activity_main.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:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/et_name"
        android:hint="Name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <EditText
        android:id="@+id/et_enroll"
        android:hint="Enrollment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <LinearLayout
        android:layout_marginTop="10dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content">
        <Button
            android:text="SAVE"
            android:id="@+id/bt_save"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content" />
        <Button
            android:text="LOAD"
            android:id="@+id/bt_load"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content" />
    </LinearLayout>
    <ListView
        android:dividerHeight="2dp"
        android:divider="@android:color/holo_orange_light"
        android:layout_marginTop="10dp"
        android:id="@+id/listshared"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

    </ListView>

</LinearLayout>
```

MainActivity.java

```
package com.example.sharedprefernce_examle;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    ListView listView;
    Button load,save;
    EditText name,enroll;
    SharedPreferences preferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listView=findViewById(R.id.listshared);
        load=findViewById(R.id.bt_load);
        load.setOnClickListener(this);
        save=findViewById(R.id.bt_save);
        save.setOnClickListener(this);
        name=findViewById(R.id.et_name);
        enroll=findViewById(R.id.et_enroll);
       // preferences=getPreferences(MODE_PRIVATE);//Activity based preferance
        preferences=getSharedPreferences("DivyanshTMU",MODE_PRIVATE);
    }

    @Override
    public void onClick(View view) {
        if (view.getId()==R.id.bt_save){
            saveDataInSharedPreferance();
        }

        if(view.getId()==R.id.bt_load){

            Map<List , List> map=loadDataToList();
            if(map.size()>0){
                MyAdapter adapter=new MyAdapter(MainActivity.this,map);
                listView.setAdapter(adapter);
            }
            else{
                Toast.makeText(this,"No data in preference
file",Toast.LENGTH_LONG).show();
            }
        }

    }

    private void saveDataInSharedPreferance(){
        SharedPreferences.Editor editor=preferences.edit();
        editor.putString(enroll.getText().toString(),name.getText().toString());
        //editor.putString("Enroll",enroll.getText().toString());
        editor.commit();
        Toast.makeText(this, "Data Saved", Toast.LENGTH_SHORT).show();

    }

    private Map<List,List> loadDataToList(){
       // SharedPreferences sp=getPreferences(MODE_PRIVATE); //DEFAULT DEVICE
AVAIABLE
        SharedPreferences sp=getSharedPreferences("DivyanshTMU",MODE_PRIVATE);
        Map<String,?> map=sp.getAll();
        HashMap<List,List> map1=new HashMap();
        List<String >name=new ArrayList<>();
        List<String>enroll=new ArrayList<>();
        for (Map.Entry<String,?> entry : map.entrySet()){
            String na=entry.getKey();
            String enro=entry.getValue().toString();
            name.add(na);
```

```
            enroll.add(enro);
            Log.e("><><><><><",na);
            Log.e("><><><><><",enro);

        }

        //map1.put("Name",name);
        map1.put(name,enroll);
        return map1;

    }
}
```

MyAdaper.java

```
package com.example.sharedprefernce_examle;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import java.util.List;
import java.util.Map;

public class MyAdapter extends BaseAdapter {
    Context mcontext;
    Map<List,List> listListMap;
    List<String>name;
    List<String>enroll;

    public MyAdapter(Context context, Map<List,List>listMap){
        mcontext=context;
        listListMap=listMap;

        extractData();
    }

    private void extractData(){
        for(Map.Entry<List,List>map1:listListMap.entrySet()){
            enroll=map1.getKey();

            name=map1.getValue();
```

```
        }

    }

    @Override
    public int getCount() {
        return name.size();
    }

    @Override
    public Object getItem(int i) {
        return null;
    }

    @Override
    public long getItemId(int i) {
        return 0;
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        if(view==null) {
            view = LayoutInflater.from(mcontext).inflate(R.layout.sharedpreflist,
null);
        }
        TextView name1=view.findViewById(R.id.tv_sname);
        TextView enroll1=view.findViewById(R.id.tv_senroll);
        name1.setText(name.get(i));
        enroll1.setText(enroll.get(i));

        return view;
    }
}
```

sharedpreflist.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"

    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">


    <TextView
        android:textStyle="bold|italic"
        android:textSize="20sp"
```

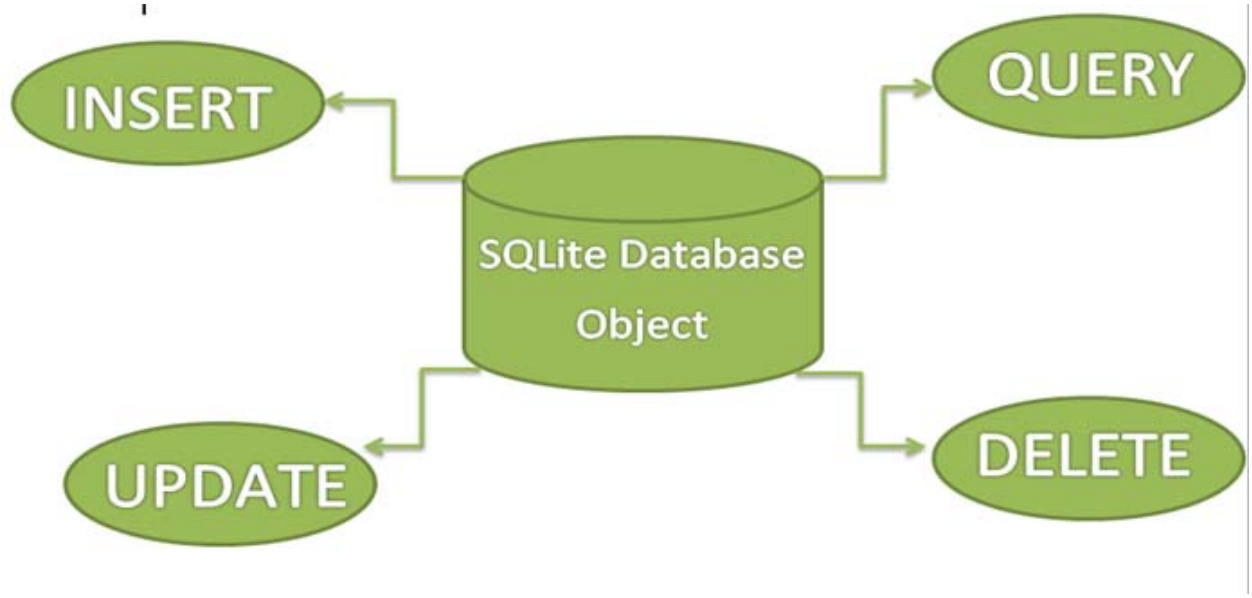
```
        android:id="@+id/tv_date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="DATA COMMING FROM SHARED PREF" />
    <LinearLayout
        android:layout_margin="10dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:textStyle="bold"
            android:textSize="20sp"
            android:text="NAME"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content" />
        <TextView
            android:textStyle="bold"
            android:textAlignment="viewEnd"
            android:layout_marginLeft="10dp"
            android:id="@+id/tv_sname"
            android:textSize="20sp"
            android:text="Name"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content" />
    </LinearLayout>
    <LinearLayout
        android:layout_margin="10dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:textStyle="bold"
            android:textSize="20sp"
            android:text="ENROLLMENT"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content" />
        <TextView
            android:textAlignment="viewEnd"
            android:textStyle="bold"
            android:layout_marginLeft="10dp"
            android:id="@+id/tv_senroll"
            android:textSize="20sp"
            android:text="ENROLLMENT"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content" />

    </LinearLayout>


</LinearLayout>
```

Sqlite

SQLite is a Structure query base database, open source, light weight, no network access and standalone database. It support embedded relational database features.



Whenever an application needs to store large amount of data then using sqlite is more preferable than other repository system like SharedPreferences or saving data in files.

Android has built in SQLite database implementation. It is available locally over the device(mobile & tablet) and contain data in text format. It carry light weight data and suitable with many languages. So, it doesn't required any administration or setup procedure of the database.

Important Note – The database created is saved in a directory: data/data/APP_Name/databases/DATABASE_NAME.

Creating And Updating Database In Android

For creating, updating and other operations you need to create a subclass or SQLiteOpenHelper class. SQLiteOpenHelper is a helper class to manage database creation and version management. It provides two methods onCreate(SQLiteDatabase db), onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion).

The SQLiteOpenHelper is responsible for opening database if exist, creating database if it does not exists and upgrading if required. The SQLiteOpenHelper only require the DATABASE_NAME to create database. After extending SQLiteOpenHelper you will need to implement its methods onCreate, onUpgrade and constructor.

onCreate(SQLiteDatabase sqLiteDatabase) method is called only once throughout the application lifecycle. It will be called whenever there is a first call to getReadableDatabase() or getWritableDatabase() function available in super SQLiteOpenHelper class. So SQLiteOpenHelper class call the onCreate() method after creating database and instantiate SQLiteDatabase object. Database name is passed in constructor call.

onUpgrade(SQLiteDatabase db,int oldVersion, int newVersion) is only called whenever there is a updation in existing version. So to update a version we have to increment the value of version variable passed in the superclass constructor.

In onUpgrade method we can write queries to perform whatever action is required. In most example you will see that existing table(s) are being dropped and again onCreate() method is being called to create tables again. But it’s not mandatory to do so and it all depends upon your requirements.

We have to change database version if we have added a new row in the database table. If we have requirement that we don't want to lose existing data in the table then we can write alter table query in the onUpgrade(SQLiteDatabase db,int oldVersion, int newVersion) method.

When you open your **database** it checks the version number and whether or not it exists. You can just "upgrade" your **database** rather than creating it new.
This **method** is called when you update your **database** version. It drops the existing tables and creates it again when onCreate **method** is called again

SQLite data type is an attribute that specifies the type of data of any object. Each column, variable and expression has related data type in SQLite.

You would use these data types while creating your tables. SQLite uses a more general dynamic type system. In SQLite, the datatype of a value is associated with the value itself, not with its container.

SQLite Storage Classes

Each value stored in an SQLite database has one of the following storage classes −

Sr.No.	Storage Class & Description
1	**NULL** The value is a NULL value.
2	**INTEGER** The value is a signed integer, stored in 1, 2, 3, 4, 6, or 8 bytes depending on the magnitude of the value.
3	**REAL** The value is a floating point value, stored as an 8-byte IEEE floating point number.
4	**TEXT** The value is a text string, stored using the database encoding (UTF-8, UTF-16BE or UTF-16LE)

5	**BLOB** The value is a blob of data, stored exactly as it was input.

SQLite storage class is slightly more general than a datatype. The INTEGER storage class, for example, includes 6 different integer datatypes of different lengths.

SQLite Affinity Type

SQLite supports the concept of **type affinity** on columns. Any column can still store any type of data but the preferred storage class for a column is called its **affinity**. Each table column in an SQLite3 database is assigned one of the following type affinities −

Sr.No.	Affinity & Description
1	**TEXT** This column stores all data using storage classes NULL, TEXT or BLOB.
2	**NUMERIC** This column may contain values using all five storage classes.
3	**INTEGER** Behaves the same as a column with NUMERIC affinity, with an exception in a CAST expression.
4	**REAL** Behaves like a column with NUMERIC affinity except that it forces integer values into floating point representation.
5	**NONE** A column with affinity NONE does not prefer one storage class over another and no attempt is made to coerce data from one storage class into another.

SQLite Affinity and Type Names

Following table lists down various data type names which can be used while creating SQLite3 tables with the corresponding applied affinity.

Data Type	Affinity
• INT • INTEGER • TINYINT • SMALLINT • MEDIUMINT • BIGINT • UNSIGNED BIG INT • INT2 • INT8	INTEGER
• CHARACTER(20) • VARCHAR(255) • VARYING CHARACTER(255) • NCHAR(55) • NATIVE CHARACTER(70) • NVARCHAR(100) • TEXT • CLOB	TEXT
• BLOB • no datatype specified	NONE
• REAL • DOUBLE • DOUBLE PRECISION • FLOAT	REAL
• NUMERIC • DECIMAL(10,5) • BOOLEAN • DATE • DATETIME	NUMERIC

Boolean Datatype

SQLite does not have a separate Boolean storage class. Instead, Boolean values are stored as integers 0 (false) and 1 (true).

Date and Time Datatype

SQLite does not have a separate storage class for storing dates and/or times, but SQLite is capable of storing dates and times as TEXT, REAL or INTEGER values.

Sr.No.	Storage Class & Date Formate
1	**TEXT** A date in a format like "YYYY-MM-DD HH:MM:SS.SSS"
2	**REAL** The number of days since noon in Greenwich on November 24, 4714 B.C.
3	**INTEGER** The number of seconds since 1970-01-01 00:00:00 UTC

You can choose to store dates and times in any of these formats and freely convert between formats using the built-in date and time functions.

myDbAdapter.java

Sample code for SQlite CURD operations in Android

```
package com.example.sqlite_inupdel_example;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class myDbAdapter extends SQLiteOpenHelper
{
    private static final String DATABASE_NAME = "TMU";    // Database Name
    private static final String TABLE_NAME = "myTable";   // Table Name
    private static final int DATABASE_Version = 1;    // Database Version
    private static final String UID="_id";     // Column I (Primary Key)
    private static final String NAME = "Name";    //Column II
    private static final String MyPASSWORD= "Password";    // Column III
    private static final String CREATE_TABLE = "CREATE TABLE "+TABLE_NAME+
            " ("+UID+" INTEGER PRIMARY KEY AUTOINCREMENT, "+NAME+" VARCHAR(255) ,"+
MyPASSWORD+" VARCHAR(225));";
    private static final String DROP_TABLE ="DROP TABLE IF EXISTS "+TABLE_NAME;
    private Context context;

    public myDbAdapter(Context context) {
```

```
    super(context, DATABASE_NAME, null, DATABASE_Version);
    this.context=context;
}

public void onCreate(SQLiteDatabase db) {

    try {
        db.execSQL(CREATE_TABLE);
    } catch (Exception e) {
        Message.message(context,""+e);
    }
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    try {
        Message.message(context,"OnUpgrade");
        db.execSQL(DROP_TABLE);
        onCreate(db);
    }catch (Exception e) {
        Message.message(context,""+e);
    }
}

public long insertData(String name, String pass)
{

    SQLiteDatabase dbb = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(NAME, name);
    contentValues.put(MyPASSWORD, pass);
    long id = dbb.insert(TABLE_NAME, null , contentValues);
    return id;
}

public String getData()
{
    SQLiteDatabase db = this.getWritableDatabase();
    String[] columns = {UID,NAME,MyPASSWORD};
   // Cursor cursor =db.query(TABLE_NAME,columns,null,null,null,null,null);
    Cursor cursor=db.rawQuery("Select * from myTable",null);
    StringBuffer buffer= new StringBuffer();
    while (cursor.moveToNext())
    {
        int cid =cursor.getInt(cursor.getColumnIndex(UID));
        String name =cursor.getString(cursor.getColumnIndex(NAME));
        String  password =cursor.getString(cursor.getColumnIndex(MyPASSWORD));
        buffer.append(cid+ "   " + name + "   " + password +" \n");
    }
    return buffer.toString();
}

public  int delete(String uname)
{
    SQLiteDatabase db = this.getWritableDatabase();
```

```
        String[] whereArgs ={uname};

        int count =db.delete(TABLE_NAME ,NAME+" = ?",whereArgs);
        return  count;
    }

    public int updateName(String oldName , String newName)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(NAME,newName);
        String[] whereArgs= {oldName};
        int count =db.update(TABLE_NAME,contentValues, NAME+" = ?",whereArgs );
        return count;
    }

}
```

MainActivity.java

```
package com.example.sqlite_inupdel_example;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    EditText Name, Pass, updateold, updatenew, delete;
    myDbAdapter helper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Name = (EditText) findViewById(R.id.editName);
        Pass = (EditText) findViewById(R.id.editPass);
        updateold = (EditText) findViewById(R.id.editText3);
        updatenew = (EditText) findViewById(R.id.editText5);
        delete = (EditText) findViewById(R.id.editText6);

        helper = new myDbAdapter(this);
    }

    public void addUser(View view) {
        String t1 = Name.getText().toString();
        String t2 = Pass.getText().toString();
        if (t1.isEmpty() || t2.isEmpty()) {
            Message.message(getApplicationContext(), "Enter Both Name and Password");
        } else {
            long id = helper.insertData(t1, t2);
            if (id <= 0) {
                Message.message(getApplicationContext(), "Insertion Unsuccessful");
                Name.setText("");
```

```
                Pass.setText("");
            } else {
                Message.message(getApplicationContext(), "Insertion Successful");
                Name.setText("");
                Pass.setText("");
            }
        }
    }

    public void viewdata(View view) {
        String data = helper.getData();
        Message.message(this, data);
    }

    public void update(View view) {
        String u1 = updateold.getText().toString();
        String u2 = updatenew.getText().toString();
        if (u1.isEmpty() || u2.isEmpty()) {
            Message.message(getApplicationContext(), "Enter Data");
        } else {
            int a = helper.updateName(u1, u2);
            if (a <= 0) {
                Message.message(getApplicationContext(), "Unsuccessful");
                updateold.setText("");
                updatenew.setText("");
            } else {
                Message.message(getApplicationContext(), "Updated");
                updateold.setText("");
                updatenew.setText("");
            }
        }

    }

    public void delete(View view) {
        String uname = delete.getText().toString();
        if (uname.isEmpty()) {
            Message.message(getApplicationContext(), "Enter Data");
        } else {
            int a = helper.delete(uname);
            if (a <= 0) {
                Message.message(getApplicationContext(), "Unsuccessful");
                delete.setText("");
            } else {
                Message.message(this, "DELETED");
                delete.setText("");
            }
        }
    }
}
```

Message.java

```java
package com.example.sqlite_inupdel_example;

import android.content.Context;
import android.widget.Toast;

public class Message {
    public static void message(Context context, String message) {
        Toast.makeText(context, message, Toast.LENGTH_LONG).show();
    }
}
```

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@android:color/holo_blue_dark">

    <TextView
        android:text="@string/username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="12dp"
        android:id="@+id/textView"
        android:textSize="18sp"
        android:textStyle="bold|italic"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:gravity="center" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:id="@+id/editName"
        android:textStyle="bold|italic"
        android:layout_below="@+id/textView"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:hint="Enter Name"
        android:gravity="center_vertical|center" />

    <TextView
        android:text="@string/password"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="13dp"
        android:id="@+id/textView2"
        android:textStyle="bold|italic"
        android:textSize="18sp"
        android:layout_below="@+id/editName"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:gravity="center"
        android:hint="Enter Password" />

    <Button
        android:text="@string/view_data"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button2"
        android:textSize="18sp"
        android:onClick="viewdata"
        android:textStyle="bold|italic"
        android:layout_alignBaseline="@+id/button"
        android:layout_alignBottom="@+id/button"
        android:layout_alignRight="@+id/button4"
        android:layout_alignEnd="@+id/button4" />

    <Button
        android:text="@string/add_user"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:textStyle="bold|italic"
        android:textSize="18sp"
        android:onClick="addUser"
        android:layout_marginLeft="28dp"
        android:layout_marginStart="28dp"
        android:layout_below="@+id/editPass"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="23dp" />

    <Button
        android:text="@string/update"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button3"
        android:onClick="update"
        android:textStyle="normal|bold"
        android:layout_below="@+id/editText3"
        android:layout_alignLeft="@+id/button4"
        android:layout_alignStart="@+id/button4"
        android:layout_marginTop="13dp" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:inputType="textPersonName"
        android:ems="10"
        android:id="@+id/editText6"
        android:layout_alignTop="@+id/button4"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:freezesText="false"
        android:hint="Enter Name to Delete Data"
        android:layout_toLeftOf="@+id/button2"
        android:layout_toStartOf="@+id/button2" />

    <Button
        android:text="@string/delete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="21dp"
        android:layout_marginEnd="21dp"
        android:id="@+id/button4"
        android:onClick="delete"
        android:textStyle="normal|bold"
        tools:ignore="RelativeOverlap"
        android:layout_marginBottom="41dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:layout_marginTop="47dp"
        android:id="@+id/editText3"
        android:textStyle="bold|italic"
        android:textSize="14sp"
        android:layout_below="@+id/button"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="7dp"
        android:layout_marginStart="7dp"
        android:hint="Current Name" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:ems="10"
        android:layout_marginTop="11dp"
        android:id="@+id/editPass"
        android:hint="Enter Password"
        android:gravity="center_vertical|center"
        android:textSize="18sp"
        android:layout_below="@+id/textView2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
```

```
        android:textAllCaps="false"
        android:textStyle="normal|bold" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:id="@+id/editText5"
        android:textStyle="bold|italic"
        android:textSize="14sp"
        android:hint="New Name"
        android:layout_alignTop="@+id/button3"
        android:layout_alignLeft="@+id/editText3"
        android:layout_alignStart="@+id/editText3"
        android:layout_marginTop="32dp" />
</RelativeLayout>
```