Experiment - 4 Date 5.12.22

Problem Definition :-
Develop an application that makes use of database.

Objectives :-
1. To understand the use of Absolute layout.
2. To understand how its use database in the application.
3. To understand the use of Builder view and its use in the application.

Theory :-

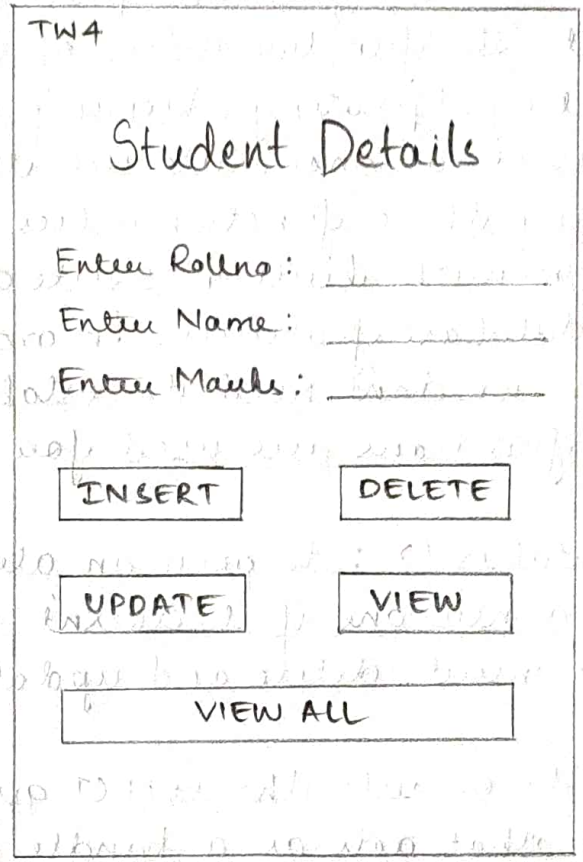
In this experiment, we build an app that uses database. We build the user interface and place components using Absolute layout. The functionalities of the app include Inserting, Deleting, Updating, Viewing a record and viewing all records at once. We use the SQLite Database for implementing these functionalities.

Android comes in with built-in SQLite database to support all relational database features. In order to access a SQLite database we don't need to establish any connection. The following functions are used for the database operations.

1. openOrCreateDatabase() : to open an already existing database or create a new one if it doesn't exist.
2. execSQL() : to insert, delete and update data in the table.
3. rawQuery() : to execute the SELECT query, in turn returns a Cursor object that acts as a handle to the result set returned by the SELECT query.

We use an instance of android.app.AlertDialog.Builder class to view all records of the table. The Builder class has a constructor that creates a builder for an alert dialog that uses the default alert dialog theme. The Builder class has methods like setTitle(), setMessage() and show(). The setTitle() is used to set the title of the Builder view. The setMessage() is used to set the content of the Builder view and show() is used to display the view on the screen. We set event listeners on every button and perform the desired functionality in the onClick method.

Design Diagram :-



XML code :- (activity_main.xml)

```
<?xml version="1.0" encoding="utf-8" ?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res
                                        /android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="50dp"
        android:layout_y="20dp"
        android:text="Student Details"
        android:textSize="30sp" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="110dp"
        android:text="Enter Rollno:"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/Rollno"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="175dp"
        android:layout_y="100dp"
        android:inputType="number"
        android:textSize="20sp" />
    <TextView
        android:id="@+id" layout_width="wrap_content"
```

```
    android : layout_height = "wrap_content"
    android : layout_x = "20dp"
    android : layout_y = "160 dp"
    android : text = "Enter Name :"
    android : textSize = "20 sp" />
<EditText
    android : id = "@+id / Name"
    android : layout_width = "150 dp"
    android : layout_height = "wrap_content"
    android . layout_x = "175dp"
    android : layout_y = "150 dp"
    android : inputType = "text"
    android : textSize = "20 sp" />
<TextView
    android : layout_width = "wrap_content"
    android : layout_height = "wrap_content"
    android : layout_x = "20 dp"
    android : layout_y = "210 dp"
    android : text = "Enter Marks :"
    android : textSize = "20 sp" />
<EditText
    android : id = "@+id / Marks"
    android : layout_width = "150 dp"
    android : layout_height = "wrap_content"
    android : layout_x = "175 dp"
    android : layout_y = "200 dp"
    android : inputType = "number"
    android : textSize = "20 sp" />
<Button
    android : id = "@+id / Insert"
    android : layout_width = "150 dp"
```

```
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="300dp"
    android:text="Insert"
    android:textSize="30dp" />
<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />
<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
    android:textSize="30dp" />
<Button
    android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="400dp"
    android:text="View"
    android:textSize="30dp" />
<Button
    android:id="@+id/ViewAll"
```

```
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_x="100dp"
    android:layout_y="500dp"
    android:text="View All"
    android:textSize="30dp" />
</AbsoluteLayout>
```

Java Code :- (MainActivity.java)

```
package com.example.tw4;
import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import as-Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity implements
                                    OnClickListener {
    EditText RollNo, Name, Marks;
    Button Insert, Delete, Update, View, ViewAll;
    SQLiteDatabase db;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        RollNo = (EditText) findViewById(R.id.RollNo);
```

```
Name = (EditText) findViewById (R.id.Name);
Marks = (EditText) findViewById (R.id.Marks);
Insert = (Button) findViewById (R.id.Insert);
Update = (Button) findViewById (R.id.Update);
Delete = (Button) findViewById (R.id.Delete);
View = (Button) findViewById (R.id.View);
ViewAll = (Button) findViewById (R.id.ViewAll);
Insert.setOnClickListener (this);
Update.setOnClickListener (this);
Delete.setOnClickListener (this);
View.setOnClickListener (this);
ViewAll.setOnClickListener (this);
// creating database and the table
db = openOrCreateDatabase ("StudentDB", Context.MODE_PRIVATE,
                                                    null);
db.execSQL ("CREATE TABLE IF NOT EXISTS student (rollno
    VARCHAR, name VARCHAR, marks VARCHAR);");
}
public void onClick (View v) {
  // inserting a record to the student table
  if (v == Insert) {
    // check for empty fields
    String rollno, marks, name;
    rollno = Rollno.getText().toString().trim();
    marks = Marks.getText().toString().trim();
    name = Name.getText().toString().trim();
    if (rollno.length()==0 || name.length()==0 || marks.length()==0) {
      showMessage ("Error", "Please enter all values");
      return;
    }
    db.execSQL ("INSERT INTO student VALUES ('" + rollno + "', '" +
```

```
        name + "', '" + marks + "');");
        showMessage("Success", "Record added");
        clearText();
    }
    // deleting a record from the student table
    if (view == Delete){
        String rollno = Rollno.getText().toString().trim();
        // check for empty roll number
        if (rollno.length() == 0) {
            showMessage("Error", "Please enter Rollno");
            return;
        }
        Cursor c = db.rawQuery("SELECT * FROM student WHERE
            rollno = '" + rollno + "'", null);
        if (c.moveToFirst()) {
            db.execSQL("DELETE FROM student WHERE rollno = '"
                + rollno + "'");
            showMessage("Success", "Record Deleted");
        }
        else {
            showMessage("Error", "Invalid Rollno");
        }
        clearText();
    }
    // updating a record in the student table
    if (view == Update) {
        String rollno, marks, name;
        rollno = Rollno.getText().toString().trim();
        name = Name.getText().toString().trim();
        marks = Marks.getText().toString().trim();
```

```
// check for empty rollno
if (rollno.length() == 0) {
    showMessage("Error", "Please enter rollno");
    return;
}
Cursor c = db.rawQuery("SELECT * FROM student WHERE
    rollno = '" + rollno + "'", null);
if (c.moveToFirst()) {
    db.execSQL("UPDATE student SET name = '" + name + "',
        marks = '" + marks + "' WHERE rollno = '" + rollno
        + "'");
    showMessage("Success", "Record Modified");
}
else
    showMessage("Error", "Invalid Rollno");
clearText();
}
// display a record from the student table
if (view == View) {
    String rollno = Rollno.getText().toString().trim();
    // check for empty rollno
    if (rollno.length() == 0) {
        showMessage("Error", "Please enter rollno");
        return;
    }
    Cursor c = db.rawQuery("SELECT * FROM student WHERE
        rollno = '" + rollno + "'", null);
    if (c.moveToFirst()) {
        Name.setText(c.getString(1));
        Marks.setText(c.getString(2));
    }
```

```
        else
            showMessage("Error", "Invalid Rollno");
        clearText();
    }
    // displaying all records
    if(view == viewAll) {
        Cursor c = db.rawQuery("SELECT * FROM student", null);
        if(c.getCount() == 0) {
            showMessage("Error", "No records found");
            return;
        }
        StringBuffer buffer = new StringBuffer();
        while(c.moveToNext()) {
            buffer.append("Rollno: ", c.getString(0) + "\n");
            buffer.append("Name: ", c.getString(1) + "\n");
            buffer.append("Marks: ", c.getString(2) + "\n\n");
        }
        showMessage("Success", buffer.toString());
    }
}

public void showMessage(String title, String message) {
    Builder builder = new Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(message);
    builder.show();
}

public void clearText() {
    Rollno.setText(""); Name.setText("");
    Marks.setText(""); Rollno.requestFocus();
}
}
```

Conclusion :-
By this teamwork, we understood the use of absolute layout, SQLite Database and Cursor class for database operations.

References :-
Android Studio 3.5 Development Essentials, Java edition, 2019
Neil Smyth / Payload Media Inc.