

# **DENSITY BASED SMART TRAFFIC CONTROL SYSTEM**

**USING CANNY EDGE DETECTION ALGORITHM FOR  
CONGREGATION TRAFFIC INFORMATION**

*Project Report Submitted by*

**VISMAYA. KK Reg. No: LAJC18MCA062**

*In Partial fulfillment for the award of the degree*

*Of*

**MASTER OF COMPUTER APPLICATIONS (MCA) APJ  
ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING  
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala - 686518]

**2018-2021**

# **AMAL JYOTHI COLLEGE OF ENGINEERING**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala - 686518]

## **DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**



### **CERTIFICATE**

This is to certify that the project entitled "**Density Based Smart Traffic Control System Using Canny Edge Detection Algorithm for Congregating Traffic information**" is a bonafide record of the work done by **VISMAYA.KK LAJC18MCA062**, during the academic year **2018-2021** carried out under our supervision. It is certified that all corrections/suggestions indicated for assessment have been incorporated in the report. The work report has been approved as it satisfies the academic requirements in respect of the project work prescribed by the university for the Master of Computer Applications Degree. Certified further, that to the best of our knowledge the exact work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this to any other candidate.

**Fr. Rubin Thottupuram**

Head of the Department

**Sr. Elsin Chakkalackal**

Project Coordinator

**Ms. Lisha Varghese**

Project Supervisor

**Expert from dept. of Computer Science and Engineering**  
Amal Jyothi College of Engineering

**External Expert appointed by the university**

## **DECLARATION**

I hereby declare that the project report "**Density Based Smart Traffic Control System Using Canny Edge Detection Algorithm for Congregating Traffic Information**" is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2018-2021.

**Date.....**

**Vismaya KK**

**KANJIRAPPALLY**

**Reg. No: LAJC18MCA062**

## **ACKNOWLEDGEMENT**

First and foremost, I thank Almighty God for his gracious guidance through the project. I take this opportunity to express my gratitude to all those who have helped me in completing the project successfully.

It has been said that gratitude is the memory of the heart. I acknowledge my deep sense of gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** for providing all the infrastructural facilities for us, our Principal **Dr. Z V Lakaparampil** for providing good faculty for guidance.

I take the immense pleasure in expressing my thanks to Head of the Department of Master of Computer Applications, **Fr. Rubin Thottupuram**, for his kind patronages in making this project a successful one. I would like to extend my sincere thanks to our coordinator **Sr. Elsin Chakkalackal** and my project guide **Ms. Lisha Varghese** for her guidance and cooperation, without which this would not have been a success.

I am indebted to my beloved teachers whose cooperation and suggestions throughout the project which helped me a lot. I also thank all my friends and classmates for their interest, dedication and encouragement shown towards the project. I convey hearty thanks to parents for the moral support, suggestion and encouragement to make this venture a success.

**Vismaya KK**

## **ABSTRACT**

As the problem of urban traffic congestion intensifies, there is a pressing need for the introduction of advanced technology and equipment to improve the state-of-the-art of traffic control. The current methods used such as timers or human control are proved to be inferior to alleviate this crisis. In this paper, a system to control the traffic by measuring the real-time vehicle density using canny edge detection with digital image processing is proposed. This imposing traffic control system offers significant improvement in response time, vehicle management, automation, reliability and overall efficiency over the existing systems. Besides that, the complete technique from image acquisition to edge detection and finally green signal allotment using four sample images of different traffic conditions is illustrated with proper schematics and the final results are verified by hardware implementation.

# CONTENT

<b>Sl. No</b>	<b>Topic</b>	<b>Page No</b>
	<b>Introduction</b>	<b>1</b>
<b>1</b>	<b>Using Git as a Version Control System</b>	<b>2</b>
1.1	Introduction to GitHub	3
1.2	Working With Git	4-8
<b>2</b>	<b>Project Documentation</b>	<b>9</b>
2.1	Introduction	10
2.1.1	Project Overview	10
2.1.2	Project Specification	10-11
2.2	System Study	12
2.2.1	Introduction	12
2.2.2	Existing system	13
2.2.3	Proposed system	13-14
2.3	Requirement Analysis	15
2.3.1	Feasibility Study	15
2.3.1.1	Economical Feasibility	15
2.3.1.2	Technical Feasibility	16
2.3.1.3	Behavioral Feasibility	16
2.4	Requirement Modeling	17
2.4.1	UML Diagrams	17-24
2.5	System Specification	25
2.5.1	Hardware Specification	25
2.5.2	Software Specification	25
2.6	Software Description	25
2.6.1	PYTHON	25
2.6.2	SQLyog	25-26
2.6.3	Android Studio	27
2.7	System Design	28
2.7.1	Architectural Design	28
2.7.2	Module Design	29-30

<b>2.7.3</b>	<b>Data Base Design</b>	<b>30-38</b>
<b>2.8</b>	<b>System Testing</b>	<b>39</b>

## **CONTENT**

<b>2.8.1</b>	<b>Introduction</b>	<b>39</b>
<b>2.8.2</b>	<b>Test Plan</b>	<b>40</b>
<b>2.8.2.1</b>	<b>Unit Testing</b>	<b>40-41</b>
<b>2.8.2.2</b>	<b>Integration Testing</b>	<b>41</b>
<b>2.8.2.3</b>	<b>Validation Testing</b>	<b>41</b>
<b>2.8.2.4</b>	<b>User Acceptance Testing</b>	<b>41-42</b>
<b>2.9</b>	<b>Implementation</b>	<b>42</b>
<b>2.9.1</b>	<b>Implementation Procedure</b>	<b>42-43</b>
<b>2.9.2</b>	<b>User Training</b>	<b>43</b>
<b>2.9.3</b>	<b>Operational Document</b>	<b>43</b>
<b>2.9.4</b>	<b>System Maintenance</b>	<b>44</b>
<b>2.10</b>	<b>Conclusion &amp; Future Enhancements</b>	<b>44</b>
<b>2.10.1</b>	<b>Future Enhancement</b>	<b>44</b>
<b>2.10.2</b>	<b>Conclusion</b>	<b>44</b>
<b>2.11</b>	<b>Bibliography</b>	<b>45</b>
<b>2.12</b>	<b>Appendix</b>	<b>46</b>
<b>2.12.1</b>	<b>Sample Code</b>	<b>46-52</b>
<b>2.12.2</b>	<b>Screen Shots</b>	<b>53-61</b>

## **LIST OF ABBREVIATIONS**

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheet

SQL - Structured Query Language

UML - Unified Modeling Language

## **INTRODUCTION**

Traffic congestion is one of the major modern-day crisis in every big city in the world. Inter metropolitan area studies suggest that traffic congestion reduces regional competitiveness and redistributes economic activity by slowing growth in county gross output or slowing metropolitan area employment growth. As more and more vehicles are commissioning in an already congested traffic system, there is an urgent need for a whole new traffic control system using advanced technologies to utilize the already existent infrastructures to its full extent. Since building new roads, flyovers, elevated expressway etc. needs extensive planning, huge capital and lots of time; focus should be directed upon availing existing infrastructures more efficiently and diligently. Edge detection technique is imperative to extract the required traffic information from the camera. It can be used to isolate the required information from rest of the image. Density of traffic is measured by comparing captured image with real time traffic information against the image of the empty road as reference image is proposed.

# **1. USING GIT AS A VERSION CONTROL SYSTEM**

## **1.1 Introduction to GitHub**

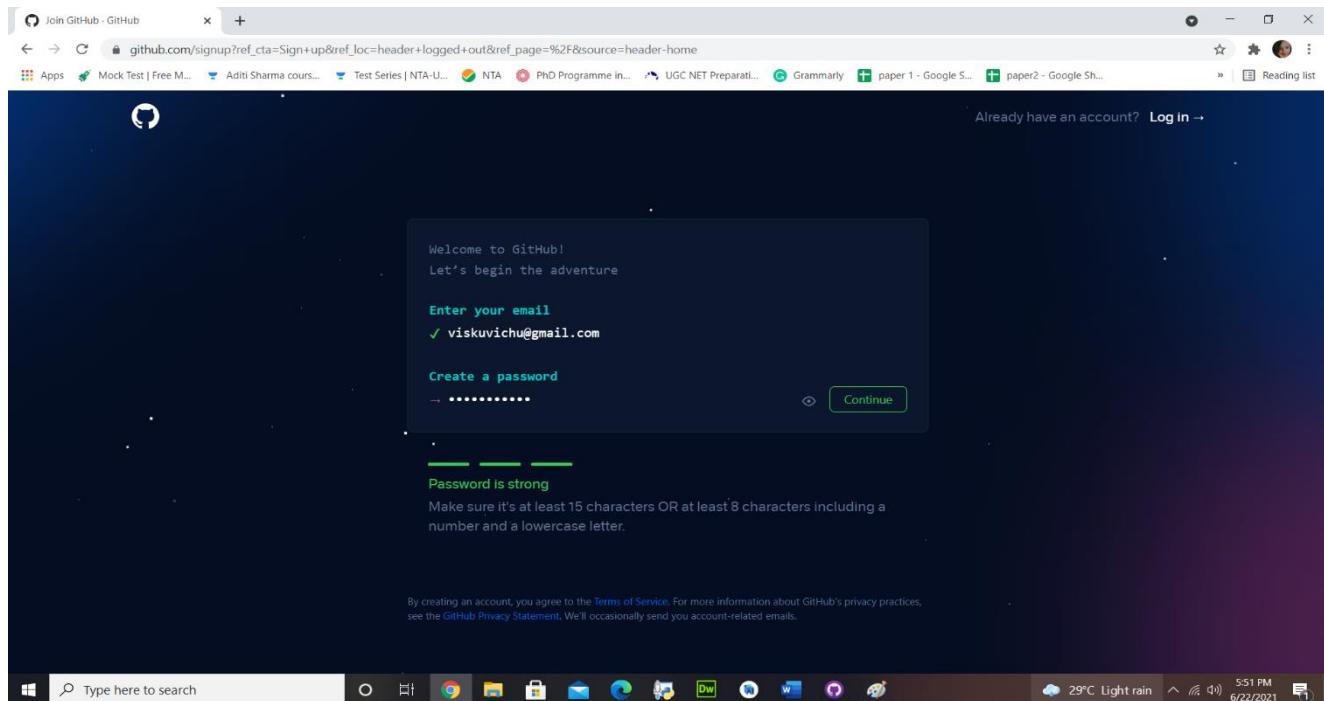
GitHub is a web-based version-control and collaboration platform for software developers. GitHub, which is delivered through a software-as-a-service (SaaS) business model, was started in 2008 and was founded on Git, an open source code management system created by Linus Torvalds to make software builds faster. And it is used to store the source code for a project and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers. GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans. Each public or private repository contains all of a project's files, as well as each file's revision history. Repositories can have multiple collaborators and can be either public or private. GitHub facilitates social coding by providing a web interface to the Git code repository and management tools for collaboration. GitHub can be thought of as a serious social networking site for software developers. Members can follow each other, rate each other's work, receive updates for specific projects and communicate publicly or privately.

### **GitHub products and features**

GitHub offers an on-premises version in addition to the well-known SaaS product. GitHub Enterprise supports integrated development environments and continuous integration tool integration, as well as a litany of third-party apps and services. It offers increased security and auditability than the SaaS version.

## 1.2 Working with Git

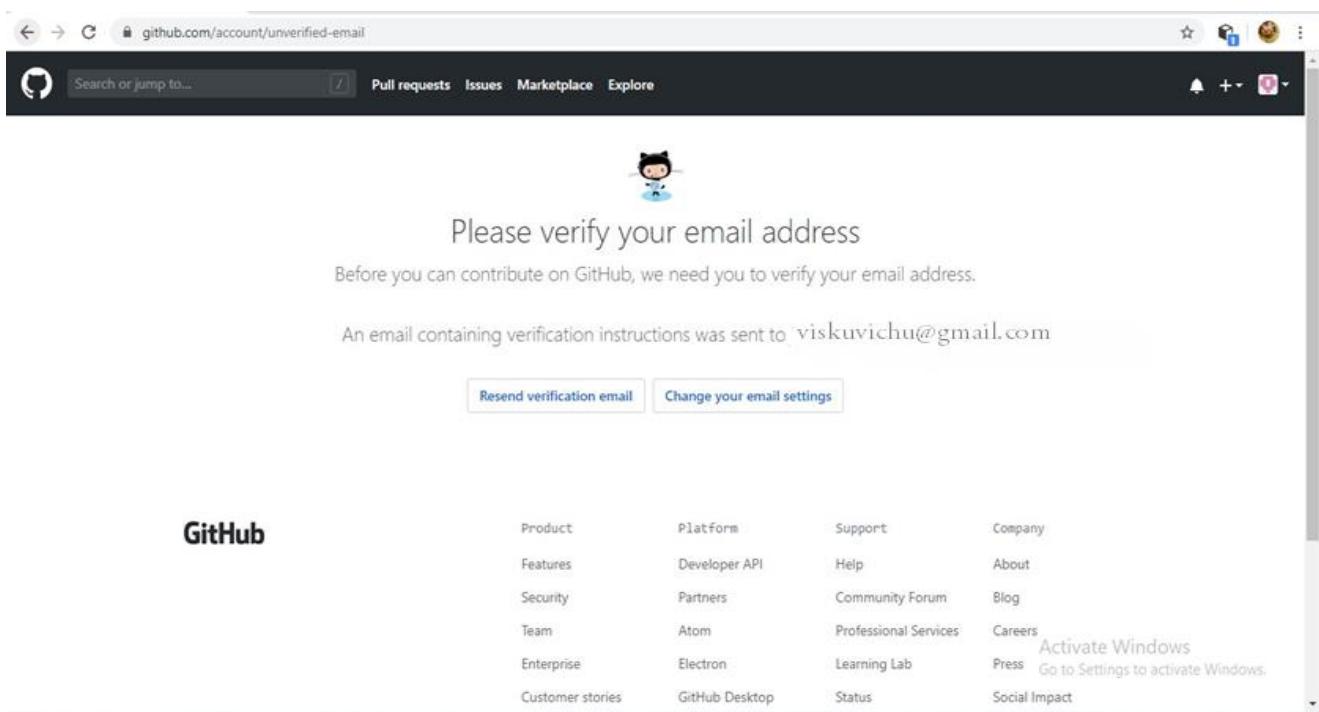
- Create an account in GitHub.com



- Once successfully sign in, set up personal account, and choose your plan by selecting the options given below based on our purpose.

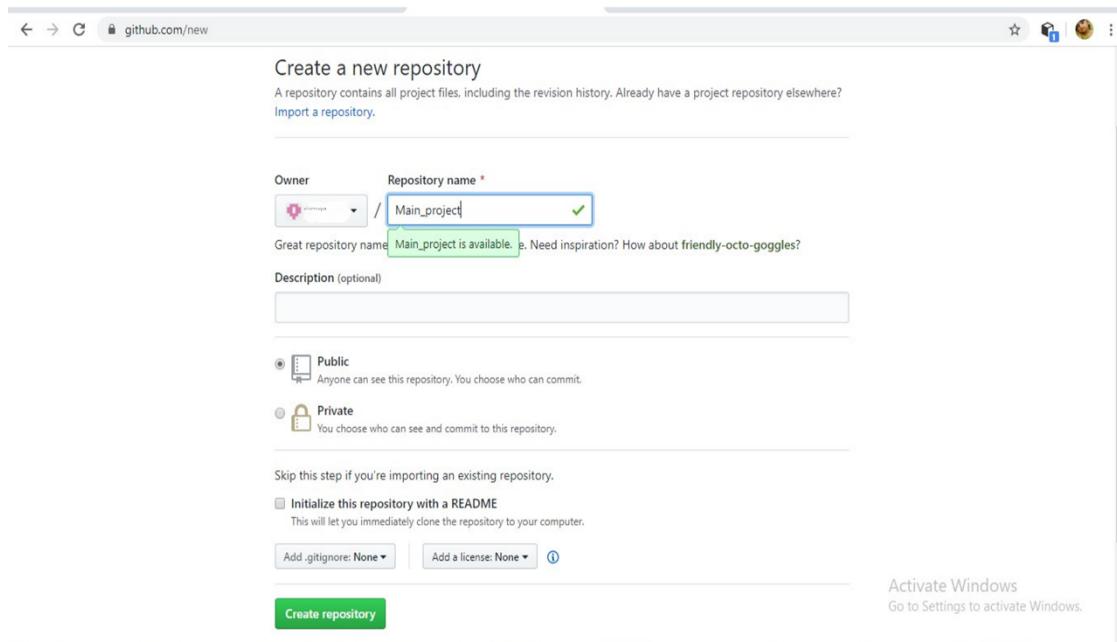
The screenshot shows a browser window with the GitHub 'join/customize' page. The URL in the address bar is [github.com/join/customize](https://github.com/join/customize). The page header includes links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header, it says 'Selected plan: Free'. The main content area features a large 'Welcome to GitHub' heading and a message encouraging users to tell GitHub about their interests. A section titled 'What kind of work do you do, mainly?' lists several options: 'Software Engineer' (I write code), 'Student' (I go to school), 'Product Manager' (I write specs), 'UX & Design' (I draw interfaces), 'Data & Analytics' (I write queries), and 'Marketing & Sales' (I look at charts). In the bottom right corner, there's a 'Activate Windows' message with a link to Settings.

- Once this step is completed then verify the email address and you can access the home page



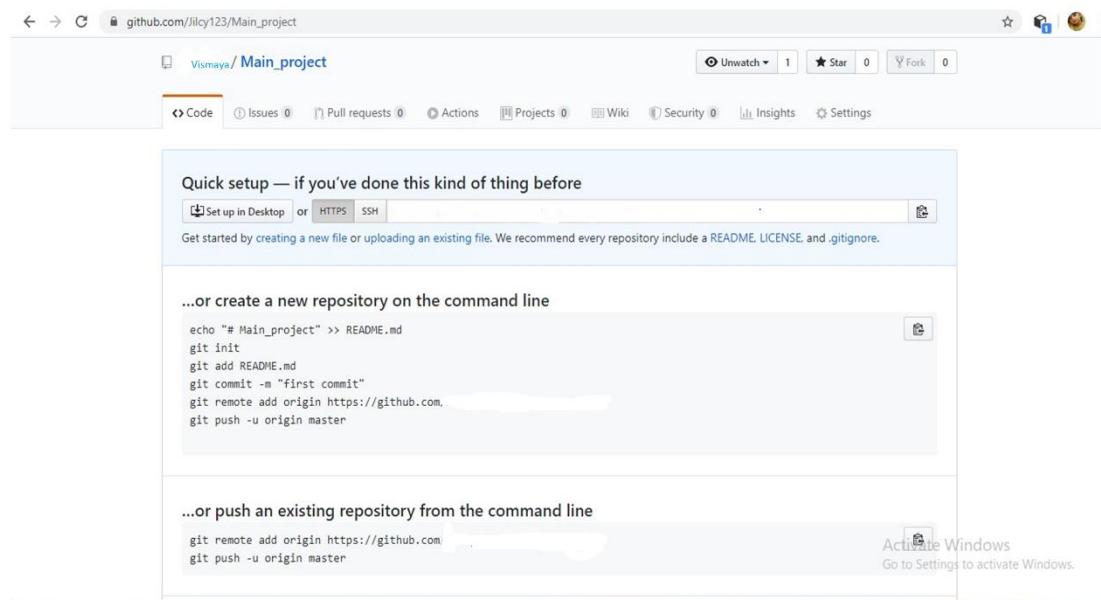
The screenshot shows a browser window with the URL [github.com/account/unverified-email](https://github.com/account/unverified-email). At the top, there's a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below that is a GitHub logo and a user icon. The main content area has a heading "Please verify your email address" with the sub-instruction "Before you can contribute on GitHub, we need you to verify your email address." It states that an email was sent to [viskuvichu@gmail.com](mailto:viskuvichu@gmail.com). There are two buttons at the bottom: "Resend verification email" and "Change your email settings".

- Create a repository in GitHub

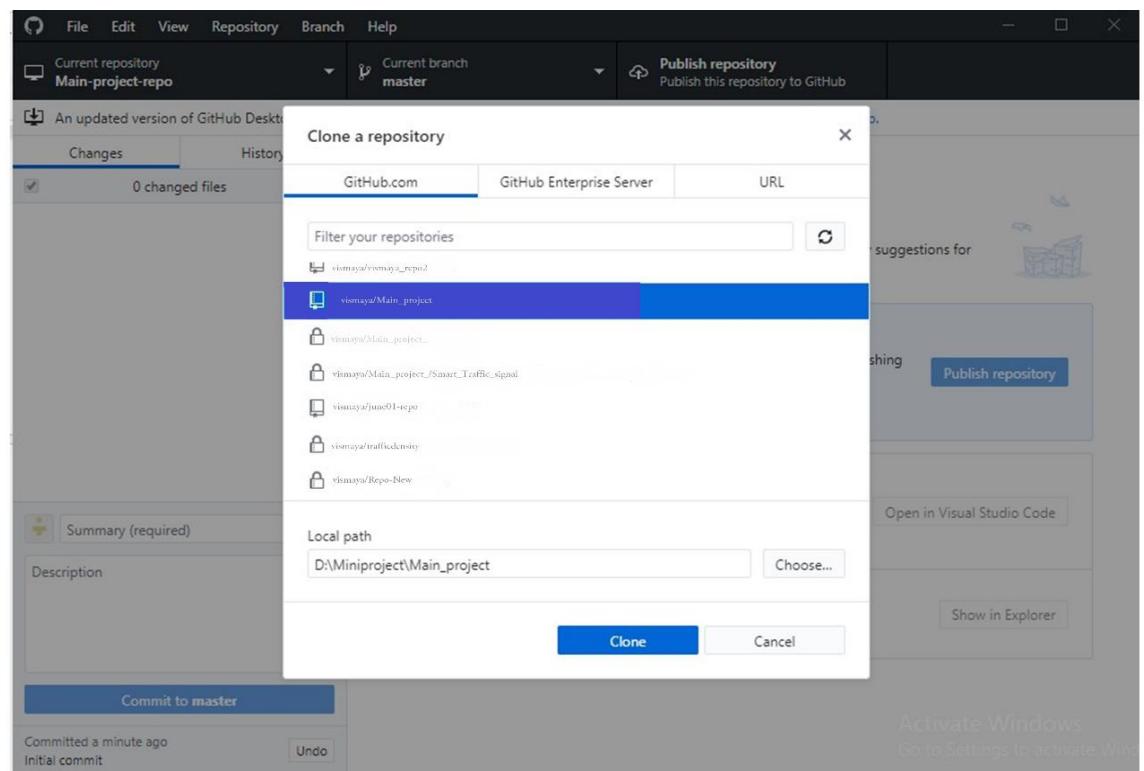


The screenshot shows a browser window with the URL [github.com/new](https://github.com/new). The title is "Create a new repository". It says "A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository." The form asks for the "Owner" (set to "viskuvichu") and "Repository name" (set to "Main\_project"). A note says "Great repository name! Main\_project is available." Below is a "Description (optional)" field with a placeholder "Add a description...". Under "Visibility", "Public" is selected with the note "Anyone can see this repository. You choose who can commit.". Under "Private", there's a note "You choose who can see and commit to this repository.". Below these are "Skip this step if you're importing an existing repository." and "Initialize this repository with a README". At the bottom are buttons for "Add .gitignore: None", "Add a license: None", and a large green "Create repository" button.

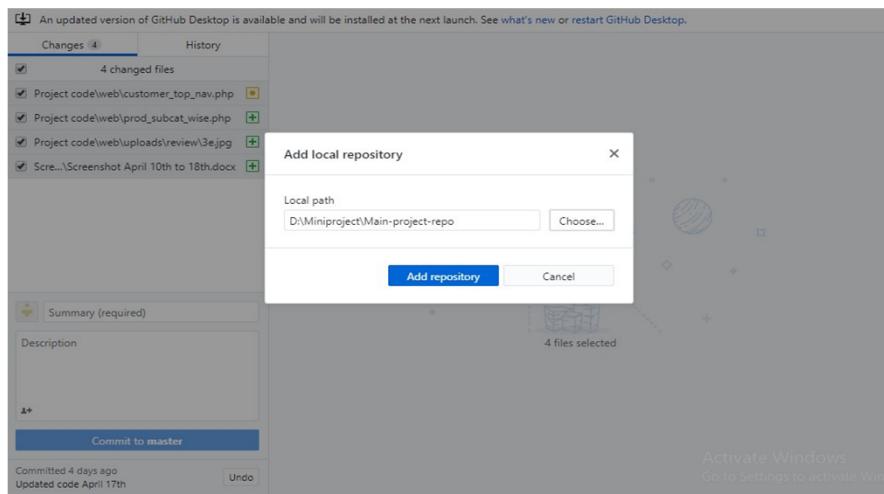
- Once repository is completed, you can setup the repository



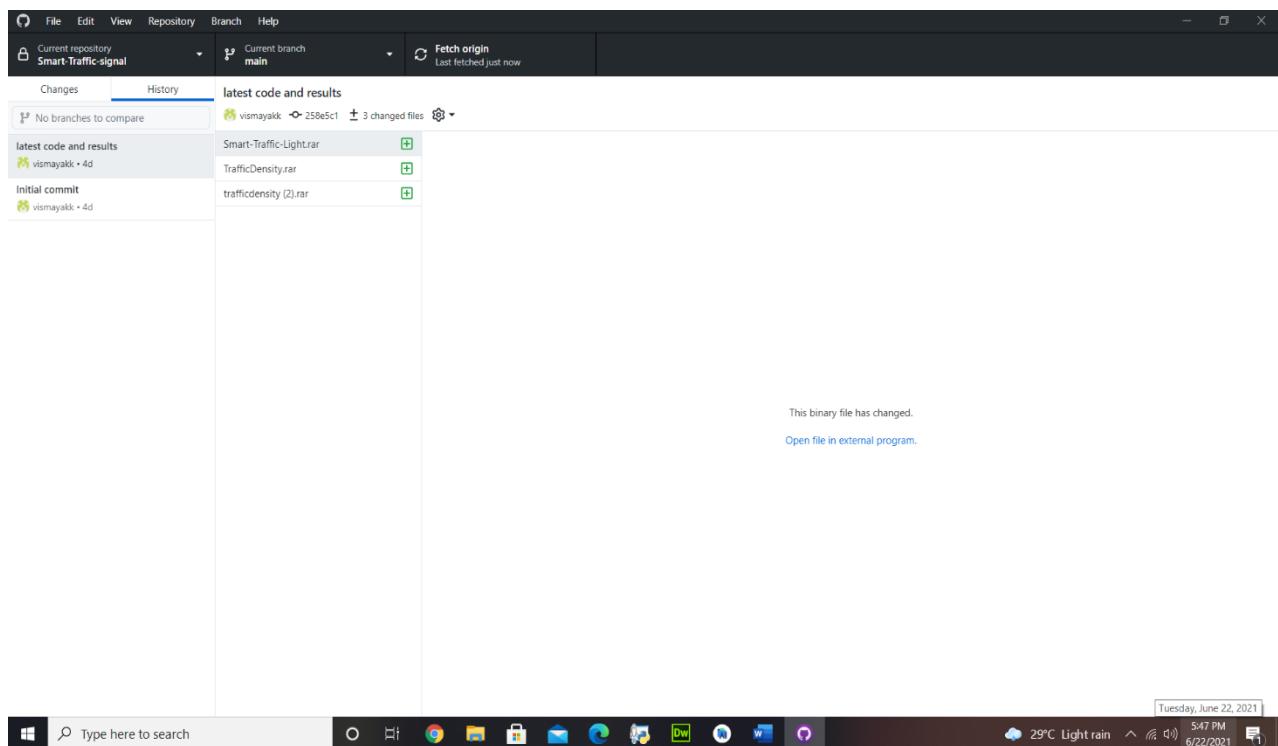
- Download and install GitHub desktop application. Once installed Go to Repository > Clone repository, and select your repository which was created in GitHub.com or Select a local system folder.



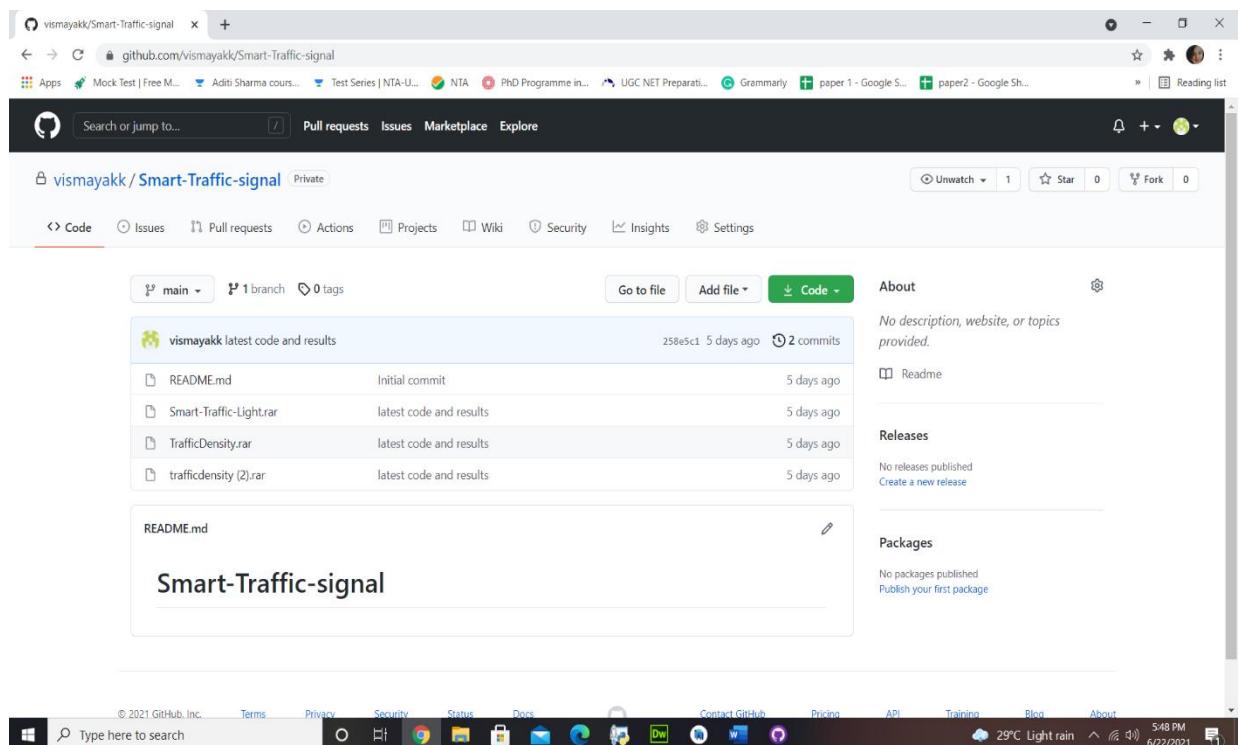
- Add local system folder, then create a local repository with folder name.



- Once repository is created, make changes on the file which is stored inside the local repository and commit to master.



- Push the local repository to origin
- Refresh GitHub.com repository to fetch commits



## **2. PROJECT DOCUMENTATION**

## 2.1 INTRODUCTION

### 2.1.1 Project Overview

Traffic congestion is one of the major modern-day crisis in every big city in the world. Recent study of World Bank has shown that average vehicle speed has been reduced from 21 km to 7 km per hour in the last 10 years in Dhaka .Intermetropolitan area studies suggest that traffic congestion reduces regional competitiveness and redistributes economic activity by slowing growth in county gross output or slowing metropolitan area employment growth .As more and more vehicles are commissioning in an already congested traffic system, there is an urgent need for a whole new traffic control system using advanced technologies to utilize the already existent infrastructures to its full extent. Since building new roads, flyovers, elevated expressway etc. needs extensive planning, huge capital and lots of time; focus should be directed upon availing existing infrastructures more efficiently and diligently.

### 2.1.2 Project Specification

imposing traffic control system offers significant improvement in response time, vehicle management, automation, reliability and overall efficiency over the existing systems. Besides that, the complete technique from image acquisition to edge detection and finally green signal allotment using four sample images of different traffic conditions is illustrated with proper schematics and the final results are verified by hardware implementation.

The system includes 4 modules. They are:

#### 1. Admin Module

The site admin has an overall control on the website. He can perform the functionalities like,

- ▶ 1. Traffic Police Registration
- ▶ 2. Traffic Island Registration
- ▶ 3. View Traffic density in real-time
- ▶ 4. Traffic signal registration
- ▶ 5. Sent notification

## **2. Traffic Police Module**

Traffic police can perform functionalities like,

- ▶ 1. Login/View/Change Password
- ▶ 2. View profile
- ▶ 3. View Traffic Density

## **3. User Module**

A User can perform the functionalities like,

- ▶ 1. User Signup/View/Change password
- ▶ 2. View profile
- ▶ 3. View Near Signal
- ▶ 4. View Traffic density

## 2.2 SYSTEM STUDY

### 2.2.1 Introduction

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

## 2.2.2 EXISTING SYSTEM

The current methods used such as timers or human control are proved to be inferior to alleviate this crisis. In this paper, a system to control the traffic by measuring the real-time vehicle density using canny edge detection with digital image processing is proposed. This imposing traffic control system offers significant improvement in response time, vehicle management, automation, reliability and overall efficiency over the existing systems.

### Drawbacks of existing system

- Not Efficient
- Human effort is needed.
- Time consuming

## 2.2.3 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly. On such consideration the system is proposed. In our proposed system there is an admin who can view and manage all the registered traffic police and users. It allows users to view traffic density. Users of this proposed system are admin, traffic police and users.

The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. In this project, a system to control the traffic by measuring the real-time vehicle density using canny edge detection with digital image processing is proposed. This imposing traffic control system offers significant improvement in response time, vehicle management, automation, reliability and overall efficiency over the existing systems. Besides that, the complete technique from image acquisition to edge detection and finally green signal allotment using four sample images of different traffic conditions is illustrated with proper schematics and the final results are verified by hardware implementation.

**Advantages of proposed system**

The system requires very low system resources and the system will work in almost all configurations. It has got following features:

A smart traffic control system availing image processing as an instrument for measuring the density has been proposed. Image processing has shown promising outcomes in acquiring real time traffic information using CCTV footage installed along the traffic light. calculating the number of vehicles may give false results if the intravehicular spacing is very small (two vehicles close to each other may be counted as one) and it may not count rickshaw or auto-rickshaw as vehicles which are the quotidian means of traffic especially in South-Asian countries. And counting number of pixels has disadvantage of counting insubstantial materials as vehicles such as footpath or pedestrians. Some of the works have proposed to allocate time based solely on the density of traffic. But this may be disadvantageous for those who are in lanes that have less frequency of traffic. Edge detection technique is imperative to extract the required traffic information from the CCTV footage. It can be used to isolate the required information from rest of the image. There are several edge detection techniques available. They have distinct characteristics in terms of noise reduction, detection sensitivity, accuracy etc.

## 2.3 REQUIREMENT ANALYSIS

### 2.3.1 Feasibility study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

#### 2.3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

The cost of project, Density Based Smart Traffic Control System Using Canny Edge Detection Algorithm for Congregating Traffic information was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open-source software.

### 2.3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. The system has been developed using HTML, CSS in front end and SQLyog in server in back end, the project is technically feasible for development. The system has been developed using HTML, CSS, Python, Android Studio and SQLyog in server back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i5 core; RAM 4GB and, Hard disk 1TB

### 2.3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

Density Based Smart Traffic Control System Using Canny Edge Detection Algorithm for Congregating Traffic information, GUI is user friendly so that users can easily use it without any training.

## 2.4 Requirement Modeling

### 2.4.1 UML diagram

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Activity diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

## 1. UML Activity Diagram

Activity Diagram describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not.

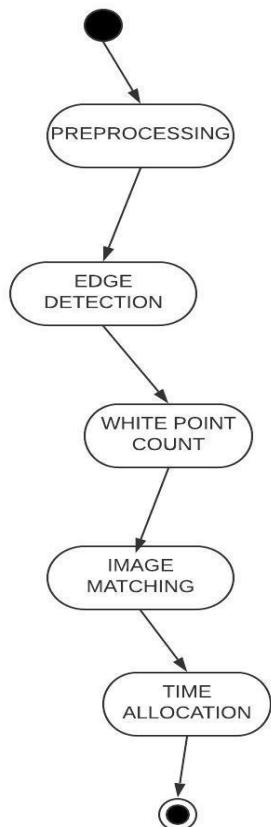


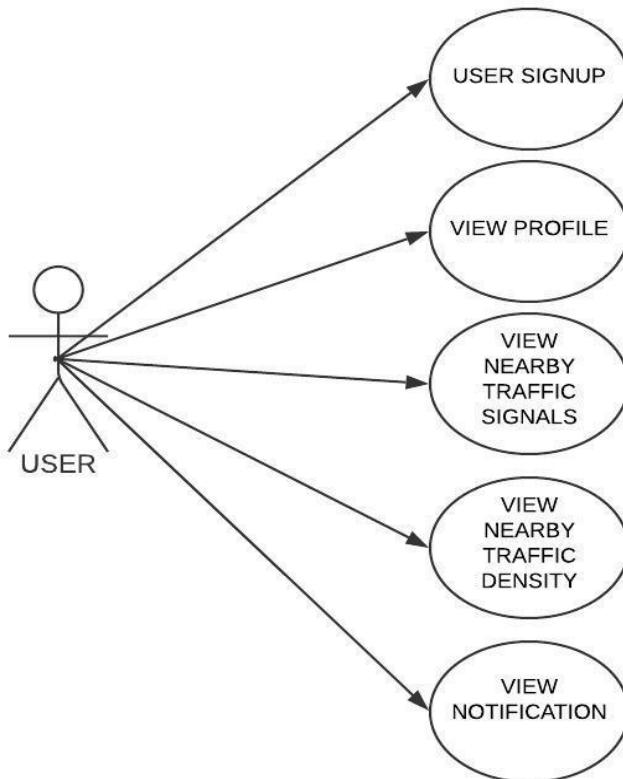
Fig. Activity diagram

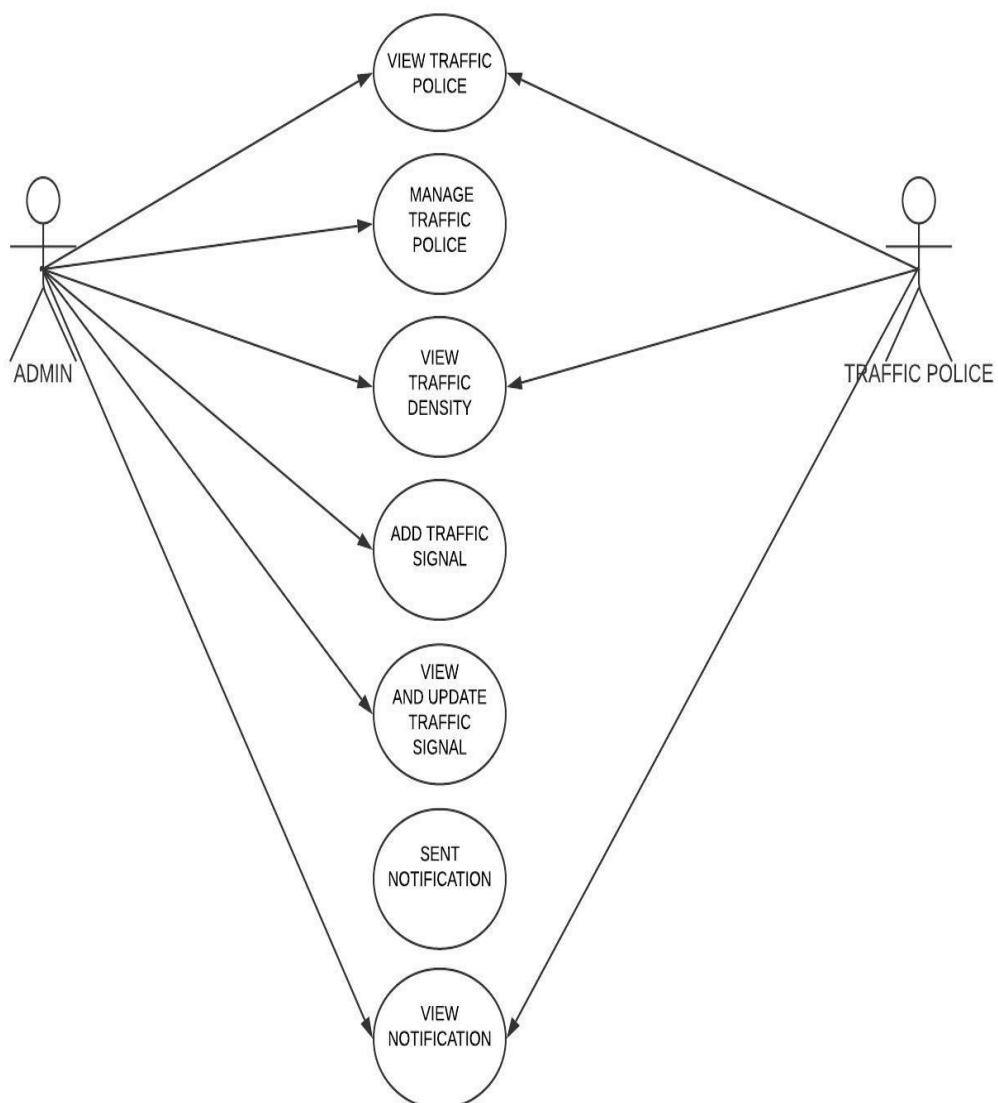
## 2. UML Use Case Diagram

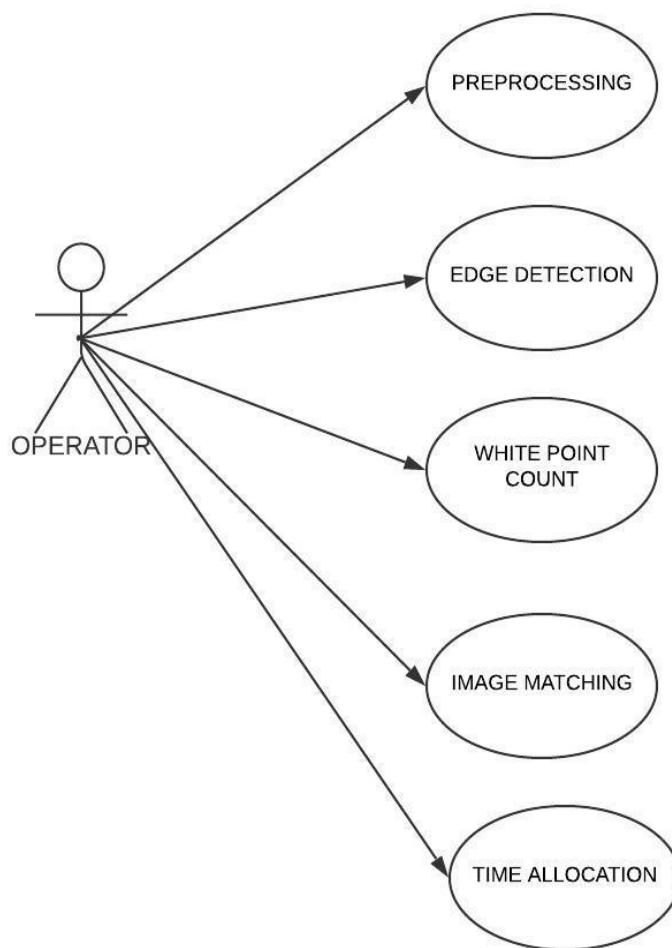
A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of realworld objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which the specific roles are played by the actors within and around the system.







### 3. UML Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner

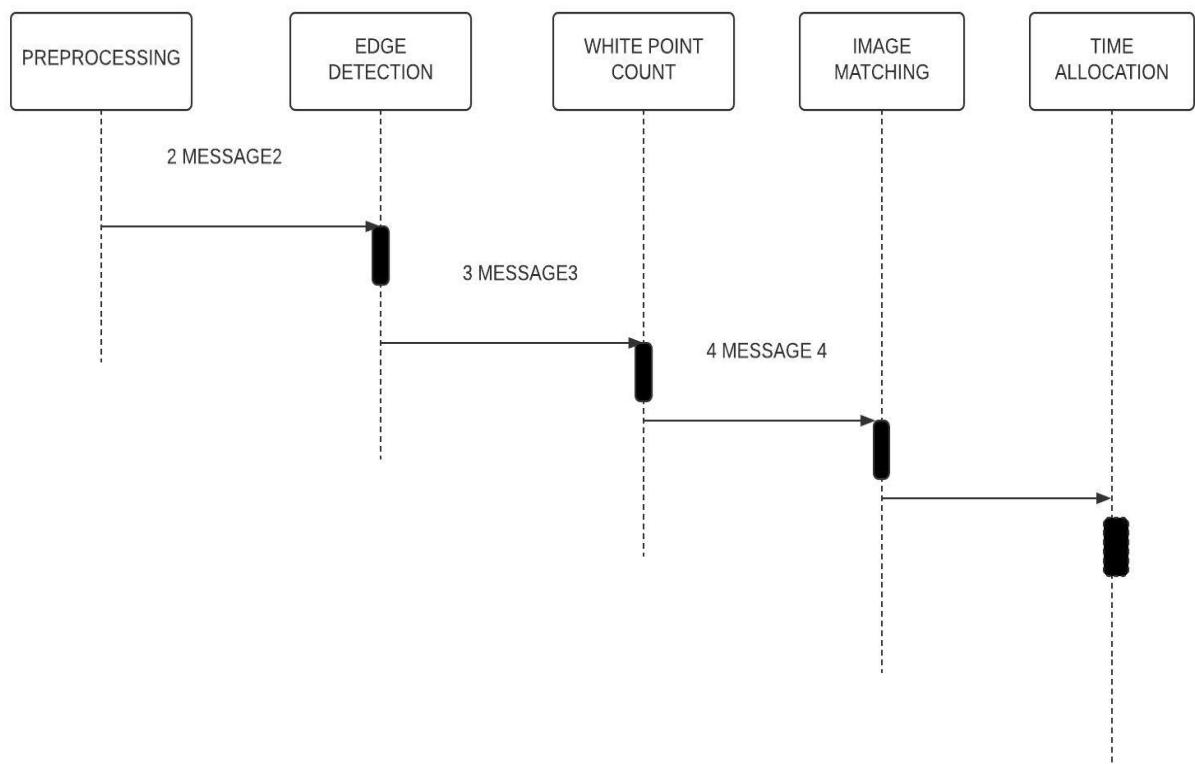


Fig. Sequence diagram for traffic density

#### 4. UML Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

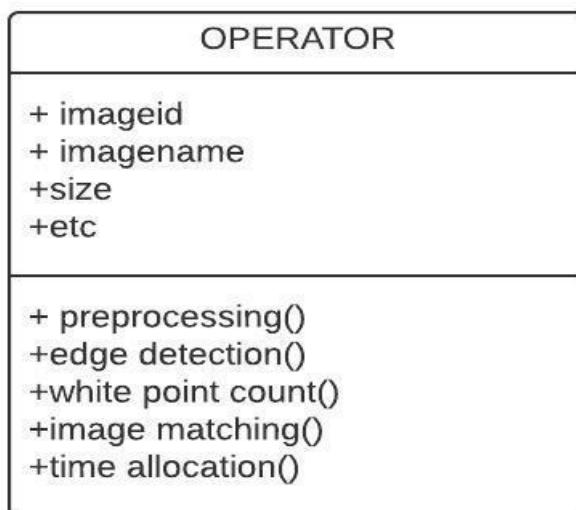


Fig. Class Diagram for smart traffic signal

## 5. UML Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

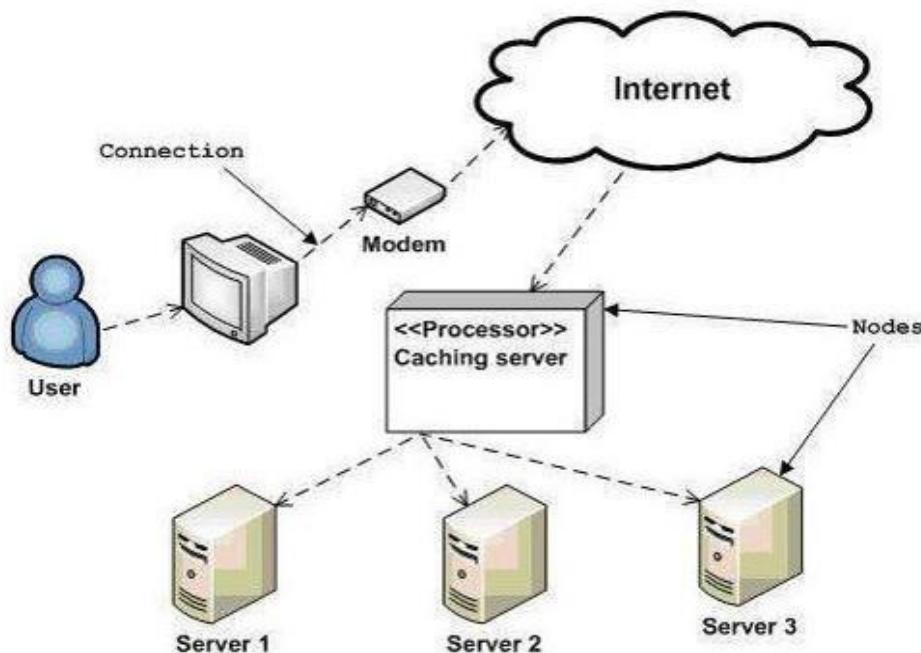


Fig. Deployment diagram

## 2.5 System specification

### 2.5.1 Hardware Specification

Processor	- Intel core i5
RAM	- 4 GB
Hard disk	- 1 TB

### 2.5.2 Software Specification

Front End	- HTML, CSS
Backend	- SQLyog
Client on PC	- Windows 7 and above.
Technologies used	- JS, HTML5, AJAX, J Query, Python, CSS, Android Studio

## 2.6 Software description

### 2.6.1 PYTHON

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3. Python 2 was discontinued with version 2.7.18 in 2020.

### 2.6.2 SQLyog

SQLyog is a GUI tool for the RDBMS MySQL. It is developed by Webyog, Inc., based in Bangalore, India, and Santa Clara, California. SQLyog is being used by more than 30,000 customers worldwide and has been downloaded more than 2,000,000 times. SQLyog is the most complete & easy MySQL GUI tool that helps save time & increase productivity through numerous powerful features such as autocomplete, query building, query profiler, Visual Schema designer and much more.

- ***Automatically Synchronize Data***

SQLyog Ultimate includes a power tool to automate and schedule the synchronization of data between two MySQL hosts. Create the job definition file using the interactive wizard. The tool does not require any installation on the MySQL hosts. You can use any host to run the tool.

- ***Visually Compare Data***

SQLyog Ultimate includes a power tool to interactively synchronize data. Run the tool in attended mode to compare data from source and target before taking action. Using the intuitive display, compare data on source and target for every row to decide whether it should be synchronized in which direction.

- ***Visually Compare and Synchronize Schema***

SQLyog Ultimate includes a power tool to interactively compare and synchronize schema. View the differences between tables, indexes, columns, and routines of two databases. The tool generates the correct scripts to synchronize the schema.

- ***Import External Data***

SQLyog Ultimate includes a power tool to import external data to transfer data from any ODBC compliant data source to MySQL. Use its interactive wizard to define its parameters. The import mechanism provides powerful and effective features, such as incremental import, and store and schedule import sessions.

- ***Schedule Backups***

SQLyog Ultimate includes a power tool to schedule the exporting of databases as SQL scripts for backup. The tool saves selected objects into a single file or multiple files. You can export schema only, or schema and data. The session establishes all required connections during the operation execution.

- ***Schedule and Report on Queries***

SQLyog Ultimate includes a power tool to schedule SQL queries. The tool generates, formats, and sends personalized emails with results returned from a scheduled query. You can also use this tool to execute maintenance queries. The interactive wizard validates queries before scheduling.

### 2.6.3 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development. Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

The following features are provided in the current stable version:

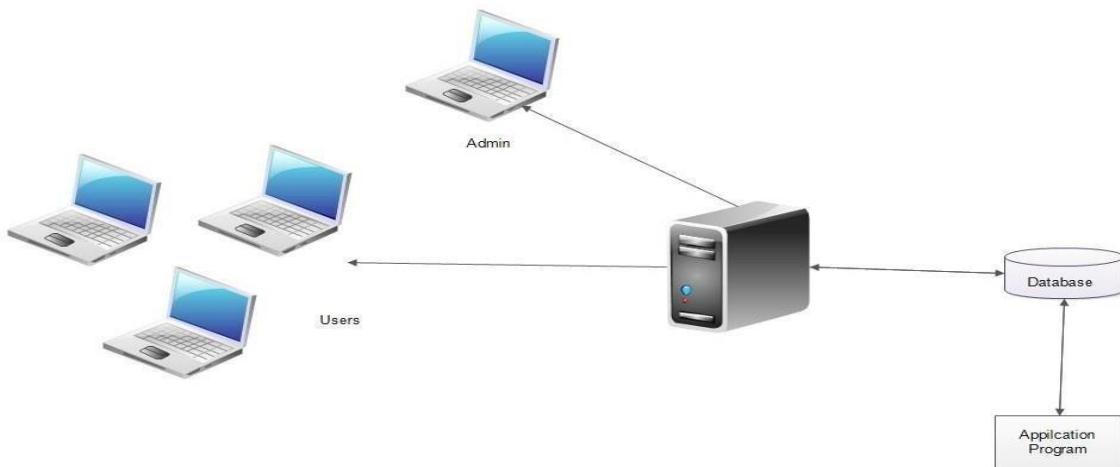
- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

## 2.7 System design

Design is the first step into the development phase for any engineered product or system.

Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

### 2.7.1 Architectural Design



The registered customer, seller, and admin can access the website through internet using their Laptop, Smart Phone, Tablet or Desktop Computer. The System's application program processes the user's request and provides the required services by taking data from the system database.

## 2.7.2 Module Design

### Admin Module

The administrator has a website of the company which is allowed to access all the services in the system.

Admin has the overall control of the system. Admin can add or update users, signals etc. Admin can View all the registered users, can able to edit traffic density and also can able to view all details.

Manage user details, add user types and traffic density	Deactivate/Activate the registered users.
Add/delete traffic signals	View traffic density

### Registered Traffic police Module

Traffic police have an application. The login credentials are sent by admin through mail. After login, traffic police can view signals, view traffic density, view notifications.

Login	View traffic signals/Track signal
Traffic density	View traffic density with images
View Notifications	Change passwords/forgot password

## Registered User Module

The User can register into this application and he/she can login. User can View profile, view traffic signal and track the signal and view traffic density.

User registration, login	View profile
Traffic density	View/track traffic signals/view traffic density
View Notifications	Change password/forgot password

### 2.7.3 Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

### ***Relational Database Management System (RDBMS)***

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

#### ***Relations, Domains & Attributes***

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

#### ***Relationships***

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key is Super Key and Candidate Keys.

#### ***Normalization***

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity.

Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies.

Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table.

There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

Normalize the data.

Choose proper names for the tables and columns.

Choose the proper name for the data.

### ***First Normal Form***

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute.

In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each nonatomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

### ***Second Normal Form***

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key.

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attribute of the relation is fully dependent on its primary key alone.

***Third Normal Form***

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on another non-key attribute.

**TABLES****Table No 01: users\_login**

Primary Key : id

A	B	C	D	E	F	G
id	Username	Password	User_type			
1	vismayakk2021@mca.majce.in	Vismaya@123	police			
2	vismayakk2021@mca.ajce.in	Vismaya@123	police			
3	amal@gmail.com	Amal@123	police			
4	admin	admin	admin			
5	nandana363@gmail.com	Nandhu	police			
6	Aswathi@gmail.com	Aswathi	police			
7	aswathik234@gmail.com	A123123a	police			
8	vishnu@gmail.com	Vishnu123	police			
9	vishnu@gmail.com	Vishnu123	police			
10	arun@gmail.com	Arun12323	police			
11	athulya@gmail.com	Athulya@123	police			
12						
13						
14						
15						
16						
17						
18						
19						

**Table No 02 : users\_add\_traffic\_police**

Primary Key : Traffic\_police\_id

Foreign Key : id

A	B	C	D	E	F	G	H	I	J	K	L
id	First_name	Last_name	Date_of_birth	Email_id	Phone_no	Police_station	Designation	Gender	Traffic_police_id_id		
1	Vismaya	kk	2021-03-16	vismaya@gmail.com	6282894418	Thalassery	head constable	female	1		
2	amal	p	1995-09-03	amal@gmail.com	9746243566	thalassery	assistant sub inspector	male	7		
3	Aswathi	N	2005-02-28	Aswathi@gmail.com	9789786789	chirakkara	head constable	female	10		
4	KK	pp	1997-12-12	aswathik234@gmail.com	6676785778	CALICUT	police constable	male	11		
5	vishnu	dev	1996-05-26	vishnu@gmail.com	9656954624	kollam	head constable	male	12		
6	vishnu	dev	1996-05-26	vishnu@gmail.com	9656954624	kollam	head constable	male	13		
7	athulya	p	1997-12-08	athulya@gmail.com	8978675645	dharmadam	senior police constable	female	15		
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											

**Table No 03: users\_traffic\_density**

Primary Key : Signal\_id

Foreign Key : id

A	B	C	D	E	F	G	H
id	Signal_status	Image	Direction	Signal_id_id			
1							
2							
3							
4							
5							
6							
7							
8							
9							
0							
1							
2							
3							
4							
5							
6							
7							

**Table No 04: users\_signal**

Primary Key: id

L28	A	B	C	D	E	F	G	H	I	J
1	id	Longitude	Latitude	Locality	Signal_name	Signal_type	Signal_info	No_of_dir		
2	3	kkjkj	dnfj	city	caltex	red	city	3		
3	4	110	45	flyover	kannur	green	around 2pm	4		
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										

## Form Design

### Home Page

	<a href="#">logout</a>
	<a href="#">Change password</a>
Menu	
Traffic Police	
Add Traffic Police	
View Traffic Police	
Traffic Density	
Traffic Signal	
Add Traffic Signal	
View Traffic Signal	
Notification	
Upload Image	

### Login

Login	
Username:	<input type="text"/>
Password:	<input type="password"/>
Submit	
<a href="#">Forgot password?</a>	

Add Traffic Police	
Firstname:	<input type="text"/>
Lastname:	<input type="text"/>
Dob:	<input type="text"/> mm / dd / yyyy <input type="button" value="Calendar"/>
Police Station:	<input type="text"/>
Designation:	police constable <input type="button" value="▼"/>
Gender:	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Others
Email:	<input type="text"/>
Phone:	<input type="text"/>
Password:	<input type="text"/>
Confirm Password:	<input type="text"/>
<input type="button" value="Register"/>	

<a href="#">View more</a>	
Firstname:	<input type="text"/>
Lastname:	<input type="text"/>
Dob:	<input type="text"/> mm / dd / yyyy <input type="button" value="Calendar"/>
Police station:	<input type="text"/>
Designation:	<input type="text"/>
Gender:	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Others
Email:	<input type="text"/>
Phone:	<input type="text"/>
<input type="button" value="Update"/>	<input type="button" value="delete"/>

<b>Add Traffic Signal</b>	
Signal Type:	<input type="text"/>
Signal Name:	<input type="text"/>
Signal Information:	<input type="text"/>
Locality:	<input type="text"/>
Latitude:	<input type="text"/>
Longitude:	<input type="text"/>
No of direction:	<input type="text"/>
<input type="button" value="Register"/>	

## 2.8 System testing

### 2.8.1 Introduction

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term's verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appears to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness is supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## 2.8.2 Test Plan

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers are always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

### 2.8.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is whitebox oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

### 2.8.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

### 2.8.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

### 2.8.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs

- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

## 2.9 Implementation

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.

- Design of methods to achieve the changeover.

### 2.9.1 Implementation Procedures

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
  - Their confidence in the software is built up.
  - Proper guidance is imparted to the user so that he is comfortable in using the application.
- Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

### 2.9.2 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

### 2.9.3 Operational Document

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

## 2.9.4 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

# 2.10 Conclusion and Future Enhancements

## 2.10.1 Future Enhancement

- Apply this in real-time signals
- Provide option to add videos of traffic signals
- Provide more accuracy
- Provide more security.

## 2.10.2 Conclusion

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, applications. The proposed system introduces facility for users to track signals and identify traffic density and it is demo for smart traffic signal. By this we can reduce time in signal.

## **2.11 Bibliography**

### **Books/References:**

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- Pankaj Jalote, “*Software engineering: a precise approach*”, 2006.
- James lee and Brent ware Addison, “*Open source web development with LAMP*”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

### **Websites:**

- [www.w3schools.com](http://www.w3schools.com)
- [www.jquery.com](http://www.jquery.com)
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- [www.agilemodeling.com/artifacts/useCaseDiagram.html](http://www.agilemodeling.com/artifacts/useCaseDiagram.html)

## 2.12 APPENDIX

### 2.12.1 SAMPLE CODE

#### Urls.py

```
from django.urls import path
from .import views
urlpatterns = [
    # path('',views.home_page),
    path('home_page',views.home_page),
    path('traffic_police',views.add_traffic_police1),
    path('traffic_signal',views.add_traffic_signal),
    path('image_edit',views.image_edit),
    path('',views.login1),
    path('notification',views.notification1),
    path('upload_image',views.upload_image),
    path('view_police',views.view_police),
    path('view_more/<i>',views.view_more),
    path('view_more_signal/<i>',views.viewmore_signal),
    path('view_signal',views.view_signal),
    path('updatepolice/<id>',views.updatepolice),
    path('updatesignal/<id>',views.updatesignal),

    path('index',views.ind),
    path('logout',views.logout1),
    path('indexlog',views.index),
    path('viewimages/<id>',views.view_image),
    path('deleteimage/<id>',views.delete_image),
    path('editimage/<id>',views.edit_image),
    path('updateimage/<id>',views.update_image),
    path('traffic_density1',views.traffic_density1),
    //////// android paths
    path('user_log',views.and_login),
    path('user_signup',views.and_sign_up),
    path('myprofile1',views.and_profile),
    path('myprofile2',views.and_profile1),
    path('notification1',views.notification11),
    path('trafficsignal1',views.traffic_signals),
    path('viewmoresignal',views.view_signal1),
    path('Track',views.Track1),
    path('trafficdensity1',views.trafficdensity),
    path('forgotpassword',views.forgotpost),
    path('sendmail/<id>/<id1>',views.sendmail),
    path('change_password', views.change_password),
]

]
```

#### Traffic density.html

```
{% extends "index.html" %}
{% block body %}

<form id="form1" name="form1" method="post" action="">
    {% csrf_token %}
```

```


|                                                 |  |  |  |  |
|-------------------------------------------------|--|--|--|--|
| <label>&lt;/label&gt;</label>                   |  |  |  |  |
| &nbsp;                                          |  |  |  |  |
| <strong>Traffic Density&lt;/strong&gt;</strong> |  |  |  |  |
| #                                               |  |  |  |  |
| Signal Name:                                    |  |  |  |  |
| <label>&lt;/label&gt;</label>                   |  |  |  |  |
| &nbsp;Signal Status:                            |  |  |  |  |
| <label>&lt;/label&gt;</label>                   |  |  |  |  |
| &nbsp;Image:                                    |  |  |  |  |
| <label>&lt;/label&gt;</label>                   |  |  |  |  |
| &nbsp;Direction:                                |  |  |  |  |


```

```

</tr>
{%
  for foo in p %
}
<tr>
  <td>{{ forloop.counter }}&nbsp;</td>
  <td>&nbsp;{{ foo.Signal_id.Signal_name }}</td>
  <td>{{ foo.Signal_status }} Vehicles</td>
  <td>&nbsp;</td>
  <td>{{ foo.Direction }}</td>

```

```

</tr>
{%
  endfor %
}
</table>
</form>
{%
  endblock %
}
```

### Add Traffic police

```

{%
  extends "index.html"
}
{%
  block body %
}

<form id="form1" name="form1" method="post" action="traffic_police">
  {%
    csrf_token %
  }
  <table width="527" border="1">
    <tr>
      <td colspan="2"><label></label>
    <td align="center"><strong>Add Traffic Police</strong></td>
    </tr>
    <tr>
      <td width="227"><label></label>
      <td>&nbsp;Firstname:</td>
      <td width="284"><label for="textfield"></label>
    <input type="text" name="textfield" id="textfield" pattern="[A-Za-z]{1-15}" title="firstname should only contain lowercase and uppercase letters" required /></td>
    </tr>
    <tr>
      <td><label></label>
      <td>&nbsp;Lastname:</td>
      <td><label for="textfield2"></label>
    <input type="text" name="textfield2" id="textfield2" pattern="[A-Za-z]{1-15}" title="lastname should only contain lowercase and uppercase letters" required /></td>
    </tr>
    <tr>
      <td><label></label>
      <td>&nbsp;Dob:</td>

```

```

<td><label for="textfield6"></label>
<input type="date" name="textfield6" id="textfield6" required /></td>
</tr>
<tr>
<td><label></label>
  &nbsp;Police Station:</td>
<td><label for="textfield3"></label>
<input type="text" name="textfield3" id="textfield3" pattern="[A-Za-z]{1-15}" title="police station should only contain lowercase and uppercase letters" /></td>
</tr>
<tr>
<td><label></label>
  &nbsp;Designation:</td>
<td><label for="select"></label>
<select name="designation" id="desig" required>
  <option value="police constable">Police constable</option>
  <option value="senior police constable">Senior police constable</option>
  <option value="head constable">Head constable</option>
  <option value="assistant sub inspector">Assistant sub inspector</option>
  <option value="sub inspector">Sub inspector</option>
</select></td>
</tr>
<tr>
<td><label></label>
  &nbsp;Gender:</td>
<td><table width="200">
<tr>
<td><label>
  <input type="radio" name="RadioGroup1" value="male" id="RadioGroup1_0" />
  Male</label></td>
</tr>
<tr>
<td><label>
  <input type="radio" name="RadioGroup1" value="female" id="RadioGroup1_1" />
  Female</label></td>
</tr>
<tr>
<td><label>
  <input type="radio" name="RadioGroup1" value="others" id="RadioGroup1_2" />
  Others</label></td>
</tr>
</table></td>
</tr>
<tr>
<td><label></label>
  &nbsp;Email:</td>
<td><label for="textfield4"></label>
<input type="email" name="textfield4" id="textfield4" pattern="[^@\s]+@[^\s]+\.[^\s]{2,}$" required/></td>
</tr>
<tr>
<td><label></label>
  &nbsp;Phone:</td>
<td><label for="textfield5"></label>
<input type="tel" name="textfield5" id="textfield5" pattern="^([6-9][0-9]{9})$" required/></td>
</tr>
<tr>
<td><label></label>
  &nbsp;Password:</td>
<td><label for="textfield7"></label>
<input type="password" name="textfield7" id="textfield7" pattern="^(?=.*\d)(?=.*[a-zA-Z]).{8,}$" title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters" required/></td>
</tr>
<tr>
<td><label></label>

```

```

    &nbsp;Confirm Password:</td>
    <td><label for="textfield8"></label>
    <input type="password" name="textfield8" id="textfield8" /></td>
</tr>
<tr>
    <td colspan="2"><input type="submit" name="Register" id="Register" value="Register"
/></td>
</tr>
</table>
</form>

{ % endblock %}

```

## Android code

### Login

```

package com.example.trafficdensity;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.DefaultRetryPolicy;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class login extends AppCompatActivity implements View.OnClickListener {
    EditText e1;
    EditText p1;
    TextView t1,t2;
    Button b2;
    SharedPreferences sh;
    String ip,url;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        e1=(EditText)findViewById(R.id.uname);
        p1=(EditText)findViewById(R.id.pass);
        t1=(TextView)findViewById(R.id.signup);
    }
}

```

```

t1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), User_signup.class);
        startActivity(i);

    }
});
t2=(TextView)findViewById(R.id.t2);
t2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), Forgot_password.class);
        startActivity(i);

    }
});

b2=(Button)findViewById(R.id.button2);
sh= PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
ip=sh.getString("url","");
url= ip+"users/user_log";
b2.setOnClickListener(this);

}

@Override
public void onClick(View v) {
    final String username=e1.getText().toString();
    final String password=p1.getText().toString();

    if(username.equals(""))
    {
        e1.setError("*");
        e1.requestFocus();
    }
    if(password.equals(""))
    {
        p1.setError("*");
        p1.requestFocus();
    }
    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    StringRequest postRequest = new StringRequest(Request.Method.POST, url,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    // Toast.makeText(getApplicationContext(), "Welcome",
                    // Toast.LENGTH_LONG).show();
                    // Toast.makeText(getApplicationContext(), "2",
                    // Toast.LENGTH_LONG).show();

                    try {
                        JSONObject jsonObj = new JSONObject(response);
                        if (jsonObj.getString("status").equalsIgnoreCase("ok")) {

                            String id=jsonObj.getString("lid");
                            String typ=jsonObj.getString("type");
                            Sharedpreferences.Editor ed=sh.edit();

```

```

        ed.putString("lid",id);
        ed.commit();
        if(typ.equalsIgnoreCase("user")) {
            SharedPreferences.Editor ed1=sh.edit();
            ed1.putString("uid",jsonObj.getString("userid"));
            ed1.commit();
            Toast.makeText(getApplicationContext(), "Welcome",
Toast.LENGTH_LONG).show();
            Intent i = new Intent(getApplicationContext(),
userhome.class);
            startActivity(i);

        }
        if(typ.equalsIgnoreCase("police")) {
            SharedPreferences.Editor ed1 = sh.edit();
            ed1.putString("pid", jsonObj.getString("policeid"));
            ed1.commit();
            Toast.makeText(getApplicationContext(), "Welcome",
Toast.LENGTH_LONG).show();
            Intent i = new Intent(getApplicationContext(),
policehome.class);
            startActivity(i);
        }
        } else {
            Toast.makeText(getApplicationContext(), "Invalid User",
Toast.LENGTH_LONG).show();
        }

    } catch (Exception e) {
        Toast.makeText(getApplicationContext(), "Error" +
e.getMessage().toString(), Toast.LENGTH_SHORT).show();
    }
},
new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
// error
Toast.makeText(getApplicationContext(), "eeeeee" + error.toString(),
Toast.LENGTH_SHORT).show();
}
}
) {

// value Passing android to python
@Override
protected Map<String, String> getParams() {
SharedPreferences sh =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
Map<String, String> params = new HashMap<String, String>();

params.put("username", username);//passing to python
params.put("password", password);

return params;
}
};


```

```
int MY_SOCKET_TIMEOUT_MS = 100000;

postRequest.setRetryPolicy(new DefaultRetryPolicy(
    MY_SOCKET_TIMEOUT_MS,
    DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
    DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));
requestQueue.add(postRequest);

}

// if(view==t1){
//     Intent i=new Intent(getApplicationContext(), Forgot.class);
//     startActivity(i);
// }

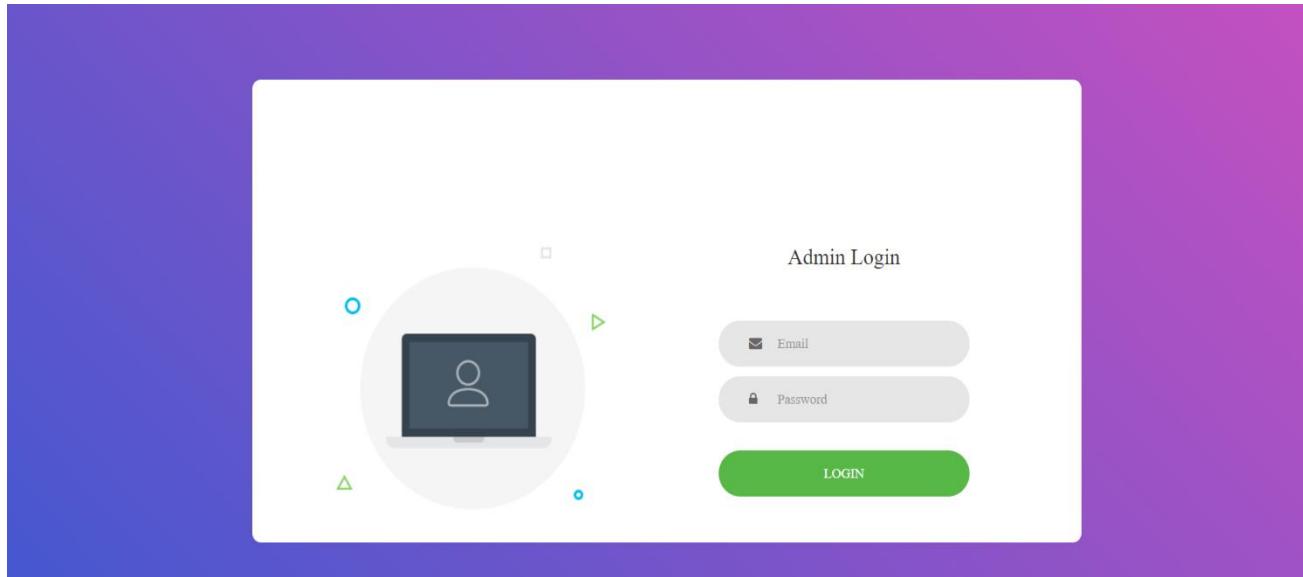
@Override
public void onPointerCaptureChanged(boolean hasCapture) {

}

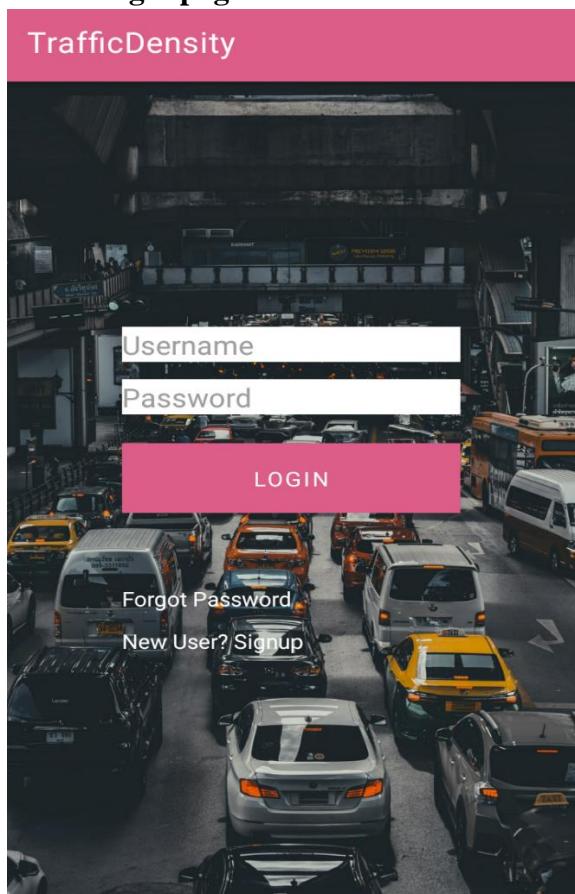
}
```

## 2.10.2 SCREENSHOTS

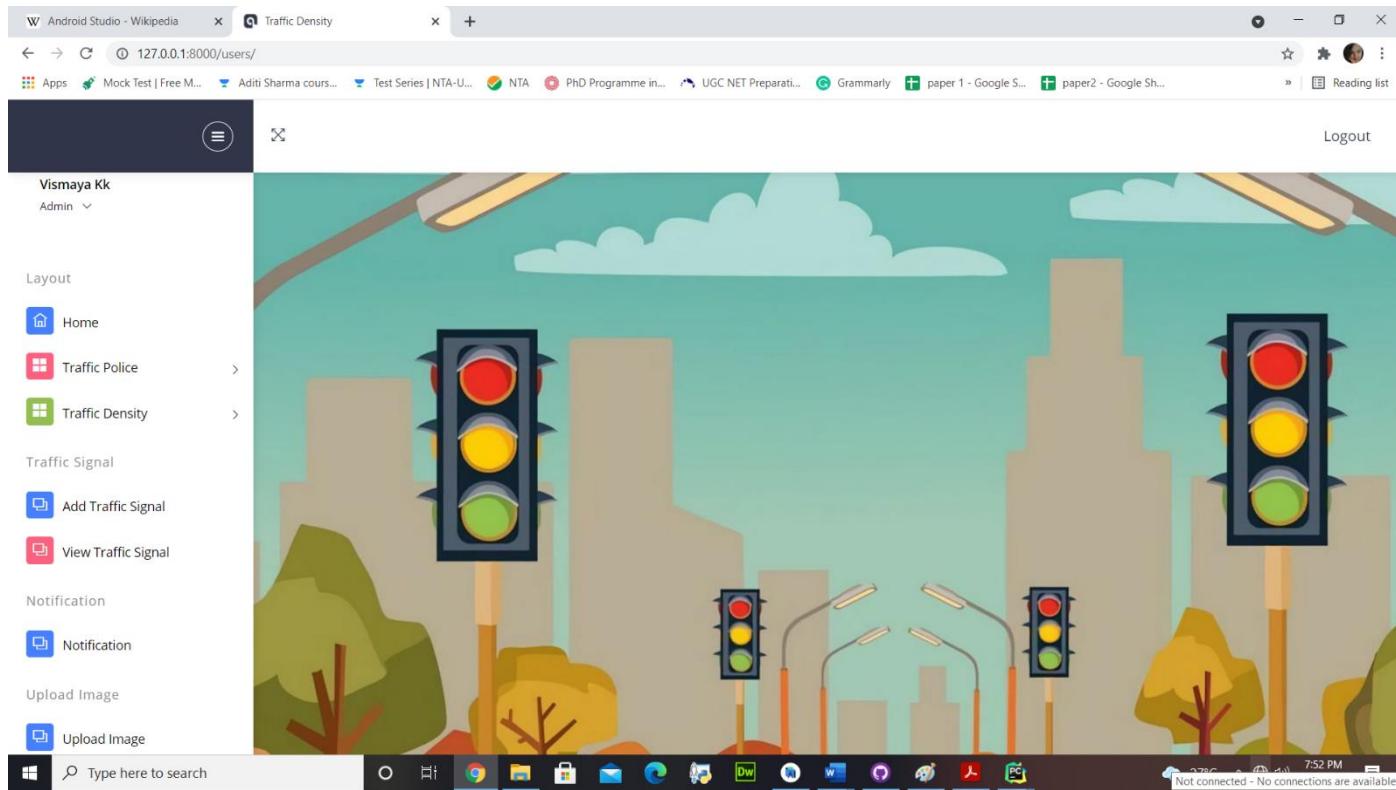
### Admin Login Page



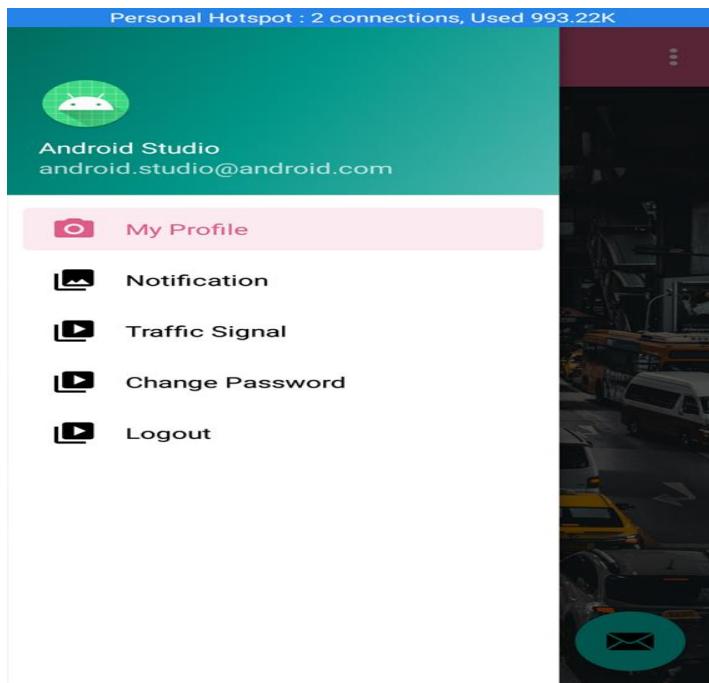
### Users Login page



## Admin Home Page



## User Home page



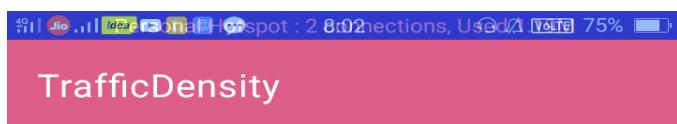
## Add Traffic Police page

Add Traffic Police	
Firstname:	<input type="text"/>
Lastname:	<input type="text"/>
Dob:	mm/dd/yyyy <input type="text"/>
Police Station:	<input type="text"/>
Designation:	Police constable <input type="button" value="▼"/>
Gender:	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Others
Email:	<input type="text"/>
Phone:	<input type="text"/>
Password:	<input type="text"/>
Confirm Password:	<input type="text"/>
<input type="button" value="Register"/>	

## Traffic Density Page

Traffic Density				
#	Signal Name:	Signal Status:	Image:	Direction:
1	thalassery	36 Vehicles		north
2	kuthuparamba	30 Vehicles		south
3	kuthuparamba	48 Vehicles		east
4	kuthuparamba	54 Vehicles		west
5	kuthuparamba	36 Vehicles		north

Traffic density user page



Direction south  
Traffic Density 30 Vehicles



Direction east  
Traffic Density 48 Vehicles



Direction west  
Traffic Density 54 Vehicles



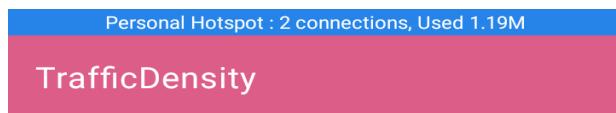
Direction north  
Traffic Density 36 Vehicles

## Add Traffic signal page

The screenshot shows a web application interface. At the top, there's a header bar with tabs like 'Android Studio - Wikipedia', 'Traffic Density', and a URL '127.0.0.1:8000/users/traffic\_signal'. Below the header is a navigation sidebar on the left containing links for 'Home', 'Traffic Police', 'Traffic Density', 'Traffic Signal' (which is currently selected), and 'Notification'. The main content area has a title 'Add Traffic Signal' and a form with fields for 'Signal Type' (dropdown menu), 'Signal Name' (text input), 'Signal Information' (text area), 'Locality' (text input), 'Latitude' (text input), 'Longitude' (text input), and 'No of direction' (text input). A 'Register' button is at the bottom of the form. The bottom of the screen shows a Windows taskbar with various icons and system status.

## View more signal

The screenshot shows a mobile application interface titled 'TrafficDensity'. At the top, it displays 'Personal Hotspot : 2 connections, Used 1.09M'. Below the title is a pink bar with the text 'View More Signal'. The main form has fields for 'Signal Name' (containing 'kuthuparamba'), 'Signal Type' (containing 'green'), 'Signal Information' (containing 'heavy traffic'), and 'Locality' (containing 'govt school pookode'). At the bottom, there are two buttons: 'TRACK' and 'TRAFFIC DENSITY'.

**Change Password Page**

### Change Password

**Current Password:**

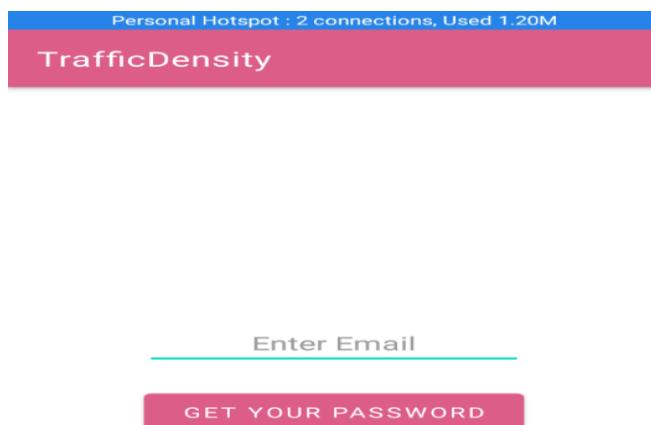
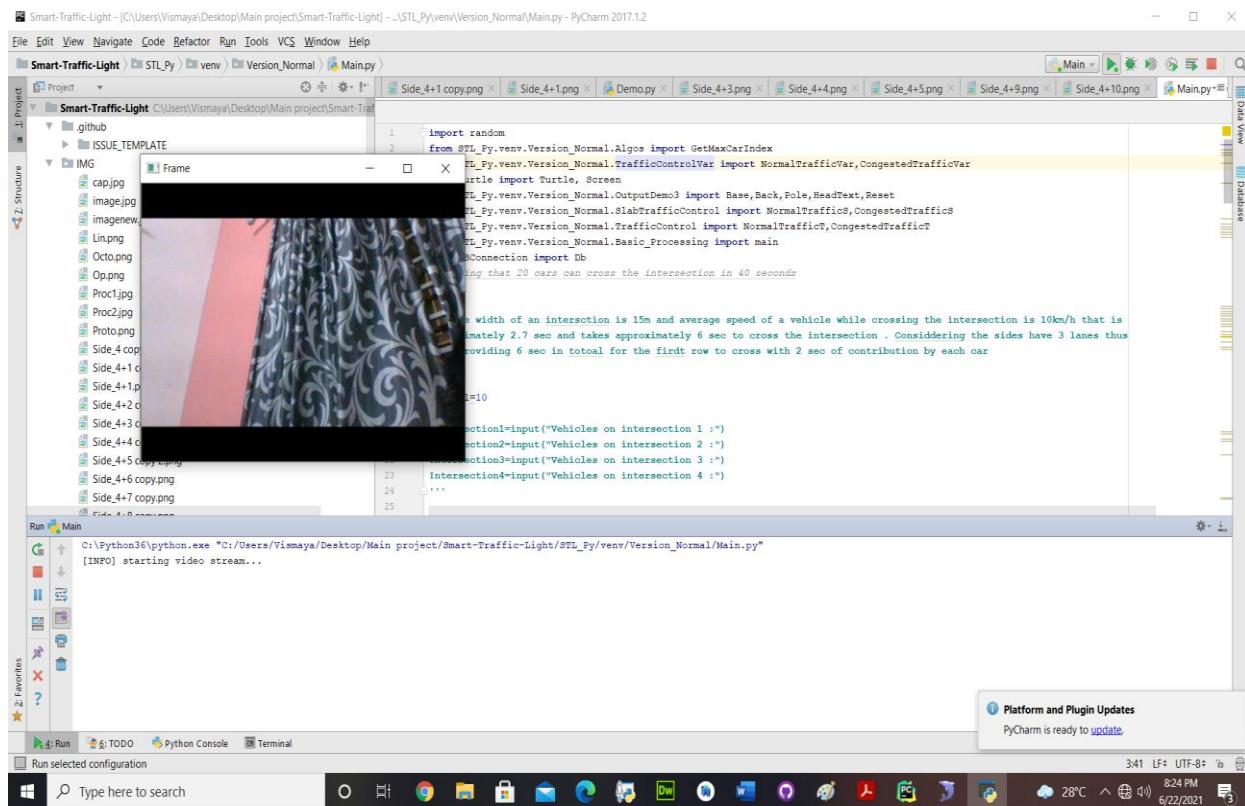
**New Password:**

**Confirm Password:**

**SUBMIT**

### Notification

A screenshot of a web browser window. The address bar shows the URL "127.0.0.1:8000/users/notification". The page content is a form titled "Notification". It has fields for "Subject:" and "Message:", and a "Submit" button. To the left of the form is a sidebar with navigation links: "Vismaya Kk" (Admin), "Layout" (Home, Traffic Police, Traffic Density), "Traffic Signal" (Add Traffic Signal, View Traffic Signal), "Notification" (Notification), and "Upload Image". The browser's toolbar at the bottom includes icons for search, refresh, and various applications like Google Chrome, Microsoft Word, and Microsoft Excel. The status bar at the bottom right shows the date and time as "6/22/2021 8:19 PM".

**User forgot password page****Camera module**

## Demo Traffic signal

