

Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering



Bachelor Thesis

Offline GPS Application for Android in Mongolia

Purevdorj Tserendorj

© 2016 CULS Prague

Declaration

I declare that I have worked on my bachelor thesis titled "Offline GPS Application for Android in Mongolia" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the bachelor thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on 30.11.2016

Acknowledgement

I am grateful especially to my supervisor Ing. Jiří Brožek, Ph.D for his patient guidance and valuable suggestions.

My next gratitude goes to Czech University of Life Sciences Prague and Ministry of Education, Czech Republic for giving me the opportunity to finance my studies.

Last but not least, I am truly thankful to my family for their endless support.

Offline GPS aplikace pro Android v Mongolsku

Souhrn

Cílem této práce je rozvíjet offline GPS aplikace pro Android v Mongolsku. Aplikace je zaměřena a má být použita pouze v Mongolsku.

Pro tento účel se používá studio Android a Java programování je provedeno. Zpočátku používáme OpenStreetMap data a také MapsForge library pro vykreslení mapy a plánování silničních výlet. Pro směřování účelu byl použit GraphHoppers. Po dokončení design aplikace, aplikace byla testována a hodnocena uživateli. Obecně 7 úkoly byly dány za účelem testování, reálné výsledky a očekávané výsledky jsou shodné na 100% ve všech úkolech.

Výhody aplikace: nevyžaduje připojení k síti, není omezen přesné adresování, služba sledování je k dispozici.

Nevýhody aplikace: žádná funkce psaní názvu adresy, připojuje pouze 2 body, neaktualizuje obrazovku.

Klíčová slova: Navigační systém, aplikace pro Android, Mongolsko, Globální systém poloha, Offline aplikace, OpenStreetMap, Mapa

Offline GPS Application for Android in Mongolia

Summary

The objective of the thesis is to develop an offline GPS application for Android in Mongolia and the application is intended to be used only in Mongolia.

For this purpose, Android Studio is employed and Java programming is executed. Initially, we use OpenStreetMap data and MapsForge library for map rendering and planning road trips. For routing purposes, GraphHopper was employed. After completing the design of the application, the application was tested and evaluated by users. In general, 7 tasks were given to test the application and test executions and expected executions are met 100 percent in all tasks.

Advantages of the developed application are: does not require network connection, does not limit by exact addressing and tracking service is available. Limitations of the application are: no function of typing name of the address, only connects 2 points and does not refresh the screen.

Keywords: Navigation system, Android application, Mongolia, Global position system, Offline application, OpenStreetMap, Map

Table of Contents

1. Introduction	8
2. Objectives and Methodology	10
2.1. Objectives	10
2.2. Methodology	10
3. Mongolian Market Description	11
3.1. Analysis of Target Users.....	11
3.2. Drivers in Mongolia.....	11
3.3. Review of Available Applications	11
3.3.1. GeoNet.....	12
3.3.2. Google Map.....	12
3.3.3. OsmAnd.....	13
3.3.4. Comparison of available applications.....	14
4. Literature Review	15
4.1. Overview of Navigation System	15
4.1.1. Global Navigaiton Satellite System	15
4.1.2. Global Position System.....	16
4.1.3. Assisted Global Position System	18
4.2. Review of Android System	19
4.2.1. Android Platform.....	19
4.2.2. Android structure.....	21
4.3. Review of the Software Development Kit	23
4.3.1. Google Maps Android API's	23
4.3.2. OsmDroid	23
4.3.3. Evaluation of SDK's.....	23
5. Architecture	25
5.1. Map Rendering	25
5.2. Navigation between screen	25
5.3. Use cases Diagram	26
5.3.1. Route planning.....	26
5.3.2. Managing Route.....	27
5.3.3. Managing Navigation	28
5.4. GUI Design	28
5.4.1. Planner UI Design	28
5.4.2. Map screen and Route UI design.....	29
6. Implementation	32
6.1. Componenet of interface	32
6.2. Data Preparation.....	33
6.3. Implementation Structure	34
6.4. Limits of Application.....	36
6.5. Application testing.....	37
7. Conclusion	39
8. Bibliography	Error! Bookmark not defined.

List of Figures

Figure 3.3.1: GeoNet User Inteface	12
Figure 3.3.2: GoogleMap User Inteface	13
Figure 3.3.3: OsmAnd User Inteface	13
Figure 4.2: Global Position System	17
Figure 4.3: Assistent Global Position system	18
Figure 4.5: Components of Android	22
Figure 5.1: Navigation Between screen	26
Figure 5.3.1: Use Case of Route Planning	27
Figure 5.3.2: Use Case of Managing Route	27
Figure 5.3.3: Use Case of Managing Navigation	28
Figure 5.4.1: Navigation Screen User interface	29
Figure 5.4.2: Navigation Screen User interface	30

List of Tables

Table 3.1: Comparison of 3 main applications	14
Table 4.1: Evaluation of the SDK's of map	24

1. Introduction

In many of countries, GPS application on smart phone and tablet is widely used by drivers in the city and out of the city. It not only simplify the driver's route also it utilize driver's time and finance. Firstly, GPS application was paid application but nowadays, there exist many applications which are free and do not require network connection. Basicly these free applications are suitable for basic users and more advanced versions are available with payments.

In Mongolian market, there are three widely used applications those have free offers regarding to offline navigation application. These applications are GeoNet, GoogleMap and OsmAnd and they have offers for worldwide. But when this worldwide map comes to Mongolian map (specially in capital city Ulaanbaatar), addressing becomes the most problematic issue. Because there exists large number of very small streets those does not even have name and included in worldwide map. Also network connection is very limited in Mongolia considering the landscape and money.

Additionally, citizens of Ulaanbaatar tend to know places by its location not by the exact address name. Thus giving an address as final destination in GPS application is most likely unnecessary tool for users. But it is necessary tool for tourists. There are many tourists holding paper maps in Mongolia and they tend to ask the place by address from someone on the street. Unfortunately, they do not get much of help from them. So the aim of this thesis is to develop offline GPS application on android that does not require network connection and with addition tool of no exact addressing. Also developed application will be offered on the market for free.

Finally, application will be developed for Android. First of all, Android platform is ideal to develope such application. Because there are many users and Android OS is open platform that has no restriction for programmer from system vendor like in iOS. Secondly, Android platform is quite widespreaded in the Mongolia considering the lower price.

The thesis is structured as follows: Chapter 3 and Chapter 4 provide all the necessary terms about systems related to the offline GPS application and potential users of this application. First of all, in Chapter 3 a short description of GPS application used by Mongolian citizens is given to have better understanding of Mongolian market. Also comparison of most used GPS applications in Mongolia is presented. Then in Chapter 4 author gives the most important terms and required survey related to the offline

navigation system based on existing literatures. This chapter is divided into three main parts: Overview of system, Review of maps and Review of Software Development Kit (SDK). Chapter 5 describes the architecture of the application including Use cases diagram and GUI design. Finally, Chapter 6 introduces the implementation and structure of author's developed application. Chapter 7 summarizes all research done in this thesis and concludes.

2. Objectives and Methodology

2.1.Objectives

The aim of the thesis is to analyse the current state and disadvantages of GPS navigation applications for use in Mongolia and design and implement navigation application addressing such disadvantages.

2.2.Methodology

Methodology of the thesis is based on study of scientific and technical sources and survey of the existing applications. Based on the gained knowledge an analysis of improved navigation application will be conducted. The application will then be implemented using Java language, utilizing map sources by the Ministry of Construction and Urban Development in Mongolia and the Ministry of Road and Transportation. The application will be deployed and tested. Possible ways of improving and extending the application will be discussed based on the results of the testing.

3. Mongolian Market Description

In this chapter we go through brief overview of Mongolian market with the short description of target users of Mongolia. By this way, we can have a complete picture of the structure of this market. Additionally, analysis of similar applications available on the Mongolian market is provided.

3.1. Analysis of Target Users

Mongolia is the 19th largest country with the 1,564,116 square kilometers landscape in the world and one of the most dispersed country (Portal n.d.). The country is located in east central Asia and sandwiched between two powerful nations: Russia to the north and China to the south. Mongolian population is just turned 3 million on year 2014 and approximately half of the population is located in the capital city Ulaanbaatar. In Mongolia, only 9991919 kilometers roads were paved including roads of capital city (National statistics office n.d.). Nowadays almost every Mongolian uses smart phone and demand for downloading GPS application on the phone is very high. Although Mongolian market offers pre-paid GPS navigation on the smart phone. According to statistical survey of Mongolian State of Governance in 2014-2015 (State of Governance n.d.), 3 of 10 people expect free (no payment) application which means more than 30 percents of the population are interested in using GPS navigation on their smart phones for free of charge.

3.2. Drivers in Mongolia

According to the National Statistic Office's survey in 2014-2015, citizens make road trips every year specially during summer. These people plan their road trips without GPS navigation and are likely to lost away or deviate from the planned route approximately by 100 kilometers and consume two or more hours to reach the destiny. This deviation is just in summer times and it expands by 2 or more times in winter times. Also this survey indicates that driver schools graduate approximately 200 new drivers in every 3 months which means offline GPS application users may increase up to 200 in every 3 months.

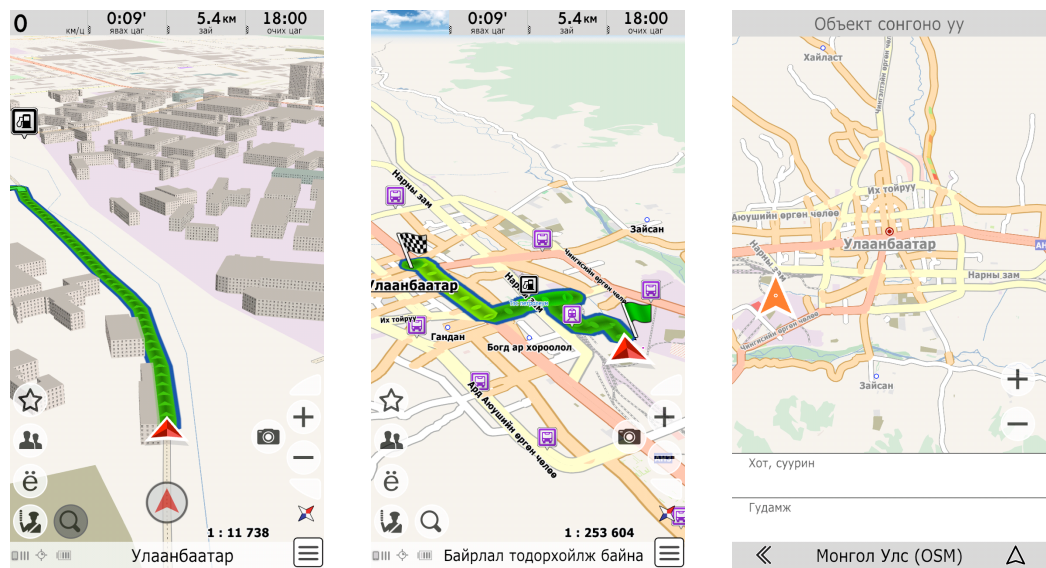
3.3. Review of Available Applications

In this section, author discusses about how many similar offline GPS navigations are available in Mongolia. According to author's survey, Mongolia has a very few number of free offline navigations and following three applications are widely used:

3.3.1. GeoNet

GeoNet is one of the biggest GPS navigator in the world that enables to use map from various map vendors. This project creates and distributes free geographic data around the world. GeoNet maps are based on OpenStreetMap (OSM) data which offers worldwide coverage and stable update.

Figure 3.3.1: GeoNet User Interface



Navigation A

Point Connection B

Point Selection C

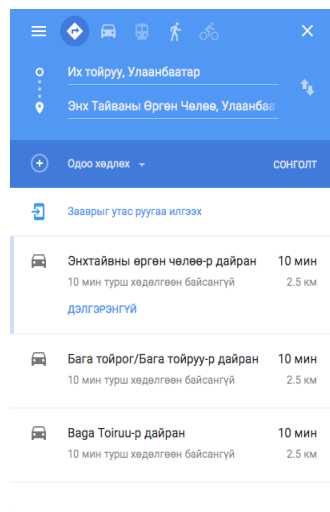
Source: Author himself

3.3.2. Google Map

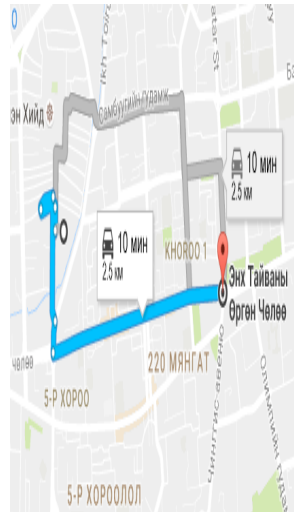
Lately, Google Map is considered to be one of the most widely used GPS navigation application in the world. However this application is very informative rather than being a professional one. There are number of cons in this application. For example:

- Online mode
- Not a opensource and only Google itself is allowed to edit the map
- Sometimes expected destination may not be reached
- No up-to-the-minute information on unusual conditions

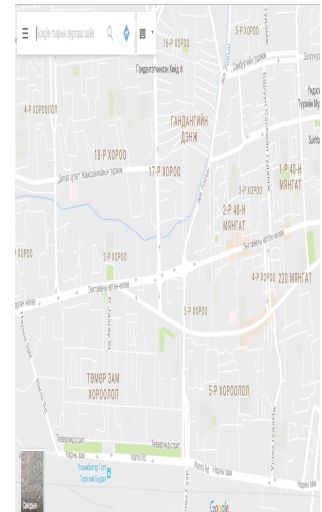
Figure 3.3.2: GoogleMap User Interface



Point Selection C



Point Connection B



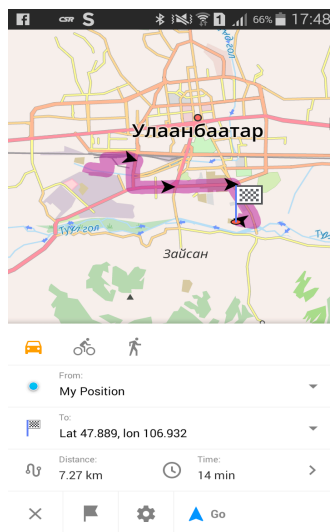
GoogleMap B

Source: Author himself

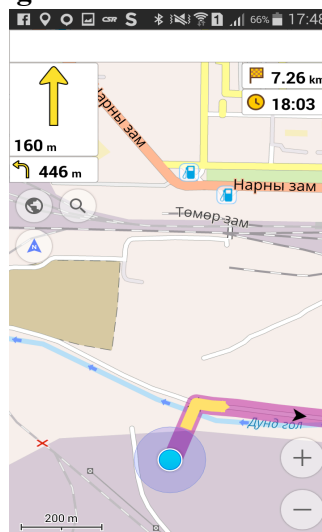
3.3.3. OsmAnd

OsmAnd stands for OpenstreetMap Automated Navigation Direction which means primary displays are based on Openstreetmap database. This navigation does not require network connection on global mobile map viewing. The android version of this application is available under the GPLv3 license and this navigation is free and open source. Also there exists payment required application version that unlocks the download limit for offline maps and provides access to Wikipedia POIs including description from the application.

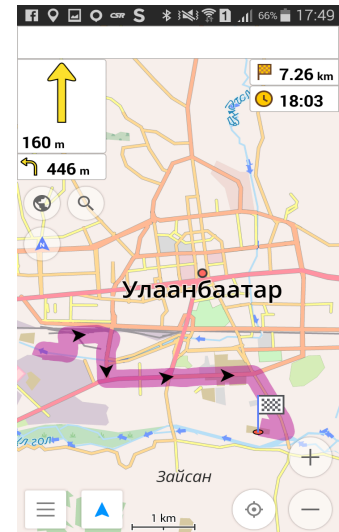
Figure 3.3.3: OsmAnd User Interface



Point Selection (A)



Navigation (B)



Point Connection (C)

Source: Author himself

3.3.4. Comparison of available applications

This comparison table shows available features of available applications. Looking at below Table 1, all three applications features are same with each others. Only differences are regarding to the offline mode, opensource and application charge.

Table 3.1: Comparison of 3 main applications

	GeoNet	GoogleMap	OsmAnd
Primarily	Yes	Yes	Yes
Raster	Yes	Yes	Yes
Vector	Yes	Yes	Yes
Can be used in Mongolia	Yes	Yes	No
Route type	any possibility	any possibility	any possibility
Offline mode	Yes	No	Yes
Payment requirement	Lite version is free for 14 days	No	No
OpenSource	Yes	No	Yes

Source: Author himself

4. Literature Review

The focus of the thesis is to develop offline GPS navigation for android in Mongolia. Consequently, literature review is concentrated on different studies analysing all the necessary features of Global Position System. This chapter has three main sections those are concentrated on Overview of the System, Review of the Map and Review of the Software Development Kit (SDK). Below author addresses the most important key factors on this thesis and then summarize it.

4.1. Overview of Navigation System

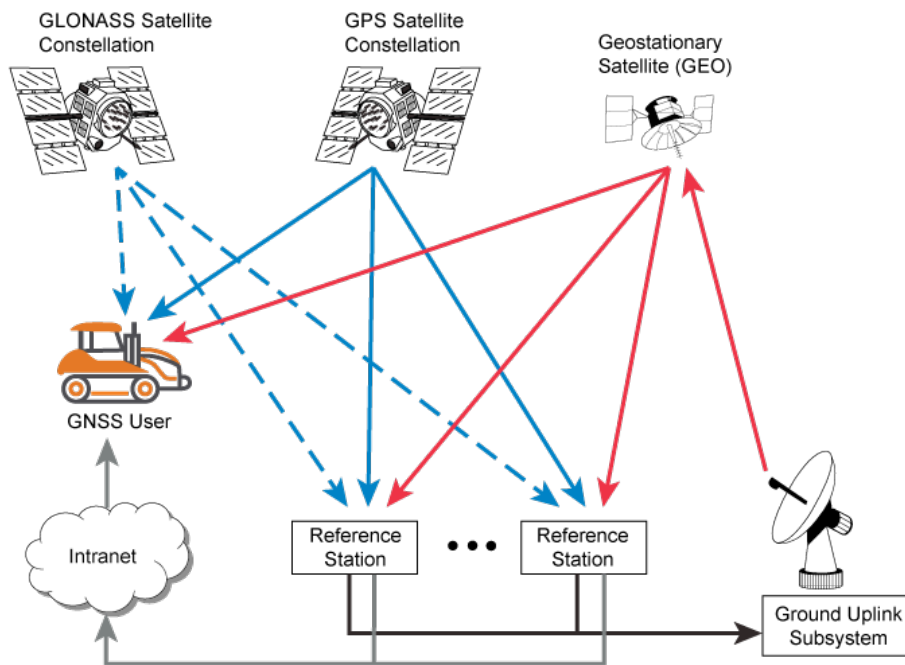
4.1.1. Global Navigation Satellite System

Global Navigation Satellite System (GNSS) is a system that uses satellite to provide geographic location of user globally. By using signals from a number of satellites, small electric receiver calculates involving information captured in signals and determine their location with expected time. According to Lance.A (National Academy of Engineering 2012) study, the system is used for tracking the position of electric receiver with high precision within in few meters to use time signal transmitted along a line of sight by radio from satellites.

Kornhauser (Kornhauser n.d.) states that signal allows the electronic receiver to calculate the current location with high precision and time synchronization. In study of Shengyue.J (Shengyue Ji 2010) generally, there are three main satellite in GNSS technologies which are GPS, GLONASS and GALILEO used by infinite number of users all over the world. According to Kornhauser (Kornhauser n.d.) each of these system has three main segments:

- Space segment: satellites in the space that transmit signals to the earth
- Control segment: management and maintenance of satellites that orbiting an earth
- User segment: receiver calculates information from signals and locate the position

Figure 4.1 Global Navigation Satellite System



Source: <http://www.precisionag.com/technology/guidance/novatel-launches-oem-positioning-solution/>

4.1.2. Global Position System

Global Position System (GPS) allows anyone on earth with a GPS receiver to get signals from the network of satellites that orbit the earth. In these signals tracking informations including time, speed and geo-coordination point are captured. In William.H (William H 1997) book, inspiration of creating GPS is came from Spuntik era in 1957 and during the Cold War in 1960s, US had developed it for military propose to track US submarines and ship navigation. At the beginning of it, only five satellites very placed on space thus the system was very slow and subs often had to wait at least an hour to change the position. The US military had increased satellites to 11 to update the position in a every minute.

According to article of Griffin.D (Griffin 2011), following features are considered regarding the reliability of GPS,:

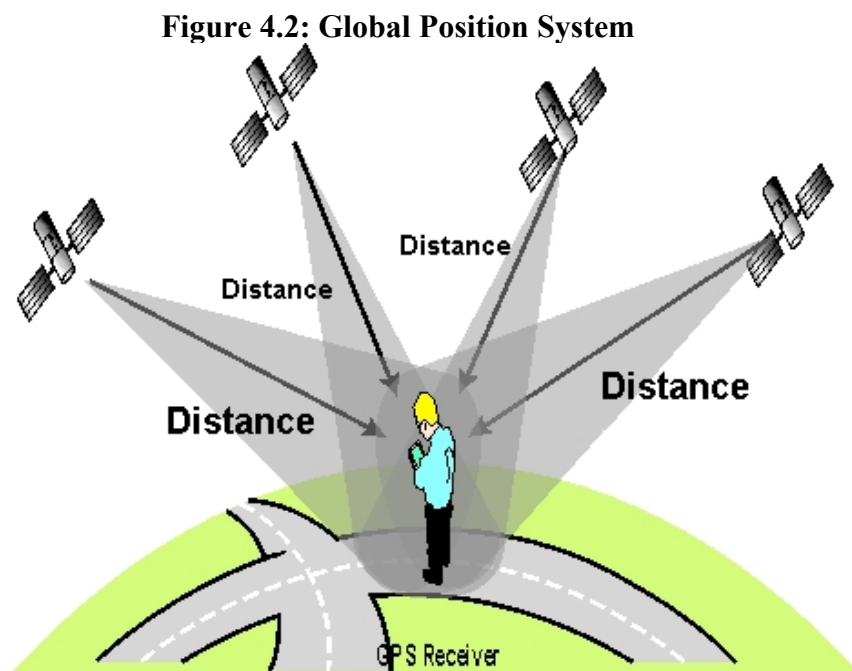
- **Position accuracy:** In most cases, it is about +/-10 metr but this accuracy fluctuate based on type of the GPS reciever. GPS satellite sends a signal with Time of Transmission to GPS reciever to measure to distance and time. Also

GPS signals are travelling at the speed of light and at least four GPS satellites are visible at a time.

- **Time accuracy:** Atomic clock usage brings very good time accuracy. Each GPS satellite owns four atomic clocks and only one of it is in use a time. Also sun-seeking solar panel and nicad batteries are charged.

In an article of Sherman (Sherman Lo 2009), it is written that mathematical algorithms is used by GPS receiver to decode the GPS satellites code that captures parameters, time stamp, correction data and other related information. And Kruger (G Kruger 1994) states that two-dimensional position is determined from that signals received from at least three satellite and these signals must capture Standard Position Receiver(SPS) code and Precise Position Service(PPS) or P code. For the three-dimensional position, signals from at least four satellite are required. Cooksey.D (Cooksey n.d.) says that there are underway programs for launching new generation of satellite that proved one more civilian signal additional to current signal.

There also exists Selective availability feature on GPS that reduces accuracy of SPS code. To increase the accuracy by 5-10 meters, this feature must be turned off. Shokouh.J (Shokouh 2013) mentions that to bring more accuracy for civilian users, Selective availability feature was turned off in year 2000 but meanwhile US government started to use Selective Denial instead to restrict the unauthorized access in a specified region.



Source: http://www.wikid.eu/index.php/Global_Positioning_System

In a year 2016, total of 31 operational GPS satellites are in orbit around the earth and they orbit the earth twice a day. 24 satellites are evenly divided into 6 orbital planes and 3 satellites are in a backup.

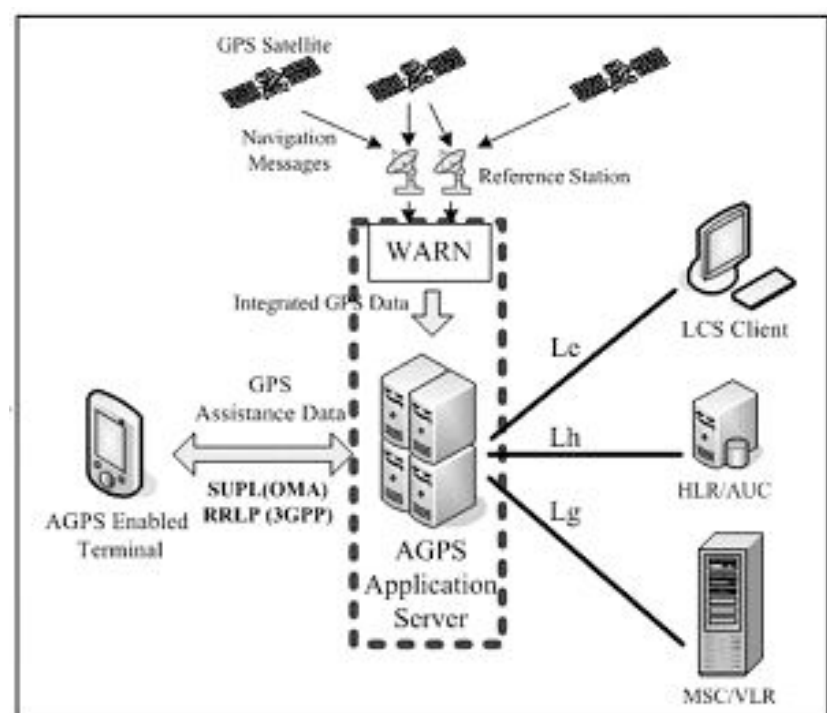
4.1.3. Assisted Global Position System

For GPS usage on cellular phones, Assisted global Position System (A-GPS) is the most suited technology to use. Because when it comes to cellular network, GPS's time preference of signal over the accuracy of position is reduced. In case of weak signal of GPS, A-GPS cooperates with cell tower data with addition to radio signals from satellites to deliver the signal faster to mobile network. Regarding to the reliability Poole.I (Poole n.d.) has written that A-GPS is slightly more accurate than GPS with accuracy of less than one meter. Also A-GPS has an operational costs due to usage of mobile network when GPS has no operational cost once the device is paid for.

Shokouh (Shokouh 2013) analyzes that A-GPS has two types of positioning types:

- i. Mobile station based: final position is calculated on the cellular device
- ii. Server station based: cellular device sends data to a server and server sends back the position data by calculating the final position in server network

Figure 4.3: Assistent Global Position system



Source: <https://technowizz.wordpress.com/2010/01/03/lbs-technologies-part-1/>

4.2. Review of Android System

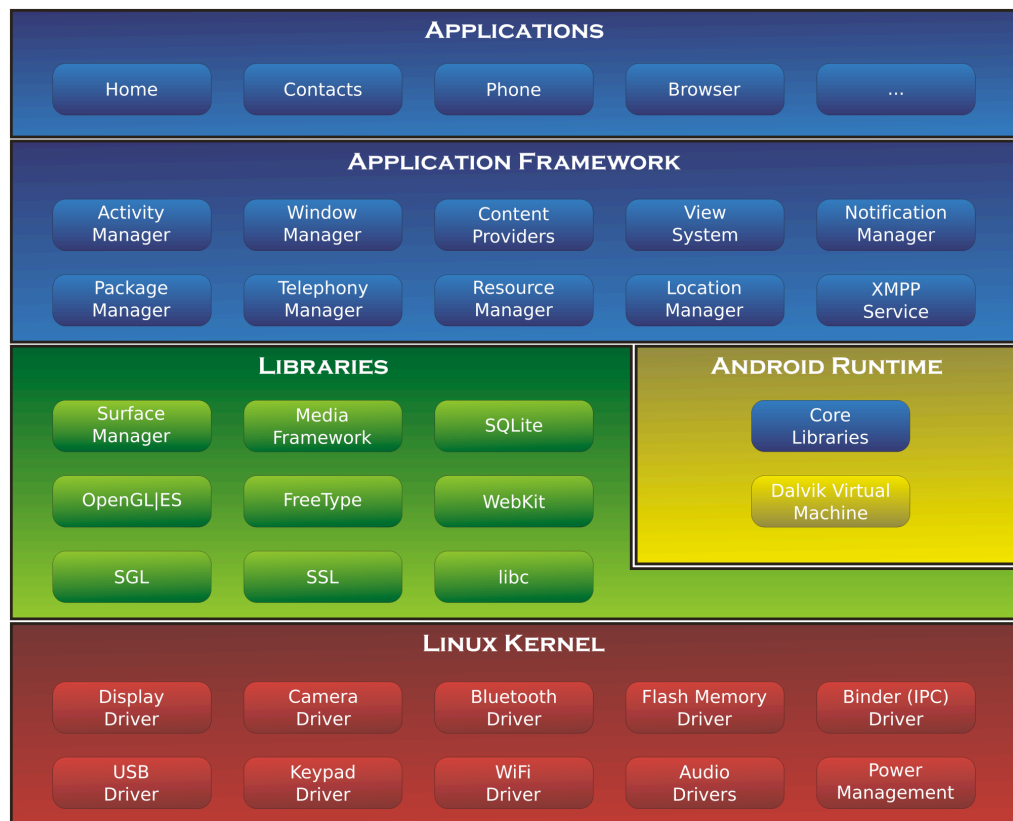
4.2.1. Android Platform

Android is an opensource operational system developed by Google and the platform of it is based on Linux Kernel. More general, Android system services are provided from Linux Kernel and link between the software stack and the hardware is known as kernel function (Android Developer n.d.). System services include security, memory management, process management, network stack, and driver model. The first stable Android was developed in October 2008 by Google and it was very innovative considering low software cost, easy to use interface and wider offer of application based on android platform. Also developers had benefited from complete API access availability, guarenteed increased productivity and user friendly developer tools.

Based on documents of (OpenCV.org 2016), to run and install applications on the Android platform the Android Software Development Kit (SDK) is required. Basically it uses the Java programming language and includes development tools, sample code, desktop emulator for testing and build in libraries to develope Android application.

According to Android Developer (Android Developer 2011), the applications run on virtual machine called Dalvik Android platform consists 5 subgroups which is named as Dalvik Virtual. As shown in Figure 4.4, there are 2 main components regarding to these subgroups. First component is linked to the hardware device which is a Linux Kerner which means application enables to contact with GPS device, camera, Bluetooth etc. Second component is linked to the software device.

Figure 4.4: Android Platform



Source: <https://novicesuresh.com/category/android-operating-system/>

- *Application layer*

Applications are developed by using Java programming language which run on this layer and all together build the the Android operation system. For example: home, contacts, phone, browser, messages, calendar, maps etc.

- *Application framework*

Development and innovation of applications are provided by application framework and implements the standard structure of the Android application. Interaction between existing applications and reuse of application's components are allowed in this framework. In Mikayelyan (Mikayelyan 2011) study, he captures that all applications consist of following frameworks:

- a. Views system of the application including lists, grids, text boxes, buttons and even an embeddable web browser.
- b. Content providers interaction between different applications' data such as contacts, photos etc.
- c. Resource Manager enables access of non-code resources such as graphics, layout files, etc.

- d. Notification Manager display alerts in the status bar
- e. Activity Manager is responsible for the lifecycle of the application

- *Libraries*

Android Developer (Android Developer 2011) states that to develop the libraries Java environment is required and it is developed in C and C++. As shown in Figure 4.4, there are nine main libraries which are Surface Manager, Media Framework, SQLite, OpenGLjES, FreeType, WebKit, SGL, SSL, libc. These libraries support Linux devices, web browser, bitmap, vector font rendering, SQL relational database, audio and image formats etc. (Android n.d.)

- *Android Runtime*

Android (Android n.d.) says that Android Runtime (ART) is the managed runtime used by applications. ART consists from core libraries which managed by Java programming language and the Dalvik virtual machine.

4.2.2. Android structure

Android system based applications vary from each others by defined components captured in an application. To develop a certain application, developer must predefine needed components and rest of the components should be isolated at install time. Mainly four different components are present to build the any kind of application. Based on specificity of an application some of components are used together or all of them. And those four components (Carlo U n.d.) are:

- i. Activity*

Usually the Activity component starts with the main activity by providing screen with user interface and the application owns number of activities itself. For example: for viewing a map, first two activities are shown in *Figure 5.6*. Each activity can start the next activity. While new activity is present, previous one is stopped and saved for the back stack. Back stack usually follows the rule of „last in, first out“ which means once Back button is clicked after current activity completion, previous activity appears from the back stack.

- ii. Service*

According to Lee.W (Lee 2012), service component is a background long-running operational performance. More specific, user can open a new application while another application still runs. For example: user can use Facebook application while play music is still processing.

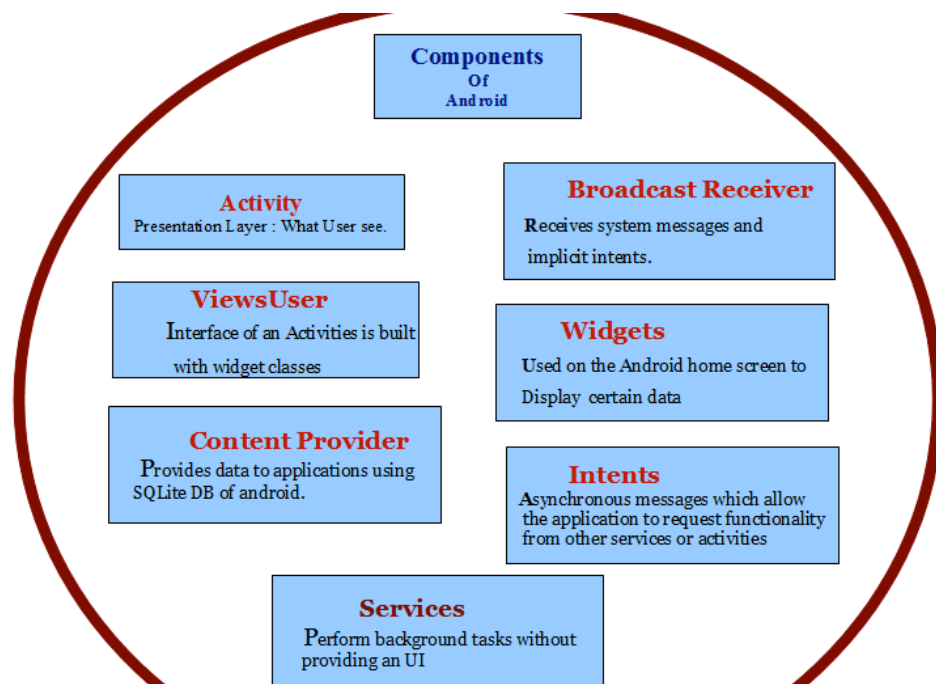
iii. *Content provider*

Content provider enables capturing data of all applications in one process. By this way, user can manipulate data in between applications. But user need his own provider to manipulate complex data from own application to other applications. For simplicity, maily used content providers are offered by Android.

iv. *Broadcast receiver*

Notification of running applications pops-up with the help of broadcast receiver while other application is running. For example: user is updating some applications while using navigation application and broadcast notify the current update status.

Figure 4.5: Components of Android



Source: <http://2.bp.blogspot.com/-o8dtusTX5CE/UCUqpE9vLZI/AAAAAAAAAD0/Shigw35qQY0/s1600/componentsOfAndroid.png>

In the website of Android Developer (Developer n.d.), it states that one of the advatanges using Android system is that components used in one application can also be used by other applications without developing components again. Explaining it: when you are capturing photo with device camera, it directs you to the another application which has needed components and delivers you the captures photo without developing a new activity.

Mikayelyan (Mikayelyan 2011) says that for security reason, Android requires that the components should be included in manifest file AndroidManifest.xml. To ensure it,

manifest needs declaration such as network connection, minimum API level, required hardware and software features for the application and needed API libraries. Addition to the manifest file, anything related to the visual presentation of the application is required in XML files.

4.3.Review of the Software Development Kit

4.3.1. Google Maps Android API's

According to writer's survey, Google Map is very useful tool to develop map application. Unfortunately Google Map does not allow to build offline GPS navigation for androids. Google Map API only for Android V2 or V3 is available for developers to add Google map in their map applications. According to Google Developer (Developer n.d.) once developer is done with registration, API key is provided to order Google maps and other Google APIs. Second step is that download of Google Play services is required to access Google Map APIs. According to Legat.V (Legat 2014), Google Map API uses OpenGL for rendering the map and it needs Google maps as a base. Developer can use maps with MapView and MapFragment or Mapview can be alone used with developer's new MapFragment (Android Developer n.d.).

4.3.2. OsmDroid

According to Google Code (Google Code 2012), OsmDroid is a third-part free open source library which renders OpenStreetMap data in to your screen and only *slf4j* android logging library is required for proper working. This basic function is extended by OSMbonusPack library. Same as Google Map, after the registration API key is provided and it allows to use MapQuest service for routing

Legat.V (Legat 2014) mentions that navigation routing is in xml format and this xml parser is kept in Library. Also OpenStreetMap POIs with Nominatim enables geocoding and reverse geocoding

4.3.3. Evaluation of SDK's

Below, author introduce that evaluation of Android map SDKs by using required features off location aware applications.

Table 4.1: Evaluation of the SDK's of map

	Google Map	Osmdroid	MapsForge	MapQuest
Map render	Vector	Raster	Vector	Raster
Overlays	Yes	Yes	Yes	Yes
Geocoding	Yes	Yes	No	Yes
Routing	Yes	Yes	No	Yes
Direction	Yes	Yes	No	Yes

Inspiration of this table is taken from the source: https://dip.felk.cvut.cz/browse/pdfcache/legatva1_2014bach.pdf

Looking at Table 2, there are a lot similarities between SDKs but major differences are given below:

- Google map requires to use Google Play Service, Google map APIs thus it is not opensource. Also it does not work on offline mode.
- Osmdroid is a library of OpenStreetMap. Basic functions are similar as Google map API and also it is not possible to use it on offline GPS application development.
- MapsForge only renders the OpenStreetMap and this does not function as a navigation.
- MapQuest functions almost same as Osmdroid and there might be some differences in the functionality.

5. Architecture

In this chapter, author deals with different approaches on how to build an intuitive application for Android devices. First two sections describe map rendering and navigation between screen. In the third section, author describes use cases diagrams along with description of the individual use cases. Lastly, images and description of GUI design of the main screen in application are provided.

5.1. Map Rendering

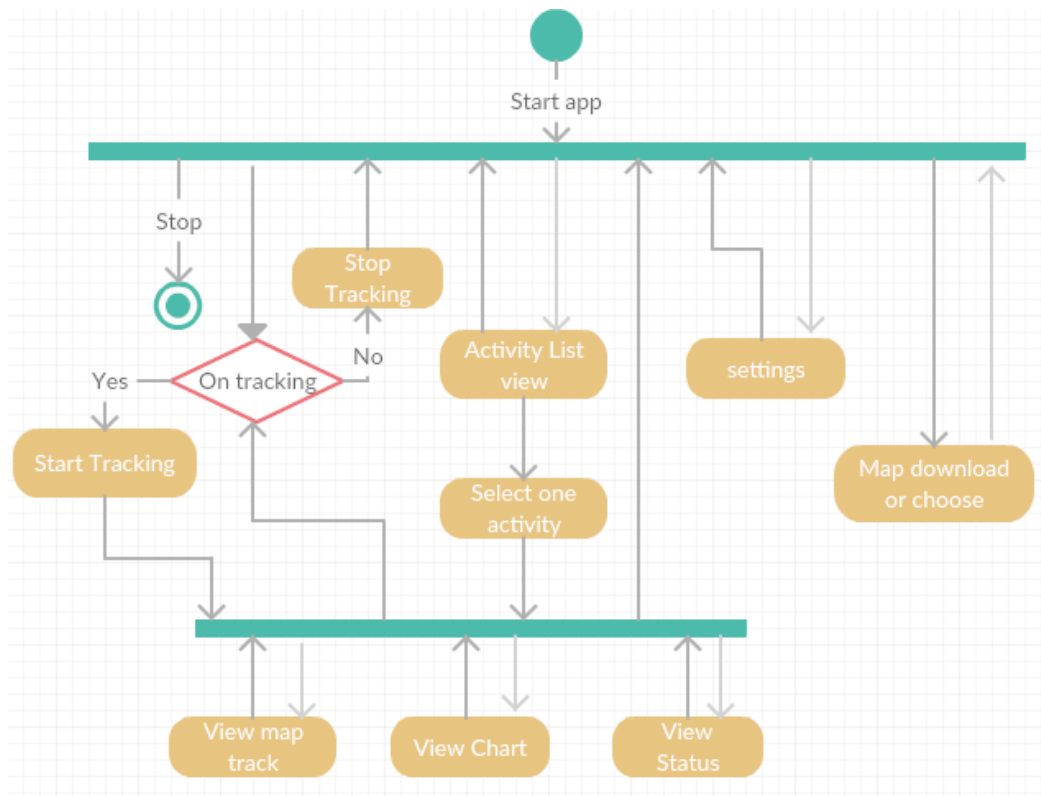
The most important features of GPS application is the map rendering and there exist a few possible methods to render the OpenStreetMap (Computer Science Institute n.d.). For example:

- Tile based rendering: Downloads a small part of a map through internet connection. Meanwhile it has a better user experience but high speed of mobile network is required.
- Memory management: This method works with offline mode but size vector map is limited. The Mongolian map size is over the 700 MB. Thus it is a big size on the mobile device but it has more liberty to use in this countryside road.
- Java OpenStreetMap (JOSM): This method is written in Java and desktop editing application and it runs on all operation systems. This is a compilated version of previous two methods. Because it supports loading stand alone GPX tracks and GPX track data from the OSM database (Basic Manual n.d.).
- The Mobile atlas creator is written in Java programing language and open source program. It can use a large number of different online maps such us OpenStreetMap and other map provider. Mobile Atlas Creator requires Java Runtime Environments 6 or higher version and high speed internet connection (Mobile Atlas Creator n.d.).

5.2. Navigation between screen

The navigation between screens is decribed in below **Figure 5.1**. The actions are connected with arrows and links the individual screens together. The navigation structure is formed by the navigation drawer pattern.

Figure 5.1:Navigation Between screen



Source: Author himself

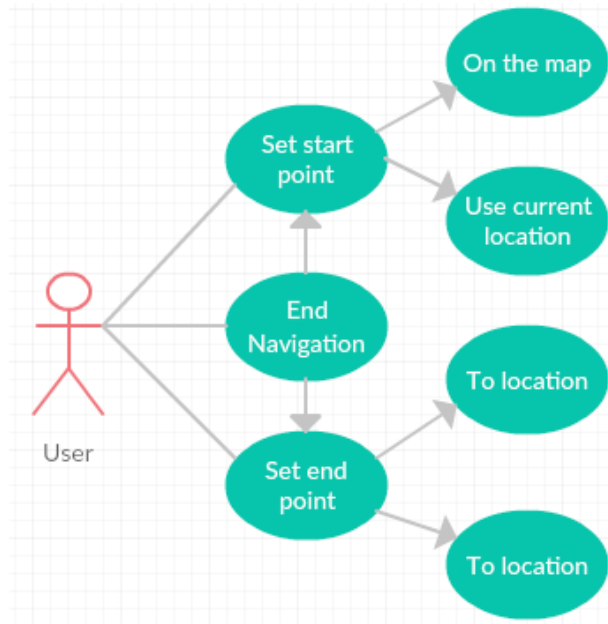
5.3. Use cases Diagram

In this section author provides use cases diagram with description of individual use cases of the application.

5.3.1. Route planning

Figure 5.3.1 illustrates the use cases associated with process of route planning. These use cases are linked to the planner screen.

Figure 5.3.1: Use Case of Route Planning



Source: Author himself

UC01 Set start point-user: sets a start point from the map or use current location.

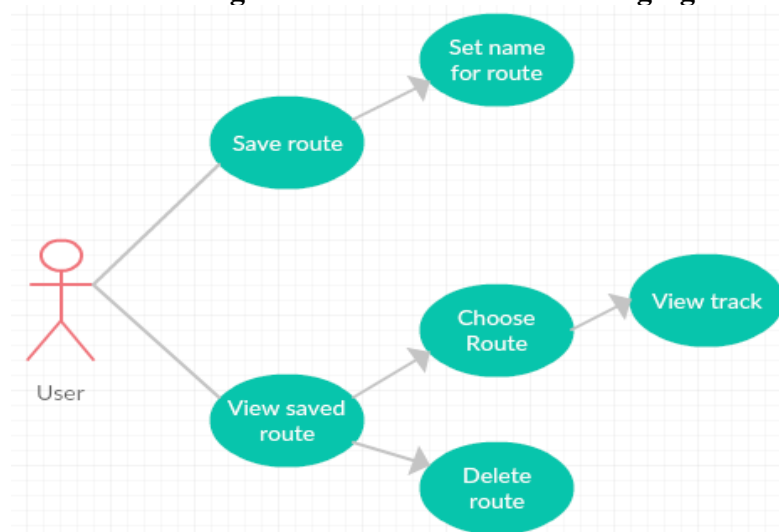
UC02 Set end point-user: set a finish point to type longitude and latitude or user can to touch on map screen.

UC03 End navigation-user: to change start and end navigation.

5.3.2. Managing Route

Figure 5.3.2 describes the use cases associated with place managements.

Figure 5.3.2: Use Case of Managing Route



Source: Author himself

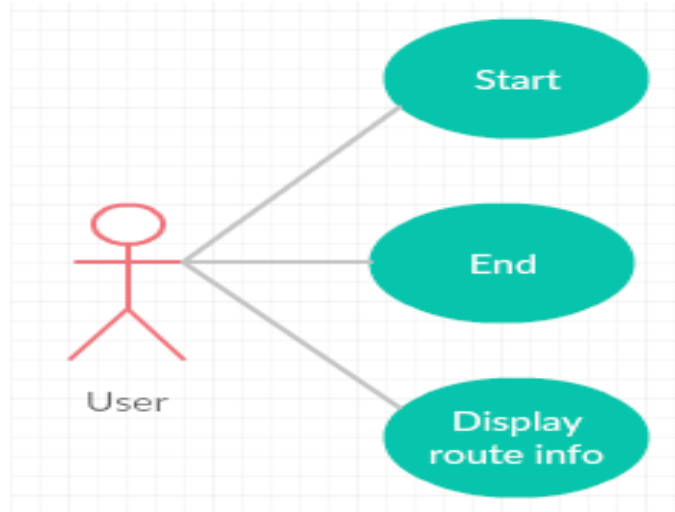
UC04 Save route: User can save the route when user records tracking

UC05 View saved route: User can view saved route and can delete a saved route

5.3.3. Managing Navigation

The use cases associated with navigation management are described in **Figure 5.3.3**. These cases are linked to the navigation screen.

Figure 5.3.3: Use Case of Managing Navigation



Source: Author himself

UC06 Start navigation: User starts the navigation process

UC07 End-User: End navigation process

UC08 Display route info: User review route info

5.4. GUI Design

In this section writer discusses about whole application screen with the user interface design. Prototype was tested by real users. Two people were invited to testing sessions and they tested several times in 3 weeks.

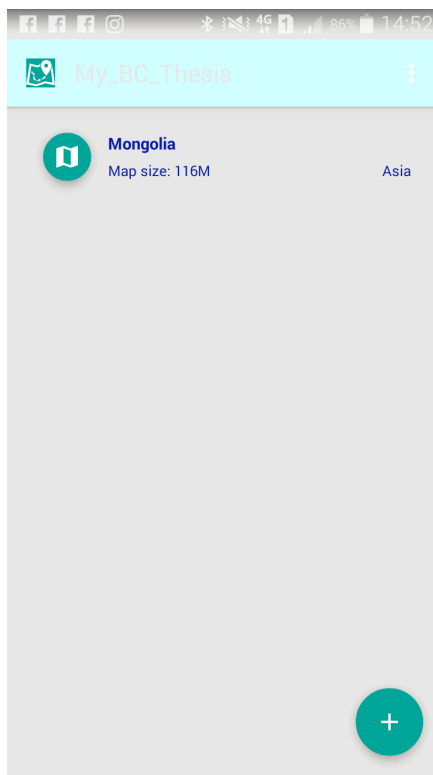
5.4.1. Planner UI Design

The main part of the whole application is planner screen. Firstly, there is a parameter for definition of the downloading a map (**Figure 5.4.1**). User can step to next move when he finishes to download a map. Once you touch the map it is automatically goes to map screen. On the map screen there are activity lists (**Figure 5.4.1**). Secondly there is a parameter for definition of activity lists. The activity lists definition group is visible and image button is on the right corner of bottom. The activity list provides access to other

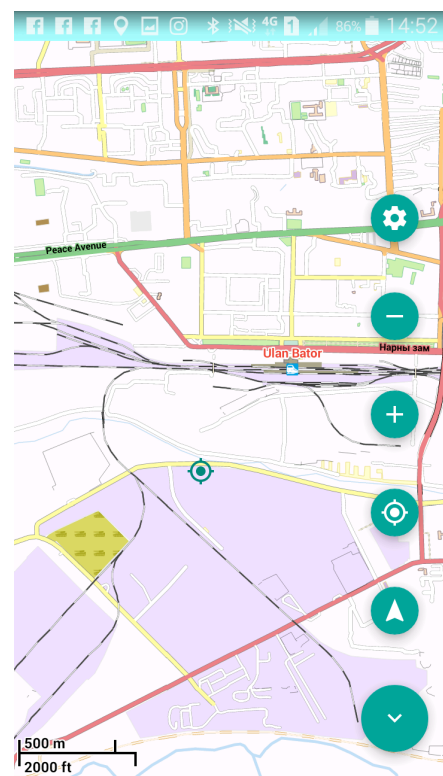
activity lists. These activity lists are Setting, Zoom controls, Find current location and Route planning. When users click the first icon from the activity lists, the Settings is opened and users can make a choice on route of algorithms. The second function of activity list is a Zoom control where users can make zoom control on the map.

When users click the third icon from the activity lists, the Finding location will work. This Navigation application does not automatically find your location. The last icon is a Route planning. Once users click a last icon the Route planning screen opens. The route planning has two choices: first one is a type longitude and latitude or second one is pointing from the map (Google n.d.).

Figure 5.4.1: Navigation Screen User interface



(A) First Screen of Application



(B) Multifunction button with MapView

Source: Author himself

5.4.2. Map screen and Route UI design

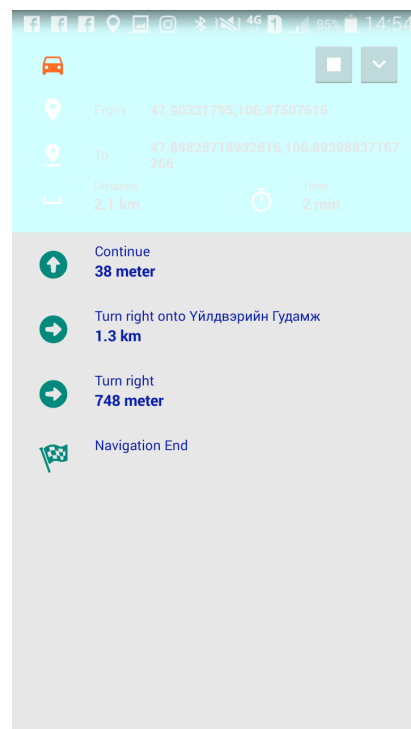
The map screen (**Figure 5.4.2**) shows map with activity lists. When users select a route planning, the screen contains two text fields. The first one is the start point and the second one is a finish point. When users click on start point, they have three options:

- First two are to type latitude and longitude or select on the map to start point

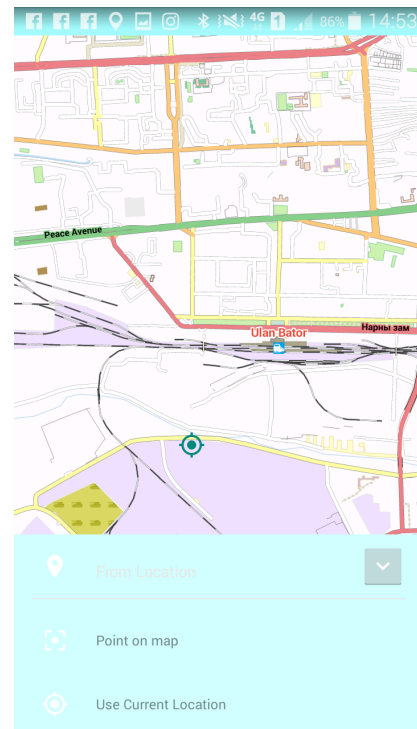
- The last option is use a current location which is the simplest way to use navigation

When users select the finish point, it's similar with the start point. Only one function is reduced from the first one which is to use the current location. Other two functions are same with start point functions which are typing a latitude and longitude and selecting from the map to finish point.

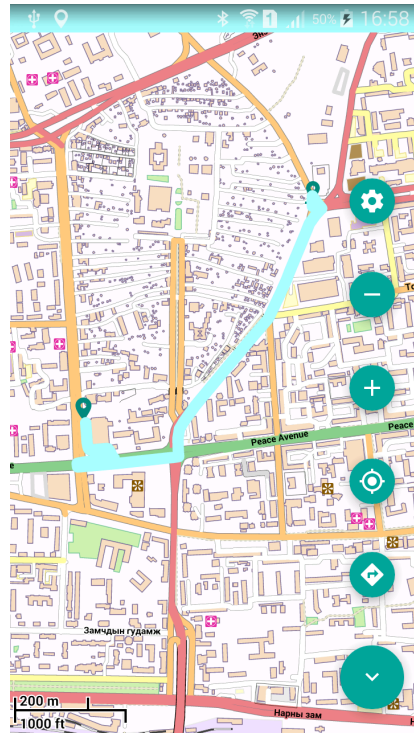
Figure 5.4.2: Navigation Screen User interface



(A)Itinerary



(B)Point Selection



(C) Tour Through the City

Source: Author himself

6. Implementation

In this section author describes the implementation process and structure of the application. The offline GPS application is implemented in Android studio for Android operating system and it is designed for running on mobile devices with Android 4.0 or higher level of android operation system. This project uses OpenStreetMap data, Mapsforge and Graphhopper open source API.

6.1. Component of interface

The Mapsforge (Computer Science Institute n.d.) project was began in 2008 at the computer science institute of Freie Universitat in Berlin and it provides offline vector map library android application. Mapsforge library requires Java programming language thus it simplify the way of API usage. To build map application only few lines code must be written. The main goal of the mapsforge project is to enable free and open toolbox perfectly.

According to OpenStreetMap (OpenStreetMap.org n.d.) following features are APIs including solutions for map rendering, map overlays, downloading maps and more:

- OSM is rendered by on-device
- Powerful and flexible overlay API
- Compact file format for map data storage
- Support multi-touch gestures
- High details (support approximately 200 OSM tags)
- Runs on all devices with android 2.3 and higher
- Small library (around 400 KB)
- Easy to use API and template application
- Free and open source

Graphopper (Github n.d.) route planning uses Dijkstra and A* algorithms¹ by bidirectional variants. Furthermore it allows you to use Contraction hierarchies² which is linked to speed mode. Flexibility mode is on without Contraction hierarchies. But this does not mean that flexibility mode is slow and Graphopper allows you to use both modes together or seperately. Features of Graphopper are:

¹ It was conceived by computer scientist Edsger W.Dijkstra in 1956. It was published 3 years later in 1959.

² Contraction Hierarchies (CH) is a technique to speed up shortest-path routing by first creating precomputed "contracted" versions of the connection graph. Source: https://en.wikipedia.org/wiki/Contraction_hierarchies

- Faster and and lightweighted responses of speed mode
- Runs on any devices such as Linux, IOS, Windows, BSD, Solaris, Raspberry PI, Android, Blackberry and other small devices with Java support
- Intense preparation and time consumption is needed for speed mode
- Pre-defined vehicles can use Graphhopper

The Geocoding is one of the most important processes used in many GPS and it is the process of transforming a location description such as longitude latitude pair of coordinates on the earth surface. In ArcGIS (ArcGIS n.d.) website, it is written that ser can geocode by entering one location description at a time or by providing many of them at once in a table. The resulting location are output as geographic features with attributes which can be used for mapping or spatial analysis.

The overlay API (Android n.d.)implemintation is designed for enriching the existing base map by adding graphical elements as overlays. The map API ensures availability of this implemintation. There are two types of overlays: tile overlays for extensive imagery addition and ground overlays for single image fixation.

The cell phone navigation is simply a turn by turn navigation to do the route planning for the users. Each movements of user is recorded on the navigation and navigation notifies them about every next changes of the destination and possible obstales on the road.

The routing mainly processes route planning between 2 or more nodes graphs. In most of map applications, it is possible to connect more than 3 destinations to link with each other. For this propose, Dijkstra’s algorithm (W.Dijkstras n.d.) is used to find the shortest path between nodes in graph.

6.2.Data Preparation

Very important thing on this application is rendering the base map. Because it is very complicated task to resolve. So author decided to find some library for rendering the base map. The whole renderer is divided into two components.

- First component is the library (the component for rendering the basemap)
- Second component uses the library to draw routes as overlay of the basemap.

Thus author used the Mapsforge library in this project that uses specific file format of the map. It uses custom binary format to be able to deliver good performance³. To be able to use Open Street Map with this library, it has to be transformed into a binary file with this format. Open Street Map provides tools for processing OSM XML files. One of the best tools for this purpose is JOSM. This is the tool for processing OSM XML files written in java that is strongly extensible by plugins. There exists also plugin for creating Mapsforge binary files called Mapsforge Map File Writer⁴. The first thing that needed to be done is converting OSM XML files into maps that can be displayed with MapsForge. Second thing is to use GraphHopper library to create routing data. Then finally author bundled these data to one folder and zipped to *asia-mongolia.ghz* and uploaded to github server.

6.3.Implementation Structure

Application must go through number of packages for the perfect functionalities and following packages are used in this project.

- **Activities**

Each application screens are captured in controller package and UI layouts forms these screens. Location of this layouts can be found in res/layout directory. Following activities and fragments captured in offline GPS navigation:

MapActivity: Mapsforge Fragment class is a layout of this class . It displays map and updating chrome where users can select positions of start point and destination point. It injects and inflates activity map content to map activity context and bring it to front. Also it works with the location listener that helps users to check whether the GPS is enabled or disabled when starting the application. And if it is disabled, application directly opens GPS location settings.

DownloadMapActivity: The DownloadMapActivity class has download process to send request for map download. The class contains methods those call to download, to generate map and to add to the listview.

MapAction: The MapAction activity contains view groups managed by separate layouts in xml file `://map_sidebar_....`

³ . Specification of the binary file can be found in:<https://github.com/mapsforge/mapsforge/blob/master/docs/Specification-Binary-Map-File.md>

⁴ Detailed instructions about installation and usage of this plugin is available in:
<https://github.com/mapsforge/mapsforge/blob/master/docs/Getting-Started-Map-Writer.md>

The model package is divided into five packages which are Database, Type, Listener and Map.

The first model package is database which contains two java classes:

i. DatabasesHelper

It is an adapter which works with sqlite database. This database contains table of places from thesis.db and the table includes following columns:

- ID is the primary identifier of database record
- TimeStamp is value of created or modified of database record.
- Longitude is the longitude value of place location
- Latitude is the latitude value of place location
- Altitude is altitude value of place altitude.

ii. DatabaseTrackingPoints

This defines methods for the working with functions to start, to finish and to state a new place points and places marker as on the map as representation. By this way, users can display or process on tracking that shows their passed road.

The second model package is datatype which contains custom data type classes such as Destination and MyMap. The Destination class stores two important properties including start point and end point. Starting point can be current location of user or custom location and ending point is custom location that generated from MapAction by typing or from MapHandlerListener by clicking on a map view. Additionally, MyMap java class stores properties such as country, size of map, download link, continent, map name and status. The purpose of this class is to work with locally saved map data.

The third model package is Map and it has has nine classes: DownloadFiles, MapDownloader, MapUnzip, MapHandler, MyDownloadAdapter, MyMapAdapter, OnDownloading, Navigator and Tracking. DownloadFiles is the main class of downloading map from remote server. Author had put map data to remote server and by this way, he could upgrade or add new country map data to server and sync with the GPS application without updating the application itself. Because it downloads map data from MapDownloader and unzips map data from MapUnzip. MyMapAdapter holds lists of all available maps. MyDownloadAdapter holds information of downloading process of map. OnDownloading class displays progress bar of the downloading process. Navigator class is bridge between GraphHopper and MapAction. It starts and stops navigation and

calculates route distance and shows direction information. Tracking class saves travelled locations to file and shows those in Analytics screen.

The fourth model package is listeners which helps action listeners to communicate between classes and modals. Author have implemented MapDownloadListener, MapFABonClickListener, MapHandlerListener, NavigatorListener, OnDownloadingListener and TrackingListener. MapDownloadingListener has actions such as downloading start, process update and ending. DownloadFile and MapDownloader classes fires the all events and DownloadMapActivity listens events and displays the map downloading proces. (GitHub n.d.)

- **Navigation**

The navigation is an essential part of the this application. When user starts navigation process, the startLocationUpdate method from the navigatorManager class is executed. So the Manager begins to request location updates every second. The location receiver obtains updates and manages in the receive method. The locationReceive is used in the NavigationFragment where the receiver locations are handled.

- **Graphics**

All resources of the graphic icons are located in res/drawable-xx and the most of logos are downloaded from IconLand.

- **Geocoding**

In the application both geocoding and reverse geocoding are used. The geocoding is used when users are setting a place by the next input.

6.4. Limits of Application

In this project, offline GPS application for Android is developed and current executed application is a base offline GPS application. Further developments can be done and at this moment, following limitations are present:

Map data: The OpenstreetMaps are used for displaying the map in this application and Geoaddress is not fully written in this map. Thus it is the major problem because when users write the address and if it does not contained on the map then users have to select near location from the map.

Route planning: Users could select types of surface that they wanted to include in route calculation or they could define route planning regarding to their preference. But the application is not able to calculate time to destination and distance of kilometer. Also

user has to refresh current location because navigation camera is not following their location.

Trip planning: Application does not have a trip planner which means application does not show connection between three or more points on the map.

6.5. Application testing

The final executed application tested by 3 users. The participants of the test were students of Technical university of Mongolia who have experience with android application. Objective of the session was to test usability of the application and find possible defect in the interface design. The participants were provided with a list of tasks. The tasks were designed to test almost every function on the application. The application was installed in Samsung E7000, Samsung Galaxy S6 and HTC with android 4.4 kit kat operating system. Below author summarizes results of expected executions and executions from participants. At the start of session the navigation drawer was displayed to simulate first launch of the application.

Task 1 Users to start application

Expected execution: clicking on the download map or choose map.

Correctly executed by: 3 participants

Task 2 users clicked on the activity list.

Expected execution: setting is the first activity of the activity list

Correctly executed by: 3 participants

Expected execution: zoom control is a second activity of activity list

Correctly executed by: 3 participants

Expected execution: set a start and finish point

Correctly executed by: 3 participants

Task 3 user wants get current location

Expected execution: find current location

Correctly executed by 3 participants

Task 4 User want to get direction

Expected execution set a start point

Correctly executed by 3 participants

Task 5 User wants to use current location set the start point

Expected execution: to use current location on the start point

Correctly executed by 3 participants

Task 6 User wants to find to get a direction

Expected execution: set end point type latitude and longitude or select on the map

Correctly executed by 3 participants

Task 7 user wants to change direction

Expected execution: end previous navigation then set new direction on the start and finish point

Correctly executed by 3 participants

7. Conclusion

In this thesis, we have developed offline GPS application for Android in Mongolia. For this purpose, Android studio is employed and Java programming is executed. The application is intended to be used only in Mongolia.

Firstly, we have surveyed OpenStreetMap data and its structure. The existing libraries were searched that can use OpenStreetMap data in offline mode and render them. MapsForge was is chosen library for map rendering and it was capable of planning road trips. For routing purpose GraphHopper was employed. After completing the design of application, application was tested and evaluated by users. Users completed 7 tasks to check the perfect functionality of the application and all of the tests were executed 100 percents.

This implementation of the application has number of limitations. The biggest limitation for the ordinary user is that application is unable to type address name which means users must mark on the map or to type the geocoordination of the place. Secondly, the application is not able to calculate time to destination and distance of kilometer. Also user has to refresh current location because navigation camera is not following their location. The last limitation is that application does not show connection between three or more points on the map.

For further research, this developed application can overcome these limitations to have more functions in the application. Unfortunately, designing and implementing those functionalities is too much time demanding in this bachelor thesis.

8. Bibliography

- Android. <https://source.android.com/devices/tech/dalvik/>.
- Android Developer. June 2011. <http://developer.android.com/guide/basics/what-is-android.html>.
- —. <http://developer.android.com/index.html>.
- —. <https://developer.android.com/guide/components/index.html>
- —. <https://developer.android.com/guide/components/fundamentals.html>.
- Android, Developer. <https://developers.google.com/maps/documentation/android-api/tileoverlay>.
- ArcGIS, Server. *SOAP SDK*.
http://help.arcgis.com/en/arcgisserver/10.0/apis/soap/SOAP_Geocode_Overview.htm.
- Carlo U, Nicola. „Android Developer Guide.“
<http://web.fhnw.ch/plattformen/androembsy/content/vorlesungsunterlagen/internet/mlrii.pdf>.
- Computer Science Institute. Freie Universitat Berlin.
<https://spline.de/static/talks/mapsforge.pdf>.
- Cooksey, Diana. *Understanding the Global Positioning System (GPS)*. Land Resources and Environmental Sciences, Montana State University-Bozeman.
- Editor Java OpenStreetMap. „Basic Manual.“
<https://ma.ellak.gr/documents/2014/09/%CF%84%CE%BF-%CE%B5%CE%B3%CF%87%CE%B5%CE%B9%CF%81%CE%AF%CE%B4%CE%B9%CE%BF-%CF%84%CE%BF%CF%85-josm.pdf>.
- G Kruger, R Springer, W Lechner. „Computer and Electronics in Agriculture.“ *Global navigation satellite system (GNSS) 11.1* (Oct 1994): 3-21.
- Github. 2015. <https://github.com/mapsforge/mapsforge/blob/master/docs/Specification-Binary-Map-File.md>.
- GitHub. <https://github.com/mapsforge/mapsforge/blob/master/docs/Getting-Started-Map-Writer.md>.
- Github, Graphhopper. <https://github.com/graphhopper/graphhopper>.
- Google. <http://www.android-graphview.org/>.
- Google Code. June 2012. <https://code.google.com/archive/p/osmbonuspack/>.
- GraphHopper. <https://graphhopper.com/>.
- Griffin, Darren. „How does the Global Positioning System work?“ June 2011.

- Kornhauser, Alain L. „Global Navigation Satellite System (GNSS).“ Operations Research and Financial Engineering, Princeton University.
- Lee, Wei-Meng. *Beginning Android 4 Application Development*. Indiana, 2012.
- Legat, Vaclav. „Mobile journey planner and navigation for cyclists.“ (Czech Technical University in Prague) Feb 2014.
- National Academy of Engineering. *Global Navigation satellite system*. Editor: Per K. Enge, and Grace X. Gao Lance A. Davis. Washington, DC: The National Academies Press, 2012.
- National statistics office. <http://www.en.nso.mn/index.php>.
- Mikayelyan, Eduard. „Developing Android Mobile Map Application with Standard Navigation Tools for Pedestrians.“ *Master thesis* (Department of Physical Geography and Ecosystem Sciences, Lund University), 2011.
- Mobile Atlas Creator. <http://mobac.sourceforge.net/>.
- OpenCV.org. 2016.
http://docs.opencv.org/2.4/doc/tutorials/introduction/android_binary_package/android_dev_intro.html.
- OpenSource. <http://mobac.sourceforge.net/>.
- OpenStreetMap.org. <http://wiki.openstreetmap.org/wiki/Mapsforge>.
- Poole, Ian. „Radia Electronics.“ http://www.radio-electronics.com/info/cellulartelecomms/location_services/assisted_gps.php.
- Portal, Mongolia. <https://en.wikipedia.org/wiki/Portal:Mongolia>.
- Server, ArcGIS. *SOAP SDK*.
http://help.arcgis.com/en/arcgisserver/10.0/apis/soap/SOAP_Geocode_Overview.htm.
- Shengyue Ji, Wu Chen, Xiaoli Ding, Yongqi Chen. „The Journal of Navigation.“ *Potential Benefits of GPS/Glonass/Galileo integration in an Urban Canyon*, 2010.
- Shokouh, Jalil. „GPS Signal for Android.“ Editor: Jalil Shokouh. *Detecting GNSS attack on smartphone* (Shokouh, Jalil), July 2013.
- State of Governance. http://www.mfa.gov.mn/?page_id=26263&lang=en.
- W.Dijkstras, Edsger. <http://algorithmsandme.in/2014/07/graphs-dijkstras-algorithm-to-find-shortest-path/>.
- William H, Guier and George C, Weiffenbach. *Genesis of Satellite Navigation*. Sv. 18, v *John Hopkins APL Technical Digest*. 1997.