REAL-TIME TRAFFIC MONITORING SYSTEM

Project Submitted To:

SRM University – AP, Andhra Pradesh

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

Bachelor of Technology

in

Computer Science and Engineering

School of Engineering and Sciences

<u>Submitted By:</u>

Group No.: 11

Under the guidance of

Ms. V. Veda Sri

Faculty, Computer Science Department

Department of Computer Science and Engineering

SRM University,AP

Neeru Konda, Mangala Giri, Guntur

Andhra Pradesh – 522240

[November, 2024]

Acknowledgement

Without acknowledging those who made it possible and whose unwavering support and encouragement make any endeavor successful, the happiness that comes with doing any assignment successfully would be lacking.

I want to thank my project from the bottom of my heart and convey how much I appreciate it. Ms. V. Veda Sri, adviser, SRM's Department of Computer Science & Engineering University, Andhra Pradesh, for her kind assistance, for providing me with the required direction, and helpful recommendations for finishing work.

Abhishekh Kumar Jha (AP23110011668)

Bhupendra Yadav (AP23110011660)

Y. Umesh Reddy (AP23110011667)

Rakesh Patel (AP23110011671)

B. Akhil (AP23110011662)

Abstract

A software program called the Traffic Management System with C++ was created with the intention of automating and controlling traffic management processes. Challan recording, vehicle and challan searches, traffic flow monitoring across many control booths, and the providing of emergency contact and medical information are just a few of the many features it offers. C++ will be used for data management and file processing. By effectively automating chores, our traffic system will show how programming can be used in practical situations like traffic control.

TABLE OF CONTENTS

1. _INTRODUCTION:

The process of managing traffic is the focus of the Traffic Management System. We will discover how to efficiently store car registration numbers and provide quick and simple access to and recovery of data. Numerous tasks are available to the user, including looking for a car and its owner, obtaining data records, and more.

Data gathering, processing, storing, and visualization are some of the elements that go into building a real-time traffic monitoring system. An organized method for creating such a system is shown below.

Image-1: Traffic Monitoring in National Highway

2. OBJECTIVES:

The objectives of this real-time traffic management system with C++ programming language are as follows:

- **Traffic Violation Management:** An interface for managing and recording traffic infractions (challans) associated with owner names and vehicle registration numbers is provided by the code.
- **Traffic Challan Search**: The application provides the ability to look for traffic tickets using the owner's name or the vehicle's registration number.
- **Vehicle Search:** The system enables search operations for vehicle records based on their registration numbers, aiding in tracking and monitoring.
- **Traffic Control Booth Monitoring:** The code tracks how many cars enter and exit the city by simulating the operation of traffic control booths at various places.
- **Provision of Emergency Information:** The system serves as a helpful resource in times of disaster by providing crucial emergency contact information and specifics about local healthcare facilities.

3. REQUIREMENTS:

- **Integrated Development Environment (IDE)**: A C++ compatible IDE like Dev C++, Code Blocks, or VS code will be needed for writing and debugging the code.
- **File System Access:** The code uses file handling operations to store and retrieve information about traffic violations. The development and deployment environment should allow for file creation, reading, and writing.
- **Operating System Compatibility:** The code is platform-independent, but your IDE and C++ compiler must be compatible with your operating system, be it Windows, MacOS, or Linux.

4. WORKFLOW FOR TRAFFIC MONITORING SYSTEM:

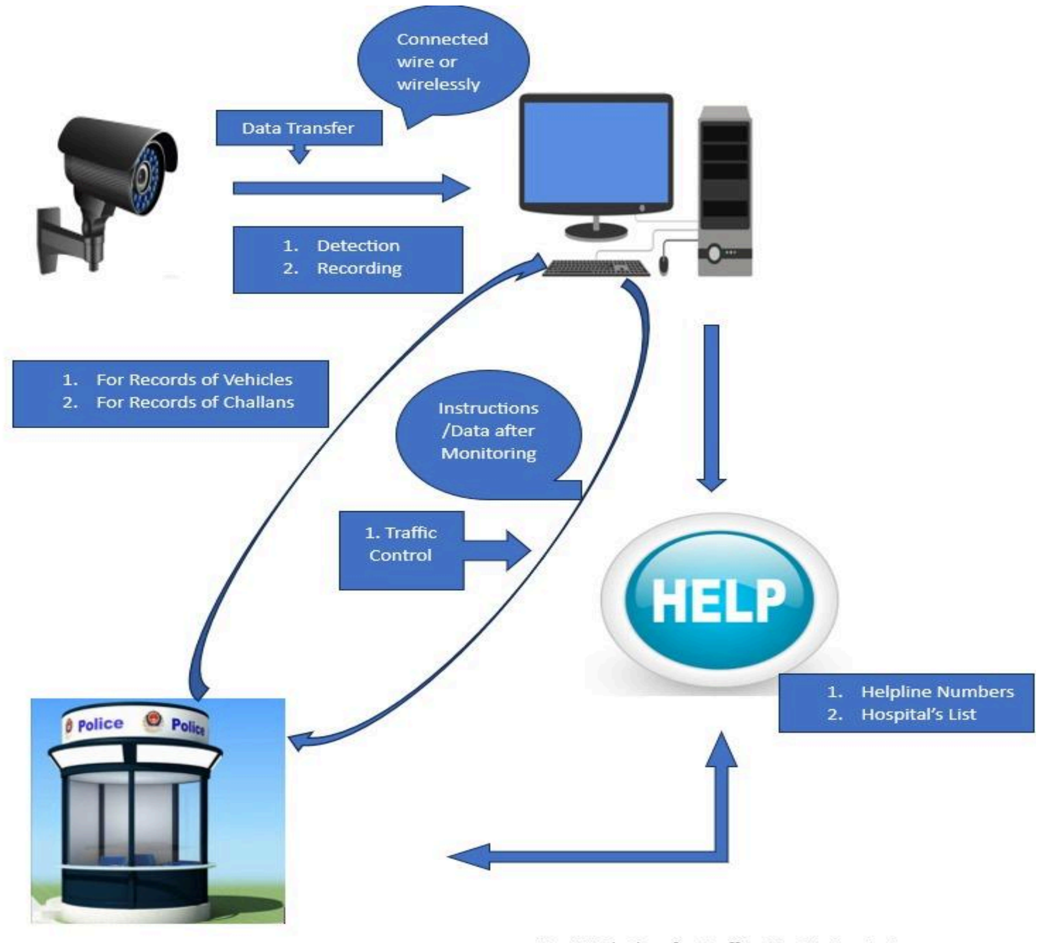

Connected wire or wirelessly
Data Transfer
1. Detection
2. Recording
1. For Records of Vehicles
2. For Records of Challans
Instructions /Data after Monitoring
1. Traffic Control
HELP
1. Helpline Numbers
2. Hospital's List
Police
Police

Fig.: Work-Flow for Traffic- Monitoring System

1. **Data Gathering:**

- Sensor Deployment: Set up a system of sensors at key points, such as LIDAR, radar, cameras, inductive loops, etc. Take into account elements such as infrastructure constraints, accident hotspots, and traffic density.
- Information Gathering: Real-time data is captured by sensors, including:
- Number and type of vehicles (cars, trucks, motorbikes)
- Direction and speed of the vehicle Patterns of traffic movement Levels of congestion weather (if applicable) Data is sent to a cloud platform or central server.
- Analysis and Processing of Data:
- Preprocessing and Data Cleaning: Remove mistakes and noise from sensor data. Standardize the units and formats of data. Address outliers or missing data.

2. Extraction of Features:

- From raw data, extract pertinent elements such Vehicle density Average velocity volume of traffic Patterns of traffic movement.
- Data Visualization and Analysis: Utilize data mining methods (such as regression, classification, and clustering) to find tendencies and patterns.
- Use maps and dashboards to visualize data and keep an eye on traffic conditions in real time.
- Create reports for planning the future and analyzing the past.

3. Traffic Management:

- Real-time Monitoring: Keep an eye on traffic conditions at all times and spot any irregularities (such as accidents or congestion). Set off alarms for urgent circumstances.
- Traffic Signal Control: To maximize flow, modify traffic signal timings in response to real-time traffic data. Put in place systems for adaptive traffic control.
- Incident Management: Quickly identify and address incidents (such as collisions or road closures). Work along with traffic management authorities and emergency services.

4. Optimization and Decision Support:

- Predictive analytics: Use machine learning algorithms and past data to forecast future traffic conditions. Determine possible areas of congestion and bottlenecks.
- Optimization Algorithms: Create optimization algorithms to identify the best options for traffic flow. Take into account elements such as traffic volume, timing of signals and the capacity of the road network.
- Scenario Planning: To assess the effects of various measures, simulate various traffic situations.

- Determine areas that could want improvement and create backup plans.

5. **Communication and User Interface:**

- User-friendly Interface: Create an interface that is easy for the general public and traffic managers to use.
- Send out notifications and traffic information in real time. Facilitate feedback and contact with the public.
- Channels of Communication: Use a variety of communication platforms to spread information, such as websites, social media, and mobile apps.
- Organize with emergency services and transportation providers.

6. EXPLANATION OF THE CODE:

There are features to keep the owner's identity, record car data, look up a vehicle using its registration number, and more. Below is a summary of the roles and their respective functions: Every feature operates as intended and without any problems.

Class Definitions

1. **TrafficCamera Class**:
   - Attributes:
     - location: The location of the traffic camera.
     - vehiclesDetected: The number of vehicles detected by the camera.
     - averageSpeed: The average speed of vehicles detected.
   - **Methods:**
     - monitorTraffic(): Simulates monitoring traffic by generating random values for vehicles detected and average speed, and prints the information.
     - detectAccident(): Randomly determines if an accident has occurred (20% chance).
     - alertAuthorities(): Alerts authorities if an accident is detected.
     - getLocation(): Returns the location of the camera.
2. **TrafficMonitoringSystem Class:**
   - Attributes:
     - cameras: A vector to store multiple TrafficCamera objects.
   - Methods:
     - addCamera(): Adds a new camera to the monitoring system.
     - startMonitoring(): Continuously monitors each camera in a loop, checking for traffic and accidents, and alerts authorities if necessary.
3. **SmartTrafficManagementSystem Class:**
   - This class serves as the main interface for the user to interact with the system.

- Methods:
    - welcome(): Displays a welcome message and prompts the user for options.
    - Various methods for displaying vehicle records, challan records, searching vehicles, controlling traffic, and providing help information.
    - controlTraffic(): Initializes a TrafficMonitoringSystem and starts monitoring traffic.
    - displayHospitals(): Displays nearby hospitals based on the selected city.

Main Function

- The ‘main()’ function creates an instance of ‘SmartTrafficManagementSystem’ and calls the ‘welcome()’ method to start the program.

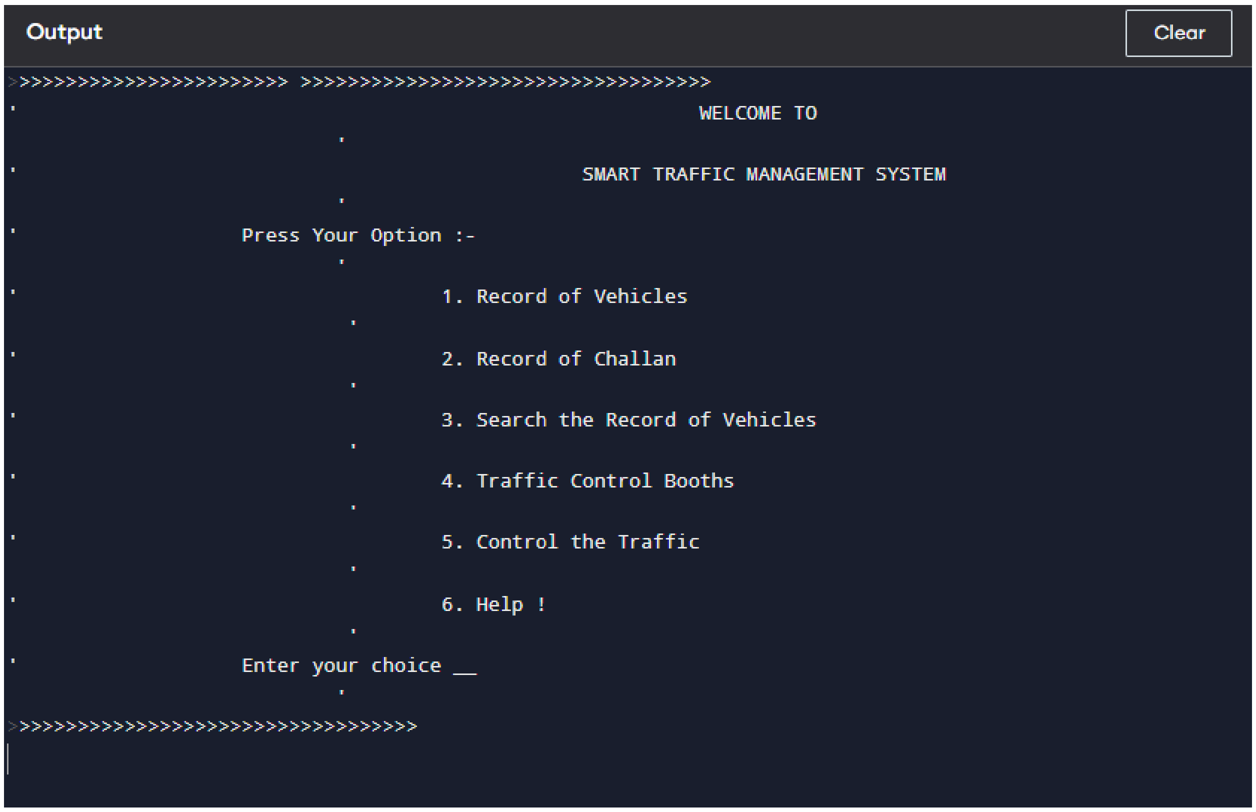
Program Flow

1. **User Interaction:**
    - The user is presented with a menu to select various options such as viewing vehicle records, challan records, searching for vehicles, controlling traffic, and accessing help information.
2. **Traffic Monitoring:**
    - When the user selects the traffic control option, the system initializes traffic cameras at specified locations and begins monitoring traffic. Each camera simulates vehicle detection and accident detection.
3. **Data Display:**
    - The system can display various records, including vehicle details and traffic violation records (challans), based on user selections. It also provides information about traffic control booths and hospitals in different cities.

Additional Features

- The program uses random number generation to simulate vehicle detection and accident occurrences, making it dynamic and interactive.
- The use of this_thread::sleep_for() introduces delays to simulate real-time monitoring and user experience.
- The program includes error handling for invalid user input, ensuring robustness.

7. OUTPUT:


Output
Clear

```
>>>>>>>>>>>>>>>>>>>>>>>> >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
'                                                  WELCOME TO
                             '
'                                        SMART TRAFFIC MANAGEMENT SYSTEM
                             '
'                    Press Your Option :-
                             '
'                                    1. Record of Vehicles
                              '
'                                    2. Record of Challan
                              '
'                                    3. Search the Record of Vehicles
                              '
'                                    4. Traffic Control Booths
                              '
'                                    5. Control the Traffic
                              '
'                                    6. Help !
                              '
'                    Enter your choice __
                             '
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

8. CONCLUSION:

This C++ Traffic Management System is an excellent illustration of how to utilize C++ for data management and file handling to create a useful, real-world application. The user may access a variety of car records as needed with the aid of this user-friendly, automated, and straightforward C++ system. It demonstrates how programming may be used to automate processes in a variety of domains, such as traffic control.

9. REFERENCES:

- **Artificial Intelligence-Enabled Traffic Monitoring System**
- **(PDF) Real-time Traffic Monitoring System based on Deep Learning and YOLOv8**

<u>GROUP-11</u>

ABHISHEKH KUMAR JHA (AP23110011668)

BHUPENDRA YADAV (AP23110011660)

Y. UMESH REDDY (AP23110011667)

AKHIL (AP23110011662)

RAKESH PATEL (AP23110011671)