

# **Palestine Polytechnic University**



**College of engineering and technology**

**Electrical and computer department**

**Graduation project**

## **A tourist guide of the old town of Hebron using smart phone**

**Project team:**

**Sara Halahla**

**Ohoud abu sebaa**

**Safa' ideas**

**Project supervisor:**

**Eng. Radwan Tahboob**

**Hebron-Palestine**

**May, 2013**

# Dedication

*To Palestine Polytechnic University, and all our teachers.*

*To our parents, brothers, and sisters*

*To our all friends and colleagues*

*To all, whom raising us to be the persons we are today, whom being with us in every step of the way, through good times and bad, thanks you for all the support that you have always give us, helping us to succeed and instilling in us the confidence that we are capable of doing anything we put in our minds.*

*Thank you for everything.*

## Acknowledgements

*Special thanks to Palestine Polytechnic university for its attention, support, and chances that give us to be graduated students, and improve our selves and abilities to be engineering.*

*We would like to thanks our supervisor eng.AymanWazwaz for his support and advice during our works, and we would also like to thanks all teachers who help us in the project, and also thanks to all teachers in electrical and computer engineering department.*

*Special thanks to our parents, families, friends, and anyone help us, and trusted deeply that we can success.*

*To each of the above, we extend our deepest appreciation.*

## **Abstract**

With the recent and fast developments in technology, smart phones offer many advanced services rather than making calls, so the number of smart phone users increases day by day. User can install and run many smart phone applications that make life easier or enjoyable.

One of the areas that users can benefit from smart phone applications is tourism. Many tourists visit a region or city in limited time which may not be enough to visit everywhere. So that tourists are obliged to select attractive spots. Usually they make this selection by searching web sites, magazines, or guidebooks. After the selection is made, the tourist decides the trip route which can take a lot of time. Tourists face several difficulties acting this way: information in guidebooks is mostly not up-to-date, spots are (partly) closed due to renovation and some timetables (museum, for example) are unavailable.

An application that can provide valuable information to guide the tourists in their trip can be so beneficial. The goal of this application is to provide tourism and geographic services through mobile phone application that encourage tourists to travel around the city and reduce efforts and shorten the time to organize the city trip.

## الملخص

مع التطورات الأخيرة والسريعة في مجال التكنولوجيا، والهواتف الذكية تقدم العديد من الخدمات المتطورة بدلا من إجراء المكالمات، وبالتالي فإن عدد مستخدمي الهواتف الذكية تزداد يوما بعد يوم. يمكن للمستخدم تثبيت وتشغيل العديد من تطبيقات الهواتف الذكية التي تجعل الحياة أسهل أو ممتعة.

واحدة من المناطق التي يمكن للمستخدمين الاستفادة من تطبيقات الهواتف الذكية هي السياحة. العديد من السياح يقوم بزيارة منطقة أو مدينة في وقت محدود قد لا يكون هذا الوقت كافيا لزيارة كل مكان. يكون السائح مهتما في اختيار أماكن جذابة من اجل زيارتها. عادة يقوم باختيار هذه الأماكن من خلال البحث في المواقع على شبكة الإنترنت، والمجلات أو في كتب الأدلة السياحية. بعد أن يحدد السائح هذه الأماكن تبدأ المعاناة في تحديد مسار الرحلة الذي يمكن أن يستغرق الكثير من الوقت. حيث يواجه السياح العديد من الصعوبات في تحديدها. إن المعلومات الواردة في كتب الأدلة السياحية هي في معظمها غير حديثة حيث يوجد فيها معلومات عن أماكن مغلقة بسبب التجديد وبعض مباني (المتحف، على سبيل المثال) غير متوفرة.

يهدف هذا التطبيق على توفر معلومات قيمة ومفيدة لتوجيه السياح في رحلتهم. ويهدف كذلك في توفير الخدمات السياحية والجغرافية. ويهدف على تشجيع السياح في التجوال في جميع أنحاء المدينة وتقليل الجهود واختصار الوقت لتنظيم جولة في المدينة.

# **CONTENT**

Title Page.....	I
Dedication.....	II
Acknowledgments .....	III
Abstract .....	IV
Content .....	VI
Glossary.....	XI
List of Figures .....	XII
List of Tables .....	XIII
CHAPTER ONE.....	1
1.1 Overview.....	2
1.2 Problem statement.....	2
1.3 Aim of project.....	2
1.4 Requirements.....	3
1.4.1 Functional requirements.....	3
1.4.2 Non-Functional requirements.....	3
1.5 Assumption and dependencies .....	4
1.6 Literature Review (related projects).....	5
1.6.1 World Mat.....	5
1.6.2 Field trip.....	5
1.6.3 Google map.....	5
1.7 Project schedule .....	6
1.8 General system description.....	7
1.9 Budget of project.....	8
1.9.1 Hardware Resources.....	8
1.10 Project risk management.....	9
1.10.1 Risk Analysis.....	9
1.10.2 Risk Effects Minimization Strategies.....	9

1.11	Report content.....	10
CHAPTER TWO.....		12
2.1	Overview.....	13
2.2	GPS Technology.....	13
2.2.1	Definition.....	13
2.2.2	GPS components.....	14
2.2.3	How it works?.....	15
2.2.4	The GPS satellite system.....	16
2.2.5	GPS in our project.....	16
2.3	Coordinate systems .....	17
2.3.1	World Geodetic System 1984 (WGS84) .....	17
2.3.2	Earth Gravitational Model (EGM96) .....	17
2.3.3	Earth Gravitational Model (EGM2008) .....	17
2.3.4	North American Datum of 1927 (NAD27) .....	17
2.3.5	North American Datum of 1983 (NAD83) .....	18
2.3.6	Coordinate systems in our project.....	18
2.4	GSM Technology.....	18
2.4.1	Definition.....	18
2.4.2	GSM transmission and frequency allocation.....	19
2.4.3	GSM network infrastructure.....	20
2.4.4	GSM Network Architecture.....	21
2.4.4.1	The Mobile Station (MS).....	21
2.4.4.2	The Base Station Subsystem (BSS).....	21
2.4.4.3	The Network and Switching Subsystem (NSS).....	22
2.4.4.4	Operation and Support Subsystem (OSS) .....	23
2.4.5	GSM In Our Project.....	23
2.5	GPRS service.....	24
2.5.1	Definition.....	24
2.5.2	GPRS and packet data in GSM.....	24
2.5.3	GPRS Network Elements.....	26
2.6	Mobile phone.....	26

2.6.1	Mobile Operating systems (mobile software platform).....	27
2.6.1.1	Android operating system .....	27
2.6.1.2	Android architecture.....	28
2.6.1.3	Android features.....	30
2.7	Programming languages .....	30
2.7.1	Java architecture.....	31
2.7.2	Java development environment (E_clips).....	32
2.7.3	Java features.....	32
CHAPTER THREE.....		33
3.1	Overview.....	34
3.2	General system block diagram.....	34
3.2.1	Mobile.....	34
3.2.2	GSM network.....	34
3.2.2.1	Why GPRS? .....	35
3.2.3	Server.....	35
3.3	System flowchart.....	36
3.3.1	Fill registration data and get coordinates.....	37
3.3.1.1	Fill registration data.....	37
3.3.1.2	Get coordinates.....	38
3.3.2	Send data and coordinates to servlet server.....	39
3.3.3	Receive data and coordinates by servlet server.....	40
3.3.4	Tourist enters the system and uses the system options.....	41
3.3.4.1	Route.....	41
3.3.4.1.1	Dijkstra's algorithm.....	42
3.3.4.1.2	Find friend.....	42
3.3.4.1.3	Food.....	44
3.4	Design options.....	44
3.4.1	Mobile.....	44
3.4.1.1	Android OS.....	44
3.4.1.2	Galaxy Note 2.....	45



3.4.2	Geographic maps.....	45
3.4.2.1	Google earth.....	46
3.4.2.2	Image of map.....	46
3.4.2.3	Google map.....	46
3.4.3	Connection Options.....	47
3.4.3.1	GPRS mobile with PC interface.....	47
3.4.3.2	HttpURLConnection Connection.....	48
CHAPTER FOUR.....		49
4.1	Overview.....	50
4.2	System classes model diagram.....	50
4.2.1	Mobile software UML diagram.....	50
4.2.1.1	Classes implementation.....	50
4.2.2	Server software UML diagram.....	58
4.2.2.1	Classes implementation.....	59
4.3	System use cases.....	61
4.4	System sequence diagram.....	63
4.4.1	Fill and send registration sequence diagram.....	63
4.4.2	Send GPS coordinates sequence diagram.....	64
4.4.3	Find friend sequence diagram.....	65
CHAPTER FIVE.....		67
5.1	Overview.....	68
5.2	Sub-System Testing.....	68
5.3	Installation and preparing the system.....	68
5.3.1	Mobile application.....	68
5.4	Testing Scenarios.....	76
5.4.1	TEST 1.....	77
5.4.2	TEST 2.....	79

CHAPTER SIX.....	80
6.1 Overview.....	81
6.2 Conclusion and Achievement.....	81
6.3 Challenges.....	81
6.4 Future work Recommendation.....	82
REFERANCES.....	83

## **GLOSSARY**

<b>Abbreviation</b>	<b>Definition</b>
<b>GPS</b>	<b>Global Position System</b>
<b>GSM</b>	<b>Global System for Mobile communications</b>
<b>OS</b>	<b>Operating system</b>
<b>GPRS</b>	<b>General Packet Radio Services</b>
<b>SDK</b>	<b>Software development kit</b>

## LIST OF TABLES

Table 1.1	Timing plane for first semester.....	7
Table 1.2	Hardware resource cost.....	8
Table 1.3	Risk analysis.....	9
Table 1.4	Minimizing the risk.....	10
Table 2.1	Data rate for GPRS software.....	25

## LIST OF FIGURES

Figure 1.1	System description.....	8
Figure 2.1	Earth-circling satellites orbits.....	14
Figure 2.2	GSM Band Spectrums.....	19
Figure 2.3	GSM Network's Components.....	23
Figure 2.4	GPRS network elements.....	26
Figure 2.5	Architecture of Android OS.....	28
Figure 2.6	Java main architecture .....	31
Figure 3.1	General system block diagram.....	34
Figure 3.2	Main system flowchart.....	37
Figure 3.3	Get coordinates flowchart .....	38
Figure 3.4	Send data and coordinates flowchart.....	39
Figure 3.5	Server received `data and coordinates flowchart.....	40
Figure 3.6	Dijkstra's algorithm pseudocode.....	42
Figure 3.7	Find Friend flowchart.....	43
Figure 4.1	Mobil UML diagram.....	52
Figure 4.2	Server UML diagram.....	59
Figure 4.3	System use case diagram.....	62
Figure 4.4	Send registration request sequence diagram.....	64
Figure 4.5	Get and send GPS coordinates sequence diagram.....	65
Figure 4.6	Find Friend sequence diagram.....	66
Figure 5.1	Opening apk file.....	69
Figure 5.2	Installing complete.....	69
Figure 5.3	Welcome screen.....	70
Figure 5.4	Registration.....	70

Figure 5.5	Old Town map and options.....	71
Figure 5.6	Famous Restaurant Town.....	72
Figure 5.7	Find Friend.....	73
Figure 5.8	Name of online users.....	73
Figure 5.9	User enter your friend name.....	74
Figure 5.10	Update location figure.....	75
Figure 5.11	Map shown static route.....	76
Figure 5.12	First GPS reading.....	78
Figure 6.1	Image map with GPS coordinates.....	82

# 1

## *Chapter One*

### *Introduction*

---

- 1.1 Overview**
- 1.2 Problem statement**
- 1.3 Aim of the project**
- 1.4 Requirements**
- 1.5 Assumption and dependencies**
- 1.6 Literature Review(related projects)**
- 1.7 Project schedule**
- 1.8 General system description**
- 1.9 Budget of project**
- 1.10 Project risk management**
- 1.11 Report content**

## **1.1 Overview**

In this chapter, we will introduce the main idea for our project, the aim we try to achieve by this system, the main requirement that will be needed in our work, some of the assumption and dependencies that we will rely on, we also made some literature review, and we mention some of these project that are related to our work and show how we will be different from there, we also show general block diagram for our system.

## **1.2 Problem statement**

The old town of Hebron contains many architectural landmarks, such as the Ibrahimi mosque, a museum, shops, and historical places. Who's the tourist doesn't know any information about it, and how to reach it. So the tourist may be getting lost.

Tour groups, it is possible members of the same group go to different places; leading to loss of time in the search for each other and cause they did not know places the rest of the group.

## **1.3 Aim of project**

- Encourage the tourism in the old town of Hebron.
- Help the tourist to access the old town by displaying map where plotted the road how to reach old town.
- Give the tourist information about historical location.
- Help tour group to be in touch with each other during their journeys.
- Eliminate the Judaization of the old town by providing tourist historical facts of the old town.



## **1.4 Requirements**

The requirements of our project will be described in two parts the functional and nonfunctional requirements as follow:

### **1.4.1 Functional requirements:**

- Our system will have stored information about historical locations of old town of Hebron, and map to guide tourist.
- The system will have a server; the server stores name, Id, and real time coordinate of each user login the system.
- The system must, track tourist mobile in real time, over its travel along given time, this will require hardware to receive the GPS coordinates, and send it through communication network to the server.
- The system must be able to receive coordinates from more than one tourist.
- The system will offer information about important services in old town of Hebron.

### **1.4.2 Non-Functional requirements:**

- Usability, the system will be used from smart phone's owner, and company's owners, so it must be usable and easy to use.
- Availability, the system will be able for any person wants to use, and also it should be available at any place during person travelling, so we rely on GPS system that have wide spread in our countries, and if some region does not have this we try to solve this limitation.
- Reliability, the system will be reliable, so it can give you a proper location of two tourists but our system may have an error rate, if one tourist is indoor and the other outdoor location, but it will be reduced because we have GSM network which may reduce the error effects.

And for all these requirements to be achieved we will need the following hardware and software:

**Hardware:**

- Mobile phones with GPS and GSM /GPRS services.
- Main server with internet connection.

**Software:**

- Mobile programming language as (JAVA, XML Programming).
- Programming software for map display.
- Graphical user interface (GUI).
- Database for maps and history.

## **1.5 Assumption and dependencies**

The following assumptions are considered in our project:

- Each person travelling to Hebron, old town having smart phone support GPS, and GPRS services he will enjoy with our application because it's only for Hebron, old town so it will make travelling into it easily.
- The application will be useful as a great tourist guide, but can also be used to dig up some information on historical events or places near you. Basically, Hebron-old town trip tracks down your exact travel route by monitoring your GPS coordinates, and find friends.

## **1.6 Literature Review(related projects)**

Many projects around the world have similar points to this project, but we try to be different, and make it suitable for our country, with reasonable cost.

And here we will mention some similar projects that related to our work:

### **1.6.1 World Mat**

The application allows confirming the booking for the flight, hotel reservations, car rental via e-mail messages and displays the Google map plans every point the user in addition to the path sync with Google calendar. [1]

### **1.6.2 Field trip**

Google launched this application solely for Android 2.3 and above, which is a tourist guide so that gives the user information and dates for the site important visited and show this information without any movement from the user, and This application works as an automatic, once approaching the user of the exact location appear on the screen of a new page showing information about the place. [2]

### **1.6.3 Google map**

This allows the application to download the map without a data connection, and be removed automatically after 30 days of the maps, and maps are downloaded. This application does not allow the user to search for points of interest or get directions. [3]

Our system differs from systems previously mentioned. Because it specifically for the old city of Hebron, so that maps are previously stored in, it will gives him information about the important places he visited.

## **1.7 Project Plan**

- **Stag1: Select the idea**

Determine the idea of the project, the motivation, and the main objective we intend to achieve.

- **Stage2: Preparing for the project**

In this stage, more and deeper determination of the tasks, and steps we want to perform, is done.

- **Stage3: Project Analysis**

In this step, a study of the all possible design options to determine our own design.

- **Stage4: Determine the project requirement**

After determine our design scheme, we specify the entire needed requirement for the user and the system, software and hardware. And try to bring them to be ready for the implementation stage.

- **Stage5: Studying the Principles**

This stage of the project is necessary to study the GSM, GPS, GPRS, mobile programming language, main server programming language.

- **Stage6: Documentation Writing**

Documenting the project will begin from the first stage to the last stage.

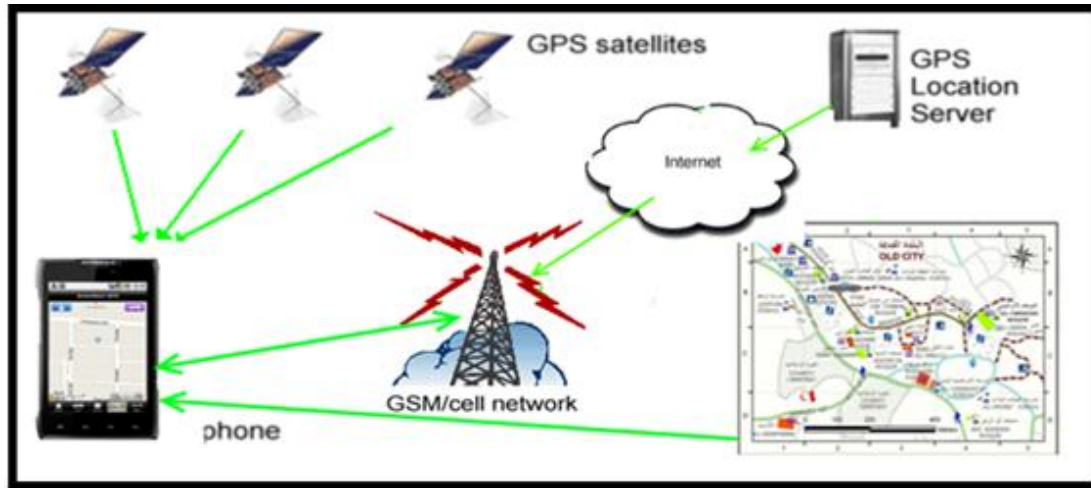
These stages represented in table 1.1:

**Table 1.1: Timing plane for first semester**

Task/ week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	■	■													
S2			■	■											
S3				■	■	■									
S4						■	■	■	■						
S5				■	■	■	■	■	■	■	■	■	■	■	■
S6					■	■	■	■	■	■	■	■	■	■	■

## 1.8 General system description

From the general description for our project idea, and main requirements, we will need major parts to achieve our goals, the hardware which used to receive the GPS signals from the satellites, and send coordinates to the server through GSM/GPRS network, the server will convert these coordinates to be suitable to our city coordinates, then sending information according to GPS location coordinates, and then using software to display tourist movement on the map. Figure 1.1 shows a main block diagram of the project parts.



**Figure 1.1 System description**

## **1.9 Budget of project**

In this section we will preview the system cost estimation in term of system Hardware.

### **1.9.1 Hardware Resources**

The following table 1.2 contains the main requirements for our system and their costs.

**Table 1.2 Hardware resources cost**

<b>Hardware Component</b>	<b>Cost</b>
1 PC with CPU 2.6 GHz, 40GB HDD, 512k RAM, monitor, keyboard and mouse	\$800
2 smart phones (GALAXY SI)	\$900
Printing Cost	\$100
<b>Total</b>	<b>\$1800</b>

## 1.10 Project Risk Management

The project may face some problem and risk that we have to declare in the early time of the project designing and manipulation, and we must avoid those problems so that the system could work in its high efficiency, so when we find a risk we will try to solve it without the total project as much as we can.

### 1.10.1 Risk Analysis

The following table shows the risks their probability and effects on the system development.

**Table 1.3 Risk analysis**

<b>Risk ID</b>	<b>Risk Description</b>	<b>Probability</b>	<b>Impact</b>
1	One of the group members dies or get sick.	10%	Critical
2	Lack of experience in using programming tools.	30%	Marginal
3	Israel may block internet service.	20%	Catastrophic
4	Israel may close the old town	10%	Critical
5	Users will not like the project; because it attacks their privacy.	30%	Critical

## 1.10.2 Risk Effects Minimization Strategies

Table 1.4 shows some of strategies we will take them to minimize the risks.

Table 1.4 Minimizing the risk

Risk ID	How to minimize
1.	The team must divide the work, if one is absent the other does his work.
2.	Read and study more on using these programming.
3,4, 5	The other risks can't be avoidable, so when they occur we will be in serious problem. We hope that none of them will happen.

## 1.11 Report Content

This report includes the following three chapters:

- **Chapter one: Introduction.**

In this chapter, we introduce a general overview about the project, the aim idea of the project, requirements, assumption and dependences, the time schedule, and the estimated cost.

- **Chapter two: Theoretical Background.**

This chapter includes the Theoretical background related to the main idea of the project, and technologies, hardware, and software used in the project.



- **Chapter three: Project Conceptual Design.**

In this chapter we talk about the design concepts; the general block diagram that show how the system works, the system main components, system flowcharts and their related options.

- **Chapter four: Detailed Design.**

In this chapter we talk about the detailed software engineering design including UML model diagram for both mobile and server application software classes. Then system use case diagram will be described, also the sequence diagram for use cases will be mentioned.

- **Chapter five: System Testing.**

In this chapter, the whole testing stage will be described, that's mean a test of an entire interconnected set of components and software for the purpose of determining proper functions and achieving the desired goals of the system. We will talk about testing of each part of the system, describe the scenarios, represent each test and it's errors, challenges and modifications, and also we will figure out the error rate.

- **Chapter six: Conclusion and Recommendations**

In this chapter, we will mention what we achieved in this project and the conclusion for all things that we have done, also we will talk about the challenges that we faced and ending with recommendation needed for the future work.

# 2

## *Chapter Two*

### *Theoretical Background*

---

- 2.1 Overview**
- 2.2 GPS Technology**
- 2.3 Coordinate systems**
- 2.4 GSM Technology**
- 2.5 GPRS service**
- 2.6 Mobile phone**
- 2.7 Programming languages**

## 2.1 Overview

In this chapter we will mention the basic theoretical information, some technologies, and learn about some devices that will be used in our project. In order to use them in an appropriate way and take the desired advantages of them. So you will be able to understand and interact with our project.

We will talk about the GPS technology, GSM/GPRS technology; a collection of hardware includes the mobiles, GPS receivers, GPRS modems.

Software will be used represented by a programming language on the mobile phone, software on the central server in order to receive and process then plot.

## 2.2 GPS technology

### 2.2.1 Definition

The global positioning system (GPS) is a system for determining the geographic location of people with GPS receivers. Consists of "constellation" of 24 well-spaced satellites that makes a revolution every 12 hours.

- Global: Anywhere on earth but not inside buildings, underground, or anywhere else not having a direct view of a portion of the sky.
- Positioning: Give answers to the following questions, Where are you? How fast are you moving and in what direction? What direction should you go to get to some other specific location?
- System: A collection of components with connections between them.

The official U.S. Department of Defense name for GPS is NAVSTAR (Navigation System with Time And Ranging) \_which is a constellation of 24 satellites orbiting the earth, broadcasting data that allows users on or near the earth to determine their spatial positions. [4]

### 2.2.2 GPS components

The GPS technology consists of the following components:

- **The Earth:** it's the major component of GPS, its mass and its surface, and the space immediately above. The mass of the Earth holds the satellites in orbit.
- **Earth-circling satellites:** as shown in figure 2.1, twenty four solar powered radio transmitters, forming a constellation such that several are visible from any point on Earth at any given time. The satellites are at middle altitude of 20200 kilometers (Km); each is in a 12 hour orbit.



Figure 2.1: Earth-circling satellites orbits [5]

- **Ground based stations:** the GPS satellites are crammed with electronics, so they require monitoring. This is done by four ground based stations located on Ascension Island, at Diego Garcia, in Hawaii, and Kwajalein. Each satellite passes over at least one monitoring station twice a day. Information developed by the monitoring station is transmitted back to the satellite, which in turn re-broadcasts it to GPS receivers.
- **GPS Receivers:** it consists of
  - An antenna whose position the receiver reports
  - Electronics to receive the satellite signals
  - A microcomputer to process the data that determines the antenna position, and to record position values

- Controls to provide user input to the receiver
  - Screen to display information.
- **The United States Department of Defense:** The U.S. DoD is developing and maintaining NAVSTAR.[4]

### 2.2.3 How it works?

GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's exact location. Essentially, the GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. Now, with distance measurements from a few more satellites, the receiver can determine the user's position.

To calculate a 2D position (latitude and longitude) and track movement, GPS receiver must be locked on to the signal of at least three satellites. With four or more satellites in view, the receiver can determine the user's 3D position (latitude, longitude and altitude). Once the user's position has been determined, the GPS unit can calculate other information, such as speed, bearing, track, trip distance, distance to destination, sunrise and sunset time and more.

The following tips show how this system works:

- 21 GPS satellites and three spare satellites are in orbit at 12,000 miles above the Earth.
- The satellites are spaced so that from any point on Earth, four satellites will be above the horizon.
- Each satellite contains a computer, an atomic clock, and a radio. With an understanding of its own orbit and the clock, it continually broadcasts its changing position and time. (Once a day, each satellite checks its own sense of time and position with a ground station and makes any minor correction.)
- On the ground, any GPS receiver contains a computer that "triangulates" its own position by getting bearings from three of the four satellites. The result is provided in

the form of a geographic position, longitude and latitude, for most receivers, within 100 meters accuracy.

- If the receiver is also equipped with a display screen that shows a map, the position can be shown on the map.
- If a fourth satellite can be received, the receiver/computer can figure out the altitude as well as the geographic position.
- If you are moving, your receiver may also be able to calculate your speed and direction of travel and give you estimated times of arrival to specified destinations.[4]

### **2.2.4 The GPS satellite system**

The first GPS satellite was launched in 1978, a full constellation of 24 satellites was achieved in 1994, and each satellite is built to last about 10 years. Replacements are constantly being built and launched into orbit. A GPS satellite weighs approximately 2000 pounds and is about 17feet across with the solar panels extended. Transmitter power is only 50 watts or less.[4]

AVASTAR uses radio waves instead of sound: The waves that are used to measure the distance are electromagnetic radiation (EM). They move a lot faster than sound. Regardless of frequency, in vacuum, EM moves at about 299,792.5Km/sec, which is roughly 186,282 statute miles per hour. [4]

### **2.2.5 GPS in our project**

We will use GPS to measure the locations of the buildings in the old city, and to determine the real time location of the moving of the tourist, using GPS receiver built in the mobile phone. Mobile needs GPS receiver plus processing to calculate position, and read assistance data from network. The handsets take GPS measurements, and report these to the central server in the network.

## **2.3 Coordinate systems**

There are many coordinates system, which are systems that used to represent position coordinates of a point or geometric elements that can appear in different formats and some of these common modern systems, can be:

### **2.3.1 World Geodetic System 1984 (WGS84):**

The World Geodetic System is one of the coordinates system used in GPS, where it consists of three-dimensional Cartesian coordinate system and an associated ellipsoid, so that WGS84 positions can be described as either XYZ Cartesian coordinates or latitude, longitude and ellipsoid height coordinates. [6]

### **2.3.2 Earth Gravitational Model (EGM96):**

Earth Gravitational Model geoids 96, defines the nominal sea level surface by means of a spherical harmonics series of degree 360, which will provides about 100km horizontal resolution. And the deviations of the EGM96 geoids from the WGS 84 reference ellipsoid range from about 150m to about 85m .[6]

### **2.3.3 Earth Gravitational Model (EGM2008):**

Earth Gravitational Model geoids 2008 is a new higher fidelity model for the original WGS 84, This new model will have a geoids with a resolution approaching 10km, requiring over 4.6 million terms in the spherical expansion, versus 130317, EGM96 and 32,757 in WGS 84.[6]

### **2.3.4 North American Datum of 1927 (NAD27) :**

North American Datum of 1927 is a datum based on the Clarke ellipsoid of 1866. The reference or base station is located at Meades Ranch in Kansas,

There are over 50,000 surveying monuments throughout the US and these have served as starting points for more local surveying and mapping efforts. [6]

### **2.3.5 North American Datum of 1983 (NAD83):**

North American Datum of 1983, is an earth-centered datum based on the Geodetic Reference System of 1980. The size and shape of the earth was determined through measurements made by satellites and other sophisticated electronic equipment; the measurements accurately represent the earth to within two meters.[6]

### **2.3.6 Coordinate systems in our project:**

The received coordinates will be in one of the common coordinate's systems that depend on the type of mobile being used to transmit the coordinates. In our system, we will depend on American coordinate systems, where our mobile operating system android, will support these types of coordinates in accurate manner.

## **2.4 GSM technology**

### **2.4.1 Definition**

GSM (Global System for Mobile communications) is a second-generation digital network, supporting voice and simple data services, including “dial-up” data and text messaging. Open, digital cellular technology used for transmitting mobile voice and data services. GSM supports voice calls and data transfer speeds of up to 9.6 kbps, together with the transmission of SMS (Short Message Service). It predates Code Division Multiple Access (CDMA), which is especially strong in Europe. Enhanced Data rates for GSM Evolution (EDGE) (also known as Enhanced GPRS (EGPRS) is faster than GSM and was built upon GSM.



The system built based on cell phone technology GSM is the dominant digital mobile phone standard for most of the world. It determines the way in which mobile phones communicate with the land-based network of towers. [7]

### 2.4.2 GSM transmission and frequency allocation

Radio systems such as GSM can have hundreds of frequencies specified for use within a specified part of the frequency spectrum,. GSM needs multiple frequencies to allow different frequencies to be used, in different geographical parts of the network, to avoid radio interference within the communication channels.

GSM operates in many frequencies, the 900MHz and 1.8GHz bands in Europe, and the 1.9GHz and 850MHz bands in the US. The 850MHz band is also used for GSM and 3G in Australia, Canada and many South American countries.

For GSM Frequency Allocation, the spectrum assigned for the various GSM bands, in each case, there is an uplink (from mobile phone to base station), and a downlink (from base station to mobile phone). It is standard practice for the uplink to use the lower frequencies in cellular systems, as the attenuation of this link will be marginally less than the corresponding downlink. Transmitting and receiving on different frequency bands is known as Frequency Division Duplex (FDD).

Each of the bands is divided into a number of radios by carriers. These carriers are separated in all cases by 200 kHz. The GSM 900 bands are each 25 MHz wide, and therefore each contains 124 carriers. The end carriers are positioned at 200 kHz inside the band edges. As shown in figure 2.2, the lowest carrier in the 900 uplink band is 890.2 MHz, and the highest is 914.8 MHz.

The carrier separation in the 1800 bands is the same as for the 900 bands. As the bands are each three times wider (at 75 MHz), this gives 374 carriers in each of the 1800 bands.

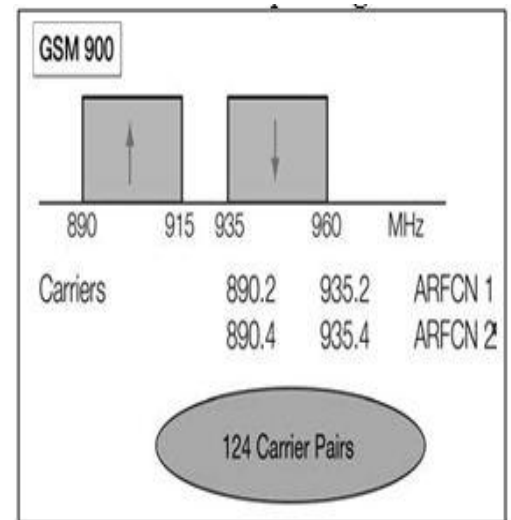


Figure 2.2: GSM Band Spectrums [7]

The 1900 bands (United States) contain 299 carriers. Carriers are assigned in pairs (uplink and downlink), and each pair is always separated by a fixed amount, equal 45 MHz at 900, and 95 MHz at 1800. A number called the Absolute Radio Frequency Channel Number (ARFCN) commonly refers to the pairs.

Having organized spectrum across most of the world, GSM's international roaming capability, allows users to access the same services when traveling abroad as at home. This gives consumers seamless and same number connectivity, in more than 218 countries. GSM carriers have roaming contracts with other GSM carriers, and typically cover rural areas more completely than competing CDMA carriers. GSM networks now cover more than 80% of the world's population. GSM satellite roaming has also extended service access to areas where terrestrial coverage is not available.

GSM also has the advantage of using SIM (Subscriber Identity Module) cards in. The SIM card, which acts as digital identity, is tied to the cell phone service carrier's network rather than to the handset itself. This allows for easy exchange from one phone to another without new cell phone service activation.

The main purpose of the GSM network is to facilitate easier access to cellular and satellite platforms across international lines. Using digital technology, it employs both speech and data channels in its system. At minimum these channels operate on the second generation (2G) network, but many use the third generation (3G) system to offer these services to clients. This enables the exchange of information at high-speed data rates via satellites and mobile cellular towers across networks and company lines. [7]

### **2.4.3 GSM network infrastructure**

GSM network is a series of virtually interconnected tower and satellite relays that send and receive digital signals allowing people to communicate wirelessly across the globe. The signal begins with a cell phone that places a call, and travels to the nearest GSM tower. That tower communicates the signal to a satellite, which in turn relays that signal to the tower nearest the call's destination, where it is then received by the call recipient. The GSM wireless network is the most popular wireless communications system today.

Original cell phone technology operated on an analog system, similar to a short-distance radio wave. This meant that many people could use the same frequency, but the frequency would vary by location.

The digital communications networks were carriers for Second-Generation (2G) cell phone technology, which featured less call interference, and increased call security, the phone's batteries lasted longer, and more calling features could be offered, since the network was expanded. The biggest advantage was that the efficiency of digital capacity lowered the startup cost of a new carrier, allowing smaller players to join the industry and offer real competition that leads to lower service prices.

GSM now carries Third Generation (3G) cell phone service as well as Fourth Generation (4G) technology, which allows users to transmit larger amounts of voice and data streams simultaneously. Each new incarnation of GSM service is compatible with each version that came before, eliminating the need for customers to upgrade their phone each time new technology is released.[7]

#### **2.4.4 GSM Network Architecture**

The GSM network can be divided into four main parts:

##### **2.4.4.1 The Mobile Station (MS)**

Consists of following two main components:

- **The Subscriber Identity Module (SIM):** It is protected by a four-digit Personal Identification Number (PIN). In order to identify the subscriber to the system.
- **Mobile equipment/terminal (ME):** There are different types of terminals distinguished principally by their power and application.

##### **2.4.4.2 The Base Station Subsystem (BSS).**

It may be divided into two parts:

- **Base Station Controller (BSC):** It controls a group of BTSs and manages their radio sources. A BSC is principally in charge of handoffs, frequency hopping, exchange functions and power control over each managed BTSs.

- **Base Transceiver Station (BTS):** it maps to transceivers and antennas used in each cell of the network. It is usually placed in the center of a cell. Its transmitting power defines the size of a cell. Each BTS has between 1-16 transceivers depending on the density of users in the cell.

#### 2.4.4.3 The Network and Switching Subsystem (NSS).

Its main function is to manage the communications between the mobile users and other users. It also includes data bases needed in order to store information about the subscribers and to manage their mobility. The different components of the NSS are:

- **Mobile Switching Center (MSC):** Performs the switching functions of the network, and provides connection to other networks.
- **Gateway MSC (GMSC):** Interconnects two networks, the cellular network and other network such as PSTN.
- **Home Location Register (HLR):** Stores information of the subscribers belonging to the coverage area of a MSC, it also stores the current location of these subscribers and the services to which they have access.
- **Visitor Location Register (VLR):** Contains information from a subscriber's HLR necessary to provide the subscribed services to visiting users.
- **Authentication Center (AuC):** It serves security purposes; it provides the parameters needed for authentication and encryption functions. These parameters allow verification of the subscriber's identity.
- **Equipment Identity Register (EIR):** stores security-sensitive information about the mobile equipment. It maintains a list of all valid terminals as identified by their International Mobile Equipment Identity (IMEI). The EIR allows then to forbid calls from stolen or unauthorized terminals.
- **GSM Interworking Unit (GIWU):** The GIWU provides an interface to various networks for data communications. During these communications, the transmission of speech and data can be alternated.

#### 2.4.4.4 Operation and Support Subsystem (OSS)

It is connected to components of the NSS and the BSC, in order to control and monitor the GSM system. It is also in charge of controlling the traffic load of the BSS. It must be noted that as the number of BS increases with the scaling of the subscriber population some of the maintenance tasks are transferred to the BTS, allowing savings in the cost of ownership of the system. [7]

All previous networks' elements are shown in the following figure that shows the relation between them.

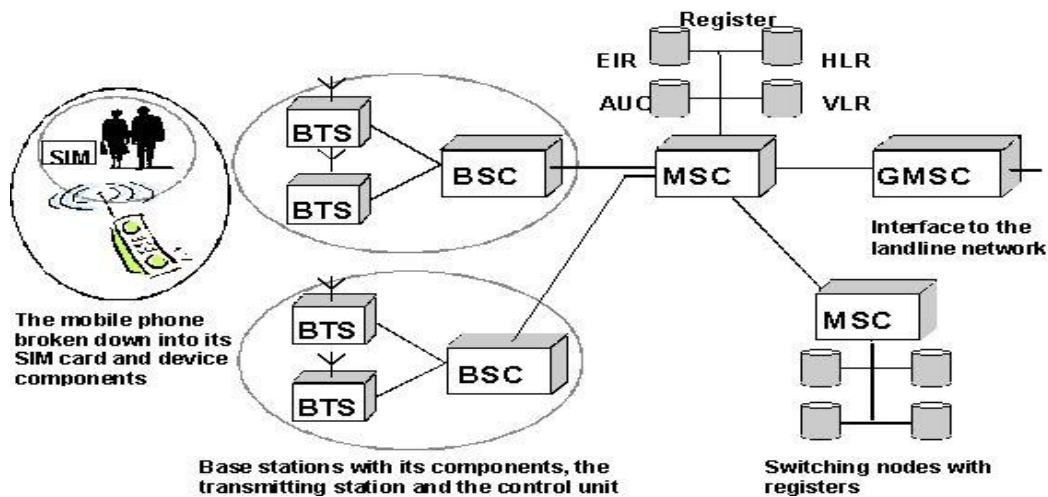


Figure 2.3: GSM Network's Components [8]

#### 2.4.5 GSM In Our Project

The usage of the GSM technology in this project represented by transferring the data from the GPS receiver (or the mobile phone) to the central server. These data will contained position request.

## **2.5 GPRS service**

### **2.5.1 Definition**

General Packet Radio Services (GPRS), is a packet-based wireless communication service that provides data rates from 56 up to 114 Kbps, and continuous connection to the Internet for mobile phone and computer users. The higher data rates allow users to take part in video conferences, and interact with multimedia Web sites, and similar applications using mobile handheld devices as well as notebook computers. GPRS is based on Global System for Mobile communication (GSM) and complements existing services such circuit-switched cellular phone connections and the Short Message Service (SMS).

In theory, GPRS packet-based services cost users less than circuit-switched services, since communication channels are being used on a shared-use, as-packets-are-needed basis rather than dedicated to only one user at a time. It is also easier to make applications available to mobile users, because the faster data rate means that middleware currently needed to adapt applications to the slower speed of wireless systems are no longer be needed. As GPRS has become more widely available, along with other 2.5G and 3G services, mobile users of virtual private networks (VPNs) have been able to access the private network continuously over wireless, rather than through a rooted dial-up connection.

GPRS also complements Bluetooth, a standard for replacing wired connections between devices with wireless radio connections. In addition to the Internet Protocol (IP), GPRS supports X.25, a packet-based protocol that is used mainly in Europe. GPRS is an evolutionary step toward Enhanced Data GSM Environment (EDGE) and Universal Mobile Telephone Service (UMTS).[9]

### **2.5.2 GPRS and packet data in GSM**

GPRS adds a packet-based data capability to the GSM system, and brings IP networking into the GSM world. As well as modifications within the handsets and the radio interface, GPRS also involves the addition of new packet routers into the GSM core network, producing a core network with separate circuit- and packet switched domains. Using GPRS, each user only requires radio resources when a

packet of information is sent or arrives, it does not require the channel for the entire duration of the data transfer. In this way, the single timeslot can be made available for more than one user, allowing radio channels to be shared, by up to seven users in GPRS.

The overall data rates in GPRS are specified for each timeslot, varying depending on what is known as the coding scheme. There are four coding schemes (CS) in GPRS (known as CS1, CS2, CS3, and CS4). The data rates for each coding scheme are shown in table 2.1. Note that each overall channel is actually designed to carry 22.8 kbps (as in standard GSM connections), and the extra capacity is actually used for error protection. Where there is a good error protection for CS1, but not so good for CS4. In practice, only coding schemes 1 and 2 are widely used.

CS1 and CS2 offer good error detection and correction with low throughput; in the first step of GPRS software only these two techniques may be used. CS3 and CS4 provide higher throughputs but have little or no error correction capabilities.

Due to the packet switched characteristics of GPRS software the allocation of the available timeslots may vary from one instant to the next. So, it may have 8 timeslots at one time and 4 later on.

Communication between the base station and any given mobile host is full-duplex but can be asymmetric; that is the down link and the uplink capacities need not be the same. Hence, the system can select the coding scheme based on the amount of data to be transferred through down link and uplink. Generally, the downlink has more frequency than the uplink. Both links can send user data as well as control data. There are some situations in which the amount of user data may be larger than the control data and vice versa. That is, the downlink may have more data to carry than the uplink. Thus, the selection of coding scheme may differ for downlink and uplink. [10]

**Table 2.1: Data Rate for GPRS Software [10]**

User Data Rate	CS1	CS2	CS3	CS4
1 Timeslot	9.05 kbps	13.4 kbps	15.6 kbps	21.4 kbps
8 Timeslot	72.4 kbps	107.2 kbps	124.8 kbps	171.2 kbps

### 2.5.3 GPRS Network Elements

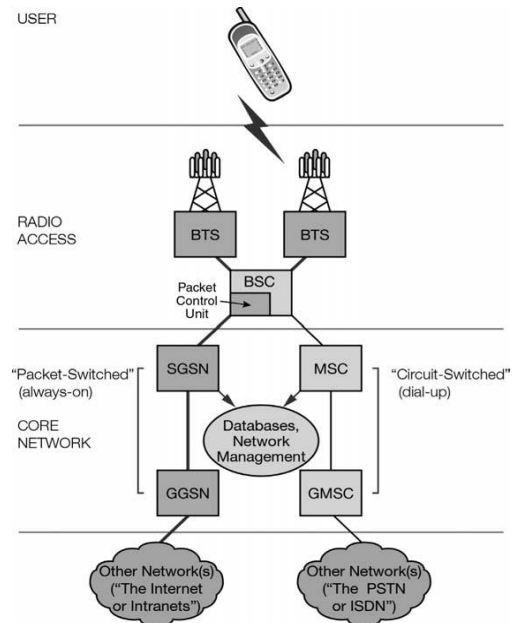
Additional elements are required to provide the General Packet Radio Services (GPRS), including:

- core network GPRS support nodes (GSNs), comprising:
  - Serving GSN (SGSN).
  - Gateway GSN (GGSN).

These elements use IP (Internet Protocol) technology to route information to and from the mobile handset. They are effectively IP routers with

modifications to allow for managing subscriber access, mobility, and IP sessions.

A GPRS data session would be handled by the GSN rather than the MSC, which is used for voice. In addition, GPRS operation requires the BTS to be GPRS-capable, and a packet control unit must be available, usually at or within the BSC. These two base station subsystem (BSS) elements allow the data packets to transfer in the right format on the correct GPRS radio channels.[7]



**Figure 2.4: GPRS network elements. [7]**

## 2.6 Mobile phone (cellular phone):

Mobile phone is one of the most sophisticated and modern techniques that spread dramatically in our societies, it transformed from a simple use to make calls, to be used nowadays in many complex and useful applications.

Back to the history of mobile phones, we found that the first mobile telephone call was made from a car in St. Louis, Missouri, USA on June 17, 1946, using the Bell System's Mobile Telephone Service.



Many updates and developments are quickly continued on mobile phones, and still updates until now, Mobile phones become nowadays a very smart, efficient, and small device that puts the whole world in your hands, and support large number of modern techniques included within this small device. [11]

### **2.6.1 Mobile Operating systems (mobile software platform):**

Like Microsoft's Windows, Apple's Macintosh, and Linux, and many others desktop operating systems, the evolution that has occurred on the mobile phone also enable us to control a mobile devices , applications and services, with their special operating systems, as the case with computer operating systems.

The most common mobile operating systems are:

- Android from Google Inc. (open source, Apache).
- BlackBerry OS from RIM (closed source, proprietary).
- Symbian OS from the Symbian Foundation (open public license).
- Windows Phone from Microsoft (closed source, proprietary).

These operating systems and many other ones, represent the operating systems for our mobile phones, and by using the appropriate programming language for each operating system, we enable to programming these phones, so we can build different applications and services we want from these mobiles. [11]

In our system Android is the operating system that we want to use in order to support our project objective requirements.

#### **2.6.1.1 Android operating system:**

In recent years Open Handset Alliance led by Google has deployed a neat, versatile, powerful, and elegant platform which was android. This platform spread widely and its market quickly growing, because it is supported by large set of hardware, software, and network carriers.

“Android is an open source platform and it is released under open source license. The Android operating system software stack consists of Java applications running on a Java based object oriented application framework on

top of Java core libraries running on a Dalvik virtual machine featuring JIT compilation. Libraries written in C include the surface manager, Open Core media framework, SQLite relational database management system, OpenGL ES 2.0 3D graphics API, Web Kit layout engine, SGL graphics engine, SSL.”[11]

### 2.6.1.2 Android architecture

The architecture of Android operating system is described as following diagram with briefly explanation:

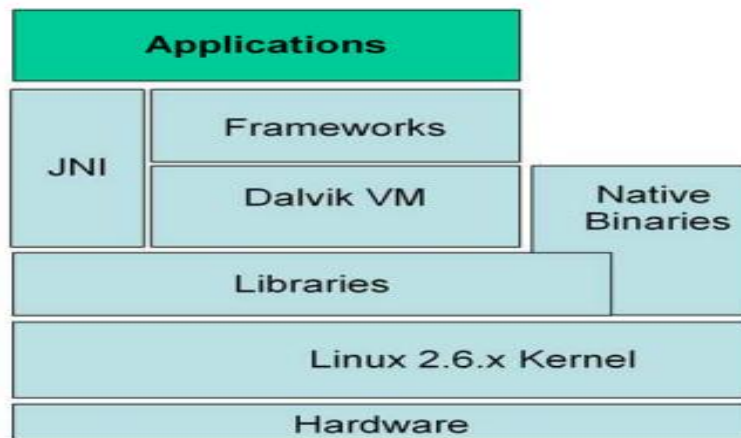


Figure 2.5: Architecture of Android OS [12]

- **Application** java is the programming language for most of applications written with android, using Android SDK.

This application when deployed is being a single file in a format (.apk). This is a modified Java Archive (JAR) file. This file contains the Java classes in a custom format (.dex) which make up the application, as well as an application manifest.

- **Application Frameworks** are also written in Java, and are based on the low level core libraries -which provide the basic subset of Java – java.io.\*, java.util.\*, etc.

And there is many level of manage and control the application, such as activity, window, package, telephony, and many other managers, that control the application you build.

- **Dalvik VM** Dalvik virtual machine which has been adapted to the specifics of mobile systems with limited CPU capabilities, low RAM and disk space, and limited battery life.

Dalvik is a register-based virtual machine, and it is neither fully J2SE nor J2ME compatible. Instead it works with its own version of Java Byte Code, pre-processing its input by using a utility called “dx” which produce “.dex” files from the corresponding Java “.class” files.

- **Native Binaries** this is a standard Linux binaries of ELF formatted, they can be created with

The Android Native Development Kit after being coded in C or C++ usually, and then compiled directly to the target processor (usually, ARM).

- **JNI** Java Native Interface which used to achieve native-level functionality, mainly by enables a Java application to directly invoke a non-Java function. Which make the runtime environment is so rich. And a Good reasons to use JNI are, efficiency, and obfuscation.

- **Libraries:** when the Dalvik VM wants to execute an operation. It will call for services from these libraries which are a collection of many libraries, all open source, and implement the low level functionality provided by the runtime.

Android support advanced point to this library which is preloaded into memory, which allows for faster load times, and doesn't waste any memory.

- **Linux :** Android uses the open source Linux Kernel as its own, the kernel is similar, but not identical, to the standard Linux kernel distribution.[12]

### **2.6.1.3 Android features:**

There are many features supported by android operating system, and here we mention some of them which give helpful points for our system objectives:

- Android supports connectivity technologies including GSM, Bluetooth, and many others, but in our project mainly we need the GSM support, and we may need the Bluetooth.
- Android operating system is powerful OS that growing worldwide and widely supported by large number of hardware, software and network, so it will be useful in our project, and available to large number of customers.
- Android support some forms of messaging, like SMS and MMS, and in our project it may be needed for communication.
- While most Android applications are written in Java, there is no Java Virtual Machine in the platform and Java byte code is not executed. Java classes are compiled into Dalvik executable and run on Dalvik, a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU. J2ME support can be provided via third-party applications.[12]

## **2.7 mobile programming language :**

In our project the phones will being programmed using java programming language, where it is an important languages, that is created by James Gosling from Sun Microsystems in 1991, and nowadays owned by Oracle, it's one of a very popular, powerful, general-purpose, concurrent, class-based, and object-oriented language.

It used to support many important applications and one of these is Android applications, which relies heavily on java fundamentals.

The Android SDK includes many standard Java libraries such as data structure libraries, math libraries, graphics libraries, and many other ones as you want.

### 2.7.1 Java architecture:

We will mention the main architecture for java programming language: where Java programming language consists mainly of four components; Java compiler, Java virtual machine, java run time environment, and the Java class libraries. And the relation between these three components can be described in brief during program execution as follow:

- You write your source code with the programming instructions.
- When you run your code it will be compiled using the language compiler that translates it from the source to Java class files.
- Run the class files on a Java virtual machine, where JVM is written specifically for a specific operating system.
- Java class libraries: that enables you to access system resources; by calling methods in the classes that implement the Java Application Programming Interface. As your program runs, it fulfills your program's Java API calls by invoking methods in class files that implement the Java API.
- Java runtime environment (JRE) that consists of the JVM and the Java class libraries.[13]

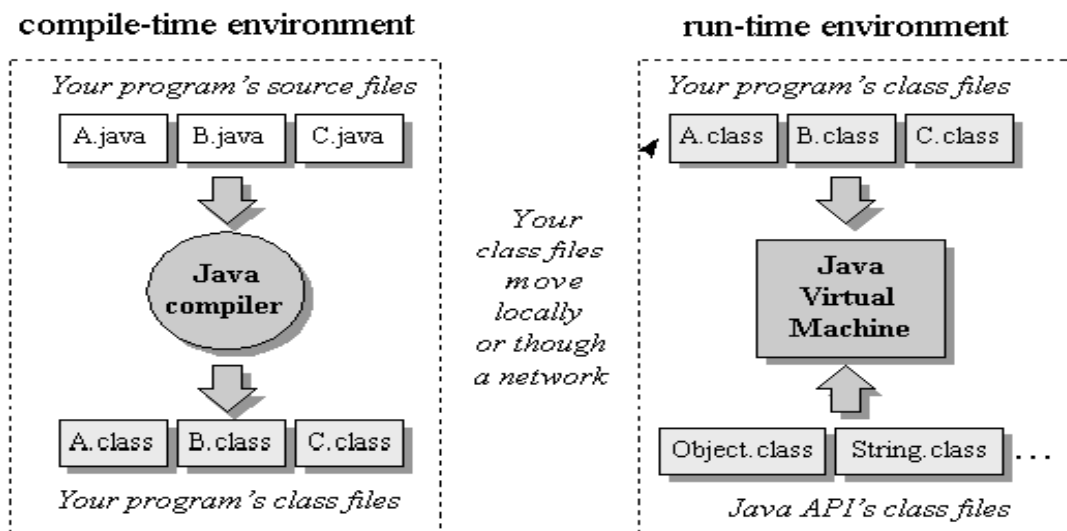


Figure 2.6: Java main architecture [13]

### 2.7.2 Java development environment (E\_clips):

Eclipse will be the development environment for our mobile phone software, where it is an integrated development environment (IDE) for Java, and also a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system.

And nowadays you can build your applications on mobile, and one of your best choices is to get started on mobile Java development using Eclipse, where to build any application on mobile you will need three main software, first will be sun's java, then Eclipse, and Eclipse ME plug-in and the Sun Java Wireless Toolkit also needed.[14]

### 2.7.3 Java features:

- **Platform Independence:** Java compilers produce byte-code instructions for the Java Virtual Machine (JVM). And use the JVM as abstraction and do not access the operating system directly. So same compiled byte code can run unmodified on several platforms.
- **Objects Oriented:** everything in a Java program is an object and everything is descended from a root object class. Except the primitive data types.
- **Strongly-typed programming language:** Java is strongly-typed, where the types of the used variables must be pre-defined and conversion to other objects is relatively strict
- **Familiar C++-like Syntax:** there is similarity between Java syntax and that of the popular C++ programming language. Which enabling the rapid adoption of Java.
- **Garbage Collection:** this feature makes your programming easier to write and less prone to memory errors, make memory use efficient and reduce the losses and unused space. where Java does not require programmers to explicitly free dynamically allocated memory.[13]

# 3

## ***CHAPTER THREE***

### ***PROJECT CONCEPTUAL DESIGN***

---

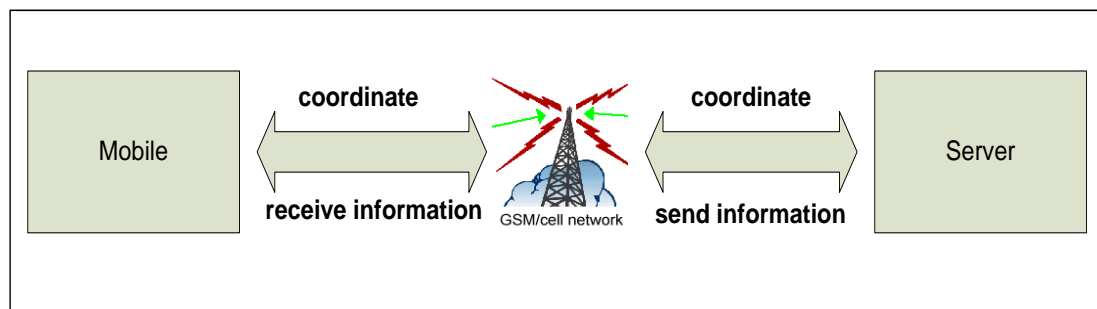
- 3.1 Overview**
- 3.2 General system block diagram**
- 3.3 System flow chart**
- 3.4 Design options**

## 3.1 Overview

In this chapter we will describe our system main parts and the design concepts in some details; we will talk about system general block diagram, the system main components, system flowcharts and their related options.

## 3.2 General system block diagram

Our system consists of many components: mobile, GSM network (service provider) and server, as shown in Fig 3.1 block diagram:



**Figure 3.1: General system block diagram**

### 3.2.1 Mobile

The main component in our system is mobile, where geographic maps stored in, and graphical user interface will be created on. The mobile phone must support GPS receiver to receive real coordinates from GPS satellite.

### 3.2.2 GSM network

The second component of our system is the GSM network. It represents the service provider in the system, which we will perform the transmission of the coordinates from the Mobile to the central server, over GSM components.

The transmission procedure will be done through GPRS, so the used network must support GPRS, which adds a packet-based data capability to the GSM system, brings IP networking into the GSM world, and involves the addition of new packet



routers into the GSM core network, producing a core network with separate circuit- and packet switched domains.

### 3.2.2.1 Why GPRS?

- **Cost:** Communication via GPRS is cheaper than through the regular GSM network.
- Customers only pay for the amount of data transported, and not for the duration of the Internet connection. Also it's cheaper than transmitting messages in SMS (short message service). So, it's more suitable to our system in order to send coordinates continuously with acceptable cost.
- **Constant Connection:** Through GPRS technology, users are constantly connected to the Internet. As GPRS services are available wherever there is GSM coverage.
- **Mobility:** GPRS provides wireless access to the Internet from any location where there is a network signal. This enables you to surf the Internet on your laptop or phone, even in remote areas. That's what we need to track the moving mobile
- **Speed:** Although new, faster technology exists today, GPRS is still faster than the older WAP (Wireless Application Protocol) and regular GSM services. GPRS data is transferred at speeds ranging from 9.6 kilobytes per second up to 14kbps.
- **Simultaneous Use:** When you access the Internet through GPRS, it does not block incoming calls through the GSM network. This enables you to make or receive voice calls while you are using our application.

### 3.2.3 Server

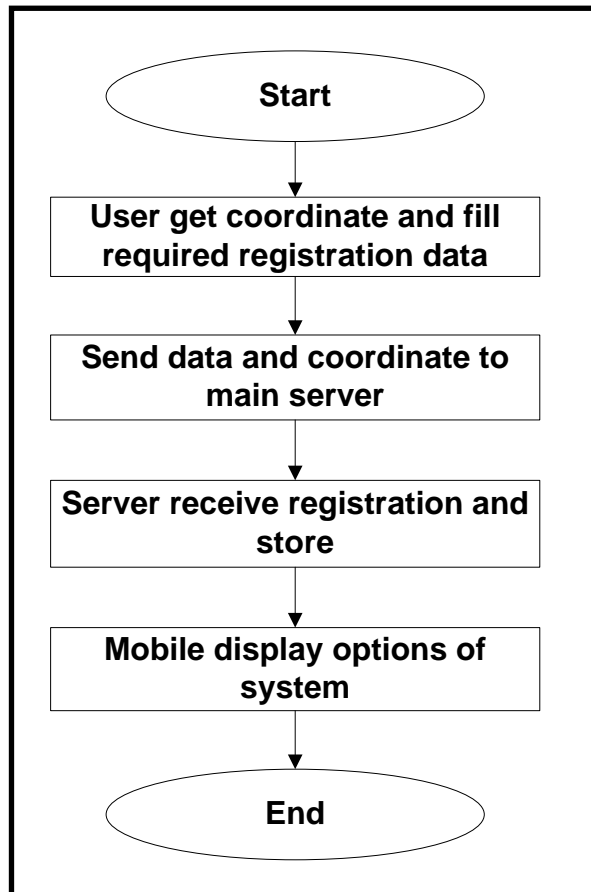
In our system we will create servlet server. A servlet is a small program that runs on a server. The term was coined in the context of the Java applet, a small program that is sent as a separate file along with a Web (HTML) page. Java applets, usually intended for running on a client, can result in such services as performing a calculation for a user or positioning an image based on user interaction.

Some programs, often those that access databases based on user input; need to be on the server. Typically, these have been implemented using a Common Gateway Interface (CGI) application. However, with a Java running in the server, such programs can be implemented with the Java programming language. The advantage of a Java servlet on servers with lots of traffic is that they can execute more quickly than CGI applications. Rather than causing a separate program process to be created, each user request is invoked as a thread in a single daemon process, meaning that the amount of system overhead for each request is slight.

### **3.3 System flowchart**

To achieve the objectives of our system, many steps will be done, and these can be described in general, as shown in the system main flowchart in figure (3.2) where the main steps are:

1. Fill the required registration data by customer, and determine the mobile user location, using GPS Technology.
2. Develop mobile application, first to send registration information, and then get the active mobile user location, and send them periodically, via GPRS.
3. Server receives registration and store information registration in database.
4. Tourist enters the system and uses the system options.



**Figure3.2: Main system flowchart**

Each one of these steps will be described in detail as follow:

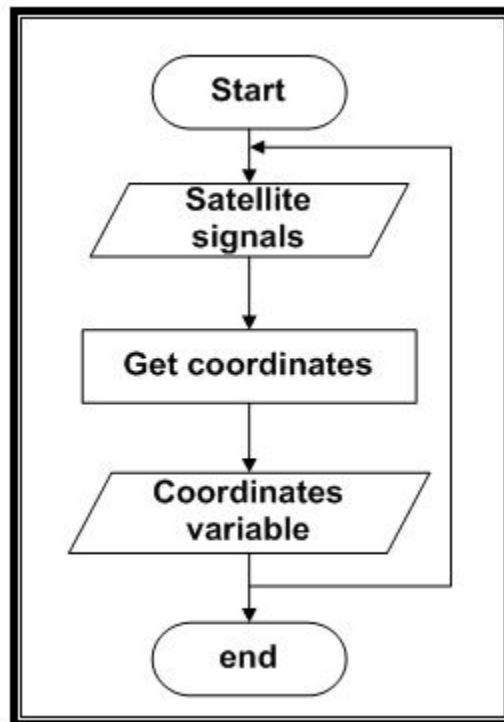
### **3.3.1 Fill registration data and get coordinates**

#### **3.3.1.1 Fill registration data**

The first step in our system is to make a registration for a new Tourist; this is done by letting the user to enter some required information, in order to be sent to the servlet server, as are quest for joining the whole system. These data sent only one time.

### 3.3.1.2 Get Coordinates

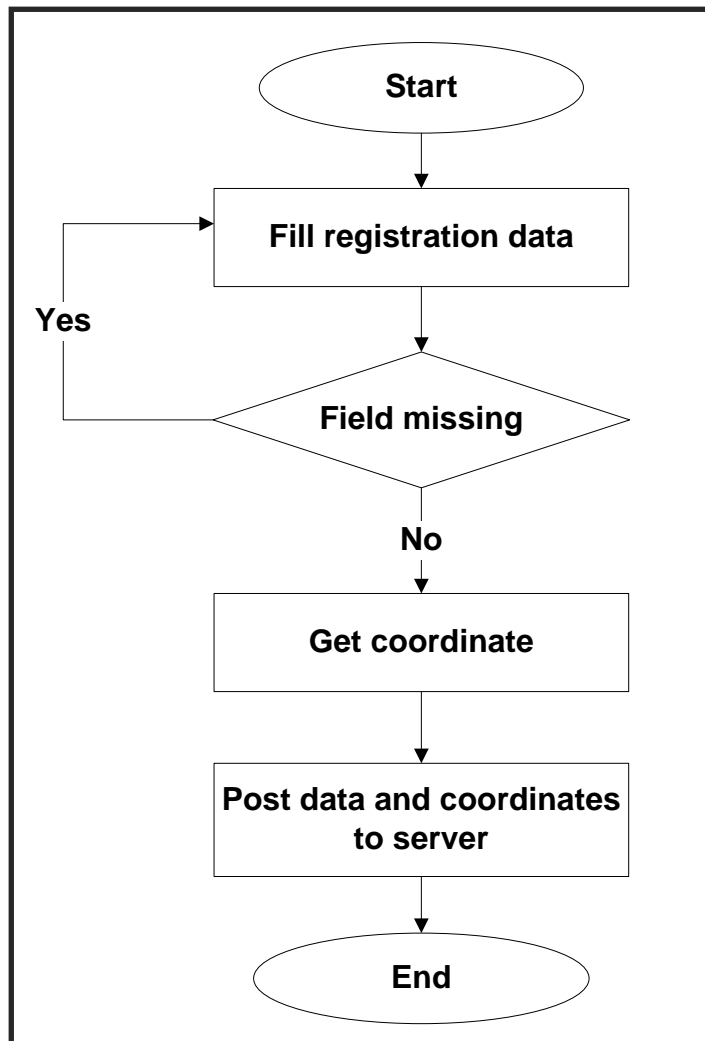
After registration completed, the next step is to determine the accurate –real time\_GPS coordinates for Mobile tourist. The used mechanism of getting these coordinates is described as shown in figure (3.3) get coordinates flowchart:



**Figure 3.3: Get coordinates flowchart**

- The received signal from the GPS satellite is processed, at the mobile's built-in receiver, to determine the longitude and latitude.
- A code will be added to the mobile, to provide access to the built-in receiver, and store the longitude and latitude parameters, at the defined coordinate's variables.

### 3.3.2 Send data and coordinates to servlet server



**Figure 3.4: Send data and coordinates flowchart**

- As shown in the figure, tourist will fill the required registration information.
- The received coordinates which already stored at the defined variables will be prepared to be posted to the servlet server.
- Post registration information and coordinate to servlet server.

### 3.3.3 Receive data and coordinate by servlet server

In the other side of our system, where the server need to continue the system work, there are main steps, start with connection side that is related to the mobile posted data, these data will be received by the server to store it in database.

The connection server side is described as shown in figure (3.5) server received data and coordinates flowchart:

- First activate the ability to receive new tourist that want to connect to our system.
- Waiting for new connection request.
- After request received the server will store data and coordinate in database.

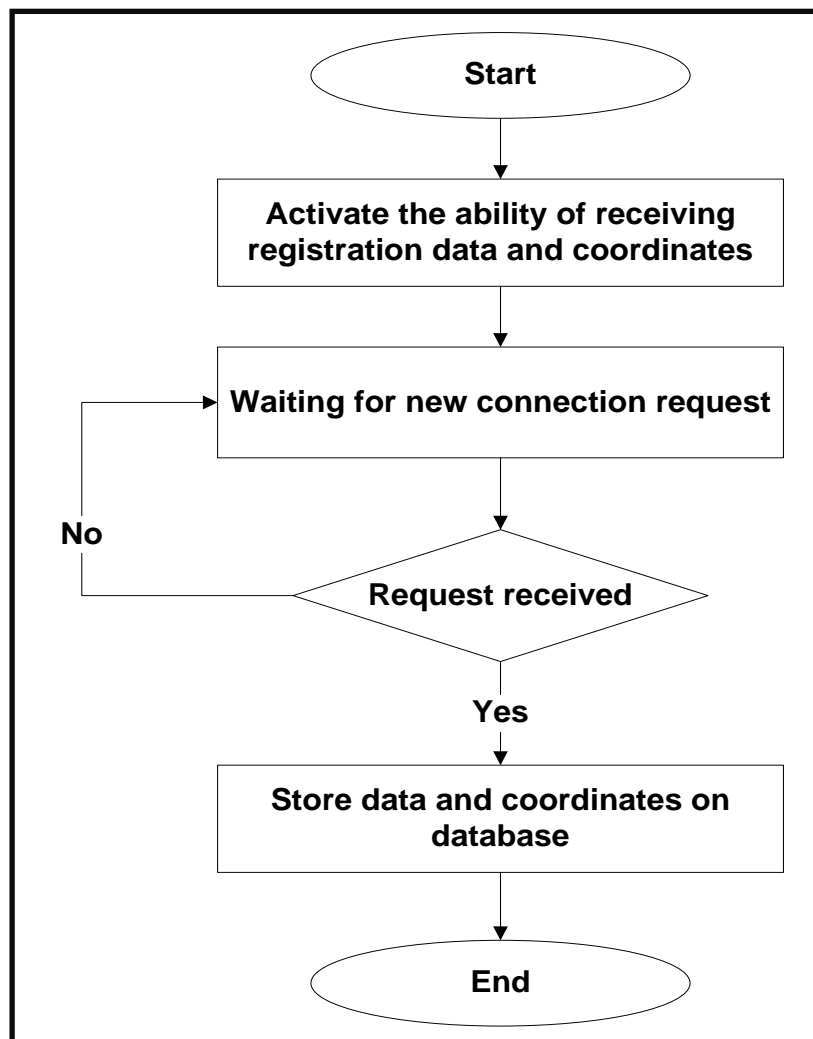


Figure 3.5: Server received data and coordinates flowchart

### **3.3.4 Tourist enters the system and uses the system options**

After tourist completed registration, he can now log in to the system, and use the options available by the system these options are route, find friend, food.

Each one of these options will be described in detail as follow:

#### **3.3.4.1 Route**

The system will draw routes for the tourist to show how to reach the most important places in the Old Town of Hebron. These routes are static. In our system we will use Dijkstra's algorithm to find the best and shortest path from a point on the map (the source) to a destination.

##### **3.3.4.1.1 Dijkstra's algorithm**

Dijkstra's algorithm, named after its discoverer, Dutch computer scientist Edsger Dijkstra, is an algorithm that solves the single-source shortest path problem for a directed graph with nonnegative edge weights [15].

Dijkstra's algorithm pseudo code:

```
// Let v1 be the origin vertex,  
// and initialize W and ShortDist[u] as  
W: = {v1}  
ShortDist [v1]:=0  
  FOR each u in V - {v1}  
    ShortDist[u]:= T [v1, u]  
  
// Now repeatedly enlarge W  
// until W includes all vertices in V  
WHILE W <> V  
  
  // Find the vertex w in V - W at the minimum distance  
  // from v1  
  MinDist: = INFINITE  
  FOR each v in V - W  
    IF ShortDist[v] < MinDist  
      MinDist = ShortDist[v]  
w: = v  
  END {if}  
END {for}  
  
  // Add w to W  
W := W U {w}  
  
  // Update the shortest distance to vertices in V - W  
  FOR each u in V - W  
    ShortDist[u]:= Min (ShortDist[u], ShortDist[w] + T [w, u])  
  END {while}
```

**Figure 3.6: Dijkstra's algorithm pseudo code[15]**

### 3.3.4.1.2 Find Friends

The system will provide a service for friends to find place of each other by doing the following steps:

- The mobile send request to the server for online user.
- Mobile receive online user from server and then chose one of them.
- After mobile chose one of online user, then send request to the server to know where the user who was chosen.
- Server search in database for selected user.



- After selected user found, server send his coordinate to the mobile, which send request.
- After mobile receive coordinate, then show these coordinates on map.
- If mobile change his location, send new coordinate location to the server.

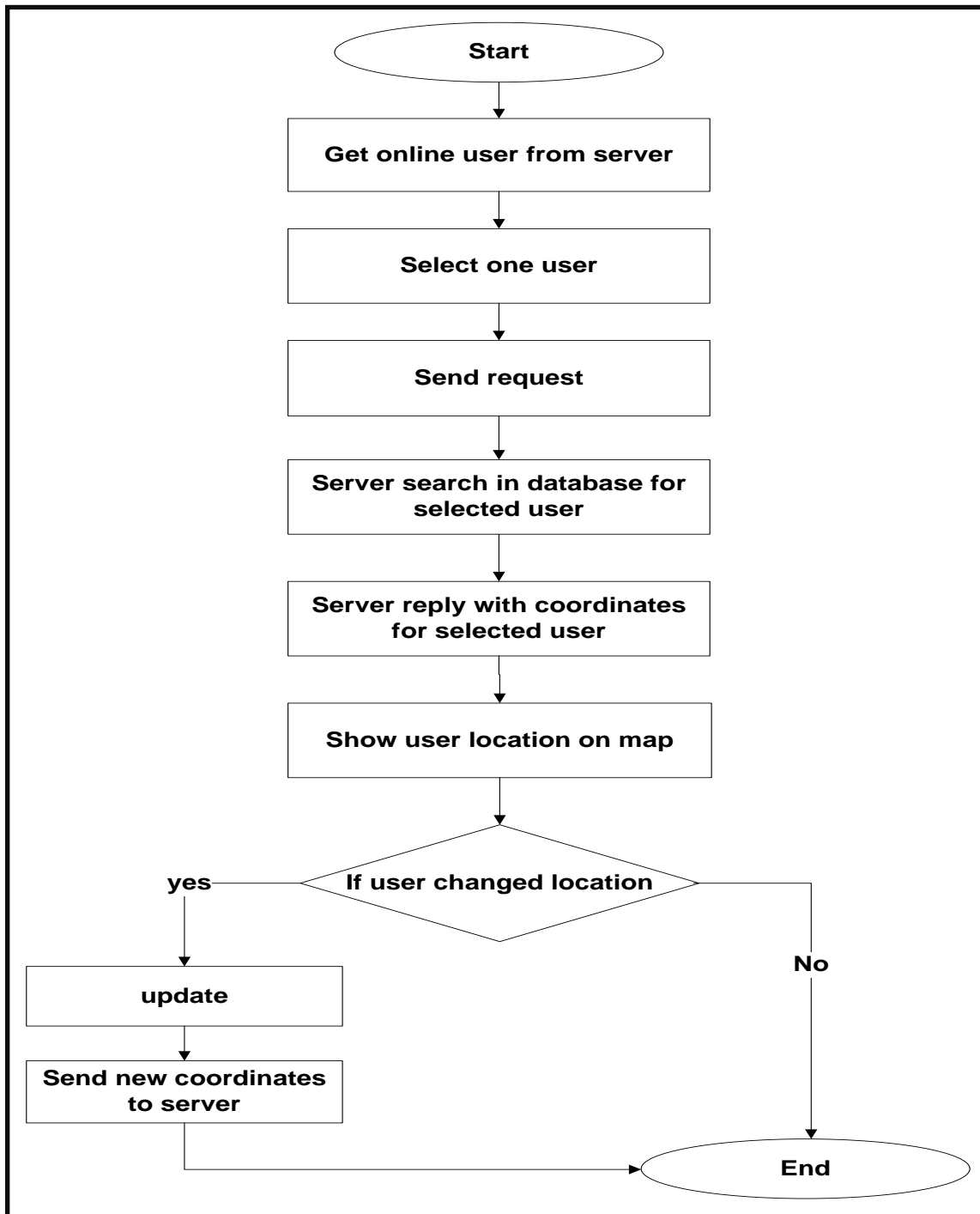


Figure 3.7: Find Friend flowchart

### **3.3.4.1.3 Food**

The system will provide service to know locations of the famous restaurants of the old town of Hebron, by plotting the coordinates of these places on the map.

## **3.4 Design options**

### **3.4.1 Mobile**

Mobile available with different operating-system, in our system android is the operating system we want to use

#### **3.4.1.1 Android OS**

Google Android is "open source software" and a competitive operating system; we choose to be used between all other alternatives, because it is taking over other leading systems with many advantages, such as:

- Android lets you change your settings faster.
- Android has a better app market, allows developers and programmers to develop applications in what is known as "application without borders".
- Updated user interface design.
- Support for extra-large screen sizes and resolutions.
- Improved power management and application control.

"Google fully developed Android and make it into an open source. Now, any phone manufacturer can use Android without expensive license fee from Google. Because it is open, manufacturer can modify Android without restriction, allowing it to fit the device they are making total freedom. This makes it a big incentive for any device

manufacturers to adopt Android. The ability to run tens of thousands of apps is another big incentive". [14]

### 3.4.1.2 Galaxy Note 2

We choose a galaxy note2 mobile phone to implement our system, it has been chosen because it contains and supports all of the required technologies that will be used to perform the project objectives. It has the following features:

- Support Operating Frequency:
  - **2G Network:** GSM 850 / 900 / 1800 / 1900
  - **3G Network:** HSDPA 850 / 900 / 1900 / 2100
- **HSDPA 850 / 900 / 2100 - N7105**
  - **4G Network:** LTE 800 / 900 / 1800 / 2600 - N7105
- Operating System

Android OS, v4.1.1, upgradeable to 4.1.2 .

- GPS: with A-GPS support and GLONASS.
- Support for SMS (threaded view), MMS, Email, Push Mail, IM, and RSS.
- Connectivity
  - Wi-Fi: 802.11 a/b/g/n, dual-band, DLNA, Wi-Fi Direct, Wi-Fi hotspot.
  - USB: micro USB v2.0 (MHL), USB Host.
- Network
  - LTE 100Mbps / HSDPA+ 42Mbps / HSUPA 5.76Mbps / 3G / EDGE / GPRS.

### 3.4.2 Geographic maps

There are many sources where our geographic maps can be taken, to create our system database that will represent the whole tracked geographic area.

### **3.4.2.1 Google earth**

Google earth is a map and geographic information software, that puts the whole world in your hands, This product released in 2005, and become nowadays available on all personal computers and mobiles, you only need to download free version of this software, then you can use it easily, which provide you with, word maps and geographical information about any region on the earth, these maps and photos was taken via satellite imagery, aerial photography, and GIS 3D globe. [16]

### **3.4.2.2 Image map**

Image map is a list of coordinates relating to a specific image, created in order to hyperlink areas of the image to many destinations (as opposed to a normal image link, in which the entire area of the image links to a single destination). [17]

### **3.4.2.3 Google Map**

A Google map is a Web-based service that provides detailed information about geographical regions and sites around the world. In addition to conventional road maps, Google Maps offers aerial and satellite views of many places. In some cities, Google Maps offers street views comprising photographs taken from vehicle. [18]

In our system, we will use Google map as reference for our regions map, because Google Maps offers several services as part of the larger Web application, as follows:

- The Google Maps application program interface (API) makes it possible for Web site administrators to embed Google Maps into a proprietary site such as a real estate guide or community service page.
- Google Street View enables users to view and navigate through horizontal and vertical panoramic street level images of various cities around the world.

- Supplemental services offer images of the moon, Mars, and the heavens for hobby astronomers.[18]

### 3.4.3 Connection Options

To implement a GPRS connection, many connection options are available. Here we will mention it and choose the most proper one:

#### 3.4.3.1 GPRS mobile with PC interface

At this option, the user data and location coordinates will be sent from tourist mobile to the main server. So an interface is needed between the adjacent mobile and the PC, this interface can be:

- **Bluetooth connection (wireless interface):** which efficient to be used here, because we want to transmit data over short range distance,(between GPRS mobile and the central server).Transmit and receive data between mobile and central server, as Bluetooth signals.
- **USB cable (wired interface):** A data cable is a long cord, which connects computer with the mobile in order to share data between them. The data cable transfers data on the form of electrical signals from mobile to central server and vice versa.

### 3.4.3.2 **HttpURLConnection Connection**

At this option we will consider the mobile phone as HTTP client, which has an application make a post directly to the PC, which will be considered as a web server.

The data will be posted directly as a parameter to the functions built on the server.

In our system we will use the second option, HttpURLConnection connection, because it more direct and practical, and it cancel the need of the PC mobile adjacent and the interface between them.

# 4

## ***CHAPTER FOUR***

### ***DETAILED DESIGN***

---

**4.1 Overview**

**4.2 System classes model diagram**

**4.3 System use cases**

**4.4 System sequence diagram**

## 4.1 Overview

In this chapter, the software engineering details will be mentioned and described, including UML model diagram for both mobile and server application software classes, with implementation for these classes, then the system use case diagram will be described, and also the sequence diagram for use cases will be mentioned.

## 4.2 System classes model diagram

### 4.2.1 Mobile Software UML diagram

A detailed description for our Mobile Application will be implemented using UML model diagram; that will describe the main classes, the attributes used in each class, methods that were defined, and a detailed description of these methods.

The mobile application software is described as shown in figure (4.1).

#### 4.2.1.1 Classes Implementation

Our system Mobile application software contains a group of classes that interact with each other, to make the whole system work properly, and complete the system objectives, and these classes are:

##### □ Hello class:

This class will be used when starting the application, as greeting screen, it consist mainly of two parts:

➤ Variables:



**mpstart:** it's media player object plays music at the beginning of the application.

**logtimer:** a thread contains the run and sleep method.

**timer:** an integer contains the number of second represent the time interval for showing the hello class and playing its music.

**Register Activity:** Intent contains the new activity that will be shown when the welcome activity finished.

➤Methods:

**onCreate:** Called when the activity is starting, this is where most initialization should go.

**run:** this is runnable thread method, that will be applied when the form being loaded, to start get the timer its interval value, and then start its down counting.

**start Activity:** Launch a new activity, this implementation overrides the current activity, clear the screen and start the next activity.

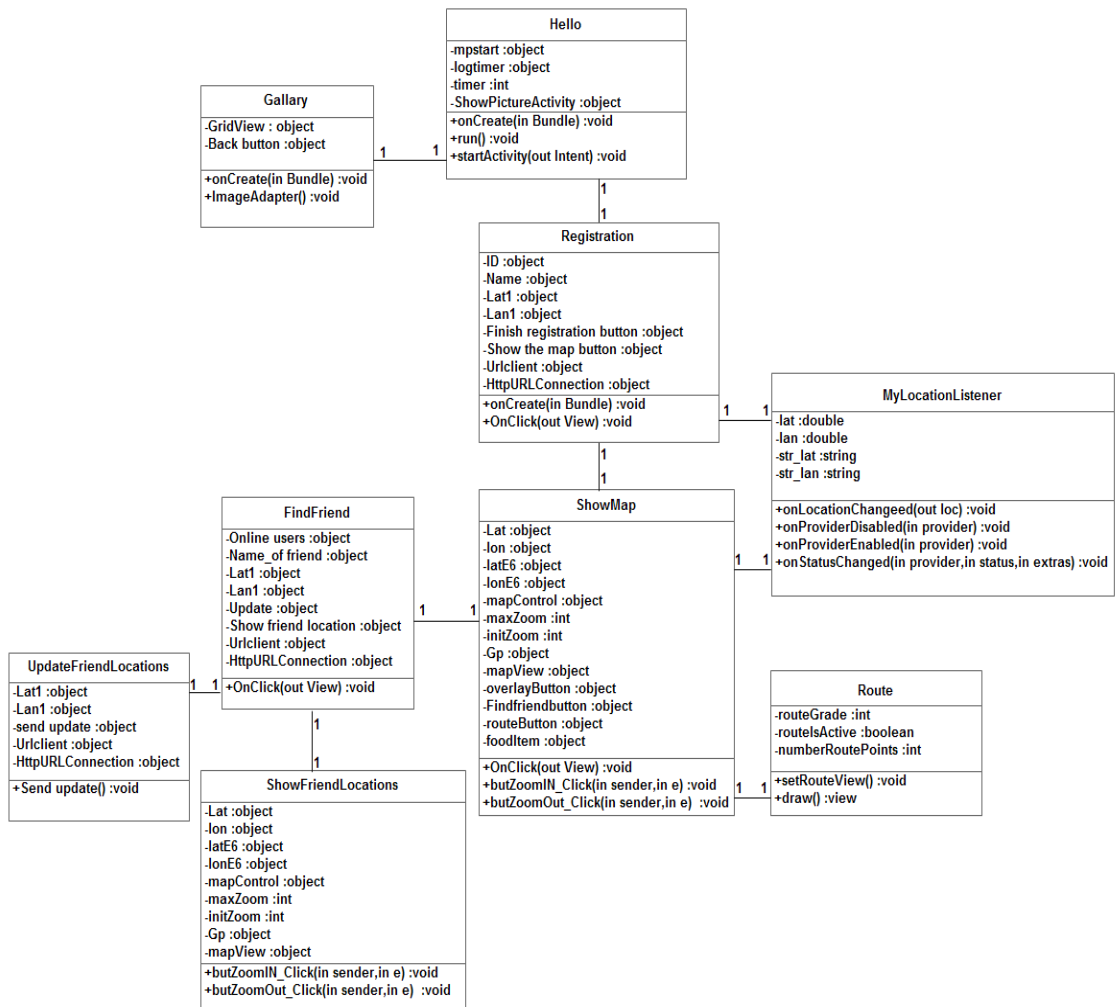


Figure 4.1: Mobile UML diagram

□ Register class:

This class will be used to allow the user to enter its information and register to system.

➤ Variables:

**ID:** an editable text view that will be filled by user, to hold the id of user.

**Name:** an editable text view that will be filled by user, to hold the name of user.

**Lat1:** an editable text view will be filled from GPS satellites with the latitude value.

**Lan1:** an editable text view will be filled from GPS satellites with the longitude

value.

**Finish registration button:** Push-buttons can be pressed, or clicked, by the user to perform an action; the action here is to post user information to main server.

**Show the map button:** Push-buttons can be pressed, or clicked, by the user to perform an action, the action here is to show map with system options.

**Urlclient:** an HTTP client required to execute HTTP requests while, handling cookies, authentication, connection management, and other features.

**HttpURLConnection:** after HTTP request accepted by server, we will use HttpURLConnection to open connection between web server and client.

➤Methods:

**onCreate:** Called when the activity is starting, this is where the initialization for the textfields should go.

**OnClick :** This is called when the user either touches the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses the suitable "enter" key or presses down on the trackball.

In our system we have two OnClick methods:

- The first method is finish registration: this method send registration was filled by user to Sarver, to be stored in database.
- The second method is show the map: this method is to start new activity.

□ **Show Map class:**

This class will be used after completing the registration process in order to select the services which system offers. It consists of:

➤Variables:

**Lat:** the value of latitude has been sent by first activity.

**Lon:** the value of longitude has been sent by first activity.

**latE6:** Convert lat in degrees into integers in micro degrees.

**lonE6:** Convert long in degrees into integers in micro degrees.

**mapControl:** is variable of A utility class to manage panning and zooming of a

map.

**maxZoom:** is integer type used to set the max zoom of map.

**initZoom:** is an integer type to initialize the first level map zoom to be shown.

**Gp:** is variable of An immutable class representing a pair of latitude and longitude, stored as integer numbers of micro degrees.

**mapView:** is MapView variable to Use the Google Maps library, to shows a map that user can pan and zoom it.

**overlayButton:** Push-buttons can be pressed, or clicked, by the user to perform an action, the action here is to put marker to show restaurant locations on map.

**Findfreindbutton:** Push-buttons can be pressed, or clicked, by the user to perform an action, the action here is to start new activity.

**routeButton:** Push-buttons can be pressed, or clicked, by the user to perform an action, the action here is to show three statics routes on map.

**foodItem:** array of overlay type ,which contains the coordinates of restaurant in Hebron old town.

➤Methods:

**OnClick:** This is called when the user either touches the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses the suitable "enter" key or presses down on the trackball.

In our system we have two OnClick methods:

- SetOverlay1: method used to put marker according to coordinates that was stored in food item array.
- SetOverlay2: start findFreind Activity.
- Route: start route activity.

**butZoomIN\_Click:** this is an event handler method that will be applied when the zoomINbutton in the screen clicked, to set the Boolean zoomIN value to be true and zoomOUT false.

**butZoomOut\_Click:** this is an event handler method that will be applied when the zoomOUT button in the screen is clicked, to set the Boolean zoomOUT value to be true and zoomIN false.

□ **Gallery class:**

This class will be used after user selects the gallery Services which system offers. It consists of:

➤ Variables:

**GridView:** that displays items in a two-dimensional, scrollable grid. The grid items are automatically inserted to the layout using a ListAdapter.

**Back button:** Push-buttons can be pressed, or clicked, by the user to perform an action; the action here is to go back to the main screen.

➤ Methods:

**OnCreat:** Called when the activity is starting, this is where the initialization for the textfields should go.

**Image Adapter:** used for display image in listview in each row. Then you have to use custom adapter for that, and using your row layout you can inflate row and fill it in listview.

□ **FindFreind class:**

This class will be used after user selects the findFreind Services which system offers. It consists of:

➤ Variables:

**Online users:** an editable text view that will be filled with names of online users by server from database.

**Name\_of friend:** an editable text view that will be filled by user, to hold the name of user need to know where his /her location.

**Lat1:** an editable text view that will be filled by server with latitude value from database according to the name\_of friend.

**Lan1:** an editable text view that will be filled by server with longitude value from database according to the name\_of friend.

**Update** : Push-buttons can be pressed, or clicked, by the user to perform an action, the action here is to start new activity.

**Show friend location**: Push-buttons can be pressed, or clicked, by the user to perform an action; the action here is to start new activity.

**Urlclient**: an HTTP client required to execute HTTP requests while, handling cookies, authentication, connection management, and other features.

**HttpURLConnection**: after HTTP request accepted by server, we will use HttpURLConnection to open connection between web server and client.

➤Methods:

**OnClick** : This is called when the user either touches the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses the suitable "enter" key or presses down on the trackball.

In our system we have two OnClick methods:

- Get online user: method used to send request to get online users to server.
- Show friend location: to start new activity.
- Update: to start new activity.

□ **Show friend location class**:

This class will be used after user select the findFreind Services which system offers. It consists of:

➤Variables:

**Lat**: the value of latitude has been sent by first activity.

**Lon**: the value of longitude has been sent by findFreind activity.

**latE6**: Convert lat in degrees into integers in micro degrees

**lonE6**: Convert long in degrees into integers in micro degrees

**mapControl**: is variable of A utility class to manage panning and zooming of a map.

**maxZoom**: is integer type used to set the max zoom of map.

**initZoom**: is an integer type to initialize the first level map zoom to be shown.

**Gp:** is variable of an immutable class representing a pair of latitude and longitude, stored as integer numbers of micro degrees.

**mapView:** is MapView variable to use the Google maps library, to shows a map that user can pan and zoom it.

➤Methods:

**butZoomIN\_Click:** this is an event handler method that will be applied when the zoomINbutton in the screen clicked, to set the Boolean zoomIN value to be true and zoomOUT false.

**butZoomOut\_Click:** this is an event handler method that will be applied when thezoomOUT button in the screen clicked, to set the Boolean zoomOUT value to be true and zoomIN false.

#### □ Update friend locations class:

This class will be used after user select the findFreind Services which system offers. It consists of:

➤Variables:

**Lat1:** an editable text view will be filled from GPS satellites with the latitude value.

**Lan1:** an editable text view will be filled from GPS satellites with the longitude value.

**Name:** an editable text view that will be filled by user, to hold the name of user.

**Send update:** Push-buttons can be pressed, or clicked, by the user to perform an action, the action here is to send request to store updated location coordinates.

**Urlclient:** an HTTP client required to execute HTTP requests while, handling cookies, authentication, connection management, and other features.

**HttpURLConnection:** after HTTP request accepted by server , we will use HttpURLConnection to open connection between web server and client.

➤Methods:

**Send update:** send connection request to server, after server accept request ,it will send updated coordinates to server to be stored in database according to name of users who sent update request.

#### □ **MyLocationListener class:**

This class will be used to register the listener with the Location Manager to receive location updates.

#### ➤ Variables:

**Lat, lan:** double variable change continuously with the location, returns the latitude and longitude of the location of travelling.

**str\_lat, str\_lan:** string variables results from converting the coordinates from double type into string type.

#### ➤ Methods:

**onLocationChanged:** Called when the location has changed, contains the coordinates data and the posting process and the timer for posting parameter, the new location, as a Location object.

**onProviderDisabled:** Called when the provider is disabled by the user. If request Location Updates is called on an already disabled provider, this method is called immediately. Its parameter, the name of the location provider associated with this update.

**onProviderEnabled:** Called when the provider is enabled by the user. Its parameter, the name of the location provider associated with this update.

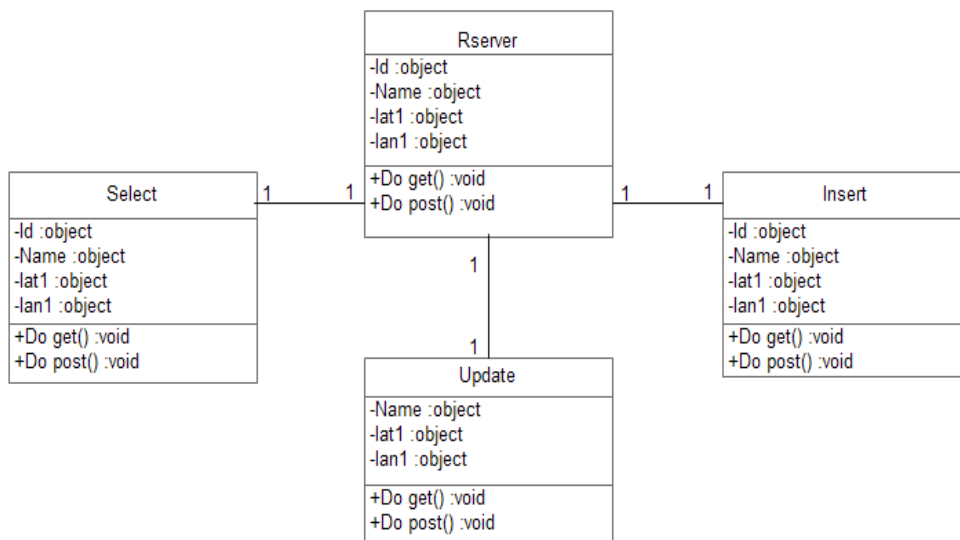
**onStatusChanged:** Called when the provider status changes. This method is called when a provider is unable to fetch a location or if the provider has recently become available after a period of unavailability.

## 4.2.2 Server Software UML diagram

A detailed description for our web server will be implemented using UML model diagram; that will describe the main classes, the attributes used in each class, methods that were defined, and a detailed description of these methods.

The Server software is described as shown in figure (4.2).





**Figure 4.2: Server UML diagram**

### 4.2.2.1 Classes Implementation

Our system web server software contains a group of classes that interact with each other, to make the whole system work properly, and complete the system objectives, and these classes are:

□ **Select class:**

This class will be used to select name and coordinates of users it consists of:

➤ **Variables:**

**Id:** store parameter that received from client.

**Name:** store parameter that received from client.

**lat1:**store parameter that received from client.

**lan1:**store parameter that received from client.

➤ **Methods:**

**Do get:** The doGet () method is the method inside a servlet that gets called every time a request from client, in order to select user client connect servlet with microsoft access database as the following code:

```

Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
String ds="pro1";
String dbUrl="jdbc: odbc:"+ds;
Connection conn = DriverManager.getConnection (dbUrl,"","");
Statement s= (Statement) conn.createStatement ();
String stm="SELECT name FROM reg  ";
s.execute (stm);
rs=s.getResultSet ();

```

#### □Insert class:

This class will be used to insert registration information in database for each user login to the system, it consists of:

#### ➤Variables:

**Id:** store parameter that received from client.

**Name:** store parameter that received from client.

**lat1:**store parameter that received from client.

**lan1:**store parameter that received from client.

#### ➤Methods:

**Do get:** The doGet() method is the method inside a servlet that gets called every time a request from client, in order to insert user registration info client connect servlet with microsoft access database as the following code:

```

Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
String ds="pro1";
String dbUrl="jdbc: odbc:"+ds;
Connection conn = DriverManager.getConnection (dbUrl,"","");
System.out.println (".....");
Statement s= (Statement) conn.createStatement ();
String stm="INSERT INTO reg VALUES
('"+ID+"', '"+name+"', '"+lt1+"', '"+ln1+"') ";
s.execute (stm);

```

#### □ Update class:

This class will be used to update users' coordinates in database, it consists of:

##### ➤ Variables:

**Name:** store parameter that received from client.

**lat1:** store parameter that received from client.

**lan1:** store parameter that received from client.

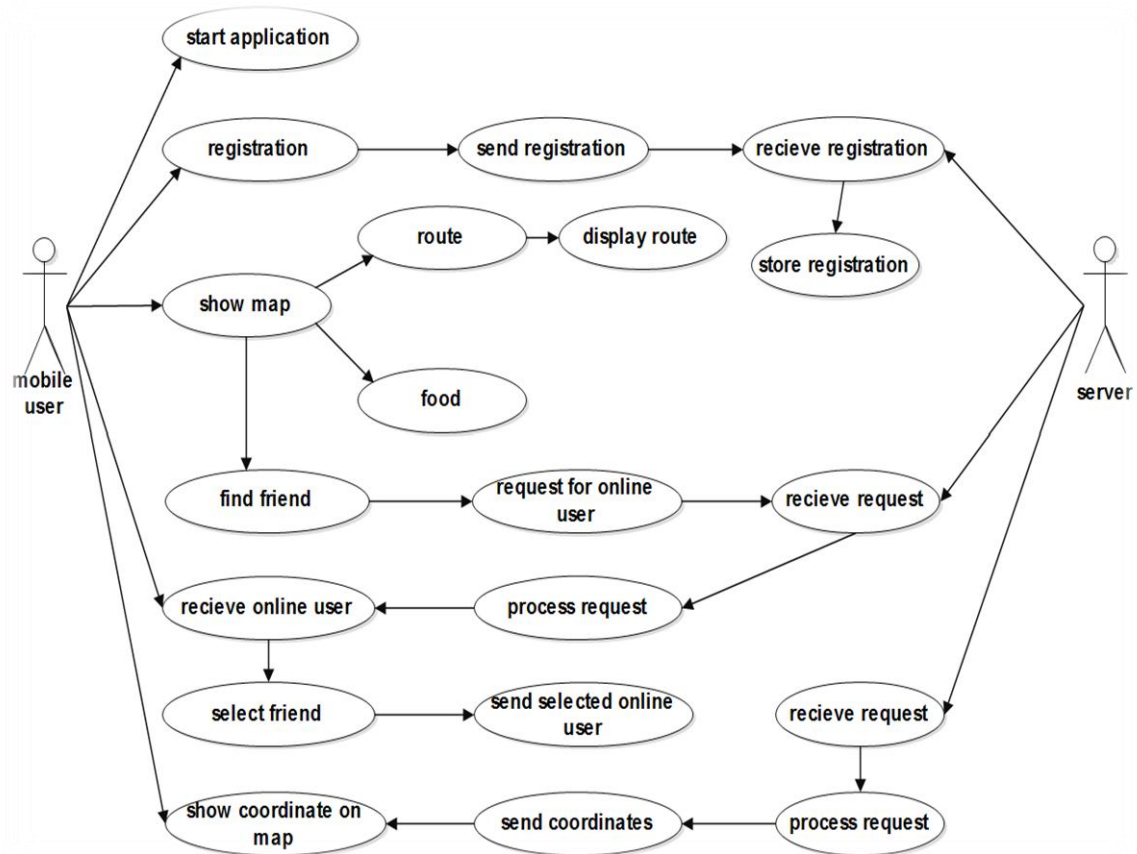
##### ➤ Methods:

**Do get:** The doGet () method is the method inside a servlet that gets called every time a request from client, in order to update user coordinate after servlet connect with microsoft access database as the following code:

```
Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");  
String ds="pro1";  
String dbUrl="jdbc: odbc:"+ds;  
Connection conn = DriverManager.getConnection (dbUrl, "", "");  
System.out.println (".....");  
Statement s= (Statement) conn.createStatement ();  
String stm1= "UPDATE reg SET lt1 = '"+lt1+"', ln1 = '"+ln1+"'  
WHERE name='"+name1+"'"  
s.execute (stm);
```

### 4.3 System use cases

A detailed of how the system outside users, and the system actions will interact with each other during the process of the system, will be described in the system use cases clearly in this section, as shown in figure (4.3) :



**Figure 4.3: System use case diagram**

From the figure we can note that:

- Two actors mainly interact to our system, the mobile user, and server.
- The mobile user who will start the mobile application.
- Then he will fill the required registration fields, and activate the action of send the registration request from mobile to the server.
- This action will activate the process of showing the old town map with current user location on it, and it will activate the options of system, where mobile user needed to select.
- If a mobile user selects food option, this will show the restaurant location on map in the old town.
- If a mobile user selects route option, this will show basic routes in the old town.

- If a mobile user select friend option, this will start friend finder screen, in this screen different actions can user do :
  1. If mobile user select show online users option, it will activate connection to server, then server receive request reply with the name of all online users that running application at this moment.
  2. Then mobile user can now select one of online users that needed to know where their location .then sends the selected name to server.
  3. Server will search in its database about coordinate of the name of user was selected.
  4. Mobile receives the coordinate of the user that was selected before, and then shows its coordinates on map.
  5. If mobile user selects update option, this will show new screen, in this screen mobile user can send new current location to server to be updated.

## 4.4 System sequence diagram

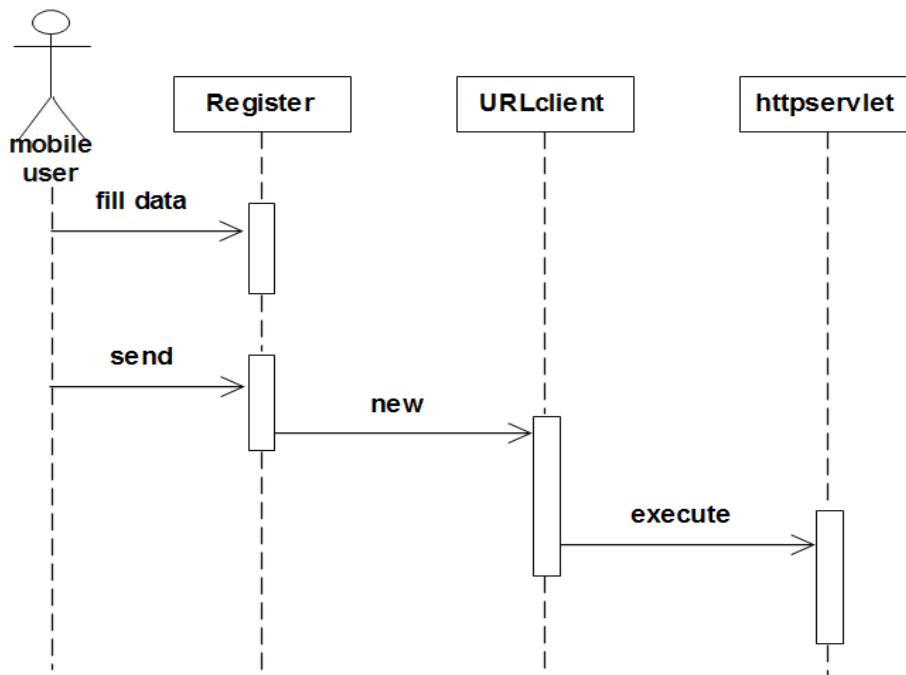
For each of the use cases a sequence diagram will be created in this section, in order to show which objects are needed to perform this case, how they will interact with each other, and the time sequence of the process steps:

### 4.4.1 Fill and send registration sequence diagram

As shown in figure (4.4); we can realize the time sequence for this use case, where the mobile user needed to fill the required registration data fields, then he click on the “finish registration ” button, this will open new http url connection, get the text from the fields, then it will execute.

**And the main methods in the system needed for this process are:**

```
publicvoid onCreate(Bundle savedInstanceState)
publicvoid onClick(View v)
publicvoid run()
```



**Figure 4.4: Send registration request sequence diagram**

#### **4.4.2 Send GPS coordinates sequence diagram**

As shown figure ( 4.5) ; we can realize the time sequence for this use case, where after registration data being send, the system will go to the tracking form automatically, and start a location manager that start a location listener ,which request the coordinates if it's available by the GPS receiver from the GPS Provider, and if it not available it will be read from the mobile network provider, and after coordinates being read this will open httpclient that execute httppost, and before post it will request the coordinates from location listener and arrange them into an array list and finally execute the post process to the server httpURL, and this process will be repeated during the listening to GPS or network provider and on location change case.

**And the main methods in the system needed for this process are:**

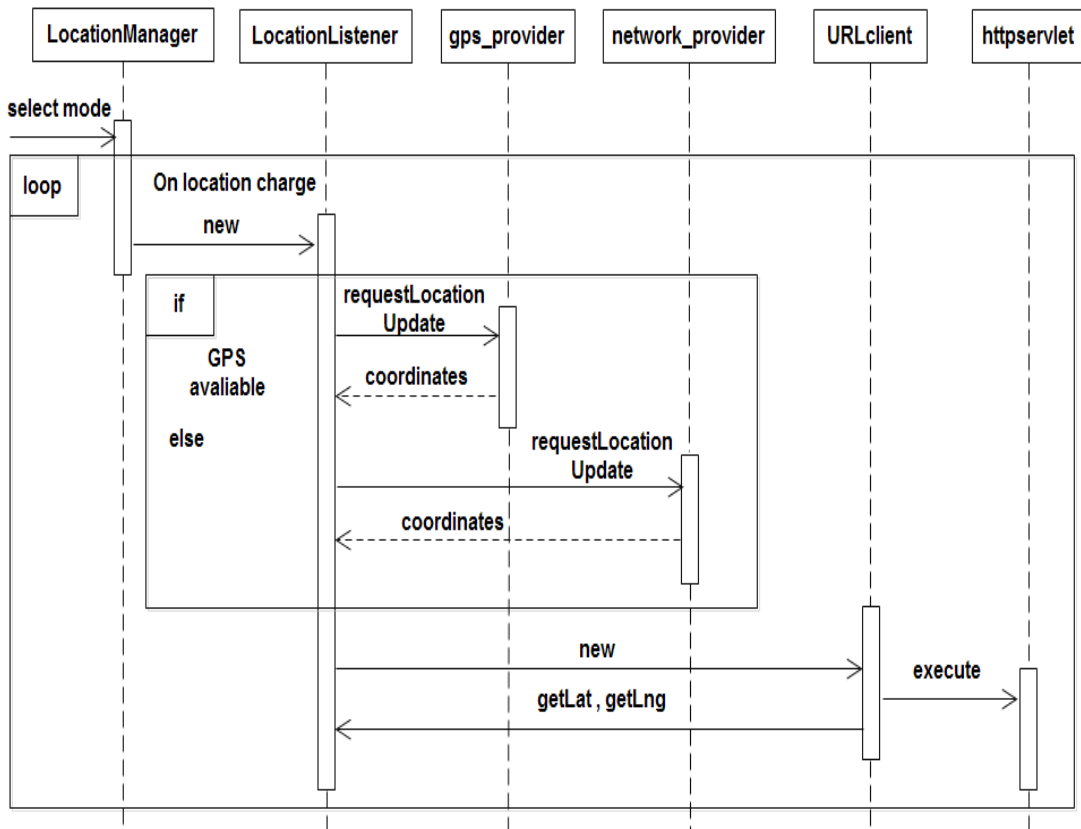
**protected void** onCreate(Bundle savedInstanceState)

**public void** onLocationChanged(Location loc)

**public void** onProviderDisabled(String provider)

**public void** onProviderEnabled(String provider)

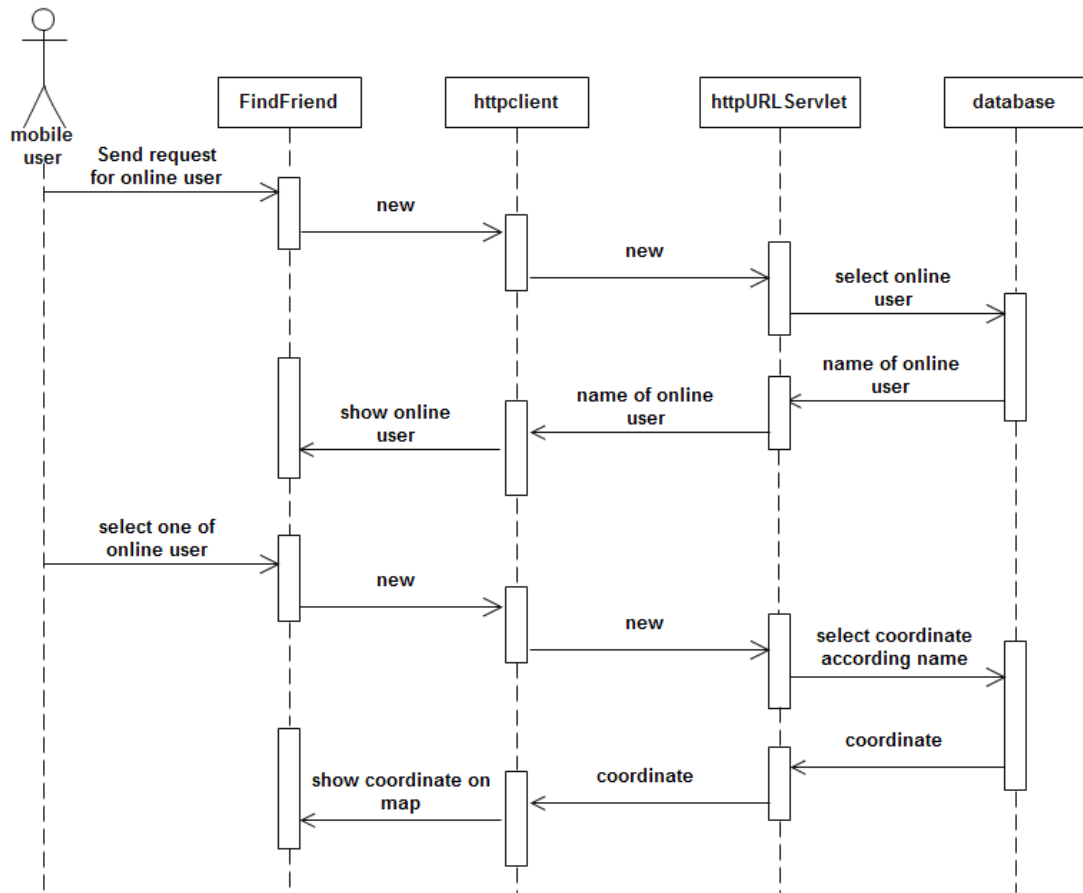
**public void** onStatusChanged(String provider, **int** status, Bundle extras)



**Figure 4.5: Get and send GPS coordinates sequence diagram**

### 4.4.3 Find Friend sequence diagram

As shown in figure (4.6); we can realize the time sequence for this use case, where the mobile user needed to select friend option.



**Figure 4.6: Find Friend sequence diagram**



# 5

## *CHAPTER FIVE*

### *SYSTEM TESTING*

---

- 5.1 Overview**
- 5.2 Sub-System Testing**
- 5.3 Installation and preparing the system**
- 5.4 Testing Scenarios**

## **5.1 Overview**

In this chapter, the whole testing stage will be described, including a test of an entire interconnected set of components and software for the purpose of determining proper functions and achieving the desired goals of the system. We will talk about testing of each part of the system, describe the scenarios, represent each test and its errors, challenges and modifications, and also we will figure out the error rate.

## **5.2 Sub-System Testing**

The main aim of this testing part is to test the main operations in the system. There are five main operations in the system. The first one is getting the correct coordinates. The second operation is posting data to the web server. The third is receiving the data and processing it to register tourist. And the fourth operation is show the map with options. The last one finds friend operation.

## **5.3 Installation and preparing the system**

At this stage, all system parts must be ready to be tested at the two sides, mobile application, and the server side.

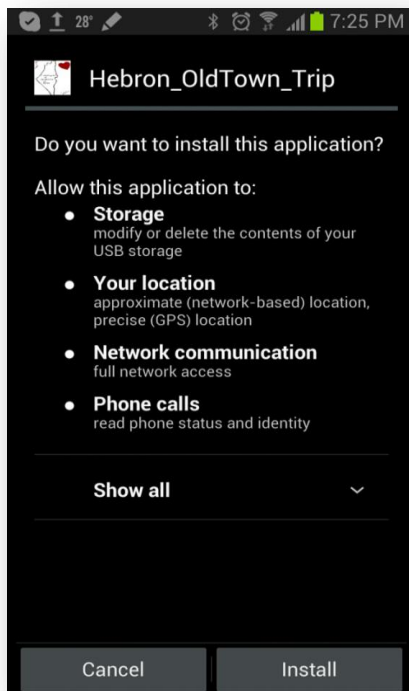
### **5.3.1 Mobile application**

An application has already been built using the Eclipse environment, to use this application it must be installed to the mobile, and to perform this we can do one of the following ways:

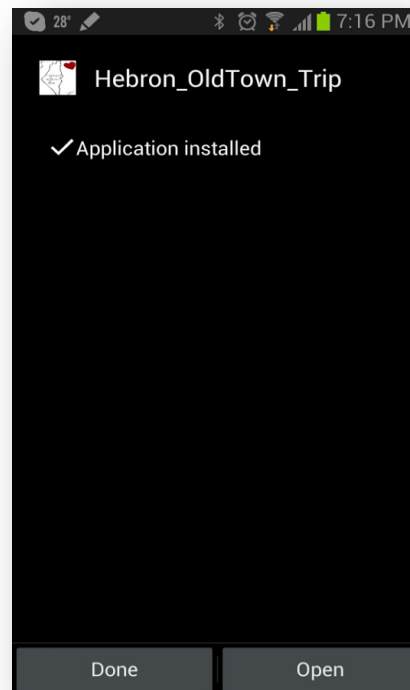
- The first is to connect the mobile phone (Galaxy SII) by the usb cable or Bluetooth connection to the PC at which the workspace of Eclipse is created. Then the following file must be copied to the mobile SD card.  
C:\Users\iyad\workspace\demor\_sample\bin\demor\_sample.apk

Open the ".apk" file, a message will appear at the screen of the mobile as shown in figure(5.1),show the name of the application and the permutations needed for it to works probably. In our application we need to access the GPS location to get the location of the tourist, and we need a full internet access for the communication between mobile (client) and PC (server) while posting data.

Click the install button to allow the installation of the application as shown at figure (5.2).



**Figure 5.1: Opening apk file**

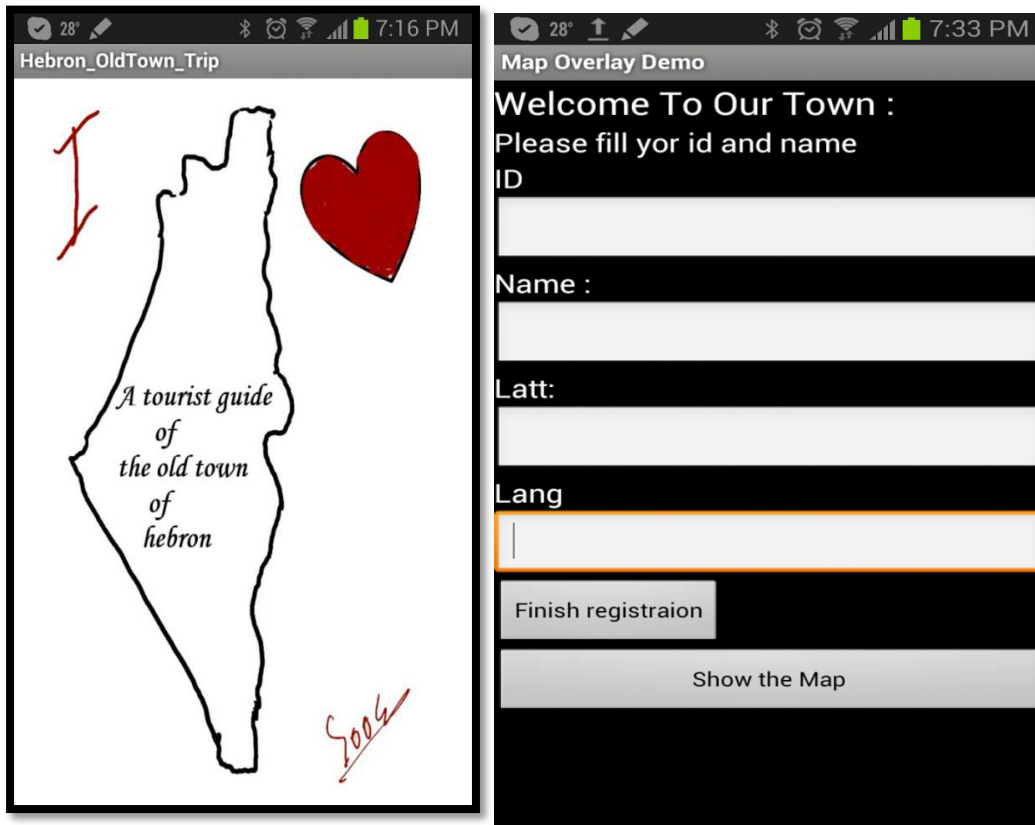


**Figure 5.2: Installing complete**

When the installation is completed, as shown in figure (5.2), click the open button to start the application:

- The second way to install the application is to connect the mobile phone directly to the Eclipse and define it as one of the active running android

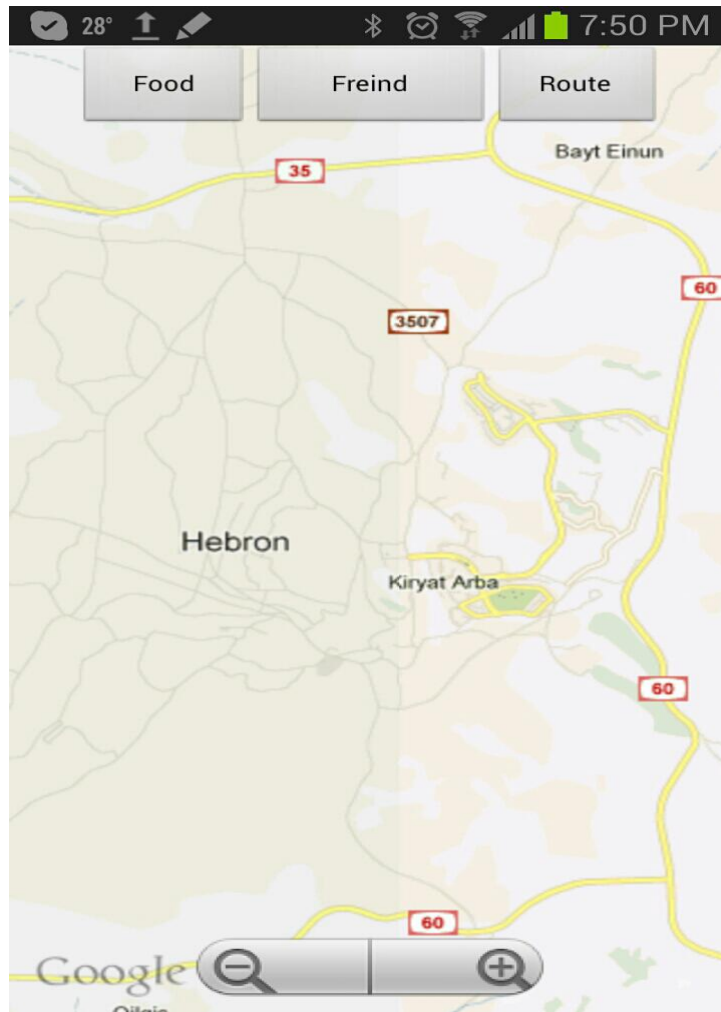
device, choose it manually to run the application on. After choosing one of the previous ways to install the application, it will be opened and became ready to use, as shown in figure (5.3).



**Figure 5.3: Welcome screen Figure 5.4: Registration**

Now, the application on the mobile is ready to be used by the user. This welcome screen will last only 7 second then immediately a new module will be opened, as shown at figure (5.4), the user must enter its information then click the "finish registration information" button.

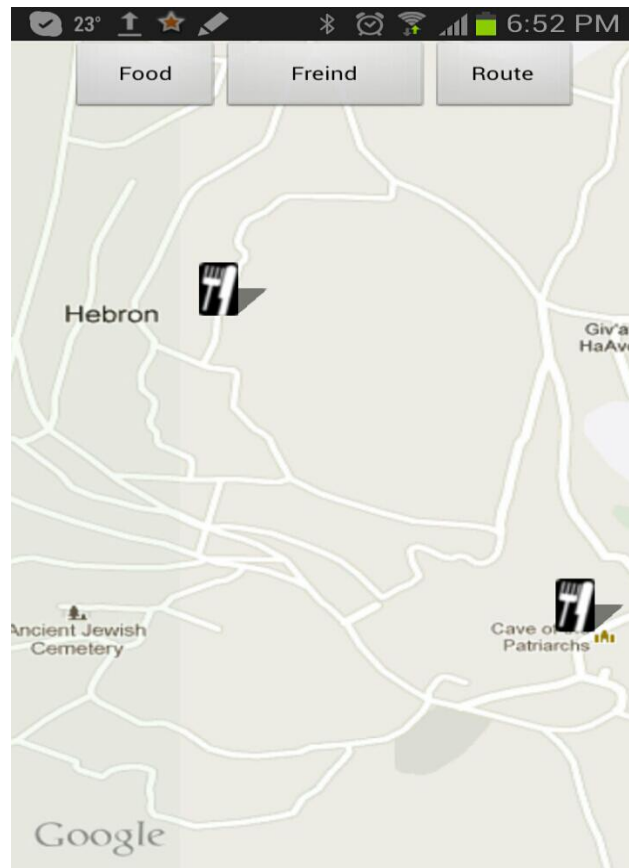
As shown, the required information is: the ID number, the name of the tourist, lat, lan. This information will be needed at the server side. All the fields are required, so if the user tries to skip this step and click the finish registration button without filling the fields a message will appear at the screen and tell him that: "All field are required, Fill it please".



**Figure 5.5: Old Town map and options**

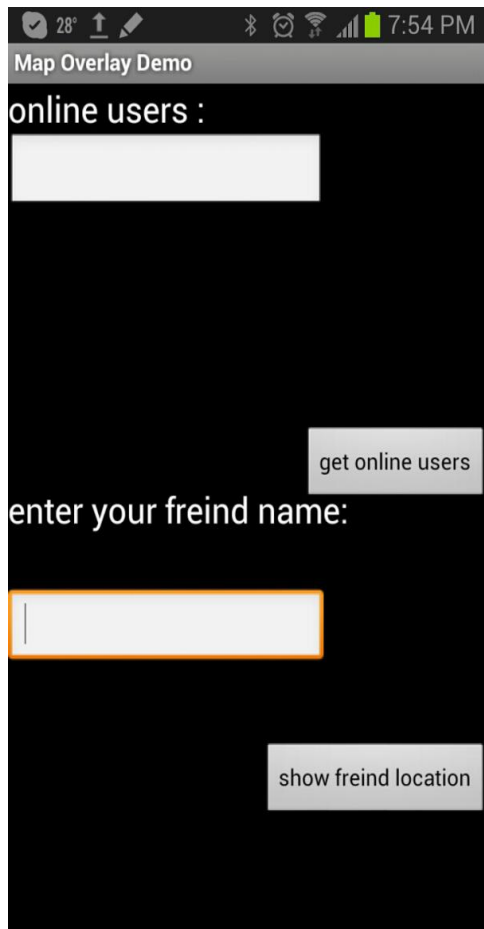
Pressing the finish registration button will post the data to the main server and move to the next module, map with options, as shown at figure (5.5).

1. For user want to show famous Restaurants location on map figure (5.6).

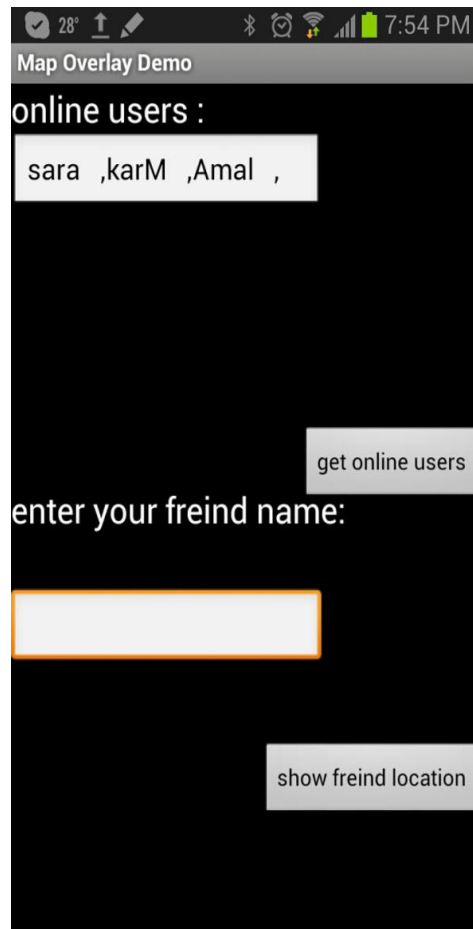


**Figure 5.6: Famous Restaurant Town**

2. For user want to show where his friends location on map as shown at figure (5.7).
3. If user wants to know online users at this moment tourist can click on button get online user.

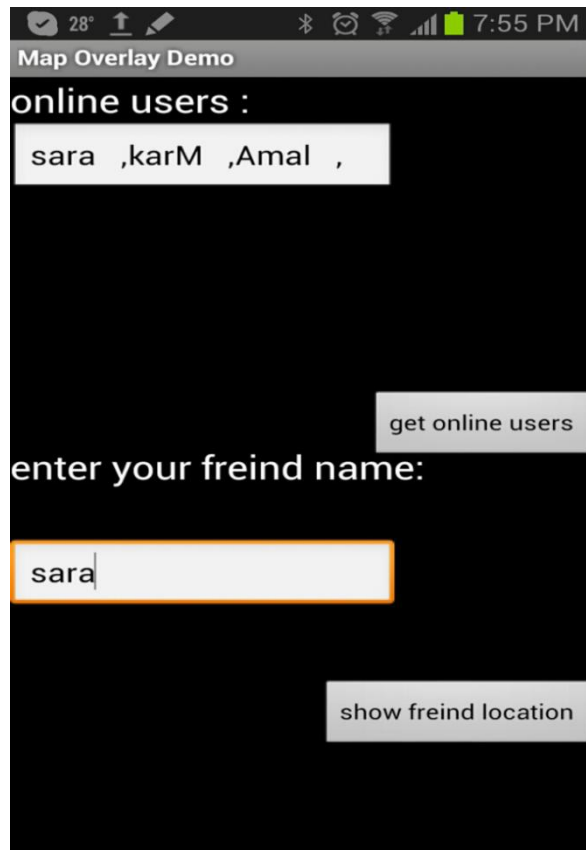


**Figure 5.7: Find Friend**



**Figure 5.8: Name of online users**

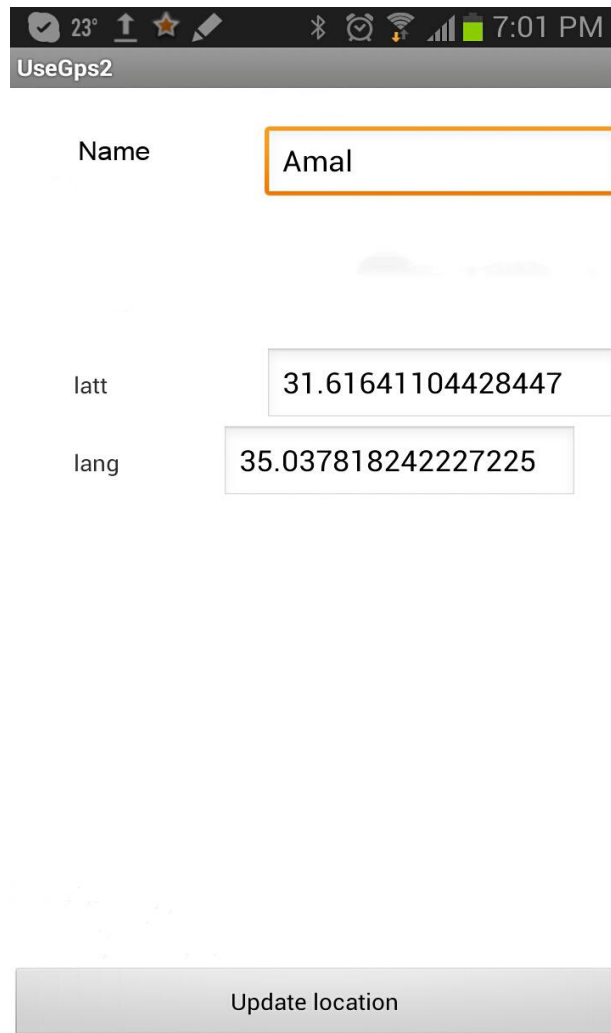
- After user click on button get online user ,the name of online users will be shown in text field as shown at figure(5.8)
- User writes the name of user, that shows his location. as shown at figure(5.9)
- User click show friend location button to show the location of the selected user on map.



**Figure 5.9: User enter your friend name**

- If user change its location and want to send its new location coordinate to the server, click on update button move to the next module.
- User get current location coordinate and show it in textfield, then user write his name in textfield as shown at figure (5.10).
- User writes his name in textfield.
- The name and coordination send to the server to update its location according his name.





**Figure 5.10: Update location figure**

4. If user clicks on route button move to the next module, map shown static route, how reach to the most important location in old town of Hebron as shown at the figure (5.11).

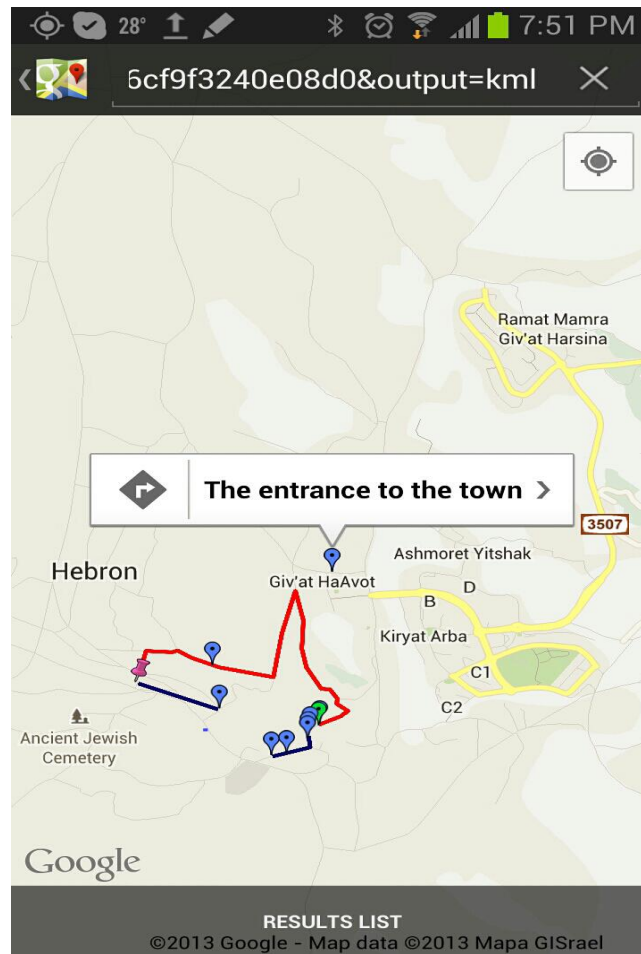


Figure 5.11: Map shown static route

## 5.4 Testing Scenarios

In this section, the scenario of starting the system with all its parts will be mentioned, including all the necessary steps which include:

- Download the mobile application and open it.
- Open the web server.
- At the mobile, fill the registration information and click the "finish registration" button.
- After the acceptance, the mobile application will start the using the application; if the GPS receiver is off, the application will inform the driver to turn it on by a text appears at the screen bottom. So:
  - If GPS is off, message= "GPS Disabled, turn on to start tracking ".
  - If GPS is on, message="GPS Enabled".

- Enable the GPS, the mobile will start receiving GPS signals and calculate its location's coordinates then show them on the screen and post them to the web server. At the same time will be ready to accept new registration request.

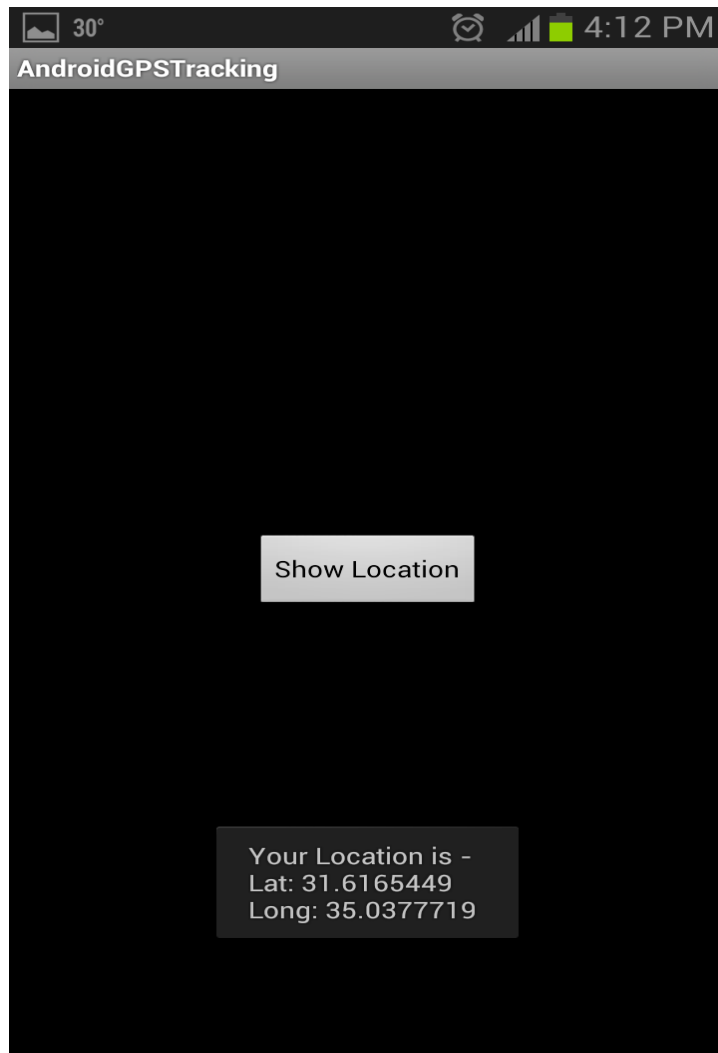
To make a test, the previous scenario including all of the previous steps is done in order. We have performed a number of tests to explore if the system works probably and to figure out the errors and challenges then make modification and corrections.

### **5.4.1 TEST 1**

The first test is applied over a small distances, this is done at the university. A person carries the mobile; open the application and start walking with a medium speed around building B to building C, with one mobile.

#### **Results**

GPS coordinates changed every 1s, and the distance 10 meters, so there was a rate error high, and in the accuracy of determined location was very low.



**Figure 5.12: First GPS reading**

## **Errors**

While testing the system the following errors happened:

- When the mobile sends the registration request to the server, it starts directly sending coordinates, before the server complete the accepting and registration of the new user, and the car information forum stay empty.
- Accuracy very low.

## **Challenges**

The mentioned errors bring the following challenges:

- Give the server the enough time to complete the registration and acceptance process.

## Modifications

The following modification is done to solve the errors and meet the challenges:

### Before modification:

```
TEN_SECONDS = 0;  
TEN_METERS = 0;  
LocationListener mlocListener = new MyLocationListener ();  
mlocManager.requestLocationUpdates (LocationManager.GPS_PROVIDER,  
TEN_SECONDS, TEN_METERS, mlocListener);
```

### After modification:

```
TEN_SECONDS = 10000;  
TEN_METERS = 10;  
TWO_MINUTES = 1000 * 60 * 2;  
LocationListener mlocListener = new MyLocationListener ();  
mlocManager.requestLocationUpdates (LocationManager.GPS_PROVIDER,  
TEN_SECONDS, TWO_MINUTES, mlocListener);
```

## 5.4.2 TEST 2

Another test is performed after applying the previous modifications, to check the system state after solving the errors. It's also over small distances, where a person moves between building C and B inside the university.

## Errors

The errors that we faced previously are limited, because GPS system reading has errors. At this test and the results are acceptable and meet the expectations.

# 6

## *CHAPTER SIX*

### *CONCLUSION AND*

### *RECOMMENDATION*

---

- 6.1 Overview**
- 6.2 Conclusion and Achievement**
- 6.3 Challenges**
- 6.4 Future work Recommendation**

## 6.1 Overview

In this chapter, we will mention what we achieved in this project and the conclusion for all things that we have done, also we will talk about the challenges that we faced and ending wither commendation needed for the future work.

## 6.2 Conclusion and Achievement

Almost all the goals of our system have been achieved. In this section the main achievements of the system are discussed and the ways of achieving it.

- We built a mobile application for given the accurate location, of the travelling tourist and send it to the server.
- We built find friend service; we were able to show friend location on map.
- We draw the statics routs that show how to reach to important places in old town of Hebron.
- We were able to show the restaurant location on map.

## 6.3 Challenges

At this project we have faced some critical points that still need a solution for them. Such as:

- In this system ,was difficult to get clear map for the old town of Hebron, and there was many option to get map:
  1. Use map as picture related with GPS coordinates, there was errors appear: Accuracy in converting GPS coordinates to pixels to be shown on picture very low as shown in figure(6.1):



**Figure 6.1: Image map with GPS coordinates**

2. Use Google map, we get an API key for our region, and to open for the first time on the old town map according to coordinate was taken before.

## 6.4 Future work Recommendation

In this project, there are some ideas that could be done or added to improve its performance, or add some capabilities, some techniques that are efficient and meet user needs. Some of these ideas are mentioned below:

- System could be improved by a connection with social accounts of users such like (Facebook, twitter).so it can import user information.
- System could be improved by dynamic routers.
- System could be improved by using timers, tell user how much time needed to reach the location he wants.



## REFERENCES:

[1]World Mate

Available at <http://andronica.com/2012/03/the-best-apps-for-android-travelers/>

[2]Field trip

Available at <http://www.tech-wd.com/wd/2012/09/28/field-trip-app/>

[3] Google map

Available at <http://www.makeuseof.com/tag/top-3-free-offline-gps-apps-android/>

[4] M. Kennedy, "The Global Positioning System and GIS an Introduction", fanrica, United State of America, 1996.

[5] Image from Top gps reviews.

Available at <http://topgpsreviews.net/wp-content/uploads/2010/08/gps-satellite-orbits.jpg>

[6] Wikipedia, the free encyclopedia, "World Geodetic System".

Available at [http://en.wikipedia.org/wiki/World\\_Geodetic\\_System](http://en.wikipedia.org/wiki/World_Geodetic_System)

[7] D. Bowler, A. Mayne, D. McNally, T. Wakefield, Introduction To Mobile

Communication Technology Services Markets, Auerbach Publication Informa, New York, 2007.

[8] Image from Bundesamt.

Available at

[https://www.bsi.bund.de/SharedDocs/Bilder/DE/BSI/Publikationen/GSM/gsm\\_e2\\_jpg.jpg?\\_\\_blob=normal&v=2](https://www.bsi.bund.de/SharedDocs/Bilder/DE/BSI/Publikationen/GSM/gsm_e2_jpg.jpg?__blob=normal&v=2)

[9] Search mobile computing, "GPRS".

Available at <http://searchmobilecomputing.techtarget.com/definition/GPRS>

[10] Search mobile computing, "GPRS".

Available at <http://searchmobilecomputing.techtarget.com/definition/GPRS>

[10] P. Calduwel Newton, DR. L. Arockiam, and Tai-hoon Kim, International Journal of Advanced Science and Technology, "A Quality of Service Strategy to Select Coding Schemes in General Packet Radio Service System".

Available at <http://www.sersc.org/journals/IJAST/vol7/1.pdf>

[11] Wikipedia, the free encyclopedia, "Mobile operating system".

Available at [http://en.wikipedia.org/wiki/Mobile\\_operating\\_system](http://en.wikipedia.org/wiki/Mobile_operating_system)

[12] From Linux to Android, Android Internals for Linux Developers, "Introduction to Android Architecture".

Available at <http://technogeeks.com/Courses/Android-Excerpt.pdf>

[13] B. Venners, artima developer, "Introduction to Java's Architecture".

Available at <http://www.artima.com/insidejvm/ed2/introarchP.html>

[14] Android compares, "Why Android".

Available at <http://androidcompare.com/#ixzz1h91s3wrS>

[15] Dijkstra's Algorithm Pseudocode

Available at

<http://www.cc.gatech.edu/aimosaic/students/Psychstudents/Shikano/Projects/AlgoNet/spa/code.html>

**[16] Wikipedia, the free encyclopedia, "Google Earth".**

**Available at** [http://en.wikipedia.org/wiki/Google\\_Earth](http://en.wikipedia.org/wiki/Google_Earth)

**[17] Image map.**

**Available at** [http://en.wikipedia.org/wiki/Image\\_map](http://en.wikipedia.org/wiki/Image_map)

**[18] Wikipedia, the free encyclopedia, "Google Maps".**

**Available at** [http://en.wikipedia.org/wiki/Google\\_Maps](http://en.wikipedia.org/wiki/Google_Maps)