Vehicle Monitoring and Routing System

Submitted in partial fulfillment of the requirements of

the degree of

Bachelor of Engineering

by

1211058 Mit Kamlesh Shah

1211061 Parshva Nalin Shah

1321004 Khyati Mukesh Thakkar

Supervisor:

Prof. Smita Sankhe

(Department of Computer Engineering)

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

(2015-16)

Project Report Approval for B. E.

This project report entitled **Vehicle Monitoring and Routing System** by

Mit Shah, Parshva Shah, Khyati Thakkar

is approved for the degree of Bachelor Of Engineering.

Examiners

1.--

2.--

Supervisors

1.--

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Mit Kamlesh Shah 1211058 --
(Signature)

Parshva Nalin Shah 1211061 --
(Signature)

Khyati Mukesh Thakkar 1321004 --
(Signature)

Date:

ABSTRACT

The GPS based Vehicle Monitoring/Tracking and Routing system is the system that makes use of GPS to provide the exact location of the vehicle. This project aims at creating an application which allows a vehicle administrator or any authentic party to monitor the vehicle in real-time using a GPS-based device possessed by its driver. This system can continuously track a vehicle and show its exact position using GPS. The proposed system is also capable of finding the shortest route to reach the destination passing through all the checkpoints which uses our proposed algorithm to find the same. This system also possesses the facility of speed monitoring and giving sound alerts to the driver if he over speeds his vehicle. Also, the system gives accident alerts to the vehicle admin via SMS.

ACKNOWLEDGEMENTS

We would like to thank Prof. Smita Sankhe for helping us throughout the project by providing us with ideal requirements of the project and guiding us to fulfil the same at each point of development of the project.

We also thank Prof. Bharathi HN, Head of Computer Dept., of KJSCE, Mumbai for encouragement and providing necessary infrastructure for pursuing the project.

We also thank all non-teaching staff for their valuable support, to complete our project.

INDEX

1. INTRODUCTION

1.1 Introduction

In today's world GPS (Global positioning system) is the most widely used technology in vehicle monitoring and Routing System. Security of public and private vehicles like school buses is of prime importance. There are various cases encountered in recent times where the children don't reach home in time or the vehicles loaded with goods are hijacked. Because of which the parents or the vehicle owners get worried. GPS technology can be used to solve this problem. Using GPS and GIS (Global information system) we can find the exact location of the vehicle.

Time is an important factor considered while traveling. Reaching on destination as soon as possible is always admirable. Google Maps provides all the routes from a source to destination. The routes provided by Google Maps are may not be the shortest/fastest route available. There needs to be a system devised which can help a person to reach from source to destination in minimal amount of time. An efficient algorithm is devised for these purposes.

In recent times, the rate in road accidents has increased considerably. For helping the victims of road accidents a quick service should be provided. We have proposed an alert system module in our project. This module is based on GPS, GPRS/GSM technology. A GSM mobile device with active GPRS connection is installed in the vehicle. The position of the vehicle which is received through GPS is sent to the concerned person and to the ambulance hotline number through SMS (Short Message Service). Immediate attention will be provided to the victim through the proposed system.

1.2 Problem Definition

Daily, drivers face problems while traveling on highways, unknown routes and different terrains. Some more problems might be of spare settlement, carjacking or accidents. When the vehicle leaves from its source for its journey, until he reaches its destination, an analyst may rely completely on the previous reports or logs that were generated. No real time information regarding the state of vehicle can ever be transmitted to the analyst. So real time tracking of the vehicle is necessary for safety and security purpose.

Real time monitoring of the vehicle is important these days due to an increasing number of accidents and robbery. Whenever such an incident takes place, the vehicle admin if monitoring on real time basis, can immediately notify nearby hospitals or police stations according to the situation he will be notified with. Also, the parents of school going kids can be informed so that they can take a sigh or for the owner of the good being travelled via transport vehicle.

VMARS is a project that covers most important and rare feature of alerts. Succinctly put, the developed system will be able to (1) Provide real-time tracking and monitoring of the vehicle. (2) Show the shortest path to the driver by applying our proposed algorithm considering all the real-time factors. (3) Alerting the driver whenever he exceeds the safe speed. (4) Proving quick and timely alert to the vehicle admin to take actions in case of an accident.

The motivation behind undertaking this project is mainly for the benefit of the parents of school-going children and notifies them about their children’s whereabouts as well as their safety in a real time environment

1.3 Scope

The system is made up of four main modules and they are:

- Location Module
- Routing Module
- Speed Alerts
- Accident Alerts

Location Module:

Google Map APIs are available for accessing the location from the device. These APIs are integrated within the software which allows us to access Google’s Location Services. The device location is extracted in terms of latitude and longitude. The location module will trace the current location from the GPS enabled device without the need of user specifically entering it. The obtained latitude and longitude will be converted to specific location by GIS (Global Information System).After mapping this Location service, one can easily see the visual display of the road map with provided location.

Routing Module:

is the most important module of this project. It will provide the driver with the path from the extracted location from location module to the destination which is supposed to be entered by the user. The main task of routing is not just to provide the path, but to provide the shortest possible distance from source to destination. For initial routing MDSP (Modified Dijkstra’s shortest path) algorithm is used. The routing from a source to destination depends on various factors such as congestion, distance, time etc. If there is any disturbance in normal course of travel, there will be frequent updation of routes. The route will be updated dynamically by applying the algorithm giving better results among MDSP and A* algorithms.

Speed Alert Module:

Providing speed alerts helps in maintaining security and safety of the kids as well as driver. This module will have predefined threshold values depending on the road type. Whenever the driver exceeds the speed and crosses this threshold value, speed alert in terms of notification will pop up on his cell phone. This will notify the driver that he has to slow down the vehicle which has crossed the permissible speed limit.

The speed alert module comprises of various thresholds. These thresholds will vary according to the various factors. Like highways will have higher values as the roads are broader and congestion free; whereas the local streets might have more congestion, so the threshold needs to be low. This will improve the safety measure of the school buses.

Accident Alert Module:

If the system detects a high probability of the occurrence of an accident, SMS will be sent on urgent basis to the Vehicle Admin with the current location of the vehicle. Then, the vehicle admin is responsible to find out the cause by contacting the driver and taking appropriate actions.

The occurrence of an accident can be noticed by the vehicle admin as the admin is watching over the complete journey of bus. The moment a vehicle stops moving for certain period, admin is expected to follow up about it as there is a chance of accident.

Also if actual accident happens, driver is supposed to contact the admin regarding incident if he is able to. The admin will get the location of the place where the accident took place and then, he can take appropriate actions depending on the circumstances present.

2. LITERATURE REVIEW

S. Sivakumar and Dr. C.Chandrasekar formulated a real-time algorithm called Modified Dijkstra's algorithm whose basic idea was to modify the Dijkstra's algorithm to make it suitably useful for finding shortest route between source and destination considering other real-time factors like traffic congestion, time and distance. They developed a tool using Java and compared the proposed Modified Dijkstra's algorithm with the existing algorithms like DKA ((sivakumar)on Jaipur database. Comparisons were made based on number of nodes visited and time taken to reach the solution.

Another author Liang Dai made a comparison between Dijkstra's algorithm and A* algorithm by implementing both of them on Ottawa city road network. He compared the running times of both the algorithms and concluded that A* can have better running time than Dijkstra's if it uses Eucledian distance as its heuristic function. Though, their time complexities are almost same, they may be chosen depending on the road network chosen

.

A.Renugambal and V.Adilakshmi Kameswari made an android application which can be run on GPS-driven mobile phones equipped by the taxi drivers. These phones send their location via GPS which also has the capacity of finding an optimal route by using min-max algorithm.

Paul Benjamin Et Al designed a GPS-based vehicle monitoring and alert system. They included a management system, fuel usage monitor and an onboard location display along with an accident and robbery alert system. Their project used Google Maps API, SMS gateway server which triggers sending SMS to nearby hospital when the inbuilt air-bags open, a panic button that can be pressed by the driver which is easily accessible to him that sends an SMS to a nearby police station. They formulated a C# code for sending location through GPS to the database.

M.A Hannanet Et Al implemented a bus monitoring system which used radio frequency identification tag (RFID) along with GPS, GPRS and GIS to monitor bus. They used an RFID reader which continuously sends an operating energy or isotropic radiated power to the RFID tag. Based on the calculated distance between the tag and the reader, it is decided whether to obtain the data from the tag or not and whether to send data to the control center or not. The time of the arrivals of the readings can be recorded too along with other processed data which is saved in the database. Then, this data is shown to the bus drivers.

3. PROJECT MANAGEMENT PLAN

3.1 Feasibility Analysis

A highly brief summary of the report and results are given below.

3.1.1 Technical Feasibility:

To develop such an application, the following technical skills and tools are required:

- Java Programming Language
- Android Software Development Kit
- Eclipse IDE (optional but recommended)
- Android Emulator and/or Android device with GPS facility

All of the above technical requirements are fulfilled. The project is technically feasible.

3.1.2 Economic Feasibility:

Considering that this is an under-grad project, ROI and profit in terms of money is not expected. However, we must and did take into consideration whether developing the product itself was within budget.

The cost of libraries, test device (or emulator), and other components was found to be within budget. The project is indeed economically feasible.

3.1.3 Legal Feasibility:

No data is saved by the application without the explicit intention of the user. Therefore, data privacy and other similar legal issues do not apply. The only condition under which the legality of this application would come into question is if the application was used to track someone without his notice. The developer, however, has no control of how the user makes use of the application and hence is not responsible for it and is absolved of any liabilities arising therefore. The product is legally feasible.

3.1.4 Operational Feasibility:

The product has high applicability among a variety of users. Being lightweight, accurate, and easy to use, a wide user base is expected. If we were to compare this application with other applications in the field of GPS, we would find that most applications are limited to a specific type, such as tracking or speed alerts. But no application may be as versatile as this one providind these mant functionalities to the user which are useful too for enhancing te overall user experience. The operational environment specifications were taken at the lowest level possible (lowest API level of Android that would support all the basic features needed in the application). Simply stated, this means that the application will run on a lot of devices. Choosing a higher API level would have meant a richer, more efficient feature set, but that would reduce the possible user base by 40% to 50%. The Operating System (Android) itself is available free of charge. The product is operationally feasible.

3.1.5 Schedule Feasibility:

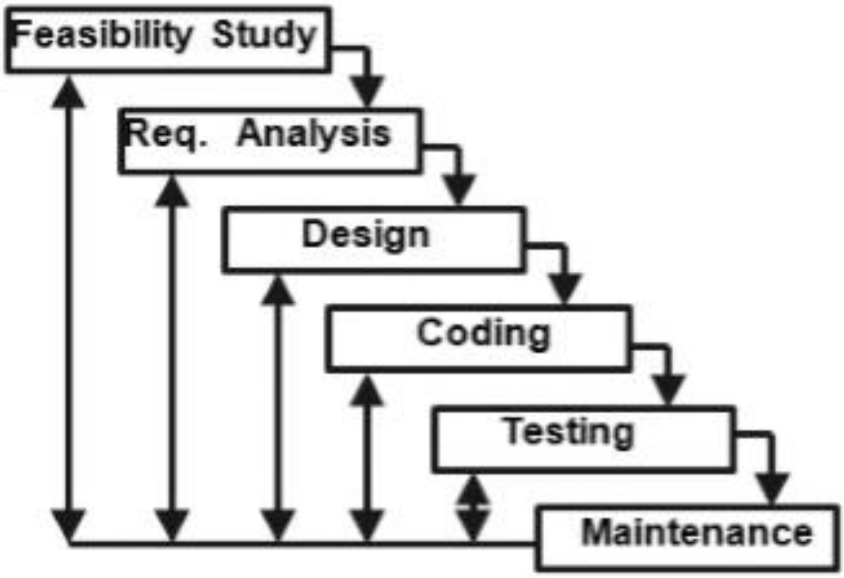
The effort and schedule break up showed that the project was likely to be delivered on time.

3.2 Lifecycle Model

The waterfall model can be used for development of project; the following phases are followed in order:

1. Requirements specification resulting in the product requirements document.
2. Design resulting in the software architecture.
3. Construction (implementation or coding) resulting in the actual software.
4. Integration.
5. Testing and debugging.
6. Installation.
7. Maintenance.

Thus the waterfall model maintains that one should move to a phase only when it’s proceeding phase is reviewed and verified. Various modified waterfall models (including Royce's final model), however, can include slight or major variations on this process. These variations included returning to the previous cycle after flaws were found downstream or returning all the way to the design phase if downstream phases deemed insufficient.



It places emphasis on documentation (such as requirements documents and design documents) as well as source code. In less thoroughly designed and documented methodologies, knowledge is lost if team members leave before the project is completed, and it may be difficult for a project to recover from the loss. If a fully working design document is present (as is the intent of Big Design Up Front and the waterfall model), new team members or even entirely new teams should be able to familiarize themselves by reading the documents.

Advantages:

- The water fall model is easy to implementation.
- For implementation of small systems water fall model is use full.
- The project requires the fulfilment of one phase, before proceeding to the next.
- It is easier to develop various software through this method in short span of time.

Flow of information:

Information would be shared by reliance to the project team which would be as per requirements.

Major Milestones:

1) Identify the proper requirements based on analysis.

2) To develop proper software model do necessary coding.

3) Test the software to identify and remove bugs.

3.3 Project Cost and time estimation

Considering that this is an under-grad project, ROI and profit in terms of money is not expected. However, we must and did take into consideration whether developing the product itself was within budget.

The cost of libraries, test device (or emulator), and other components was found to be within budget. The project was completed by the time of submission and presentation within a span of about 8 months.

3.4 Resource plan

The project relies on human resources along with some technological requirements. Besides, the project team, a room for them to work in, a desktop computer and an Android Device for designing and developing the application.

Labour:
The project team which includes 3 members:

- Mit Shah
- Parshva Shah
- Khyati Thakkar

Equipment:
Desktop PC with 1GB RAM and internet connectivity.
Materials:

- Desktop PC with 1GB RAM and Android Studio 1.5.1.
- A mobile phone with Android OS v5.0 Lollipop with internet connectivity and GPS facility.

3.5 Task & Responsibility Assignment Matrix

The responsibilities and roles given to individuals in order to carry out the required and activities are listed in the table given below:-

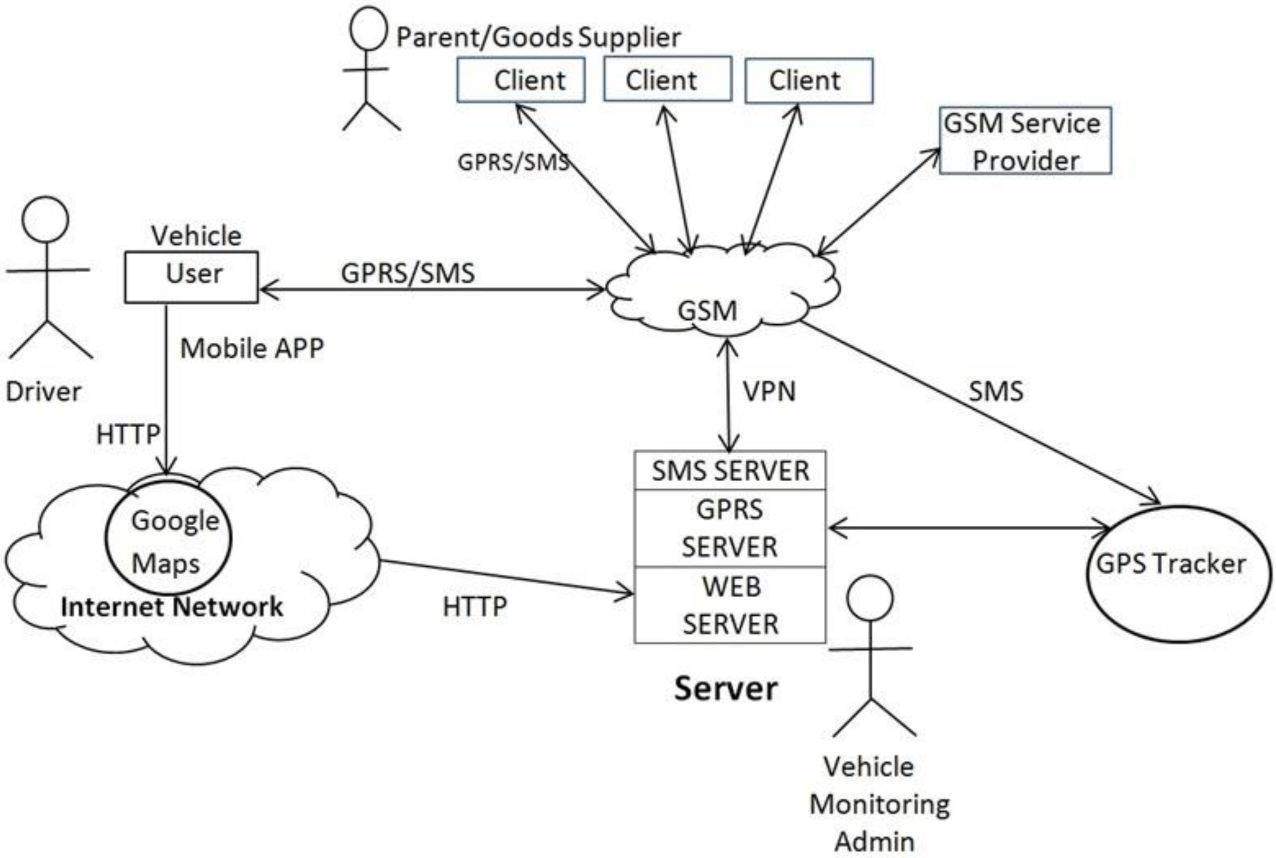
Person	Role	Responsibilities
Mit Shah	Software developer	Information gathering, Algorithm analysis and selection, coding ,testing, analysis.
Parshva Shah	Software developer	Information gathering, Algorithm analysis and selection, coding , testing, analysis.
Khyati Thakkar	Software developer	Information gathering, Algorithm analysis and selection, coding , testing, analysis.

3.6 Project Timeline Chart

Task	Start date	End date
Proposal	10th April 2015	10th April 2015
Software requirement specification(SRS)	2nd August 2015	16th September 2015
Literature survey	1st August 2015	15th September 2015
Algorithm	12th September 2015	1st October 2015
Android Project	2nd October 2015	2nd November 2015
Test accuracy	25th October 2015	20th November 2015
Software project management plan	1st November 2015	15th November 2015
Software design description	11th November 2015	19th November 2015
Report	19th November 2015	21st November 2015
Suggestions' collection of modifications and additional features	4th December 2016	2nd February 2016
Incorporation of suggestions and additional features	2nd February 2016	15th April 2016
Testing of final project	14th March 2016	25th April 2016

4. PROJECT ANALYSIS AND DESIGN

4.1 Software Architecture Diagram



4.2 Architectural style and justification

Our proposed system consists of GPS module, GSM modem, SMS Server and GPRS Server. The System continuously tracks the vehicle through a GSM device with GPS enabled in the device.

Management System :

The management system receives information from the vehicle via a SMS gateway server which is installed at the server side i.e. vehicle monitoring admin. The SMS is sent through VPN (Virtual Private Network).

SMS is used to inform clients about any delay or other important notifications. Driver sends an SMS to vehicle monitoring admin requesting for alternate route in case of a rally or unexpected blockage of a road or if any accident has occurred blocking the road.

GSM :

GSM module provides the GSM service through GSM service provider. All the clients are connected with GSM and GPRS service. This module helps in sending SMS to the vehicle admin.

Driver:

Driver will open the application and the map will be loaded once the GPS and internet is enabled on driver mobile. During the journey, the vehicle will be tracked through client application. In case of accident the SMS is generated and sent to the vehicle admin.

Clients:

Clients are the one who will be using this service. Clients will be able to track the ongoing vehicle in real time manner

4.3 Software Requirements Specification Document

4.3.1 Purpose

- The “Software Requirements Specification” (SRS) will provide a detailed description of the software requirements for the Vehicle Monitoring and Routing System (VMARS).

- Project's target and its UI along with hardware and software requirements are described by this document

- This project provides the users a safe and secure feeling for their loved ones and gives them continuous updates about them.

- The users can view the exact positions of the people, can put the speed of their vehicle under scrutiny and can get the alerts on the mobile phone if accident occurs.

4.3.2 Product Scope

- Finding Shortest Route Module
- Location monitoring Module
- Speed Monitoring
- Alerts in Case of Accidents

4.3.3 References

- The information related to format of project is referred from

- [1] Dr. C.Chandrasekar, S. Sivakumar, “*Modified Dijkstra’s Shortest Path Algorithm* ” IJIR in Computer Engineering (An ISO 3297: 2007 organization)

- [2] Liang Dai, “*Fast Shortest Path Algorithm for Road Network and Implementation*” Carleton University School of Computer Science, 2005 COMP 4905

- [3] A.Renugambal , V.Adilakshmi Kameswari ,“*Finding Optimal Vehicular Route Based On GPS* ” ,IJCSIT, Vol. 5 (2) , 2014

- [4] Shen Wang 1, Soufiene Djahel 2 and Jennifer McManis1 , “*A Hybrid Vehicular Re-routing Strategy with Dynamic Time Constraints for Road Traffic Congestion Avoidance*”

- [5] Nishtha Kesswani,Dinesh Gopalani ,“*Design And Implementation Of Multi-Parameter Dijkstra’s Algorithm* ,Ijaer 2011, September

4.3.4 Overall Description

4.3.4.1 Product Perspective

- This product is based on merging of certain existing system to develop a new full fledge product.

- The SRS defines a component view of Vehicle Monitoring and Routing System, relates the software and interface requirements.

4.3.4.2 Product Functions

- Location -The current real-time location of the Vehicle will be provided to the vehicle admin on the server side.
- Route-The optimal Route will be shown to the driver from a given source to a given destination passing through all the given checkpoints which is calculated by the proposed algorithm.
- SMS Alerts-An SMS Alert will be sent to the vehicle admin with the location of the vehicle in case the application detects an accident.
- Speed Alerts – A sound alert will be given to the driver if he is found to drive the vehicle over a predefined threshold speed value and the sound will stop only when the driver brings the vehicle to a speed less than that threshold value.

4.3.4.3 User Classes and Characteristics

a) For Users:

- User will be able to see the location and they will have to monitor the speed of the vehicle and if the vehicle's speed goes above a predetermined limit for that road, he will be notified about it and will be advised to slow down because safety should be the first preference whether it includes safety of people or goods supply.
- User can see the shortest Route so that he can take that route and reach to the destination as early as possible.
- If there is a chance of any Accident taking place the SMS notification will be sent to the number fed in the system.
- Users won't be able to make changes to the system and will be questioned by the vehicle admin if they take a detour or deviate from the path shown by the product.

b) For Administrators:

- An administrator is the one with high privileges
- Main task of an administrator will be the management and updation of users' information, location and routes.
- Administrators can view, modify and delete the personal information and travelling information of users if necessary.
- The product owners can ping the vehicle admin anytime for asking about the whereabouts of the vehicle he is supposed to track and monitor.

4.3.4.4 Operating Environment

- On the server side, the vehicle admin will have a mobile phone with Android OS and Internet facility.
- The driver will have a mobile phone with Android Os and Internet and GPS facility.

4.3.4.5 Design and Implementation Constraints

- Must be coded efficiently enough to run well without using much data.
- Minimum memory required for application :25MB space for each.
- The database will be created and maintained online in a way that makes it of reasonable and manageable size.

4.3.4.6 User Documentation

- User Manual will be provided with Frequently Asked Questions(FAQ's)

4.3.4.7 Assumptions and Dependencies

ASSUMPTION:

- This response time may increase if the network is slow or there is a connection error.
- The time taken for a requested response may vary depending on the location and network strength.

4.3.5 External Interface Requirements

4.3.5.1 User Interfaces

- Driver side : Map with source, destination, checkpoints' markers, current location and route from source to destination passing through all the checkpoints.

4.3.5.2 Hardware Interfaces

The application can be used on a android mobile phone which has the following specifications:

Memory: 25MB or more

GPS, Internet facilities (preferably 3G or more)

Internet is needed for running the application. All hardware are required to get connected to internet for hardware side interfacing.

4.3.5.3 Software Interfaces

This software package is developed using java as the front end. Microsoft SQL server as the back end to store the database.

Operating System: Android 2.3.6 or more.

Languages: Java .

Database: MS SQL server.

4.3.5.4 Communications Interfaces

- It will be connected to the internet, through a GSM device using GPRS.
- GSM messaging Service will be used for communication in case of emergency like an accident.

4.3.6 System Features

4.3.6.1 Accident Alerts:-

- In case of a possibility of an accident, an SMS will be sent on urgent basis to the Vehicle Admin with the current location of the vehicle. Then the vehicle admin is responsible to find out the cause by contacting the driver and taking appropriate actions.

4.3.6.2 Routing:-

- Shortest path will be provided to the driver.
- The shortest path provided will be from a particular source to destination via checkpoints.

4.3.6.3 Speed Monitoring:-

- Speed of the Vehicle is monitored
- If the speed of a vehicle exceeds a permissible limit, he will be alerted to slow down by a sound alarm.

4.3.6.4 Location:-

- The current location of the vehicle will be shown to the user.
- The location of the vehicle will be monitored by another application which is possessed by the vehicle admin.
- The updates of location will be sent at fixed interval of time.

4.3.6.5 Feedback:-

- The user is provided the freedom of expression by giving them the flexibility of suggesting changes in the system but before or after the d-day to the vehicle admin.
- The user is also provided power of rating the service which would help other users.

4.3.7 Other Nonfunctional Requirements

4.3.7.1 Performance Requirements

It is expected that the database will perform functionally all requirements specified.

- The performance of the system should be fast and accurate.
- Responses to view information must not take more than 5 seconds to load on the screen.

4.3.7.2 Safety Requirements

- A backup of the database must be taken from time to time daily ensuring timely recovery in case of any severe loss or a power cut.
- Also information of the users must be stored in the database with proper authorization.

4.3.7.3 Security Requirements

- The database must follow a standard authorization so as to prevent any mis usage of the private information.
- Users need not be aware of other users' account information.

4.3.7.4 Business Rules

- The developer can modify or update the system and its features.
- The client will be able to modify the non-technical part such as adding checkpoints etc. But updating the database, changing authorization etc. will be entirely upon the developer.
- The user will be only able to view and request for information for the vehicle.

4.3.8 Other Requirements

- Other requirements include an ownership of a large memory online for database storage, a faster processor for faster loading.
- Rights for Monitoring of vehicle should be taken care of.

4.3.9 Appendix A: Glossary

A: admin, abbreviation, acronym, assumptions

B: business rules

C: class

D: dependencies

F: functional requirement

G: GUI

M: member

N: non-functional requirement

O: operating environment

S: safety, security, system features

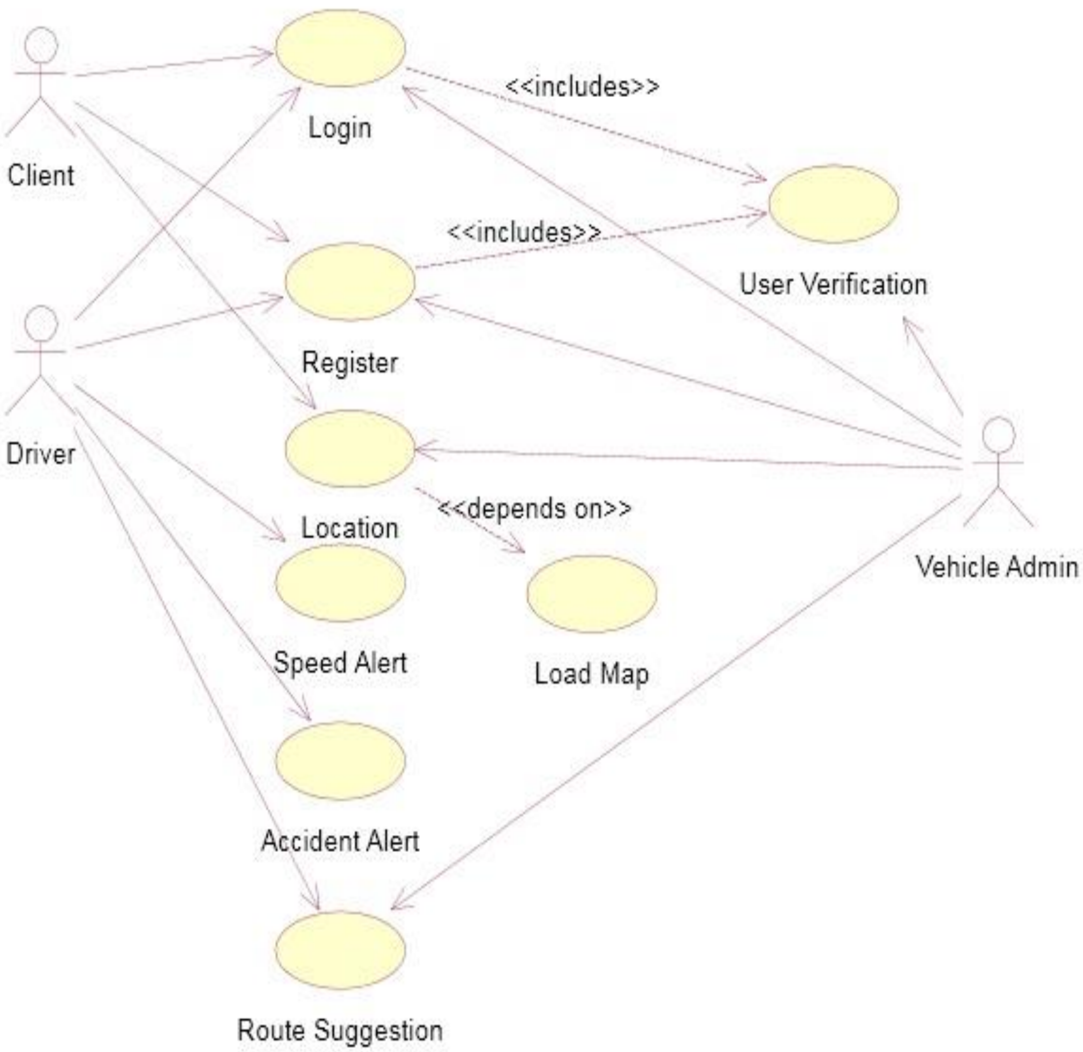
U: user class and characteristics

The following are the list of conventions and acronyms used in this document:-

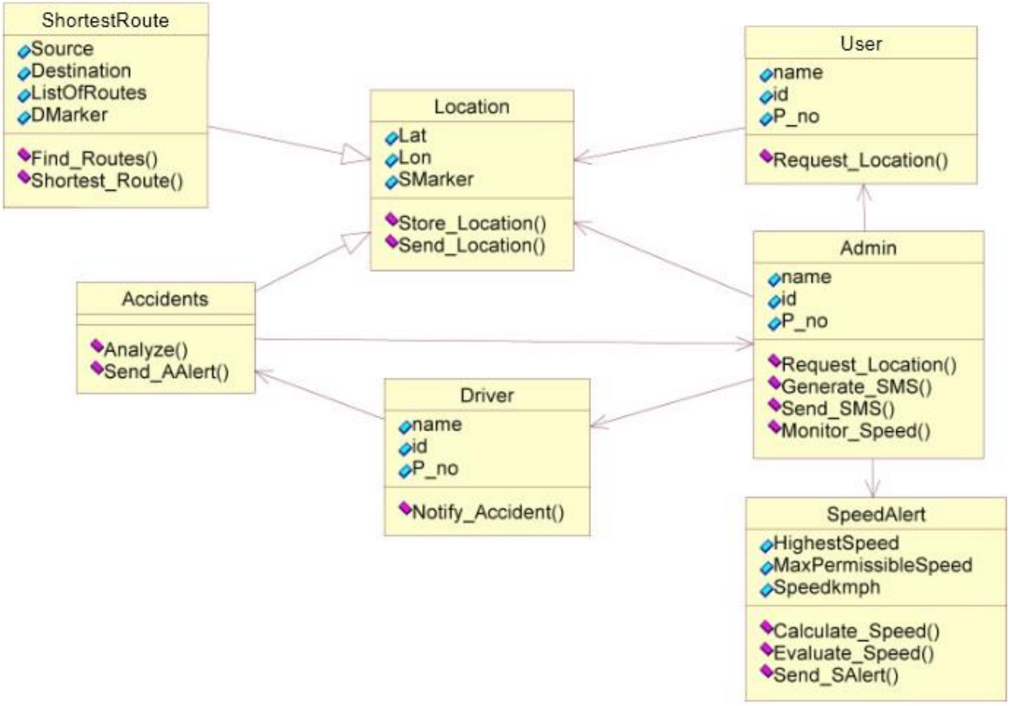
1. SQL: - structured query language; used to retrieve information from a database
2. JSP:-Java Servlet Page.

4.3.10 Diagrams

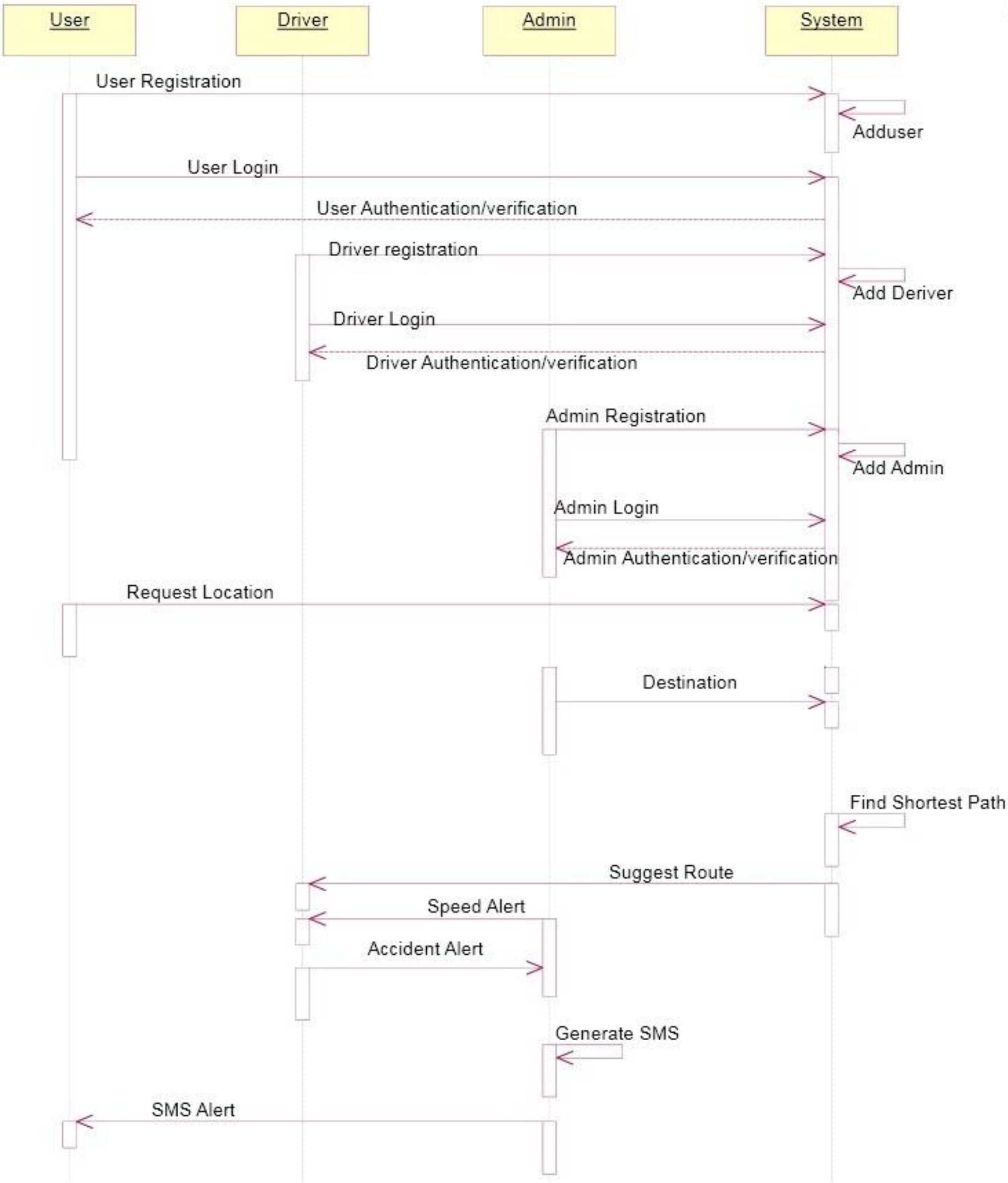
4.3.10.1 Use Case Diagram



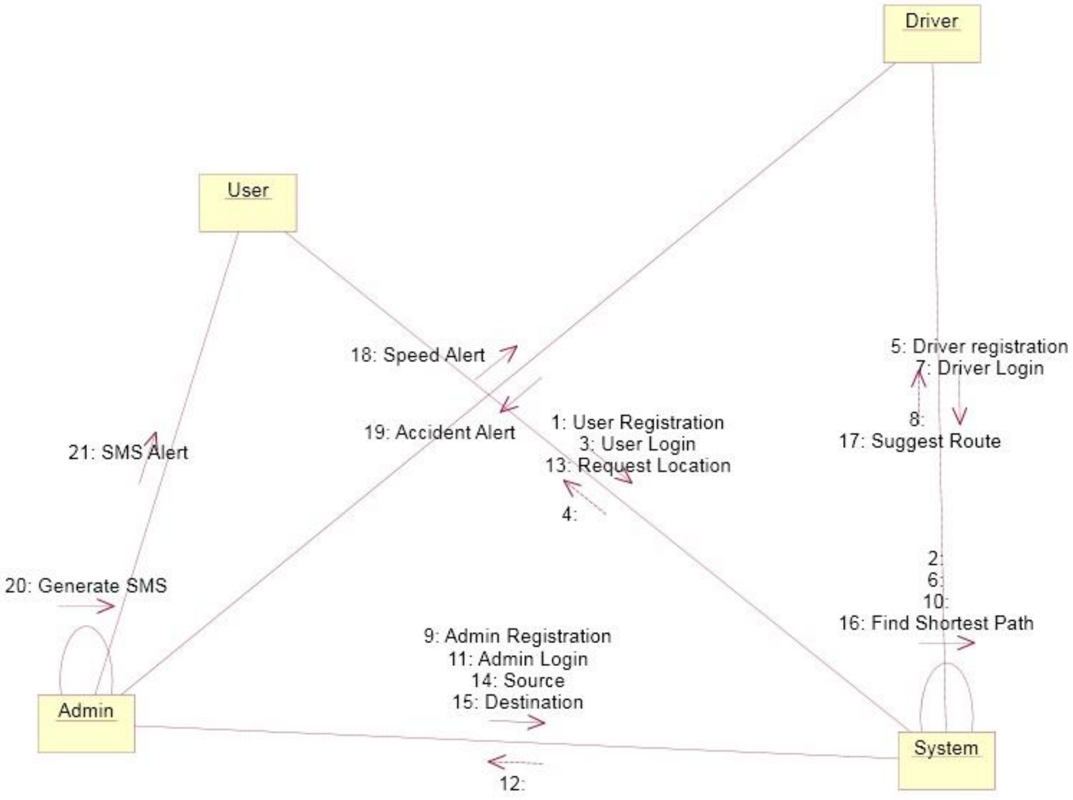
4.3.10.2 Class Diagram



4.3.10.3 Sequence Diagram



4.3.10.4 Collaboration Diagram



4.4 SOFTWARE DESIGN DESCRIPTION

Software Design Document (SDD) attempts to describe the design specifications used in the development of the Vehicle Monitoring And Routing System. It describes the system and architectural design, as well as the user interface design. It also contains requirement traceability matrix that maps and traces user requirement with test cases.

4.4.1 Database Design

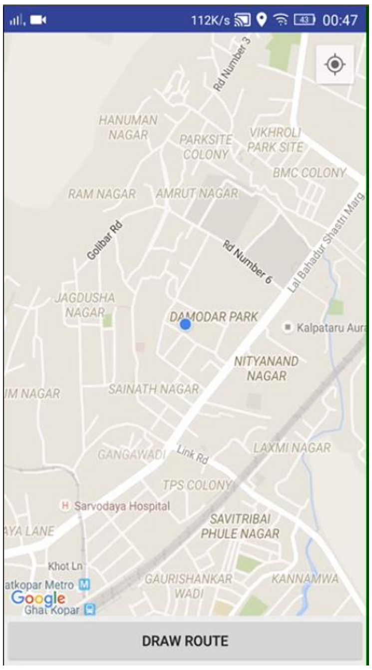
The database for the application is maintained on vmars.pe.hu website and is hosted using Hostinger. The database is MySQL database on PHPMyadmin. It contains two columns, latitude and longitude both of type double. Three files are maintained on Hostinger, init.php which initializes database connectivity, latlong.php which stores coordinate values from the application into the database and get_latlong.php which retrieves coordinate values from the database and passes to the application.

4.4.2 UI Design

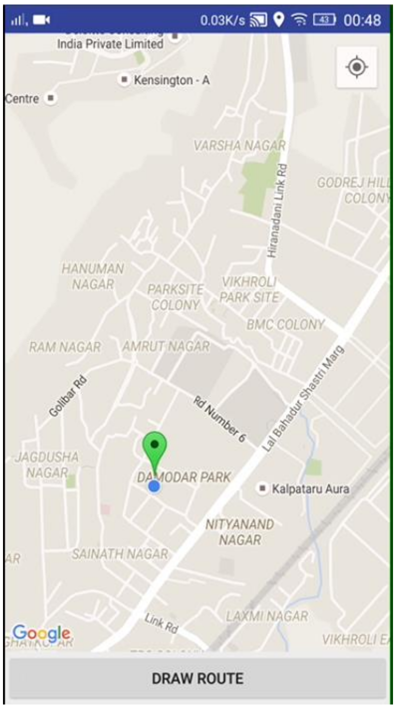
4.4.2.1 Driver Application

- On startup

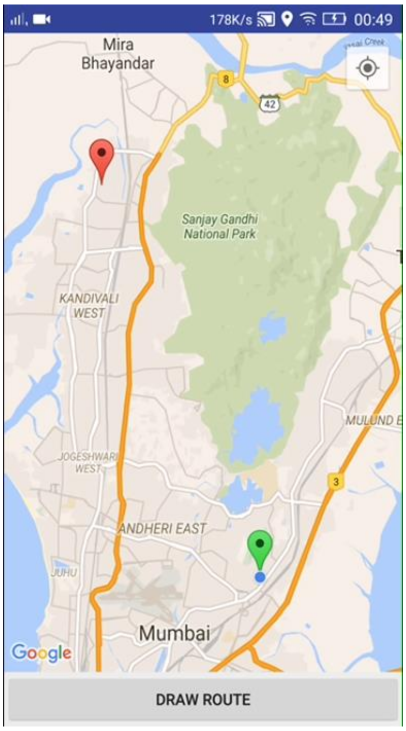
- After tapping on zoom button



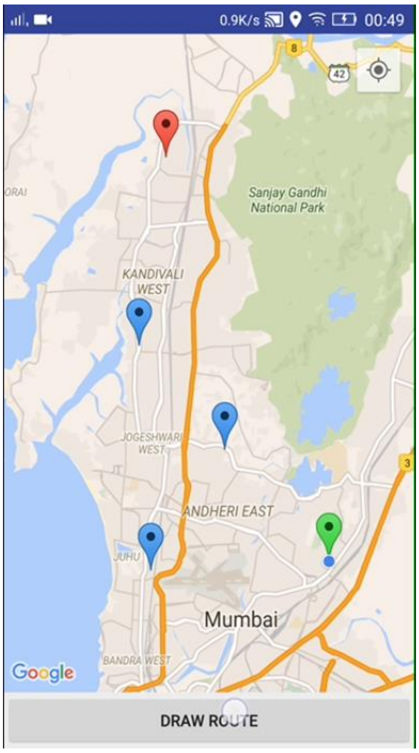
- After selection Source



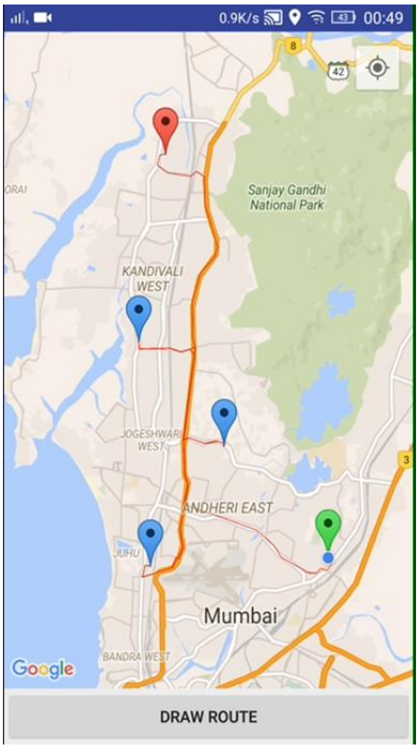
- After selecting destination



- After selecting checkpoints

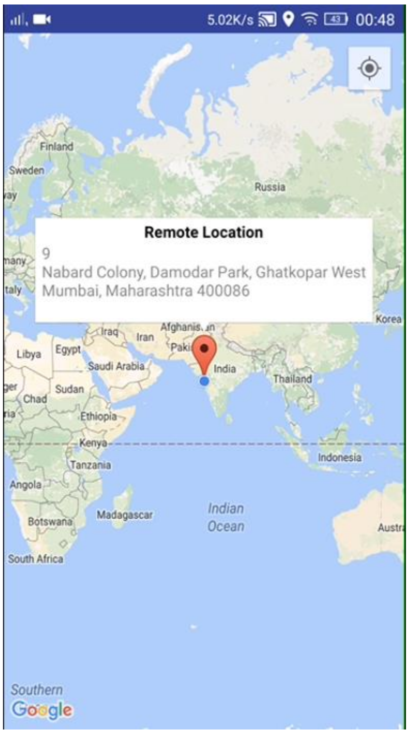


- After pressing “DRAW ROUTE” button

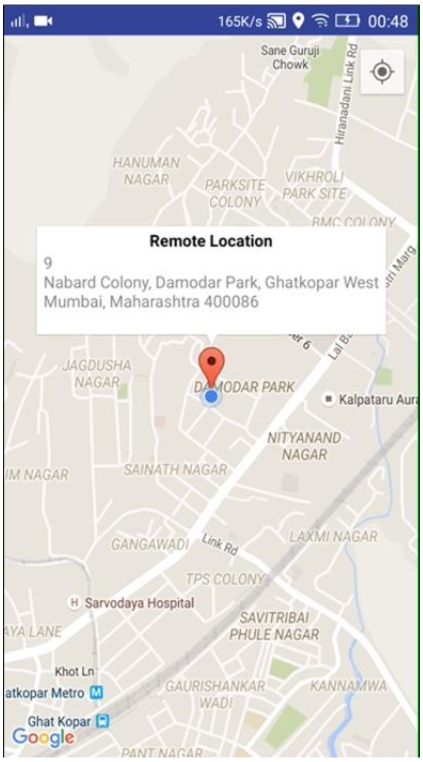


4.4.2.2 Parent Application

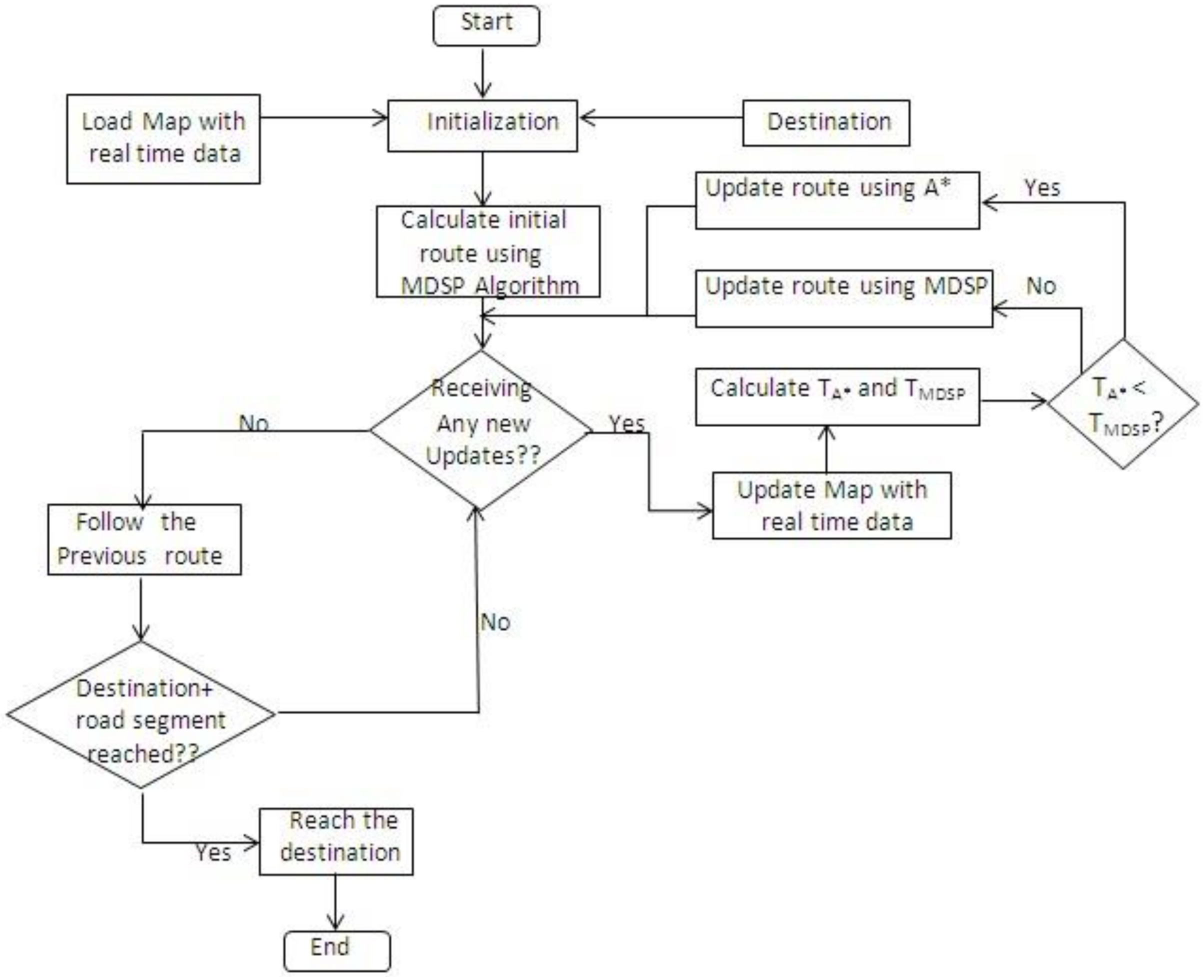
- On startup



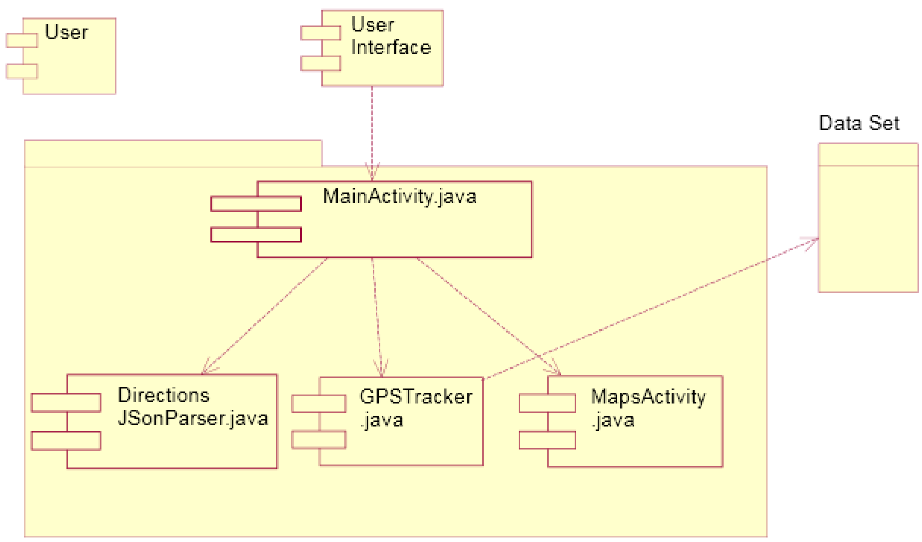
- On pressing zoom button



4.4.3 Procedural Design



4.4.4 Component Diagram



5. PROJECT IMPLEMENTATION

5.1 Main Algorithm

The vehicle routing problem is a different from normal shortest path problem, which are having links that will represent road maps with its junctions.

Many shortest paths algorithms are proposed but according to research work, Dijkstra's shortest path algorithm is the most accurate when there is a single source - single destination problem. Normal Dijkstra's algorithm considers only the weights or distance between the nodes for selection of the shortest path. Taking the real world networks into consideration, there is need to modified original Dijkstra’s algorithm to modified one which is known as “multi-parameter Dijkstra’s algorithm” (MDSP) that considers multiple parameters into consideration. Along with the distance between any two nodes, it considers time factor taken to travel from the source to the destination, congestion at path etc. so that the user can select the desired optimum route based on his/her preferences.

The available time to provide an alternate path must be limited due to road network constraints. It should take very less time to provide the alternative path. Thus, we will classify the existing solutions into two categories and they are : MDSP and Heuristic.

5.1.1 “Modified Dijkstra’s Shortest Path Algorithm” (MDSP)

Researchers proposed a new shortest path algorithm named as “Modified Dijkstra’s Shortest Path algorithm” (MDSP). In this algorithm not only single parameter of weight is considered but multiple parameters need to be considered.

Researchers identified that there was a need to find an efficient shortest path route for the road network. Hence, they developed a new shortest path algorithm by modifying the Dijkstra’s shortest path algorithm. This algorithm shows the better results than the existing Dijkstra’s shortest path algorithm but it take high computational time than the existing algorithm.

This algorithm shows better results than the existing Dijkstra’s shortest path algorithm on real time road network.

This algorithm considers multiple parameters like time, distance and congestion for finding possible shortest route from the source to destination. The algorithm’s pseudo code is as below:

```
// Let source be the origin vertex  and initialize Visited and ShortestDistance[u] as
Visited := {source}
ShortestDistance[source] := 0

// Let user choose any preference among Distance, Time and Congestion factors
ACCEPT Choice

// Update distance between every nodes as per their corresponding factors
FOR each vertex pair [u,v]
        Case Distance: //Do nothing
        Case Time: //Update according to time factor
                Distance[u,v] := Distance[u,v] * time factor
```

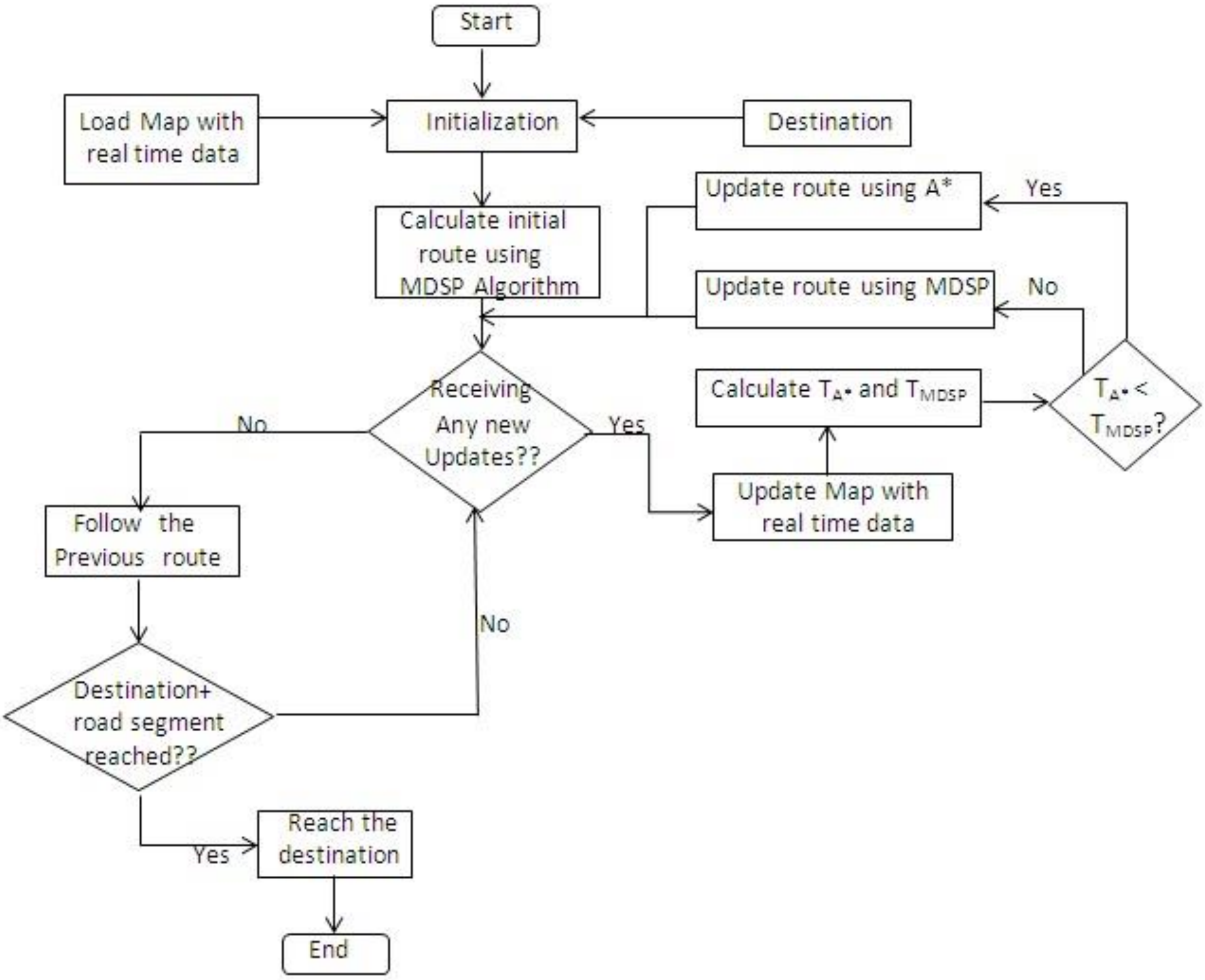
```
    Case Congestion: //Update according to congestion factor
            Distance[u,v] := Distance[u,v] * congestion factor
END FOR

FOR each vertex in V - {source}
    ShortestDistance[u] := Distance[source,u]
END FOR

//Add vertices to Visited until Visited includes all vertices in V
WHILE Visited not equal to V

 // Find the vertex w among remaining vertices closest to the source
    MinimumDistance := INFINITE

    FOR each v in remaining vertices

        IF ShortestDistance[v] < MinimumDistance
            MinimumDistance = ShortestDistance[v]
            w := v
        END IF
    END FOR

    // Add w to Visited list
    Visited:= Visited union {w}

    // Update the minimum distance to vertices in remaining vertices

    FOR each u in remaining vertices
        ShortestDistance[u] :=
        Minimum of (ShortestDistance[u],ShortestDistance[w] + Distance[w,u])
    END FOR
END WHILE
```

5.1.2 Heuristic Algorithms (HA)

The A* algorithm is used to integrate a heuristic function into a search procedure. Instead of selecting next node with the least cost (which is measured from the start point), the selection of the node is based on the cost from the start node plus an estimate of proximity to the destination (a heuristic estimate). This project uses Euclidean distance as estimated distance to the destination. In the searching, the cost of a node V could be calculated as:

The proposed strategy is a combination of both MDSP and A* which helps in meeting requirement of dynamic time constraints of real road traffic scenarios.

5.2 Flow Chart:



- Initially, the shortest path is found using Modified Dijkstra's Algorithm. On receiving any new updates like new route, or change in congestion, or blockage of a road segment, an alternate route is calculate using A* Search Method and travel time of both the approaches are compared.
- The route with lower travel time period is chosen and route is updated accordingly.
- This process is repeated till the destination road segment is reached.

5.3 Programming Language used for Implementation

Java programming language is used for coding the android application

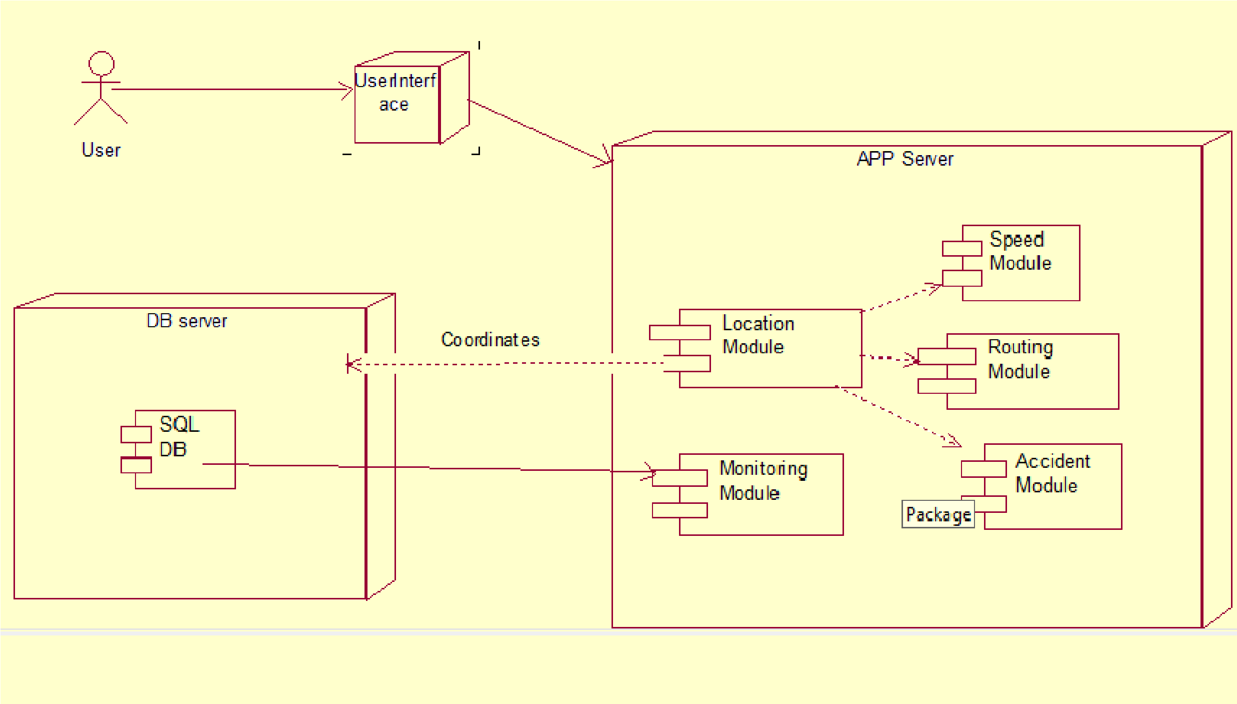
PHP scripting language is used for posting and extracting location values to and from the online database.

5.4 Tools used

For developing android application, Android Studio 1.5.1 was used.

For debugging of the application, mobile phone with Android lollipop v5.0 was used

5.5 Deployment diagram



6. INTEGRATION & TESTING

6.1 Testing Approach

The testing approach that was followed in the project began at the component level and this testing was worked outwards toward the integration of the entire system.

Testing approach was an umbrella activity in the development of the system. Each module was tested at its completion before transitioning to the next component.

At the end of each component, these different components were integrated and testing was done to check the working of the integrated system.

- Unit testing:

The significance of Unit Testing in this project was prominent since the algorithms had different components which needed to be run and tested on their own for the algorithms to run smoothly and give the most optimum output. The plan for conducting unit testing was as follows:

i. Conduct unit testing for each functional components like location and speed module

ii. Testing the application when the GPS and internet was off while opening the application.

iii. Providing error messages or warning messages to user for turning on GPS and Internet.

iv. For speed module, the ringing bell sound was tested when the user crosses predefined threshold speed.

Test Case No.	Threshold	Observed Speed	Alarm Rings?
1	40kmph	16kmph	No

2	40kmph	36kmph	No
3	40kmph	57kmph	Yes
4	30kmph	40kmph	Yes
5	30kmph	27kmph	No

v. For accident module, the system was tested if the vehicle doesn't move for predefined time; it is considered that there might be chance of an accident.

vi. All interface testing was done.

- Integrated testing:

Integration testing is used to check integration of different components. Tests were done after integrating different modules as we progressed. For instance, once binning was added to the system, a test was done to check the working of the system with knn algorithm and binning algorithm together. We check for data integrity. We must make sure that the results are not absurd. The test result provides a clear picture of the integrated functioning of components.

Testing schedule:

6.2 Testing Plan

- **Define testing strategy**
  Examine your application, System environment, testing resources in order to determine your testing goals.

- **Define test subjects**
  Divide your application into modules or function to be tested.

- **Define tests**
  Determine the types of tests you need for each module.

- **Create coverage**
  Link each test with the testing requirements.

- **Design test steps**
  Develop manual tests by adding steps to the tests in test plan tree. Test steps describe the tests operations, the point to check and the expected output of each test.

- **Automate tests**
  For each test that you decide to automate, create test scripts with a third party testing tool.

Test Title	**Date**
Dijkstra's Algorithm	27th July 2015
MDSP Algorithm	29th August 2015

A* Algorithm	4th September 2015
Proposed Algorithm Testing	18th November 2015
Location Module	30th October 2015
Route Module	12th December 2015
Integrated tests for additional features of routing with proposed Algorithm and checkpoints	15th January 2016
Speed Module and its integration with the application	15th March 2016
Accident Module and its integration	20th April 2016
Interface flow test	10th May 2016

6.3 Unit Test Cases

Unit test cases were conducted on each of the algorithms. The values were recorded

Table 1. Record:1 Dijkstra's Algorithm

Algorithm	Distance	Travel Time
Charni Road-Ghatkopar	20 km	51 minutes
Borivali - Kandivali	5.8 km	26 minutes
Mumbai - Mahad	156 km	3 hours 42 minutes

Table 2. Record:2 MDSP Algorithm

Algorithm	Distance	Travel Time
Charni Road-Ghatkopar	23 km	44 minutes
Borivali - Kandivali	5.9 km	24 minutes
Mumbai - Mahad	169 km	3 hours 27 minutes

Table 3. Record:3 A* Algorithm

Algorithm	Distance	Travel Time
Charni Road-Ghatkopar	21 km	49 minutes
Borivali - Kandivali	6.1 km	25 minutes

Mumbai - Mahad	171 km	3 hours 24 minutes

6.4 Integrated System Test Cases

Algorithm	Distance	Travel Time
Charni Road-Ghatkopar	23 km	44 minutes
Borivali - Kandivali	5.9 km	24 minutes
Mumbai - Mahad	171 km	3 hours 24 minutes

7. CONCLUSION & FUTURE WORK

7.1 Conclusion

After successful implementation of the proposed project and deployment in the form of an android application, it will be much beneficial for the parents of school-going students. Right from the factors like safety of the students to the satisfaction of their parents, this system can be used by a school for giving assurance to the parents as it can monitor real-time position of the buses and even ensure that the driver does not over-speed. Also, the proposed system detects the occurrence of an accident with high probability so that immediate actions can be taken by the vehicle admin by looking into its details.

7.2 Future work

This project can be extended to provide all the facilities to multiple monitoring agencies and multiple drivers associated with each agency by providing secure login interface for each user.

References

- The information related to format of project is referred from
- [1] Dr. C.Chandrasekar, S. Sivakumar, "*Modified Dijkstra's Shortest Path Algorithm* " IJIR in Computer Engineering (An ISO 3297: 2007 organization)
- [2] Liang Dai, "*Fast Shortest Path Algorithm for Road Network and Implementation*" Carleton University School of Computer Science, 2005 COMP 4905
- [3] A.Renugambal , V.Adilakshmi Kameswari ,"*Finding Optimal Vehicular Route Based On GPS* " ,IJCSIT, Vol. 5 (2) , 2014
- [4] Shen Wang 1, Soufiene Djahel 2 and Jennifer McManis1 , "*A Hybrid Vehicular Re-routing Strategy with Dynamic Time Constraints for Road Traffic Congestion Avoidance*"
- [5] Nishtha Kesswani,Dinesh Gopalani ,"*Design And Implementation Of Multi-Parameter Dijkstra's Algorithm* ,Ijaer 2011, September

Appendix

A: admin, abbreviation, acronym, assumptions

B: business rules

C: class

D: dependencies

F: functional requirement

G: GUI

M: member

N: non-functional requirement

O: operating environment

S: safety, security, system features

U: user class and characteristics

I) Minimum System Requirement

- The application runs of mobile phone with Android OS v2.3.6 or more.
- The phone running the driver side application should have a good internet connection, GPS facility and enough balance to send SMS'
- The phone running parent side application application should have a good internet connection.

II) User's Manual

1. Overview

VMARS consists of two applications viz VMARS Driver and VMARS Parent which primarily help to keep track of whereabouts of the vehicle.

2. Install applications

Install the driver application in the android device of the driver who will be driving the vehicle which should have a good internet plan (preferable 3G or more) and GPS.

Install the parent application in the android device of the vehicle admin who is appointed by the monitoring agency for keeping track of the whereabouts of the driver.

3. Pre-requisites

Before running the driver application, GPS should be on and internet connection should be established. The parent application should be run after keeping internet connection established.

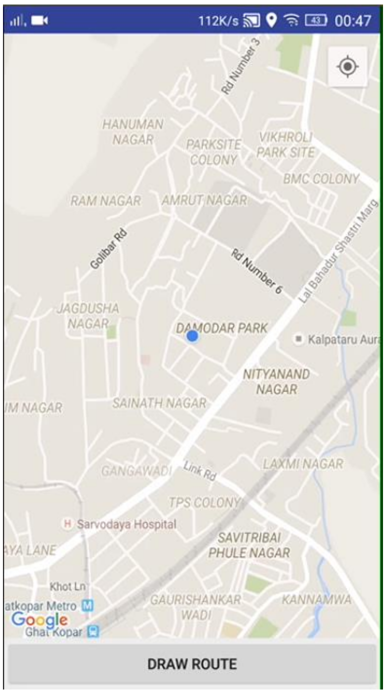
If these conditions are not met, the application may not work as expected. The application may suggest the user to turn on data or GPS if they are not turned on.

4. Routing

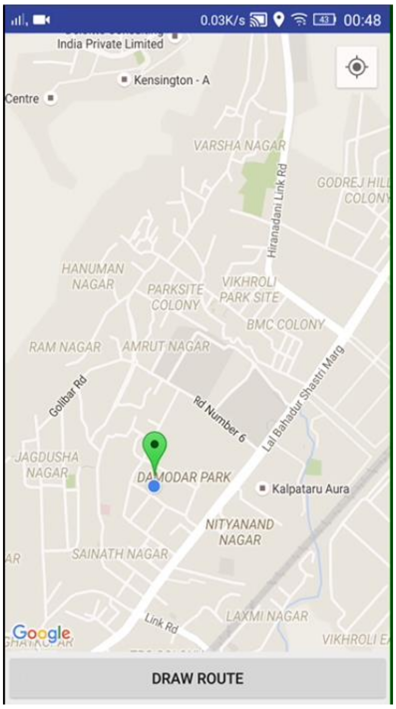
Step 1: Open the driver application.

Note: The position of the driver will be stored to the database as soon as the driver application is opened.

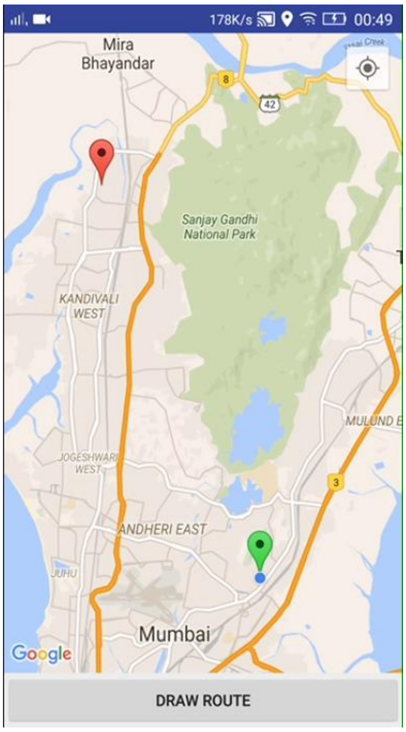
Step 2: Click on the zoom button  for zooming to the driver's location.



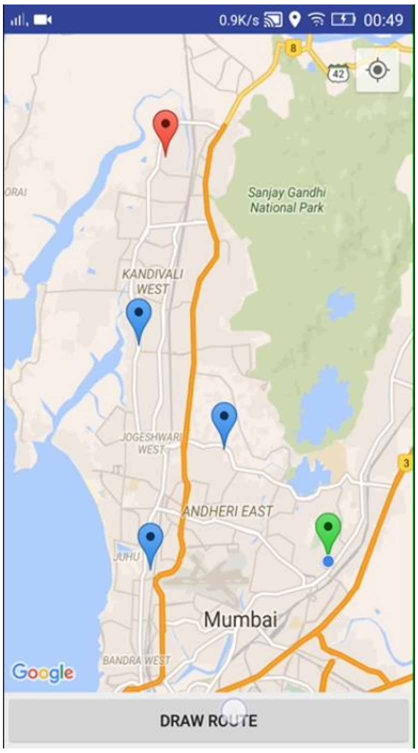
Step 3: Tap on the map the place where driver wants to select as source of the journey.



Step 4: Tap on the map the place where driver wants to select as destination of the journey.

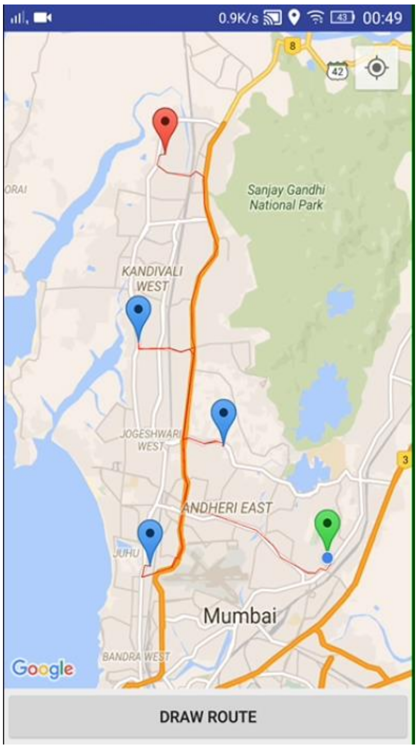


Step 5: Tap on the map the places where driver wants to select as checkpoints of the journey i.e. the places through which the driver needs to pass while travelling from source to destination.

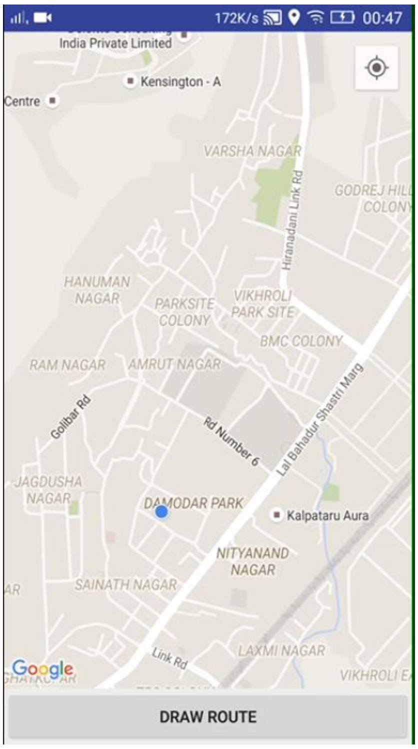


Note: A maximum of 8 checkpoints can be selected on the map.

Step 6: When source, destination and all the checkpoints (optional) are selected, route can be drawn on the map by pressing the “DRAW ROUTE” button.



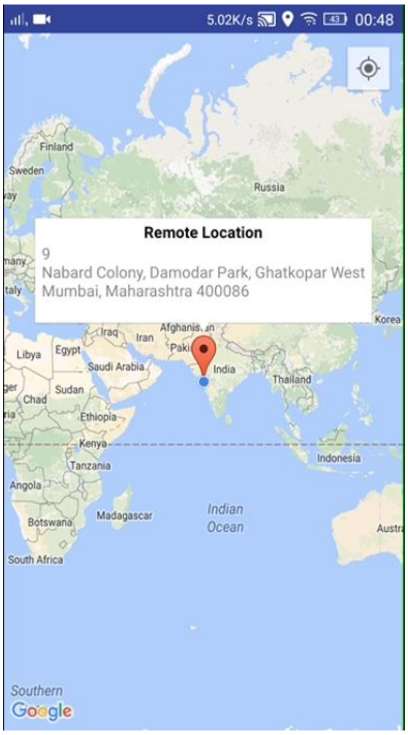
Step 7: Long press on the map to reset all the markers.



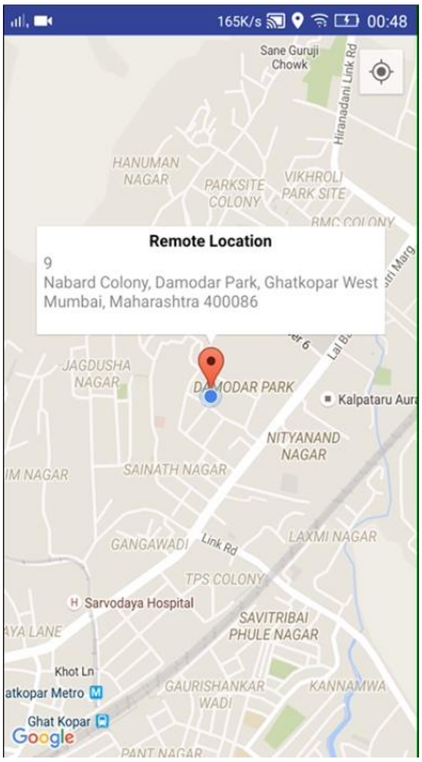
Warning: Once the map is reset, the previous markers cannot be recovered.

5. Monitoring

Step 1: Open the parent application.



Step 2: Click on the zoom button  for zooming to the driver's remote location.


165K/s 00:48
Remote Location
9
Nabard Colony, Damodar Park, Ghatkopar West
Mumbai, Maharashtra 400086
Sane Guruji Chowk
Hiranadani Link Rd
HANUMAN NAGAR
PARKSITE COLONY
VIKHROLI PARK SITE
BMC COLONY
JAGDUSHA NAGAR
DAMODAR PARK
Kalpataru Aura
NITYANAND NAGAR
SAINATH NAGAR
GANGAWADI
Link Rd
LAXMI NAGAR
TPS COLONY
Sarvodaya Hospital
SAVITRIBAI PHULE NAGAR
Khot Ln
GAURISHANKAR WADI
KANNAMWA
Ghat Kopar
PANT NAGAR
Google