SD-6501 Assignment 2

**Jing Li ID:21902204**

Table of Contents

[Introuction : 1](#_Toc19604987)

[Conceptual Framework: 2](#_Toc19604988)

[Advanced Feature 1:Database implementation and management: 2](#_Toc19604989)

[Create: 2](#_Toc19604990)

[1.Layout: 2](#_Toc19604991)

[2. Behaviour: 3](#_Toc19604992)

[Create Database-SQL: 4](#_Toc19604993)

[Read: 5](#_Toc19604994)

[1.Layout: 5](#_Toc19604995)

[2. Behaviour: 7](#_Toc19604996)

[Update: 8](#_Toc19604997)

[1.Layout: 8](#_Toc19604998)

[2. Behaviour: 8](#_Toc19604999)

[Delete: 9](#_Toc19605000)

[1.Layout: 9](#_Toc19605001)

[2. Behaviour: 9](#_Toc19605002)

[Discussion of constraints encountered, and strategies applied during the development 9](#_Toc19605003)

[Advanced Feature 2: GridView and GridItemView **Error! Bookmark not defined.**](#_Toc19605004)

[1. Layout **Error! Bookmark not defined.**](#_Toc19605005)

[2. Behaviour **Error! Bookmark not defined.**](#_Toc19605006)

[Conclusion 11](#_Toc19605007)

# Introuction :

In Assignment 2 , I will add two main functions. One main function is the Database implementation and management, I attach video to show steps.

I upload a demo video to display the whole progress like below list:

Create: After Clicking “Save” button in the ListActicity to create Database.

Read: In the mainActivity ,after clicking Options Menu, and click AdminAccount, It will jump to one AdminActivity to display a SQL data .

Update: In AdminActivity, after input then click “Update”button.

Delete: In AdminActivity, after input”IDNum”,click “Delete”button.

Function 2 is one improvement about the content display.

# Conceptual Framework:

I will achieve the input information into a SQL with a DbHandler Class.

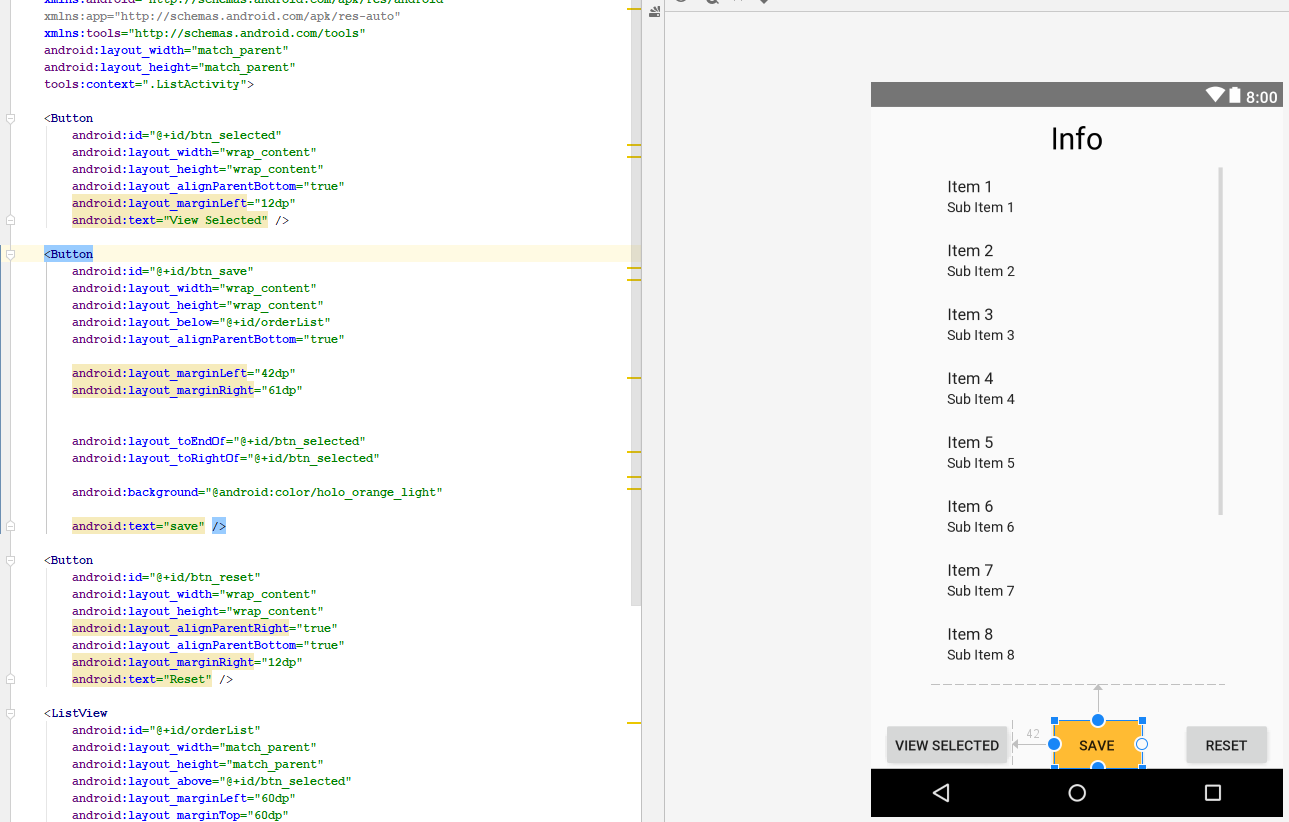
Following the logic , I will create ListActicity,AdminActivity.

# Advanced Feature :Database implementation and management:

## Create:

### 1.Layout:

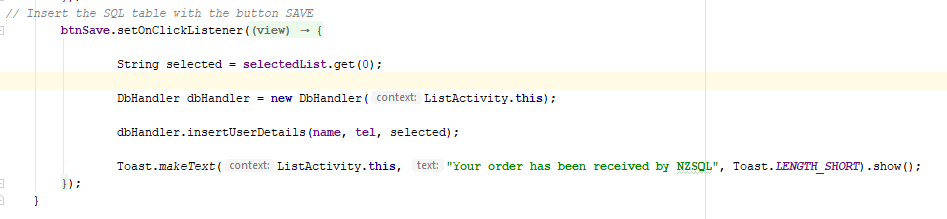
In the ListActicity ,at the base of assignment 1 including ViewSelected and Reset functions, I will add a Button “btn\_save" to create the SQL with the selectedList in the \_list.xml file.



### 2. Behaviour:

Then I will add a setOnClicklistener method to the Button “btn\_save" in ListActivity:I also add a “Toast” information to help the users to ensure their order . Inside the OnClicklistener method, I will insert a message to create the SQL Table. Message include 3 variables : name, tel, selected ;

The whole method like this snap picture below,



There is a Logic whch is need to think: Where does selected value come from ?

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

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

The selectedList will be set to hold only one value.

SQL is inserted by a selectedList.

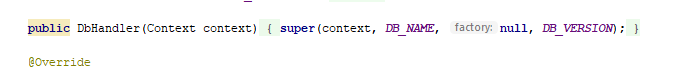
I need create a method “insertUserDetails()” in a DaHandler.java Class(new built)To create the database.

Create Database-SQL:

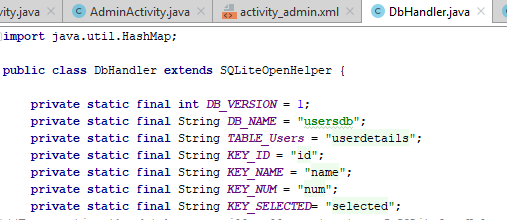
Create a new java class called DbHandler that I have used here contains DbHelper class that extends SQLiteOpenHelperclass and perform all database related operations.

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

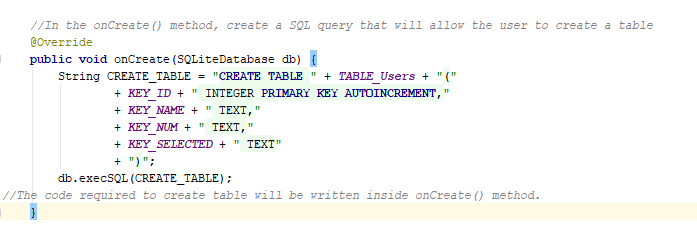
Pass the DB\_NAME and DB\_VERSION in the superclass within my constructor.



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

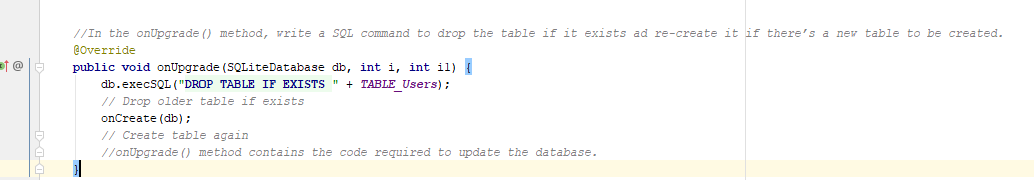


To make it simpler, modify the argument name in the onCreate() method to db. In the onCreate() method, create a SQL query that will allow the user to create a table.



Add the unimplemented methods (onCreate() and onUpgrade()) and a constructor. Configure the constructor to only accept Context argument.

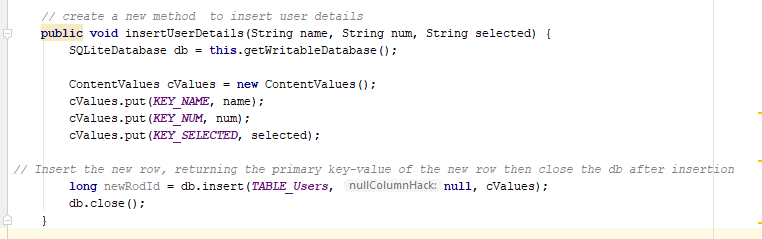
In the onUpgrade() method, write a SQL command to drop the table if it exists ad re-create it if there’s a new table to be created.



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 above 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.

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

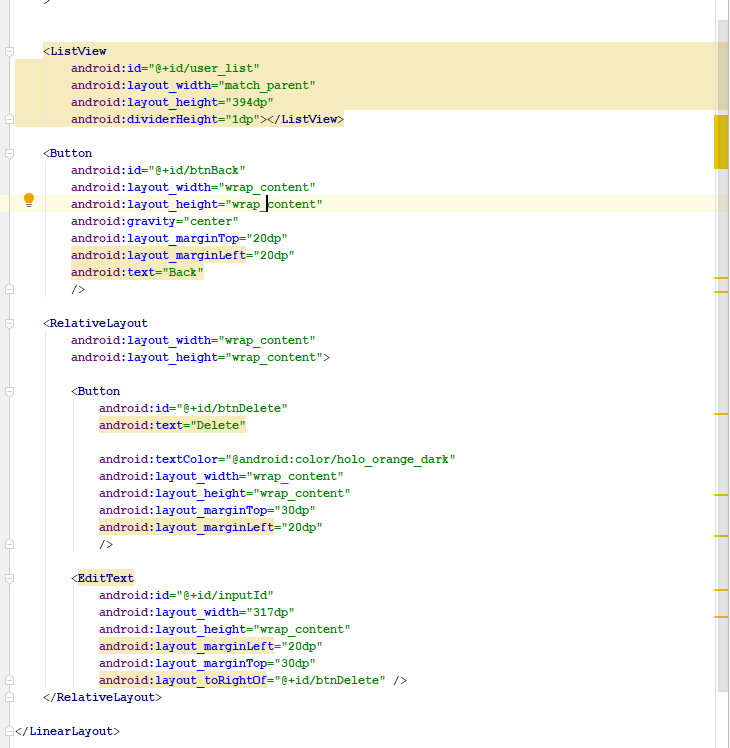
## Read:

The adminsraters can see all users ordered information in AdminActivity.

### 1.Layout:

Create a new empty activity called AdminActivity.

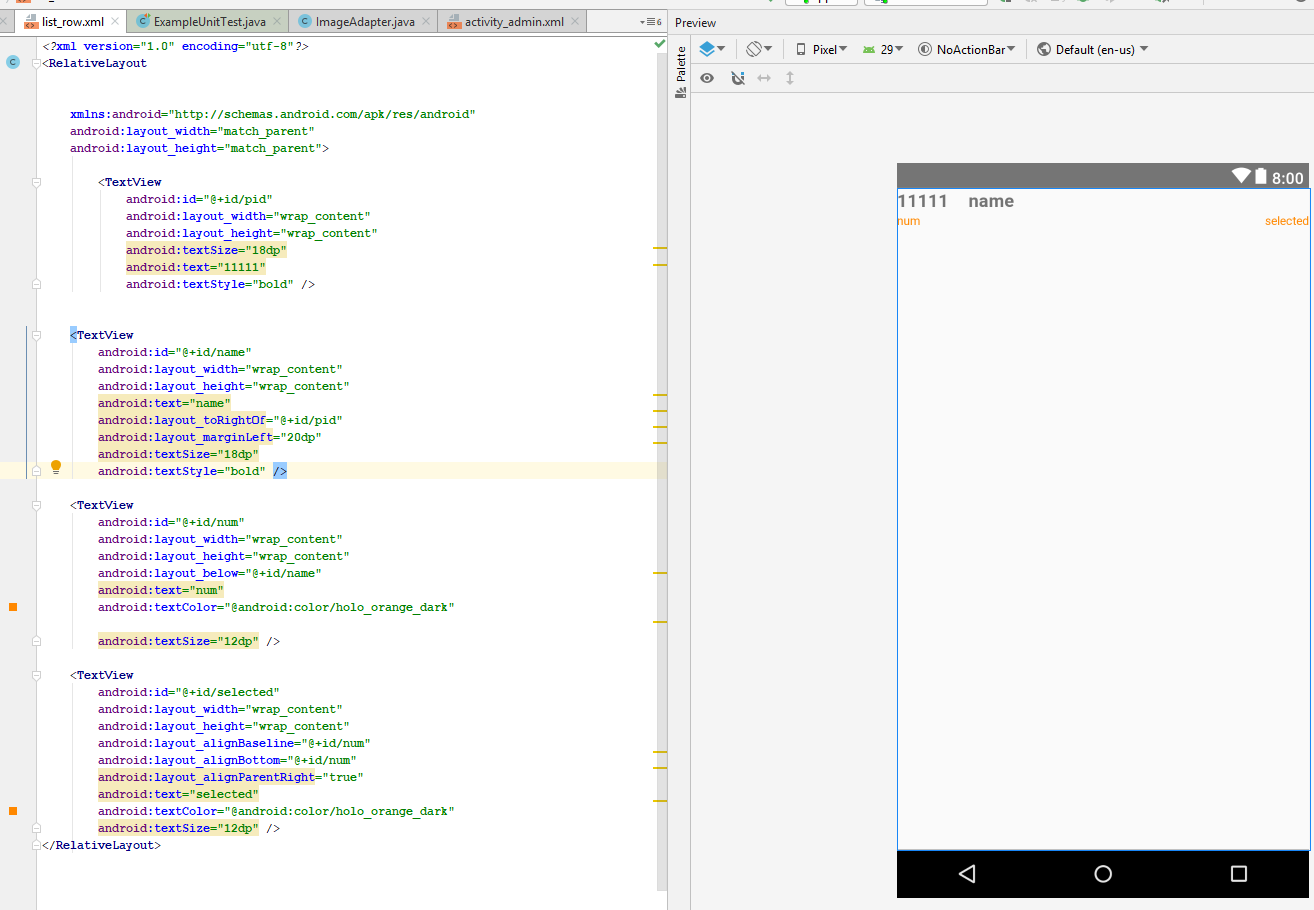
While configuring this, set the name of the layout to activity\_admin.xml



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

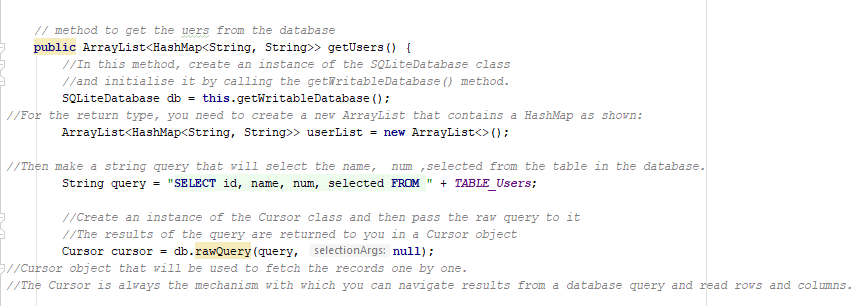
Because the ListView will display rows in it as shown, I need to make another layout that will hold the row values.

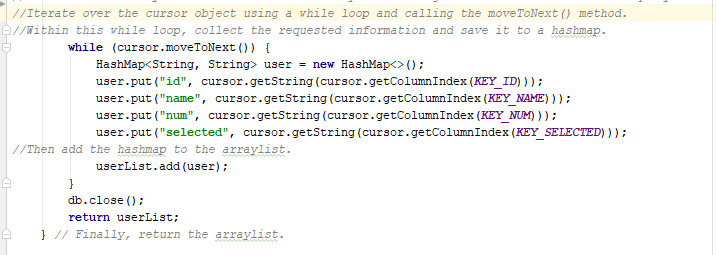
So I create and call this layout list\_row.xml.



### 2. Behaviour:

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.





In the onCreate() method, create a DbHandler instance and initialise it with the ‘this’ keyword.

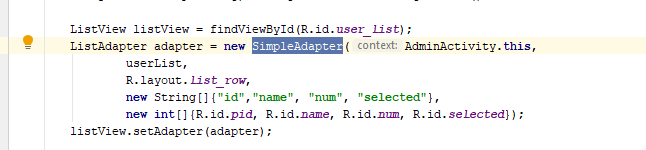


Similar to what I did in the DbHandler class, create an ArrayList that contains a HashMap. This will receive the returned value when the db instance calls the getUser() method.



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

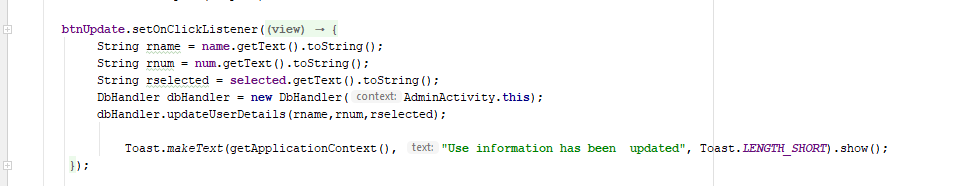
To display the list, add the following code that uses the “SimpleAdapter” to hold the information from the database.



## Update:

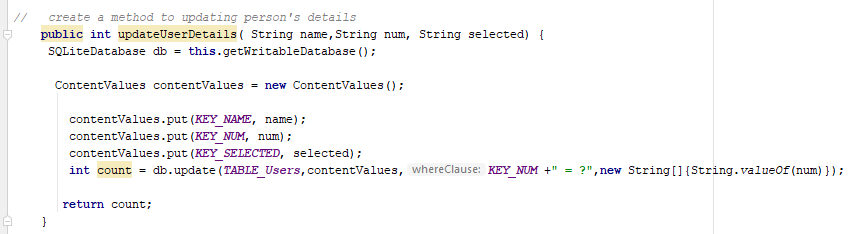
### 1.Layout:

Set the onClickListener for the Update button. When this button is clicked, it should update the Ordered information got from three EditText input.



### 2. Behaviour:

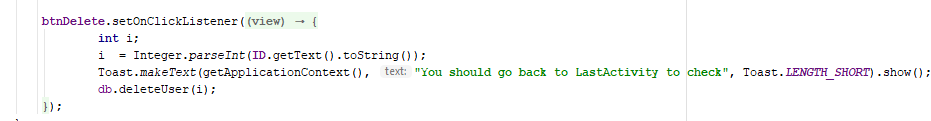
I create a method to update users' order details：



## Delete:

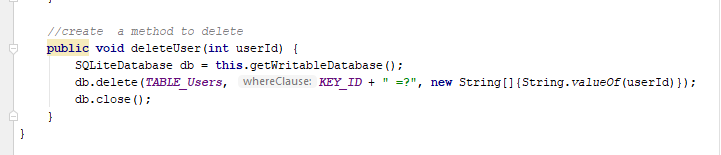
### 1.Layout:

Set the onClickListener method for the Delete Button . When this button is clicked, it should delete the ordered information by the ID .got from EditText input.



### 2. Behaviour:

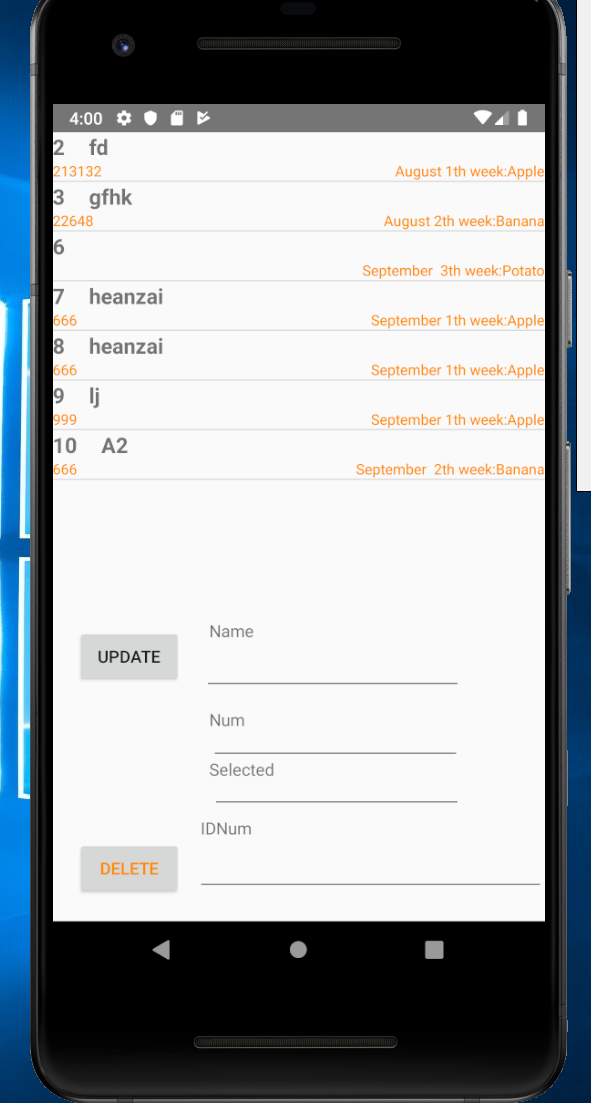
I create a method to delete the selected users' order details based on”ID”



# Discussion of constraints encountered, and strategies applied during the development

During the development , I find it is hard to delete ,because I need the “id” to delete, if “id” doesn’t show, it is hard to clarify the Num of ID, So I make “id” will be displayed also in the list so as to read and check.

In the above code, I put “id” also saved into a hashmap.



# Conclusion

Above all , Database implementation (CRUD functions) have made the App become a better product to use , User can manage the order information and CRUD in one phone .

If this app want to be in market use , It needs the SQL in the cloud service and more testing to run well in different kinds of phones ,including IOS,Android.