

ANDROID APPLICATION OF RESTAURANT FINDER

by

SPOORTHY REDDY PARNE

B.Tech, Jawaharlal Nehru Technological University, India, 2010

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2012

Approved by:

Major Professor
Dr. Daniel Andresen

Abstract

Mobile applications can be one of the best ways to keep consumers engaged with a brand as they are on the move. With the increase in demand for smartphones and efficiency of wireless networks, the demand for mobile applications has increased incredibly. Android is one of the most popular open source platforms that offers the developers full access to the framework API's so as to build innovative applications.

The main aim of this project is to build an Android application that helps the users to find a Restaurant in a specified location and according to the specified tastes. The main features provided by the Restaurant Finder application are as follows:

- 1) Basic Search where the user can search for a particular restaurant based on any keyword and Advanced Search where the user can specify the category, rating and the distance range for the restaurants.
- 2) Google Maps that shows the top 5 restaurants in the city of the current location and the routes to a particular restaurant.
- 3) The users can write a review, see the reviews and invite a friend/colleague to meet at a particular restaurant.
- 4) Google Calendar where the user can mark an event.

Table of Contents

List of Figures	v
List of Tables	vi
Acknowledgements.....	vii
Chapter 1 - Introduction.....	1
Chapter 2 - Motivation.....	1
Chapter 3 - Requirement Analysis	2
3.1 Requirement Specification.....	2
3.1.1 Functional Requirements	2
3.1.2 Software Requirements.....	2
3.1.3 Hardware Requirements.....	3
Chapter 4 - System Architecture and Design.....	4
4.1 System architecture.....	4
4.2 System Design	6
4.2.1 Use Case Diagram.....	6
4.2.2 Class Diagram	7
Chapter 5 - Android Framework Components.....	11
5.1 AndroidManifest.xml file	11
5.2 Activities.....	13
5.3 Intents.....	13
Chapter 6 – Implementation.....	15
6.1 Graphical User Interface.....	15
6.1.1 Home Screen	16
6.1.2 Login and Register Screen	17
6.1.3 Basic Search Results Screen	18
6.1.4 Advanced Search Screen.....	18
6.1.5 Restaurant Details Screen	20
6.1.6 Write Review and Show Reviews Screen.....	21
6.1.7 Send Invitation and Invitation List Screens	22

6.1.8 Invitation Details and Map view screen.....	22
6.1.9 Calendar Screen	23
6.1.10 Show Route Screen	24
6.2 Database.....	24
Chapter 7 - Testing.....	25
7.1 Unit Testing	25
7.2 Performance Testing.....	28
7.2.1 Home Screen.....	28
7.2.2 Search Screen.....	30
7.3 Compatibility Testing	31
7.3.1 Portrait mode.....	32
7.3.1 Landscape mode.....	32
Chapter 8 – Conclusion.....	33
Chapter 9 – Future Work	33
Chapter 10 – References	33

List of Figures

Figure 1 Android Architecture showing the major components of Android OS	4
Figure 2 Connection between PHP (server) and Android (client) using HTTP and JSON	5
Figure 3 Use Case Diagram	6
Figure 4 Class Diagram	7
Figure 5 Home Screen	17
Figure 6 Login Screen and Registration screen	17
Figure 7 Search Result	18
Figure 8 Advanced Search	19
Figure 9 Restaurant Details screen	20
Figure 10 Screens for writing and viewing reviews.	21
Figure 11 Send Invitation and Invitation List screens	22
Figure 12 Invitation Details and Map View screens.....	23
Figure 13 Calendar Screen.....	23
Figure 14 Show Route screen	24
Figure 15 Performance testing on home screen for 500 users	28
Figure 16 Performance testing on home screen for 1000 users	29
Figure 17 Performance testing on home screen for 2000 users	29
Figure 18 Performance testing on search screen for 500 users.....	30
Figure 19 Performance testing on search screen for 1000 users.....	30
Figure 20 Performance testing on search screen for 1500 users.....	31
Figure 21 Portrait mode	32
Figure 22 Landscape mode	32

List of Tables

Table 6.1 Lines of Code.....	15
Table 7.1 Test Cases for Unit Testing	27

Acknowledgements

I would like to take this opportunity to thank my major professor Dr. Daniel Andresen for his valuable guidance and encouragement throughout the project and my Masters program.

I would also like to thank Dr. Mitchell Neilsen and Dr. Torben Amtoft for graciously accepting to serve on my committee. It was a privilege to be able to take courses under you, which provided me with more practical guidance and experience. My special thanks to Dr. Mitchell Neilsen for providing the Android devices for testing of my project.

Finally, I would like to thank the administrative and technical support staff of the department of CIS for their support throughout my graduate study.

Chapter 1 - Introduction

Restaurant Finder is an Android application that helps the users to find the restaurants according to their requirements. The main aim of the project is to provide the users an easy to use application that helps them to find restaurants according to their tastes and convenience. The users can also give their opinions about the restaurants by giving a rating and writing a review. The Google Maps API shows the Top 5 restaurants present in the current city and also the route to the restaurants. The application also makes it easy for the user to send out invitations to his/her friends/colleagues for meetings and also mark the events in their calendar.

The motivation behind the project is discussed in Chapter 2; the requirement analysis is done in Chapter 3 and then followed by the system architecture and design in Chapter 4, Android Framework in Chapter 5, implementation in Chapter 6, Testing in Chapter 7, finally the conclusion and future work of the project.

Chapter 2 - Motivation

The main motivation of this application is to learn the mobile application development. I was always curious to know how things work in mobile. This led me to develop the Restaurant Finder Android application. There are few Restaurant Finder applications in the android market of which Yelp is the most popular one. Yelp deals not only with the restaurants but also many other businesses. It also acts like a social networking application which connects people doing various businesses. I started this application with the intention of making something simple and only related to restaurants.

Chapter 3 - Requirement Analysis

The project involved analyzing the design of few applications so as to make the application more user friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the android version had to be chosen so that it is compatible with most of the Android devices. Hence Android 2.3.3 Gingerbread version was chosen.

3.1 Requirement Specification

3.1.1 *Functional Requirements*

- Graphical User interface which the user
- Provide accessibility to the application through Wi-Fi or cellular network.
- MySQL that stores the information to be displayed to the user.

3.1.2 *Software Requirements*

For developing the application the following are the Software Requirements:

Operating System: Windows 7

Language: Android SDK, Java

Database: MySQL

Tools: Eclipse IDE, Android Plug-in for Eclipse

Technologies used: Java, MySQL, PHP, XML.

Debugger: Android Dalvik Debug Monitor service

Server: XAMPP Server

For running the application the following are the Software Requirements:

Operating System: Android 2.3 or higher versions

Network: Wi-Fi Internet or cellular Network

3.1.3 Hardware Requirements

For developing the application the following are the Hardware Requirements:

Processor: P IV or higher

RAM: 256 MB

Space on disk: minimum 512MB

For running the application:

Device: Android version 2.3 and higher

Minimum space to execute: 5.0MB

Chapter 4 - System Architecture and Design

4.1 System architecture

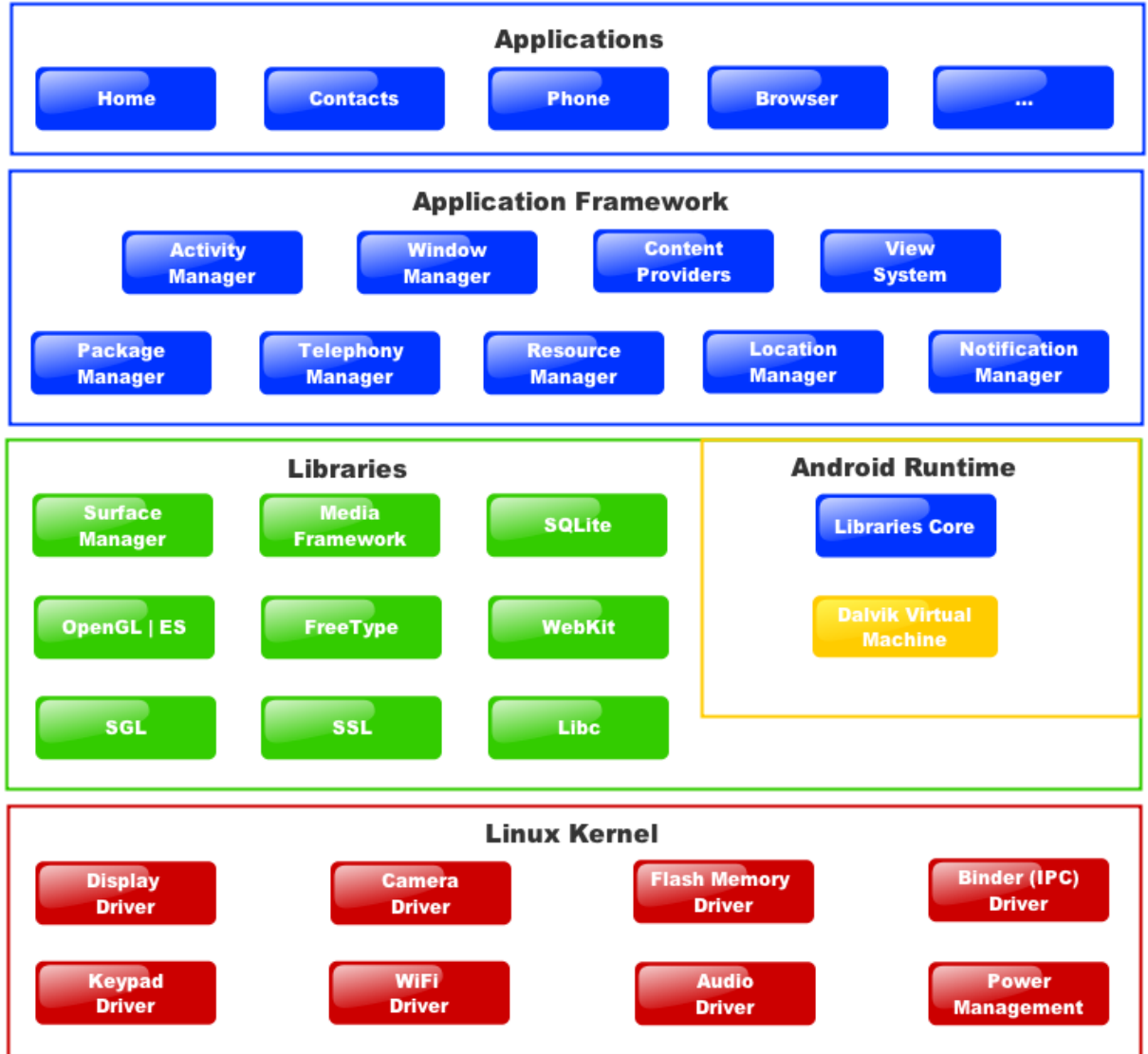


Figure 1 Android Architecture showing the major components of Android OS (Android Wiki, 2010)

The above figure [1] shows the diagram of Android architecture. Android OS is basically a software stack with various layers with each layer offering different services to the layer above it. The layers include a Linux Kernel which is responsible for interaction with the hardware. Libraries are written in C or C++ and are specific to the hardware of the system.

Android Runtime consists of Dalvik Virtual machine and Core Java libraries responsible for running the applications[2]. It is the Application Framework that our application interacts with.



Figure 2 Connection between PHP (server) and Android (client) using HTTP and JSON (<http://fahmirahman.wordpress.com/2011/04/21/connection-between-php-server-and-android-client-using-http-and-json/>)

The architecture shown in above figure [3] is used to show the search results, store the user details, user review and user invites. The data from android goes to webserver (PHP) to database server (MySQL). PHP is used here because of the interaction it can offer with the databases. On Android, HTTP protocol is used to connect with the webserver (PHP). JSON (JavaScript Object Notation) is a lightweight text-based open standard designed for human-readable data interchange and it is used in this application to send data from Android device to PHP Script.

When the application is executed, it connects the device to the PHP script on the server. PHP script fetches the response data which is encoded to JSON format and then sent back to the device. The data is parsed and displayed according to the requirement.

4.2 System Design

Once the features to implement were decided and the architecture was drafted, a system design was needed to implement the application. This section depicts the design using the Unified Modeling Language. The following were the three diagrams that are used to implement and test the application.

- Use Case Diagram
- Class Diagram

4.2.1 Use Case Diagram

A use case diagram captures the actors and the role they perform in a system. It depicts the roles performed by each actor. The two actors for this project are an *Unregistered User* and a *Registered User*.

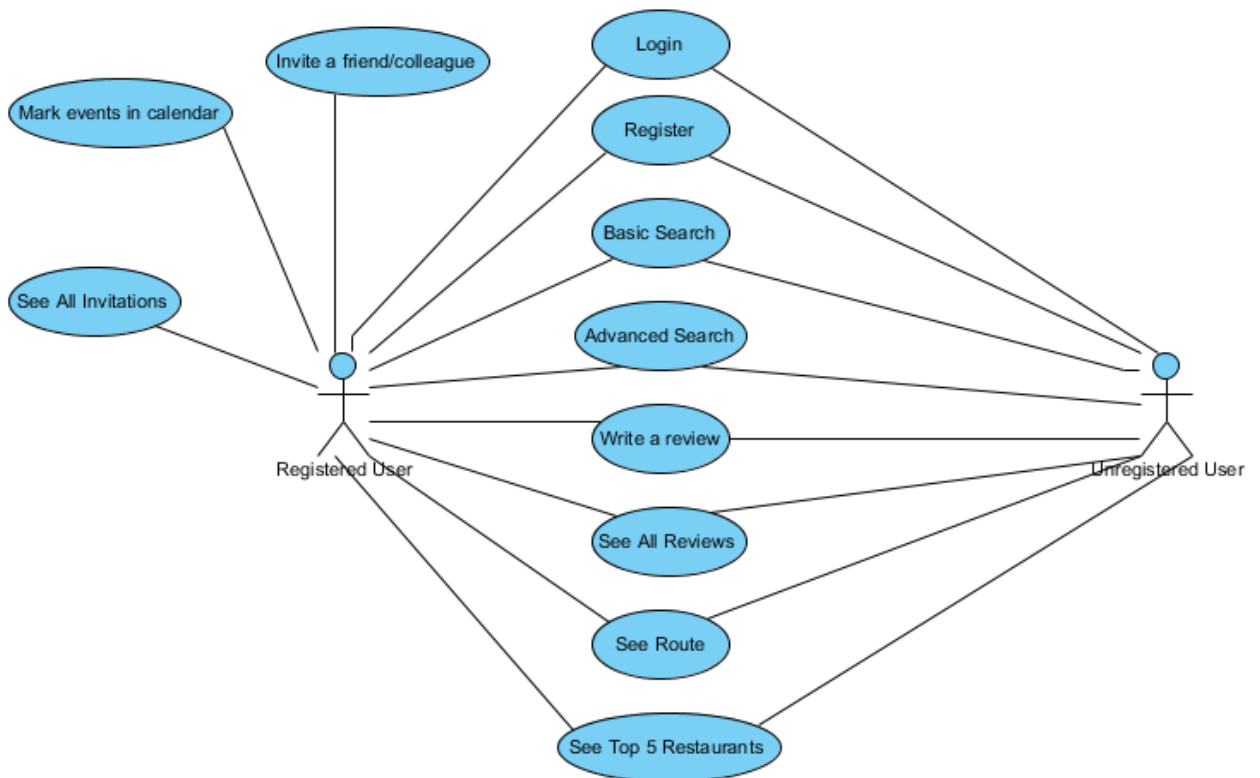


Figure 3 Use Case Diagram

4.2.2 Class Diagram

The class diagram below represents the important classes implemented for the application with their operations and return types. The diagram also shows how each class is associated with other classes in the system. Each activity is represented as a class and the attributes and functions associated with the class are represented within.

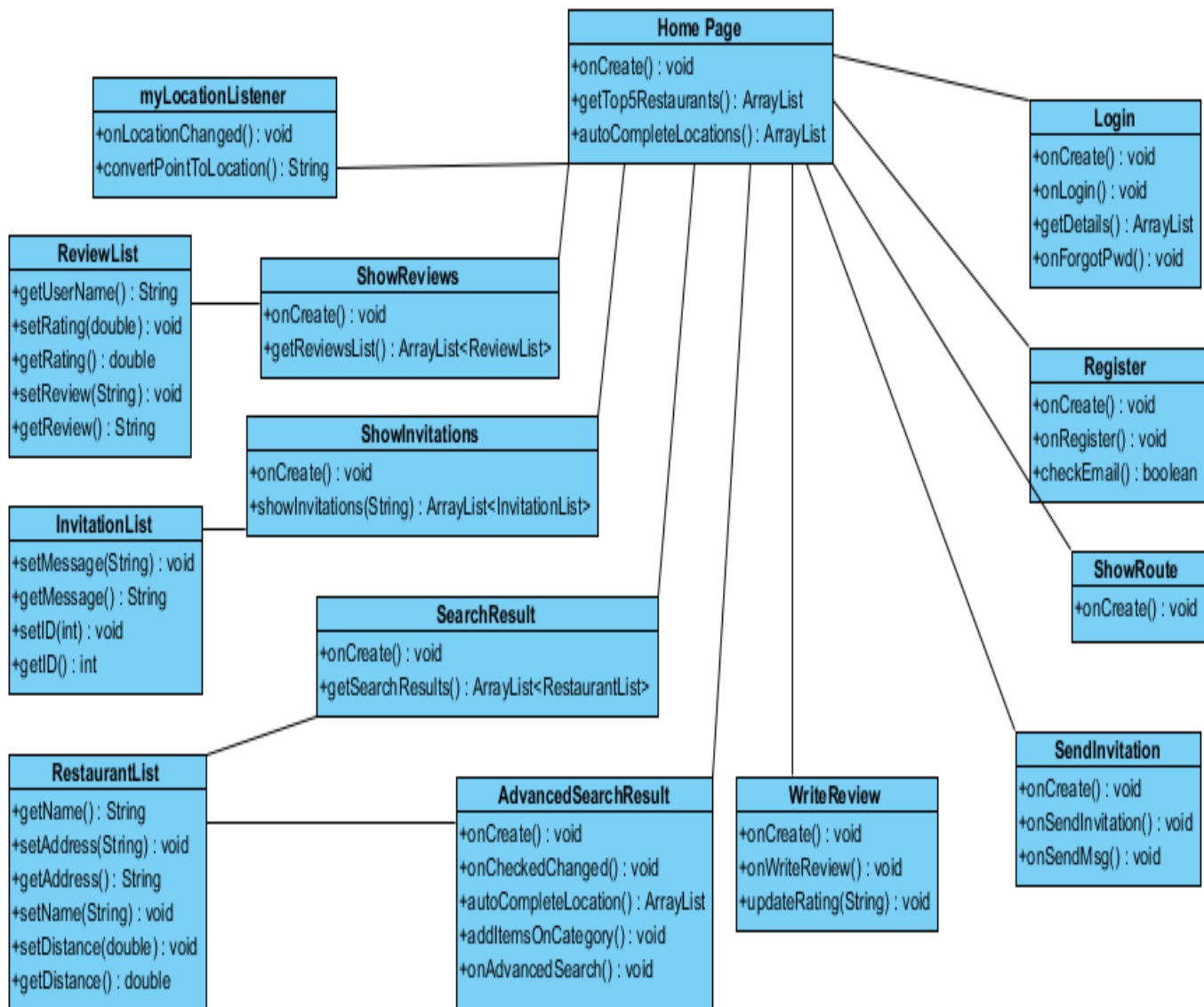


Figure 4 Class Diagram

RestaurantFinder Activity

This class is the start page of the application wherein the users' current location is tracked using the available Network Provider. Based on the current city, the top 5 restaurants are pin pointed on the Google Map.

The user can start searching for restaurants using the Basic (Keyword) search or the Advanced Search. The user can also login or register with the application.

Login Activity

This class enables the user to login to the application so as to view the invitations received. The user can login by entering the username and password. If the user enters the wrong credentials he/she is not allowed to login. In case the user forgets the details, they can be sent to his email address if he clicks on "Forgot password?" and gives a valid email address.

Register Activity

This class lets the user to register with the application. The user has to register by giving a username, a valid password and an email address. Each registered user will have a unique username. All the user details are stored in a database located on a server. This enables the user to login from any device as the data is centrally located on a server. Certain constraints have been set on the password and the email is checked for a valid format.

SearchResult Activity

This activity displays all the restaurants which are the result of the basic keyword search. All the restaurants are represented in a List View with their Name and First Line of the address shown.

AdvancedSearchResult Activity

This activity accepts the input from the user where the user can select category of the restaurant, the rating and the distance range. The user has an option of choosing the current address as source or entering a different address.

AdvancedSearchList Activity

This activity displays the list of restaurants which result from the users' input from the AdvancedSearchResult Activity. The user can see the name of the restaurant, the distance from the source address and also the address of the restaurant.

RestaurantDetails Activity

This activity shows the details of the selected restaurant such as name, address, phone number, rating and number of reviews. In this activity the users also have an option seeing the review, writing a review, seeing the route. The registered user has a privilege of viewing and sending out invitations.

SendInvitation Activity

This activity enabled the user to send invitations for a meeting at a particular restaurant. The user can select the date and time from a Date Picker and a Time Picker respectively.

ShowReviews Activity

This activity helps user in seeing all the reviews of a particular restaurants written by various users.

WriteReview Activity

In this class the review given by a user for a particular restaurant is stored in a centralized database.

ShowRoute Activity

In this class the user can view the route to a particular restaurant from the selected source with the help of Google API. This can be seen either in Maps application or in the web browser according to the users' choice.

InvitationList Activity

This activity displays all the invitations that a user received in the form of a list view. The user also has an option of seeing the meeting places in a map view.

InvitationDetails Activity

This activity displays the details of the invitation such as the sender's name, the restaurant name where the meeting has been scheduled, the address and the subject. The user also has an option of marking the event in their personalized calendar present on the device. This is done with the help of calendar intent.

In this application the Restaurant information, the user details, the reviews and invitations posted by the user are stored in a database located on a centralized server. This helps the user in using the application from any android device as it is not personalized.

Chapter 5 - Android Framework Components

In this section, the various components of an android application such as activity, intent etc are discussed

5.1 AndroidManifest.xml file

Every android application must have an AndroidManifest.xml file in its root directory. This file lists out all the activities, intents, intent-filters, permissions etc the application user. This file is responsible for providing all the information about the application to the Android system. It also has the minimum SDK version needed to run the application. The permissions to access the internet, write onto a calendar etc are also declared in this file.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.rednif.RestaurantFinder"
    android:versionCode="1"
    android:versionName="1.0" >

<uses-sdk android:minSdkVersion="10" android:targetSdkVersion="8"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_CALENDAR"></uses-permission>
<uses-permission android:name="android.permission.READ_CALENDAR"></uses-permission>

<application
    android:icon="@drawable/icon"
    android:label="@string/app_name"
    android:debuggable="true" >

    <activity
        android:name=".RestaurantFinderActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category
                android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>

    <activity
        android:name=".Login"
        android:label="@string/app_name" >
    </activity>
```

```
<activity
    android:name=".Register"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".About"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".SearchResult"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".Contact"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".AdvancedSearchResult"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".ShowRoute"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".RestaurantDetails"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".AdvancedSearchList"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".WriteReview"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".ShowReviews"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".SendInvitation"
    android:label="@string/app_name" >
</activity>
```

```

<activity
    android:name=".ShowInvitations"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".InvitationDetails"
    android:label="@string/app_name" >
</activity>

<activity
    android:name=".ShowMapView"
    android:label="@string/app_name" >
</activity>

<uses-library android:name="com.google.android.maps" />

</application>

</manifest>

```

Since the application uses Google Maps API to show the top 5 restaurants and to show the route to a particular restaurant, the library has been mentioned.

5.2 Activities

An activity provides a means of interaction to the user. It provides a window where the UI can be designed according to the window. Almost all activities interact with the users.

This application has the following activities: RestaurantFinder Activity, Login Activity, Register Activity, SearchResult Activity, AdvancedSearchResult Activity, AdvancedSearchList Activity, RestaurantDetails Activity, WriteReview Activity, ShowReviews Activity, ShowRoute Activity, SendInvitation Activity, ShowInvitations Activity and InvitationDetails Activity

5.3 Intents

The activities are activated through messages called intents. It basically has information that is required by the component. It helps in launching the activity or to do something with the existing activity. An intent passed to `startActivity()` is delivered only to an activity. The intent object holds the name of the component that handles the event, the action to be performed, data on which the operation is performed, the key value pairs to send some additional information. The intents that are used in this application are `ACTION_SEND` and `ACTION_EDIT`. The Restaurant Finder Application makes use of intents to send the users'

details or users' suggestions/opinions in an email and also to mark the events in Google Calendar.

Intent to send an email:

```
Intent email = new Intent(Intent.ACTION_SEND);
email.putExtra(Intent.EXTRA_EMAIL, new String[]{ to});
email.putExtra(Intent.EXTRA_SUBJECT, subject);
email.putExtra(Intent.EXTRA_TEXT, message);
email.setType("message/rfc822");
startActivity(Intent.createChooser(email, "Choose an Email client :"));
```

Intent to create a new calendar event:

```
Calendar beginTime = Calendar.getInstance();
    beginTime.set(year, month-1, day, hours, min);
    startMillis = beginTime.getTimeInMillis();
    Calendar endTime = Calendar.getInstance();
    endTime.set(year, month-1, day, hours+1, min);
    endMillis = endTime.getTimeInMillis();
```

```
Intent l_intent = new Intent(Intent.ACTION_EDIT);
l_intent.setType("vnd.android.cursor.item/event");
l_intent.putExtra("title", "Invitation from "+uname.getText().toString());
l_intent.putExtra("description", rdesc.getText().toString());
l_intent.putExtra("eventLocation", rname.getText().toString()+"\n"+raddress.getText().toString());
l_intent.putExtra("beginTime", startMillis);
l_intent.putExtra("endTime", endMillis);
l_intent.putExtra("allDay", 0);
l_intent.putExtra("eventStatus", 1);
l_intent.putExtra("visibility", 0);
l_intent.putExtra("transparency", 0);
l_intent.putExtra("hasAlarm", 1);
startActivity(l_intent);
```

Chapter 6 – Implementation

The Restaurant Finder application is designed from a user point of view. The user friendly design helps the users in accomplishing their task with ease. Attempts have been made to keep the design simple and understandable.

The screens were designed in XML and the business logic was written in Java. The database used is MySQL where all the information related to restaurants and users is stored. The web service used to connect MySQL database with Android is PHP. The PHP files are located on the XAMPP Server. Google Maps API has been used to make it easy for the user to figure out the top restaurants and to find the route to a particular restaurant.

The total lines of code written in this application is

Language	LOC
Java	4061
PHP	332
XML	$(1594*2) + 150 = 1744$

Table 6.1 Lines of Code

Debugging of the application throughout the development is done using Dalvik Debug Monitor Server (DDMS). DDMS provides port-forwarding services, screen capture on the device, thread and heap information on the device, logcat, process, and radio state information, incoming call and SMS spoofing, location data spoofing etc.[1]. DDMS is also used to verify the location based services implemented in the application.

6.1 Graphical User Interface

The user interface is kept simple and understandable. The user need not take any additional effort to understand the functionality and navigation in the application. The colors are chosen in such a way that user can easily understand where the input has to be given. Hints are given to help the user in giving the correct input.

The following are the main screens and features in this application.

- Home Screen
- Login Screen
- Registration Screen
- Basic Search Results
- Advanced Search Input Screen
- Advanced Search Result Screen
- Restaurant Details Screen
- Show Reviews Screen
- Write a Review Screen
- Send Invitation Screen
- Invitation List Screen
- Invitation Details Screen
- Show Route Screen
- Calendar Screen

6.1.1 Home Screen

The Home page is the starting page of the application. It has a google map that shows the top 5 restaurants in the current city. The navigation to the pages like About Us, Contact Us, Login , Register can be done from this page. The basic search and advanced search also can be done from this screen

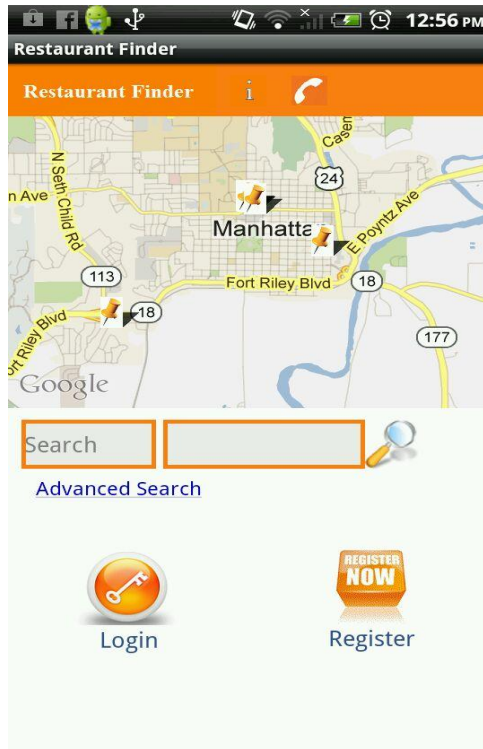


Figure 5 Home Screen

6.1.2 Login and Register Screen

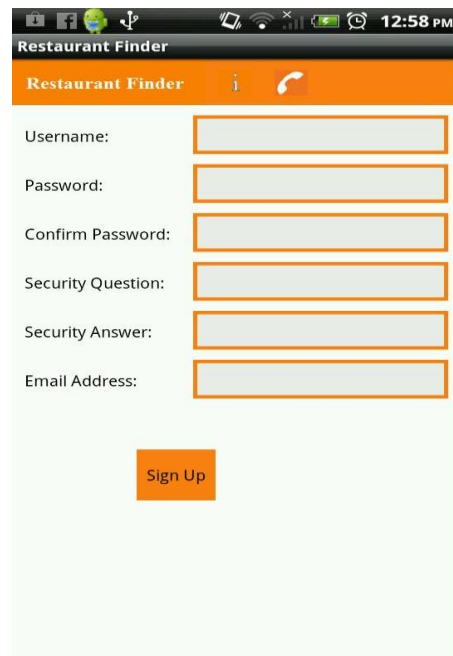
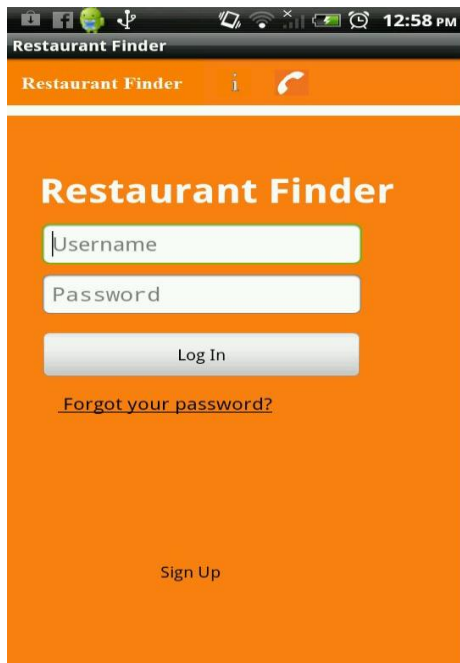


Figure 6 Login Screen and Registration screen

The user can login and register from the above screens.

6.1.3 Basic Search Results Screen

In the basic search result the user can see the list of restaurants and the addresses of the restaurants.

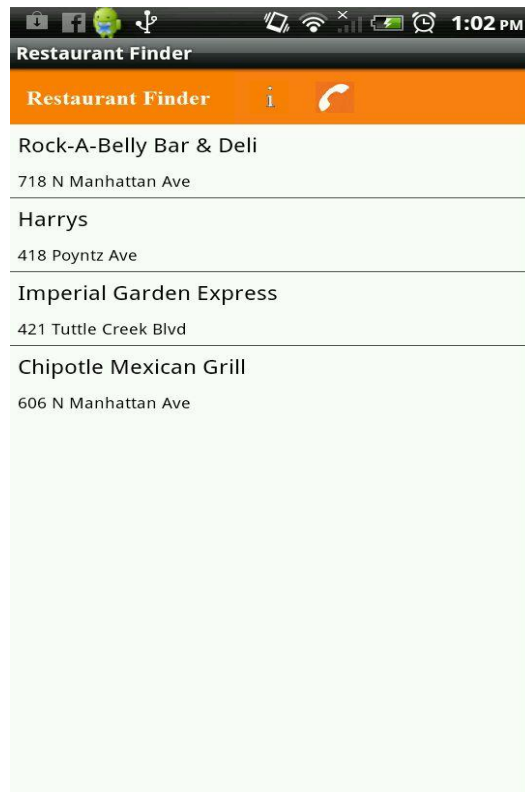


Figure 7 Search Result

6.1.4 Advanced Search Screen

In the Advanced Search screen the user can select a category, rating and distance range from a list of choices. The user also has an option of choosing the current address as starting point or can give some other address. In the advanced search result screen the user can view the list of restaurants, the distance and the address.

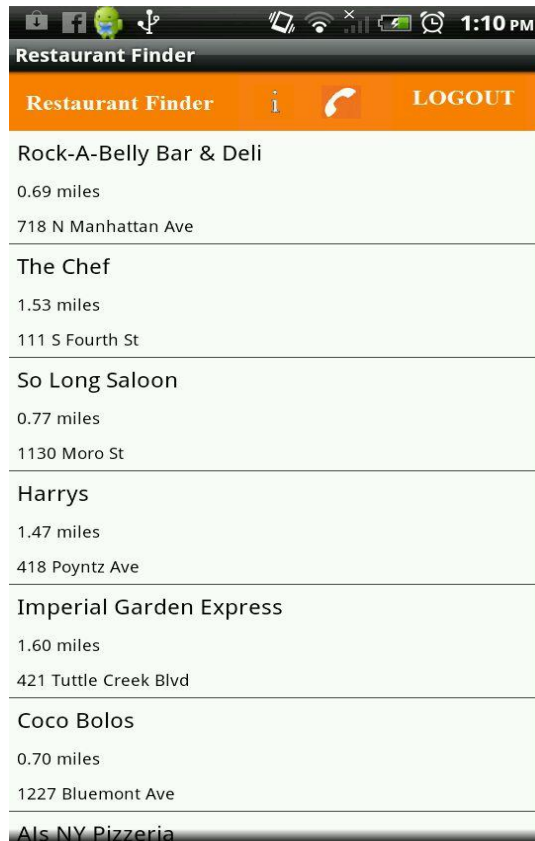
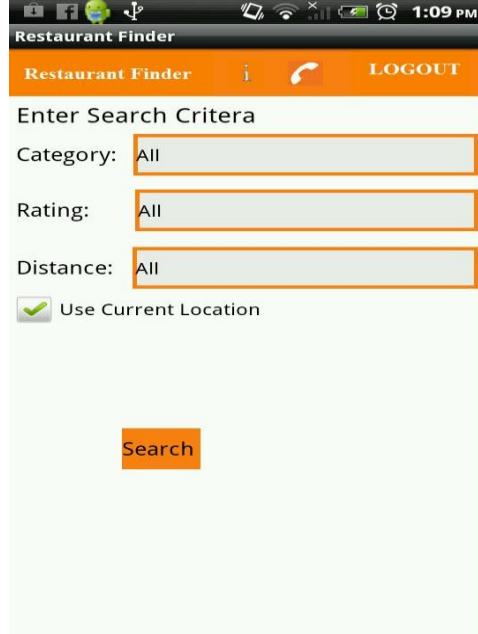
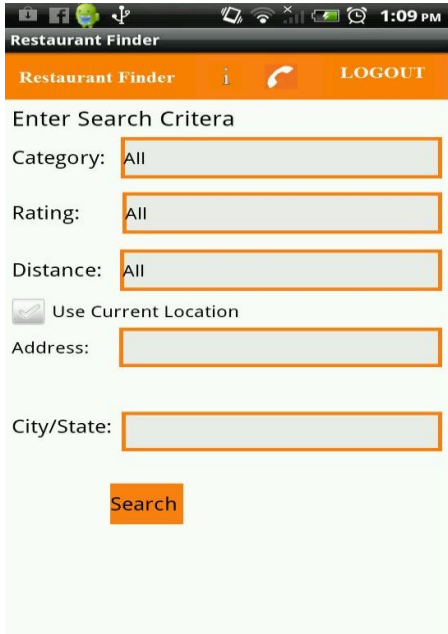


Figure 8 Advanced Search

6.1.5 Restaurant Details Screen

The restaurant details screen shows the details of the selected restaurant.

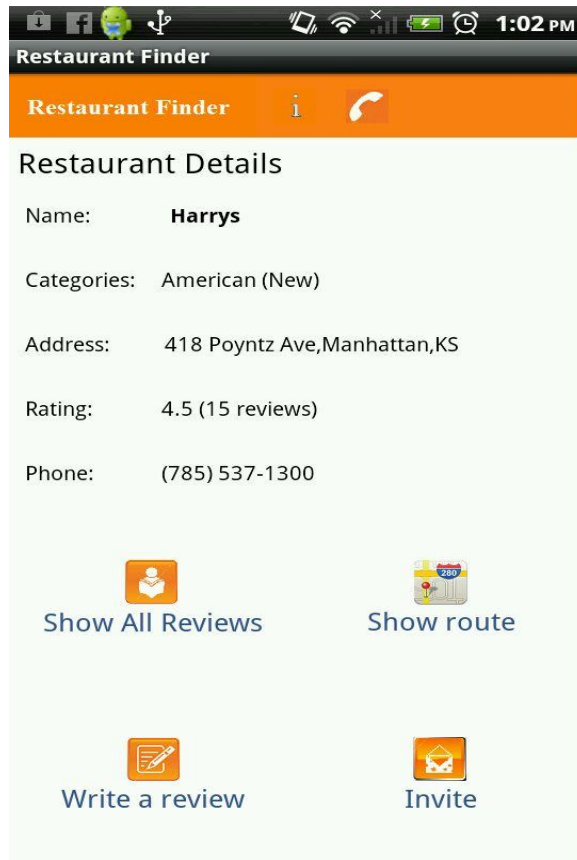


Figure 9 Restaurant Details screen

6.1.6 Write Review and Show Reviews Screen

The user can write a review and see all the reviews of a selected restaurant from these screens. The review includes the rating and an opinion about the restaurant.

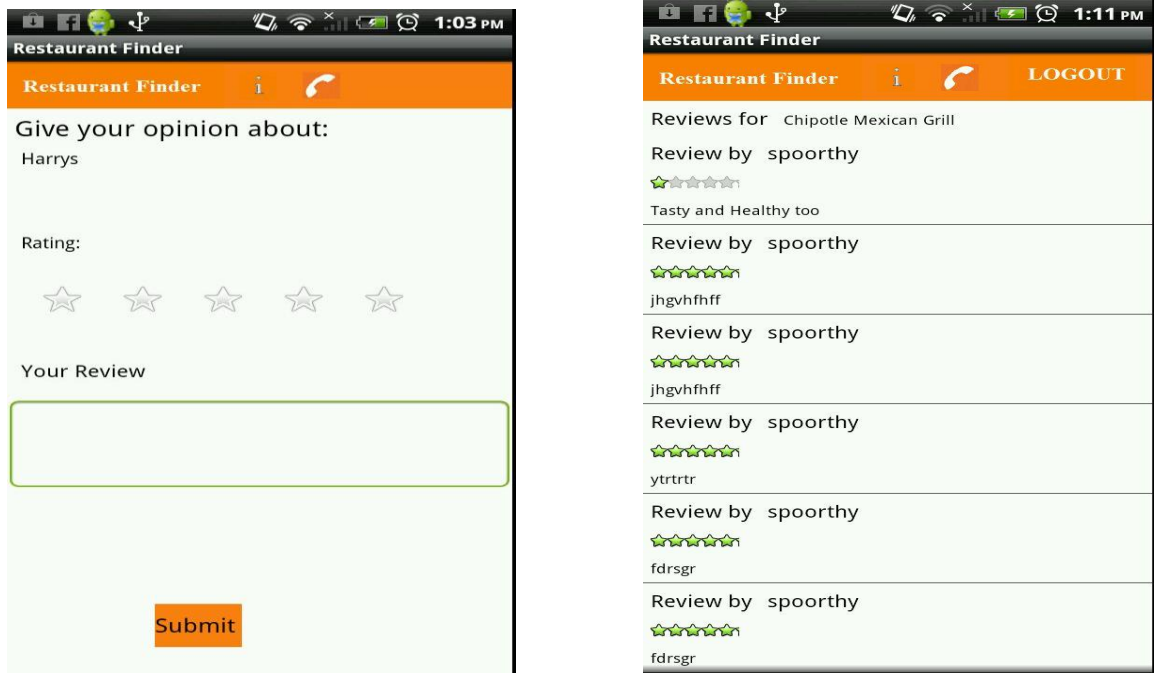


Figure 10 Screens for writing and viewing reviews.

6.1.7 Send Invitation and Invitation List Screens

The following are the screens for inviting a friend/colleague and for viewing the list of invitations along with the sender's name, location, date, time and purpose of meeting. Only register users can use this feature of this application.

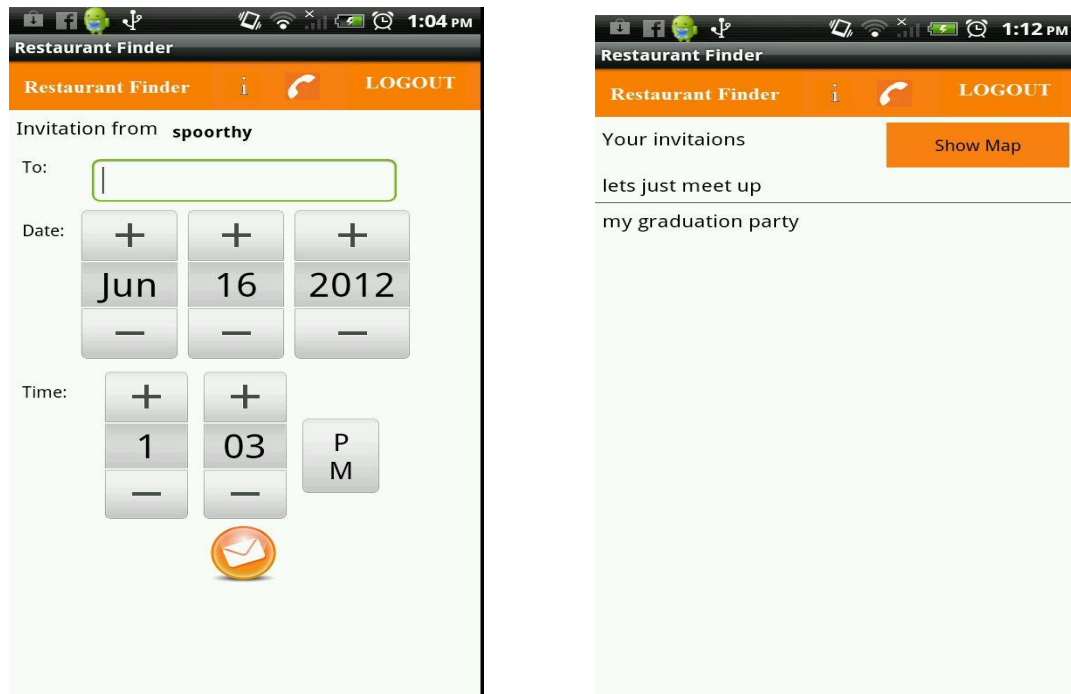


Figure 11 Send Invitation and Invitation List screens

6.1.8 Invitation Details and Map view screen

The details of the invitation can be seen in the map by clicking on the location or normally by selecting an invitation from the list.

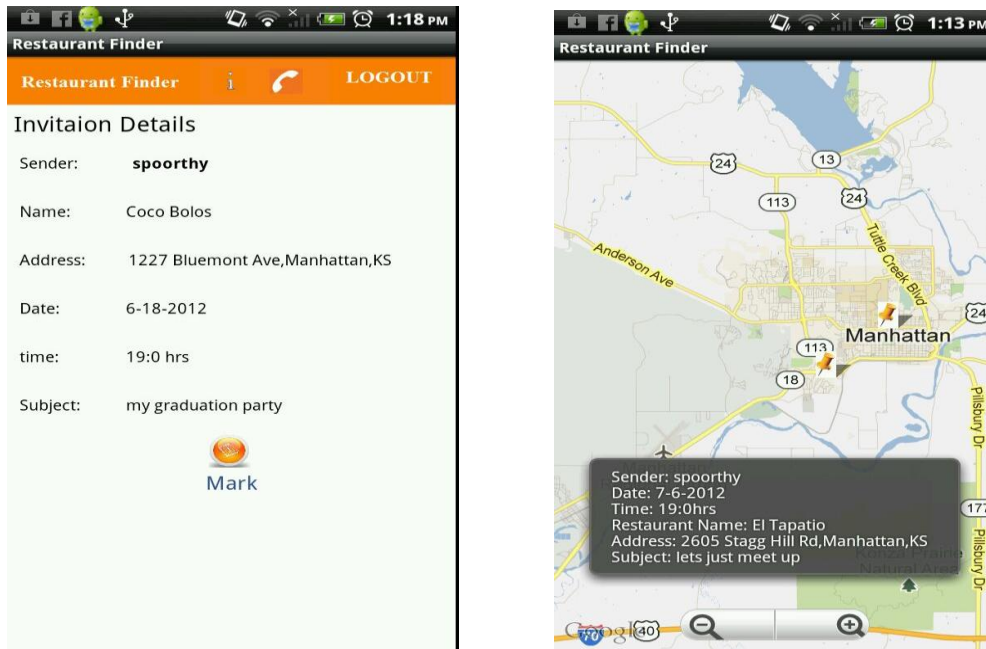


Figure 12 Invitation Details and Map View screens

6.1.9 Calendar Screen

The user can also mark the invitations in his personal calendar by clicking the mark button on the invitation details screen. The subject, date, time and location will be filled automatically in the calendar.

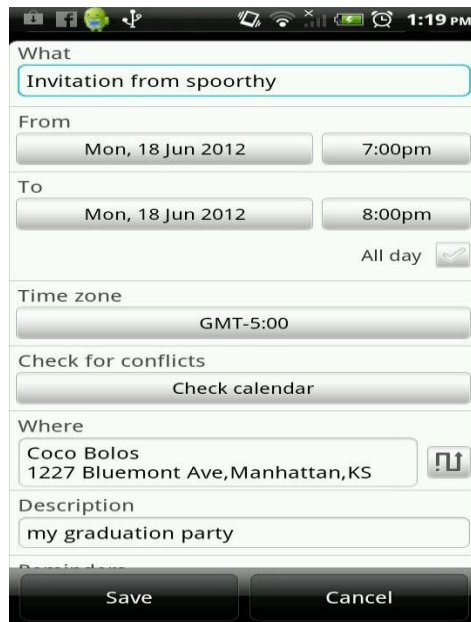


Figure 13 Calendar Screen

6.1.10 Show Route Screen

For the basic search an alert box will pop up where the user have to give a start address but foe advanced search it immediately redirects him to the google map.

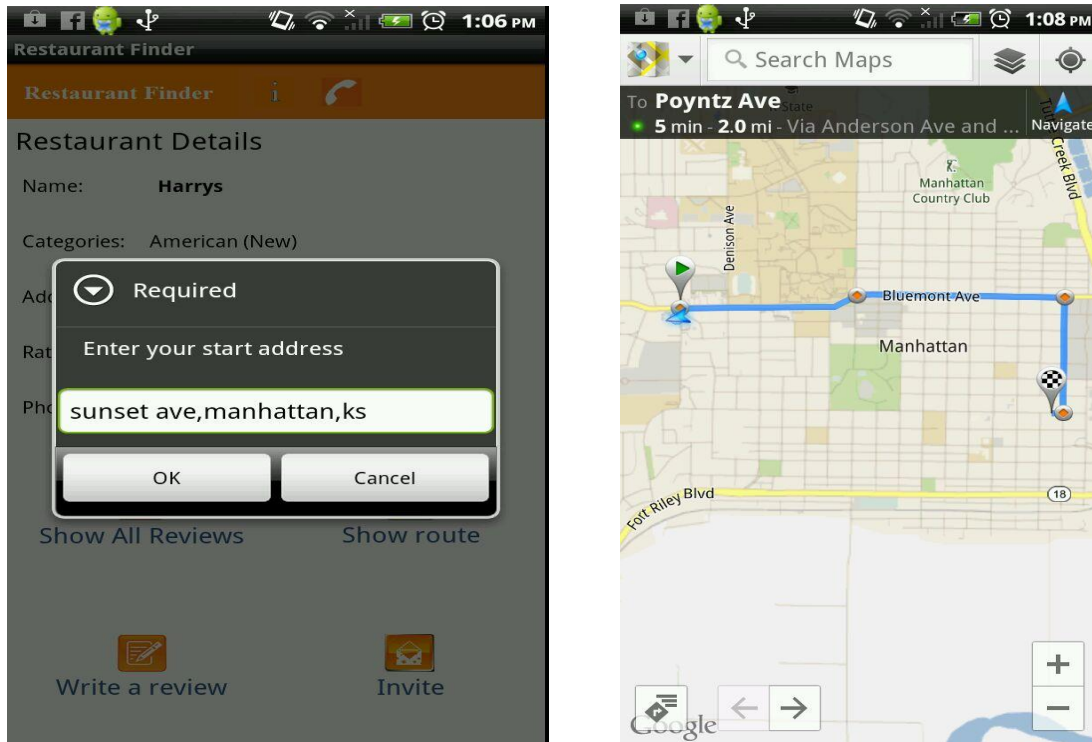


Figure 14 Show Route screen

6.2 Database

The database used for this application is MySQL which is present on a central server. The main table which has the information of the restaurant was populated by crawling www.yelp.com. Then the pages have been parsed to extract the restaurant information. Swing Parser has been used to parse the simple HTML pages of Yelp

Chapter 7 - Testing

7.1 Unit Testing

In unit testing, various modules have been tested individually. This has been done manually to test if the expected result is actually seen on the screen. The following are test cases with the help of which the application has been tested.

Sr. No	Test Case Description	Expected Result	Actual Result
1	On load of home screen	Show Top 5 restaurants in the current city	Pass
2	On clicking search image button without any user input	Display an alert box prompting the user about an invalid input	Pass
3	On clicking search image button with user input	Display a list of restaurants present in the specified location and that match the keyword.	Pass
4	On Clicking search button without any user input	Display an alert box prompting the user to either choose the current location or enter an address	Pass
5	On Clicking search button	Display a list of restaurants	Pass

	with complete user input	according to the requested category, rating and distance range	
6	On Clicking a particular restaurant in the list	Display a screen showing the details of the selected restaurant	Pass
7	On Clicking the submit button on Write a Review page	Update the rating and number of reviews in the database	Pass
8	On Clicking “Show Reviews” button on Restaurant Details page	Display a list of all the reviews written by users about the selected restaurant	Pass
9	On Clicking “Show Route” button on Restaurant Details screen	Show the route to the selected restaurant from the specified source address	Pass
10	On Clicking Invite button on Restaurant Details screen without logging in.	Navigate to the login screen.	Pass

11	On Clicking “Invite” button on Restaurant Details Screen after logging in	Show a screen where the user can send invitations to multiple users.	Pass
12	On Clicking “See Invitations” button on Restaurant Details Screen without logging in.	Navigate to the login screen.	Pass
13	On Clicking “See Invitations” button on Restaurant Details Screen after logging in.	Display a list of invitations that the logged user received.	Pass
14	On Clicking an invitation	Display the details of the invitation like the sender name, meeting place, time and the subject.	Pass
15	On Clicking “Mark” button on Invitation Details screen	Open the google calendar with the details populated.	Pass

Table 7.1 Test Cases for Unit Testing

7.2 Performance Testing

Performance testing has been done to measure the responsiveness of the application to the workload such as increasing users' requests. JMeter was used to create the users and to analyze the performance. The parameters were chosen randomly till the application performed consistently. The test has been done on a home Wi-Fi network with a speed of 5Mbps

System Configuration:

Operating System: Windows 7

Processor: Intel® Core™ i5-2450M CPU @ 2.50GHz

RAM: 6GB

The home page was tested by analyzing the performance of the PHP page which is responsible for returning top 5 restaurants in the current city by tracking the current location of the user through Wi-Fi, Cellular network or GPS. It was really important to have a good response time for this screen as it is the home page.

7.2.1 Home Screen

The graph below explains the characteristics of the PHP page for 500 user requests.

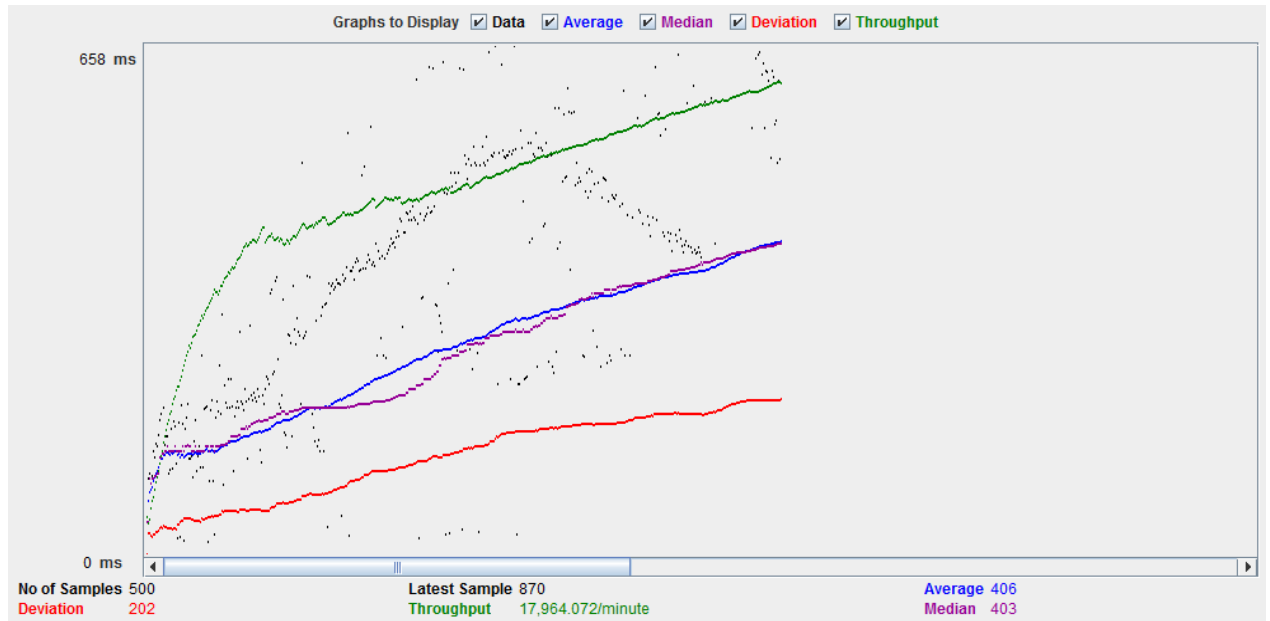


Figure 15 Performance testing on home screen for 500 users

It is clear from the graph that the application performs consistently for 500 users as it has an average response time of 0.4 sec.

The following are the graphs for 1000 users and 1500 user requests on the home page

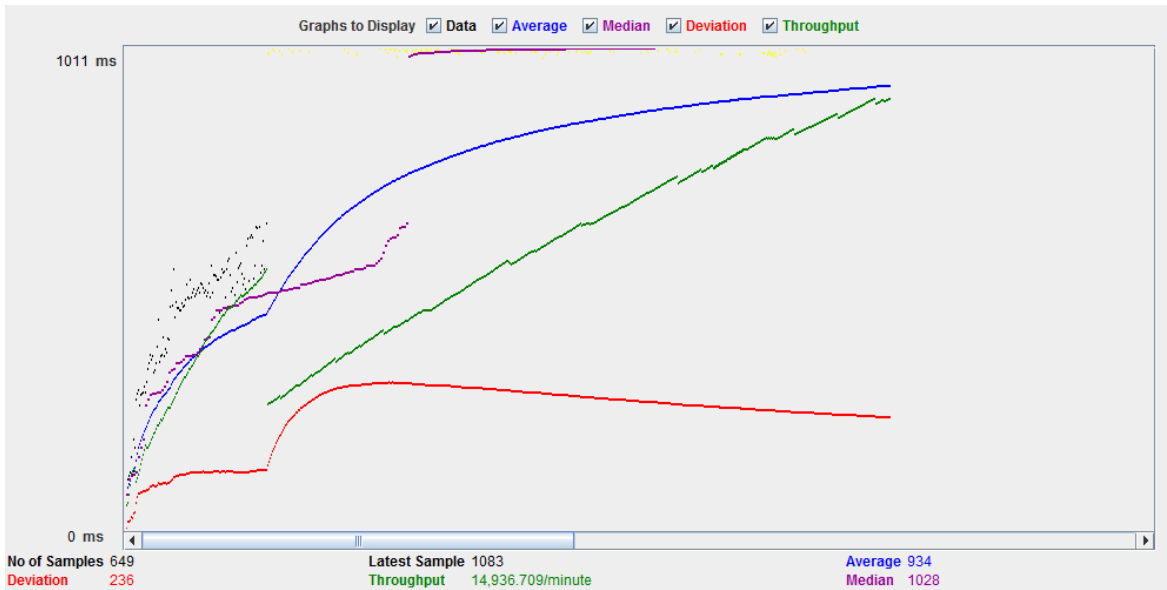


Figure 16 Performance testing on home screen for 1000 users

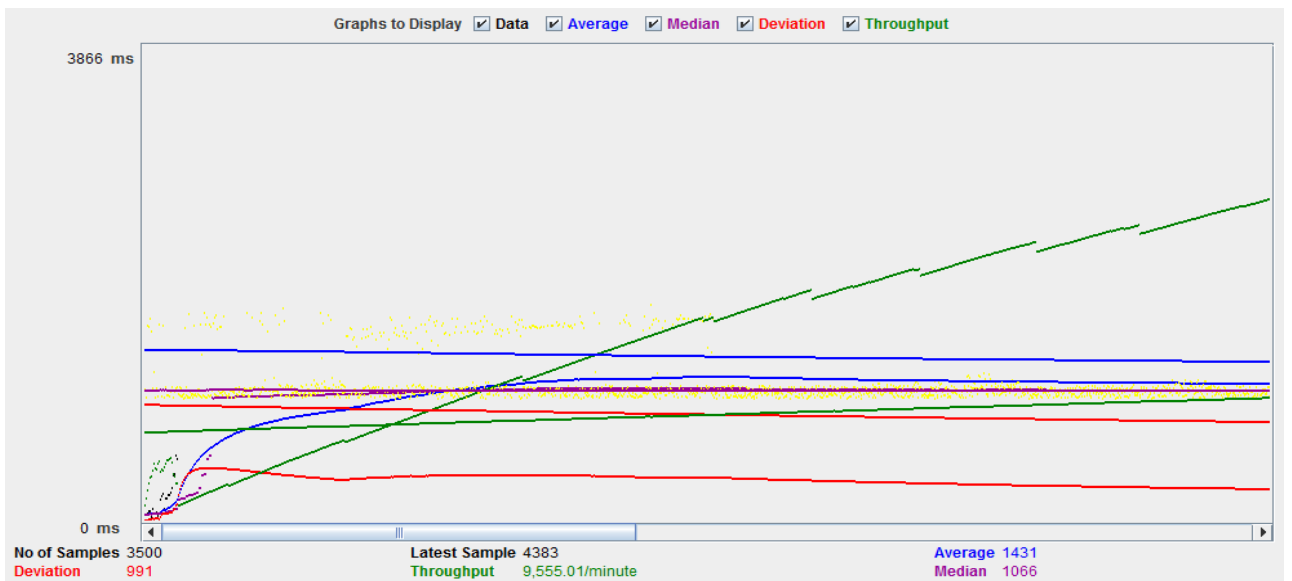


Figure 17 Performance testing on home screen for 2000 users

The application performs consistently for 1000 users as the average response time is 0.9 sec but it might not respond quickly enough for 2000 or more users as the average response time is as high as 1.4 sec.

7.2.2 Search Screen

The graph below explains the characteristics of the PHP page for 500, 1000 and 1500 user requests.

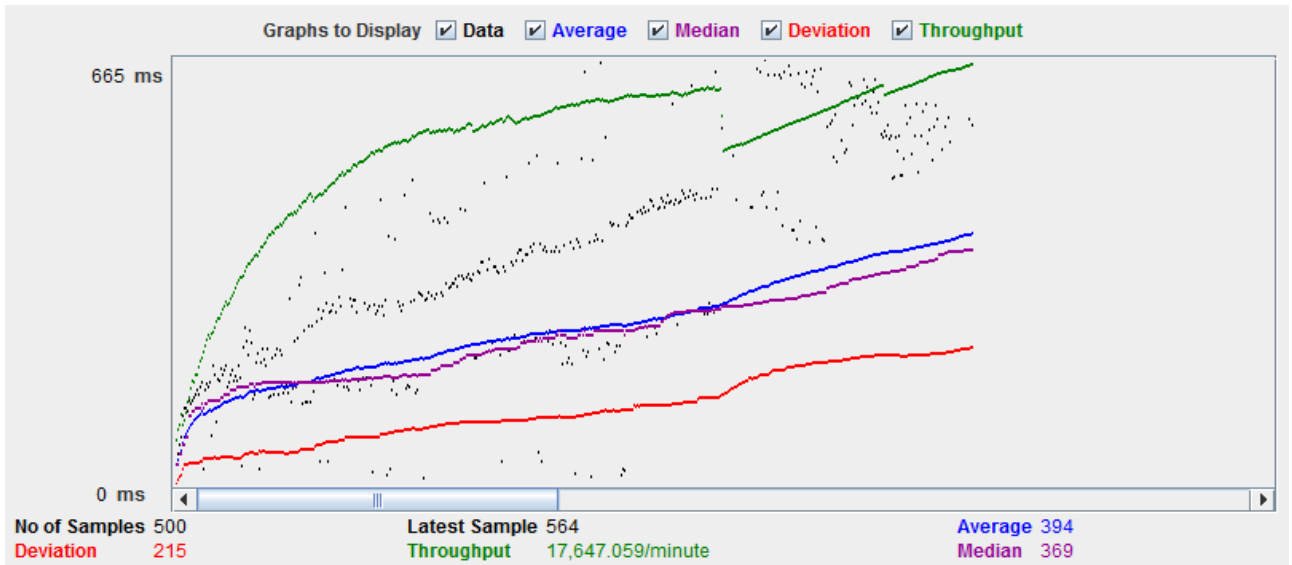


Figure 18 Performance testing on search screen for 500 users

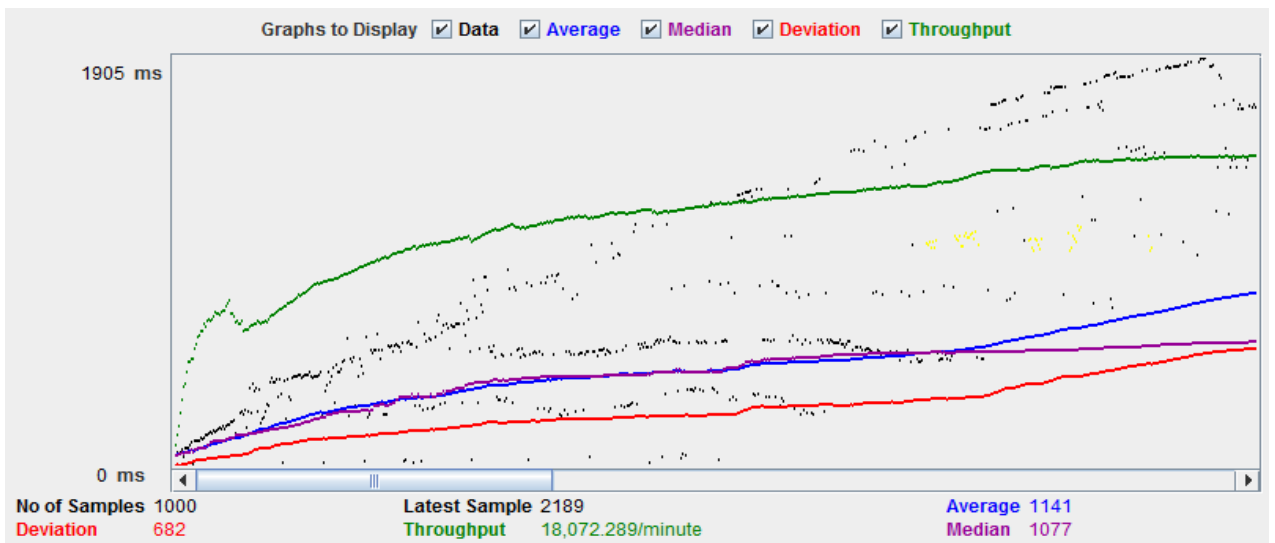


Figure 19 Performance testing on search screen for 1000 users

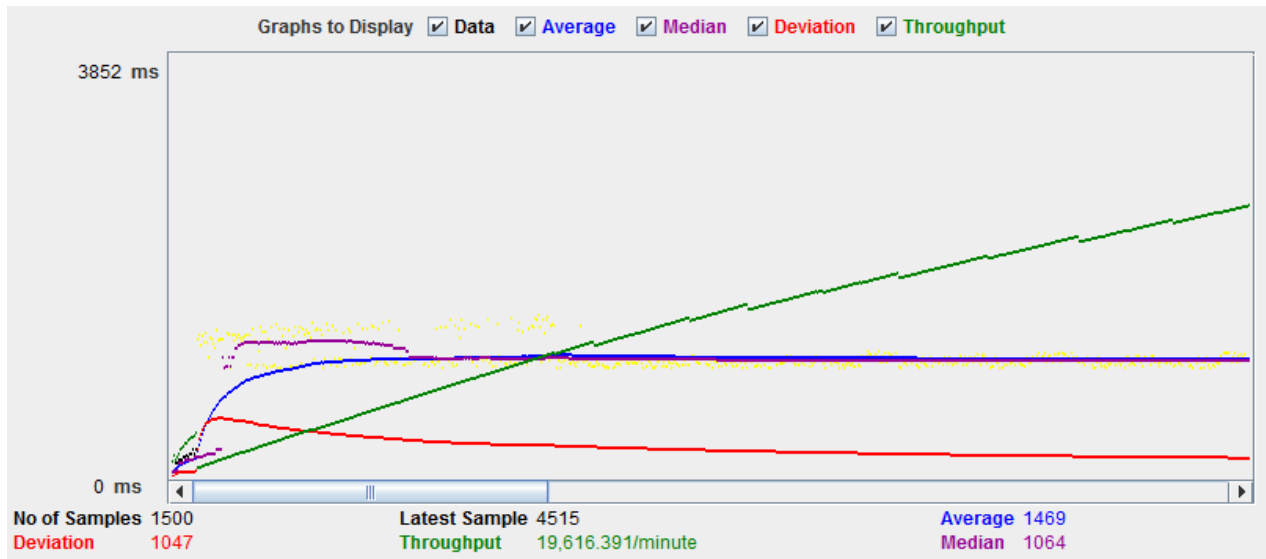


Figure 20 Performance testing on search screen for 1500 users

From the above graphs it can be observed that the applications performs consistently for 1500 user requests with an average response time of 1.4sec and a throughput of almost 20,000 requests per minute.

7.3 Compatibility Testing

This application was mainly designed for android phones as it helps the users find the restaurants when they are on the move. Generally they try to carry something handy like cell phones with them and not the tablets. Different android phones have different screen sizes and resolution. The application has been tested for its compatibility with different screen sizes on the emulator.

7.3.1 Portrait mode

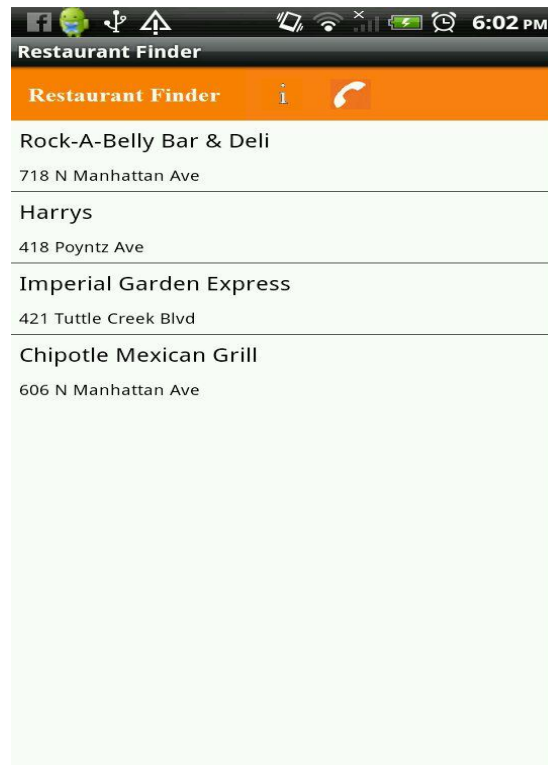


Figure 21 Portrait mode

7.3.1 Landscape mode

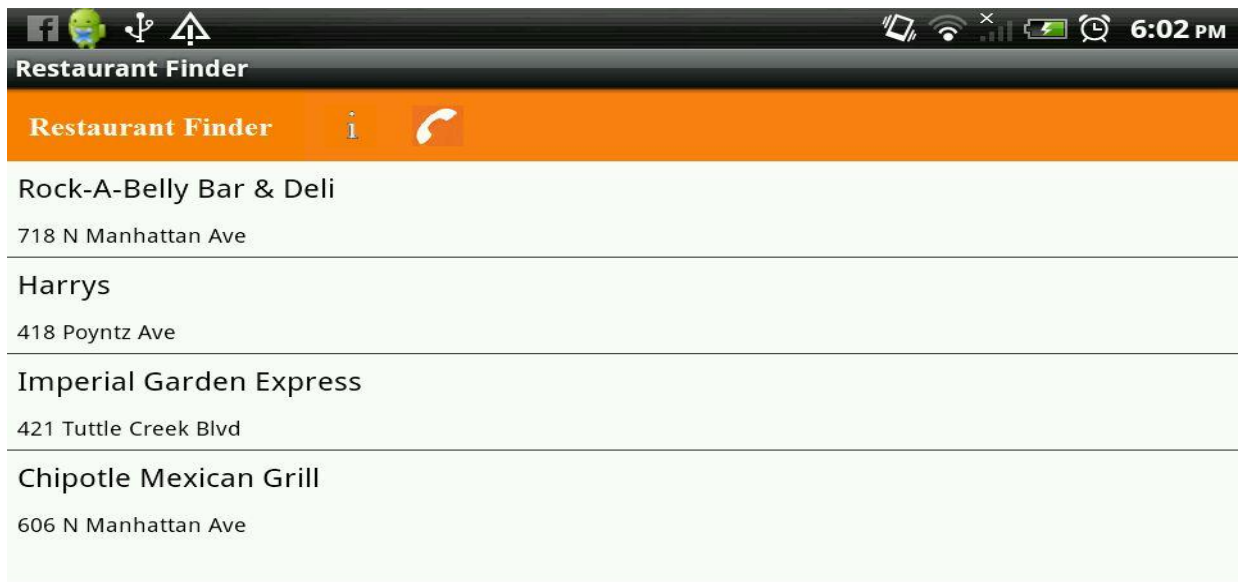


Figure 22 Landscape mode

Chapter 8 – Conclusion

This is my first attempt in developing a mobile application which gave me a basic understanding of development and challenges of mobile application development. The main aim of the project is to provide an easy to use application for searching the restaurants. The application has been implemented and tested on real devices.

Chapter 9 – Future Work

The application can be improved in many ways and can be extended to support more devices like the tablets and iOS devices. Following are some of the possible extensions:

- The application can be extended to provide a more advanced search by taking the zip code as the input.
- For the selected distance range, all the restaurants in the distance range can be displayed irrespective of the city.
- There can also be ways to directly navigate the user to the restaurant's site or provide a way to call the restaurant when clicked on the phone number.

Chapter 10 – References

[1] Android Architecture

http://en.androidwiki.com/wiki/Diagram_of_the_Android_architecture

[2] Android Architecture Description

<http://www.android-app-market.com/android-architecture.html>

[3] Connection between PHP (server) and Android (client) Using HTTP and JSON

<http://fahmirahman.wordpress.com/2011/04/21/connection-between-php-server-and-android-client-using-http-and-json/>

[4] Android Developer Guide

<http://developer.android.com/>

[5] Android Coding

<http://stackoverflow.com/>

[6] Crawling and Parsing

<http://www.oracle.com/technetwork/java/index.html>

[7] Source of restaurants information

<http://www.yelp.com/>