NZ WEEKLY Fruit Vegetable APP

Purpose: I want to create an APP to help the customer to see the content in advance and to order the fruits weekly by the name and TEL information .

This is a real need when I order something feeling unconvent because I cannot see the order content in advance, and the order system is not in time because in the form of Email.

**Plan -Activity-Order:**

Toolbar accountMenu

TextView

toolbar

* Input (e.g)
* name: Jason
* TEL:0291251989)

TextView

User Information

Log in button

1.First week -Apple

2.Second week-Kiwi

Image view

For LOGO Picture

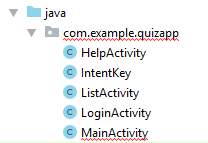
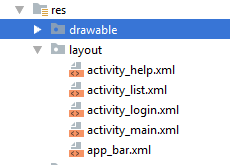
GO

selected

reset to save

I will use at least two Android features that were not yet covered in class just like below list:

1. Add LOGO picture in the MainActicity:
2. Input the user information to log in.
3. Limit the right that user cannot choose after they select.

I will generate a class “IntentKey”to implement a static String value to transfer all the customer information between the activities, which can be understood as a courier number and can be unified management in case of writing wrongly every time .(OOP)

First, I will design the layout of the activities throough the .xml file.

MainActicity: for display the logo and the company; navigating to the log in

LoginActivity: for log in and jumping to the next activity.

1. I will choose the color for “ backgroundColor”, “ButtonbackgroundColor” and “txtColor” from the website <https://coolors.co/> .For a beautiful looking , I will design many colors for different buttons. Take the “SubmitButton ” for example. I have changed it different the other button background color. As you can see in the screen of APP there are two button color between “LOG In” and “SUBMIT’.

2. I can change the APP name to the “NZ WEEKLY LIJING App” in the strings.xml file .

3. I can drag the “textview” and “button” to the layout .xml file to design the picture.

Second, I will use Method:onClickListener; intent ; EditText; ListView +Arrayadapter

such as Intent for jumping to new activity using a button click .

Function 1: Click Button （btnLogin）will navigate to next activity.

I declare “private Button btnLogin;”

“setContentView (R.layout.activity\_main);”

// one method make activity to load the content of .xml and generate related objects.

“btnLogin = find ViewById(R.id.***btnLogin***);”

// After creating the objects, use findViewById to find out the correspond object.

// I think findViewById is the Implementation of Object-Oriented Programming (OOP) concepts, because all the action must be done by the Object.

// I use next code to link the “onClick” method to the “btnLogin” object.

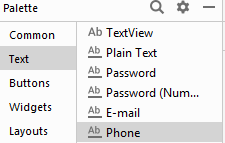
// inside the “onClick “, there is “intent” to transfer from MainActivity to LoginActivity.

**“btnLogin**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent(MainActivity.**this**, LoginActivity.**class**);  
 startActivity(intent);  
 }  
});”

Function2: Input the user information for log in

At first, I will drag two EditText in the activity\_login.xml , and make one inputType is phone.

This place is not password type because this APP just need the phone, without password.



Second ,I layout this two EditText,and add a button”btn/Go”.And I add the hint content for both .

At last， I want to use intent to transfer the data of EditText by clicking the btn/Go button.

Firstly, I will implement the function to input content to two values.

**private** EditText **et\_name**;  
**private** EditText **et\_tel**;

**et\_name** = findViewById(R.id.***et\_name***);  
**et\_tel** = findViewById(R.id.***et\_tel***);

String name = **et\_name**.getText().toString();  
String tel = **et\_tel**.getText().toString();

Add data into the Intent object using the putExtra() methods.

“

Intent intent = **new** Intent(LoginActivity.**this**, QuestionActivity.**class**);  
intent.putExtra(IntentKey.***USER\_NAME***, name);// Add extended data to the intent

intent.putExtra(IntentKey.***USER\_PHONE***,tel);

”

Then in the new Activity – ListActivity.I will get the value by the following code.

“

**private void** getData() {  
 Intent intent = getIntent(); // return the intent that started this activityString name = intent.getStringExtra(IntentKey.***USER\_NAME***);  
 String tel= intent.getStringExtra(IntentKey.***USER\_PHONE***);  
}

“

Above all , I have achieved the function that input user information and jump to the next activity and display the transferred information in the new ListActivity.

**textDisplay** = findViewById(R.id.***textDisplay***);  
**textDisplay**.setText(**"name:"**+**name**+**"TEL:"**+**tel**);

Function3: I will use a listView to display the information of product everyweek.

//At first, I will get a ListView object –list to prepare ,but this list should at first write like this

**private** ListView **list**;

// then under the “setContentView(R.layout.***activity\_list***);”

Get the list from the reference.

**list** = findViewById(R.id.***orderList***);

inttead of using this below code directily.

“ListView list = (ListView) findViewById(R.id.***orderList***);” at the beginning

ArrayList<String> productlist = **new** ArrayList<>();

In java call the method ,create the dynamic arraylist, and add the data

Create new method to add content to the new ArrayList object-“productList”.

**private void** createList() {  
 **productList**.add(**"August 1th week:Apple"**);  
 **productList**.add(**"August 2th week:Banana"**);  
 **productList**.add(**"August 3th week:Potato"**);  
 **productList**.add(**"August 4th week：Tomato"**); }

The I add the arrayList to an adapter and then set the adapter to the listview

//ceate ArrayAdapter a new object "adapter" and set it with type and content.



//I will use this code to combine listview and adapter to add the data

List.setAdapter(adapter);

//I also use below two line code to permit to display the multiple choice with box.





The result is that you can see a list ,but I want to add a button”btnSelected” toclick to display select the chosen week.

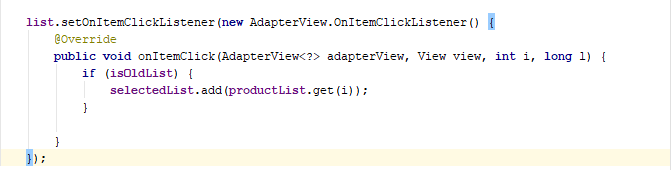
So I create a new dynamic ArrayList object –“selectedList” to achieve the select function

**public** ArrayList<String> **selectedList** = **new** ArrayList<>();

// then again add data to the “selectedList”

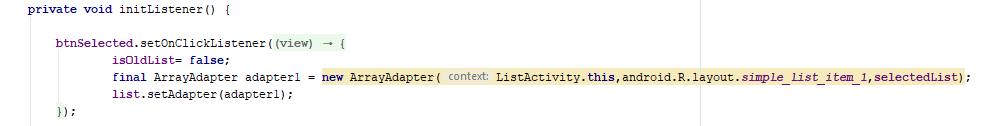


“btnSelected” will work after clicking the item in the list, so I add the method of “list.setOnItemClickListener( new)” to after adding data to the new ArrayList “selectedList”



Now I wanna use the method setOnClickListener to run “btnSelected” button to select the new list

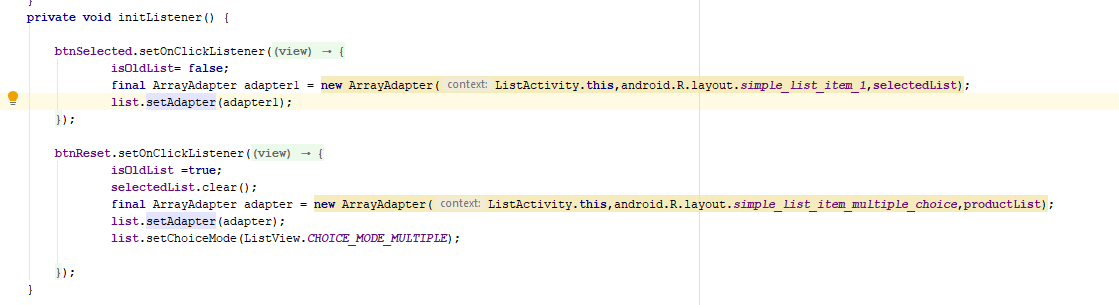
and add the arraylist to a new “adapter1” at the same time set the “adapter1” to the listview.



Now, I will achieve the feature which not yet covered in class , I want to limit the right that user cannot choose after they select.

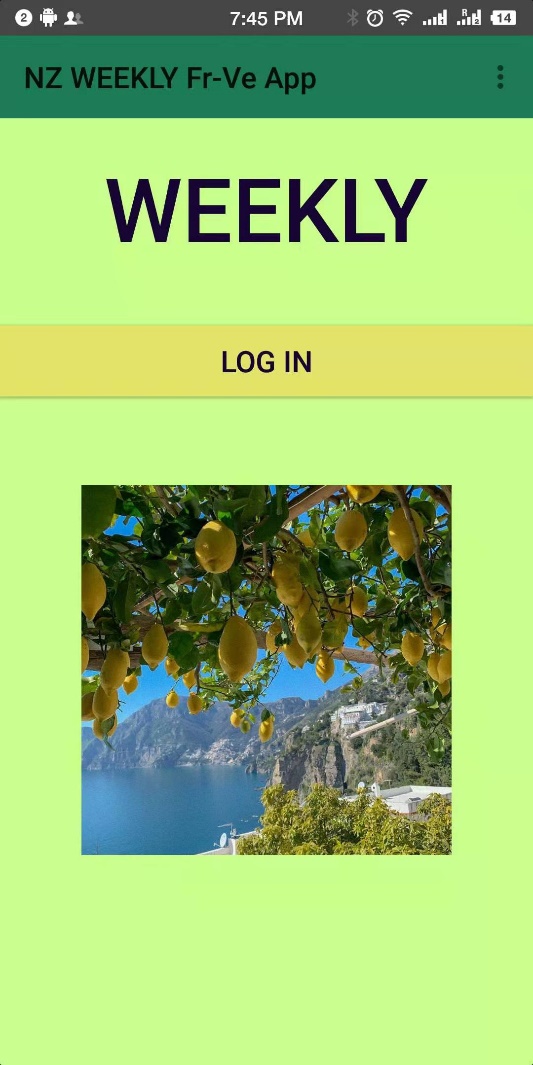
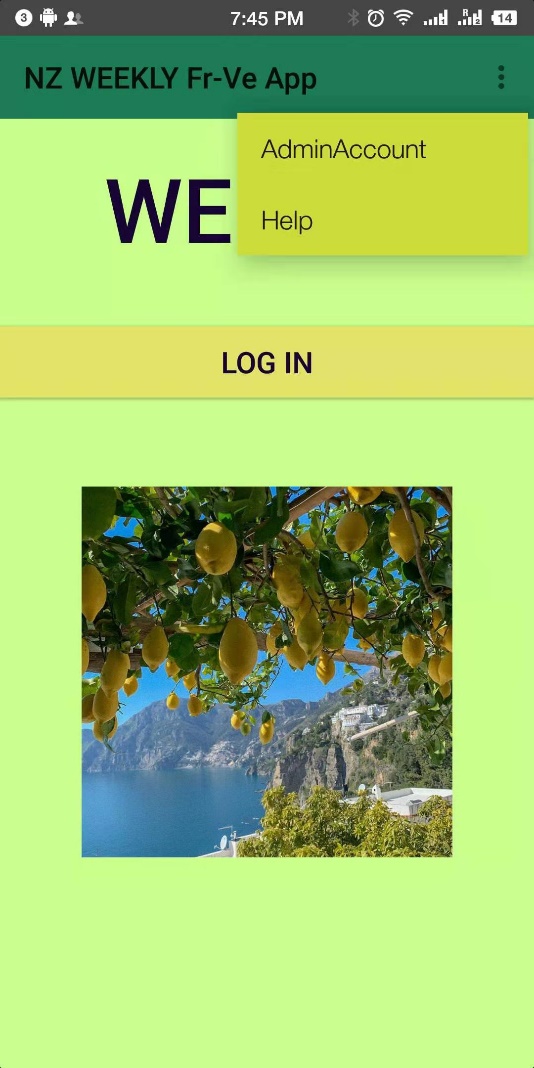
So I add one Boolean considitonat the beginning: **“private boolean isOldList** = **true**;”

And I use an if statement to decide whether or not is the “isOldList” to touch off the choose function.Just like this below code.



Finally，I want to customise a new toolbar to display the menu .

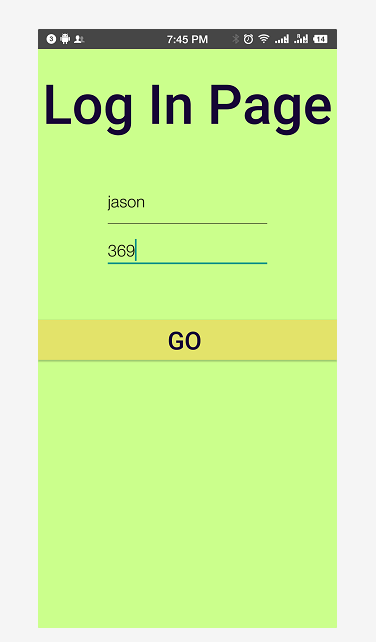
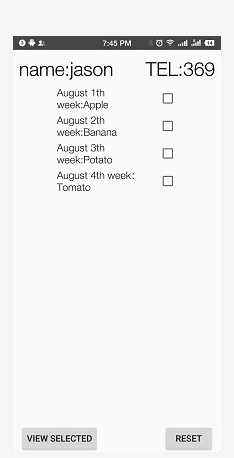
I also use intend to jump to a new HelpActivity to help customer to explain this APP.

**Assignment2** Plan : I will achieve one admin account (which is shown in the above picture )can see all the ordered information in a activity.

The admin account also can see order content and can add text to the ListView list.

**Assignment 2:**

I will achieve ：Click ToolBar-AdminAccount, jump to one AdminActivity to display a SQL data . SQL is inserted by a selectedList, boss can see all users ordered information in AdminActivity.

Table of Contents

[1.ListActicity: 7](#_Toc18230516)

[2. DaHandler.java Class 8](#_Toc18230517)

[I. Create a new java class 8](#_Toc18230518)

[II. Add the unimplemented methods (onCreate() and onUpgrade()) 8](#_Toc18230519)

[III. Create Database 8](#_Toc18230520)

[In the onCreate() method, create a SQL query 8](#_Toc18230521)

[V. In the onUpgrade() method, write a SQL command to drop the table 9](#_Toc18230522)

[VI. In order to create new users, add a new method called insertUserDetails() 9](#_Toc18230523)

[VII. Create a new method called getUsers() to view user details. 9](#_Toc18230524)

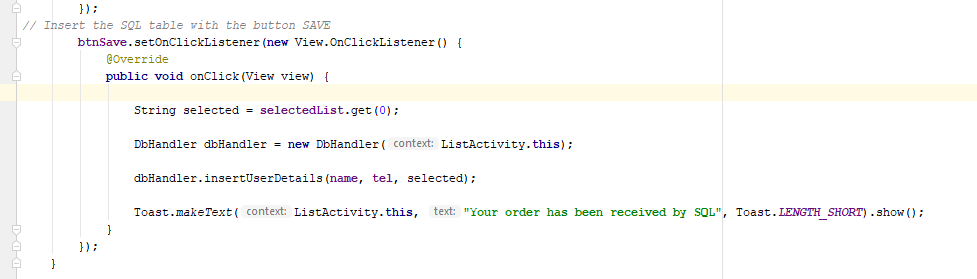
[3.AdminActivity: 11](#_Toc18230525)

[4. Open the AndroidManifest.xml file 13](#_Toc18230526)

# 1.ListActicity:

1.I will add a Button “btn\_save" to create the SQL with the selectedList. In the \_list.xml

2.Then I will add a OnClocklistener method to the Button “btn\_save" in ListActivity:



3. Inside the OnClocklistener method, I will insert a message to create the SQL Table.

4. message include 3 variables : name, tel, selected ;

Where does selected value come from ?

I will get it from selectedList.get(0) like below:

String selected = **selectedList**.get(0);

5.The selectedList will be set to hold only one value ?

6.I need create a method “insertUserDetails()” in a DaHandler.java Class(new built)

To create the database.

7. I also add a “Toast” information to help the users to ensure their order .

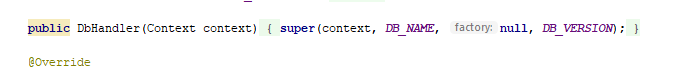
# 2. DaHandler.java Class

1. Create a new java class called DaHandler that is a subclass from SQLiteOpenHelper class.
2. Add the unimplemented methods (onCreate() and onUpgrade()) and a constructor. Configure the constructor to only accept Context argument.
3. Create Database

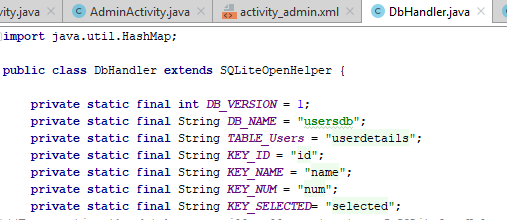
The example that I have used here contains **DbHelper** class that extends **SQLiteOpenHelper**class and perform all database related operations.

For creating the database we will call constructor of **SQLiteOpenHelper** class using **super()**.

Pass the DB\_NAME and DB\_VERSION in the superclass within your constructor.

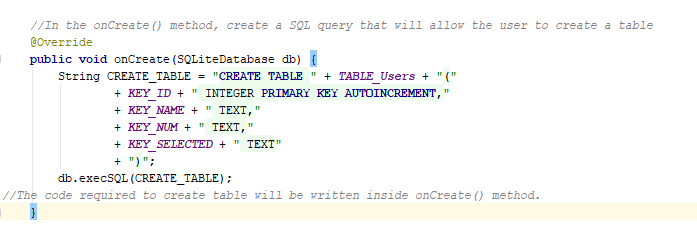


Where are the ***DB\_NAME*** and ***DB\_VERSION***) from? I declare them at first.

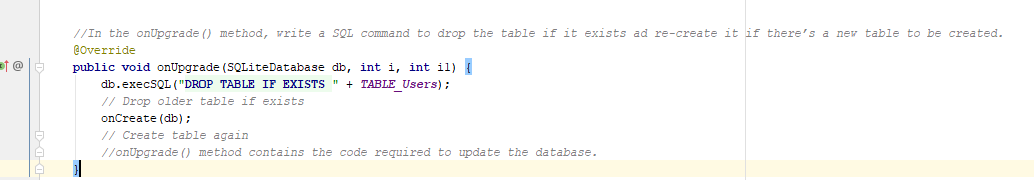


1. To make it simpler, modify the argument name in the onCreate() method to db.

In the onCreate() method, create a SQL query that will allow the user to create a table.

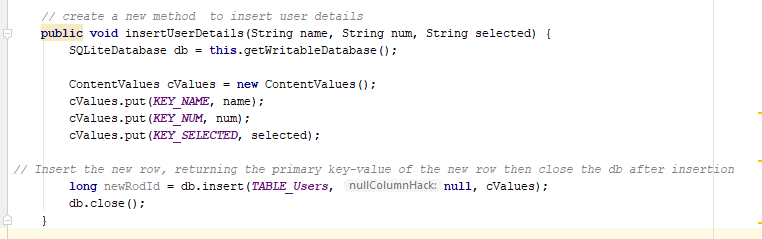


1. In the onUpgrade() method, write a SQL command to drop the table if it exists ad re-create it if there’s a new table to be created.



1. In order to create new users, add a new method called insertUserDetails() within this class. This method will require three (3) parameters namely name, location and designation all of String types, I have created them at the 3rd step.

Still within the insertUserDetails() method, add the required code and call the necessary method to get the data repository in write mode.



In the above example the insert operation is handled by insertUserDetails()

Method. It takes name , num, selected as 3 arguments and insert them into table.

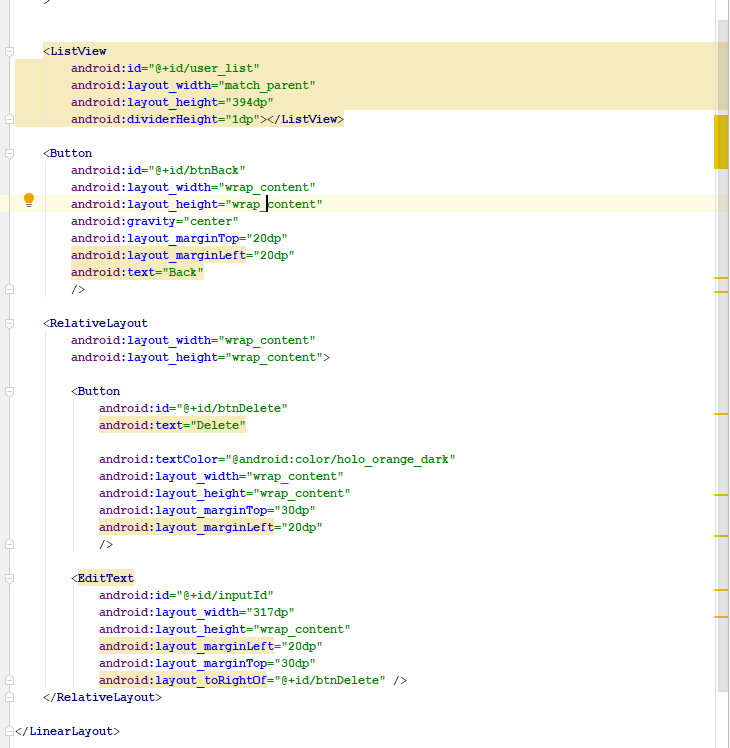
 We have to first add all the values in **ContentValues** object ---“cValues”and then finally insert into table using **insert()** method of **SQLiteDatabase** class.

1. Create a new method called getUsers() to view user details. This method should return an array list of hash map that contains strings for key and values.



# 3.AdminActivity:

1. Create a new empty activity called AdminActivity. While configuring this, set the name of the layout to activity\_admin.xml



2. For the \_admin layout, I am adding the following widgets:

Because the ListView will display rows in it as shown, I need to make another layout that will hold the row values. I create and call this layout list\_row.xml.



1. Create an instance of the Intent class, listView instance and a button within the class.
2. In the onCreate() method, create a DbHandler instance and initialise it with the ‘this’ keyword.



1. Similar to what you did in the DbHandler class, create an ArrayList that contains a HashMap.

This will receive the returned value when the db instance calls the getUser() method.



1. Initialise the ListView and Button instances by referencing their equivalent widget ids in the activity\_admin.xml .

To display the list, add the following code that uses the “SimpleAdapter” to hold the information from the database.



8. Set the onClickListener for the back button. When this button is clicked, it should go back to the MainActivity.java.



4. Open the AndroidManifest.xml file to verify that the new activity has been added.

I also want to use this now to set the display activity name for the AdminActivity.

