2010

Food Phone Application

Shengyu Li
San Jose State University

FOOD 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

May 2010

ABSTRACT

FOOD 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. People sometimes may just know what food they wish to eat instead of the restaurants' name. Without knowing any restaurants' names, our food application's search only requires the name of the dish (e.g., hamburger, spaghetti, etc) in order to get the list of restaurants that serve these items and their corresponding information (e.g., location, hours, phone number, item's price, etc.).

An advantage of using my food application is the system not only includes Google Map, but any information other users have inputted. When a user wants to input a food item, one can either upload the item's picture or a template picture to the server and input the rating and comments about the specific food item. With the rating option, my project calculates a cumulative rating result based around the original and other user's input. There is also the option of having the users input a zip code to better identify where to find the food.

Based on the phone's capability, the system also needs to figure out the physical phone location. This requires the phone to receive the GPS signal. As a result, users can search/upload the local restaurants' food without inputting the current location.

Acknowledgements

I feel extremely grateful to my project advisor, Dr. Chris Pollett, for his encouragement, management and support from Fall 2009 to Spring 2010. This project would not have been possible without his technical advisement, guidance and thoughtful insights from his lectures and our weekly discussion sessions. The first time we talked about this project was the summer of 2009, when I started learning about phone applications and was motivated by Dr. Pollett to take his CS185C (Phone application) course in the Fall of 2009. It is a very useful course for me to learn how to develop applications on both iPhone and Android phones. Special thanks also goes to the San Jose State University Computer Science, which provided me with an Android G1 phone to develop on.

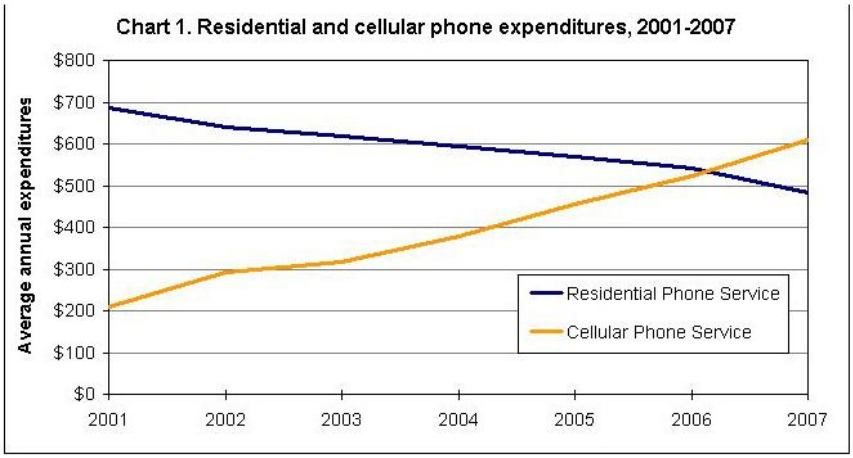
I also want to thank my committee members Dr. Robert Chun and Dr. Chris Tseng, who took the time to discuss this project with me and offering me helpful suggestions and support.

Finally, I want to extend my appreciation to Louis Zhang and Scott Chan, who gave me lots of useful suggestions on high-level application design. Last but not least, my auntie Wai-Shueng Lee, my friends Mason Tse, Chris Chan and Philip Lay who helped me to proofread my report.

Contents

1. 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. In the Figure 1.1 below, from 2001 to 2007, while the residential phone expenditures are dropping, the cellular phone expenditures are increasing dramatically in all age groups.



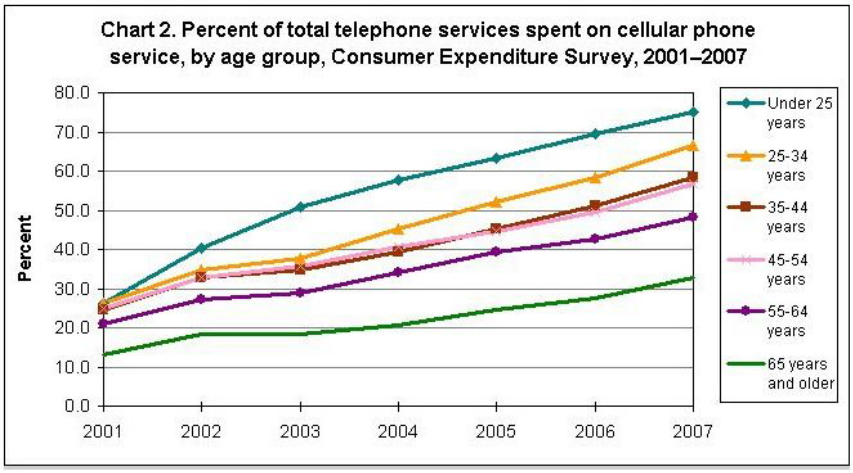


Figure 1.1 Cellular Phone Statistics

In order to satisfy the large number of cell phone customers, the mobile phone industry is making improvements in cell phone hardware as well as software. Hence the science fiction devices of yesteryear have become reality. Cellular telephones today are remarkable in their advancement. Whereas cell phones of yesterday were useful simply as communication devices, for many people today, a cell phone is not only just a tool to communicate with others; it serves also as an alarm clock, calculator, internet browser, notepad, camera, calendar, music player, voice recorder, GPS and computer game. This has been made possible with the advancement of cell phones. Today, cell phones are not just telephones; they are small computers with heavy processing power.

One recent development in cell phone usage is online restaurant reviews. Before deciding what to eat for dinner or stepping into a restaurant, people will often consult restaurant reviews through their smart phones. In order to help people search and order food with their smart phones, my graduate project is to design and implement a food menu search phone application.

2. Current Existing Products

In the past, a person had to rely on either their own experiences or on a recommendation from a friend in order to find a good restaurant. However, recent advances in technology have created new avenues for finding a good place to eat. Today, one can simply pull out their GPS and look up a Point of Interest (POI). Alternatively, one can plan ahead and use their home computer to look up restaurants through their favorite search engine. Lastly, one can pull out their cell phone and look up the right place by using special phone applications.

In chapter 2, we are going to explore some of these systems in detail: section 2.1 will discuss information regarding Stand-alone GPS systems; section 2.2 is going to discuss relevant web based applications; section 2.3 covers information about phone applications.

2.1 Stand-alone GPS

Global Positioning System (GPS), called NAVSTAR-GPS officially, is comprised of 27 satellites that are orbiting the earth. Each of these satellites emits a signal. The basic design of GPS is similar to typical ground-based radio navigation systems. By receiving microwave signals precisely from a constellation of satellites, a GPS is capable of determining its current location, speed, time and a calculated direction to a destination.

In this section, we are going to discuss how GPS is used in daily life, the Point Of Interest (a GPS feature), and the advantages and disadvantages of general GPS devices.

2.1.1 GPS POI

Most GPS models include a POI (Point of Interest) feature, which could be very useful for the user to get brief information about different businesses in a given range of the current location. Of particular interest to us is the ability of a GPS to locate a food-related POI; the most related subject is the search for the restaurants nearby from a GPS device. Most likely, the search result would be a list of restaurants and if a user selects one of the restaurants, a GPS will provide some brief information (address of this restaurant, phone number, and the distance between the current location to this restaurant). When the user decides to go to this restaurant, the GPS device will instantly calculate the appropriate path and estimated time of arrival.

2.1.2 Pros and Cons

One of the reasons GPSs have become so common is GPS signals are open and available to the public for free. The manufacturers of GPSs did not have to pay for or maintain the system of satellites used to triangulate a location. Many GPS manufacturers have taken advantage of this by creating devices that merely “grab” GPS signals and calculate a given location. For this reason, GPSs are very affordable.

Most GPS devices have big screens. With oral instructions and a large screen to display the map, it is very handy to use when a user is driving and getting the directions immediately. Once a user buys a GPS or a car with the GPS feature, the device itself comes with the current map and POI. Everything is already set up correctly and the POI is simple to use and can serve users for general interest.

However, if the GPS embedded in the vehicle, one can only use this device while within the car. Sometimes when users elsewhere, or in their friend’s car, it is not

possible to retrieve information from their GPS device. Therefore, an individual GPS device will be more handy and flexible.

Another drawback of most stand-alone GPSs is that the information within the device is dated. Most of the GPS devices include the Map and POI, the database is static in the device; users are required to pay additional money in order to obtain updated maps and POI information. Another limitation of standalone GPSs is that the device only provides very brief information about the restaurant. There is no other way for users to get more information through the GPS device. Besides, GPS device is only a display device; there is no interactive feature. Users are not able to input and share any information such as comments about the restaurant, food, ratings, and uploading pictures.

2.2 Web Based Application

The number of web applications have grown dramatically as the Internet has developed. As long as a user has a computer with internet access and a browser installed, he can make a request to a static URL web application server. All the technical detailed implementations are hidden from the users, it is quite easy for anybody to use anytime.

2.2.1 Web Application Architecture

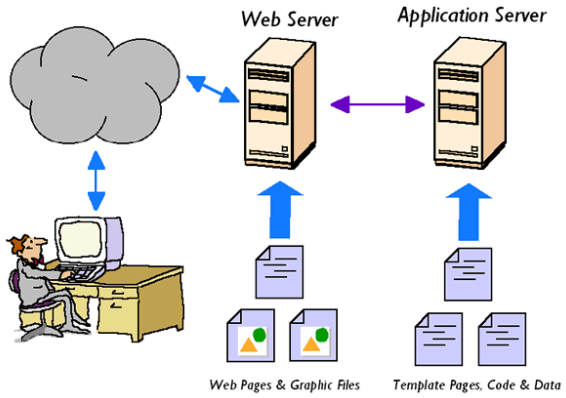


Figure 2.1 Web Application Architecture

Figure 2.1 shows the architecture of web server application. Typically, the client uses a web browser to communicate with its web application server. The web application server supports and generates different types of dynamic content and sends back to its clients. When a browser sends a request through the Internet to the corresponding web application server, unlike the web server which just retrieves static information, it generates dynamic result by using different template pages, code and data and sends it back to its computer clients.

2.2.2 Web Application Usage

There are many different types of web applications in the market. They are bank account web applications, student registration system for a college administration office to manage student record, or just a simple message board for people to chat in their free time.

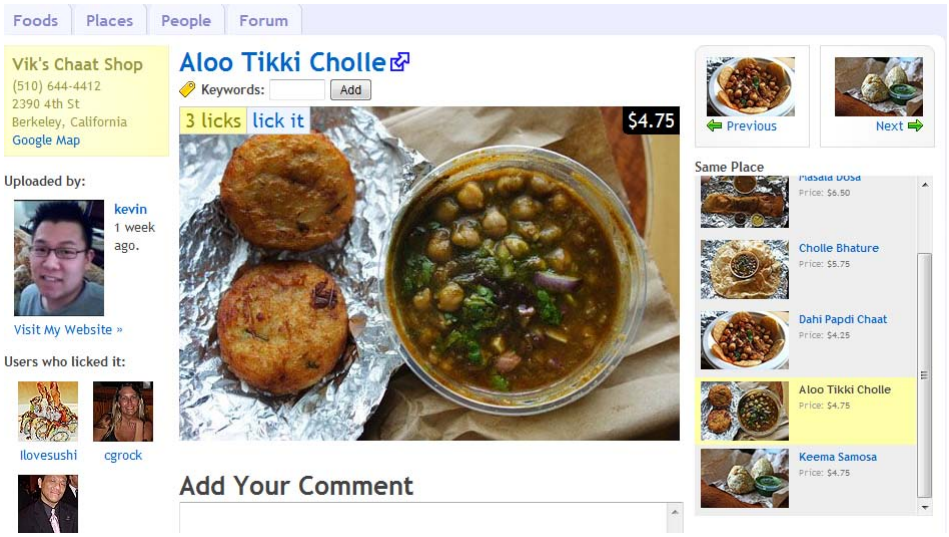


Figure 2.2 Food web Application

Figure 2.2 shows a web application focus on food item. Users can search for a food item online; they can also upload or view each food item online. Under each food item, there is some information such as the food picture, food price, customers' reviews, restaurant's name, phone number, direction and maps may display under each food item.

2.2.4 Pros and Cons

The web based application is very simple to use. Users are not required to have much technical skill and all they need to do is just stay at home and retrieve the information of their interest.

Comparing to GPS devices, the web base application has colorful user interfaces. People tend to use web-based applications on home computers, which tend to have much bigger screens than a typical GPS device. Also much more detail information provided by other users can be displayed on the web than on a GPS device.

However, web applications do not have all of the efficient features that a GPS has. Even if one has the address of the restaurant, in order to get directions to the restaurant; the current location is needed since there is no GPS receiver embedded in a computer.

Since web application's client is most likely a web browser, a web application cannot motivate users to input feedback. Also, most people find it inconvenient to bring their computer with them everywhere just to give feedback for food orders. Since the minimum requirements of using web applications are computer and Internet data, most people cannot input feedback while at the restaurant and most probably will also lose the interest of giving feedback by the time one gets back home.

2.3 Phone-based Application

As described in the previous section, GPS and web applications are useful in different situations. Because a smart phone comes with a lot of useful features such as GPS receiver, camera, notepad, timer, and calculator etc., smart phone applications can be easier to use when one is on the go than your typical GPS or web application. Smartphones are also very small, which makes it easier to carry around with you. Since most come with an internal camera, users can also easily upload pictures and ratings and feedback while they are still in the restaurant.

2.3 Phone-based Application

As described in the previous section, GPS and web applications are useful in different situations. Because a smart phone comes with a lot of useful features such as GPS receiver, camera, notepad, timer, and calculator etc., smart phone applications can be easier to use when one is on the go than your typical GPS or web application.

Smart phones are also very small, which makes it easier to carry around with you. Since most come with an internal camera, users can also easily upload pictures and ratings and feedback while they are still in the restaurant.

2.3.1 Phone Application:

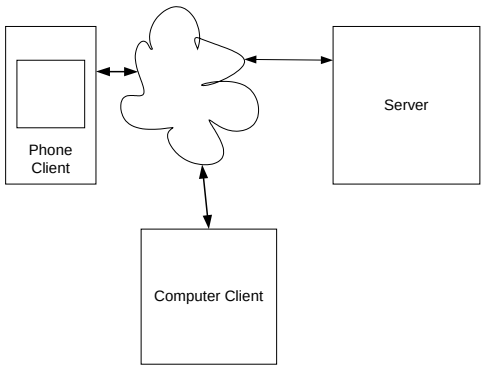


Figure 2.3 Phone Application Architecture

We have already discussed some characteristic of a web based application in the previous section. Most of those benefits carry over when used on a smart phone. In fact, there are lots of web server can handle communication with both phone and computer clients. Figure 2.3 show general phone application architecture. We can see how both phone and computer can communicate with the server via the Internet.

One successful example is Google Maps, one of the most popular map applications. Similar to GPS devices, some applications provides point of interest for users to retrieve their desired information. By using Google Maps, similar to GPS' POI feature, users can search for points of interest in different categories. Unlike a GPS, as long as internet access is provided, users can get the most updated information at no charge.

Google Maps is one of the most popular map applications. In our case, if a user wants to search for a specific business (such as Milk Pail Market in Mountain View), they can get the following output:

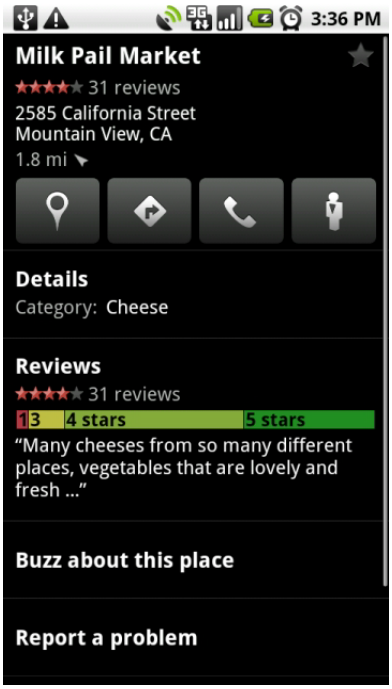


Figure 2.4 Google Maps Phone Android Application

As we can see from the Figure 2.4, Google Maps is very useful for users who need information on a particular place when they already know which place they have in mind. The result output of Google Maps contains brief information (such as restaurant's name, address, phone number, directions to the restaurant). In order to show other people this information, a user can use the "Google Buzz" layer by Google Maps. Because a Google Android user is automatically set up with a new Google mail account and all Gmail accounts are automatically enrolled in Google Buzz, Google Maps provides an option for users to buzz about this place so that the others can instantly see related information.

The applications we focus here are search engine products, especially for food and restaurant searches. In order to search for a restaurant within a region, Yelp is a popular site with phone application for many people to use because it provides reviews from real people.

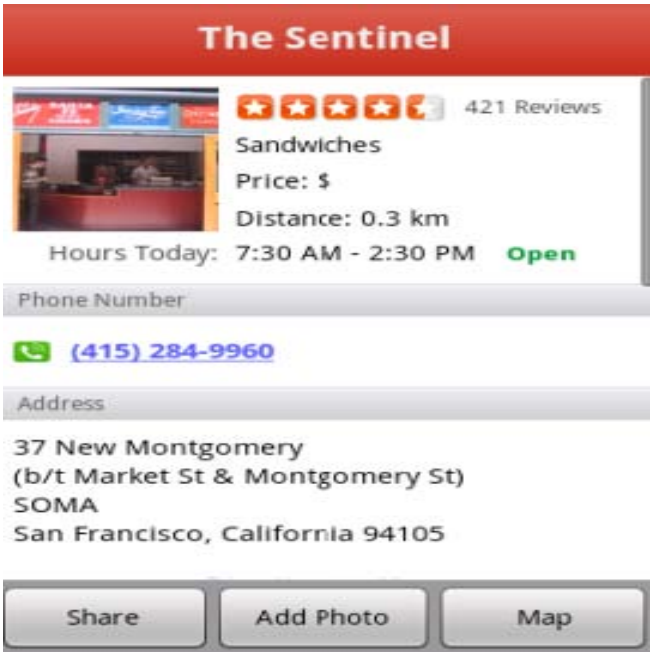


Figure 2.5 Yelp Phone Android Application

Similar to the Google Maps, as shown in figure 2.5, Yelp provides the ability to search by restaurant address, shows the distance from the current location to the restaurant, and displays ratings and reviews about the restaurant. It also provides restaurants' business hours (with the current status as open or closed). Additionally, the application looks much better due to all the colorful pictures that convey a sense of

the restaurant's style and the type of food items offered. Users can also add more photos and write their comments about the restaurant.

Comparing to Google Maps, Yelp contains much more information about restaurants, including menus, descriptions of food items, and their related ratings and comments from users. In fact, so many users use and trust the system that Yelp has become a restaurant community place where people come to discuss and critique the food and service. Google Maps is more useful for general points of interest while Yelp is primarily focused on restaurant reviews. Consequently, Yelp provides more detailed restaurant information compared to Google Maps. From a user's point of view, Yelp may be better in finding new restaurants.

Figure 2.6 Urbanspoon on the iPhone

Another interesting phone application regarding food is Urbanspoon (Figure 2.6). Urbanspoon is a free iPhone application which finds a restaurant as long as users

shake their phones. There are three columns display on the screen: the first column is the current location; second column is the type of food; third one is the price range. The users can shake it again until they find what they like. Then they can see this restaurant's phone number and address. For more detail, they can launch to the web server.

Urbanspoon is more like a game to offer people who have problems deciding what to eat, suggestions of what's in their area. When a person doesn't have any idea what to eat, it is a good idea to just pick it for them. However, there are sometimes people know what they want, they just don't know which restaurant provides this food. By using Urbanspoon, users may end up shaking their phone the whole day without going anywhere.

Additionally, when users need further information about the restaurant, they need to launch to web browser to the web server and view the detail in the phone screen. It is less mobile user-friendly when the users have to use their phone's web browser to view restaurant information. Also, it is less likely users would like to upload pictures and comments because of the inconvenience of using the web browser to do feedback on the phone.

Here is some comparison about some relevant systems in the market nowadays:

Phone Application	Review	Ratings	User-friendly Phone UI	Actual Price	Search for Individual Food	Upload Pictures	Menu
Google Maps (POI on Food)	Some	Some	Yes	No	No	No	No
Yelp	Yes	Yes	Yes	No	No	Yes	No
Urbanspoon	Yes	Yes	No	No	No	No	No
BooRah Restaurant Locator	Yes	Yes	Yes	No	No	No	Some
iCrave	Yes	Yes	No	Yes	Yes	No	No
My Food Phone Application	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 2.1 Comparison of phone applications' features

We are comparing several popular phone restaurants' supported features in Table 2.1. This is the motivation of developing my phone application. As other applications are all focus on restaurants, mine is focus on food items. The advantage of this design is we can give users a specific food so that we can provide detail information about this specific food (such as actual price, pictures, ratings, reviews). Besides, since the information is from the restaurant, we are providing the entire menu so that users can order food on the phone with ease. Instead of wasting time shaking the phone (like Urbanspoon and iCrave), we provide users to search a specific food item within any regions.

3. Design

3.1 Goal and Requirements

For most businesses, getting new customers is just as important as keeping old ones. If we can assist restaurants in providing more information to their potential customers, it would most likely encourage new customers to visit those restaurants.

Similar to Google Maps and Yelp, this application would encourage users to utilize our system as long as their phones have internet data services. With our food phone application, users are able to type full or partial names of food items and search for it in the range (from current location or a specific zip code).

Compared to Yelp, which only has some food items or general information about a particular restaurant, our food application provides customers a better idea of what they can order. For example, Yelp only provides dollar signs to indicate the general price of food in a restaurant. Our food application, by listing all items in the menu, allows users to know how much they need to pay before they make decisions on whether they want to walk into the restaurant or not. Besides, when users want to order food by phone rather than walking into the restaurant, they can check all desired food items from the restaurant's menu with pricing, ratings, and pictures.

Furthermore, the application helps with propagating food information by allowing users to upload more pictures to the server as well as store more comments and ratings each of the food items.

3.2 Architecture

The project is based on a typical server-client model as shown in Figure 3.1 below:

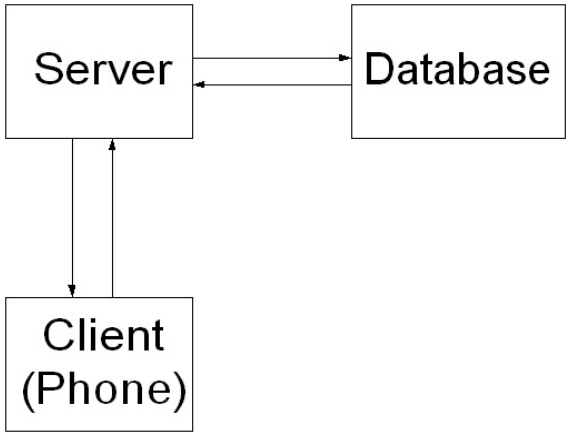


Figure 3.1: Server-Client Architecture

The developing environment includes a remote computer, acting as a web server, with a static domain name set up, and a G1 (Android phone), acting as a client.

Because this project is based on a relatively small scale database, in order to reduce the number of hardware components, the database is also set up on the server side. We are using pre-setup XAMPP distribution which contains Apache (version 2.0), MySQL (version 4.1.20) and PHP (phpMyAdmin version 2.6.2). By setting up the PHP web server, an Android phone and Google phone emulator, act as clients, can indirectly access the database via the PHP server.

For the client, we also installed Eclipse and Android emulator plug-in for development and testing purposes. The client is written in (Android) Java language. The project is tested on both emulator and a G1 (Android) phone.

3.3 Supported Features

This section is going to describe the main functionalities of this application. Our new phone app includes the following features: searching food from a defined region, displaying a list of restaurants, getting the menu of a particular restaurant, uploading pictures and comments to the server through an Android phone.

3.3.1 Search Food

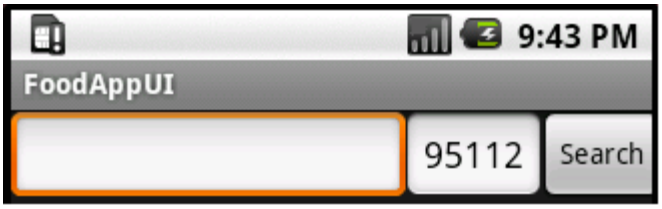


Figure 3.2 Search Food Screen

When users are looking for a specific food item, they input either partial or full name of the food. For example, you can just type “noodle” and search for all noodle items from all restaurants in the current location zip code. Figure 3.1 shows a user entry to find a restaurant in 95112 areas (95112 is the current location zip code). Users can also type a valid zip code for the search scope. The search result will contain a list of food from different restaurants. If the user picks any of the food items from the list, the phone will display the selected food item with detail information such as food pictures, ratings, comments, and the information of the restaurant.

3.3.2 Get Menu

Getting a restaurant’s menu is one of the most important features for this phone application. When a user launches the application, a list of restaurants within the

current location will be displayed in the phone. Further detailed information of a restaurant (see Figure 3.3 below) will be displayed as the user selects a restaurant from the list. So the user can get the entire menu of the selected restaurant. Each food contains detail information which helps user to have better idea so that they can order the food from the phone.

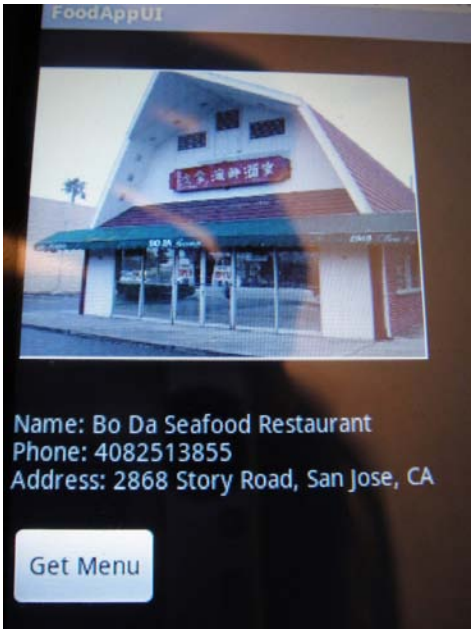


Figure 3.3 Our Food Phone Application Resturant View

3.3.3 Upload Picture

Another useful feature is that the user can take a picture with the phone, and upload it to our server. After users select a restaurant and its food category, they are allowed to take and upload pictures to the server. This is a feature that lots of phone applications, such as iCrave, Urbanspoon and BooRah Restaurant Locator, do not support. Allowing users to have this feature can definitely motivate users to use our food application more frequently.

3.3.4 Comment:

Unlike the GPS and web application, with the picture, rating and comment features available on the phone, this phone application is better suited for users to give instant feedback about the food items. Some phone applications, like Yelp and Google Maps, allowed users to read and write any comments for food items in the system. Users can also rate the food items using the phone application. However, they only provide a web browser UI for users to access, which could be a discourage from reviewing the food items because the less user friendly web browser UI.

3.4 Design Consideration

3.4.1 Platform

In many markets, Android is one of the earliest significant software platforms for open source mobile applications. Its operating system is Linux kernel-based and user applications are built in Java by using the Android SDK. Due to the simple set up, open sources and the code being based on regular Java syntax, developing Android has big potential market in the near future. One of the big comparators is the iPhone. In order to develop applications on the iPhone, a Mac OS is required. In order to develop Android application, programmers can develop code on both Mac and Windows systems. This makes Android applications more flexible.

By installing the Android add-on Android SDK, we can develop and run the appropriate version of Android phone applications on both of the emulator and the real Android Google phone. The Android emulator is pretty easy to set up and use. For most of the features for our food application, we can use the Android emulator. For some of the features such as instant file storage, usage of camera and location

detection, using the real Android phone would be more reliable. Besides, we can also see all the debug messages from Eclipse when we are testing both on the emulator and on the real phone.

3.4.2 Application Functionality Design

When using this application, users are not allowed to use their phone to create a new food or restaurant. This is because we want to avoid people entering incorrect information to our system. For example, we do not want somebody to input the wrong price/food item to affect a restaurant's reputation. Besides, incorrect information could cause our users to lose their trust in our system, discouraging them from using it. On another hand, users are allowed to input information such as comments, ratings, and pictures. This is a nice feature to encourage users to upload more pictures, give some feedback, and rate the food items they just ate. With real users' review and ratings, our phone application's information will be more trustable.

Another thing to encourage users using our application is having a user friendly phone application UI design. Unlike web applications and GPS devices, which have bigger screen, Android phone's screen size (resolution 340X240) is limited. When we want to develop a phone application, the user interface is one of the most important factors and must be handled delicately. Some phone applications, such as Urbanspoon and iCrave, just launch the web browser on the phone to display food information. Within the small screen, it is not easy for user to see or click any buttons and hyper links on the screen. Our application, however, gave users a comfortable UI to work on. For example, all of our buttons and text message are big and easy to click. For multiple

items (restaurants/food), we tried to use a lot of list view with a scrollable bar on the side so that the users can have more active space to view each item.

Another example the project uses is the menu buttons in order to save the screen space. If a user does not push the menu button, the bigger size of the screen could be displayed to the user.

Additionally, when we slide the keyboard out, the screen will automatically switch from landscape to portrait. As a result of defining both landscape and portrait views for each activity, we can have both keyboards, virtual and physical, facilitating user inputs, which help users to have an easier time typing in their inputs.

3.4.3 Modular Design

We separated the project into four components: database, PHP code (server), Java core code (client) and Android Java code (client). The PHP code helps the server connect, manage, and query the MySQL database through a corresponding URL. The Java core code connects to the server using the correct URL, does the parsing and translates the value into an abstract level to make the data ready for the next level (Android Java code) to use. For the PHP code and Java core, both of them contain the implementation of all defined major APIs. In this way, the server and client can communicate easily.

Separating the code into different level simplified the debugging process. For example, when we work on the PHP server section, all we need to do is to use a browser to access the server via the corresponding URL. If the output is not as what we expected, the error would reside in the PHP code only.

After we completed the major functions, we worked on the next model, which is the Java core code section. This is the middle module; the main purpose of this module is to generate a valid corresponding URL and send the request to the PHP server. As we mentioned before, with the same name as the PHP function, if the return result is the same as calling the result from the URL, we know that the URL is correct. So we will get the result as a String value. Furthermore, Java core module parses the result into Java corresponding data type for the Android UI module to use later.

The next module is Android Java core; it is the main component of the project. It handles all User Interfaces inputs and outputs. Besides, the module also handles the

phone features, such as getting the current location, taking pictures through the camera, uploading and downloading pictures to the server and so forth.

3.4.3 Application Design

Firstly, we need to design a protocol between the client and the server. It takes two steps to do so, the first step, called Application Design, is the highest level to define what kind of tasks the food server and client can handle.

We listed all the possible inputs and outputs for this phone application:

Outputs:		Inputs:
a)Location		1)Location
b)Menu		2)Search Range
c)Rating for Food		3)Rating for food
d)Restaurant Name		4)Food Name
e)Food Price		5)Restaurant Name
f)Food Comment		6)Food Price
g)Picture		7)Food Comment
h)Phone Number		8)Picture

Figure 3.4 Mapping of the Inputs and Outputs

This step is very important because it helps a developer understand what kind of information the user can get from and send to the server. After that, we can map each of the output to the some of the inputs to define all getter and setter functions:

Getter:

- a) = 5), 1), 2)
- b) = 5), 1)
- c) = 4), 5), 1)
- d) = 1), 2)
- e) = 4), 5), 1)
- f) = 4), 5), 1)
- g) = 4), 5), 1)

Setter:

- c) = 4), 3), 5), 1)
- f) = 4), 5), 1), 7)
- g) = 4), 5), 1), 8)

Figure 3.5 Getter and Setter functions

Application design is a fundamental step since all the major features are based on this step. It defines a high level protocol between clients (mobile phone) and the server. As this is done, we can move on to the implementation design step, which will be discussed in the next chapter.

4. Implementation

There are lots of web base applications which utilize LAMP (Linux as operating system, Apache HTTP Server, MySQL and PHP) software stack as it is an open source. As it is easy to use and also reliable and reasonable in performance, we are using LAMP set up on our server side.

4.1 Database

Similar to many web applications, for restaurants and their menu's storage and user logging information, it is necessary to have a database set up for the system.
Since this system's database is simple, MySQL database is used for this phone application.

4.2 PHP

PHP, which can be replaced by Perl or Python, is the only language that runs on our server side. Like other web application, others can use a browser to retrieve data from our database via PHP server.

4.2.1 Database Access

One of the most important roles for the PHP module is helping users to communicate with our database indirectly. From the previous section, we defined all public function names precisely in the PHP file, which means, the server will return correct result only if users put the correct URL with correct function names and parameter names with appropriate values. Otherwise, user will receive an error

message saying this is not a valid function to call. Therefore, all users won't have any direct access to the database.

4.2.2 Upload File

Another responsibility is accepting files (picture files) from clients. As it is built on a Linux system, since the given default setting for upload directory is unknown, within the PHP file, it is necessary to change the change the permission code in order to allow users to have write permission.

Secondly, since the uploaded pictures will only display on the phone screen, for a better performance, the picture resolution does not really need to be high. This can save lots of space for the server and downloading and uploading time from the server to transfer files.

4.3 Java Core

After we made sure the PHP server is working correctly for all valid URL, we can start to work on the Java core module. The main responsibility for Java core module is to assembly all URL for all functions with correct parameters and appropriate value , make sure they can get exact output as we use a browser and send request to the PHP server. Theoretically, this module could be written in other language as well as it could generate URL and parse the result into appropriate form to the next module (Android UI module). Java is obviously the best solution in here because using Java to build the URL and parsing the result are very simple and most importantly, Android is already include the JDK library.

Same as doing basic java networking code, we need to set up an URL object with an appropriate URL, then use a HttpURLConnection to open a new connection and read in data by using the Inputstream.

When we get the http response from our PHP server, as it is not the result from a web services, Java core module needs to handle the parsing so that the next module can retrieve this result easily.

4.4 Java Android

Since this is a phone application, the Java Android is the most crucial part for the project. It directly interface with users, handles all user input events to trigger internal functionalities for the phone and display user understandable output to the screen. Besides, this module tried to take the advantages of Android smart phone's advance features which will be described in the following sections.

4.4.1 GPS Location

Taking the advantage of having GPS receiver embedded in the Android phone, our application uses LocationManager to obtain the current location information from the GPS satellite signal. With this GPS features, the user can easily get the current location in order to search for specific food item from the corresponding nearby restaurant.

4.4.2 Camera

Android phone G1 module comes with a camera device. The picture's resolution could be up to 240x320. For performance concern, since we need to download and upload pictures from server to clients and the picture quality does not need to be high

(as long as its clear enough to display on the phone screen), we set the pictures' size as 240x320 (the same as the screen phone size).

In Java, we implemented SurfaceHolder.Callback to define a surface, then used the Camera from the Android to control the camera activities, and write the image file onto the phone's SD card.

4.4.3 Fancy List View

By default, we can only create a simple list view on the phone. The list is only text message which are stored in an array variable. When we need to create a fancy list view, it is necessary to extend the ArrayAdapter in order to add customized style of into the list view. In this way, we can add different components dynamically to the list and display them on the phone at runtime.

4.4.4 Pictures

Uploading and downloading pictures to the server are important features for this phone application. In order to make sure the system can refer to the correct picture for its corresponding food item or restaurant correctly, we gave the restaurant's picture name as the restaurant's name plus the location (longitude and latitude). This is because that in given a location, there should be only one restaurant at that point; and we only have one picture for this restaurant. As a result, there should not have any duplicated picture name for this. For the food item, since it's within the same restaurant, assume there are no duplicated food items. We use the food name plus the restaurant picture name string with an picture counter number so that when the user click the "next" or "previous" it will display the correct food picture on the screen. After the client

downloaded the pictures to the phone’s SD card, with the correct naming schema for all food pictures, user can click the next button to display the next picture.

5. Testing

5.1 Testing Strategy

Our strategy for testing code is as follows: we first test each individual module to ensure their correctness; then we put all modules together and test the integrated system on a simulator; finally, we will test our application on a real phone. In this section, we will go through each step in detail.

5.1.1 Individual Modules

Since each module is written in a different language and serves different functionalities, we will test each module individually to ensure correctness of each before putting them together as the final application.

5.1.1.1 PHP

To test the PHP server code, we would use pre-defined URLs to test each function. We would launch the URLs in a browser, and then check if the corresponding output is correct. For example, for a given API:

> ArrayList<Restaurant> getResList(double LatMax, double LatMin, double LongMax, double LongMin, int str, int end)

We can test this function on the PHP server side by using the following URL:

http://localhost:8888/foodserver.php?function=getResList&LatMax=38&LatMin=37.3&LongMax=-121&LongMin=-122&str=0&end=2

We are trying to the first two restaurants within this location (37.3<latitude<38 and -122 <longitude<-121). If the function is working correctly, we will get a list of restaurants like this as shown in Figure 5.1:

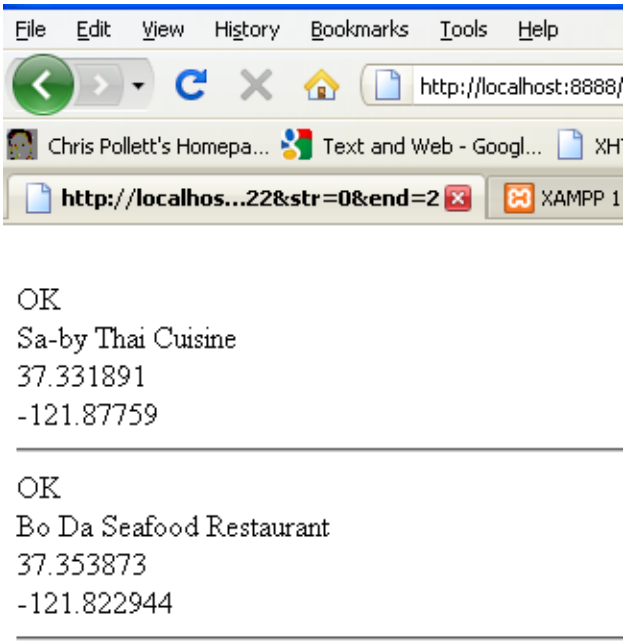


Figure 5.1 Actual Output of getResList function

In Figure 5.1, “OK” is the status indicating success after calling this function. Invoking this link returns two restaurants. It also lists the restaurants’ location (in latitude and longitude). By using a browser for testing, we can limit the test scope to the PHP module only. As long as returned results match our expected results, we can ensure correctness of those PHP functions under test (getResList in this case).

The following table 5.1 shows the correctness test of the main functions of our phone application. As we can see all actual results match with our expected results. All major phone functionalities tests are passing.

Table 5.1: pass if actual result is equal to expected result

	Expected Result	Actual Result	Status
getResList	5	5	Pass
getResMenu	5	5	Pass
getRes	1	1	Pass
getFood	5	5	Pass
searchFood	4	4	Pass
searchFoodbyZipCode	4	4	Pass

5.1.1.2 Java Core

We test our Java core by calling test functions in the main method. These test functions mainly provide text transferring features. We can hard code the current location to get a list of restaurants within the range. To simplify the testing process, we can even test this module in a separate pure Java project and use the system print out method to print debug messages on the screen.

5.1.1.3 Android Java Code

After passing all testing cases for PHP and Java core modules, we then test the Android Java code and the entire product. We will test the product in an emulator first. Test cases will only include a subset of those designed for the entire product because some features, such as GPS receiver, camera, and transferring picture files, cannot run properly on the Android emulator. We have to use the real phone to test those features.

5.2 Performance

We will measure performance of our application. Section 5.2.1 will discuss networking performance, whereas Section 5.2.2 will discuss convenience of UI.

5.2.1 Network Performance

We measure network performance by measuring the time it takes a client to finish downloading restaurant's information and food information based on different numbers of results the server return. In order to measure the elapsed time, we connect the phone to our development computer to read out log message from the phone application. We log the timestamps before and after a download. The less time it takes the higher performance this application provides. Figure 5.2 shows the download time for different numbers of restaurants, and Figure 5.3 shows the download time for different numbers of food. It takes 98 ms and 116 ms to download one restaurant's information and to download food information on average, respectively. This result indicates the average time for downloading food information is slightly higher than that for downloading restaurant's information. This is because in order to find the food, we still need to find the corresponding restaurant from our database first. However, we believe any performance bottleneck for the download times should be related to network conditions. Overall, the performance of our system is very acceptable from a usage perspective.

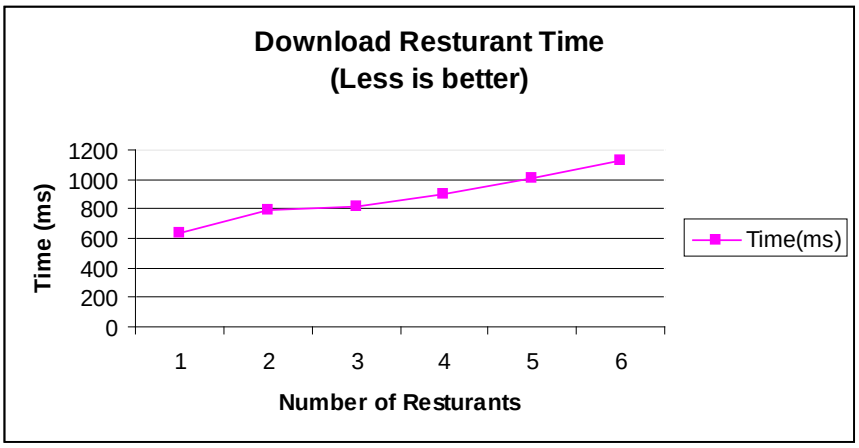


Figure 5.2 Download Restaurant Time

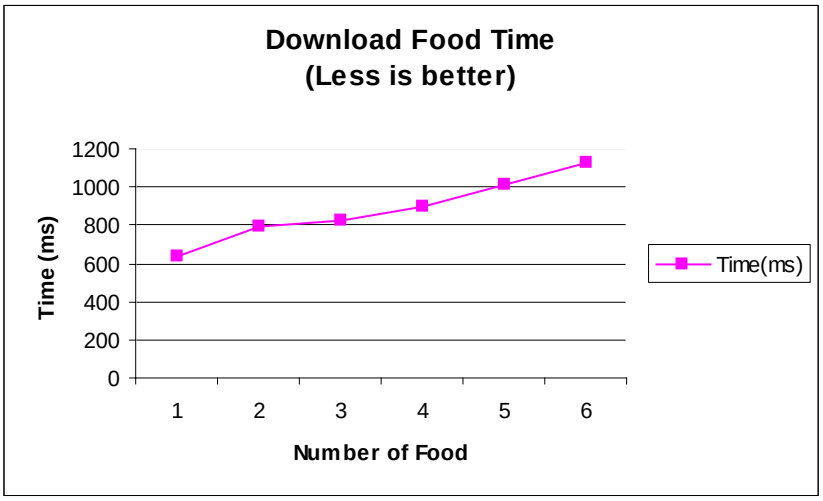


Figure 5.3 Download Food Time

5.2.2 Convenience of UI

We measure convenience of UI by comparing the number of steps we need to go through in order to search for one specific food item. The fewer steps it takes, the better UI design the system is because fewer steps indicates ease of use. Figure 5.4 shows a comparison of looking for different food items by Yelp, iCrave and our food application. Test result showed that our application gave the best search results whiling requiring the least user interactions. Other application can return irrelevant result while

requiring many user steps at the same time. For example, when we searched for “Ma Po ToFu” using Yelp, the first search result is some Korean ToFu soup, which is not relevant to the food (a Chinese dish) we are looking for at all. Users may need to have some basic knowledge of the food in order to pick some “possible” restaurants that serve “Ma Po ToFu”. It took us 20 UI actions to get the best match result. When using iCrave or our phone application, it took much fewer steps to find the same “Ma Po ToFu” food item, since both of them allow users to search for specific food item. As a result, a search bar specific for food item is a valuable feature when we are looking for a food item by phone.

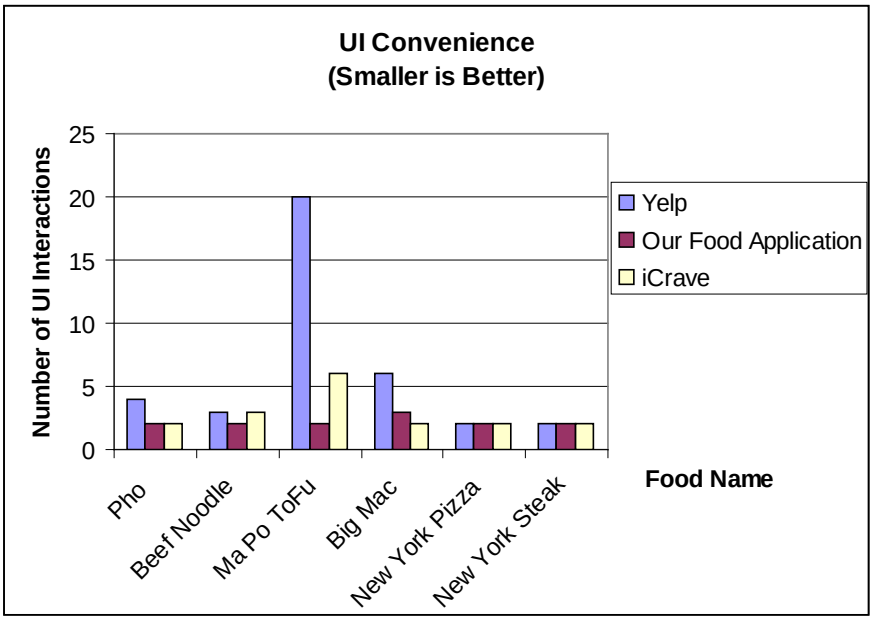


Figure 5.4 UI Convenience

5.3 Issues and Solutions

In the process of developing this application, we encountered several issues due to the complexity of the whole system as well as the limited budget available to this project.

5.3.1 Set up Server

When we were setting up the server, we ran into some communication issues with both the emulator and the real phone. In addition, we had issues to invoke store procedure from the PHP server. We will also discuss some different between pure Java code and Android Java code, list view icon's picture resolution issues and how to solve the multiple users access server issues.

5.3.1.1 Emulator

If we are hosting a server on a different machine from that running the emulator, we must use an explicit IP address in order for the client to access the server. If we use our machine to both run the client and host the server, we can use "localhost" as the IP on the client machine to talk to the server. However, our client is a mobile phone emulator. Even if we use the Android emulator to access the server hosted on the same machine, we could not use "localhost" to reference our server. Instead, we have to explicitly specify the server's IP address.

5.3.1.2 Real Android Phone

If we use a real phone to run our application, within the same subnet, we need to use a static IP or a domain name to access the server. This is because the router is not smart enough to know how to forward port properly when the server is assigned a

dynamic IP address. However, using static IP setting to do port forwarding in our test environment is not feasible because we do not have the access to modify such settings. To solve this problem, we use a public server with a static domain name to host our server so that the phone (both the real phone and the simulator) can access the server at anytime, from anywhere.

5.3.2 Stored Procedure

Our original modular design uses stored procedures, with the same API for all main functions, to access the database. This way, the PHP code is only responsible for calling the corresponding stored procedure after accepting user input. This design is neat because each module's responsibility is very clear and equally distributed. However, after we implemented the stored procedures, they only worked fine when we invoked them from within MySQL directly; invoking them from our PHP code does not work as expected because PHP did not work well with MySQL due to versioning issues. So in the end, we changed our design to embed SQL statements into PHP code directly.

5.3.3 Pure Java code vs. Android code

Even though Android applications are Java based, we still need to pay attention to the differences between pure Java code and Android Java code. For example, as mentioned from the section (Section 5.1.1.2 Java Core Module), when we tried to test our Java Core module, we tested it as a pure Java project. But when we switched it back to Android Java code. All of the print screen message statements need

to be cleared; otherwise, we would get the following error message, as shown in Figure 5.5 below.

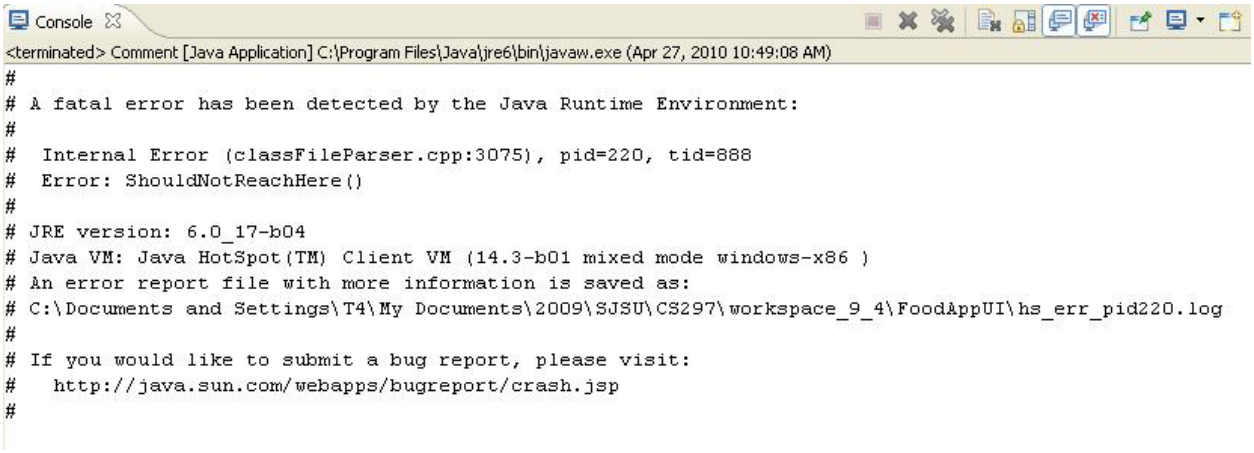


Figure 5.5 Android Run Time Error

This error cannot be resolved by simply deleting files related to this error message from the project directory and then cleaning and rebuilding the project. We circumvented this error by deleting the project (while keeping all its files) and then importing the project to the IDE one more time.

5.3.4 Picture Resolution

For the Android module's Fancy List view feature, it is necessary to convert a picture into a smaller icon. If the size of the picture is too big (maximum image size of 2592x1944), we will get a run time error messages.

We use pictures with smaller resolution to solve this problem. For consistency and performance, we choose 340X240 as the standard resolution (same as the phone screen's native resolution) for image files for uploading food pictures.

5.3.5 Multiple users

Because we only have one available phone for use in our project, we did not get a chance to test the server's capability of handling multiple clients. In theory, our system should be able to handle multiple clients well. We do not need to maintain a state between the server and the client. Other issues, such as naming schema for pictures of a specific food, need to be handled specifically. In this case, the server should be responsible for assigning permanent file names to each picture after they are uploaded.

6. Conclusion and Future Improvement

This paper contains the motivation, design patterns, detailed implementation and testing processes for our food phone application. We went through many issues at the time of working on the project. By resolving each of them, we have expanded our knowledge of updated technology such as Android phone programming, Java programming, PHP programming, and MySQL statements. The food application runs smoothly within the internet environment. This is only a beta release and there are more improvements for the future release:

Similar as a GPS receiver, it would be nice to have a full GPS features such as displaying maps, directions, and current speed implemented on the phone. Additionally, the phone application includes the voice for turn by turn directions. In this way, the phone application can fully replace the typical GPS device. As an open resource, it would be definitely have big market for all Android phone owners. Also since the Android phone already has a phone feature, it would be convenient for the user to make a call to the restaurant with a touch on the phone screen.

The scope of this project is to implement a phone application for user to obtain food/restaurant information easily through the Internet on the Android. The current assumption for the restaurant owner to upload their menu is to send to the administrator. However, since we've already set up a server for the phone to access. It makes a sense if the web server is set up as a web application so that the user can upload menu information through the web.

Since social networking is a current mainstream trend, our food phone application would become very popular if users can share the information about a specific food item to our friends with ease.

7. 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

[2003] Clarke Bishop: What is a Web Application Server?, Retrieved January 2, 2010 from resultantsys web site: http://www.resultantsys.com/index.php/general/what-is-a-web-application-server/

[2008] Ethan: Urbanspoon on the iPhone Retrieved March 23, 2010 from urbanspoon web site: http://www.urbanspoon.com/blog/27/Urbanspoon-on-the-iPhone.html