FOOD MENU PHONE APPLICATION

A Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Shengyu Li

December 2009

ABSTRACT

FOOD MENU PHONE APPLICATION

by Shengyu Li

This project is about implementing a food menu application for users to search and upload food information by using a mobile phone. The project is scheduled to be finished in two semesters. This semester, I wrote two testing programs which are using Google Maps from a phone, and obtaining a list of zip codes which geographic short distance to the given zip code. These two test programs will be helpful for implementing phone location next semester. I also designed and constructed a food database in this semester, this will help to set up a server contain a database for phone to access next semester.

Table of Contents

List of Figures

1.0 INTRODUCTION

Social networking has become increasingly popular especially in today's mobile phones. Cell phone technology nowadays allows innovative applications to be implemented in many forms. So in order to help people search for specific type of food with their smart phone, my graduate project is to design and implement a food menu search phone application.

Sometimes people crave specific food, but don't know which restaurant provides it. By using our food search phone application, users can obtain a list of restaurants' information such as restaurants' location, open/close hours, phone number, item's price, rating, so on and so forth.

There are other similar applications to our food search program like Google Maps and Yelp. In Google Maps, user can search many types of information (including food and restaurant). However, the result of Google Maps only contains information (restaurant's name, address, phone number, direction to the restaurant) in brief. More detail about the food item would be necessary in order to bring users' passion to visit the restaurant. Alternatively, Yelp is geared towards searching for restaurant titles for reviews and rating.

With my food menu phone application, users will be able to input the food information whenever they want to. Our user friendly phone application interface encourages users to enlarge our system as long as the phone has internet data services.

The main functionalities are search and upload food from the mobile phone. In this case, when a user is searching a food item, the phone program needs to make a request to a web server and display the result when it receives the response from the server. On the other hand, when a user is uploading food item information, the phone program posts the information to the server and server saves it to its database.

The developing environment includes my Latitude Dell laptop, acts as the web server, and a G1 (Android phone), acts as the client. For the server, I am using XAMPP for Apache distribution which contains default MySQL and PHP. By setting up this PHP web services, my phone as the client can easily access the database (using MySQL) via HTTP. The server is written in PHP. On the developing machine, I have also installed Eclipse and Android emulator add-in for developing and testing purposes. The client is written in JAVA. The project is tested on both emulator and a G1 (Android) phone.

In this paper, I am going to discuss my progress in implementing this food menu application. The next sections will discuss the deliverables in detail for my CS297 course this semester. At the end, there will be a conclusion for the deliverables for this food phone application.

2.0 DELIVERABLES

2.1 Deliverables

In order to present the progress I made in CS297 for the project, in this section, I am going to describe several deliverables I completed this semester:

2.1.1 Deliverable 1

Deliverable 1 was to write a test program to use the GPS feature for our phone application. Currently, the application is able to simulate the location within the emulator by sending in the coordinate with DDMS tool. In order to use the Google Maps, it is necessary to obtain a Maps API Key by getting the MD5 Fingerprint of my signing certificate, getting the MD5 fingerprint of the SDK debug certificate and then registering the certificate fingerprint with the Google Maps service.

One of the advantages of having a phone application to search food items is mobility. With the GPS features, the user can easily get the current location in order to search for specific food item from the corresponding nearby restaurant. This program is a proof of concept of how to use the Google Map API in order to get the current location's map.

Here is some source code snippet in order to achieve this:

```
private class MyLocationListener implements LocationListener
    {
        @Override
        public void onLocationChanged(Location loc) {
            if (loc != null) {
                GeoPoint p = new GeoPoint(
                        (int) (loc.getLatitude() * 1E6),
                        (int) (loc.getLongitude() * 1E6));
                mc.setCenter(p);
                mc.setZoom(16);

            }
        }
        }
```

This code snippet implements the LocationListener interfaces in order to display the given location on the map to as a point. After implemented our own LocationListener, in our MapActivity class' onCreate method, we need to find our current location by using the LocationManager and use our defined LocationListener to display it on the map:

```
    LocationManager lm = 
(LocationManager)getSystemService(Context.LOCATION_SERVICE);

    locationListener = new MyLocationListener();

    lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, 
ocationListener);

    mapView = (MapView) findViewById(R.id.mapview1);

    mc = mapView.getController();
    boolean on = true;
```

Finally after we finished our onCreate method, it is necessary to remove the updates when we pause the application:

```
        public void onPause(){

         lm.removeUpdates(locationListener);

         super.onPause();

        }
```

2.1.2 Deliverable 2

Deliverable 2 was to write a test program to obtain a list of zip codes which have geographic short distance to the given zip code. Besides integrating Deliverable 1, the program is getting the current location zip code and obtains the list of zip codes for the related cities.

When we are searching for one place according to the zip code, we expect not only the inputted zip code, but also the zip code geographically close to the searching zip code. The Deliverable 2 is the feature for my phone application to get a list of zip codes near to the search location. The way I achieve this is by sending a request to a free web service and getting the results back to the phone.

With the source code from Deliverable 1, we have got the current location and displayed it on the map. The next thing we need to do is getting the address from the current location by using Geocoder:

```
     Geocoder geocoder = new
Geocoder(getBaseContext(),Locale.getDefault());

     List<Address> addresses1 =
geocoder.getFromLocation(loc.getLatitude(), loc.getLongitude(), 1);

        StringBuilder sb = new StringBuilder();

          Address address1 = addresses1.get(0);
          sb.append(address1.getPostalCode()+"\n");

        //discuses this function as following
          getNearCities(address1.getPostalCode());
```

In the getNearCities function, with our zip code, we call construct the corresponding URL in order to get a list of geographic zip code from the web services:

```
     String location =
"http://codebump.com/services/placelookup.asmx/GetPlacesWithin?Authenti
cationHeader=hXhFedrtFe%2F6K6wLhXGHgnzCAOFh%2F800kU3Fm2M3f2SKR1zSmw3ILG
%2BZs1iRszJqeCeuJLJikkSRbNVcwFR4kIfz%2BWfUM31S&place="+zipcode+"&state=
&distance=4&placeTypeToFind=ZipCode";
```

Then set up the socket to send this URL to the web services in order to get the result:

```
     HttpURLConnection urlConn = (HttpURLConnection)
url.openConnection();

     BufferedReader in = new BufferedReader(new InputStreamReader(
                       urlConn.getInputStream()));
```

2.1.3 Deliverable 3

The purpose of Deliverable 3 was to design food item database and user interface. It is important to have a well-defined database. In this step, we have the ER diagram and table defined for the food menu phone application to use.

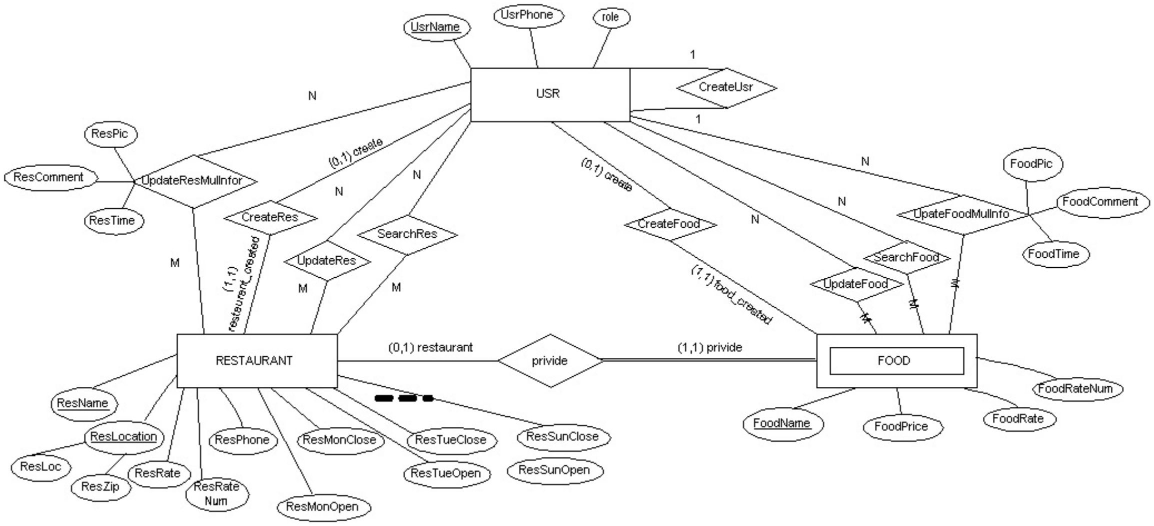

Figure 1. ER Diagram

Figure 1 is the ER Diagram of my food server database. Food, user, and restaurant are the three main entities for this diagram. It demonstrates the relationships between user and restaurant include how a user create a restaurant, how do they search or update the restaurant's information. This helps user to create, search and modify a restaurant easily. Similar relationships between a user and a food item include how a user create, search or update a food's information. Because a specific food item is provided by a restaurant, I made food as weak entity type for restaurant. In this way, the ER diagram is ready in order to form the schema which is described on the next page.

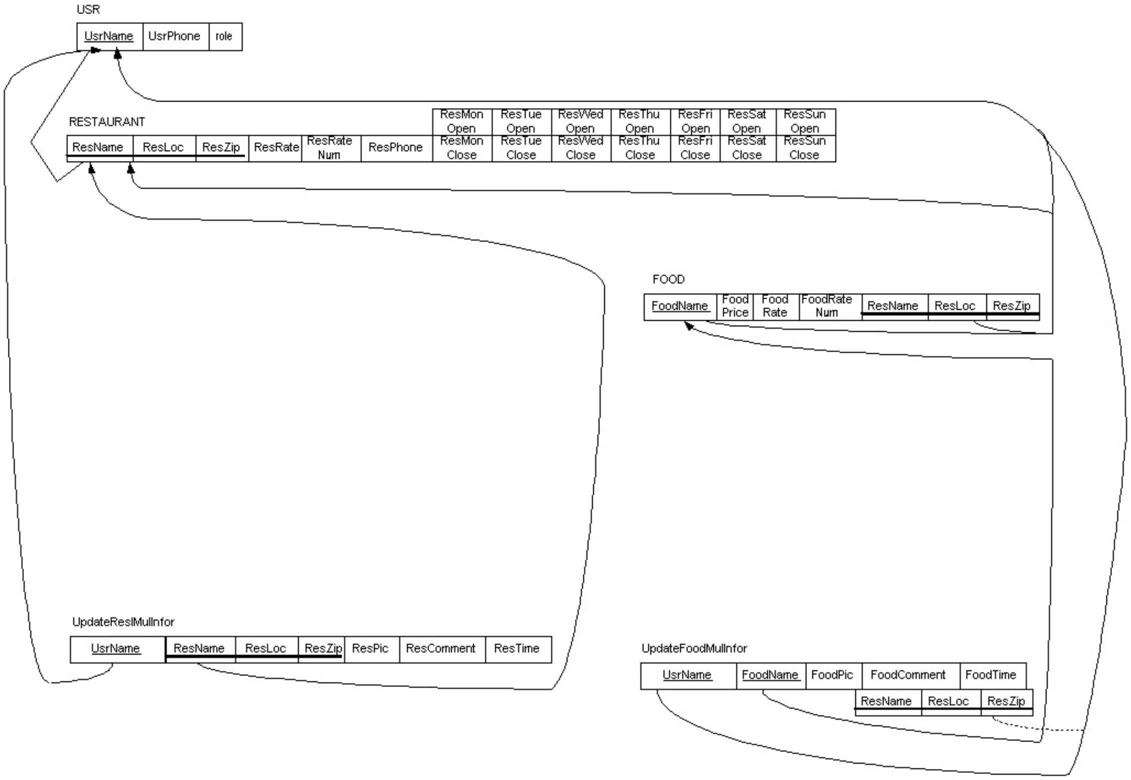

Figure 2. Schema

To the ER diagram of Figure 1, I applied a standard ER-to-Relational mapping to obtain the tables and foreign keys of Figure 2. The diagram helps to create a food database with corresponding tables with all fields and keys set up.

2.1.4 Deliverable 4

Deliverable 4 was to construct a food items table with real restaurants’ menus. One of the most important factors for the food menu phone application is to have a reasonable set of data ready to be searched in the database for customers to use. Having the database well-defined by the end of Deliverable 3 and having the data inputted from Deliverable 4 will be essential factors for the success of my phone application.

Some examples of mySQL statements as following:

Insert a new Restaurant:

```
     INSERT INTO `fooddata`.`restaurant` (
     `ResName` ,
     `UsrName` ,
     `ResLoc` ,
     `ResRate` ,
     `ResPhone` ,
     `ResMonOpen` ,
     `ResMonClose` ,
     `ResTueOpen` ,
     `ResTueClose` ,
     `ResWedOpen` ,
     `ResWedClose` ,
     `ResThuOpen` ,
     `ResThuClose` ,
     `ResFriOpen` ,
     `ResFriClose` ,
     `ResSatOpen` ,
     `ResSatClose` ,
     `ResSunOpen` ,
     `ResSunClose`
     )
     VALUES (
     'pho queen noodle house', 'Kelly', '1133 Tasman drive, Sunnyvale,
CA94089', '3', '4087457753', '9', '21', '9', '21', '9', '21', '9', '21',
'9', '21', '9', '21', '9', '21'
     );
```

Insert a new Food item:

```
     INSERT INTO `fooddata`.`food` (
     `FoodName` ,
     `FoodPrice` ,
     `FoodRate` ,
     `ResName` ,
     `ResLoc` ,
     `UsrName`
     )
     VALUES (
     'Jumbo shrimps, jumbo crab claw, pork & chicken (Dry or Soup with
thin rice noodle)', '8.75', '4.5', 'pho queen noodle house', '1133
Tasman drive, Sunnyvale, CA94089', 'Kelly'
     );
```

Insert a new User:

```
     INSERT INTO `fooddata`.`usr` (
     `UsrName` ,
     `UsrPhone` ,
     `role` ,
     `UsrLoc`
     )
     VALUES (
     'Angela', '4082593876', '1', 'Siant Lucia street, San Jose, CA 
95127'
     );
```

3.0 Conclusion

In this paper, we discussed some details of how I developed my phone application. We also covered some characteristics and special features for each of my deliverable for CS297 this semester. These include how I got current location and display it in the map, and how I got a list of geographic short distance to the given zip code and how I designed and constructed the food server database. With the deliverables implemented as a proof of concept, I will be able to concentrate on how to implement the actual application with a user friendly interface, besides, I will create a stable backend to communicate with a web server embed a fully developed table with some actual food data.

REFERENCES

Google Maps API Concepts. Retrieved August 27, 2009 from Google web site: http://code.google.com/apis/maps/documentation/index.html.

Leena Rao. (April 2, 2009). Citing Websites. In *Yelp Focuses on Mobile, New And Improved iPhone App Coming Soon*. Retrieved August 27, 2009 from techcrunch web site: http://www.techcrunch.com/2009/04/02/yelp-focuses-on-mobile-new-and-improved-iphone-app-coming-soon/

Geographic Queries on Google App Engine. Retrieved August 27, 2009 from MetaCarta Labs web site: http://labs.metacarta.com/blog/27.entry/geographic-queries-on-google-app-engine/

[1989] The Geographic Database – Logically Continuous and Physically Discrete.Peter Aronson. 1989.

[1993] Geographic Regions: A New Composite GIS Feature Type. Jan van Roessel and David Pullar. 1993