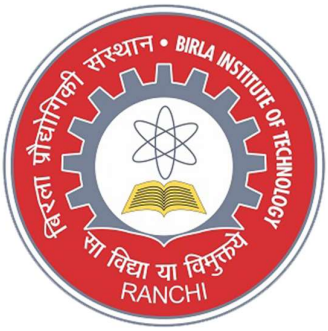
Birla Institute of Technology,Mesra,

Patna Campus



MI-Assignment

Name-Shubham Sourabh

Roll-Btech/15044/18

Sec-CSE 6th

MainActivity.java

```
package com.example.papi.sqlapp;

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

public class MainActivity extends AppCompatActivity {

    //Initializing fields
    DatabaseHelper myDB;
    EditText edit_name, edit_surname, edit_marks, edit_id;
    Button addData, viewData, updateData, deleteData;
    String name, surname, marks, id;
```

```
    boolean isUpdated;
    TextView textView;

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

        //Initialize Database
        myDB = new DatabaseHelper( this );

        //Initialize EditText
        edit_name = findViewById( R.id.name );
        edit_surname = findViewById( R.id.surname );
        edit_marks = findViewById( R.id.marks );
        edit_id = findViewById( R.id.id );

        // TextView
        textView = findViewById( R.id.textView5 );
        textView.setText( "Important Notes:\n1. Both Date and Time will be
stored automatically on the time of insertion.\n2. Existing Date and Time will
be updated when you update your data." );

        //Initialize Button
        addData = findViewById( R.id.button );
        viewData = findViewById( R.id.button2 );
        updateData = findViewById( R.id.button3 );
```

```
    deleteData = findViewById( R.id.button4 );

    //Call Methods
    AddData();
    viewData();
    updateData();
    deleteData();
}

//Adding or inserting data to database
public void AddData(){

    addData.setOnClickListener( new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            name = edit_name.getText().toString();
            surname = edit_surname.getText().toString();
            marks = edit_marks.getText().toString();

            //Current Date and Time
            Date date1 = new Date();
            String date = DateFormat.getDateTimeInstance(). format(date1);

            boolean isInserted = myDB.instertData( name, surname, marks, date);
```

```
            if(isInserted == true){
                Toast.makeText( MainActivity.this, "Data is inserted",
Toast.LENGTH_SHORT ).show();
            }
            else
                Toast.makeText( MainActivity.this, "Data is not inserted",
Toast.LENGTH_SHORT ).show();
        }
    } );
  }

  //For viewing data in database
  public void viewData(){

    viewData.setOnClickListener( new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        Cursor res = myDB.getData();

        if (res.getCount() == 0){
          showMessage("Error", "Data not found!");
        }

        else{
          StringBuffer buffer = new StringBuffer();
          while (res.moveToNext()){
            buffer.append( "ID: " + res.getString( 0 ) + "\n" );
```

```
                buffer.append( "Name: " + res.getString( 1 ) + "\n" );
                buffer.append( "Surname: " + res.getString( 2 ) + "\n" );
                buffer.append( "Marks: " + res.getString( 3 ) + "\n" );
                buffer.append( "Insertion/Updation Date:\n" + res.getString( 4 ) +
"\n\n" );
            }

            showMessage( "Data", buffer.toString() );

        }
      }
    } );
  }

  //For updating existing data in database
  public void updateData(){

    updateData.setOnClickListener( new View.OnClickListener() {
      @Override
      public void onClick(View view) {

        id = edit_id.getText().toString();
        name = edit_name.getText().toString();
        surname = edit_surname.getText().toString();
        marks = edit_marks.getText().toString();

        //Current Date and Time
```

```
            Date date1 = new Date();
            String date = DateFormat.getDateTimeInstance(). format(date1);

            boolean isUpdated = myDB.updateData( id, name, surname, marks,
date );

            if (isUpdated == true){
               showMessage( "Update", "Your data has been successfully
updated!" );
            }
            else
               showMessage( "Update failed", "Cannot Update your data :(" );
         }
      } );
   }

   //For deleting data in the database
   public void deleteData(){

      deleteData.setOnClickListener( new View.OnClickListener() {
         @Override
         public void onClick(View view) {

            id = edit_id.getText().toString();

            Integer res = myDB.deleteData( id );
            if(res > 0){
```

```
                Toast.makeText( getApplicationContext(), "Row effected",
Toast.LENGTH_SHORT ).show();
            }
            else{
                Toast.makeText( getApplicationContext(), "Row not effected",
Toast.LENGTH_SHORT ).show();
            }

        }
    } );
  }

  //Method for creating AlertDialog box
  private void showMessage(String title, String message) {

    AlertDialog.Builder builder = new AlertDialog.Builder( this );
    builder.setCancelable( true );
    builder.setTitle( title );
    builder.setMessage( message );
    builder.show();
  }
}
```

Database.java

```
package com.example.papi.sqlapp;

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

/**
 * Created by papi on 02-06-2021.
 */

public class DatabaseHelper extends SQLiteOpenHelper {

    //Initialize all the fields needed for database
    public static final String DATABASE_NAME = "Students.db";
    public static final String TABLE_NAME = "student_data";
    public static final String COL_1 = "ID";
    public static final String COL_2 = "Name";
    public static final String COL_3 = "Surname";
    public static final String COL_4 = "Marks";
    public static final String COL_5 = "Date";
    public static final String LBR = "(";
    public static final String RBR = ")";
    public static final String COM = ",";
```

```
    //Just pass context of the app to make it simpler
    public DatabaseHelper(Context context) {
        super( context, DATABASE_NAME, null, 2 );

    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        //Creating table

        db.execSQL( "create table " + TABLE_NAME + LBR + COL_1 + " INTEGER
PRIMARY KEY AUTOINCREMENT" + COM +
                COL_2 + " TEXT" + COM + COL_3 + " TEXT" + COM + COL_4 + "
INTEGER" + COM + COL_5 + " INTEGER" +RBR );

        // Another way of writing the CREATE TABLE query
        /* db.execSQL( "create table student_data (ID INTEGER PRIMARY KEY
AUTOINCREMENT, Name TEXT, Surname TEXT," +
                "Marks INTEGER, Date TEXT)" );*/

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {
```

```
        //Dropping old table
        db.execSQL( "DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate( db );

    }

    //Insert data in database
    public boolean instertData(String name, String surname, String marks, String
date){

        //Get the instance of SQL Database which we have created
        SQLiteDatabase db = getWritableDatabase();

        //To pass all the values in database
        ContentValues contentValues = new ContentValues();
        contentValues.put( COL_2, name );
        contentValues.put( COL_3, surname );
        contentValues.put( COL_4, marks );
        contentValues.put( COL_5, date);

        long result = db.insert( TABLE_NAME, null, contentValues );

        if(result == -1)
            return false;
        else
            return true;
    }
```

```
   //Cursor class is used to move around in the database
   public Cursor getData(){

      //Get the data from database
      SQLiteDatabase db = getWritableDatabase();
      Cursor res = db.rawQuery( "select * from " + TABLE_NAME, null );
      return res;
   }

   //Update fields of database using ID (Unique identifier)
   public boolean updateData(String id, String name, String surname, String
marks, String date){

      SQLiteDatabase db = getWritableDatabase();

      ContentValues contentValues = new ContentValues(  );
      // When you want to update only name field
      if(surname.equals( "" ) && marks.equals( "" )){
         contentValues.put( COL_1, id );
         contentValues.put( COL_2, name );
         contentValues.put( COL_5, date);
      }
      // When you want to update only surname field
      if(name.equals( "" ) && marks.equals( "" )){
         contentValues.put( COL_1, id );
         contentValues.put( COL_3, surname );
```

```
    contentValues.put( COL_5, date);
}
// When you want to update only marks field
if(name.equals( "" ) && surname.equals( "" )){
    contentValues.put( COL_1, id );
    contentValues.put( COL_4, marks );
    contentValues.put( COL_5, date);
}
// When you want to update name and surname field
if(marks.equals( "" ) && !name.isEmpty() && !surname.isEmpty()){
    contentValues.put( COL_1, id );
    contentValues.put( COL_2, name );
    contentValues.put( COL_3, surname );
    contentValues.put( COL_5, date);
}
// When you want to update marks and surname field
if(name.isEmpty() && !marks.isEmpty() && !surname.isEmpty()){
    contentValues.put( COL_1, id );
    contentValues.put( COL_3, surname );
    contentValues.put( COL_4, marks );
    contentValues.put( COL_5, date);
}
// When you want to update name and marks field
if(surname.isEmpty() && !name.isEmpty() && !marks.isEmpty()){
    contentValues.put( COL_1, id );
    contentValues.put( COL_2, name );
```

```
            contentValues.put( COL_4, marks );
            contentValues.put( COL_5, date);
        }
        // When you want to update every data field
        if(!id.isEmpty() && !name.isEmpty() && !surname.isEmpty() &&
!marks.isEmpty()){
            contentValues.put( COL_1, id );
            contentValues.put( COL_2, name );
            contentValues.put( COL_3, surname );
            contentValues.put( COL_4, marks );
            contentValues.put( COL_5, date);
        }

        // UPDATE query
        db.update( TABLE_NAME, contentValues, "ID = ?", new String[]{id} );
        return true;
    }

    //Delete data from the databse using ID (Primary Key)
    public Integer deleteData(String id){

        SQLiteDatabase db = getWritableDatabase();
        return db.delete( TABLE_NAME, "ID = ?", new String [] {id} );
    }
}
```

AndroidManifest.xml

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

    <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>
        <provider
            android:authorities="com.example.papi.sqlapp"
            android:name="com.example.papi.sqlapp.DBContentProvider"/>
    </application>

</manifest>
```

Layout(activity_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="com.example.papi.sqlapp.MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="52dp"
        android:text="Add Data"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.054"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.84" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
    android:text="Name"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.049"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.063" />

<EditText
    android:id="@+id/name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.502"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.034" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Surname"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.049"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.17" />

<EditText
    android:id="@+id/surname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.502"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.133" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Marsks"
    app:layout_constraintBottom_toBottomOf="parent"
```

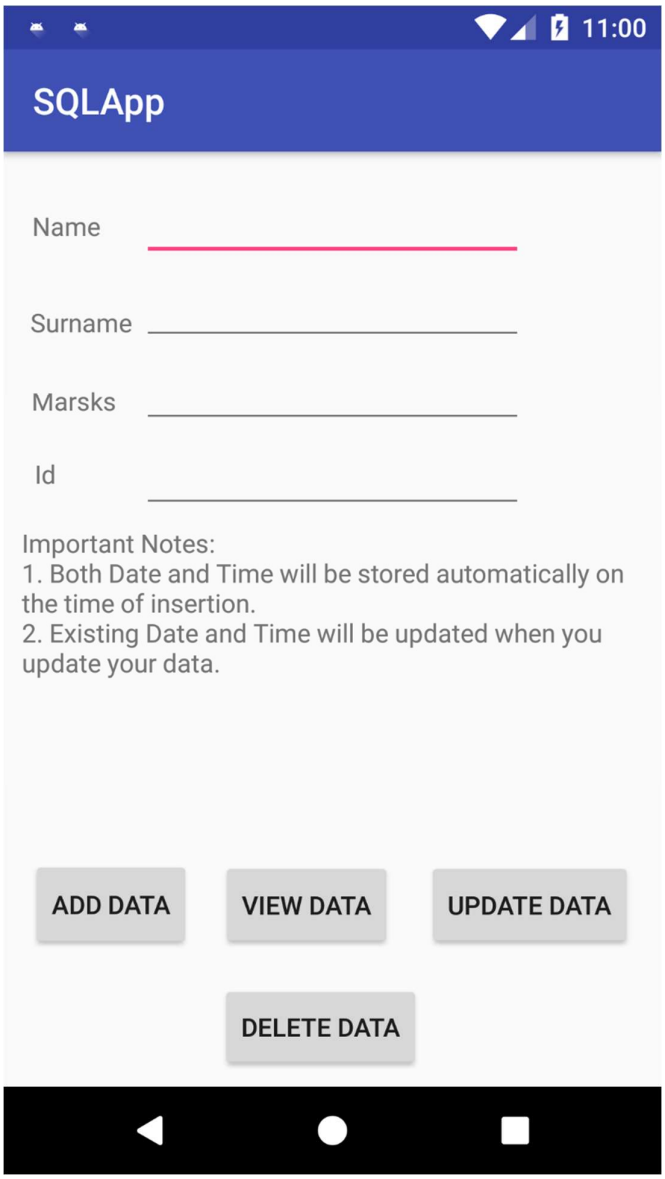
```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.049"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.258" />

<EditText
    android:id="@+id/marks"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.502"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.232" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="51dp"
    android:text="View Data"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

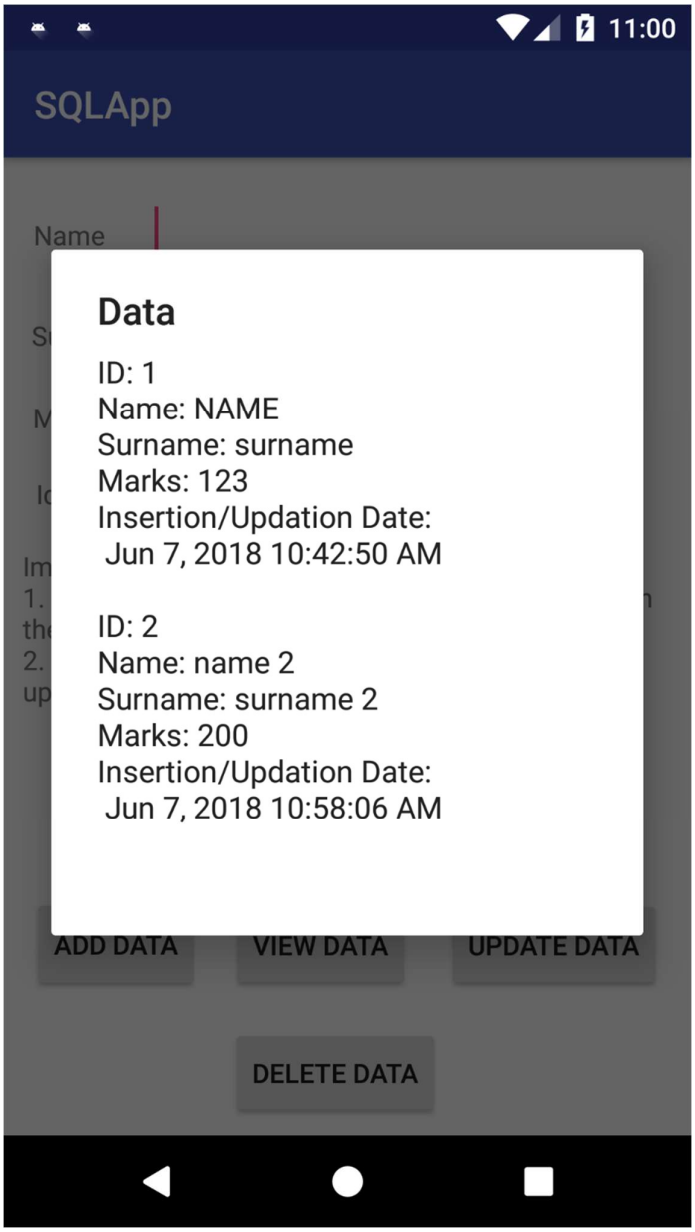
```
    app:layout_constraintHorizontal_bias="0.447"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.84" />

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="51dp"
    android:text="Update data"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.939"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.84" />

<EditText
    android:id="@+id/id"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.331" />

<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="50dp"
    android:text="delete data"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.474"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.984" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Id"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.049"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
```

```
        app:layout_constraintVertical_bias="0.339" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:paddingStart="10dp"
        android:paddingEnd="10dp"
        android:layout_marginTop="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.049"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.453" />

</android.support.constraint.ConstraintLayout>
```

Output:-


11:00
SQLApp
Name
Surname
Marsks
Id
Important Notes:
1. Both Date and Time will be stored automatically on the time of insertion.
2. Existing Date and Time will be updated when you update your data.
ADD DATA
VIEW DATA
UPDATE DATA
DELETE DATA


11:00
SQLApp
Name
Data
ID: 1
Name: NAME
Surname: surname
Marks: 123
Insertion/Updation Date:
Jun 7, 2018 10:42:50 AM
ID: 2
Name: name 2
Surname: surname 2
Marks: 200
Insertion/Updation Date:
Jun 7, 2018 10:58:06 AM
ADD DATA
VIEW DATA
UPDATE DATA
DELETE DATA