**DATABASE HANDLER CLASS:**

package com.example.testingapp;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
public class DatabaseHandler extends SQLiteOpenHelper {  
 private static final int *DATABASE\_VERSION* = 1;  
 protected static final String *DATABASE\_NAME* = "StudentDatabase";  
 public DatabaseHandler(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 String sql = "CREATE TABLE students " +  
 "( id INTEGER PRIMARY KEY AUTOINCREMENT, " +  
 "firstname TEXT, " +  
 "email TEXT ) ";  
 db.execSQL(sql);  
 }  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 String sql = "DROP TABLE IF EXISTS students";  
 db.execSQL(sql);  
 onCreate(db);  
 }  
}

**MAIN ACTIVITY JAVA CODE:**

package com.example.testingapp;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.KeyEvent;  
import android.view.View;  
import android.widget.Button;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import java.util.List;  
  
public class MainActivity extends AppCompatActivity {  
  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Button buttonCreateStudent = (Button) findViewById(R.id.*buttonCreateStudent*);  
 buttonCreateStudent.setOnClickListener(new OnClickListenerCreateStudent());  
 countRecords();  
 readRecords();  
  
  
 }  
 public void countRecords() {  
 int recordCount = new TableControllerStudent(this).count();  
  
  
  
 TextView textViewRecordCount = (TextView) findViewById(R.id.*textViewRecordCount*);  
 textViewRecordCount.setText(recordCount + " records found.");  
  
 }  
  
 public void readRecords() {  
 LinearLayout linearLayoutRecords = (LinearLayout) findViewById(R.id.*linearLayoutRecords*);  
 linearLayoutRecords.removeAllViews();  
 List<ObjectStudent> students = new TableControllerStudent(this).read();  
 if (students.size() > 0) {  
 for (ObjectStudent obj : students) {  
 int id = obj.id;  
 String studentFirstname = obj.firstname;  
 String studentEmail = obj.email;  
 String textViewContents = studentFirstname + " - " + studentEmail;  
 TextView textViewStudentItem= new TextView(this);  
 textViewStudentItem.setPadding(0, 10, 0, 10);  
 textViewStudentItem.setText(textViewContents);  
 textViewStudentItem.setTag(Integer.*toString*(id));  
  
 textViewStudentItem.setOnLongClickListener(new OnLongClickListenerStudentRecord());  
  
 linearLayoutRecords.addView(textViewStudentItem);  
 }  
 }  
 else {  
 TextView locationItem = new TextView(this);  
 locationItem.setPadding(8, 8, 8, 8);  
 locationItem.setText("No records yet.");  
 linearLayoutRecords.addView(locationItem);  
 }  
 }  
  
  
  
  
  
}

**MAIN ACTIVITY XML CODE:**

*<?*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"  
 android:orientation="vertical"  
 android:gravity="center">  
  
 <Button  
 android:id="@+id/buttonCreateStudent"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:text="Create Student"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.231" />  
  
 <TextView  
 android:id="@+id/textViewRecordCount"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/buttonCreateStudent"  
 android:layout\_marginBottom="436dp"  
 android:gravity="center"  
 android:padding="1dp"  
 android:text="0 records found"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 tools:layout\_editor\_absoluteX="0dp"  
 tools:ignore="MissingConstraints" />  
  
 <ScrollView  
 android:id="@+id/scrollViewRecords"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/textViewRecordCount"  
 tools:layout\_editor\_absoluteX="0dp"  
 tools:layout\_editor\_absoluteY="4dp"  
 tools:ignore="MissingConstraints">  
  
 <LinearLayout  
 android:id="@+id/linearLayoutRecords"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"></LinearLayout>  
 </ScrollView>  
  
 </android.support.constraint.ConstraintLayout>

**STUDENT INPUT CLASS CODE:**

<RelativeLayout 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" >  
 <EditText  
 android:id="@+id/editTextStudentFirstname"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:hint="Student Firstname"  
 android:singleLine="true" >  
 <requestFocus />  
 </EditText>  
 <EditText  
 android:id="@+id/editTextStudentEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_below="@+id/editTextStudentFirstname"  
 android:hint="Student Email"  
 android:singleLine="true" />  
</RelativeLayout>

**TABLE CONTROLLER STUDENT:**

package com.example.testingapp;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class TableControllerStudent extends DatabaseHandler {  
 public TableControllerStudent(Context context) {  
 super(context);  
 }  
 public boolean create(ObjectStudent objectStudent) {  
 ContentValues values = new ContentValues();  
 values.put("firstname", objectStudent.firstname);  
 values.put("email", objectStudent.email);  
 SQLiteDatabase db = this.getWritableDatabase();  
 boolean createSuccessful = db.insert("students", null, values) > 0;  
 db.close();  
 return createSuccessful;  
 }  
 public int count() {  
 SQLiteDatabase db = this.getWritableDatabase();  
 String sql = "SELECT \* FROM students";  
 int recordCount = db.rawQuery(sql, null).getCount();  
 db.close();  
 return recordCount;  
 }  
 public List<ObjectStudent> read() {  
 List<ObjectStudent> recordsList = new ArrayList<ObjectStudent>();  
 String sql = "SELECT \* FROM students ORDER BY id DESC";  
 SQLiteDatabase db = this.getWritableDatabase();  
 Cursor cursor = db.rawQuery(sql, null);  
 if (cursor.moveToFirst()) {  
 do {  
 int id = Integer.*parseInt*(cursor.getString(cursor.getColumnIndex("id")));  
 String studentFirstname = cursor.getString(cursor.getColumnIndex("firstname"));  
 String studentEmail = cursor.getString(cursor.getColumnIndex("email"));  
 ObjectStudent objectStudent = new ObjectStudent();  
 objectStudent.id = id;  
 objectStudent.firstname = studentFirstname;  
 objectStudent.email = studentEmail;  
 recordsList.add(objectStudent);  
 } while (cursor.moveToNext());  
 }  
 cursor.close();  
 db.close();  
 return recordsList;  
 }  
  
 public ObjectStudent readSingleRecord(int studentId) {  
 ObjectStudent objectStudent = null;  
 String sql = "SELECT \* FROM students WHERE id = " + studentId;  
 SQLiteDatabase db = this.getWritableDatabase();  
 Cursor cursor = db.rawQuery(sql, null);  
 if (cursor.moveToFirst()) {  
 int id = Integer.*parseInt*(cursor.getString(cursor.getColumnIndex("id")));  
 String firstname = cursor.getString(cursor.getColumnIndex("firstname"));  
 String email = cursor.getString(cursor.getColumnIndex("email"));  
 objectStudent = new ObjectStudent();  
 objectStudent.id = id;  
 objectStudent.firstname = firstname;  
 objectStudent.email = email;  
 }  
 cursor.close();  
 db.close();  
 return objectStudent;  
 }  
 public boolean update(ObjectStudent objectStudent) {  
 ContentValues values = new ContentValues();  
 values.put("firstname", objectStudent.firstname);  
 values.put("email", objectStudent.email);  
 String where = "id = ?";  
 String[] whereArgs = { Integer.*toString*(objectStudent.id) };  
 SQLiteDatabase db = this.getWritableDatabase();  
 boolean updateSuccessful = db.update("students", values, where, whereArgs) > 0;  
 db.close();  
 return updateSuccessful;  
 }  
 public boolean delete(int id) {  
 boolean deleteSuccessful = false;  
 SQLiteDatabase db = this.getWritableDatabase();  
 deleteSuccessful = db.delete("students", "id ='" + id + "'", null) > 0;  
 db.close();  
 return deleteSuccessful;  
 }  
  
  
  
  
}

**OBJECT STUDENT:**

package com.example.testingapp;  
  
public class ObjectStudent {  
  
  
 int id;  
 String firstname;  
 String email;  
 public ObjectStudent(){  
 }  
}

**ONCLICK LISTNER CREATE STUDENT:**

package com.example.testingapp;  
  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
public class OnClickListenerCreateStudent implements View.OnClickListener {  
 @Override  
 public void onClick(View view) {  
 final Context context = view.getRootView().getContext();  
 LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 final View formElementsView = inflater.inflate(R.layout.*student\_input\_form*, null, false);  
 final EditText editTextStudentFirstname = (EditText) formElementsView.findViewById(R.id.*editTextStudentFirstname*);  
 final EditText editTextStudentEmail = (EditText) formElementsView.findViewById(R.id.*editTextStudentEmail*);  
 new AlertDialog.Builder(context)  
 .setView(formElementsView)  
 .setTitle("Create Student")  
 .setPositiveButton("Add",  
 new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int id) {  
 String studentFirstname = editTextStudentFirstname.getText().toString();  
 String studentEmail = editTextStudentEmail.getText().toString();  
 ObjectStudent objectStudent = new ObjectStudent();  
 objectStudent.firstname= studentFirstname;  
 objectStudent.email= studentEmail;  
 boolean createSuccessful = new TableControllerStudent(context).create(objectStudent);  
  
 ((MainActivity)context).countRecords();  
 ((MainActivity) context).readRecords();  
  
 if(createSuccessful){  
 Toast.*makeText*(context, "Student information was saved.", Toast.*LENGTH\_SHORT*).show();  
 }else{  
 Toast.*makeText*(context, "Unable to save student information.", Toast.*LENGTH\_SHORT*).show();  
 }  
  
  
  
  
 dialog.cancel();  
 }  
 }).show();  
  
 }  
}

**ONLONGCLICK LISTNER STUDENT RECORD:**

package com.example.testingapp;  
  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
public class OnLongClickListenerStudentRecord implements View.OnLongClickListener {  
  
 Context context;  
 String id;  
  
 public void editRecord(final int studentId) {  
 final TableControllerStudent tableControllerStudent = new TableControllerStudent(context);  
 ObjectStudent objectStudent = tableControllerStudent.readSingleRecord(studentId);  
 LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 final View formElementsView = inflater.inflate(R.layout.*student\_input\_form*, null, false);  
 final EditText editTextStudentFirstname = (EditText) formElementsView.findViewById(R.id.*editTextStudentFirstname*);  
 final EditText editTextStudentEmail = (EditText) formElementsView.findViewById(R.id.*editTextStudentEmail*);  
 editTextStudentFirstname.setText(objectStudent.firstname);  
 editTextStudentEmail.setText(objectStudent.email);  
 new AlertDialog.Builder(context)  
 .setView(formElementsView)  
 .setTitle("Edit Record")  
 .setPositiveButton("Save Changes",  
 new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int id) {  
 ObjectStudent objectStudent = new ObjectStudent();  
 objectStudent.id = studentId;  
 objectStudent.firstname = editTextStudentFirstname.getText().toString();  
 objectStudent.email = editTextStudentEmail.getText().toString();  
 boolean updateSuccessful = tableControllerStudent.update(objectStudent);  
  
 ((MainActivity) context).countRecords();  
 ((MainActivity) context).readRecords();  
  
 if(updateSuccessful){  
 Toast.*makeText*(context, "Student record was updated.", Toast.*LENGTH\_SHORT*).show();  
 }else{  
 Toast.*makeText*(context, "Unable to update student record.", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 dialog.cancel();  
  
  
  
  
 }  
 }).show();  
  
 }  
  
 @Override  
 public boolean onLongClick(View view) {  
  
  
  
  
 context = view.getContext();  
 id = view.getTag().toString();  
 final CharSequence[] items = { "Edit", "Delete" };  
 new AlertDialog.Builder(context).setTitle("Student Record")  
 .setItems(items, new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int item) {  
  
 if (item == 0) {  
 editRecord(Integer.*parseInt*(id));  
 }  
 else if (item == 1) {  
 boolean deleteSuccessful = new TableControllerStudent(context).delete(Integer.*parseInt*(id));  
 if (deleteSuccessful){  
 Toast.*makeText*(context, "Student record was deleted.", Toast.*LENGTH\_SHORT*).show();  
 }else{  
 Toast.*makeText*(context, "Unable to delete student record.", Toast.*LENGTH\_SHORT*).show();  
 }  
 ((MainActivity) context).countRecords();  
 ((MainActivity) context).readRecords();  
 }  
  
  
 dialog.dismiss();  
 }  
 }).show();  
  
 return false;  
 }  
}