A

Project Report On

EMERGENCY VEHICLE DISPATCH SYSTEM

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

COMPUTER SCIENCE AND ENGINEERING

BY

NIKHIL R (1602-19-733-082)

RAMAKANTH SVN (1602-19-733-118)

Under the guidance of

Dr.V.Sireesha, ASSISTANT PROFESSOR

Department of Computer Science Engineering

Vasavi College Of Engineering

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31 2021

ACKNOWLEDGEMENT

With the grace of god ad with successful completion of the work, we wish to express thanks to our sincere thanks to our dynamic and beloved principal *DR.S. V. RAMANA SIR* for giving us an opportunity of doing this mini-project work. We are greatly thankful to our head of the department **Dr.T. Adilakshmi Madam** and guides **Dr.V.Sireesha** for their encouragement given. We are also thankful to their imparting knowledge with us during preparation of this mini project.

TABLE OF CONTENTS

1. ABSTRACT

OBJECTIVE:

The objective of this project is to find and allocate an available emergency vehicle, taking source and destination as inputs and providing an optimum path during the course of travel.

WORKING:

Here the user first needs to select the source and then the user is asked to either choose a hospital or a nearby hospital from the list of hospitals. If due to any technical reasons like non availability of hospital beds, closed admissions, lack of medical equipment in destined hospital or need of increase in criticality of case and advanced medical supervision is required, a grace period of 10 minutes will be given to confirm the status of admission and the ambulance will reroute to the nearby hospital or the hospital selected by the use and in case it also fails, re-routing to a different hospital gets enabled.

ALGORITHM:

Dijkstra algorithm is implemented in this project to find the shortest path between source and destination.

2. INTRODUCTION

In this approach initially a map is shown which consists of different locations and hospitals. Each of which is a node. Source location and hospital destination are to be selected from the map. Optimum path and estimated fare are displayed to the user. Based on the availability, an emergency vehicle would be dispatched to the source location. Now, given the distance and traffic optimal path is generated and the vehicle is ready to go. In case of any complications at the destined location, re-routing of path to a near by hospital is also made available. Based on these the fair would be charged at the end.

Attributes Provided:

  • Places as nodes
 • Distance and traffic density as the edges weight with connected vertices

During the course of travel since the traffic density isn't constant, different traffic densities are allocated between each of the locations. The densities range from low to very high. As the traffic density increases, weight of the path is increased. So the distance and traffic densities are deciding factors for weight of that particular path. Based on the traffic density levels, average speed for each level is fixed. The average of all paths speeds is considered as final estimated speed. Based on the distance and traffic density weight is decided and shortest weighed path is chosen as optimal path and is displayed to the user.Considering the distance and traffic fare is generated and all the details are stored in the files including the source,destination,fare,user's info and the time of booking.
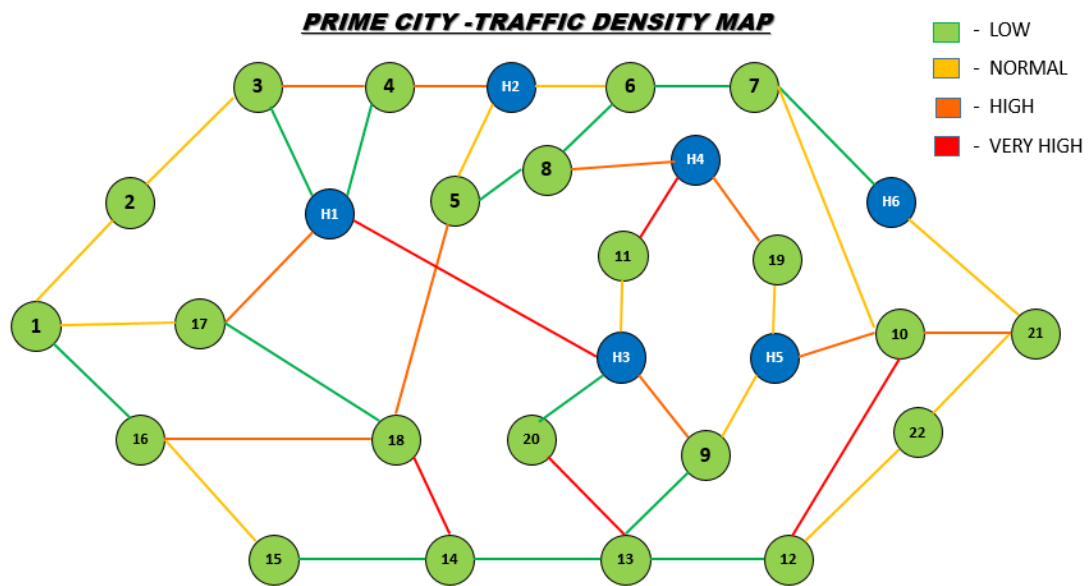
ASSUMPTIONS:

  • Traffic densities are fixed between two locations
  • Average speed is assumed based on traffic density

3. DESIGN

In this project, name of the city considered is "Prime City". A total of 22 locations and 6 hospitals are considered in the map and are connected in various ways. Below is the demonstration of the city map. Blue colored vertices indicate hospitals and green ones indicate locations.

PRIME CITY - LOCATION MAP
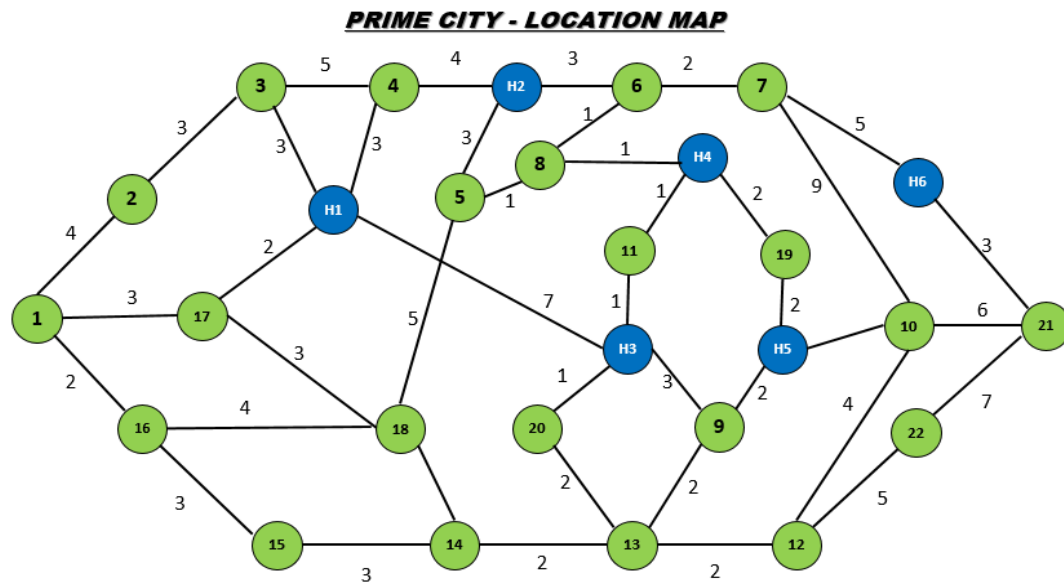
Keeping in view the traffic densities between locations of the given city, the traffic hierarchy is classified into 4 stages. Stage-1 indicates low traffic, stage-2 indicates normal traffic, stage-3 indicates high traffic and stage-4 indicates very high traffic (possibly railway gates or tollgates of any sort). Below is the demonstration of Traffic density map.

PRIME CITY - TRAFFIC DENSITY MAP

Legend:
- LOW
- NORMAL
- HIGH
- VERY HIGH

When the source location is selected, the user is asked to either choose a hospital or a nearby hospital would be allocated. If due to any technical reasons like non availability of hospital beds, closed admissions, lack of medical equipment in destined hospital or need of increase in criticality of case and advanced medical supervision is required, a grace period of 10 minutes will be given to confirm the status of admission. In case of a any issue, re-routing to a different hospital gets enabled.

4. IMPLEMENTATION OF CODE

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<conio.h>
#include<dos.h>
#define INF 99999999
int size,source_chosen,destintion_chosen;
int i,j,bookingStatus,choice;;
int hosp=22,optimization_path,reroute;;
int status,finalCost,reroutedFare=0;
int allDistances[28];
char hospitalName[50]= "";//Empty string for hosp name
char sourceName[50]= "";//Empty string for source name
int total_graph[28][28];// Graph for both distance and traffic
int graph[28][28];//graph for distances
int traffic_graph[28][28];// graph for traffic
void choose();
void signup();
void login();
void display();
void Hospitaldetermine(int);

struct user
{
 char name[20];
 char password[20];
 long long int phoneNumber;
 char gmail[30];
};
void signup()
{
struct user u1;
FILE *fp;


fp=fopen("D:\\nikhil.txt","a");
if(fp==NULL)
{
 printf("\n file does not exist");

}
printf("\n enter the name\n");
scanf("%s",u1.name);
fprintf(fp,"Name = %s \n",u1.name);
printf(" enter the password\n");
```

```c
scanf("%s",u1.password);
fprintf(fp,"Password = %s \n",u1.password);
printf(" enter the phoneNumber\n");
scanf("%lli",&u1.phoneNumber);
fprintf(fp,"PhoneNumber = %lli \n",u1.phoneNumber);
printf(" enter the gmail\n");
scanf("%s",u1.gmail);
fprintf(fp,"Gmail = %s \n",u1.gmail);
fclose(fp);
printf("\nAccount created Successfully\n");
choose();
}

void login()
{

FILE* fp;
int wordExist1=0;
int bufferLength1 = 255;
char search1[100];
printf("Enter Username\n");
scanf("%s",search1);

char line1[bufferLength1];
fp = fopen("D:\\nikhil.txt", "r");
while(fgets(line1, bufferLength1, fp))
{
 char *ptr1 = strstr(line1, search1);
 if (ptr1 != NULL)
 {
  wordExist1=1;
  break;
 }
}
fclose(fp);

int wordExist2=0;
int bufferLength2 = 255;
char search2[100];
printf("Enter Password\n");
scanf("%s",search2);

char line2[bufferLength2];
fp = fopen("D:\\nikhil.txt", "r");
while(fgets(line2, bufferLength2, fp))
{
 char *ptr2 = strstr(line2, search2);
```

```c
  if (ptr2 != NULL)
  {
   wordExist2=1;
   break;
  }
 }
 fclose(fp);

 if(wordExist1==1 && wordExist2==1)
 {
  printf("\n Successfully Logged In!");
 }
 else
 {
  printf("\n No such account exists");
  exit(0);
 }


}



void display()
{
 FILE *fp;
 char ch;

printf("\n=====================================================
=========================================\n\n");
 printf("\t\t WELCOME TO \n \n");
 printf("\t\t PRIME HEALTH CARE \n \n");
 fp=fopen("D:\\nikhil.txt","r");
 ch=getc(fp);

 while(ch!=EOF)
 {
  printf("%c",ch);
  ch=getc(fp);
 }


printf("\n=====================================================
========================================\n\n");
 fclose(fp);
 choose();
}
```

```c
void dijkstra(int graph[][size], int source)
{
int cost[size][size],distance[size];
int traffic_graph[28][28];
int visited[size], i, j, cnt, minDistance, nextNode;
int optimization_path;
// Initialize cost matrix
for(i = 0; i < size; i++)
{
   for(j = 0; j < size; j++)
   {
      if(graph[i][j] == 0)
      {
         cost[i][j] = INF;//If source and node aren't directly connected
      }
      else
      {
      cost[i][j] = graph[i][j];//If source and node are directly connected
      }
   }
}
// distance of roots directly conneted to source
for(i = 0; i < size; i++)
{
distance[i] = cost[source][i];// Distance from itself = 0
visited[i] = 0;// since traversing starts from the source, that node is clearly
visited
}
distance[source] = 0;// source is the starting point
visited[source] = 1;
cnt = 1;
while(cnt < size - 1)
{
   minDistance = INF;
   for(i = 0; i < size; i++)
   {
      if(distance[i] < minDistance && !visited[i]) //Adjacent and un-visited
      {
         minDistance = distance[i];
         nextNode = i;
      }
   }
visited[nextNode] = 1;
for(i = 0; i < size; i++) //Relaxation function
{
   if(!visited[i])
```

```c
        {
            if(minDistance + cost[nextNode][i] < distance[i])
            {
                distance[i] = minDistance + cost[nextNode][i] ;

            }
        }
    } cnt++;
    }
    for(i = 0; i < size; i++)
    {
        allDistances[i]=distance[i];
        //printf("%d ", allDistances[i]);
    }
}
void costfunction( int units)//Fare estimation function
{
    finalCost=units*25;
    reroutedFare+=finalCost;
    printf("\nEstimated fare for given distance is %d
rupees\n",reroutedFare);
    if(reroute==1)
    printf("\n re-routing charges are %d\n",finalCost);
    // Fare per one kilometer is 15 rupees

}
void Hospitaldetermine(int hos)// Resolving name to destination
{
    hospitalName[0]= '\0';
if(hos+1==23)
{
//hospitalName[50] = Central Los Santos Medical Center';
 strcat(hospitalName,"Central Los Santos Medical Center");
}
if(hos+1==24)
strcat(hospitalName,"Mount Zonah Medical Center");
if(hos+1==25)
strcat(hospitalName,"Sandy Shores Medical Center");
if(hos+1==26)
strcat(hospitalName,"The Bay Care Center");
if(hos+1==27)
strcat(hospitalName,"Portola Trinity Medical Center");
if(hos+1==28)
strcat(hospitalName,"St.Fiacre Medical Center");

}
void sourceDetermine(int source_chosen)// Resolving name to source
```

```c
{

if(source_chosen==1)
strcat(sourceName,"Georgopol");
if(source_chosen==3)
strcat(sourceName,"Novo");
if(source_chosen==4)
strcat(sourceName,"Lipovka");
if(source_chosen==5)
strcat(sourceName,"SanMartin");
if(source_chosen==6)
strcat(sourceName,"YasnayaPolana");
if(source_chosen==7)
strcat(sourceName,"Severny");
if(source_chosen==8)
strcat(sourceName,"Pecado");
if(source_chosen==9)
strcat(sourceName,"Gronnus");
if(source_chosen==10)
strcat(sourceName,"Blomster");
if(source_chosen==11)
strcat(sourceName,"Sosnovka");
if(source_chosen==12)
strcat(sourceName,"Pochinki");
if(source_chosen==13)
strcat(sourceName,"Miramar");
if(source_chosen==14)
strcat(sourceName,"Livik");
if(source_chosen==15)
strcat(sourceName,"Sanhok");
if(source_chosen==16)
strcat(sourceName,"Rozhok");
if(source_chosen==17)
strcat(sourceName,"LosLeones");
if(source_chosen==18)
strcat(sourceName,"Gatka");
if(source_chosen==19)
strcat(sourceName,"Zharki");
if(source_chosen==20)
strcat(sourceName,"NorthYanton");
if(source_chosen==21)
strcat(sourceName,"Mylta");
if(source_chosen==22)
strcat(sourceName,"Midstein");
}
void choose ()
{
```

```c
    printf("\n");
y : printf("\n 1.SIGNUP");
 printf("\n 2.LOGIN");
 printf("\n 3.DISPLAY ");
 printf("\n 4.EXIT");
 printf("\n Enter your choice\n");
 scanf("%d",&choice);
 switch(choice)
 {
  case 1:signup();
  break;
  case 2:login();
  break;
  case 3:display();
  break;
  case 4:exit(1);
  default :printf("Invalid choice");

 }
}

void main()
{
FILE *fp;
printf("\n\t\t*****WELCOME TO PRIME CITY'S AMBULANCE DISPATCH
SERVICE*****\n");
choose();// Account creation and credential verification
size = 28;
int graph[28][28]={

            {0,4,0,0,0,0,0,0,0,0,0,0,0,0,2,3,0,0,0,0,0,0,0,0,0,0,0,0},
            {4,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            {0,3,0,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0,0},
            {0,0,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,4,0,0,0,0,0},
            {0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,5,0,0,0,0,3,0,0,0,0,0},
            {0,0,0,0,0,0,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0},
            {0,0,0,0,0,2,0,0,0,9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5},
            {0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0},
            {0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,3,0,2,0},
            {0,0,0,0,0,9,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,4,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0},
            {0,0,0,0,0,0,0,0,0,4,0,0,2,0,0,0,0,0,0,0,5,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,2,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,2,0,3,0,0,5,0,0,0,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,3,0,0,0,0,0,0,0,0,0,0},
            {2,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,4,0,0,0,0,0,0,0,0,0},
            {3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,2,0,0,0,0,0},
```

```c
                {0,0,0,0,0,0,0,0,0,0,0,0,5,0,4,2,0,0,0,0,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0},
                {0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0},
                {0,0,0,0,0,0,0,0,0,6,0,0,0,0,0,0,0,0,0,7,0,0,0,0,0,3},
                {0,0,0,0,0,0,0,0,0,0,5,0,0,0,0,0,0,0,7,0,0,0,0,0,0,0},
                {0,0,3,3,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,7,0,0,0},
                {0,0,0,4,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,2,0,1,0,0,0,0,0,0,0,1,0,0,7,0,0,0,0,0},
                {0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,2,4,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,5,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0,0,0}
        };

        int traffic_graph[28][28]={

                {0,2,0,0,0,0,0,0,0,0,0,0,0,0,1,2,0,0,0,0,0,0,0,0,0,0,0},
                {2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                {0,2,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0},
                {0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,1,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0,2,0,0,0,0},
                {0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0},
                {0,0,0,0,0,1,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1},
                {0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0},
                {0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,3,0,2,0},
                {0,0,0,0,0,0,4,0,0,0,2,0,0,0,0,0,0,0,0,0,1,0,0,0,3,0},
                {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,4,0,0},
                {0,0,0,0,0,0,0,0,4,0,0,2,0,0,0,0,0,0,1,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,4,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,4,0,0,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,0,0},
                {1,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,2,0,0,0,0,0,0,0,0,0},
                {2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,3,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,0,0,0,0,0,4,0,2,1,0,0,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,2,0},
                {0,0,0,0,0,0,0,0,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,1,0,0,0},
                {0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,2},
                {0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0},
                {0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,3,0,0,0,0,0,0,4,0,0,0},
                {0,0,0,3,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,3,0,2,0,0,0,0,0,0,1,0,0,4,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,3,0,0,4,0,0,0,0,0,0,3,0,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,0,0,2,3,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0},
                {0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0}

        };
```

```c
for(i = 0; i < size; i++)
{
  for(j = 0; j < size; j++)
  {
    total_graph[i][j]=graph[i][j]+traffic_graph[i][j];
  }
}
printf("Select source from the following:\n");
printf("****Prime City
Locations***\n1.Georgopol\n2.LosSantos\n3.Novo\n4.Lipovka\n5.SanMart
in\n6.YasnayaPolana\n7.Severny\n8.Pecado\n9.Gronnus\n10.Blomster\n11
.Sosnovka\n12.Pochinki\n13.Miramar\n14.Livik\n15.Sanhok\n16.Rozhok\n
17.LosLeones\n18.Gatka\n19.Zharki\n20.NorthYanton\n21.Mylta\n22.Mids
tein\n");
scanf("%d",& source_chosen);
sourceDetermine(source_chosen);
q : printf("Select destintion from the following:\n");
printf("\t\t***Hospitals***\n1.Central Los Santos Medical
Center\n2.Mount Zonah Medical Center\n3.Sandy Shores Medical
Center\n4.The Bay Care Center\n5.Portola Trinity Medical
Center\n6.St.Fiacre Medical Center\n\tOR\n\n");
printf("Enter 0 to choose shortest hospital\n");
scanf("%d",&destintion_chosen);
dijkstra(total_graph, source_chosen-1);
r: printf("\n");
int shortestDistance=allDistances[22];

i=22;
int p=0;//array index of same optimal distance paths
while(i<28)
{

  if(allDistances[i]<shortestDistance)//for equal paths make <=
  {
    //if(allDistances[i]==shortestDistance)
    //{
    // similar_distances[p]=allDistances[i];
      //p++;
    //}
    //else

      shortestDistance=allDistances[i];
      hosp=i;
  }
  i++;
}
if(shortestDistance==0)//while re-routing if same destintion is chosen
```

```c
{
   allDistances[hosp]=999999;
   goto r;
}


if (destintion_chosen==0)//Nearest hospital
{
   printf("\nConsidering traffic and distance :\n");
   Hospitaldetermine(hosp+1);
   printf("Nearest hospital is : %s",hospitalName);
   printf("\nIt is %d kms away\n\n",shortestDistance);
   costfunction(shortestDistance);


}
else //Manually chosen destintion
 {
   while(destintion_chosen < 7)
   {
      hosp = destintion_chosen+21;
      Hospitaldetermine(hosp+1);
      printf("\nIt is %d kms away",allDistances[destintion_chosen+21]);
      //destination gets coverted to node number in city [(1-6) + 21 = (22-
27)]
      costfunction(allDistances[destintion_chosen+21]);
      break;
   }
}
printf("Would you like to confirm your ambulance
booking?\npress(1/0)\n");
scanf("%d",&status);
if(status==1)
{
   printf("Ambulance booked successfully!!\n");
}
if(status)
{
      fp=fopen("D:\\nikhil.txt","a");

   printf("\n\t***DETAILS REGARDING DISPATCH***\n\n");
   printf("source : %s\n",sourceName);
   printf("destination : %s\n",hospitalName);
   printf("fare : %d rupees\n",reroutedFare);
   printf("Booking date is : %s \n",__DATE__);
      printf("Booking time is : %s \n",__TIME__);
```

```c
        fprintf(fp,"source : %s\n",sourceName);
    fprintf(fp,"destination : %s\n",hospitalName);
    fprintf(fp,"fare : %d rupees\n",reroutedFare);
    fprintf(fp,"Booking date is : %s \n",__DATE__);
        fprintf(fp,"Booking time is : %s \n",__TIME__);

        fclose(fp);

}
/*for re-routing:
make destination as source =>
destination is hospital so its node number > 22 now make that node as
source

now ask to choose destination =>
exclude present hosp from destination list and show; shortest is available;
store destination

apply dijkstra*/

printf("would u like to reroute?(1/0)\n");
scanf("%d",&reroute);
if(reroute==1)
{

    char hospitalName[50]= "";//Empty string for hosp name
    char sourceName[50]= "";//Empty string for source name
    source_chosen=hosp+1;
    printf("\n%s\n",hospitalName);
    goto q;


}
```
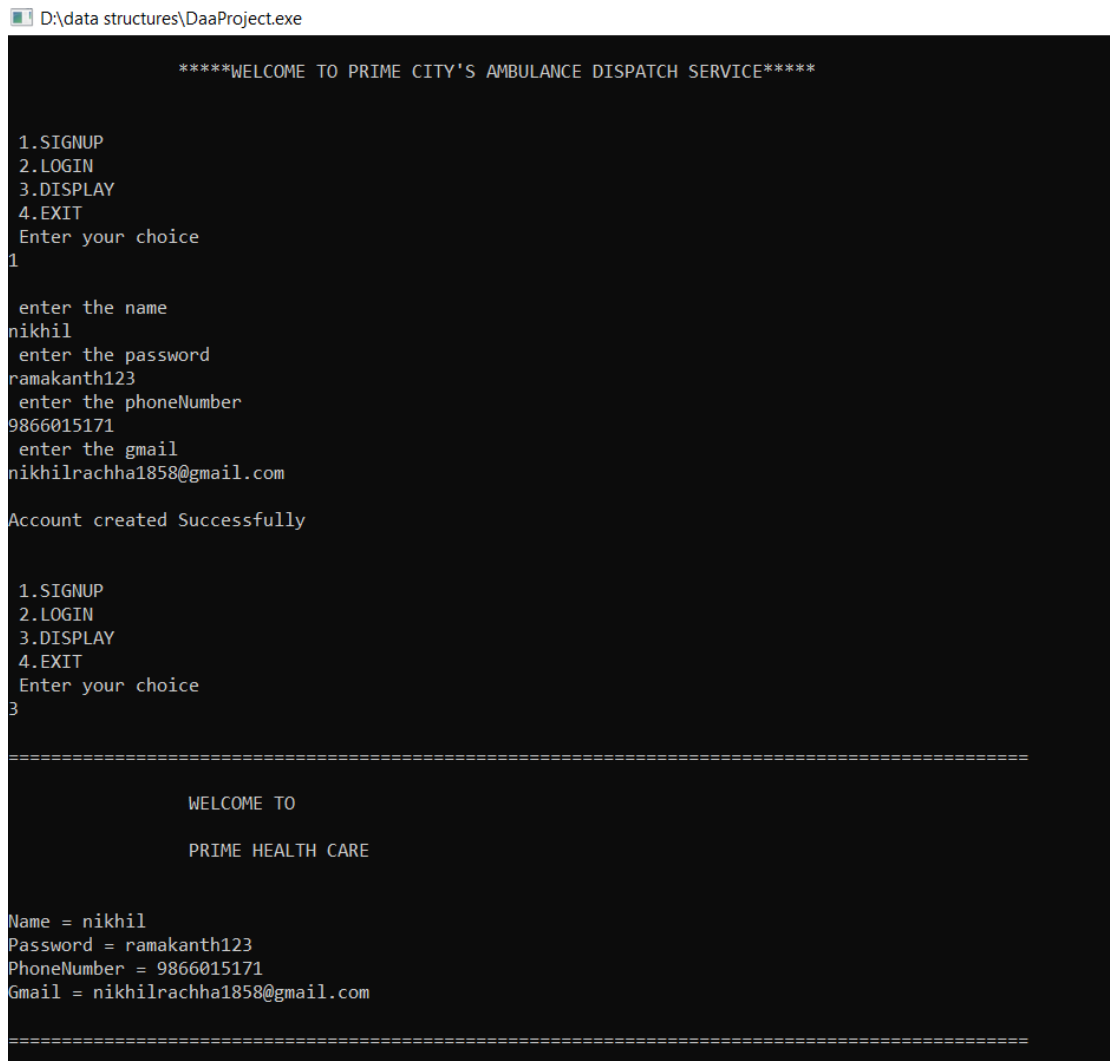
6. OUTPUT SCREENSHOTS

SIGNUP UP BY ENTERING YOUR DETAILS IN THE EMERGENCY VEHICLE DISPATCH SERVICE

D:\data structures\DaaProject.exe

```
 Successfully Logged In!Select source from the following:
****Prime City Locations***
1.Georgopol
2.LosSantos
3.Novo
4.Lipovka
5.SanMartin
6.YasnayaPolana
7.Severny
8.Pecado
9.Gronnus
10.Blomster
11.Sosnovka
12.Pochinki
13.Miramar
14.Livik
15.Sanhok
16.Rozhok
17.LosLeones
18.Gatka
19.Zharki
20.NorthYanton
21.Mylta
22.Midstein
12
Select destintion from the following:
               ***Hospitals***
1.Central Los Santos Medical Center
2.Mount Zonah Medical Center
3.Sandy Shores Medical Center
4.The Bay Care Center
5.Portola Trinity Medical Center
6.St.Fiacre Medical Center
        OR

Enter 0 to choose shortest hospital
0
```

Choose the source location from the given 22 locations in the map and and enter the destination to which hospital would you like to go or else enter 0 to get to the nearest Hospital from the source.

```
12
Select destintion from the following:
              ***Hospitals***
1.Central Los Santos Medical Center
2.Mount Zonah Medical Center
3.Sandy Shores Medical Center
4.The Bay Care Center
5.Portola Trinity Medical Center
6.St.Fiacre Medical Center
        OR

Enter 0 to choose shortest hospital
0


Considering traffic and distance :
Nearest hospital is : Mount Zonah Medical Center
It is 9 kms away


Estimated fare for given distance is 225 rupees
Would you like to confirm your ambulance booking?
press(1/0)
1
Ambulance booked successfully!!
```

Based on the hospital chosen,fare is generated considering the parameters such as distance and traffic between the two places.

Enter 1 to confirm the hospital booking.

```
D:\data structures\DaaProject.exe

Estimated fare for given distance is 225 rupees
Would you like to confirm your ambulance booking?
press(1/0)
1
Ambulance booked successfully!!

        ***DETAILS REGARDING DISPATCH***

source : Pochinki
destination : Mount Zonah Medical Center
fare : 225 rupees
Booking date is : Jun 21 2021
Booking time is : 17:46:57
would u like to reroute?(1/0)
1


Select destintion from the following:
            ***Hospitals***
1.Central Los Santos Medical Center
2.Mount Zonah Medical Center
3.Sandy Shores Medical Center
4.The Bay Care Center
5.Portola Trinity Medical Center
6.St.Fiacre Medical Center
        OR

Enter 0 to choose shortest hospital
2
```

All details related to the the dispatch are shown which consists of the source,destination,fare,booking date and time.

If you want to re-route to the nearby hospital or any other hospital re-reouting is enabled.

Enter 0 to re-route to nearby hospital or choose your option from the remaining hospitals.

```
D:\data structures\DaaProject.exe
4.The Bay Care Center
5.Portola Trinity Medical Center
6.St.Fiacre Medical Center
        OR

Enter 0 to choose shortest hospital
2



It is 22 kms away
Estimated fare for given distance is 775 rupees

 re-routing charges are 550
Would you like to confirm your ambulance booking?
press(1/0)
1
Ambulance booked successfully!!

        ***DETAILS REGARDING DISPATCH***

source : Pochinki
destination : Sandy Shores Medical Center
fare : 775 rupees
Booking date is : Jun 21 2021
Booking time is : 17:46:57
would u like to reroute?(1/0)
0

-------------------------------
Process exited after 121.3 seconds with return value 0
Press any key to continue . . . _
```

The dispatch has been successfully done and all the details regarding to the user's ride are been displayed.

6. TEST CASES

Test case 1:Choosing the source from the given location and showing the shortest path .

```
 Successfully Logged In!Select source from the following:
****Prime City Locations***
1.Georgopol
2.LosSantos
3.Novo
4.Lipovka
5.SanMartin
6.YasnayaPolana
7.Severny
8.Pecado
9.Gronnus
10.Blomster
11.Sosnovka
12.Pochinki
13.Miramar
14.Livik
15.Sanhok
16.Rozhok
17.LosLeones
18.Gatka
19.Zharki
20.NorthYanton
21.Mylta
22.Midstein
12
Select destintion from the following:
             ***Hospitals***
1.Central Los Santos Medical Center
2.Mount Zonah Medical Center
3.Sandy Shores Medical Center
4.The Bay Care Center
5.Portola Trinity Medical Center
6.St.Fiacre Medical Center
        OR

Enter 0 to choose shortest hospital
0
```

Test case 2: Confirming the dispatch and providing the details of the dispatched service.

D:\data structures\DaaProject.exe

```
Estimated fare for given distance is 225 rupees
Would you like to confirm your ambulance booking?
press(1/0)
1
Ambulance booked successfully!!

        ***DETAILS REGARDING DISPATCH***

source : Pochinki
destination : Mount Zonah Medical Center
fare : 225 rupees
Booking date is : Jun 21 2021
Booking time is : 17:46:57
would u like to reroute?(1/0)
1


Select destintion from the following:
              ***Hospitals***
1.Central Los Santos Medical Center
2.Mount Zonah Medical Center
3.Sandy Shores Medical Center
4.The Bay Care Center
5.Portola Trinity Medical Center
6.St.Fiacre Medical Center
        OR

Enter 0 to choose shortest hospital
2
```

Test case 3: Re-routing to the nearby hospital or any other hospital.


```
D:\data structures\DaaProject.exe

Estimated fare for given distance is 225 rupees
Would you like to confirm your ambulance booking?
press(1/0)
1
Ambulance booked successfully!!

        ***DETAILS REGARDING DISPATCH***

source : Pochinki
destination : Mount Zonah Medical Center
fare : 225 rupees
Booking date is : Jun 21 2021
Booking time is : 17:46:57
would u like to reroute?(1/0)
1


Select destintion from the following:
            ***Hospitals***
1.Central Los Santos Medical Center
2.Mount Zonah Medical Center
3.Sandy Shores Medical Center
4.The Bay Care Center
5.Portola Trinity Medical Center
6.St.Fiacre Medical Center
        OR

Enter 0 to choose shortest hospital
2
```

7.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 rescue station, thus to shorten the response time for future incidents. In this paper, we analyze the key constraints and relevant punishments, of vehicle dispatching and redistribution problem, and quantify the punishment coefficient in levels. Then we point out that the emergency vehicle dispatching and redistribution problem is a bilevel programming problem, as a result of which dijstra algorithm is efficient to solve the problem.

8. REFERENCES

Algorithm Design text book :Jon Klienberg and Eva tardos

 https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/

 https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/