# CS 564

# Database Management Systems

University of Wisconsin - Madison

Madison, WI 53706

## Project Stage #5

**Submitted by:**

Jin Ruan

Shuang Wu

Qing Li

**Submitted to:**

Prof. AnHai Doan

**Date Due**: Apr 24, 2015

- **Platforms & Languages:**

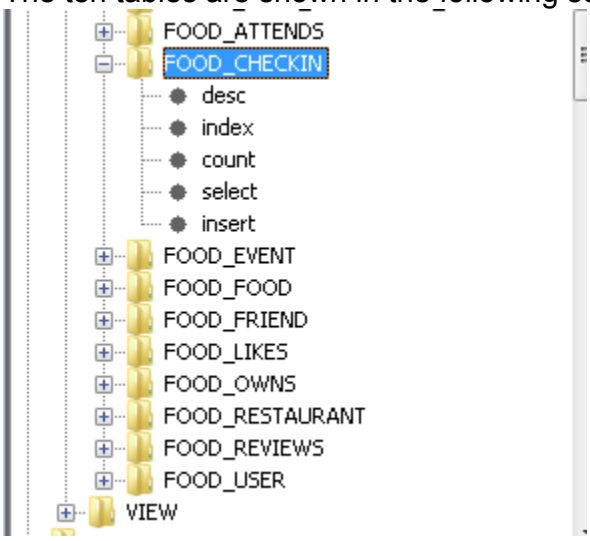
We used SQLite as our relational database. Struts including jsp, java, xml is used as our web application.

- **Database Tables:**

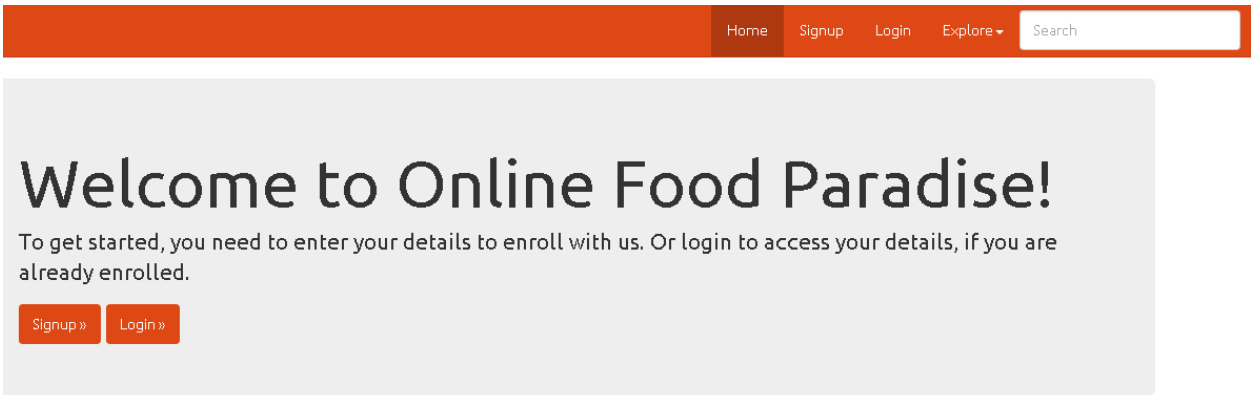
We have ten tables in total. Here is one sample of creating a table:

```
CREATE TABLE COMMMAPSDEV.food_user (
        USER_ID      VARCHAR2(16) NOT NULL,
        USER_PASSWORD        VARCHAR2(16) NOT NULL,
        NICKNAME        VARCHAR2(16) NOT NULL,
          AVERAGE_RATINGS          NUMBER (2,1),
        MEMBER_SINCE   DATE DEFAULT (sysdate),
          PRIMARY KEY (USER_ID));
```

The ten tables are shown in the following screenshot:




- **Here is the interface looks like:**

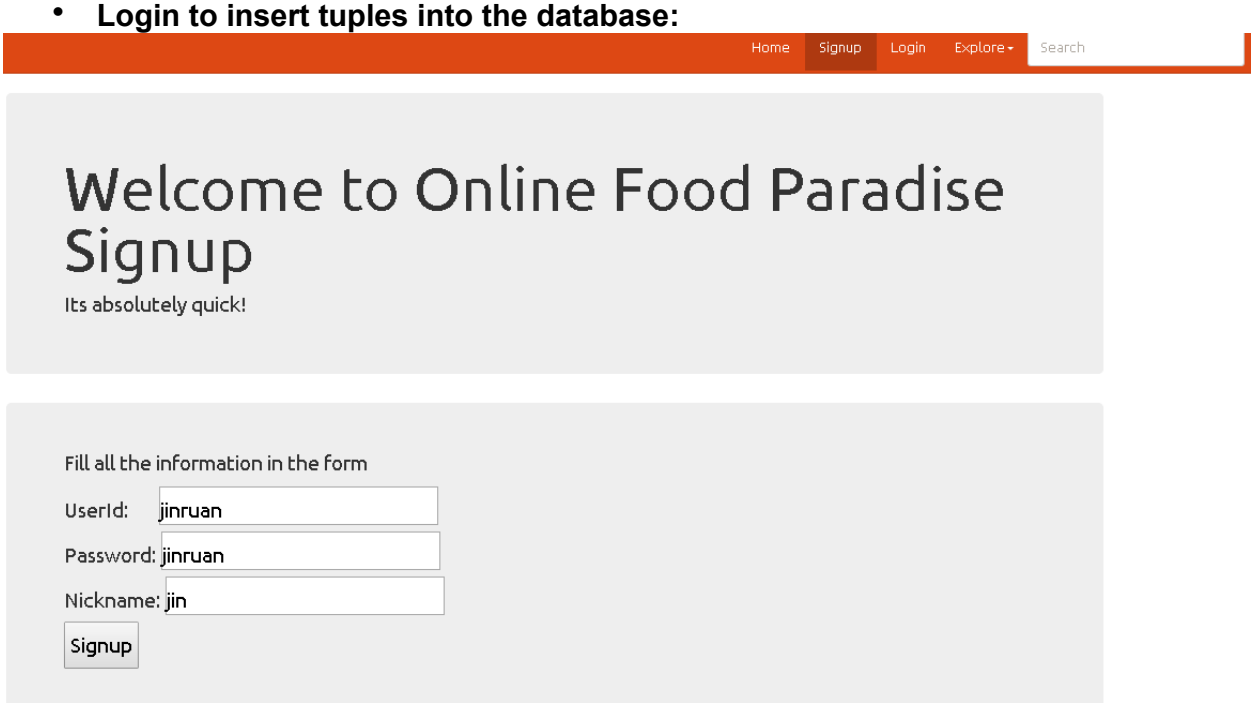



- **Here is the codes look like (Struts was used as MVC.):**

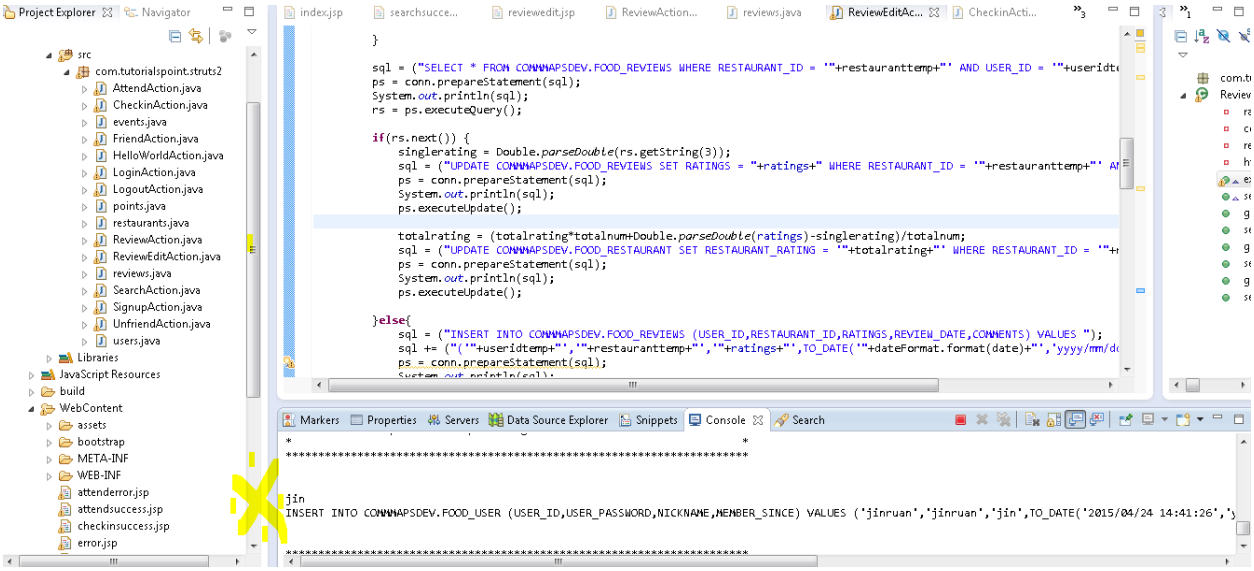
```
}

sql = ("SELECT * FROM COMMMAPSDEV.FOOD_REVIEWS WHERE RESTAURANT_ID = '"+restauranttemp+"' AND USER_ID = '"+useridte
ps = conn.prepareStatement(sql);
System.out.println(sql);
rs = ps.executeQuery();

if(rs.next()) {
    singlerating = Double.parseDouble(rs.getString(3));
    sql = ("UPDATE COMMMAPSDEV.FOOD_REVIEWS SET RATINGS = "+ratings+" WHERE RESTAURANT_ID = '"+restauranttemp+"' AN
    ps = conn.prepareStatement(sql);
    System.out.println(sql);
    ps.executeUpdate();

    totalrating = (totalrating*totalnum+Double.parseDouble(ratings)-singlerating)/totalnum;
    sql = ("UPDATE COMMMAPSDEV.FOOD_RESTAURANT SET RESTAURANT_RATING = '"+totalrating+"' WHERE RESTAURANT_ID = '"+r
    ps = conn.prepareStatement(sql);
    System.out.println(sql);
    ps.executeUpdate();

}else{
    sql = ("INSERT INTO COMMMAPSDEV.FOOD_REVIEWS (USER_ID,RESTAURANT_ID,RATINGS,REVIEW_DATE,COMMENTS) VALUES ");
    sql += ("('"+useridtemp+"','"+restauranttemp+"','"+ratings+"',TO_DATE('"+dateFormat.format(date)+"','yyyy/mm/d
    ps = conn.prepareStatement(sql);
    System.out.println(sql);
    ps.executeUpdate();

    totalrating = (totalrating*totalnum+Double.parseDouble(ratings))/(totalnum+1);
    totalnum +=1;
    sql = ("UPDATE COMMMAPSDEV.FOOD_RESTAURANT SET RESTAURANT_RATING = '"+totalrating+"', RESTAURANT_RATING_NUM =
    ps = conn.prepareStatement(sql);
    System.out.println(sql);
```

Tomcat v7.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre7\bin\javaw.exe (Apr 24, 2015, 2:32:25 PM)

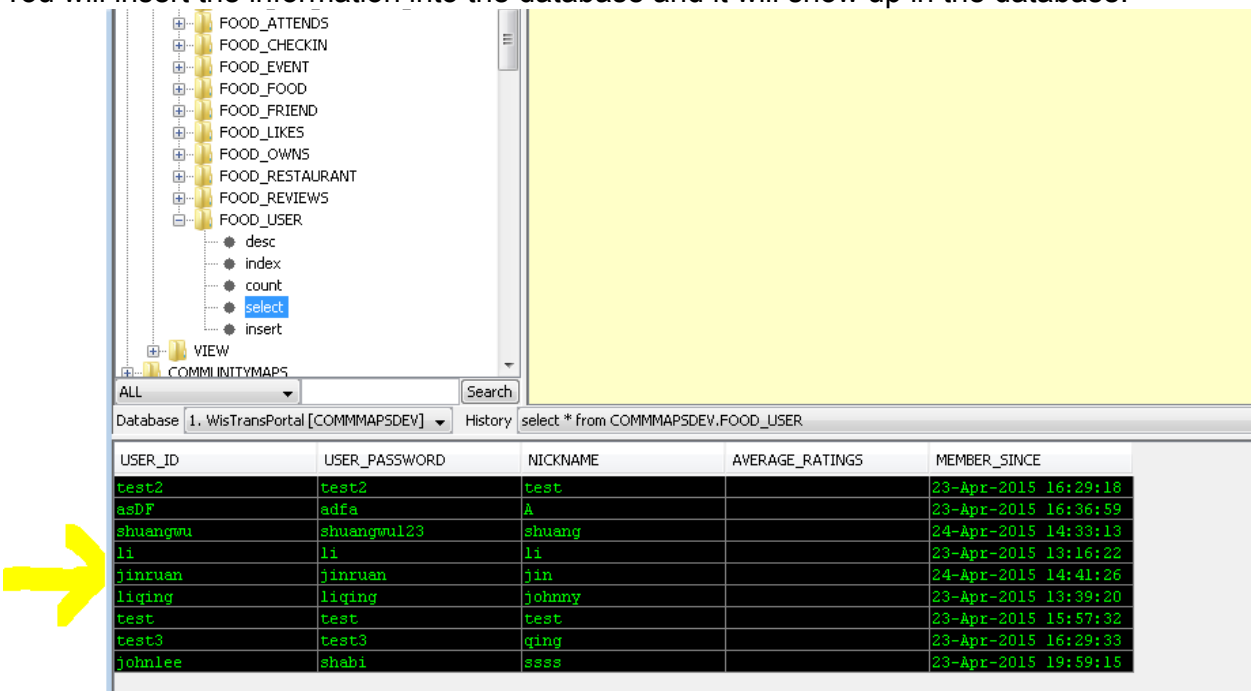
- **Login to insert tuples into the database:**




After filling the signup form, the codes will read the information in the frontend and translates it into the SQL insert query, here is the screenshot of the codes and the sql query it translated:

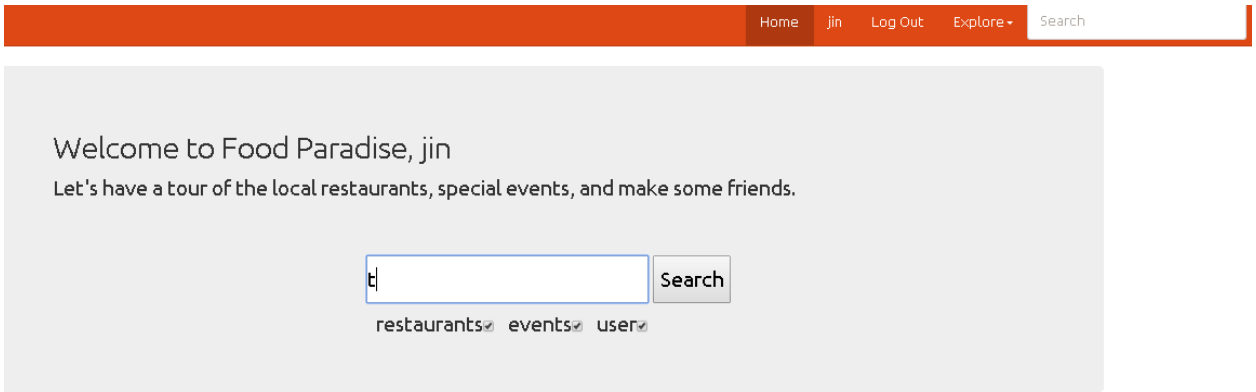



You will insert the information into the database and it will show up in the database:




| USER_ID | USER_PASSWORD | NICKNAME | AVERAGE_RATINGS | MEMBER_SINCE |
|---|---|---|---|---|
| test2 | test2 | test | | 23-Apr-2015 16:29:18 |
| asDF | adfa | A | | 23-Apr-2015 16:36:59 |
| shuangwu | shuangwu123 | shuang | | 24-Apr-2015 14:33:13 |
| li | li | li | | 23-Apr-2015 13:16:22 |
| jinruan | jinruan | jin | | 24-Apr-2015 14:41:26 |
| liqing | liqing | johnny | | 23-Apr-2015 13:39:20 |
| test | test | test | | 23-Apr-2015 15:57:32 |
| test3 | test3 | qing | | 23-Apr-2015 16:29:33 |
| johnlee | shabi | ssss | | 23-Apr-2015 19:59:15 |

- **Search query will ask queries over database:**




Then since you selected restaurants, events, and users, it will ask all the matched restaurants, events, and restaurants. The query here involves **joint operator**.
The SQL query the codes made will be:

```
SELECT * FROM COMMMAPSDEV.FOOD_EVENT,COMMMAPSDEV.FOOD_RESTAURANT WHERE
EVENT_NAME LIKE '%t%' AND COMMMAPSDEV.FOOD_RESTAURANT.RESTAURANT_ID =
COMMMAPSDEV.FOOD_EVENT.RESTAURANT_ID AND ROWNUM < 10
SELECT USER_ID,NICKNAME,AVERAGE_RATINGS,MEMBER_SINCE FROM
COMMMAPSDEV.FOOD_USER WHERE NICKNAME LIKE '%t%' AND ROWNUM < 10
```

The returned tuples will show up in the webpage:

Matched Restaurants

| Name | Address | Latitude | Longitude | Rating | Rating_Number | Review | Check In |
|---|---|---|---|---|---|---|---|
| test1 | test1 | 80 | 80 | 2.1 | 12 | Review | Check In |
| test2 | test2 | 1 | 1 | 5 | 1 | Review | Check In |

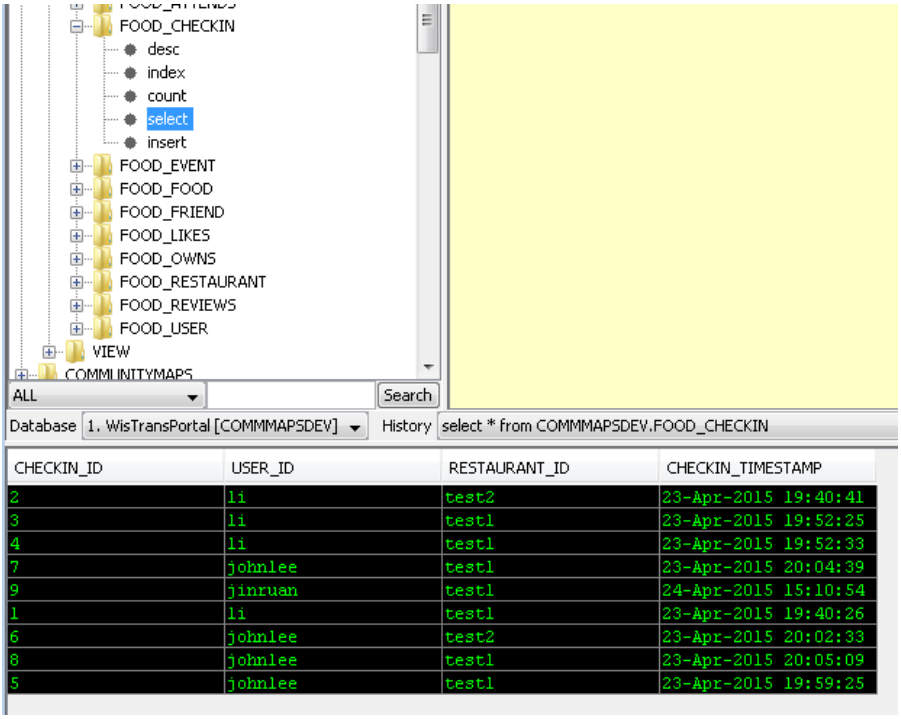
Matched Events

| Name | Restaurant | Time | Event Cost | Description | Attend |
|---|---|---|---|---|---|
| party | test1 | 2015-04-23 16:36:59.0 | | | Attend |

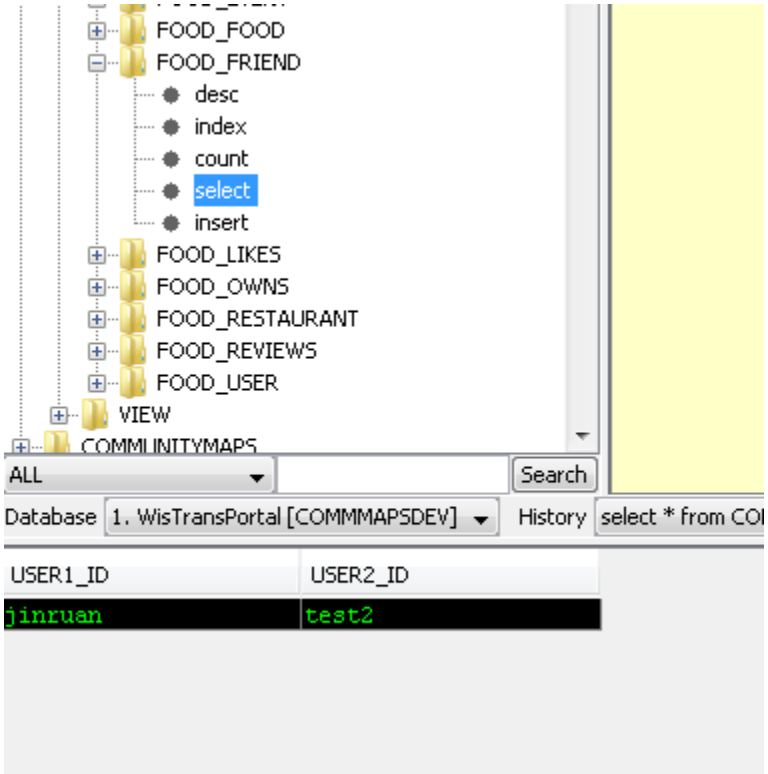
Matched Users

| Nickname | Average Rating | Member_Since | Friend | Unfriend |
|---|---|---|---|---|
| test | | 2015-04-23 16:29:18.0 | Friend | Unfriend |
| test | | 2015-04-23 15:57:32.0 | Friend | Unfriend |

As you can see from the above screenshot, you can check in or review a restaurant, attend a event, friend or unfriend other users.
For the friend, attend and checkin, after you clicking the button, it will add a record into the specific table in the database.

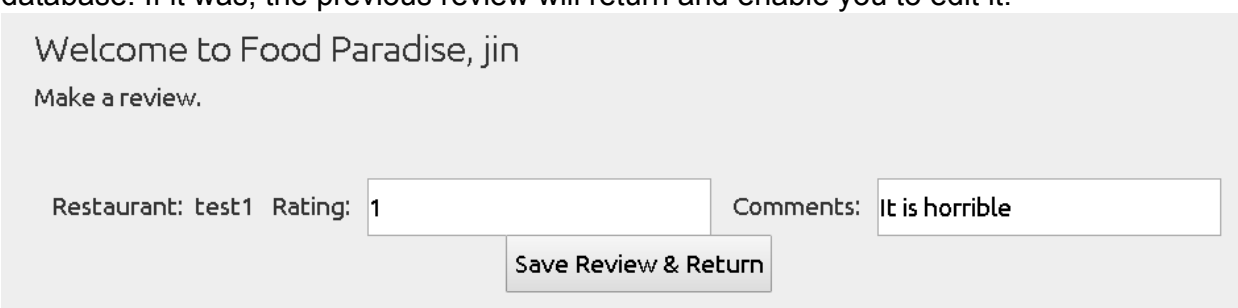

FOOD_CHECKIN
desc
index
count
select
insert
FOOD_EVENT
FOOD_FOOD
FOOD_FRIEND
FOOD_LIKES
FOOD_OWNS
FOOD_RESTAURANT
FOOD_REVIEWS
FOOD_USER
VIEW
COMMUNITYMAPS
ALL
Search
Database 1. WisTransPortal [COMMMAPSDEV]
History select * from COMMMAPSDEV.FOOD_CHECKIN
CHECKIN_ID USER_ID RESTAURANT_ID CHECKIN_TIMESTAMP
2 li test2 23-Apr-2015 19:40:41
3 li test1 23-Apr-2015 19:52:25
4 li test1 23-Apr-2015 19:52:33
7 johnlee test1 23-Apr-2015 20:04:39
9 jinruan test1 24-Apr-2015 15:10:54
1 li test1 23-Apr-2015 19:40:26
6 johnlee test2 23-Apr-2015 20:02:33
8 johnlee test1 23-Apr-2015 20:05:09
5 johnlee test1 23-Apr-2015 19:59:25

desc
index
count
select
insert
FOOD_FOOD
FOOD_FRIEND
FOOD_LIKES
FOOD_OWNS
FOOD_RESTAURANT
FOOD_REVIEWS
FOOD_USER
VIEW
COMMUNITYMAPS
ALL
Search
Database 1. WisTransPortal [COMMMAPSDEV]
History select * from COMMMAPSDEV.FOOD_EVENT
EVENT_ID EVENT_NAME RESTAURANT_ID EVENT_TIMESTAMP EVENT_COST DESCRIPTION
john party test1 23-Apr-2015 16:36:59




- **Review function will query, insert, and update record in the database**

When you click the review button, it will firstly check if the review was already in the database. If it was, the previous review will return and enable you to edit it.




After you change the previous review, it will do the modification query, which will update the original value.

```
UPDATE COMMMAPSDEV.FOOD_REVIEWS SET RATINGS = 2 WHERE RESTAURANT_ID =
'test1' AND USER_ID = 'jinruan'
UPDATE COMMMAPSDEV.FOOD_RESTAURANT SET RESTAURANT_RATING =
'2.076923076923077' WHERE RESTAURANT_ID = 'test1'
```

If the user didn't make a review before, after user submit the review, it will insert a review into the database.

For the unfriend button, it will delete the record that is in the friend table.

For other functions like login, it will do the select query to ask the database to return the records satisfying conditions.