**MAKERERE  UNIVERSITY**

**TRAFFIC LIGHTS COMPLIANCE AND PENALTY SYSTEM**

By

**BSE 23 – 14**

EMBEDDED AND WEB SYSTEM.

DEPARTMENT OF NETWORKS

SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

A Project Report Submitted to the School of Computing and Informatics Technology

for the Study Leading to a Project in Partial Fulfillment of the

Requirements for the Award of the Degree of Bachelor of

Science in Software Engineering of Makerere University.

Supervisor

Dr Rashidah Kasauli.

Department of Networks

School of Computing and Informatics Technology, Makerere University

rashidahk@gmail.com, +256-41-540628, Fax: +256-41-540620

July, 2023

**Declaration**

We, group BSE 23 - 14 hereby declare that the work presented is original and has never been submitted for an award to any university or institution of higher learning.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Names** | **Registration Number** | **Student**  **Number** |
| 1 | NAKIBINGE BONNY | 1900712552 | 19/U/12552/PS |
| 2 | KATEMBEKO CHRISTOPHER | 1900724692 | 19/U/24692/PS |
| 3 | OTIM DERRICK MURUNGI | 1900712613 | 19/U/12613/PS |
| 4 | NSUBUGA ALPHA FRANCIS | 1900708652 | 19/U/8652/EVE |

**Approval**

This project report titled Traffic Lights Compliance and Penalty System has been submitted for examination with my approval as the supervisor of group **BSE 23 -14.**

DR RASHIDAH KASAULI

Department of Networks

School of Computing and Informatics Technology;

College of Computing and Information Sciences,

Makerere University

Signature: ................................................... Date: ............................

Supervisor

**Acknowledgements**

We would like to express our sincere gratitude to all individuals who have contributed to the successful completion of this report on the Traffic Light Compliance and Penalty System.

First and foremost, we would like to extend our deepest appreciation to our project supervisors, Dr. Rashidah Kasauli and Dr Mary Nsabagwa, for their guidance, support, and valuable insights throughout the development and documentation process. Their expertise and encouragement have been instrumental in shaping this project.

Furthermore, we would like to acknowledge the invaluable assistance provided by Makerere University towards giving us the opportunity to gain advance knowledge in the field of software development and engineering. Their provision of resources, technical support, and access to necessary infrastructure have been crucial in the successful execution of this project.

We would also like to extend our appreciation to our friends and family for their unwavering support and encouragement throughout this endeavor.

Lastly, we are grateful to all the individuals who have contributed their time, feedback, and expertise during the testing and evaluation phases of the project.

Without the collective efforts and support of the aforementioned individuals and organizations, this project would not have been possible. Thank you all for your contributions and commitment to this project.

# 

**Abstract**

The Traffic Light Compliance and Penalty System (TLCPS) is a technological solution designed to detect and capture vehicles violating traffic lights, issue penalties and notify the drivers thorough SMS notifications. This report presents an overview of the system, including its objectives, design, implementation, testing, performance, maintenance, and support requirements.

The system utilizes a combination of traffic lights through led light simulation, pi cameras, ultrasonic sensor, and a Raspberry Pi control system to monitor and capture traffic violations. Violations are detected through analyzing the motion vehicles at red lights. Once the vehicle is analyzed, the license plate is identified and analyzed to obtain the text license plate, penalty notifications are generated and sent to the registered vehicle owners via SMS using the Africa’s Talking API. During the development phase, comprehensive testing was conducted to ensure the system's functionality, accuracy, and reliability. Test scenarios were defined to simulate traffic violations, and the system's ability to detect and capture violating vehicles was validated. Evidence capturing, penalty issuance, and SMS delivery were also tested to ensure their proper functioning.

To maintain the system's performance, regular service and maintenance are essential. Documentation of service and support, including future updates, problem solutions, requested modifications, and support to clients, is necessary for efficient maintenance and customer satisfaction. Any anomalous operating conditions, both in the computer system and used instruments, should be addressed through precautionary steps and proper testing.

Performance requirements, such as the maximum time taken before providing output, should be met to ensure efficient operation. Support for clients should include prompt response to inquiries, troubleshooting assistance, and guidance on system usage.

In terms of future improvements, incorporating changes may be necessary to enhance the system's functionality or address evolving traffic regulations. Upgrades should be carefully planned and executed to ensure a seamless transition from the old system to the new one. Data transfer from the old system's SQL database to the new system should be carried out systematically on the cloud to prevent data loss or inconsistencies.

In conclusion, the Traffic Light Compliance and Penalty System provides an effective solution for promoting traffic safety and enforcing compliance with traffic lights regulations. With proper maintenance, support, and continuous improvement, the system can contribute to reducing traffic violations and enhancing road safety in our communities.

**Blog website:** <https://sites.google.com/view/bse23-14/home>

**Git hub:**  https://github.com/bo-nny/Traffic-Lights-Compliance-And-Penalty-System.git

**Software Design Document**

**SDD Table of Contents**

[1. INTRODUCTION 11](#_Toc140497462)

[1.1 Purpose 11](#_Toc140497463)

[1.2 Scope Of the System 11](#_Toc140497464)

[1.3 Objective (s) 11](#_Toc140497465)

[1.3.1 Specific Objectives 11](#_Toc140497466)

[1.4 The Benefits of The System 11](#_Toc140497467)

[1.5 Overview Of the Document. 12](#_Toc140497468)

[1.6 Definitions and Acronyms 12](#_Toc140497469)

[2. SYSTEM OVERVIEW 12](#_Toc140497470)

[3. SYSTEM ARCHITECTURE 14](#_Toc140497471)

[3.1 Architectural Design 14](#_Toc140497472)

[3.2 Decomposition Description 15](#_Toc140497473)

[3.3 Design Rationale 18](#_Toc140497474)

[4. DATA DESIGN 20](#_Toc140497475)

[4.1 Data Description 20](#_Toc140497476)

[4.2 Data Dictionary 20](#_Toc140497477)

[5. COMPONENT DESIGN 22](#_Toc140497478)

[5.1 Motion detection module 22](#_Toc140497479)

[5.2 License plate capture module 22](#_Toc140497480)

[5.3 Data processing module 23](#_Toc140497481)

[5.4 Penalty module 23](#_Toc140497482)

[6. HUMAN INTERFACE DESIGN 24](#_Toc140497483)

[6.1 Overview of User Interface 24](#_Toc140497484)

[6.2 Screen Images 24](#_Toc140497485)

[6.3 Screen Objects and Actions 26](#_Toc140497486)

[7. REQUIREMENTS MATRIX 27](#_Toc140497487)

[8. APPENDICES 28](#_Toc140497488)

**List of Figures**

[Figure 3.1 : module diagram showing the system components 14](#_Toc140497489)

[Figure 3.2 : level 0 diagram showing major system components. 15](#_Toc140497490)

[Figure 3.3 : Structural decomposition diagram showing the decomposition of system components 16](#_Toc140497491)

[Figure 3.4 : sub-system model diagram showing the other sub-system involved with the system. 17](#_Toc140497492)

[Figure 3.5 : sequence diagram showing the follow of activities in the system. 18](#_Toc140497493)

[Figure 4.1 : table showing the driver profile data entities, their relationships and description 20](#_Toc140497494)

[Figure 4.2 : table showing user profile and their entities and data structure 21](#_Toc140497495)

[Figure 4.3 : table showing the captured plates 22](#_Toc140497496)

[Figure 6.1 : Showing the camera interface. 24](#_Toc140497497)

[Figure 6.2 : showing the driver interface 25](#_Toc140497498)

[Figure 6.3 : showing the user interface login form. 25](#_Toc140497499)

[Figure 6.4 : showing the user interface for the data. 26](#_Toc140497500)

[Figure 1.1: Components inter-connected to Each other System 57](#_Toc140497501)

[Figure 1.2: Connection To Power Supply 58](#_Toc140497502)

[Figure 1.3: insert USB Mouse and Keyboard Violation 58](#_Toc140497503)

[Figure 1.4: PI camera 58](#_Toc140497504)

[Figure 1.5: Connection of the HDMI to the Raspberry pi and Screen 59](#_Toc140497505)

[Figure 1.6: showing the login buttons 59](#_Toc140497506)

[Figure 1.7: Screen showing Display of the system. 60](#_Toc140497507)

**List of Tables**

[Table 1.1 : table showing Acronyms and definitions. 12](#_Toc140497508)

[Table 6.1 : Table showing screen objects and their actions 26](#_Toc140497509)

[Table 7.1 : Table tracing the requirements in the SRS to the system components. 27](#_Toc140497510)

[Table 8.1 : image of the Ultra sonic sensor connected to the breadboard 28](#_Toc140497511)

[Table 8.2 : Data flow diagram showing how the data elements flow in the system. 29](#_Toc140497512)

[Table 3.1: Design details (check all that apply to your project. Make sure you can defend what you tick) 42](#_Toc140497513)

[Table 4.1: Inspection plan and performance 43](#_Toc140497514)

[Table 5.1: Checklist of the Installation and system acceptance test 51](#_Toc140497515)

[Table 5.2: Installation Procedure Check 52](#_Toc140497516)

[Table 6.1: Performance and maintenance details 54](#_Toc140497517)

# INTRODUCTION

## Purpose

This software design document describes the architecture and system design of the Traffic Lights Compliance and Penalty System (TLCPS). The purpose of this document is to provide a comprehensive overview and understanding of the system's functional and non-functional requirements, as well as its high-level architecture, component design, and implementation plan.

This SDD is intended for the software development team responsible for building and implementing the system, as well as other stakeholders such as project managers, quality assurance testers, and system administrators. This document serves as a reference point for the software development team and provides a framework for ensuring that the software is designed to meet the requirements outlined in the software requirements specification (SRS) document

## Scope Of the System

The traffic lights system software is designed to track drivers and riders who violate the traffic lights red signal. The system is composed of hardware components such as cameras and sensors that detect violations and software components that process the data in order to extract the license plates and identify the violating driver. The software will be capable of automatically issuing fines to violating drivers after capturing and analyzing the number plates and notifying the violating drivers though SMS notification.

## Objective (s)

The primary goal of the software is to increase safety on the roads by reducing the number of traffic violations which can lead to accidents and injuries along the roads. The system will also help to enforce traffic laws by the law enforcement agencies and improve traffic flow in the junctions.

### Specific Objectives

1. To identify and capture the license plate images of the vehicles that violate the traffic red light signal.
2. To extract the license plate number from the images captured using an OCR trained model.
3. To analyze the number in order to identify the violating driver and issue a penalty automatically to through a text.
4. To create evidence of the traffic light violators to ensure law enforcement with evidence.

## The Benefits of The System

1. Increased safety on the roads by reducing the number of traffic violations and accidents.
2. Increased traffic flow especially at the junctions by better enforcing the traffic red lights laws.
3. Reduce the work load of the traffic police officers who will no longer need to physically monitor the traffic flow on roads.
4. To a larger extent, revenue will be generated to the government through the fines issued to the traffic violators.

## Overview Of the Document.

The document is comprised of eight sections. The section is an introduction that highlights the purpose of the document and its intended audience. The second section the system overview that provides a general description of the functionality, context and design of the project. The third section system architecture which comprises of the architectural design and the decomposition of the system. The fourth section data design which is composed of the data dictionary and the data description and dictionary of the system. The fifth section is the component design that looks at all the components proposed in the system. The sixth section is the human interface design which contains the overview of the human interfaces, screen images and actions of the system intended to be developed. The requirements matrix that traces components and data structures to the requirements in SRS document. The last section is the appendices which is optional but includes and other kind of information that may be useful to the system and can provide a better understanding to the system.

## Definitions and Acronyms

Table . : table showing Acronyms and definitions.

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| API | Application Programming Interface |
| OCR | Optical Character Recognition. |
| RDBMS | Rational Database Management System |
| SDD | Software Design Document. |
| SRS | Software Requirement Specification. |
| TLCPS | Traffic lights compliance and penalty system. |

# SYSTEM OVERVIEW

The traffic lights compliance and penalty system is a software application that is designed monitor and manage traffic lights enforcement rules along roads. The system collects real time data from the sensors and the cameras and analyzes the image through optical character recognition (OCR) well trained model. The identified license plate is then stored and analyzed to obtain the driver details and finally, issues fines through notifying the traffic lights violators to control the violation of traffic lights.

The system is built using event-based architecture, where system modules intercommunicate with each other by sharing events. Events are triggered by motion sensed by the sensor and passed on to other components. Pipe and filter model architecture is also employed in development of the TLCPS in that the output from one module is used as the input in the other module.

Client-Server architecture is used only when the traffic officer interacts with the system interface to obtain reports and maintain the system functionality. In order to ensure system reliability and availability, the system is designed with redundant components and also built with robust security measures such as login details to protect against unauthorized access to unauthorized users. The project functionality is broken down into different steps that all together make a joint system design and they are as follows:

1. The system uses a machine learning model algorithm to analyze license plate and make intelligent decisions in identifying the number plates and OCR to analyze the license plate in order to obtain the license plate number.
2. The system also includes a violation detection module comprised of sensors to detect traffic violations such as running the red light.
3. The system is to ensure real-time automatic issuing of penalties through generating notifications to the traffic lights violating drivers through a system’s notification gateway.
4. The system design includes a dashboard that allows authorized traffic officers to monitor and manage how the system performs in case the license plate number is not found in the database.

**The project designs.**

In order to ensure complete functionality and better architecture, the system is built in different levels or layers: license plate capture, number recognition and database identification search of the number, penalty implementation.

1. The license plate capture is the first layer that will be responsible for identifying motion of vehicles at red light and capturing the license plates of the violating drivers. The design at this level is composed of hardware components such as the Ultra sonic sensor, Raspberry camera connected to the raspberry pi board that manages the functionality of the components.
2. Number recognition is the second layer of the system that provide the general purpose as to why the project is necessary. Number recognition is where images of the system are analyzed to obtain the number plate text. To achieve the recognition, machine learning such as the use of the Yolov5 is used to ensure license plate automation.

In this level, license automation, this OCR is used to convert the image into text. The license plate image is converted into a text that can be stored in the data base and analyzed.

1. Database Identification search is the next deployment of the project. The database is involved to ensure store and identification of the driver details for the identified license plate. We assume that the database contains details of the drivers and if the identified license plate is check in the database, it is either found or not and if found, the corresponding details of the driver are obtained if not found that the traffic officers have to look for the vehicle in order to obtain the driver details such that the driver is included in the database. SQL statements are used to ensure full functionality in the database.
2. Penalty implementation the last step of the system design and help to fully implement the major goals and objectives of the system. Issuing penalties is the main reason as to why the project is considered to be relevant. The design of this component is to use the SMS API to send a SMS message to the violating drivers whose number plates have been identified in the database.

# SYSTEM ARCHITECTURE

## Architectural Design

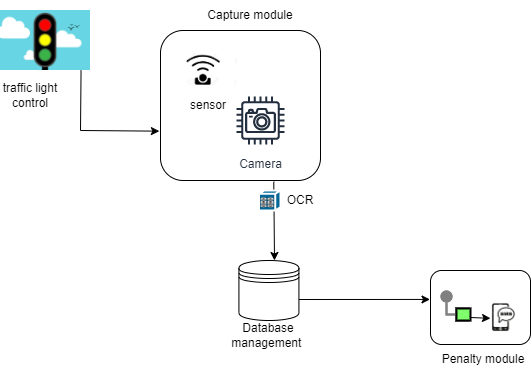


Figure . : module diagram showing the system components

The Traffic Lights Compliance and Penalty System is designed using a modular architecture to ensure flexibility, scalability and maintainability. These modules communicate with eachother through share events. The events are triggered by some action and passed to the different modules of the system. The modules of the system as shown in the diagram above as all together make up a fully functional system.

1. Motion Detection and Capture module: the capture module is triggered by an event from the traffic light when the traffic light turns red. This module belongs to event based architecture and will help in capturing of license plate of vehicles which is triggered by an event from the sensor after detecting motion of the vehicle at red light.
2. OCR: is a submodule of the system that will help to convert images the captured images into text. the module takes in the output of the capture module. Therefore, if no images are captured then the sub – module is inactive. This submodule belongs to pipe and filter architecture where by it takes in the output from the capture module.
3. Database management: is the module of the system used to store, retrieve and run queries on data. The database of the system shall store driver information such as name, date of birth, license plate, permit number etc. and run queries to the license plate number that will be passed to it to identify the details of the violating drivers. the information shall be retrieved and can be viewed by the traffic officer to ensure full system functionality.
4. Penalty module is a system module that will issue penalties to the violating drivers therefore highlighting major functionality of the system build. The module will use the SMS API to issue SMS texts to the driver phones. This module demonstrates the last architecture which is the client-server architecture.

## Decomposition Description

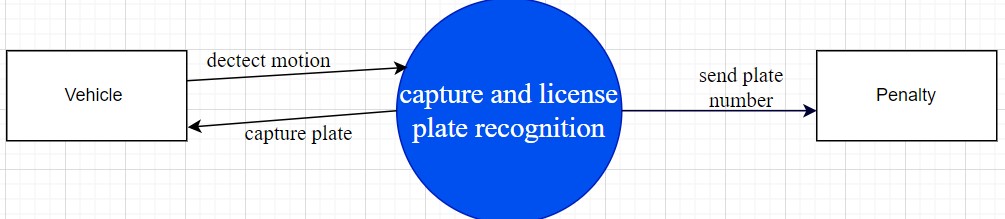


Figure . : level 0 diagram showing major system components.

The diagram explains the level 0 diagram of the TLCPS. It shows the overall system components at level 0 meaning show how the system will operate at a basic level of understanding of the stakeholders.

The vehicle represents the environment that the system will operate In or collect information from. The basic understanding of the level 0 explains that the TLCPS will detect motion of the vehicle, capture the license plate of the vehicle and finally issue the penalty. Level 0 diagram doesn’t look at the constraints of the system component such as the red light has to be on in order to detect motion.

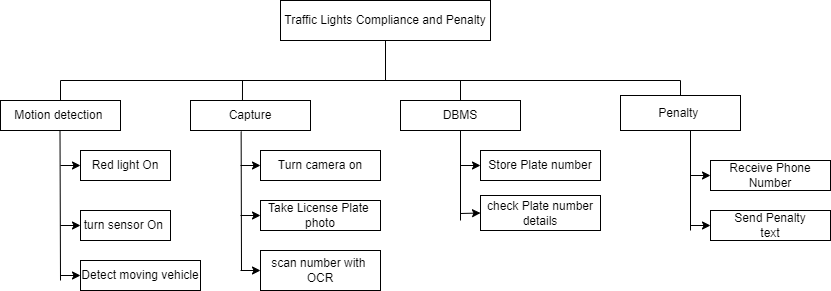


Figure . : Structural decomposition diagram showing the decomposition of system components

The Structural decomposition diagram above represents the hierarchical relationships between the system components. The TLCPS is represented by one box at level 1 and broken down into subsystem at the next level. The subsystems (motion detection, capture, DBMS, and Penalty) are then each broken into subsystems on the next level as function that each component performs. This will allow designers know how the system components perform their tasks and know how the interact with each other. The structural decomposition diagrams above clearly identify what each component of the TLCP system is supposed to perform.

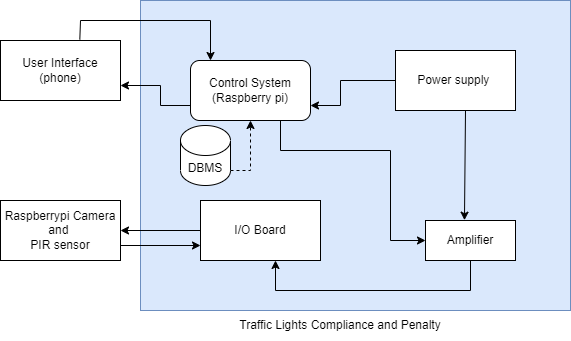


Figure . : sub-system model diagram showing the other sub-system involved with the system.

The sub system module architectural diagram above identifies the other subsystem modules that the TLCP system components interact with and show the relationships between the subsystem and the modules of the system.

Some of the other subsystems associated with the TLCP system shall be:

1. Power Supply sub system: this Subsystem is responsible for providing power supply to the system components to ensure that they start up and perform their respective functions.
2. Amplifier: this is a subsystem that allow even distribution of the power across the other components of the system. Such as amplifies power to the control system (Raspberry Pi) and the I/0 Board connected to the sensors.

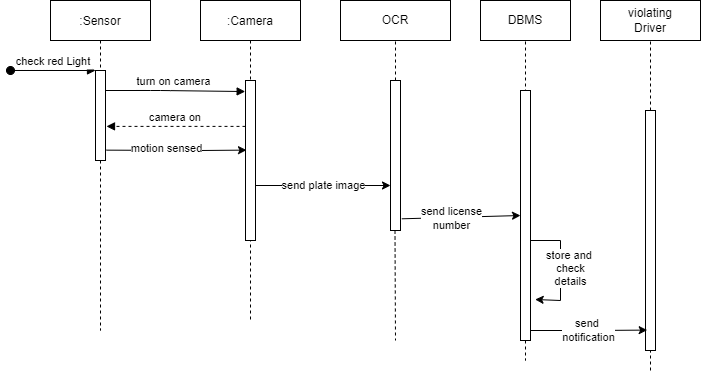


Figure . : sequence diagram showing the follow of activities in the system.

The TLCP system sequence architectural diagram represents the flow of activities within the system. It defines the individual components of the system and how events are activated and handled in the system through a sequential pattern. Actions are performed form the first to the last without going back and shows which component receives what information.

The diagram helps designers to gain a better understanding of how the system performs action to avoid loops and component interchange.

## Design Rationale

An event-based architecture is selected because it best aligns with the system's requirements and needs for real-time processing and integration with external systems. Events are the basis for communication among the system components and the functionality is triggered by the events.

There are many reasons as to why the selected architecture best suits the design of the system but the major reason are as follows:

1. The system involves multiple modules that need to communicate with each other in an asynchronous and loosely coupled manner. The architecture allows each module to operate independently and respond to events as they occur, without needing to wait for other modules to complete their tasks. For example, the capture module won’t necessarily need

to wait for the penalty module to issue the penalty to the 1st violating driver in order for it to capture another image of a violating driver. This can make the system more scalable and easier to maintain over time.

1. The system involves real-time processing of data, such as capturing license plate images from violating drivers at red light and extracting license plate numbers. An event-based architecture allows the system to respond quickly to new data as it becomes available, without the need for manual intervention or scheduling.
2. The system shall be involved in integration with external systems, such as a database of registered vehicles and vehicle owners. An event-based architecture can facilitate this integration by allowing the system to send and receive events in a standardized format that can be easily consumed by external systems.

In terms of critical issues and trade-offs, there are a few considerations to keep in mind when using an event-based architecture. One of the potential issues we identified is the potential for event overload. This occurs the system receives too many events at once and becomes overwhelmed. This can be mitigated by implementing event throttling or other strategies to limit the number of events that the system can process at any given time.

Another trade-off considered is the complexity of designing and implementing an event-based architecture. This architecture requires careful attention to event formats, event handling, and event-driven programming techniques. It also requires a clear understanding of the system's requirements and use cases to ensure that the events are designed appropriately.

In the development, Client - Server architecture will also be used in the development of the system to enable the traffic officers interact with the system. They act as clients who request requests from the system, in response the corresponding response is received. There could be many reasons as to why to a certain extent this architecture was used and they are not highlighted in this document but there is important reason that were identified in the deployment of the system and are as follows:

1. Client server architecture provides better security than other architectures. In the architecture the server is responsible for managing access to resources, which can

be used to protect sensitive data from unauthorized access. Therefore, the architecture will provide a platform for only the authorized Traffic officer in the server to access the database to retrieve the information.

1. The client-server architecture is capable of managing the data and application, which mean that any updates or maintenance can be performed centrally on the server updates are accessed anywhere. Therefore, the architecture will ensure that all authorized traffic officers will be able to access the same updates and will be able to work on the same data and application.

However much the architecture was used and beneficial in the development, it is not efficient enough to communicate all the system component and is can’t not handle multiple request or overload at the same time since the server is likely to crush due to over load. The TLCP system components do not involve request however they share information among them and to enable efficiency, each component works independently to the real time data which the client server architecture cannot handle because all responses come from the server and therefore some components will have to wait for response.

Also due to the single point of operation, the client-server architecture was not chosen because if the server fails, due to multiple requests that come in every time, then the system will have to fail.

# DATA DESIGN

## Data Description

The system will use a database to store and retrieve information. The major data and system entities that will be stored, processed and organized are:

User profile: the user profile will contain information about each user of the system including the names, title, identification number, password and some other relevant information such as location of work and residence. In the case of our traffic light compliance and penalty system, the user are the traffic officers IT personnels.

Driver profile: the driver profile will define the information about the drivers with there information including names, phone contact, age, sex, date of birth, permit number, vehicle license plate number, residence, place of work.

All our data will be stored in a data base using a relational database management system (RDBMS). The RDBMS will create tables to store the data from the user profile and Driver profile in tables and create the relationship in the data. With the help of MySQL database, we shall be able to manipulate the data. The data items involved in the system shall include:

1. OCR license plate text
2. Penalty text data
3. Motion detection data from the sensor
4. License plate images from the camera.

## Data Dictionary

Figure . : table showing the driver profile data entities, their relationships and description

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | Data Type | Field Length | Description |
| Age | Int | 2 | Age of all registered  drivers |
| Contact | Int | 10 | Phone number for the driver on which they can be accessed. |
| Date Of Penalty | Date | 9 | Date when penalty is  issued for violation |
| DOB | Date | 9 | Date of birth of the driver |
| id | Int | 2 | Unique identifier for a driver |
| license Plate | String | 12 | Number plate  uniquely identifying each car. |
| Name | String | 30 | Name of the driver. |
| Permit No | String | 15 | Identifier that the driver is permitted to drive a vehicle |
| Place of work | String | 10 | Company or  organization where the  driver works from. |
| Residence | String | 10 | Place where the driver lives. |
| sex | String | 1 | Gender of the driver. |

Figure . : table showing user profile and their entities and data structure

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | Data Type | Field Length | Description |
| Contact | Int | 10 | Phone number of the  traffic officer |
| Id | Int | 2 | Unique identifier of the office |
| Name | String | 24 | Name of the traffic  officer |
| Password | String | 8 | Secret character for logging onto the system |
| Place Of Work | String | 10 | Area or work station |
| Residence | String | 10 | Place where office  lives |
| Sex | String | 1 | Gender of the traffic officer. |

Figure . : table showing the captured plates

|  |  |  |
| --- | --- | --- |
| Entity Name | Data Type | Field length |
| Id | Int | 8 |
| License plate | Varchar | 8 |
| Date of Capture | Date | 10 |

# COMPONENT DESIGN

## Motion detection module

The motion detection module is the module the triggers the events and is responsible for identifying violations of traffic lights compliance.

**Pseudocode:**

While traffic light is **red**.

Sensor is switched on and camera is turned on.

**If** vehicles move past the sensor,

Motion Is detected.

## License plate capture module

The license plate capture is the module the triggers the event of capturing and takes on the event that was trigger by the sensor to detect motion to determine that the vehicle should be captured therefore, the license plate is captured.

This module also takes on the responsibility of capturing the license plate number as a text using the OCR sub system model. The OCR is a model trained to analyze the license plate in order to capture the text on the plate

**Pseudocode:**

**While** camera is on.

Capture license plate of moving vehicle.

**If** image captured.

Convert to text.

Clear the text to obtain the actual plate.

Confirm the degree of accuracy.

## Data processing module

Data processing module is the module responsible for storing, analyzing and verifying the existence of the license plate in the database. The database uses SQL statements to analyze the license plate number inform of a text. The module is triggered by the actions from the output from the License plate capture module.

**Pseudocode:**

While text accuracy is greater than 80%.

Store text in the database.

Match/compare the text to check its existence.

If Text exits.

Using SQL statement obtain the contact number and permit details.

## Penalty module

The penalty module is triggered after obtaining the analysis of the data processing module. It takes into account of the contact and permit number related to the license plate number that was captured.

A penalty text is forwarded to the contact obtained for the violating driver.

In addition, the penalty module will be responsible for managing reports. It will be able to pull the records of the license plate that has been captured and issued a violation penalty. It will follow the penalty by checking the status of the penalty.

The traffic officers will be involved in the module for analyzing and check the status of the penalties and verifying the degree of accuracy of the license plate number.

# HUMAN INTERFACE DESIGN

## Overview of User Interface

The user interface of system will aim at providing an intuitive and user-friendly experience for users. The system will mostly be an embedded system that takes into account of the sensor interface, camera interface to input the data into the system.

The Ultra sonic sensor interface will sense the motion of the vehicles the pass infront of it when the red light is on implying that the there is a violation of the traffic lights. And on the other hand, the camera interface is responsible for capturing the images of the vehicles and keeping track of the traffic light violations.

The system outputs the results of the analysis through sending messages to the violating drivers’ mobile phones as texts and also provide reports to the user through a web interface that will provide the status of the penalty and feedback of invalid license plate numbers.

## Screen Images

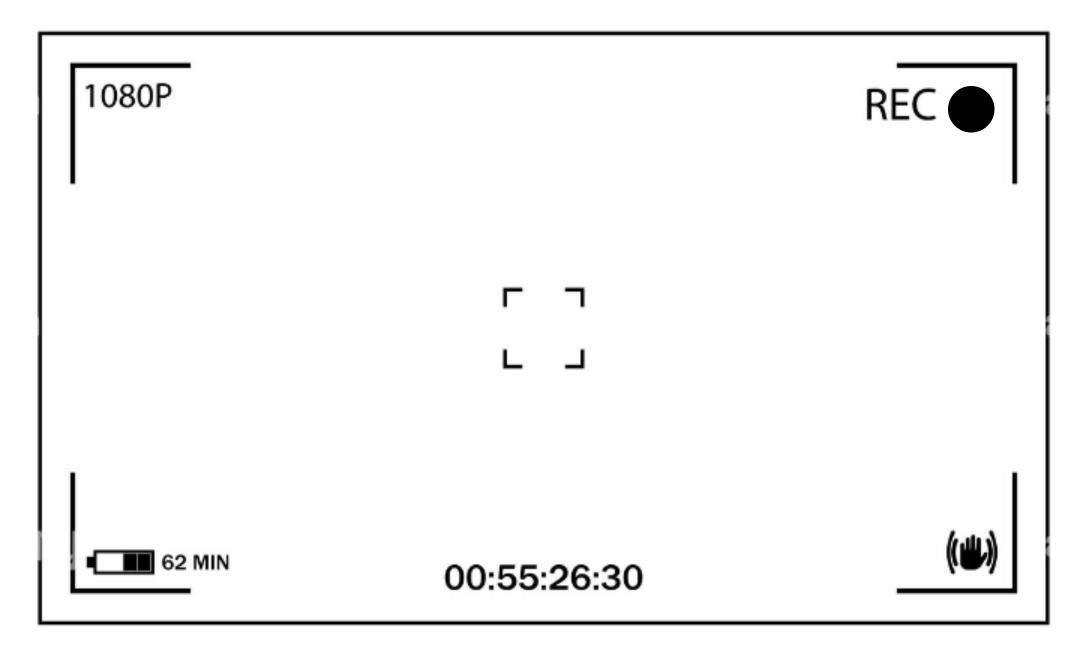


Figure . : Showing the camera interface.

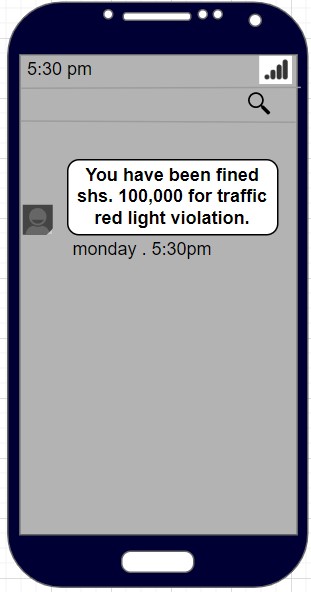


Figure . : showing the driver interface



Figure . : showing the user interface login form.

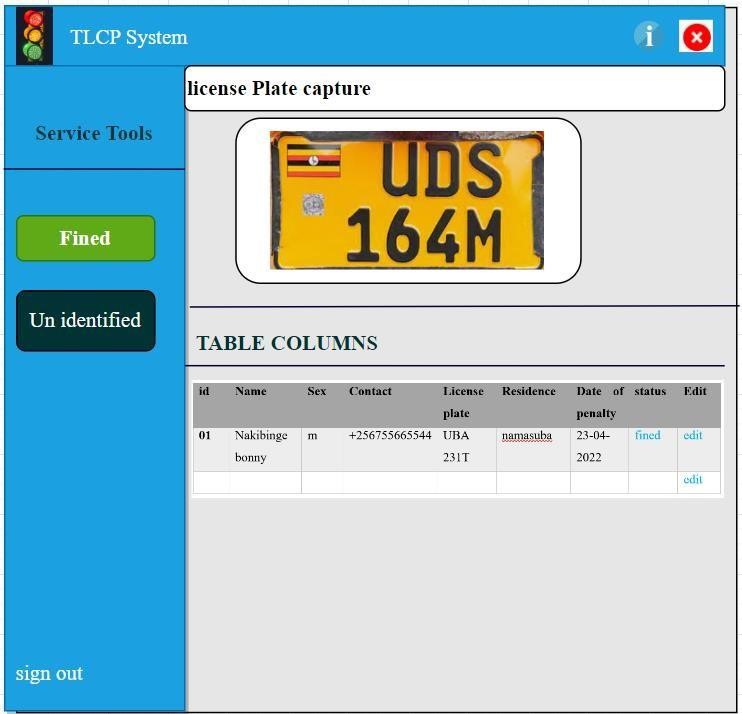


Figure . : showing the user interface for the data.

## Screen Objects and Actions

Table . : Table showing screen objects and their actions

|  |  |  |
| --- | --- | --- |
| **Screen Object** | **Screen object image** | **Action** |
| Officer Id input |  | Allow Traffic Officer  identification |
| Password input |  | Allow Traffic Officer  authentication |
| Login button |  | Enter system display on click. |
| Forgot Password? |  | Go to forgot password menu for the traffic officer to reset the password. |
| Fined button |  | Open table filtered with fined traffic violators. |
| unidentified button |  | Filters license plates that are not identified. |
| Visualization Screen |  | Show the officer that system is operating. |
| Edit |  | Allows the traffic officer assign penalty manually. |
| Close |  | Close the system interface |
| About button |  | Provide information about the traffic lights compliance and  penalty system |
| Sign out |  | Logs out the traffic officer personnel. |

# REQUIREMENTS MATRIX

Table . : Table tracing the requirements in the SRS to the system components.

|  |  |  |  |
| --- | --- | --- | --- |
| **System Component** | **Functional**  **Requirement** | **Priority** | **Description** |
| Motion detection  module | NPCS 3 | High | Detect motion of the vehicle when it is red light. |
| License plate capture module | NPCS 1, NPCS 2 | High | 1. Capture license plate images 2. Scan the plate using OCR to obtain the license plate text. |
| Data processing  module | DLM 1 | High | 1. Store the license plate text. 2. Match the plate to obtain   Attached information eg name, contact, permit number of the driver |
| Penalty module | DF 1, DF 2 | high | Issues penalty through the SMS notification. |

# APPENDICES

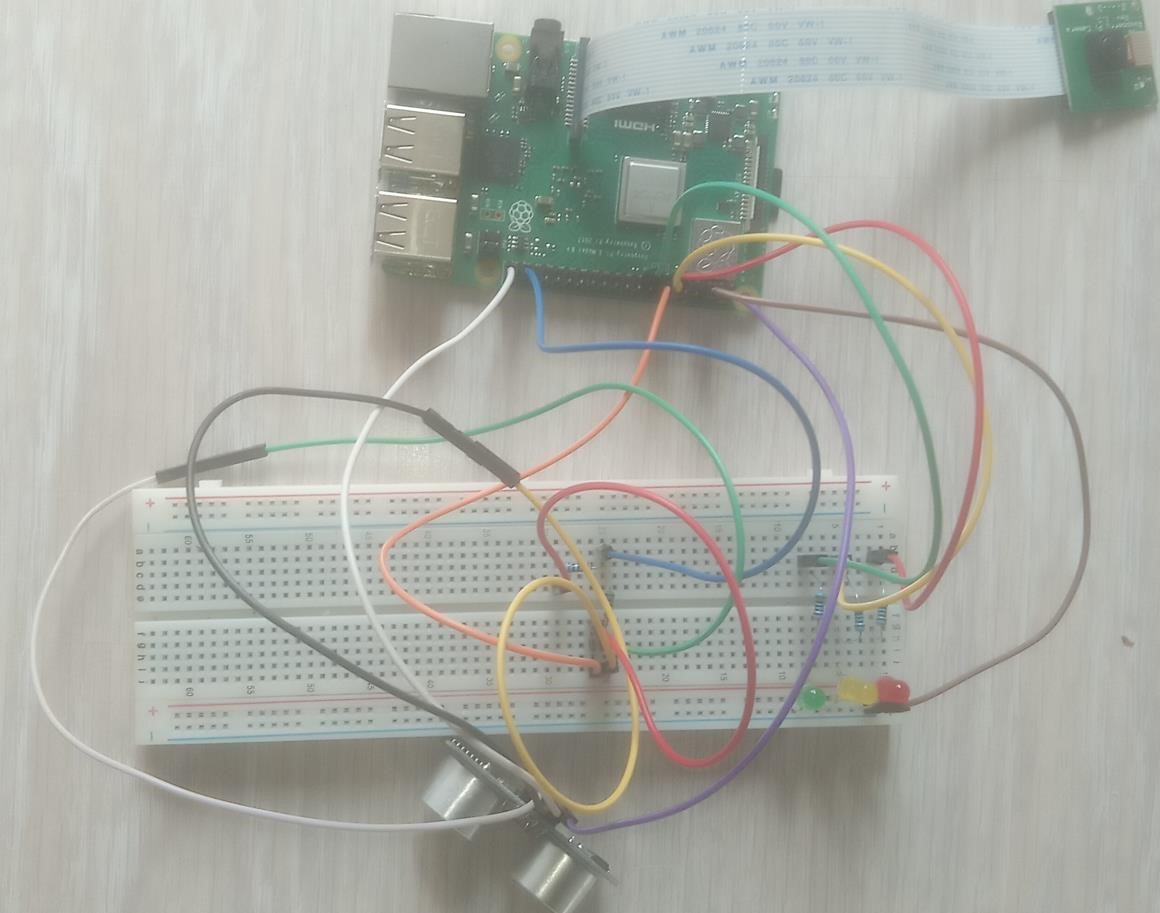


Table . : image of the Ultra sonic sensor connected to the breadboard

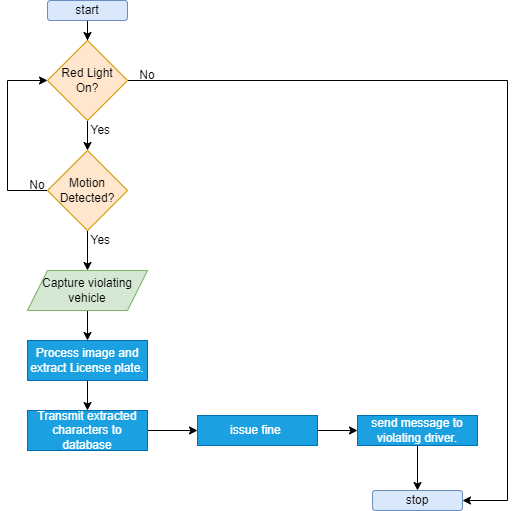


Table . : Data flow diagram showing how the data elements flow in the system.

**Final Report**

**Report Table of Contents**

[1. Chapter 1: Introduction 34](#_Toc140497518)

[1.1 Background and scope of the project 34](#_Toc140497519)

[1.2 Overview of the document 34](#_Toc140497520)

[1.2.1 Introduction 34](#_Toc140497521)

[1.2.2 System Specifications 34](#_Toc140497522)

[1.2.3 Design Output 34](#_Toc140497523)

[1.2.4 Inspection and Testing 35](#_Toc140497524)

[1.2.5 Installation and System Acceptance Test 35](#_Toc140497525)

[1.2.6 Performance, Servicing, Maintenance, and Phase Out 35](#_Toc140497526)

[1.2.7 Conclusion and Recommendations 35](#_Toc140497527)

[1.2.8 Appendix A: User Manual 35](#_Toc140497528)

[2. Chapter 2: System Specifications 35](#_Toc140497529)

[2.1 Version of requirements and Version Control 35](#_Toc140497530)

[2.2 Input 36](#_Toc140497531)

[2.3 Output 36](#_Toc140497532)

[2.4 Functionality 37](#_Toc140497533)

[2.5 Limitations and safety 37](#_Toc140497534)

[2.5.1 Internet connectivity 37](#_Toc140497535)

[2.5.2 Image quality and variations 38](#_Toc140497536)

[2.5.3 Safety Precaution 38](#_Toc140497537)

[2.6 Default settings 38](#_Toc140497538)

[2.7 Special requirements 38](#_Toc140497539)

[2.7.1 Hardware specifications 38](#_Toc140497540)

[2.7.2 Software dependencies 38](#_Toc140497541)

[2.8 Errors and alarms 39](#_Toc140497542)

[2.8.1 License plate not detected 39](#_Toc140497543)

[2.8.2 Text recognition failure 39](#_Toc140497544)

[2.8.3 Alarm 1: Failure to Issue Penalty 39](#_Toc140497545)

[3. Chapter 3: Design output 39](#_Toc140497546)

[3.1 Implementation (coding and compilation) 39](#_Toc140497547)

[3.2 Device interfaces 40](#_Toc140497548)

[3.2.1 Ultra sonic sensor 40](#_Toc140497549)

[3.2.2 A breadboard 40](#_Toc140497550)

[3.2.3 LED lights (Light-Emitting Diodes) 40](#_Toc140497551)

[3.2.4 Computing devices such as computer, flat screen 40](#_Toc140497552)

[3.3 Hardware environment 40](#_Toc140497553)

[3.4 Documentation 41](#_Toc140497554)

[3.4.1 User Manuals 41](#_Toc140497555)

[3.4.2 Technical Specifications 41](#_Toc140497556)

[3.4.3 Design Documents 41](#_Toc140497557)

[3.4.4 Installation Guides 41](#_Toc140497558)

[4. Chapter 4: Inspection and testing 42](#_Toc140497559)

[4.1 Introduction 42](#_Toc140497560)

[4.2 Test plan and performance 44](#_Toc140497561)

[4.2.1 Test Objectives of the development of the system 44](#_Toc140497562)

[4.2.2 Test Coverage 44](#_Toc140497563)

[4.2.3 Test Methods and Techniques 45](#_Toc140497564)

[4.2.4 Test Procedures 45](#_Toc140497565)

[4.2.5 Test Data and Environments 45](#_Toc140497566)

[4.2.6 Test Execution and Reporting 45](#_Toc140497567)

[4.2.7 Test objectives 45](#_Toc140497568)

[4.2.8 Scope and Relevancy of tests 46](#_Toc140497569)

[4.2.9 Levels of tests 47](#_Toc140497570)

[4.2.10 Types of tests 47](#_Toc140497571)

[4.2.11 Sequence of tests 48](#_Toc140497572)

[4.2.12 Configuration and calculation tests 49](#_Toc140497573)

[4.3 Precautions 49](#_Toc140497574)

[4.3.1 Anomalous conditions 49](#_Toc140497575)

[4.3.2 Precautionary steps taken 50](#_Toc140497576)

[5. Chapter 5: Installation and system acceptance test 50](#_Toc140497577)

[5.1 Input files 50](#_Toc140497578)

[5.2 Supplementary files 51](#_Toc140497579)

[6. Chapter 6: Performance, servicing, maintenance, and phase out 53](#_Toc140497580)

[6.1 Service and maintenance 53](#_Toc140497581)

[6.2 Performance and Maintenance 53](#_Toc140497582)

[6.3 Chapter 7: Conclusion and Recommendations 55](#_Toc140497583)

# Abbreviations/Acronyms

HTML: Hyper Text Markup Language

LED: Light-Emitting Diode

OCR: Optical Character Recognition

OS: Operating System PHP: Hypertext Preprocessor

SSD: Solid-State Drive

SMS: Short Message Service

SSH: Secure Shell

TLCPS: Traffic Light Compliance and Penalty System

USB: Universal Serial Bus

# Chapter 1: Introduction

## Background and scope of the project

The traffic lights compliance and penalty system are an implementation that aims to address the issuing of penalties to drivers that have violated the traffic lights on the red light for stop. The identification of these specific drivers is through detection of their license plates using a deep learning model that is trained using Yolov5 to capture license plates. The system utilizes different technologies such as OpenCV, Python, PHP, HTML, Tesseract-OCR, deep learning using Yolov5 to achieve its objectives.

The system's main purpose is to automate the process of license plate capturing, detection, recognition, and analysis to extract the extract and analysis to obtain the corresponding captured driver. The detection is aided with the use of an ultra-sensor that helps to measure the distance within which the vehicle is, and detect it. The distance measuring is to ensure that the vehicle detected on fall in the particular lane of the road. It’s the constraints that help the system not to detect vehicles out of its lane of the road for the detection.

By implementing computer vision techniques using OpenCV and Python, the system can detect and extract license plate regions from images or video streams from the pi camera frame relating to the trained deep learning model. It then utilizes Tesseract-OCR for accurate text recognition of the extracted license plate text. Additionally, the system integrates with the Africa’s Talking API to send

SMS notifications to the driver’s contact issuing a penalty corresponding the captured license plate.

## Overview of the document

The implementation, testing, and validation results of the Traffic Lights compliance and penalty system are included in this document. The document is organized into numerous sections to make it easy to grasp the system's implementation, testing, installation, and maintenance aspects.

### Introduction

This chapter provides an introduction to the document presenting the background and scope of the project. It also offers an overview of the entire document and sets the context for the subsequent chapters as well the system’s implementation details.

### System Specifications

In this chapter, the complete system specifications are described. It includes the versioning of requirements, input details, output formats, system functionalities, limitations, safety precautions, default settings, special requirements, errors, and possible alarms. The chapter also covers aspects such as version control and tools used for code management like git.

### Design Output

This chapter entails the implementation details, coding, and compilation aspects. It includes information about the development tools used, any anomalies encountered during the implementation, device interfaces and equipment supported, hardware and software operating environment, and documentation details.

### Inspection and Testing

This chapter focuses on the inspection plan and performance of the system. It covers the inspection of design outputs, documentation, software development environment, and the results of the inspection. The chapter also discusses the test plan and performance, including test objectives, scope, types of tests, test cases, and expected results.

### Installation and System Acceptance Test

In this chapter, the installation process and system acceptance testing are addressed. It includes details about input files, supplementary files, installation qualification, and a checklist for the installation procedure.

### Performance, Servicing, Maintenance, and Phase Out

Here, the requirements for service, maintenance, performance, and support are discussed. The chapter covers topics such as problem detection and solutions, functional maintenance, performance improvement suggestions, and the phase-out process.

### Conclusion and Recommendations

The final chapter provides a conclusion based on the findings of the report. It summarizes the key points discussed throughout the document and offers recommendations for further improvements or enhancements.

### Appendix A: User Manual

This appendix provides a user manual for the traffic lights compliance and penalty system. It includes detailed instructions on how to use the system, where to seek help or support, and any accompanying visual aids such as screenshots or diagrams.

Finally, the document concludes with a section for final approval for use, where the identification of the responsible personnel, validation status, and any remarks can be recorded.

# Chapter 2: System Specifications

## Version of requirements and Version Control

The version of the requirements document for the Traffic lights compliance and penalty system is 1.0. Throughout the development process, changes have been made to the requirements document to accommodate evolving project needs and stakeholder feedback. The changes made between versions Python 3.7.0 and raspberry pi 3B+. The change in the python version 3.9.0 to 3.7.0 inside the project folder to create a safe virtual environment to run the project dependencies. The change in the version of the raspberry pi 3B+ from raspberry pi 4 was also made to the system.

The reason for the change in the python version was to accommodate the python dependencies that couldn’t be installed on our raspberry pi 3B+. In order to make the system more efficient to our software components, the device provides a way to install the python 3.7.0 inside the project folder in order not to affect other components that are installed on python 3.9.0. Therefore, the virtual environment inside the folder takes into account of python 3.7.0.

The reason for the change in the raspberry pi version to 3B+ was due to the cost funds of the whole project. And also, both version of the raspberry pi (4 and 3) performs the same functions. However, version 3 is weaker but we created software components that are lighter to take into account of the efficiency and performance of the system.

To manage versioning of the code and ensure effective collaboration, version control tools such as Git were utilized. These tools allow for tracking changes, branching, merging, and maintaining a history of code revisions. By leveraging version control, we managed different versions of the software and identified specific versions (such as 3.9, 3.7) based on the commit history and tagging.

## Input

The Traffic lights compliance and penalty system receives various inputs, which are crucial for its functionality and performance as well. They act as the basis to which the system operates.

**Input 1**: Image or video data

Before we capture the images, the system detects using the ultra-sonic sensor a violating vehicle that is within a specific distance and the pi camera video is obtained at that moment. The system then accepts image or video data as input for license plate detection and recognition and the input is obtained from the raspberry pi camera.

**Input 2**: User configuration settings

Users can provide configuration settings, such as detection thresholds, or text analysis options. The region of interest and detection thresholds are inputs obtained from the deep learning model that detects the license plate region and therefore, they influence the system's behavior and customization.

You can state the traffic officer login details as follows:

**Input 3**: Traffic Officer Login Details

The traffic officer login details are required to access the system. The traffic officer needs to enter their user id and password in the respective fields to authenticate themselves and gain access to the system's functionalities. This helps to ensure confidentiality and security of the system.

## Output

**Output 1**: Detected license plate information

The system runs inference and outputs the detected license plate regions within the input images or video frames using the labels from the trained model. It provides bounding box coordinates which indicate the location of the license plates.

**Output 2:** Extracted license plate text

The system performs text analysis using Tesseract-OCR on the detected license plate regions and it outputs the recognized text which represents the alphanumeric characters present on the license plates. The plate is stored in a relational database in a table containing a trigger that matched the plates to the corresponding plate in the driver table and the driver details are obtained to be stored in the other table such that the text can be constructed.

**Output 3**: SMS notifications

After analyzing the license plate text and obtaining the driver contact using the triggers, the system can send SMS notifications to the violating driver for the penalty issuing. The notification includes information such as the license plate number, and location where the vehicle was captured from.

## Functionality

**Functionality 1**: License plate detection

The system utilizes computer vision techniques like Open Cv to detect and localize license plates within images or video frames. It employs object detection algorithms such as a deep learning-based model that uses the single shot multi-box detector (SSD) that predicts the object bounding boxes and class labels in a single pass, to identify license plate regions accurately.

**Functionality 2**: License plate recognition

After the images are captured, they are sent to the server and inference I performed to them to obtain the region of interest where the plate Is located and cropped out.

The system then performs optical character recognition (OCR) on the detected license plate regions to extract the alphanumeric characters. By leveraging Tesseract-OCR, the system can recognize and interpret the license plate text with high accuracy to obtain the text from the image boundaries.

**Functionality 3**: SMS sending

The system integrates with the Africa’s Talking API to send SMS notifications to the driver’s contacts who are captured violating the traffic lights. After analyzing the license plate information, the system composes and sends SMS messages to predefined contacts issuing a penalty.

**Functionality 4:** Traffic Officer Performance Metrics

To assess the performance and effectiveness of the system, a performance monitoring feature was implemented for monitoring the system by the authorized IT traffic officer. This functionality allows the tracking and analysis of various metrics related to the system performance such as capturing and identifying the violating drivers and accuracy of the captured number plates.

## Limitations and safety

### Internet connectivity

The system requires an active internet connection to interact with the Africa’s Talking API for sending SMS notifications. Without internet access, the SMS sending functionality will be unavailable.

### Image quality and variations

The accuracy of license plate detection and recognition heavily depends on the quality of input images or video frames. Challenges such as low lighting, blur, or frame movement or video display affect the system during it performance. This is most likely due to low power system component CPU i.e. the raspberry pi 3 B+.

### Safety Precaution

Data privacy

The traffic lights compliance and penalty system handle sensitive information, such as license plate numbers and personal data for the drivers. Appropriate measures are taken in place to ensure the privacy and security of this information, adhering to relevant data protection regulations through allowing only the authorized administrator to access the system to monitor its performance.

## Default settings

By default, the system should be able to perform License plate detection based on a default object detection model with predefined detection thresholds that was trained to perform the detection through the Open Cv.

For License plate recognition, by default Tesseract-OCR settings for license plate text extraction should be installed to run the extraction program.

For SMS sending, Africa’s Talking API configuration with predefined message content and driver contact identification should be deployed to perform issuing of penalties.

## Special requirements

### Hardware specifications

The system requires a more powerful raspberry pi with sufficient processing power and memory to perform real-time or near real-time license plate detection and recognition. GPU acceleration and Coral platform to optimize for running machine learning models effectively allowing for faster and real time influence of the raspberry pi performance, but it is not mandatory.

### Software dependencies

The system relies on several software dependencies including OpenCV, Python, Tesseract OCR, Africa’s Talking API, and any other libraries or frameworks specified in the documentation. These dependencies must be installed and configured correctly for the system to function as intended.

## Errors and alarms

### License plate not detected

One of the most frequent errors encountered in the system was the failure to detect any license plate regions in the input images or video frames based on the license plate detection model. The error is identified when no label region if encountered around the detected license plate.

This could be due to factors such as poor image quality, incorrect region of interest selection, or limitations of the object detection algorithm therefore to enhance the system to avoid the error we can use the coral platform to improve functionality or deploy the system with better hardware components.

### Text recognition failure

Another common error was when the system cannot accurately recognize the license plate text. if the system fails it produces an error message due to failure to detect the text.

Some of the factors that may contribute to recognition failures include poor image quality, complex background patterns, or variations in license plate design.

To address this error, training the model with more data and through more steps can improve the detection of the images captured. Deploying other OCR platform such as Google cloud Vision, Amazon Textract, OCRopus could be more accurate than tesseract.

### Alarm 1: Failure to Issue Penalty

In the event of an error or issue with issuing penalties due to failure to send SMS, the system generates an alarm to address the failure. The error could occur due to network connectivity problems, incorrect Africa’s Talking API configuration, or limitations imposed by the SMS service provider.

To go around the error, always verify and check the internet connections of the system and ensure maintenance activities and always confirm the API used.

# Chapter 3: Design output

## Implementation (coding and compilation)

The system was developed using different programming languages that communicate together to accomplish the system functionalities. Python, a widely and versatile programming language was used to run the sensor and led lights that simulate the traffic lights control system to perform object detection using a trained model. The entire process of object detection and license plate capture was facilitated by python.

The system has a web interface thats developed using the web-based programming languages such as HTML, PHP, JAVA SCRIPT.

Finally, the raspberry pi system runs on the bullseye operating system 32-bit software.

For coding and development, the system was developed using an integrated development environment that utilizes Visual Studio Code that connects to the raspberry pi through a remote connection using SSH. It provides a streamlined development experience with code editing, debugging, and version control integration.

The frameworks and libraries used to enhance functionality include open Cv that was used for image processing, computer vision tasks, and object detection. They provide functions and algorithms for manipulating and analyzing images and videos. Additionally, Yolov5 library was employed in the system for machine learning tasks specifically for training and deploying object training models. Yolov5 offers a high-level API that simplifies the process of building and training models within platforms such as Jupiter note book and Collab an online platform.

Version control such as Git were employed to manage the source code and track changes throughout the development process and as the system relied on python, build tools like pip is utilized to install and manage the required dependencies. Pip simplifies the installation process and ensures that the necessary libraries and packages on the raspberry pi control system.

During the development process, various anomalies and issues were encountered. These anomalies include software bugs, performance bottle necks or compatibility problems in libraries and software platforms.

## Device interfaces

### Ultra sonic sensor

These are used to detect the vehicle that is violating the traffic lights. The reason behind the use of the ultra-sonic sensor was to both detect the vehicles and create a difference between the lanes of the road. The roads consist of two lanes the right and the left, therefore the ultra-sonic sensor will only detect the vehicles within a given distance for which the traffic lights are ordering and it can also be used to detect distance.

### A breadboard

These were used for prototyping circuits without the need for soldering. It consists of a grid of interconnected metal strips and holes that allows to insert and connect electronic components easily. The breadboard provided a platform to connect the ultra-sonic sensor and LED lights to the Raspberry Pi.

### LED lights (Light-Emitting Diodes)

These were used to visually simulate the traffic light control system.

### Computing devices such as computer, flat screen

These were used to access the system and to visualize the web interface and also help in operating the and programming the raspberry pi.

## Hardware environment

Consists of components such as raspberry pi, power supply, peripherals, bread board.

The raspberry pi 3 model B+ is a micro control processing system unit for managing and controlling the system function.

The power supply unit such as a USB power adapter is used to supply the power to the raspberry pi to power it up.

Peripherals highlight the additional devices that are connected to the raspberry pi to add on its functionality. These include flat screen, keyboard, mouse, and the pi camera module for the raspberry pi.

**3.3.1 Software Environment** consist of different operating software the is installed on the raspberry pi such as Raspbian, Raspberry pi OS. Additionally, other software used on the system include, python 3.7.0, Yolov5, OpenCV 4.5, Tesseract 4.1, HTML, PHP, and JAVA Script.

The system runs on a network and we deployed a network router such as a phone that provides interconnection between the devices that interacts the system to a local area network and internet access.

## Documentation

### User Manuals

User manuals provide instructions and guidelines on how to use the system effectively. They explain the system's features, functionalities, and operation, helping users understand how to interact with the system and achieve their desired tasks.

### Technical Specifications

Technical specifications provide detailed information about the system's architecture, components, interfaces, and requirements. They serve as a technical reference for developers, system administrators, and other technical personnel involved in the implementation, maintenance, and troubleshooting of the system.

### Design Documents

Design documents outline the system's design approach, methodology, and overall structure. They describe the system's modules, their interactions, and the rationale behind design decisions. Design documents are important for developers, as they provide insights into the system's design principles and assist in understanding its internal workings.

### Installation Guides

Installation guides provide step-by-step instructions for installing and configuring the system. They ensure that the system is properly set up and ready for use, guiding system administrators or end-users through the installation process.

Table .: Design details (check all that apply to your project. Make sure you can defend what you tick)

|  |  |  |
| --- | --- | --- |
| Topics | **Design output** | |
| **Good programming**  **practice**  Efforts made to meet the recommendations for good programming practice... | Source code is...  Modulized  Encapsulated  Functionally divided  Strictly compiled  Fail-safe (handling errors) | Source code contains...  Revision notes  Comments  Meaningfull names  Readable source code  Printable source code |
| **Dynamic testing**  Step-by-step testing made dynamically during the implementation... | All statements have been executed at least once  All functions have been executed at least once  All case segments have been executed at least once  All loops have been executed to their boundaries  Some parts were not subject to dynamic test  Comments: | |

# Chapter 4: Inspection and testing

## Introduction

The inspection and testing phase of the Traffic lights compliance and penalty system was crucial to ensure its overall quality, functionality, and adherence to the specified requirements. This phase involves a systematic and thorough evaluation of the system's components, features, and performance to identify any discrepancies, errors, or areas for improvement.

The primarily objectives of the inspection and testing include, Quality Assurance to verify that the system meets the desired quality standards such as accuracy of the license plate text from the captured images, identification of the license plate and the driver that’s violating the traffic light.

Functionality Validation was essential to ensure the system’s responsiveness to perform its specified functions.

Risk mitigation to identify and address any potential risks, issues, or vulnerabilities that could affect the system's performance or security. Such as dark environment that restricts the system from detecting and capturing the license plates.

Table .: Inspection plan and performance

|  |  |  |  |
| --- | --- | --- | --- |
| Topics | **3.3.1 Inspection plan and performance** | | Date / Initials |
| **Design output**  Results from the Design Output section inspected... | Program coding structure and source code  Evidence of good programming practice  Design verification and documented reviews  Change-control reviews and reports  Comments:  The code best operates on the raspberry pi. | | 10/06/2023 –  NB  11/06/2023 –  KC |
| **Documentation**  Documentation inspected... | System documentation, flow charts, etc.  Test results  User manuals, On-line help, Notes, etc.  Contents of user manuals approved  Comments:  Images were included to aid understandability. | easy | 10/06/2023 –  NB  11/06/2023 –  NA  11/06/2023 –  OD |
| **Software development environment**  Environment elements inspected... | Data integrity  File storage Access rights  Code protection  Installation kit, replication and distribution  Comments:  Data is stored and easily managed in a  relational database ie Maria DB | | 10/06/2023 –  NB  11/06/2023 –  KC  10/06/2023 –  NB  11/06/2023 –  NA  11/06/2023 –  OD |
| Topics | **3.3.1 Inspection plan and performance** | | Date / Initials |
| **Result of inspection**  Approval of inspection. | Inspection approved  Comments:  The system inspection is approved by all the group members to ensure full functionality. | | 09/06/2023 –  NB  09/06/2023 –  NA  11/06/2023 –  OD  11/06/2023 –  KC |

## Test plan and performance

### Test Objectives of the development of the system

The test objectives for the Traffic lights compliance and penalty system outlined the specific goals of the testing phase. This included verifying the functionality of the traffic lights, assessing the accuracy of the object detection algorithm, evaluating the performance of the system under different traffic conditions, ensuring compliance with regulatory requirements and ensuring the issuing of penalties to violating drivers.

### Test Coverage

The test plan defined the scope of the testing activities for the Traffic lights compliance and penalty system. It identified the components and functionalities of the system that need to be tested, such as the hardware components (Raspberry Pi, sensors, LEDs), software modules (object detection model, web interface), and their interactions.

### Test Methods and Techniques

The test plan specified the testing methods and techniques to be employed for the Traffic lights compliance and penalty system. This included functional testing to ensure proper functioning of each component, performance testing to assess response times and system scalability, security testing to identify vulnerabilities, and usability testing to evaluate user interaction with the web interface.

### Test Procedures

The test plan outlined the step-by-step procedures for conducting the tests specific to the

Traffic lights compliance and penalty system. It included instructions for setting up the hardware components, configuring the software environment, simulating traffic scenarios, and collecting test data. Additionally, it defined the expected results and criteria for passing each test.

### Test Data and Environments

The test plan identified the test data required for evaluating the Traffic lights compliance and penalty system. This involved generating sample traffic patterns, capturing real-world traffic data, or using simulated data for testing purposes. The test plan specified the necessary test environments including the hardware and software configurations required to run the system.

### Test Execution and Reporting

The test plan detailed how the tests will be executed and how the results will be recorded and reported for the Traffic lights compliance and penalty system. It would include information on the testing team responsible for carrying out the tests, the schedule for conducting the tests, and the format for documenting the test outcomes. This ensures that the testing process is systematic, well documented, and allows for tracking any issues or deviations encountered.

### Test objectives

The main objective of the system to detect, capture license plate and issue a penalty, a Traffic Violation Detection and Penalty Issuance Test was performed.

#### Description

The Traffic Violation Detection and Penalty Issuance Test aims to verify the system's ability to detect and capture vehicles that violate traffic lights, as well as issue penalties by sending SMS notifications. This test ensures that the system accurately identifies traffic violations, captures the license plate clearly, and effectively delivers penalty notifications to the responsible vehicle owners.

#### Why

The test was conducted to ensure the system's effectiveness in enforcing traffic regulations, deterring violations and promoting road safety. By validating the detection and penalty issuance functionalities, we can ensure that the system operates reliably and contributes to the overall traffic management objectives.

**Steps Taken and What Was Tested.**

1. **Set up the Test Environment:**

Prepare the physical setup with the traffic lights, cameras, ultra- sensors, and Raspberry Pi control system and configure the system software and ensure proper connectivity using the bread board and jumper cables.

1. **Define the Test Scenario:**

Simulate traffic scenarios where vehicles may potentially violate traffic lights. such as redlight running. Then determine the expected behavior of the system when a violation occurs.

1. **Perform Test Violations:**

Trigger traffic violations intentionally by simulating vehicles violating traffic lights and validate that the system detects and captures the vehicle license plate accurately.

1. **Capture Evidence:**

Verify that the system captures relevant images or videos of the violating vehicles and ensure that the captured images are clear, identifiable, and timestamped through observing the system’s extracted license plate from the image.

1. **Penalty Issuance:**

Confirm that the system generates penalty notifications for the identified violations and validate that the penalty notifications the reason for the penalty and where it was captured.

1. **SMS Delivery:**

Verify that the system sends SMS notifications through the Africa’s Talking API to the registered vehicle owners associated with the violations and ensure that the SMS notifications

### Scope and Relevancy of tests

1. **Coverage:**

The tests covered all critical aspects of the system related to detecting and capturing license plates, extracting the license plate text and issuing penalties through delivering SMS notifications. This includes testing the functionality of the traffic lights, cameras, ultra-sensors, Raspberry Pi control system, image processing algorithms, license plate recognition and penalty generation.

1. **Volumes:**

The tests considered varying volumes of traffic lights violations to ensure that the system can handle different scenarios effectively. This includes testing the system's performance and scalability in detecting and capturing license plates accurately.

1. **Complexity:**

The tests addressed the complexity of the system by evaluating its ability to handle different license plates extracting from the images and conditions in which the system finds hard time to perform its objective. This includes testing the system's robustness and reliability in detecting license plates in real-time and capturing the actual plates in challenging environments.

Therefore, the tests align with the project objectives which include the detection, capture, and penalty issuing for traffic lights violations. They focus on verifying that the system can accurately detect violations, capture the necessary license plates and penalties associated registered vehicle owners.

### Levels of tests

#### 1. Module Test

The module tests focused on testing the individual components or modules that contribute to achieving the system's objective. This included testing the functionality and accuracy of modules such as the traffic light detection module, image processing module, license plate recognition module, penalty generation module, and SMS delivery module. The module tests ensure that each component performs its specific tasks correctly and contributes to the overall objective of detecting and capturing license plates of the violating drivers.

1. **Integration Test.**

The integration tests aimed at verifying the seamless integration and communication between different modules of the system. This involved testing the interactions and data exchange between the modules to ensure that information flows correctly throughout the system. For example, it included testing the integration of the traffic light detection module with the image processing module, the integration of the license plate recognition module with the penalty issuing module, and the integration of the penalty issuing module with the SMS delivery module. The integration tests ensured that the modules work together harmoniously to achieve the objective of detecting violations and issuing penalties.

1. **System Acceptance Test.**

The system acceptance test focused on evaluating the system as a whole to ensure it meets the defined acceptance criteria and fulfills the intended objective. This involved conducting end-to-end tests that simulate real-world traffic scenarios to verify the system's capability to detect and capture vehicles violating traffic lights accurately. Additionally, it ensured the proper generation and delivery of penalty notifications through SMS. The system acceptance test validates that the system performs as expected, meets the specified requirements, and successfully achieves the objective of detecting violations and issuing penalties.

### Types of tests

1. **Input Tests**:

Input tests focused on validating the system's ability to handle different types of inputs effectively. In the context of the system, input test involved providing various traffic scenarios as inputs to test how the system responds to different license plate detection. These tests ensure that the system can accurately detect and process different types of input data related to license plates of the violating vehicles.

1. **Functionality Tests**:

Functionality tests assess whether the system functions correctly and performs the intended operations. In this case, functionality tests verified that the system accurately detects traffic violations, captures images or videos of the violating vehicles, extracts license plate information, issues penalty notifications, and sends SMS notifications to the respective vehicle owners. The tests ensured that the core functionalities of the system align with the objective of detecting and capturing traffic violations while issuing penalties through SMS.

1. **Boundary** **Tests.**

Boundary tests evaluated the system's behavior at the boundaries of its capabilities. For example, it involved testing the system's response to extreme or edge cases, such as vehicles violating traffic lights at high speeds or in challenging lighting conditions. The tests ensured that the system can or can’t handle and process data at the limits of its expected operational scenarios, identifying any potential issues or vulnerabilities in the system's performance.

1. **Performance Tests.**

Performance tests assed the system's performance in terms of speed, accuracy, and resource utilization. In the context of the traffic lights violations and penalty system, performance tests measured the system's response time in detecting and capturing violations, the accuracy of license plate recognition, and the efficiency of penalty generation and SMS delivery. These tests ensured that the system meets performance expectations and can handle the anticipated volume of traffic violations effectively.

### Sequence of tests

#### 1. Test Case: Red-Light Violation Detection

Test Procedure: Simulate a scenario where a vehicle runs a red light.

Test Data: a vehicle violating the red lights traffic and it’s captured by the system.

Expected Result: The system accurately detects the red-light violation, captures clear images or videos of the violating vehicle, extracts the license plate using tesseract and generates a penalty with the appropriate details of the owner if the exists.

#### 2. Test Case: Penalty Notification Delivery

Test Procedure: Verify the delivery of penalty notifications through SMS.

Test Data: Simulate sending penalty notifications to registered vehicle owners.

Expected Result: The system sends SMS notifications to the respective vehicle owners associated with the violations, delivering the penalty information accurately and in a timely manner.

#### 3. Test Case: Penalty Management

Test Procedure: Validate the system's penalty management functionality.

Test Data: Track penalties issued to violators and their corresponding records through the system’s dashboard.

Expected Result: The system correctly records and manages penalty information allowing easy retrieval of penalty records and tracking of violators who have been issued penalties.

### Configuration and calculation tests

1. **Configuration Test.**

Platform Compatibility: Verify that the system is compatible with the chosen hardware platform, such as Raspberry Pi and that all necessary components (traffic lights, pi-cameras, ultra-sensors) are properly connected and configured.

Network Integration: Test the system's ability to integrate with the network infrastructure, ensuring that it can send and receive data, including SMS notifications through the Africa’s Talking API.

Integration with Other Systems: Validate the integration of the system with any external systems or APIs required for functionality, such as license plate recognition or penalty management databases.

1. **Calculation Test.**

Known Input Validation: Provide specific input scenarios with predetermined outputs to validate the accuracy of calculations and algorithms used in the system.

Algorithm Verification: Ensure that the system's algorithms for detecting traffic violations, extracting license plate information, and generating penalty notifications produce the expected outputs for known inputs. This involves validating the accuracy and reliability of the system's calculations and algorithms.

## Precautions

### Anomalous conditions

Anomalies are unexpected or abnormal behaviors, conditions or events that deviate the expected or intended operation of the system. Therefore, we encountered some anomalies in the interaction between the system and the instruments used such as the traffic led light could sometimes fail to follow the pattern of the lights and also incase an image is identified and the text is not recognizing the system could fail however much the image contains the plate number.

### Precautionary steps taken

1. Raspbian Configuration.

Ensuring that the Raspbian operating system is properly configured for the system's requirements. This involves setting up the appropriate software packages, libraries, and dependencies, as well as configuring system settings for optimal performance and compatibility.

1. Device Compatibility.

Verifying the compatibility of the hardware devices used in the system, such as cameras, sensors, and Raspberry Pi, with Raspbian. This may involve installing device drivers, adjusting configurations, and conducting compatibility tests to ensure smooth integration and operation.

1. System Stability:

Monitoring the system's stability and addressing any potential anomalies or undesired operating conditions specific to the Raspbian environment. This includes managing system resources effectively, monitoring for system errors or crashes, and implementing appropriate error handling mechanisms.

# Chapter 5: Installation and system acceptance test

## Input files

1. Raspbian Operating System.

This is the primary file that needs to be installed on the Raspberry Pi's SD card. It contains the necessary components and configurations to run the Raspbian operating system.

1. System Configuration Files.

These files include various configuration settings specific to the Raspberry Pi, such as network settings, user accounts, system preferences, and hardware configurations. Examples of such files are `/etc/hostname`, and `/boot/config.txt`.

3.Python Scripts.

The system may consist of multiple Python scripts responsible for different functionalities, such as traffic light detection, license plate recognition and penalty issuing. These Python files contain the code for implementing these features.

4.Libraries and Dependencies

Depending on the specific requirements of the system, certain libraries and dependencies may need to be included in the installation media. These files ensure that the system has access to the necessary resources and functions to run properly. Examples may include the TensorFlow library for object detection, the OpenCV library for image processing, and MySQL for database connections.

## Supplementary files

**Readme Files**

These files provide instructions, guidelines, or additional information about the system. They typically include details on how to set up the system, install dependencies, configure settings, and troubleshoot common issues. Readme files are useful for users and developers to understand the system's functionality and usage.

License Agreements.

These files contain the licensing terms and conditions for the software used in the system. They specify the rights and limitations associated with using, modifying, or distributing the software. License agreements ensure compliance with open-source or proprietary software licenses and protect the intellectual property rights of the software.

Documentation.

This may include user manuals, technical guides, or API documentation that provide comprehensive information about the system's features, functionalities, and usage. The documentation assists users and developers in understanding how to interact with the system, utilize its capabilities, and integrate it into other applications if needed.

Table .: Checklist of the Installation and system acceptance test

|  |  |
| --- | --- |
| Topics | **Installation summary** |
| **Installation method**  Automatic or manual installation... | Automatic - installation kit located on the installation media  Manual - Copy & Paste from the installation media  Comments:  Since the raspberry pi operates on a linux OS, we used copy and paste to install the files and libraries. |
| Topics | **Installation summary** |
| **Installation media**  Media containing the installation files... | Diskette(s)  CD-ROM  Source disk folder (PC or network)  Download from the Internet  Comments:  The Raspbian OS is downloaded from the internet therefore an internet connection is required. Together with downloading software like SD card formatter the helps to prepare the sd card for the OS. |
| **Installed files**  List of (relevant) installed files, e.g. EXE- and DLLfiles, spreadsheet Add-ins and Templates, On-line  Help, etc. | * DBK files * Plg files * Opt files |

Table .: Installation Procedure Check

|  |  |  |
| --- | --- | --- |
| Topics | **Installation procedure** | Date / Initials |
| **Authorization**  Approval of installation in actual environment. | Person responsible:  NAKIBING BONNY | 28TH MAY  27023 |
| **Installation test**  The following installations have been performed and approved... | Tested and approved in a test environment  Tested and approved in actual environment  Completely tested according to test plan  Partly tested (known extent of update)  Comments: | 09/06/2023 –  NB  09/06/2023 –  NA  11/06/2023 –  OD  11/06/2023 –  KC |

# Chapter 6: Performance, servicing, maintenance, and phase out

## Service and maintenance

Due to the need for change in software equipment and in the environment, software has to continuously change in order for it to remain functional and be able to perform its tasks to accomplish a certain set of requirements.

To ensure proper functionality and reliability of the traffic lights compliance and penalty system, as a team we performed several tasks to make sure our system can be fully functional and efficient through capturing, license extraction and issuing of penalty to the violating driver.

For maintenance procedures, we performed regular inspection of the hardware components such as the sensors, resistors in the system, lights and the raspberry pi to ensure that they are functioning at full capacity. Software inspections of the raspberry pi such that it is always up to date. And before installing any other libraries, frameworks and other dependencies, we always first update the system through running “sudo apt-get update” command such that we are able to detect and fix bugs in our software and also recover from security patches and finally to enhance the performance of the system.

We also performed maintenance and training through the user manual. IT traffic officers were trained of how the system works and operates by providing step by step procedures for routine maintenance procedures, how to trouble shoot and problem and how to resolve the problem. This was to elevate the system performance and avoid issues of misuse of the system.

## Performance and Maintenance

Service and maintenance activities are crucial to keep the system functioning optimally. Regular inspections of hardware components (raspberry pi, sensor, pi camera), software updates (Raspbian, python, Yolov5), database maintenance, and data backups are necessary to prevent unexpected failures and minimize downtime. By performing these activities, the system will operate smoothly and ensure reliable performance.

Performance requirements are also significant in this phase. The system should meet specific performance criteria, which may include the maximum time taken to deliver an SMS after identification of the violating driver. Monitoring and optimizing the system's performance are essential to meet user expectations and maintain a satisfactory user experience. Performance testing and monitoring tools can be used to assess and improve system performance.

Effective user support is another critical aspect during this phase. We will help the IT traffic officers by providing technical assistance, troubleshooting guidance, user training, and prompt response to inquiries or issues raised by IT traffic officers towards the system hardware components. User support ensures that users can effectively utilize the system, resolves problems in a timely manner, and enhances overall user satisfaction.

The causes for incorporating changes into the system raised due to system performance to visualize the violating license plates and modification in the system components such as the pi camera and the sensors to maintain full functionality. And upgrades should be incorporated into the system through updating the software remotely using the SSH to access the raspberry pi control system.

Since the data is on the MySQL database on the cloud, in terms of moving from the old system to a new system, we shall provide an analysis on the schema and design of our existing SQL database to understand the tables and relationships and datatypes. Then, during the transfer from the old system to the new system, we shall create a strategy to migrate our data to the new system. Then extract the data that needs to be moved to the new system through executing SQL queries. And finally, transfer and map the data to the new system using the SQL query.

Table .: Performance and maintenance details

|  |  |  |
| --- | --- | --- |
| Topics | **Performance and maintenance** | Date / Initials |
| **Problem / solution** | When the number of license plates identified are note known in the database with driver details, there will be need to always carryout hardware checks especially on the raspberry pi camera to check the efficiency of vision and also trained and update the model further more with more data. | 09/06/2023 –  NB  09/06/2023 –  NA |
| **Functional maintenance** | If there is upgrade in the raspberry pi version, hardware needs to be changed to reflect fast performance of the system. Therefore, regular software and hardware checks need to be performed onto the system through making new installations and upgrades. | 11/06/2023 –  NB  11/06/2023 –  KC |
| Topics | **Performance and maintenance** | Date / Initials |
| **Functional expansion and performance improvement** | List of suggestions and requests, which can improve the performance of the computer system. eg   * Use of more advanced cameras that have a wide coverage view and clear. * Use of coral to increase the speed and functioning of the raspberry pi camera. | 11/06/2023 |

## Chapter 7: Conclusion and Recommendations

In conclusion, the Traffic Light Compliance and Penalty System has been successfully developed to detect and capture vehicles violating traffic lights and issue penalties through SMS notifications. Throughout the development process, various inspections and tests were conducted to ensure the system's quality, functionality, and compliance with requirements.

The system has demonstrated its effectiveness in accurately detecting violations, capturing license plate information, and generating penalty notifications. It has proven to be reliable in delivering SMS notifications to registered vehicle owners associated with the violations. The integration with the Raspberry Pi control system and other components has been seamless, providing a robust and efficient solution. However, there are some challenges in spend in capturing that need to be addressed using better hardware platform such as raspberry pi 4 and coral to speed up functionality.

**Recommendations.**

Based on the successful implementation of the Traffic Light Compliance and Penalty System, the following recommendations are made:

**Deployment and Operational Considerations**.

Prior to deploying the system in a live traffic environment, it is recommended to thoroughly evaluate and test its performance and scalability in the real world. Conducting pilot tests in a controlled setting can help identify any potential challenges and fine-tune the system for optimal operation.

**Continuous Monitoring and Maintenance.**

Implementing a comprehensive monitoring system to continuously monitor the performance and health of the system. This will help identify any issues or anomalies promptly and allow for proactive maintenance and troubleshooting. Regular software updates and patches should be applied to ensure the system remains secure and up to date.

**User Support and Training.**

Providing a comprehensive user support and training to the IT traffic officer. This will enable them to effectively utilize and manage the system, handle penalty management, and address any technical issues that may arise. Clear documentation and user guides should be made available for easy reference.

**Data Security and Privacy.**

Ensuring strict adherence to data security and privacy regulations, including secure storage and transmission of captured images and personal information. Regular audits and risk assessments should be conducted to identify and address any potential vulnerabilities or risks.

**Manuals Table of Contents**

[1. Appendix A: User Manual 57](#_Toc140497584)

[1.1 Installation and Setup: 57](#_Toc140497585)

[1.2 Traffic Monitoring: 58](#_Toc140497586)

[1.3 Detection and Capture: 58](#_Toc140497587)

[1.4 Monitoring Performance. 59](#_Toc140497588)

[2. Seeking Help: 60](#_Toc140497589)

[3. Sources 61](#_Toc140497590)

# Appendix A: User Manual

## Installation and Setup:

Ensure that the Raspberry Pi control system, traffic lights, cameras, and ultra-sensors are properly set up and configured according to the system's installation instructions.

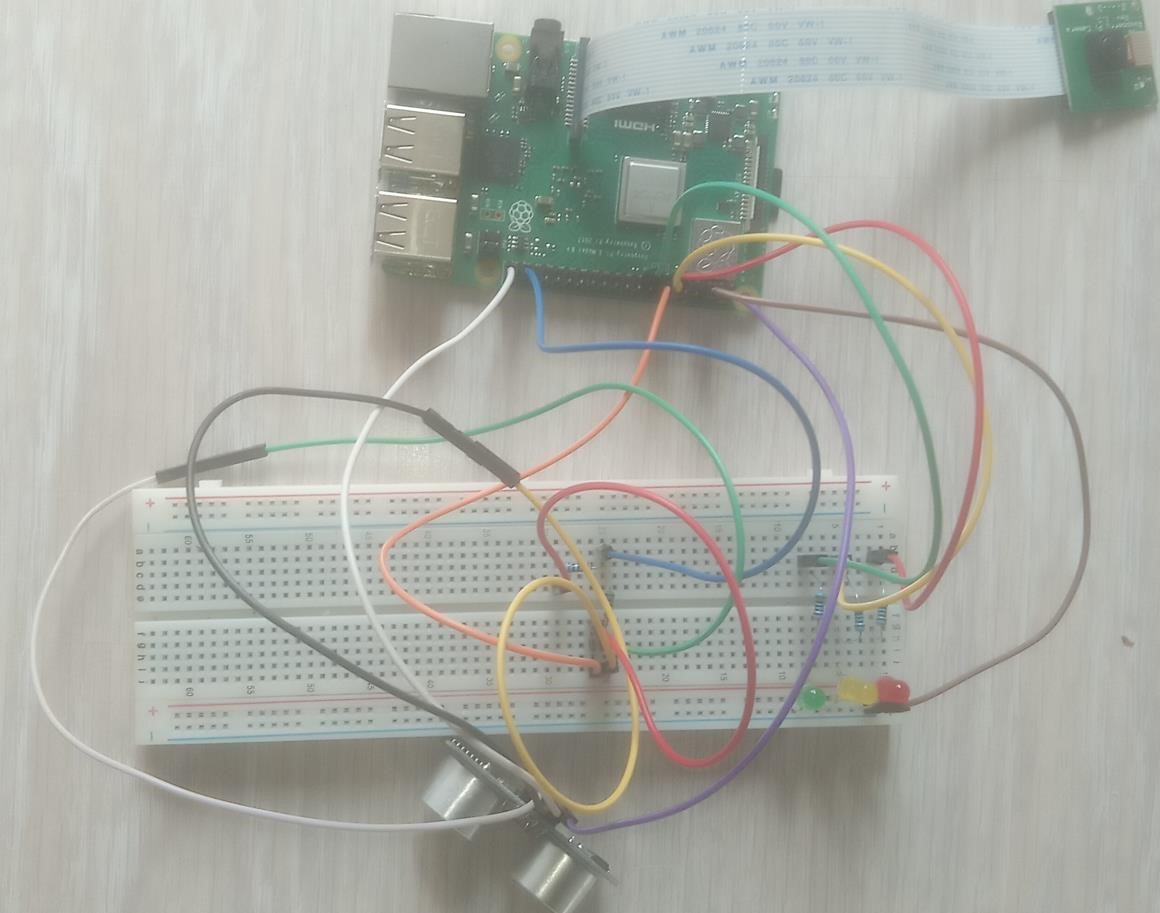


Figure .: Components inter-connected to Each other System

Activation:

Power on the Raspberry Pi and ensure that the system software is running. Verify that the system components are properly connected and communicating.



Figure .: Connection To Power Supply

## Traffic Monitoring:

The system will continuously monitor the traffic lights and capture images or videos when a violation occurs. It uses image processing algorithms to detect license plates and record relevant information.

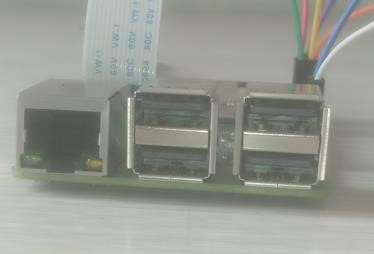


Figure .: insert USB Mouse and Keyboard Violation

## Detection and Capture:

When a vehicle violates a traffic light, the system will detect the violation based on predefined criteria. It will capture clear and identifiable images or videos of the violating vehicle, including the license plate using the pi camera.



Figure .: PI camera

## Monitoring Performance.

To monitor the performance and accomplishing of tasks, connect to the same network with system and access the user interface of the system through SSH connection for only authorized traffic personnel will be allowed to login.

Connect the HDMI to the screen and view the raspberry pi desktop

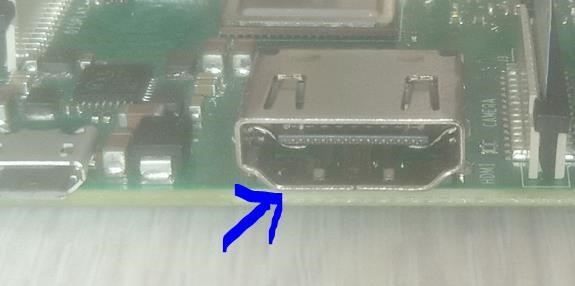
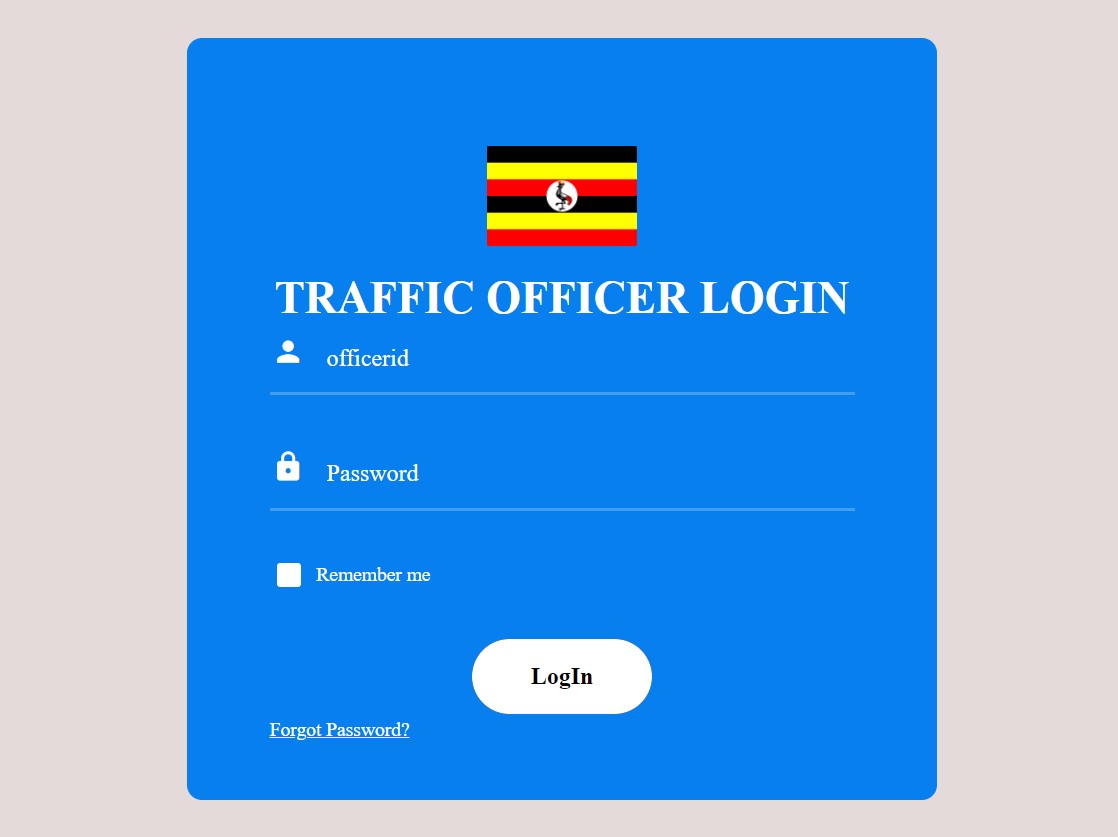


Figure .: Connection of the HDMI to the Raspberry pi and Screen



Enter officer Id

Enter password

Press to login

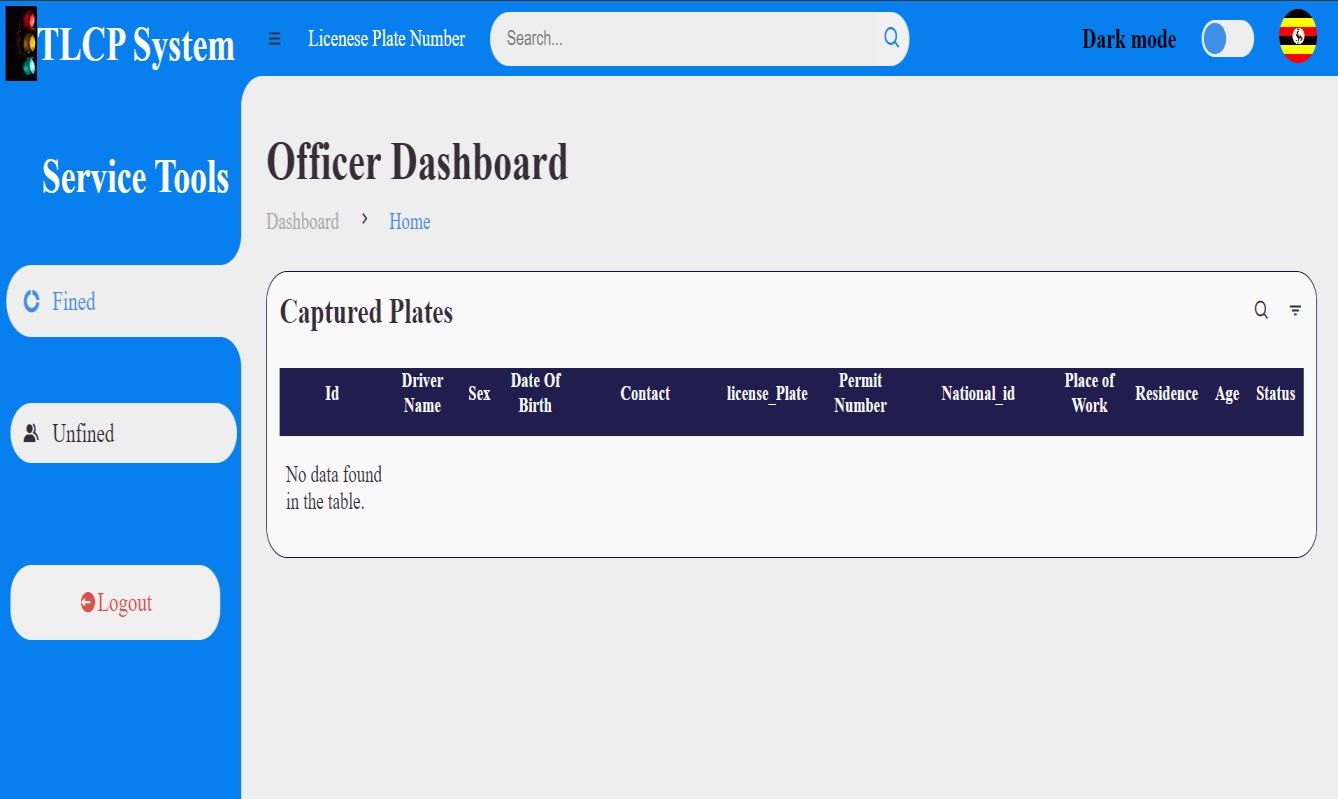
Press for

password

recovery

Figure .: showing the login buttons

:



View for captured plates

Log out from system

Search for numbers

Figure .: Screen showing Display of the system.

# Seeking Help:

If you require assistance with the Traffic Light Compliance and Penalty System, you can reach out to the following resources:

User Manual: Refer to the system's user manual for detailed instructions on installation, setup, and usage. The manual should provide step-by-step guidance on system operation and troubleshooting.

Online Support: Visit the system's official website or online support portal, knowledge base articles, and troubleshooting guides. These resources can provide answers to common questions and help address technical issues.

Contact Support: If you encounter any technical issues or need personalized assistance, contact the system's support team. They can provide direct support, guidance, and further troubleshooting steps.

Name: Nakibinge Bonny Name: Katembeko Christopher

Contact: 0755665544 Contact: 0770829724

Role: Developer. Role: Project Co-Ordinator

Name: Nsubuga Alpha Name: Otim Derrick Mulungi

Contact: 0705310594 Contact: 0786604400

Role: Project Manager Role: Developer

# Sources

**Blog website:** <https://sites.google.com/view/bse23-14/home>

**Git hub:** https://github.com/bo-nny/Traffic-Lights-Compliance-And-Penalty-System.git

|  |  |  |
| --- | --- | --- |
| **Final approval for use** | | |
| Identification: | |  |
| Responsible for validation: | |  |
| Remarks: | | |
| Date: | Signature: | |