使用單位：技術處　　姓名：

技術：Sqlite 與 Xml 下載技術查核表(進階二)　　開始時間：

Sqlite 與 Xml 教學查核表(進階二)

說明：此文件接續上份文件的第三部分完成後，有關於第四部份與後續的教學介紹。

- 有關於此教學查核表學習前的必要條件：

  A. 對於軟體程式寫作已經有初步的了解。
  B. 已經對於 Android JAVA 撰寫有一定的能力。
  C. 已經完成 Sqlite 與 XML 教學查核表的相關內容。

- 有關於本查核表的最終有價值產品：

  有能力可以從 Server 上面下載 Xml Data 並且存入 Sqlite 資料庫中。最後用 Sql select 的方式將資料整理成列表，並且進一步點選內容。

- 查核表完成的時間：4 個小時以內。

- 注意事項：請依照步驟進行，不可以任意跳過未完成的步驟。若該步驟是關於查核或是時做練習，必須要將作業(成品)，交給主管進行查核，該項目請由主管簽名。

執行步驟

第一部分：設計概念與準備事項

1. 承接上一份查核表，我們已經完成了食物種類的查核表下載工作，接著要處理列表動作。

   以下是畫面是意圖參考。

2. 請打開 SQL 與 XML 查核表所完成的專案 SqliteTest，如果這個查核表尚未完成到第三部分，請先完成。 _____

3. Layout 設計：請依照步驟 1 的設計概念設計食物種類的 Layout 檔案，取名為 nutrition_type.xml。(請注意，我們將使用 custom_link_icon.png 當作按鈕的製作圖案。可以從此網址進行下載，將可找到此圖檔) _____
http://www.v7idea.com.tw/download/testAndroidCustonObj.zip

以下是 layout 範例：

處理一：(此範例將會用到兩個指定顏色，請先新增到/res/values/strings.xml

```
<color name="gray_type1">#DEDEDE</color>
<color name="gray_type2">#CDCDCD</color>
```

處理二：請先匯入 custom_link_icon.png 到/res/drawable 目錄中。

處理三：layout 範例：

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="@color/gray_type1" >
        <TextView android:id="@+id/nutritionTypeText"
            android:layout_width="60dp"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_centerVertical="true"
            android:layout_marginLeft="5dp"
            android:layout_marginRight="5dp"
            android:text="A09"
            android:textSize="18dp" />
        <TextView android:id="@+id/nutritionTypeName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerVertical="true"
            android:layout_marginRight="5dp"
            android:layout_toLeftOf="@+id/imageView1"
            android:layout_toRightOf="@+id/nutritionTypeText"
            android:text="食物種類 2"
            android:textSize="18dp" />
```

```
    <ImageView android:id="@+id/imageView1"
        android:layout_width="50px"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:layout_marginRight="5dp"
        android:src="@drawable/custom_link_icon" />
</RelativeLayout>
```

4 Layout 設計：畫面捲動所需要的 layout，取名為 scroll_list.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <ScrollView
        android:id="@+id/contactScroll"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true" >
        <LinearLayout
            android:id="@+id/contactListArea"
            android:layout_width="match_parent"
            android:layout_height="match_parent" >
        </LinearLayout>
    </ScrollView>
</RelativeLayout>
```

5 Layout 設計：請依照步驟 1 的設計概念設計食物種類的 Layout 檔案，取名為 nutrition_food.xml。

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="@color/gray_type1" >
        <TextView
            android:id="@+id/foodName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:layout_marginLeft="5dp"
```

```
        android:layout_marginRight="10dp"
        android:layout_marginTop="5dp"
        android:layout_toLeftOf="@+id/clickButton"
        android:text="麥當勞薯條"
        android:textSize="18dp" />
    <ImageView
        android:id="@+id/clickButton"
        android:layout_width="35px"
        android:layout_height="35px"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="5dp"
        android:layout_marginTop="5dp"
        android:src="@drawable/custom_link_icon" />
    <TextView
        android:id="@+id/calText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/foodName"
        android:layout_below="@+id/clickButton"
        android:layout_marginLeft="5dp"
        android:text="350 大卡"
        android:textSize="15dp" />
    <TextView
        android:id="@+id/unitText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/foodName"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_toLeftOf="@+id/clickButton"
        android:text="100 克"
        android:textSize="15dp" />
</RelativeLayout>
```

6 我們為了增加互動的效果，所以在按鈕按下的同時增加一個小的動畫，讓使用者可以知道自己已經觸發了按鈕。因此製作一個動畫檔預備按鈕觸發事件中使用。

Step 1: 在/res/ 目錄中增加一個新的目錄/anim

Step 2: 在這個目錄下增加一個動畫定義檔案：button_animation.xml

Step 3: 在這個檔案中增加動畫的描述 xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android" >
 <scale android:interpolator="@android:anim/accelerate_decelerate_interpolator"
        android:fromXScale="1.0"
        android:toXScale="0.9"
        android:fromYScale="1.0"
        android:toYScale="0.9"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="100"
        android:fromAlpha="1.0"
        android:toAlpha="0.7"
   />
</set>
```

第二部分：第一個畫面的設計 (食物分類：MainActivity.java)

1. 設計概念：MainActivity.java 承襲前一個查核表中的流程，在一開始會檢查是否有資料是否已經存在資料庫中等動作，在此將流程做一個完整的描述，並且加入產生列表的流程：
   Step1：使用 sql command 去確認資料庫是否資料： *select * from nutritiontype*
   Step2：如果沒有資料，就必須從網站中下載 XML 檔案，並且將資料放入資料庫中。
   Step3: 使用 sql command 去篩選出資料，產生食物類別的列表。

2. 增加一個自訂類別：NutritionType (這個類別不繼承其他任何物件) ,去實作單一食物類別的物件
   (在這個專案中增加一個新的類別，並且取名為 NutritionType.)
   以下是相關範例：

```
/**
 * 食物類別物件
 * @author Louis_Chuang
 * 這個類別物件是用來示範如何自訂一個類別物件
 */

public class NutritionType {

    private String typeCode;
    private String typeName;
    private int backgroundColor;
    private Intent activityIntent;
```

```
    private RelativeLayout thisLayout;
    private Context parentContext;

    /**
     * 建構這個物件類別，用來產生食物類別的物件
     * @param context 母容器
     * @param typeCode 這個類別代碼
     * @param typeName 類別名稱
     * @param backgroundColor 背景顏色
     * @param activityIntent 導入下個畫面所需要使用的Intent
     */

    public NutritionType(Context context, String typeCode, String typeName, int
backgroundColor, Intent activityIntent) {

        // 將變數值存入Grobal中，以便日後進行使用

        this.typeCode = typeCode;
        this.typeName = typeName;
        this.backgroundColor = backgroundColor;
        this.activityIntent = activityIntent;
        this.parentContext = context;

        thisLayout = generateLayout();

    }

    /**
     * 產生這個類別物件 (僅限內部使用)
     * @return 回傳RelativeLayout物件
     */

    private RelativeLayout generateLayout() {

        RelativeLayout tempLayout;

        tempLayout =(RelativeLayout) ((Activity)
parentContext).getLayoutInflater().inflate(R.layout.nutrition_type, null);

        TextView typeCodeText = (TextView)
```

```
tempLayout.findViewById(R.id.nutritionTypeText);
            TextView typeNameText = (TextView)
tempLayout.findViewById(R.id.nutritionTypeName);
            ImageView clickButton = (ImageView)
tempLayout.findViewById(R.id.clickButton);

            tempLayout.setBackgroundResource(backgroundColor);
            typeCodeText.setText(typeCode);
            typeNameText.setText(typeName);

            // 設定按紐的按下動作

            clickButton.setOnClickListener(new OnClickListener() {

                @Override
                public void onClick(View thisView) {

                    // 設定按下後的動畫

                    Animation buttonAnimation = null;
                    buttonAnimation = AnimationUtils.loadAnimation(parentContext,
R.anim.button_animation);
                    buttonAnimation.setAnimationListener(new AnimationListener() {
                        @Override
                        public void onAnimationEnd(Animation animation) {
                            parentContext.startActivity(activityIntent);
                        }
                        @Override
                        public void onAnimationRepeat(Animation animation) {
                        }
                        @Override
                        public void onAnimationStart(Animation animation) {
                        }});
                    thisView.startAnimation(buttonAnimation);
                }});
            return tempLayout;

        }

        /**
```

```
     * 讓使用者可以取來產生出來的layout;
     * @return 產生完成的layout;
     */
    public RelativeLayout getContentLayout() {

        return this.thisLayout;

    }
}
```

3. 實作：在 MainActivity 中實作產生列表的部分，請先增加一個新的方法 generateList()：

```
private void generateList() {
    // 以下程式碼是由進階版的查核表進行製作
    String sql = "";
    SqliteTestApp thisApp = (SqliteTestApp) getApplicationContext();

        MyDatabaseAdapter DBAdapter = null;
        DBAdapter = thisApp.getDBAdapter();

        RelativeLayout mainLayout = (RelativeLayout)
this.getLayoutInflater().inflate(R.layout.scroll_list, null);
        LinearLayout contactListArea = (LinearLayout)
mainLayout.findViewById(R.id.contactListArea);
        contactListArea.setOrientation(LinearLayout.VERTICAL);

        if(DBAdapter != null) {

            Log.d("DBAdapter","DBAdapter存在並且檢查");

            DBAdapter.open();

            sql = "select * from nutritiontype";
            Cursor resultCur = DBAdapter.query(sql, null);

            if (resultCur != null) {

                if(resultCur.moveToFirst()) {

                    // 表示有找到資料
```

```
                int i = 0;
                int backgroundColor = R.color.gray_type1;

                for (i = 0; i < resultCur.getCount(); i ++ ) {

                        String nutritionTypeId =
resultCur.getString(resultCur.getColumnIndex("typeid"));
                        String nutritionTypeCode =
resultCur.getString(resultCur.getColumnIndex("code"));
                        String nutritionTypeName =
resultCur.getString(resultCur.getColumnIndex("name"));

                        if(backgroundColor == R.color.gray_type2) backgroundColor =
R.color.gray_type1;
                        else backgroundColor = R.color.gray_type2;

                        Intent foodIntent = new Intent(this, NutritionList.class);
                        foodIntent.putExtra("typeid", nutritionTypeId);

                        NutritionType thisTypeObj = new NutritionType(this,
nutritionTypeCode, nutritionTypeName, backgroundColor, foodIntent);
                        if(thisTypeObj != null) {

                                RelativeLayout singleLayout =
thisTypeObj.getContentLayout();
                                if(singleLayout != null) {

                                        contactListArea.addView(singleLayout);

                                }

                        }

                        resultCur.moveToNext();

                }
        }

    } else {
```

```
                Log.d("resultCur","無法產生cursor");
                Toast.makeText(this, "找不到任何食物類別的資料",
Toast.LENGTH_LONG).show();

            }

            resultCur.close();
            DBAdapter.close();

            setContentView(mainLayout);


        }

    }
```

4 由於以上的方法會呼叫到新的 Class，必須要新增一個新的 java 檔案 NutritionList.java 這個檔案繼承 Activity.

5 請在 MainActivity 的 OnCreate() 方法與非同步背景工作 `downloadBackgroundTask` 的流程中加入 generateList()。

第三部分：第二個畫面的設計 (食物分類：NutritionList.java)

1. 設計概念：
   Step1：使用 sql command 去確認資料庫是否資料： *select * from nutritiontfood*
   Step2：如果沒有資料，就必須從網站中下載 XML 檔案，並且將資料放入資料庫中。
   Step4：接收來自於 intent 的資料，取出 typeid 值
   Step3: 使用 type id 值與 sql command 去篩選出資料，產生食物的列表。

2. 重要：需要在 AndroidManifest.xml 中去增加一個 Activity 設定：

```
<activity android:name=".NutritionList"
            android:label="@string/nutrition_list_title">
        </activity>
```

3. 先製作 NutritionItem.java (一個自訂的物件類別)，主要是用來顯示有單一食物的自訂畫面物件。
   以下是範例:

```
private String foodCode;
private String foodName;
private String weightValue;
```

```
private String calValue;
private int backgroundColor;
private Intent activityIntent;
private RelativeLayout thisLayout;
private Context parentContext;

/**
 * 建構這個物件類別，用來產生食物類別的物件
 *
 * @param context 母容器
 * @param foodCode 這個類別代碼
 * @param foodName 類別名稱
 * @param weightValue 重量單位
 * @param calValue 卡路里
 * @param backgroundColor 背景顏色
 * @param activityIntent 導入下個畫面所需要使用的Intent
 */

public NutritionItem(Context context, String foodCode, String foodName, String weightValue,
String calValue,
                     int backgroundColor, Intent activityIntent) {

        // 將變數值存入Grobal中，以便日後進行使用

        this.foodCode = foodCode;
        this.foodName = foodName;
        this.weightValue = weightValue;
        this.calValue = calValue;
        this.backgroundColor = backgroundColor;
        this.activityIntent = activityIntent;
        this.parentContext = context;

        thisLayout = generateLayout();

    }

    /**
     * 產生這個類別物件 (僅限內部使用)
     *
     * @return 回傳RelativeLayout物件
```

```
     */

    private RelativeLayout generateLayout() {

        RelativeLayout tempLayout;

        tempLayout = (RelativeLayout) ((Activity)
parentContext).getLayoutInflater().inflate(R.layout.nutrition_food, null);

        TextView foodNameText = (TextView) tempLayout.findViewById(R.id.foodName);
        TextView calText = (TextView) tempLayout.findViewById(R.id.calText);
        TextView weightText = (TextView) tempLayout.findViewById(R.id.unitText);
        ImageView clickButton = (ImageView) tempLayout.findViewById(R.id.clickButton);

        tempLayout.setBackgroundResource(backgroundColor);
        foodNameText.setText(foodName);
        calText.setText(calValue);
        weightText.setText(weightValue);

        // 設定按鈕的按下動作

        clickButton.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View thisView) {

                // 設定按下後的動畫

                Animation buttonAnimation = null;
                buttonAnimation = AnimationUtils.loadAnimation(parentContext,
R.anim.button_animation);
                buttonAnimation.setAnimationListener(new AnimationListener() {

                    @Override
                    public void onAnimationEnd(Animation animation) {
                        // TODO Auto-generated method stub

                        // Intent intent = new Intent(parentContext, (Class<?>)
                        // targetActivity);
                        parentContext.startActivity(activityIntent);
```

```
                }

                @Override
                public void onAnimationRepeat(Animation animation) {
                    // TODO Auto-generated method stub

                }

                @Override
                public void onAnimationStart(Animation animation) {
                    // TODO Auto-generated method stub

                }
            });

            thisView.startAnimation(buttonAnimation);

        }
    });

    return tempLayout;

}

/**
 * 讓使用者可以取來產生出來的layout;
 *
 * @return  產生完成的layout;
 */

public RelativeLayout getContentLayout() {

    return this.thisLayout;

}
```

4 設計 NutritionList 的 OnCreate()方法：

```
@Override
```

```
        protected void onCreate(Bundle savedInstanceState) {
            // TODO Auto-generated method stub
            super.onCreate(savedInstanceState);

            // 繼續以下的處理程序

            SqliteTestApp thisApp = (SqliteTestApp)
getApplicationContext();


            MyDatabaseAdapter DBAdapter = null;

            DBAdapter = thisApp.getDBAdapter();

            boolean ifExist = false;
            String sql = "";

            if(DBAdapter != null) {

             Log.d("DBAdapter","DBAdapter存在並且檢查");

             DBAdapter.open();

             sql = "select * from nutritionfood";
                 Cursor resultCur = DBAdapter.query(sql, null);

                 if (resultCur != null) {

                     if(resultCur.moveToFirst()) {

                         ifExist = true;
                         Log.d("尋找是否已經下載資料","曾經已經下載過資料
了");
                         Toast.makeText(this, "之前已經下載過資料庫的資
料了", Toast.LENGTH_LONG).show();

                     }

                 } else {
```

```
                    Log.d("resultCur","無法產生cursor");

                }

                resultCur.close();
                DBAdapter.close();

            }

            if(ifExist == false) {  // 表示資料還沒有下載

             // 要先下載資料
             loadingProgress = new ProgressDialog(this);

    loadingProgress.setProgressStyle(ProgressDialog.STYLE_SPINNER);
                loadingProgress.setMessage("連結主機，下載食物資料
中.....");
                loadingProgress.setIndeterminate(false);
                loadingProgress.setCancelable(false);

                loadingProgress.show();

                String dataUrl =
"http://www.v7idea.com.tw/xml/NutritionXMLSmall.xml";

                Log.d("MainActivity","重新下載資料!");

                downloadBackgroundTask thisTask = new
downloadBackgroundTask();
                thisTask.execute(dataUrl);

            } else {
              generateList();
            }
        }
```

5. 我們需要設計一個背景處理去下載食物的相關資料，以下是相關的範例程式碼：

```
private class downloadBackgroundTask extends AsyncTask<String,
Integer, String> {
```

```
        // public ProgressDialog loadingProgress1 = null;
        private SqliteTestApp thisApp;

        @Override
        protected String doInBackground( String... urls) {
            // TODO Auto-generated method stub


        int count = urls.length;
          String getResult = "";

          if (count > 0) {

            Log.d("FORM_POST:","參數：" + urls[0]);
            //getResult = MeetingRoom.this.httpFormPost(urls[0],
urls[1], null, null, "UTF-8");
            thisApp = (SqliteTestApp) getApplicationContext();
            getResult = thisApp.getStringFromURL(urls[0]);

          }

          Log.d("LOGIN_GET_RESULT", "取得資料" + getResult);
          // alertbox("回應值",getResult);

          return getResult;

        }

        protected void onProgressUpdate(Integer... progress) {
          // setProgressPercent(progress[0]);


       }

       protected void onPostExecute(String result) { // 回傳之後後續動
作;

         SqliteTestApp thisApp = (SqliteTestApp)
getApplicationContext();
```

```
        if(loadingProgress.isShowing() == true) {

            loadingProgress.hide();
            loadingProgress.dismiss();

        }

        if(result.contentEquals("WEBERROR")) {

        } else {

            boolean ifSuccess = false;
            // 準備要放有關於產生資料的相關程序。

            DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
            DocumentBuilder db;

            try {

                db = dbf.newDocumentBuilder();
                // Document doc = db.parse(mainMapXMLString);
                InputStream thisStream =
thisApp.String2InputStream(result);
                Document doc = db.parse(thisStream);

                Log.d("產生主街區物件","已經整理出一個XML Document");

                NodeList thisNode =
doc.getElementsByTagName("Food_Nutrition");

                if(thisNode.getLength() > 0) {

                    for (int i = 0; i < thisNode.getLength(); i ++ )
{

                        Node singleNode = thisNode.item(i);

                        if (singleNode.getNodeType() ==
Node.ELEMENT_NODE) {
```

```
                            Element thisElement = (Element)
singleNode;
                            String nutritionid =
thisApp.getAttributeValueByXMLTagName(thisElement, "NutritionID");
                            String typeid =
thisApp.getAttributeValueByXMLTagName(thisElement,
"NutritionTypeID");
                            String name =
thisApp.getAttributeValueByXMLTagName(thisElement, "FoodName");
                            String code =
thisApp.getAttributeValueByXMLTagName(thisElement, "FoodCode");
                            String unittype =
thisApp.getAttributeValueByXMLTagName(thisElement, "DefaultWeight")
+ thisApp.getAttributeValueByXMLTagName(thisElement,
"WeightUnitType");
                            String calorie =
thisApp.getAttributeValueByXMLTagName(thisElement, "Calorie");

                            if(nutritionid != null &&
nutritionid.length() > 0) {

                                // 表示有取得值
                                MyDatabaseAdapter DBAdapter = null;
                               DBAdapter = thisApp.getDBAdapter();
                               String sql = "";
                               boolean ifExist = false;

                               if(DBAdapter != null) {

                                DBAdapter.open();
                                SQLiteDatabase thisDatabase =
DBAdapter.getWritableDatabase();

                                sql = "select * from nutritionfood
where nutritionid = ?";
                                Cursor resultCur =
DBAdapter.query(sql, new String[]{nutritionid});

                                if (resultCur != null) {
```

```
                                    if(resultCur.moveToFirst()) {

                                        ifExist = true;
                                        Log.d("尋找是否已經下載資料","
曾經已經下載過資料了");

                                    }

                                }

                                resultCur.close();

                                if(ifExist == false) {

                                    sql = "INSERT INTO
nutritionfood(nutritionid, code, name, "
                                            + "typeid, unittype,
calorie) values(?,?,?,?,?,?)";

                                    thisDatabase.execSQL(sql, new
String[]{nutritionid, code, name, typeid, unittype, calorie});
                                }
                            thisDatabase.close();
                                DBAdapter.close();

                            }


                        }


                    }

                }

            }catch (ParserConfigurationException e1) {
```

```
                // TODO Auto-generated catch block
                e1.printStackTrace();
            } catch (SAXException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
         }
         Log.d("下載主街區資料","下載資料成功!!");
         generateList();
        }
}
```

6. 產生資料的方法：generateList()

```
private void generateList() {

        // 篩選出傳入的typeid值;

        Bundle thisBundle = this.getIntent().getExtras();
        typeId = thisBundle.getString("typeid");

    // 以下程式碼是由進階版的查核表進行製作

    String sql = "";
    SqliteTestApp thisApp = (SqliteTestApp) getApplicationContext();

       MyDatabaseAdapter DBAdapter = null;
       DBAdapter = thisApp.getDBAdapter();

       RelativeLayout mainLayout = (RelativeLayout)
this.getLayoutInflater().inflate(R.layout.scroll_list, null);
       LinearLayout contactListArea = (LinearLayout)
mainLayout.findViewById(R.id.contactListArea);
       contactListArea.setOrientation(LinearLayout.VERTICAL);

       if(DBAdapter != null) {

        Log.d("DBAdapter","DBAdapter存在並且檢查");
```

```
        DBAdapter.open();

        sql = "select * from nutritionfood where typeid = ? order by
code";
        Cursor resultCur = DBAdapter.query(sql, new
String[]{typeId});

        if (resultCur != null) {

            if(resultCur.moveToFirst()) {

                // 表示有找到資料

                int i = 0;
                int backgroundColor = R.color.gray_type1;

                for (i = 0; i < resultCur.getCount(); i ++ ) {

                    String nutritionTypeId =
resultCur.getString(resultCur.getColumnIndex("typeid"));
                    String foodCode =
resultCur.getString(resultCur.getColumnIndex("code"));
                    String foodName =
resultCur.getString(resultCur.getColumnIndex("name"));
                    String foodId =
resultCur.getString(resultCur.getColumnIndex("nutritionid"));
                    String calValue =
resultCur.getString(resultCur.getColumnIndex("calorie"));
                    String weightValue =
resultCur.getString(resultCur.getColumnIndex("unittype"));

                    if(backgroundColor == R.color.gray_type2)
backgroundColor = R.color.gray_type1;
                    else backgroundColor = R.color.gray_type2;

                    Intent foodIntent = new Intent(this,
NutritionDetail.class);
                    foodIntent.putExtra("nutritionid", foodId);
```

```
                    NutritionItem thisTypeObj = new
NutritionItem(this, foodCode, foodName, weightValue, calValue,
backgroundColor, foodIntent);
                    if(thisTypeObj != null) {

                        RelativeLayout singleLayout =
thisTypeObj.getContentLayout();
                        if(singleLayout != null) {

                            contactListArea.addView(singleLayout);

                        }

                    }

                    resultCur.moveToNext();

                }



            }

        } else {

            Log.d("resultCur","無法產生cursor");
            Toast.makeText(this, "找不到任何食物類別的資料",
Toast.LENGTH_LONG).show();

        }

        resultCur.close();
        DBAdapter.close();

        setContentView(mainLayout);


    }

}
```