

Unit 8

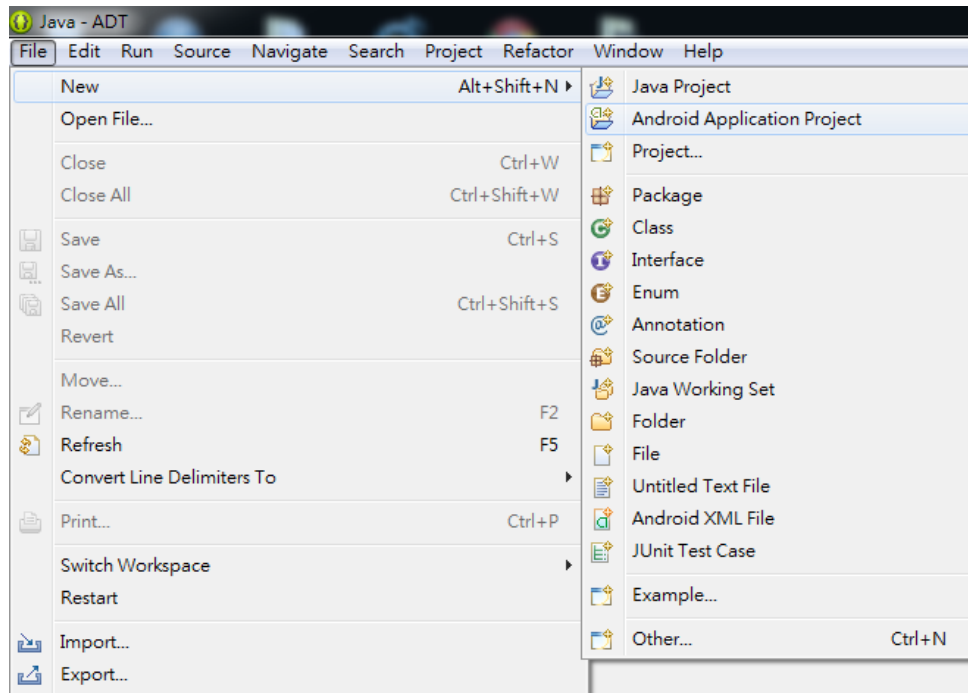
User Interface

8.2 Android & SQLite

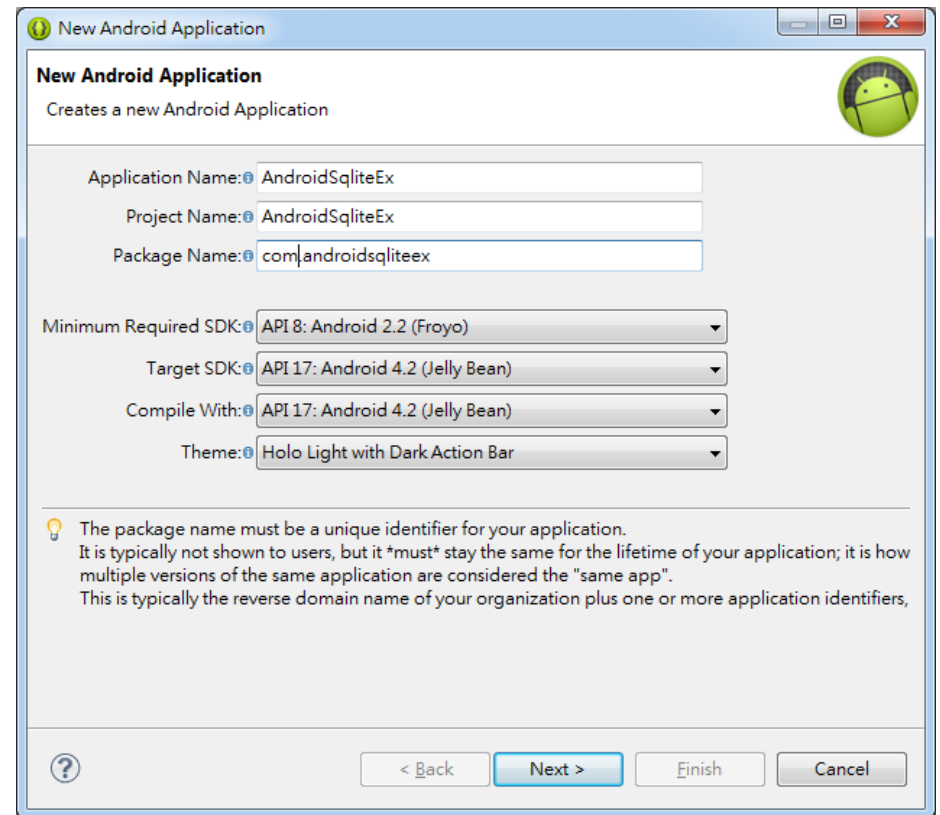
建立專案

建立專案

新增一個Android Application Project

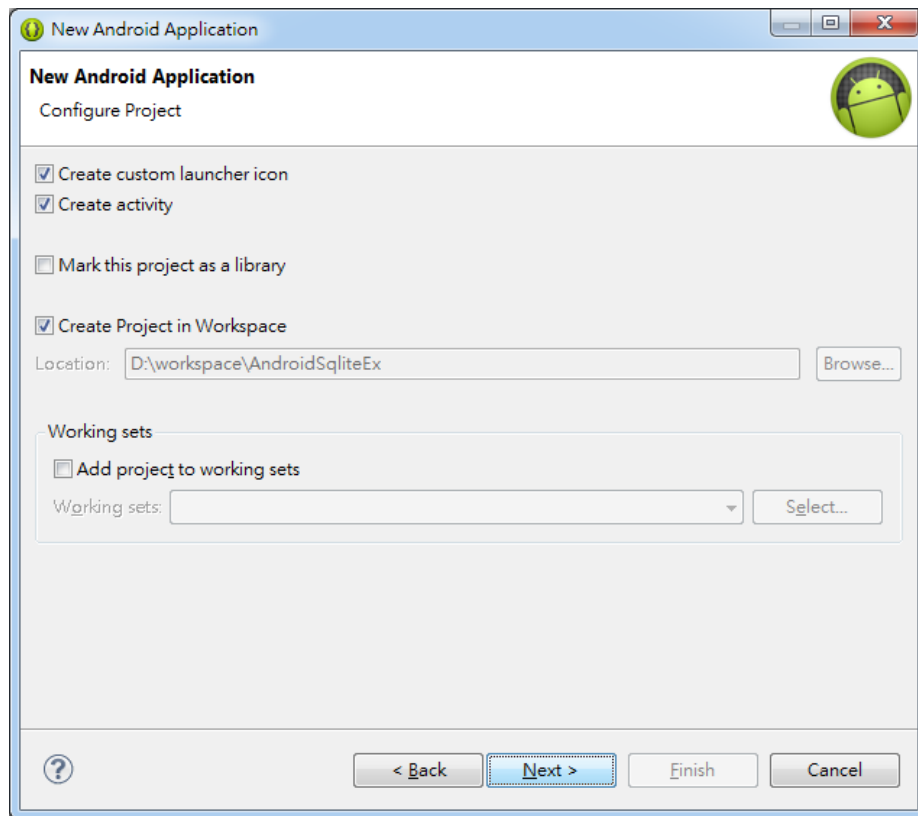


依開發者需求進行設定

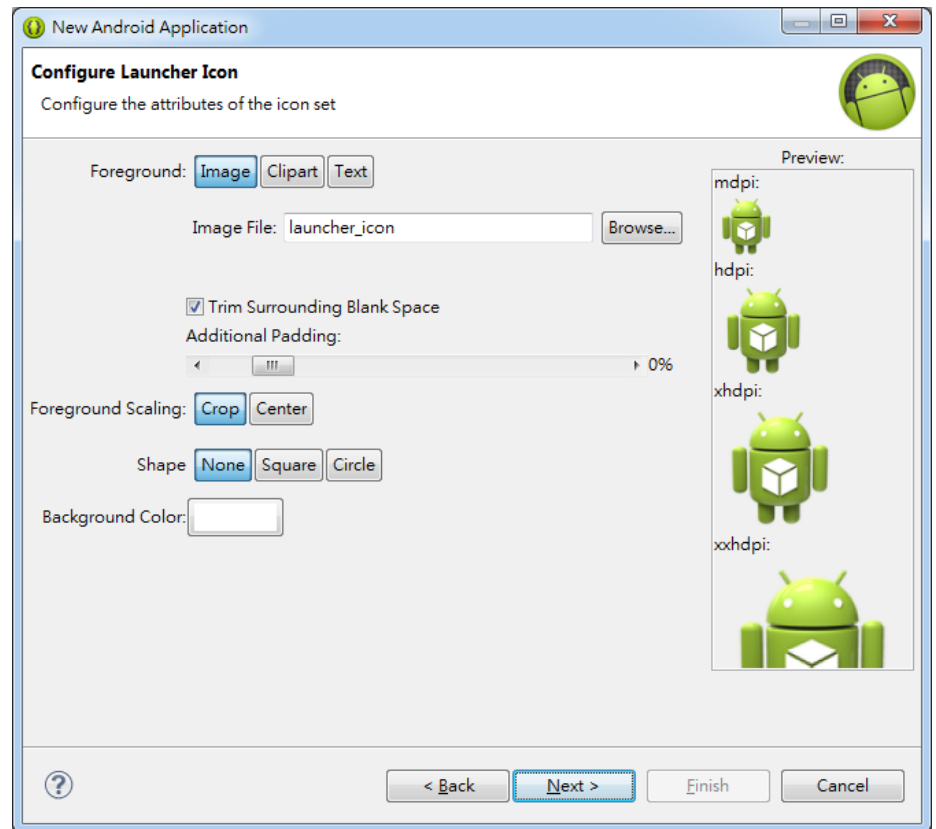


建立專案

直接按下一步即可

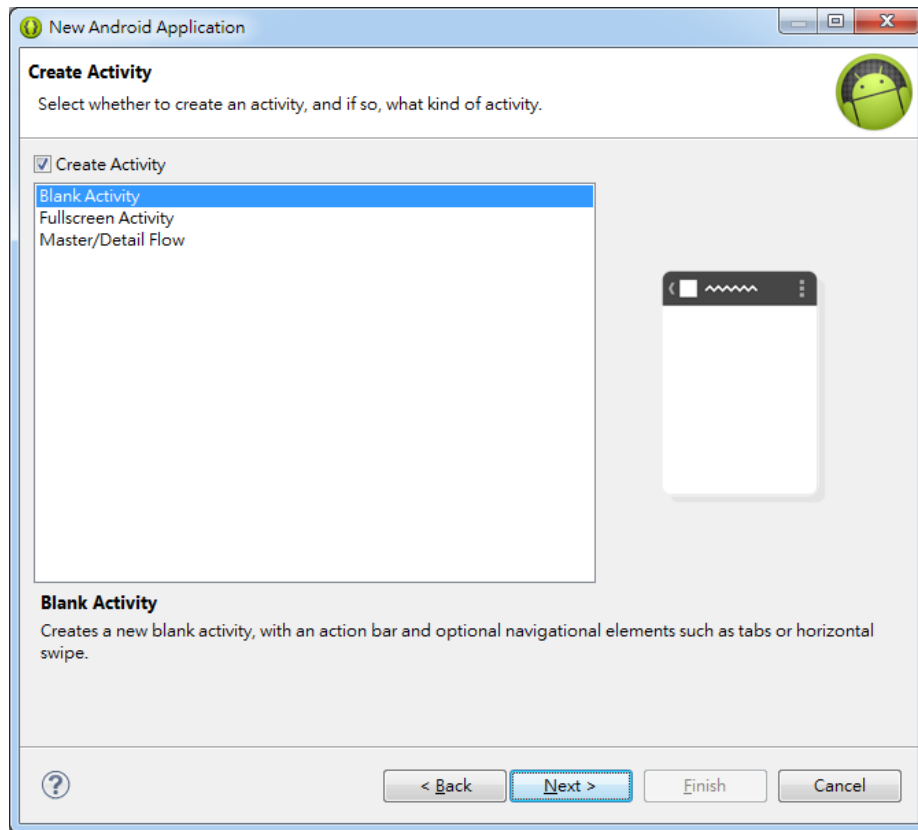


設置 APP ICON，預設為Android機器人圖示

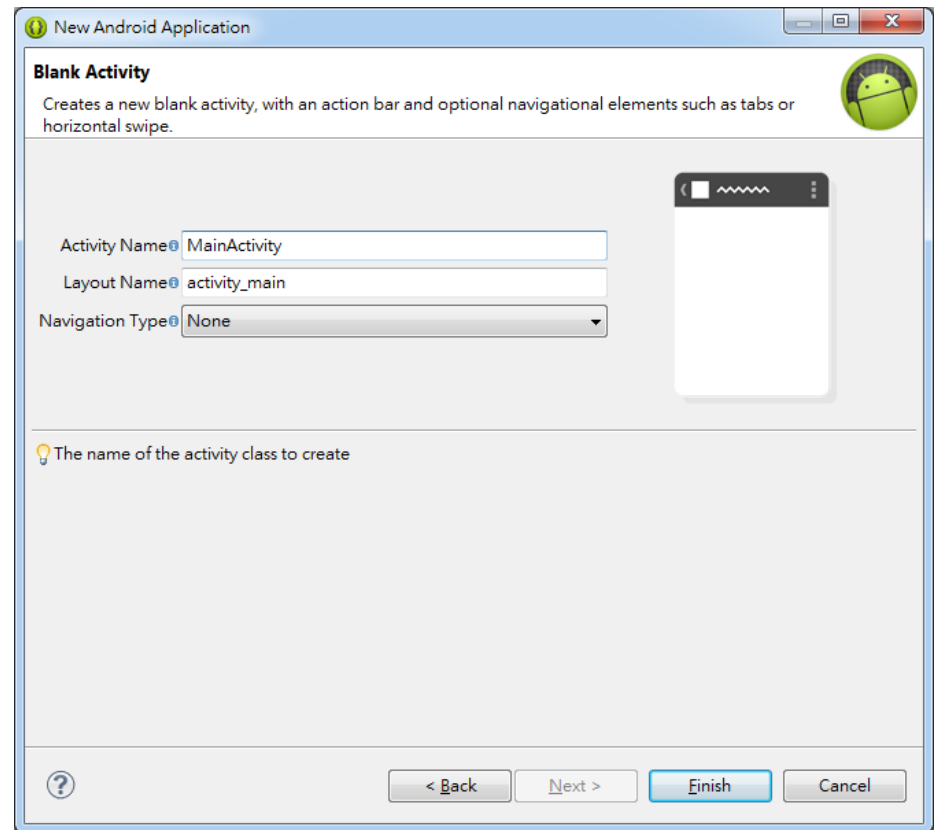


建立專案

設置 Activity，大部分使用 Blank Activity

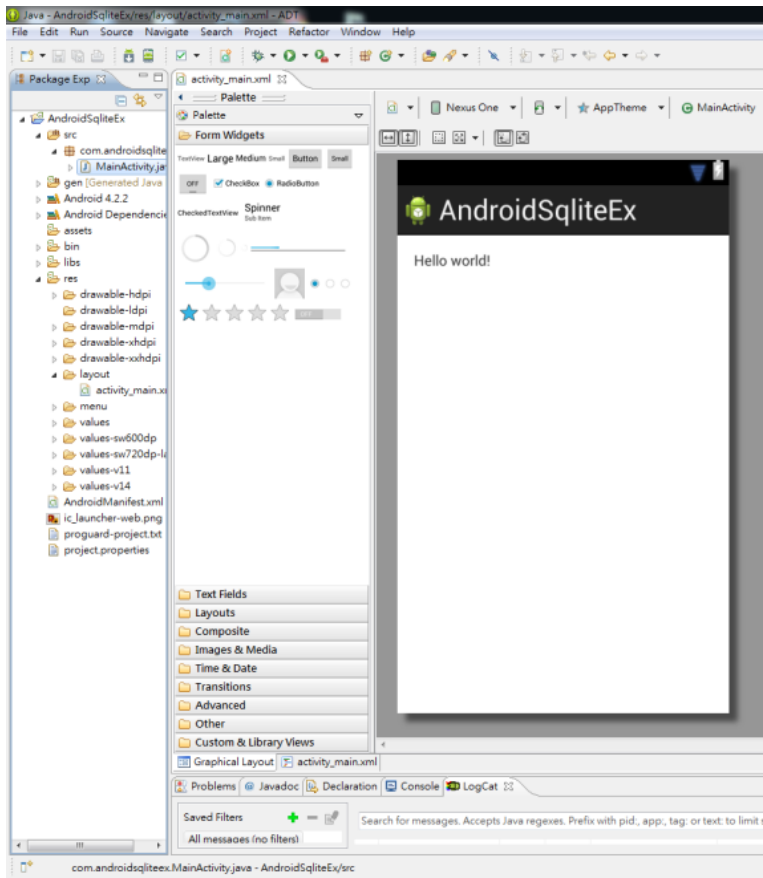


設定 Activity 跟 Layout 名稱



建立專案

Project 新增完畢



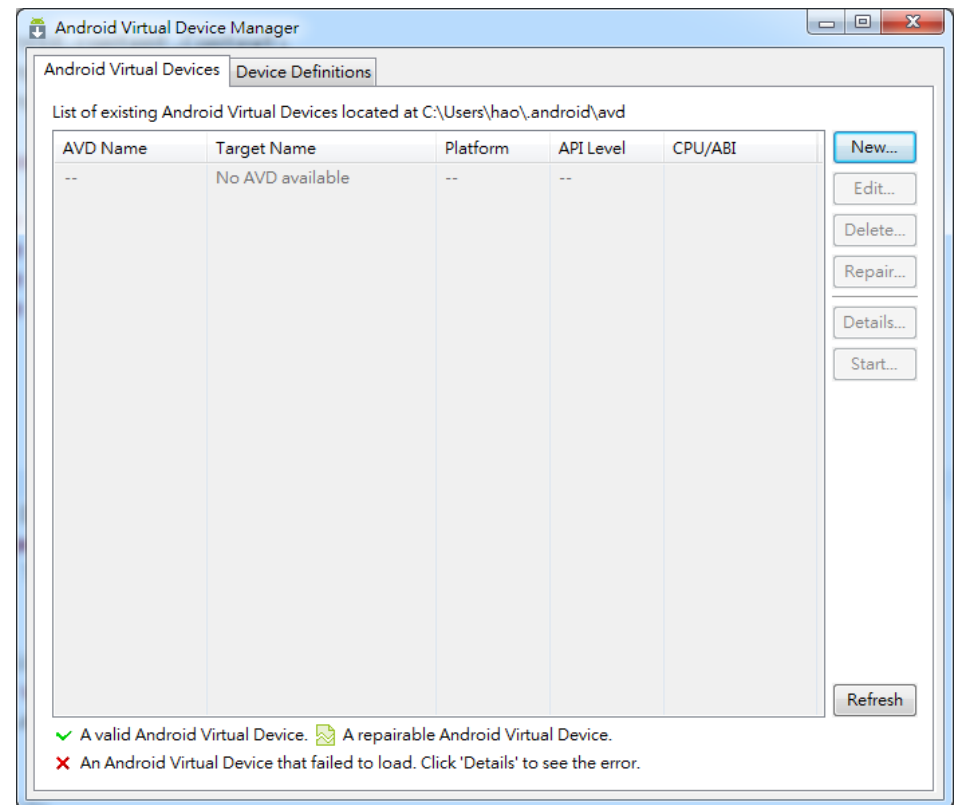
建置模擬器

建置模擬器

按下手機圖示準備新增一個模擬器

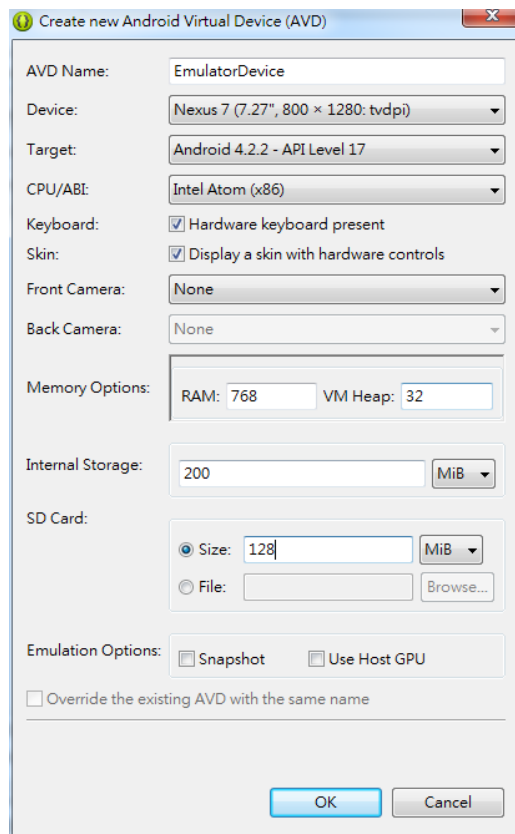


按下 **New** 新增模擬器

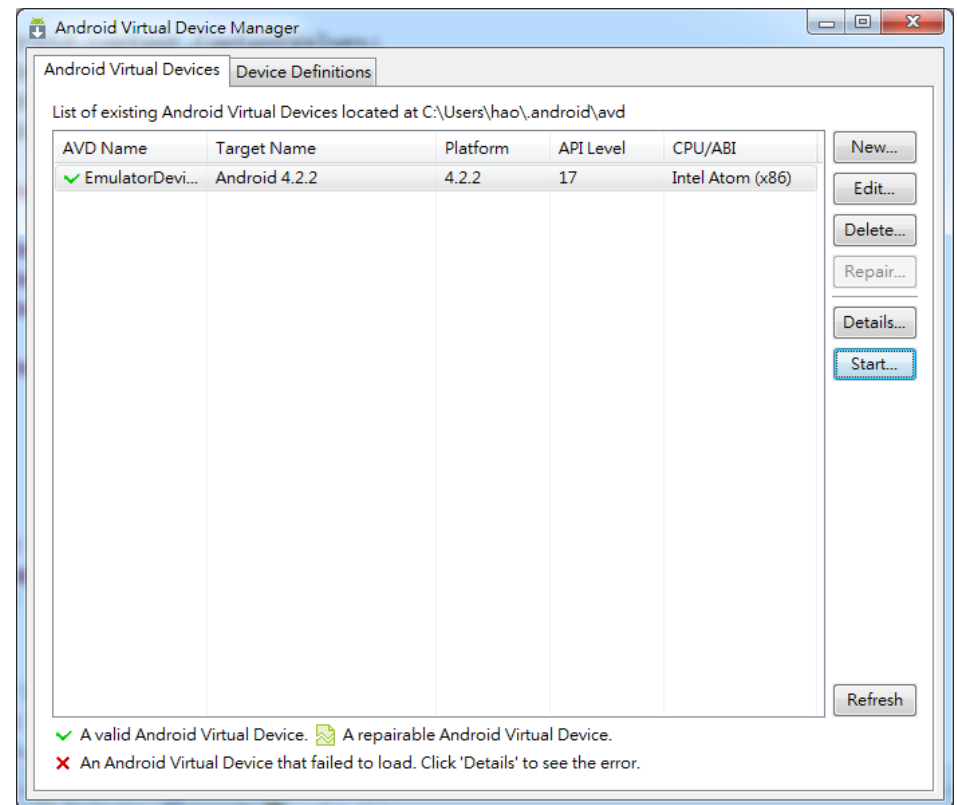


建置模擬器

進行模擬器設置，設置完成後按下OK

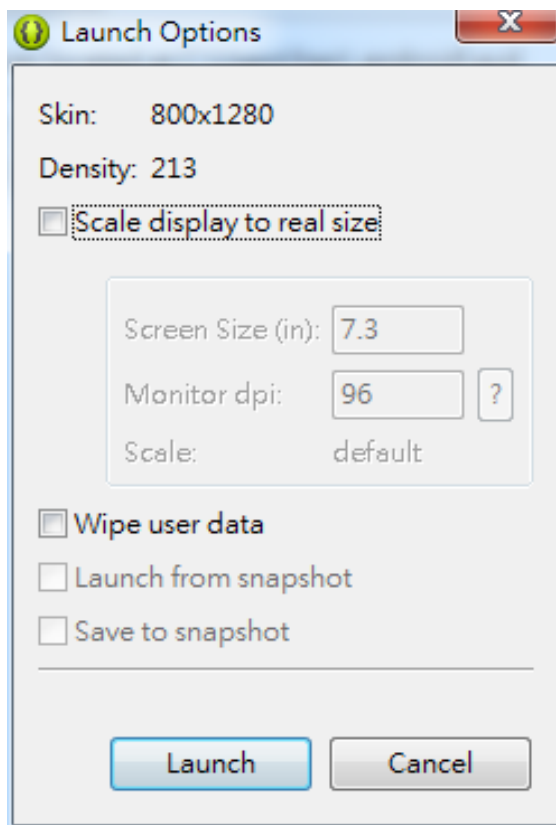


按下 **Start** 啟動模擬器

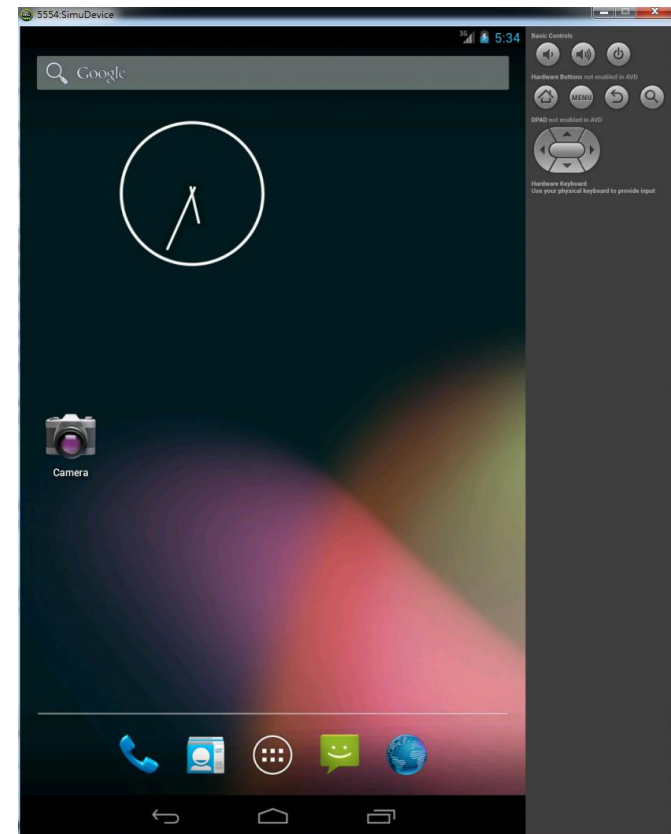


建置模擬器

出現 Launch Options 視窗按下 Launch 即可



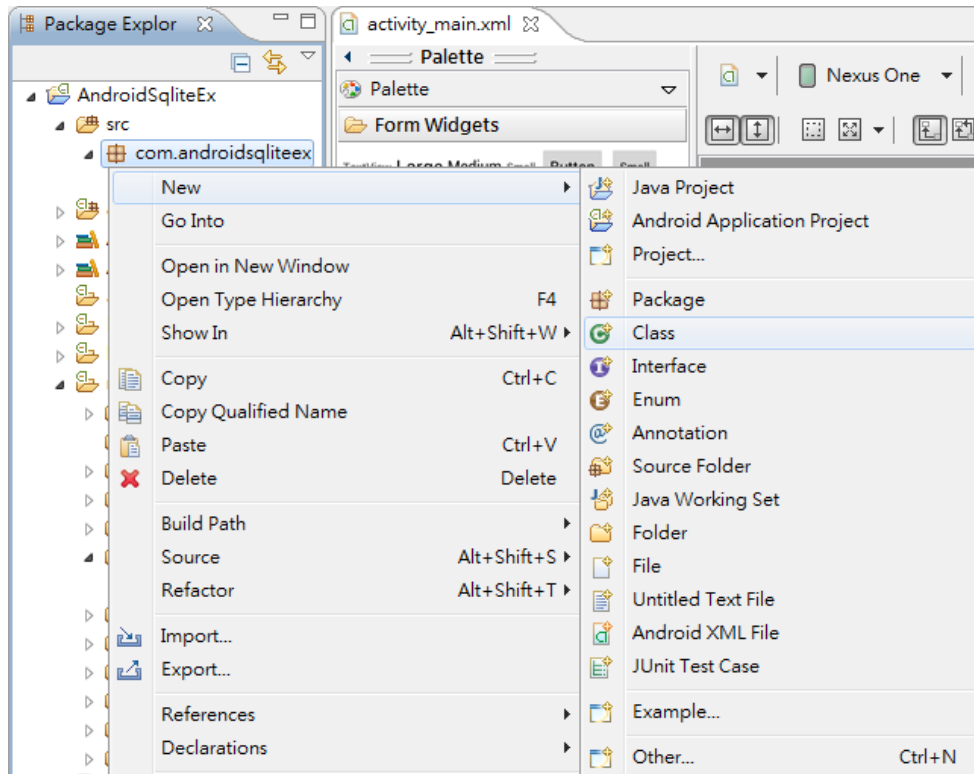
模擬器剛開始啟動會需要一些時間，此介面即是 Android 系統的畫面



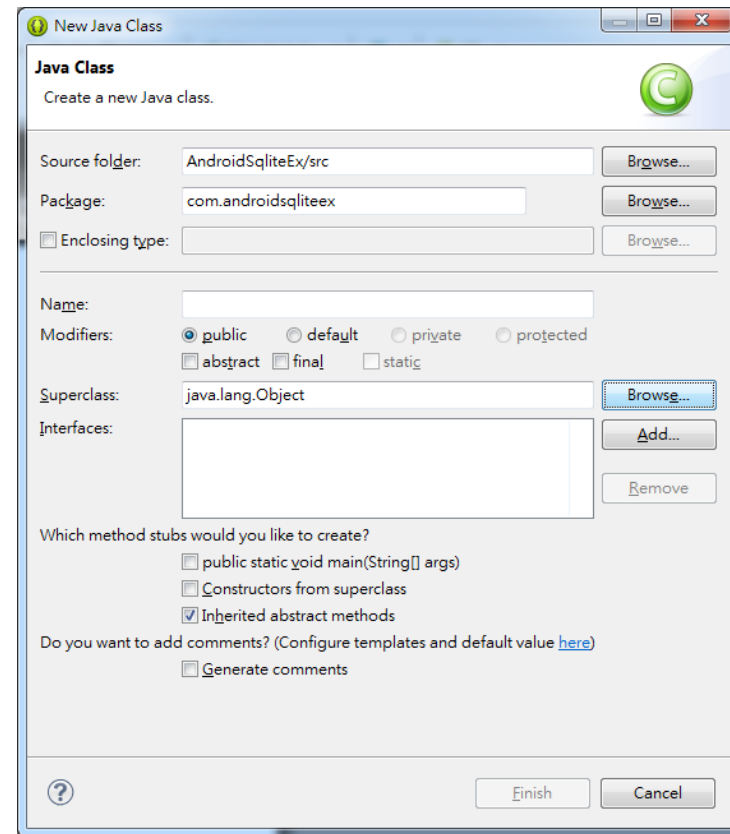
建立資料庫

建立資料庫

在 **package** 上按右鍵新增一個 **Class**

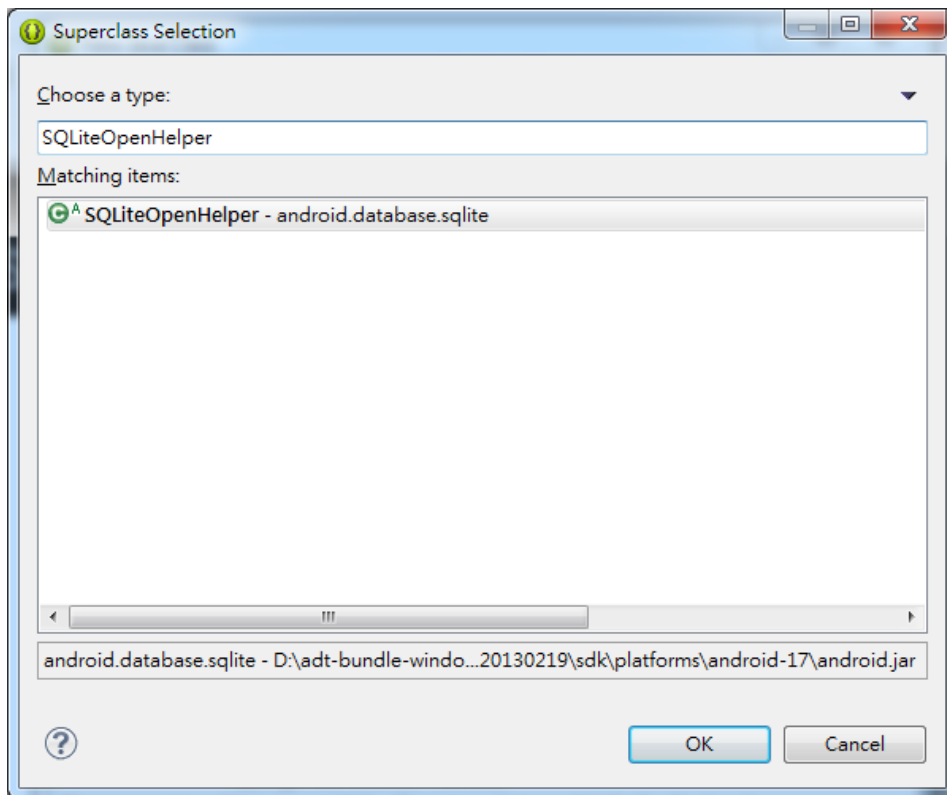


按下 **Browse** 進行設定

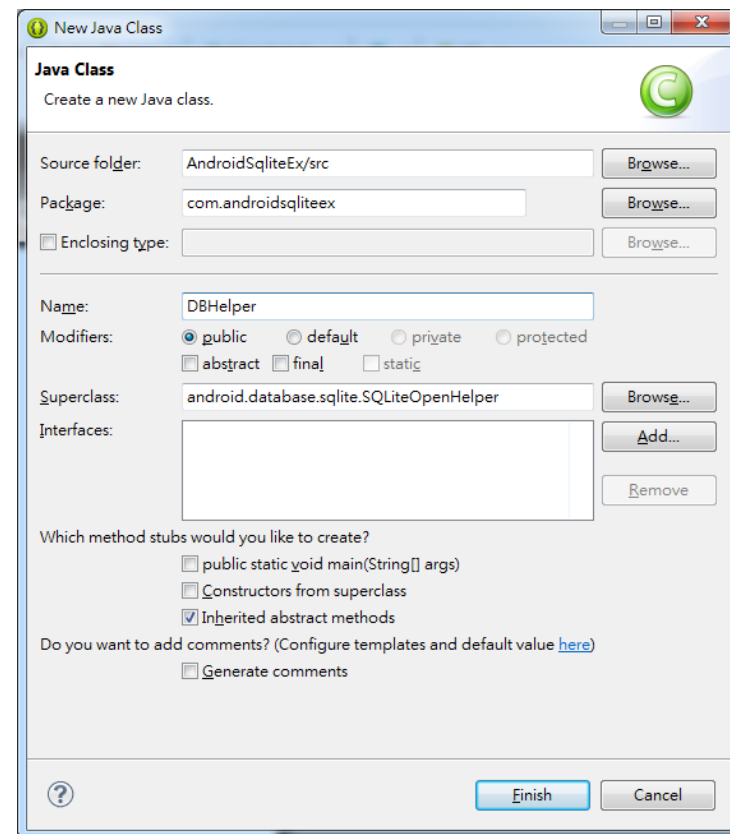


建立資料庫

選擇 SQLiteOpenHelper 類別



設定 Class 名稱按下 Finish



建立資料庫

新增完 Class 的初始程式碼

```
activity_main.xml DBHelper.java
1 package com.androidsqliteex;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteDatabase.CursorFactory;
6 import android.database.sqlite.SQLiteOpenHelper;
7
8 public class DBHelper extends SQLiteOpenHelper {
9
10     public DBHelper(Context context, String name, CursorFactory factory,
11                     int version) {
12         super(context, name, factory, version);
13         // TODO Auto-generated constructor stub
14     }
15
16     @Override
17     public void onCreate(SQLiteDatabase arg0) {
18         // TODO Auto-generated method stub
19     }
20
21
22     @Override
23     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
24         // TODO Auto-generated method stub
25     }
26 }
27
28
29
```

建立資料庫

```
activity_main.xml DBHelper.java
1 package com.androidsqliteex;
2
3 import android.content.ContentValues;
4 import android.content.Context;
5 import android.database.sqlite.SQLiteDatabase;
6 import android.database.sqlite.SQLiteOpenHelper;
7
8 public class DBHelper extends SQLiteOpenHelper {
9
10     private static final String DATABASE_NAME = "SCHOOL";
11     private static final int DATABASE_VERSION = 1;
12     private static final String TABLE_NAME = "STD";
13
14     private static final String TABLE_CREATE =
15         "CREATE TABLE " + TABLE_NAME + " ( " +
16         " S_no INTEGER NOT NULL, " +
17         " S_name TEXT, " +
18         " S_phone TEXT, PRIMARY KEY (S_no)); ";
19
20     public DBHelper(Context context) {
21         super(context, DATABASE_NAME, null, DATABASE_VERSION);
22         // TODO Auto-generated constructor stub
23     }
24
25     @Override
26     public void onCreate(SQLiteDatabase db) {
27         db.execSQL(TABLE_CREATE);
28     }
29
30     @Override
31     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
32         db.execSQL(" DROP TABLE IF EXISTS " + TABLE_NAME);
33         onCreate(db);
34     }
35 }
```

- ❑ 各別宣告資料庫名稱、資料庫版本、資料表名稱
- ❑ 將新增資料庫的語法放入字串 TABLE_CREATE
- ❑ 更改 name 以及 version
- ❑ 透過 execSQL 執行 TABLE_CREATE
- ❑ 透過這段程式碼可進行資料庫改版

建立資料庫

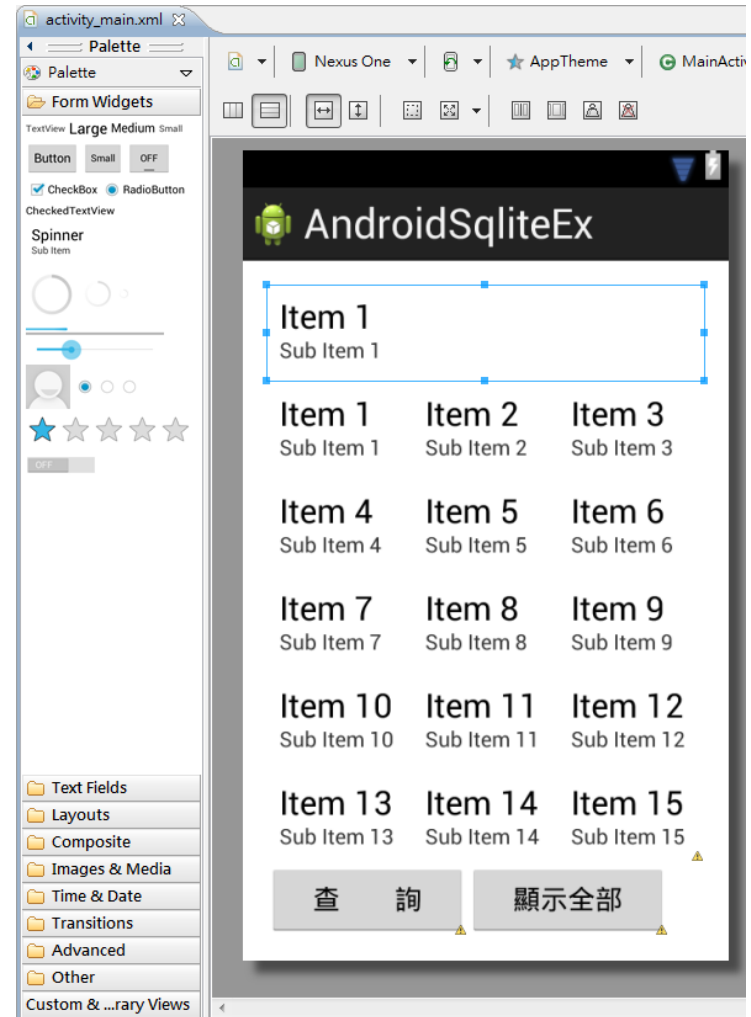
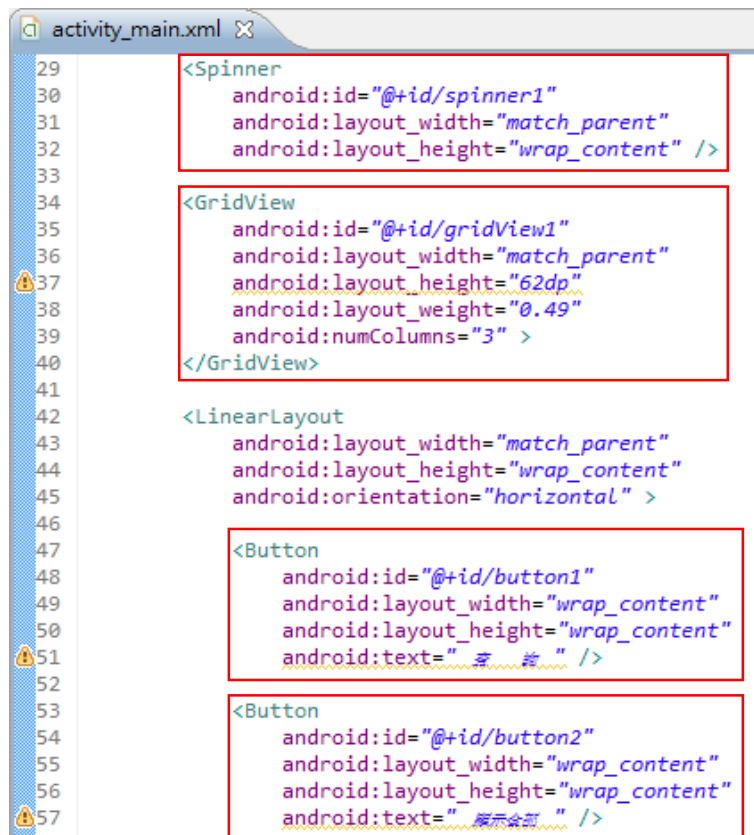
使用 ContentValues 可以進行資料庫的
Insert

```
36 public void insert() {  
37     SQLiteDatabase db = getWritableDatabase();  
38     ContentValues[] values = new ContentValues[3];  
39     for(int i=0; i<values.length; i++)  
40         values[i] = new ContentValues();  
41  
42     values[0].put("S_no", 610139006);  
43     values[0].put("S_name", "廖翔翔");  
44     values[0].put("S_phone", "0912345678");  
45  
46     values[1].put("S_no", 610139008);  
47     values[1].put("S_name", "黃豪豪");  
48     values[1].put("S_name", "0923456789");  
49  
50     values[2].put("S_no", 610139012);  
51     values[2].put("S_name", "張綸綸");  
52     values[2].put("S_phone", "0934567890");  
53  
54     for(ContentValues row : values){  
55         db.insert(TABLE_NAME, null, row);  
56     }  
57     db.close();  
58 }
```

介面設計

介面設計

在 `activity_main` 分別新增 `Spinner` 、
`GridView` 、 `Button` 等元件



介面設計

```
28 @Override
29 protected void onCreate(Bundle savedInstanceState) {
30     super.onCreate(savedInstanceState);
31     setContentView(R.layout.activity_main);
32
33     initDB();
34     findViews();
35     SelectAll();
36
37     // 開啟DB從STD table 讀取 S_no column 將每一筆 S_no 逐一放入 ArrayList
38     SQLiteDatabase db = dbHelper.getReadableDatabase();
39     ArrayList<String> SnoList = new ArrayList<String>();
40
41     Cursor C = db.query("STD", SnoColumns, null, null, null, null, null);
42     while (C.moveToNext()) {
43         String S_no = C.getString(0);
44         SnoList.add(S_no);
45     }
46     // 將 ArrayList 的內容放入 Spinner
47     ArrayAdapter<String> SnoAdapter = new ArrayAdapter<String>(this,
48         android.R.layout.simple_spinner_dropdown_item, SnoList);
49     SnoAdapter
50         .setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
51     spSno.setAdapter(SnoAdapter);
52 }
```

- 第 33 ~ 35 行：分別呼叫 3 個副程式
- 第 38 行：使程式可以讀取 Database
- 第 41 行：使用 **Cursor** 指向查詢的欄位
- 第 42 ~ 45 行：使 **Cursor** 逐一指向每一筆 record 並放入 **ArrayList**
- 第 47 ~ 51 行：將 **ArrayList** 放入 **Spinner**

介面設計

初始化 DBHelper Class 並呼叫 insert 副程式

```
54 // 初始化 dbHelper
55 private void initDB() {
56     if (dbHelper == null)
57         dbHelper = new DBHelper(this);
58     dbHelper.insert();
59 }
```

第 63 ~ 66 行：取得 activity_main 的原件
第 68 ~ 81 行：兩個 Button 元件 Click 事件

```
61 // 取得 XML 元件
62 private void findViews() {
63     spSno = (Spinner) findViewById(R.id.spinner1);
64     showGridView = (GridView) findViewById(R.id.gridView1);
65     selectBtn = (Button) findViewById(R.id.button1);
66     selectAllBtn = (Button) findViewById(R.id.button2);
67
68     selectBtn.setOnClickListener(new Button.OnClickListener() {
69         @Override
70         public void onClick(View v) {
71             spinSelect = "S_no =" + spSno.getSelectedItem().toString();
72             selectClick();
73         }
74     });
75
76     selectAllBtn.setOnClickListener(new Button.OnClickListener() {
77         @Override
78         public void onClick(View v) {
79             SelectAll();
80         }
81     });
82 }
```

介面設計

第 84 ~ 104 行：將 **Spinner** 選取學號的學生資料顯示出來

第 106 ~ 125 行：是將所有學生資料表中的 record 全部顯示出來

```
84 private void selectClick() {
85     SQLiteDatabase db = dbHelper.getReadableDatabase();
86     ArrayList<String> AllList = new ArrayList<String>();
87     Cursor selectSno = db.query("STD", AllColumns, spinSelect, null, null,
88         null, null);
89     AllList.add("學號");
90     AllList.add("姓名");
91     AllList.add("電話");
92     while (selectSno.moveToNext()) {
93         String S_no = selectSno.getString(0);
94         String S_name = selectSno.getString(1);
95         String S_phone = selectSno.getString(2);
96
97         AllList.add(S_no);
98         AllList.add(S_name);
99         AllList.add(S_phone);
100     }
101     ArrayAdapter<String> AllAdapter = new ArrayAdapter<String>(this,
102         android.R.layout.simple_spinner_dropdown_item, AllList);
103     showGridView.setAdapter(AllAdapter);
104 }
```

```
106 private void SelectAll() {
107     SQLiteDatabase db = dbHelper.getReadableDatabase();
108     ArrayList<String> AllList = new ArrayList<String>();
109     Cursor C1 = db.query("STD", AllColumns, null, null, null, null, null);
110     AllList.add("學號");
111     AllList.add("姓名");
112     AllList.add("電話");
113     while (C1.moveToNext()) {
114         String S_no = C1.getString(0);
115         String S_name = C1.getString(1);
116         String S_phone = C1.getString(2);
117
118         AllList.add(S_no);
119         AllList.add(S_name);
120         AllList.add(S_phone);
121     }
122     ArrayAdapter<String> AllAdapter = new ArrayAdapter<String>(this,
123         android.R.layout.simple_spinner_dropdown_item, AllList);
124     showGridView.setAdapter(AllAdapter);
125 }
```

介面設計

按下 **Run** 執行 Project

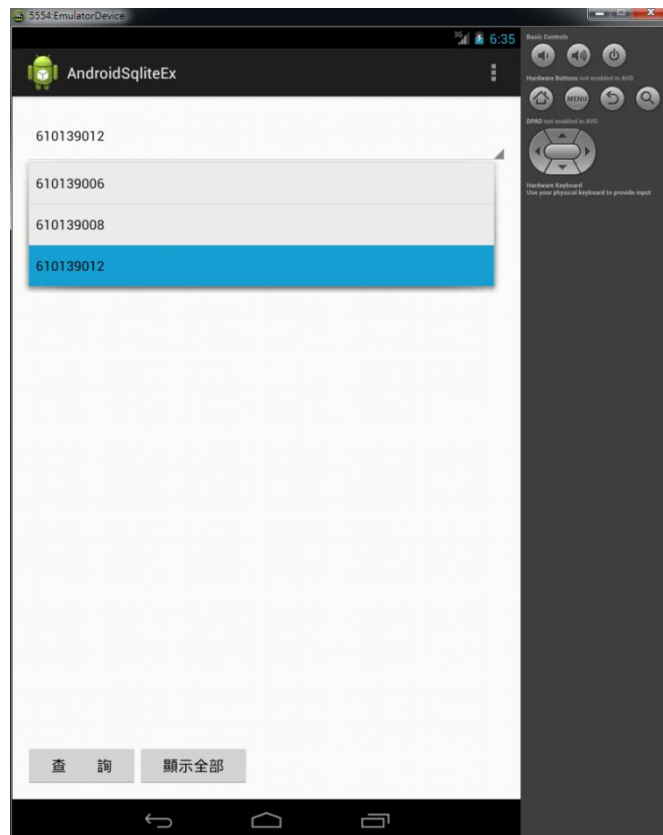


剛開始進入畫面會顯示學生資料表中全部的資料

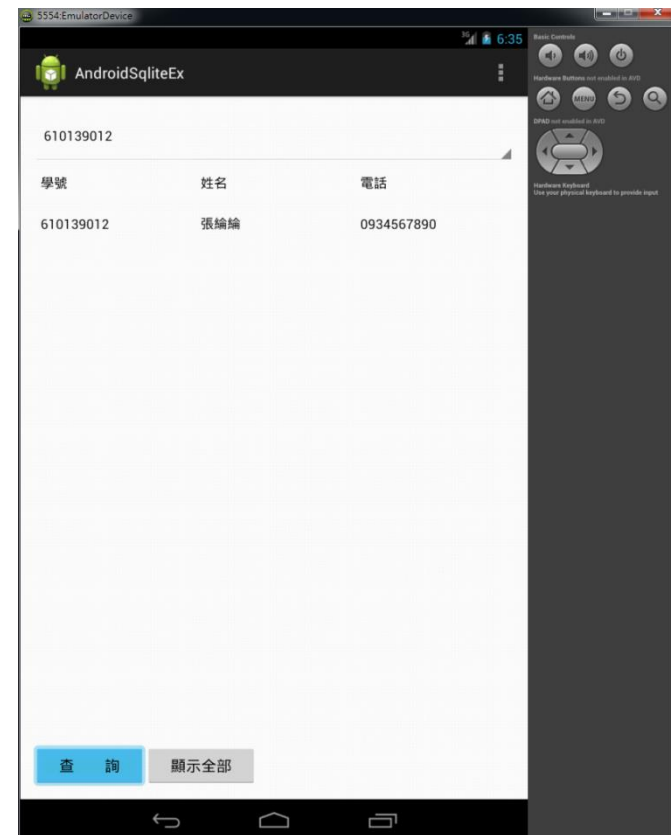


介面設計

選擇其中一個學號



按下查詢後就會顯示此學號學生的個人資料



介面設計

按下**顯示全部**則會顯示所有學生的資料

