National University Of Computer and Emerging Sciences

Route Optimization for Field Agent's Journey Plan in Sales and Distribution
By Techlogix

Muhammad Zubair 15L-4034
Muhammad Mujahid Iqbal 15L-4105
Muhammad Zain Jaffery 15L-4125
Hafiz Nokhaiz Ali 15L-4347

External Project Advisor (Techlogix):
Mr. Mehran Khizar

Internal Project Advisor:
Dr. Kamran Lodhi

B.S. Computer Science
Final Year Project: 2019

Department of Computer Science
FAST NU, Lahore, Pakistan

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the:

Title: **Route Optimization for Field Agent’s Journey Plan in Sales and Distribution(By Techlogix)**is the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as **Plagiarism.** All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: _______________

Student 1

Name:Muhammad Mujahid Iqbal

Signature: ______________________

Student 2

Name:Zain Jaffery

Signature: ______________________

Student 3

Name:Muhammad Zubair

Signature: ______________________

Student 4

Name:Hafiz Nokhaiz

Signature: ______________________

Table of Contents

List of Tables

List of Figures

Abstract

Route Optimizer is an online automated solution for route optimization and route planning for a field agent’s journey plan in sales and distribution. Considering some possible factors that affect route planning, this system provides an optimized automated solution that facilitates the field agents and their manager at real time. This project is based on Mobility, Web and Artificial Intelligence Algorithms. In this document, presented are some of the popular literature reviews that overlap with our project domain. By going through these research papers, we have analyzed the problem with different aspects and then elaborated the solutions accordingly. The major approaches used are the TSP (Travelling salesman Problem) and VRP (Vehicle Routing Problem) and in order to elaborate the system,the engineering approach is also presented which includes architectural strategies, functionalities and methodology of the system. Furthermore, some APIs are also explored that will allow us to implement the required solution. The system applies genetic algorithm and with the consideration of some route obstructing factors it provides close to minimal solutions according to the nature of GA. From this project we hope to build a reliable and valuable optimization system for the industry.

1. Introduction

1.1 Overview

The solution of route optimization problem is very important for those service providers whose agents have to go in fields to meet with different customer. As it takes time and money to visit customers, it is very important for the agents to traverse through these customers in such an order and following those paths through which resources can be utilized in most efficient way. For companies, it is also very important to decide at run time which agent will go to new customer. For this reason, providing alternate routes is very important. These routes should be optimized considering all the factors that affect route planning.

1.2 Purpose

The purpose of this project is to develop a system that can help users to find the best possible route for travelling between different set of locations at any given time. Its main feature is to enable users to get best possible route considering some factors like traffic and weather etc.
Currently, a field agent is assigned a set of customers to visit daily. The sequence of visiting customers should be performed by Route Optimizer instead of KPO/Manager manually enlisting and sequencing the customers per Field Agent. This Route plan should be optimized in a way that minimum transit time is achieved using the best possible route.

1.3 Goals and Objectives

Our ultimate goal is to come up with a system which would recommend best route according to the provided set of locations to its users on the basis of their available (real time) information and other factors effecting route planning.

1.4 Project Scope

The motive of our project is to lessen the work of manager for providing the route to the sales man for visiting the customers.
Our system will eventually:

- Provide the best possible route plan to field agents.
- Manage multiple agents' route plan at run time.
- Allow to add new co-ordinates of customers at runtime.
- Give information regarding road blockage, weather and traffic etc.
- Show the map with the locations of all the customers.
- Take suggestions from other field agents regarding road blockages etc.

1.5 Elaboration of Problem

Generally, the focus of our problem is that a field agent is assigned a set of locations to visit daily. As of now, the sequence of visiting customers is being done by KPO/Manager manually enlisting and sequencing the locations to visit per Field Agent. This task should be performed by Route Optimizer instead of KPO/Manager.

Our Problem can be broken into sub problems as:

- The route plan should be optimized in such a way that minimum transit time is achieved for **cost saving**.
- Out of all alternate routes, **best possible route** should be selected.
- Route should be judged based on **factors** like traffic situations, weather conditions etc.
- **Road closures** of any type should be tackled by the system at real-time as well as prior to the route definition.
- Best possible route will be assigned to all agents in an **optimized sequence automatically** by the system.

1.6 Outline

As the goal of our project is to find the best route for the given set of locations incorporating different factors. In Literature review some of the applicablealgorithms with several approaches have been elaborated, with some addition exploration of APIs. The the most feasible solution to the travelling salesman problem can be achieved by applying the Genetic Algorithm. Google Map APIs also provide help in thus regard. Later in the requirements chapter the main functional and non functional requirements have been discussed in order to provide the best optimized solution. An overview of GUI and system's working is explained further and in the next session the implementation methods and strategies are discussed. Then after the test cases the detailed analysis of experiments and results is stated followed up by the final conclusion.

2. Literature survey / related work

We have considered some of the important research papers in this literature review section [3], [12], [22], and [21],which lie in our project domain.

There are many problems in the context of route and transport problems. Following are 2 famous routing problems:

1. **Vehicle Routing Problem**
2. **Travelling Salesman Problem**

The first one is defined as sequence of delivery/recovery points in which proper route is to be selected in an order considering some constraints. Salesman problem, on the other hand, is a problem where a sales person needs to visit different nodes or points only once and is required to come back at the starting points in minimum distance leading to reduce cost.
In this chapter the main focus has been done on the variety of algorithms used to solve the route optimization problem.

2.1 Techniques for route optimization

2.1.1 Introduction

This is the review of the paper [21]. Human life required transportation system. People need to get from one point to another all the time. Due to increase in vehicle production, traffic is becoming congested. This congestion causes hectic delays in traffic, increases cost and wastage of time etc. This urges us to try different routes. To avoid traffic delays, alternate routes should be available. An effective route is that which saves time and cost. Shortest path is not necessarily the best as it may be costlier where traffic is congested even if route is shortest. There are multiple techniques for optimizing routes. Most of them uses Graphs and consider one location as a node and distance between 2 nodes as edge. In addition, there are multiple algorithms which can be used for different set of route optimization related problems. All of them can be categorized into two major techniques called hard computing techniques and soft computing techniques. Each consists of different type of algorithms.

2.1.2 Hard Computing Techniques

These methods include approaches like deterministic reasoning and binary logic to deal with factors like uncertainty, precision etc. Some of them include:

2.1.2.1 Dijkstra's Algorithm

This is the groundwork of all algorithms that include shortest path in some way. This algorithm basically calculates the minimum distance from a home point (node) to all available nodes. It traverses the nodes sequentially followed thorough shortest distance from origin [14].

2.1.2.2 A* Search Algorithm

This algorithm is concerned with path discovery and node traversal. It uses heuristic functions to improve time performance in general. It involves some machine learning techniques as well.

2.1.2.3 Arc Flags

As sign posts guide humans to reach an intersection, arc flags can be defined as additional sign posts in a graph. Due to this, these flags change graph into segments and remove single edges from Dijkstra's algorithm so that they are ultimately removed from shortest distance as well. They also add vectors of flags to each edge indicating weather corresponding edge leads to destination or not [15].

2.1.2.4 Contraction hierarchies

These contraction hierarchies are methods such that they introduce shortcuts in the nodnetworks by certifying that the points (nodes in our case) are well-ordered by significance first. After that, a hierarchy is mapped by selecting and contracting the lowest priority node, iteratively. If we contact a node M, then this means we are changing the shortest paths which are passing through shortcuts [16]. These hierarchies produced due to contraction assign a priority or significance level to every node.

After the process of contraction is done, , the nodes are then contracted in a hierarchy based on priority by eliminating them totally from the graph and also swapping shortcuts to save the shortest-path distances linking more highest priority nodes [14].Soft Computing Techniques

These techniques are also called intelligent route optimization. They differ from usual analytical algorithms. They utilize computational methods to represent uncertain and vague concepts etc. Following are some of them that can be used for route optimization.

2.1.2.5 Fuzzy Logic

This type of algorithm is an addition of boolean algebra which caters concept of incomplete/partial truth i.e. the static values between totally true and totally false. This method operates on the logic of estimation rather than precision. It has emerged from human nature which works by approximation.

2.1.2.6 Artificial Neural Network (ANN)

It is also called neural network. It works on the biological principal of human nervous system. Its most basic unit is a neuron, which works in parallel to resolve problems. Key feature of ANN is that it learns by examples. We fed it some examples then it fine tunes itself to form accurate relations between neurons [17].

In actual, each property in any problem is assigned weight randomly. Then we give some accurate and tested examples to ANN, and then it starts to learn from those examples and fine tune those weighs so that those weighs also give accurate results.

2.1.2.7 Genetic Algorithms (GA)

This method resolves around human evolution paradigm. In it, bits of strings are used as a basic unit called a chromosome. It has a specific sequence at random. We then form more of such chromosomes and then do crossover with over chromosomes to form mutation. In this way, chances of more accurate results improve after each iteration called generation. Accuracy of each chromosome is tested by a fitness function, which in our case will be an optimized route [18].

2.1.2.8 Ant Colony Algorithm

This algorithm is an exceptional algorithm used for determining route which is optimal based on the behavior of ants which try to seek food. This type of algorithm feats the behavior of the real-life ants when they are searching for food. Deep studies tell us that ants throw a tiny quantity of pheromone (a chemical), on its way as it starts travelling from nest to the source of food. Upon their return, ants are required to track the same exact way which is noticeable by the pheromone traces and also drop more of it along the way back. Those ants which are routing through the shortest path are expected to arrive much earlier and consequently boost the amount of pheromone drops in their way at a faster rate than those ants who are taking a longer route. Using these rules, shortest and optimized paths are found. Travelling salesman problem can be solved using this approach [19]Agent Based Software Engineering A.K.A ABSE

This algorithm is a combination of software engineering and agent based-computing. It is just the application of agent-based approach in developing softwares. Agent based computation is defined on the principle of real life agents in which a specific agent is assigned a task and the other employee just requires that agent should complete the task in any way as desired. In this approach, agent is to perform independently. Only results are expected. As this approach is new, there is not sufficient data to declare it best or worst. Wide-ranging literature on the correctness of ABSE could be found at [20]. This approach works on the principles of decomposition (breaking larger problem into sub parts and dividing tasks between agents), abstraction, organization (collaboration of different methods and tasks of agents), interaction (between agents) ABSE encourage agents to take up different roles in route optimization to monitor and report status, traffic, events and distance of road network in order for another agent to make choice on route to deploy. This technique allow for dynamic approach to solve shortest path problem where multi way and multi scenarios are involved.

2.1.3 Conclusion:

Upon analyzing different algorithms, it can be concluded that when we are given enough amount of traffic data and details, then hard computing algorithms can be used like Dijkstra and A* algorithm etc. can be used .But since traffic situations involves dynamic data collection where we need to predict most of the time rather than hard coded result, we will use soft computing techniques in that case like ANN, Genetic algorithm etc. A mixture of these two can also be used (ABSE) which uses static parameters like assigning a specific job to specific agent and also uses dynamic parameters like using neural networks to predict traffic blockages etc.

2.2 Research on the optimization of vehicle distribution routes in logistics enterprises

2.2.1 Introduction:

This is the review of the paper [3].Route optimization is very important in industry of logistics supply. In logistics when a vehicle leaves the start point than usually it has to stop at many pointson the way and supply things. Now in which order these points would be covered is important. If these points are covered in efficient way then money and time can be saved. As logistics companies usually have number of vehicles, they also have to manage the path of each vehicle at distinct points.

Main goals to be achieved:

- Reduce Time
- Make efficient operation
- Reduce Distance
- Reduce cost

2.2.2 VRP Analysis

In this segment some of the VRP solutions are highlighted which majorly include

2.2.2.1 Saving Algorithm

1. Select one point as start point and end point as well we will call it as "depot".
2. Calculate cost of joining two customers and eliminating route back to our depot.
3. Use these saving to form a route.
4. Do not break any links that were formed earlier. In this step stop when all customer are on route [5].

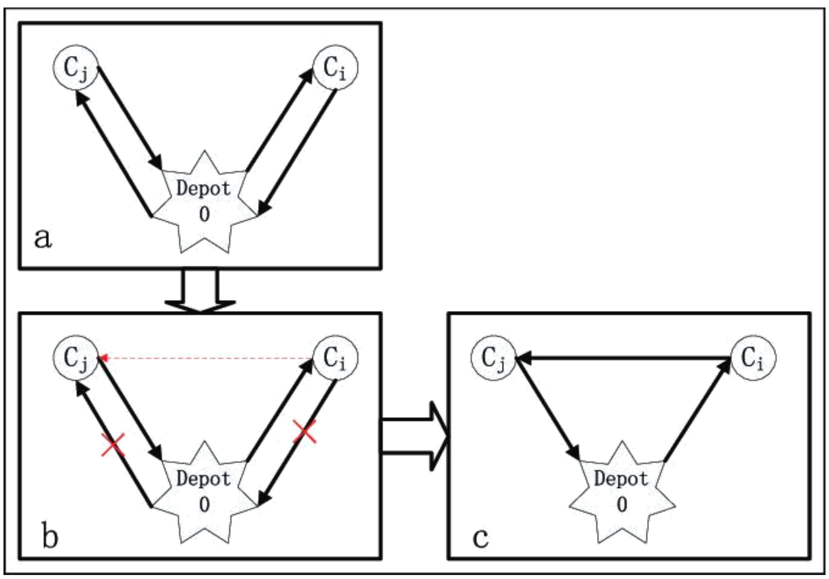

Figure 1: Working of Saving Algorithm [4]

2.2.2.2 Sweep algorithm:

1. Take a center point or depot.
2. Assume each vehicle has a certain capacity.
3. Draw a ray starting from depot.

4. Go clock wise or anti clockwise and add customer to route.
5. Start a new route when vehicle cannot serve more customers.
6. Re-optimize each route.

2.2.2.3 Value insertion method:

It works on.

- Shortest distance
- Least costly concept.

2.2.2.4 Summary of VRP algorithms

	Saving algorithm	**Sweep algorithm**	**Insertion method**
Algorithm used	VRP	VRP	VRP
Use of heuristics	Yes	No	Yes
Considerations in adding point in route	Saving Cost (time, distance)	Capacity of vehicle	Shortest distance
Use of clock wise and anti-clock wise Technique	NO	Yes	No

Table 1: Comparison of VRP Algorithms

2.2.3 Conclusion:

As in our project at this point of time we are considering that we are sending our agents on same type of vehicle (usually bike) so limitation of or capacity of vehicle is not important so among these algorithms, thesaving algorithm and insertion algorithm will be more considerable.

2.3 Redesign of the supply of mobile mechanics based on a novel genetic optimization algorithm using Google Maps API

2.3.1 Introduction:

This is the review of the paper [12]. Route optimization is a general form of the TSP (TravellingSalesman Problem) and VRP (Vehicle Routing Problem). In this paper a novel approach has been devised to solving the Multiple Traveling Salesman problem (mTSP) using Multi-Chromosome Genetic Algorithm.

The main objective of the system was to reduce the financial loss and enhance the service level of the mobile mechanics that had to collect the tools from warehouses and provide the services at different locations keeping under consideration some additional constraints such as limited travelling distance of each salesman, time window and the wage criteria.

2.3.2 Methodology:

A Google map API can map the locations on the basis of provided longitude and latitude vales (in the form of an excel file) of the nodes. By which the distances between the nodes can be obtained (in terms of length or time) which are further used as heuristics (genes for each chromosome) and then the genetic algorithm is applied which gives us the optimized routes for each salesman. (MATLAB and web based technologies were used to develop a framework in order to generate an automated system to invoke the ultimate optimized route for each salesperson.)

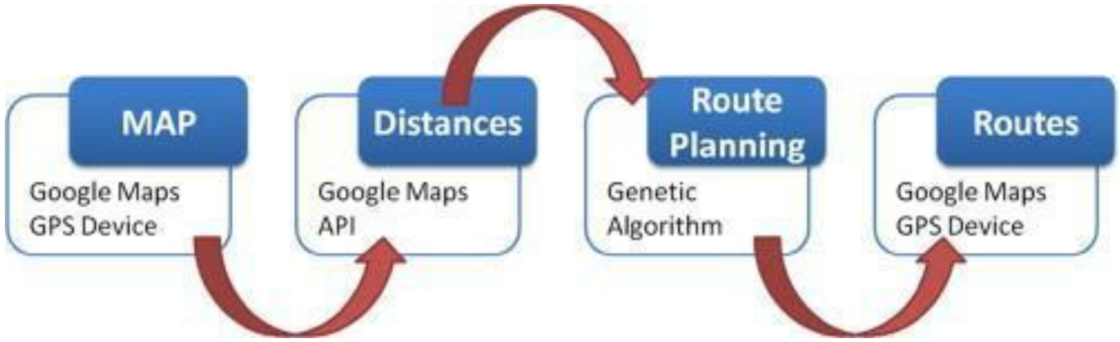

Figure 2: Methodology of Implementation [12]

The objective cost function for each salesman is defined as:

$$\text{minimize} \sum_{k=1}^{m} (E_k + \lambda \cdot \max(E_k - S, 0)) + m \cdot c_m \quad [12]$$

Where

- **Ek** represents the arc length between 2 nodes (if includedin the kth salesman's tour).
- **S** represents the maximum length of any tour in the solution.
- **λ** represents the degree of the punishment.
- **m** represents the feasible number of salesman in the route plan.
- **Cm**represents the cost of the involvement of the salesperson.

This function also includes a penalty for the salesperson that exceeds the maximal tour length.

2.3.3 Multi-chromosome genetic algorithm:

In this technique, each salesman is taken as a separate chromosome and the genes representing the distances (on the basis of duration or length) of that salesman to different locations (nodes) for a single tour.
Five phases are considered in a genetic algorithm.

- **Initial population:**

  Firstly, the initial population is generated consisting of chromosomes with randomly arranged genes.

- **Fitness function:**

  The fitness function is actually taken as the sum of overall path distance of each salesperson inside an individual (chromosome) with a fix number of cities per route.

- **Selection:**

  The chromosome with the least fitness value is then selected (further iterated) for the production of next off springs.

- **Crossover and Mutations:**

  Several operators for the novel representation are implemented. Some of the important ones are.

  - *In-route mutations*: i.e. Mutations within a chromosome e.g. genes sequence flipor swapping genes etc.
  - *Cross route mutations*: i.e. Multiple chromosomes are updated in one go.

  In order to reach the maximum optimality, more complex operators were applied (mostly cross route mutations)

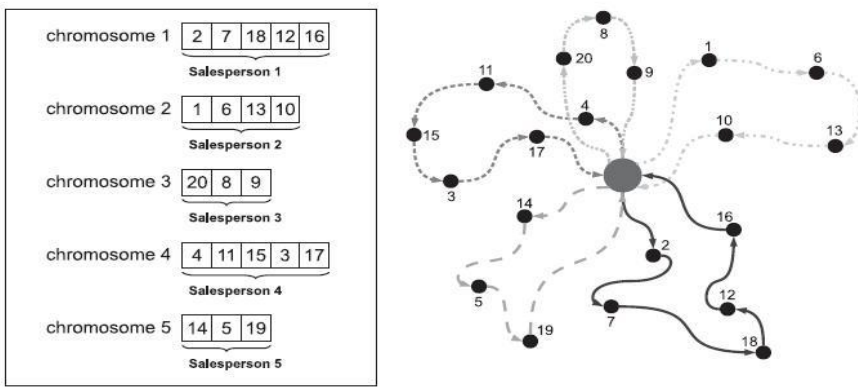

Figure 3: Example of multi-chromosome representation for a 20 city mTSP with 5 salesperson

2.3.4 Analysis and Conclusion:

The numerical analysis depicts that this novel multi-chromosome method provides us a cheaper and quick solution as compared to the methodologies of genetic algorithm applied previously for such (TSP) problems. This technique can converge to the optimized solution faster. As in the testing example, multi-chromosome approach needed only **7411** iterations to find the value 996, whereas the other technique required **18,861** iterations to reach that value.

2.4 Android Application of Traveling Salesman Problem Using the Hybrid Genetic Algorithm (GA and Local Search) on Google Maps

2.4.1 Introduction:

This is a review of paper [22]. TSP is the NP-hard problem means that every time it will not provide optimal solution and there is no exact algorithm which can solve this problem in polynomial time. We can obtain optimal solution by exhaustive enumeration and evaluation. To solve this problem, we have to get all the routes and then we will select the route which has shortest length.

A Genetic algorithm is well known search techniques in the field of computer sciences which can find optimal solution of problems (including TSP). In Genetic algorithm we have an initial population of solutions after applying GA operators (Fitness function and mutations) best population maintained and we get an optimal solution or near to optimal solution after further evaluations. Genetic Algorithm has some problems one of which is local optimum if GA converges to local optimum rather than global optimum we will not get optimal solution. There is another problem with GA that it cannot search the whole space. So to overcome these hazards they use hybrid of GA and local search (iterative hill climbing). They use android because it is portable and assessable easily. They integrate this application with Google API so they get real time experience by showing routes, locations and different stops.

2.4.2 Traveling salesman problem:

The traveling salesman problem (TSP) is an optimization problem and has several applications, such as distribution chain, planning and scheduling. This problem is about find is the best route between different locations (n locations) providing a starting point and every location visited only once and after that return to the starting position again.

2.4.3 Genetic Algorithm:

Genetic Algorithm is an evolutionary algorithm. In start we have a population (solutions of problem optimized or not) on which we apply fitness function from which we check how much the solution is good. After applying crossover mutations we have another population in this way we get our optimal solution or near to it.GA will terminate on some termination condition until this condition did not match GA will run same process again.

2.4.4 Local Search Technique:

They use the iterative hill climbing with GA (Hybrid GA) to solve this TSP. Hill Climbing algorithm search the better solution in its neighborhood. If this algorithm finds a better solution it will update current solution with this better solution. If the newly find solution is not optimum it will update current solution with local optimum solution and stop the algorithm.

2.4.5 Hybrid Genetic Algorithm:

In travelling salesman problem Genetic algorithm did not give optimal route so the better way to find optimal solution is HGA which is use GA with local search (Iterative Hill Climbing).

2.4.6 Experiment:

They perform experiments on five different samples. Each sample includes different number of locations.

- First with 5locations.
- Second with 8 locations.
- Third with 12 locations.
- Forth with 16 locations.
- Fifth with 20 locations.

2.4.7 Results:

No	Data	Number of cities	Distance (meters)					
			GA			HGA		
			Best	Worst	Average	Best	Worst	Average
1	Case1	5	333390	333390	333390	333390	333390	333390
2	Case2	8	518053	595128	533484.6	518053	518053	518053
3	Case3	12	667143	872809	750450.6	622172	719781	679885.6
4	Case4	16	999926	1107099	1060378.4	823899	977451	906274
5	Case5	20	1175422	1314313	1260734.8	1001333	1289746	1185912.8

Table 2: GA verses HGA results [22]

The results show that hybrid genetic algorithm give better route than GA. In 3rd, 4th and 5th experiment HGA will give better results as compared to simple GA. But in 1st and 2nd experiment both results are same. This shows that HGA perform well when data is larger because on small data difference is not prominent.

2.4.8 Conclusion:

In this paper, an application is developed with Hybrid Genetic Algorithm to solve travelling salesman problem and experiments were performed on this application to show that it would be better than simple GA. Results of experiments have proved that HGA perform better means find better solution than GA.HGA perform very well when problem is more complex.

2.5 Review Summary:

Paper	Authors	Area of research	Description
Review of routing problems.	Bale, D. L. T. Ugwu, C. Nwachukwu, E. O	Different Optimization techniques.	This paper discusses history of routing problem, describes different algorithms to solve Optimization problems. Some of these algorithms are Dijkstra’s Algorithm, A* Algorithm, ant-colony algorithm, Ann etc.
Research on the optimization of vehicle distribution routes in logistics Enterprises	Zhigou Fan1 ,Mengkun Ma2	Route optimization.	This paper discusses about vrp algorithms`
Redesign of the supply of mobile mechanics based on a novel genetic optimization algorithm using Google Maps API	Király, A. and Abonyi, J.	Route optimization using Multi-Chromosome Genetic Algorithm.	This paper Utilizes Google maps API and applies A novel technique of GA in finding mTSP solution with Additional constraints.
Android Application of Traveling Salesman Problem Using the Hybrid Genetic Algorithm (GA and Local Search) on Google Maps	Narwadi and Subiyanto	Route optimization using hybrid Genetic Algorithm (hybrid of GA and Iterative Hill Climbing).	This paper compare the result of solution provided by GA and HGA and find that HGA perform better.

Table 3: Summarizing Literature Review

3. Requirements and design

This chapter gives a detailed overview of all the functional specifications of Route Optimization for Field Agent's Journey Plan in Sales and Distribution system. It includes the purpose, intended audience, domain overview and system requirements and the system gui design.

3.1 User Characteristics

Users of Route Optimization for Field Agent's Journey Plan in Sales and Distribution system

- **Sales Agent of supply chain**
  The user will use our system to get path which he have to follow for sales and distribution.

  User characteristics are as follows.

  i. The user should be computer literate.
  ii. The user should be of at least 18 years old.
  iii. The user can be preferably male.
  iii. The user should be able to understand English language.
  iv. The user should be able to use our system in any environment having internet access.

3.2 Domain Overview

The solution of route optimization problem is very important for those service providers whose agents have to go in fields to meet with different customer. As it takes time and money to visit customers, it is very important for the agents to traverse through these customers in such an order and following those paths through which resources can be utilized in most efficient way. For companies, it is also very important to decide at run time which agent will go to new customer. For this reason, providing alternate routes is very important. These routes should be optimized considering all the factors that affect route planning.

Generally, the focus of our problem is that a field agent is assigned a set of locations to visit daily. As of now, the sequence of visiting customers is being done by KPO/Manager manually enlisting and sequencing the locations to visit per Field Agent. This task should be performed by Route Optimizer instead of KPO/Manager.

Our Problem can be broken into sub problems as:

- The route plan should be optimized in such a way that minimum transit time is achieved for cost saving.
- Out of all alternate routes, best possible route should be selected.
- Route should be judged based on factors like traffic situation, peak hours etc.
- Road closures of any type should be tackled by the system at real-time as well as prior to the route definition.
- Best possible route will be assigned to all agents in an optimized sequence automatically by the system.

3.3 Functionality

3.3.1 Functional Requirements

- The system shall use APIs (google maps) to fetch data and compute shortest path b/w locations and keep track of possible conditions. E.g. road blockages, weather etc.
- The agent shall login to his own mobile device.
- The system shall assign the optimized routes before the start of agent's journey.
- Each user shall be shown a map displaying his respective optimized path which he has to follow.
- Agent shall check in to the corresponding location by pressing a button (market visited) and the notification shall be received at the server.
- The user shall be prompted on missing out a location on his way.
- The system will keep track of each location (i.e. Visited/unvisited) and agent at real time.
- The system will broadcast the notification to the specific agent in case of any alteration in his route.
- The system can dynamically add/delete the agents and locations.
- System shall record the agent’s journey time.
- Agents shall navigate through the map for path guidelines.
- Managers shall login to the system.
- Managers shall add all the data related to locations in repository.

3.3.2 Non-Functional Requirements

3.3.2.1 Usability:

Our system shall require no time training for the user (field agent). The agent will be presented a simple interface by which he will view his path and mark the corresponding visited locations.

3.3.2.2 Performance:

Once the routes have been optimized and allotted to each agent, later in case of any conditional changes (e.g. road blockages, etc.), the system shall not re-optimize the entire system, rather in order to enhance the efficiency and performance time, it shall update only the (nearby) effected agents' routes (within seconds).

3.3.2.3 Security:

Each agent shall be isolated from other agents' journey plan.
Map APIs’ keys shall be issued from google reserving all rights and licenses.

3.3.2.4 Compatibility:

At **server side**, our system will be implemented as a web based application and it will be completely compatible with devices which support web access.
At **client side**, the portal shall be completely compatible on any android device (higher than lollipop) and require internet access on each device used by the agents.

3.3.2.5 Reliability:

The system shall handle the addition of new agents and locations dynamically. Our system shall device the best possible solution to traverse a number of locations in an optimized sequence.

3.3.2.6 Extensibility:

Modular approach will be used. So at any time replacing or adding something new would not be an issue. I.e. once before the start of their respective journeys.

3.4 Assumptions

Our system will be based on following assumptions.

- Intended users (**sales agents**) are computer literate and educated enough to operate and understand the system.
- Intended users (**managers**) are computer literate and educated enough to operate and understand the system.
- User data will be locations (longitude and latitude).
- Sales agents have continuous internet connectivity.
- Managers have high speed internet and efficient devices.

3.5 System Architecture

- **Client server**

We should use client server architecture because managers and agents behave as client. Managers add data to the system and agents shall use processed results of our system. System is our server which will perform computations to compute output which would be used by clients later.

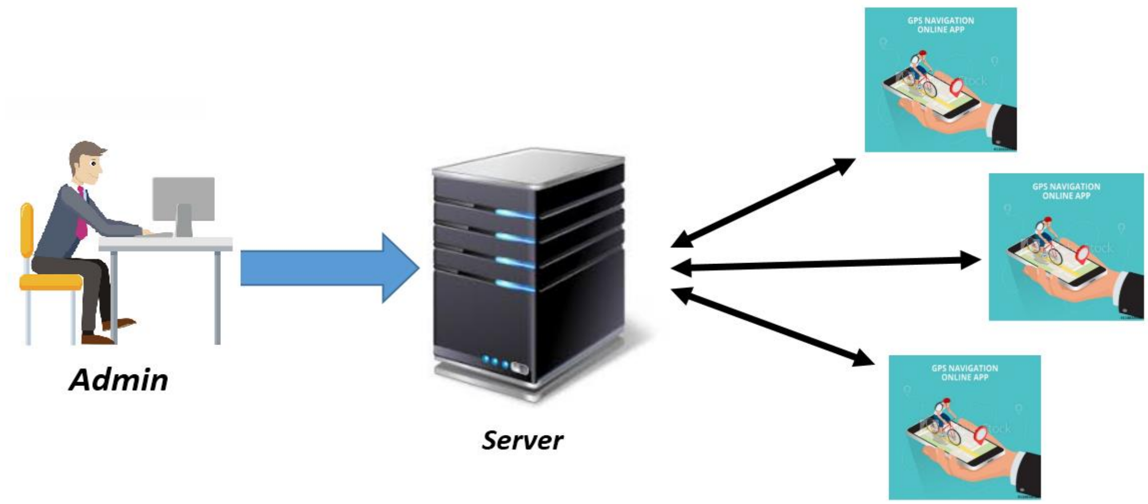

Figure 4: System architecture design

3.6 Methodology

3.6.1 Data entry:

A manager will enter all the locations in a file with longitude, latitude and name of place. Alternatively there would be another way in which manager will enter all the locations with the help of **Google maps** by taking the pin to specific place and entering the name of locations. Manager will do that for all locations.

After this, the file is ready for the use of our system.

3.6.2 System's working:

Our system will use this file to show all customers on map with tags and give all these locations to Genetic algorithm to find the optimal path to visit all these customers.
There is another way in which we will get optimal path from Google APIs without applying GA from scratch. This approach will be used at runtime in case of any conditional change in the path of an agent to maintain the load on the server.

3.6.3 Sales agents input:

Now every sales agent will get their optimal path which they have to follow we also have to record path of sales agent so that we can compare it with optimal path.

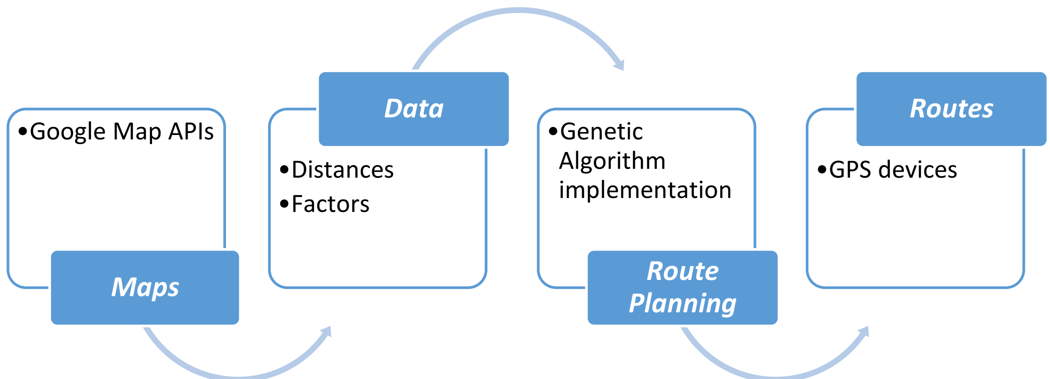

Figure 5: Methodology

Note: the functionalities may alter depending upon the timely industrial requirements (from Techlogix) and availability of resources.

3.7 GUI

3.7.1 Home page

The admin will beshown this (main)page at the start. Who is in charge of selecting the appropriate functionality from this page. i.e Get Path, Optimized path, fetch data, agent details.

- On selecting **Optimized path,** admin can get the complete optimized solution on the basis of locations fetched through the file (add file) and number of agents (from agent details).

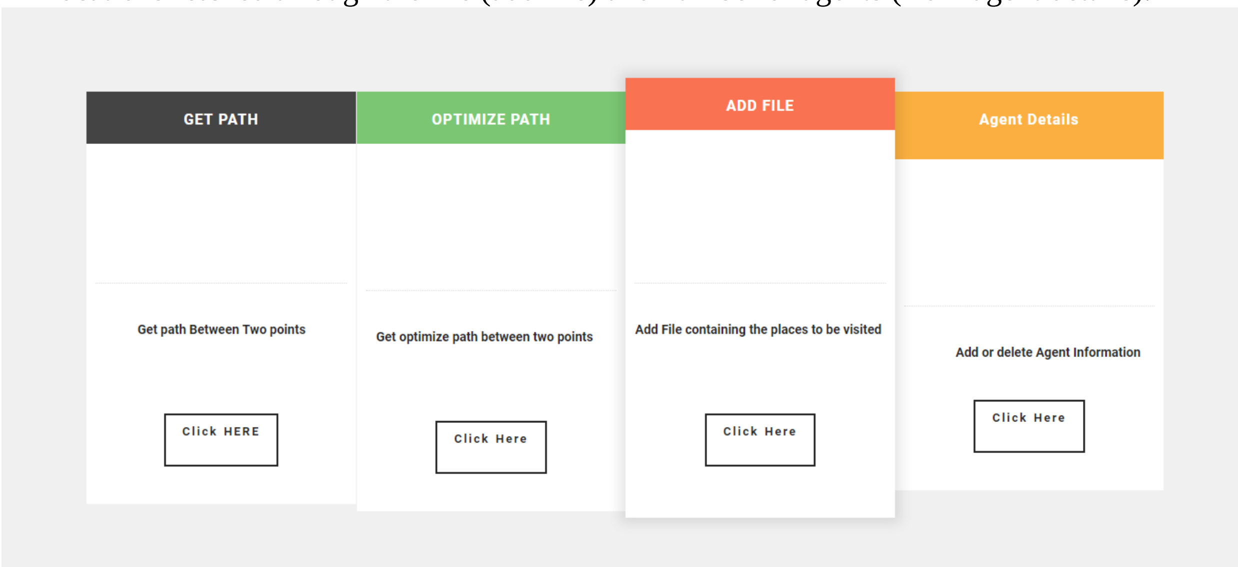

Figure6: home page

3.7.1.1 Get path

- On selecting **Get path from the home page,** the admin can get the shortest path between 2 locations and can also add waypoints in between (optoinal).

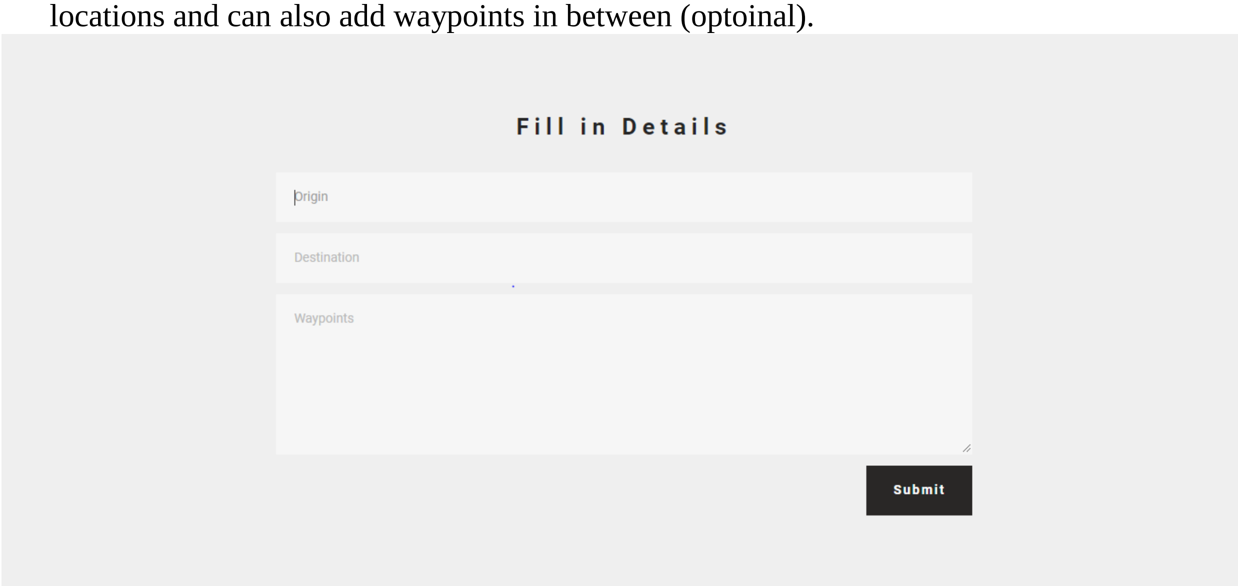

Figure7: Get path

3.7.1.2 Add file

By selecting "**Add file**" from the hpme page, the admin can uplad the excel file containing the set of alllocations to be optimized.

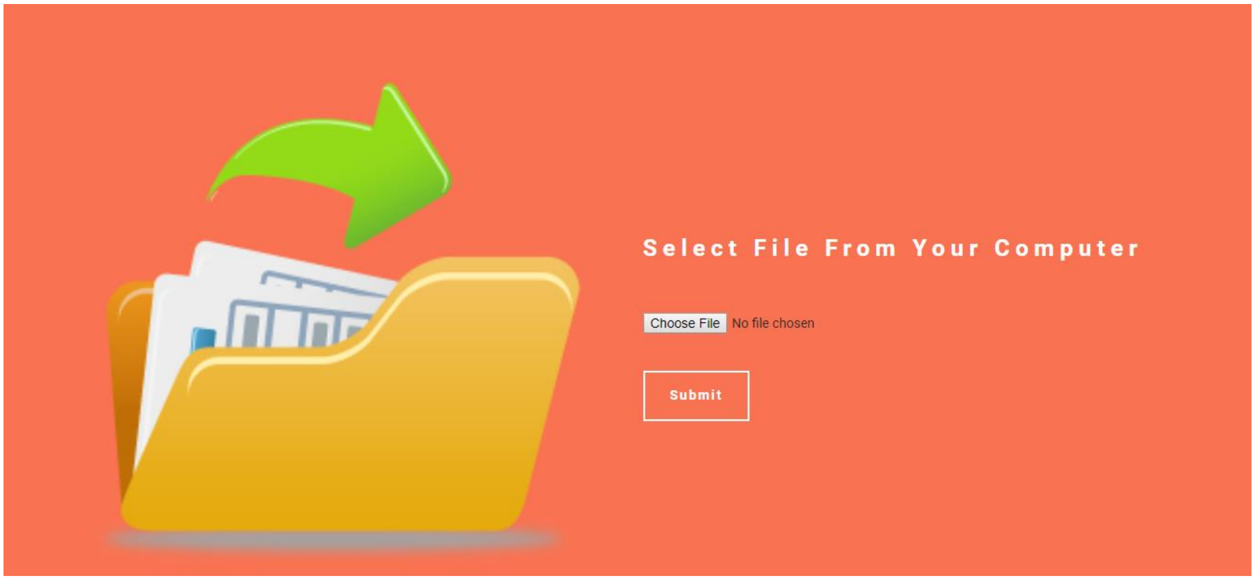

Figure8: Add file

3.7.1.3 Agentdetails

By clcking on "**Agent details**" from he home page,admin is navigated to this page having 2 options i.e add agent and delete agent.

- In the first page "**Add agent**",admin must enter the agent credentials.

Figure9:Add agent

- In the second page “**Deleteagent**”, admin may select the agent o be deleted by clicking the X icon.

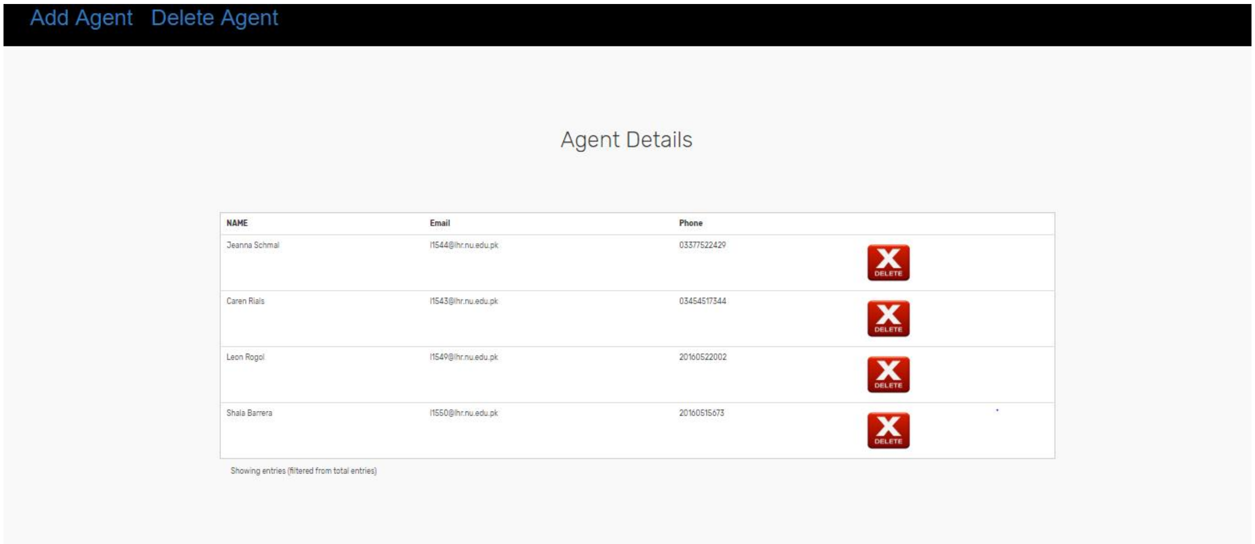

Figure10: Delete agent

3.8 Conclusion

The complete analysis of system requirements, architectural strategiesand the gui design reveals that the system will be comprised of two portals i.e. admin and agent where the admin will be the in charge of all the actions to be performed in the system in order to provide the optimized path which will be assigned to the agents. Agent’s portal will be in the form of an app (mobile application) through which he will receive optimized path and then we will track his path through that app.

4. Implementation and test cases

For this project we have worked on different types of APIs mainly focusing on Google maps APIsin order to map the locations and get the shortest distances using **genetic algorithm**.Some of the useful APIs that have helped in our implementation have been stated in this chapter. We have used the data set provided by techlogix and manipulated the functionalities of different APIs in this chapter. We have also developed an application that controls the agent portal.

4.1 Explored APIs

In addition to the literaturereview, we have done some research on APIs that can be used to achieve our desired goals.
We have explored the following two APIs.

1. Google APIs
2. HERE APIs

4.1.1 Google Maps APIs:

Google maps APIs are the most famous related to maps in world. They provide us lot of services.
The important data that is used by google is [13]

- Recommended speed limit of the roads.
- Traffic analysis
- Number of turns

4.1.1.1 Types of google APIs:

1. Geocoding API
2. Roads API
3. Distance Matrix API
4. Direction API

There are numerous other APIs associated with the above presented APIs. These function as a whole to produce certain results on maps. For example, process of converting address to geographic coordinates is called geocoding which is the main purpose of Geocoding API and process of converting geographic coordinates to address is called reverse geocoding, which we can get from associated APIs of Geocoding API. Similarly, marking location points on maps is also a feature of a API. Following figure will explain the process of marking location points.

Making a point on Map as Marker with Google API

Figure 11: Mapping Points With Google API As Provided By Techlogix [6]

4.1.1.1.1 Geocoding API

This API provides the service to change address to geo coordinates and vice versa [7].

4.1.1.1.2 Road API

This service can be used premium user only. Most importantly, this API will help us to get speed limit of road [8].
The speed limit data provided can be:

- not real-time
- Estimated
- Inaccurate
- Incomplete
- Out dated

4.1.1.1.3 Distance Matrix API

It provides distance and time to travel between two points [9].
Time information is based on these factors:

1. Predictive traffic information,
2. Depending on the start time specified in the request, different modes of transportation

4.1.1.1.4 Direction API

This API helps to find routes between Point A and B [10]. The parameters of this are:

- Start and End point
- Way points which can be maximum 25
- Arrival time
- Departure time
- Alternative routes flag
- Avoid option: Like tolls and highways
- Optimized route option

4.1.1.2 How Google Collects Road Closure Information?

As you can see in below picture Google has started implementing this feature where people would report and provide information to google. So this information would be very important for making google app more accurate. Google also is owner of Waze app so google also get data from this application. There may be other sources from where the company gets data which are not publically told. For example, it is rumored that Google collects information data from all devices (android-based)

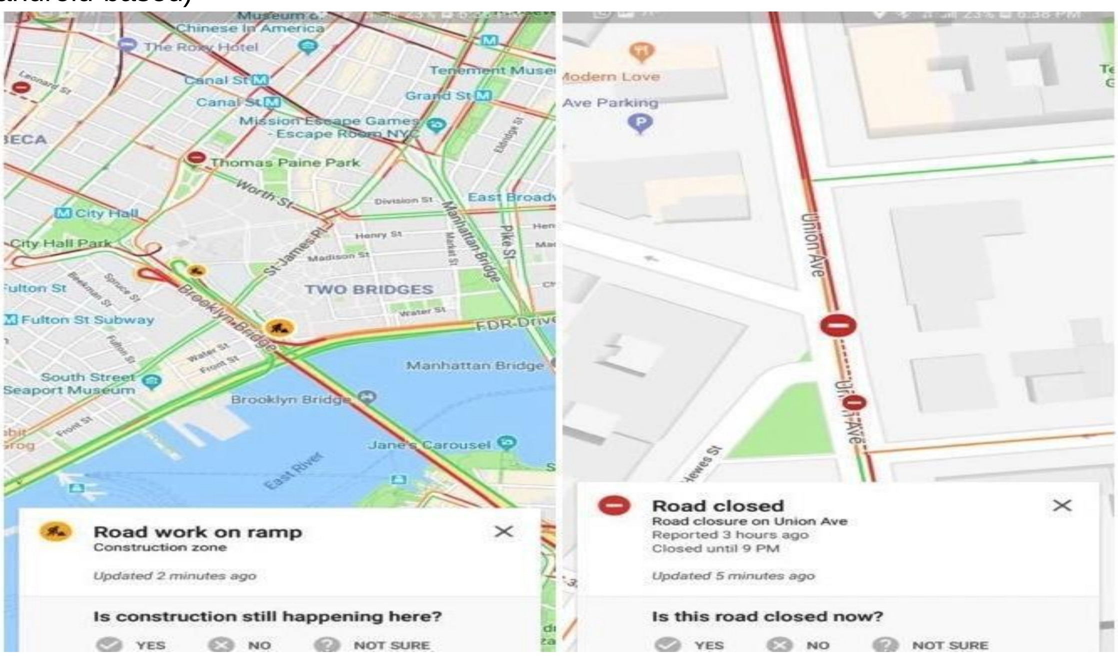

Figure 12: Google Collecting Data [13]

4.1.2 HERE API

4.1.2.1 About

This HERE API provides SDKs and APIs for web and mobile apps regarding maps, navigation and location based services. These allows us to solve complex problems very easily by providing fast and flexible access to mappings, geocoding, live positioning ,routing, data from live traffic etc.

It also provides location based services for IoT, mobile and embedded systems. It also provides advanced algorithms based on locations for fleet management.

4.1.2.2 Features

This API called the "HERE Maps API" for different platforms like JavaScript can be used to envision map figures, precise directions, translating addresses to co-ordinates(geocoding) and live traffic.

Following are the API's which are included in HERE MAPS and their functionalities.

4.1.2.3 MAPS AND SATELLITE TILES:

Here Tile API, which provides images of maps in a user friendly choice of presentation styles, including satellite, terrain and traffic. This basically provides detailed street level map to screen which is like Google Maps. We have the choice to get the maps in satellite view (image-like maps from far away), terrain (street level design of map drawn using color combinations and lines etc.), and traffic (information of live traffic shown via different lines).

4.1.2.4 VENUE MAPS:

This API can call more than 11,000 sites using PNG or JS tiles. This includes floor levels or fully designed models with geometry/polygons, exit/enter doors, stores, elevators etc.

4.1.2.5 CAR AND PEDESTRIAN ROUTING:

This feature guides the traveler move by move during drive and during walk both including directions with detailed instructions available in multiple languages. This feature is copied from google map feature in which assistant of software guides us after each interval via voice and text etc. Travelers can change preferences like shortest or fastest route, restrictions, tolls or motorway etc.

4.1.2.6 GEOCODING:

This API gets an address as input and translates it into geo co-ordinates as precise as possible.

4.1.2.7 PLACES:

This API enables us to find and discover points of interest. Results will be in the form of name, address, category, location, image, and contact info etc.

4.1.2.8 REALTIME AND HISTORICAL TRAFFIC TILES:

This is the most useful API which developers can use in their apps to display live traffic information on the devices having maps. This includes transportation instances, live historic traffic events for major towns and cities of the world.

4.1.2.9 CUSTOME LOCATION EXENSION:

With this API, we can save, manage and retrieve custom made POIs and polygons, shapes etc.

4.1.2.10 PLATFORM DATA EXTENSION:

This is also can be a beneficial API.It allows to get additional HERE Map Content (for advanced user), including height and slope values, curvature, speed limits and traffic lights.

4.1.2.11 COMPARISON WITH GOOGLE’s API

We did a comparison of this HERE API with GOOGLE MAPS API. Following shows images of maps (same co-ordinates) rendered by these API’s including live traffic information.

HERE API
Longitude: 74.3604598, Latitude: 31.5105876

Figure 13: Here API Rendered Map

GOOGLE MAPS API
Longitude: 74.3604598, Latitude: 31.5105876

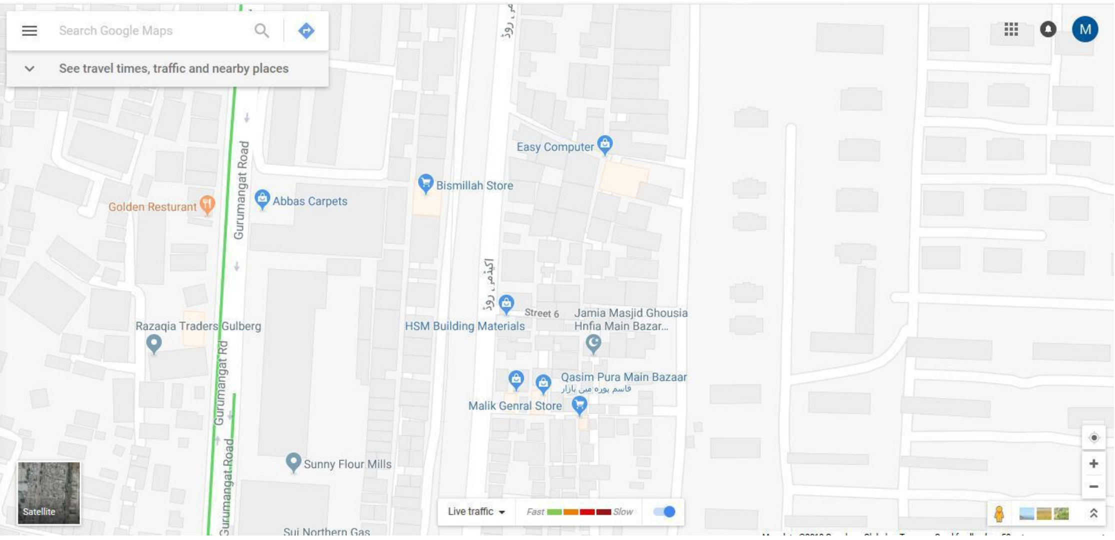

Figure 14: Google API Rendered Map

4.1.2.12 DRAWBACKS

This API is well developed and very useful regarding the functionalities it provides. It provides almost every type of details that we require in our project like plotting multiple co-ordinates, translating addresses into geo coordinates, live traffic and incidents information etc. But the major problem we will face in implementing this API is that is not applicable in PAKISTAN. It can be clearly seen from the above figures that GOOGLE API shoes detailed map including places and streets but HERE API only shows LAHORE CITY and some areas. It does not show any details of streets and roads even when fully zoomed. This proves that this API is not well developed for Pakistani users. Therefore, this is not useful for applications built for Pakistan.

4.2 Connection between waypoints

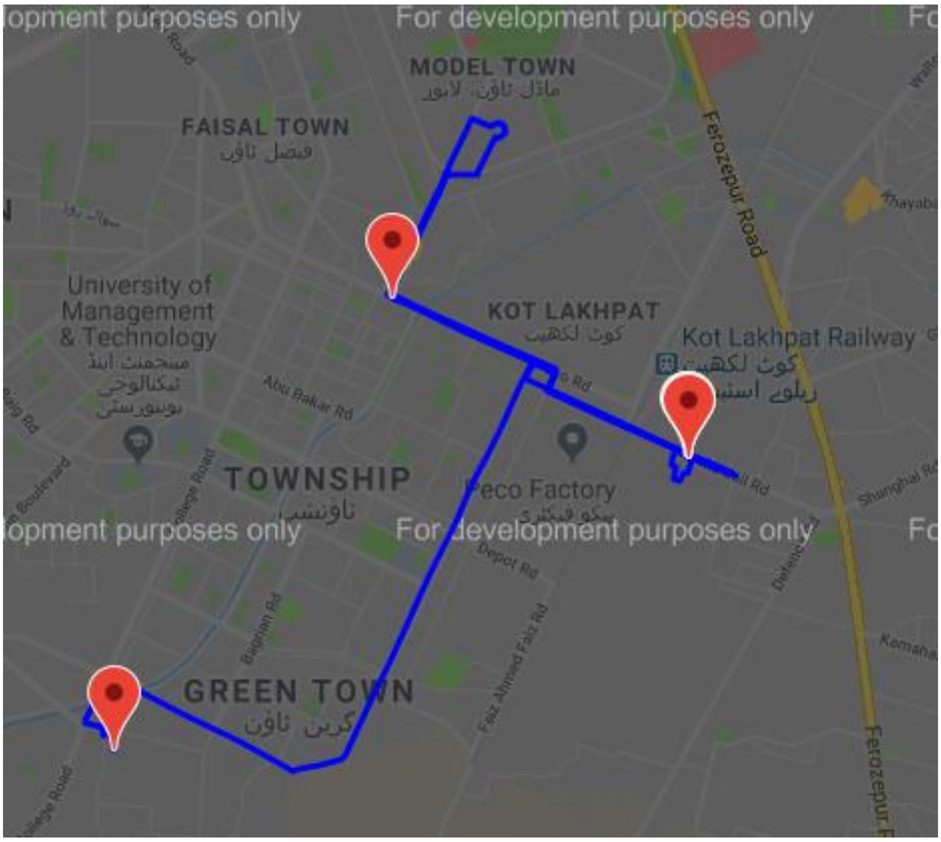

Figure 15: use of polylines to connect points

By fectching locationsin the form of longitude and latitudewe have connected the locations on google map by using **direction API.** Andfor the representation of the connected paths we have used polylines.

4.3 OpenWeatherMap API

Weather is considered to be one of the important factors that hinder route planningconsidering the vehicle typeof the field agent (specially motor bike). So we have incorporated theOpenWeatherMap API in our algorithm as well through which we are fetching the current weather data for any location. This API allows us to access current weather data for any location including over 200,000 cities. Current weather is frequently updated based on global models and data from more than 40,000 weather stations.Our main focus in this regard is to get those locations where it is raining currently so that those locations are omitted because a salesman on a bike cannot goto those locations where it is raining.
Below screenshot from our application shows real time weather data of each geolocation.In this screenshot,on the left(in green portion) is the the live weather results of those geolocations which are clear,mist and smoke respectively.

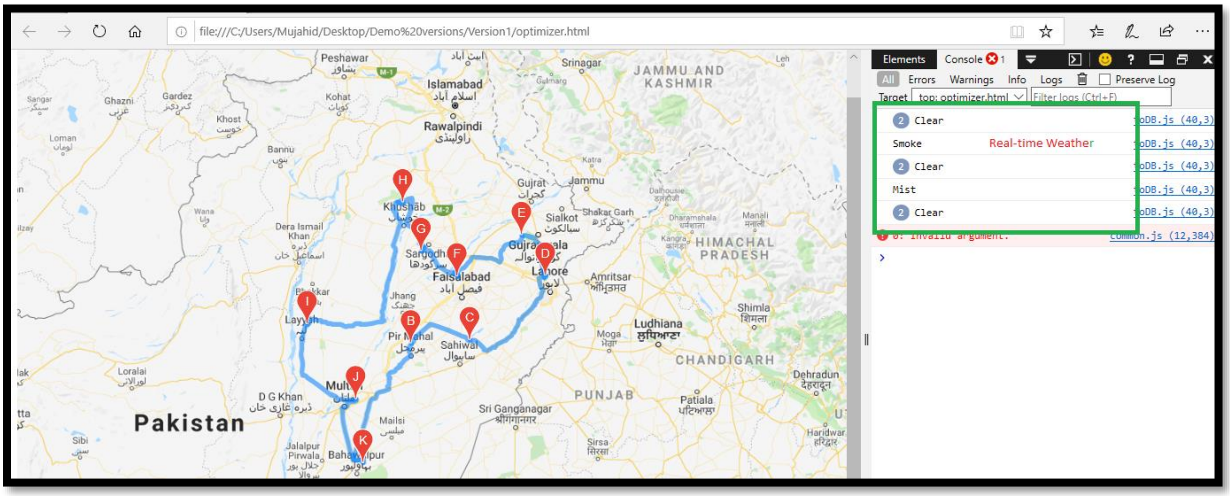

Figure 16:openWeatherMap API

4.4 The Agent Portal(Mobile App.)

We have managed to consider some of the important factors that affect route planning and incorporated them in our**Genetic Algorithm**byusing some of the aboveAPIs.The algorithm will provide us the set of locations for each field agent to be visited on that day.On the sart of the day, the admin will run the algorithm on the Admin portal and the optimized set of locations with a particular sequence for each agent (covering all locations in minimum possible transit time)will be placed automatically on the **online server** (Firestore) in the form of points i.e. longitude and latitude.
From the server the locations will be then fetched and mapped onto the mobile application (the Agent Portal)of the corresponding agent.
Below picture shows complete round trip path being shown on the agent’s mobile app.Numbers 1-14 shows the stops in sequence which agent has to visit.

Figure 17: The Agent Portal

4.5 Test case Design and description

Our system constitutes two portals i.e. web portal (operated by admin) and mobile portal (operated by the field agent). For all of the below mentioned test cases, the internet connectivity is a pre-condition and for mobile devices, the GPS must be turned on. The environment used for testing the web portal is Windows 10 (Browser: Chrome) and for the mobile portal is tested on android version 8.0 Oreo.

Test Case Number	Test Case Name
1.1.1	Admin login
1.1.2	Add locations
1.1.3	Add agent
1.1.4	Optimize path
1.1.5	Agent login
1.1.6	Start day
1.1.7	Check in location
1.1.8	End journey

Table 4: Test Cases

4.5.1 Admin Login

Test Case ID:	1.1.1	**QA Test Engineer**	Muhammad Zain Jaffery
Test case Version:	1	**Reviewed By**	Muhammad Zubair
Test Date:	4/21/2019	**Use Case Reference(s)**	N/A
Revision History			
Objective	Admin must login to the web portal		
Product/Ver/Module	Admin Login Module.		
Environment:	Operating System: Windows 10, Browser: Chrome		
Assumptions:	System must have active internet connection.		
Pre-Requisite:	Admin must be registered before logging into the system		
Step No.	**Execution description**	**Procedure result**	
1	Admin enters his username and password into username and password fields and clicks on the login button	System will check credentials and login the admin into the web portal.	
Comments: The system is working fine.			
	☒Passed ☐Failed ☐Not Executed		

4.5.2 Add Locations

Test Case ID:	1.1.2		**QA Test Engineer**	Muhammad Zain Jaffery
Test case Version:	1		**Reviewed By**	Muhammad Zubair
Test Date:	4/21/2019		**Use Case Reference(s)**	N/A
Revision History				
Objective	The excel file containing set of locations must be uploaded			
Product/Ver/Module	Add locations module			
Environment:	Operating System: Windows 10, Browser: Chrome			
Assumptions:	The file contains all the details of locations to be visited			
Pre-Requisite:	Admin must be logged in to the web portal			
Step No.	**Execution description**		**Procedure result**	
1	Admin clicks on Add file button		System redirects the admin to the Add file page.	
2	Admin clicks on choose file button		System prompts for choosing a file from local directory	
3	Admin selects the excel file containing all locations from local directory		System uploads the excel file	
Comments: The system is working fine				
	☒Passed ☐Failed ☐Not Executed			

4.5.3 Add Agent

Test Case ID:	1.1.3		**QA Test Engineer**	Muhammad Zain Jaffery
Test case Version:	1		**Reviewed By**	Muhammad Zubair
Test Date:	4/23/2019		**Use Case Reference(s)**	N/A
Revision History				
Objective	A new agent must be added to the system			
Product/Ver/Module	Add agent module			
Environment:	Operating System: Windows 10, Browser: Chrome			
Assumptions:	The admin has the authority to add agent			
Pre-Requisite:	Admin must be logged in to the web portal			
Step No.	**Execution description**		**Procedure result**	
1	Admin clicks on Agent details button		System redirects the admin to the Agent details page	
2	Admin clicks on add agent tab		System navigates the admin to add agent page	
3	Admin fills in the credentials (name, email, phone) of the agent and clicks on submit		The new agent is added to the system. The system displays the list of all agents	
Comments: The system is working fine				
	☒Passed ☐Failed ☐Not Executed			

4.5.4 Optimize Path

Test Case ID:	1.1.4	**QA Test Engineer**	Nokahiz Ali
Test case Version:	1	**Reviewed By**	Muhammad Zain Jaffery
Test Date:	3/14/2019	**Use Case Reference(s)**	N/A
Revision History			
Objective	The system must run Genetic Algorithm and output the optimized path		
Product/Ver/Module	Optimization Module		
Environment:	Operating System: Windows 10, Browser: Chrome		
Assumptions:	The details of the agents and locations are fetched from the excel file		
Pre-Requisite:	Admin must be logged in to the web portal		

Step No.	**Execution description**	**Procedure result**
1	Admin clicks on Optimize path button	System redirects the admin to the Optimizer page and shows the locations to be visited on the map
2	Admin selects default input parameters and clicks Start Optimizer	System displays the path between the locations through polylines in an optimized sequence
Comments: The system is working fine		
	☒Passed ☐Failed ☐Not Executed	

4.5.5 Agent Login

Test Case ID:	1.1.5	**QA Test Engineer**	Nokhaiz Ali
Test case Version:	1	**Reviewed By**	Muhammad Zain Jaffery
Test Date:	4/12/2019	**Use Case Reference(s)**	N/A
Revision History			
Objective	Agent must login to the mobile portal		
Product/Ver/Module	Agent Login Module.		
Environment:	Android device (version: 8.0)		
Assumptions:	System must have active internet connection, and GPS must be turned on		
Pre-Requisite:	Agent must be added in the system (by admin) before logging into the system		

Step No.	**Execution description**	**Procedure result**
1	Agent enters his username and password into username and password fields and clicks on the login button	System will check credentials and login the agent into the mobile portal. The system
Comments: The system is working fine.		
	☒Passed ☐Failed ☐Not Executed	

4.5.6 Start Day

Test Case ID:	1.1.6	**QA Test Engineer**	Muhammad Zubair
Test case Version:	1	**Reviewed By**	Nokhaiz Ali
Test Date:	4/26/2019	**Use Case Reference(s)**	N/A
Revision History			
Objective	Agent must start traversing through the path assigned to him		
Product/Ver/Module	Agent’s Journey Module		
Environment:	Android device (version: 8.0)		
Assumptions:	System must have active internet connection, and GPS must be turned on		
Pre-Requisite:	Agent must be logged in to the mobile portal		
Step No.	**Execution description**	**Procedure result**	
1	Agent clicks on Start Journey button	System shows the path of the agent and starts tracking the location of the agent. And broadcast is on the web portal (Admin portal)	
Comments: The system is working fine.			
	☒Passed ☐Failed ☐Not Executed		

4.5.7 Check-in Location

Test Case ID:	1.1.6	**QA Test Engineer**	Muhammad Zubair
Test case Version:	1	**Reviewed By**	Nokhaiz Ali
Test Date:	4/29/2019	**Use Case Reference(s)**	N/A
Revision History			
Objective	The server (Admin portal) must be notified about the location visited by the agent		
Product/Ver/Module	Agent’s Journey Module		
Environment:	Android device (version: 8.0)		
Assumptions:	System must have active internet connection, and GPS must be turned on		
Pre-Requisite:	Agent must be logged in to the mobile portal		
Step No.	**Execution description**	**Procedure result**	
1	Agent clicks on check-in location after visiting the required location.	System sends a notification to the Admin portal	
Comments: The system is working fine.			
	☒Passed ☐Failed ☐Not Executed		

4.5.8 End Journey

Test Case ID:	1.1.8	**QA Test Engineer**	Muhammad Mujahid
Test case Version:	1	**Reviewed By**	Muhammad Zain Jaffery
Test Date:	4/29/2019	**Use Case Reference(s)**	N/A
Revision History			
Objective	The server (Admin portal) must be notified about the journey completion of the agent		
Product/Ver/Module	Agent's Journey Module		
Environment:	Android device (version: 8.0)		
Assumptions:	System must have active internet connection, and GPS must be turned on		
Pre-Requisite:	Agent must be logged in to the mobile portal		
Step No.	**Execution description**	**Procedure result**	
1	Agent clicks on End Journey buttons	System sends a notification to the Admin portal	
Comments: The system is working fine.			
	☒Passed ☐Failed ☐Not Executed		

4.6 Test Metrics

Metric	Purpose
Number of Test Cases	8
Number of Test Cases Passed	8
Number of Test Cases Failed	0
Test Case Defect Density	0
Test Case Effectiveness	0

Table 5: Test Metrics

4.7 Traceability Matrix

Table 6: Traceability Matrix

Requirements	Test Case No.	Implemented/Not Implemented	Pass/Fail	Comments
System shall allow admin to log into the Admin portal	1.1.1	Implemented	Pass	--
System shall provide feature for admin to add Locations	1.1.2	Implemented	Pass	--
System will allow the admin to add a new agent	1.1.3	Implemented	Pass	--
System will run genetic algorithm and output the optimized path	1.1.4	Implemented	Pass	--
System shall provide a login screen to agent on the mobile portal	1.1.5	Implemented	Pass	--
System shall provide feature for agent to start his journey	1.1.6	Implemented	Pass	--
System shall allow the agent to mark the visited location on the android device	1.1.7	Implemented	Pass	--
System shall allow the agent to mark his journey as completed on the android device	1.1.8	Implemented	Pass	--

4.8 Software bug report

4.8.1 Bug 1

Submitter
Name: Nokhaiz Ali
Email: nokhaizali711@gmail.com

Date Seen
10th April, 2019

Versions
Android 8.0 (Oreo)

Bug Description
During the login of sales agent, user is not able to login with right credentials.

Severity
Major

Steps to Reproduce
- Apply log on user name and password.
- Trace these variables to find the cause of bug.

Actual Behavior
Instead of showing menu screen to the user app is not allowing the user to login.

Expected Behavior
After entering right credentials showing menu screen to the user.

Troubleshooting/Testing Steps
- Trace the variables with help of log.
- Find that data is not coming in that variable from our database.
- After that apply solution for the fault we find during troubleshooting

Workaround
All the agents' details are stored at server i.e. web portal. In case if there is a temporary issue with fetching the credentials there can be internet or database connectivity issue. Hence solution is to ensure the internet connectivity at both portals.

4.8.2 Bug 2

Submitter
Name: Nokhaiz Ali
Email: nokhaizali711@gmail.com

Date Seen
18th April, 2019

Versions
Android 6.0 (Marshmallow)

Bug Description
When clicking the get path button the app is not showing path to user which is major issue.

Severity
Major

Steps to Reproduce
- Run the app
- Find that we have old version of sdk which is causing problem to run the app successfully
- Apply log on array list which is saving locations from our data base.
- Trace this variable to find the cause of bug.

Actual Behavior
App is not showing path after clicking get path button.

Expected Behavior
After clicking get path button our app should show the path to the user.

Troubleshooting/Testing Steps
- Run the app.
- Find that we have old version of sdk which is causing problem to run the app successfully.
- Apply log on array list which is saving locations from our data base.
- Trace this variable to find the cause of bug.

Workaround
Our mobile portal is compatible with android 7.0 version or above. So it will never work on android devices below 7.0. hence the only solution is to run the application on android 7.0 versions or above.

4.9 Conclusion

We have researched on several factors that affect route planning and have come up with different helpful APIs of Google for our project which includeDirection API, Distance API and geocoding API. And some of the other APIs as well which majorly include the weather API. From these APIs we areretrieving the data which is required for our Genetic Algorithm to workefficiently. After the inplementation we have successfully run the test cases and resolved the bugsas well which depicts that the system is working efficiently.

5. Experimental results and analysis

For the experimental analysis of our system we conducted a test. We took some random sets of locations (table 7) and applied our implemented genetic algorithm which gave us the optimized path and minimum transit time for each set. Then in order to measure the correctness of our solutions we compared each solution with all the possible combinations of the corresponding set of locations by setting the arrangement of genes (chromosome) in such a way that it gives us the transit time for each sequence respectively.

Number of locations tested	Global Maximum transit time (minutes)	Our system resulting transit time (minutes)	Brute Force transit time (minutes)
4	22	16.75	15.7
3	18	18	17
5	70	58	51
6	37	34	32
7	75	54	50
8	78	70	64
9	92	86	81
9	63	50	47
10	61	51	46
15	81	76	Not possible
20	95	86	Not possible

Table 7: Comparison of results

It can be clearly seen that our system results are close to the global minimum but not exactly minimal. This is observed because of the typical nature of genetic algorithm which usually gets stuck in **local optimum.**

5.1 Performance under weather conditions

We have incorporated weather factor in our GA and for is verification we have tested our algorithm with some location where weather condition is raining. In this situation our algorithm over looks such location and excludes this location from our deviced optimized path.

5.2 Experiment example

Referring to table 7 (first row), the detailed test results for 4 locations are elaborated as an example in figure 17.
In this example we have considered 4 random locations within the city for which the possible combinations are 24 (i.e. 4! = 4*3*2*1=24). For evaluation purpose, the readings are stated in the form of a histogram.

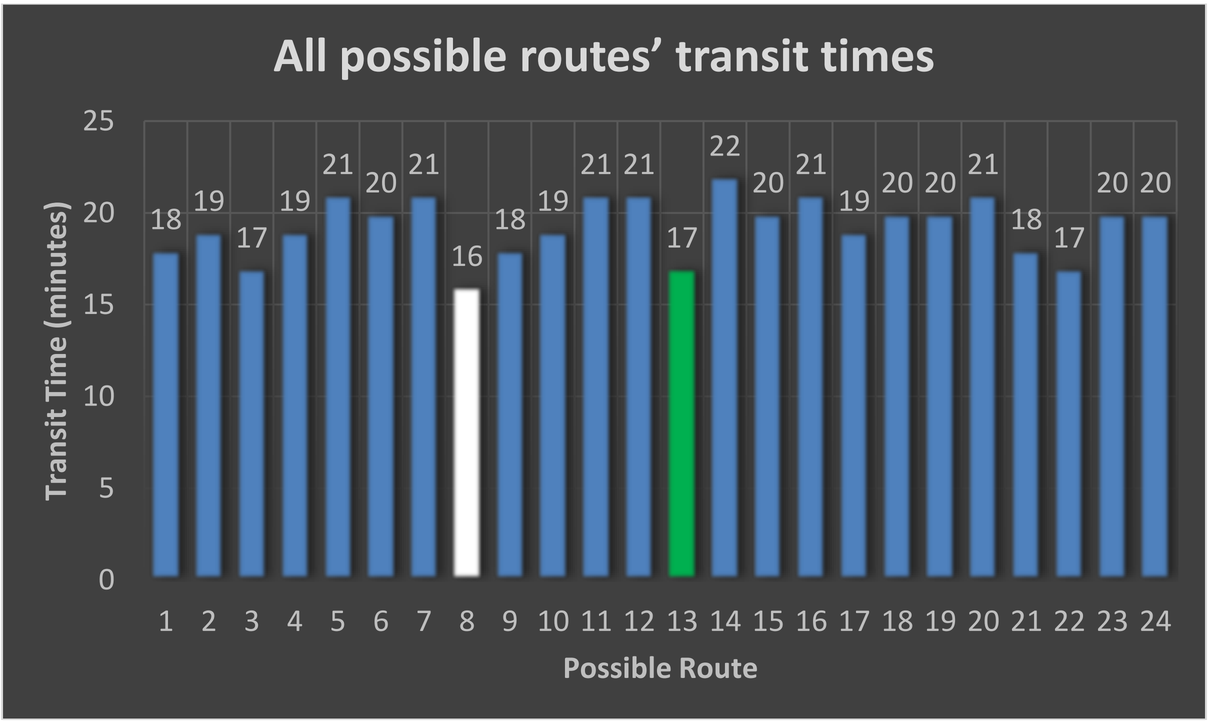

Figure 18: Histogram for all possible routes transit time for 4 random locations

Along the x-axis are all possible routes for the selected 4 random locations and on the y-axis we have the transit time in minutes. Each bar represents the transit time for each route in which the white bar shows the **global optima** and the green bar represents the optimized route provided by our system. So here it can be seen that our result is very close to the global optima.

6. Final Conclusion

Our aim was to achieve a better path for salesman so the cost of operation can be reduced. This problem is subdomain of VRP and TSP. During this project we studied many algorithms related to this problem and we selected genetic algorithm as it provides better efficiency and is easy to work with. As a source of data we used google maps APIs and weather API. In google APIs our major focus was on direction API. Then we incorporated whole of this data to run our application on admin side which was controlled by manager. And then we sent that data to firebase and then agent was able to get it through android application. Our major focus was on functional requirements but we also considered areas like security as well which come under area of non-functional requirements.

During analysis phase of our project we compared the solutions provided by our algorithem with the global optimum solurion. Usually our solution gave the results close to global optima but in some cases it gave the global optimum solution as we increased number of generations and ran it on limited amount of points. So as the result of these outcomes in future more work can be done towards reliability and scalability of this project.

7. References

[1] Punjab IT Board. (2018). Rasta [Online]. Available: https://play.google.com/store/apps/details?id=com.pitb.rasta

[2] Waze. (2016). Waze - GPS, Maps, Traffic Alerts & Live Navigation [Online]. Available:https://play.google.com/store/apps/details?id=com.waze

[3] Zhigou Fan and Mengkun Ma 2018 IOP Conf. Ser.: Earth Environ. Sci. 108 052007

[4] Google.cm. (2018). *saving algorithm - Google Search*. [online] Available at:https://www.google.cm/search?q=saving+algorithm&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjj2v-PrIveAhUMLcAKHZ5pBr4Q_AUIDigB&biw=1366&bih=608#imgrc=euvyEYBXLnAn3M: [Accessed 17 Oct. 2018].

[5] Aswani.ieor.berkeley.edu. (2018). [online] Available at: http://aswani.ieor.berkeley.edu/teaching/FA15/151/lecture_notes/ieor151_lec18.pdf [Accessed 17 Oct. 2018].

[6] Google Developers. (2018). *Markers | Maps JavaScript API | Google Developers*. [online] Available at: https://developers.google.com/maps/documentation/javascript/markers [Accessed 17 Oct. 2018].

[7] Google Developers. (2018). *Get Started | Geocoding API | Google Developers*. [online] Available at: https://developers.google.com/maps/documentation/geocoding/start [Accessed 17 Oct. 2018].

[8] Google Developers. (2018). *Overview | Roads API | Google Developers*. [online] Available at: https://developers.google.com/maps/documentation/roads/intro [Accessed 17 Oct. 2018].

[9] Blogs.bing.com. (2018). *Bing Maps Distance Matrix API Launches Today*. [online] Available at: https://blogs.bing.com/maps/2017-10/bing-maps-distance-matrix-api-launches-today [Accessed 17 Oct. 2018].

[10] Google Developers. (2018). *Get Started | Directions API | Google Developers*. [online] Available at: https://developers.google.com/maps/documentation/directions/start [Accessed 17 Oct. 2018].

[11] Estrada, M., Smith, C. and Smith, C. (2018). *Google Maps is about to get Waze's best feature*. [online] BGR. Available at: https://bgr.com/2018/06/29/google-maps-vs-waze-incident-reports-user-interaction/ [Accessed 17 Oct. 2018].

[12] Király, A. and Abonyi, J. (2015). *Redesign of the supply of mobile mechanics based on a novel genetic optimization algorithm using Google Maps API*. [Online] https://www.sciencedirect.com. Available at: https://ac.els-cdn.com/S0952197614002553/1-s2.0-S0952197614002553-main.pdf?_tid=e83008e2-cdd6-475b-a7ff-d03f94dad000&acdnat=1539869419_00ee25da377101d6dce6062546d8bf83 [Accessed 18 Oct. 2018].

[13] Anon, (2018). [online] Available at: https://www.quora.com/How-does-Google-maps-update-the-traffic-and-roads-under-maintaince-information [Accessed 17 Oct. 2018].

[14] R. Geisberger, "Advanced Route Planning in Transportation Networks," des Karlsruher Instituts für Technologie, 2011

[15] M. Baum, "On Preprocessing the Arc-Flags Algorithm," KIT – University of the State of Baden-Wuerttemberg and National Laboratory of the Helmholtz Association, 2011.

[16] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks," Available at: http://algo2.iti.kit.edu/schultes/hwy/contract.pdf.

[17] T. Bossomajer, and G. David, "Complex Systems," Cambridge University Press, Cambridge, UK, 2000.

[18] N.K. Kuppuchamy, and R. Manimegalai, "Manet Routing: Optimization by Genetic and Fuzzy Logic Approach," Journal of Theoretical and Applied Information Technology, 2013

[19] J. E. Bell, and P. R. McMullen, Ant colony optimization techniques for the vehicle routing problem. Elsevier Ltd Advanced Engineering Informatics, 2004 vol 18 pp. 41–48.

[20] N. R. Jennings, On Agent-Based Software Engineering. Artificial Intelligence, 2000, 117 pp. 277–296

[21] https://www.ijser.org. (2016). *ROUTE OPTIMIZATION TECHNIQUES: AN OVERVIEW*. [online] Available at: https://www.ijser.org/researchpaper/ROUTE-OPTIMIZATION-TECHNIQUES-AN-OVERVIEW.pdf [Accessed 15 Oct. 2018].

[22] Teguh, Narwadi and Subiyanto, "An application of traveling salesman problem using the improved genetic algorithm on android google maps" in the AIP Conference Proceedings 1818, 020035 (2017); doi: 10.1063/1.4976899 [Online]. Available: http://dx.doi.org/10.1063/1.4976899.