

“SMART VEHICLE NAVIGATION SYSTEM”

A PROJECT

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the supervision of

Dr. Pradeep Kumar Gupta

(Associate Professor)

By

Dhairya Agarwal (151383)

to



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT, SOLAN – 173234

HIMACHAL PRADESH, INDIA

CERTIFICATE

I hereby declare that the work presented in this report entitled “**Smart Vehicle Navigatin System**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2018 to December 2018 under the supervision of **Dr. Pradeep Kumar Gupta**.

The work done embodied in the report has not been appeased for the award of any other degree or diploma.

(Student Signature)

Dhairya Agarwal
(151383)

This is to certify that the above affirmation made by the candidate is true to the best of my knowledge.

Dr. Pradeep Kumar Gupta
Associate Professor
Department of Computer Science and Engineering and Information Technology

Dated:

ACKNOWLEDGEMENT

I would like to take the opportunity to thank and express my deep sense of gratitude to my mentor and project guide **Dr Pradeep Kumar Gupta** for his immense support and valuable guidance without which it would not have been possible to reach at this stage of our final year project.

I am also obliged to all my faculty members for their valuable support in their respective fields which helped me in reaching at this stage of my project. My thanks and appreciations also go to my colleagues who have helped me out with their abilities in developing the project.

Date:

Dhairya Agarwal

TABLE OF CONTENTS

1. INTRODUCTION.....	02-06
1.1 Introduction.....	02
1.2 Problem Statement.....	03
1.3 Objectives.....	04
1.4 Methodology.....	04-06
1.4.1 Global Positioning System	
1.4.2 Project OSRM Enhanced Graph Model	
2. LITERATURE SURVEY.....	06-17
3. SYSTEM DEVELOPMENT.....	18-36
3.1 System Requirements.....	18-21
3.1.1 Android as an Operating System	
3.1.2 GPS Based Location Tracker for accessing location	
3.1.3 MapBox	
3.2 Implemented Libraries/Model/Algorithm Used.....	22-28
3.2.1 General Options provided by OSRM	
3.2.2 Significant interfaces objects of OSRM	
3.3 Implementation	28-36
3.3.1 Accessing Map from Maps SDK for Android	
3.3.2 Setup permissions	
3.3.3 Add a map	
3.3.4 Lifecycle methods	
3.3.5 User Location	
3.3.6 Route generation	
3.3.7 Route progress	
4.PERFORMANCE ANALYSIS.....	37-44
4.1 Algorithm efficiency.....	37-40
4.1.1 The Simple Model and Path Search	
4.1.2 Our Detailed Model	

4.2 Implemented result screenshots of the system.....	40-44
4.2.1 User location	
4.2.2 Calculate and draw route	
4.2.3 Route progress	
5.CONCLUSIONS.....	45
5.1 Conclusions	
6.RFERENCES.....	46-47

LIST OF FIGURE

Fig1: A workflow for background map design.....	14
Fig2: Request functionality of OSRM.....	22
Fig3: Response functionalities OSRM.....	23
Fig4: Access token for the project.....	30
Fig5: Lifecycle method.....	32
Fig6: Accessing location component.....	33
Fig7: Route Generation method.....	36
Fig8: A typical directing scenario directing a turn limitation.....	37
Fig9: Edge prolonged model.....	38
Fig10: Optimized path for bike through San Francisco’s dodging turns plus high speed traffic.....	39
Fig11: User Location showing JUIT.....	41
Fig12: Route generation from JUIT to Chandigarh.....	42
Fig13: Route generation from JUIT Kandaghat.....	42
Fig14: Navigation started from JUIT.....	43
Fig15: Turn By Turn Navigation in JUIT.....	43
Fig16: Navigation proceeded ahead.....	43
Fig17: 12 min Remaining to reach the destination.....	43
Fig18: Destination had arrived.....	44
Fig19: Application Notification while the App is temporary close.....	44

LIST OF TABLE

Table1 : Options provided by Nearest Service.....	23
Table2 : Option provided by Route service.....	24-25
Table3 : Option provided by Match service.....	26
Table4 : Route progress API.....	34-35

ABSTRACT

If we account for more than half of currently used hand held systems, Android OS, is providing its users with pronounced chance to innovates and to implements things in mobile devices. Initial as a mobile OS, the array of devices is compatible with Android OS is even motivating the market in the course of PC knowledge. Because of which the need for compactness have increased by hops and bounds. Users had already started implementing applications for each and every need. The major part of the project includes Mobile App Developments of the GPS based navigation system using an Algorithm based on OSRM in which with the help of device; any GPS permitted handsets could be located. Although these targeted user, which are situated in remote locations or the crowded locations, he/she must always require a network connectivity and GPS enabled system. Primarily, these apps are implemented not for android systems only, but these applications can be run on cross platforms with specific devices supports in terms of google map and iOS map Services. These apps are free and currently online.

For prediction of routes and exact assurance of navigation paths ahead of time, calculations have been utilized depending on the OSRM model. Various SDK's and libraries have also been taken from MapBox Map SDK, where the Direction Api helps in navigation of the vehicle.

CHAPTER 1: INTRODUCTION

The smart vehicle navigation system is basically the satellite base system. The navigation systems make the use of many different technology and parameter. Vehicle manufacturer, Navigation device manufacturer and Map data maker work together to develop best navigation systems that are precise and accurate in term of directions and shortest path providing, in coordination with the geographical positioning system of streets and places. A typical navigation device has a GPS embedded with in itself, visual display units, audio units, computer processor with the specified algorithm for various functionalities and a maps database. Other feature like, traffic density calculators, language preferences, including various point of interest like hotel, pool, petrol pump etc and other interesting and operating feature like speed alerts, live views etc can also be incorporated if the availability and cost is not a problem for you.

The Navigation system is a structure that aid in navigation, route determination, etc. These systems might be fully on board of a vehicles, or these systems might be placed somewhere else and connect mobile location signals within a vehicles or vessel, or they might use a combination of these methods. In this project, I had implemented the project for mobile application or an Android Application with the help of various technologies and models.

1.1 Introduction

One of the most basic use of maps are to find out where one are located and the destination locations. After this, onr usually want to use maps to find the best path to another locations.

On a digital maps, this is usually provided with the help of turn by turn navigation using navigation device embaded with GPS services Turn by turn Navigation are one of the basic features of some GPS navigation devices where directions for a chosen route are continuously introduced to the users in the form of written or visual guidance. These application keep the user up to date about the shortest route to the destinations, and are often informed accordingly to changing factor such as traffic and street conditions. Turn-by-turn navigation systems generally use an electronic voice to direct the user of the application which direction to choose whether to turn left or right, the route or street names, and how much distance to turn. Mathematically, turn by turn navigation system are based on the shortest path algorithm within itself, which examine how to identify the best path that meets some criteria

(shortest, best, cheapest, fastest, etc.) between two locations in a huge network. The open source routing machine algorithm or OSRM is providing our forthcoming Direction API which is implemented as the superior and a complete graphed models that provide many feature beyond the basic important topologies.

Navigation system is capable of:

- providing maps, that might be displayed to user in human understandable and clear format in the form of texts or in graphical formats.
- determining these vehicles or vessel's locations through sensor, map, or data with the help of external sources.
- providing various suggested directions to the user in charge of a vehicle's or vessel's via text or speech.
- providing route directly to an autonomous vehicles such as a robotic probed data or guided new user to that locality.
- providing data on nearby vehicle's or vessel's, or other hazards or obstacle.
- providing analysed and predicted data on traffic condition and suggesting alternate directions for the destination.

1.2 Problem Statement

The objective of this proposal is to keep track the vehicle's location by installing GPS system. We need to develop a computer software and a smart phone application. The computer software will get the information from GPS and helps the vehicle to control room to keep track of location of the vehicle and monitor the speed of vehicle. The smart phone application helps the students and faculty members which uses the navigation service to get information about the real time location of traffic arriving at during their travelling. Nowadays, many public feel discomfort to find for whatever they all require from a search machine by typing arguments, particularly when they do not knew the keywords that they should be writting. For instance, a traveller facing to find the preferable direction to a location, but he/she would like to know more about the route such as its traffic, point of interests, road conditions, suitable path during their traveling and etc. This kind of situations, the user's would prefers an progressive searching purpose formulated with advanced models for these, which is capable to search openly with database and yield the related data.

In account, there exists many progressive navigation systems but most are incapable to deliver

routes exactly as well as data of situation they may face during their traveling near the area such as property, supermarket, mall, hospital, etc. where the traffic may exist.

Therefore, an educational, dependable and exact directive system is very significant in this technical era. This directive system must be able to explore the preferable information with the appropriate and consistence application that stores this information and return the computed path to destinations. The directive system must be user welcoming and able to develop data proficiently.

1.3 Objectives

The general objective of this project is to provide the smart navigation system implementing a routing algorithm which is designed to provide advanced information and provide wide facilities for not only providing efficient and best route but also the is used with the project of MapBox which are providing with map sdk's and various other map facilities like markers, tools for customization of maps, etc.

The specific objective of the project is: -

- To change a mobile application that is capable to capture real time traffic congestion and deliver it to server's along with coordination details.
- To develop any system that can uses Open Source Routing Machine Model for providing turn by turn directions efficiently.
- To develop any system that can provide route information precisely to the client side.
- To develop any system which is able to navigate user's to the certain destination.

1.4 Methodology

1.4.1 Global Positioning System

GPS navigation device, GPS receiver, or simply GPS are the devices that has the ability to receive data from GPS satellites and then to calculate the system's geographical positions. Using appropriate software's, the device is capable of displaying the location on a map, and they may offer direction.

Using the GPS compiled data and subject to sophistication of the installed GPS software's, a GPS device are used as an vehicle navigation system that may be used in a number of context, including:

- map, includes street maps, displayed in human understandable format via texts or in a graphical format,
- turn-by-turn navigation system directs the human driving the vehicle or vessel via texts or speeches,
- Direction that fed directly to an autonomous vehicles such as a robotic probes,
- traffic congestion of maps (depicting either historically or real-time data) and suggests alternative direction,
- information on nearby devices such as restaurant, petrol station, and tourist attraction.
- GPS devices may have the ability to indicate:
 - the streets or route available,
 - traffic congestions and alternative route,
 - streets or routes that might be taken to reach to the destinations,
 - if some paths are congested (now or historically) then what is the best route to take,
 - The location of food stalls, banks, motels, fuel, airport or other place of interests,
 - the shortest path between the two location,
 - the different option to drive on highway or back road.

1.4.2 Project OSRM Enhanced Graph Model

Offering effective direction and best route for pedestrian, cyclist or driver is substantially much simpler way of finding the best connections in street networks. This open routing machine, OSRM is powering an upcoming Direction API, which is implemented as an enhance and detail graphed model that provides many features beyond the basic topology abilities. With OSRM, one not only get the best shortest or fastest route, but can also takes into account turn restriction and weighted cost like waiting at traffic signals, applying brakes and accelerating at sharp turn, or steep hill for riders. OSRM uses contraction hierarchies internally. From my understanding, this technique is way superior to Dijkstra's algorithm which is considered as the shortest path algorithm when it comes to calculating the distance between two locations in real world data.

Basic algorithm for route finding is Dijkstra's algo, which solve the difficulty of finding best path, or the route with a minimum weight. Conceptually, it works like: to find the path from A to B, start from the basis point A, and in each iterations, find and settle the closest unsettled intersections. Settling mean assigning a final path and distance. And this is the best or the

shortest intersections to any previously settled one which are still not settled itself. Intersection adjacent to settled one are assigned to a tentative distance's that might decrease if a shorter distance or a better path is found during the search. In this sense, the procedure searches outward round the source point until it finds the destination. This looks like a decent model to find a path between A to D, but when we see from a closer look, it has some major shortcomings. It does not account for turn restriction, a very common occurrence in the street system.

CHAPTER 2: LITERATURE SURVEY

Paper1: “Smart Vehicle Navigation System Using Hidden Markov Model and RFID Technology” By 1. Reza Malekian •2. A. F. Kavishe •3. B. T. Maharaj •4. P. K. Gupta •5. G. Singh •6. H. Waschefort

Published online: 16th -6- 2016 Springer Science and Business Media in New York 2016

The street transportation of difficult good had remained the viewed as one of major for study with increasing frequency in coming years. Global positioning system is the base for vehicle location devices, which are used to track vehicles in movement. However, this tracking application suffers - inaccuracy and other limitations. In addition, the real-time tracking of transport through areas being shielded from GPS satellite are difficult. In this research paper, author had addressed the development of a smarter vehicle navigation system which is capable of using triangular radio frequency identifications based on data about various navigation paths. For prediction of these paths and accurate determination of navigation route in advance, predictive algo had been used based with the Hidden Markovs model. At the base of the system there exists field, programmable gate arrays board and hardware for the collection of navigation info. A message protocols and a file to store driver habit and other data have begun to be designed. From experimental result obtained, an accurated navigation routes prediction is consistently and accurately attained by the designed system. In addition, the once-off troubles of the driver’s habits had been cleaned out positively.

Model Implemented (Hidden Markov Model)

It was realized that in this HMM the grouping of watched yield esteems gives data about the succession of states. Whenever demonstrated is utilized, a HMM at that point is eyewitness which will just watch a succession of yield tokens specifically. The hidden states and the state change and discharge probabilities are considered earlier learning. The use of the HMM at the prescient calculation has been utilized in biotechnical fields to ponder protein structures and hereditary qualities. Applying HMM to a vehicle route framework requires just that the route way be spoken to as a Markov procedure too.

Vehicle Navigation Path Prediction in this paper

Vehicle route ways are generally tedious in nature because of characteristic requirements that limit the opportunity of the driver. A standout amongst the most widely recognized

characteristic requirements is time; most drivers endeavor to diminish the measure of time spent going between their cause and goal. Anyway the possible route way that a vehicle driver settles on is affected by new imperatives from the street system and condition when all is said in done. Cruel climate conditions, poor activity conditions and the accessibility of fuel will all factor into the basic leadership of the driver and influence the inevitable route way. The proposed framework will gather driver propensities over some stretch of time and, utilizing this information, play out a static expectation on a vehicle's route way. The forecast is viewed as static, as it won't be refreshed as the vehicle is being driven along the route way. The conceivable route way will be demonstrated as a HMM whose parameters are gotten from the driver's current driving practices.

Main objective of this literature survey work is to implement a framework that fit for foreseeing a vehicle's path ways, which is depended on the current driving practices of drivers by utilizing a HMM deduction calculation. These current way of driving practices are gathered by an equipment detecting stage and RFID technologies.⁶The programming and the calculations used in this experiment work all turned out great. Actual calculations were thoroughly tried and the whole forecast programming suite was subjected to unit tests to keep changes after periodic time from influencing its center usefulness. Furthermore, the UART correspondence module was all around structured.

No planning issue or information accuracy issues happened through the execution of work. The Viterbi calculation, which is a deduction calculation for the HMM, was actualized utilizing Python programming dialect. This calculation gave the doubtlessly arrangement of states that a Markov procedure were experienced enough to give the succession of perceptions. Correspondence conventions between the sensors and FPGA, and also the FPGA and a PC, were planned. A RFID hardware's and label set was utilized to accumulate topographical information and additionally trigger the social affair of ecological information by means of the sequential association being crossed over through the PC.

Paper 2: “Dijkstra Shortest Path Algorithm Using the Global Positioning System(GPS)”

By Pooja Singal , R.S.Chhillar

International Journal Computer Application (0975 – 8887) Volume 101– No.6th, September 2014

GPS uses the position parameter that included the Dijkstra's calculation as gps is utilized to locate the accurate way starting with one node then onto the next node. Dijkstra's shortest path

calculation is otherwise called a most precise and accurate way of calculation between nodes. It vales is only measured on positive weights. In this paper, Global Positioning System (GPS) is used for including another necessary Dijkstra's calculation. From this current position is recovered anytime. By utilizing this current position, the separation can be resolved starting with one node then onto the next node.

Dijkstra's calculation is designed by Dutch PC researcher Edsger Dijkstra in 1956 and distributed in 1959, Its diagram is depended on seeking calculation, as it fathoms the single source most limited way issue. It is connected just on positive weights charts and not negative diagrams or the negative values. This calculation is frequently utilized in steering. Dijkstra's calculation is utilized for finding the shortest path among the variour nodes with least expense. We will utilize it in most calculation for finding shortest paths.

“For example: - Let the vertices with in a graph be the cities and the edges which link these vertices are driving distance from one city to another city. Dijkstra’s Shortest path algorithm is used to find out the shortest route from one city to another city with minimum cost. It solves only the problems with nonnegative costs, i.e., $C(m,n) \geq 0$ for all (m, n) belongs to E , Where C is the cost & E is the edges for a graph.

Dijkstra's algo is used to find the most constrained route in a diagram. GPS is used in Dijkstra's algo to get current position of each center. Partition is also decided from this position. A count is proposed for this. Regardless, simply speculative thought is given, helpful execution isn't given in this paper. Thusly, in future you can complete this unemployable paper practically. This thought of GPS is moreover used on others briefest way estimations like A^* , Warshals figuring and Bellman Ford count, etc.

Paper 3: “GPS Based Vehicle Navigation System Using Google Map”

By “1. Ch. ChakradharaRao, Department of CSE, GMRIT, RAJAM Andhra Pradesh, India

2. P.Pushpalatha , Department of CSE, GMRIT, RAJAM Andhra Pradesh, India

3. N.AdityaSundar Department of CSE, ANITS, Visakhapatnam Andhra Pradesh, India

Global positioning System or GPS is being use for wide variety of application. In this paper, the principal focuse is on building an upgraded version of GPS based vehicle navigational system or route framework that is utilizing Google Maps. Presently, the most used principle issues as for open transportation is the management of heavyor big transports. The management of these involves a considerable measure of issues to the specialists. Moreover, while managing

the administration of open transport, one ought to likewise remember the expense of activity. This paper gives a possible, efficient and minimal effort answer for the above issue. This framework is to find the courses in which the buses are transporting and travelling and reviewing the current location of vehicles on a displayed outline of the map that helps in following the bus conveniently and numerous other applications like speed screen, burglary screen and so on.

We have followed a two wheeler with the help of this module and furthermore in coordination with the Google maps which showed the area, speed, date, time and accuracy. Alternate applications included are speed observing, robber checking and armada the executives. It very well may be utilized in any vehicle which helps in following the vehicle in which the module is set. This likewise encourages us in consistently screen the vehicle. This whole framework can be executed in a transport station which can enable transport to transport experts to persistently screen the transports. A similar idea can be actualized in a mechanized vehicle by utilizing CAN (Controller Area System) transport for getting more refined applications like programmed speed control, course mapping and so forth.

The application highlights both visual and sound-related signs, by presenting a selection of courses to the desired paths. After the client has picked their favored course, the framework guides them from their present location to the nearest fitting and nearest bus stop. Once there, it asks them to what extent the user will need to hold up before their bus arrives – apparently, once the time is close by, they may at present need some help in confirming that they're boarding the correct transport.

In the wake of getting on the transport, the framework will at that point illuminate them what number of stops are left before theirs, and afterward caution them when it's time to make them aware of their destination. After getting off the vehicle, the application will continue giving directions, directing the client to their destination.

Paper 4: “Smart and real-time traffic congestions estimation and clustering technique for urban vehicular road”

By:1. Vishwajeet Pattanaik 2. Vishwajeet Pattanaik,3. Mayank Singh Computer Science and Engineering Deptt., Krishna Engineering College Ghaziabad, India 201 007,4. P.K. Gupta Department of Electrical, Electronic, and Computer Eng University of Pretoria, South Africa, 0028,5. S.K. Singh Department of Computer Science and Eng Indian Institute of Technology BHU, Varanasi, 221005

Street traffic blockage is turning into a genuine worry in developing countries and effects the economy of the countries gravely. Expanding traffic and congestion on urban streets shows a certified risk to the financial development and livability of city locales. Most traffic clogs are caused because of unplanned and impromptu street systems, high volumes of vehicles and nearness of basic clog regions. Traffic clogs represent a risk to the economy as well as to nature. Overflow impact from blocked fundamental streets to auxiliary streets and side boulevards as elective courses frequently prompts more blockage; expanding the odds of crashes and mishaps because of tight separating and consistent halting and-going. The accompanying paper exhibits a savvy clog shirking procedure by assessing the degree of continuous traffic blockage on urban street arranges and predicts another most limited course to the goal. The proposed framework utilizes K-Means Clustering Algorithm to appraise the size of clog on various streets and after that utilizes Dijkstra's Algorithm to anticipate the briefest course. When the client inputs the goal into the framework, the framework predicts the briefest course from the client's present area. The procedure is emphasized at each crossing point until client achieves the goal.

The proposed procedure was tried more than a few street maps of New Delhi. The results obviously demonstrate radical decrease in movement time, contrasted with way discovering strategies utilizing ordinary strategies. Executing the proposed framework in Navigation frameworks could lessen financial what's more, natural misfortunes caused because of blockage. Moreover, since the framework depends on ongoing traffic and powerfully predicts courses as driver's advancement towards their goal, there is negligible possibility that individuals beginning from the source to a similar goal could ever cause clog for different drivers. The proposed framework is planned as a proof of idea, henceforth it depends on basic calculations like: k-implies grouping, raised frame and Dijkstra's Algorithm. The framework has high time-space unpredictability, which could be decreased altogether by utilization of most recent bunching and way discovering calculation. Moreover, executing the framework over Aggravated Computing Systems could improve the framework's execution. The proposed framework is only a stage towards radicalizing cheaper yet solid traffic clog recognition and avoiding frameworks, helping spare economy, condition, and health.

Paper 5: “POSITIONING AND NAVIGATION SYSTEM USING GPS SYSTEM”

By 1. J. Parthasarathy Member Technical Staff, Sun Microsystems Pvt ltd, India,

2. Divyasree chambers, Off-Langford road, Bangalore-560027, India.

In this paper, a portion of the thoughts of position and route utilizing GPS were investigated, GPS is a satellite-based route framework made up of a system of 24 satellites put into space by the U.S. Division of Defense. This paper gives the utilization of a handheld GPS recipient in the territories of exact situating, mapping areas, exploring over the mapped areas all around effectively. The reason for this paper is to grand the encounters that brought about in structuring a positioning and route, which can be utilized as a moving compass, directing to any mapped goal, giving the data about close by spots, vacation spots, oil bunks and so forth. The Magellan 310 handheld GPS which is being utilized for building up the proposed framework enables clients to interface with a PC through RS-232 Serial Interface and the convention utilized by the gadget for correspondence is NMEA 0183. This paper gives a contextual investigation during the time spent structuring such a framework and its restrictions in the time where through separation estimations from a couple of more satellites, the beneficiary can decide the client's position and show it on the unit's electronic guide. It additionally depicts the usage subtleties and their outcomes acquired what's more, along their upsides and downsides when contrasted and the other existing situating and route framework intended for essential handheld GPS (without having advance highlights like arrangement for electronic guide and so forth in the gadget), It likewise addresses a portion of the upgrades that can be cultivated in the structured framework. This paper is composed by giving a concise prologue to GPS innovation and after that managing equipment portrayal and NMEA Messages alongside a contextual analysis usage detail of the framework, its highlights and potential upgrades in the framework.

This framework created is a small scale of what should be possible with the utilization of GPS innovation, since expense being an imperative for further progression, it gives the clients a wide range of applications by utilizing an essential GPS beneficiary which is costing less. In this way the actualized framework can clear path for creating different frameworks utilizing top of the line GPS beneficiaries with parcel of new highlights and progressions, with the goal that applications, for example, vehicle following, vacationer direct programming and so on can be produced for the clients.

Paper 6: “An Efficient Indoor Location Tracking System and Navigation System Using Simple Magnetic Maps Matching”

By 1. Binu P K,2. Akhil krishnan R,3. Pranavkumar A Department of Computer Science and Applications, Amrita School of Engineering,4. Amritapuri Amrita Viswa Vidyapeetham, Amrita University, India {binupk78@gmail.com, akhilkrishnan555@gmail.com}

With the design of GPS and related advances, open air route is essentially conceivable with incredible exactness these days. In any case, we are still needing a productive, dependable and less expensive innovation for indoor route. In this paper we put forward the possibility of an android based application which empowers proficient indoor route and tracks the accurate area of individuals inside expansive structures. In spite of the fact that unique advancements like Wi-Fi, Bluetooth, RFID etc., are accessible for indoor route and area following, the fundamental issue is that the usage and upkeep costs are high. Likewise, the change in climatic condition influences the working and debases its execution. In this work, we consolidate the creative thought of attractive field distinguishing proof innovation which utilizes the normally happening attractive fields inside the solid structures with streamlined attractive guide coordinating calculation and the thought from Route fighter calculation for successful indoor situating and route. A Smart telephone with attractive field sensor and accelerometer empowers the application to give the best indoor route office to the end clients.

In this work, we propose a viable android based application for following people at indoor areas by exploiting earth's attractive field. Our framework requires no extra framework or usage cost. All we need is an advanced mobile phone with inbuilt magnetometer and gyration sensors. The application enables the client to follow the accurate area of another client and furthermore to direct a client from a source end to goal end in an indoor condition. In our work, we took the thought from Google Maps Route fighter calculation and joined it with Simplified Attractive Map Matching calculation to accomplish the objective. The application was tried effectively in a chose indoor area of our establishment and it gives precise outcome in various climatic conditions. Contrasted with the current such applications, our application is light weight, easy to understand furthermore, expends truth be told, less memory and CPU time. In this work, the application can control the clients just from a preidentified source to a pre-distinguished goal with a explicit strolling design. In future, this application can be improved to direct an end client from a specific source inside the structure to an appropriate goal independent of their strolling designs.

Paper 7: “Designing of Mobile Map with Open Source Mapping Platforms”

By 1. Chi yuan GU, Design of Mobile Maps with Open Source Mapping Platforms

The expanding utilization of mobile phones come along a increasingly versatile map applications which anyway show up for the most part in institutionalized styles. The point of the work is to make an intelligent map with a existing structure for the utilization on mobile

phones demonstrating georeferenced paper articles. A writing survey should concentrate on variety of maps by and large, portable guide plan and potential connections with portable maps. In the conceptual part of the research paper, an examination of the 'default' Google Maps API and open source mapping stages like Leaflet for various criteria with respect to their properties and particularities will be achieved. Based on the comparisons, a work process for making an intelligent and particularly versatile map with the assistance of open source mapping stages will be described. The created work process will fill in as a reason for the address and exercise 'Mobile Cartography'. The reasonable work goes for building up a model of the prior referenced versatile guide for the Android working framework focusing on representation and guide cooperation. The developed work of submit two printed variants of basic lecture together with the advanced form of Mobile Cartography. The advanced work to incorporate the content depiction and every single required data and programming to run the model. It is urged to distribute the proposition on the production server Qucosa of SLUB. The significant discoveries will be displayed in the type of an A2 shading notice.

Map Design Workflow

First step is to decide the substance of this guide. And after the set up that related information that will be added to the guide. At last, style the foundation layer of the base map. The entire work process of structuring the background map.

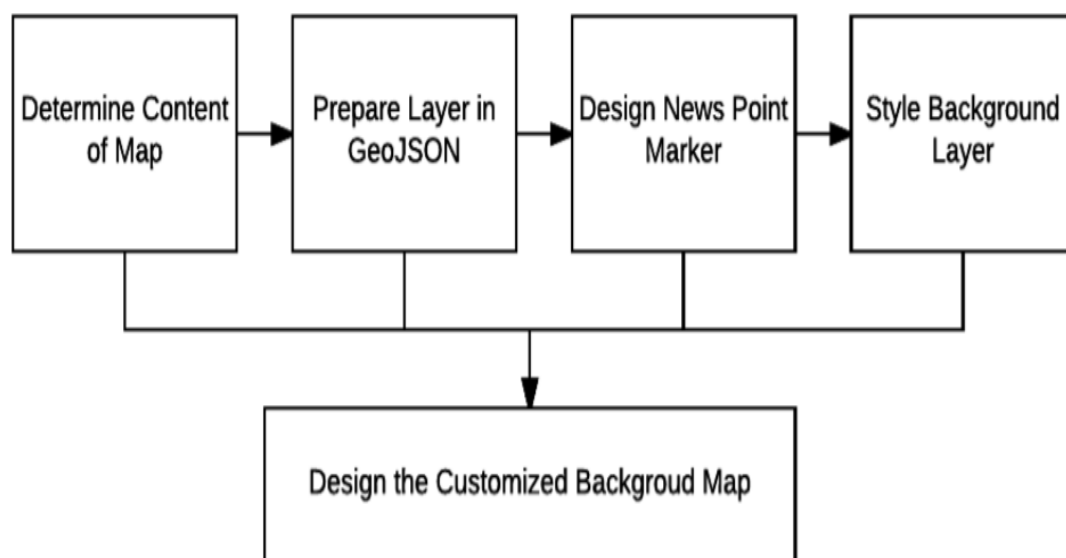


Fig.1. A workflow for background map design

After designing to planning, we get custom foundation map by Mapbox. These base map can be seen as the —Slippy Maps‖ which let you dish around and zoom. In addition, at that point the —Slippy Maps‖ will be incorporated into the site page by Leaflet.js. In the meantime, some interactivities will be joined with some current modules and APIs. At last, this web guide will be incorporated into a versatile application by Android Studio.

The utilization of Web View holder in Android application gives a chance of making a cross breed Android application, that quickens the way toward creating portable guide. Structured web maps can be coordinated into the portable application straightforwardly. When running the application, the sprinkle screen comes up and it connects to the classification movement consequently in no time flat. In the classification page, clients can choose the news that they are intrigued as per the classification. Subsequent to taping the guide style symbol in the upper right, it goes to the principle map movement. At the underlying dispatch of the guide, purposes of news stories are shown on the web map which is inside a WebView holder. These point markers are displayed as groups to manage the issue of data flood.

This proposal exhibits a method for making an interactive mobile map for demonstrating geotagged newspaper dependent on open-source stage. As the consequence of this work, the advanced pattern of mobile mapping is gotten a handle on by alluding to later related investigations. Additionally, this theory gives a viewpoint of the well-known web mapping APIs. Through an unpleasant correlation, a superior comprehension of utilizing these APIs comes up. With respect to the usage part, there are two viewpoints – plan of tweaked guide and improvement of this prototypical versatile guide application. Mapbox is a valuable internet mapping apparatus that makes the structure for web map helpful. The foundation map is structured by setting different related properties including substance of guide, shade of layers, and the text style of names. All control can be appeared in an agreeable UI. Then again, Android gives a WebView compartment, which can show the substance of website page, for engineers.

This compartment permits building up a cross breed Android application that gets more help from web procedures. This contextual analysis effectively utilizes the open-source techniques particularly for some JavaScript libraries including Leaflet API, MarkerCluster module, jQuery, and jQuery portable. These open-source strategies permit cartographers doing web mapping individually, which results in such a large number of fascinating applications come up. Moreover, acing these related systems can help us in picturing and examining geospatial information. Through this contextual analysis, clearly an attractive guide can be effectively

planned by open source stages. Numerous techniques are offered to make an assortment of maps with novel astonishing style. Furthermore, the hole between the web guide and portable guide is getting littler. For this situation consider, the center capacities are acknowledged by web mapping system, then the web guide can be coordinated into a local Android application effectively.

Paper 8: “ANDROID APPLICATION USING GPS NAVIGATION”

By 1.Bagrecha Komal S,2. Bramhecha Amit R,3.Chhaged Sneha S,4.Chhaged Sneha S,
5.Khivsara B.A

Existence in the period of Group IV frameworks and innovation is developing and changing at pretty much every nanosecond. Mobile phines and Palmtops are additionally no special case for the equivalent. Wherever we continue attending the word Android and Maps which is at its pick of accomplishment among youth. Because of fast development and colossal changes in Android, we chose why not to plan an application which can client inclined according to their need. Along these lines, we planned a lot of numerous applications like Place Marking, Shortest Path Plan and Weather Forecasts for Next 4days. These applications work with co-appointment with Google Maps in live condition to monitor the Places and focal points according to client for some time later. Likewise, these applications locate the briefest way and climate expectation for next four days in degree centigrade just as Fahrenheit

In recent years the phone has turned into the significant part of the individuals. It is important for human being to have a ground-breaking gadget which will give everyone in the offices other than fundamental office, facilities in smart phones. Android give such usefulness which empowers the engineers to structure such applications which will make a straightforward versatile to smart one. "Android is based on the open Linux Kernel. Besides, it uses a custom virtual machine that has been intended to streamline memory and equipment assets in a versatile situation. Android is open source it tends to be generously stretched out to fuse new front line advancements as they rise. The stage will keep on advancing as the engineer network cooperates to fabricate creative portable applications." The proposed framework will give the offices to the clients when client is more up to date to wherever through the application named places catalog. Another utilization of proposed framework is briefest way will give the administration to client by mining the most limited separation among source and goal. The last application named GPS Alarm of proposed framework will give the warning to the client of landing of its predefined goal while going as caution.

The test tended to be addressed by the mobiles was the capability to get definite area from the predefined top favorites choices, finding the most limited way with quicker speed and advising the client about the predetermined goal while traveling. Brought up from the research paper that numerous applications had been developed, however some of these subject are to be programmed developed, acknowledgment, time segmented, while those applications don't give definite data about location which client required and accurately brief way to find the shortest path in term of distance. Subsequent to examining numerous research paper about the same and furthermore tended to numerous issues yet the one's referenced above are the serious issue so the proposed framework will endeavor to take care of numerous issues related with speech recognition and time segmentation. So as to help the client who is more up to date to the city, at venturing out time and to get the regular alternate ways, Places Directory, Smart separation and climate estimate are amongst the best alternatives.

The primary target of Android applications utilizing GPS navigated route is to give rules or guideline to the individual who is more up to date in the city and keeping in mind their travelling convenience. As proposed framework incorporates application Smart distance that gives the shortest path to achieve the goal with which that important time of individual gets saved. Another application named Places Directory provide the facility to discover different places adjacent to the individual that person needed for the most part to visit every day. Again another application named Weather Forecasting gives administration as far as climate data of determined spot with pictorial information.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 System Requirements

3.1.1 Android as an Operating System

Android is a mobile operating system i.e. android OS which is a modified version of Linux open source Unix-like operating systems, developed by a startup, Android, Inc. We know that as Google entered the mobile market, it purchased the copywriters of Android and to encourage independent developmental work, it released the software and the developer tools under the open source Apache License. This permissive licensing allowed the Operating System and related software to get modified and distributed to and by enthusiastic developers, network operators and device manufacturer.

3.1.1.1. Dalvik Virtual Machine Architecture

Modified version of JAVA language are used for application development with the help of Dalvik VM which is used to run the mobile app on Android devices. This Dalvik VM can be viewed as the modified version of JVM which constrained in term of storage and processing speed and convert the java bytecode in form of JVM compatible .class files to be compatible by Dalvik VM and is converted in. dex which is executable before installation.

3.1.1.2. Application Interface and Hardware Supports

Based on the Direct Manipulations, on screen object had been programmed so that it can respond to real world entities like swiping, tapping, touching etc. Boasting of the fast and responsive touch screen, the Operating System support various dedicated and tested hardware like various proximity sensor, gyroscope, magnetometers and accelerometers etc. The Home Screen is considered to be analogue to Desktop in the Windows OS.

It is powered by Google Play Store, millions of application can be readily downloaded and used. Application is available in. apk format. Google provides the SDK, free of cost which can support a comprehensive set of developed tools which includes an IDE (Eclipse), debugger, and support for the emulator and for the sample code etc. Android even support C/C++ extension or bytecode through JNI and the support is also available by the Native Development Kit (NDK).

3.1.1.3. Architecture

Android which is based on Linux kernel, middlewares, API and various existing and developing libraries are written in C language. Hardware platform is generally of the ARM architecture with all the later support which is being available for x86 and MIPS. All the GNU library are not supported in the system, hence restricting the porting of Windows application on Android. Users are not given the ROOT access and hence they only have access to data partitions on flash memory and not to the whole system which can holds OS and the boot files and all other sensitive read only partition.

3.1.1.4. Memory Management

The Operating System support multi-threading which is dependent on instant memory availability, this can kill's application's so as to reduce the overloading. RAM management is so much important that power feeding is at its least. As-far-as third party requests is considered, the app SDK is provided with the ample of library entities such as Service, Experience Task and Foreground Task for employed with application lifetime.

3.1.2. GPS Based Location Tracker for accessing location

The application GPS navigation system which is GPS service, based on application which will help us in locating and tracking the device location that exact geo-positions of the user depends upon the current location or situation. The Geo-positioning which is displayed on map view in our android set and which display functioning can be an analogue to the current usage of map services like Google Map / Nokia Map / iOS Map Services. Some key point about the App are:

- the user location will be retrieve from the database so to centrally control the permission for viewing.
- to restrict the user access and user authentication is even supported.
- Periodic refreshing intervals is also important so that each time geo-location change after a fixed intervals of time, the value in database should be updated.
- All the user plans would be having unique ID (UID) which could be used for searching for the device location.
- The application would have additional support in terms of
 - Road Views and Satellite Views
 - Positions on the Map

- Extracting Addresses from the Map
- For locating multiple users (support for Multiple positions)
- Zooming in or Zooming Out
- App User Data Manipulation or password
- management of UID support

Offering effective turn by turn directions for pedestrian, cyclist or driver is much simpler calculating the utmost directly connected in the street networks. This open source routing machine Project, OSRM powering forthcoming Direction API tools an improved and complete display model that capture many feature beyond the simple topologies. With OSRM, we not only acquire the direct and fastest routes, but unique that takes into description of chance restriction and weighted cost like waiting during traffic lights when it is red, applying brakes and accelerating at harsh turn, or vertical hills for bicycle riders.

3.1.3. MapBox

The Mapbox provides a developers platform which is used across manufacturing industries to implement practice application that can solve problem with map, its data's, and three-dimensional analysis. Mapbox's tool forms the structure block for the support of every part of web and the mobile map development process. Whether one goals are to shape a attractive maps to match ones websites or build a fully functional geo-processing applications, mapbox have ones app concealed.

These directives are introducing us to the constructing block of Mapbox and how we can:

- Working with Mapbox's vigorous information.
- Styling our maps down toward smallest detail.
- download or create information.
- Developing a fully featured webapp and mobile apps.
- Extending our app's function with web services for geo-coding, spatial analysis, directives, and more.
- Create stable maps programmatically.

3.1.3.1. Design a map

Tradition map project are one of the essential function of Mapbox Studio. MapBox delivers some advanced applications for tapping these customizations at our fingertips through the help of Mapbox Studio styling editor. In Mapbox Studio, we can start with our own template

styles or designer styles which can style each separate layer to our exact requirement.

We can see the Basic styles, the template styles made up from a limited set of layer. We can also see a custom styles that use all the same fundamental data, but that has been modified by altering colors and font in Mapbox Studio styling editor.

The Mapbox is styling editor is to fully introduced map editors that provides us with total power over all the styles of our map directly in our browsers. Whether we start with Mapbox styles or starts from scratches, the style possibility is virtually limitless. With using this Mapbox's styling editor, we can:

- Generate style that can be changing dynamically which is based on various zoom levels.
- Using customized fonts.
- Setting customized alignments, pitches, offsets, and more for our labels.
- Established color, weight, and opacity of our map layer.
- Filtering all the tile sets is based on attribute.

3.1.3.2 MapBox provide various feature for Mobile applications

Mapbox's provide a Map SDK's for IOS's and Android's for displaying our map in built-in application. The Map SDK for iOS and Android's are planned to be dropped in replacement for MapKit and Google's Map SDK. The Map SDK should be familiar to the mobile developer who has knowledge of either the technologies. Often, our map be able to be changed for Mapbox for altering a sole line of my program.

Every SDK's belongs with bundled 5 Mapbox design map styling and could be handled by any practice designed generated with the Mapbox's Studio styling editor. If we use our Mapbox mobile SDKs, we'll also gain access to mobile procedure console, which provide a continuous update view of periodic active operators, map convention, and region by region metric.

3.1.3.3. Use of satellites imagery

Mapbox Satellite remains a global base map of uninterrupted satellites and aeriels imagery that we can use as a blank canvas or the overlay for our own statistics. It comprises of multiple images bases, we color the correct it and inform it as new descriptions which becomes available.

Mapbox Satellites uses universal satellites and aerial's imagery from profitable provider, NASA's team, and USGS's. As cities develops and lands change, we enhance fresher, vibrant,

and more beautiful images. The recent zoom level offering includes:

0–8: MODIS from 2012–2013.

9–12: Landsat 5 and 7 from 2010–2011.

13–19: is a combination of open and proprietized source, including DigitalGlobe GBM 2011+ for our world, USDA, NAIP 2011–2013 are the contiguous United State, and open aerial images from Denmark’s, Finland’s, and part of Germany.

3.1.3.4. Access Tokens for accessing Map api

To use these Mapbox tools, API, or SDK, we'll be needing a Mapbox access-token. Mapbox use access token to associate API request with our accounts. we can find our access token, make new ones, or remove existing one on your Access Token page or programmatically using a Mapbox-Token API's.

3.2 Implemented Libraries/Model/Algorithm Used (Open Source Routing Machine)

3.2.1. General Options provided by OSRM

OSRM model can be use as a collection (libosrm) via C++ in its place of using it from side to side the HTTP interfaces and osrm routed. Osrn allow for fine fine-tuning point in OSRM model which has considerably fewer overheads. Now is a quick overview into how do we get to use libosrm in the forthcoming v5 release. Takings the appearance at the sample codes that lives in the model directories. At this point all we can ever required to knew about libosrm are that it is a small description of whatever the type do and wherever to find records on it:

3.2.1.1. REQUEST:

Parameter	Description
service	One of the following values: <code>route</code> , <code>nearest</code> , <code>table</code> , <code>match</code> , <code>trip</code> , <code>tile</code>
version	Version of the protocol implemented by the service. <code>v1</code> for all OSRM 5.x installations
profile	Mode of transportation, is determined statically by the Lua profile that is used to prepare the data using <code>osrm-extract</code> . Typically <code>car</code> , <code>bike</code> or <code>foot</code> if using one of the supplied profiles.
coordinates	String of format <code>{longitude},{latitude};{longitude},{latitude};{longitude},{latitude} ...</code> or <code>polyline({polyline})</code> or <code>polyline6({polyline6})</code> .
format	Only <code>json</code> is supported at the moment. This parameter is optional and defaults to <code>json</code> .

Fig.2. REQUEST FNCTIONALITIES OF OSRM

Passing any value like {option=value} is possible. The polylines follow's Google's polyline arrangement with exactness of 5 by defaulting and can be created using this packages.

GET:

```
/{servic0}/{version0}/{profile0}/{coordinates0}[.{format0}]?option=valu&option=value
```

3.2.1.2. Responses

Every reaction objects had a code things maintaining the unique strings which are below or which are a service codes:

Type	Description
Ok	Request could be processed as expected.
InvalidUrl	URL string is invalid.
InvalidService	Service name is invalid.
InvalidVersion	Version is not found.
InvalidOptions	Options are invalid.
InvalidQuery	The query string is syntactically malformed.
InvalidValue	The successfully parsed query parameters are invalid.
NoSegment	One of the supplied input coordinates could not snap to street segment.
TooBig	The request size violates one of the service specific request size restrictions.

Fig.3. RESPONSE FNCTIONALITIES OF OSRM

Message are a elective human understandable fault messages. While all position type are services dependents. In any case of errors the HTTP's code are 400. Else the HTTP's code will be 200 and codes will display "Ok". Every reply to the object has the data version properties containing timestamps from the unique OpenStreetMap's file. This fields are optional. we can consider that if data version parameters are not set on osrm extract stages or OSRM file has not osmosis replication timestamp section.

GET:

```
{
  "code": "OK",
  "message": "Everything worked"
}
```

3.2.2. Service provided by OSRM

3.2.2.1. Nearest Service

Snap of the coordinates to the road networks, provides the nearby ‘n’ matches.

Whether the coordinate will only support single long, lat entry.

This accumulation to general selections the subsequent option is supported by the service are:

Table1 : Options provided by Nearest Service

Options	Value	Descriptions
numbers	Integers ≥ 1 (default 1)	No. of nearest segment that should be returned.

Responses to Nearest Service

If the code which is request was successfully ‘Ok’ else the service will be dependent on the general status code. Waypoint arrays of waypoint object are sorted by space to the input coordinates. Each objects have at least the subsequent additional properties:

nodes: “Array of OpenStreetMap node ids”.

GET:

`http://{service}/nearest/v1/{profile0}/{coordinates0}.json1?number={number}`

3.2.2.2. Route service

It helps to find the fastest direction between nodes in the complete order.

In accumulation to the over-all options the subsequent option is to be supported for these service:

Table2 : Option provided by Route service

Option	Values	Description
Step	true , false	Reverted routes step for every route legs
annotation	False,true, node, speeds, durations , distances ,datasource ,weights	Return additional data for every coordinates along with all the route geometry.
geomety	polyline, polyline6 , geo-json	Reverted route geometry format
overviews	simplified, false , full	Add overviews geometries either filled, simplified rendering to high zoom lvl it can be displayed on
continuein g straights	false, true ,default	Force the paths to keep going in a straight at waypoint limiting u-turn there even is if it is faster. Default value depend on the profiles.
Waypoint	{index};{index}...	Treat the input coordinate indicated by given index as waypoint in returned as Match objects. Default is to treat all input coordinate as waypoint.

Response to Route service

Codes: if the requested code was successfully Ok otherwise we see the service which is dependent has a general code status.

Waypoint: Arrays of Waypoints object are representing all the waypoint in order:

route: An array of path's objects, which is order by decreasing recommendations rank.

GET:

```
/route1/{profile0}/{coordinates0}?altrnatives={ false| true }&step={ false|true }&geometries={ polyline0|polyline6|geo-json }&overview={ full|simplified|false }&annotations={ true|false }
```

3.2.2.3. Match service

Maps similar matches or snap given GPS point to the street networks in the most reasonable

ways. Please note the requests may results multiple sub traces. Large jump in these timestamp or improbable transition lead to traces split if the complete match is not found. The algorithm may not be completely able to match all the points. Outlier are removed if any one of them cannot be matched effectively.

In adding to these general decisions the subsequent option are maintained for these services:

Table3 : Option provided by Match service

Option	Values	Description
step	true , false	Reverted route step for each route
geometry	polyline, polyline6 , geo- json	Reverted route geometry format
annotation	true , false node, distances ,duration s , datasource ,weights , spe eds	Return additional data for every coordinates along the route geometries.
overviews	simplified, false, full	Add the overviews geometries of either full or simplified depending on the highest zoom lvel which might be displayed on or non at all.
timestamp	{timestamp} [;{timestamps} ...]	Timestamps for each data location in second UNIX's. Timestamp needs to be in monotonically accumulating.
tidy	true , false	Allow the input tracker modifications to obtain better matching qualities for noisy trackers.
waypoint	{index}{index}...	Treat the input coordinate indicated by given index as waypoint in return Matched objects. Defaults are to treat all input coordinate as waypoint.

Radius for every points should be a standard errors of the locations measured in meter from the locations. which Uses the Location.getAccuracy() on our Android or we can also use CLLocation .horizontalAccuracy in iOS. These values are used, to control which point

should be considered as a candidate and how commonly each candidates are of larger radiuses mean far away from candidates. The area to be search is to be selected such that, right candidates would be reflected at 99.9 % of the time.

Response to Match service

Code: If the request code is successfully Ok, otherwise we will see the services dependent on general status code.

Trace point: Arrays of Waypoints object representing all the point of traces in order. If the trace points are committed by maps matching because there is an outlier, then the entry's will be null. Each Waypoints objects has the subsequent additional property:

- **matching_index:** Indices to the Path objects in matching the sub traces are also matched to.
- **waypoints_index:** Indices of the waypoints inside the matched paths.
- **alternativecount:** No. of likely alternatives matching for these drop points. A value 0 indicates that if this point is matched clearly or not. Then we split the trace at this point for the incremental map matching.
- **matching:** Arrays of Paths objects that assemble the trace.
- **confidence:** Confidences of these similar float values between 0 to 1. Where 1 is considered that the corresponding is correct.

GET:

```
/match1/v1/{profile0}/{coordinates0}?steps={false|true}&geometrics={polyline1|poline6|geo-json}&overview={simplified|false|full}&annotations={false|true}
```

3.2.2 Significant interfaces objects of OSRM

EngineConfiguration - for preparing an OSRM's instance or instances it can just need to configures certain property and constraint. E.g. storage configuration which are the base paths such as France. OSRM model from which we have derived and loaded France.osm. osrm.secondary file. This similarly let us to set constraint such as maximum number of positions allowed for specified service.

OSRM mechanism - The main Routing Machine type with various function such as Routes and Tables. We have to initialize a EngineConfig. OSRM does all heavy lifting for

us. Every functions takes their own parameter, e.g. the Routefunction which take RouteParameter, and a out reference to the JSON results that get filled. The returned values are the Status, indicating errors or success.

Status - Status is a type of wrapping Errors or Oks for indicating errors or success, separately.

TableParameters - are the examples of parameters types for the Routing Machine function .

In these cases Table expect it own parameter as TableParameter. We can view its wrapping with two vector, source and destination which are indices into our own coordinate for the table services to constructs a matrices from empty source or destination. If we ask ourself where are the coordinates coming from, we can view TableParameter inheriting from the BaseParameter.

BaseParameters - the most important hold coordinate and the few other option property that we don't need for the basic usage, these specified parameters type inherits from BaseParameter to these member of attribute. Such that which mean our TableParametertype has coordinate, source and destinations member's attribute and the few other that we can ignore for now.

Coordinates - are the wrapper around along longitude, latitude pair. We generally don't care about these (longitude, latitude) pairs vs (longitude, latitude) pairs but we don't want anyone to by chance mix them up, so both the latitude and longitude coordinates are strictly typed as wrappers around integer or floating point.

Parameter for various other service. All other Parameter we needed for Routing engine service.

JSON is a total sum type resemblance of JSON. The Open Routing engine service function which takes a out refference to a JSON. JSON is currently implementing mapbox or variants which are alike to Boosts. Variants. There are two way to which we can work with these sum types: either by providing visitors so that it can act on every type on visitations or we can use the get functions in case we're sure of the structure. The JSON structures are printed down in the form of HTTP's API.

3.3. Implementation

Components of map navigation

Any navigation system has user face component that can interact with the other user and back-end component which computes the shortest path based on the user's inputs and the available information. In these guides we will mainly focus on back end component that rely especially on map information.

User facing component of map navigation

- User interfaces control for setting the original and destination position
- Maps displaying the visual represents the path to take
- Turn by turn navigation that accurately describe the navigation manoeuvre's at appropriate time
- Path status that include current progress and estimated time of arrival at the destinations
- Points

Back end components of navigation system

- the navigation profile which describe the mode of transportation to use for routing algorithm
- A path finder that find the shortest possible routes between the original and the destination for the selected location profile
- A direction system that changes the path into commands for turn by turn navigations

3.3.1. Accessing Map from Maps SDK for Android

Mapbox Map SDK is an open source tool set for viewing variety maps in our android application of project.

3.3.1.1. Install the Map SDK

Before starting to develop the applications or projects with Map SDK, we needed to add the map SDK as my dependency. We can find the subsequent dependencies given underneath in the Maven Central repository. Although these documents show us how we can insert the stable versions of the Map SDKs inside of our projects, we can also use these night builders (i.e. SNAPSHOTs) or these beta versions, if these are available.

- Start your Android Studio.
- Open up our application build.gradle files.
- our project minSdkVersion is at API 23 or higher.
- Under these dependencies, add a new implementational dependencies lines for the latest mapbox-android-sdks.
- providing these data in my toolbar, then pressing Sync Project with Gradle File.

```
repositories {
    mavenCentral()
```

```

}
dependencies {
    implementation 'com.mapbox. mapboxsdk :mapbox-android-sdk:7.3.4'
}

```

Note: Here we might had mismatching's of Gradle dependency once we add the Mapbox's Map SDKs for our Android project. Don't forget that we can still use excluding groups to remove certain dependencies:

```

implementation ('com.mapbox .mapboxsdk :mapbox-android-sdk:7.3.4'){
    exclude group: 'groupname', module: 'modulename'}

```

Running these gradle application module name here: dependency in our command lines will display a list of dependency. ./gradle app: dependency work if we can have a Gradle wrappers. These are helpful for troubleshoot of nimble Gradle configuration where various library are included in a project. We can view these dependencies that specifies library can bring and where conflict may be happening.

3.3.1.2. Get the access token

If one do not have any Mapbox's account, he had to sign it up, and then he had to navigate to his Account page. Providing the **default public key token** to the memory. After we have added new Map SDKs as dependencies in our Android projects, we have to open the R.strings.xml files, create a new String resources, and paste this access token.

```

<strings name="mapboxs_accesstoken">MAPBOXACCESSTOKEN</strings>

```

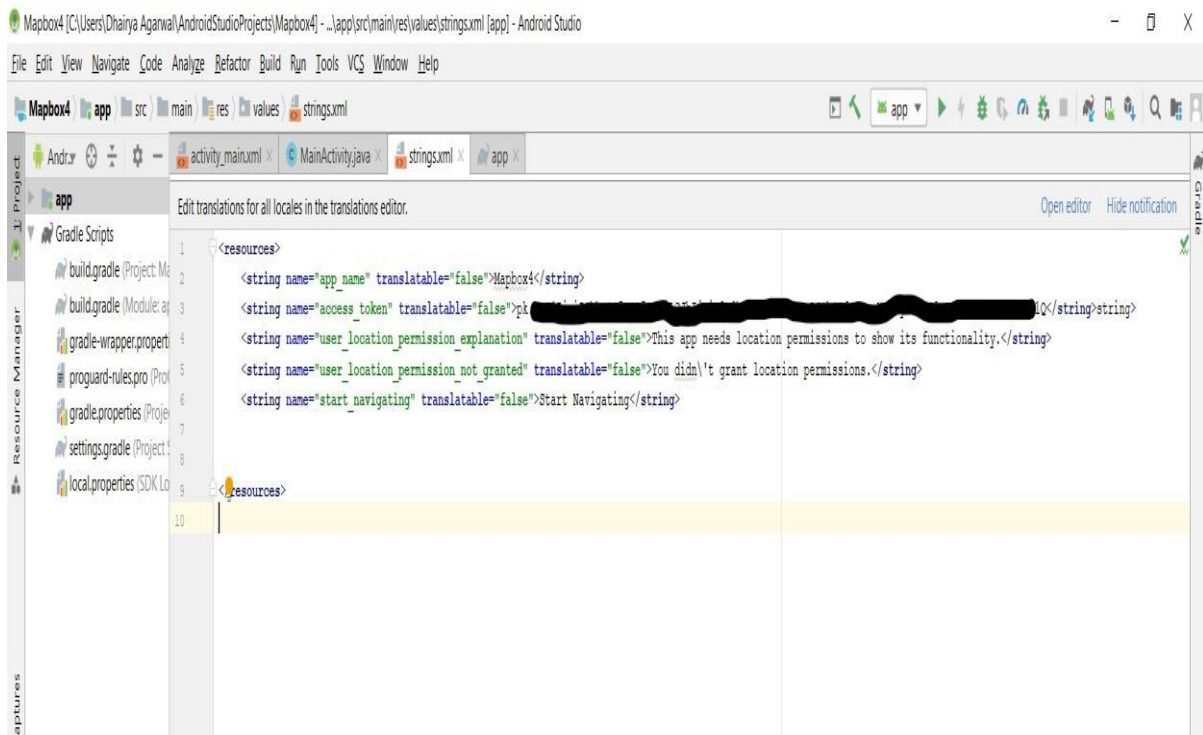


Fig.4. Access token for the project

Then we pass key into my Map SDK's, I had to place this access pass key inside my application onCreate() method.

```
class MyApplication1 extends Applications {
    @Override
    public void onCreate() {
        supr.onCreate();
    }
}
```

```
Mapbox.getInstance(getApplicationContext(), YOURMAPBOXACCESSTOKEN);
    }
```

Map SDK's also provides a setToken() in for us if we want to change this Mapbox access tokens during runtime. Many Mapbox API requires Mapbox token.

Setting this new token enable us with multiple Mapbox tool with conjunction with each another. These methods allow us to set a tokens before specific Mapbox's tools which are used instead of setting the tokens initially and these are being required to use the same access tokens for all other Mapbox related request.

Code in OnCreate() method

```
Mapbox.setAccessToken(YOUR_MAPBOX_ACCESS_TOKEN);
```

3.3.2. Setup permissions

Initial with the version 5.0 of the Map SDKs, we can use the Manifest file features to decrease the need to comprise any Map SDKs which require thing in my app manifest files. We'll need to add the well or the Coarse locations permissions if need to plan to display the user location on the maps or to get the user locations data. This user location's permissions should be check during the runtime with the help of the PermissionsManager.

In Manifest File:

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

3.3.3. Add a map

This Open Source in the JVM files of my activities where we'd like to includes the map's and then add this code below to this file.

Code for map

```
mapView1 = findViewById(R.id.mapView);
mapView1.onCreate(savedInstanceState);
mapView1.getMapAsync(new OnMapsReadyCallback() {
.....
//add Map style here
..... }
}
```

3.3.4. Lifecycle methods

MapView contain it own lifecycle method for handling OpenGL lifecycles, which is to be called openly so that it can contain Activity's. In order for this apps to correctly call the MapView lifecycle method, we must override the subsequent lifecycle method in the Activity's that contain the MapView's and then calls the corresponding MapView methods. For example, in my onStart() methods look like this:

```
@Override
protected void onStart() {
    super .onStart(){.....};
    mapView1 .onStart(){.....};}
}
```

Like on the onStart() overrides the subsequent lifecycle method also needed to be dominated and included in corresponding MapView methods:

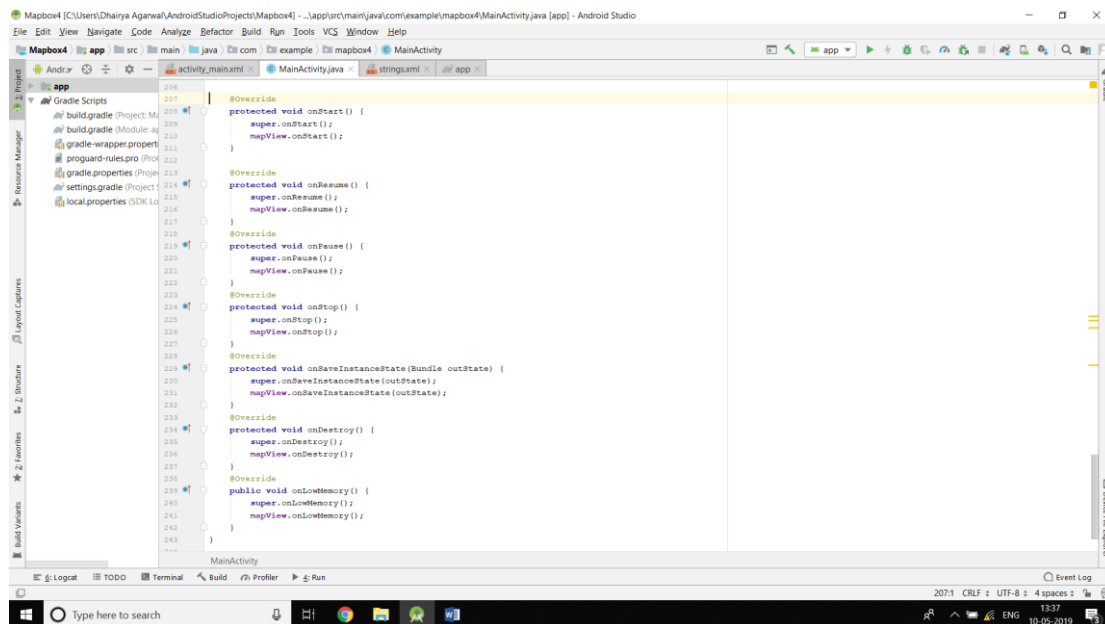


Fig.5. LIFECYCLE METHOD OF MAP

3.3.5 User Location

Navigation application frequently use the user current location as the original location when requesting a path. With this Navigation SDK, it is done using the LocationEngine class. By default, this Navigation SDK’s generated class will use the best position engine obtainable and display the user location’s with the default styling. These guides walk through how to customized both the user’s location icon and location’s engine.

3.3.5.1. User location in map view

To display the user location we could use Mapbox’s LocationsComponent to show the user’s location on the maps. Displaying the user’s location require introduction of several class related to both specified to the Mapbox and the user location. Import the subsequent classes below the class we've already imported our NavigationActivity’s file’s.

We'll also be using various several variable through the process of addition of a user location to the maps. Declare the subsequent variables in the NavigationsActivity.class below the current mutable, isolated MapView mapView1;

Now we are ready to enable this location component and display the user location.

- we will also be needing to add two applications to the NavigationActivity.class : LocationEngineListener and PermissionListener.
- The update to the onStart(), onStop(), and onDestroy() methods is also done

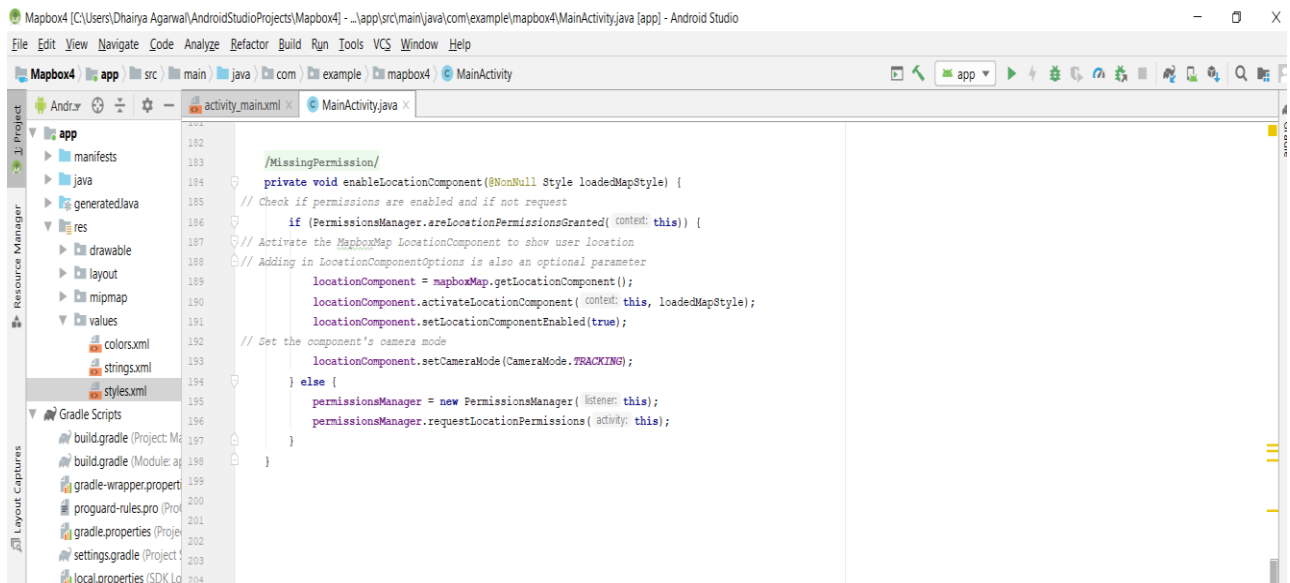


FIG.6. Accessing location component

3.3.6. Route generation

The main Navigation SDK's use path generated by the Mapbox Direction API. The Navigation SDK NavigationRoute.class wrap the MapboxDirection class with parameter that should be set in order for some navigation sessions to successfully begin.

We can use this NavigationRoute.Builder class to be created by a new request to Mapbox Direction API. There are over a dozen methods that can be used to customize our requests, but only three are required:

- accessTokens
- original location
- destination location

This response will be a NavigationRoute's object, which we can use to show a path on a map or start a navigation's sequence.

3.3.7 Route progress

Back in this NavigationActivity file, we have to import the necessary class for addition of a button and then opening a new direction finding view. Then we will add a button with a click listener. When this button is clicked, we start the NavigationLauncher.

Finally, we'll need to specify the button that should change, to be able to enable after a marker is further more added to the map i.e when the destination location is defined. We can accomplish this by setting button to enable and set the background to mapboxBlue color

define in the color.xml file.

This class contain all progresses data at any given interval during a navigation's sessions. This progress include information's for current paths, legs and steps that the user is traversing along with. With every valid location's update's, a new route will progress which will be generated by using the latest data

Table 4:Route progress API

RouteProgress APIs	Description
directionRoute	The path acquire from the direction API and is being used for the navigation.
distancesTraveled	The total distances that the user has travel along the path.
legIndices	The path's current legs indices that the user is on.
currentLegs	The parh's current legs as a path Leg objects.
distancesRemaining	Provide the distance outstanding in meter till the user reache the end of the path.
durationsRemaining	The estimated period residual till the user's arrives at their destination location.
fractionsTraveled	The float value between 0 and 1 gives the total percentage of the user's that has completed in the navigation, based on space.
currenstLegProgress	return the currentsLegProgress object's with data specific to the current path leg. we can also access these steps data through this objects.

RouteProgress APIs	Description
remainWaypoint	No. of waypoint remaining on the current path.
currentStepsPoint	List the Point representing the present step geometries.
upcomingStepsPoint	List the Point representing the forthcoming step geometry.
voiceInstructions	The current VoiceInstructions for the provided segment along the path.

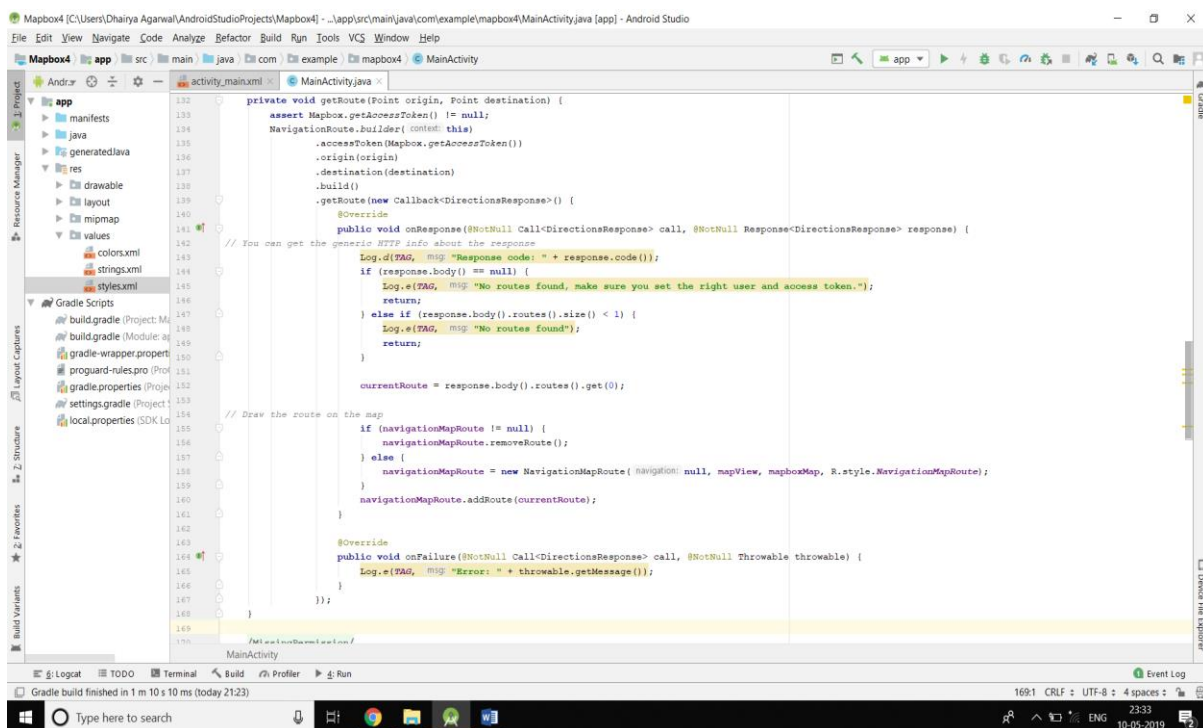


FIG.7. Route Generation method

Device notifications

By default, device notifications display the upcoming maneuver icon's and instruction's texts, as well as distance remaining to said maneuverer. By default, the notification will always be display on the notification bar unless otherwise overridden to a custom notification as outlined below.

Method `NavigationNotification.updateNotification` i.e `RouteProgress.routeProgress` will be called each time that the Navigation SDK's creates a new `RouteProgress` update.

CHAPTER 4: PERFORMANCE ANALYSIS

4.1 Algorithm efficiency

4.1.1 The Simple Model and Path Search

Diagram are the scientific instruments for portrayal of elements and the association between the courses. In any basic chart models of the street organizes, the different elements are crossing points and the courses between them which structure the associations. Overabundance edges or hubs are added to the model for the bend of the street, for example the geometry of the course. The heaviness of an edge signifies the weighted expense to cross it in the picked method of transportation, for example travel time for vehicle steering, remove for strolling bearings, or a mix of time and vitality utilization for electric vehicles.

One of the essential algorithms for directing is Dijkstra's Algorithm, which takes care of the issue of finding a most brief way, I. e. a way of least weight. Basically, it works this way: to course from A to B, begin from a source A_n , and in every cycle, find and settle the closest agitated crossing point. Settling implies doling out a complete separation. What's more, this is the nearest crossing point to any recently settled ones which is as yet not settled itself Connecting neighbouring established ones are allocated a provisional separation that could diminish if a lesser distance—a improved path—is discovered amid the inquiry. As it were, the calculation seeks radially round the cause until it finds the goal.

This resembles a better than average approach to discover a way from A to B, however at a more critical look it has real deficiencies. To begin with, it doesn't represent turn confinements, a typical event in the street arrange. Much more terrible, this inadequacy is by structure. How about we take a gander at a true model and its diagram portrayal.



Fig.8. A typical directing scenario directing a turn limitation.

Expect we are endeavoring to discover a way from a to b. The pursuit settles hubs an and along these lines f. The connection to b isn't investigated as it is illegal, and the pursuit proceeds on the circle c, d, e. And afterward it is ceased, on the grounds that hub f is as of now settled beforehand with a lower weight. The detour street is never found as a feasible way. Also, on the off chance that you enable the inquiry to return to the crossing point after it is settled, you penance productivity and in this way common sense.

4.1.2 Our Detailed Model

The takeaway is that the straightforward model does not ensure way finding when turn confinements are included. To settle this, OSRM changes the street organize diagram. Rather than demonstrating convergences and the streets interfacing them, it shows streets and abandons one to the next. This is known as the edge-extended model. In the edge-extended model we consider connects between road when it is conceivable to abandon one road to other. As it were, streets are presently our elements of intrigue and turns are the associations that we might want to show. Investigate the accompanying outline of the edge extension:

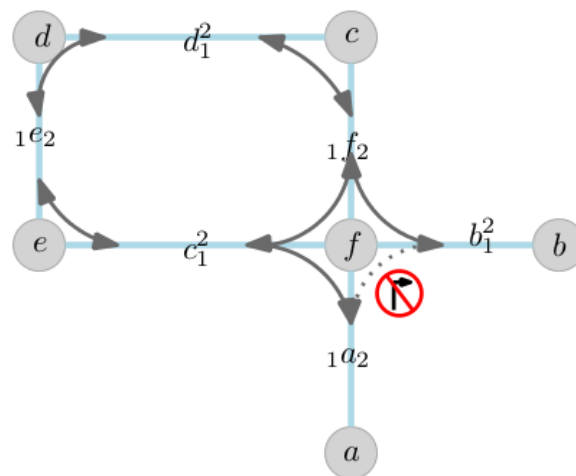


Fig.9. Edge prolonged model—OSRM model streets and actually probable turns somewhat naively modelling the road network.

For each edge we include a hub speaking to every bearing of movement. The dark edges in the image currently reflect abandoning one street to the following. The heaviness of each (extended) edge is the expense of its first (unique) edge in addition to the expense of the turn. Presently, while going in direction of edge from a to b, we essentially need to cross the circle thus discover the detour street.

Clearly, we are including substantially more details into the chart, which has increases the

memory prerequisites of a factor of 3– 4. After cautious improvement the informational index for planet-wide vehicle steering devours about 36 GB of RAM.

The Expressive Power of the Model

The definite edge extended model is clearly helpful for vehicle steering, however is it essential for bicycle courses, which have numerous less turn confinements? Incidentally, the model has a few highlights that make it helpful for an assortment of movement modes. How about we take a gander at two illustrative models.

- Firstly, when going on a bicycle it bodes well to limit the quantity of turns on a course. In urban regions with gridded boulevards there might be a few similarly long courses, yet the best one would incline toward long straightaways to unpredictable turns, particularly where a turn would cross traffic.
- Secondly, the edge with which I had abandon street to the next is significant. More extreme the edge, more you have to de-accelerate and quicken once more.

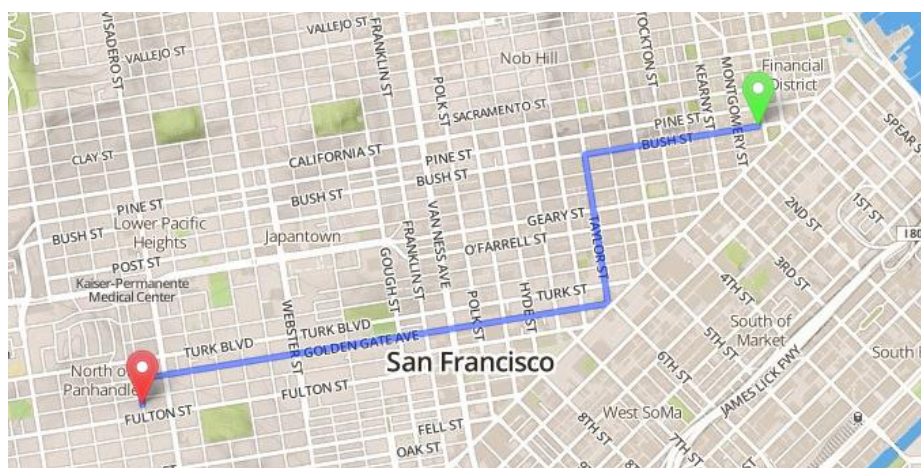


FIG.10. Optimized path for bike through San Francisco’s dodging turns plus high speed traffic.

Turn expenses are a feature for vehicle. By unequivocally displaying each turn, OSRM can represent the expense related with individually turn. For example, how tough is the slowing at a harsh turn, to what extent is normal hold up at a crossing point. Turn points, traffic signals, stop signs, and different components were altogether considered.

Future Developments

We are preparing for what will be a lot of energizing new and ground-breaking highlights. One

of them is improved turn limitations, similar to Michigan Lefts that prohibit going over a specific intersection, yet over various ways. See the accompanying precedent from the OpenStreetMap Wiki..

Example of a composite intersection not permitting for various U-turns.

At long last, we trust that no single steering profile will fit each utilization case and those individuals has assorted and abstract inclinations this ought to be configurable during the inquiry time. We are dealing with conveying profoundly configurable steering profiles, and our first outcomes show that it is conceivable to chop down pre-processing and retention prerequisites much further.

Comparison of OSRM and other Algorithm

Venture was directed with an examination investigation between three OpenStreetMap(OSM) steering motor APIs (CloudMade, MapQuest Open and OSRM) and Google Maps API where we can discover the outcomes where all administrations have a mainland inclusion except for OSRM, the OSM directing motor (OSRM) give the quickest outcomes.

4.2 Implemented result screenshots of the system

In order to check the strength of the method, there are some challenging situation that had been tested and confirmed. These test case is specified as below:-

- The best route provided by the “Smart Navigation System”.
- Turn by turn navigation route.

These analyzing processes are to ensure that the system has the ability to deliver consistent and correct data. To check the navigation and other module by using getting hold of method.

4.2.1 User location

Route applications frequently utilize the client's present area as the starting point while mentioning a course. With the Navigation SDK, this is finished utilizing the LocationEngine class. As a matter of course, the Navigation SDK will utilize the best area motor accessible and show the client's area with the default styles depicted beneath. This guide strolls through how to alter both the client area symbol and the area motor.

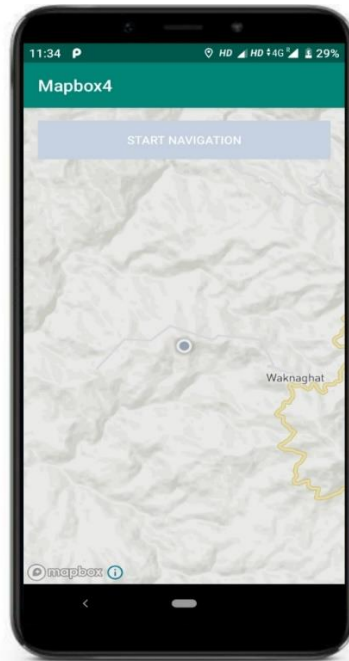


Fig.11. User Location showing JUIT

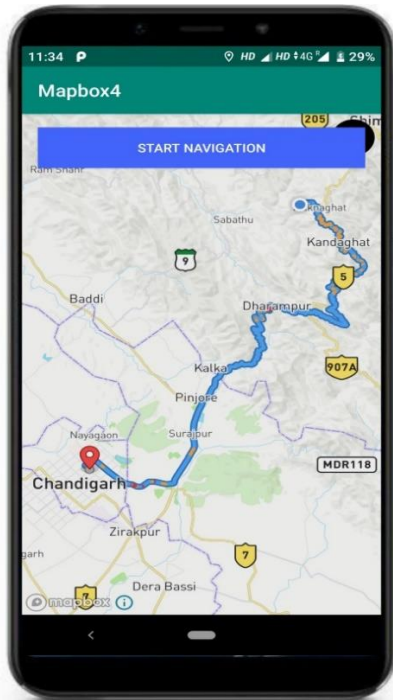
4.2.2 Calculate and draw route

Next, I had calculated and drawn a path between the user location and destination where the marker is added. Firstly, we will import sets of class which are required to compute and draw a paths on the map for the user's using the Mapbox's Navigation SDK's and the Mapbox's Android Service Direction API wrapper.

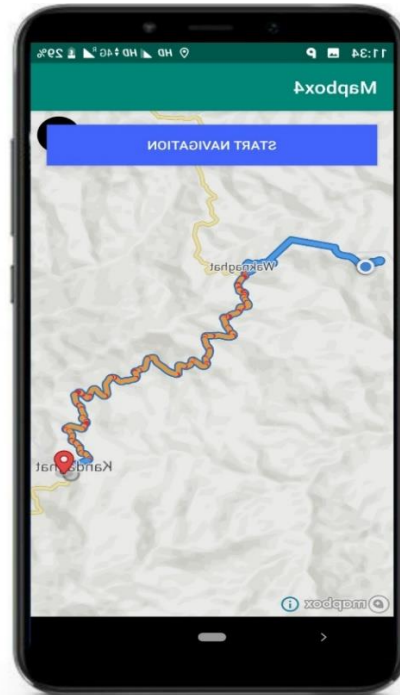
Next, I have added a few more variables that I had being using while computing and drawing the paths.

Now we are ready to mark a new `getRoute()` method. These methods will be requiring a original and destination location to build a `NavigationRoute()`. After providing an original and destination location to these methods, I had make a request to the Direction API. Once I have received a positive response, this response is then stored as the `currentRoute`. Then, I had add the `currentRoute` to my map. I had firstly started by checking if a path already exist. If it does, then remove it and added the `currentRoute()` to tmy map as the `NavigationMapRoute`.

Now I had written the `getRoute()` method, which is still needed to be called when user adds a marker's to the maps. Then I had called the `getRoute()` method in the `onMapClick()` method.



**Fig.12. Route generation from JUIT
To chandigarh**



**Fig.13. Route generation from JUIT
to Kandaghat**

4.2.3 Route progress

Tracking the user progress along the path is a key to providing helpful and prompted navigation instruction. RouteProgress class contain all the user progress data along the route's, including leg and step. The object provide inside ProgressChangeListener's, allowing us to get distance measurement and the percentage of route complete's, current step indices, and much more.

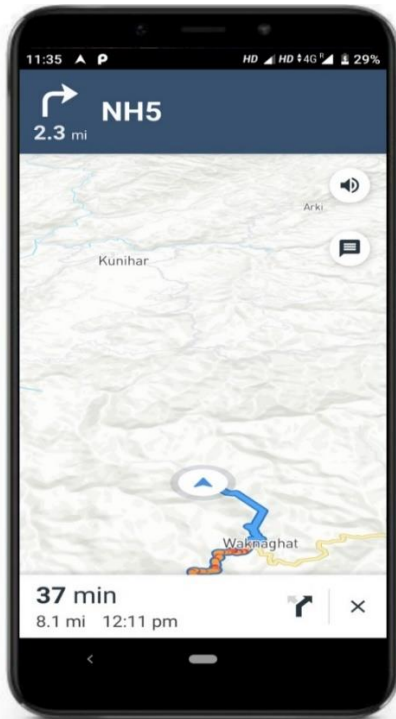


Fig.14. Navigation started from JUIT

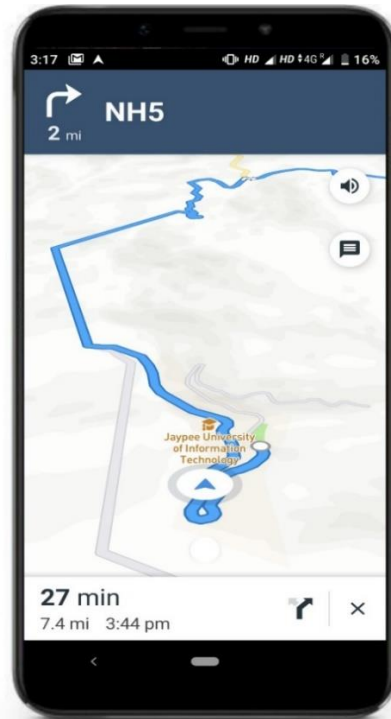


Fig.15. Turn By Turn Navigation in JUIT

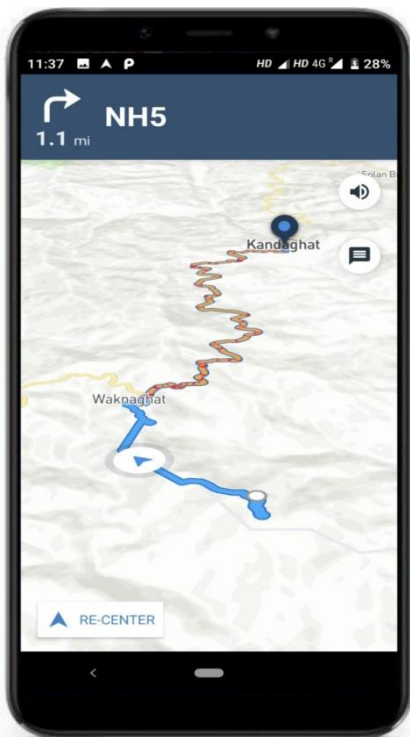


Fig.16. Navigation proceeded ahead

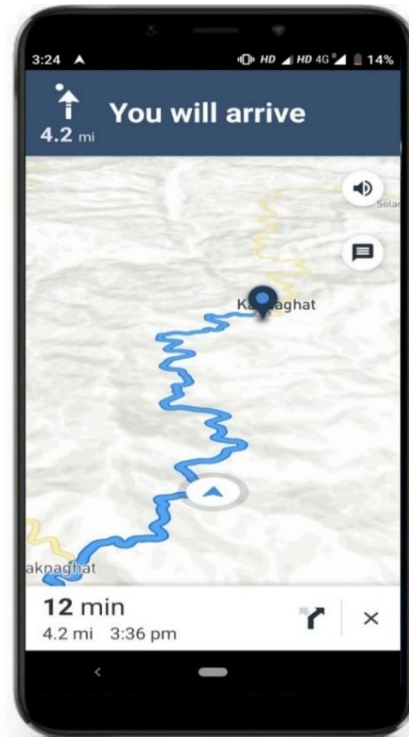


Fig.17. 12 min Remaining to reach the destination

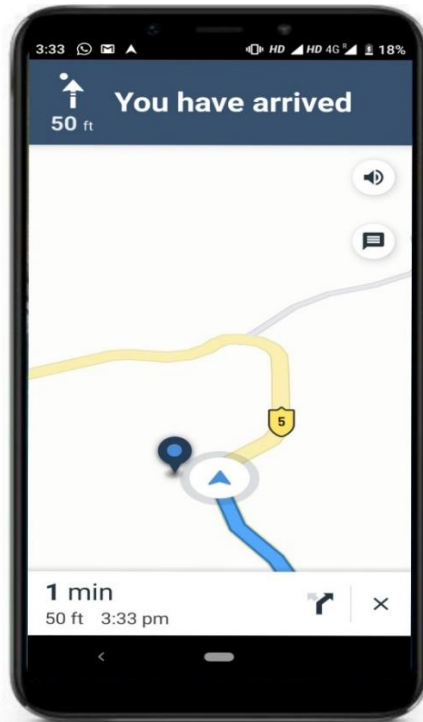


Fig.18. Destination had arrived

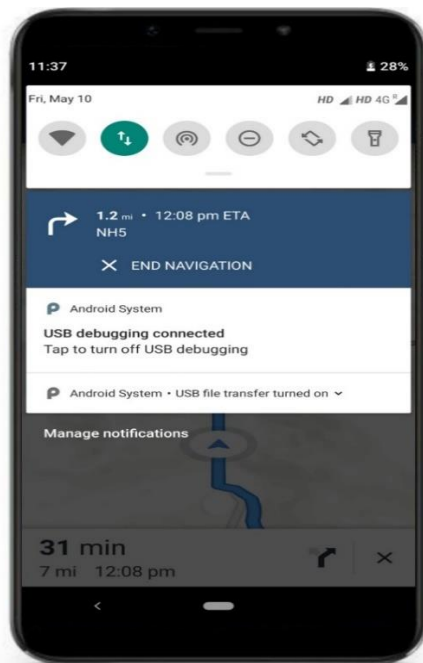


Fig.19. Application Notification while the App is temporary close

CHAPTER 5: CONCLUSIONS

5.1 Conclusions

The purpose of this study is to get familiar with the technology of Android application development and Open Source Routing Machine Model which is generally used for finding not only the shortest path but the best path analyzing all the problems persist on the road or while traveling like traffic congestion, turn by turn prediction, etc.

Able to compute route himself without exhausting the google directions API's in my project, the routes or the navigation between the location are calculated by the Directions API which is provided by a web services provided by the MapBox that provide HTTP interfaces using OSRM Model for routing, thus the organization is required to have internet connections in order to process the OSRM algorithm. The suggested answer is to improve our routing algorithm that is capable to compute the instructions for the navigation with in the remote area perfectly and to navigate and display all the minute details changing every minute on the road.

REFERENCES

1. Malekian, R., Kavishe, A.F., Maharaj, B.T., Gupta, P.K., Singh, G. and Waschefort, H., 2016. Smart vehicle navigation system using hidden Markov model and RFID technology. *Wireless Personal Communications*, 90(4), pp.1717-1742.
2. Singal, P. and Chhillar, R.S., 2014. Dijkstra Shortest Path Algorithm using Global Positioning System. *International Journal of Computer Applications*, 101(6), pp.12-18.
3. Ch. ChakradharaRao¹, P. Pushpalatha², N. AdityaSundar³, “GPS Based Vehicle Navigation System Using Google Maps”, *International Journal of Computer Science and Information Technologies*, Vol. 4 (6), 2013, pp. 979-982
4. MapBox - <https://www.mapbox.com/help/android-navigation-sdk/>
5. Google - "Android Studio 3.4 Canary 5 available". *Android Studio Release Updates*. November 26, 2018. Retrieved November 29, 2018.
6. Vishwajeet Pattanaik², Vishwajeet Pattanaik³, Mayank Singh⁴ Computer Science and Engineering Deptt., Krishna Engineering College Ghaziabad, India 201 007,⁴. P.K. Gupta⁵ Department of Electrical, Electronic, and Computer Eng University of Pretoria, South Africa, 0028,⁵. S.K. Singh⁵ Department of Computer Science and Eng Indian Institute of Technology BHU, Varanasi, 221005 ,“Smart and real-time traffic congestions estimation and clustering technique for urban vehicular road”, 2016 IEEE Region 10 Conference (TENCON) — Proceedings of the International Conference, pp. 1-4
7. J. Parthasarathy¹ Member Technical Staff, Sun Microsystems Pvt ltd, India, Divyasree chambers, off-Langford road, Bangalore-560027, India., “POSITIONING AND NAVIGATION SYSTEM USING GPS SYSTEM”,pp. 5 , *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, Volume XXXVI, Part 6, Tokyo Japan 2006, pp. 23-34
8. Binu, P.K., Krishnan, R.A. and Kumar, A.P., 2016, December. An efficient indoor location tracking and navigation system using simple magnetic map matching. In *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)* (pp. 1-7). IEEE.
9. 1.Bagrecha Komal S,². Bramhecha Amit R,³. Chhajed Sneha S,⁴. Chhajed Sneha S,⁵. Khivsara B.A, “ANDROID APPLICATION USING GPS NAVIGATION” pp. 1-7,

1 st International Conference on Recent Trends in Engineering & Technology, Mar-2012 Special Issue of International Journal of electronics, Communication & Soft Computing Science & Engineering, ISSN: 2277-9477

10. TseHiu Yeung 2 Dr Ray Cheng, “Virtual Campus Navigation system”, pp.1-54, 2013, *Hong Kong Journal of Computer Applications*.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 14-05-2019

Type of Document (Tick): Ph.D Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: DHAIRYA AGARWAL Department: C.S.E Enrolment No 151383

Contact No. 8629010507 E-mail. dhairyaagarwal1234@gmail.com

Name of the Supervisor: Dr Pradeep Kumar Gupta

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): Smart Vehicle Navigation System

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 55
- Total No. of Preliminary pages = 7
- Total No. of pages accommodate bibliography/references = 2

Dhairya Agarwal
(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found Similarity Index at 13 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

[Signature]
14/may/2019
(Signature of Guide/Supervisor)

[Signature]
Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
<u>14/05/2019</u>	<ul style="list-style-type: none"> • All Preliminary Pages • Bibliography/Images/Quotes • <u>14</u> Words String 	<u>12%</u>	Word Counts	<u>12,334</u>
Report Generated on			Character Counts	<u>67,826</u>
<u>15/05/2019</u>			Submission ID	<u>41</u>
		<u>1130714883</u>	File Size	<u>2.91 MB</u>

Checked by [Signature]
Name & Signature

[Signature]
15.05.2019
LIBRARIAN

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

University of Information Technology
Waknaghat, Distt. Solan, (Himachal Pradesh)
Pin Code: 173201