TRAFFIC OFFENCE REPORTING SYSTEM

Project Report Submitted By

ARYA SASI

Reg. No.: AJC20MCA-2025

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]

2021 -2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**TRAFFIC OFFENCE REPORTING SYSTEM"** is the bonafide work of **ARYA SASI (Reg.No:AJC20MCA-2025)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

Ms. Merin Chacko
Internal Guide

Ms. Nimmy Francis
Coordinator

Rev.Fr.Dr.Rubin Thottupurathu Jose
Head of the Department

DECLARATION

I hereby declare that the project report **“TRAFFIC OFFENCE REPORTING SYSTEM”** is a bonafided 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 2021-2022.

Date :21-07-2022	**ARYA SASI**
KANJIRAPPALLY	**Reg. No: AJC20MCA-2025**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms**. **Nimmy Francis** for the valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms**. **Merin Chacko** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

ARYA SASI

ABSTRACT

Roads are one of the main means of transport worldwide. When using the roads there are certain rules and regulations to be observed. Failure to observe these regulations can land one in jail, life sentence, death or disabled for the rest of their life. Some people fail to observe the rules and they end up committing a traffic offense.The main objective of this project for the better performance of Traffic Offense Reporting services and to manage that services through computerized system.

“ Traffic Offense Reporting” comprises of a web application that aims at helping the public to report the traffic offenses easily. The application will also contain the road traffic act which helps the drivers/ general public know their rights and the procedures they are supposed to follow, once caught with a traffic offense. Additionally, it seeks to reduce corruption through the introduction of an online payment system where the offenders will pay their fines or bails. It mainly contain four actors.

- Admin
- Officers
- Users
- Offender

.

CONTENT

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The actual strength of this project lies not merely in regulating traffic laws and offences, but also in developing deeper bonds with users and providing a high standard of service and support. A software programme called the Online Traffic Offense Reporting System Project eliminates the need for additional human record-keeping and report-generating hours. This website maintains the data in a central location that is simultaneously accessible to all users. It oversees the database's historical data. New thing that included in this project is that the public also have access to the web where he/she can view all the service such as reporting offense,rules,offense and fines.The Admin can access and manage all the data and features of the project while the Staff has only limited access. This system stores the list of traffic offenses and along with this data is the fine or penalty rate in each of these.This project has a user side where he/she can view and register the complaints. The Admin can also add the officers if he/she wants.Its is a State based Traffic offence reporting system.

1.2 PROJECT SPECIFICATION

The proposed system is a website in which user can report offense online. Also that the user can see the rules and offense details by accessing website.

The system includes 4 actors. They are:

- **Admin:**

  Overall control of the system.
- **Officers:**

  Control over application.
- **Users:**

  View services and can make an information request according to their needs.

  Can report Traffic Offense and its related documents.
- **Offender:**

  Payment of committed offence and Vie committed offence list.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is the process of acquiring and analysing data, diagnosing issues, and using the data to suggest system changes. The system users and system developers must communicate extensively during this problem-solving process. A system analysis or research should be the first step in any system development process. The system is meticulously examined and assessed. The system analyst takes on the role of an interrogator and investigates the operation of the current system in great detail. The system's input is identified, and the system as a whole is viewed. The various procedures can be linked to the outputs from the organisations. Understanding the issue, identifying the pertinent and important variables, evaluating and synthesising the many elements, and selecting the best or, at the very least, most acceptable course of action are all part of system analysis.

The process must be thoroughly studied using a variety of methodologies, including questionnaires and interviews. To reach a conclusion, the information gathered by these sources must be carefully examined. Understanding how the system works is the conclusion. The current system is the name of this system. Now, the current system is carefully examined, and issue areas are found. Now the designer takes on the role of a problem-solver and attempts to fix the problems the company is experiencing. The solutions are replaced with proposals. The best option is then selected after an analytical comparison of the proposal and the existing system. The user is given the opportunity to approve or reject the suggestion. On user request, the proposal is assessed and appropriate revisions are made. As soon as the user is content with the suggestion, this loop breaks.

The process of acquiring and analysing data in order to use it for future system studies is known as preliminary study. Initial research is a problem-solving activity that necessitates close coordination between system users and developers. It conducts a number of feasibility studies. These investigations provide an approximate estimate of the system activities, which can be used to determine the tactics to be used for an efficient system research and analysis.

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. The existing system have the disadvantages such as time consuming and expensive.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system customers can view all information related to traffic offense and can report offense.

2.3 DRAWBACKS OF EXISTING SYSTEM

- No proper online management of system
- Human effort is needed.
- It is difficult to maintain important information in books.
- There is no automated report generation.
- Lack of validations.
- No proper Data collection and processing.

2.4 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 and user attractive for growth of service center,on such consideration the system is proposed. In our proposed system there is admin who can view all the customers. It allows public to report offense and the offender do their penality by using online payment method .Users of this proposed system are admin,officer,public and offender. The software programme that reduces the amount of time needed to manually maintain records and provide reports This programme maintains the data in a central location that is simultaneously accessible to all users. Managing historical data in a database is extremely simple. No specific training is required for the distributors to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. It is very easy to record the information of offense and payment in the databases.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is relatively easy to create and put into operation. The system works in practically any setup and calls for very little in the way of system resources. It has the following attributes:

- **Enhanced safety: -**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

- **Ensure data precision: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

- **Availability and continuity of services: -**

The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is conducted to determine whether the project will, upon completion, fulfil the objectives of the organisation in relation to the work, effort, and time invested in it. A feasibility study enables the developer to predict the project's usefulness and potential future. A system proposes the workability, which includes the influence on the organisation, capacity to satisfy user needs, and efficient use of resources, is the basis for a feasibility study. As a result, before a new application is accepted for development, it often undergoes a feasibility assessment.

The document outlines the project's viability and contains a number of factors that were carefully taken into account throughout this project's feasibility study, including its technical, economic, and operational viabilities. It has the following characteristics: -

3.1.1 Economical Feasibility

Cost and benefit analyses are required to support the emerging system. Criteria to make sure that focus is placed on the project that will yield the best results the earliest. The price that would be involved in developing a new system is one of the variables.

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 was created as part of a project; hence, there are no manual expenses associated with it. Additionally, the fact that all of the resources are already at hand indicates that the system may be developed affordably.

The cost of the Traffic Squard project was broken down into three categories: system costs, development costs, and hosting costs. All calculations indicate that the project was developed at a modest cost. As open source software was used to develop it entirely.

3.1.2 Technical Feasibility

The system needs to be assessed first from a technical standpoint. An overview design of the system's requirements in terms of input, output, programmes, and procedures must serve as the foundation for the assessment of this viability. The inquiry must next advise the kind of equipment, necessary procedure for building the system, and means of once it has been created, using the system after having identified an outline system.The following technical difficulties came up throughout the investigation:

- Is the proposed technology compatible with the current technology?
- Can the system grow if it is improved?

The project should be designed in such a way that the necessary functionality and performance are met within the constraints. The project uses cryptographic methods and calls for a high resolution scanning device. The system may still be used even though the technology may become outdated after a while because a newer version of the same software still works with an earlier version. Therefore, this project only has a few limitations. The system was created using PHP for the front end and a MySQL server for the back end; it is technically feasible to complete the project.The system was created using PHP for the front end and a MySQL server for the back end; it is technically feasible to complete the project. The system was also well-performing, with an Intel i3 core processor, 4 GB of RAM, and a 1 TB hard drive.

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 advantageous because, when created and implemented, it would achieve the goals. The project is deemed to be behaviorally feasible after carefully weighing all behavioural factors.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	Intel core i3
RAM	-	4 GB
Hard disk	-	1 TB

3.2.2 Software Specification

Front End	-	HTML, CSS
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

In addition to being used as a general-purpose programming language, PHP is a server-side scripting language created for web development. Currently, PHP is present on 2.1 million web servers and more than 244 million websites. The reference implementation of PHP, which was first developed by Rasmus Ledorf in 1995, is now created by the PHP group. The recursive acronym PHP:HypertextPreprocessor has replaced the original meaning of PHP, which was personal Home page. A PHP processor module on a web server interprets PHP code to produce the final web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also grown to include a command-line interface and can be used in standalone mode, which is incompatible with the GNU General Public License (GPL) due to restrictions on the use of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL92" refers to the standard released in 1992, "SQL: 1999" refers to the standard released in 1999, and "SQL: 2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to sssdefine what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

  You ought to give it a shot if that is what you're after. In addition to your other apps, web servers, and other software, MySQL Server can function smoothly on a desktop or laptop while requiring little to no maintenance. You can modify the settings to utilise all the RAM, CPU power, and I/O capacity if you dedicate an entire machine to MySQL.MySQL Server works in client/server or embedded systems.

  The MySQL Database Software is a client/server system made up of a multi-threaded SQL server that supports several client programmes and libraries, administration tools, and a broad range of application programming interfaces (APIs). Additionally, we provide MySQL Server as an integrated multi-threaded library that you can link into your programme to create a standalone offering that is smaller, faster, and simpler to operate.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Any engineered system or product's development process begins with design. A creative process is design. The secret to an efficient system is a decent design. The process of using different methodologies and concepts to specify a process or a system in enough detail to allow for its physical implementation is referred to as "design." One way to describe it is as the process of using different methodologies and concepts to specify a device, a process, or a system in enough detail to allow for its physical reality. Regardless of the development paradigm that is employed, software design forms the technical core of the software engineering process.The architectural detail needed to construct a system or product is developed through the system design. This programme has also through the best possible design phase, fine tuning all efficiency, performance, and accuracy levels, as in the case of any systematic technique. A user-oriented document is converted into a document for programmers or database staff throughout the design phase. The two stages of system design development are logical design and physical design.

4.2 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.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- Statechart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between system components. A approach for identifying, outlining, and organising system requirements is called a use case. The word "system" here refers to a thing that is being created or run, like a website for mail-order product sales and services. UML (Unified Modeling Language), a standard language for the modelling of real-world objects and systems, uses use case diagrams.

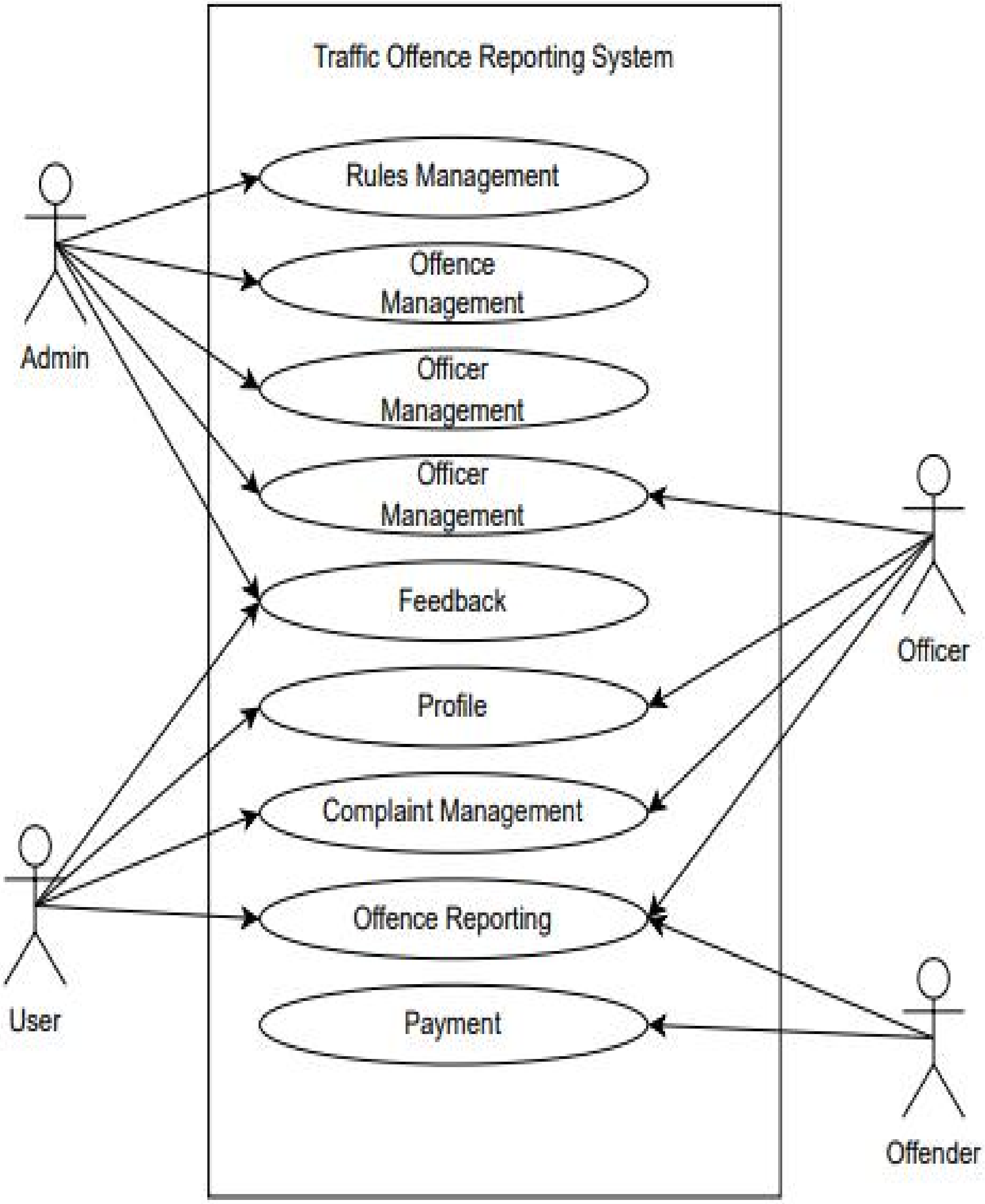
The planning of general requirements, the validation of a hardware design, the testing and debugging of a software product in development, the creation of an online help reference, or the completion of a job focused on customer support are all examples of system objectives. For instance, use cases in a product sales context can involve customer service, item ordering, catalogue updating, and payment processing. There are four elements in a use case diagram.

- 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 are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a wayso that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Use-Case Diagram



4.2.2 SEQUENCE DIAGRAM

A sequence diagram essentially shows how things interact with one another sequentially, or the order in which these interactions occur. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence diagrams show the actions taken by the components of a system in chronological order. Businesspeople and software engineers frequently use these diagrams to record and comprehend the requirements for new and current systems.

Sequence Diagram Notations –

i. **Actors –** In a UML diagram, an actor represents a particular kind of role that interacts with the system and its objects. An actor is always beyond the purview of the system that we want to use the UML diagram to represent. We employ actors to portray a variety of roles, including those of human users and other outside subjects. In a UML diagram, an actor is represented using a stick person notation. In a sequence diagram, there might be several actors.

ii. **Lifelines –** A lifeline is a named element in a sequence diagram that represents an individual participant. So, in a sequence diagram, each incident is represented by a lifeline. A sequence diagram's lifeline elements are at the top.

iii. **Messages –** Messages are used to show how objects communicate with one another. The messages are displayed on the lifeline in chronological sequence. Arrows are how messages are represented. A sequence diagram's main components are lifelines and messages.

Messages can be broadly classified into the following categories:

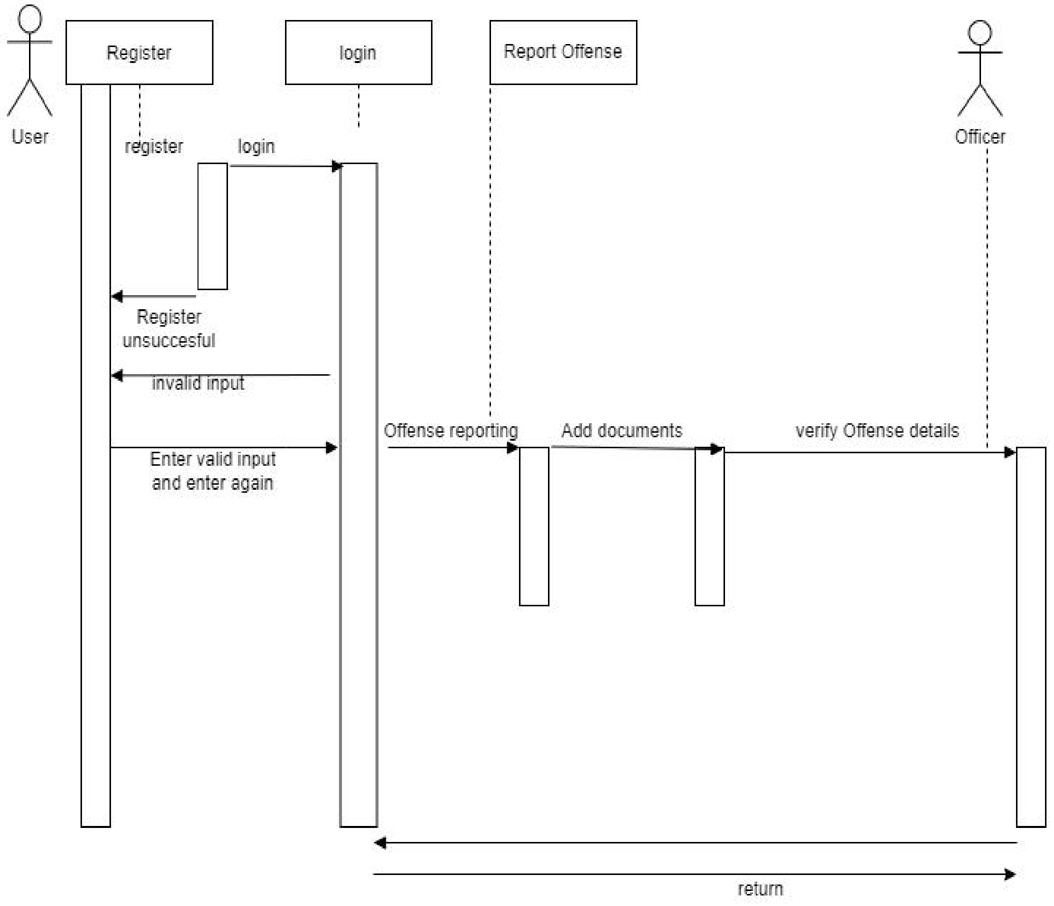
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

iv. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

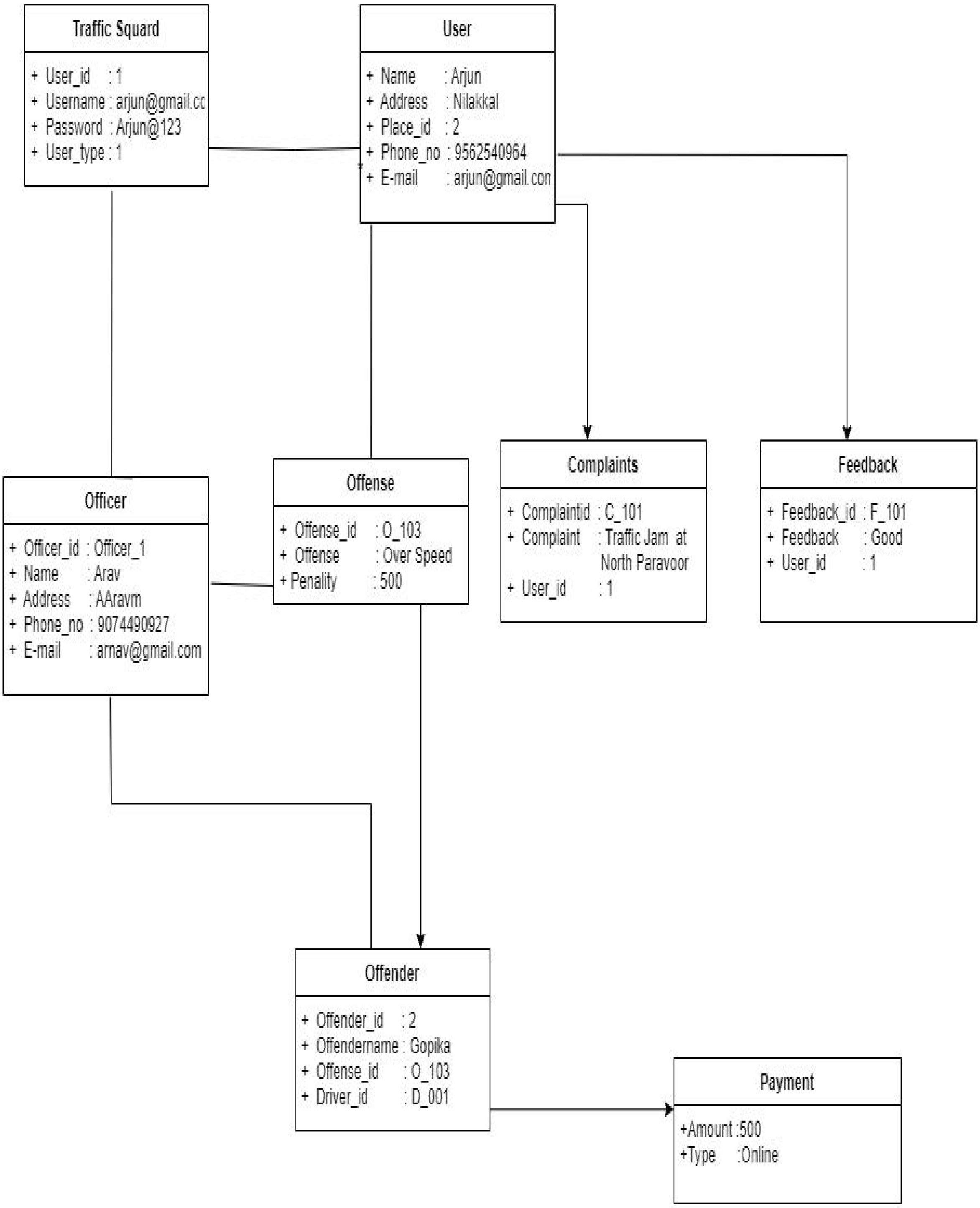
- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Sequence Diagram



4.2.3 Object Diagram

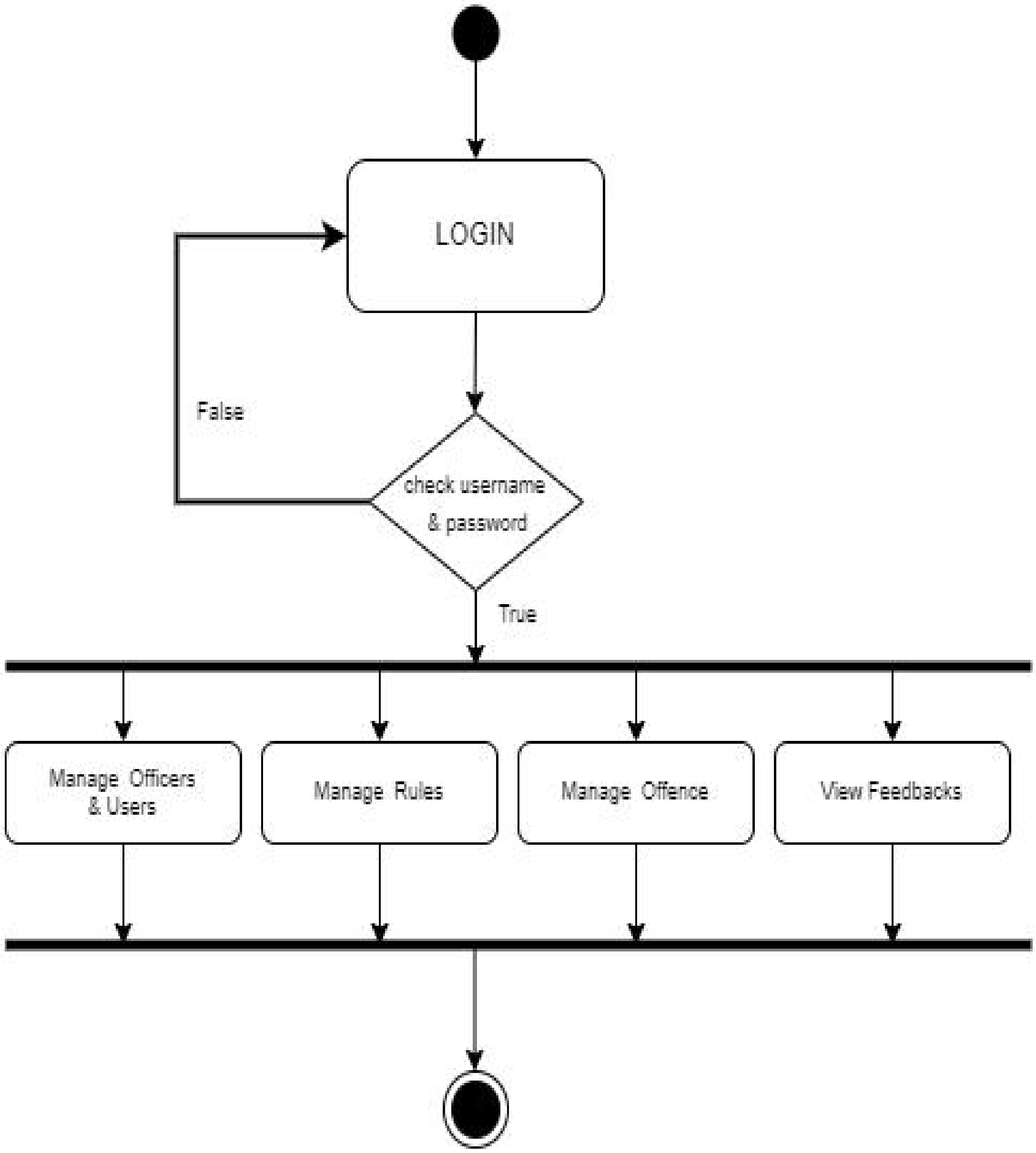
Since class diagrams are the source of object diagrams, class diagrams are a prerequisite for object diagrams. An instance of a class diagram is represented by an object diagram. Class and object diagrams both use the same fundamental ideas. The static view of a system is also represented by object diagrams, but this static view represents a momentary snapshot of the system. To represent a group of items and their connections as an instance, object diagrams are employed.



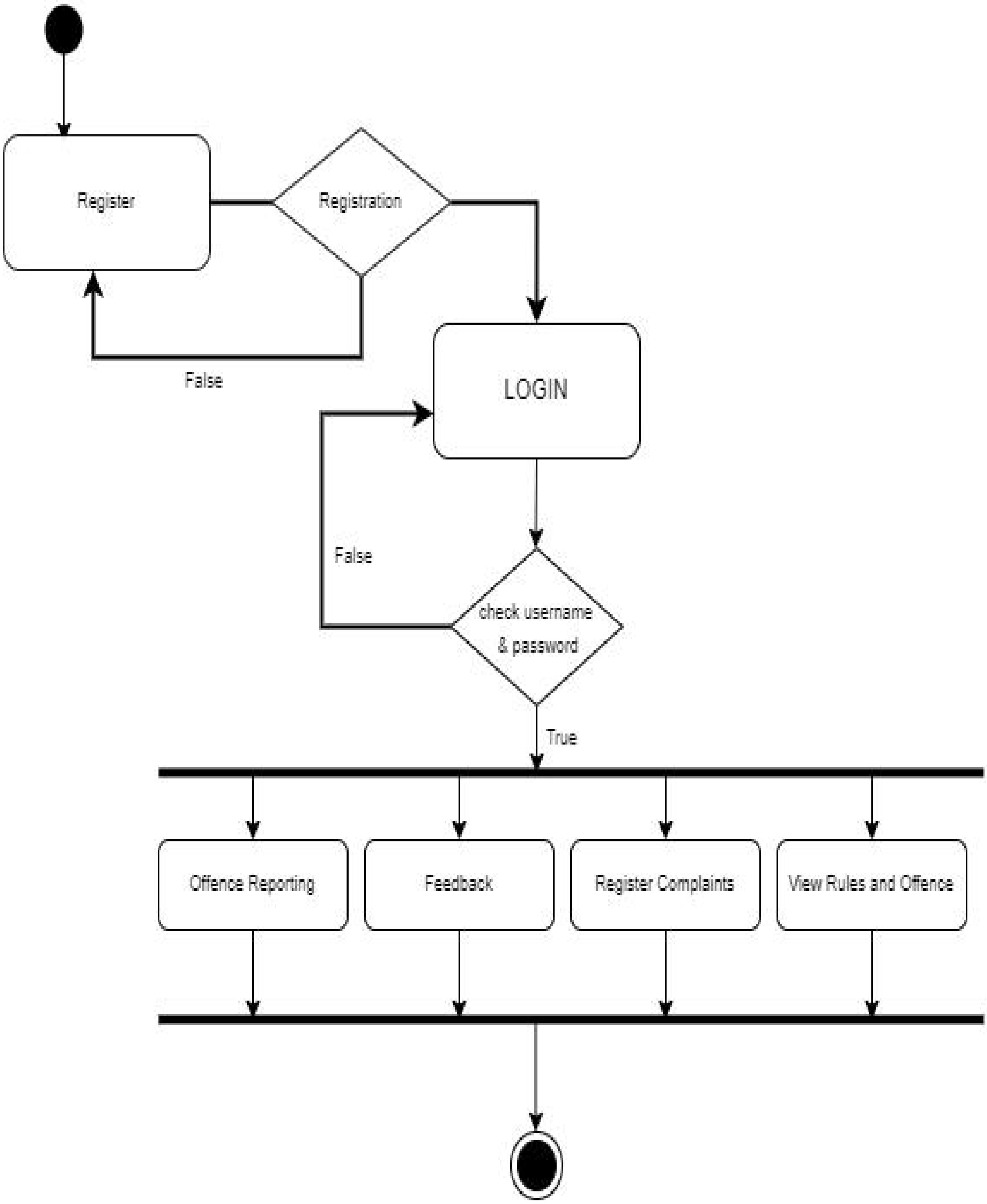
4.2.4 Activity Diagram

Another crucial UML diagram for describing the system's dynamic elements is the activity diagram. An activity diagram is essentially a flowchart that shows how one activity leads to another. The action might be referred to as a system operation. One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams use many features, such as fork, join, etc., to cope with all types of flow control.An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

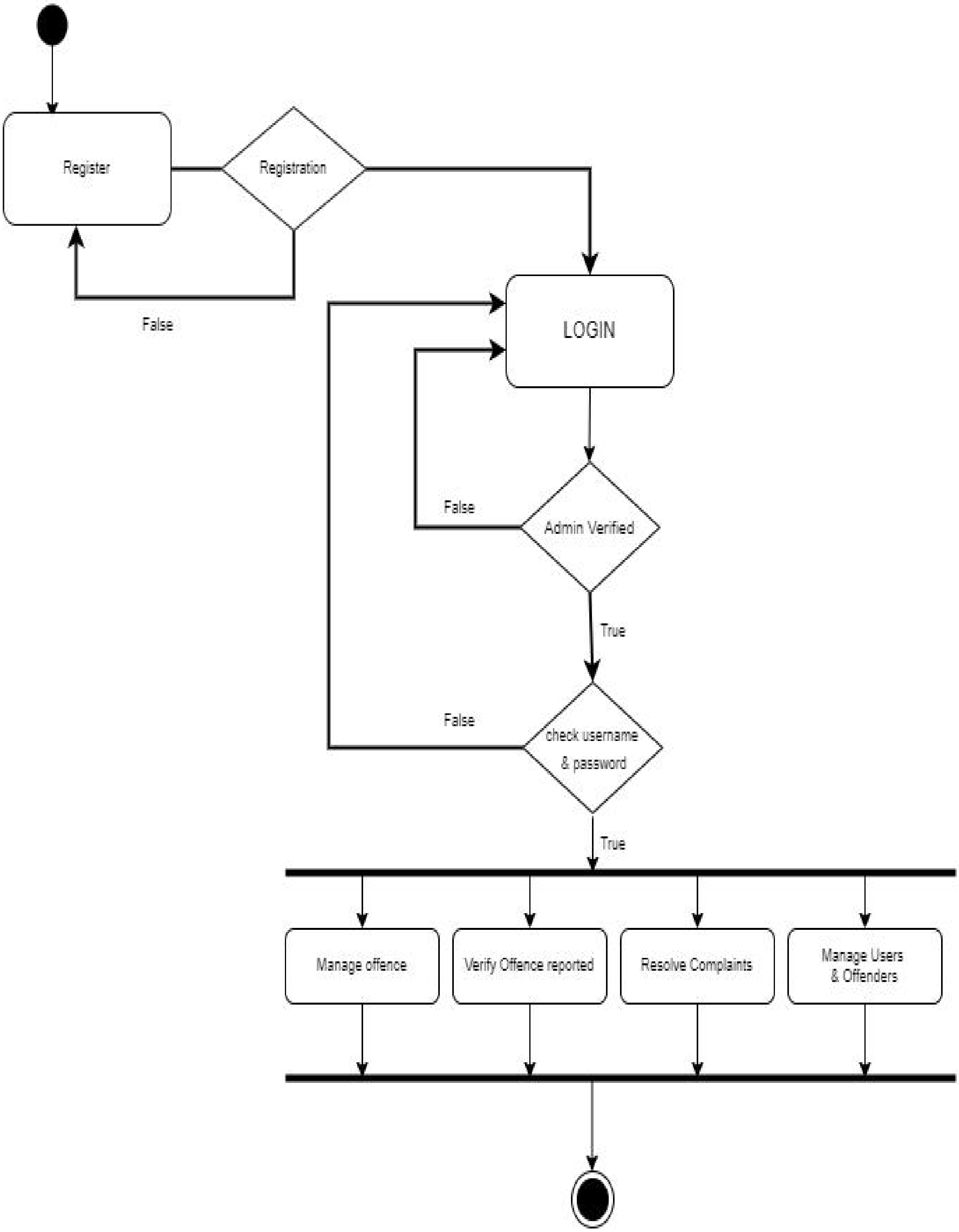
ADMIN



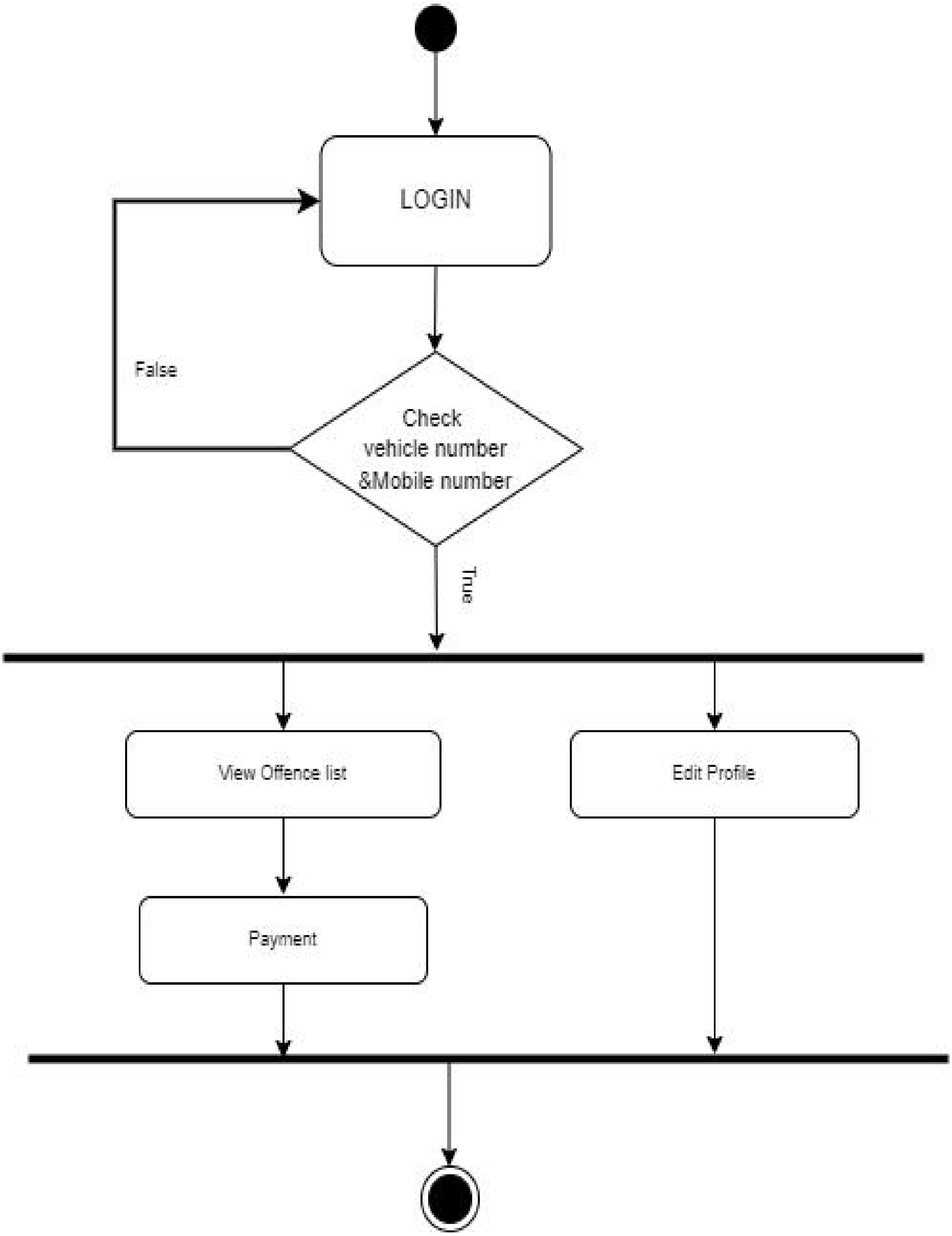
USER


Register
Registration
False
LOGIN
False
check username & password
True
Offence Reporting
Feedback
Register Complaints
View Rules and Offence

OFFICER

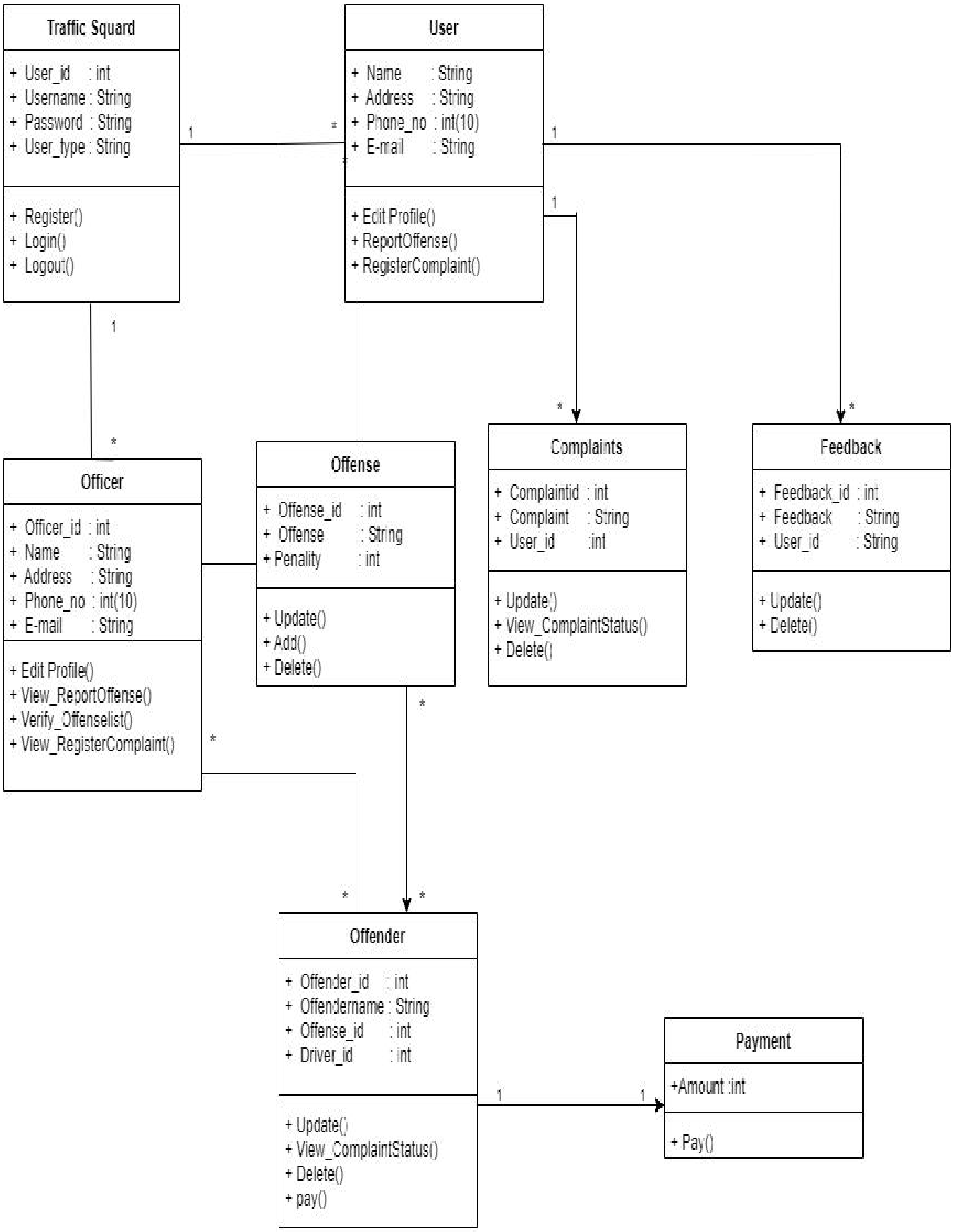

Register
Registration
False
LOGIN
False
Admin Verified
True
False
check username & password
True
Manage offence
Verify Offence reported
Resolve Complaints
Manage Users & Offenders

OFFENDER


LOGIN
False
Check
vehicle number
&Mobile number
True
View Offence list
Edit Profile
Payment

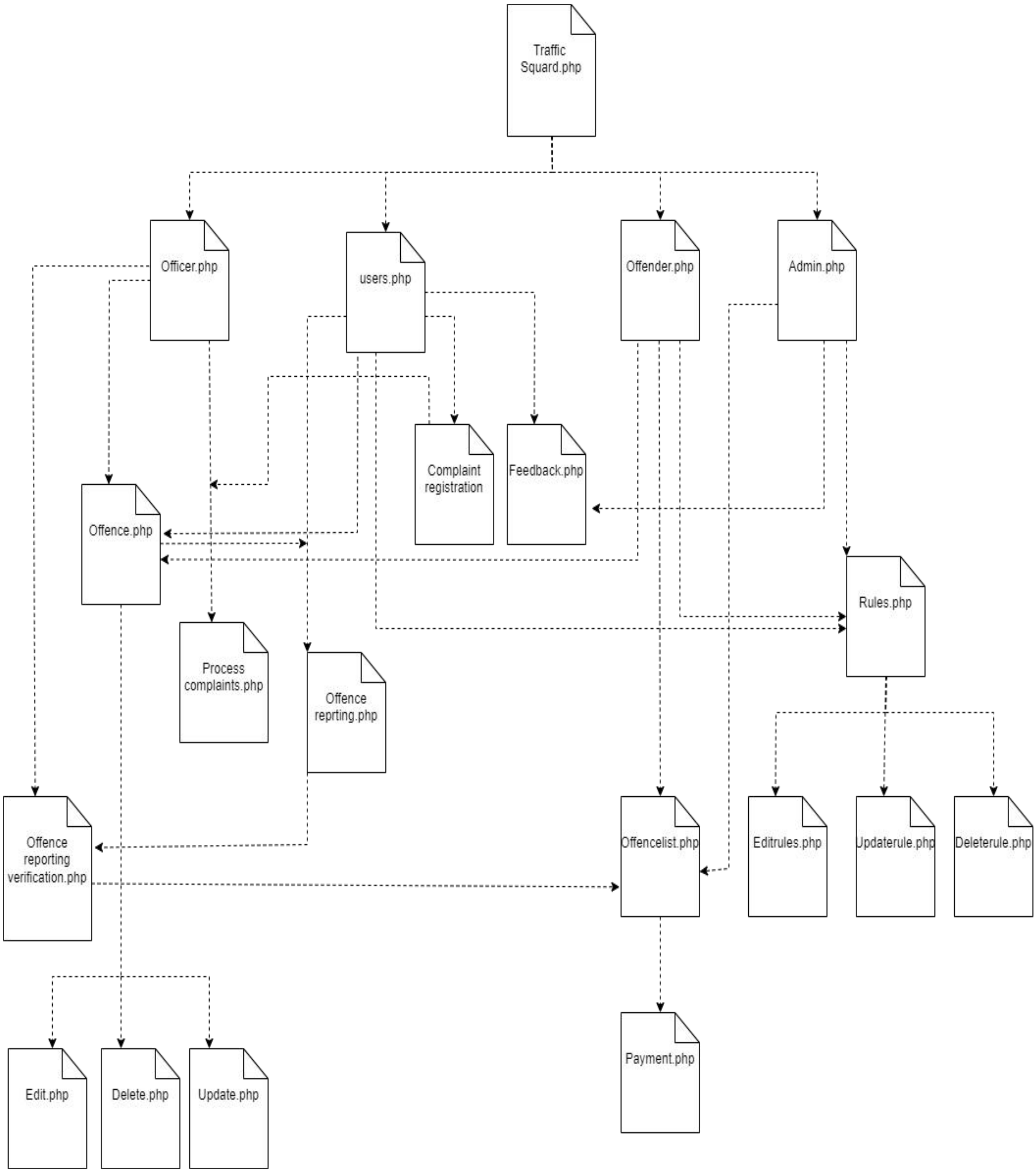
4.2.5 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling.



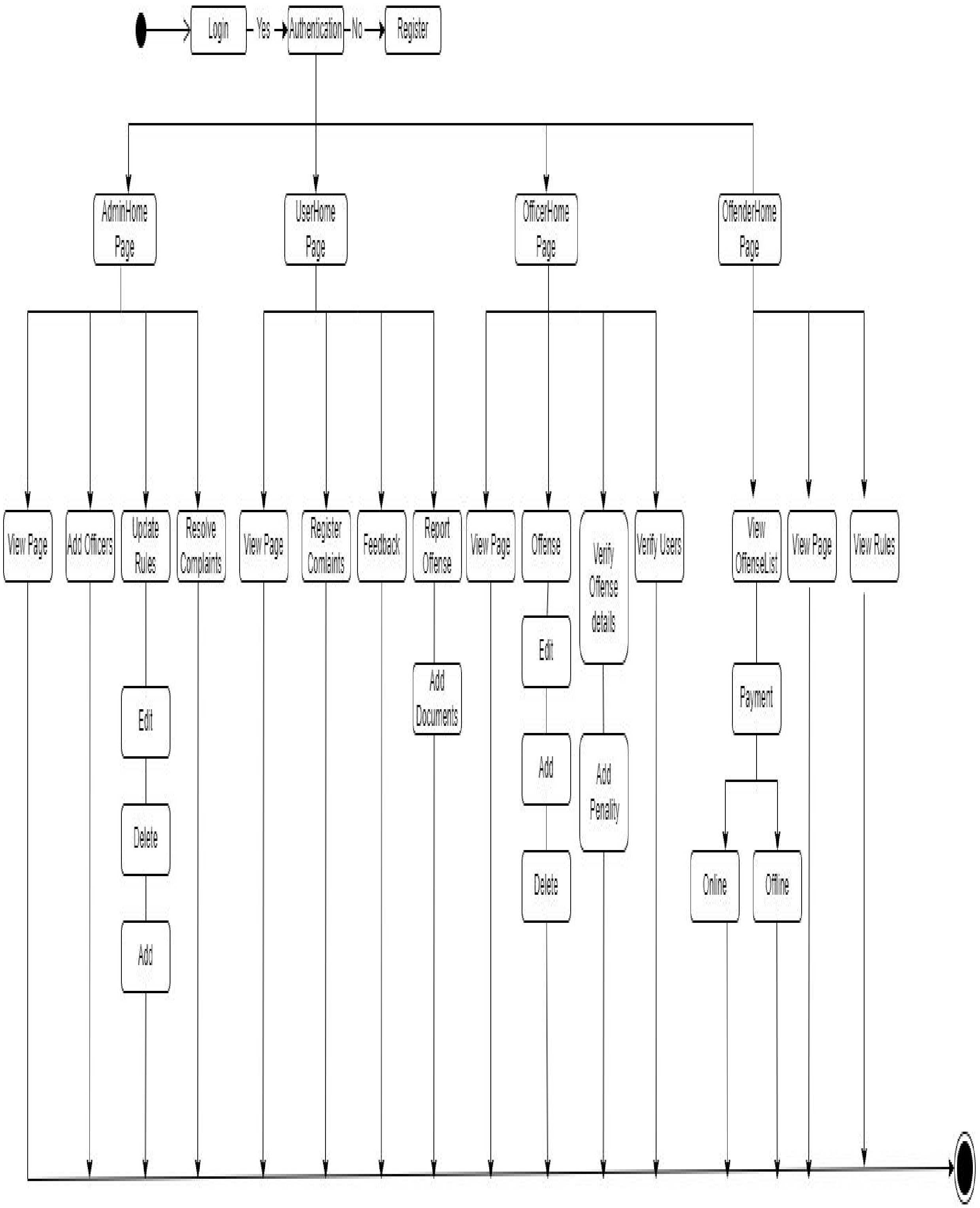
4.2.6 Component Diagram

A component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.



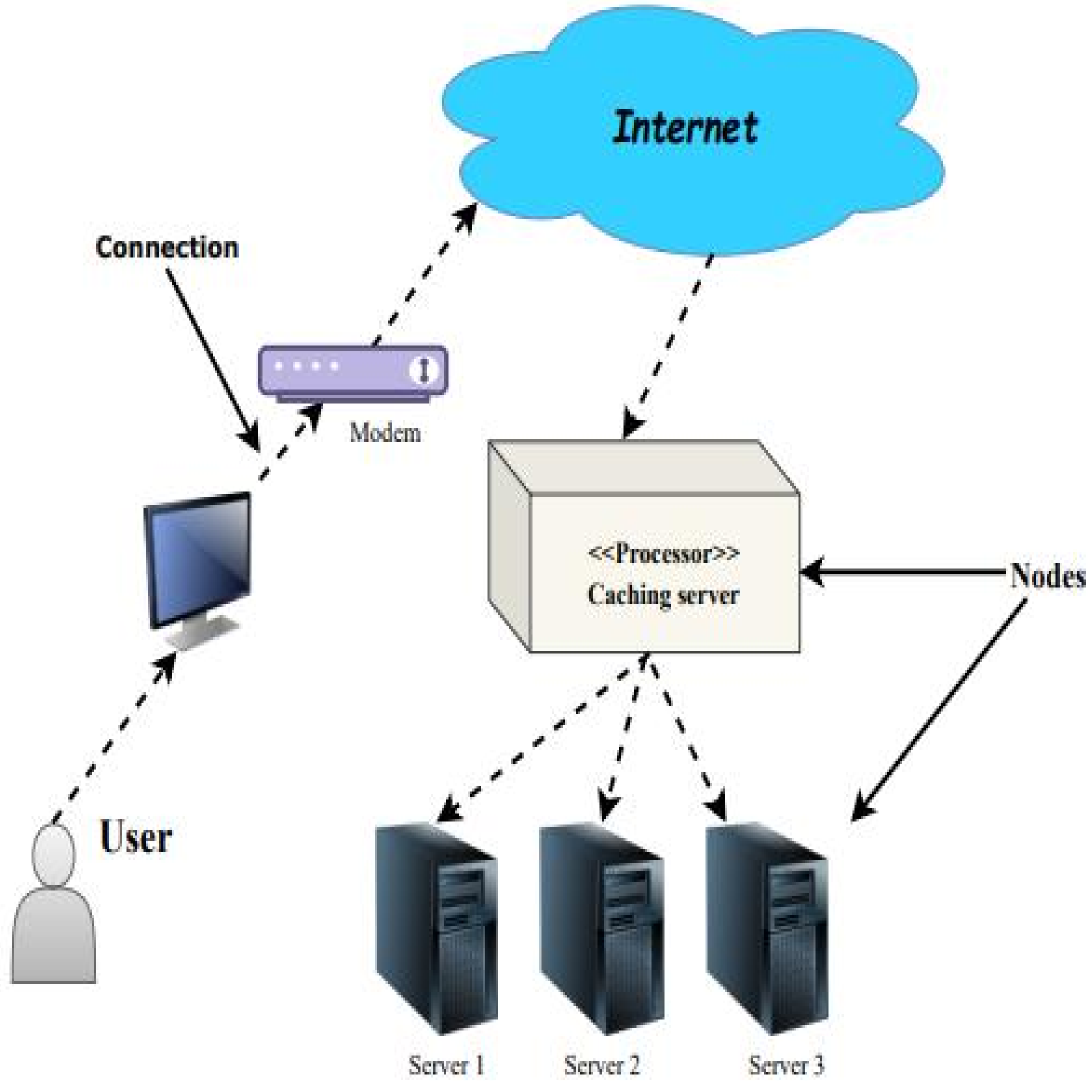
4.2.7 State Diagram

A state diagram, also known as a state machine diagram or statechart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language (UML).



4.2.8 Deployment Diagram

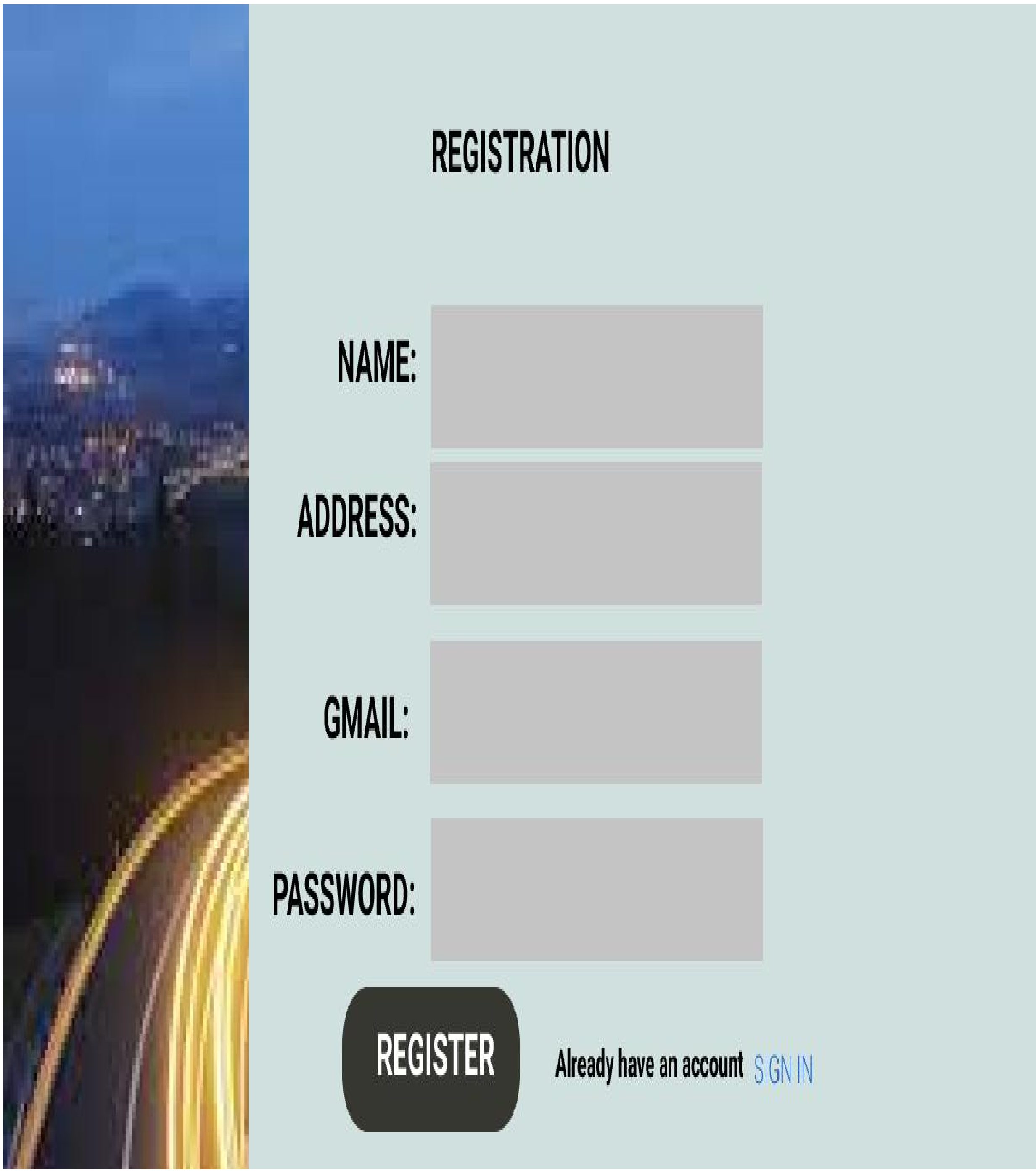
An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system. By using it, you can comprehend how the hardware will physically deliver the system. In contrast to other UML diagram types, which primarily depict the logical components of a system, deployment diagrams assist describe the hardware structure of a system.



4.3. USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

Form Name : User Registration



REGISTRATION

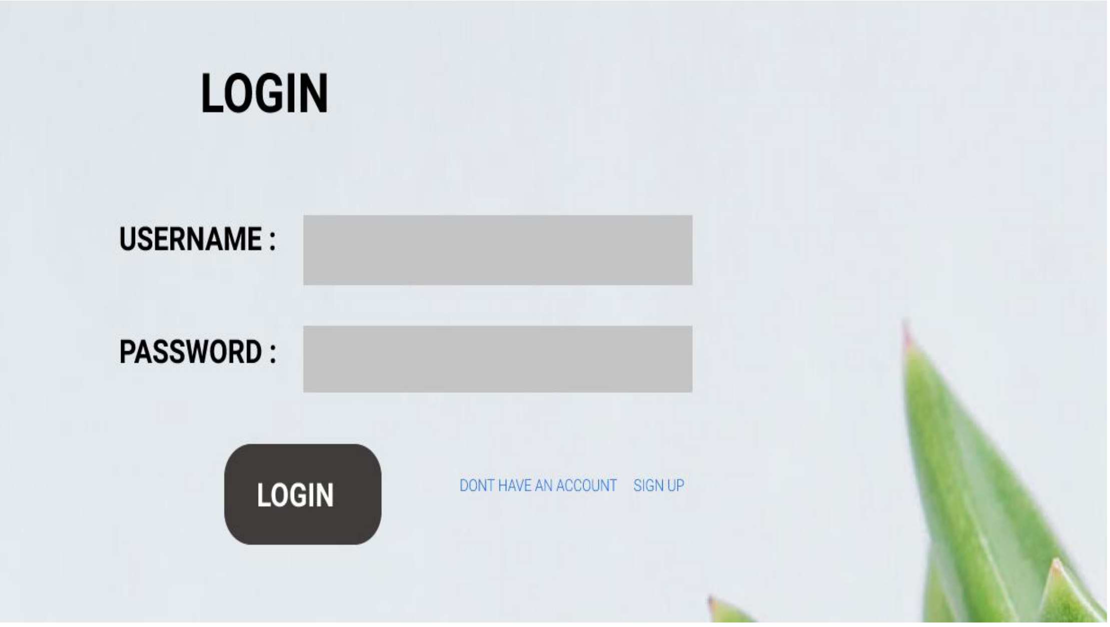
NAME:

ADDRESS:

GMAIL:

PASSWORD:

Form Name : User Login

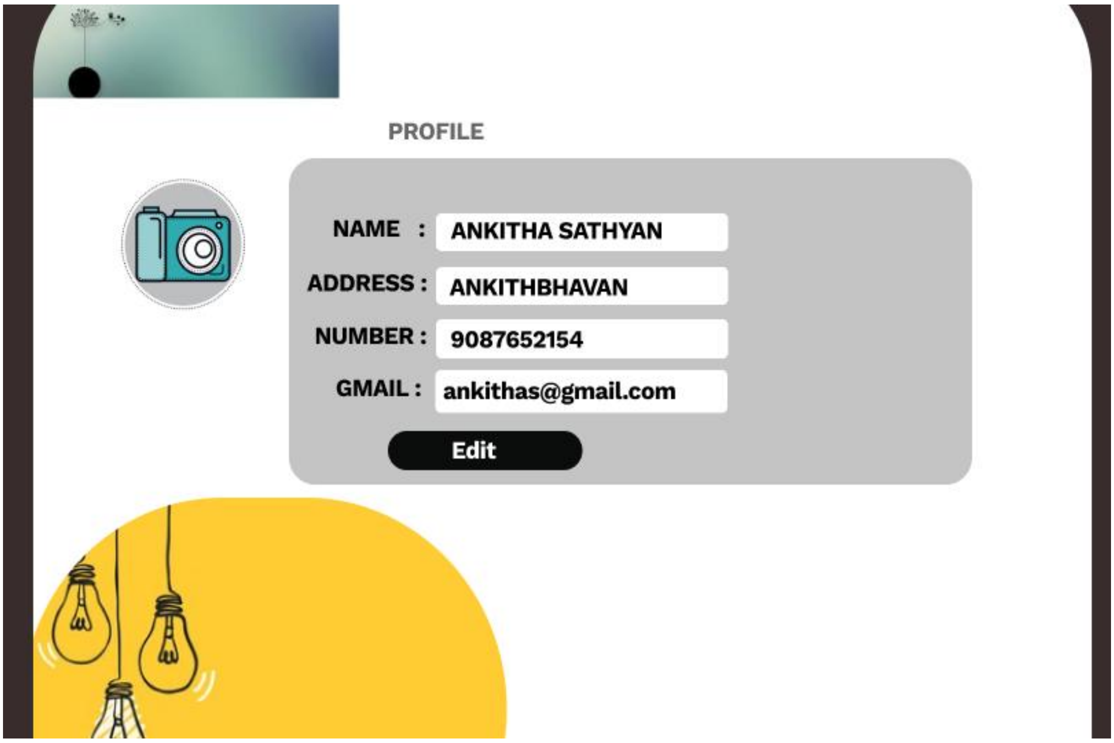


LOGIN

USERNAME :

PASSWORD :

Form Name : Profile

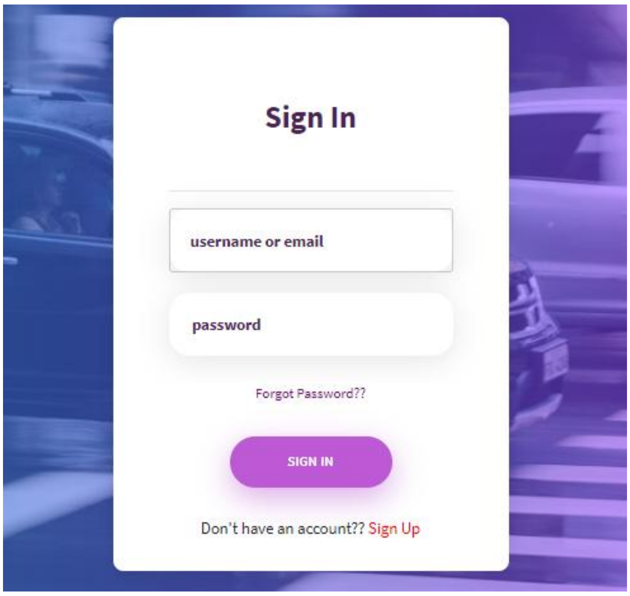


PROFILE

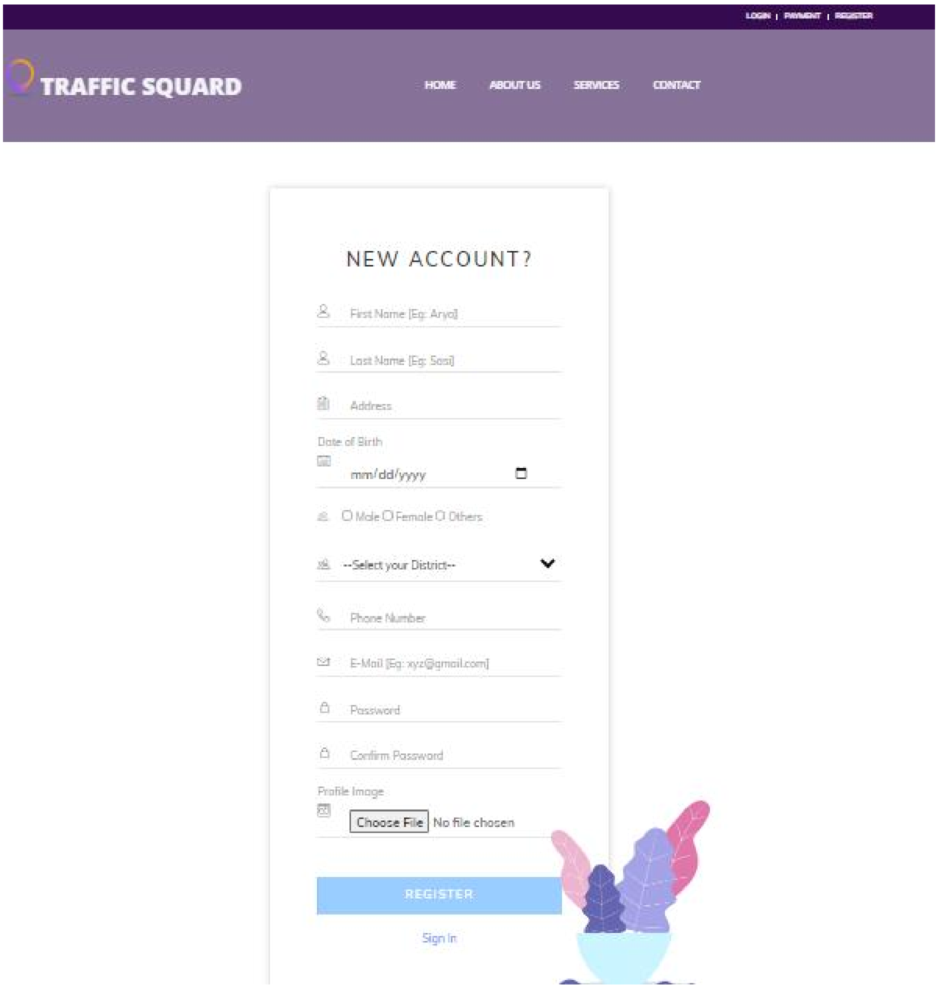
NAME :	ANKITHA SATHYAN
ADDRESS :	ANKITHBHAVAN
NUMBER :	9087652154
GMAIL :	ankithas@gmail.com

4.3.2 OUTPUT DESIGN

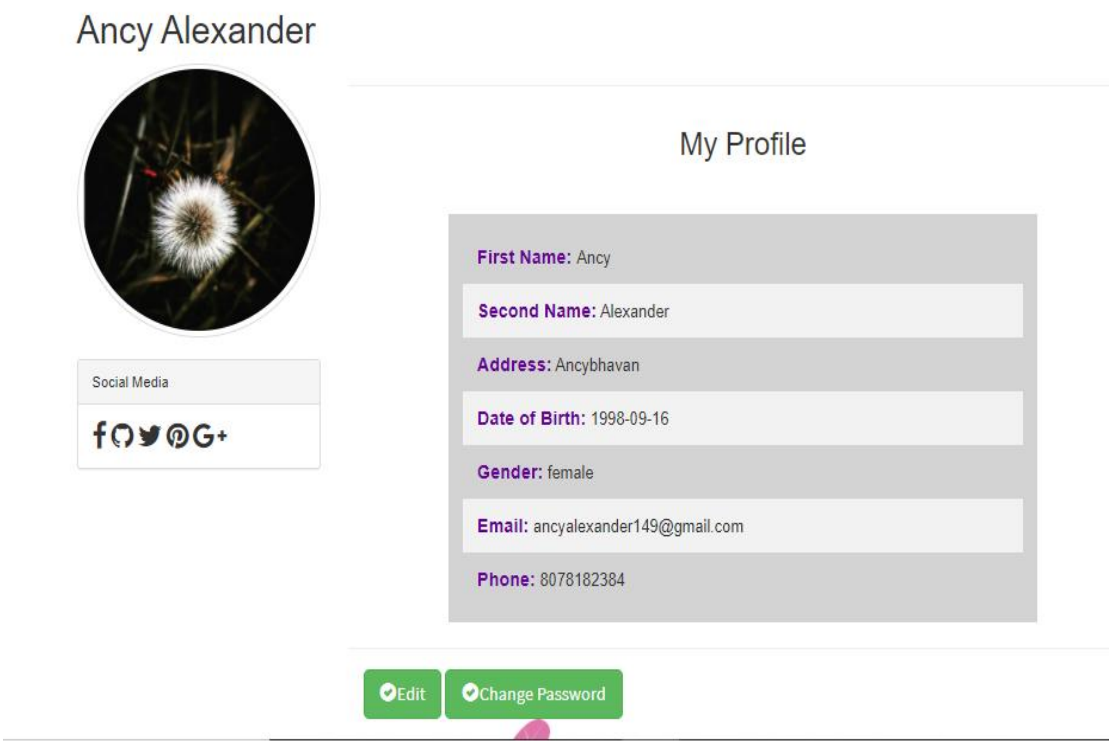
User Login



User Registration



User Profile



Ancy Alexander

Social Media

My Profile

First Name: Ancy

Second Name: Alexander

Address: Ancybhavan

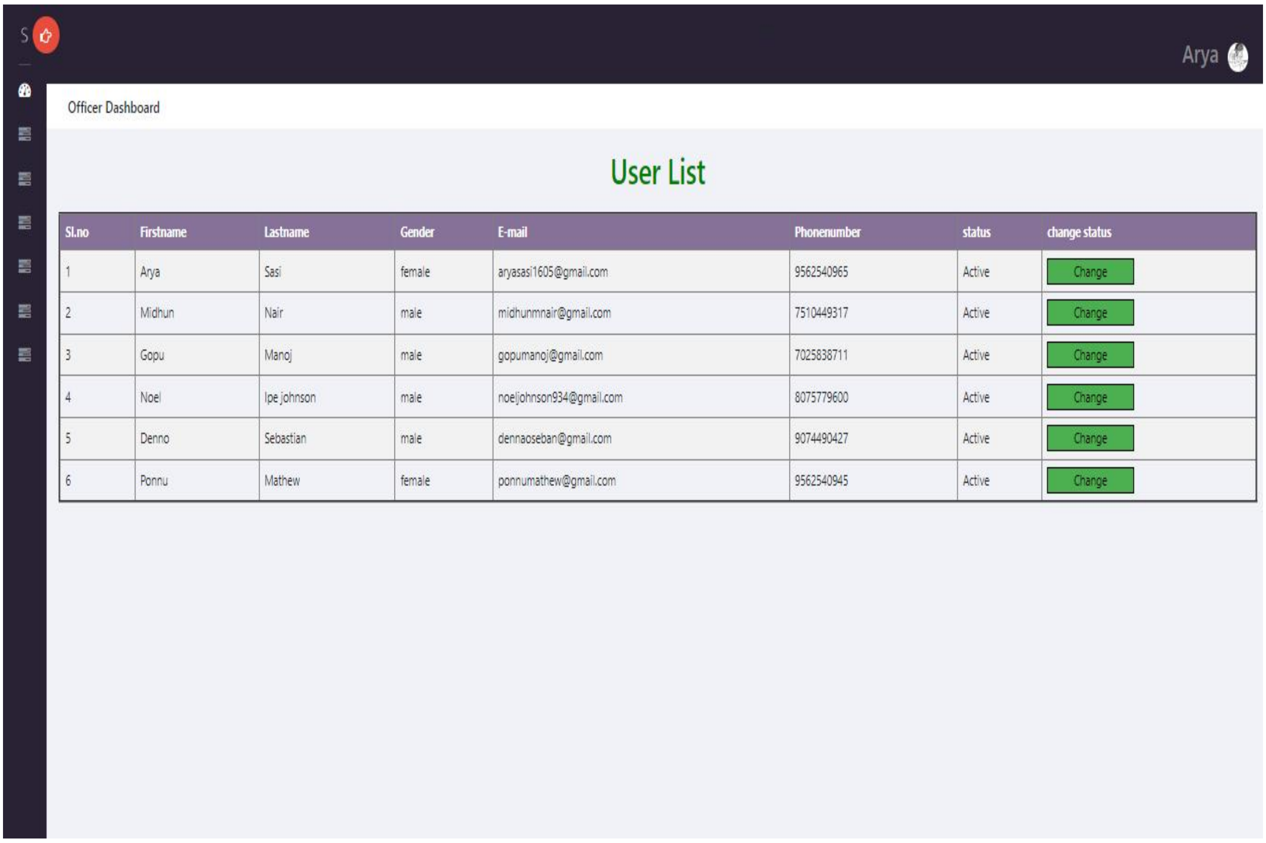
Date of Birth: 1998-09-16

Gender: female

Email: ancyalexander149@gmail.com

Phone: 8078182384

Officer Userlist



User List

Sl.no	Firstname	Lastname	Gender	E-mail	Phonenumber	status	change status
1	Arya	Sasi	female	aryasasi1605@gmail.com	9562540965	Active	Change
2	Midhun	Nair	male	midhunmnair@gmail.com	7510449317	Active	Change
3	Gopu	Manoj	male	gopumanoj@gmail.com	7025838711	Active	Change
4	Noel	Ipe johnson	male	noeljohnson934@gmail.com	8075779600	Active	Change
5	Denno	Sebastian	male	dennaoseban@gmail.com	9074490427	Active	Change
6	Ponnu	Mathew	female	ponnumathew@gmail.com	9562540945	Active	Change

4.4 DATABASE DESIGN

A database is a structured system with the capacity to store information and allows users to retrieve stored information quickly and effectively. Any database's primary goal is its data, which demands protection.

There are two stages to the database design process. The user needs are obtained in the first step, and a database is created to as clearly as possible meet these criteria. This process, known as information level design, is carried out independently of all DBMS.The design for the specific DBMS that will be used to construct the system in issue is converted from an information level design to a design in the second stage. Physical Level Design is the stage where the characteristics of the particular DBMS that will be used are discussed. Parallel to the system design is a database design. The database's data arrangement aims to accomplish the two main goals listed below.

- Data Integrity
- Data independence

4.4.1 Relational Database Management System (RDBMS)

In a relational model, the database is shown as a set of relations. Each relation resembles a file or table of records with values. A row is referred to as a tuple, a column heading is referred to as an attribute, and the table is referred to as a relation in formal relational model language. A relational database is made up of a number of tables, each with its own name. In a story, each row represents a group of associated values.Relations, Domains & Attributes

A relation is a table. Tuples are the units of a table's rows. An ordered group of n elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A group of atomic values make up a domain D. Choosing a data type from which the domain's data values are derived is a typical way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name.Each value in a relation is atomic and cannot be broken down.

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 are Super Key and Candidate Keys.

4.4.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. The formal process of normalising data structures in a way that reduces duplication and fosters integrity. Using the normalisation technique, superfluous fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. A row in a table is uniquely identified by a key. Primary keys and foreign keys are the two different kinds of keys.A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalised.

It means placing things in their natural form, as the name suggests. By using normalisation, the application developer aims to establish a coherent arrangement of the data into appropriate tables and columns, where names may be quickly related to the data by the user. By removing recurring groups from the data, normalisation prevents data redundancy, which puts a heavy strain on the computer's resources. These consist of:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

According to the First Normal Form, each attribute's domain must only include atomic values, and each attribute's value in a tuple must be a single value from that domain. In other words, 1NF forbids using relationships as attribute values within tuples or relations within relations. Single atomic or indivisible values are the only attribute values that are permitted under 1NF. The data must first be entered into First Normal Form. This can be accomplished by separating data into tables of a 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 non-atomic 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

No non-key attribute should be functionally dependent on a portion of the main key for relations where the primary key has several attributes, according to Second Normal Form. This involves breaking down each partial key into its dependent characteristics and setting up a new relation for each one. Keep the original primary key and any properties that are entirely dependent on it in your database. This procedure aids in removing data that depends only on a small portion of the key. If and only if a relation satisfies all the requirements for first normal form for the primary key and all of the non-primary key qualities of the relation are completely dependent on the primary key alone, then that relation is said to be in second normal form.

Third Normal Form

Relation should not have a non-key attribute that is functionally determined by another non-key attribute or by a collection of non-key attributes, according to the Third Normal Form. The primary key should not be transitively dependent, in other words. The non-key attributes that functionally determine other non-key attributes are decomposed in this way put up in relation. This procedure is used to eliminate anything not wholly dependent on the Primary Key. Only when a relation is in second normal form and, more importantly, when its non-key characteristics do not depend on those of other non-key attributes, is it considered to be in third normal form..

TABLE DESIGN

- **tbl_login**

Primary key:Login_id

Role references role_id in tbl_Role.

R_id references r_id in tbl_Registeration.

Field	Datatype	Constraints	Description
Login_id	int	Primary key	Login identity Number
Username	varchar(50)	NOT NULL	Username for Login
Password	varchar(50)	NOT NULL	Password for Login
Role	int	Foreign key	Role of User
Status	bit	NOT NULL	Status of User
R_id	int	Foreign Key	Registeration Id

- **tbl_Officer**

Primary key:Officer_id

R_id references r_id in tbl_Registeration.

District_id references District_id in tbl_District.

Field	Datatype	Constraints	Description
Officer_id	int	Primary key	Officer Unique identity
fname	Varchar(100)	NOT NULL	Officer Firstname
lname	Varchar(100	NOT NULL	Officer Lastname
address	Varchar(100)	NOT NULL	Officer Address
dob	Date	NOT NULL	Officer Date of Birth
email	Varchar(50)	NOT NULL	Email of Officer
District_id	int	Foreign key	District
R_id	int	Foreign key	Register Id
Status	bit	NOT NULL	Status -active/blocked

- **tbl_Offense**

Primary key:Offense_id

Field	Datatype	Constraints	Description
Offense_id	int	Primary key	Offence unique identity number
Offense	Varchar(50)	NOT NULL	Offence
description	Varchar(100)	NOT NULL	Description of Offence
amount	int	NOT NULL	Amount to be paid

- **tbl_Offense_report**

Primary key:Offensereport_id

User_id references User_id in tbl_User.

Offense_id references Offense_id in tbl_Offense.

District_id references District_id in tbl_District.

Field	Datatype	Constraints	Description
Offensereport_id	Int	Primary key	Offence reporting unique identity number
User_id	int	Foreign key	Reported user id
Vehicle_no	int	NOT NULL	Vehicle number that comit offence
Reported date	Date	NOT NULL	Reported date
Reporting date	Date	NOT NULL	Reporting time and date
Offence_id	int	Foregin key	Offence id from tbl_offence
district_id	int	Foreign key	District where offence happens
Place	Varchar(50)	NOT NULL	Place in the district
image	Varchar(100)	NOT NULL	Evdience of offence comitted
Status	bit	NOT NULL	Status-approved/rejected
Payment	Varchar(10)	NOT NULL	Payment-unpaid/paid

- **tbl_Complaints**

Primary key:Offensereport_id

R_id references r_id in tbl_Registeration.

District_id references District_id in tbl_District.

Field	Datatype	Constraints	Description
complaintid	int	Primary key	Complaint id
R_id	Int	Foreign Key	Registeration id of user
complaint	Varchar(100)	NOT NULL	Complaint/meaasage
Date	date	NOT NULL	Date of complaint registration
District_id	Int	Foreign Key	District where complaint registered
Place	Varchar(20)	NOT NULL	Location in the district where the problem is
C_status	Varchar(20)	NOT NULL	Status-approved/processed/rejected

- **tbl_Feedback**

Primary key: Feedback_id

R_id references r_id in tbl_Registeration.

Field	Datatype	Constraints	Description
Feedback_id	int	Primary key	Feedback unique identity number
R_id	int	Foreign key	Registeration id of user who gives the feedback
F_date	Date	NOT NULL	Date of feedback updated
feedback	Varchar(50)	NOT NULL	Message in feedback
F_status	Int	NOT NULL	Status-readed/unreaded

- ## tbl_Registration

Primary key:R_id

Field	Datatype	Constraints	Description
R_id	int	Primary key	Registration id
firstname	Varchar(20)	NOT NULL	Firstname of user
lastname	Varchar(20)	NOT NULL	Lastname of user
Address	Varchar(30)	NOT NULL	Address
dob	date	NOT NULL	Date of birth
gender	Varchar(20)	NOT NULL	Gender
Gmail	Varchar(20)	NOT NULL	Email of the user
phno	Varchar(20)	NOT NULL	Mobile number
image	Varchar(200)	NOTNULL	Profile image
status	Int(10)	NOT NULL	Status-active/blocked

- ## tbl_Offender

Primary key:offender_id

Vehicle_id references vehicle number in tbl_drives

Field	Datatype	Constraints	Description
offender_id	int	Primary key	Offender identity number
Offender name	Varchar(10)	NOT NULL	Offender Name
vehicle_id	int	Foregin key	Vehcile number from tbl_drives

- **tbl_Role**

Primary key:role_id

Field	Datatype	Constraints	Description
role_id	int	Primary key	Role identity
role	varchar(	NOT NULL	Role description

- **tbl_Rules**

Primary key:Rule_id

Field	Datatype	Constraints	Description
Rule_id	int	Primary key	Rule identity number
Rule	varchar(20)	NOT NULL	Rule description

- **tbl_Driver**

Primary key:Driver_id

Field	Datatype	Constraints	Description
Driver_id	int	Primary key	Driver unique identity number
Drivername	varchar(100)	NOT NULL	Drivername/ownername
Gender	Varchar(10)	NOT NULL	Gender of driver
Vehicle_no	Int(20)	NOT NULL	Vehicle number
address	Varchar(50)	NOT NULL	Adress of vehicle registration
Phone Number	Varchar(10)	NOT NULL	Mobile number registered with vehicle
Gmail	Varchar(30)	NOT NULL	Email registered with vehicle
Vehicle Type	Varchar(30)	NOT NULL	Vehicle type-2 wheeler/4-wheeler/3-wheeler etc..
District	Varchar(30)	NOT NULL	District where vehicle is registered

- **tbl_Users**

Primary key:User_id

R_id references r_id in tbl_registeration.

Field	Datatype	Constraints	Description
Uer_id	int	Primary key	User unique identity
R_id	int	Foreign Key	User Register id
firstname	Varchar(20)	NOT NULL	User Firstname
lastname	Varchar(20)	NOT NULL	User Lastname
Address	Varchar(30)	NOT NULL	Address of user
dob	date	NOT NULL	Date of birth of user
gender	Varchar(20)	NOT NULL	gender
Gmail	Varchar(20)	NOT NULL	Email id that the user have
phno	Varchar(20)	NOT NULL	Mobile number
image	Varchar(200)	NOTNULL	Profile image of user
status	Int(10)	NOT NULL	Status-active/blocked

- **tbl_District**

Primary key:district_id

Field	Datatype	Constraints	Description
district_id	int	Primary key	District identity number
district	varchar(20)	NOT NULL	Name of district

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the process of carefully controlling the execution of software in order to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and walkthroughs as well. Verifying that what has been specified matches what the user truly desired is the process of validation.

The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code, searching for issues and obtaining statistics without actually running the code. Dynamic analysis examines how software behaves while it is running in order to offer data like execution traces, timing profiles, and test coverage details.

Testing is a collection of activities that can be planned ahead of time and carried out in a methodical manner. Testing starts with individual modules and progresses to the integration of the full computer-based system. There are many rules that can be used as testing objectives, and testing is necessary for the system testing objectives to be successful. As follows:

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 test is successfully carried out in accordance with the aforementioned aims, it will reveal software bugs. Additionally, testing shows that the software functions seem to operate in accordance with the specification and that the performance requirements seem to have been satisfied.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are 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.

5.2 TEST PLAN

A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer programme, its documentation, and associated data structures are all created by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfils the purpose for which it was intended. In order to solve the inherent issues with allowing the builder evaluate what they have developed, there is an independent test group (ITG). Testing's precise goals should be laid forth in quantifiable language.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

5.2.1 Unit Testing

Unit testing concentrates verification efforts on the software component or module, which is the smallest unit of software design. The component level design description is used as a guide when testing crucial control paths to find faults inside the module's perimeter. the level of test complexity and the untested area determined for unit testing. Unit testing is white-box focused, and numerous components may be tested simultaneously. To guarantee that data enters and exits the software unit under test properly, the modular interface is tested. To make sure that data temporarily stored retains its integrity during each step of an algorithm's execution, the local data structure is inspected. 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.

Before starting any other test, tests of data flow over a module interface are necessary. All other tests are irrelevant if data cannot enter and depart the system properly. An important duty during the unit test is the selective examination of execution pathways. Error circumstances must be foreseen in good design, and error handling paths must be put up to cleanly reroute or halt work when an error does arise. The final step of unit testing is boundary testing. Software frequently fails at its limits.

In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. All unused code was eliminated, and it was confirmed that every module was functional and produced the desired outcome.

5.2.2 Integration Testing

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a programme structure that has been determined by design using unit tested components. The programme as a whole is tested. Correction is challenging since the size of the overall programme makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All of the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive programme structure also developed when discrepancies in programme structures were eliminated.

5.2.3 Validation Testing or System Testing

The testing process comes to an end here. This involved testing the entire system in its entirety, including all forms, code, modules, and class modules. Popular names for this type of testing include system tests and black box testing.

The functional requirements of the software are the main emphasis of the black box testing approach. That example, using Black Box testing, a software engineer can create sets of input conditions that will fully test every programme requirement.

The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors.

5.2.4 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.

CHAPTER 6
IMPLEMENTATION

6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. It can be regarded as the most important stage in creating a successful new system since it gives users assurance that the system will operate as intended and be reliable and accurate. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation. The user department now bears the most of the workload, faces the most disruption, and has the biggest influence on the current system.If the implementation is not carefully planned or controlled, it can create chaos and confusion.Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system that satisfies organisational needs must be implemented properly. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards can it be put into use. 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 change over.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. People have early reservations about the software, but we must watch out that they do not become more resistant by making 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.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. The people who will be involved must have faith in their ability to contribute to the goal and benefits anticipated from the computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports, and execute other important tasks through user training.

6.2.2 Training on the Application Software

The user will need to receive the essential basic training on computer awareness after which the new application software will need to be taught to them. This will explain the fundamental principles of how to use the new system, including how the screens work, what kind of help is displayed on them, what kinds of errors are made while entering data, how each entry is validated, and how to change the date that was entered. Then, while imparting the program's training on the application, it should cover the information required by the particular user or group to operate the system or a certain component of the system.

6.2.3 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".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for users to report offense online and view all information related to traffic.The person who comit offence can view their offence details and they also can pay the amount through online by using this system.The system also provide all relevent details that are related to traffic such as the rules,driving tips,offence details.

7.2 FUTURE SCOPE

For Future,We can Improve the Quality of this website by Adding API locating and rating of users who report offence that are approved by officers and also we can add a new fuction to users those who report fake offence comitment having punished.It may help to improve the correctness of reporting and the users must be aware about the punishment and they will not repeat fake offence reporting then.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- PankajJalote, "So*ftware 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
- www.jquery.com
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html
- www.bootstrap.com

CHAPTER 9

APPENDIX

9.1 Sample Code

Login

```
<html lang="en">
<head>
<title>Traffic squard</title>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="description" content="Travelix Project">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="styles/bootstrap4/bootstrap.min.css">
<link href="plugins/font-awesome-4.7.0/css/font-awesome.min.css" rel="stylesheet"
type="text/css">
<link rel="stylesheet" type="text/css" href="plugins/OwlCarousel2-2.2.1/owl.carousel.css">
<link rel="stylesheet" type="text/css" href="plugins/OwlCarousel2-2.2.1/owl.theme.default.css">
<link rel="stylesheet" type="text/css" href="plugins/OwlCarousel2-2.2.1/animate.css">
<link rel="stylesheet" type="text/css" href="styles/main_styles.css">
<link rel="stylesheet" type="text/css" href="styles/responsive.css">
</head><body>
<div class="super_container">
<!-- Header --><header class="header">
<!-- Top Bar -->
<div class="top_bar"><div class="container"><div class="row">
<div class="col d-flex flex-row">
<div class="user_box ml-auto">
<div class="user_box_login user_box_link"><a href="login.php">login</a></div>
<div class="user_box_register user_box_link"><a href="register.php">register</a></div>
</div></div></div></div></div>
<!-- Main Navigation -->
<nav class="main_nav">
<div class="container">
<div class="row">
```

```
<div class="col main_nav_col d-flex flex-row align-items-center justify-content-start" style="
margin-top: -2px;">
<div class="logo_container">
<div class="logo"><a href="#"><img src="images/logo.png" alt="">traffic squard</a></div>
</div><div class="main_nav_container ml-auto">
<ul class="main_nav_list">
<li class="main_nav_item"><a href="index.html">home</a></li>
<li class="main_nav_item"><a href="about.html">about us</a></li>
<li class="main_nav_item"><a href="services.html">Services</a></li>
<li class="main_nav_item"><a href="contact.html">contact</a></li>
</ul></div>
<div class="content_search ml-lg-0 ml-auto">
<svg version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
width="17px" height="17px" viewBox="0 0 512 512" enable-background="new 0 0 512 512"
xml:space="preserve">
<g><g><g>
<path class="mag_glass" fill="#FFFFFF"
d="M78.438,216.78c0,57.906,22.55,112.343,63.493,153.287c40.945,40.944,95.383,63.494,153.28
7,63.494
s112.344-22.55,153.287-63.494C489.451,329.123,512,274.686,512,216.78c0-57.904-22.549-
112.342-63.494-153.286C407.563,22.549,353.124,0,295.219,0c-57.904,0-112.342,22.549-
153.287,63.494C100.988,104.438,78.439,158.876,78.438,216.78zM119.804,216.78c0-
96.725,78.69-175.416,175.415-175.416s175.418,78.691,175.418,175.416c0,96.725-
78.691,175.416-
175.416,175.416C198.495,392.195,119.804,313.505,119.804,216.78z"</g></g><g><g>
<path class="mag_glass" fill="#FFFFFF"
d="M6.057,505.942c4.038,4.039,9.332,6.058,14.625,6.058s10.587-2.019,14.625-
6.058L171.268,369.98c8.076-8.076,8.076-21.172,0-29.248c-8.076-8.078-21.172-8.078-
29.249,0L6.057,476.693C-2.019,484.77-2.019,497.865,6.057,505.942z"/>
</g></g></g></svg>
</div>
```

```
<form id="search_form" class="search_form bez_1">
<input type="search" class="search_content_input bez_1">
</form><div class="hamburger">
<i class="fa fa-bars trans_200"></i></div></div></div></div></nav></header>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="icon" type="image/png" href="images/icons/favicon.ico"/>
        <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" type="text/css" href="fonts/iconic/css/material-design-iconic-font.min.css">
        <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">
        <link rel="stylesheet" type="text/css" href="vendor/css-hamburgers/hamburgers.min.css">
        <link rel="stylesheet" type="text/css" href="vendor/animsition/css/animsition.min.css">
        <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">
        <link rel="stylesheet" type="text/css" href="vendor/daterangepicker/daterangepicker.css">
        <link rel="stylesheet" type="text/css" href="css/util.css">
        <link rel="stylesheet" type="text/css" href="css/main.css">
</head><style>
input[type=text] {
   padding: 12px 20px;
   display: inline-block;
   border: 1px solid #ccc;
   border-radius: 4px;
   box-sizing: border-box;}
button{
  background-color: #4CAF50;
   border: none;
   color: white;
   padding: 12px 30px;
   text-decoration: none;
   margin: 4px 2px;
   cursor: pointer;
}
```

```
canvas{
  /*prevent interaction with the canvas*/
  pointer-events:none;
}

</style><div class="container-login100" style="background-image: url('images/c4.jpg');">
          <div class="wrap-login100 p-l-55 p-r-55 p-t-80 p-b-30" style="margin-top: 189px;">
<form class="login100-form validate-form" action="login_action.php" method="post">
<span class="login100-form-title p-b-37">Sign In</span>
<div id="validation-message" style="color:red;text-align:center;">
<?php if(isset($_GET['error']))
echo $_GET['error'];
                    ?><hr></div>
<div class="wrap-input100 validate-input m-b-20" data-validate="Enter username or email">
<input class="input100" type="text" name="username" placeholder="username or email">
<span class="focus-input100"></span></div>
<div class="wrap-input100 validate-input m-b-25" data-validate = "Enter password">
<input class="input100" type="password" name="pass" placeholder="password">
<span class="focus-input100"></span></div>
<center><div class="wrap-input100 validate-input m-b-25" data-validate = "Forget password">
<a href="in.php"><font color="purple">Forgot Password??</font></a>
</div><div class="container-login100-form-btn">
<button type="submit" class="login100-form-btn">Sign In</button></div>
<br><div class="text-center">Don't have an account??
<a href="register.php" class="txt2 hov1">
<font color="red">Sign Up</font></a></div></form></div></div>
                    <?php
if (@$_GET['registered'] == 'true')
   echo ("<SCRIPT LANGUAGE='JavaScript'>
    window.alert('You are Succesfully Registered')</script>");
?>
</body></html>
```

```
        <div id="dropDownSelect1"></div>
        <script src="vendor/jquery/jquery-3.2.1.min.js"></script>
        <script src="vendor/animsition/js/animsition.min.js"></script>
        <script src="vendor/bootstrap/js/popper.js"></script>
        <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
        <script src="vendor/select2/select2.min.js"></script>
        <script src="vendor/daterangepicker/moment.min.js"></script>
        <script src="vendor/daterangepicker/daterangepicker.js"></script>
        <script src="vendor/countdowntime/countdowntime.js"></script>
        <script src="js/main.js"></script>
</body></html></body>
<div class="copyright"><div class="container"><div class="row">
<div class="col-lg-3 order-lg-1 order-2  ">
<div class="copyright_content d-flex flex-row align-items-center">
<div><!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->
Copyright &copy;<script>document.write(new Date().getFullYear());</script> All rights reserved
<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. --></div>
</div></div></div></div></div></div>
<script src="js/jquery-3.2.1.min.js"></script>
<script src="styles/bootstrap4/popper.js"></script>
<script src="styles/bootstrap4/bootstrap.min.js"></script>
<script src="plugins/OwlCarousel2-2.2.1/owl.carousel.js"></script>
<script src="plugins/easing/easing.js"></script>
<script src="js/custom.js"></script>
</body>
</html>
```

Register

```
<?php include "dbconnect.php";
$sq = mysqli_query($con,"SELECT * FROM `tbl_district`");
$user='1';
?>
<html lang="en">
<head>
<title>Traffic Squard</title>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="description" content="Travelix Project">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="styles/bootstrap4/bootstrap.min.css">
<link href="plugins/font-awesome-4.7.0/css/font-awesome.min.css" rel="stylesheet"
type="text/css">
<link rel="stylesheet" type="text/css" href="plugins/OwlCarousel2-2.2.1/owl.carousel.css">
<link rel="stylesheet" type="text/css" href="plugins/OwlCarousel2-2.2.1/owl.theme.default.css">
<link rel="stylesheet" type="text/css" href="plugins/OwlCarousel2-2.2.1/animate.css">
<link rel="stylesheet" type="text/css" href="styles/main_styles.css">
<link rel="stylesheet" type="text/css" href="styles/responsive.css">
</head>
<body>
<div class="super_container">
	<!-- Header -->
	<header class="header">
		<!-- Top Bar -->
<div class="top_bar"><div class="container"><div class="row">
<div class="col d-flex flex-row"><div class="user_box ml-auto">
<div class="user_box_login user_box_link"><a href="login.php">Login</a></div>
<div class="user_box_login user_box_link"><a href="offender_login.php">Payment</a></div>
<div class="user_box_register user_box_link"><a href="register.php">Register</a></div>
</div></div></div></div></div>
```

```
        <!-- Main Navigation -->
        <nav class="main_nav"><div class="container" ><div class="row" >
<div class="col main_nav_col d-flex flex-row align-items-center justify-content-start">
<div class="logo_container">
<div class="logo"><a href="#"><img src="images/logo.png" alt="">traffic squard</a></div>
</div>
<div class="main_nav_container ml-auto" >
<ul class="main_nav_list">
<li class="main_nav_item"><a href="index.html">home</a></li>
<li class="main_nav_item"><a href="about.html">about us</a></li>
<li class="main_nav_item"><a href="services.html">Services</a></li>
<li class="main_nav_item"><a href="contact.html">contact</a></li>
</ul></div></div></div></div></nav>
    </header>
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <!-- LINEARICONS -->
        <link rel="stylesheet" href="fonts/linearicons/style.css">
        <!-- STYLE CSS -->
        <link rel="stylesheet" href="css/style.css"></head>
    <body><div class="wrapper"><div class="inner" >
    <img src="images/image-1.png" alt="" class="image-1"></img>
<form action="register_action.php?type=<?php echo $user;?>" method="post"
enctype="multipart/form-data" onsubmit="return Validate()" style="margin-top: 203px;">
    <h3>New Account?</h3>
            <?php
            if(isset($_GET['error']))
                echo $_GET['error'];
            ?>
<div class="form-holder"><span class="lnr lnr-user"></span>
<input type="text" name="nme" id="nme" class="form-control" title="First Name"
placeholder="First Name [Eg: Arya]" required onchange="Validate();">
</div><span id="msg1" style="color:red;"></span><script>
function Validate()
{      var val = document.getElementById('nme').value;
```

```
   if (!val.match(/^[A-Z][A-Za-z]{2,}$/))
   {    document.getElementById('msg1').innerHTML="Start with a Capital letter & Only alphabets
without space are allowed!!";
                              document.getElementById('nme').value = "";
  return false;      }
document.getElementById('msg1').innerHTML=" ";
   return true;   }
</script><div class="form-holder">
<span class="lnr lnr-user"></span><input type="text" name="lnme" id="lnme" class="form-
control" title="Last Name" placeholder="Last Name [Eg: Sasi]" required onchange="Validate1();">
</div><span id="msg2" style="color:red;"></span>
<script>
function Validate1()
{   var val = document.getElementById('lnme').value;
if (!val.match(/^[A-Z ][a-z]{1,}\s+[a-z]{0,}$/))
   {     document.getElementById('msg2').innerHTML="Start with a Capital letter.Only Aplhabets
are allowed";
 document.getElementById('lnme').value = "";
      return false;       }
document.getElementById('msg2').innerHTML=" ";
   return true;
}
</script><div class="form-holder"><span class="lnr lnr-apartment"></span>
<input type="text" name="add" id="add" class="form-control" title="Address"
placeholder="Address" required onchange="Validname();"></div>
<span id="msg3" style="color:red;"></span>
<script>
function Validname()
{
   var val = document.getElementById('add').value;

   if (!val.match(/^[A-Z][a-z][0-9]\s+{3,}$/))
   {
      document.getElementById('msg3').innerHTML="Start with a Capital letter";
```

```
                              document.getElementById('add').value = "";
       return false;
   }
document.getElementById('msg3').innerHTML=" ";
   return true;
}
</script><div class="form-holder"><span class="lnr lnr-calendar-full"></span>
<label for="image" style="text-align: left;">Date of Birth</label>
<input type="date" name="dob" id="myDate" class="form-control" min="1980-01-01"
max="2004-12-30" title="Date of birth" placeholder="Date of Birth" max="<?php echo date("Y-m-
d")?>" ></div>
<span id="demo" style="color:red;"></span><script>
function myFunction()
{         var x = document.getElementById("myDate").max;
document.getElementById("demo").innerHTML = "Invalid Date!!";
}
    </script>
<div ><div style="padding-top: 7px;padding-bottom: 23px;">
<span class="lnr lnr-users"></span>
&nbsp&nbsp&nbsp <input type="radio"  name="gender" value="male"> Male
<input type="radio" name="gender" value="female"> Female
<input type="radio" name="gender" value="others"> Others
</div></div>
                              <div class="form-holder"><span class="lnr lnr-users">
</span><select class="tkvsoft"  name="district" style="border-style: initial;padding-top: 18px;
padding-bottom: 10px;  margin-bottom: 4px;width: 100%;border-bottom: 1px solid #e6e6e6;"
required>
<option>&nbsp &nbsp &nbsp &nbsp --Select  your District--</option>
<?php while($row=mysqli_fetch_assoc($sq)){ ?>
<option value="<?php echo $row['d_name']; ?>"><?php echo $row['d_name']; ?></option>
                                    <?php } ?>
</select></div><div class="form-holder">span class="lnr lnr-phone-handset"></span>
<input type="text" name="phn"  maxlength="10" id="phn" class="form-control" title="Phone
Number" placeholder="Phone Number"  pattern="[0-9]{10}" required onchange="Validat();">
```

```
</div><span id="msg4" style="color:red;"></span>
<script>
function Validat()
{   var val = document.getElementById('phn').value;
    if (!val.match(/^[7-9][0-9]{1,9}$/))
    {  document.getElementById('msg4').innerHTML="Only Numbers are allowed and must contain
10 number"
  document.getElementById('phn').value = "";
        return false;
    }
document.getElementById('msg4').innerHTML=" ";
    return true;
}
</script><div class="form-holder"><span class="lnr lnr-envelope"></span>
<input type="email" name="email" id="email" title="E-Mail" class="form-control"
placeholder="E-Mail [Eg: xyz@gmail.com]" required onchange="return Validata();"></div>
<span id="msg5" style="color:red;"></span>
<script>
function Validata()
{
    var val = document.getElementById('email').value;

    if (!val.match(/([A-z0-9_\-\.]){1,}\@([A-z0-9_\-\.]){1,}\.([A-Za-z]){2,4}$/))
    {
        document.getElementById('msg5').innerHTML="Enter a Valid Email";
  document.getElementById('email').value = "";
        return false;
    }
document.getElementById('msg5').innerHTML=" ";
    return true;
}                    </script>
<div class="form-holder"><span class="lnr lnr-lock"></span>
<input type="password" name="pwd" id="pwd" class="form-control" title="Password"
placeholder="Password" required onchange="return Validp();"></div>
```

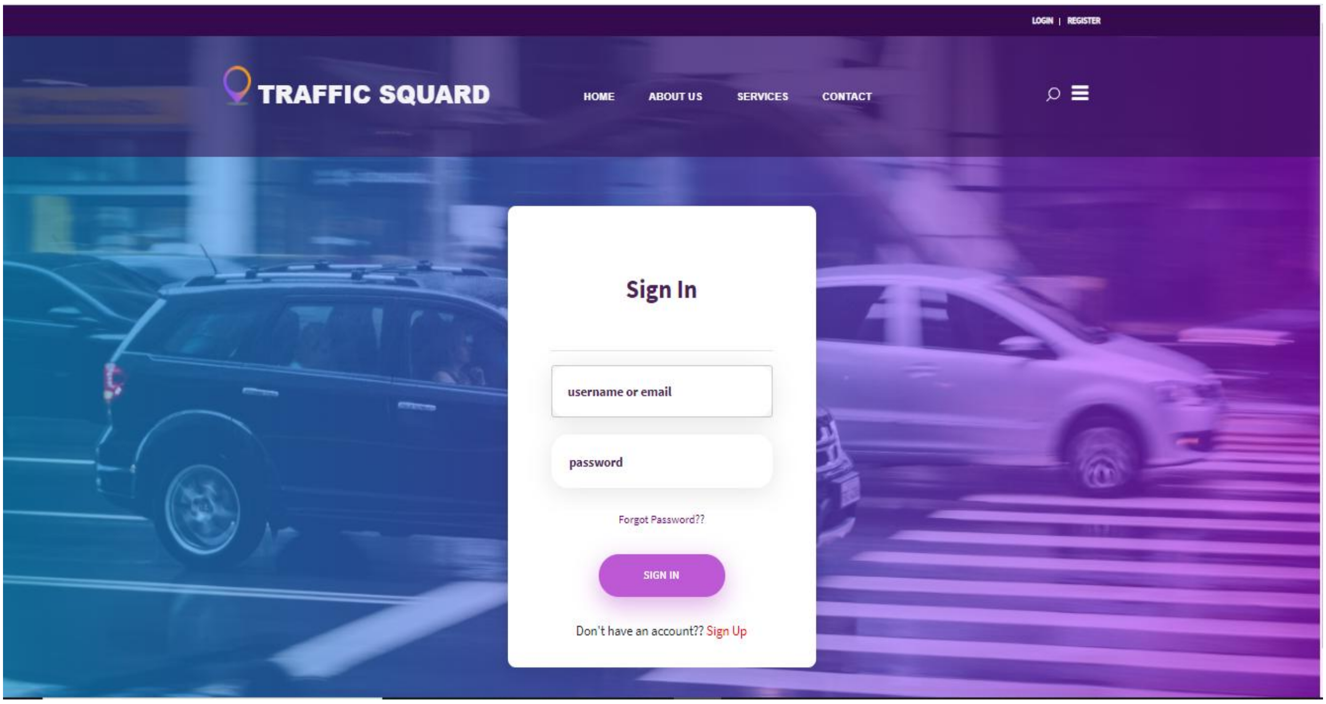
```
<span id="msg6" style="color:red;"></span>
<script>
function Validp()
{
   var val = document.getElementById('pwd').value;
   if (!val.match(/^[A-Za-z0-9!-*@]{5,15}$/))
   {   document.getElementById('msg6').innerHTML="Password should contain atleast 5 
characters";
document.getElementById('pwd').value = "";
       return false;     }
document.getElementById('msg6').innerHTML=" ";
   return true;  }
</script><div class="form-holder">  <span class="lnr lnr-lock"></span>
<input type="password" name="confirm" id="confirm" title="Confirm Password" class="form-
control" placeholder="Confirm Password" required onchange="return check();">
</div><span id="msg7" style="color:red;"></span><script>
         function check()
{   var pas1=document.getElementById("pwd");
  var pas2=document.getElementById("confirm");
          if(pas1.value=="")
         {   document.getElementById('msg7').innerHTML="Password can't be null!!";
                  pas1.focus();
                  return false;}
         if(pas2.value=="")
         {     document.getElementById('msg7').innerHTML="Please confirm password!!";
                  pass2.focus();
                  return false;   }
         if(pas1.value!=pas2.value)
         {     document.getElementById('msg7').innerHTML="Passwords does not match!!";
                  pas1.focus();
                  return false;     }
    document.getElementById('msg7').innerHTML=" ";
         return true;      }
         </script><div class="form-holder"><span class="lnr lnr-picture"></span>
```

```
        <label for="image" style="text-align: left;">Profile Image</label>
        <!-- upload upto 20 images!-->
<input type="file" name="img" id="img" class="form-control" accept="image/jpg,
image/JPG,image/JPEG, image/jpeg, image/png, image/PNG" required onchange="return
Validp();">
</div><script type="text/javascript">

function load_image(id,ext)
{   if(validateExtension(ext) == false)
  {   alert("Upload only JPEG or JPG or PNG format ");
    document.getElementById("img").value = "";
         document.getElementById("file").focus();
    return;  }}

function validateExtension(v)
{
    var allowedExtensions = new Array("jpg","JPG","jpeg","JPEG","png","PNG");
    for(var ct=0;ct<allowedExtensions.length;ct++)
    {  sample = v.lastIndexOf(allowedExtensions[ct]);
       if(sample != -1){return true;}   }
    return false;
}
</script><div>
<button type="submit" name="submit" value="Register">Register</button>
</div><br><center><a href="login.php">Sign In</a></form>
<img src="images/image-2.png" alt="" class="image-2"></img></div></div>
<script src="js/jquery-3.3.1.min.js"></script><script src="js/main.js"></script>
</body><!-- This templates was made by Colorlib (https://colorlib.com) -->
</html>
<div class="copyright"><div class="container"><div class="row">
<div class="col-lg-3 order-lg-1 order-2  ">
<div class="copyright_content d-flex flex-row align-items-center">
<div><!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->
Copyright &copy;<script>document.write(new Date().getFullYear());</script> All rights reserved
```

```
<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. --></div>
                                        </div></div></div></div></div></div>
<script src="js/jquery-3.2.1.min.js"></script>
<script src="styles/bootstrap4/popper.js"></script>
<script src="styles/bootstrap4/bootstrap.min.js"></script>
<script src="plugins/OwlCarousel2-2.2.1/owl.carousel.js"></script>
<script src="plugins/easing/easing.js"></script>
<script src="js/custom.js"></script><script>
$(document).ready(function()
{       $(".tkvsoft").on('change', function()
        {var t=this;
if($(t).val()==2)
                $("#tkvsoft").show();
        else
                $("#tkvsoft").hide();



        });


        });
        </script>
</body>
```

9.2 Screen Shots

Login



Sign In

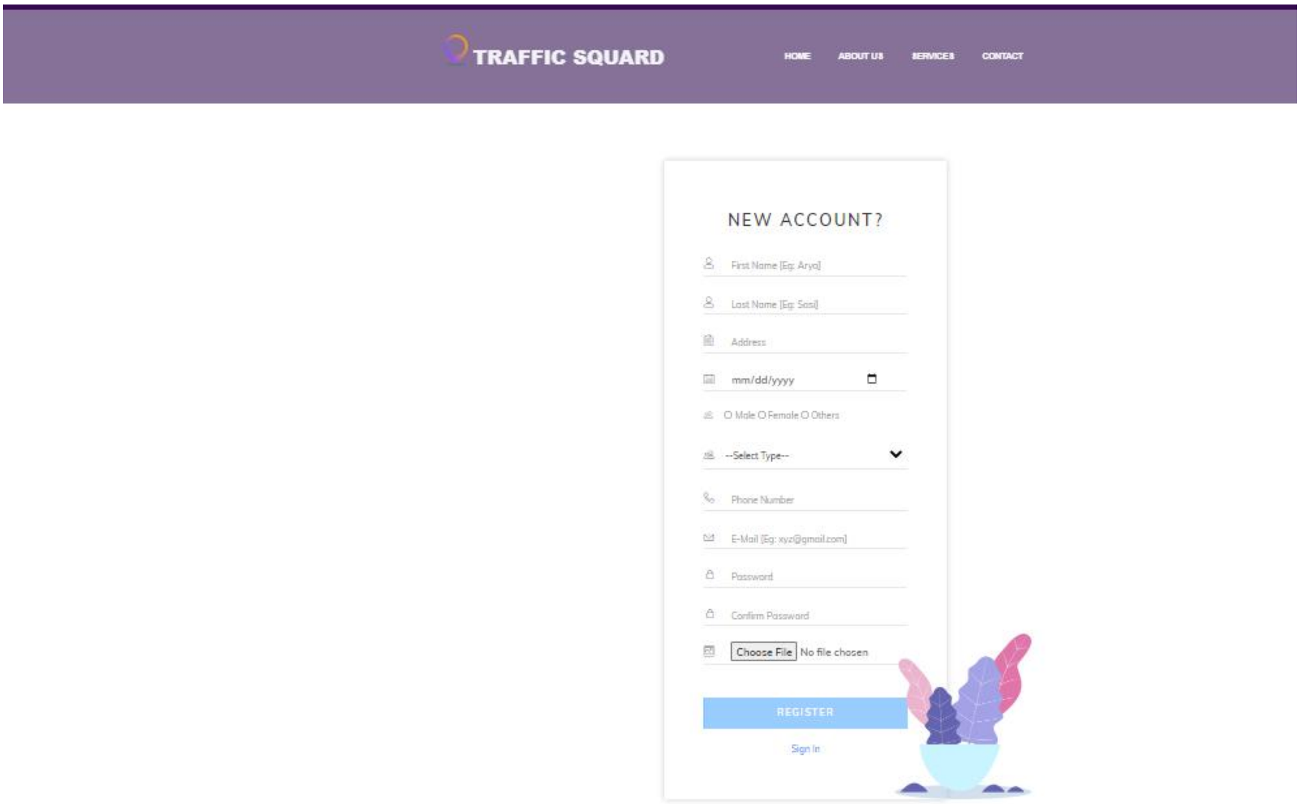
username or email

password

Forgot Password??

Don't have an account?? Sign Up

Registration



NEW ACCOUNT?

First Name [Eg: Arya]

Last Name [Eg: Sasi]

Address

mm/dd/yyyy

○ Male ○ Female ○ Others

--Select Type--

Phone Number

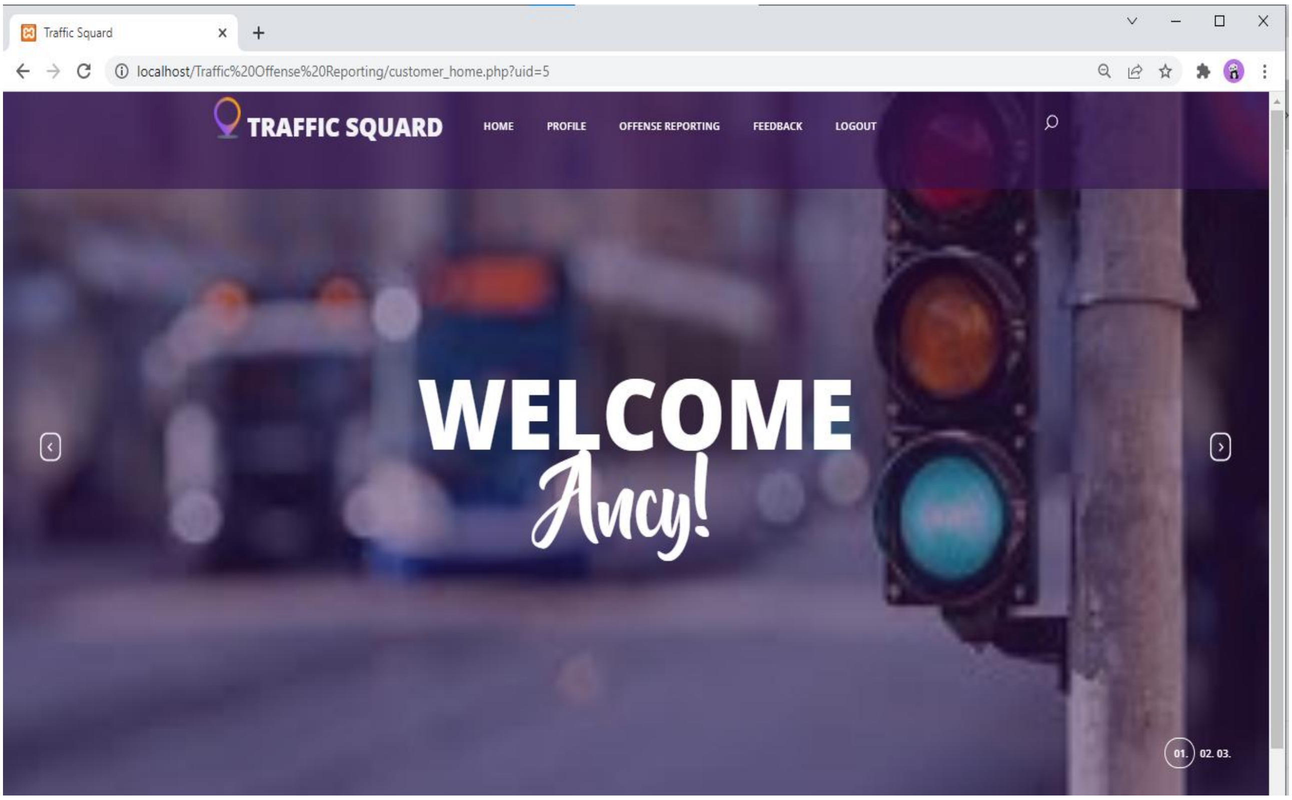
E-Mail [Eg: xyz@gmail.com]

Password

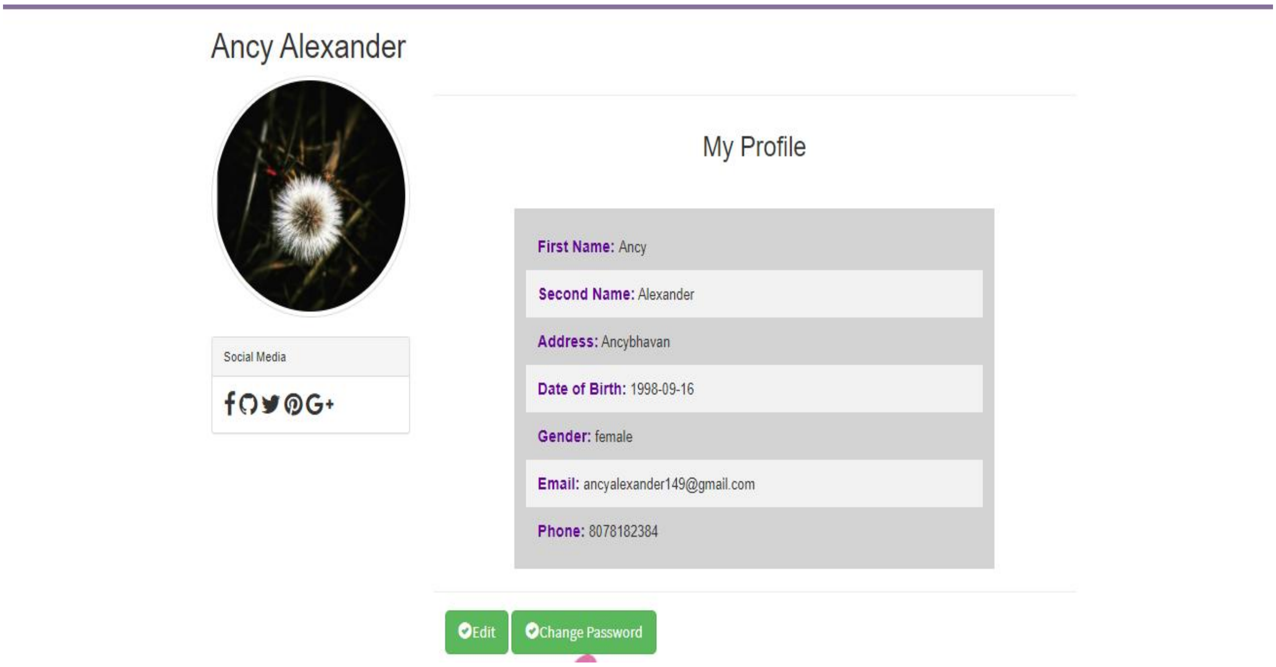
Confirm Password

No file chosen

User Homepage



User Profile



Ancy Alexander

My Profile

First Name: Ancy

Second Name: Alexander

Address: Ancybhavan

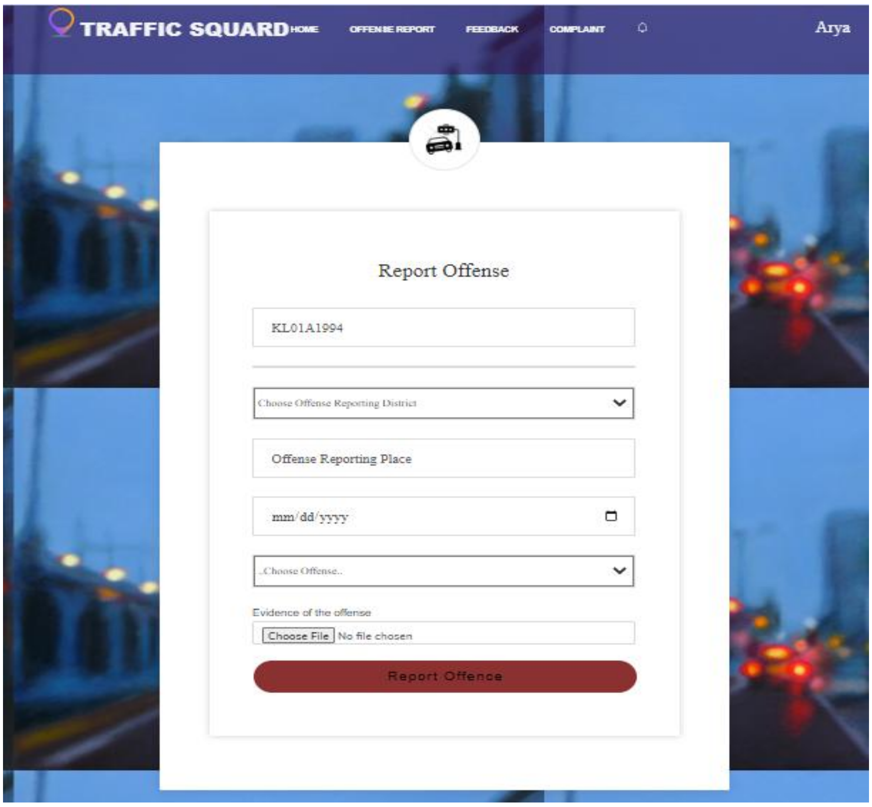
Date of Birth: 1998-09-16

Gender: female

Email: ancyalexander149@gmail.com

Phone: 8078182384

User Offence Reporting



Report Offense

KL01A1994

Choose Offense Reporting District

Offense Reporting Place

mm/dd/yyyy

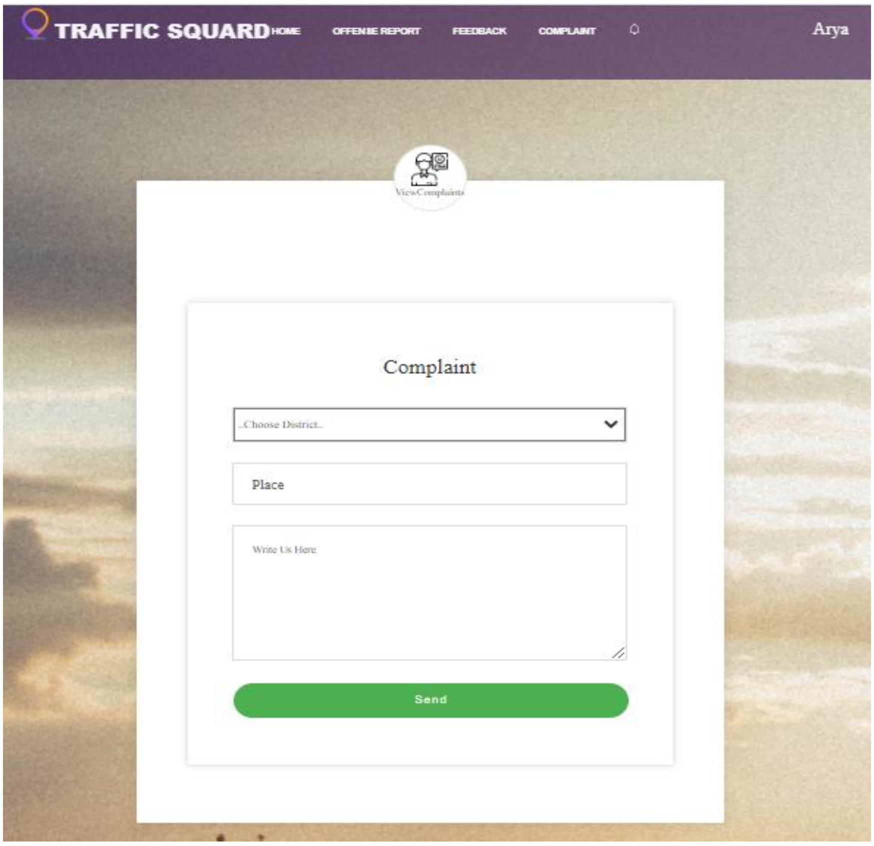
..Choose Offense..

Evidence of the offense

Choose File No file chosen

Report Offence

User-Complaint Form



Complaint

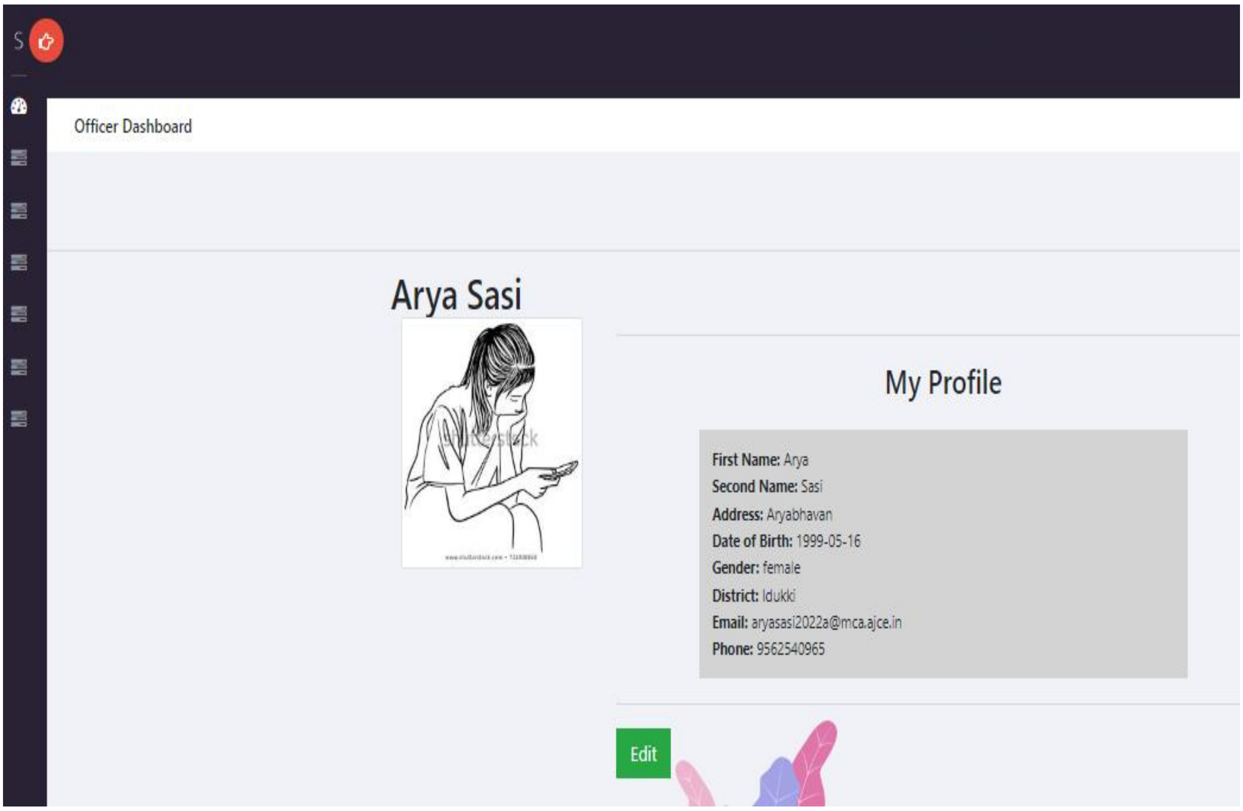
..Choose District..

Place

Write Us Here

Send

Officer Profile

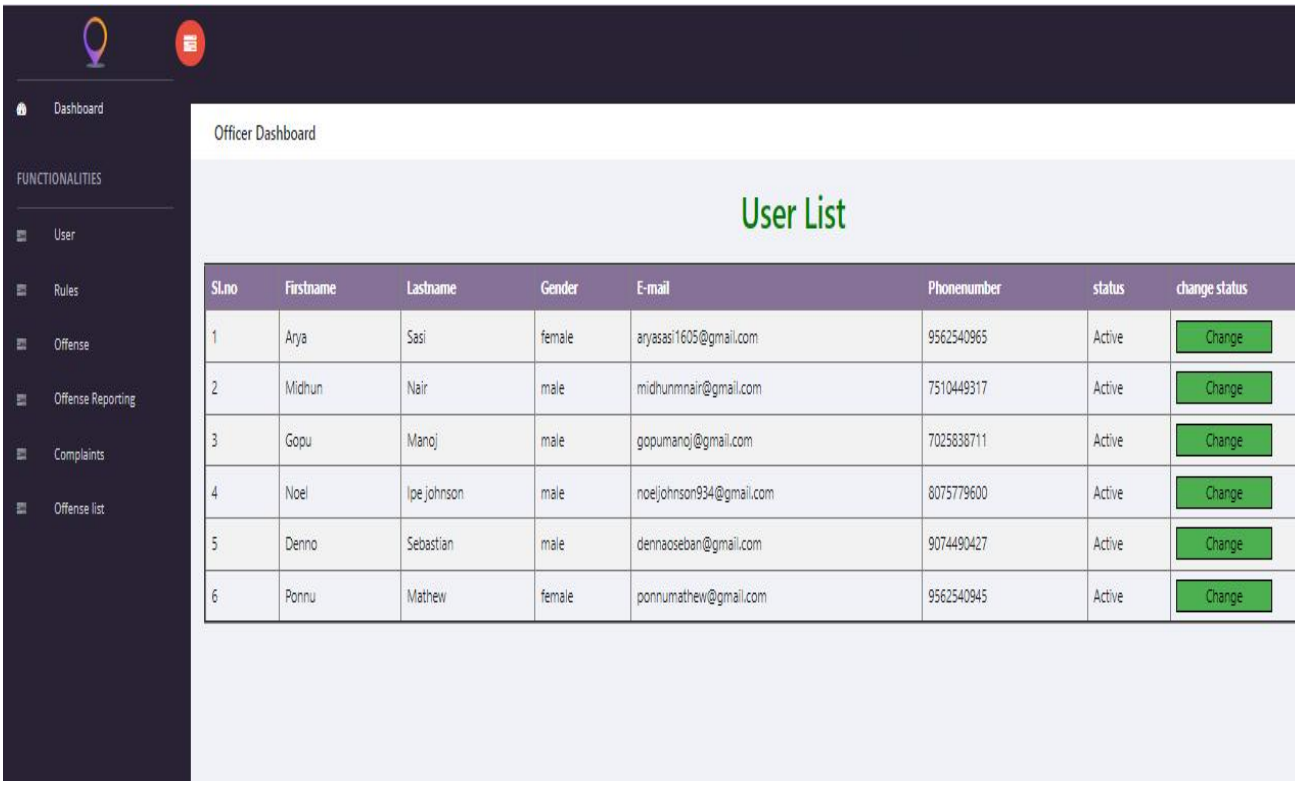


Arya Sasi

My Profile

First Name: Arya
Second Name: Sasi
Address: Aryabhavan
Date of Birth: 1999-05-16
Gender: female
District: Idukki
Email: aryasasi2022a@mca.ajce.in
Phone: 9562540965

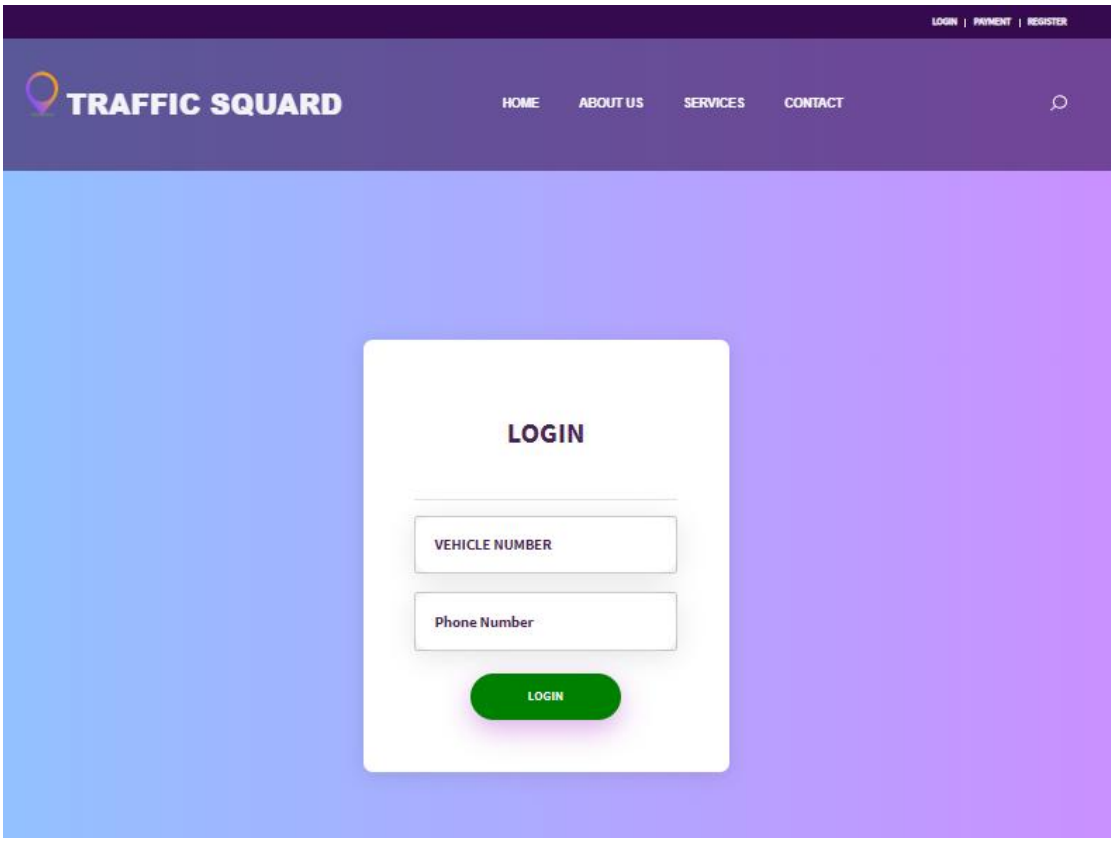
Officer Userlist



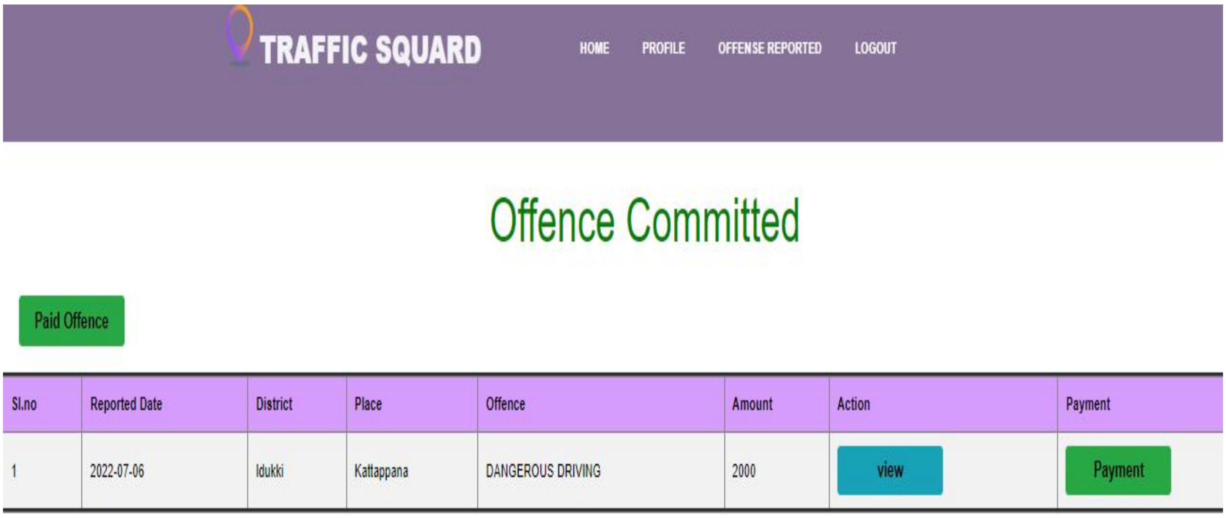
User List

Sl.no	Firstname	Lastname	Gender	E-mail	Phonenumber	status	change status
1	Arya	Sasi	female	aryasasi1605@gmail.com	9562540965	Active	Change
2	Midhun	Nair	male	midhunmnair@gmail.com	7510449317	Active	Change
3	Gopu	Manoj	male	gopumanoj@gmail.com	7025838711	Active	Change
4	Noel	Ipe johnson	male	noeljohnson934@gmail.com	8075779600	Active	Change
5	Denno	Sebastian	male	dennaoseban@gmail.com	9074490427	Active	Change
6	Ponnu	Mathew	female	ponnumathew@gmail.com	9562540945	Active	Change

Offender Login



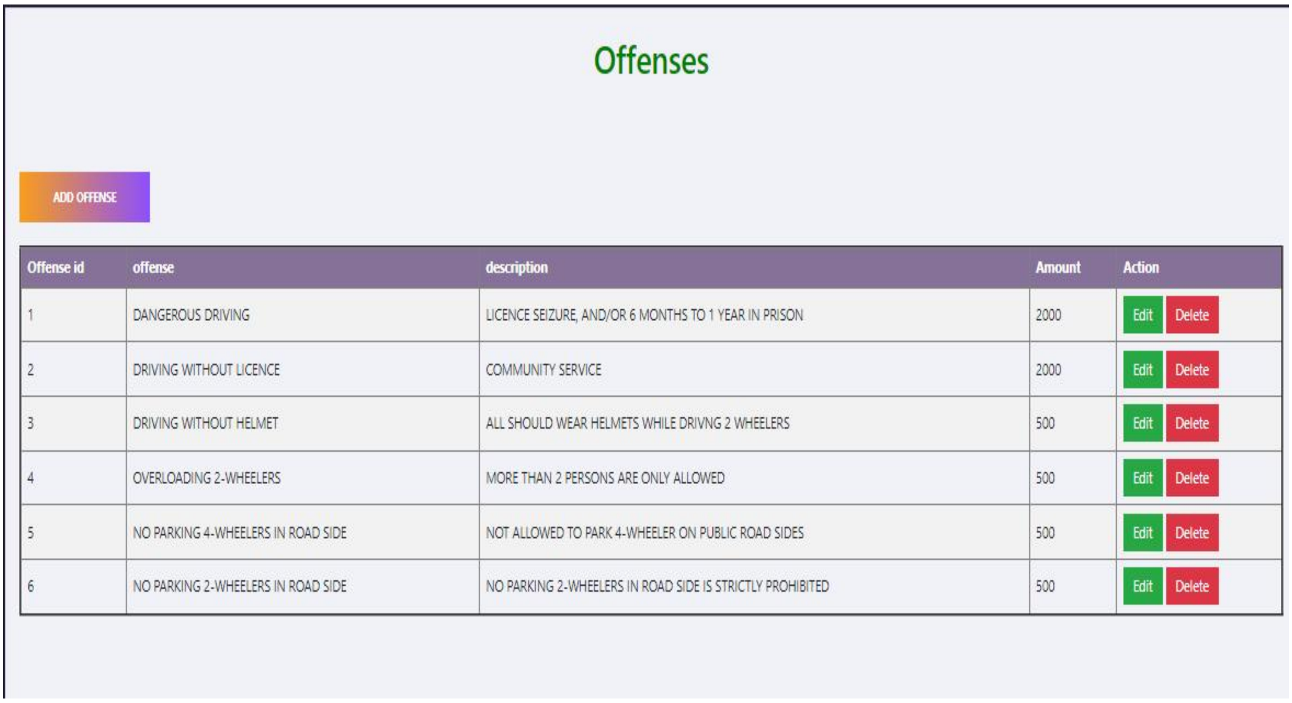
Offender-Offence committed list



Offence Committed

Sl.no	Reported Date	District	Place	Offence	Amount	Action	Payment
1	2022-07-06	Idukki	Kattappana	DANGEROUS DRIVING	2000	view	Payment

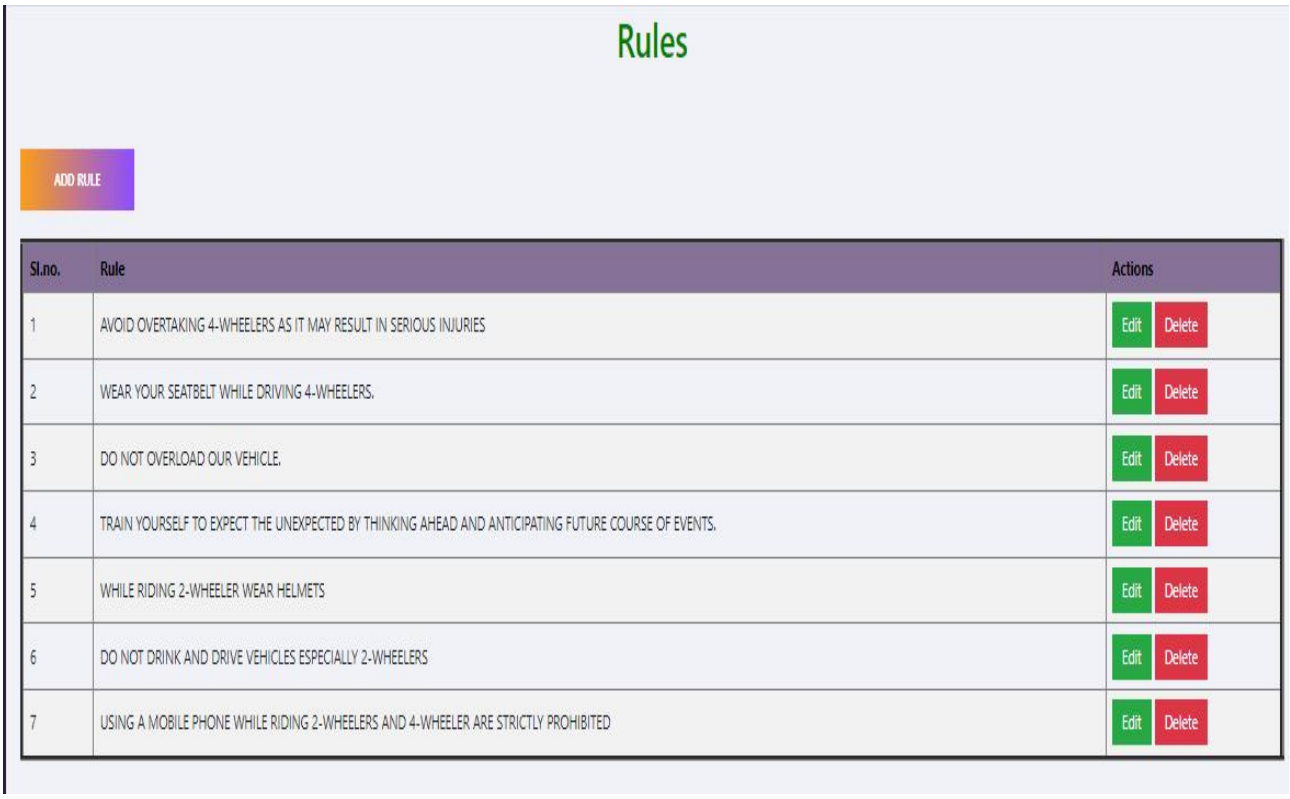
Admin -Offence



Offenses

Offense id	offense	description	Amount	Action
1	DANGEROUS DRIVING	LICENCE SEIZURE, AND/OR 6 MONTHS TO 1 YEAR IN PRISON	2000	Edit Delete
2	DRIVING WITHOUT LICENCE	COMMUNITY SERVICE	2000	Edit Delete
3	DRIVING WITHOUT HELMET	ALL SHOULD WEAR HELMETS WHILE DRIVNG 2 WHEELERS	500	Edit Delete
4	OVERLOADING 2-WHEELERS	MORE THAN 2 PERSONS ARE ONLY ALLOWED	500	Edit Delete
5	NO PARKING 4-WHEELERS IN ROAD SIDE	NOT ALLOWED TO PARK 4-WHEELER ON PUBLIC ROAD SIDES	500	Edit Delete
6	NO PARKING 2-WHEELERS IN ROAD SIDE	NO PARKING 2-WHEELERS IN ROAD SIDE IS STRICTLY PROHIBITED	500	Edit Delete

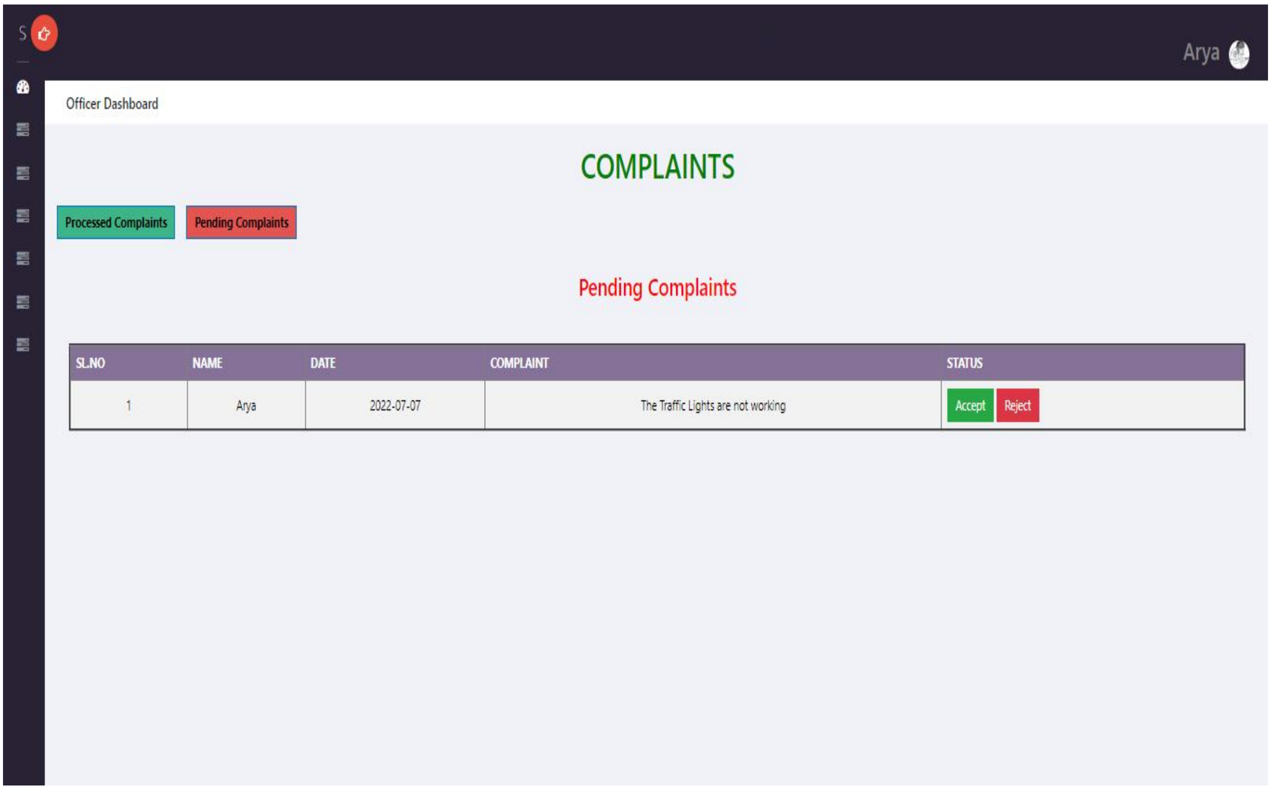
Admin -Rules



Rules

Sl.no.	Rule	Actions
1	AVOID OVERTAKING 4-WHEELERS AS IT MAY RESULT IN SERIOUS INJURIES	Edit Delete
2	WEAR YOUR SEATBELT WHILE DRIVING 4-WHEELERS.	Edit Delete
3	DO NOT OVERLOAD OUR VEHICLE.	Edit Delete
4	TRAIN YOURSELF TO EXPECT THE UNEXPECTED BY THINKING AHEAD AND ANTICIPATING FUTURE COURSE OF EVENTS.	Edit Delete
5	WHILE RIDING 2-WHEELER WEAR HELMETS	Edit Delete
6	DO NOT DRINK AND DRIVE VEHICLES ESPECIALLY 2-WHEELERS	Edit Delete
7	USING A MOBILE PHONE WHILE RIDING 2-WHEELERS AND 4-WHEELER ARE STRICTLY PROHIBITED	Edit Delete

Officer User Complaint View

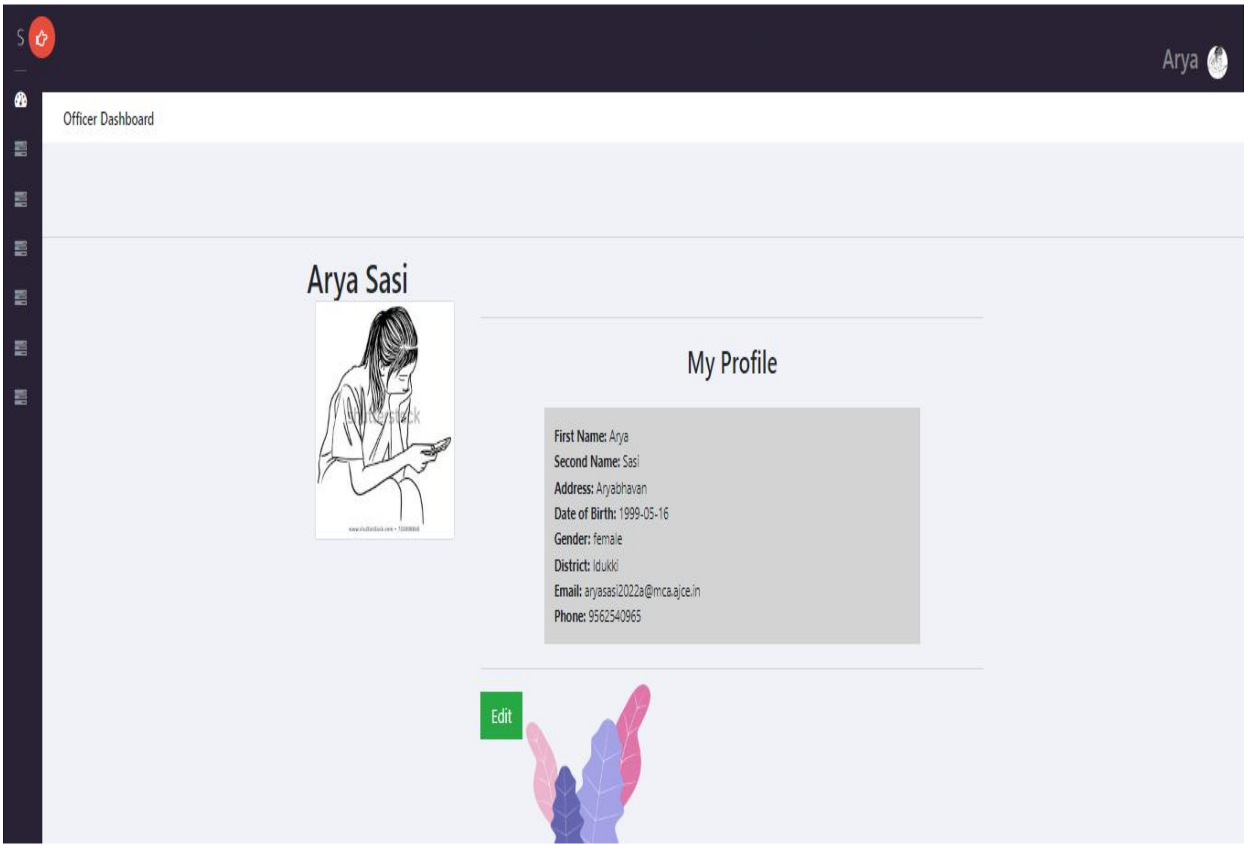


COMPLAINTS

Pending Complaints

SL.NO	NAME	DATE	COMPLAINT	STATUS
1	Arya	2022-07-07	The Traffic Lights are not working	Accept Reject

Officer Profile



Arya Sasi

My Profile

First Name: Arya
Second Name: Sasi
Address: Aryabhavan
Date of Birth: 1999-05-16
Gender: female
District: Idukki
Email: aryasasi2022a@mca.ajce.in
Phone: 9562540965