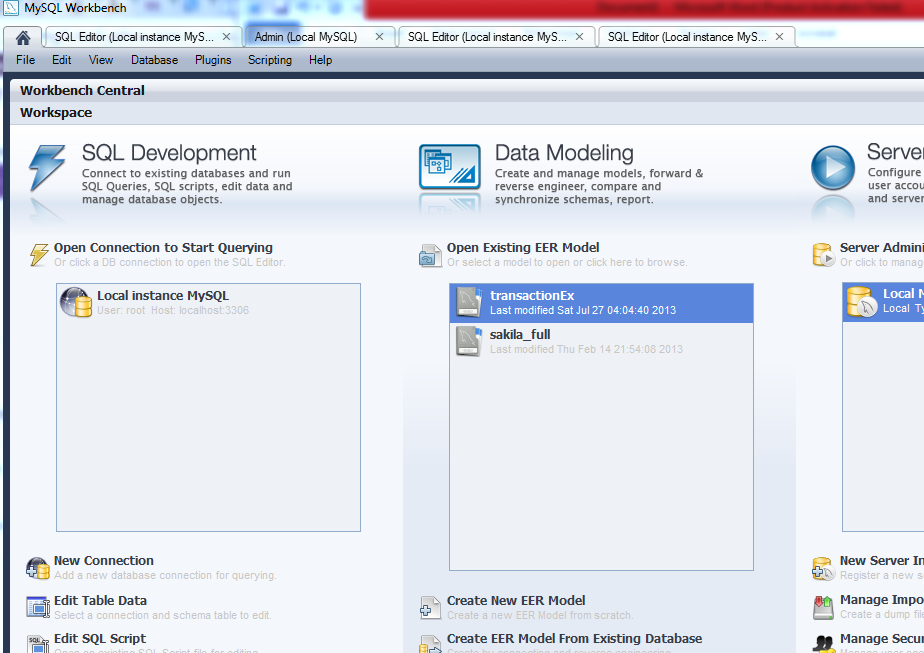
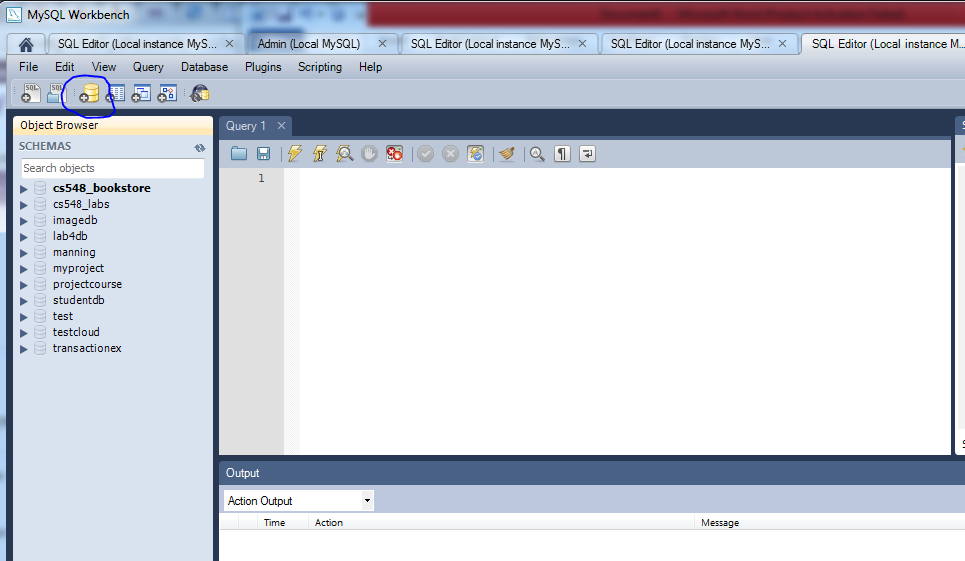
**Phonegap and MySQL database Example**

1. **Setup and programming for server side with database.**

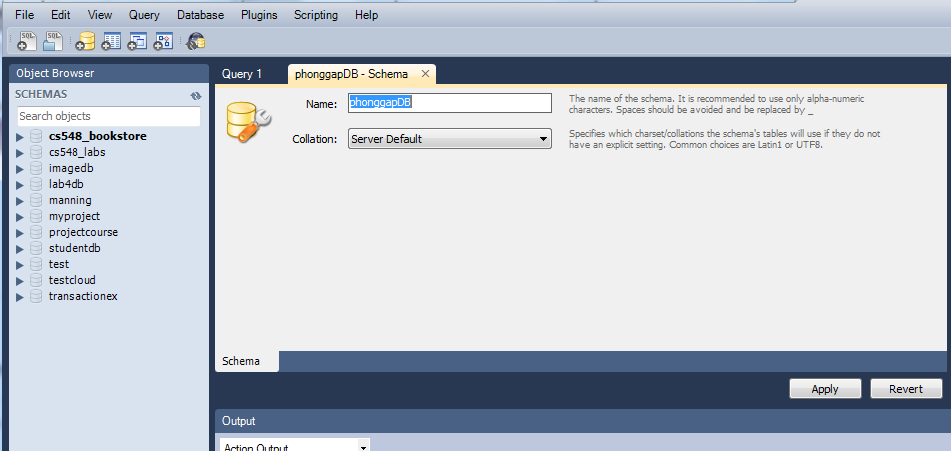
* Installed mySQL server and MySQL workbench.
* Open mySQL workbench, create a new database and a table as following:



Click on “ Local Instance MYSQL”

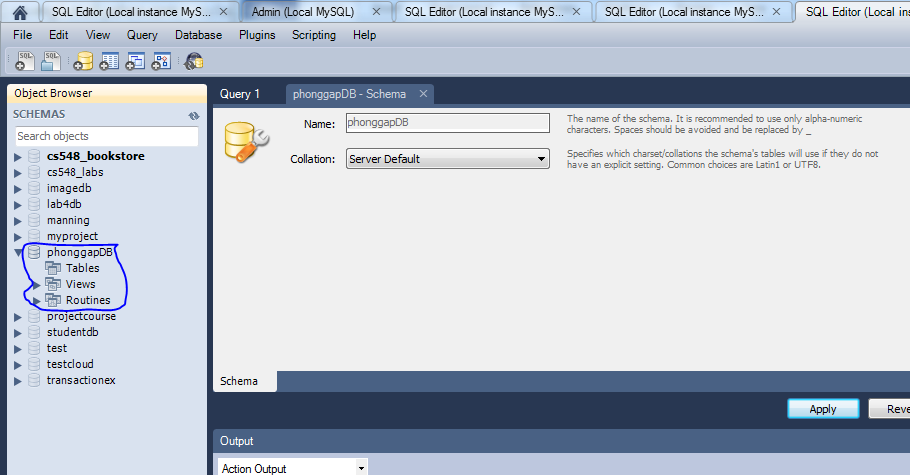


Click on create database icon to create a new database.

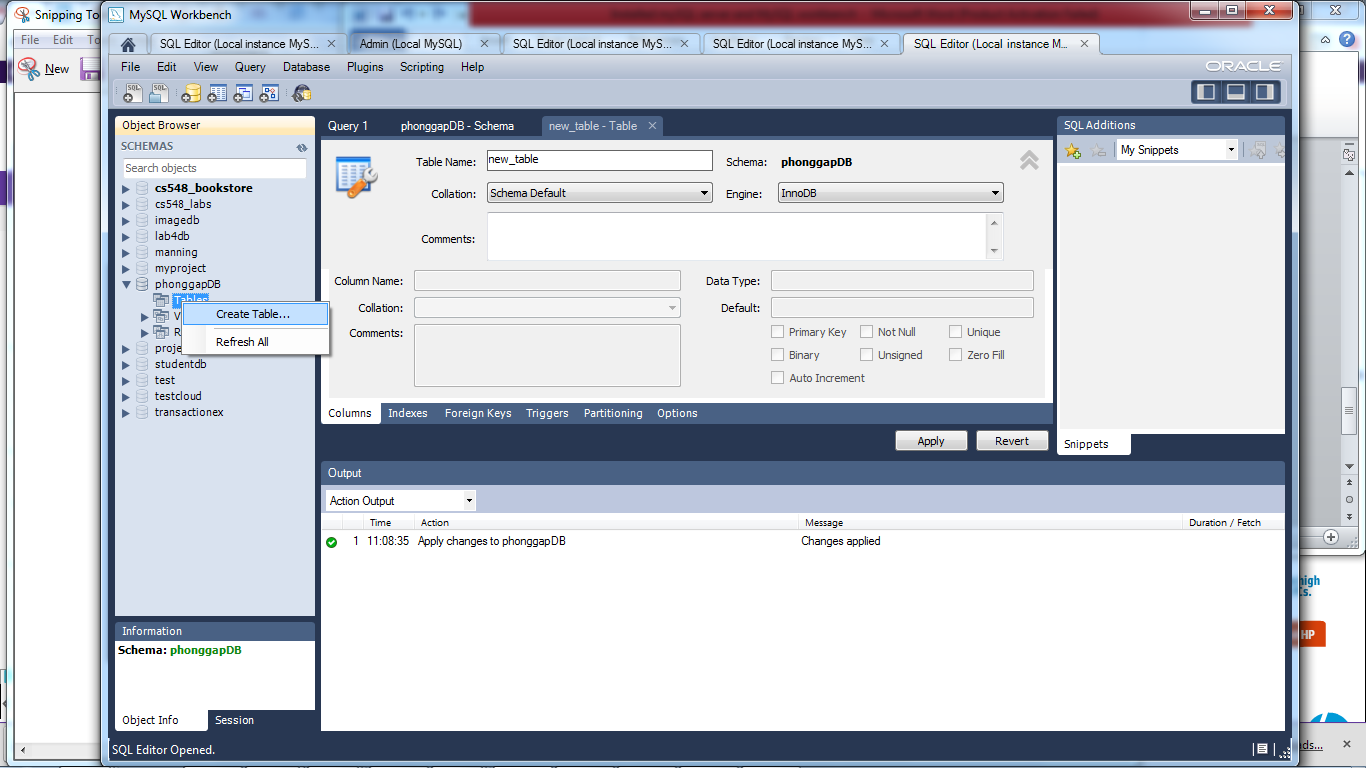


For example: type the name of database as “phonegapDB”

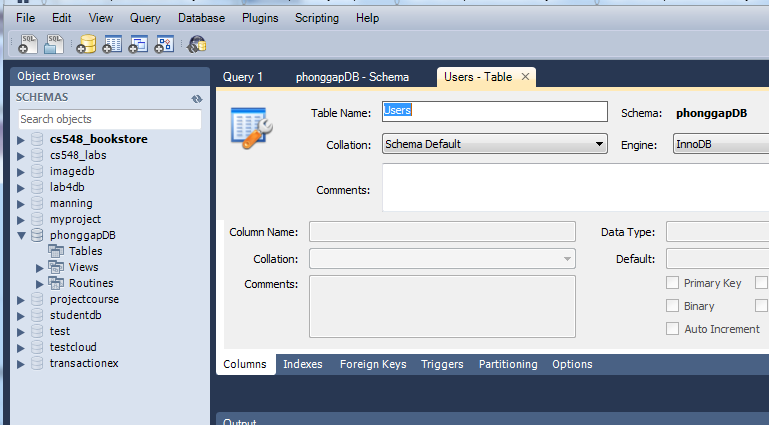
Click Apply.



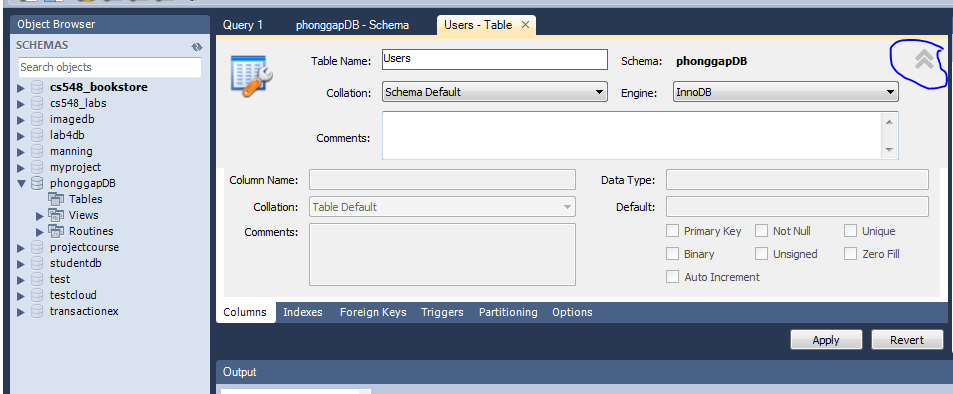
Right click on Tables of database phonggapDB to create a new table.



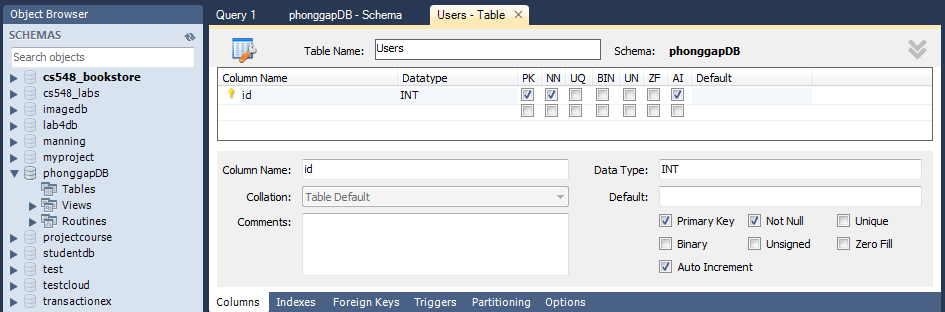
For example, enter table name as Users to create Users table.



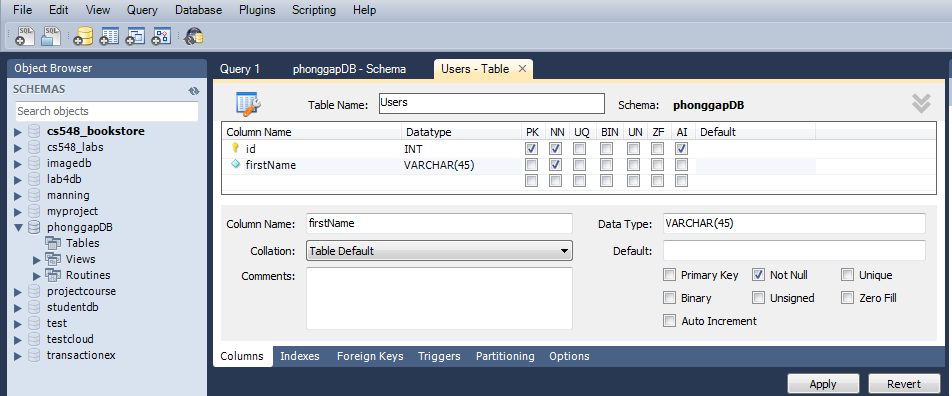
Click the icon “up” to go to the screen to add some columns.



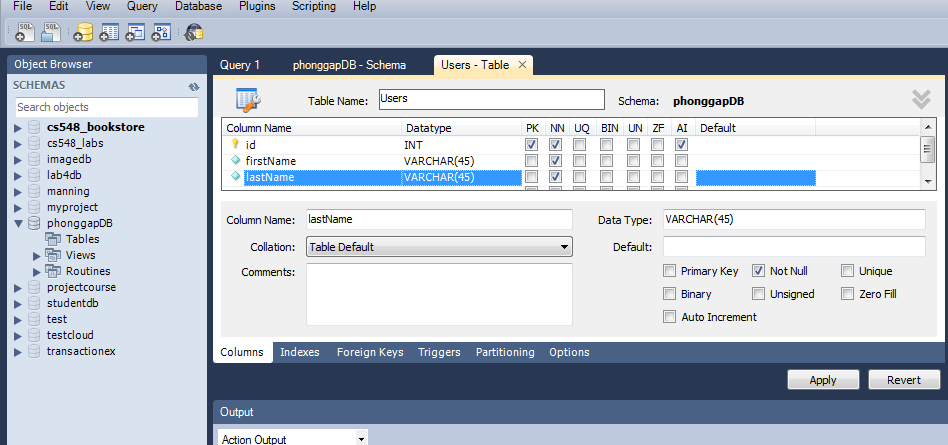
Add “id” column as following



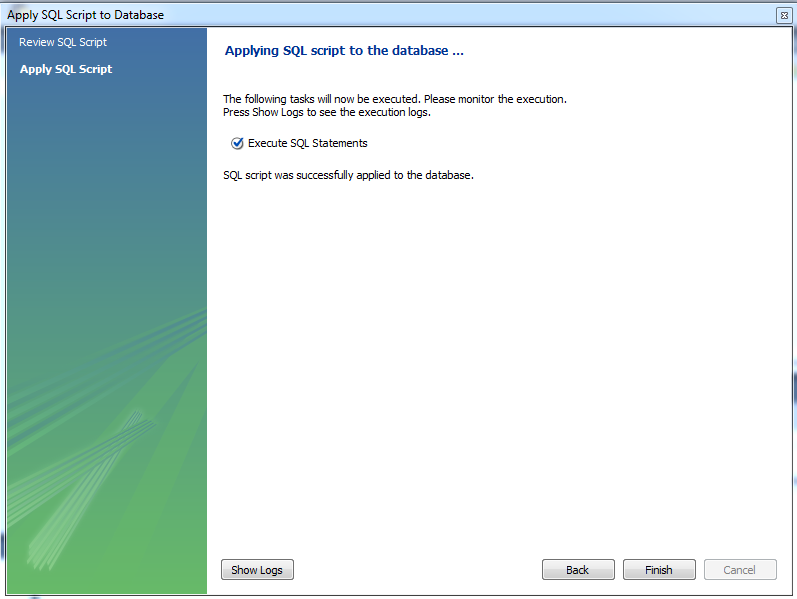
Add “firstName” column as following:



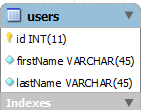
Add “lastName” column as following:



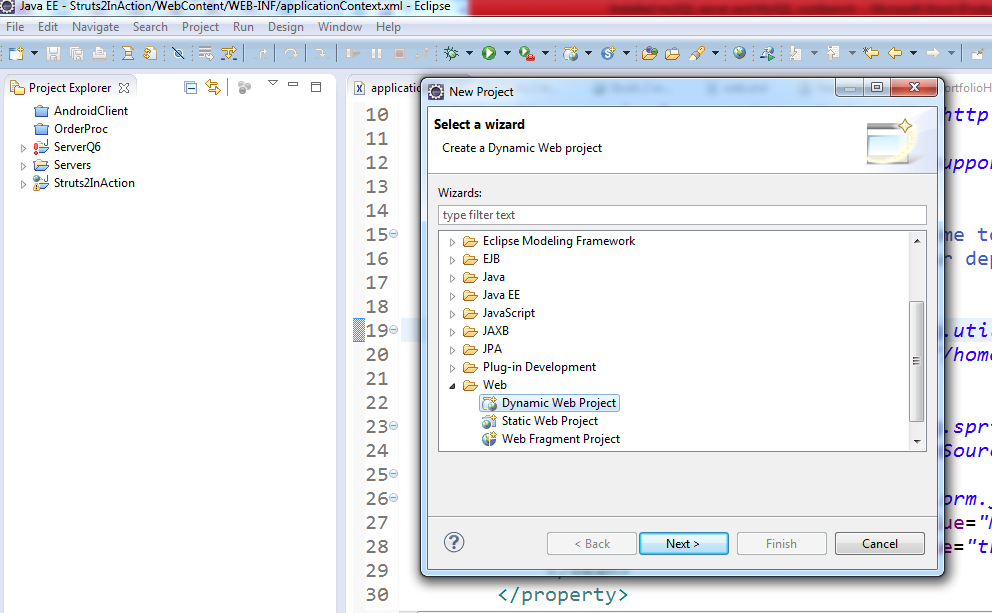
Click Apply

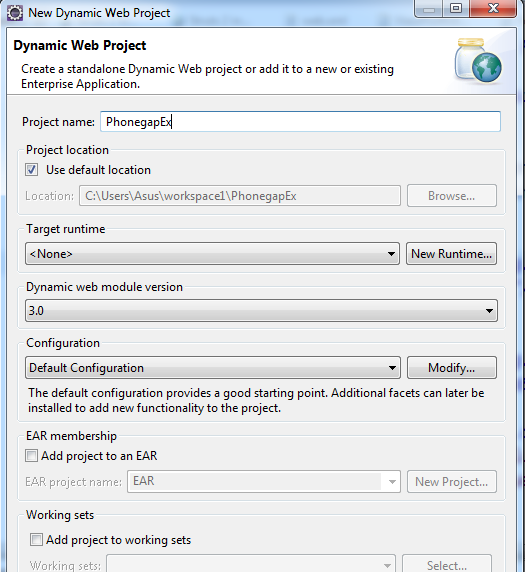


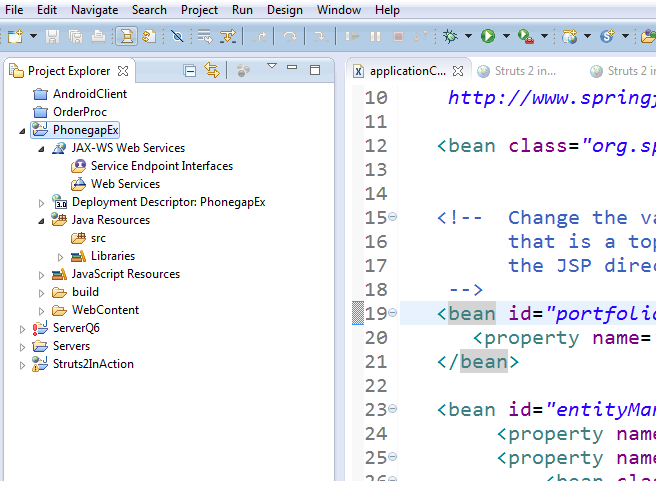
Now we got the following table.



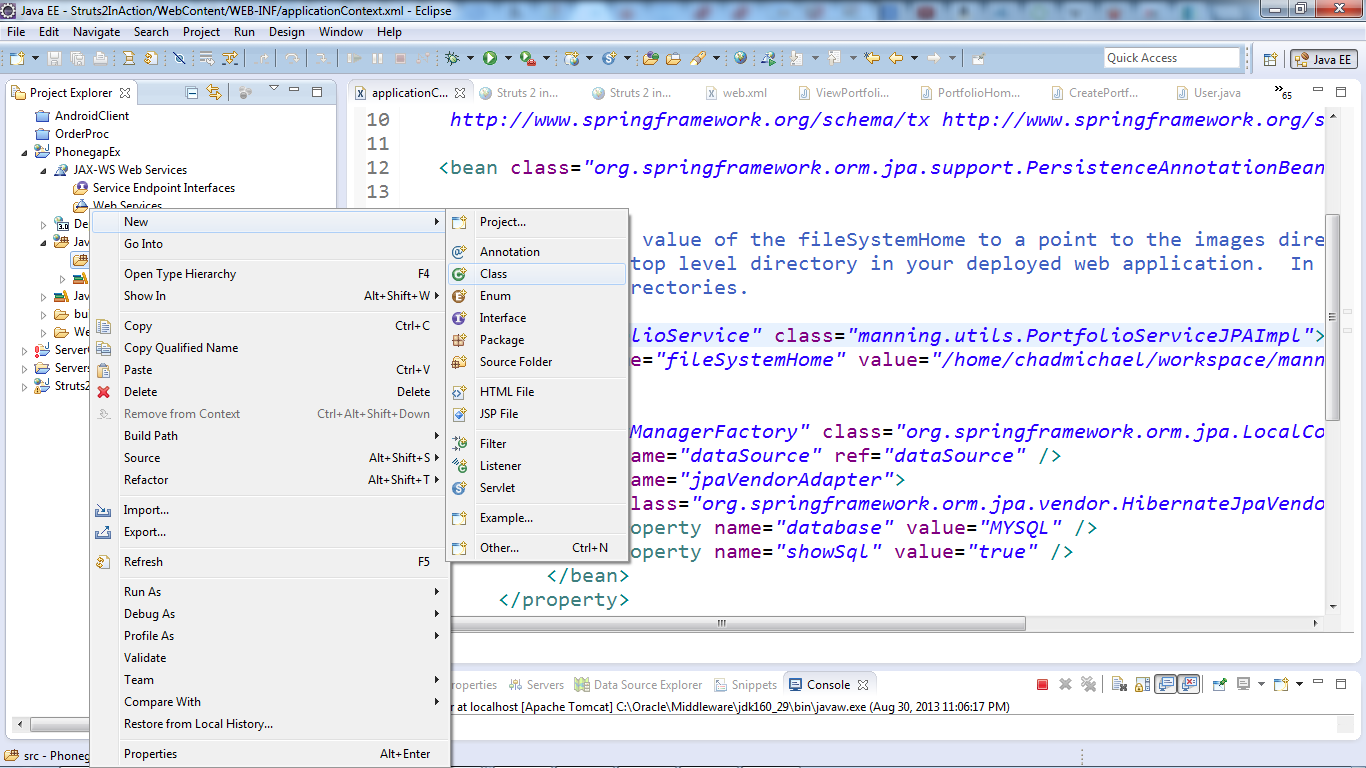
Write some java code to manipulate our database.



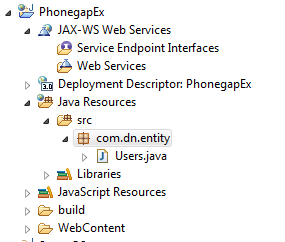


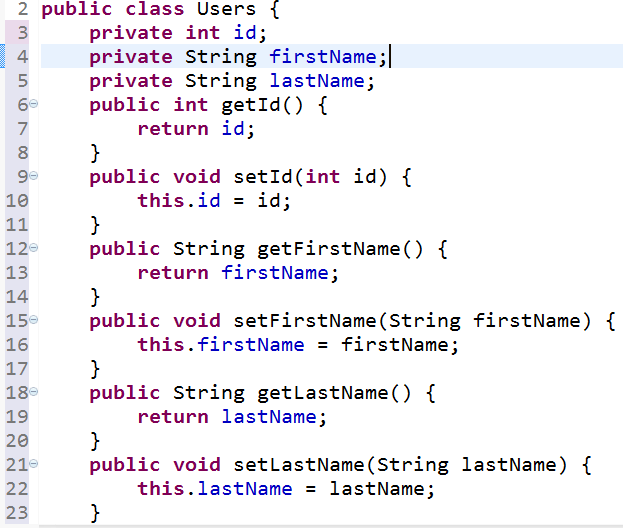


Click on “src” create new class

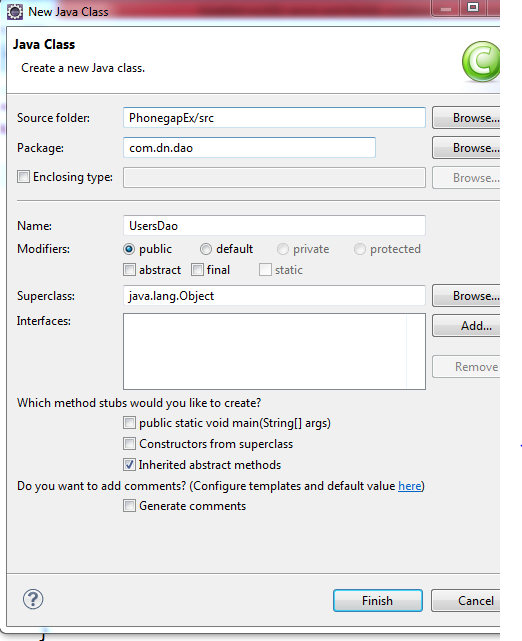


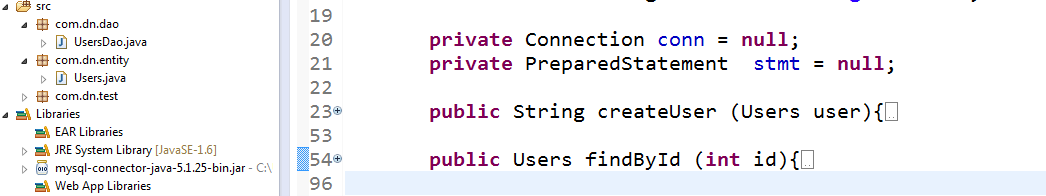
Enter Name class : Users and package” com.xxx.xxx”





Create Dao (data access object) by using jdbc low level ( this part can be replaced by jdbc template or Hibernate or so on).



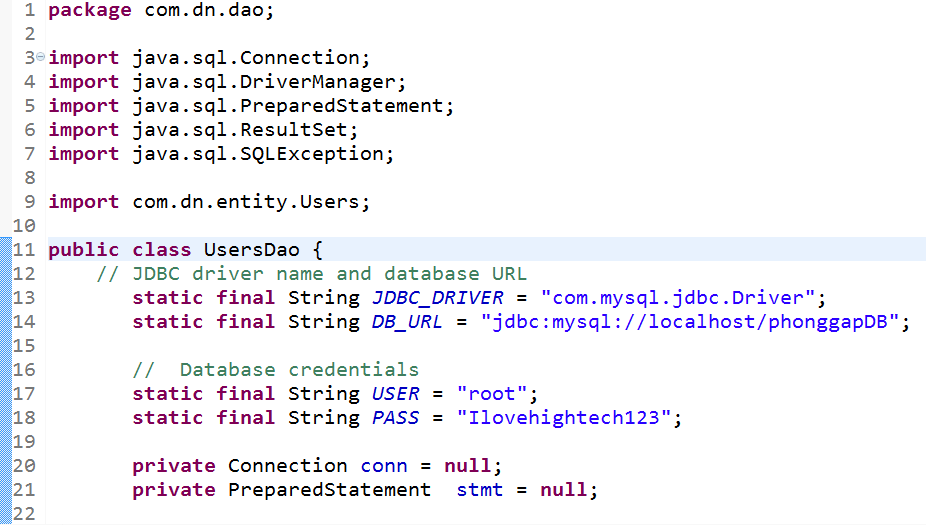


The UsersDao has 2 methods.

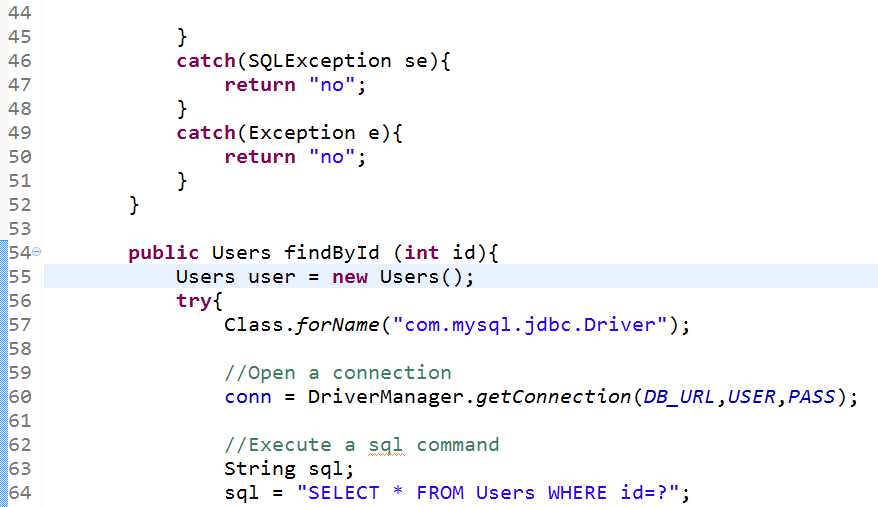
createUser: user to create a user to database. Return “yes” if save to database successfully; “no” if fails.

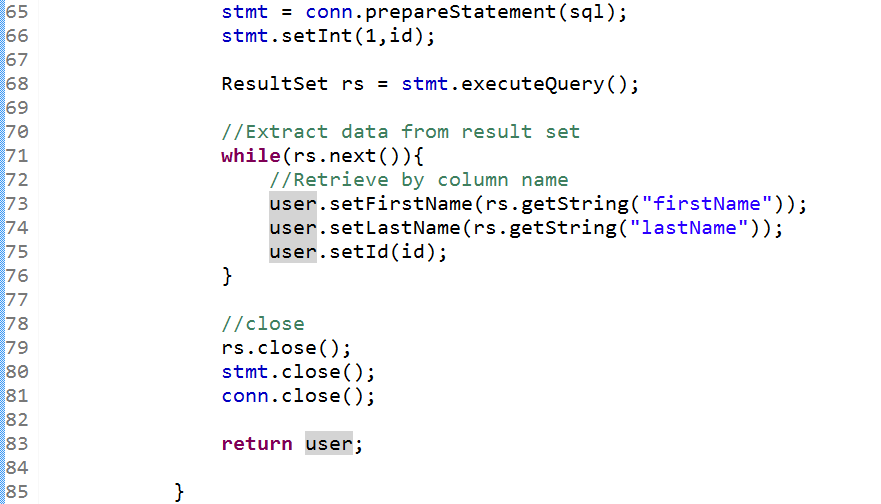
findById: find a user by id.

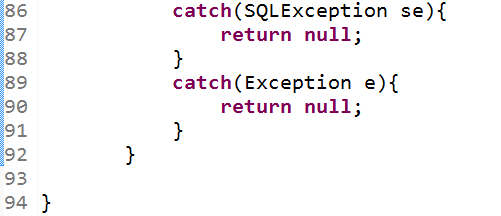
Source code



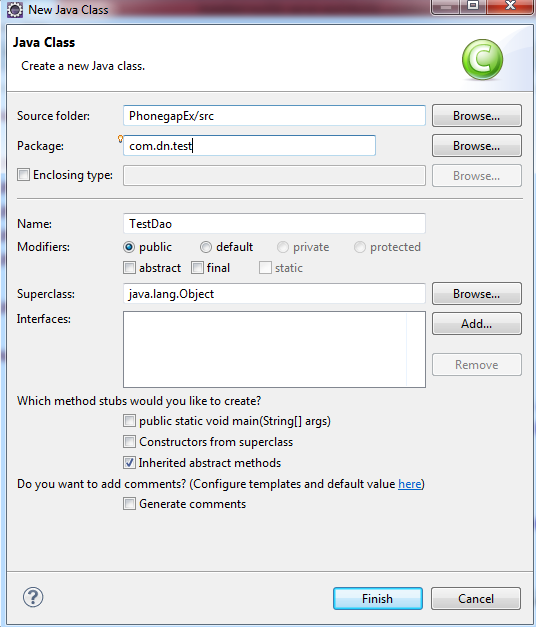


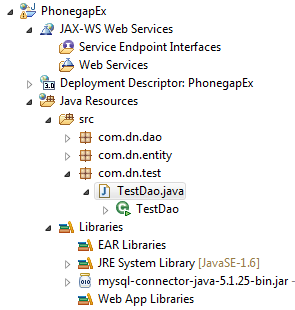




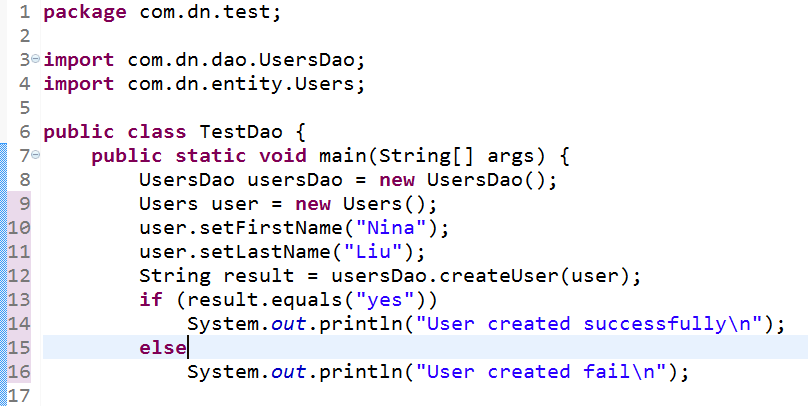


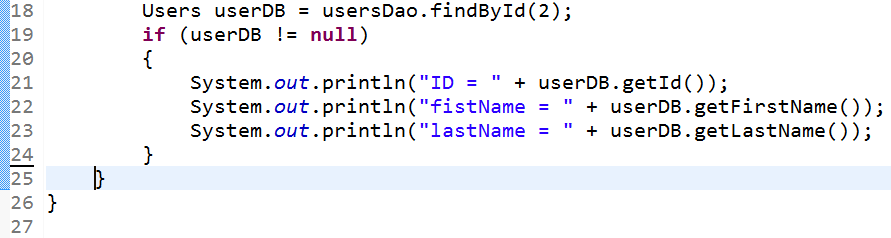
Do a little test with UsersDao class by using simple main method not JUnit.



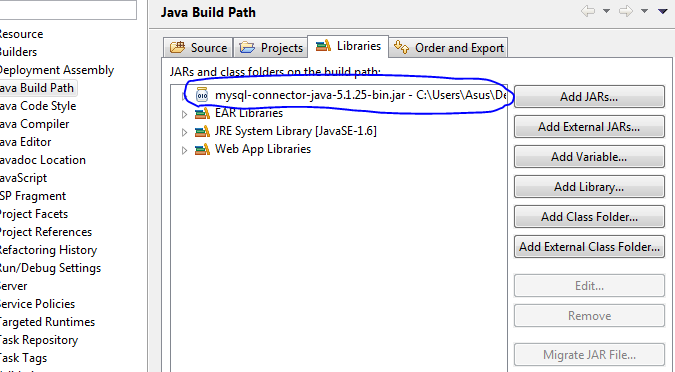


Source code

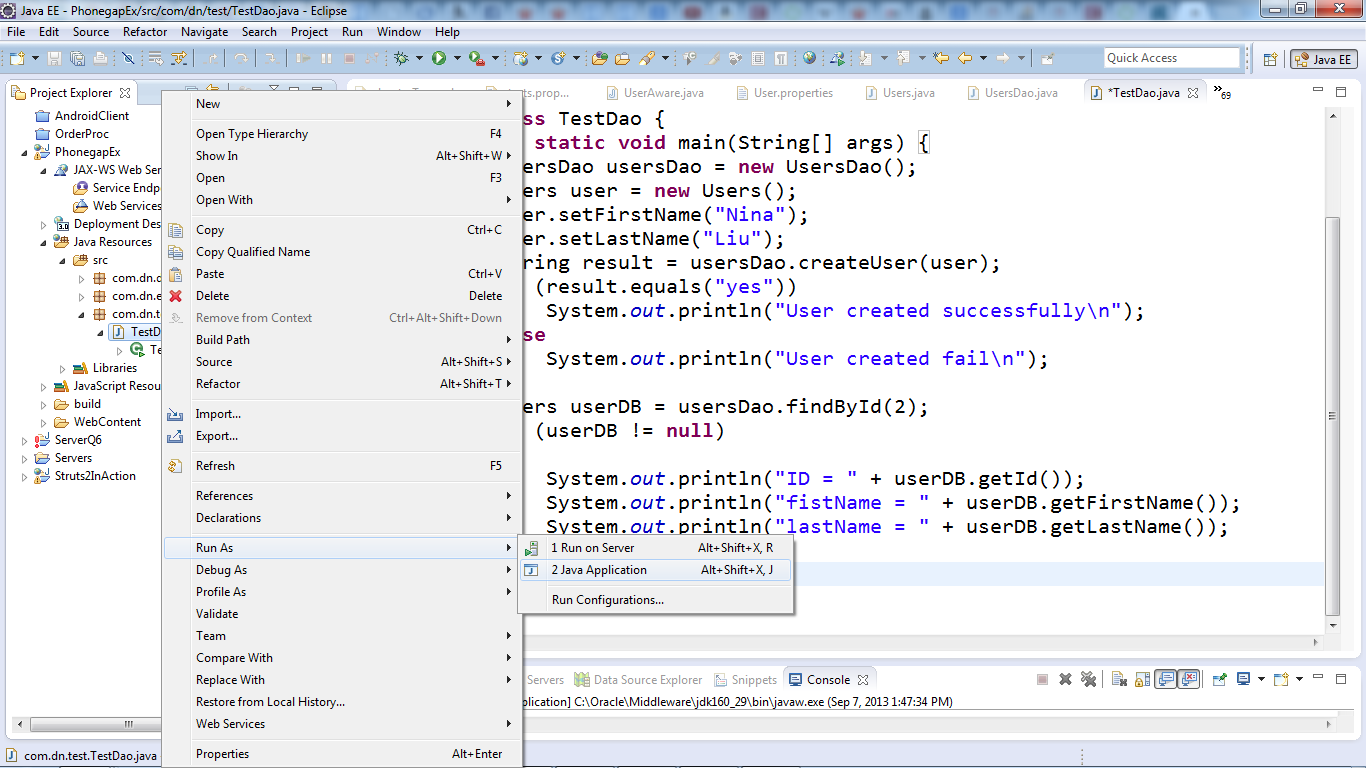


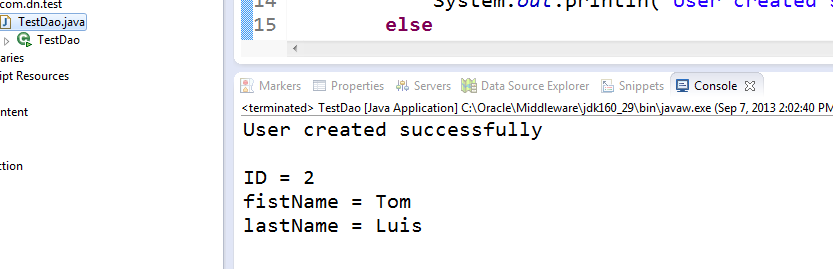


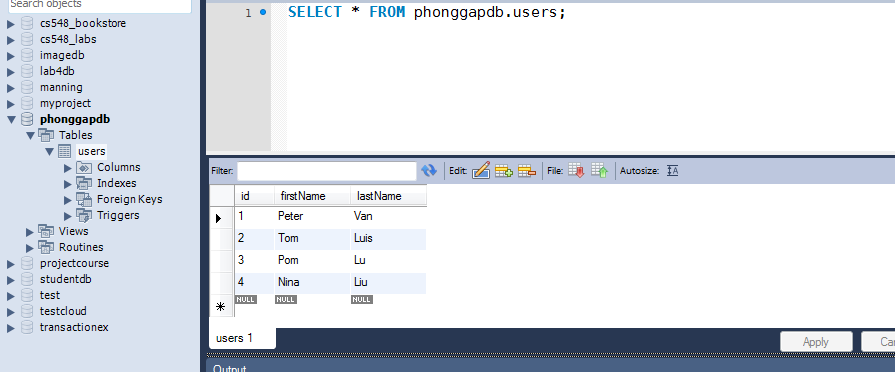
Remember: we need to add mysql-connector-java-xxx to the library



Click on TestDao and run



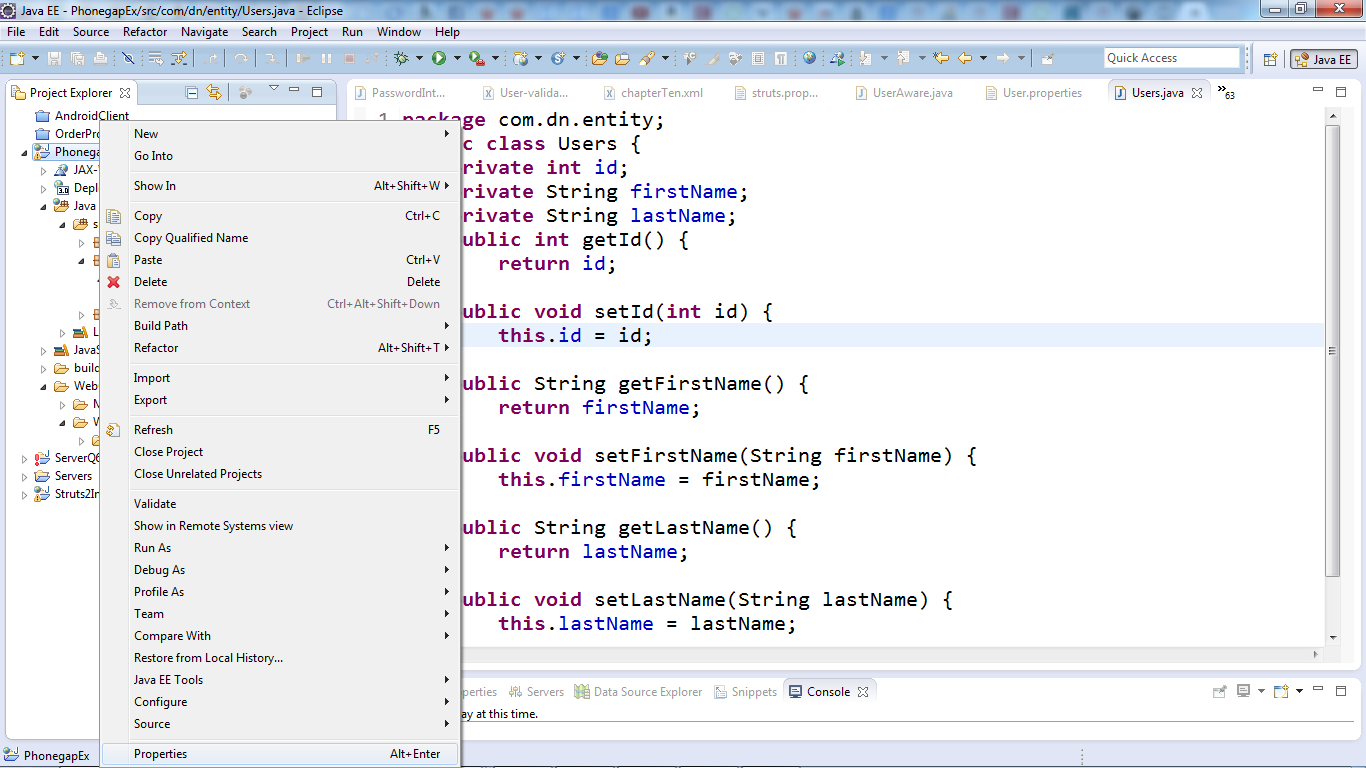


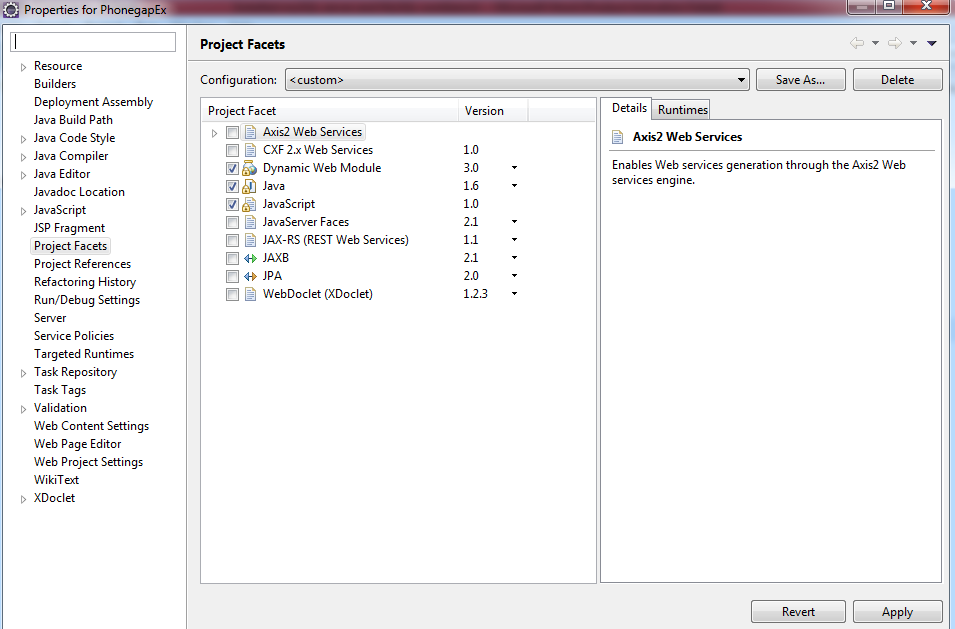


From mysql workbench, we see the user(Nina, Liu) is saved to database and findById(2) gave the right result.

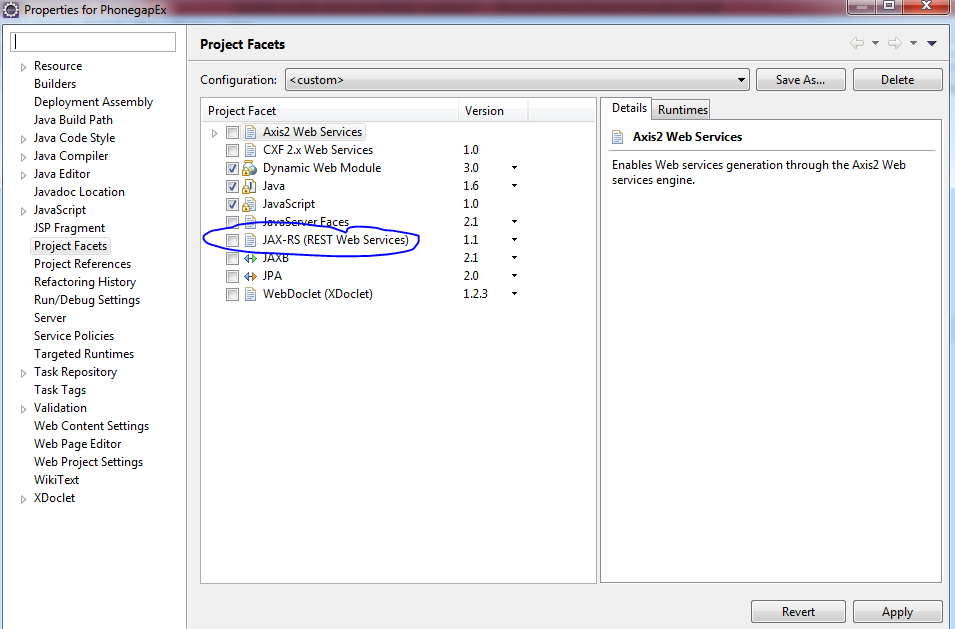
Next create a little Rest web service which wraps around UsersDao.

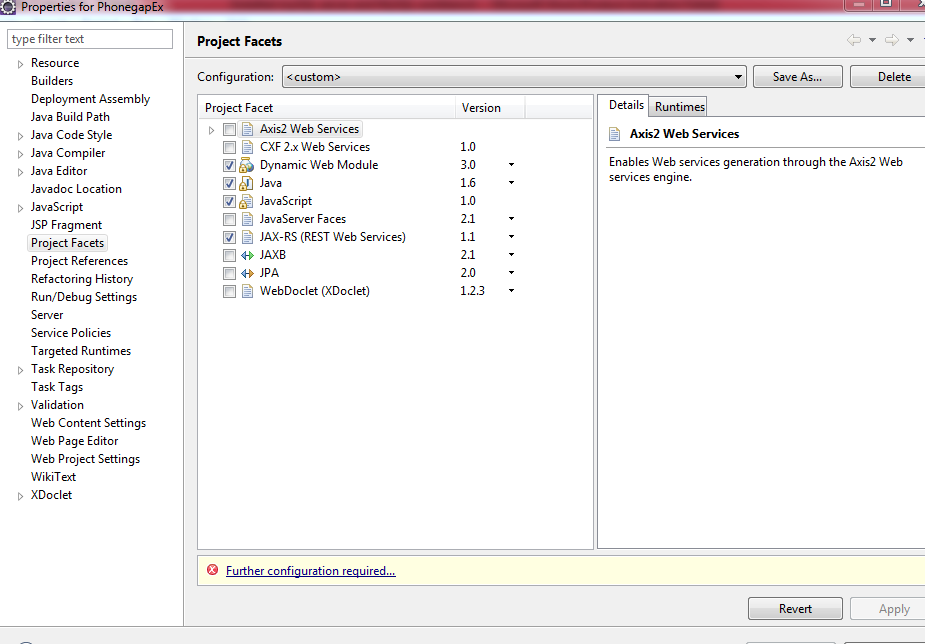
Right click on project and click Properties.



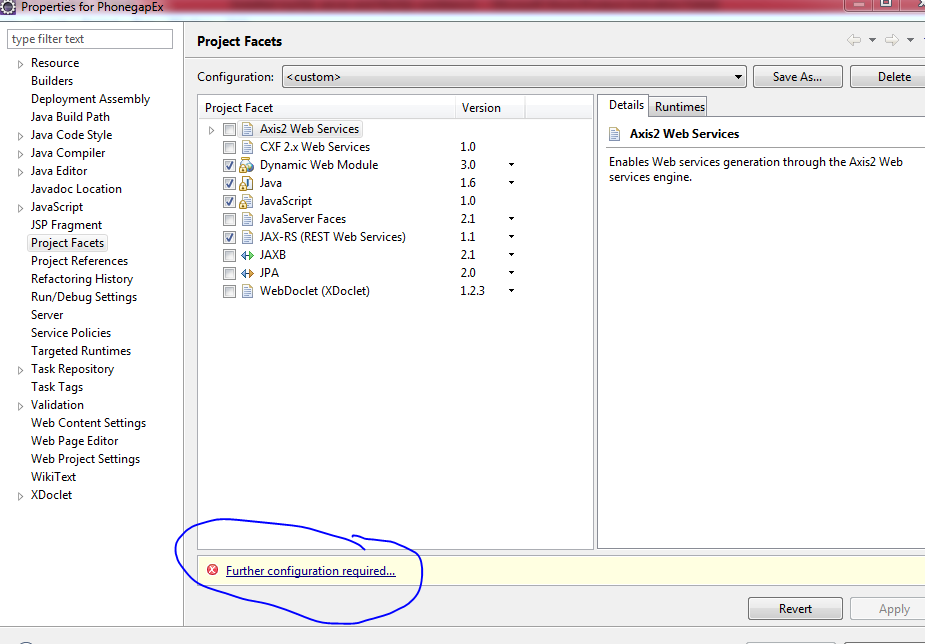


Click on JAX-RS

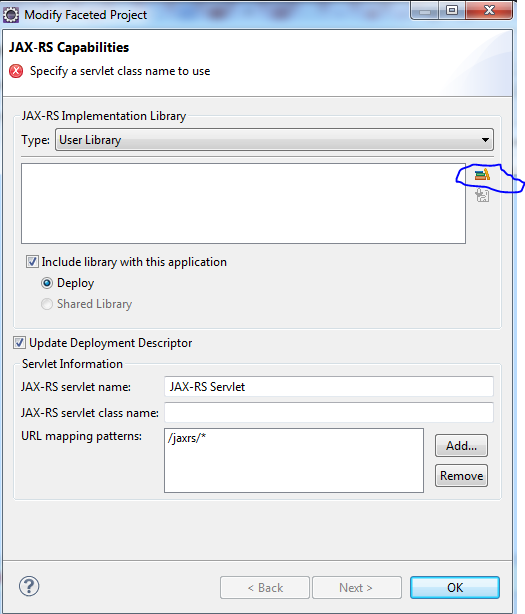


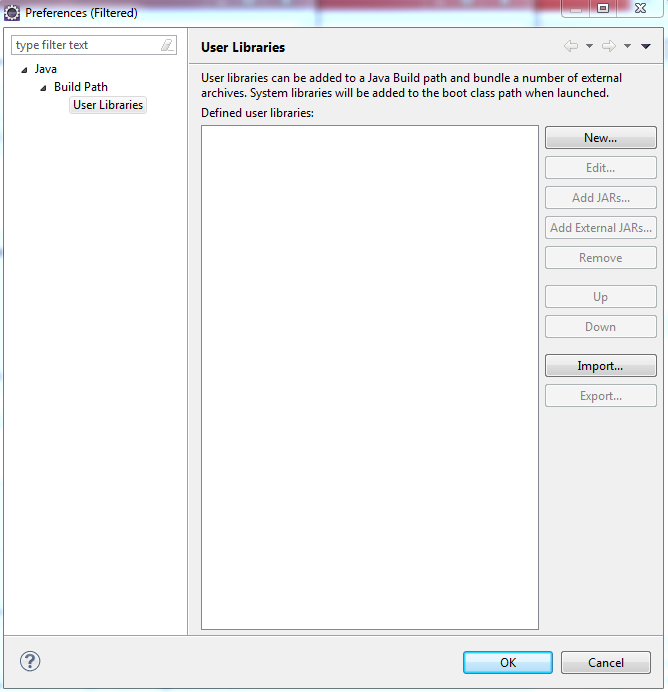


Click on further configuration



Click on Manage Library



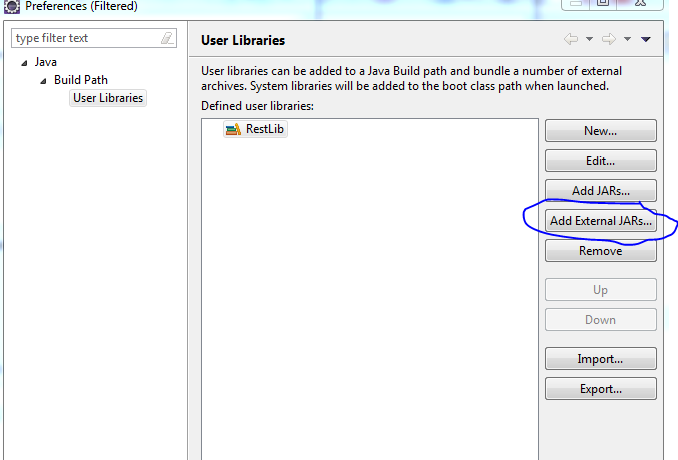


Click New

Give a name, For example RestLib



Click Ok.

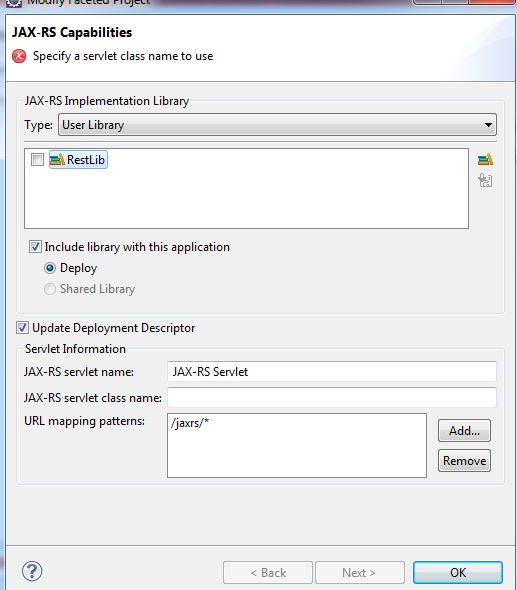


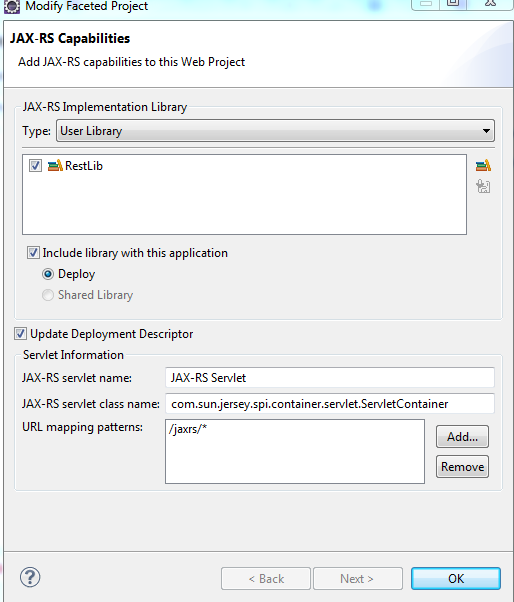
Click on Add external JARs

Add some necessary libraries

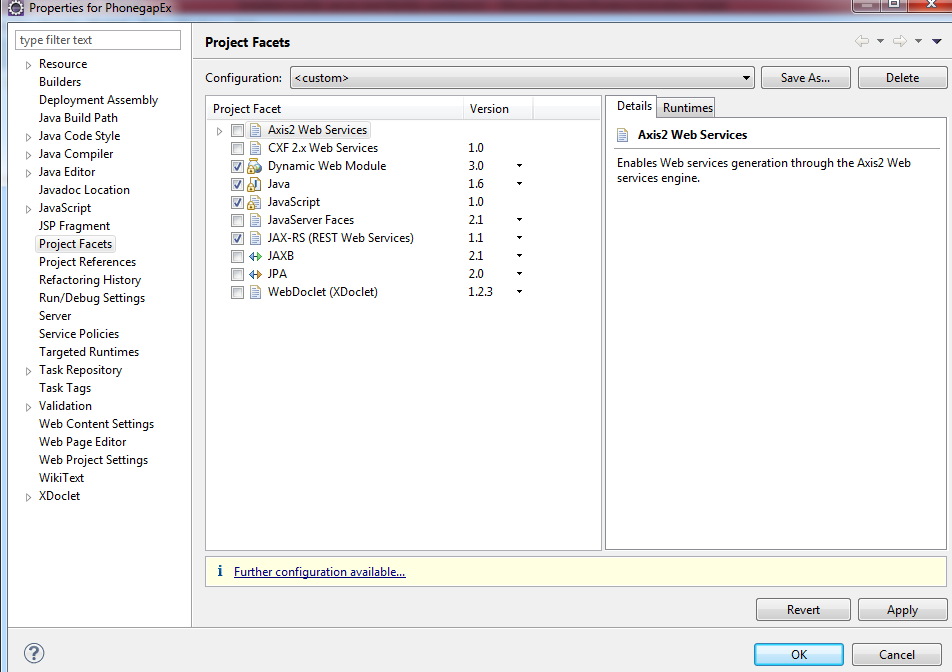


Click Ok



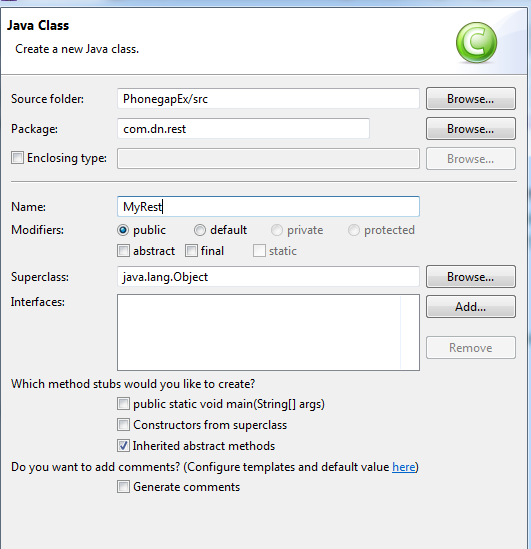


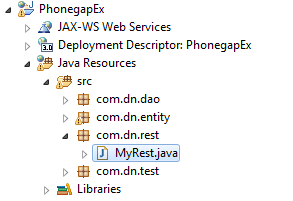
Click Ok



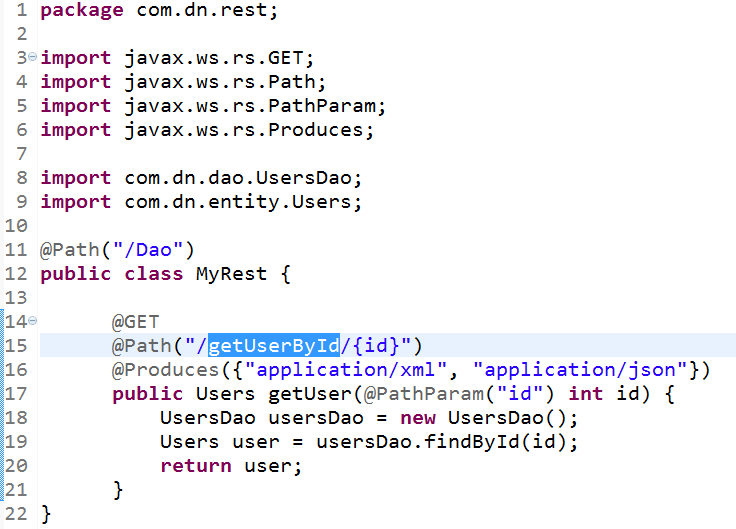
Click Ok. The web.xml is created as following



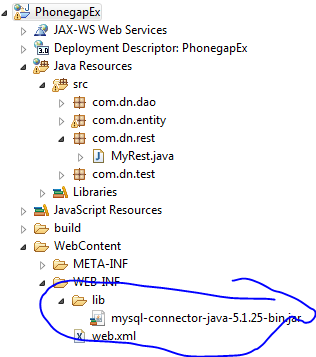




Put the following code to MyRest.java



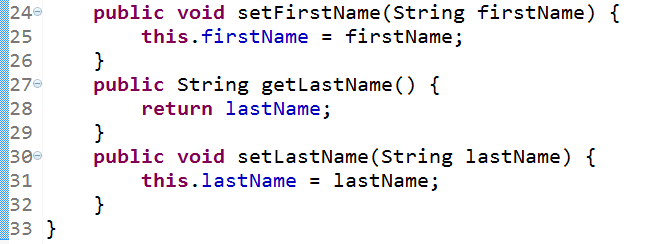
Before run the project on Tomcat server, we need to copy mysql-connector-java-xxx to lib folder as following



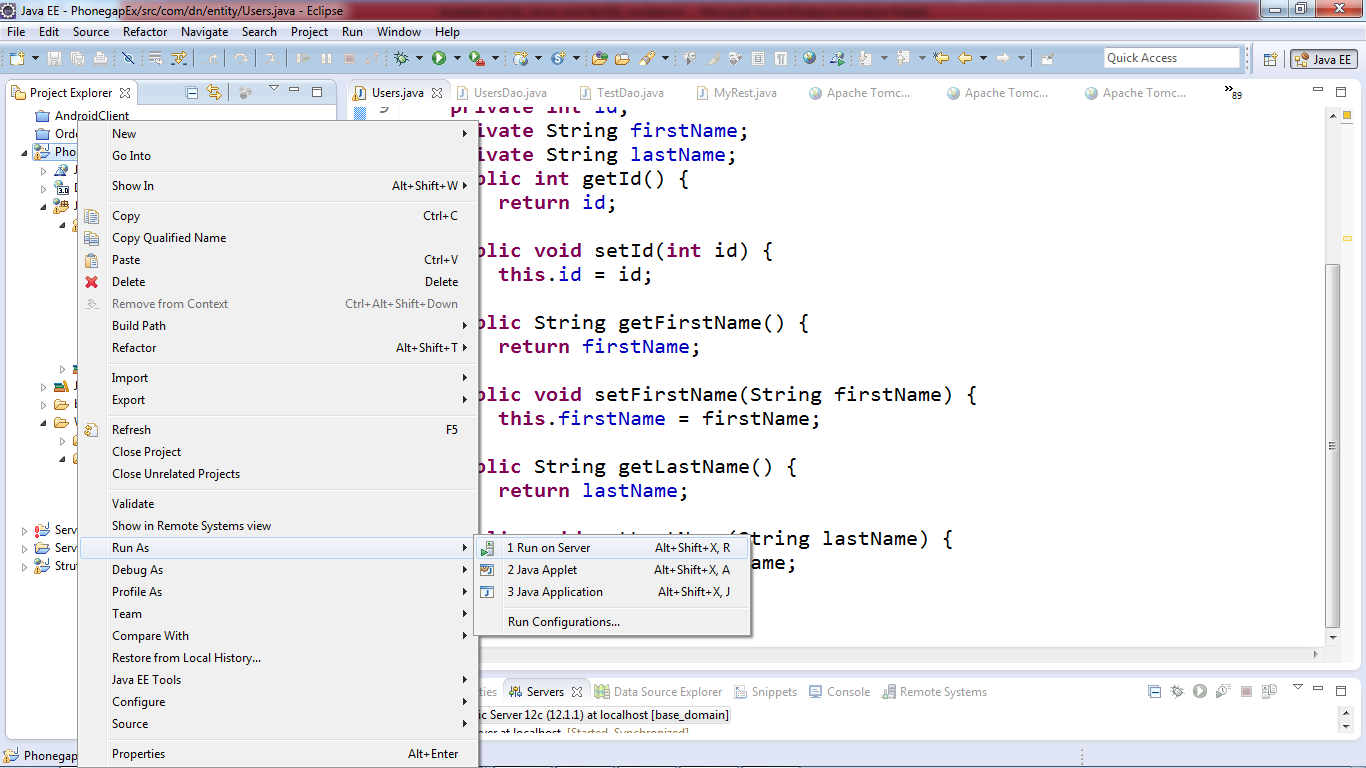
And

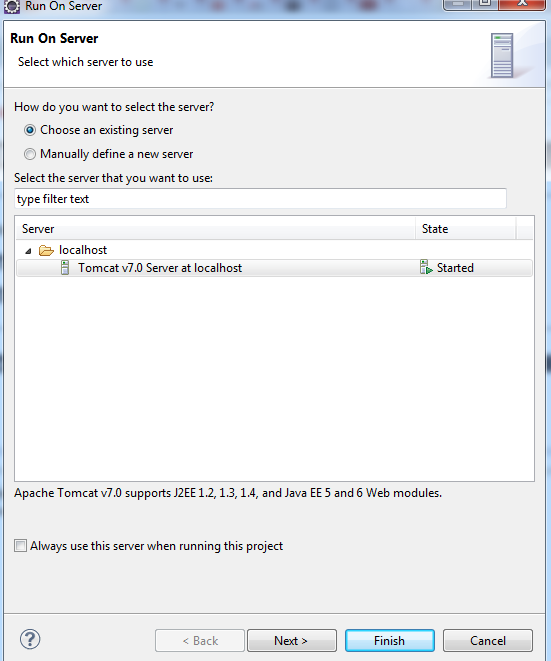
Change the code of the Users.java a little bit as following





Now run the project on Tomcat server





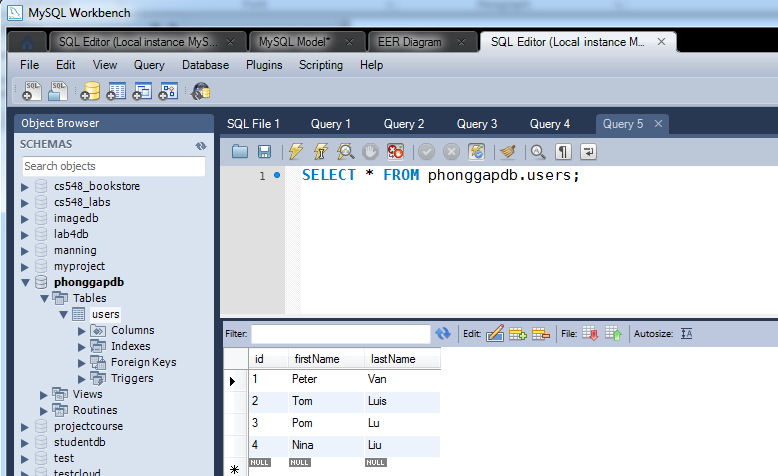
Use the browser to test the result.

Go to the link

<http://localhost:8080/PhonegapEx/jaxrs/Dao/getUserById/2>

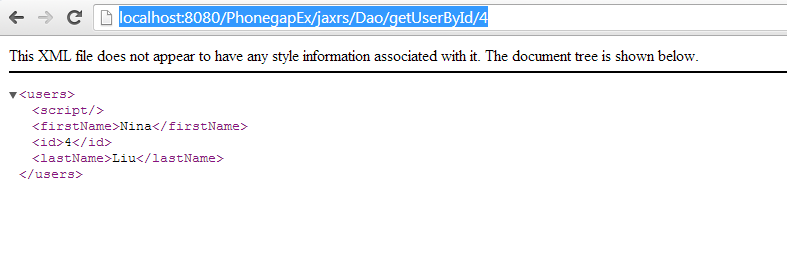
* Get the user who has id =2





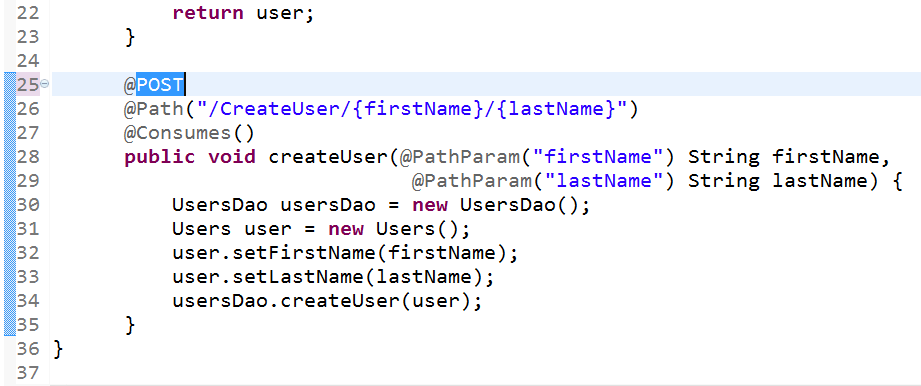
Try again with

<http://localhost:8080/PhonegapEx/jaxrs/Dao/getUserById/4>



Right now, the MyRest.java has only getUserById. Add more method to MyRest.java as the following.





And now we can save a new user to database by using uri with a POST method

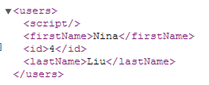
<http://localhost:8080/PhonegapEx/jaxrs/Dao/CreateUser/firstName/lastName>

For example, to save a user with first name : Tina, last name: Le 🡪

http:// localhost:8080/PhonegapEx/jaxrs/Dao/CreateUser/Tina/Le + POST method. More detail: next section.

1. **Programming for client side.**

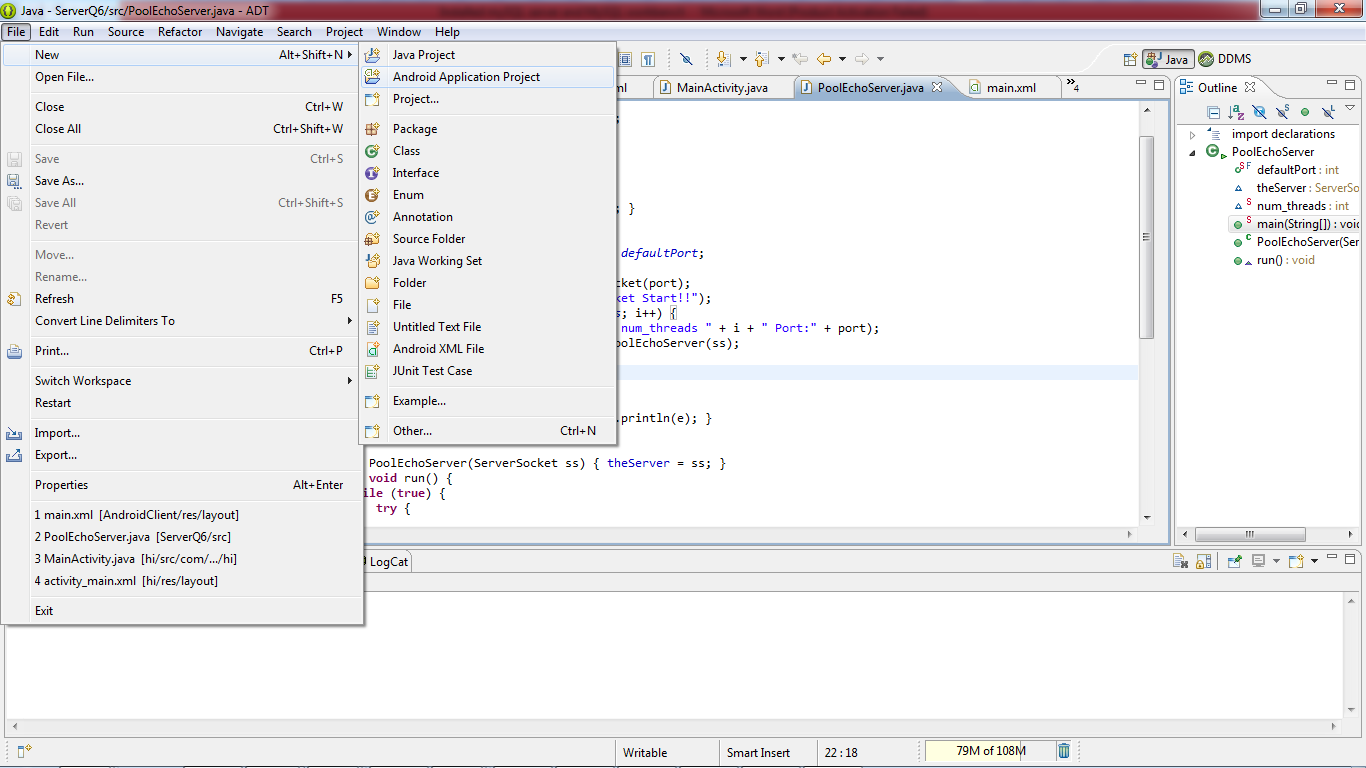
The client side can use the Rest Web service in the above to get information of user from database. For example, access this link <http://localhost:8080/PhonegapEx/jaxrs/Dao/getUserById/4>, we will get

 in XML or in Json.

The client can also create a new user and save to database. For example,

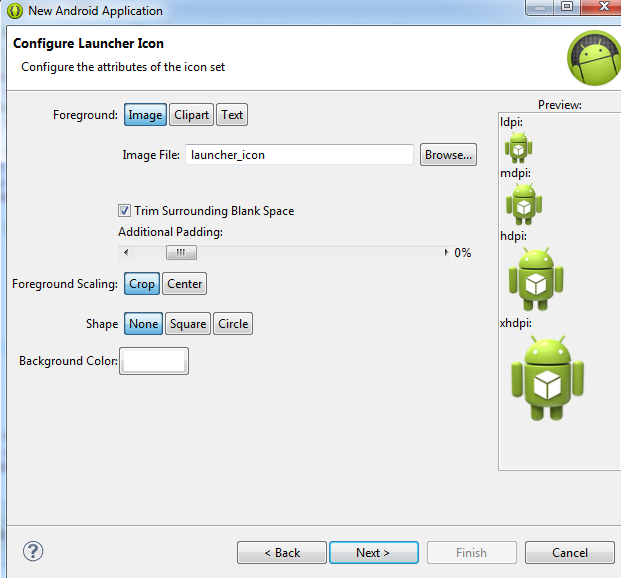
In this section, we will use html, jQuery, and Phonegap to build Android application. Even we don’t know much about Android programming.

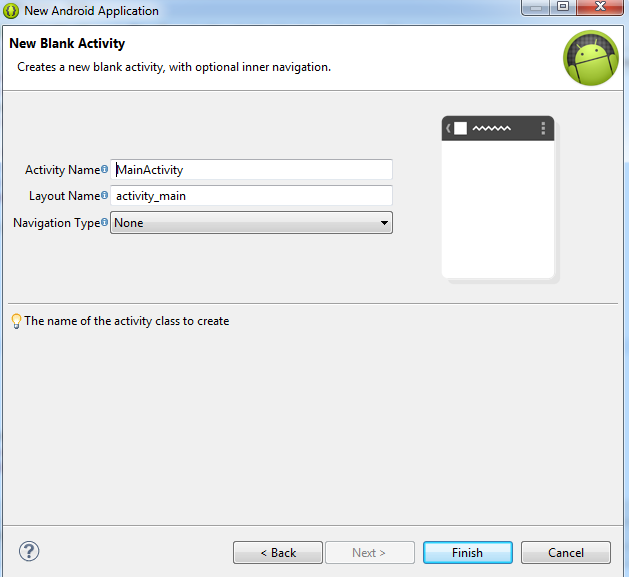
Open Eclipse ADT-bundle and create a new Android project.



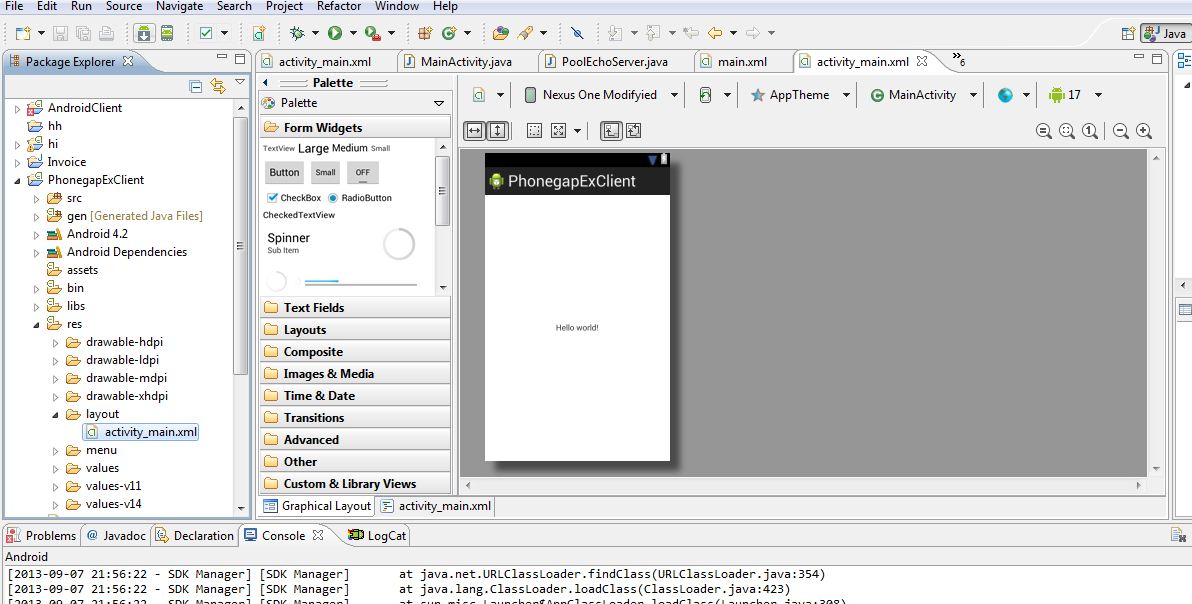


Click next, next..

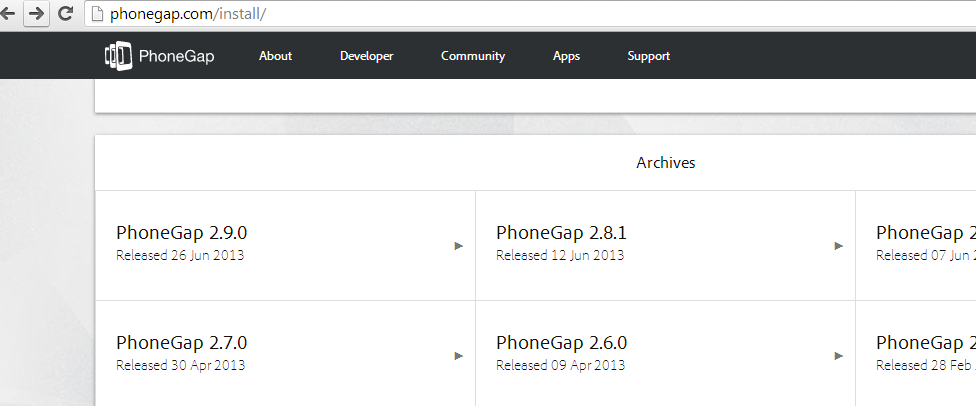




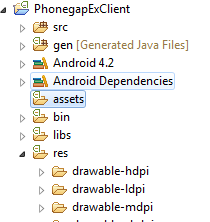
And finish

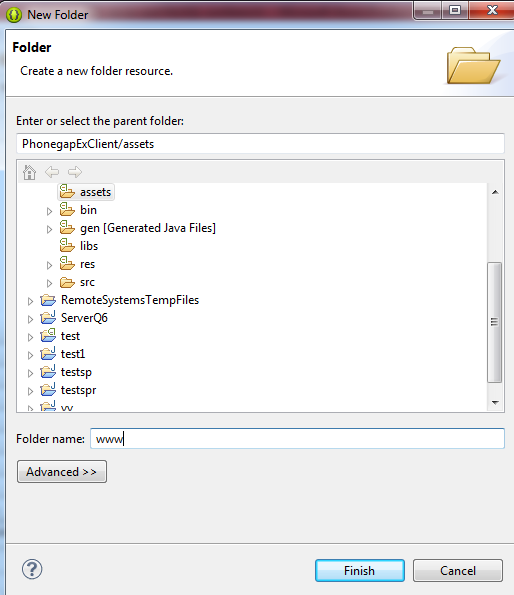


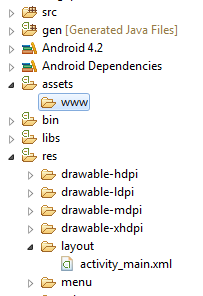
Go to phonegap website and download phonegap package.



Create a folder www in the assets folder







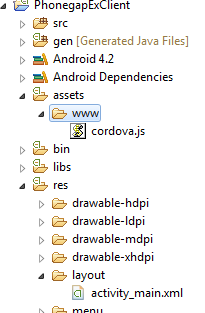
After downloading phonegap, unzip it



Go to lib folder, Android folder



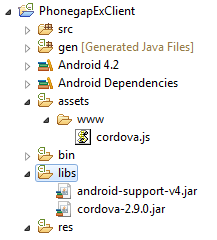
Copy cordova.js to www folder

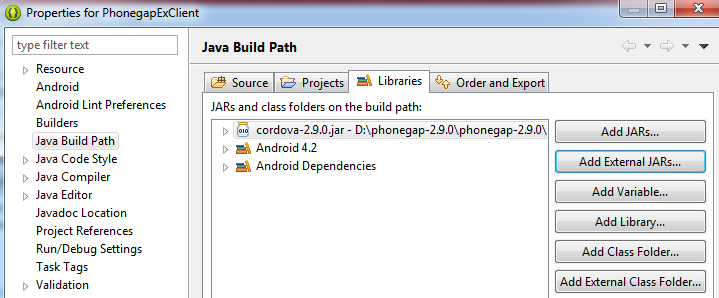


Also copy

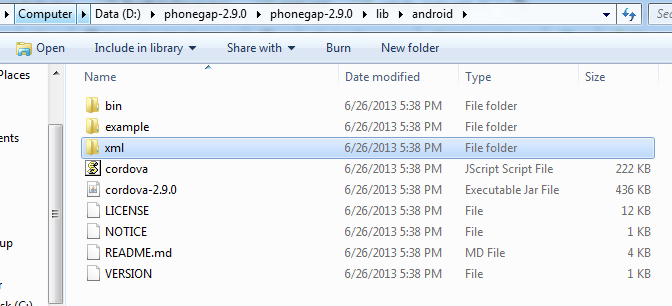


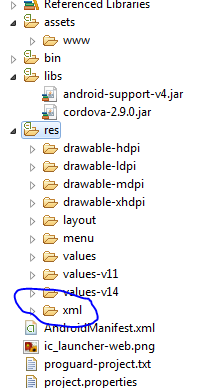
To libs folder and add it to classpath





Copy xml folder to res folder

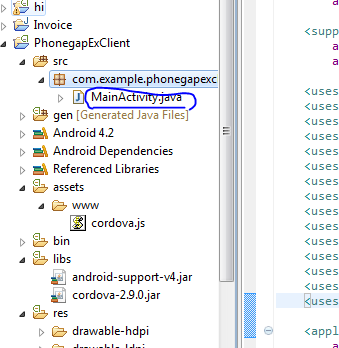


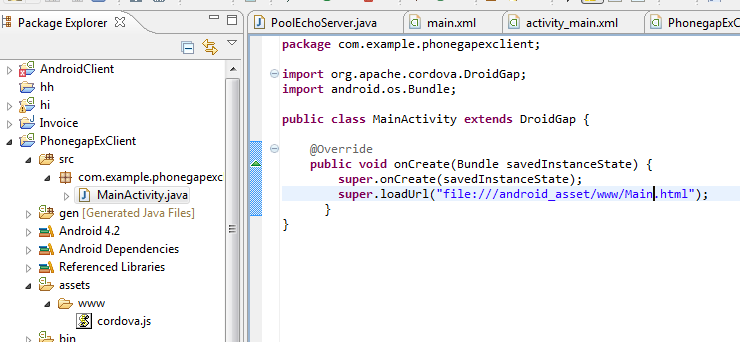


Open AndroidManifest.xml and add some code for screen and permission as following



Open MainActivity.java and change the code into

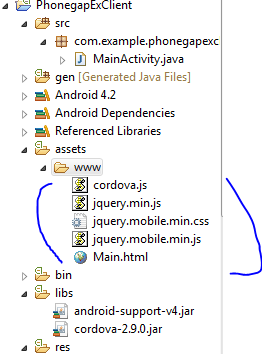




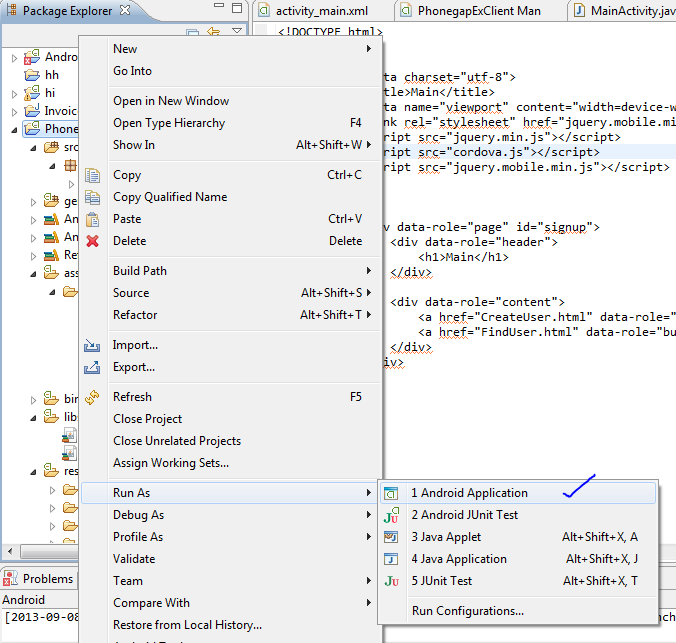
Add Main.html and some js libraries to www folder

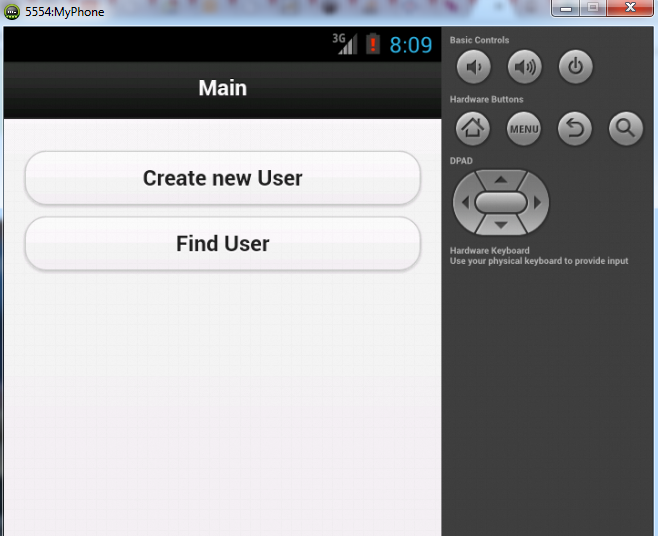
Main.html



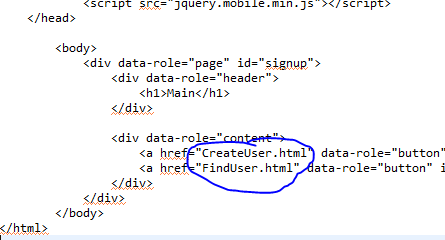


Click Run Android Application





Next we need add 2 files: CreateUser.html and FindUser.html to www folder

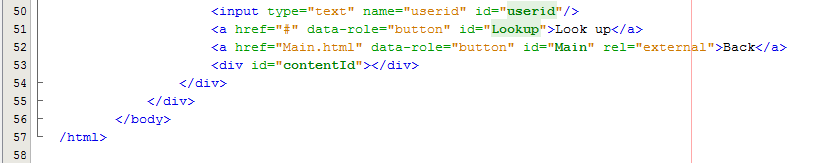


First, doing with FindUser.html

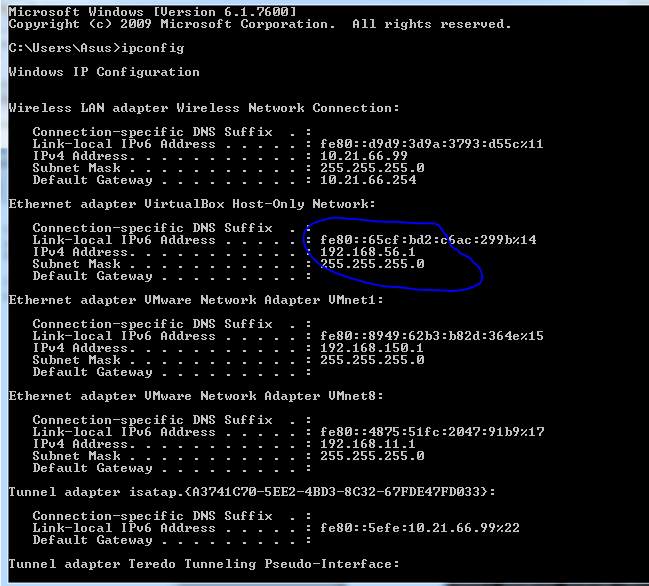
FindUser.html





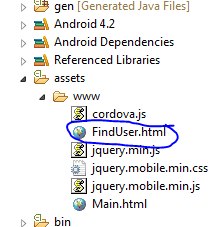


For the link, we cannot use localhost, we need to replace with our host ip

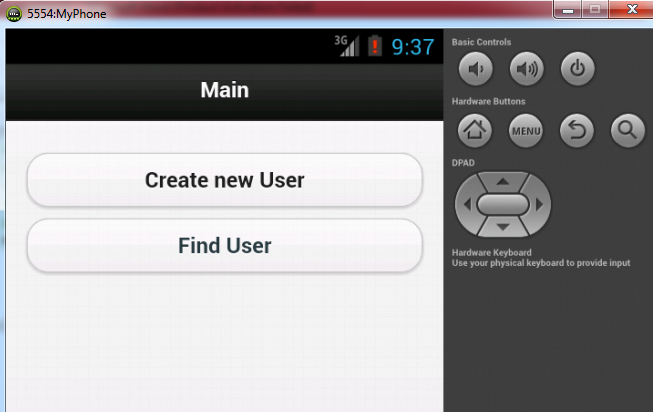


Use 192.168.56.1

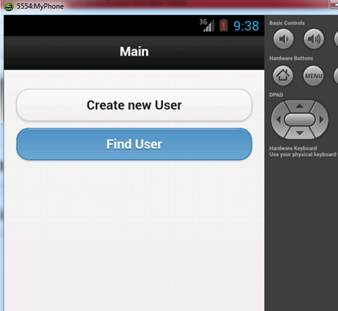


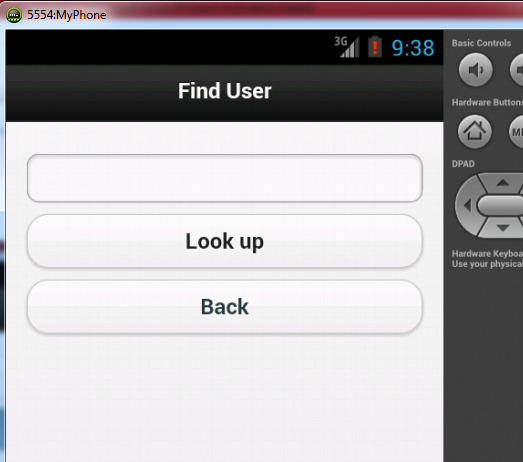


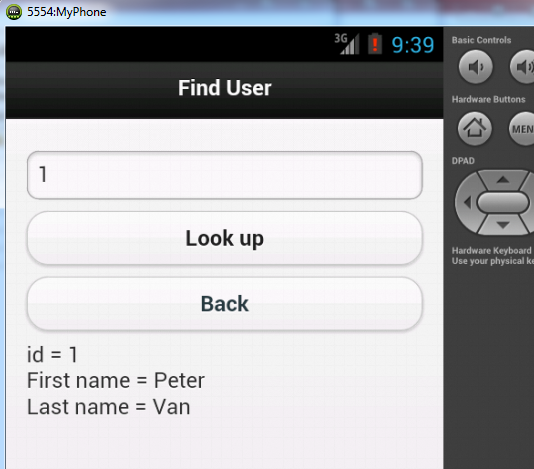
Now run the program

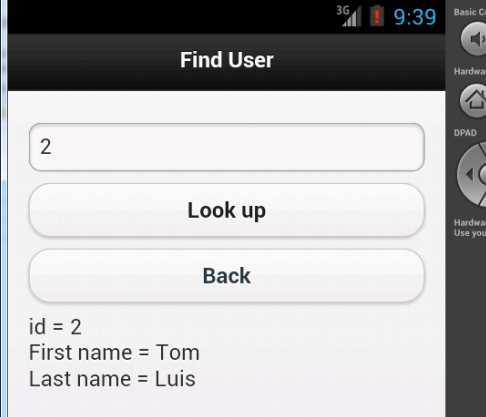


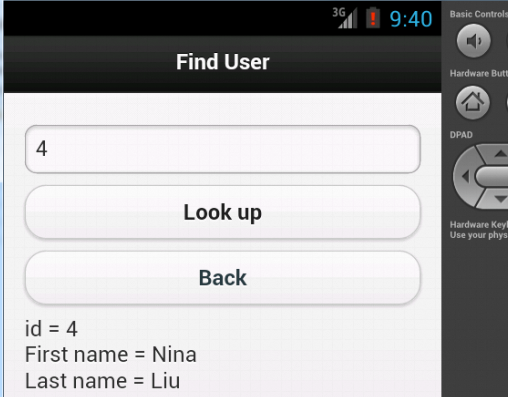
Click on Find User



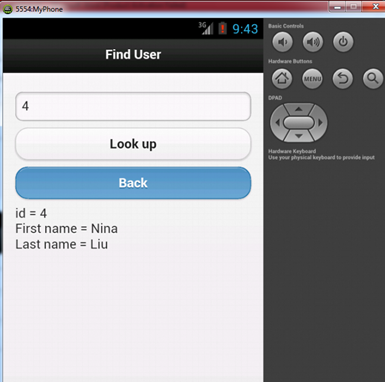








Click Back



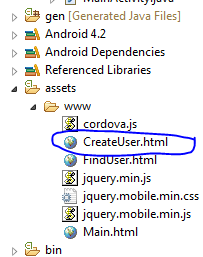


Second, doing with CreateUser.html

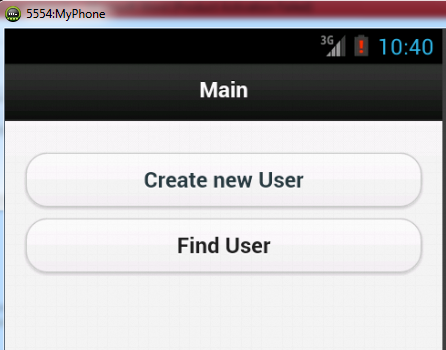
CreateUser.html



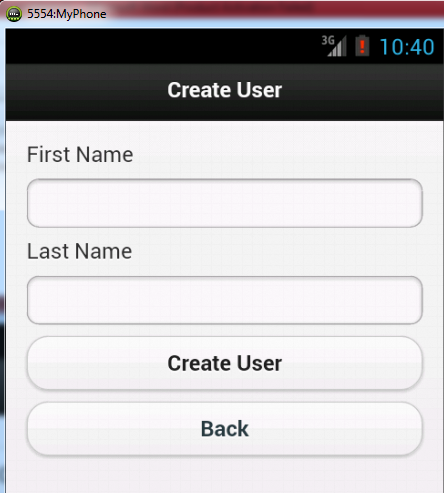


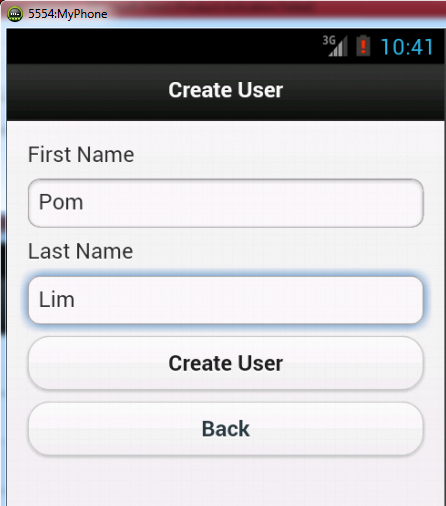


Run the program

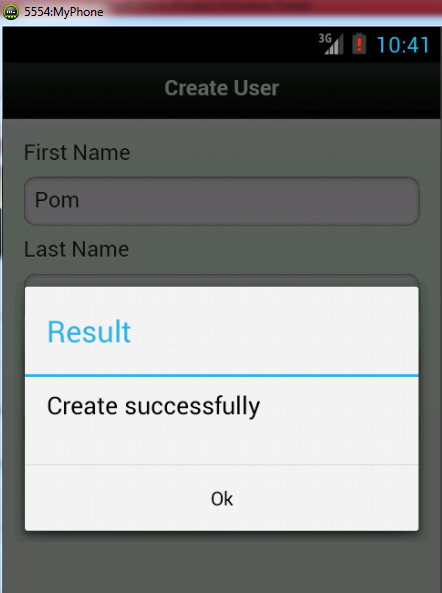


Click Create New User



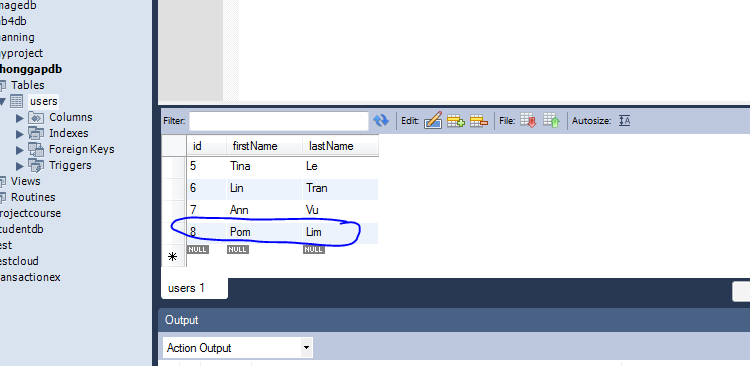


Click Create User



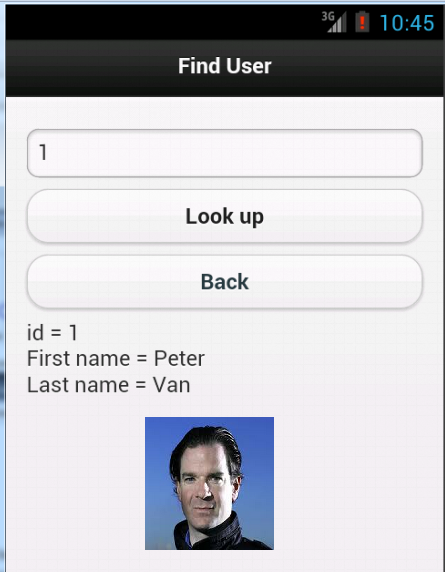
Click Ok

Check database



-------------------------------------------------------

The above example can be added extended more such as add more column (image 🡪 allow upload image and retrieve image)



Or save information of users from server to contact database of the phone by using phonegap api : navigator.contacts.create();…..