ST. XAVIER'S COLLEGE

(Affiliated to Tribhuvan University)

Maitighar, Kathmandu

Final Year Project Report
On
"KHOJ"
A GPS BASED SMS ENABLED ATM TRACKING SYSTEM
[CSC- 404]

For the partial fulfillment of Bachelor's Degree of Computer Science and Information Technology

Under the Supervision of
Mr. Vishnu Kumar Rana
Head of Department
Department of Computer Science
St. Xavier's College

Submitted by
Rashna Neupane (T.U. Exam Roll No. 857/067)
Sajan Rai (T.U. Exam Roll No. 864/067)

Submitted to
ST. XAVIER'S COLLEGE
Department of Computer Science
Maitighar, Kathmandu, Nepal

October, 2014

“KHOJ”
A GPS BASED SMS ENABLED ATM TRACKING SYSTEM

[Course Code: CSC-404]

A final year project submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science in Computer Science and Information
Technology awarded by Tribhuvan University

Submitted by
Rashna Neupane (T.U. Exam Roll No. 857/067)
Sajan Rai (T.U. Exam Roll No. 864/067)

Submitted to
ST. XAVIER’S COLLEGE
Department of Computer Science
Affiliated to Tribhuvan University

October, 2014

ST. XAVIER'S COLLEGE
MAITIGHAR, KATHMANDU, NEPAL
Post Box :7437
Contact: 4221365,4244636
Email: ktm@sxc.edu.np

सेन्ट जेभियर्स कलेज
माईतीघर, काठमाडौं, नेपाल
पो.ब.नं. : ७४३७
फोन : ४२२१३६५,४२४४६३६
ईमेल : ktm@sxc.edu.np

Celebrating 25 years
1988-2013

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the Department of Computer Science for acceptance, a project report entitled **"KHOJ"- A GPS BASED SMS ENABLED ATM TRACKING SYSTEM** submitted by **Rashna Neupane (T.U. Exam Roll No. 857/067)** and **Sajan Rai (T.U. Exam Roll No. 864/067)** in partial fulfillment for the degree of Bachelor of Science in Computer Science & Information Technology.

…………………………..
Mr. Vishnu Kumar Rana
Project Supervisor
Head of Department
Department of Computer Science
St. Xavier's College

……………………………..
External Examiner
Tribhuvan University

……………………………..
Mr. Vishnu Kumar Rana
Head of Department
Department of Computer Science
St. Xavier's College

ACKNOWLEDGEMENT

Our forthright gratefulness goes to our supervisor **Mr. Vishnu Kumar Rana**, HOD, Department of Computer Science, St. Xavier’s College, for whole hearted support and for providing us the opportunity to undertake this case study.

We would like to extend our sincere gratitude to **Mr. Jitendra Manandhar, Er. Rajan Karmacharya and Mr. Bal Krishna Subedi**, Lecturers, Department of Computer Science, St. Xavier’s College, for their kind and co-operative support, valuable time and guidance as well as suggestions. Their useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

We are also grateful to our teachers **Mr. Nitin Malla, Mr. Sansar Jung Dewan, Miss Shova Shrestha, Er. Sanjay Kumar Yadav, Er. Anil Kumar Sah and Mr. Ganesh Yogi** for their constant support and guidance.

We are thankful to the **Mr. Sunim Shrestha, Mr. Om Nath Shrestha**, **Mr. Bibek Konda** and **Mr. Sagar Rijal**, Lab Assistants, Department of Computer Science, St. Xavier’s College, for their support.

We would like to thank **Sparrow SMS** for providing us with the SMS service required for the implementation of our project objectives.

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during the preparation of this project report.

Thanking You,

Rashna Neupane (T.U. Exam Roll No. 857/067)

Sajan Rai (T.U. Exam Roll No. 864/067)

ABSTRACT

The motivation for every location based information system is to assist with the exact information, at right place in real time with personalized setup and location sensitiveness. Mobile applications can be one of the best ways to keep consumers engaged with a brand as they are on the move. With the increase in demand for smartphones and efficiency of wireless networks, the demand for mobile applications has increased incredibly. Android is one of the most popular open source platforms that offers the developer's full access to the framework API's so as to build innovative applications.

This report describes **"Khoj"**, a GPS based mobile application built on the Android platform that provides location-based services for retrieving map-based directions. The main aim of this project is to build an Android application that helps the users to find direction of nearest ATMs of a particular bank. Mapping enables the user to visualize the relative geographic location in real time. The Location Based Service is used to identify the current location of the user. The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information anywhere on the Earth, where there is an unobstructed line of sight to four or more GPS satellites. It is freely accessible by anyone with a GPS receiver. The position is then displayed, perhaps with a moving map display or latitude and longitude. Once the location of the user is retrieved, he is provided with the options to select the required bank. The application then lists the ten nearest ATMs of the corresponding bank in ascending order with respect to the distance of the ATM from user's current location. Google Map API is then used to display the route of each ATM to the user.

In addition, for the availability and simplification of the service for all kind of mobile phone users, a short code message service is provided, in which the users will be able to query the ATM locations of the particular bank and get the five nearest ATM locations of the preferred bank from the user's current location via the SMS service.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ATM	Automatic Teller Machine
GPS	Global Positioning Sensor
JSON	JavaScript Object Notation
API	Application Programming Interface
GIS	Geographical Information System
JVM	Java Virtual Machine
XML	Extensible Markup Language
JDK	Java Development Kit
SDK	Software Development Kit
SMS	Short Message Services
ADT	Android Development Tools
ER	Extended Relation
API	Application Programming Interface
DFD	Data Flow Diagram

CHAPTER 1: INTRODUCTION

1.1 Background

This report describes Khoj, a GPS based mobile application built on the Android platform that provides location-based services for retrieving map-based directions. The main aim of this project is to build an Android application that helps the users to find direction of nearest ATMs of a particular bank. Mapping enables the user to visualize the relative geographic location in real time. The Location Based Service is used to identify the current location of the user. The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information anywhere on the Earth, where there is an unobstructed line of sight to four or more GPS satellites. It is freely accessible by anyone with a GPS receiver. The position is then displayed, perhaps with a moving map display or latitude and longitude. Once the location of the user is retrieved, he is provided with the options to select the required bank. The application then lists the ten nearest ATMs of the corresponding bank in ascending order with respect to the distance of the ATM from user's current location. Google Map API (Application Programming Interface) is then used to display the route of each ATM to the user.

In addition, for the availability and simplification of the service for all kind of mobile phone users, a short code message service is provided in the application in which the users will be able to query the ATM locations of the particular bank and get the ATM locations of the preferred bank via the SMS service. The short code is a number to which a SMS or text message can be sent. A short code is fewer digits than a 10-digit telephone number. A short code may be specific to one mobile operator or "common" and supported by all major mobile operators.

1.1.1Android Application

As smart phones and tablets become more popular, the operating systems for those devices become more important. Android application uses Android SDK API to manage the GPS Sensor, Google Maps API to show the Map powered by Google Maps, to display the markers about events on the Map. The most important part is acquiring the user's Location. It's important to manage it properly because the aim is to get the most accurate location. Android utilizes GPS and Android's Network Location Provider to acquire the user location. Android's Network Location Provider determines user location using cell tower and Wi-Fi signals, providing location information in a way that works indoors and outdoors. Android provides a best performance model to obtain user location. The application starts listening to location updates just after it started. [1]

1.1.2 ATM

The ATM, or Automated Teller Machine, allows bank customers to withdraw money without the need for a live bank teller. These machines are available all over the world, and customers from one bank may almost always use ATM machines from other banks, though they may be charged a fee. The basic concept and design of the ATM has changed little since its invention, though security and encryption methods are constantly being changed and improved. [2]

1.2 Problem Statement

There might have been number of situations where ATM users might have searched for an ATM for a particular bank nearby to withdraw cash. The users might not be able to withdraw money in a convenient way in case they are unknown about the particular place or the direction to the ATM booth from the location they are located. Also, the users may be interested to know about the ATMs nearby in a particular place. For all these reasons, the mobile application to locate the ATMs is required which can provide the information just with the help of few clicks on the mobile phones. This application can be handy for the users to locate the ATMs in unknown places.

With the development of the application **"Khoj"**, it makes easier way to locate the ATM of the preferred bank for android users, and helps to find the direction between the user and the nearest ATM using the geo locations on the interactive map. In addition to this, a short code message service is provided to non-android users as well, in which the users are able to query the ATM locations of the particular bank in order to get the

ATM locations of the preferred bank via the SMS service making the day to day life of the users more convenient.

1.3 Objectives and Scopes

1.3.1 Objectives

After going through the research, it can be gathered that there is a huge scope of application development in mobile domain. Following the same notion, we can also develop application that can tackle following issues:

- Location positioning technologies
- Query processing
- Cache management

Applications can be developed on Android platform of Open Handset Alliance led by Google. Google provides simulated environment and standard development kit for developing Android applications. [10] We chose Android as it is also open source. The GPS based application can help user to find ATM location of preferred bank indicated by user within certain range. The main purpose of the proposed system is to track the ATMs of the corresponding banks those are available in different places. In addition, it will be able to show the direction from the user's current destination to the nearby ATM of preferred bank. Besides that, this system is planned to provide service for all kind of mobile phone users with the help of a short code message service in which the users will be able to query about the ATM locations of the particular bank and get the ATM locations of the preferred bank via the SMS service [13].

1.3.2 Scopes

The objectives of the project are as follows:

- To retrieve the user's current geological coordinates.
- To locate the nearest ATM of a particular bank in a specified location based on geolocation in which the user's location is automatically generated with the help of GPS enabled device.

- To let the user see the locations on Map to find out how far he is from the expected location.
- To provide an interactive map to locate ATM and show the direction between user and ATM.
- To show alternative route from the source to the destination.
- To provide simple and easy user interface.
- To make the application handy for the users to locate the ATMs in unknown places.
- To provide list of nearest ATMs of particular bank through SMS targeting those users who use all kinds of mobile phones.

1.4 Project Features

The screens of the application are designed in XML and the business logic is written in Java. The database used is MySQL where all the information related to user's current location and geolocation and other information regarding each ATM is stored. The web service used to connect MySQL database with Android is PHP. The PHP files are located on the XAMPP Server. Google Maps API has been used to make it easy for the user to figure out the nearest ATM location of the required bank and to find the route to the particular ATM.

The application first retrieves the user's current location with the help of GPS. Once the location of the user is retrieved, he is provided with the options to select the required bank. The application then lists the ten nearest ATMs of the corresponding bank in ascending order with respect to the distance of the ATM from user's current location. Google Map API (Application Programming Interface) is then used to display the route of each ATM to the user. In addition, for the availability and simplification of the service for all kind of mobile phone users, a short code message service is provided in the application in which the users will be able to query the ATM locations of the particular bank and get the five nearest ATM locations of the preferred bank from the user's current location via the SMS service.

1.5 Requirement and Feasibility Analysis

1.5.1 Requirement Analysis

The main purpose of Requirement Analysis/Specification document is to describe the functional and nonfunctional requirement of the Project. All the requirements specified here are high priority and has been specified according to the requirement analysis. This document is intended to clarify the actual need of the system and verify its functionality with other member involved to design the system.

1.5.1.1 Functional Requirements

The functional requirement of the system is divided into three categories or module.

i. Retrieve current location

- The user shares his location by enabling GPS service.
- The application displays user's current location as toast message.
- The user's location is updated for particular period
- Google Map API key is generated
- Marker is placed on user's current location

ii. Display nearest ATMs

- Place API key for finding nearest places to the Google maps server
- The server sends result to the application in JSON format
- Ten nearest ATMs are listed by the application based on the user's current location
- The marker is placed for the particular location.

iii. Route to the ATM

- The user's and ATM's geolocations i.e. Latitude and Longitude are sent to Google map server.
- The Application gets distance from Google Map Server.
- The distance is displayed on the list of each ATMs

- The Application gets points between user's location and ATM's location from Google Map Server. These points are joined as polyline.
- The polyline is drawn on the map to show the route from the user's current location to the selected ATM's location.
- The application navigates the user to the selected ATM's location.

1.5.1.2 Non-functional Requirements

The non-functional requirement of the system is divided into three categories:

i. Performance Requirements

- The application provides the features of real world scenario.
- The system shall perform its function by retrieving, storing and processing data within few milliseconds.
- Performance depends upon the availability the internet.
- The system even displays error message if any error occurred during performing any task in system by the user.

ii. Security Requirements

- There is no any authentication login or registration to the application.
- The application user must provide his GPS location as well have access to the internet.
- The users can get the required information through SMS service as well.

iii. System Requirements

a. Hardware Requirements

"Khoj" is an android mobile application, the primary hardware requirement is as follows:

- GPS
- Processor: 1 GHz

- Memory Required: 256 MB
- Android mobile with OS version 2.3 or above

b. Software Requirements

- Operating System: Android
- Java JDK 1.6.0 or above
- Eclipse IDE with Latest Android SDK
- Database: MYSQL

1.5.2 Feasibility Analysis

The feasibility of the project is analysed in this phase and public proposal is put forth with a very general plan and some cost estimates. During system analysis the feasibility study of the proposed system is carried out. For feasibility analysis, some understanding of the major requirements for the system is essential. The key considerations involved in the feasibility analysis are:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

1.5.2.1 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

The existing system should be able perform as suggested and the proposed equipment's have the technical capacity to handle the data in the new system. The proposed system provide adequate response to inquiries, regardless of the size of the files and the system should be able to be upgraded if developed.

1.5.2.2 Operational Feasibility

The operational feasibility is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. There must be sufficient support for application resources.

The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system. The system should be able to be used and work properly if it is being developed and implemented.

1.5.2.3 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the administration can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available.

1.6 SOFTWARE DESCRIPTION

1.6.1 Android

Android is an Open Source Operating System for Mobile Devices. It’s one of the latest technology in computer field. It’ a Linux Based OS. It’s most popular OS for mobile devices. The Open Handset Alliance is a group of hardware and software developers, including Google, NTT DoCoMo, Sprint Nextel, and HTC, whose goal is to create a more open cell phone environment. The first product to be released under the alliance is the mobile device operating system, Android.

Android, is a Java-based operating system that runs on the Linux 2.6 kernel. It is very lightweight and full featured. Google is one member of the Open Handset Alliance.

Google, after purchasing the original developer of Android, released the Operating System under the Open Handset Alliance.

1.6.2 Java

Java is one of the programming languages used for develop android applications. Most of the Android Applications are developed in java. It is a powerful OOP language. Java was designed to meet all the real world requirements with its key features as follows:

- Simple and Powerful
- Secure
- Portable
- Multithreaded
- Architecture-neutral
- Distributed
- Interpreted and High performance
- Dynamic

1.6.3 XML

XML stands for Extended Mark-up Language. It is also a Markup Language Similar to HTML. Every Values in XML are between the TAGs. It is mostly used for Transfer Data between Networks. But in our project it’s only used for UI Design. In Android application with Eclipse IDE UI, External resources such as Strings, String Arrays, colors are in XML format. The default Drag and Drop UI Design automatically generate Xml code depends dragged elements or you can write xml code for UI. Manifest.xml is one of the Important Xml files in android application development. It contains all the permissions for the application, API Keys, Minimum and Target SDK Version of the Application.

1.6.4 JSON

JSON stands for JavaScript Object Notation. It is a text-based open standard designed for human-readable data interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for many languages. In JSON the values are not between the tags .All the values in JSON are Objects. It is Object based format. The JSON format was originally specified by Douglas Crockford, and is described in RFC 4627. The official Internet media type for JSON is application/json. The JSON filename extension is .json. The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML. Although JSON was originally based on a non-strict subset of the JavaScript Scripting language. Code for parsing and generating JSON data is readily available for a large variety of programming languages. One potential pitfall of the free-form nature of JSON comes from the ability to write numbers as either numeric literals or quoted strings.

1.6.5 API Keys

There are two API keys used in this project: Google Maps API Key (for use Google maps) and Places API Key (for find nearest ATMs). The following are the steps to get API Keys:

- Go to this link https://code.google.com/apis/console and sign in with the Gmail account
- Create new Project or rename the Default API Project. The Project name should be started with small letter and it should be the application name.
- Click services tab in the project Pane. There are list of services displayed. From these services, enable the following Google Maps Android API V2.
- For Places API, provide the organization name and website address.

- Click API Access tab in the project Tab pane. First click “create new Android key” give the Sha1 key and package name. Both are separated by “;” i.e. SHA1;packagename
- To Find the SHA1 key In Eclipse Window -> Preferences ->Android -> build. The SHA1 key displayed as “SHA1 Fingerprint: key”. Put the SHA1 key and package name and click “create”.
- Click “Create new server Key” to block the particular server to access your application. Give the IP address of the server and click “create”. Otherwise don’t put any IP address and click “create”. This is the API key for Places.

Steps to add Google Play Service Library:

- File -> Import -> Existing Android Code Into Workspace
- “Select the Google play service library project” and click Finish. Now the New Project is imported to the workspace.
- Project -> Properties -> Android in “Library Tab” click Add and select the Google Play Service Library Project and click Apply. Now Google Play Service Library is added to the Project.

CHAPTER 2: LITERATURE REVIEW

An Automated Teller Machine or ATM allows a bank customer to conduct their banking transactions from almost every other ATM machine in the world. As we went through the history of ATM, we found that as often the case with inventions, many inventors contribute to the history of an invention of ATM. [2]

In 1939, Luther Simjian patented an early and not-so-successful prototype of an ATM. He came up with the idea of creating a "hole-in-the-wall machine" that would allow customers to make financial transactions which then implemented in 20 patents in 1939. After six months, the bank reported that there was little demand for the new invention and discontinued its use. [4] After that different inventors came up with their concepts and applied at various places. However, some experts have the opinion that James Goodfellow of Scotland holds the earliest patent date of 1966 for a modern ATM, and John D White (also of Docutel) in the US is often credited with inventing the first free-standing ATM design. In 1967, John Shepherd-Barron invented and installed an ATM in a Barclays Bank in London. [5] Don Wetzel invented an American made ATM in 1968. However, it wasn't until the mid to late 1980s that ATMs became part of mainstream banking [6].

After the use of ATM in banking was accepted by people, the concept of mobile banking also came in use. Now, in this modern informatics life, technological advancement has completely revolutionized the world causing humans to integrate on them in their lifetime, especially on mobile phone. Moreover, the mobile gadgets are much sophisticated nowadays, that is, it looks much alike a computer with more compact and tinier but allows information to be stored on the mobile devices. [5]

Increasingly small, stylish, and powerful mobile phones are now as ubiquitous as they are indispensable. Hardware advancements have made mobiles smaller and more efficient while including an increasing number of peripherals. Beginning with cameras and media players, mobiles now include GPS systems, accelerometers, and touch screens. While these hardware innovations should prove fertile ground for software development, the applications available for mobile phones have generally lagged behind the hardware. [3] Corporate as well as Consumer products are increasingly geared towards mobility and Location Based Services. This is where our study focuses on the ability of mobile devices which provides software solutions which are location sensitive [1]. For the development of such mobile based application, some technologies are used like LBS, GPS etc.

Location based services (LBS) are information and entertainment services accessible with mobile devices through mobile network and utilizing the ability to make use of the geographical position of the mobile device. Mobile devices send and receive radio signals with any number of cell site base stations fitted with microwave antennas. These sites are usually mounted on a tower, pole or building, located throughout populated areas, then connected to a cabled communication network and switching system. [12]

The Global Positioning System (GPS) is a U.S. owned utility. It provides users with positioning, navigation, and timing (PNT) services. The Global Positioning System (GPS) is actually a constellation of 27 Earth-orbiting satellites (24 in operation and three extras in case one fails).The orbits are arranged so that at anytime, anywhere on Earth, there are at least four satellites "visible" in the sky. A GPS receiver's job is to locate four or more of these satellites, figure out the distance to each, and use this information to deduce its own location. [8]

GPS for mobile communications have been widely applied to collect detailed information on travel trajectory. Nowadays, people can use their cell phones to get direction, track their friends, searching the nearest amenities or keep an eye on their kids. According to Carter (2006), the U.S. FCC (Federal Communications Commission) has mandated the E911 (Enhance 911), which requires that the location of any mobile phone used to call 9-1-1 can be determined. This will require the manufacturers to install a GPS receiver in all mobile phones. As a result, people are able to use their mobile phone to get driving directions. [13]

Prior work on ATM locator has been done by ‘MoneyPass’. MoneyPass’ is a nationwide surcharge-free ATM network with more than 20,000 ATMs. MoneyPass ATM Locator provides the search options by entering an address, a zip code or using the current location of the application user. It includes the cellular IS-41 MAP (mobility application part) that consists of using a two-level hierarchy of location registers called home location registers and visitor location registers to track mobile locations using registration notification messages. Though it makes easier way to locate the ATM for android user, the limitation was that it was unable to help the non-android users. [14]

Likewise in the context of Nepal, nLocate is an application for modern smart phones and web browsers that lets to look up business information such as banks, ATMs, restaurants and many more. The location-based app also lets to discover interesting events and offers in your surroundings. nLocate started as an android application that tracks the user's location to provide a clear direction to the business users are looking for. [15]

Thus, focusing on the limitations of these applications we are working on the project named **"Khoj"** which is an android application that will find the nearest ATM center within the specified location. First of all, the application locates the ATM based on geolocation on the map of the user which is automatically generated with the help of a GPS enabled device. The user is then allowed to choose the preferred bank after which the nearest ATM booth of that particular bank is shown along with the direction of that ATM booth from the user's location. Also, this application will be able to show other alternative route from the source to the destination. Moreover, our project is able to facilitate the non-android users. For the availability and simplification of the service for all kind of mobile phone users, a short code message service will be provided in the application in which the users will be able to query the ATM locations of the particular bank and get the ATM locations of the preferred bank via the SMS service. The short code is a number to which a SMS or text message can be sent. A short code is fewer digits than a 10-digit telephone number. A short code may be specific to one mobile operator or common and supported by all major mobile operators.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Project Management Strategy and Tools

Project management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements [16]. A Project Management is a controlled process of initiating, planning, executing and closing down a project [17]. To ensure that the project meets the deadline, Is developed within an acceptable budget, and fulfill customer expectations and specifications, effective project management is necessary.

Following are the project management strategies:

3.1.1 Research

Research is the systematic study of the materials and the source. Some of the research methodologies done for the software development are:

- Research on the Internet
- Research about the ATM users in Nepal.
- Read various papers and articles for locating ATMs nearby.

3.1.2 Goal

Effective project knows which operational goals make a difference in the strategy and method for keeping these visible to all [16]. At the beginning of the project supervisor needs to tell the project team how the project fits into or support the periphery. Once the project is under way, progress against these goals needs to be measured and communicated on an ongoing basis [17].

3.1.3 Team Structure

Team structure is defining the specific duties and work for the project and developing the software.

Team: 4 members and 1 supervisor

Role	Team
Supervisor	Mr. Vishnu Kumar Rana
Designer/Developer	Sajan Rai
Designer/Developer	Rashna Neupane

Table 1: Team Structure

3.1.4 Issue Resolution

Projects are usually initiated to resolve an issue. The best project know how to assign an ownership of an issue to a project team, to analyze smartly on problem-solving, decision making, planning method and then move to resolution.

Responsibilities of Supervisor:

- Schedule the project
- Share information

Responsibilities of Team Members:

- Research about the project

- Analyze and design
- Development and testing
- Implementation
- Report Submission

3.2 System Analysis

System analysis is a detailed study of various operations performed by a system. It is a process of collecting and analyzing facts with respect to the existing system operation of the situation prevailing so that an effective computerized system may be designed and implemented if proved feasible.

That is a structural process related to four significant phases. They are study phase, design phase, development phase and implementation phase. A good analysis is essential for the development of a new improved system.

3.2.1 Existing System

Previously the mapping was based on the GIS (Geographical Information System) Technology. This is very Complex Technology. Cartography is the art and science of graphically representing a geographical area, usually on a flat surface such as a map or chart; it may involve the superimposition of political, cultural, or other non-geographical divisions onto the representation of a geographical area. DBMS was used for storing map data.

3.2.2 Proposed System

Nowadays GPS is mostly used for Location Based Services. GPS (Global Positioning System) is satellite based. A network of satellites continuously transmit coded information, which makes it possible to precisely identify locations on earth by measuring distance from the satellites. These satellites are developed by U.S Defense.

Every point in earth get signal from at least 4 satellites. The main advantage of this System is accuracy. GPS has 3 Segments these are Space Segment, Control Segment and User Segment. The user Segment has GPS receiver. The proposed system uses GPS service as well as the advanced technologies like Google Maps API service to locate and route the direction of certain places.

3.3 System Design

3.3.1 Module Description

3.3.1.1 Display current location

This module display's the user's current location as a toast message when the user shares his location to the application. The user's location is retrieved by either GPS or Network Provider. The user's location is then updated for particular time and distance period. For Using Google Map to this Application Google Maps API key is generated, and the marker is placed in the user's current location.

3.3.1.2 Display nearest ATMs

In This Module, ten nearest ATMs are listed by the application based on the user's current location. This Application sends user's current location's Latitude and Longitude, radius of the area to be searched, bank code to search for particular bank on the basis of selection made by the user, and Place API key for finding nearest places to the Google maps server. The server sends result to the application in JSON format. This format is parsed to String for use into Application. This data contains Name, Address and Latitude and Longitude point of the ATMs. From these latitude and longitude points of all the nearest ATM's, the marker is placed for the particular location.

3.3.1.3 Route to the ATM

In this module, the application navigates the user to the particular location of the ATM which the user selected. First of all, the user's and ATM's geolocations i.e. Latitude and Longitude are sent to Google map server. The Application gets points between user's location and ATM's location from Google Map Server. These points are joined as polyline, and the polyline is drawn on the map to show the route from the user's current location to the selected ATM's location. This Application gets distance from Google Map Server. The distance is displayed on the list of each ATMs. It also updated for particular time and distance period.

3.3.2 Algorithm

The Algorithm for **"Khoj"** is as follows:

Step 1: Start of the application.

Step 2: Go to step 3 to get the nearest ATM location of preferred bank through the application or go to Step 8 to get the list of nearest ATM locations through SMS.

Step 3: Share the current location through GPS. If GPS is enabled, go to Step 5 else go to Step 4.

Step 4: Click the button provided in the application in order to navigate to the Location access and Network settings of the phone and enable the GPS and network.

Step 5: Choose the preferred bank from the list provided.

Step 6: Choose the preferred nearest ATM location of the corresponding bank from the list of the ATM locations obtained in ascending order with respect to the user's current location.

Step 7: Display the route from the user's current location to the preferred ATM location in Google Map.

Step 8: Compose the message (For eg. APIKHOJ GBL LAG to obtain the nearest Global IME Bank Limited from Lagankhel) and send it to 5455.

Step 9: The nearest five ATMs of corresponding bank from the corresponding place is listed in a message inbox.

Step 10: End of the application

3.3.3 Flowchart

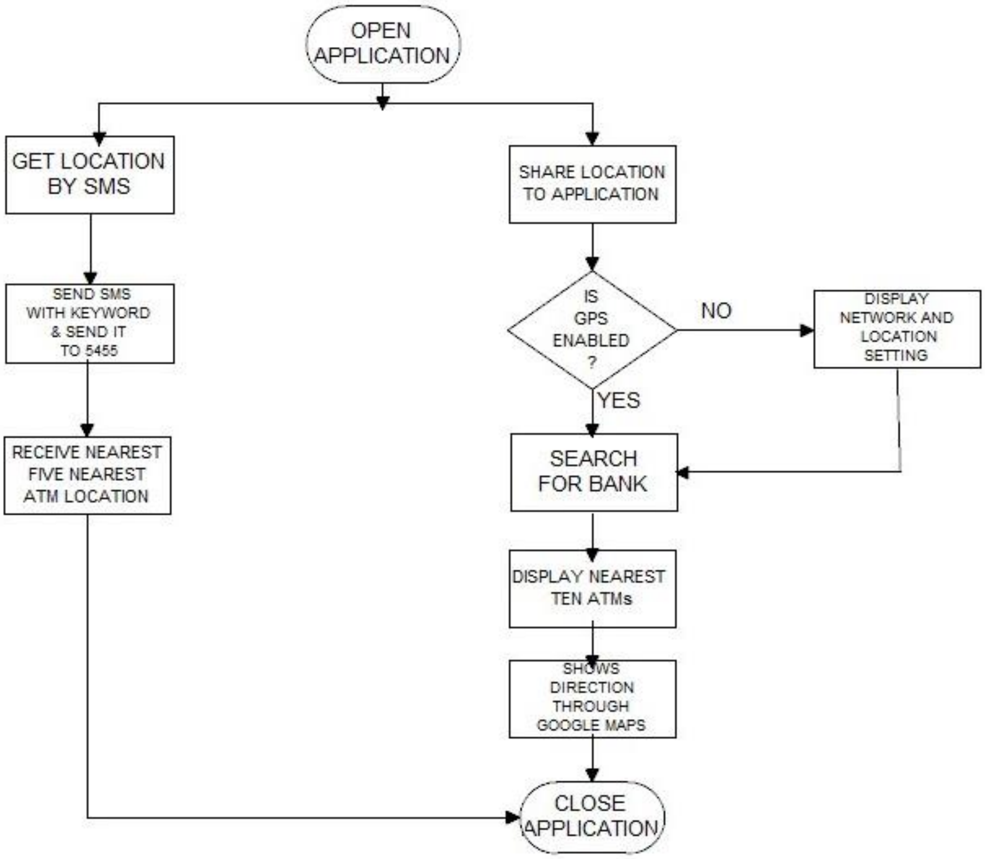

OPEN APPLICATION
GET LOCATION BY SMS
SHARE LOCATION TO APPLICATION
SEND SMS WITH KEYWORD & SEND IT TO 5455
IS GPS ENABLED ?
NO
DISPLAY NETWORK AND LOCATION SETTING
YES
RECEIVE NEAREST FIVE NEAREST ATM LOCATION
SEARCH FOR BANK
DISPLAY NEAREST TEN ATMs
SHOWS DIRECTION THROUGH GOOGLE MAPS
CLOSE APPLICATION

Figure 1 Flowchart showing the workflow of the project

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem.

Figure 1 represents the flowchart of the project. First of all, the user needs to share his location through GPS. But if GPS is not enabled, the user is navigated to the Location Access and Network settings of the phone. After the location is shared, the appropriate bank is to be clicked and then the nearest ATM locations of the corresponding bank is shown with nearest ten ATMs in ascending order with respect to the user's current location.

3.3.4 Context Diagram

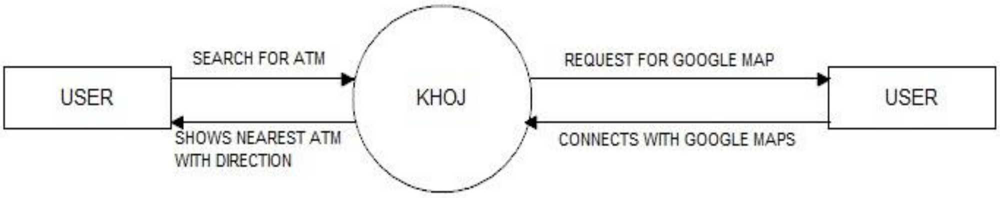


Figure 2 Context Diagram

A System Context Diagram (SCD) in software engineering and systems engineering is a diagram that defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it. This diagram is a high level view of a system. It is similar to a block diagram.

Figure 2 represents the Context Diagram of the project. According to the figure above, the user requests for the ATM location of the particular bank to the application "Khoj". The application then requests for the google map with to the Google Maps API. Then, the Google API connects the application with Google Map which further displays the nearest route from user's current location to ATM location to the user.

3.3.5 DFD up to Level 2 for major Processes

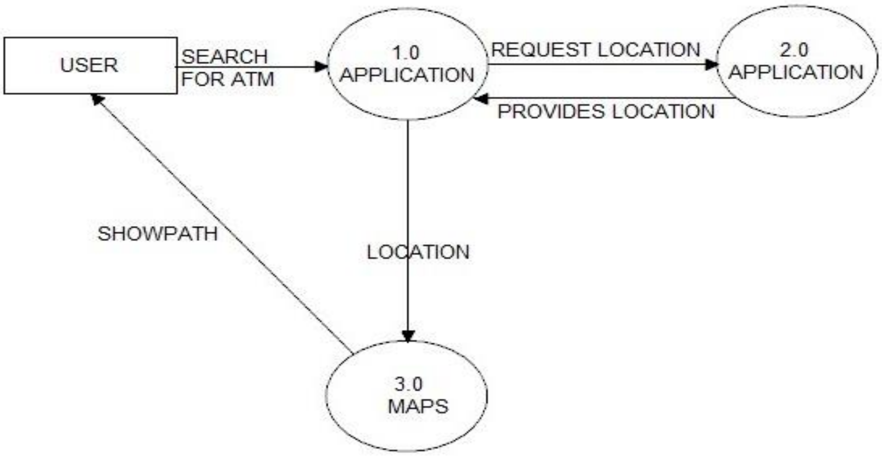


Figure 3 DFD Level 1

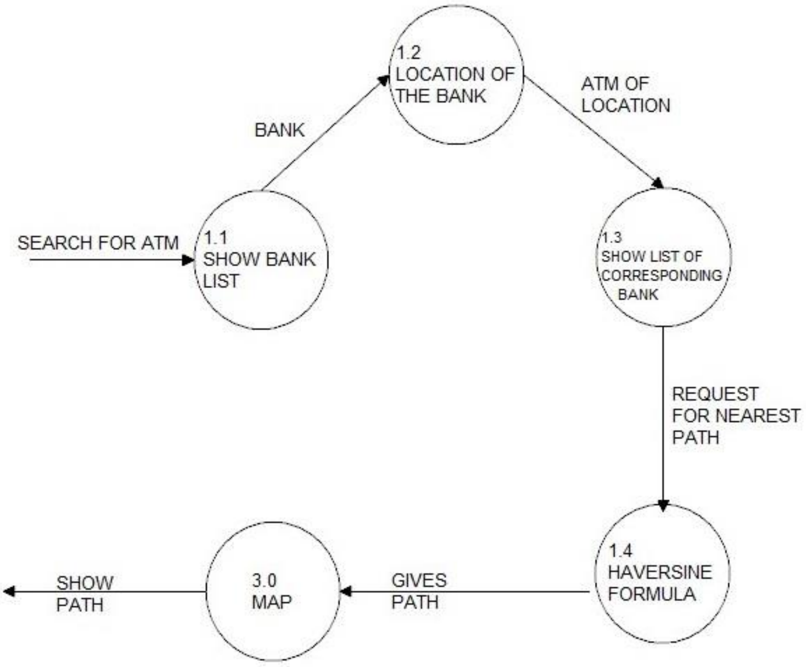


Figure 4 DFD Level 2 Expanding Process1

Figure 3 represents the DFD Level 1 of the project whereas Figure 4 represents the DFD Level 2 expanding Process 1. A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

3.3.6 Use Case Diagram

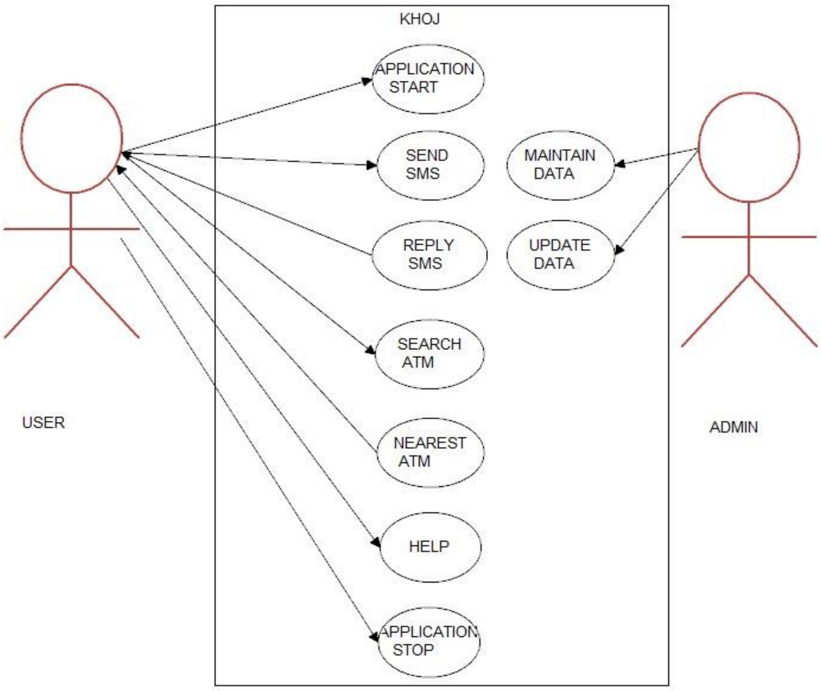


Figure 5 Use Case Diagram for Khoj

Figure 5 represents the Use Case Diagram of the project . A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

3.3.7 Class Diagram

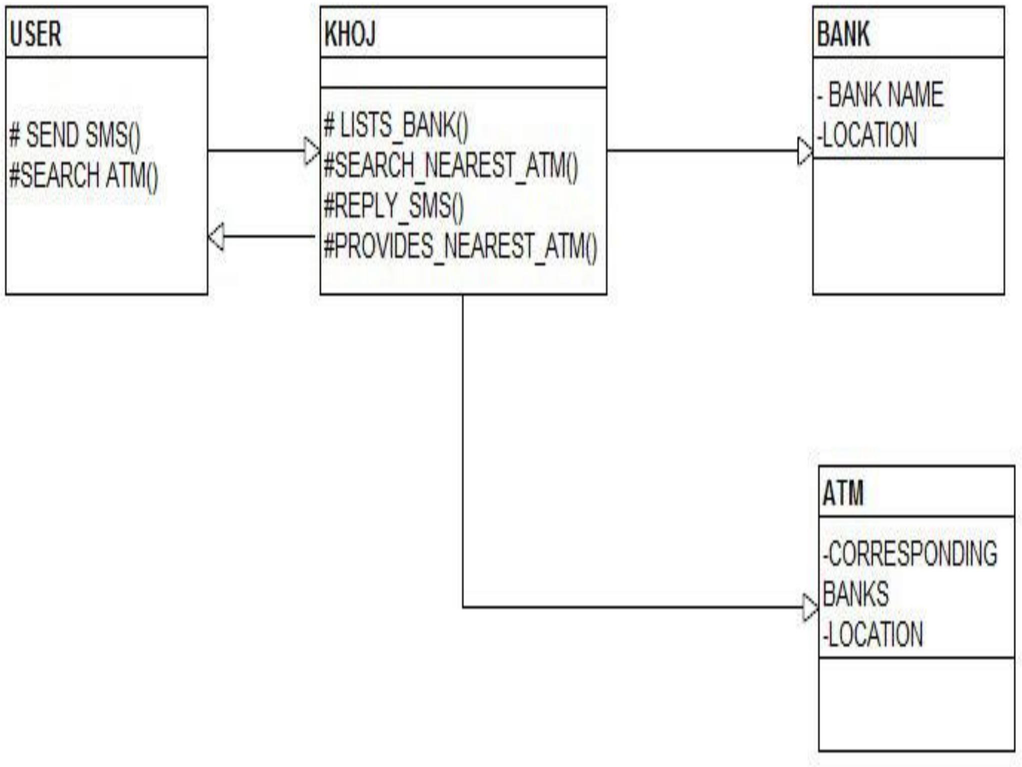


Figure 6 Class Diagram for Khoj

Figure 6 represents the Class Diagram of the project. The class diagram is the main building block of object oriented modelling. It is used both for general conceptual modelling of the systematics of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling.

3.3.8 System Sequence Diagram

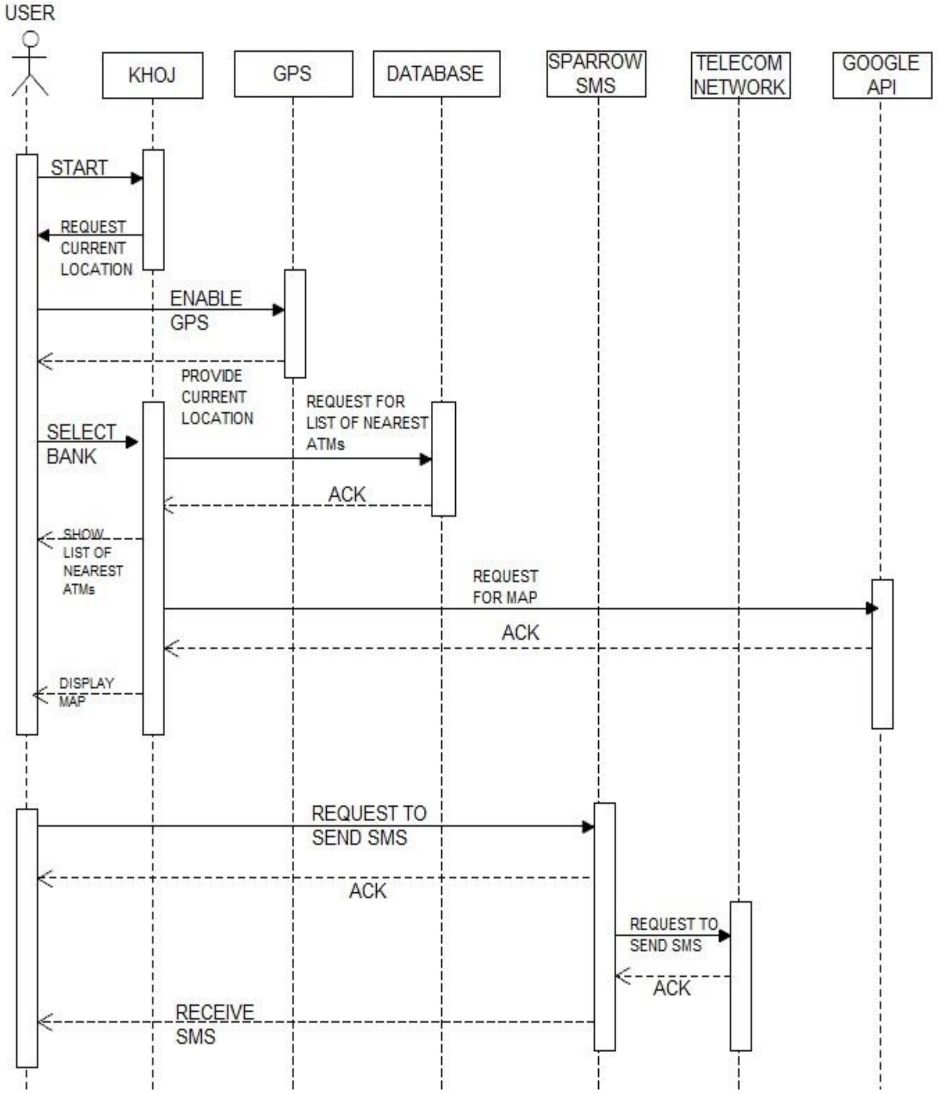


Figure 7 Sequence diagram for Khoj

Figure 7 represents the Sequence Diagram of the project. A system sequence diagram (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate, their order, and possible inter-system events. Overview. System sequence diagrams are visual summaries of the individual use cases.

3.4 Project Schedule

3.4.1 GANTT Chart

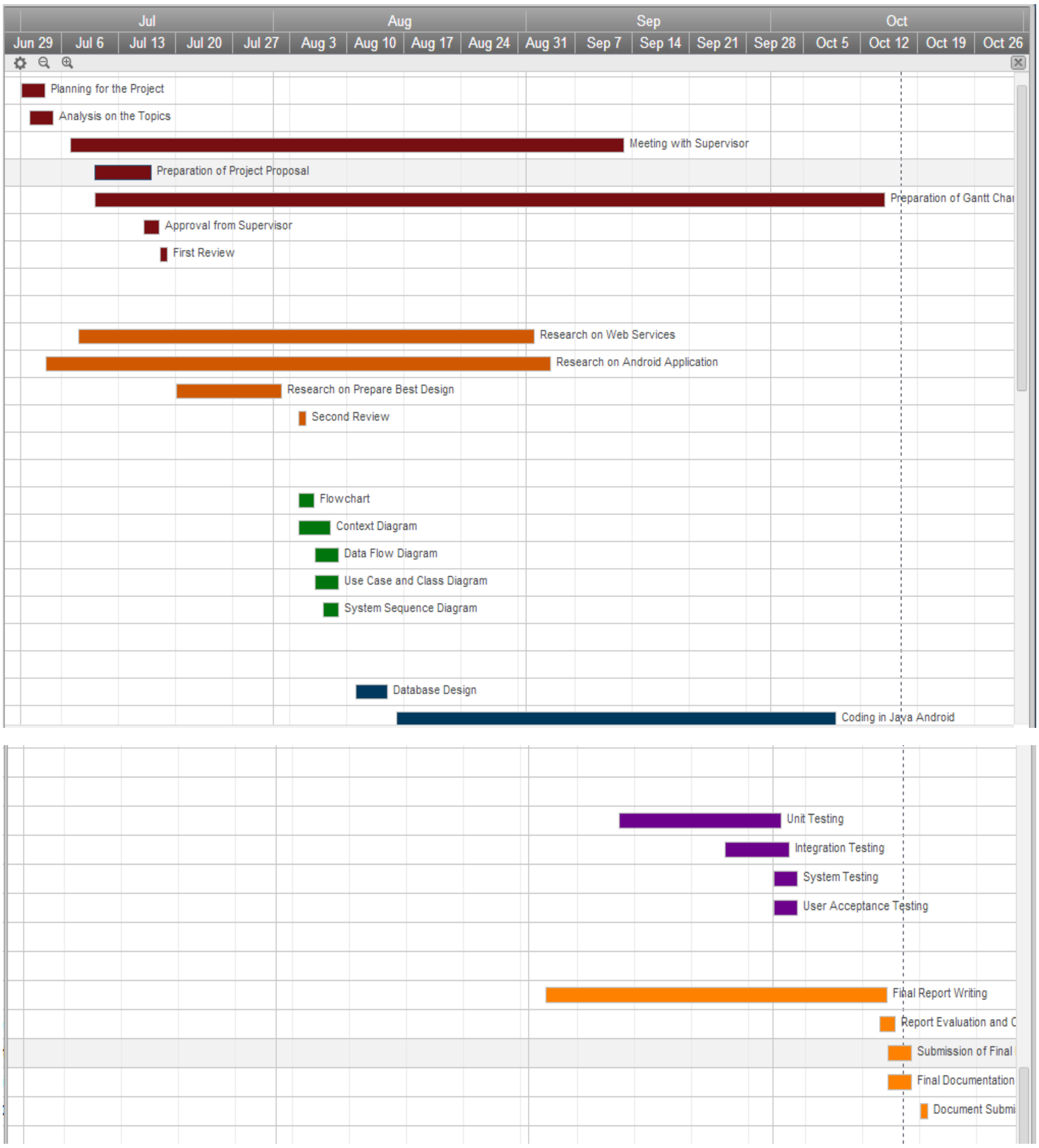


Figure 8 Gantt chart

3.4.2 Time Schedule

Time Schedule			
		Planned	
Task ID	**Task Description**	**Start Date**	**Finished Date**
1	Preliminary Work		
1.1	Planning for the Project	7/1/2014	7/3/2014
1.2	Analysis on the Topics	7/2/2014	7/5/2014
1.3	Meeting with Supervisor	7/7/2014	9/12/2014
1.4	Preparation of Project Proposal	7/10/2014	7/16/2014
1.5	Preparation of Gantt Chart and Project Schedule	7/10/2014	10/14/2014
1.6	Approval from Supervisor	7/16/2014	7/17/2014
1.7	First Review	7/18/2014	7/18/2014
2	Research Work		
2.1	Research on Web Services	7/8/2014	9/1/2014
2.2	Research on Android Application	7/4/2014	9/3/2014
2.3	Research on Prepare Best Design	7/20/2014	8/1/2014
2.4	Second Review	8/4/2014	8/4/2014
3	Design		
3.1	Flowchart	8/4/2014	8/5/2014
3.2	Context Diagram	8/4/2014	8/7/2014
3.3	Data Flow Diagram	8/6/2014	8/9/2014
3.4	Use Case and Class Diagram	8/6/2014	8/9/2014
3.6	System Sequence Diagram	8/7/2014	8/9/2014
4	Implementation		
4.1	Database Design	8/11/2014	8/14/2014
4.3	Coding in Java Android	8/16/2014	10/8/2014
5	Testing		
5.1	Unit Testing	9/12/2014	10/1/2014
5.2	Integration Testing	9/25/2014	10/2/2014
5.3	System Testing	10/1/2014	10/4/2014
5.4	User Acceptance Testing	10/1/2014	10/4/2014
6	Dissertation		
6.1	Final Report Writing	9/3/2014	10/14/2014
6.2	Report Evaluation and Conclusion	10/14/2014	10/15/2014
6.3	Submission of Final Report Copy	10/15/2014	10/17/2014
6.4	Final Documentation Printing and Binding	10/15/2014	10/17/2014
6.5	Document Submission to College	10/19/2014	10/19/2014

Table 2: Time Schedule of Khoj

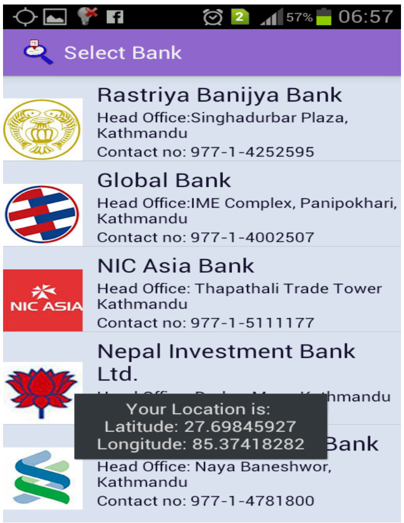
3.5 System Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The increasing visibility of software as a system element and the cos associated with a software failure are motivating force for well planned, through testing. Software is tested from different perspectives. The following are test cases with the help of which the application has been tested:

3.5.1 Unit Testing

In unit testing, various modules have been tested individually. This has been done manually to test if the expected result is actually seen on the screen. We independently unit tested and validated the supporting classes in com.adifynepal.khoj package, which are designed to provide re-usable functionality used across different features such as:

i. Figure 9 represents the screenshot for retrieving the user’s current location with the help of GPS service as unit testing.



Rastriya Banijya Bank
Head Office:Singhadurbar Plaza, Kathmandu
Contact no: 977-1-4252595

Global Bank
Head Office:IME Complex, Panipokhari, Kathmandu
Contact no: 977-1-4002507

NIC Asia Bank
Head Office: Thapathali Trade Tower Kathmandu
Contact no: 977-1-5111177

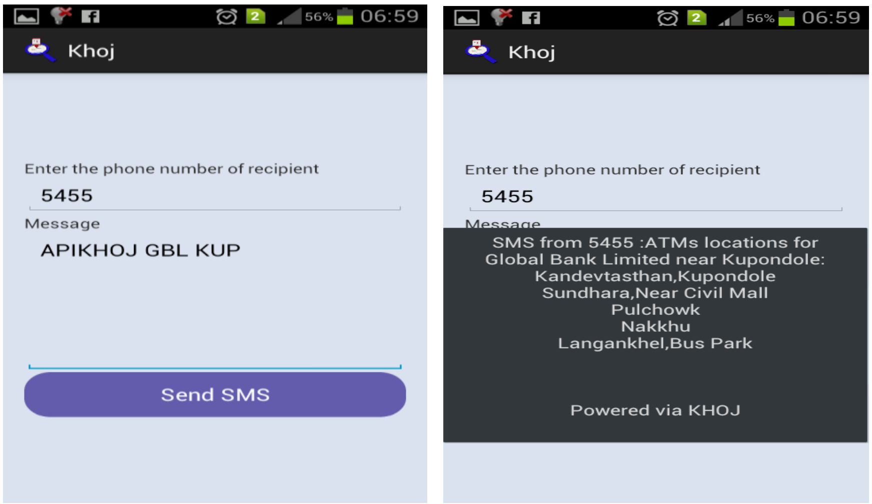
Nepal Investment Bank Ltd.
hmandu

Your Location is:
Latitude: 27.69845927
Longitude: 85.37418282

Bank
Head Office: Naya Baneshwor, Kathmandu
Contact no: 977-1-4781800

Figure 9 Unit Testing for retrieving user's location and listing Bank names

ii. Figure 10 represents the unit testing performed for the SMS based service.



Enter the phone number of recipient
5455
Message
APIKHOJ GBL KUP

Enter the phone number of recipient
5455
Message

SMS from 5455 :ATMs locations for
Global Bank Limited near Kupondole:
Kandevtasthan,Kupondole
Sundhara,Near Civil Mall
Pulchowk
Nakkhu
Langankhel,Bus Park

Powered via KHOJ

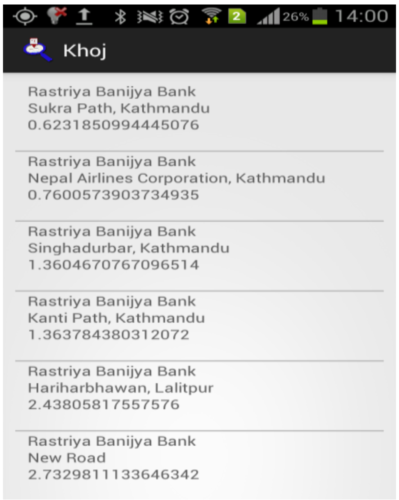
Figure 10 Unit testing for SMS service

Each classes were tested as a standalone Java console application using Unit test cases.

3.5.2 Integration Testing:

After the unit testing, these units were integrated together to perform a complete task and were tested again. This testing was mainly emphasized to check whether the integrated part worked well or not. Separate development units that make up a component of the system are tested to ensure that they work together [20]. Software and hardware are integrated and full range of system test is conducted so that we check for interrupt priority and the performance of each interrupt handling procedures. Following are the results for the integration testing performed:

i. Figure 11 represents parsing the JSON results and calculating surface distance from user’s location to the location of each ATM and listing the nearest ATM location when particular bank is clicked.



Rastriya Banijya Bank
Sukra Path, Kathmandu
0.6231850994445076

Rastriya Banijya Bank
Nepal Airlines Corporation, Kathmandu
0.7600573903734935

Rastriya Banijya Bank
Singhadurbar, Kathmandu
1.3604670767096514

Rastriya Banijya Bank
Kanti Path, Kathmandu
1.363784380312072

Rastriya Banijya Bank
Hariharbhawan, Lalitpur
2.43805817557576

Rastriya Banijya Bank
New Road
2.7329811133646342

Figure 11 Integration testing for List of nearest ATM locations from user's current location

ii. Figure 12 represents displaying the route in Google Map using Google Map API as an integration testing:

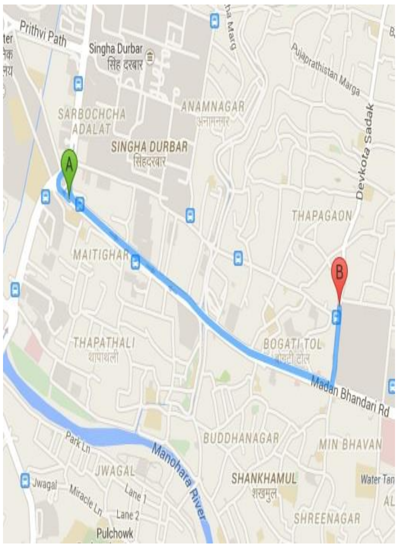


Figure 12 Integration testing for Google map showing route from user's location to ATM

3.5.3 System Testing

"Khoj" was system tested on an Android powered mobile phone with Android Icecream Sandwich OS (Version 4.0.4). Using the app on an actual handset in real life scenarios revealed some interesting insights which would have been otherwise difficult to detect in a simulated environment. We used Eclipse IDE to develop the application which would not support in displaying the Google map in simulated environment. The actual usage of the application was able to display the Google Map along with the required result. In addition, some of the performance enhancements were discovered during field testing. Moreover, the usage of a real mobile phone helped to obtain the result for the SMS service provided where the SMS service is available for both the android and non-android users. The screenshot of final result of system testing is displayed as follows in Figure 13.

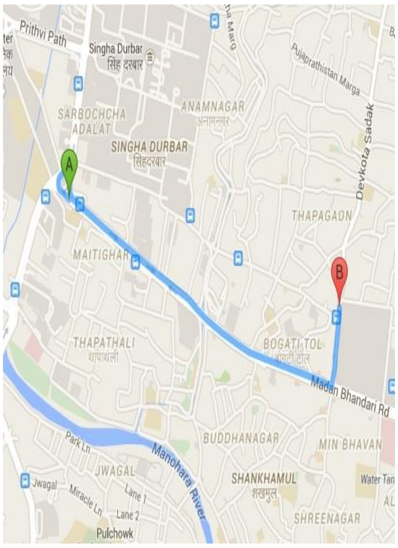


Figure 13 System testing for Google map showing route from user to ATM

3.5.4 User Acceptance Testing

User acceptance test is a test conducted to determine if the requirements of a specification are met. The application was mainly designed for android phones as it helps the users find the direction to the nearest ATMs required when they are on the move. Generally they try to carry something handy like cell phones. Different android phones have different screen sizes and resolution. The application has been tested for its compatibility with different screen sizes on the emulator and the requirement of the specification was met.

3.6 System Implementation

3.6.1 System development environment

The screens were designed in XML and the business logic was written in Java. The database used is MySQL where all the information related to user's current location and geolocation and other information regarding each ATM is stored. The web service used to connect MySQL database with Android is PHP. The PHP files are located on the XAMPP Server. Google Maps API has been used to make it easy for the user to figure out the nearest ATM location of the required bank and to find the route to the particular ATM.

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

3.6.2 System Development tools:

For developing the application, we need the following applications pre-installed:

- Eclipse IDE
- Android SDK

Apart from these, we also require the following for using Google map services in the application.

- Google APIs Add-on
- Android Virtual device Targeting Google APIs.

CHAPTER 4: RESULT ANALYSIS

4.1 Result

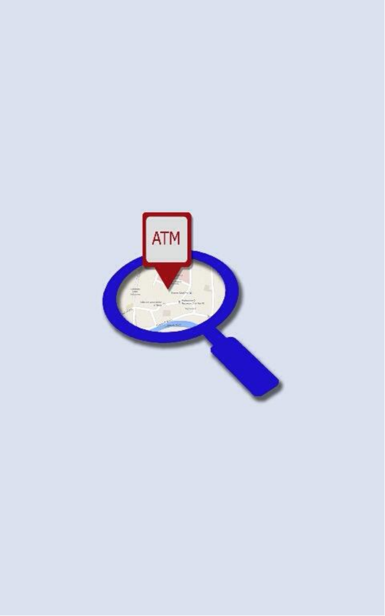


Figure 14 Splash Screen of the application

Figure 14 represents the first activity of the application which displays the splash screen consisting of the application's logo.

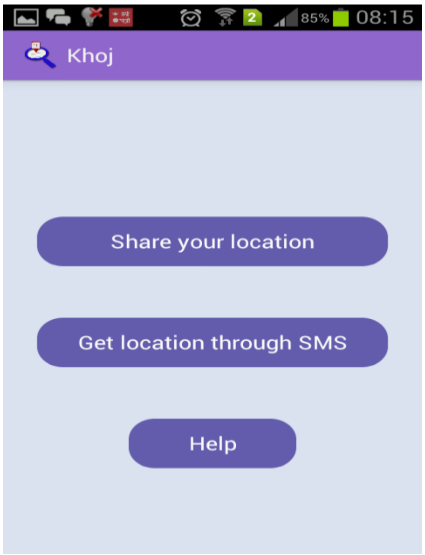
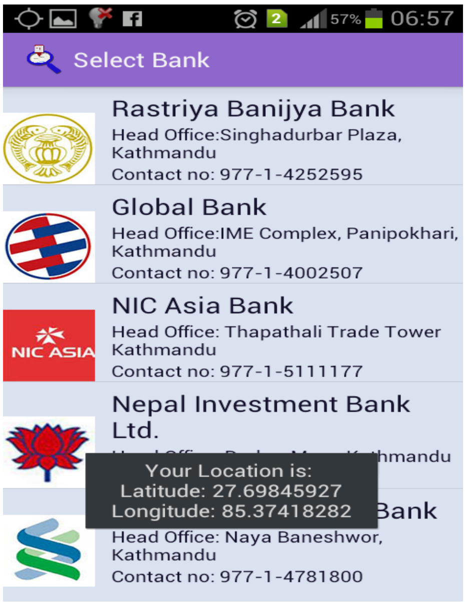


Figure 15 Screenshot showing ways to extract the nearest ATM location

Figure 15 represents the second activity in the application whose user interface consists of the ways to extract the nearest ATM locations. The nearest ATM locations of the preferred bank can be extracted by enabling GPS service of the device or through SMS service.



Select Bank

Rastriya Banijya Bank
Head Office:Singhadurbar Plaza, Kathmandu
Contact no: 977-1-4252595

Global Bank
Head Office:IME Complex, Panipokhari, Kathmandu
Contact no: 977-1-4002507

NIC Asia Bank
Head Office: Thapathali Trade Tower Kathmandu
Contact no: 977-1-5111177

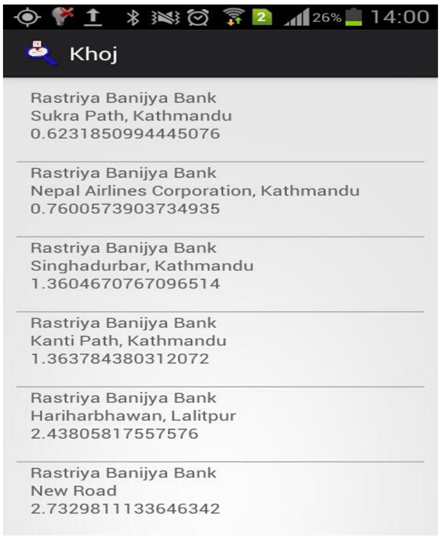
Nepal Investment Bank Ltd.
hmandu

Your Location is:
Latitude: 27.69845927
Longitude: 85.37418282

Bank
Head Office: Naya Baneshwor, Kathmandu
Contact no: 977-1-4781800

Figure 16 Selection of the respective Bank

Figure 16 above shows the lists of the banks in which the users are allowed to select the preferred bank. Also it displays the geo-position of the user's current location.



Khoj

Rastriya Banijya Bank
Sukra Path, Kathmandu
0.6231850994445076

Rastriya Banijya Bank
Nepal Airlines Corporation, Kathmandu
0.7600573903734935

Rastriya Banijya Bank
Singhadurbar, Kathmandu
1.3604670767096514

Rastriya Banijya Bank
Kanti Path, Kathmandu
1.363784380312072

Rastriya Banijya Bank
Hariharbhawan, Lalitpur
2.43805817557576

Rastriya Banijya Bank
New Road
2.7329811133646342

Figure 17 List of Nearest locations of the corresponding Bank

Figure 17 in the above activity shows the list of the nearest ATMs of the selected bank along with the distance in ascending order with respect to the user's current location.

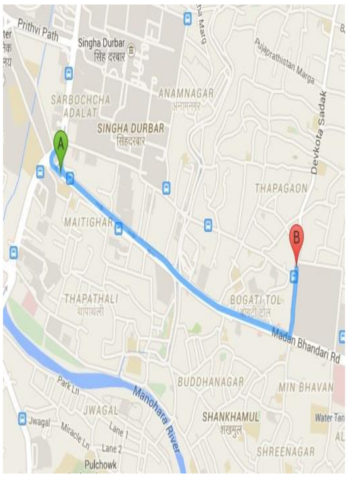
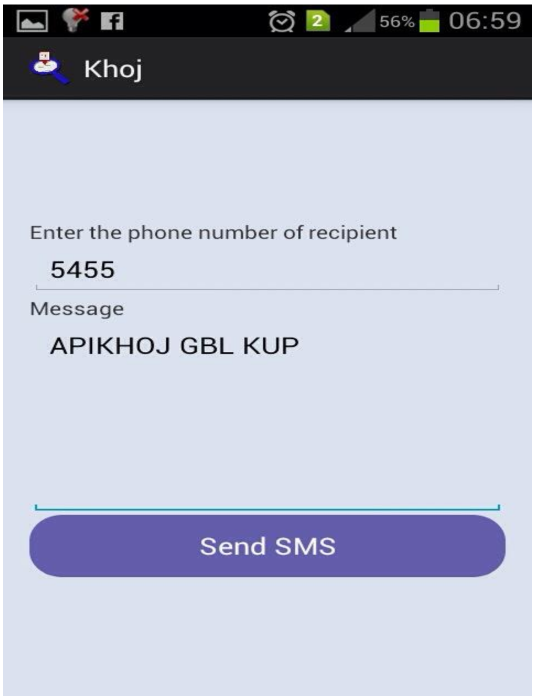


Figure 18 Google map showing route from user to nearest preferred ATM location

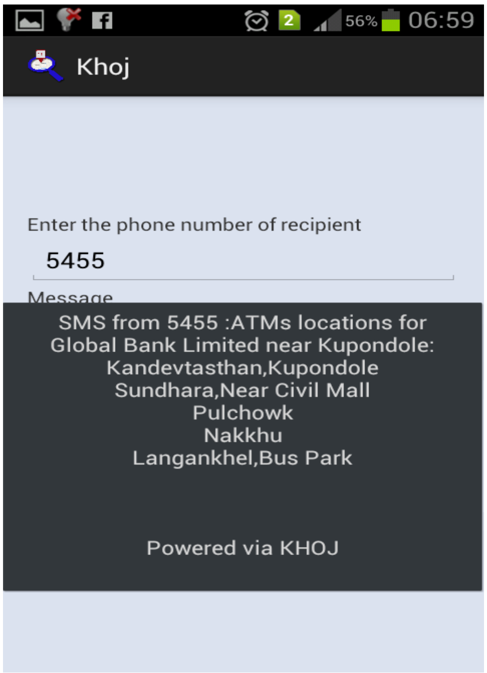
Figure 18 shows the google map which displays the route from the user's location to the selected ATM location.



Enter the phone number of recipient
5455
Message
APIKHOJ GBL KUP

Figure 19 SMS GUI Interface for querying nearest ATMs of corresponding bank

The above Figure 19 shows the user interface for querying the nearest ATM locations of the preferred bank through SMS.



Enter the phone number of recipient

5455

Message

SMS from 5455 :ATMs locations for Global Bank Limited near Kupondole:
Kandevtasthan,Kupondole
Sundhara,Near Civil Mall
Pulchowk
Nakkhu
Langankhel,Bus Park

Powered via KHOJ

Figure 20 List of nearest ATMs from the user's current location

The above figure 20 shows the list of five nearest ATM locations of the preferred bank which is obtained through SMS when the user queries for a particular bank with the specification of certain location.

4.2 Critical Analysis

This project **"Khoj"** has been developed for finding nearest ATMs which navigates to the particular ATM as per the selection of the user. It also displays that ATM's name, address and distance. These details have been collected from Google maps Server. The application was chosen to be developed in Android as it is parallel to iOS (supported by Apple) in terms of facilities it provide and is also open source. The screens were designed in XML and the business logic was written in Java. The database used is MySQL where all the information related to user's current location and geolocation and other information regarding each ATM is stored. The web service used to connect MySQL database with Android is PHP. The PHP files are located on the XAMPP

Server. Google Maps API has been used to make it easy for the user to figure out the nearest ATM location of the required bank and to find the route to the particular ATM.

During the research analysis phase, the study was performed in the company SCT (Smart Choice Technology) that looks after the entire ATM's of the banks under their network. Initially, the objective of the project also included providing the information of the ATM's which were under maintenance or out of service. As per the suggestion given by the SCT, it was not convenient to include the information due to security and authorization reasons. So, the requirement was modified accordingly.

This Application first finds user's current location by using Network Provider or GPS Sensor on user's mobile. Once the location of the user is retrieved, he is provided with the options to select the required bank. The application then lists the ten nearest ATMs of the corresponding bank in ascending order with respect to the distance of the ATM from user's current location. Google Map API (Application Programming Interface) is then used to display the route of each ATM to the user.

Here, for finding location of ATM, the GPS technology with Google Map API was used. As android is an open source, this application can be used for further improvements in many Smartphones. Though this project achieved all of the mentioned goals and objectives in some form, there remain some deficiencies. One of the main deficiency is that the project would be based on the internet connection i.e. ATM locator needs data connection to fetch Banks/ATMs data and maps. The overall performance of the app would depend on the speed of the data services. It would get much faster results on Wi-Fi networks, while GPRS and Edge would very much vary in the service provider. One of the main problems is the lack of spread of the wireless network into the countryside. In developing country like Nepal, the wireless technology is in very nascent stage. In major cities and areas, the problem of network congestion is also an important issue. The percentage of service operators not meeting the congestion rate benchmarks has risen subsequently. But still, we have somehow tried to solve this problem by providing a short code message service for the availability and simplification of the service for all kind of mobile phone users, in which the users will be able to query the ATM locations of the particular bank and get the five nearest ATM locations of the preferred bank from the user's current location via SMS service.

Moreover, in addition to location of the prior ATM, not only ATM, different places and important locations like Petrol Pump, Hospital, Ambulance, Cinema halls etc. can be added as another contextual element to the locator. So, that the user can get the more benefit from the application and use the application to the fullest.

4.3 Limitations and Future Enhancements

The major limitations of the project would be the internet-based dependencies i.e. ATM locator needs data connection to fetch Banks/ATMs data and maps. The overall performance of the app would depend on the speed of the data services. It would get much faster results on Wi-Fi networks. While GPRS and Edge performance would vary depending on the service provider. Bandwidth consumed to fetch Banks/ATMs info would take only few KBs but using maps would significantly utilize more data. The application can be improved in many ways and can be extended to support more devices like the tablets and iOS devices.

The application can be improved and extended in several ways. Here we explore some of these exciting possibilities and discuss our future plans for adding new features that build upon and compliment the current functionality. In addition to location of the prior ATM, not only ATM, different places and important locations like Petrol Pump, Hospital, Ambulance, Cinema halls etc. can be added as another contextual element to the locator. So, that the user can get the more benefit from the application and use the application to the fullest.

Finally, we would like to create a server side application to provide services such as allowing users to track friends' location in real time, provide personalized place recommendations based on the user's past history of visited and allow users to review or rate places they visit and share their experiences with other users of the application. This would entail communicating with server side protocols to send and retrieve location specific information.

4.4 Conclusion

In this project an Android Application was developed for finding nearest ATMs which navigates to the particular ATM as per the selection of the user. The application was chosen to be developed in Android as it is parallel to iOS (supported by Apple) in terms of facilities it provide and is also open source. The screens were designed in XML and the business logic was written in Java. The database used is MySQL where all the information related to user's current location and geolocation and other information regarding each ATM is stored. The web service used to connect MySQL database with Android is PHP. The PHP files are located on the XAMPP Server. Google Maps API has been used to make it easy for the user to figure out the nearest ATM location of the required bank and to find the route to the particular ATM. There are Two API Keys used in this Android Application. One for displaying Google map in Application (Google Maps API key for android) and another one for finding nearest ATMs (Places API Key) which are unique.

First of all, the application retrieves user's current location by using Network Provider or GPS Sensor on user's mobile. Once the location of the user is retrieved, he is provided with the options to select the required bank. The application then lists the ten nearest ATMs of the corresponding bank in ascending order with respect to the distance of the ATM from user's current location. Google Map API is then used to display the route of each ATM to the user.

In addition, for the availability and simplification of the service for all kind of mobile phone users, a short code message service is provided in the application in which the users will be able to query the ATM locations of the particular bank and get the five nearest ATM locations of the preferred bank from the user's current location via SMS service.

REFERENCES

[1] “Location based profiling” internet: http://www.beprojectreport.com/download/B3-LOCATION-BASED-PROFILING-USING-GPS [July 7, 2014]

[2] “Top 10 things about ATM” internet: http://content.time.com/time/specials/packages/article/0,28804,1914560_1914558_1914559,00.html [July 10, 2014]

[3] “CSCA-common short code administration” internet: http://www.usshortcodes.com/about-sms-short-codes/sms-marketing faqs.php#.U8UMEvmSwwA [July 9, 2014]

[4] “Automatic Teller Machines – ATM-The ATM Machine of Luther Simjian By Mary Bellis” internet: http://inventors.about.com/od/astartinventions/a/atm.htm [July 10, 2014]

[5] “Automated teller machines (ATMs) (per 100,000 adults) “internet: http://data.worldbank.org/indicator/FB.ATM.TOTL.P5[July 10, 2014]

[6] “Useful Information on ATM Machines ByTom Streissguth, eHow Contributor” internet:http://www.ehow.com/about_6546686_useful-information-atm-machines.html [July 7, 2014]

[7] “IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012ISSN (Online): 1694-0814 www.IJCSI.org” internet: http://ijcsi.org/papers/IJCSI-9-1-2-237-242.pdf [July 8,2014]

[8] “A thesis on ATM locator mobile application” internet: http://umpir.ump.edu.my/4444/1/CD6502_LIM_YEN_LENG.pdf [July 9, 2014]

[9] “GOOGLE PLACE FINDER” internet: http://www.astiwz.com/project-list [July 12, 2014]

[10] “International Journal of Advances in Engineering & Technology, Mar 2011.” Internet: http://www.ijaet.org/media/0001/Location-Based-Services-using-Android-mobile-Operating-System-Copyright-IJAET.pdf [July 12, 2014]

[11] “Android and iPhone Application for ATM/Bank Locater” internet: http://www.ndottech.com/solutions/mobile/android-and-iphone-application-for-atmbank-locater [July 9, 2014]

[12] Professional Android™ Application Development Author: Reto Meier [July 12, 2014]

[13] GPS Signal Acquisition and Tracking– An Approach towards Development of Software based GPS Receiver By Dinesh Manandhar, Yongcheol Suh, Ryosuke Shibasaki [July 12, 2014]

[14]"MoneyPass ATM Locator" internet: http://www.moneypass.com/atm-locator.aspx [July 1, 2014]

[15] "nLocate" internet : http://www.nlocate.com/about-us/ [Sept 3,2014]

[16] NiladriMallickChoudhuri, "Project Management Fundamentals", in ICT Infotech India Ltd., pp. 3. [September 5, 2014]

[17] Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich, "Modern System Analysis and Design", in Prentice Hall, 4th Edition, 2005, Chap. 3 pp.7. [September 11, 2014]

[18] G. Booch, J. Rumbaugh, I. Jacobson, "The Unified Modeling Language User Guide", in Addison-Wesley, Reading, Massachusetts, 1999, pp. 99. [September 12, 2014]

[19] John E. Bentley, "Software Testing Fundamentals-Concepts, Roles, and Terminology", in SUGI 30, Paper 141-30, 2004, pp. 1-5. [September 20, 2014]

[20] M. C. Bob Hughes, Software Project Management, 2005. [October 12, 2014]

APPENDIX

Coding Snippets:

//Splash.java : The code below generates the splash screen for the application.

```
public class Splash extends Activity {
        private boolean mIsBackButtonPressed;
        private static final int SPLASH_DURATION = 4000; //4 seconds
        private Handler myhandler;
       @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_splash);
            myhandler = new Handler();
          // run a thread to start the home screen
            myhandler.postDelayed(new Runnable() {
            @Override
            public void run()
            {

              finish();

              if (!mIsBackButtonPressed)
              {
                 // start the home activity
                 Intent intent = new Intent(Splash.this, Share.class);
                 Splash.this.startActivity(intent);
              }

            }

          }, SPLASH_DURATION);
        }

        //handle back button press
        @Override
        public void onBackPressed()
        {
          mIsBackButtonPressed = true;
          super.onBackPressed();
        }

      @Override
      public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the action bar if it is present.
            getMenuInflater().inflate(R.menu.activity_splash, menu);
            return true;
      }
}
```

Output:

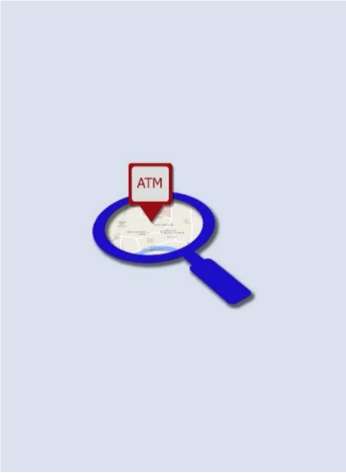


Figure 21 Screenshot of Splash Screen

//Share.java: The code below creates the User Interface for clickable buttons that provide the ways for finding the nearest ATMs of the preferred bank.

```
public class Share extends Activity{
         // GPSTracker class
           GPSTracker gps;
           protected static final Context context = null;
           public void onCreate(Bundle savedInstanceState)
                 {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_share);

        getActionBar().setBackgroundDrawable(getResources().getDrawable(R.drawa
        ble.bg));
     addListenerOnButton();
        }
        public void addListenerOnButton()
        {
                Button click1= (Button)findViewById(R.id.automatic);
                click1.setOnClickListener(new OnClickListener()
                {

                        public void onClick(View arg0)
                        {
                                gps = new GPSTracker(Share.this);
                                // check if GPS enabled
                                if(gps.canGetLocation()){

             double latitude = GPSTracker.getLatitude();
             double longitude = GPSTracker.getLongitude();
                     Toast.makeText(getApplicationContext(), "Your Location is:
        \nLatitude: " + latitude + "\nLongitude: " + longitude,
        Toast.LENGTH_LONG).show();
```

```
      Intent link=new Intent(Share.this,Banklist.class);
                    Share.this.startActivity(link);
    }else{
      // can't get location
      // GPS or Network is not enabled
      // Ask user to enable GPS/network in settings
      gps.showSettingsAlert();
    }

  }
});
        Button click2= (Button)findViewById(R.id.sms);
        click2.setOnClickListener(new OnClickListener()
        {
              public void onClick(View arg0)
              {
                    Intent link=new Intent(Share.this,Sms.class);

                    //link.setAction()
                    Share.this.startActivity(link);
              }
        });
        Button click3= (Button)findViewById(R.id.abt);

        click3.setOnClickListener(new OnClickListener()
        {
              public void onClick(View arg0)
              {
                    Intent link=new Intent(Share.this,Help.class);
                    //link.setAction()
                    Share.this.startActivity(link);
              }
        });
        }
}
```

Output:

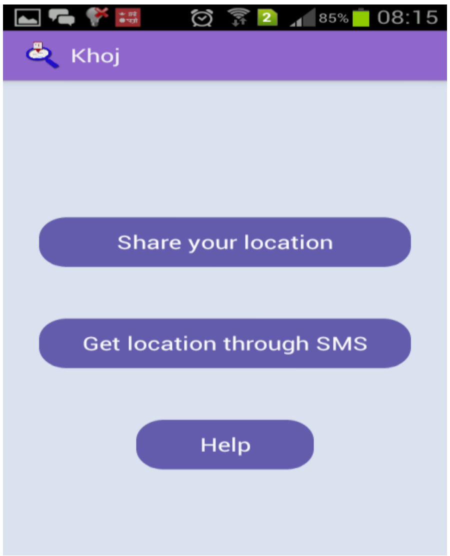
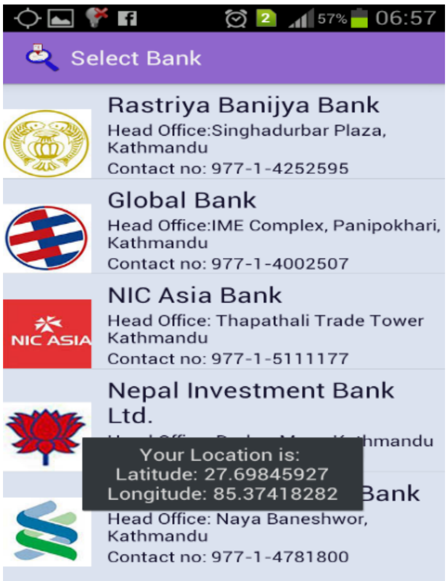


Figure 22 Screenshot of GUI for first activity

//BankList.java: The code below creates the list of available banks which generates the corresponding nearest ATM locations when clicked.

```
protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_banklist);
        getActionBar().setBackgroundDrawable(
                getResources().getDrawable(R.drawable.bg));
        getActionBar().setTitle("Select Bank");
        mListView = (ListView) findViewById(R.id.mylist);

        List<HashMap<String, String>> data = GetSampleData();

        SimpleAdapter adapter = new SimpleAdapter(this, data,
                     R.layout.banklist_row, new String[] {
       "image","bname","loc","phone" },
                     new int[] {
       R.id.image,R.id.bname,R.id.loc,R.id.phone});

       mListView.setAdapter(adapter);

       // onClick

       mListView.setOnItemClickListener(new OnItemClickListener() {

              @Override
              public void onItemClick(AdapterView<?> parent, View view,
                            int position, long id) {

                     if (position == 0) {
                            Intent link = new Intent(Banklist.this,
              AtmListActivity.class);
                            link.putExtra("bankName", "RBB");
                            Banklist.this.startActivity(link);
                     }
              }
       });

}
        @Override
  public void onBackPressed() {
    android.os.Process.killProcess(android.os.Process.myPid());
    super.onBackPressed();
  }
  public void webServiceCallFinished(String response, String invokerId) {
    if (response == null || response.equalsIgnoreCase("[]")) {
          Toast.makeText(this, "Error connecting to the server.. Please check your
         internet connection",
                        Toast.LENGTH_LONG).show();
               } else {
                  try {
                           System.out.println("hello");
                     JSONObject atmJSONObject = new JSONObject(response);
                     String ack = atmJSONObject.getString("ack");
                     if (ack.equalsIgnoreCase("failure")) {
                        Toast.makeText(getApplicationContext(),
atmJSONObject.getString("ack_message"),
                              Toast.LENGTH_SHORT).show();
                        return;
                     }
                     JSONArray atmJSONArray =
atmJSONObject.getJSONArray("result");
                     for (int i = 0; i < atmJSONArray.length(); ++i) {
                        JSONObject JSONObject = atmJSONArray.getJSONObject(i);
                        int id = JSONObject.getInt("id");
                        System.out.println("id");
                        String bank = JSONObject.getString("bankCode");
                        System.out.println(bank);
                                 if(JSONObject.has("name")){
                                    for(int j=0; j<atmJSONArray.length(); ++j){
                              JSONObject JSONObjectInner =
atmJSONArray.getJSONObject(j);
                              String lat = JSONObjectInner.getString("lat");
                              }


                              }
                     }
                  } catch (JSONException e) {
                    Toast.makeText(this, e.getMessage(),
Toast.LENGTH_SHORT).show();
                    e.printStackTrace();
                  }
                }
              }
        List<HashMap<String, String>> GetSampleData() {
                  int[] pic = new
int[]{R.drawable.a1,R.drawable.a2,R.drawable.a3,R.drawable.a4,R.drawable.a5};

                  List<HashMap<String, String>> list = new
ArrayList<HashMap<String, String>>();

                  HashMap<String, String> map = new HashMap<String, String>();



                  map.put("bname", "Rastriya Banijya Bank");
                  map.put("loc", "Head Office:Singhadurbar Plaza,Kathmandu");
                  map.put("phone", "Contact no: 977-1-4252595");
                  map.put("image", String.valueOf(pic[0]));
                  list.add(map);

}

}
```

Output:



57% 06:57

Select Bank

Rastriya Banijya Bank
Head Office:Singhadurbar Plaza, Kathmandu
Contact no: 977-1-4252595

Global Bank
Head Office:IME Complex, Panipokhari, Kathmandu
Contact no: 977-1-4002507

NIC Asia Bank
Head Office: Thapathali Trade Tower Kathmandu
Contact no: 977-1-5111177

Nepal Investment Bank Ltd.
hmandu

Your Location is:
Latitude: 27.69845927
Longitude: 85.37418282

Bank
Head Office: Naya Baneshwor, Kathmandu
Contact no: 977-1-4781800

Figure 23 Screenshot displaying user's geolocation as well as banklist

// WebServiceCall.java

```
package com.adifynepal.khoj;

public interface WebServiceCall {
        public void webServiceCallFinished(String response, String invokerId);
}
```

// WebServiceUtils.java

```
public class WebServiceUtils {
        public static HttpClient getHttpClient(){
                HttpParams httpParams = new BasicHttpParams();
      HttpConnectionParams.setConnectionTimeout(httpParams, 5000);
      HttpConnectionParams.setSoTimeout(httpParams, 60000);
      HttpClient httpClient = new DefaultHttpClient(httpParams);

      return httpClient;
        }
}
```

```
// WebserviceActivity.java

import android.app.Activity;
public abstract class WebserviceActivity extends Activity implements
WebServiceCall {

}

// TabsPagerAdapter.java

public class TabsPagerAdapter extends FragmentPagerAdapter {
  private String[] tabNames = { "Class Schedule", "Classes", "Tutors", "Students",
"Payment Reminder", "Inbox" };
  private HomeActivity ctx;
  public TabsPagerAdapter(HomeActivity ctx, FragmentManager fm,
android.app.ActionBar actionBar, TabListener tabListener) {
    super(fm);
    this.ctx = ctx;
    for (int i = 0; i < tabNames.length; i++) {
      actionBar.addTab(actionBar.newTab().setText(tabNames[i])
          .setTabListener((TabListener) tabListener));
    }
     Handler handler = new Handler();
            handler.postDelayed(new Thread(){
                  public void run(){
                        TabsPagerAdapter.this.ctx.setPagerAdapter();
                  }
            }, 100);
  }

  @Override
  public Fragment getItem(int index) {
    return null;
```

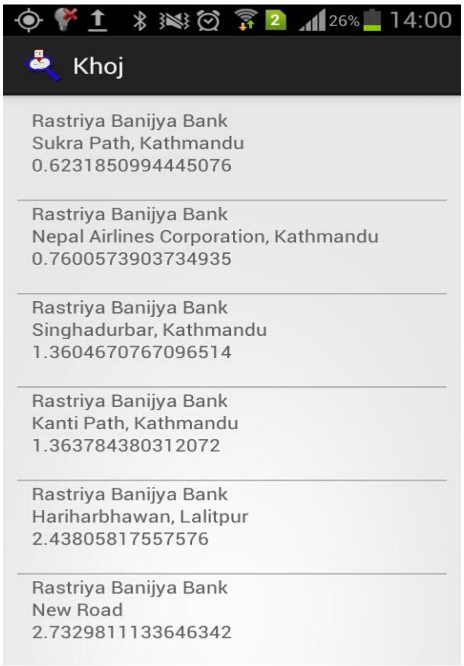
```
   }
   @Override
   public int getCount() {
      return tabNames.length;
   }
}

// StreamUtils.java
public class StreamUtils {
        public static String getStringFromStream(InputStream is){
                try{
                String result = "";
        BufferedReader br = new BufferedReader(new InputStreamReader(is));
        String temp = "";
        while((temp = br.readLine())!=null){
                result += temp;
        }

        return result;
                }catch(Exception e){
                        e.printStackTrace();
                }
                return null;
        }

        public static String xmlNodeToString(Element elem){
                try{
                        Transformer transformer =
                        TransformerFactory.newInstance().newTransformer();
                        transformer.setOutputProperty(OutputKeys.INDENT, "yes");

                        StreamResult result = new StreamResult(new StringWriter());
                        DOMSource source = new DOMSource(elem);
                        transformer.transform(source, result);

                        String xmlString = result.getWriter().toString();
                        return xmlString;
                }catch(Exception e){
                        e.printStackTrace();
                }
                return "";
        }
}
```

//AtnListActivity.java

```
public class AtmListActivity extends WebserviceActivity implements
WebServiceCall {
   public static ArrayList<Atm> atmArrayList;
   String a;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.atm_listview);
       a = getIntent().getStringExtra("bankName");
      System.out.println(a);
     // getActionBar().setTitle("Student List");
      Button button = (Button) findViewById(R.id.addBtn);
      button.setVisibility(View.GONE);

  List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
                   nameValuePairs.add(new BasicNameValuePair("request_name",
"getAtms"));
        nameValuePairs.add(new BasicNameValuePair("lat", "myLatitude"));
        nameValuePairs.add(new BasicNameValuePair("lng", "myLongitude"));
        nameValuePairs.add(new BasicNameValuePair("radius", "15"));
```

```
    nameValuePairs.add(new BasicNameValuePair("bankCode", a));
    WebserviceAsyncTask asyncTask = new
WebserviceAsyncTask(AtmListActivity.this, "Atm Search",
        "Searching nearest Atms...", nameValuePairs, "fetcha");
    asyncTask.setWebServiceCallFinishedListener(new WebServiceCall() {
      @Override
      public void webServiceCallFinished(String response, String invokerId) {
        AtmListActivity.this.webServiceCallFinished(response, invokerId);
      }
    });
    asyncTask.execute(ApplicationSettingsFilePath.baseUrl);
  }

  public void webServiceCallFinished(String response, String invokerId) {
    if (response != null) {
      atmArrayList = new ArrayList<Atm>();
      try {
              JSONObject atmJSONObject = new JSONObject(response);
              String ack = atmJSONObject.getString("ack");
              if (ack.equalsIgnoreCase("failure")) {
                Toast.makeText(getApplicationContext(),
atmJSONObject.getString("ack_message"),
                    Toast.LENGTH_SHORT).show();
                return;
              }
              JSONArray atmJSONArray =
atmJSONObject.getJSONArray("result");
              for (int i = 0; i < atmJSONArray.length(); ++i) {
                JSONObject JSONObject = atmJSONArray.getJSONObject(i);
                int id = JSONObject.getInt("id");
                System.out.println("id");
                String name = JSONObject.getString("name");
                String address = JSONObject.getString("address");
}
                String lat = JSONObject.getString("lat");
                String lng = JSONObject.getString("lng");
                String bankCode = JSONObject.getString("bankCode");
                String distance = JSONObject.getString("distance");
                System.out.println(bankCode);
                atmArrayList.add(new Atm(id, name, address, lat, lng,
      bankCode, distance));
        }

      } catch (JSONException e) {
          Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
          e.printStackTrace();
      }
      ListView listView = (ListView) findViewById(R.id.atmlist);
      AtmListAdapter adapter = new AtmListAdapter(this, atmArrayList);
      if(atmArrayList.size() == 0){
          Toast.makeText(getApplicationContext(), "No records",
Toast.LENGTH_SHORT).show();
      }
      listView.setAdapter(adapter);
      System.out.println("adapterset");
      listView.setOnItemClickListener(new OnItemClickListener() {
          public void onItemClick(AdapterView<?> adapterView, View v, final int
position, long id) {
                  Map.atmObject = atmArrayList.get(position);
                  Intent intent = new Intent(AtmListActivity.this, Map.class);
                  AtmListActivity.this.startActivity(intent);
          }
      });
  }
  }
}
```

Output:



Rastriya Banijya Bank
Sukra Path, Kathmandu
0.623185099445076

Rastriya Banijya Bank
Nepal Airlines Corporation, Kathmandu
0.7600573903734935

Rastriya Banijya Bank
Singhadurbar, Kathmandu
1.3604670767096514

Rastriya Banijya Bank
Kanti Path, Kathmandu
1.363784380312072

Rastriya Banijya Bank
Hariharbhawan, Lalitpur
2.43805817557576

Rastriya Banijya Bank
New Road
2.7329811133646342

Figure 24 Screenshot for nearest locations of the corresponding Bank

//Map.java : The code below connects the application with Google Map and displays the route along with marker form user's current location to the preferred ATM's location.

```
protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_map);
     v2GetRouteDirection = new GMapV2GetRouteDirection();
               String destLat = atmObject.lat.toString();
               String destLon = atmObject.lng.toString();
               destLat1 = Double.parseDouble(destLat);
    destLon1 = Double.parseDouble(destLon);
     MapFragment MapFragment = (MapFragment) getFragmentManager()
          .findFragmentById(R.id.map);
          mGoogleMap = MapFragment.getMap();

   // Enabling MyLocation in Google Map
     mGoogleMap.setMyLocationEnabled(true);
     mGoogleMap.getUiSettings().setZoomControlsEnabled(true);
mGoogleMap.getUiSettings().setCompassEnabled(true);
mGoogleMap.getUiSettings().setMyLocationButtonEnabled(true);
mGoogleMap.getUiSettings().setAllGesturesEnabled(true);
mGoogleMap.setTrafficEnabled(true);
mGoogleMap.animateCamera(CameraUpdateFactory.zoomTo(12));
markerOptions = new MarkerOptions();
marker=new MarkerOptions();
myLatitude=GPSTracker.getLatitude();
myLongitude=GPSTracker.getLongitude();
fromPosition = new LatLng(myLatitude,myLongitude);
toPosition = new LatLng(27.6843379,85.2959103);
GetRouteTask getRoute = new GetRouteTask();
getRoute.execute();

  mGoogleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
//mGoogleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
// mGoogleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
// mGoogleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);

}

private class GetRouteTask extends AsyncTask<String, Void, String> {
private ProgressDialog Dialog;
String response = "";
@Override
protected void onPreExecute() {
     Dialog = new ProgressDialog(Map.this);
     Dialog.setMessage("Loading route...");
     Dialog.show();
}
@Override
protected String doInBackground(String... urls) {
     //Get All Route values
```

```
 document = v2GetRouteDirection.getDocument(fromPosition, toPosition,
GMapV2GetRouteDirection.MODE_DRIVING);
                response = "Success";
            return response;
        }
        @Override
        protected void onPostExecute(String result) {
            mGoogleMap.clear();
            if(response.equalsIgnoreCase("Success")){
            ArrayList<LatLng> directionPoint =
v2GetRouteDirection.getDirection(document);
            PolylineOptions rectLine = new PolylineOptions().width(10).color(
                    Color.RED);

            for (int i = 0; i < directionPoint.size(); i++) {
                rectLine.add(directionPoint.get(i));
            }
            // Adding route and marker on the map
            mGoogleMap.addPolyline(rectLine);
            marker.position(fromPosition);
            marker.draggable(true);
            mGoogleMap.addMarker(marker);
            markerOptions.position(toPosition);
            markerOptions.draggable(true);
            mGoogleMap.addMarker(markerOptions);
            }
             Dialog.dismiss();
        }
    }
    @Override
    protected void onStop() {
        super.onStop();
        finish();
 }
          }
```

Output:

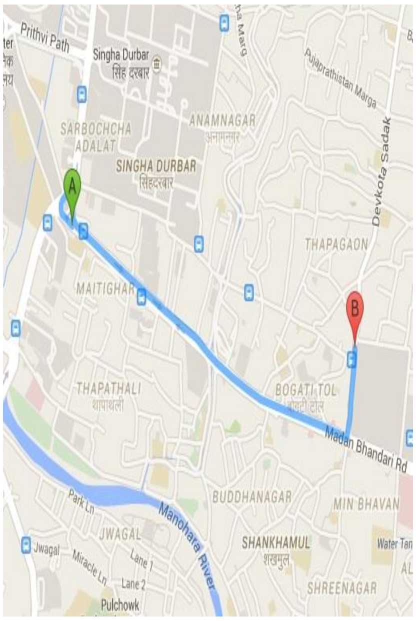


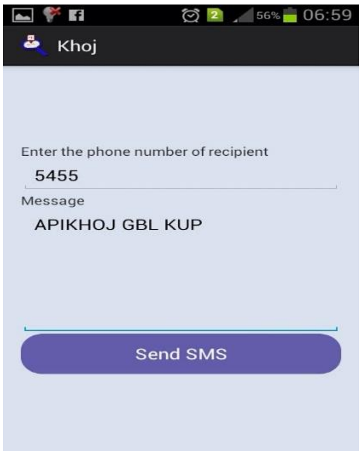
Figure 25 Screenshot of Google map showing route from user to nearest preferred ATM location

//Sms.java: The below code provides the application with the function of writing and sending the SMS.

```
public class Sms extends Activity {
            Button btnSendSMS;
            EditText txtPhoneNo;
            EditText txtMessage;

            /** Called when the activity is first created. */
            @Override
            public void onCreate(Bundle savedInstanceState)
            {
               super.onCreate(savedInstanceState);
               setContentView(R.layout.activity_sms);

               btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
               txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);
```

```
          txtMessage = (EditText) findViewById(R.id.txtMessage);
          btnSendSMS.setOnClickListener(new View.OnClickListener()
          {
            public void onClick(View v)
            {
              String phoneNo = txtPhoneNo.getText().toString();
              String message = txtMessage.getText().toString();
              if (phoneNo.length()>0 && message.length()>0)
                sendSMS(phoneNo, message);
              else
                Toast.makeText(getBaseContext(),
                  "Please enter both phone number and message.",
                   Toast.LENGTH_SHORT).show();
            }
          });
              }
       //---sends an SMS message to another device---
private void sendSMS(String phoneNumber, String message)
{
  String SENT = "SMS_SENT";
  String DELIVERED = "SMS_DELIVERED";
  PendingIntent sentPI = PendingIntent.getBroadcast(this, 0,
    new Intent(SENT), 0);
  PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0,
    new Intent(DELIVERED), 0);
  //---when the SMS has been sent---
  registerReceiver(new BroadcastReceiver(){
    @SuppressWarnings("deprecation")
            public void onReceive(Context arg0, Intent arg1) {
      switch (getResultCode())
      {
        case Activity.RESULT_OK:
          Toast.makeText(getBaseContext(), "SMS sent",
              Toast.LENGTH_SHORT).show();
          break;
                   case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                       Toast.makeText(getBaseContext(), "Generic failure",
              Toast.LENGTH_SHORT).show();
          break;
                   case SmsManager.RESULT_ERROR_NO_SERVICE:
          Toast.makeText(getBaseContext(), "No service",
              Toast.LENGTH_SHORT).show();
          break;
                   case SmsManager.RESULT_ERROR_NULL_PDU:
                   Toast.makeText(getBaseContext(), "Null PDU",
              Toast.LENGTH_SHORT).show();
          break;
                   case SmsManager.RESULT_ERROR_RADIO_OFF:
          Toast.makeText(getBaseContext(), "Radio off",
              Toast.LENGTH_SHORT).show();
          break;
      }
    }
  }, new IntentFilter(SENT));

  //---when the SMS has been delivered---
  registerReceiver(new BroadcastReceiver(){
    public void onReceive(Context arg0, Intent arg1) {
      switch (getResultCode())
      {
        case Activity.RESULT_OK:
          Toast.makeText(getBaseContext(), "SMS delivered",
              Toast.LENGTH_SHORT).show();
          break;
        case Activity.RESULT_CANCELED:
          Toast.makeText(getBaseContext(), "SMS not delivered",
                    Toast.LENGTH_SHORT).show();
                break;
            }
        }
    }, new IntentFilter(DELIVERED));
 SmsManager sms=SmsManager.getDefault();
sms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI);
            }
}
```

Output:



Enter the phone number of recipient
5455

Message
APIKHOJ GBL KUP

Figure 26 Screenshot of SMS GUI Interface for querying nearest ATMs of corresponding bank

Following code shows the permissions used in AndroidManifest.xml of the android application:

```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"
/>
```

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
```

Following code shows the activities or the java classes used in the application in AndroidManifest.xml of the android application:

```
    <activity
    android:name="com.adifynepal.khoj.Share"
    android:label="@string/app_name" >
  </activity>

  <activity
    android:name="com.adifynepal.khoj.Sms"
    android:label="@string/app_name" >
  </activity>
  <activity
    android:name="com.adifynepal.khoj.SmsReceiver"
    android:label="@string/app_name" >
  </activity>
  <activity
    android:name="com.adifynepal.khoj.GMapV2GetRouteDirection"
    android:label="@string/app_name" >
  </activity>
  <activity
    android:name="com.adifynepal.khoj.Banklist"
    android:label="@string/app_name" >
  </activity>
  <activity
    android:name="com.adifynepal.khoj.Help"
```

```
android:label="@string/app_name" >
</activity>
<activity
   android:name="com.adifynepal.khoj.Map"
   android:label="@string/app_name"
   android:screenOrientation="landscape" >
</activity>
<activity
   android:name="com.adifynepal.khoj.GPSTracker"
   android:label="@string/app_name" >
</activity>
   <activity
   android:name="com.adifynepal.khoj.AtmListActivity"
   android:label="@string/app_name" >
</activity>
```