Project Documentation

on

Automated Traffic Monitoring System

SAYANIKA BHOWMIK (Reg. no.: 201900082)

Table of contents

Automated Traffic Monitoring System

Introduction-

The project “Automated Traffic Monitoring System” is being proposed to bring substantial changes in the Traffic Enforcement system. The system has been conceived with the following vision.

- To bring transparency in enforcement of traffic laws and rules and in challenging traffic violations.
- To reduce rash and negligent driving by quality enforcement.
- To avoid conflicts between police & public during traffic law enforcement
- To increase awareness of traffic rules and regulations among road users
- To expedite processing of traffic violation cases and ensure speedy disposal.

Abbreviations

SRS	Software Requirement Specification
SPMP	Software Project Management Plan
TMS	Traffic Monitoring System
DBMS	Database Management System
OS	Operating System

Software Project Management Plan

Evolution of the SPMP

This document is subject to changes. The assumptions, dependencies and constraints for the project, the detailed time and resource planning for each phase as mentioned in this SPMP can change during the project. Changes in this information will lead to a new SPMP.

Resource Requirements

The primary resource needed to complete the project is the human resource. The distribution of roles and responsibilities are mentioned above. Other resource includes financial and development tools including

both hardware and software tools.

Hardware:

- A computer with at least 4GB RAM and i5 processor
- Basic peripherals
- OS: Windows 11/10/8/7 or UNIX or MAC

Software:

- C++
- Database (SQL)

Special Requirements:

- Good internet connectivity

Project Estimates

Development Effort and Time Estimation

So, according to the Basic COCOMO model, considering Organic project

Total Lines of Code (LOC) = 750

Total Kilo Lines of Code (KLOC) = 0.750

Hence, we can calculate the following-:

1. Effort = 2.4 (KLOC)^1.05 PM

Effort Estimation => 1.774 PM

2. Tdev = 2.5 (Effort)^0.38 Months

Time estimation => 3.109 Months

Critical Assumptions and Constraints

a) The Automated Traffic Monitoring System depends largely on the availability of data of all vehicles registered in the concerned state. It means, information regarding ownership data of vehicles registered in the State should be made available to traffic police for sending challans for traffic violations.
b) There shall be a mechanism for automatic transfer of ownership information from the Transport Department to the Traffic Police. In case of change of ownership, the transfer details should be automatically updated in the traffic police database, on a real-time basis.
c) It is assumed that the wireless or wired connectivity is available on mobile vehicles for data as well as voice in the area where the project will be implemented.
d) Initially the operation and use of hand held and related software should be imparted to the cities where this project is desired to be implemented.
e) On completion of projects the States will be willing to take upon themselves to continue the work and take the project objective down to all police station levels
f) District Traffic Management Centre experts will undertake training programs for resource building with respect to assisting and solving field problems and update software wherever required.
g) All the States will outsource the maintenance and management of the system initially till the department officers sufficiently assimilate required skills in maintaining and managing the system themselves.

Dependencies

The project requirements must be established and defined first before the designing stage. However, it is unavoidable that the Project Requirements might change during the Design Stage as new requirements arise.

Software Requirement Specification

Smart traffic management system goals

- Prioritize moderate traffic conditions by analyzing real-time traffic situations.
- Providing congestion-free traffic.
- Improvising traditional ticketing with an automated E-bill payment system.

- Speed sensors to warn commuters over speed violations.
- Provide a smart lighting system that reserves renewable energy sources.
- Offer safe and punctual public transportation.
- Eradicate pollution.
- Advanced traffic monitoring systems at intersections and narrow road ends to provide the right traffic guidance through GPS and GIS.
- Optimizing road networking systems, through building IoT, enabled quick and better communication systems.

Implementing This System Will Give Us/Outcome Measured:

- Save time & fuel cost
- Reduce Accident & CO_2 emission.
- To provide more transparency in enforcing traffic law by using cameras and e-enforcement.
- Effective e-governance to manage, monitor and administer.
- To empower police with a better system for enforcing traffic discipline.
- Facilitate in identifying frequent violators and initiate appropriation corrective action.
- To have complete data of motor vehicle owners address, license holder's particular, and violation particulars.

Objective of the project

The main objective of the entire activity is to automate and to manage the process of day to day activities of the traffic like: Record of Vehicles, Record of Challan, Search the Record of Vehicles, Traffic Control Booths, Control the traffic, Help.

SYSTEM DEVELOPMENT MODEL

AGILE SOFTWARE DEVELOPMENT (ASD)

Agile development is a conceptual framework mostly applied in software engineering projects. This type of

methodology minimizes risk by developing software in short time boxes; this is known as iteration, which

normally last from one to four weeks. There are a number of ASD methodologies: Dynamic System

Development Model, Crystal Methods and Scrum.

Scrum is a framework that is used to implement Agile development.

- Scrum is one of the agile methodologies designed to guide teams in the iterative and incremental delivery of a product. Often referred to as “an agile project management framework”.

- Traditional project management methods fix requirements in an effort to control time and cost; Scrum on the other hand, fixes time and cost in an effort to control requirements.

Features of the Enforcement Application Software:

1. Record of Vehicles
2. Record of Challan
3. Search the Record of Vehicles
4. Traffic Control Booths
5. Control the traffic
6. Help

To provide more transparency in enforcing traffic law by using cameras and e-enforcement.

Effective e-governance to manage, monitor and administer.

To empower police with a better system for enforcing traffic discipline.

Facilitate in identifying frequent violators and initiate appropriation corrective action.

To have complete data of motor vehicle owners address, license holder’s particular, and violation particulars.

Respective offenders automatically tracked and notices with enhanced fines are sent to repeated offenders.

Functional requirements

- It allows the officer to see through the record of the vehicle
- Search a Vehicle Using registration
- Search a Vehicle Using Name of the owner
- It allows the officer to apply for a new challan
- Search a Challan Using registration number
- Search a Challan Using Name of the owner
- Vehicle Registration

- See traffic of particular places and guide along with Vehicles Going out of the City & Vehicles coming into the City
- Help assistance for nearby hospital for emergencies

Non -functional requirements

Performance requirements

Performance requirements define response times for system functionality. Although the system is developed suiting for the least system performances, the performance of the system will highly depend on the performance of the hardware and software components of the installing computer. The load time for user interfaces, the time to verify vehicle information, time taken for returning query results should be fast.

Safety Requirements

There are several levels in traffic monitoring system, Access to the various subsystems will be protected by a user log in screen that requires a user name and password. This gives different views and accessible functions of user levels through the system. Maintaining backups ensure the system database security.

Software Quality Attributes

- Availability – The TMS shall be available all hours.
- Correctness – extent to which program satisfies specifications, fulfils user's mission objectives.
- Efficiency – How much smaller number of resources and time are required to achieve a particular task through the system.
- Flexibility – Ability to add new features to the system and handle then conveniently.
- Integrity – How the system would insecure the information in the system and how it avoids the data losses, referential integrity in database tables and interfaces.
- Maintainability – How easy is to keep the system as it is and correct defects with making changes.
- Portability – The TMS shall run in any OS.
- Usability – The TMS gives direct input on how real users use the system.

Design Phase

UML Diagrams for Hotel Management System

UML diagrams can be used as a way to visualize a project before it takes place or as documentation for a project afterward. The various Diagrams given below give a representation of how the HMS will be designed.

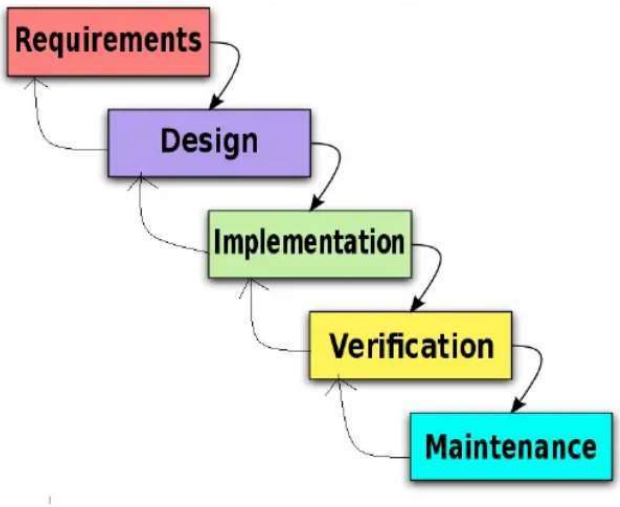
Class diagram- it illustrates the classes in a system, attributes and operations of each class and also the relationship between each class.

Use case diagram - It gives a graphic over-view of the actors involved in a system directly. It shows how different functions needed by the actors how they are interacted.

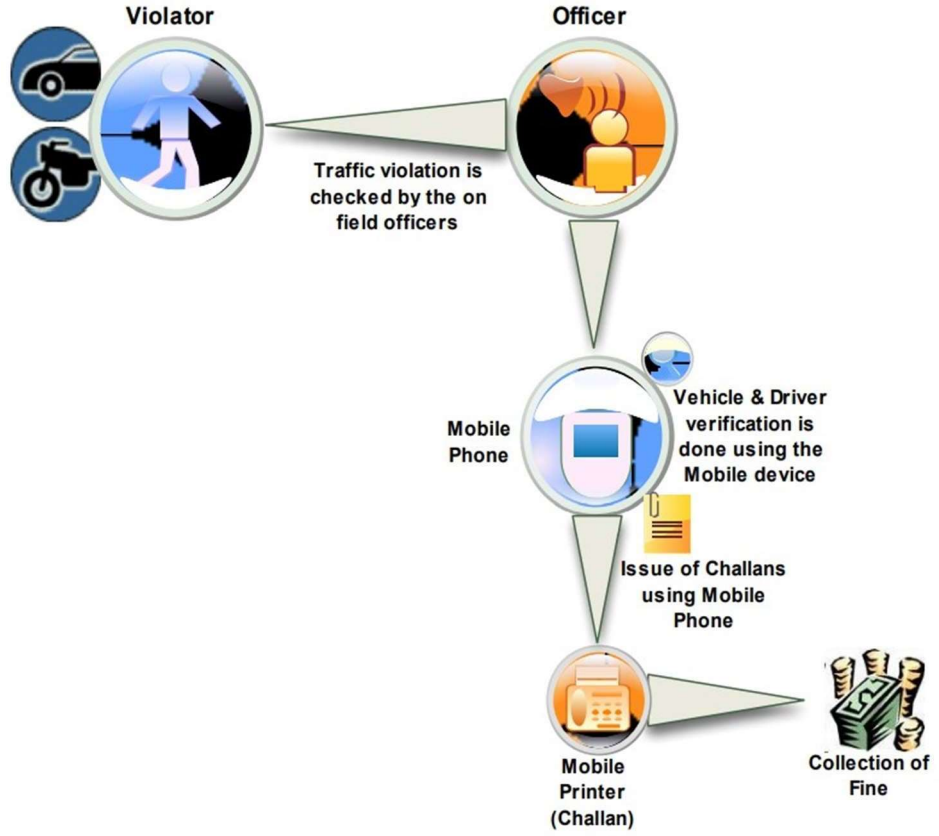
Sequence diagram - The main purpose of a sequence diagram is to define event sequences that result in some desired outcome.

Activity diagram - The main purpose of a sequence diagram is to define event sequences that result in some desired outcome.

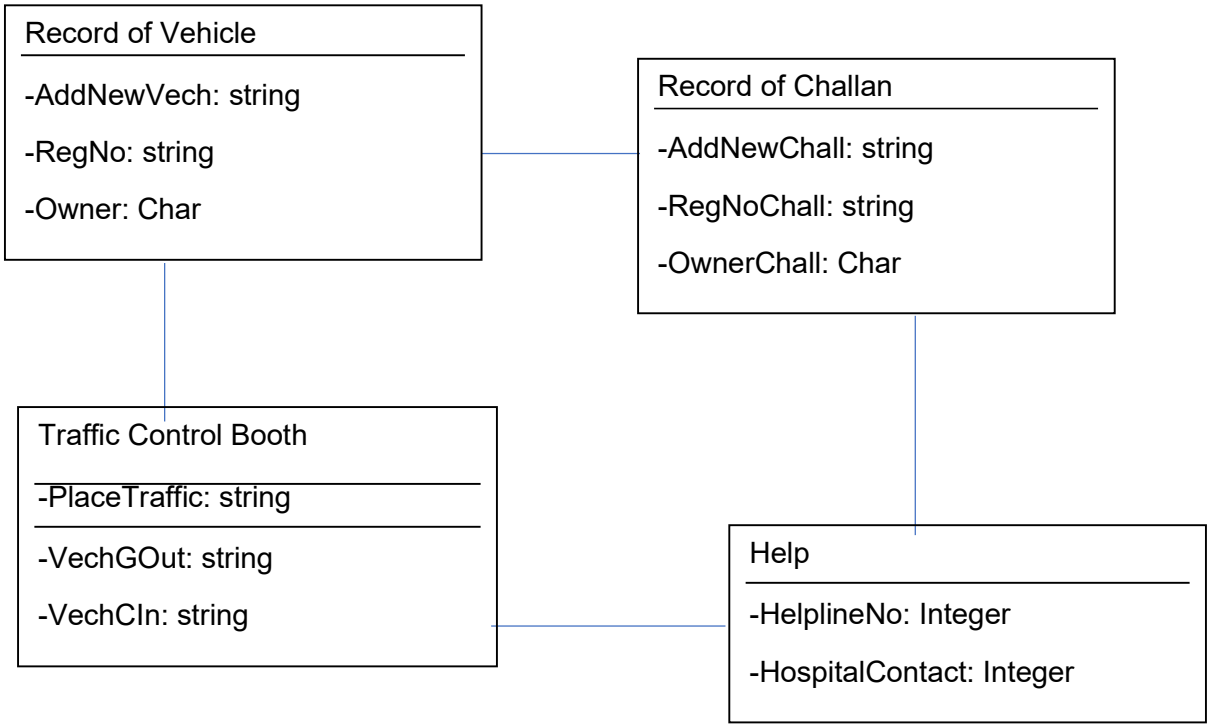
Development Process-



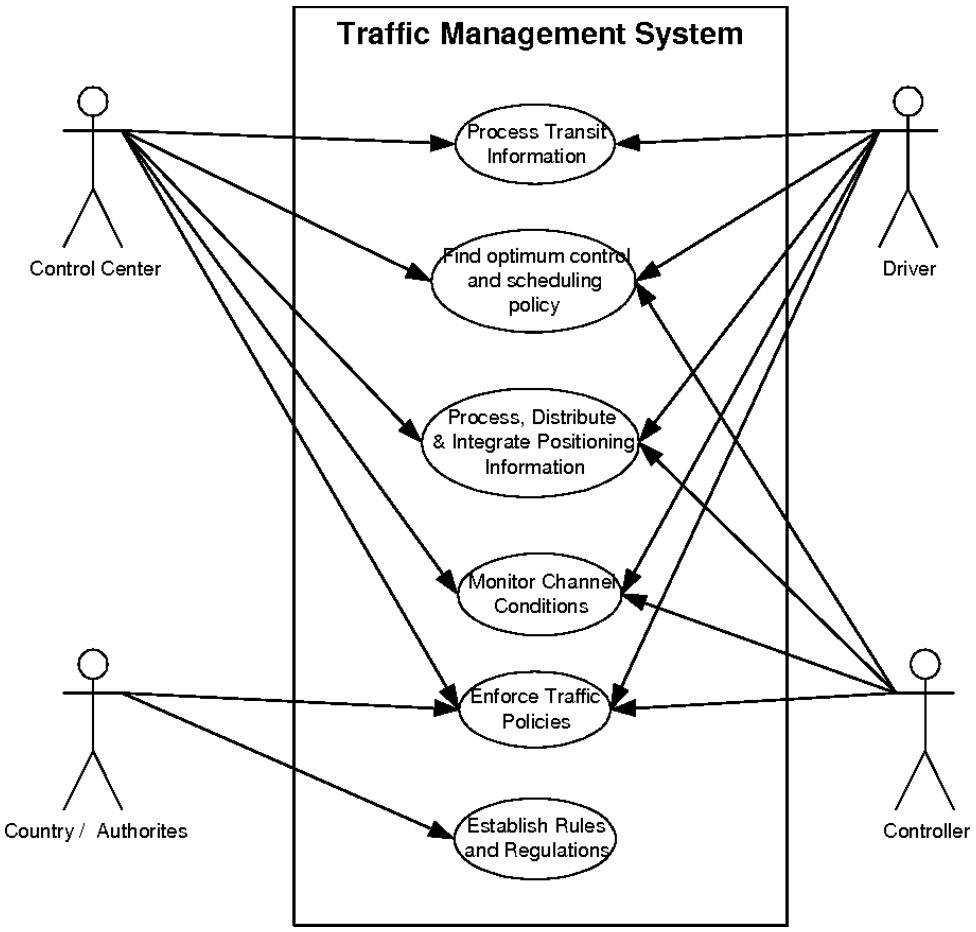
Process flow Diagram



Class Diagram



Use Case Diagram



TESTING

- Unit Test Summary and Results.

	Test	Use Case Derivation	Test Result
	A. REPORTS		
	1. Generate a Report on vehicle records	Generate a Report Use Case	Passed
	2. Generate a Report on Vehicle Challan	Generate a Report Use Case	Passed
	3. Generate a Report on Default Price Exceptions	Generate a Report Use Case	Passed
	4. Generate a Report on Pending Cash Outs	Generate Pending CheckOut Report Use Case	Passed
	B. USER ADMINISTRATION		

	1. Find Vehicle records 2. Add a New Vehicle record 3. Update a vehicle Info. 4. Clear information in Text Field 5. Delete a Vehicle record	Administer User Profile Use Case Administer User Profile Use Case Administer User Profile Use Case Administer User Profile Use Case Administer User Profile Use Case	Passed Passed Passed Passed Passed

	Test	Use Case Derivation	Test Result
C. Payment			
	1. Generate Bill 2. Retrieve Payment Information 3. Clear the Payment Screen	Make Payment Use Case Make Payment Use Case Make Payment Use Case	Passed Passed Passed Passed

- Integration Test Summary

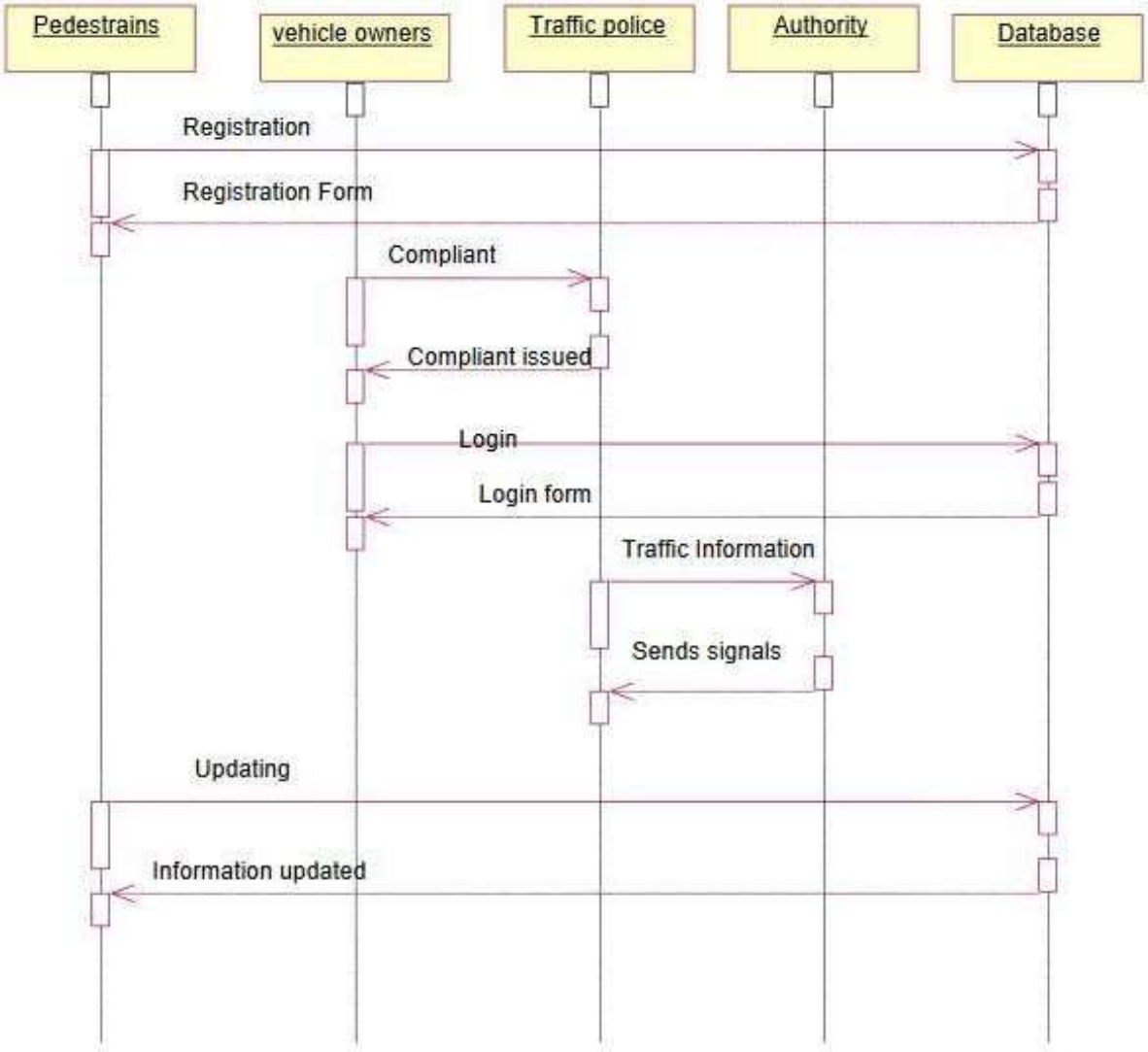
	TEST	USE CASE DERIVATION	TEST RESULT
1.	Record of Vehicles	Record of Vehicles Use Case	Passed
2.	Record of Challan	Record of Challan Use Case	Passed
3.	Search the Record of Vehicles	Search the Record of Vehicles Use Case	Passed
4.	Traffic Control Booths	Generate Report Use Case	Passed
5.	Control the Traffic	Control the Traffic Out Report Use Case	Passed
6.	Hospital Help	Hospital Help Use Case	Passed
7.	Manager Performs Room Availability	Room Use Case	Passed
8.	View a Particular Customer Record	Customer Record Use Case	Passed

COST ESTIMATION

Generating the module and placing the traffic: 3-4lacs (estimation)

Practise of police officers along with the project: time requirement with their increment

SEQUENCE DIAGRAM



CODE:

```
#include<bits/stdc++.h>
#include <fstream>
#include <chrono> //for delay
#include <stdlib.h>
#include<ctime>
```

```
// ***** Smart Traffic Management Solution *****

//keep the record of vehicles                    == recOfVeh()
//keep the record of challan done                == recOfChall()
//Search the record of vehicles                  == vehSearch()
//Display information of traffic in control booths == trafContBooth()
//Helpline Information and nearby hospitals      == helpInfo()
//control the traffic                            == trafCont()


using namespace std;


class SmartTrafficManagementSystem{


public:

   int welcome()
   {
      system("clear");

      time_t tt;
      struct tm * ti;
      time (&tt);
```

```
        ti = localtime(&tt);


        cout<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<endl<<"
" <<asctime(ti);

        delay1();


        system("clear");



cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;


cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;

        cout<<"**'
'**"<<endl;

        cout<<"**'
'**"<<endl;

        cout<<"**'                              WELCOME TO
'**"<<endl;

        cout<<"**'
'**"<<endl;

        cout<<"**'                         SMART TRAFFIC
MANAGEMENT SYSTEM                                   '**"<<endl;

        cout<<"**'
'**"<<endl;

        cout<<"**'
'**"<<endl;

        cout<<"**'
'**"<<endl;
```

```
        cout<<"**'
'**"<<endl;
        cout<<"**'              Press Your Option :-
'**"<<endl;
        cout<<"**'
'**"<<endl;
        cout<<"**'                      1.Record of Vehicles
'**"<<endl;
        cout<<"**'                      2.Record of Challan
'**"<<endl;
        cout<<"**'                      3.Search the Record of Vehicles
'**"<<endl;
        cout<<"**'                      4.Traffic Control Booths
'**"<<endl;
        cout<<"**'                      5.Control the Traffic
'**"<<endl;
        cout<<"**'                      6.Help !
'**"<<endl;
        cout<<"**'
'**"<<endl;
        cout<<"**'              Enter your choice ___
'**"<<endl;
        cout<<"**'
'**"<<endl;
        cout<<"**'
'**"<<endl;
        cout<<"**'
'**"<<endl;
        cout<<"**'
'**"<<endl;
        cout<<"**'
'**"<<endl;
```

```
        cout<<"**'
'**"<<endl;
        cout<<"**'
'**"<<endl;

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;



        int choice{0};
        cin>>choice;

        if(choice > 0 && choice < 7)
        {
            switch (choice)
            {
                case 1:
                    system("clear");
                    recOfVeh();
                    break;

                case 2:
                    system("clear");
                    recOfChall();
                    break;
```

```
        case 3:
            system("clear");
            vehSearch();
            break;

        case 4:
            system("clear");
            trafContBooth();
            break;

        case 5:
            system("clear");
            trafContBooth();
            break;

        case 6:
            system("clear");
            helpInfo();
            break;
    }
} else{
    cout<<"Enter Valid option !!"<<endl;
    delay();

    system("clear");
    welcome();
```

```
    }
    return 0;
}

int delay()
{
    using namespace std::this_thread; // sleep_for, sleep_until
    using namespace std::chrono; // nanoseconds, system_clock, seconds

    sleep_for(nanoseconds(1000000));
    sleep_until(system_clock::now() + seconds(1));
}
int delay1()
{
    using namespace std::this_thread; // sleep_for, sleep_until
    using namespace std::chrono; // nanoseconds, system_clock, seconds

    sleep_for(nanoseconds(1000000000));
    sleep_until(system_clock::now() + seconds(1));
}
int delay2()
{
    using namespace std::this_thread; // sleep_for, sleep_until
    using namespace std::chrono; // nanoseconds, system_clock, seconds

    sleep_for(nanoseconds(100000000));
    sleep_until(system_clock::now() + seconds(1));
```

```
}
int recOfVeh()
{

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*                                    WELCOME TO
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*                              SMART TRAFFIC
MANAGEMENT SYSTEM                              *"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*                                 * Record of Vehicles *
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*              Press Your Option :-
*"<<endl;
    cout<<"*
*"<<endl;
```

```
cout<<"*                              1. Add a New Vehicle
*"<<endl;
cout<<"*                              2. Search a Vehicle Using registration
number                                 *"<<endl;
cout<<"*                              3. Search a Vehicle Using Name of the
owner                                  *"<<endl;
cout<<"*
*"<<endl;
cout<<"*                    Enter 0 For Home
*"<<endl;
cout<<"*                    Enter your choice __
*"<<endl;
cout<<"*
*"<<endl;
cout<<"*
*"<<endl;
cout<<"*
*"<<endl;
cout<<"*
*"<<endl;

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;

int ROVChoice{0};
cin>>ROVChoice;
switch (ROVChoice)
{
  case 0:
    system("clear");
```

```
            welcome();
            break;

        case 1:
            recOfVeh_1();
            break;

        case 2:
            recOfVeh_2();
            break;

        case 3:
            recOfVeh_3();
            break;

    }
    return 0;
}
int recOfVeh_1()
{
    system("clear");
    fstream fio;
    string line;
    cout << "                Welcome to Registration of Vehicles\n"<<endl;
    cout<<"Enter Registration number of the Vehicle in the first line "<<endl;
    cout<<"Enter Name of the owner in the second line "<<endl;
    cout<<endl<<"Enter './' to Exit ";
```

```
    fio.open("/home/lamecodes/CLionProjects/untitled/cmake-build-
debug/RecordOfVehicles.txt", ios::app | ios::out | ios::in);

    // Execute a loop If file successfully Opened
    while (fio) {

        // Read a Line from standard input
        getline(cin, line);

        // Press -1 to exit
        if (line == "./")
            break;

        // Write line in file
        fio << line << endl;
    }
    cout<<"Data Entered successfully !!"<<endl;
    delay();

    system("clear");
    recOfVeh();

}

int recOfVeh_2()
{
    system("clear");
```

```
    string path = "/home/lamecodes/CLionProjects/untitled/cmake-build-debug/RecordOfVehicles.txt";
    ifstream file( path.c_str() );


    if( file.is_open() )
    {
       cout << "                Welcome to Registration of Vehicles\n" ;


       cout <<endl<< "Write the Registration number of the vehicle you want to searching for\n" ;
       string word ;
       cin >> word ;


       int countwords = 0 ;
       string candidate ;
       while( file >> candidate ) // for each candidate word read from the file
       {
          if( word == candidate ) ++countwords ;
       }
       if (countwords > 0){
          cout << "The registration number " << word << "' has been found in our records."<<endl ;


          int choice;
          cout<<endl<<"Enter 1 to go to home screen and enter 2 for again entering the registration number ";
          cin>>choice;
          (choice ==1) ? welcome(): recOfVeh_2();
```

```
        }
        else{
            cout<<"Registration number does not foud !!";
            int choice;
            cout<<endl<<"Enter 1 to go to home screen and enter 2 for again
entering the registration number ";
            cin>>choice;
             (choice ==1) ? welcome(): recOfVeh_2();
        }
    }

    else
    {
        cerr << "Error! 401!\n" ;
        delay();
        welcome();

    }


}

int recOfVeh_3()
{
    system("clear");
```

```
    string path = "/home/lamecodes/CLionProjects/untitled/cmake-build-debug/RecordOfVehicles.txt";
    ifstream file( path.c_str() );


    if( file.is_open() )
    {
       cout << "                Welcome to Registration of Vehicles\n" ;


       cout <<endl<< "Write the Name of the Owner of the vehicle you want to searching for\n" ;
       string word ;
       cin >> word ;


       int countwords = 0 ;
       string candidate ;
       while( file >> candidate ) // for each candidate word read from the file
       {
          if( word == candidate ) ++countwords ;
       }
       if (countwords > 0){
          cout << "The Name of the Owner " << word << "' has been found in our records."<<endl ;


          int choice;
          cout<<endl<<"Enter 1 to go to home screen and enter 2 for again entering the Name of the Owner";
          cin>>choice;
          if (choice ==1)
             welcome();
```

```
            else
                recOfVeh_2();


        }
        else{
            cout<<"Name of the Owner does not foud !!";
            int choice;
            cout<<endl<<"Enter 1 to go to home screen and enter 2 for again entering the Name of the Owner ";
            cin>>choice;
            if (choice ==1)
                welcome();
            else
                recOfVeh_2();
        }
    }


    else
    {
        cerr << "Error! 401!\n" ;
        delay();
        welcome();


    }


}


int recOfChall()
```

```
{

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;

    cout<<"*
*"<<endl;

    cout<<"*                                             WELCOME TO
*"<<endl;

    cout<<"*
*"<<endl;

    cout<<"*                                  SMART TRAFFIC
MANAGEMENT SYSTEM                                *"<<endl;

    cout<<"*
*"<<endl;

    cout<<"*
*"<<endl;

    cout<<"*
*"<<endl;

    cout<<"*
*"<<endl;

    cout<<"*                                       * Record of Challan *
*"<<endl;

    cout<<"*
*"<<endl;

    cout<<"*                    Press Your Option :-
*"<<endl;

    cout<<"*
*"<<endl;

    cout<<"*                              1. Add a New Challan
*"<<endl;

    cout<<"*                              2. Search a Challan Using registration
number                               *"<<endl;
```

```
        cout<<"*                                3. Search a Challan Using Name of the
owner                                            *"<<endl;

        cout<<"*
*"<<endl;

        cout<<"*                        Enter 0 For Home
*"<<endl;

        cout<<"*                        Enter your choice __
*"<<endl;

        cout<<"*
*"<<endl;

        cout<<"*
*"<<endl;

        cout<<"*
*"<<endl;

        cout<<"*
*"<<endl;

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;


        int ROCChoice{0};
        cin>>ROCChoice;
        switch (ROCChoice)
        {
          case 0:
            system("clear");
            welcome();
            break;


          case 1:
```

```
            recOfChall_1();

            break;

        case 2:
            recOfChall_2();
            break;

        case 3:
            recOfChall_3();
            break;
        default:
            cout << "please enter a valid case" << endl;

    }
    return 0;
}

int recOfChall_1()
{

    system("clear");
    fstream fio;
    string line;

    cout << "              Welcome to Challan Management System\n"<<endl;
```

```
        cout<<"Enter Registration number of the Vehicle in the first line "<<endl;
        cout<<"Enter Name of the owner in the second line "<<endl;
        cout<<endl<<"Enter './' to Exit ";
        fio.open("/home/lamecodes/CLionProjects/untitled/cmake-build-
debug/RecordOfChallan.txt", ios::app | ios::out | ios::in);

        // Execute a loop If file successfully Opened
        while (fio) {

            // Read a Line from standard input
            getline(cin, line);

            // Press -1 to exit
            if (line == "./")
                break;

            // Write line in file
            fio << line << endl;
        }
        cout<<"Data Entered successfully !!"<<endl;
        delay();

        system("clear");
        recOfChall();


    }
```

```
int recOfChall_2()
{

    system("clear");
    string path = "/home/lamecodes/CLionProjects/untitled/cmake-build-debug/RecordOfChallan.txt";
    ifstream file( path.c_str() );

    if( file.is_open() )
    {
        cout << "            Welcome to Challan Management System\n"<<endl;

        cout <<endl<< "Write the registration number of the vehicle you want to searching for\n" ;
        string word ;
        cin >> word ;

        int countwords = 0 ;
        string candidate ;
        while( file >> candidate ) // for each candidate word read from the file
        {
            if( word == candidate ) ++countwords ;
        }
        if (countwords > 0){
            cout << "The Registration number '" << word << "' has been found in our records."<<endl ;

            int choice;
```

```
            cout<<endl<<"Enter 1 to go to home screen and enter 2 for again
entering the registration number.";
            cin>>choice;
            if (choice ==1)
               welcome();
            else
               recOfChall_2();


         }
         else{
            cout<<"Registration number does not foud !!";
            int choice;
            cout<<endl<<"Enter 1 to go to home screen and enter 2 for again
entering the registration number.";
            cin>>choice;
            if (choice ==1)
               welcome();
            else
               recOfChall_2();
         }
      }


      else
      {
         cerr << "Error! 402!\n" ;
         delay();
         welcome();
```

```
    }
}

int recOfChall_3()
{
    system("clear");
    string path = "/home/lamecodes/CLionProjects/untitled/cmake-build-debug/RecordOfChallan.txt";
    ifstream file( path.c_str() );

    if( file.is_open() )
    {
        cout << "              Welcome to Challan Management System\n"<<endl;

        cout <<endl<< "Write the Name of the Owner of the vehicle you want to searching for\n" ;
        string word ;
        cin >> word ;

        int countwords = 0 ;
        string candidate ;
        while( file >> candidate ) // for each candidate word read from the file
        {
            if( word == candidate ) ++countwords ;
        }
        if (countwords > 0){
            cout << "The Name of the Owner " << word << "' has been found in our records."<<endl ;
```

```
            int choice;
            cout<<endl<<"Enter 1 to go to home screen and enter 2 for again
entering the Name of the Owner";
            cin>>choice;
            if (choice ==1)
               welcome();
            else
               recOfChall_3();


         }
         else{
            cout<<"Name of the Owner does not foud !!";
            int choice;
            cout<<endl<<"Enter 1 to go to home screen and enter 2 for again
entering the Name of the Owner ";
            cin>>choice;
            if (choice ==1)
               welcome();
            else
               recOfChall_3();
         }
      }


      else
      {
         cerr << "Error! 402!\n" ;
         delay();
```

```
welcome();


    }
  }


  int vehSearch()
  {



cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*                                      WELCOME TO
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*                                SMART TRAFFIC
MANAGEMENT SYSTEM                                        *"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*                                * Search the Record of Vehicles *
*"<<endl;
```

```
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*                                        Enter The Vehicles Registration
Number                                          *"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*                    Enter 0 For Home
*"<<endl;
        cout<<"*                    Enter your choice ___
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;



        string path = "/home/lamecodes/CLionProjects/untitled/cmake-build-
debug/RecordOfVehicles.txt";
```

```
ifstream file( path.c_str() );
system("clear");

if( file.is_open() )
{

   string word ;
   cin >> word ;

   int countwords = 0 ;
   string candidate ;
   while( file >> candidate ) // for each candidate word read from the file
   {
      if( word == candidate ) ++countwords ;
   }
   if (countwords > 0){
      cout << "The registration number " << word << "' has been found in
our records."<<endl ;

      int choice;
      cout<<endl<<"Enter 1 to go to home screen and enter 2 for again
entering the registration number ";
      cin>>choice;
      if (choice ==1)
         welcome();
      else
         vehSearch();
   }
```

```
        else{
            cout<<"Registration number does not foud !!";
            int choice;
            cout<<endl<<"Enter 1 to go to home screen and enter 2 for again
entering the registration number ";
            cin>>choice;
            if (choice ==1)
               welcome();
            else
               vehSearch();
         }
      }
      else
      {
         cerr << "Error! 401!\n" ;
         delay();
         welcome();


      }









   }

   int trafContBooth()
   {
```

```
// Traffic Control Booths

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*                        WELCOME TO
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*                   SMART TRAFFIC
MANAGEMENT SYSTEM                    *"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*                   * Traffic Control Booths *
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*          Press Your Option :-
*"<<endl;
    cout<<"*
*"<<endl;
    cout<<"*               1. Jalandhar Traffic Control Booth
*"<<endl;
```

```
	cout<<"*				2. Amritsar Traffic Control Booth
*"<<endl;
	cout<<"*				3. Chandigarh Traffic Control Booth
*"<<endl;
	cout<<"*
*"<<endl;
	cout<<"*			Enter 0 For Home
*"<<endl;
	cout<<"*			Enter your choice __
*"<<endl;
	cout<<"*
*"<<endl;
	cout<<"*
*"<<endl;
	cout<<"*
*"<<endl;
	cout<<"*
*"<<endl;

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;


	int TCBChoice{0};
	cin>>TCBChoice;
	switch (TCBChoice)
	{
	   case 0:
	      system("clear");
	      welcome();
	      break;
```

```
        case 1:
            trafContBooth_1();
            break;

        case 2:
            trafContBooth_2();
            break;

        case 3:
            trafContBooth_3();
            break;

    }


    return 0;
}
int trafContBooth_1()
{
    system("clear");

    for (int i = 0; i < 100; ++i) {

        cout<<"                Jalandhar City Traffic Control Booth
"<<endl;
        cout<<"Vehicles Going out of the City                Vehicles coming into
the City"<<endl;
```

```
        cout<<endl<<"      "<<i+1<<"
"<<i+5<<endl;
        delay();


        system("clear");


      }


   }


   int trafContBooth_2()
   {
      system("clear");


      for (int i = 0; i < 100; ++i) {


        cout<<"                  Amritsar City Traffic Control Booth
"<<endl;
        cout<<"Vehicles Going out of the City                Vehicles coming into
the City"<<endl;
        cout<<endl<<"      "<<i+5<<"                                    "<<i*7<<endl;
        delay();


        system("clear");


      }
```

```
}

int trafContBooth_3()
{

    system("clear");

    for (int i = 0; i < 100; ++i) {

        cout<<"                    Chandigarh City Traffic Control Booth
"<<endl;
        cout<<"Vehicles Going out of the City                 Vehicles coming into
the City"<<endl;
        cout<<endl<<"      "<<i*16<<"
"<<i*22<<endl;
        delay();

        system("clear");

    }

}

int helpInfo()
{
    //Helpline Information and nearby hospitals

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*                                        WELCOME TO
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*                                 SMART TRAFFIC
MANAGEMENT SYSTEM                                    *"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*                             * Helpline Information And Nearby
Hospitals *                          *"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*              Press Your Option :-
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*                          1. Helpline Number
*"<<endl;
        cout<<"*                          2. Hospitals in Amritsar
*"<<endl;
        cout<<"*                          3. Hospitals in Chandigarh
*"<<endl;
```

```
        cout<<"*
*"<<endl;
        cout<<"*                    Enter 0 For Home
*"<<endl;
        cout<<"*                    Enter your choice __
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;
        cout<<"*
*"<<endl;

cout<<">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>>>>>>>>"<<endl;

        int CTTChoice{0};
        cin>>CTTChoice;
        switch (CTTChoice) {
            case 0:
                system("clear");
                welcome();
                break;

            case 1: {
                system("clear");
                string line;
```

```
        ifstream myfile("/home/lamecodes/CLionProjects/untitled/cmake-build-debug/HelpNumbers.txt");
        if (myfile.is_open()) {
            while (getline(myfile, line)) {
                cout << line << '\n';
            }
            myfile.close();
        } else cout << "Error! 403!";

        int choice;
        cout << endl << "Enter 1 to go Home Page" << endl;
        cin >> choice;
        if (choice == 1) {
            system("clear");
            welcome();
        } else {
            cout << endl << "Enter Valid option !!";
            cout << endl << endl << "Enter 1 to go Home Page" << endl;
            cin >> choice;
            if (choice == 1) {
                system("clear");
                welcome();
            }
        }
        break;
    }
    case 2: {
```

```
            system("clear");
            string line;
            ifstream myfile("/home/lamecodes/CLionProjects/untitled/cmake-build-debug/HAmritsar.txt");
            if (myfile.is_open()) {
               while (getline(myfile, line)) {
                  cout << line << '\n';
               }
               myfile.close();
            } else cout << "Error! 403!";

            int choice;
            cout << endl << "Enter 1 to go Home Page" << endl;
            cin >> choice;
            if (choice == 1) {
               system("clear");
               welcome();
            } else {
               cout << endl << "Enter Valid option !!";
               cout << endl << endl << "Enter 1 to go Home Page" << endl;
               cin >> choice;
               if (choice == 1) {
                  system("clear");
                  welcome();
               }
            }
         }
            break;
```

```
        case 3: {


            system("clear");
            string line;
            ifstream myfile("/home/lamecodes/CLionProjects/untitled/cmake-build-debug/HChandigarh.txt");
            if (myfile.is_open()) {
                while (getline(myfile, line)) {
                    cout << line << '\n';
                }
                myfile.close();
            } else cout << "Error! 403!";


            int choice;
            cout << endl << "Enter 1 to go Home Page" << endl;
            cin >> choice;
            if (choice == 1) {
                system("clear");
                welcome();
            } else {
                cout << endl << "Enter Valid option !!";
                cout << endl << endl << "Enter 1 to go Home Page" << endl;
                cin >> choice;
                if (choice == 1) {
                    system("clear");
                    welcome();
                }
```

```
            }
            break;
        }
    }
    return 0;
  }
};
int main()
{
  SmartTrafficManagementSystem ob1;
  ob1.welcome();
}
```

GRANTT CHART