

# Embedded System Project Laboratory

## 嵌入式系統專題實驗

### 實驗 (八)

## 期末 Project

授課教師：曾嘉影 老師

班 級：I3B

座 號：54

姓 名：黃子睿

## 題目 1：

期末 Project

### 實驗內容及問題說明

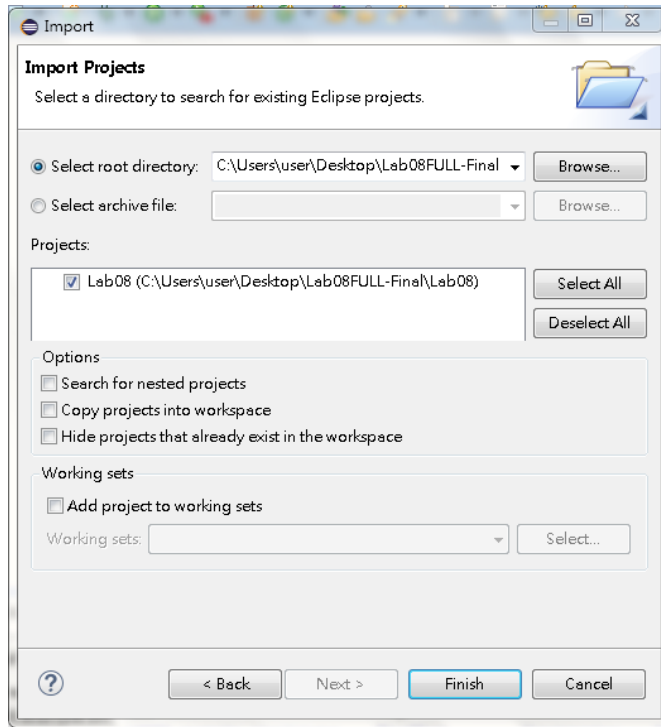
範例檔 Lab08FULL-Final.zip (網路硬碟一:分享名稱I3090)為Android應用SQLite資料庫的範例，請設計一個利用到SQLite資料庫功能並結合”實驗-5-初見Google Map”、”實驗-6-路徑規劃”、”實驗-7-附近搜尋”之期末Project

PS:

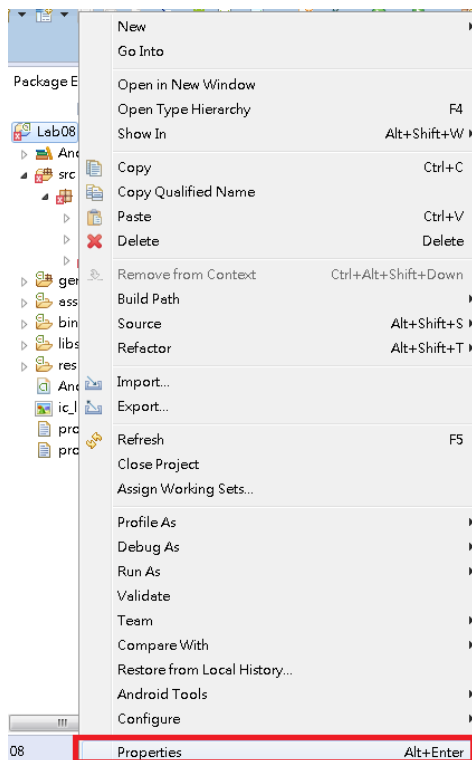
1. 有關SQLite可參考 <http://jim690701.blogspot.tw/2012/06/android-sqlite.html>
2. 可參考android market或google play的app應用程式發揮你的創意完成一個功能完整的Project
3. 記得記錄實驗步驟及結果！

# 實驗程式及輸出畫面

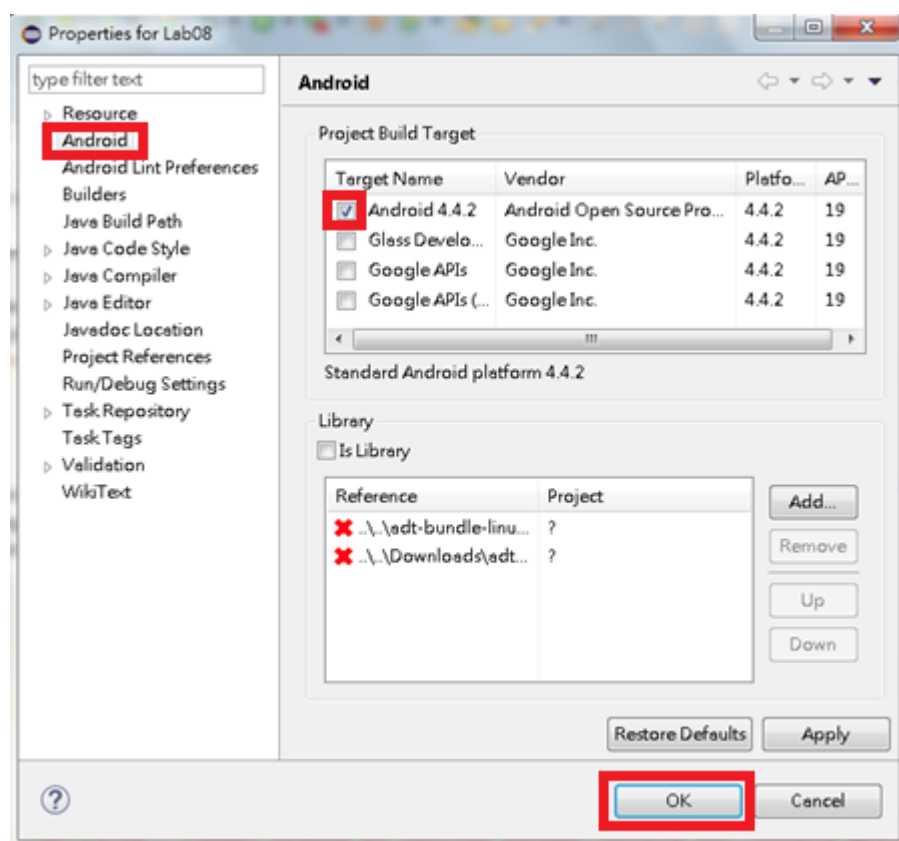
## 1. 匯入網路硬碟的範例檔案



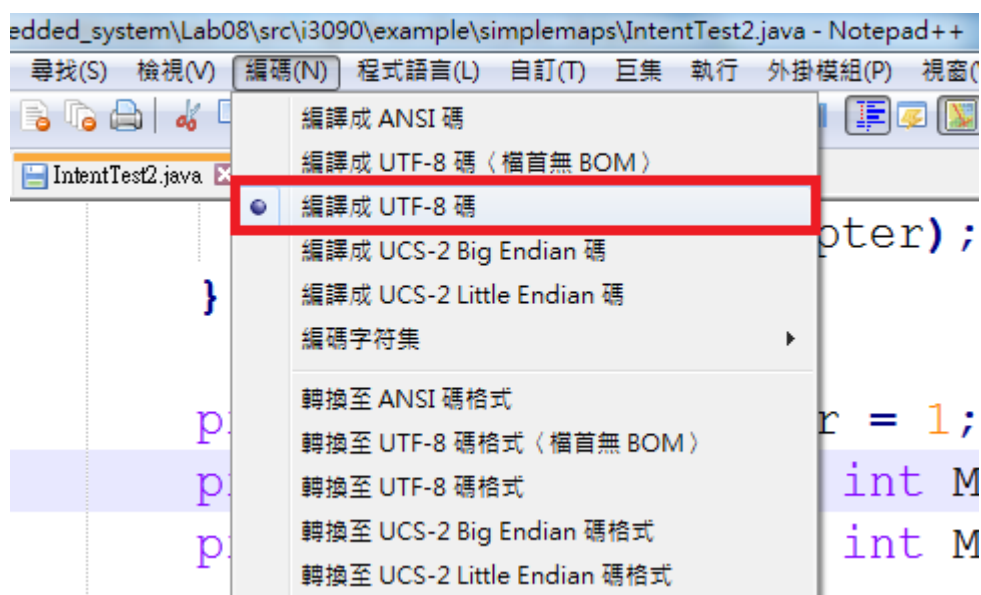
## 2. 對專案點選右鍵選擇Properties



3. 點選Android，勾選Android 4.4.2更改版本後按下確定



4. 將java檔案先以Notepad++打開後將編碼改成UTF-8 避免亂碼



## 5. 修改IntentTest2.java檔案，為每個case都加上break修改BUG，並修改

DELETE方法，同時增加了清空全部的項目

```
private int mNoteNumber = 1;
protected static final int MENU_INSERT = Menu.FIRST;
protected static final int MENU_DELETE = Menu.FIRST+1;
protected static final int MENU_CLEAR = Menu.FIRST+2;

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // TODO Auto-generated method stub
    menu.add(0, MENU_INSERT, 0, "新增");
    menu.add(0, MENU_DELETE, 0, "刪除");
    menu.add(0, MENU_CLEAR, 0, "清空");
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // TODO Auto-generated method stub
    switch(item.getItemId()) {
        case MENU_INSERT:
            String noteName = " Note " + mNoteNumber++;
            mDbHelper.create(noteName);
            fillData();
            break;
        case MENU_DELETE:
            if(checkValid(row+"") == 1)
                updateID(row);
            row = 0;
            fillData();
            break;
        case MENU_CLEAR:
            num=0;
            mDbHelper.clear();
            fillData();
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

## 6. 修改DB.java檔案，增加clear函式。

```
//remove database
public void clear() {
    db.delete(DATABASE_TABLE, null, null);
}
```

## 7. 修改IntentTest2.java和DB.java檔案，修改delete按鈕功能，同時增加

合法性的判定防止BUG產生，其中updateID的功能，將會即時更新

row和ID，並在點擊後更改按鈕顏色

```
deletetbtn.setOnClickListener( new OnClickListener() {
    public void onClick(View v) {
        String Text = edit.getText().toString();

        if(checkValid(Text) == 1)
            updateID(Integer.valueOf(Text));
        row = 0;
        v.setBackgroundColor(Color.rgb(239, 2477, 255));
    }
});

private void updateID(int R){
    mDbHelper.delete(R);
    int t = mDbHelper.getCount();

    for(int i = R; i <= t; i++){
        mDbHelper.updateRow(i+1,i);
        String s = mDbHelper.get(i).getString(1);
        int j;
        for(j = 0; j < s.length(); j++){
            if(!Character.isDigit(s.charAt(j)))
                break;
        }
        String neww = i+"";
        neww += s.substring(j, s.length());
        mDbHelper.update(i, neww);
    }
    fillData();
    num = t;
}

private int checkValid(String Text){
    int valid = 1;

    for(int i = 0; i < Text.length(); i++){
        if(!Character.isDigit(Text.charAt(i))){
            valid = 0;
            break;
        }
    }
    if(Text.equals("") || Text.charAt(0)=='0')
        valid = 0;
    edit.setText("");
    return valid;
}

//update2
public boolean updateRow(long rowId, long newR) {
    ContentValues args = new ContentValues();
    args.put(KEY_ROWID, newR);

    return db.update(DATABASE_TABLE, args, KEY_ROWID + "=" + rowId, null) > 0;
}
```

8. 修改record按鈕功能，為noteName新增加ID在前面，並在點擊後更改

按鈕顏色

```
recordbtn.setOnClickListener( new OnClickListener() {  
    public void onClick(View v) {  
        String noteName = "從" + from + "到" + to;  
        mDbHelper.create(++num + noteName);  
        fillData();  
        v.setBackgroundColor(Color.rgb(74, 172, 255));  
    }  
});
```

9. 增加一個導航按鈕，用以切回上一個Activity直接導航，並傳參數過去

```
startbtn.setOnClickListener( new OnClickListener() {  
    public void onClick(View v) {  
        bundle.clear();  
        bundle.putString("KEY_TEXT1", from);  
        bundle.putString("KEY_TEXT2", to);  
        intent.putExtras(bundle);  
        IntentTest2.this.setResult(MainActivity.RESULT_OK, intent);  
        IntentTest2.this.finish();  
    }  
});
```

10. 將所點擊的ID呈現在上方的輸入框，並更改導航位置和更改顏色

```
@Override  
protected void onListItemClick(ListView l, View v, int position, long id) {  
    super.onListItemClick(l, v, position, id);  
    row = (int)id;  
    edit.setText(row+"");  
  
    String s = mDbHelper.get(row).getString(1);  
  
    String[] cmds = s.split(" ");  
    int i = 1;  
    for(String d:cmds){  
        if(i == 4)  
            from = d;  
        if(i == 8)  
            to = d;  
        i++;  
    }  
    v.setBackgroundColor(Color.rgb(197, 230, 255));  
}
```

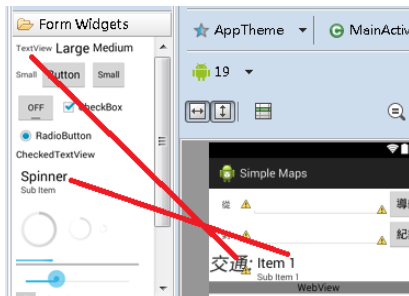
## 11. 回到MainActivity.java覆寫 onActivityResult方法，用以將

IntentTest2.java傳回的參數接收住，並直接呼叫map的找路徑涵式

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 0 && resultCode == Activity.RESULT_OK) {
        Log.d("123", "sucesss");
        bundle = data.getExtras();
        from=bundle.getString("KEY_TEXT1");
        to=bundle.getString("KEY_TEXT2");
        edit.setText(from);
        edit2.setText(to);
        webView.loadUrl( "javascript:route()");
    }
}
```

## 12. 來到map\_v3.xml拉出TextView，並改成交通方式，和拉出Spinner



## 13. 回到MainActivity.java增加spinner用以選擇交通方法。

```
private Spinner spinner;
private String[] list = {"開車", "步行", "單車", "大眾運輸"};
private ArrayAdapter<String> listAdapter;
private String mode="DRIVING";

spinner = (Spinner)findViewById(R.id.mySpinner);

listAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, list);
spinner.setAdapter(listAdapter);

spinner.setOnItemSelectedListener(new OnItemSelectedListener(){
    @Override
    public void onItemSelected(AdapterView<?> arg0, View arg1,int position, long arg3) {
        if(position == 0)    mode = "DRIVING";
        if(position == 1)    mode = "WALKING";
        if(position == 2)    mode = "BICYCLING";
        if(position == 3)    mode = "TRANSIT";
        webView.loadUrl( "javascript:route()");
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {
        // TODO Auto-generated method stub
    }
});
```

#### 14. 增加getMode方法，將Map與Android做連結

```
@JavascriptInterface
public String getMode(){
    return mode;
}
```

#### 15. 修改map\_v3.html增加selectedMode讓其能夠與Android做連結

```
function route() {
    from=Android.getFrom();
    to=Android.getTo();
    selectedMode=Android.getMode();
    var request = {
        origin: from ,
        destination: to,
        travelMode: google.maps.TravelMode[selectedMode]
    }
```

#### 16. 修改路徑顏色，改成青色

```
directionsRenderer = new google.maps.DirectionsRenderer({
    polylineOptions:{strokeColor: "cyan"}
});
```

## 實驗結果

初始畫面將會看到兩個輸入框和交通方式





先輸入 大同大學 和 中山國小 並按下導航，可以看到青藍色路徑



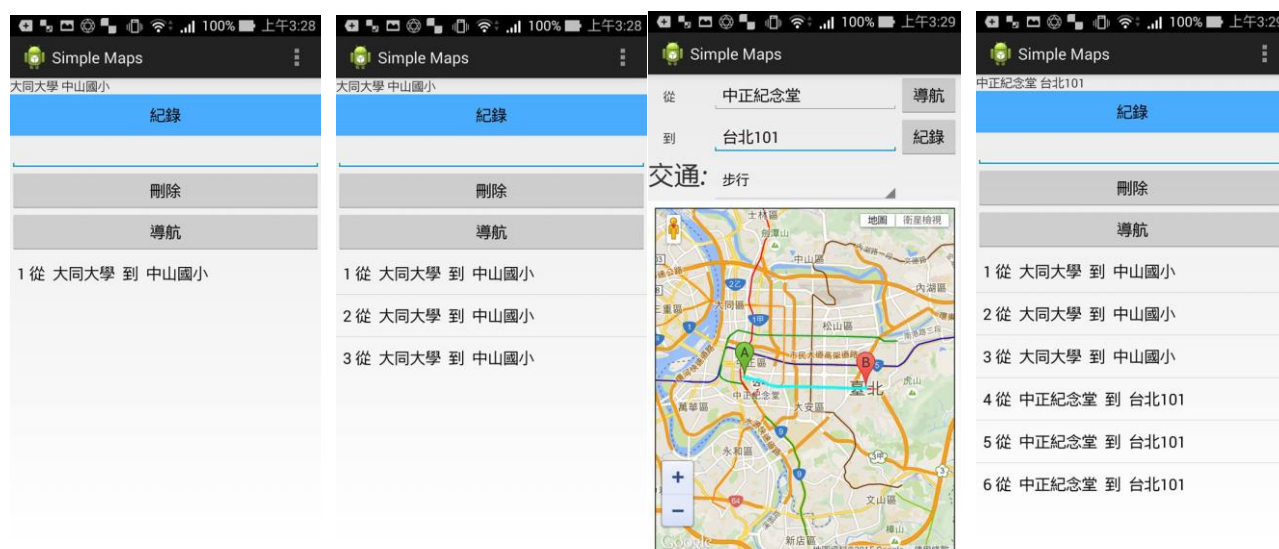
接著更改交通方式改成步行，會發現路徑有改變



按下紀錄後切到第二個畫面，由於沒有紀錄因此必須再按下一次紀錄  
增加新的紀錄



增加完紀錄後記錄的按鈕會變成藍色的，且記錄欄位還標有ID，接著  
連續增加多筆不同紀錄



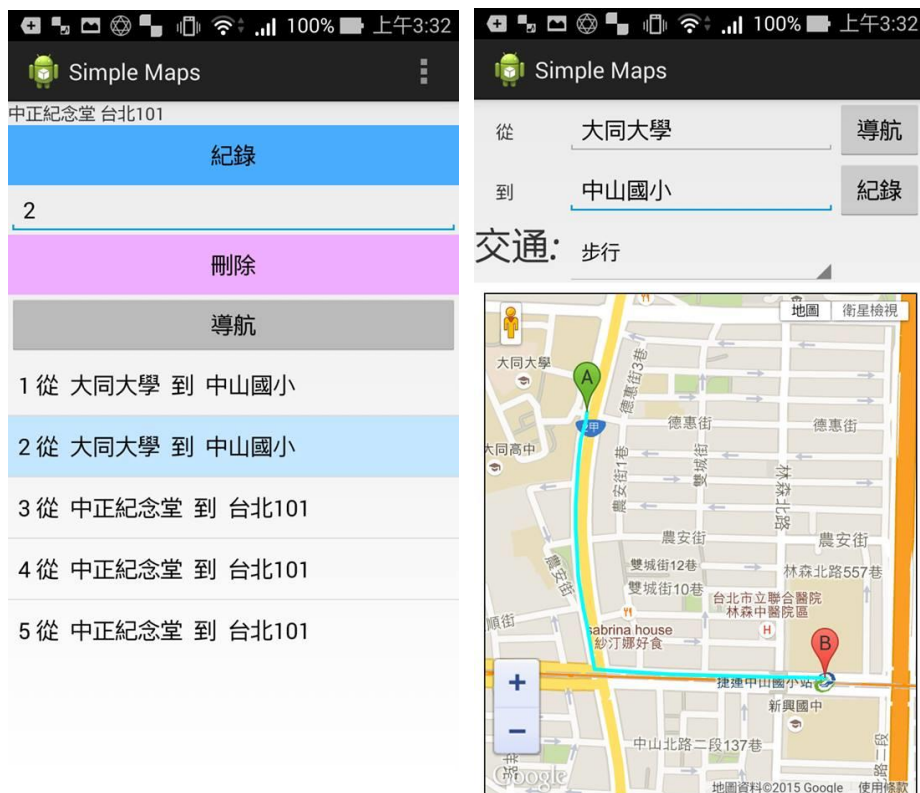
點選ID 3後，所選擇的紀錄會被標以顏色，並在上方輸入欄填入ID



按下刪除後，刪除按鈕會變色，下方欄位會將記錄清空，同時更新ID，且上方輸入框也會跟著清空



接著點選ID 2後選擇導航按鈕，會切回原始頁面，並在輸入框填入原始紀錄，同時搜尋該筆路徑



測試右上角的設定按鈕，選擇ID 3 並按下刪除，可以清除該筆紀錄



新增Note筆記，底下會多出筆記欄位



測試清空按鈕，可以將所有紀錄都清空



## 實驗紀錄

### 在實驗中遇到一些問題

Q: 看不到新建立的.db檔?

Ans: 上網查資料後，發現是因為沒有root的權限關係。

Q: 想刪掉同個row，但刪除後row又在不同行怎麼辦?

Ans: 直接從資料庫裡即時更新row。

Q: 不太了解SQLite是什麼?

Ans: 這方面由於老師快速帶過，所以寫程式前花很大時間在研究程式碼

後來了解到這是一個用來存手機資料庫的東西，裡面的語法基本上跟SQL還蠻相像的。

Q: 怎麼取用資料庫裡的資料?

Ans: 一開始很煩惱這問題，還去研究cursor怎麼使用，打算自己寫一個query來查詢，後來同學跟我說DB.java裡面已經有寫好get的方法了還教我怎麼使用，非常感謝他。

## 思考分析

這次學到了很多事情，發現原來Android可以直接與google Map做溝通連結，將兩者的參數互相傳遞，這是跟以前實驗不同之處。以往實驗感覺偏向寫網頁與Android做分離，這次則是將兩者融合再一起非常有趣。而這次的SQLite手機版的資料庫取用，要如何使用也收穫蠻多的，知道可以用SQLiteDatabase下去做新增修改刪除查詢，雖然與網頁版的SQL用法不太一樣卻也非常方便。這次因為有上次的debug經驗，了解到Log.d的好用之處，因此除錯方面也進步不少。從一開始到現在學期結束有了許多收穫，了解到手機程式原來那麼有趣、不懂的資料該如何查詢...等，將來如果從事手機app相關方面的工作，我想必定能夠寫出好用的app程式。