

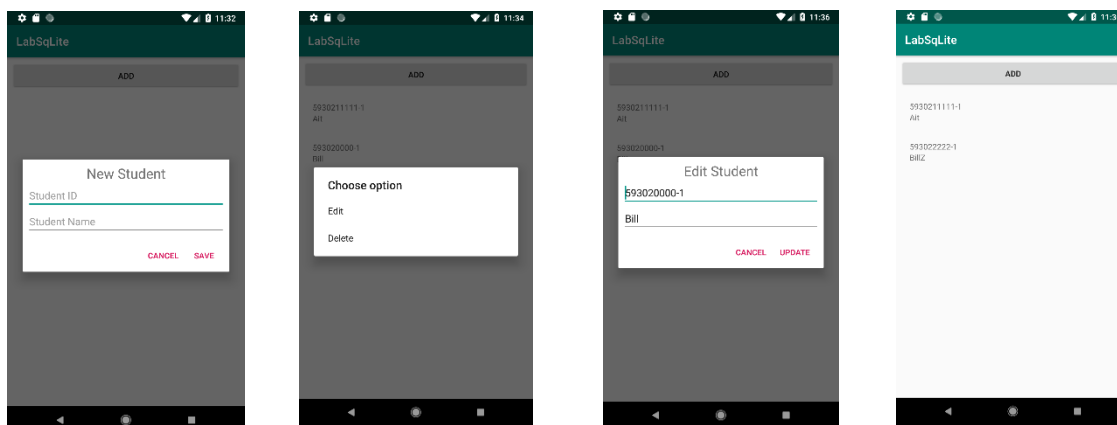
## LAB 6 SQLite

เอกสารประกอบการปฏิบัติการรายวิชา 342267 MOBILE DEVICE PROGRAMMING

### วัตถุประสงค์

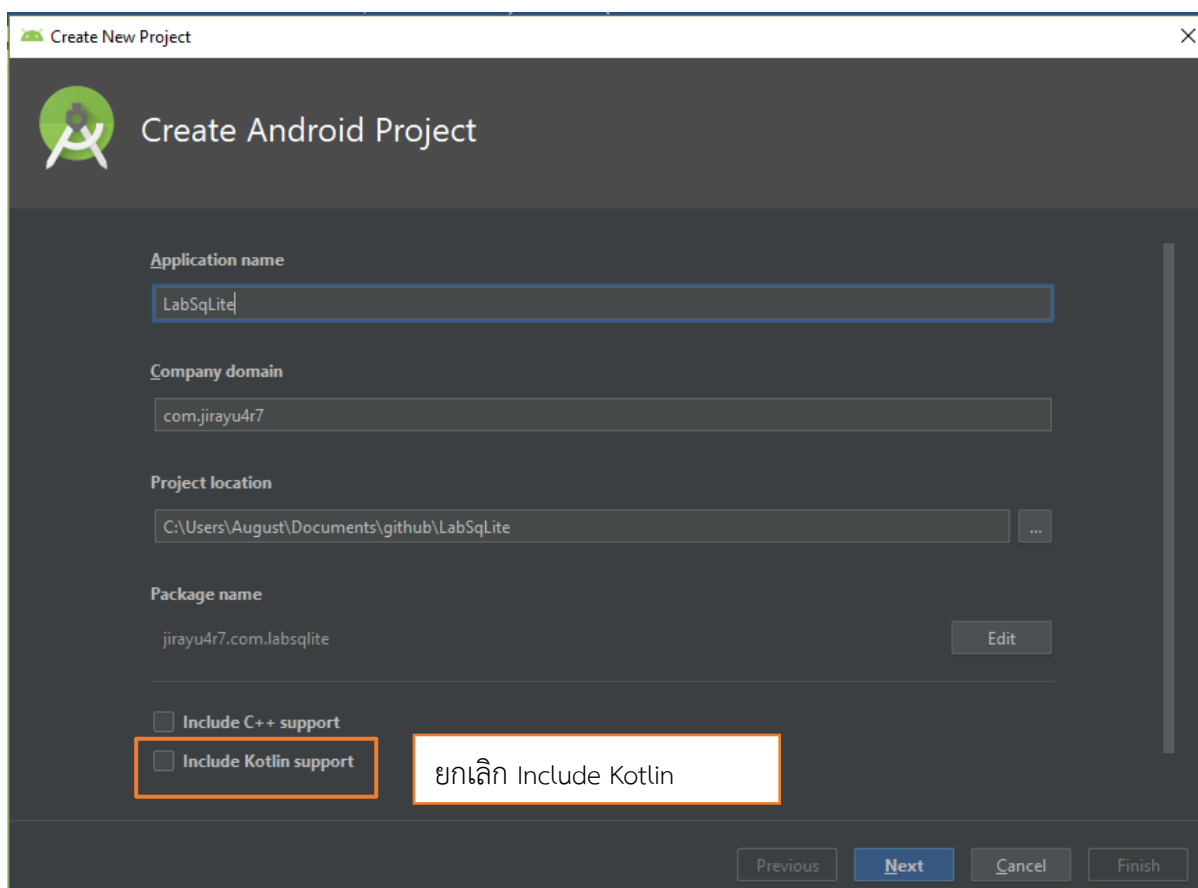
1. เพื่อให้ให้นักศึกษาสามารถออกแบบและใช้งาน SQLite ร่วมกับ RecyclerView ได้
2. เพื่อให้ให้นักศึกษาแอปพลิเคชันในการ เพิ่ม ลบ แก้ไข และดึงข้อมูล โดยใช้ SQLite ได้

### หน้าจอการทำงาน



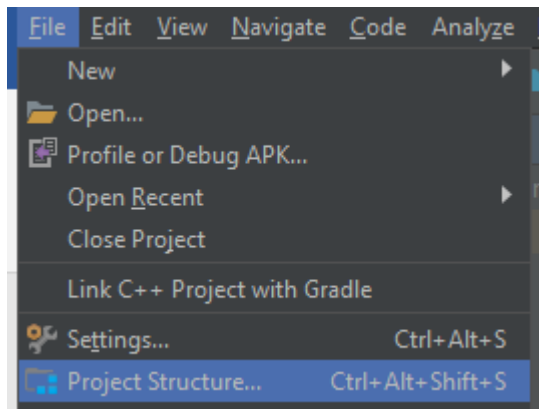
### การสร้างโปรเจกใหม่

ตั้งชื่อโปรเจก Application name (“LabSQLite”) และ ยกเลิก Include Kotlin support

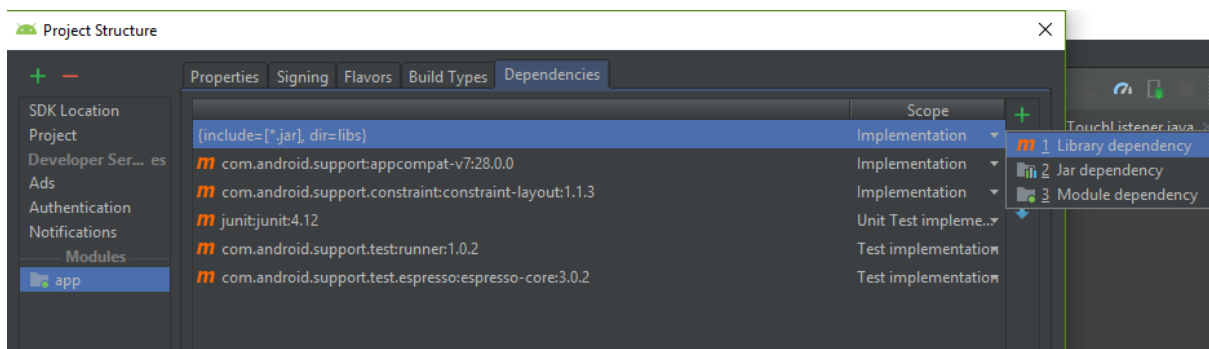


## การเพิ่ม Library ของ RecyclerView

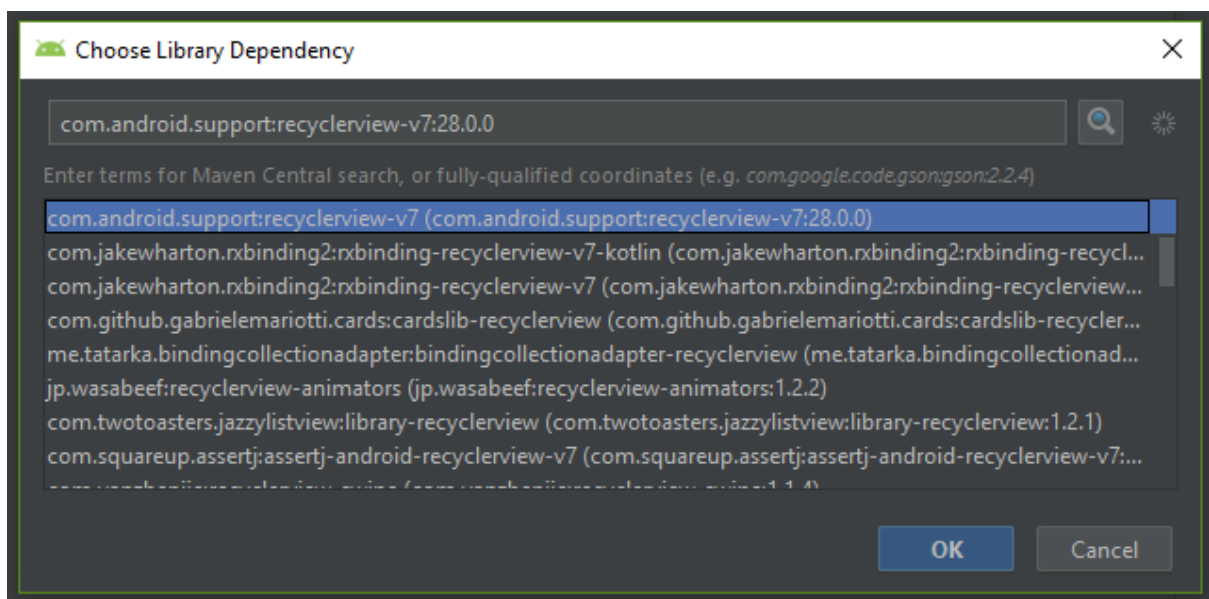
ไปที่เมนู File > Project Structure



แล้วเลือกapp ด้านซ้ายมือ แล้วเลือกแท็บ Dependencies กดเพิ่ม Library dependencies

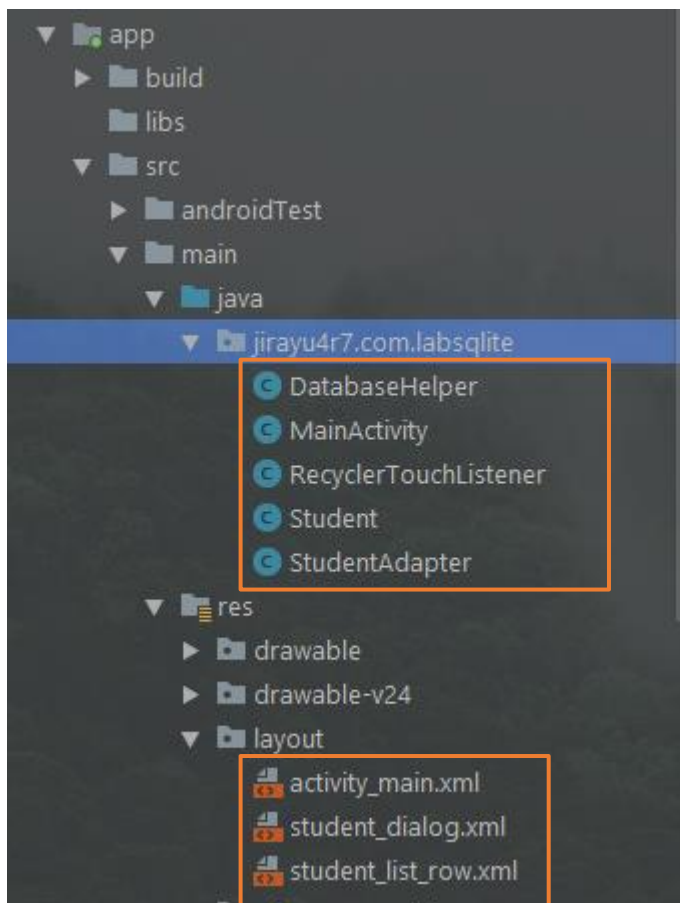


ให้พิมพ์ “com.android.support:recyclerview” เพิ่มค้นหา และกดปุ่ม OK แล้วรอให้โปรแกรม Sync เสร็จ



## โครงสร้างของโปรเจค

ให้นักศึกษาสร้าง Java Class และ ไฟล์ XML ตามโครงสร้างด้านล่าง



## การออกแบบ Layout

activity\_main.xml >> หน้า main

student\_dialog.xml >> หน้า dialog เพิ่ม แก้ไข

student\_list\_row.xml >> item view แสดงข้อมูลนักศึกษาภายใน RecyclerView

## การสร้างการทำงาน

DatabaseHelper.java >> Class สำหรับจัดการเกี่ยวกับ Database

MainActivity.java >> Class สำหรับจัดการเกี่ยวกับ UI

RecyclerViewTouchListener.java >> Class สำหรับจัดการเกี่ยวกับ การกด ของ RecyclerView

Student.java >> Java bean ของข้อมูลนักศึกษา

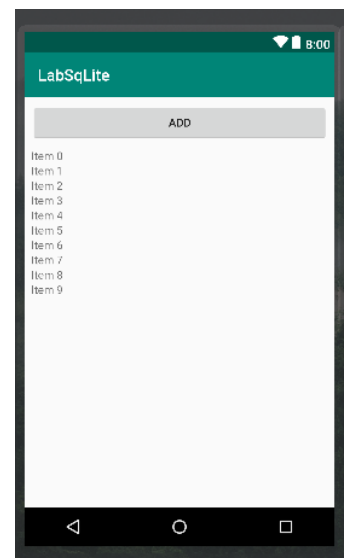
StudentAdapter.java >> Class สำหรับจัดการข้อมูลภายใน RecyclerView

## การออกแบบ Layout

ไฟล์ activity\_main.xml

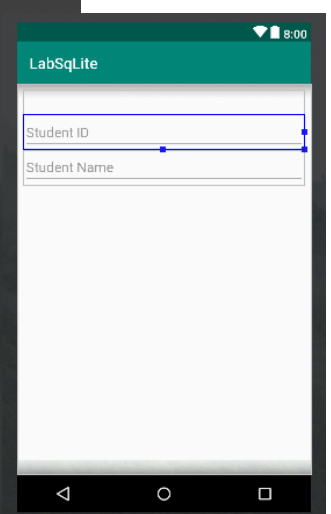
```
<Button
    android:id="@+id/button_add"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="Add"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<android.support.v7.widget.RecyclerView
    android:id="@+id/recycler_view_student"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button_add" />
```



ไฟล์ student\_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="8dp">
    <TextView
        android:id="@+id/text_view_dialog_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="24sp"/>
    <EditText
        android:id="@+id/edit_text_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Student ID"/>
    <EditText
        android:id="@+id/edit_text_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Student Name"/>
</LinearLayout>
```



## ไฟล์ student\_list\_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="8dp"
    android:layout_margin="8dp">

    <TextView
        android:id="@+id/text_view_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/text_view_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

## การสร้างการทำงาน

### เปิดไฟล์ Student.java

กำหนดตัวแปร id, name เป็นประเภท String ไว้เพื่อเก็บ รหัสนักศึกษา, ชื่อนักศึกษา และสร้าง getter, setter, constructor method ให้เรียบร้อย และสร้างตาราง Student

```
3  public class Student {
4      public static final String TABLE_NAME = "students";
5
6      public static final String COLUMN_ID = "std_id";
7      public static final String COLUMN_NAME = "std_name";
8
9      private String id;
10     private String name;
11
12     // Create table SQL query
13     public static final String CREATE_TABLE =
14         "CREATE TABLE " + TABLE_NAME + "("
15         + COLUMN_ID + " TEXT,"
16         + COLUMN_NAME + " TEXT"
17         + ")";
18
19     public Student(String id, String name) {
20         this.id = id;
21         this.name = name;
22     }
23
24     public Student() {}
25
26     public String getId() { return id; }
27
28     public void setId(String id) { this.id = id; }
29
30     public String getName() { return name; }
31
32     public void setName(String name) { this.name = name; }
33
34 }
35
```



## เปิดไฟล์ DatabaseHelper.java

ให้สร้าง Class **DatabaseHelper** Extends **SQLiteOpenHelper** โดยต้องOverride เมธอด **onCreate** และ **onUpdate** และต้องสร้างตัวแปรประเภท **String** **DATABASE\_VERSION**, **DATABASE\_NAME** นอกจากนี้สร้าง constructor รับ parameter เป็น **Context** แล้วกำหนด super ชื่อดาต้าเบสและเวอร์ชัน

```
public class DatabaseHelper extends SQLiteOpenHelper {

    // Database Version
    private static final int DATABASE_VERSION = 1;

    // Database Name
    private static final String DATABASE_NAME = "student_db";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, factory: null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(Student.CREATE_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int il) {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " + Student.TABLE_NAME);

        // Create tables again
        onCreate(db);
    }
}
```

## สร้างเมธอด insertStudent ใน DatabaseHelper.java

ประกาศเป็น public void และรับ parameter เป็น id และ name เป็นตัวแปรชนิด String

```
public void insertStudent(String id, String student) {
    // get writable database as we want to write data
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put(Student.COLUMN_ID, id);
    values.put(Student.COLUMN_NAME, student);

    db.insert(Student.TABLE_NAME, nullColumnHack: null, values);
    db.close();
}
```

### สร้างเมธอด getStudent ใน DatabaseHelper.java

ประกาศเป็น public และรับ parameter เป็น id เป็นตัวแปรชนิด String เพื่อส่งค่าข้อมูลของ Student กลับไป

```
public Student getStudent(String id) {
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = db.query(Student.TABLE_NAME,
        new String[]{Student.COLUMN_ID, Student.COLUMN_NAME},
        selection: Student.COLUMN_ID + "=?",
        new String[]{String.valueOf(id)},
        null, null, null, null);

    if (cursor != null) {
        cursor.moveToFirst();
    }

    Student student = new Student(
        cursor.getString(cursor.getColumnIndex(Student.COLUMN_ID)),
        cursor.getString(cursor.getColumnIndex(Student.COLUMN_NAME)));

    cursor.close();
    return student;
}
```

### สร้างเมธอด getAllStudent ใน DatabaseHelper.java สำหรับดึงข้อมูลของนักศึกษาทั้งหมดมาแสดง

```
public List<Student> getAllStudents() {
    List<Student> students = new ArrayList<>();

    String selectQuery = "SELECT * FROM " + Student.TABLE_NAME;

    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);

    if (cursor.moveToFirst()) {
        do {
            Student student = new Student();
            student.setId(cursor.getString(cursor.getColumnIndex(Student.COLUMN_ID)));
            student.setName(cursor.getString(cursor.getColumnIndex(Student.COLUMN_NAME)));
            students.add(student);
        } while (cursor.moveToNext());
    }

    db.close();
    return students;
}
```

สร้างเมธอด updateStudent ใน DatabaseHelper.java เพื่อแก้ไขข้อมูลนักศึกษา

```
public int updateStudent(Student student) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();

    values.put(Student.COLUMN_ID, student.getId());
    values.put(Student.COLUMN_NAME, student.getName());

    return db.update(Student.TABLE_NAME, values, whereClause: Student.COLUMN_ID + " = ?",
        new String[]{String.valueOf(student.getId())});
}
```

สร้างเมธอด deleteStudent ใน DatabaseHelper.java เพื่อลบข้อมูลนักศึกษา

```
public void deleteStudent(Student student) {
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(Student.TABLE_NAME, whereClause: Student.COLUMN_ID + " = ?",
        new String[]{String.valueOf(student.getId())});
    db.close();
}
```



## เปิดไฟล์ StudentAdapter.java

สำหรับ Class StudentAdapter ทำการ extends RecyclerView.Adapter  
<StudentAdapter.MyViewHolder> แล้ว Implement method ‘onCreateViewHolder’,  
‘onBindViewHolder’, ‘getItemCount’

จากนั้นให้สร้างตัวแปรประเภท Context, และ List<Student> แล้วกำหนดค่าผ่าน constructor และสร้าง inner class “MyViewHolder” extends RecyclerView.ViewHolder แล้วทำการ BindView ใน constructor

```
public class StudentAdapter extends RecyclerView.Adapter<StudentAdapter.MyViewHolder> {  
  
    private Context context;  
    private List<Student> studentList;  
  
    public class MyViewHolder extends RecyclerView.ViewHolder {  
        public TextView idTextView;  
        public TextView nameTextView;  
  
        public MyViewHolder(View view) {  
            super(view);  
            idTextView = view.findViewById(R.id.text_view_id);  
            nameTextView = view.findViewById(R.id.text_view_name);  
        }  
    }  
  
    public StudentAdapter(Context context, List<Student> notesList) {  
        this.context = context;  
        this.studentList = notesList;  
    }  
  
    @NonNull  
    @Override  
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {  
        View itemView = LayoutInflater.from(viewGroup.getContext())  
            .inflate(R.layout.student_list_row, viewGroup, attachToRoot: false);  
  
        return new MyViewHolder(itemView);  
    }  
  
    @Override  
    public void onBindViewHolder(@NonNull MyViewHolder myViewHolder, int position) {  
        Student student = studentList.get(position);  
  
        myViewHolder.idTextView.setText(student.getId());  
        myViewHolder.nameTextView.setText(student.getName());  
    }  
  
    @Override  
    public int getItemCount() {  
        return studentList.size();  
    }  
}
```

## เปิดไฟล์ RecyclerViewTouchListener.java

สร้าง Interface class ClickListener และประกาศตัวแปร ClickListener, GestureDetector

```
private ClickListener clicklistener;  
private GestureDetector gestureDetector;  
  
public interface ClickListener {  
    void onClick(View view, int position);  
    void onLongClick(View view, int position);  
}
```

ประกาศ constructor รับ parameter Context, RecyclerView, ClickListener

```
public RecyclerViewTouchListener(Context context,  
                                final RecyclerView recycleView,  
                                final ClickListener clicklistener) {  
  
    this.clicklistener = clicklistener;  
    gestureDetector = new GestureDetector(context,  
        new GestureDetector.SimpleOnGestureListener() {  
            @Override  
            public boolean onSingleTapUp(MotionEvent e) {  
                return true;  
            }  
  
            @Override  
            public void onLongPress(MotionEvent e) {  
                View child = recycleView.findChildViewUnder(e.getX(), e.getY());  
                if (child != null && clicklistener != null) {  
                    clicklistener.onLongClick(child,  
                        recycleView.getChildAdapterPosition(child));  
                }  
            }  
        }  
    );  
}
```

สำหรับคลาส 'RecyclerViewTouchListener' implements 'RecyclerView.OnItemTouchListener'  
แล้วจะ implements methods 'onInterceptTouchEvent' ตามคำสั่งด้านล่าง ส่วน 'onTouchEvent' และ  
'onRequestDisallowInterceptTouchEvent' ไม่ต้องเพิ่มคำสั่ง

```
public class RecyclerViewTouchListener implements RecyclerView.OnItemTouchListener
```

ในเมธอด 'onInterceptTouchEvent'

```
@Override
public boolean onInterceptTouchEvent(@NonNull RecyclerView recyclerView,
                                     @NonNull MotionEvent motionEvent) {
    View child = recyclerView.findChildViewUnder(motionEvent.getX(),
        motionEvent.getY());
    if (child != null && clicklistener != null &&
        gestureDetector.onTouchEvent(motionEvent)) {
        clicklistener.onClick(child,
            recyclerView.getChildAdapterPosition(child));
    }
    return false;
}
```

## เปิดไฟล์ MainActivity.java

ประกาศตัวแปรใน class MainActivity

```
private StudentAdapter mAdapter;  
private List<Student> studentList = new ArrayList<>();  
private RecyclerView studentsRecyclerView;  
private Button addButton;  
  
private DatabaseHelper db;
```

ประกาศเมธอด 'createStudent', 'updateStudent', 'deleteStudent' ใน class MainActivity แล้วเพิ่มคำสั่งดังภาพด้านล่าง

```
private void createStudent(String id, String name) {  
    db.insertStudent(id, name);  
    Student student = db.getStudent(id);  
    if (student != null) {  
        studentList.add(index 0, student);  
        mAdapter.notifyDataSetChanged();  
    }  
}  
  
private void updateStudent(String id, String name, int position) {  
    Student student = studentList.get(position);  
  
    student.setId(id);  
    student.setName(name);  
  
    db.updateStudent(student);  
  
    studentList.set(position, student);  
    mAdapter.notifyItemChanged(position);  
}  
  
private void deleteStudent(int position) {  
    db.deleteStudent(studentList.get(position));  
  
    studentList.remove(position);  
    mAdapter.notifyItemRemoved(position);  
}
```

ประกาศเมธอด 'showStudentDialog' ใน class MainActivity

```
private void showStudentDialog(final boolean shouldUpdate, final Student student, final int position) {
    LayoutInflater inflaterAndroid = LayoutInflater.from(getApplicationContext());
    View view = inflaterAndroid.inflate(R.layout.student_dialog, root null);

    AlertDialog.Builder alertDialogBuilderUserInput = new AlertDialog.Builder(context MainActivity.this);
    alertDialogBuilderUserInput.setView(view);

    final EditText inputId = view.findViewById(R.id.edit_text_id);
    final EditText inputName = view.findViewById(R.id.edit_text_name);
    TextView dialogTitle = view.findViewById(R.id.text_view_dialog_title);
    dialogTitle.setText(!shouldUpdate ? "New Student" : "Edit Student");

    if (shouldUpdate && student != null) {
        inputId.setText(student.getId());
        inputName.setText(student.getName());
    }
    alertDialogBuilderUserInput
        .setCancelable(false)
        .setPositiveButton(shouldUpdate ? "update" : "save", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialogBox, int id) {

            }
        })
        .setNegativeButton(text "cancel",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialogBox, int id) {
                    dialogBox.cancel();
                }
            }
        ));

    final AlertDialog alertDialog = alertDialogBuilderUserInput.create();
    alertDialog.show();

    alertDialog.getButton(AlertDialog.BUTTON_POSITIVE).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (TextUtils.isEmpty(inputId.getText().toString())) {
                Toast.makeText(context MainActivity.this, text "Enter ID!", Toast.LENGTH_SHORT).show();
                return;
            } else {
                alertDialog.dismiss();
            }

            if (shouldUpdate && student != null) {
                updateStudent(inputId.getText().toString(),
                    inputName.getText().toString(),
                    position);
            } else {
                // create new note
                createStudent(inputId.getText().toString(), inputName.getText().toString());
            }
        }
    });
}
```



ประกาศเมธอด 'showActionsDialog' ใน class MainActivity

```
private void showActionsDialog(final int position) {
    CharSequence colors[] = new CharSequence[]{"Edit", "Delete"};

    AlertDialog.Builder builder = new AlertDialog.Builder(context: this);
    builder.setTitle("Choose option");
    builder.setItems(colors, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            if (which == 0) {
                showStudentDialog(shouldUpdate: true, studentList.get(position), position);
            } else {
                deleteStudent(position);
            }
        }
    });
    builder.show();
}
```

คำสั่งในเมธอด onCreate() ของ class MainActivity.java

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

    addButton = findViewById(R.id.button_add);
    studentsRecyclerView = findViewById(R.id.recycler_view_student);

    db = new DatabaseHelper(context: this);
    studentList.addAll(db.getAllStudents());

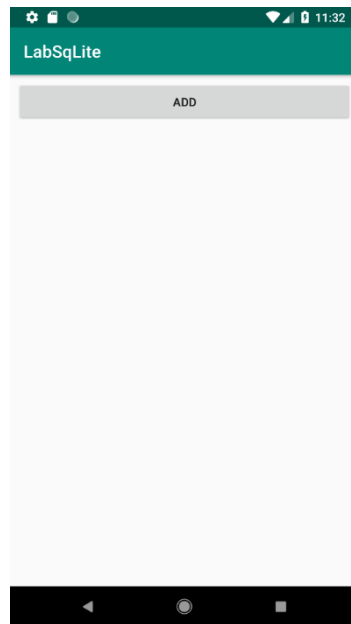
    addButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            showStudentDialog(shouldUpdate: false, student: null, position: -1);
        }
    });

    mAdapter = new StudentAdapter(context: this, studentList);
    RecyclerView.LayoutManager mLayoutManager = new LinearLayoutManager(getApplicationContext());
    studentsRecyclerView.setLayoutManager(mLayoutManager);
    studentsRecyclerView.setItemAnimator(new DefaultItemAnimator());
    studentsRecyclerView.setAdapter(mAdapter);

    studentsRecyclerView.addOnItemTouchListener(new RecyclerViewTouchListener(context: this,
        studentsRecyclerView, new RecyclerViewTouchListener.ClickListener() {
            @Override
            public void onClick(View view, final int position) {
            }

            @Override
            public void onLongClick(View view, int position) { showActionsDialog(position); }
        }
    ));
}
```

ทดลองรันแอปพลิเคชัน ให้สามารถเพิ่มข้อมูลนักศึกษาได้



## Assignment 6

ให้นักศึกษาร่างแอปพลิเคชันที่แสดงข้อมูลใน RecyclerView โดยประกอบด้วย รหัสนักศึกษา รายชื่อ และแตรัการศึกษา, และสามารถเพิ่ม ลบ แก้ไข ข้อมูลได้ ดังภาพด้านล่าง

การส่งงานให้แคปโค้ดลงส่งเป็น PDF และ Record video การทำงานทั้งหมดของแอปพลิเคชัน Ass6\_รหัสนักศึกษาไม่มีขีด\_secX.rar

