

**University of Missouri-Kansas City**

**Design and Analysis of Algorithms**

(COMP-SCI 5592)

**FALL SEMESTER PROJECT**

**EMERGENCY VEHICLE DISPATCHING SYSTEM**

**by**

**Sreeya Reddy Daripalli (16222142)**

**Sirisha Mantramurthy (16235068)**

**Sai Karthikeya Addagada (16231320)**

**Gopala Krishna Avinash Tummalapalli (16230748)**

**Table of Contents**

* Problem Statement
* Proposed Approach
* Assumptions
* Implementation Languages and Platforms
* About Dijkstra Algorithm
* BIG-O Complexity of Dijkstra Algorithm
* Illustrating Diagram
* Screenshots
* References

**Problem Statement**

The Problem Statement here states that we must implement an algorithm such that it sends an Emergency Vehicle required to the zip code from which the request has occurred, and the requests must be processed one after the other.

**Proposed Approach**

As per the given problem statement, we must design an efficient real-time system. So, we have decided to design a system where several zip-codes are taken as nodes(vertices) of a graph and the distance (in miles) between these nodes as the edges of the graph. We have allocated the three types of Emergency Vehicles i.e.; Ambulance, Fire Truck and Police to the zip codes such that using the Dijkstra’s Algorithm we can find the shortest path and distance between the nodes. We have also implemented other codes in order to check the availability of the Emergency Vehicle required at the zip code where the vehicle is required, if the vehicle is present then it is allocated to that particular zip code else the Dijkstra’s Algorithm works and the zip code which is closest and has the Emergency Vehicle required is retrieved and allocated. This is a short description of what we implemented to solve the problem.

**Assumptions**

To implement the Emergency Vehicle System, we have made several assumptions and they are:

* We have taken vertices of the graph as zip codes and the zip codes which we have taken are from 64150 to 64160 and each of these zip codes is assumed to be a digit lying between 0 to 9.
* We have taken a text file where in the graph present is assumed to give the edges weight between the connected vertices.
* We assumed the number of Emergency vehicles for each node to be fixed.
* We also assumed that once an Emergency vehicle is requested and allocated to a zip code, then another Emergency Vehicle of the same type replaces it.
* We assumed that there is no compulsion for each node to have all the three types of Emergency vehicles.

**Implementation Languages and Platforms**

* Java
* Eclipse IDE

**About Dijkstra Algorithm**

Dijkstra algorithm is used to find the shortest path between nodes in a graph, there are various real-life applications build on this Dijkstra algorithm for example in our day to day life we use navigation maps to reach our destination in less time this was build using Dijkstra algorithm. It was developed by a computer scientist named Edsger W Dijkstra in 1956. This algorithm is used to find the shortest path between nodes, where it selects a node as a starting point and selects the neighboring node which has the edge with low weight and continues until it reaches the specified destination node. This is the most efficient way to find shortest path between nodes.

**BIG-O Complexity of Dijkstra’s algorithm**

Basically, the time and space complexity of Dijkstra’s algorithm can be calculated in two cases:

* Using sorted input
* Using unsorted input

The complexity of Dijkstra algorithm with |v| vertices and |E| edges when it is sorted is as follows.

* **Time Complexity**: O((|v|+|e|) log v)
* **Space Complexity**: O(v)

The complexity of Dijkstra algorithm with unsorted information is as follows.

* **Time** **Complexity**: O(|v|^2)
* **Space** **Complexity**: O(v)

The algorithm we used to find the solution of emergency vehicle dispatching system is **Dijkstra Algorithm**.

The time complexity of the code we are using is



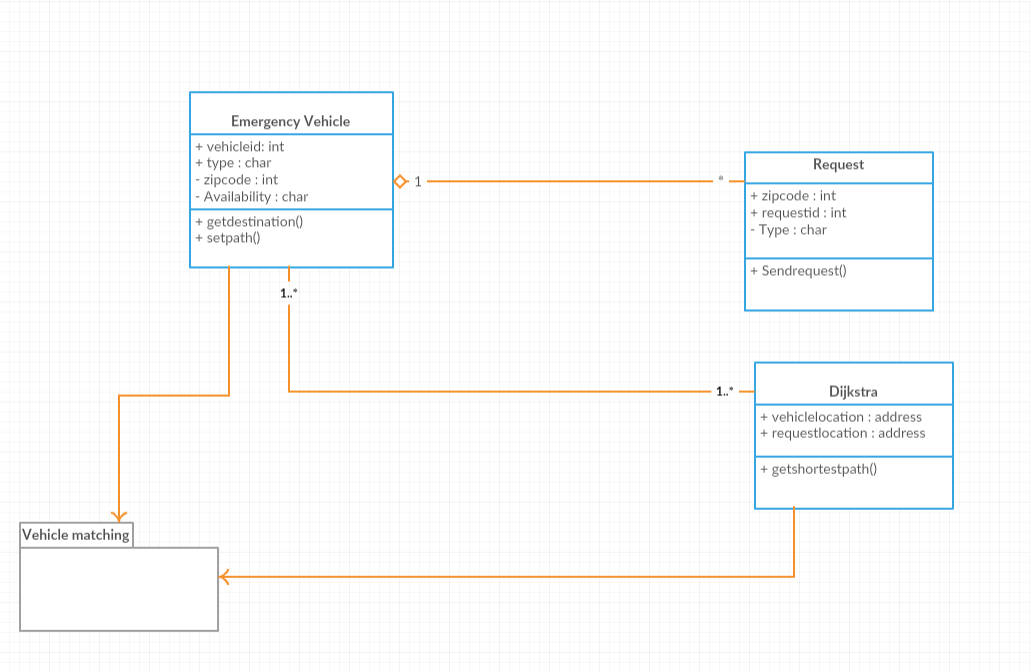
The space complexity of the code we are using is



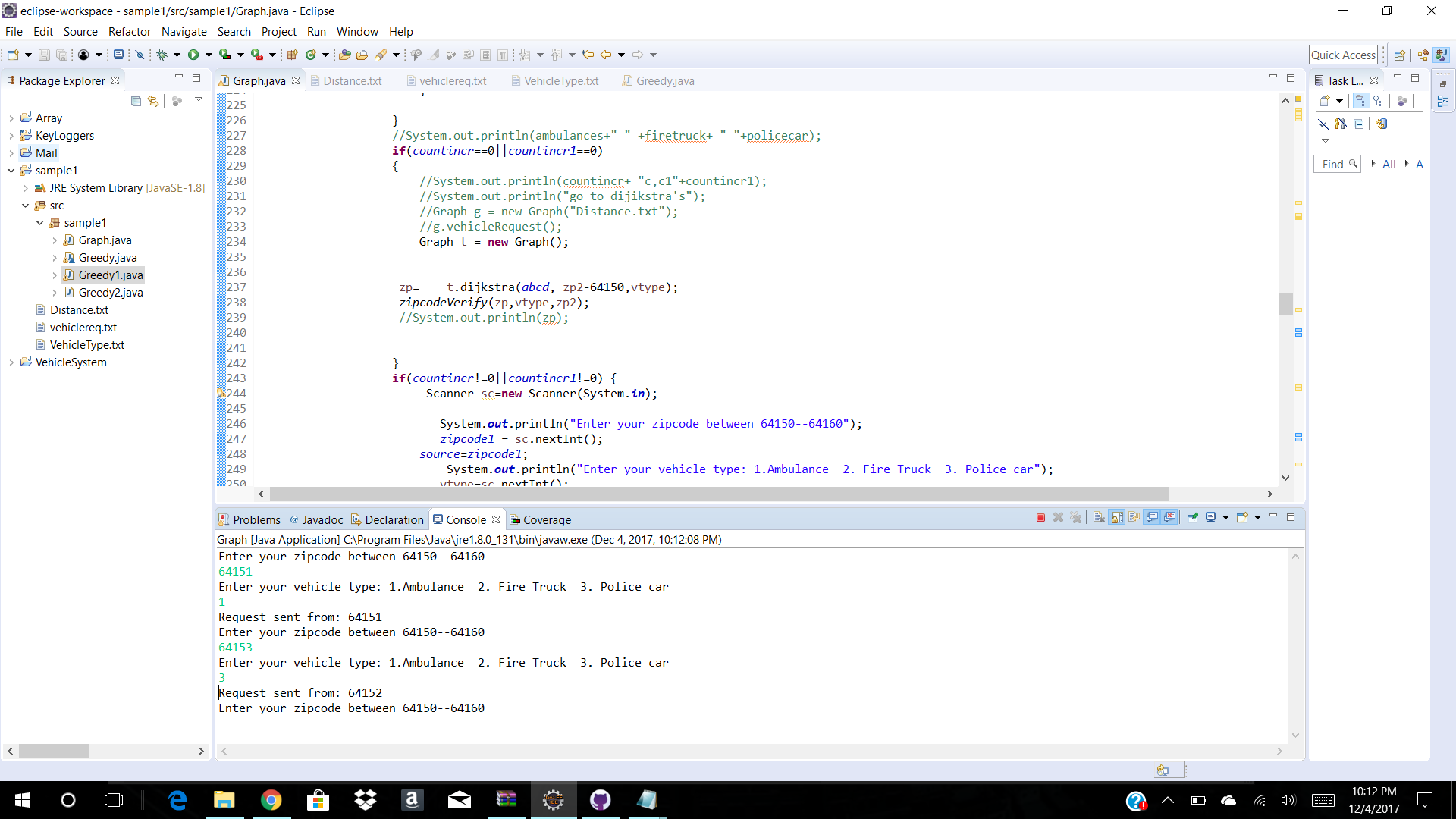
**Illustrating Diagram**

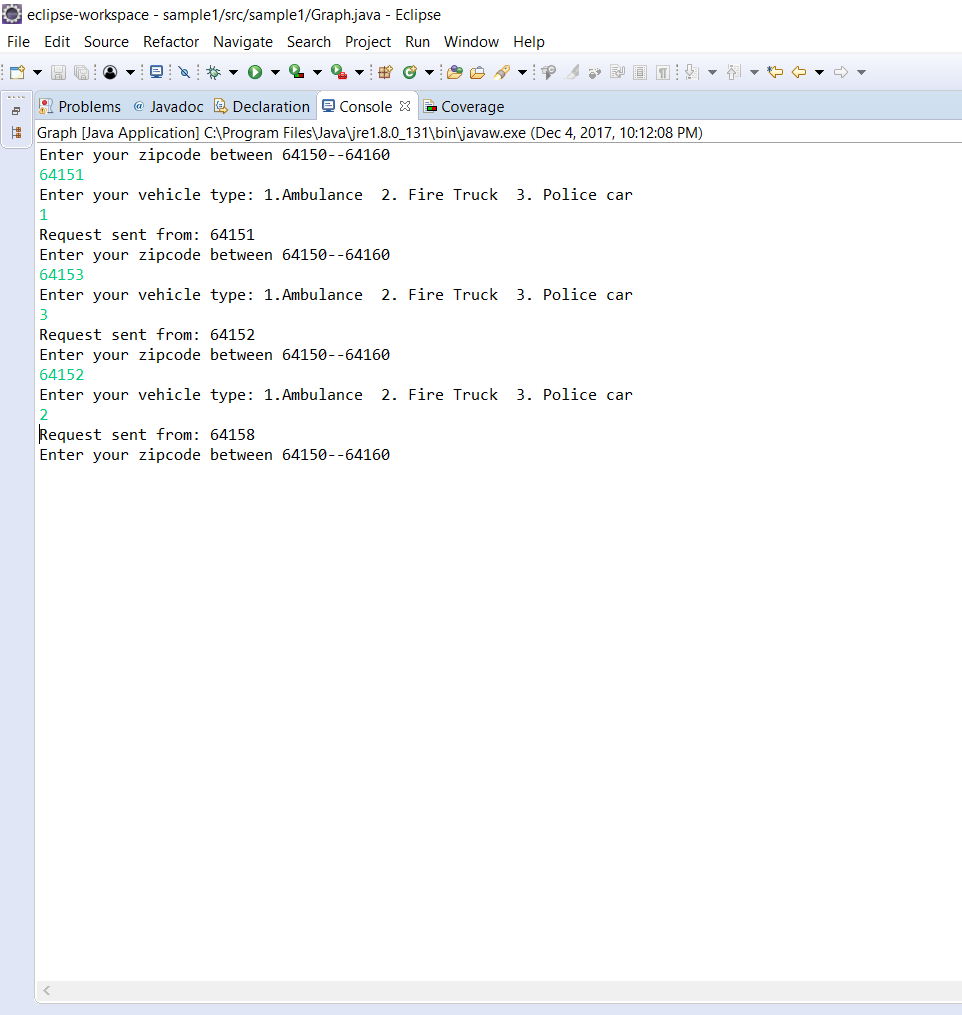
**Class diagram:**

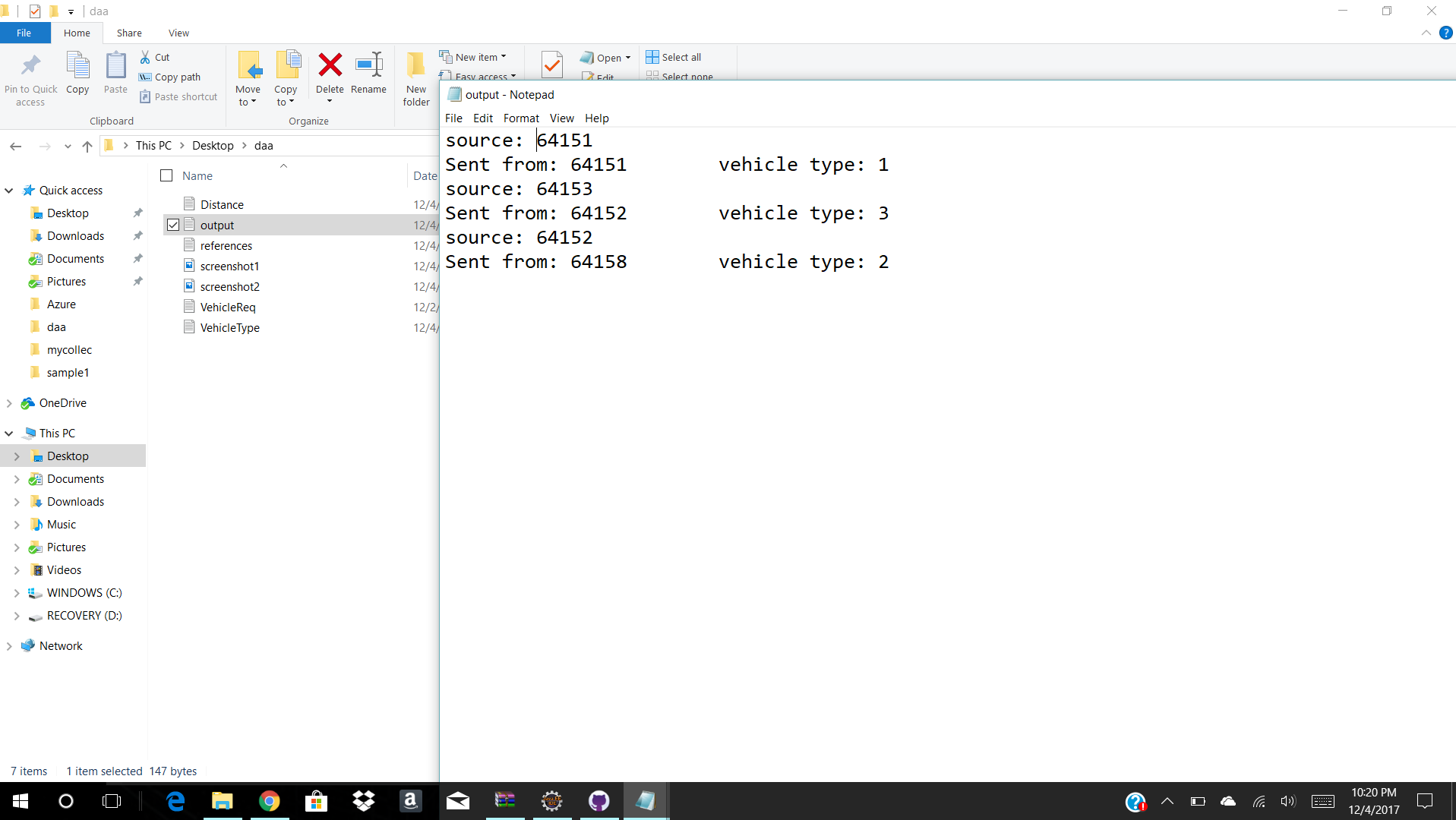
This class diagram depicts the basic structure of our view how to solve the emergency vehicle dispatching system. We have three classes they are Emergency Vehicle, Request and Dijkstra we also have a vehicle matching package. We have two relationships one aggregation relationship between request class and emergency vehicle class and second one is association relationship which is between emergency vehicle class and Dijkstra class. Emergency vehicle class and Dijkstra class are having a simple line with vehicle matching package.



**Screenshots**

****

****

****

**References**

* <http://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/>
* <http://www.vogella.com/tutorials/JavaAlgorithmsDijkstra/article.html>
* <https://codereview.stackexchange.com/questions/15062/reading-a-line-from-a-text-file-and-splitting-its-contents>
* <https://stackoverflow.com/questions/35147544/how-to-read-a-txt-file-into-a-2d-array-java>
* https://stackoverflow.com/questions/1994255/how-to-write-console-output-to-a-txt-file
* <https://www.youtube.com/watch?v=72BEuCPMgTo>
* <https://medium.com/@ssaurel/calculate-shortest-paths-in-java-by-implementing-dijkstras-algorithm-5c1db06b6541>
* <https://rosettacode.org/wiki/Dijkstra%27s_algorithm>
* <http://www.vogella.com/tutorials/JavaIO/article.html>
* <https://www.tutorialspoint.com/java/java_files_io.htm>
* <https://www.programcreek.com/2011/03/java-write-to-a-file-code-example/>

**GitHub Link:** https://github.com/SreeyaReddy/AlgorithmsProject