

The background of the slide features three white ambulances with orange and blue stripes, parked in a row. The text "Emergency Vehicle Dispatching System" is overlaid in a large, bold, black font. The ambulances have "HAMILTON COUNTY EMS" and "AMBULANCE" written on their sides.

Emergency Vehicle Dispatching System

Smrithi Agrawal – 2019140001
Prathamesh Bhoi - 2019140008
Karen Castelino - 2019140010



Project Briefing

- Aim: To dispatch an emergency vehicle.
- The vehicle types 1. Ambulance 2. Fire Truck and 3. Police Car.
- JTable: Vehicle ID, Vehicle Type and the Distance between the zip codes are stored.
- By taking the current location, minimum distance between the zip codes in the locality are calculated.
- Algorithm for calculating minimum Distance: Dijkstra's algorithm.
- Available desired vehicle is dispatched.

Technical Requirements

Language Used: JAVA

Software: ECLIPSE

Operating System: Windows

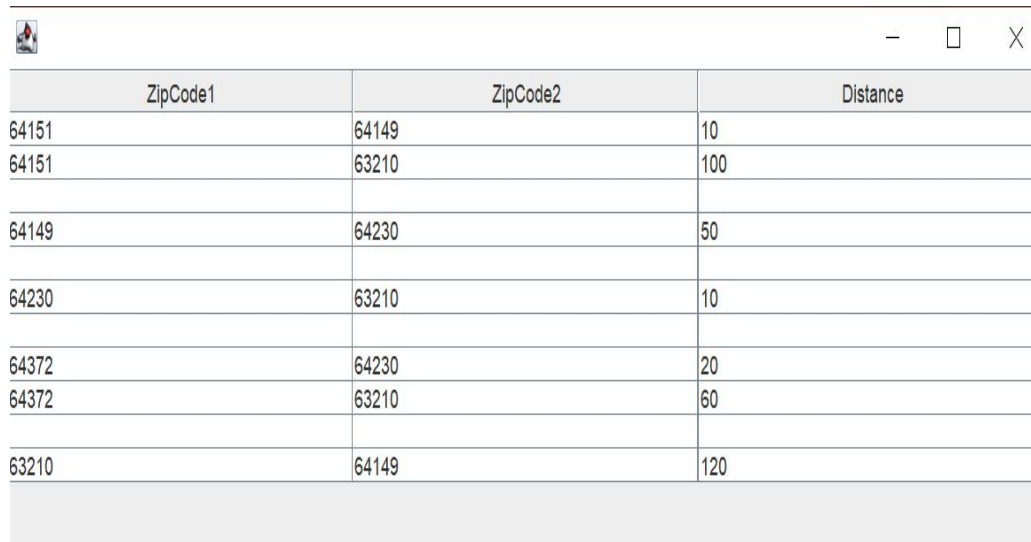


Work Flow

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

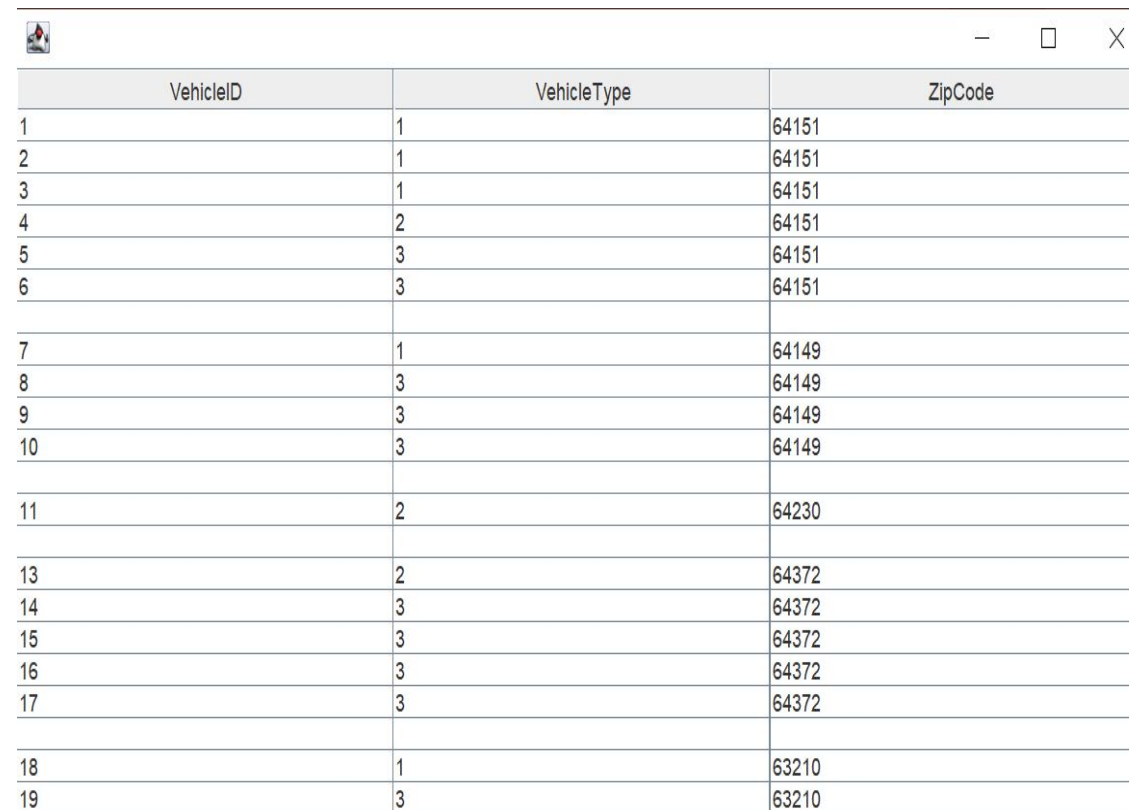
Step 1 – Creating Tables

TABLE 1 GIVES THE VEHICLE DETAILS.



ZipCode1	ZipCode2	Distance
64151	64149	10
64151	63210	100
64149	64230	50
64230	63210	10
64372	64230	20
64372	63210	60
63210	64149	120

TABLE 2 GIVES DISTANCE BETWEEN TWO ZIP CODES.



VehicleID	VehicleType	ZipCode
1	1	64151
2	1	64151
3	1	64151
4	2	64151
5	3	64151
6	3	64151
7	1	64149
8	3	64149
9	3	64149
10	3	64149
11	2	64230
13	2	64372
14	3	64372
15	3	64372
16	3	64372
17	3	64372
18	1	63210
19	3	63210

Step – 2 Input the zipcode

- The dialogue box is created by importing `javax.swing.JOptionPane`



Step 3 – Finding Shortest Path

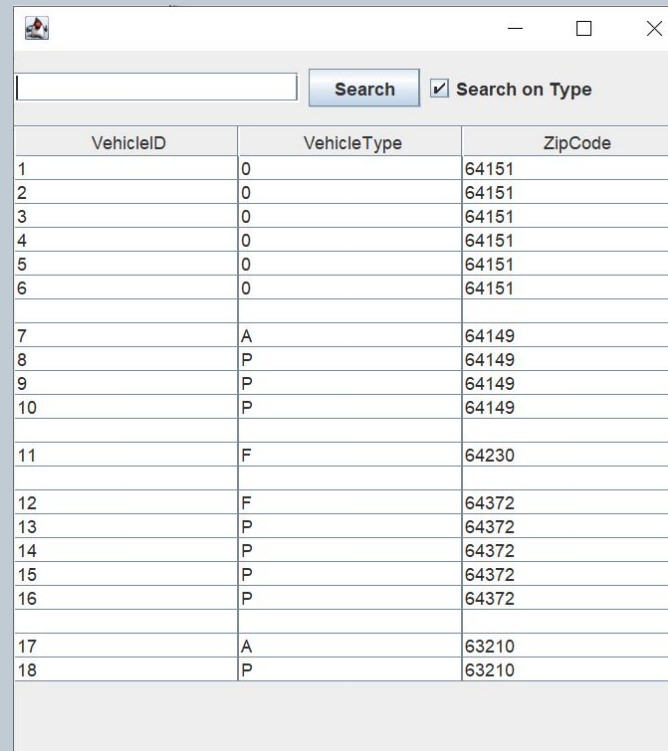
```
Given zipcode is: 64151

Using Dijkstra's Algo,
shortest paths from given zipcodes are displayed below
Distance to 64151: 0.0
Distance to 64149: 10.0
Distance to 64230: 30.0
Distance to 64372: 50.0
Distance to 63210: 60.0
Selected vehicle P is available at 64149
Selected vehicle P is available at 64149
Selected vehicle P is available at 64149
Selected vehicle P is available at 64372
Selected vehicle P is available at 64372
Selected vehicle P is available at 64372
Selected vehicle P is available at 64372
Selected vehicle P is available at 63210
```

- We displayed the shortest paths calculated from the given location the all other zip codes available in the locality using Dijkstra's algorithm in the CONSOLE.

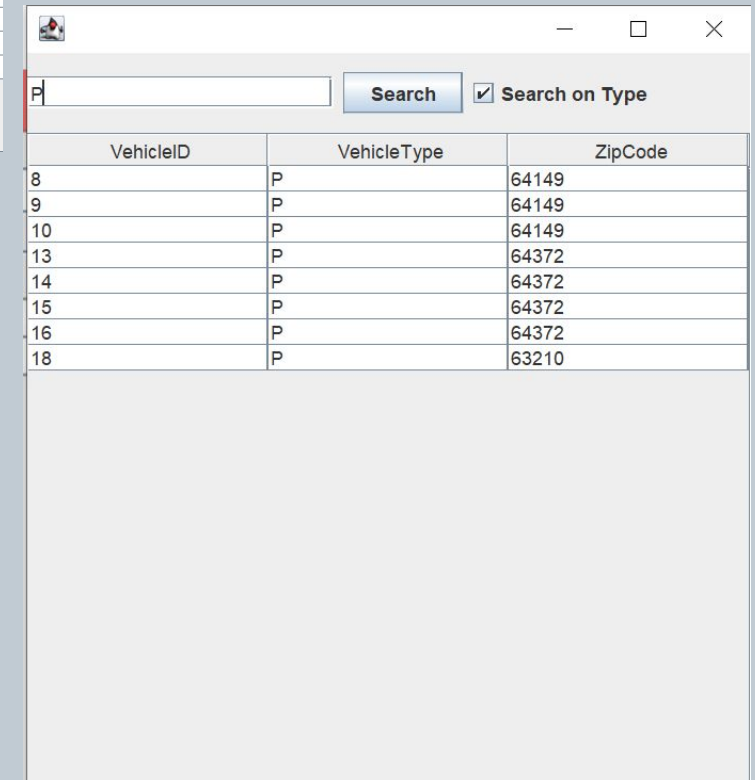
STEP 4 – AVAILABILITY CHECK

- In order to get the availability, we have implemented a search function where the emergency vehicle type (A, F or P) is entered in the Search Dialogue Box.
- The zip codes are displayed along with the vehicle ID.



The screenshot shows a search window with a search bar containing '0', a 'Search' button, and a checked 'Search on Type' checkbox. The table below displays the results of this search.

	VehicleID	VehicleType	ZipCode
1		0	64151
2		0	64151
3		0	64151
4		0	64151
5		0	64151
6		0	64151
7		A	64149
8		P	64149
9		P	64149
10		P	64149
11		F	64230
12		F	64372
13		P	64372
14		P	64372
15		P	64372
16		P	64372
17		A	63210
18		P	63210

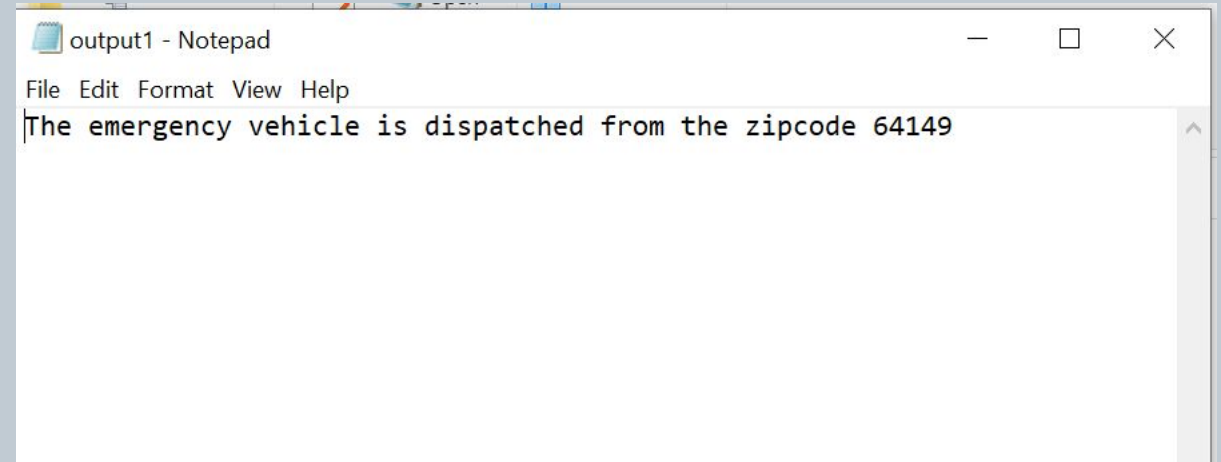


The screenshot shows the same search window, but the search bar now contains 'P'. The table below displays the results of this search.

	VehicleID	VehicleType	ZipCode
8		P	64149
9		P	64149
10		P	64149
13		P	64372
14		P	64372
15		P	64372
16		P	64372
18		P	63210

STEP 5- DISPATCH VEHICLE ALLOCATION

- Read string sCurrentLine.
- Printing the dispatch vehicle zipcode in a new output1 file.



A screenshot of a Windows Notepad application window. The title bar reads "output1 - Notepad". The menu bar includes "File", "Edit", "Format", "View", and "Help". The text area contains the string "The emergency vehicle is dispatched from the zipcode 64149".

```
output1 - Notepad
File Edit Format View Help
The emergency vehicle is dispatched from the zipcode 64149
```

Complexity Calculation

We have used DIJKSTRA's Algorithm which gives the efficient time complexity of **$O(E \log V)$**
where E – No. of Edges , V – No. of Vertices

References

- ❖ <https://www.youtube.com/watch?v=EbL1pj3tOgQ>
- ❖ <https://pdfs.semanticscholar.org/d69a/69142f67573c9584eb6e220ca749b2bf30bb.pdf>
- ❖ <https://stackoverflow.com/questions/8265307/file-input-for-dijkstras-algorithm>
- ❖ <https://stackoverflow.com/questions/4615814/dijkstra-and-fileinput-java>
- ❖ <https://stackoverflow.com/questions/1994255/how-to-write-console-output-to-a-txt-file>
- ❖ <https://stackoverflow.com/questions/22066387/how-to-search-an-element-in-a-jtable-java>

Thank You!

We are open to questions now.

π