

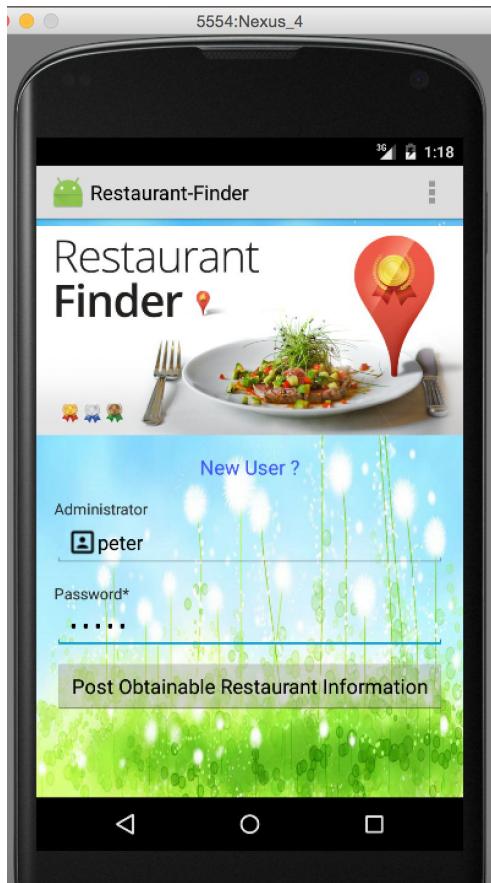
Design Documentation

Step1:

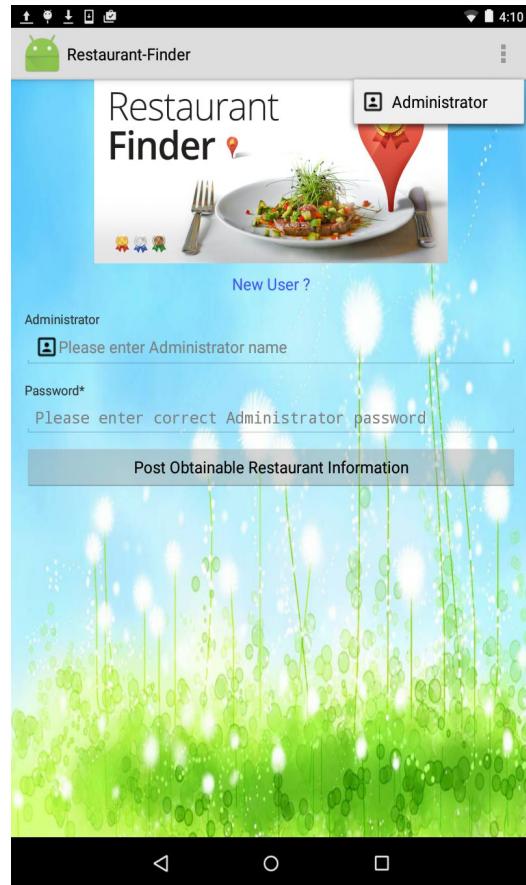
To support two devices, we design two layouts of our application to fix their screen sizes and densities.

One is used for Tablet (Nexus7) and the other is used for phone (Nexus4).

There are two group of screenshots which are obtained by different screen sizes. In the two tests, we give different test values by hand to verify our code correctness.

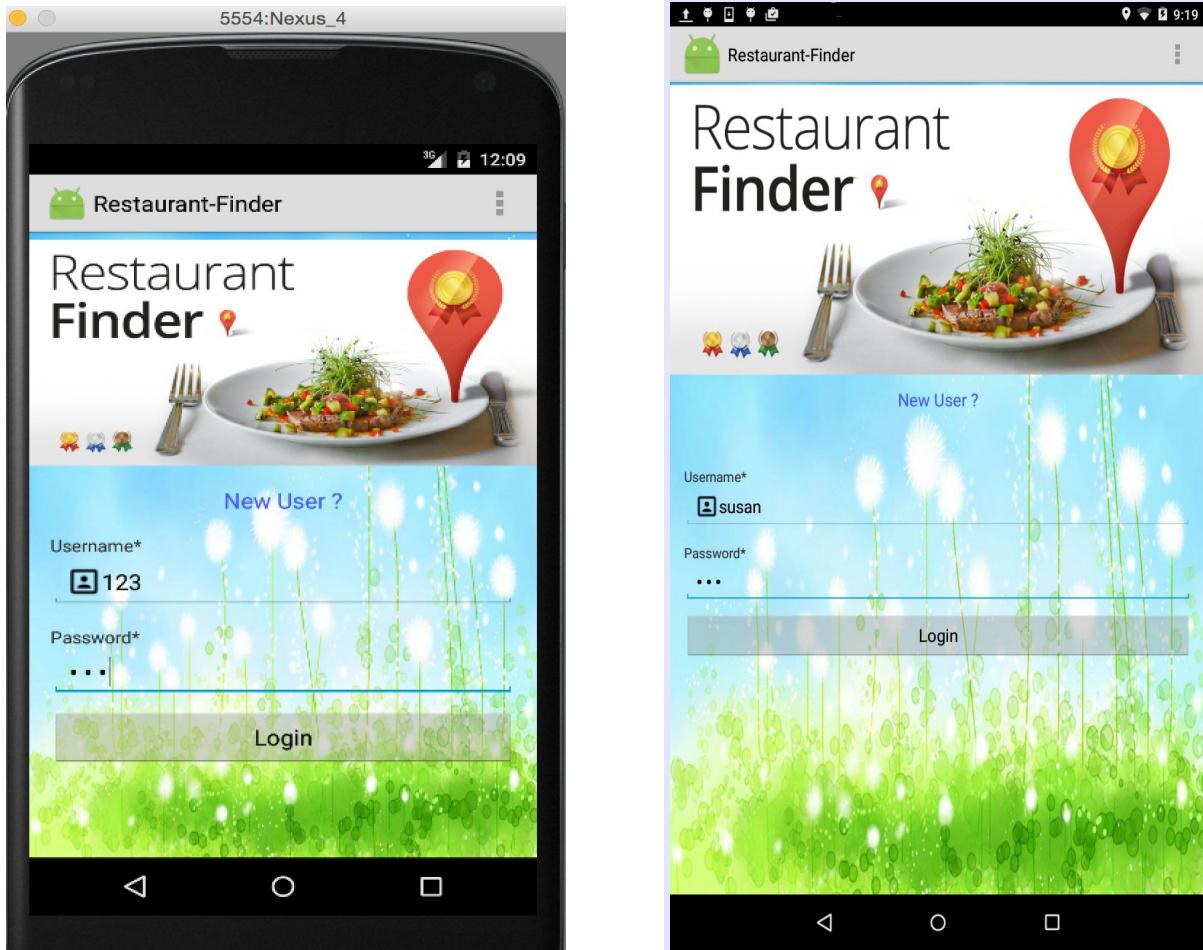


Nexus4

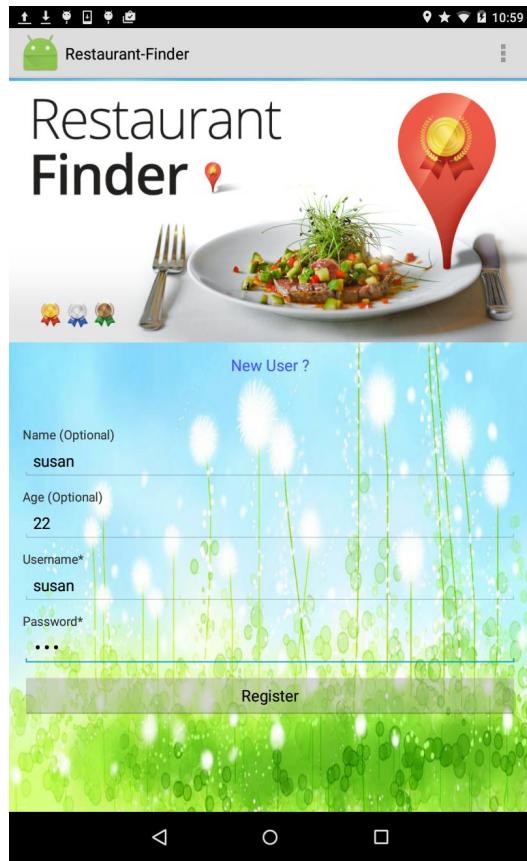
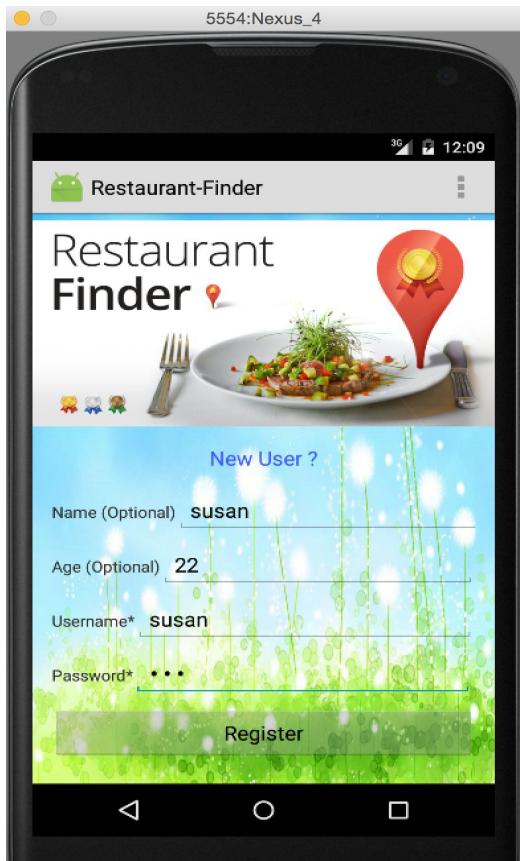


Nexus7

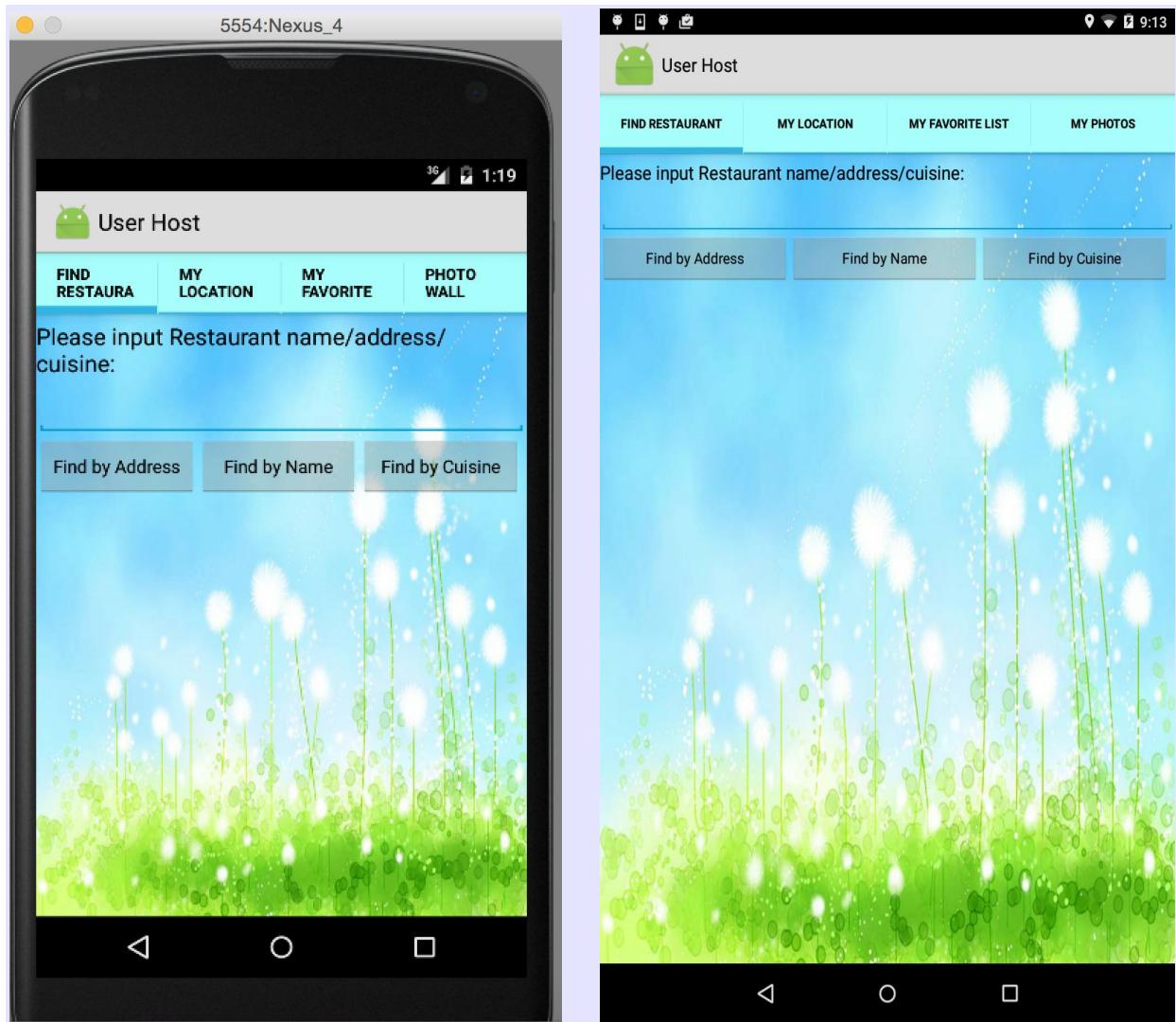
Administrator page: When the app first running, the administrator need to post obtainable restaurant information by his identifications. Click the Post Obtainable Restaurant Information go to the login page.



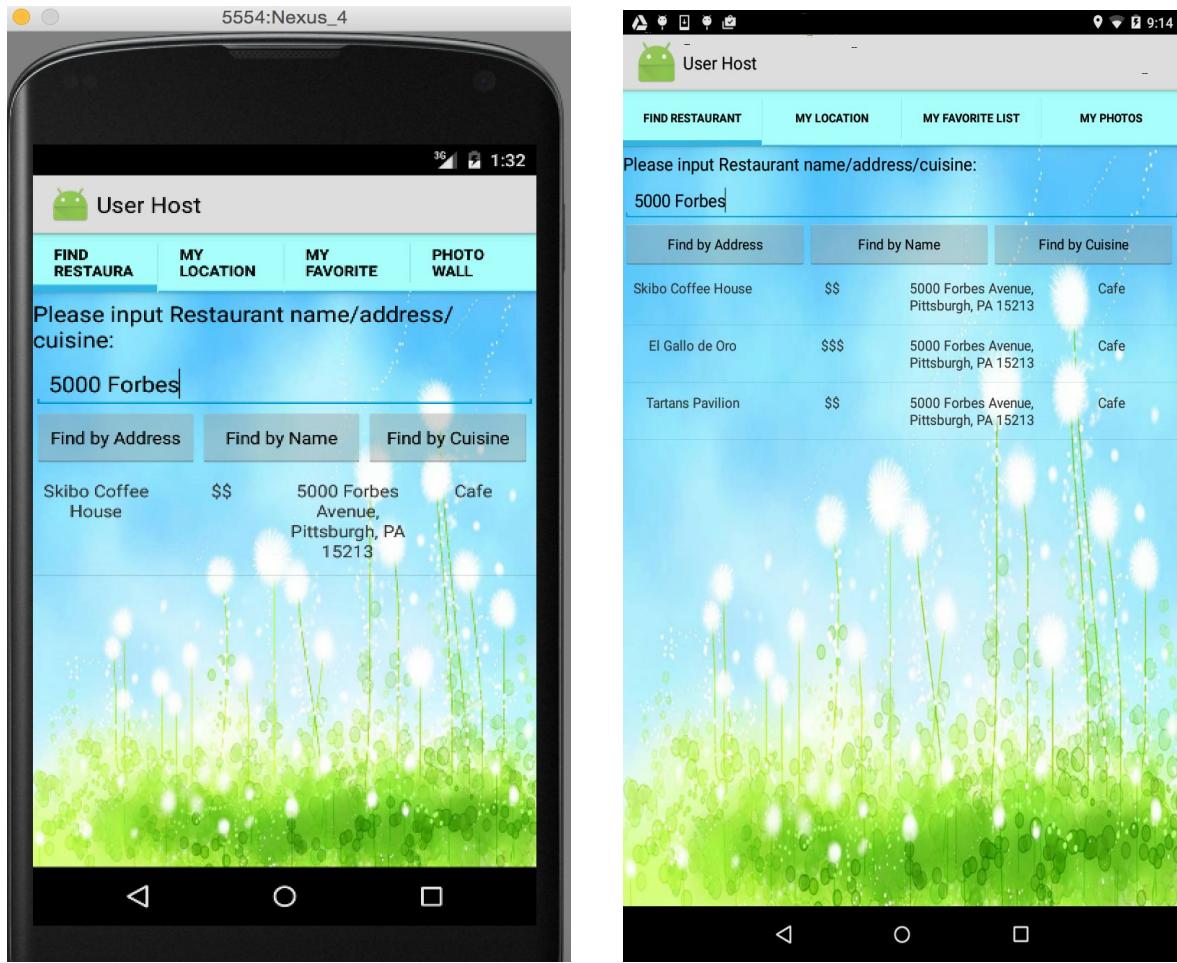
Login page: user enter username and password, and click login button to submit, then goes to UserHost page



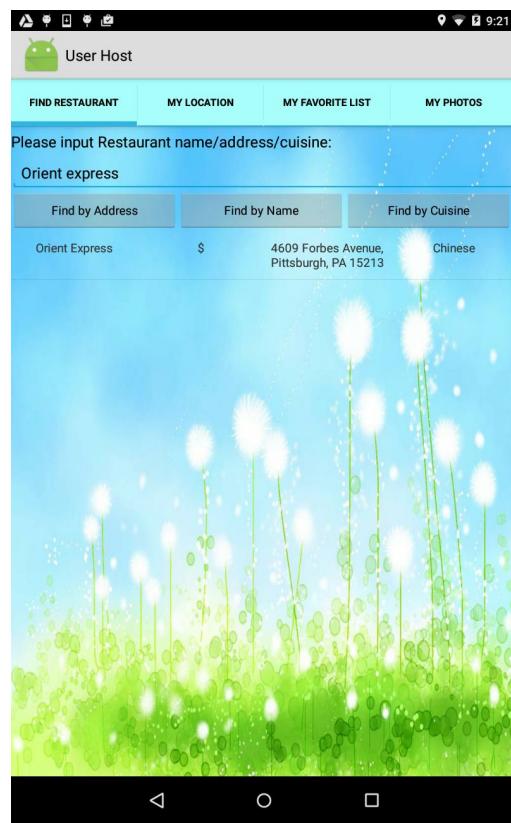
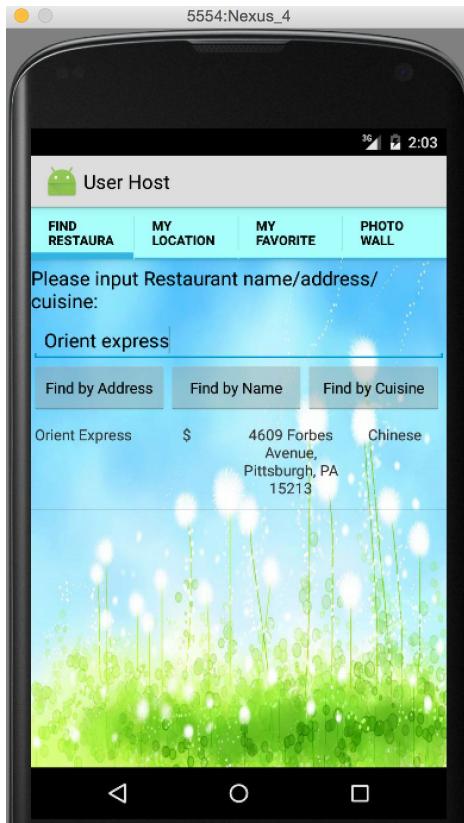
Register page: User clicks New User, then goes to Register page. User enter information to register and then goes to login page by clicking Register button.



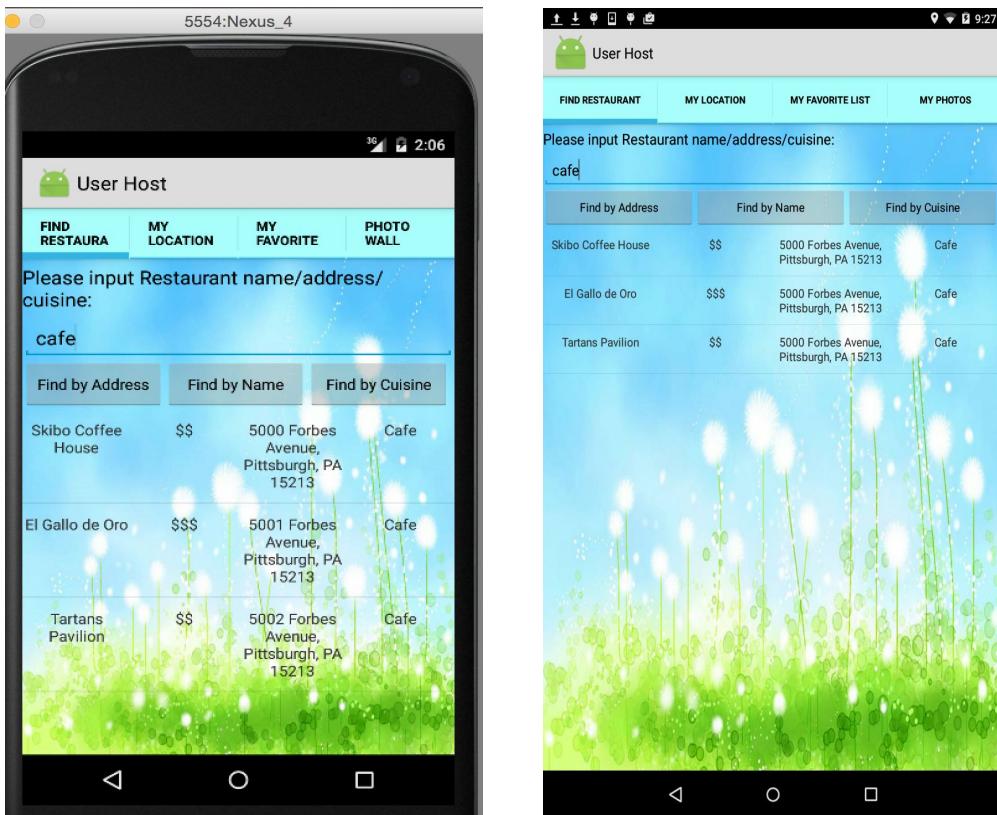
It is the User Host Page, There are mainly four choices which user could make.



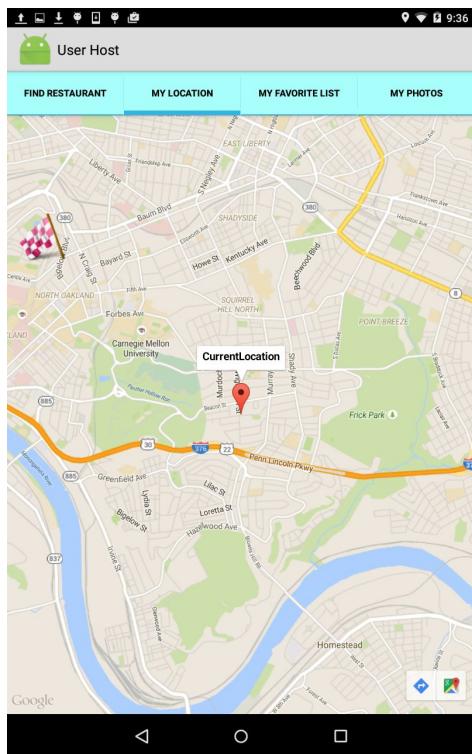
UserHost page: Click Find Restaurant by address. Click ListView item to go to Restaurant Host page.



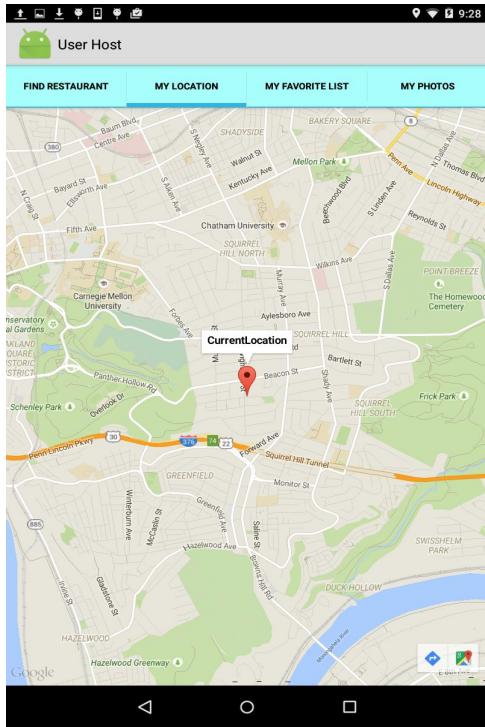
UserHost page: Click Find Restaurant by name.Click ListView item to go to Restaurant Host page.



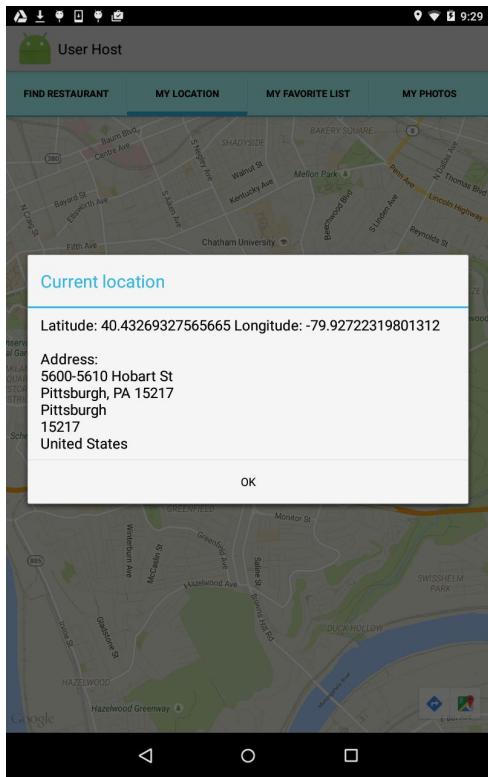
UserHost page: Click Find Restaurant by Cuisine. Click ListView item to go to Restaurant Host page.



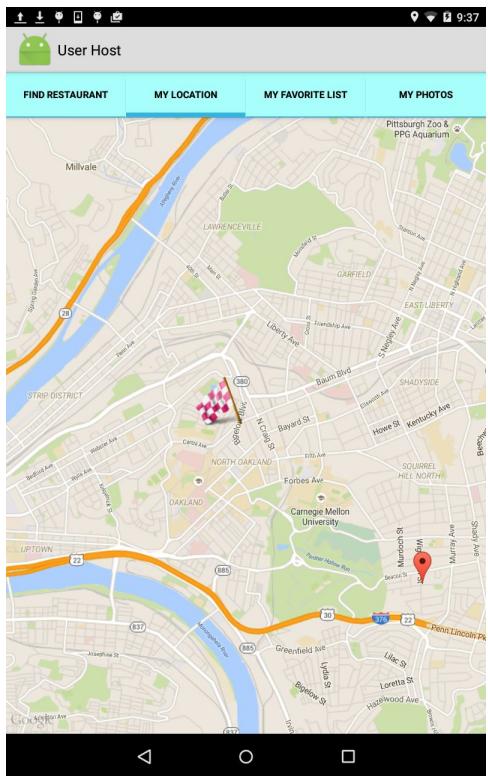
UserHost page: Click My location Tab.



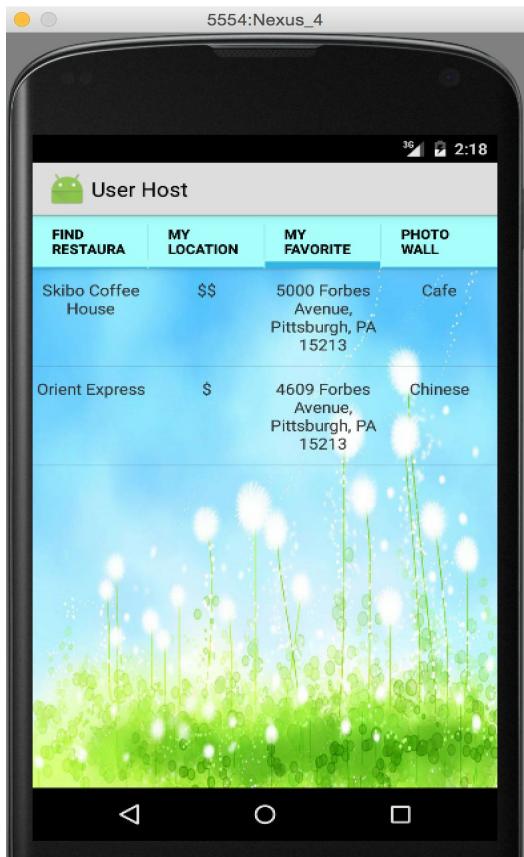
UserHost page: click the user current location marker
Because AVD can not use GPS and not display the map, there is only one screenshot.



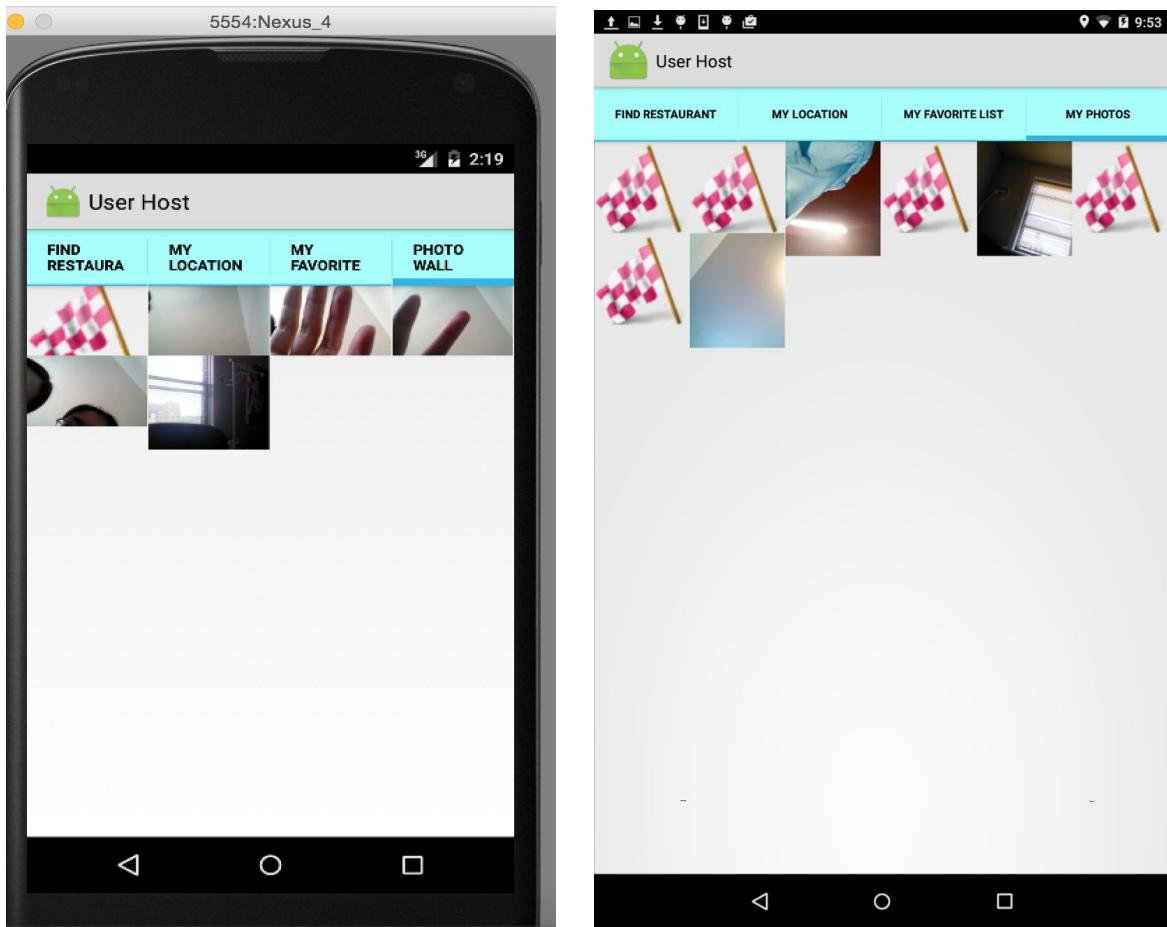
UserHost page: click the current location marker title



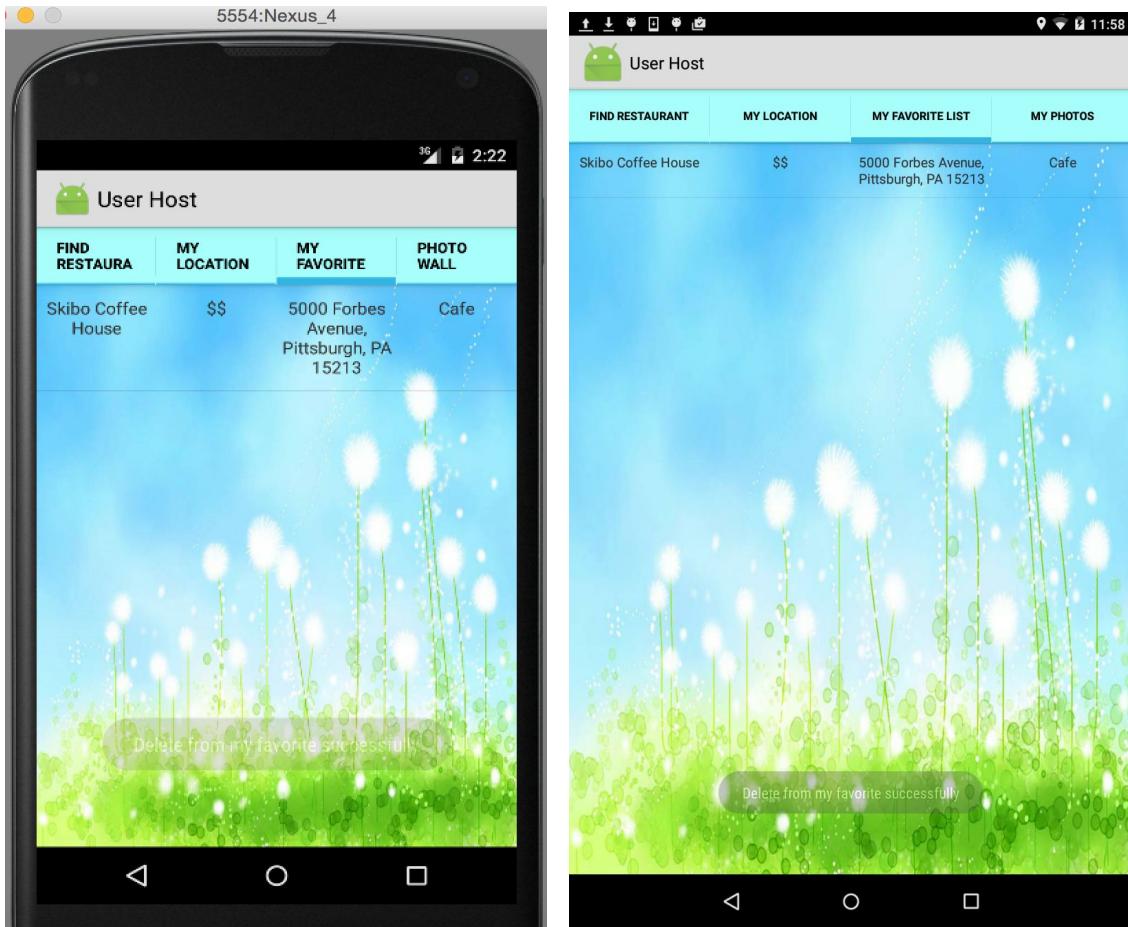
UserHost page: find nearby restaurant which shows with the flag in the google map.



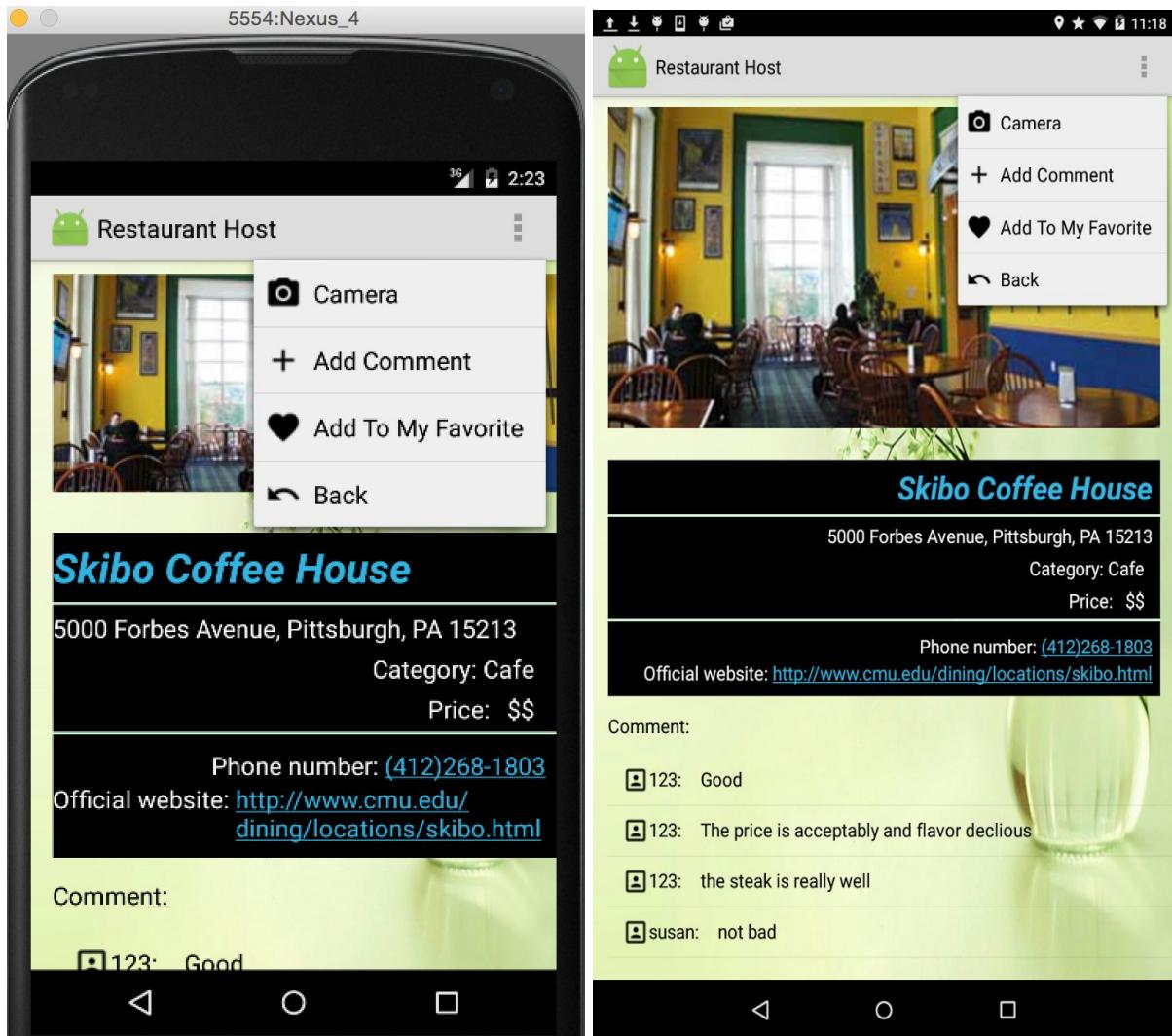
UserHost page: Click My Favorite List, showing all restaurants in user's favorite list. Click ListView item to go to Restaurant Host page.



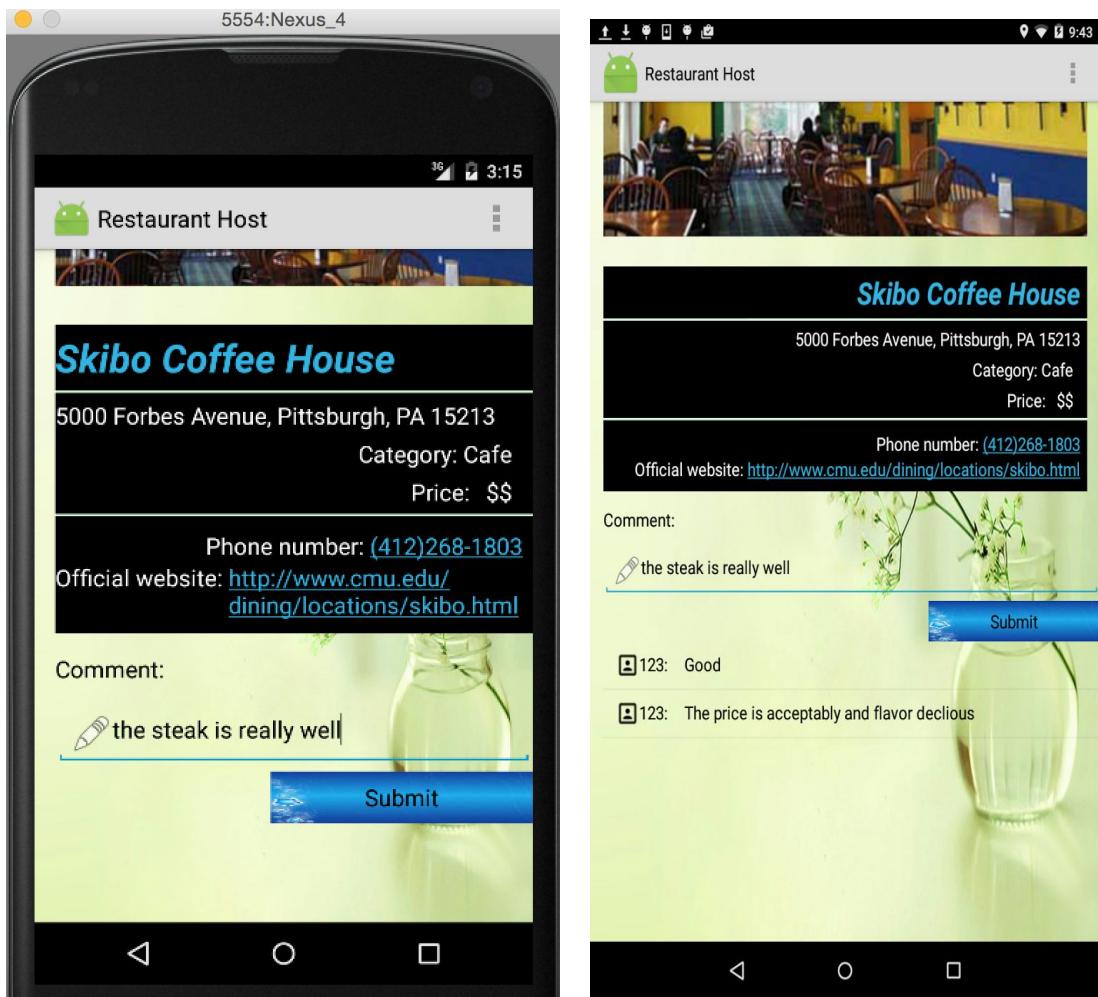
UserHost page: click MY PHOTOS, showing all photos in this app



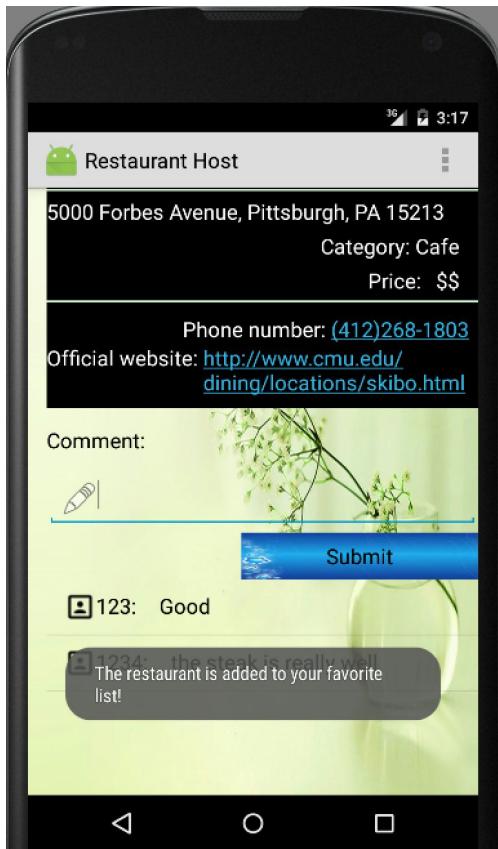
UserHost page:slide a item in the list and delete it from my favorite list



Restaurant Host Page: Click item in the list and then enter Restaurant Host Page



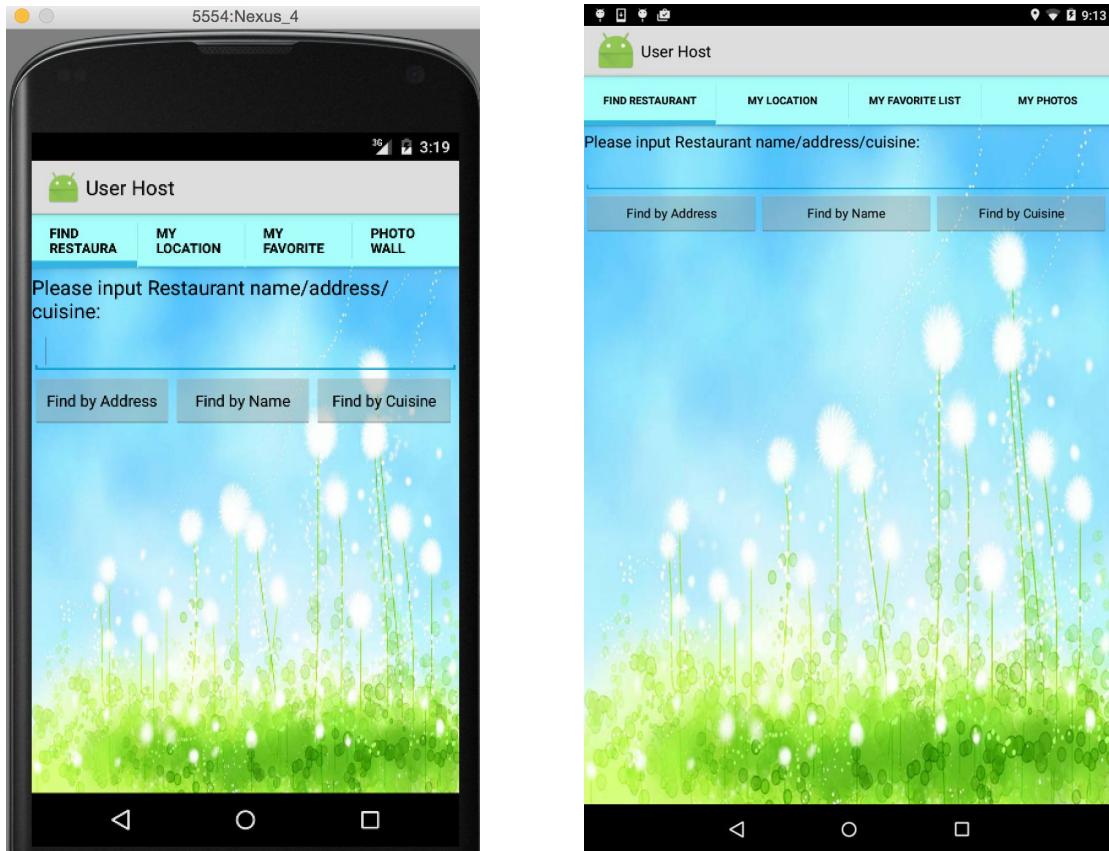
Restaurant Host Page: Click Add Comment action bar to input comments.



Restaurant Host Page: click Add to my favorite list and restaurant is added successfully



Restaurant Host Page:click Camera in the action bar.AVD can not choose this functionality.



Restaurant Host Page:click Back in the action bar.

Step2:

All activities in our project are saved in presentation package, which are used to interact with user input and show output. This package can be seen as user interface package. There are several sub-packages in the presentation package, including adapter, intents, loginPage, restaurantHost, and userHost packages.

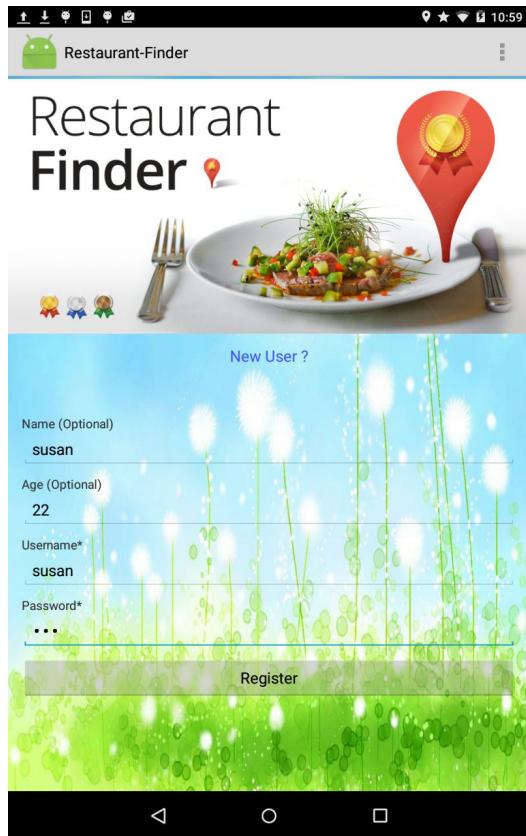
1. loginPage package.

loginPage package includes MainActivity class, LoginFragment class, Register Fragment class, BackHandledFragment, BackHandledInterface and initializeDB fragment.

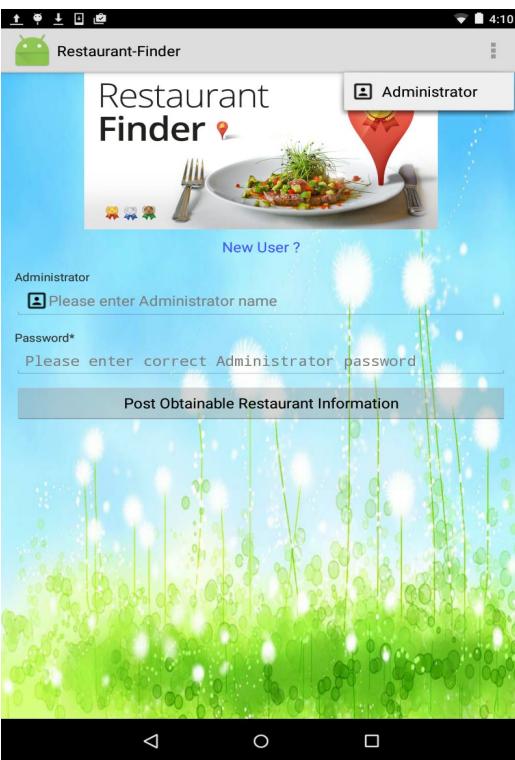
1.1 MainActivity:



1-a



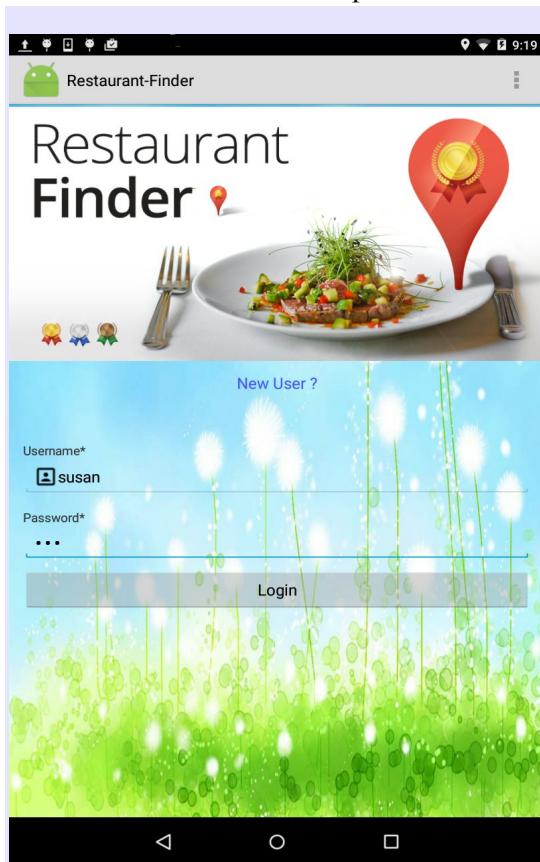
1-b



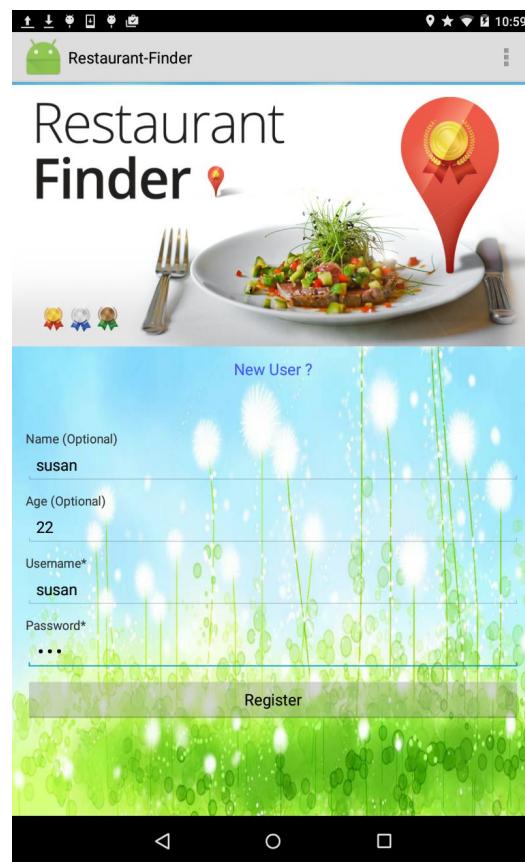
1-c

When a user open the Restaurant_Finder app, MainActivity page is the first page which the user will use. In this activity, it contains two TextViews and one fragment container. The first TextView is just a header which show our app theme. The second TextView is used to register an account when a new user click this TextView. In the part of fragment container, we design two fragmens including LoginFragment and RegisterFragment to implement the login and register functionalities.

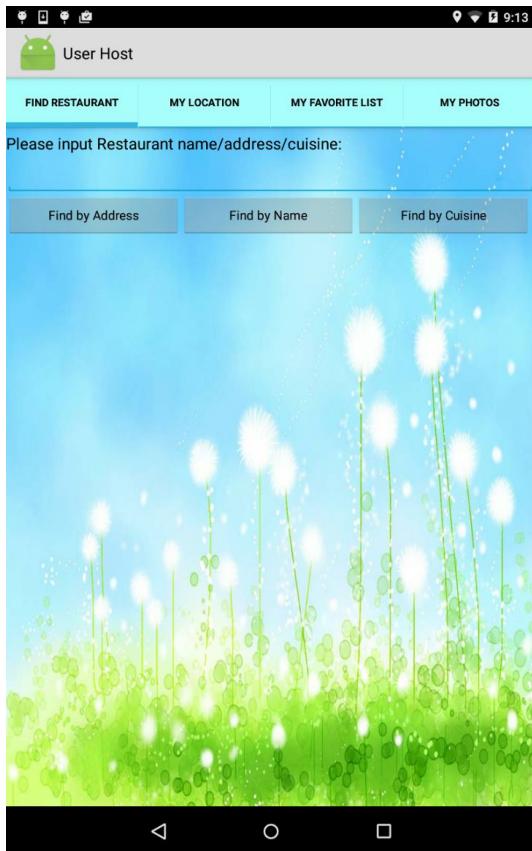
In the MainActivity, the default fragment is LoginFragment. The screenshot of LoginFragment is 1-a. If user already register this app before, he could input the username and password, then click the login button directly to enter userHost page. In this process, the input are username and password. When user click the login button, it will pass the information about username and password to server's database. If the username and password could be found in database, user could login the UserHost page successfully, otherwise, the Loginfragment will be replaced by RegisterFragment, the screenshot of RegisterFragment is 1-b. For a new user, In the 1-b page, user could enter the Name, Age, Username and Password, then click the register button to use this app. When user enter the valid information and click the register button, the information will be sent and stored in server database. At the meantime, this RegisterFragment page will be replaced by LoginFragment. The user interface will jump to 1-a. Now, user could enter the username and password which are used to register this app just now, then click the login button. The user could enter his host page successfully. The screenshot 1-c is the administrator page. Administrator could use his valid identification to post restaurant information for user.



1-c



1-d

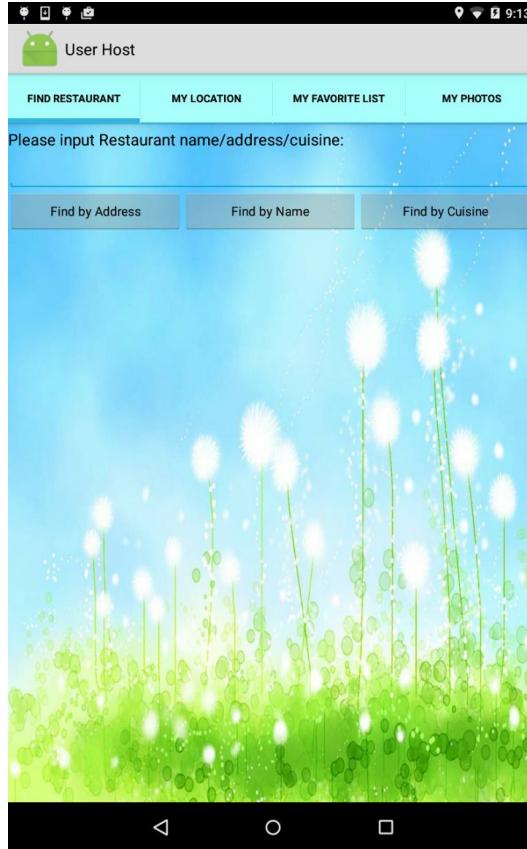


1-e

In the screenshot of 1-c, user has enter his username and password. When he click the login button and the information of username and password are checked by server successfully. He could enter the UserHost page which displays like the screenshot 1-e. In this process, the MainActivity will jump to the UserHostActivity. When a newer user enter the information like the screenshot 1-d and the information is valid, this page will jump to the screenshot of 1-c. Now the new user could login this app normally. When new user enter the correct information, the MainActivity also jump to UserHostActivity.

2. userHost packages:

2.1 UserHostActivity:



Once user has login into the app from login page successfully, he will goes to his user page, which designs in userHost package includes five activities: UserHostActivity, FavoriteListActivity, findRestaurantActivity, findMyLocationActivity and showPhotosActivity. UserHostActivity class defines the user page of this App. It includes four TabHosts that will contain the Tabs. When particular Tab is be selected, it will intents to a specific activity using specific Intent method in intent package. The first tab is navigated to findRestaurantt activity. The second tab is navigated to FindMyLocationactivity. The third tab is navigated to findFavoriteList activity. The fourth tab is navigated to showPhotosActivity. There are FOUR sub-packages in userHost package: findMyLocation package, findRestaurant package , myFavoriteList package and Myphotos package.

2.2 findRestaurant package:

FindRestaurantActivity:



findRestaurant package includes FindRestaurantActivity class. It accepts user input of restaurant name/ address/ cuisine. It outputs a list of available restaurants. There are three buttons in FindRestaurantActivity. There are listening to be clicked. The first button is to search restaurant by user input address. The second is to search by input name. The third is to search by input cuisine. Each of the buttons will call databaseHandler to search Restaurant Table based on three different user's inputs. Database will then return ArrayList<Restaurant>. We will use a ListView in this class to show each restaurant item based on the search result. ListViewAdapter in adapter package will designs a restaurant item showing in the ListView. Clicking the item in the ListView will starts an intent to navigate to the corresponding restaurant page. The intent will also pass a Restaurant object to restaurant page.

2.3 myFavoriteList package:

In this package, it includes one activity named FavoriteListActivity and onefragments named displayMyfavorite.

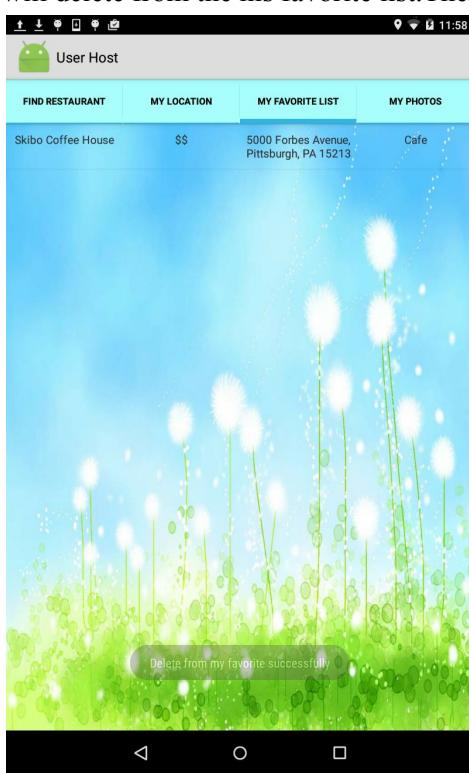
In the FavoriteListActivity, it mainly contains two buttons and one fragment_container. In this activity, the default fragment is displayMyfavorite fragment. When user enter this activity, it will display the list of user's favorite restaurants. If user want to choose one of the restaurant in this list, he could click the item respectively. The page will jump from MyfavoriteList activity (screenshot of 2-3-a)to RestaurantHostActivity(screenshot of 2-3-b).



2-3-a



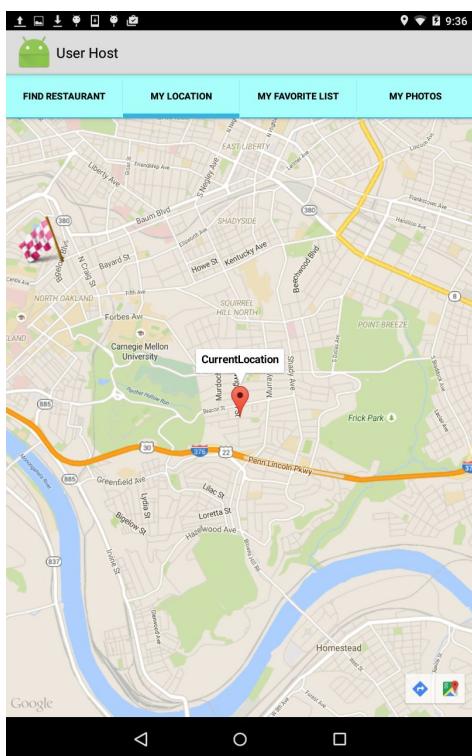
2-3-b



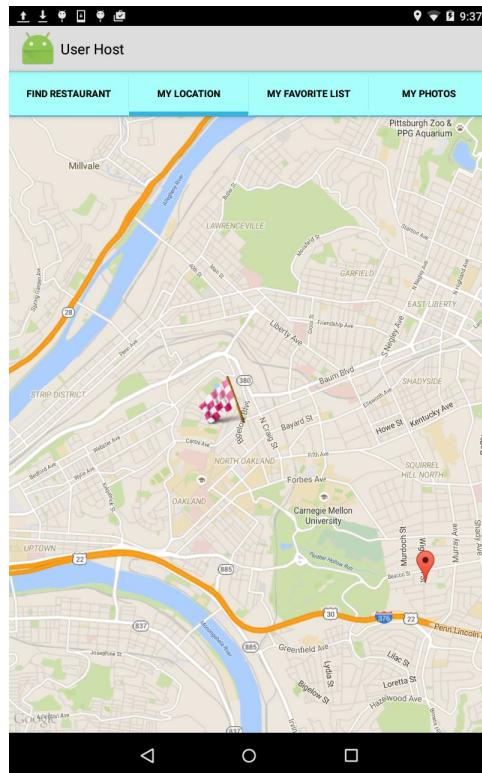
2-3-c

2.4 findMyLocation package:

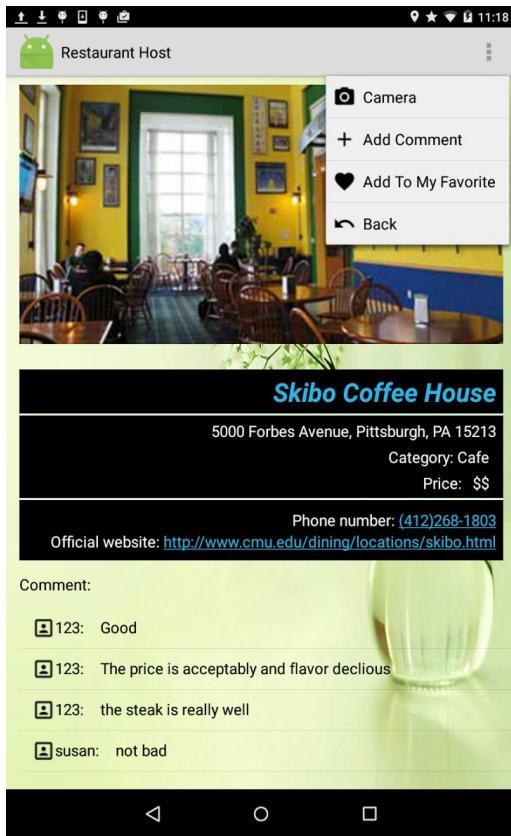
In this package, it includes an activity named MapsActivity and LocationAddress class. If user click tab MYLOCATION, the page will go to MapsActivity and it can display user current location like screenshot 2-4-a. If there are nearby restaurants in the map such as pink flag which represent a nearby restaurant, the page display like 2-4-b. If user click the flag marker title, it could enter the corresponding restaurant page like screenshot 2-4-c.



2-4-a



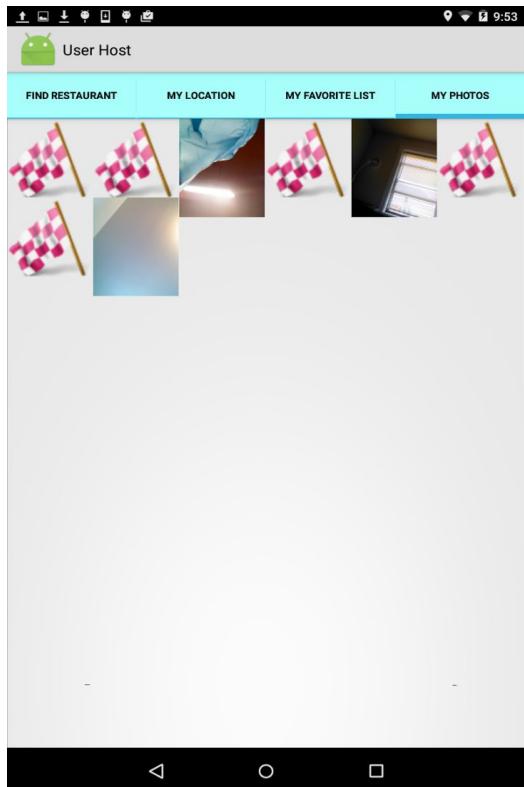
2-4-b



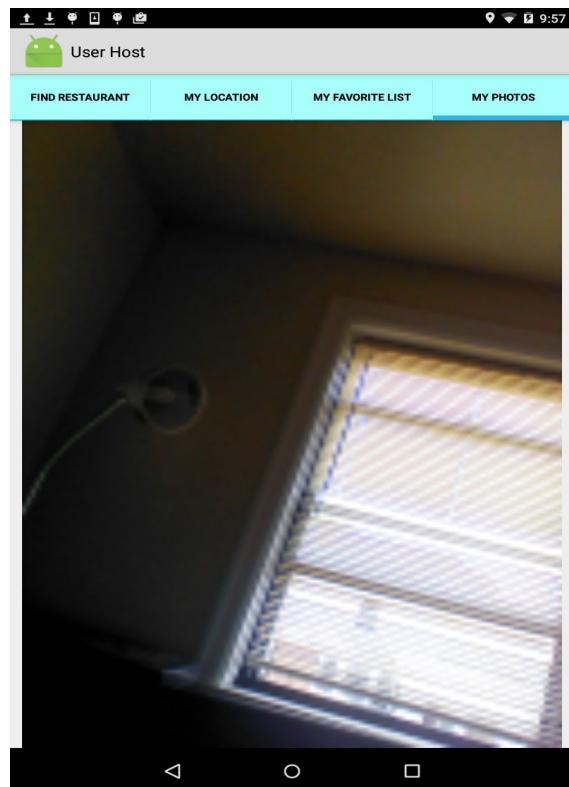
2-4-c

2.5 Myphotos package

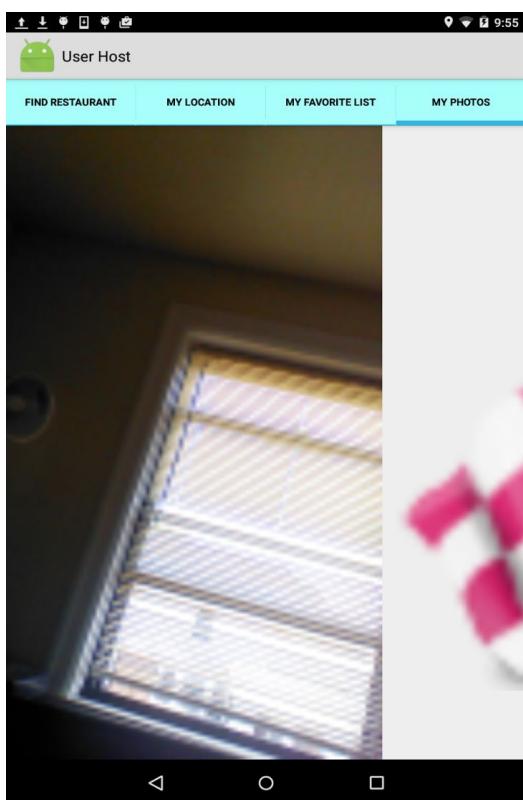
If user click Myphotos tab,it will display all photos in this app like screenshot 2-5-1.User could click one photo randomly and the selected photo will zoom in like screenshot 2-5-2.User also could slide the photo wall like screenshot 2-5-3.If user click the full-screen photo,it will zoom out and page return screenshot 2-5-1.



2-5-1



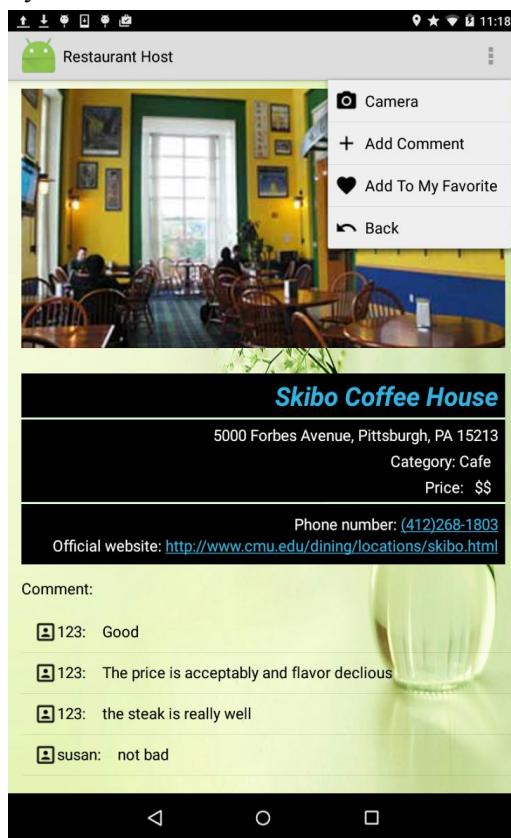
2-5-2



2-5-3

3. restaurantHost package

3.1 RestaurantHostActivity:



RestaurantHostActivity class accepts the Restaurant object from User Host page as its input. It will outputs the name, address, cuisine, phone, website, and price information using TextView. In addition, it will read restaurant ID from object and use it to ask database to return Restaurant comments. Then, outputs these comments using listview in CommentsFragment class. CommentViewAdapter class in Adapter package defines each list item to output comment description. RestaurantHostActivity class contains an action bar. In this action bar overview, it contains four functionalities such as take a photo by camera, add comment, add this restaurant to my favorite list and back to User Host page.

If user find a restaurant which he likes very much and want to add it in his favorite list, user could click the ADD TO MY FAVORITE item, it will show a toast representing user add this restaurant into his favorite list successfully like screenshot 3-1-a.

If user want to add a comment to this restaurant,he could click add comment action bar,enter his comment and click the submit button.Then the page will like screenshot 3-1-b.

If user want to take a photo,he could click camera, the page like screenshot 3-1-c.

If user want return back to User Host page,he could click back item.The page will go to User Host page and show like screenshot 3-1-d.



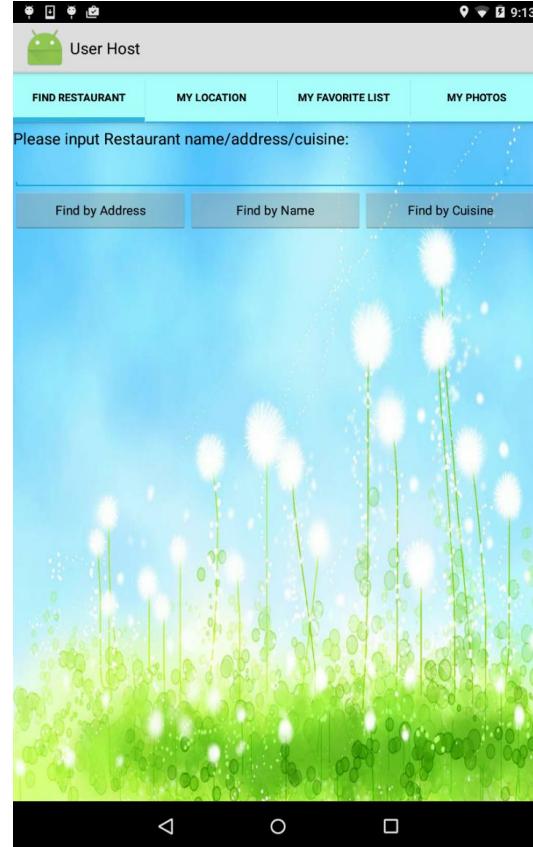
3-1-a



3-1-b



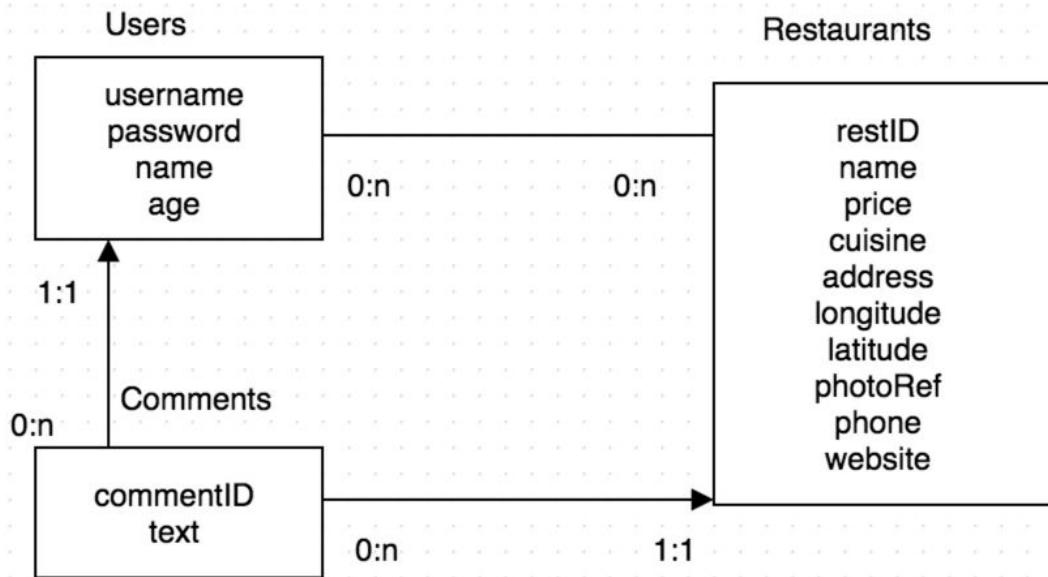
3-1-c



3-1-d

Step 3

3.1 UML diagram



3.2 Table Schema

3.2.1 User Table

User
<code>userName</code> VARCHAR(255) NOT NULL
<code>name</code> VARCHAR(255)
<code>age</code> INTEGER
<code>password</code> VARCHAR(255)
PRIMARY KEY (<code>userName</code>)

3.2.2 Restaurant Table

Restaurant
res_id INTEGER NOT NULL
resName VARCHAR(255)
longitude REAL
latitude REAL
address VARCHAR(255)
price REAL
cuisine VARCHAR(255)
phone VARCHAR(255)
website VARCHAR(255)
PRIMARY KEY (res_id)

3.2.3 Comment

Comment
comment_id INTERGER NOT NULL AUTO_INCREMENT
text VARCHAR(255)
PRIMARY KEY (comment_id)

3.2.4 Favorite_Restaurant

Favorite_Restaurant
userName VARCHAR(255) NOT NULL
res_id INTEGER NOT NULL
FOREIGN KEY (userName) REFERENCES User(userName) ON DELETE CASCADE
FOREIGN KEY (res_id) REFERENCES Restaurant(res_id) ON DELETE CASCADE
PRIMARY KEY (userName, res_id)

3.2.5 Restaurant_Comment

Restaurant_Comment
comment_id INTEGER NOT NULL
res_id INTEGER NOT NULL
FOREIGN KEY (comment_id) REFERENCES Comment(comment_id) ON DELETE CASCADE
FOREIGN KEY (res_id) REFERENCES Restaurant(res_id) ON DELETE CASCADE
PRIMARY KEY (comment_id)

3.3 class diagram for SQL query

Create	Read
<ul style="list-style-type: none"> - createTableUsers(): String - createTableComments(): String - createTableRestaurants(): String - createTableUserComments(): String - createTableRestaurantComments(): String - createTableUserRestaurant(): String 	<ul style="list-style-type: none"> - readTableUsers(user): User - readTableComments(Comment): Comment - readTableRestaurants(Restaurant): Restaurant - readTableUserComments(Integer[]): Integer[] - readTableRestaurantComments(Integer[]): Integer[] - readTableUserRestaurant(Integer[]): Integer[]
Update	Delete
<ul style="list-style-type: none"> - updateTableUsers(User): String - updateTableComments(Comment): String - updateTableRestaurants(Restaurant): String - updateTableUserComments(Integer[]): String - updateTableRestaurantComments(Integer[]): String - updateTableUserRestaurant(Integer[]): String 	<ul style="list-style-type: none"> - deleteTableUsers(User): String - deleteTableComments(Comment): String - deleteTableRestaurants(Restaurant): String - deleteTableUserComments(Integer[]): String - deleteTableRestaurantComments(Integer[]): String - deleteTableUserRestaurant(Integer[]): String

Note: in update and delete operation, the return string is either “success” or “fail” to tell the user whether the operation is executed successfully.

Step4:

1 entities package:

There are three models we are going to use in this project: Restaurant, User and Comments. These three objects will be utilized in presentation tier and interact with database.

Comment class: Include text variables, and all getter and setter methods about the variable.

Restaurant class: Include name, address, price, cuisine, phone, website, longitude and latitude variables, and all getter and setter methods about the variables.

User class: Include username, password, name and age variables, and all getter and setter methods about the variables.

2 exception package:

2.1 DuplicatedUsernameException: it may happens in MainActivity. When a guest want to register to be a new user but input a username that already exists in database, then throw this exception.

2.2 UsernameNotFoundException: it may happens in MainActivity. When a user inputs a wrong username that cannot be found in database, then throw this exception.

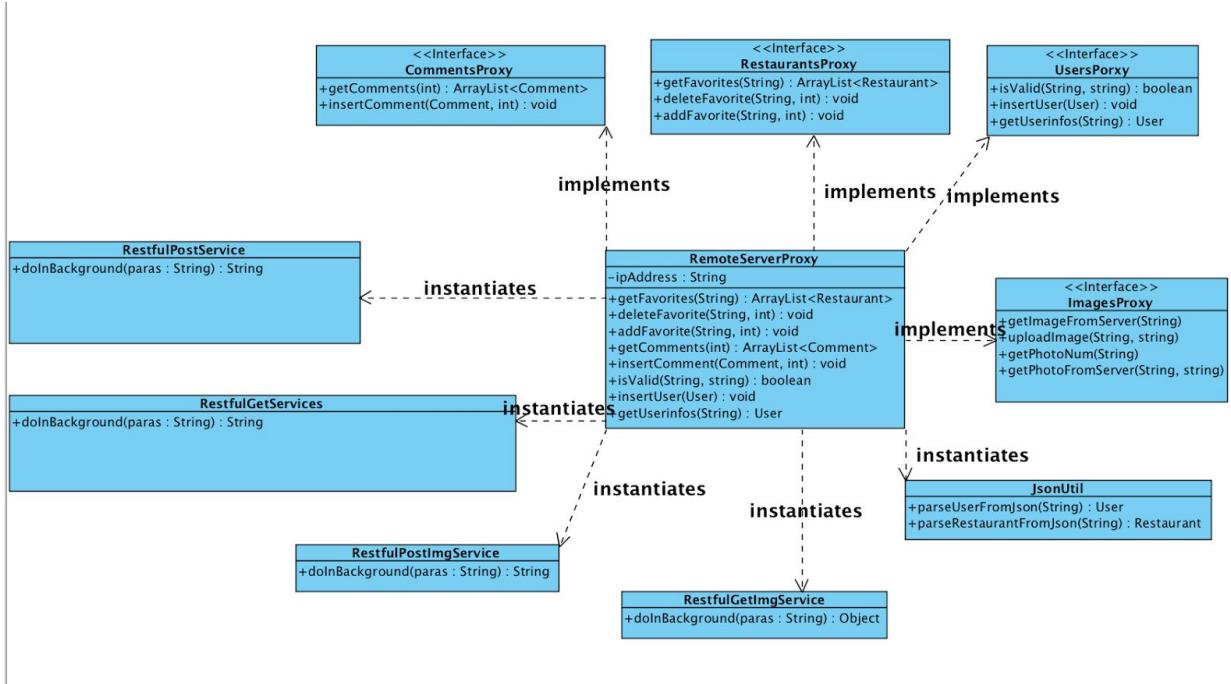
2.3 NullInputException: it may happens in all Activities that need to read user's input. When user does not provide a required input, then throw this exception.

2.4 ExceptionHandler: this includes several fix methods to used to fix the corresponding exceptions. To fix DuplicatedUsernameException, the MainActivity will shows RegisterFragment again and let user retype a username. To fix UsernameNotFoundException, the MainActivity will shows LoginFragment again and let user retype a username. To fix NullInputException, the current Activities will make a Toast to print error message and stop the intent to transfer to start another activity or fragment.

Step5:

5.1 Client

5.1.1 Class Diagram



5.1.2 Local Service

As for local service, we provide GPS, Accelerometer and Camera, user can get their location through GPS service, also they can get a restaurant randomly by shaking the tablet. What is more, users can access camera and take photo, these are local services provided.

5.1.3 Remote Service

5.1.3.1 Category of Services

To reduce the CPU cost on mobile phone, as long as considering the limited storage in mobile phone, we decided put our data in the server which enables the consistency within different users.

Four are main category of data service provided by server, aka user account service, comments service, restaurants service and Images service. All of these four are mainly about data and image manipulation.

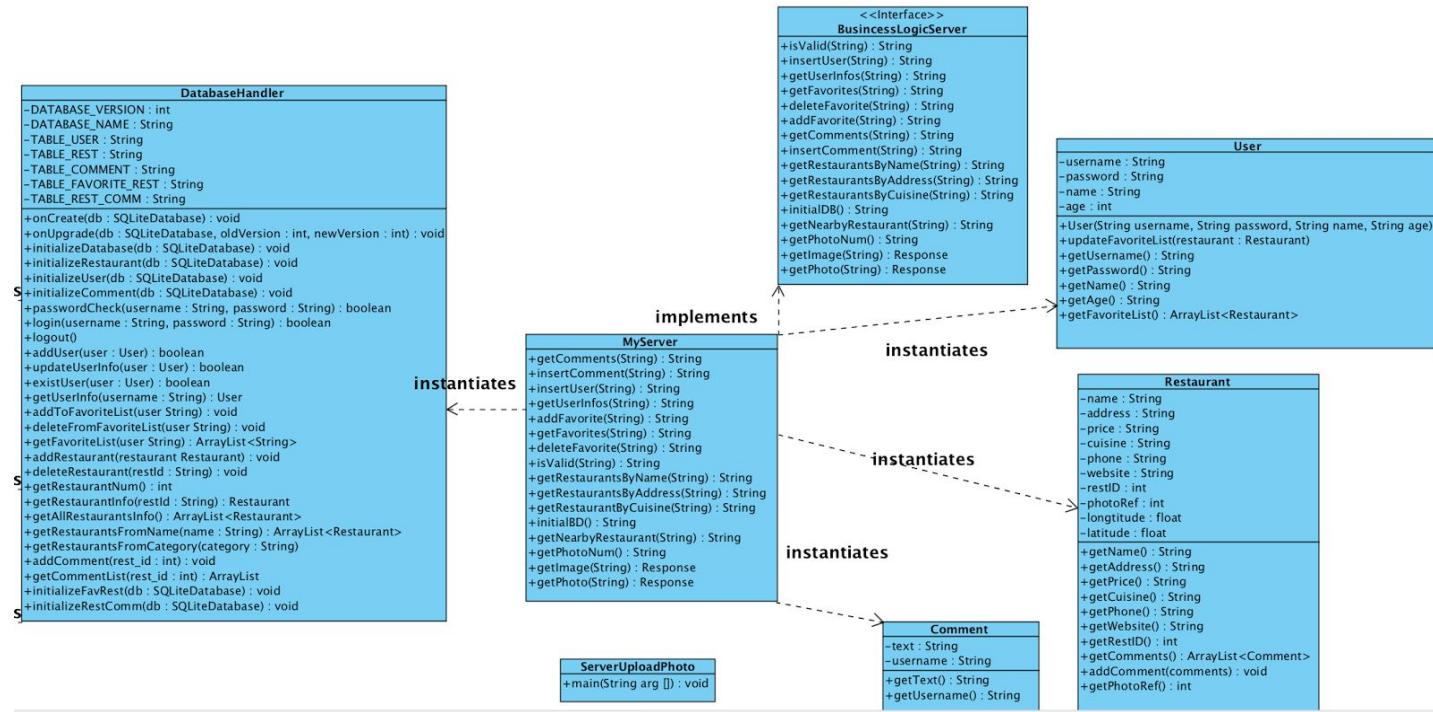
5.1.3.2 Server Proxy

We build a server proxy in client to enable client access server easily. The concrete service provided are: get comments of a specific restaurant; push a user's comment of a specific restaurant to the server which can be viewed by all the users; pull the list of a specific user's favorite restaurant; pull the list of available restaurants based on user's search input; when a user try to login, check him is valid user or not; for a specific user, fetch all his/her information including name, password, list of favorites and all photos; when a new user register, his/her data also will be stored in the server; and when a user upload a photo, the photo will be stored in the server as well.

The server proxy is able to provide all these remote services through calling RestfulGetService, RestfulPostService, RestfulGetImgService and RestfulPostImgService.

5.2 Server:

5.2.1 class diagram:



5.2.2 Business logic Tier

The business logic mainly completed by three components, the first is **BusinessLogicService**, the second is **ServerUploadPhoto** and the other one is the instances in **models** package.

5.2.2.1 BusinessLogicService

This service tries to operate differently according to the query of client, it can get comments of a specific restaurant; update a user's comment of a specific restaurant server; get the list of a specific user's favorite restaurant; check whether a user is valid or not given his/her name and password; for a specific user, fetch all his/her information including name, password, list of favorites; when a new user registers, store his/her data to the database. And all these services will be operated on the corresponding models which are user, account and restaurant.

5.2.2.2 ServerUploadPhoto

This service is opening a socket and listening a port from client. It can get InputStream from client, creates a file output stream to write to the input image file with a name of currentTimeMillis, read the file byte by byte and finally save the image into a folder.

5.2.2.3 Models

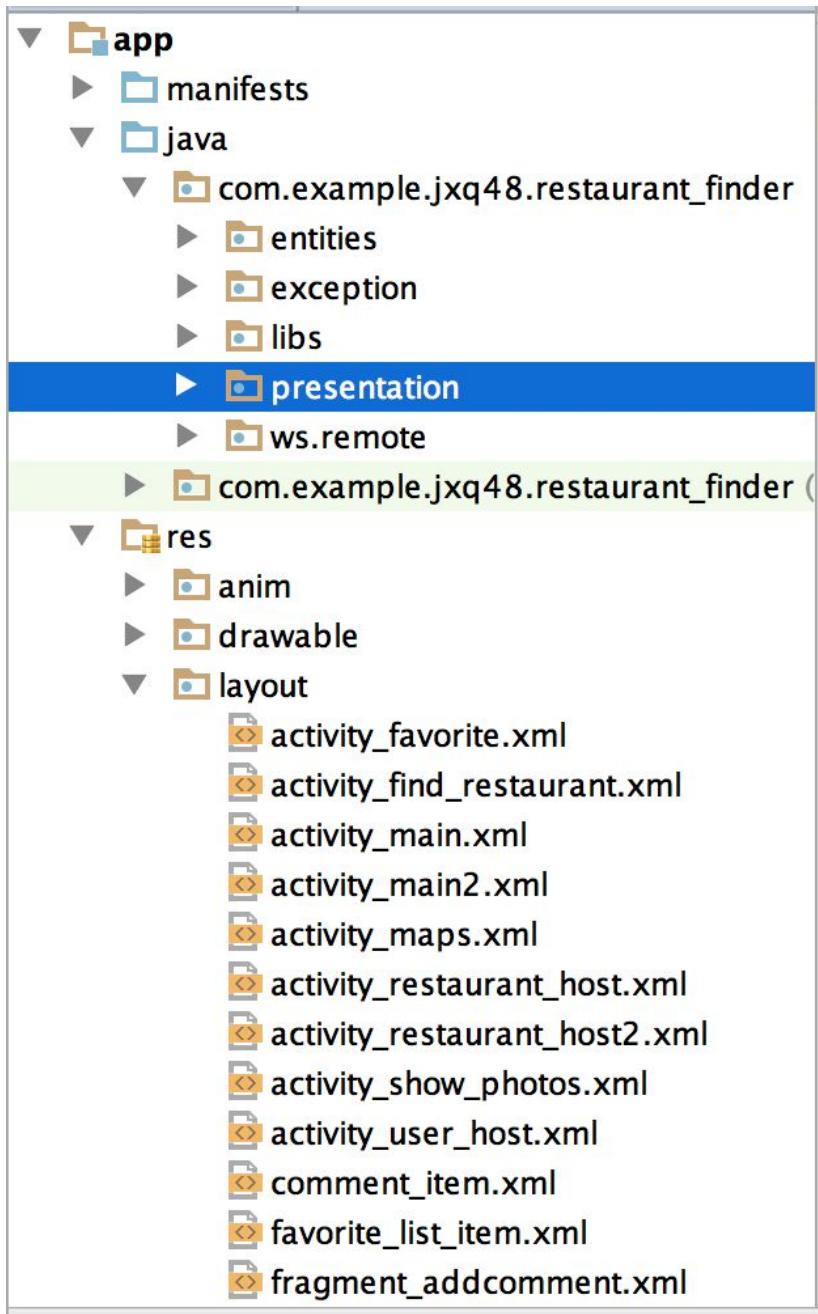
The models contain user, account and restaurant. All the data manipulation will be reflected on the specific instance of model. When data is returned back to client, a instance of corresponding model will be returned (either in JSON format or through `toString()` method).

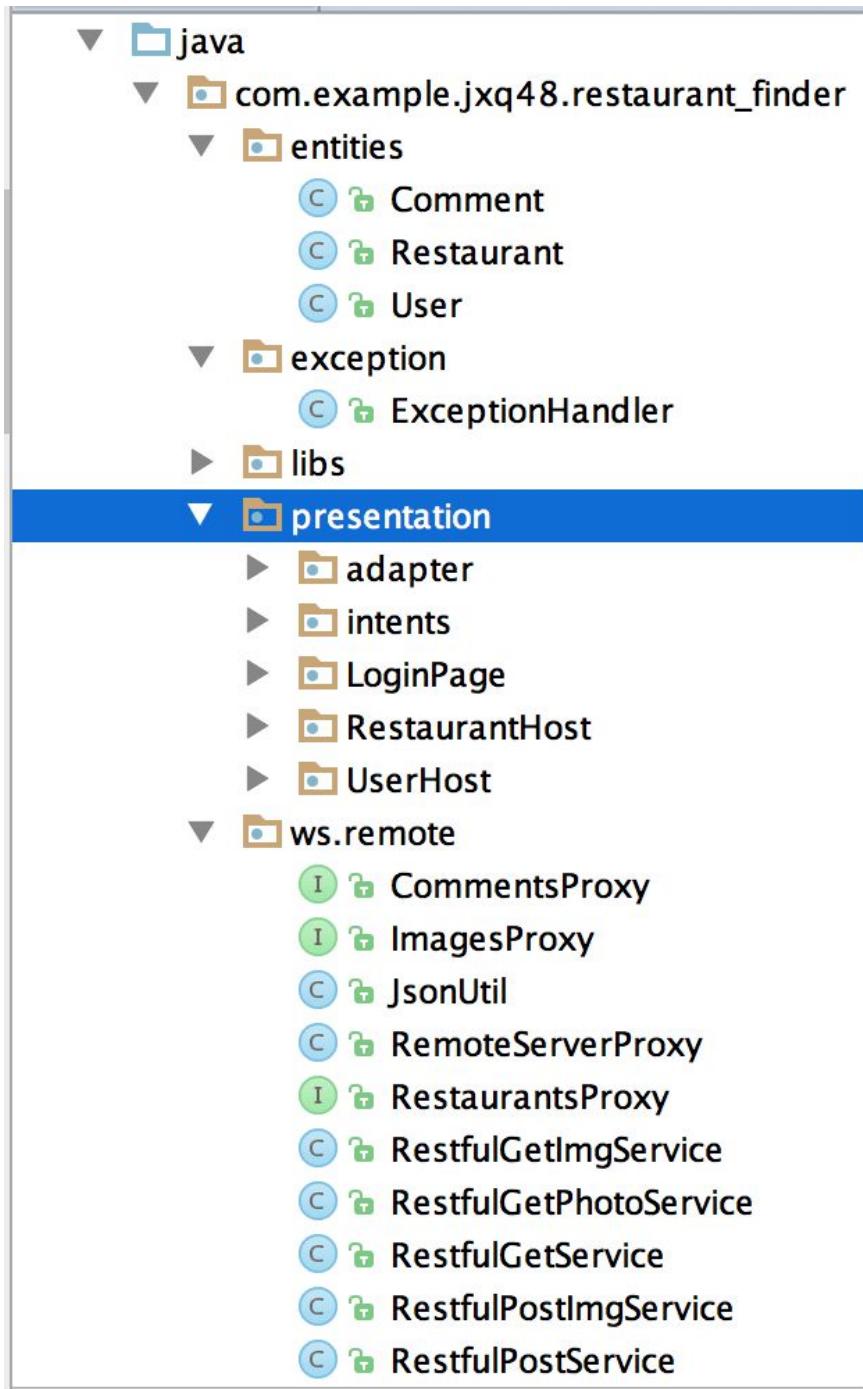
5.2.4 Storage Tier

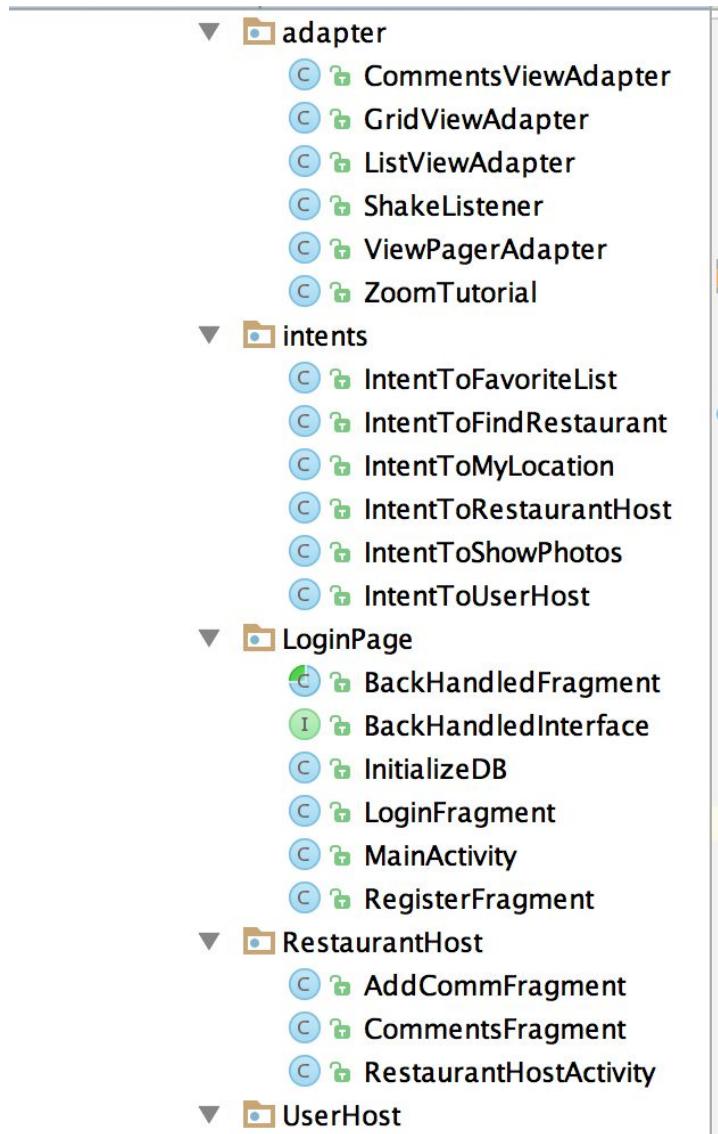
Use Mysql Database to store users and restaurants data, there are six tables: Users Table, Comments Table, Restaurants Table. These three are entity tables, Favorite_restaurant table, user_comment table, restaurant_comment table, these three are relationship tables. The detail description of table schema is in.

Project Class Design:

This is our project directory. We design presentation tier in package presentation. The package DBLayout is a content provider. The package entities is application tier. The package ws is used to organize our integration tier.

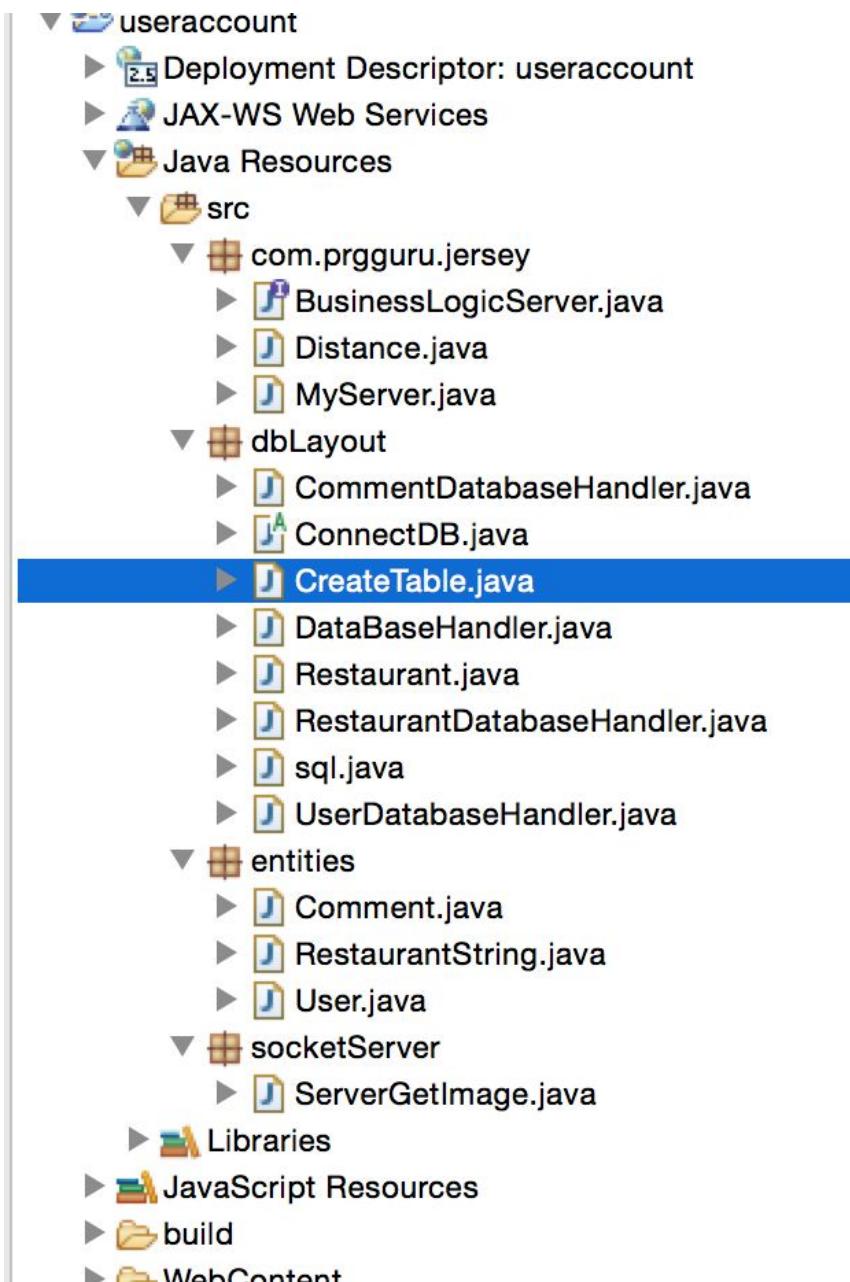






5-2-3

The screenshots 5-2-1, 5-2-2, 5-2-3 are client and UI project directory.

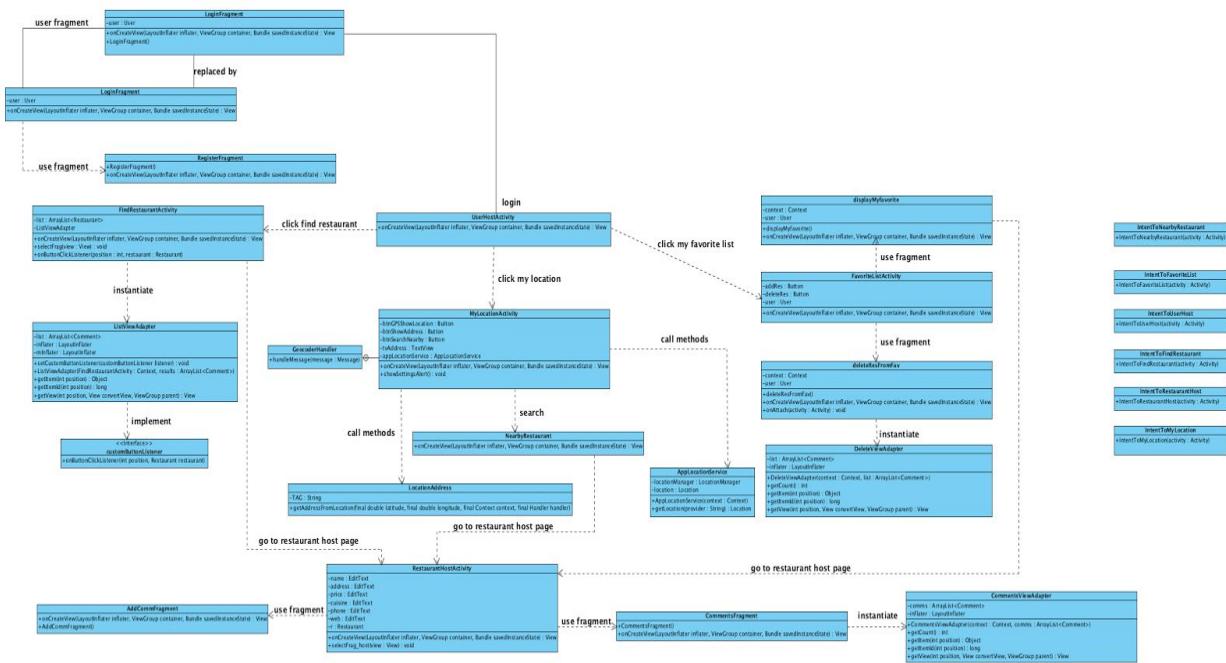


5-2-4

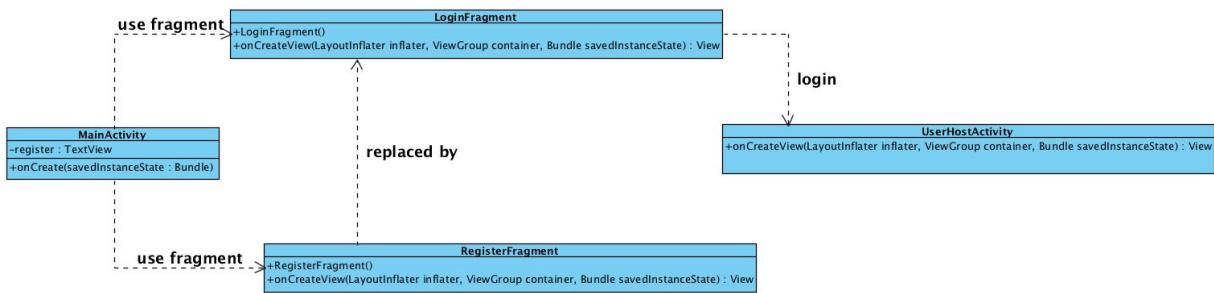
The Screenshot 5-2-4 is the remote server directory in Eclipse.

Class diagram:

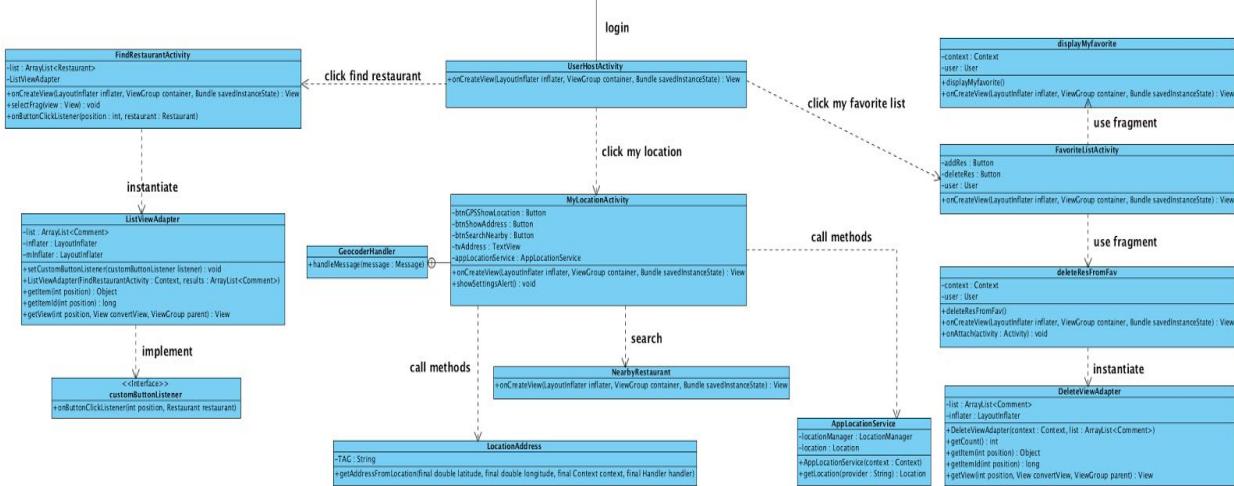
Client:



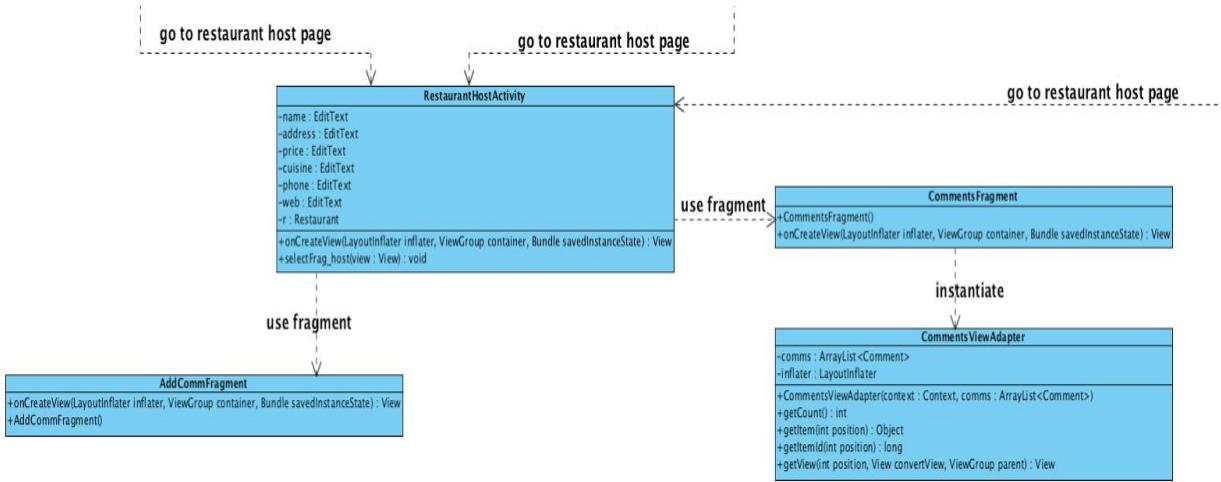
LoginPage:



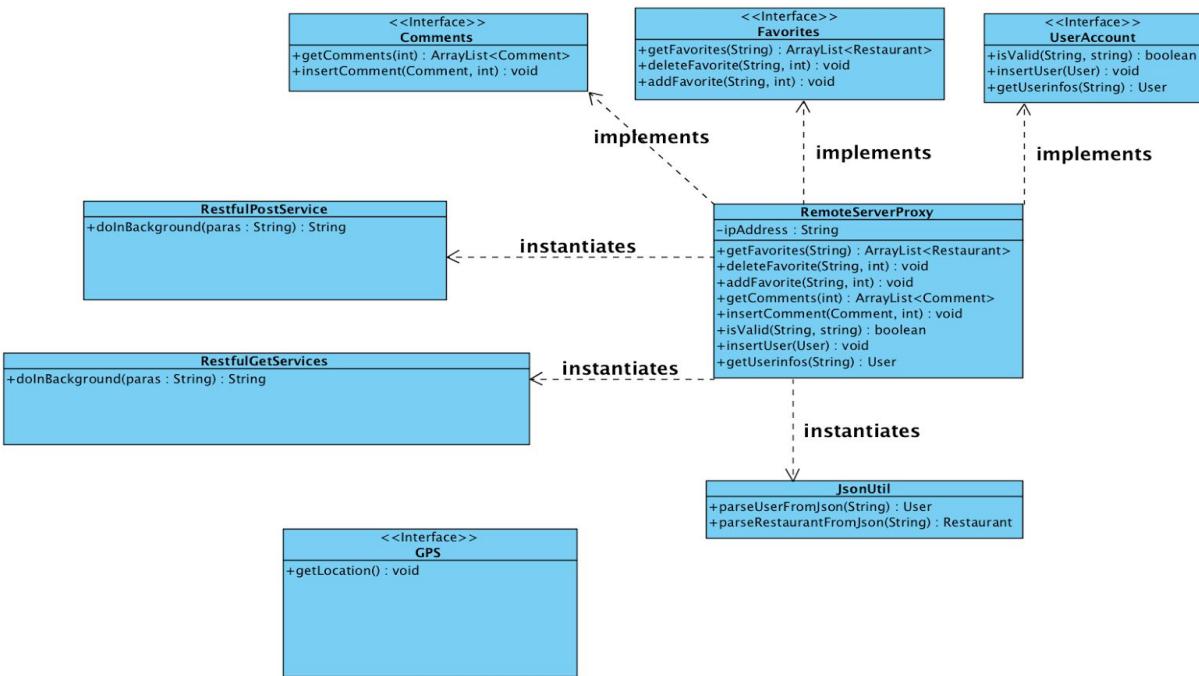
UserHost:



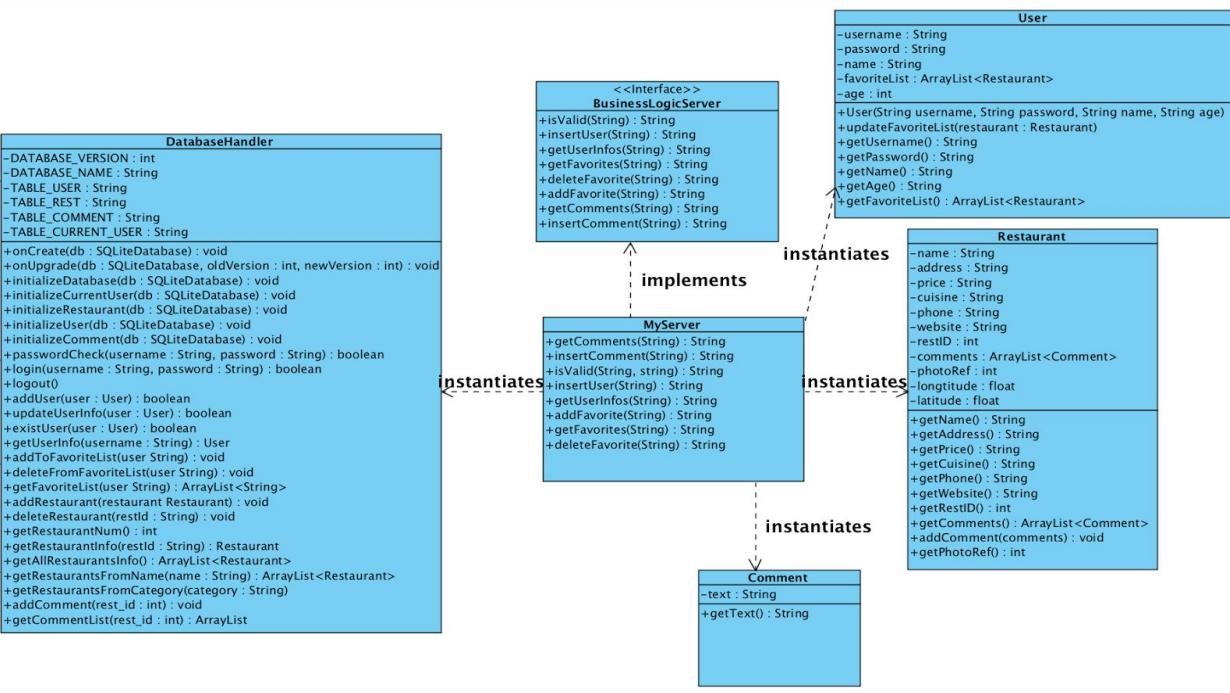
RestaurantHost:



Remote Server Proxy:



Server:



Database:

