EMERGENCY VEHICLE DISPATCHING

A MINI-PROJECT REPORT

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

at

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING, BANGALORE-560068

IV SEMESTER

Course Code: 16CS274

DESIGN AND ANALYSIS OF ALGORITHMS

DAYANANDA SAGAR UNIVERSITY

CERTIFICATE

This is to certify that the Design and Analysis of Algorithm Mini-Project report entitled **“Emergency Vehicle Dispatching”** being submitted by____________

Namitha K (ENG19CS0197)

Neti Sahithi (ENG19CS0205)

Pranav NG (ENG19CS0227)

Prarthana G (ENG19CS0229)

to the Department of Computer Science and Engineering, School of Engineering, Dayananda Sagar University, Bangalore, for the 4th semester B.Tech C.S.E of this university during the academic year 2020-2021.

Date:___________

____

Signature of the Faculty in Charge

____

Signature of the Chairman

ACKNOWLEDGEMENT

We are pleased to acknowledge the Faculty **Prof Bindu Madavi K.P,Assistant Professor,** Department of Computer Science & Engineering for her invaluable guidance, support, motivation and patience during the course of this mini- project work.

We extend our sincere thanks to **Dr. Sanjay Chitnis, Chairman,** Department of Computer Science & Engineering who continuously helped us throughout the project and without his guidance, this project would have been an uphill task.

We have received a great deal of guidance and co-operation from our friends and we wish to thank one and all that have directly or indirectly helped us in the successful completion of this mini-project work.

Team Members

Namitha K (ENG19CS0197)

Neti Sahithi (ENG19CS0205)

Pranav NG (ENG19CS0227)

Prarthana G (ENG19CS0229)

Abstract

Response time is a key factor in the emergency vehicle dispatching problem. Because regional emergency vehicles are limited, vehicle gaps will be created in the rescue station after vehicles are dispatched to several accidents, which affects quick response to the subsequent incidents. To solve this problem, we are using Dijkstra’s algorithm to find the shortest rescue path for current accidents and allotting the emergency vehicles for which they requested.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 About the Problem:

In India historically we have had different phone numbers to call police, fire and ambulance services namely 100, 102, 103. This system was designed at the time of a regulated telecom sector with only one telecom provider across India and one in each metro. Hence any call to these emergency numbers were routed to a call agent/ dispatcher of that emergency service and handled by the emergency personnel themselves. The system was not designed for emergency response initially but as an emergency contact.

1.2 About DAA:

The main aim of this project is using Dijkstra's algorithm to find the shortest between nodes and respond to requests and allotting emergency vehicles for which they requested. We are using python language to implement the same.

There are three types of emergency vehicles:

Ambulance

Fire truck

Police car

The three attributes to be considered are:

Zip codes as nodes

distance between two nodes

vehicle type

Chapter 2

Problem Statement

PROBLEM: What is the shortest distance an emergency vehicle needs to travel in order to reach the consumer?

Chapter 3

LITERATURE REVIEW

- The project mainly aims to dispatch an emergency vehicle to the desired location inserted.
- The vehicle types given are 1. Ambulance 2. Fire Truck and 3. Police Car.
- For the input like Vehicle ID, Vehicle Type and the Distance between the zip codes are manually inserted and are stored in JTables.
- By taking the current location, minimum distance between the zip codes in the locality are calculated.
- For calculating the minimum distance between the zip codes, famous Dijkstra's algorithm is used and the values are displayed in console.
- Availability of the desired vehicle is searched, and the minimum distance zip codes location dispatches the emergency vehicle

Chapter 4

SYSTEM REQUIREMENTS

4.1 Functional Requirements

- **Users must be able to enter pin code and emergency vehicles needed.**
- **The distance between user and nearest available station should be printed.**

4.2 Software and Hardware Requirements

Hardware Requirements

•Modern operating system
•32/64-bit CPU
•4 GB Ram

Software Requirements

•Python 3
•An editor like atom, PyCharm or jupyter notebook

Chapter 5 SYSTEM DESIGN

5.1 Architecture/Data Flow Diagrams

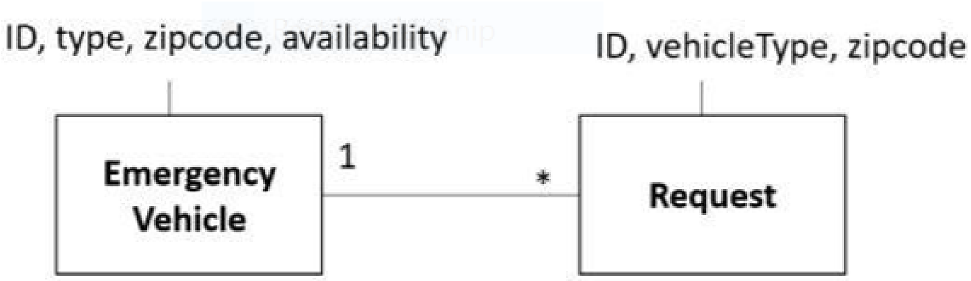


Figure 2

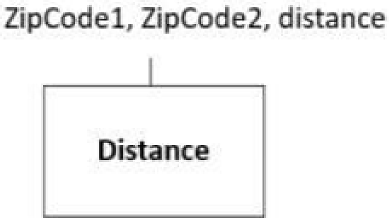


Figure 3

5.2 MODULES

The Following steps are implemented in the project.

Module 1. Creating Tables.
Module 2. Inserting the current zip code.
Module 3. Finding Shortest Paths.
Module 4. Availability Check Step
Module 5. Dispatch Vehicle Allocation.

Chapter 6

SYSTEM IMPLEMENTATION

6.1 Module Description

We are trying to design an emergency vehicle dispatch system.
Let's assume that there are three different types of emergency vehicles:
(i)Ambulance
(ii)Fire Truck
(iii)Police Car
Let's also assume that every request needs only one emergency vehicle.

Implement an algorithm that processes requests one by one. For each request, the algorithm should try to find the closest available emergency vehicle.
Make up the data for the project. We can generate the data file at run time or have it sorted in files and read it at run time.

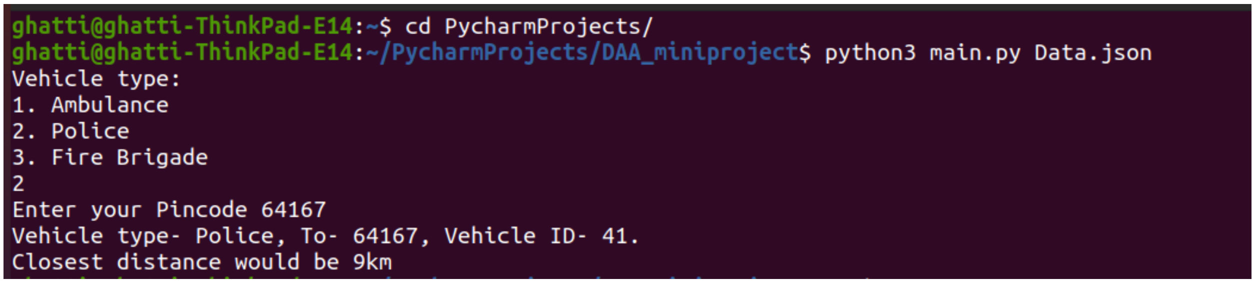
6.2 Pseudocode

```
Dijkstra(Graph, source):
        for each vertex v in Graph:
        dist[v] = infinity        // initial distance from source to vertex v is set to infinite
        previous[v] = undefined        // Previous node in optimal path from source
        dist[source] = 0        // Distance from source to source
        Q = the set of all nodes in Graph
        while Q is not empty: // main loop
        u = node in Q with smallest dist[ ]
        remove u from Q
        for each neighbor v of u:        // where v has not yet been removed from Q.
        alt = dist[u] + dist_between(u, v)
        if alt < dist[v]
        dist[v] = alt
        previous[v] = u
        return previous[ ]
```

Chapter 7

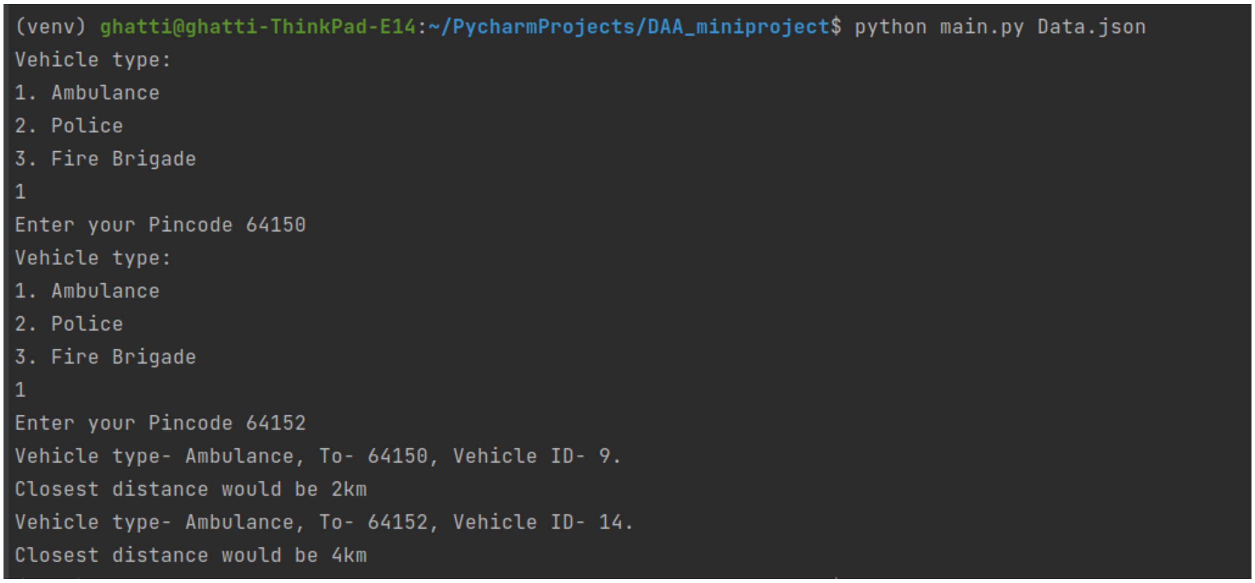
TESTING

1. A request of one type of vehicle from one pincode



```
ghatti@ghatti-ThinkPad-E14:~$ cd PycharmProjects/
ghatti@ghatti-ThinkPad-E14:~/PycharmProjects/DAA_miniproject$ python3 main.py Data.json
Vehicle type:
1. Ambulance
2. Police
3. Fire Brigade
2
Enter your Pincode 64167
Vehicle type- Police, To- 64167, Vehicle ID- 41.
Closest distance would be 9km
```

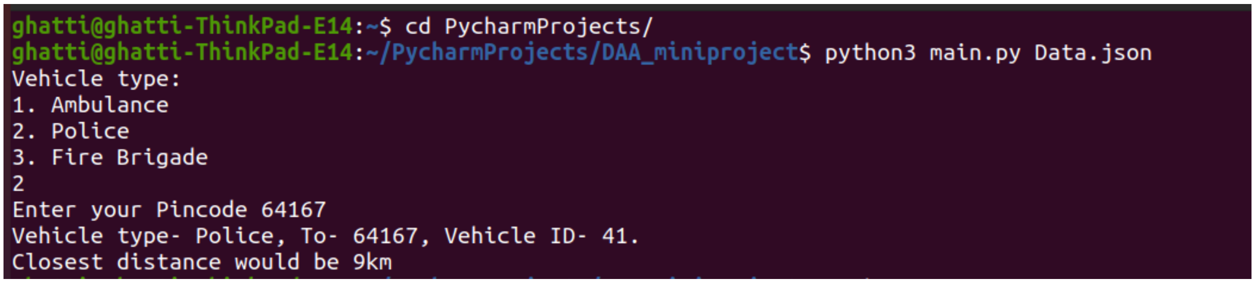
2. A request of one type of vehicle from two pincode



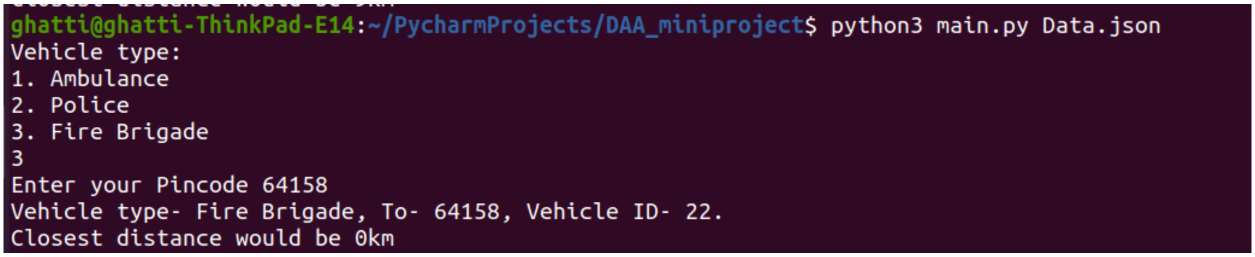
```
(venv) ghatti@ghatti-ThinkPad-E14:~/PycharmProjects/DAA_miniproject$ python main.py Data.json
Vehicle type:
1. Ambulance
2. Police
3. Fire Brigade
1
Enter your Pincode 64150
Vehicle type:
1. Ambulance
2. Police
3. Fire Brigade
1
Enter your Pincode 64152
Vehicle type- Ambulance, To- 64150, Vehicle ID- 9.
Closest distance would be 2km
Vehicle type- Ambulance, To- 64152, Vehicle ID- 14.
Closest distance would be 4km
```

Chapter 8

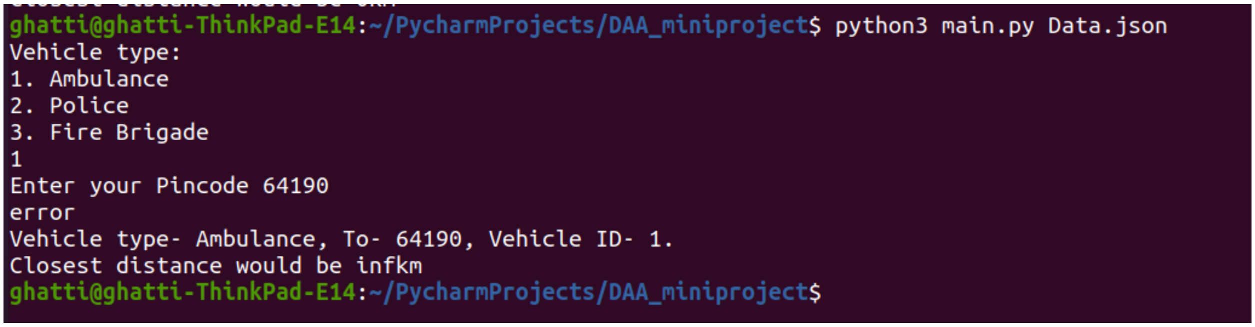
OUTPUT SCREENSHOTS



```
ghatti@ghatti-ThinkPad-E14:~$ cd PycharmProjects/
ghatti@ghatti-ThinkPad-E14:~/PycharmProjects/DAA_miniproject$ python3 main.py Data.json
Vehicle type:
1. Ambulance
2. Police
3. Fire Brigade
2
Enter your Pincode 64167
Vehicle type- Police, To- 64167, Vehicle ID- 41.
Closest distance would be 9km
```



```
ghatti@ghatti-ThinkPad-E14:~/PycharmProjects/DAA_miniproject$ python3 main.py Data.json
Vehicle type:
1. Ambulance
2. Police
3. Fire Brigade
3
Enter your Pincode 64158
Vehicle type- Fire Brigade, To- 64158, Vehicle ID- 22.
Closest distance would be 0km
```



```
ghatti@ghatti-ThinkPad-E14:~/PycharmProjects/DAA_miniproject$ python3 main.py Data.json
Vehicle type:
1. Ambulance
2. Police
3. Fire Brigade
1
Enter your Pincode 64190
error
Vehicle type- Ambulance, To- 64190, Vehicle ID- 1.
Closest distance would be infkm
ghatti@ghatti-ThinkPad-E14:~/PycharmProjects/DAA_miniproject$
```

Chapter 9

CONCLUSION

After the accidents, emergency vehicles should be dispatched to perform effective emergency rescue, and they should also be redistributed according to the potential risks of the coverage area of the rescue station, thus to shorten the response time for future incidents.

The ultimate goal of the current study is to put the proposed dispatching system into practical uses. The proposed dispatching system requires further advancements in social infrastructure. More efforts need to be put on real-world considerations.

Future studies on the present issue can include considerations of more severe scenarios, such as disasters where a large number of EVs are required. Amendments to the current dispatching strategy could be in need for those scenarios.

Chapter 10

References

https://pdfs.semanticscholar.org/d69a/69142f67573c9584eb6e220ca749b2bf30bb.pdf

https://stackoverflow.com/questions/8265307/file-input-for-dijkstras-algorithm

Dijkstra Algorithms - an overview | ScienceDirect Topics

Save and load Python data with JSON | Opensource.com