MAKERERE UNIVERSITY

Project Title: Automatic Highway Surveillance System

By

BSE 21-1

TYPE OF SYSTEM: **EMBEDDED 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. RUTH MBABAZI MUTEBI

Department of Networks

School of Computing and Informatics Technology, Makerere University

Supervisor email,rmbabazi@cit.ac.ug

JAN 2022

Declaration

We, group BSE 21-1, hereby declare that the work presented is original and have never been submitted for an award to any university or institution of higher learning

	Names	Registration Number	Student Number
1	NYOMBI NELSON	17/U/9420/EVE	217013671
2	MWEBAZEKENNETH	17/U/6872/EVE	217004234

Approval

This project report titled Automatic Highway Surveillance System has been submitted for examination with my approval as the supervisor of group BSE 21-1.

DR. RUTH MBABAZI MUTEBI

Department of Networks

School of Computing and Informatics Technology;

College of Computing and Information Sciences,

Makerere University

Signature: ..... Date: 12 Jan 2022

Supervisor

BSE 4200 Project Submission Letter

Date: 03/02/2022

Dean, School of Computing and Informatics Technology
Makerere University

Dear Madam

RE: Project Group Number B21-1

Following the Project Presentation held on the 13/01/2022, this is to confirm that the above group has effected all the minor corrections as recommended by the panel, to my satisfaction.

I therefore recommend the group for the award of the Degree of Bachelor of Science in Software Engineering of Makerere University.

Yours truly,

Mary Nsabagwa

Internal Examiner

Dedication

Uganda Police, Department of traffic.

Acknowledgement

As BSE 21-1 group, we would like to place on record our profound sense of gratitude to our Supervisor Dr. Ruth Mbabazi Mutebi from the Computer Science Department College of Computing and Information Sciences Makerere University Kampala for the support, guidance, and providing us with a conducive environment for learning and successfully carrying out our final year project.

Special gratitude also goes out to Dr.Mary Nsabagwa for the continued guidance and assistance in teaching and mentoring us into the real world experience of the IT profession.

In a special way, we would like to appreciate the contribution of our parents in this journey of education. We pray that God rewards them for their effort that has been financial, moral and psycho-social.

Abstract

This document details the software design document, System implementation, and testing and validation report for the Automatic Highway Surveillance System.

The SDD describes the architecture and system design of the Automatic Highway Surveillance System that will guide the system implementation. It covers the representation of software components, interfaces and data necessary for the implementation phase. The purpose of this document is to act as a guide and reference for code development and for supervision purposes. It covers system architecture, data design, component design and human interface design.

System implementation, and testing and validation report captures the background and scope of the project, system specification, version of requirements and version control. It details how the system was implemented, the programming languages used, design outputs and the testing phase details.

List of figures.

List of tables

Blog site: https: //sites.google.com/view/automatic-highway-surveillance/home

Github:https://github.com/AHSS21

SOFTWARE DESIGN DOCUMENT

Table of contents

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of the Automatic Highway Surveillance System that will guide the system implementation. It covers the representation of software components, interfaces and data necessary for the implementation phase.

The purpose of this document is to act as a guide and reference for code development and for supervision purposes.

It covers system architecture, data design, component design and human interface design.

1.2 Scope

Automatic Highway Surveillance System is a traffic control software system that intends to curb over speeding on Highways by simplifying the way over speeding vehicles are monitored and followed up. The system shall monitor the road twenty-four hours, seven days a week for all occurrences of over speeding cars. The traffic officer on duty shall be able to view over speeding occurrences that happened even while they were off the road. This automatic aspect shall allow the traffic police to follow up on those offensive cars involved in over speeding as captured in the database.

1.3 Overview of the document.

This document introduces you to the purpose of the document, scope and its intended audience. It then introduces you to the overall system overview of the project.

Section 3 contains the system architecture which in turn details the architectural design, decomposition description and rationale the design decision is based on.

Section 4contains the data description and data dictionary, component design, human interface design that spells out the overview of user interface, screen images, screen objects and actions.

This document follows the IEEE convention.

2. SYSTEM OVERVIEW

This is the first version of Automatic Highway Surveillance System (AHSS). It comes in response to the need of simplifying how over speeding crimes are monitored and followed up along highways in a call to curb over speeding as one of the commonest traffic offenses. Over speeding being one of the major causes of road accidents, there has been a need for an automatic system to monitor and archive traffic offenses while they happen live and clear.

The system shall detect the passing vehicle, measure, record, and grade its speed. Once the speed of the target vehicle is above the accepted speed, the number plate of the car shall be captured using the system's camera and together with the offensive speed, this data will be sent into the database.

In case the traffic officers are on duty (Daytime), they will be able to review this data, stop the car, and issue a speed ticket to the driver. Otherwise (night), data stored will be used for follow up once the officers return to duty.

The system shall allow traffic officers to retrieve recent records about offenders and the offenses that have occurred. The system then displays a weekly report of over speeding occurrences. The traffic officers are able to print a weekly report of over speeding offenses. The system shall also allow the Head of Traffic to add and allocate traffic officers to different highways. The system shall allow traffic officers to be registered and also log in.

3. SYSTEM ARCHITECTURE

Architectural Type; Layered architecture.

AHSS uses this type of architecture given that it has three layers i.e.,

a) The layer of sensors that interacts with the cars.
b) The layer for processing the data. (business logic)
c) The presentation layer for displaying the data at the interface.

3.1 Architectural Design

Modules

1. Roadside speed recorder

➢ Captures automatic speed at that time.

➢ It sends a receipt containing details of the car speeding offense into the database which includes the speed of the over speeding car in km/hr., screenshot of automobile number plate and the time when the offense occurred.

2. Interface

➢ Displays data from the road side speed recorder

➢ Creates monthly, weekly and daily traffic reports.

➢ Display traffic history

➢ Analyses traffic data for simple statistical models

➢ This information can therefore be used for issuing fines to traffic offenders

Relationships

Use case diagram

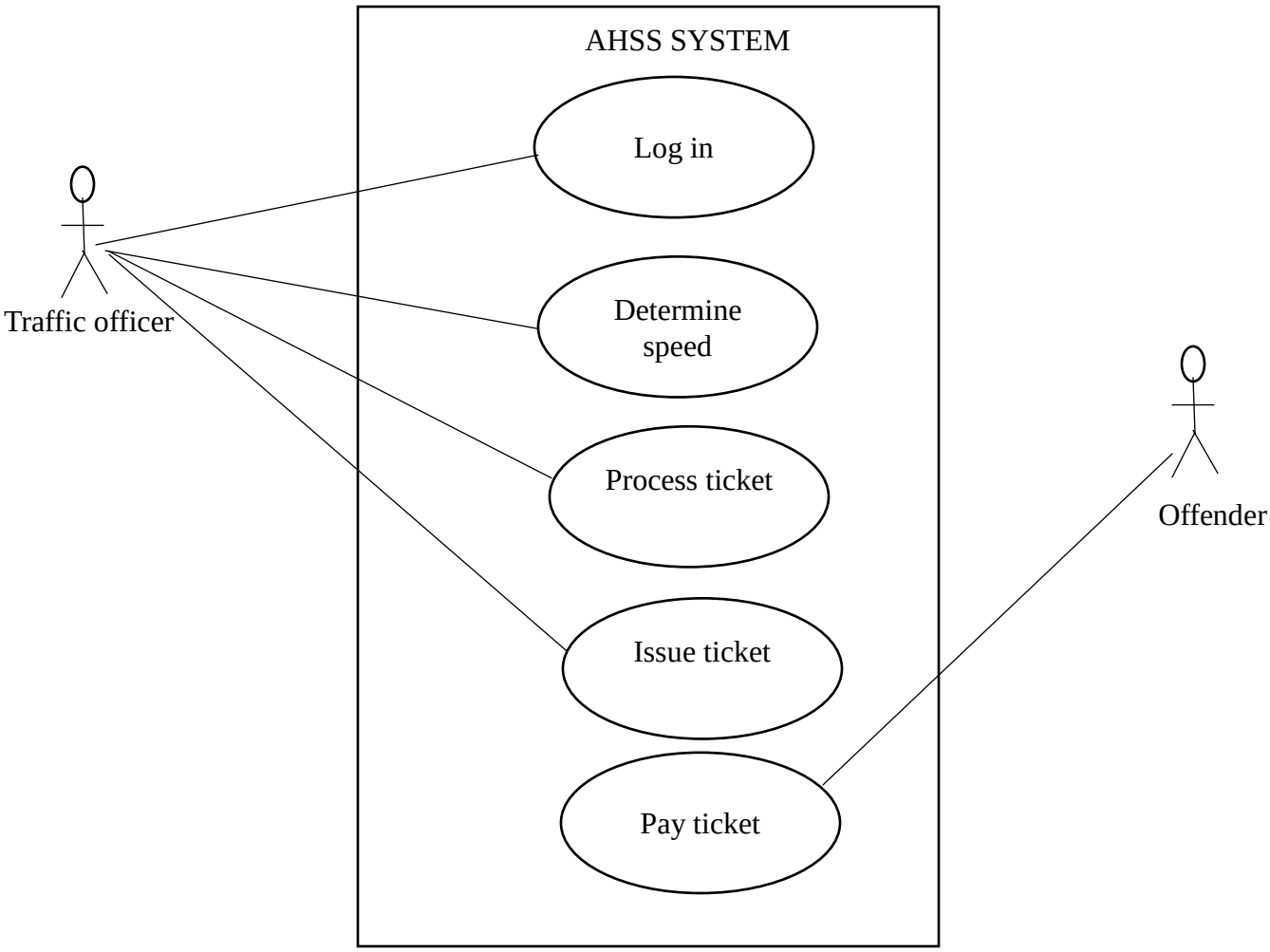

Figure 1 Use case diagram

The Traffic officer logs into the system. The speed of the incoming car is determined by the sensors along the road. If the speed is above the one mandated by the traffic police department, a ticket is processed. The ticket is then issued to the offender who then pays it.

Sequence diagram

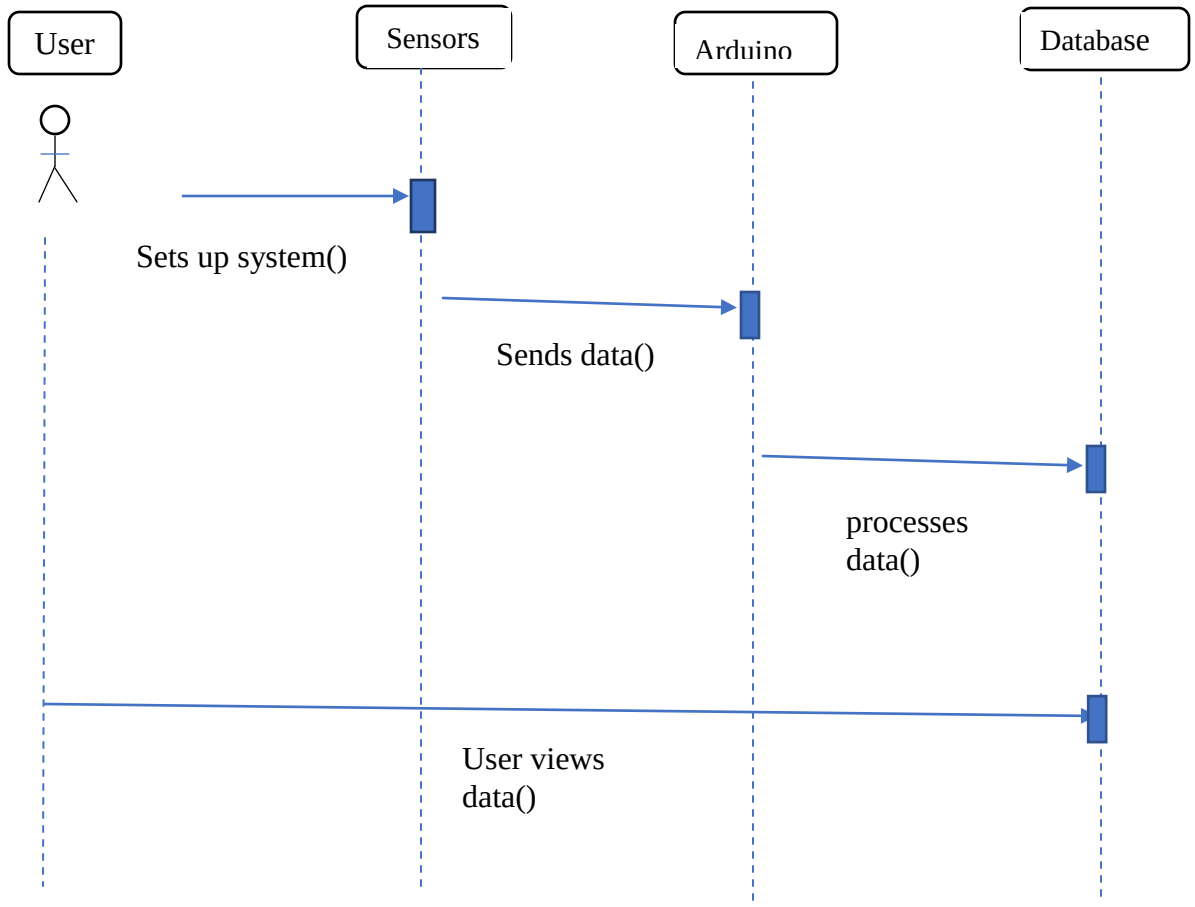

Figure 2 Sequence diagram

SYSTEM OVERVIEW DIAGRAM

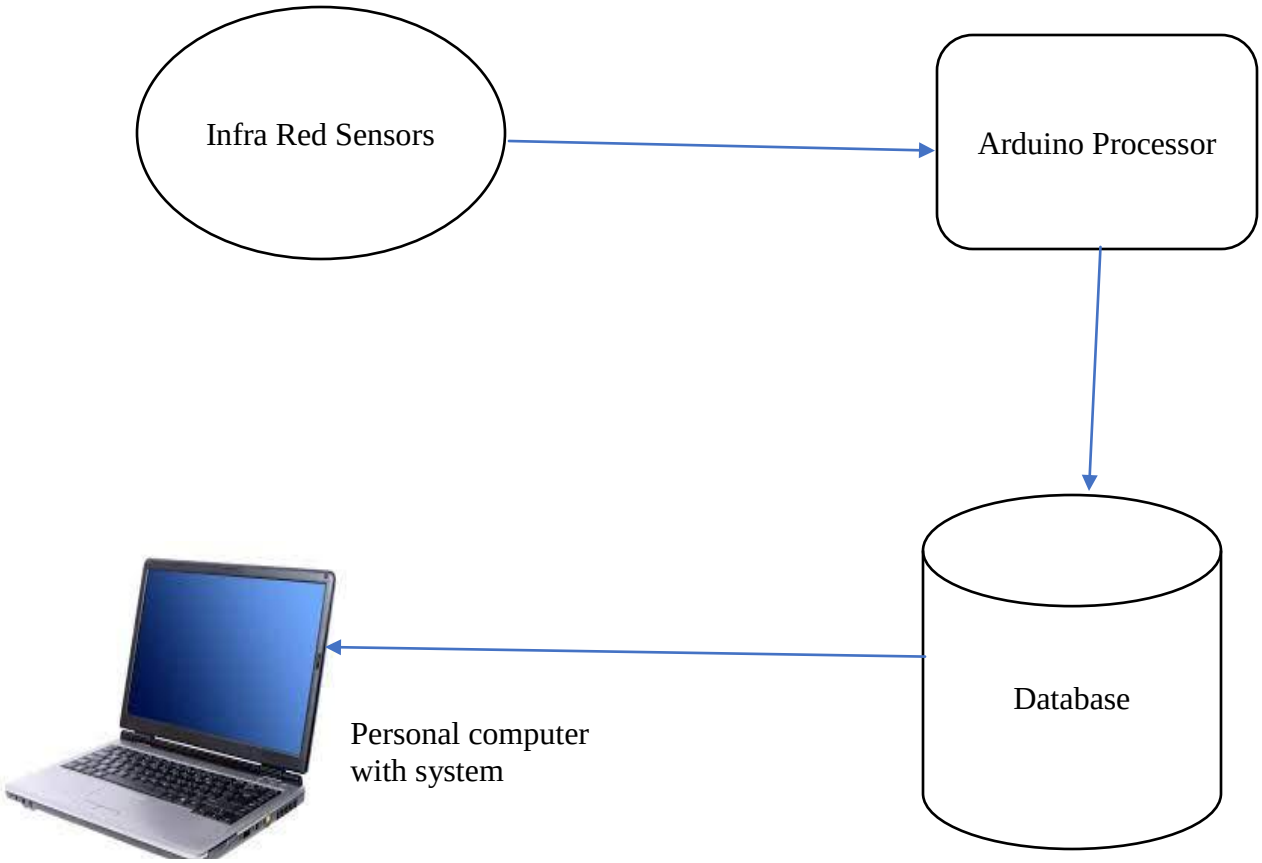

Figure 3 System overview

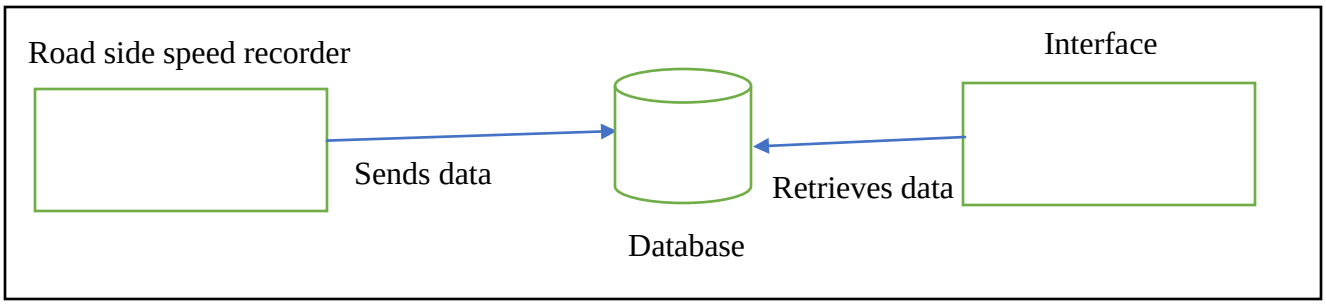

Figure 4 Relationship between recorder and system interface.

Figure 3 is a simple descriptive diagram showing the relationship between the roadside speed recorder, database and the user interface. The data is sent to the database by the roadside speed recorder and thereafter retrieved by the interface for analysis and modeling.

Relationship between speed recorder and database.

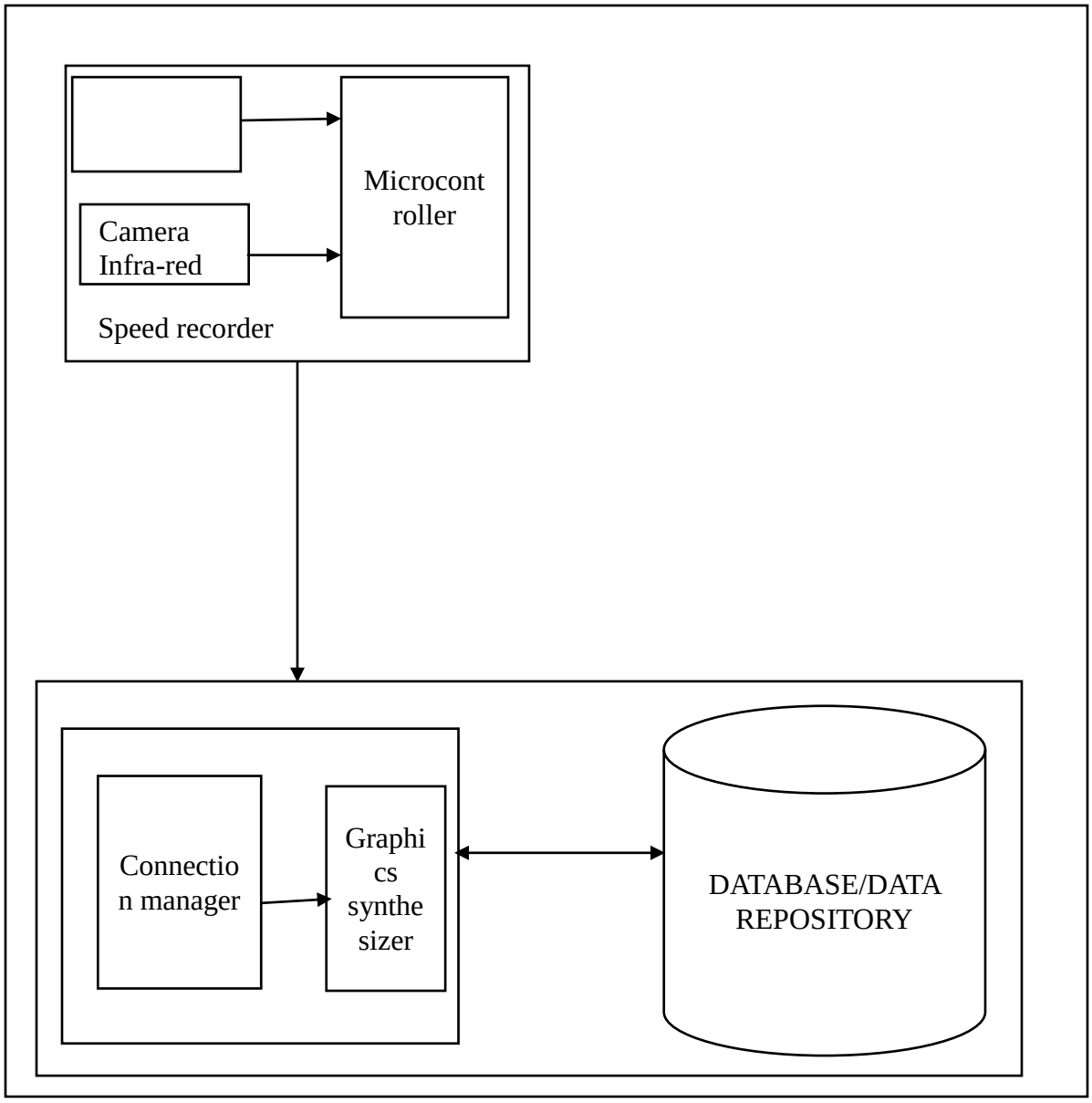

Figure 5 Relationship between speed recorder and database.

3.2 Decomposition Description

Speed recorder

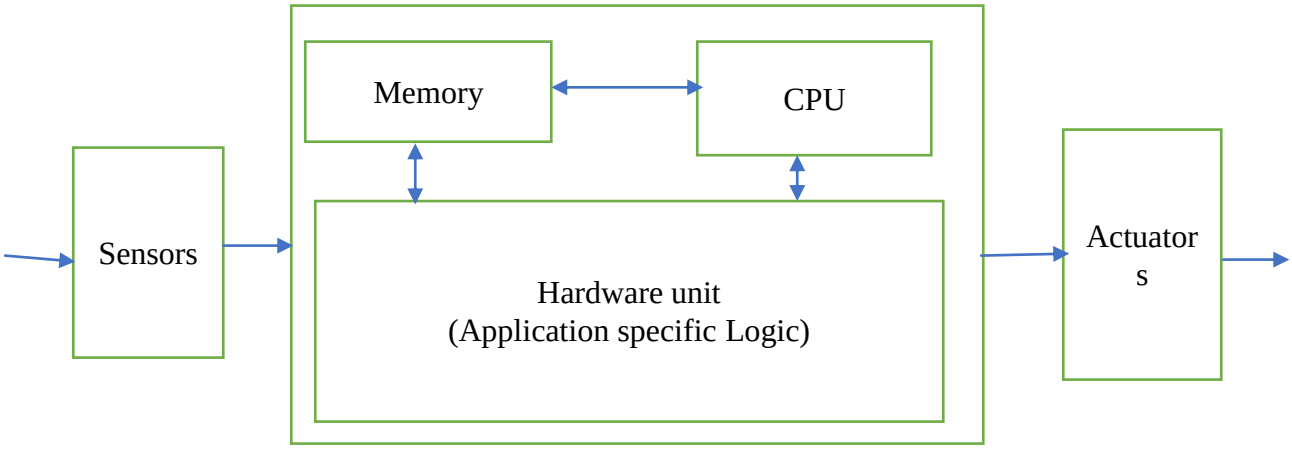

Figure 6 Pictorial representation of the speed recorder.

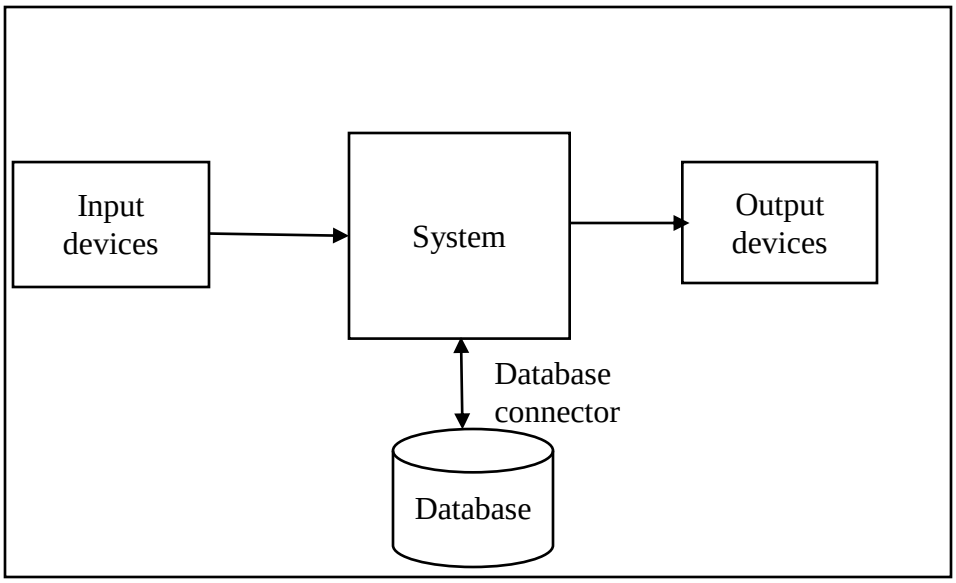

Figure 7 Decomposition of the system.

Process flow

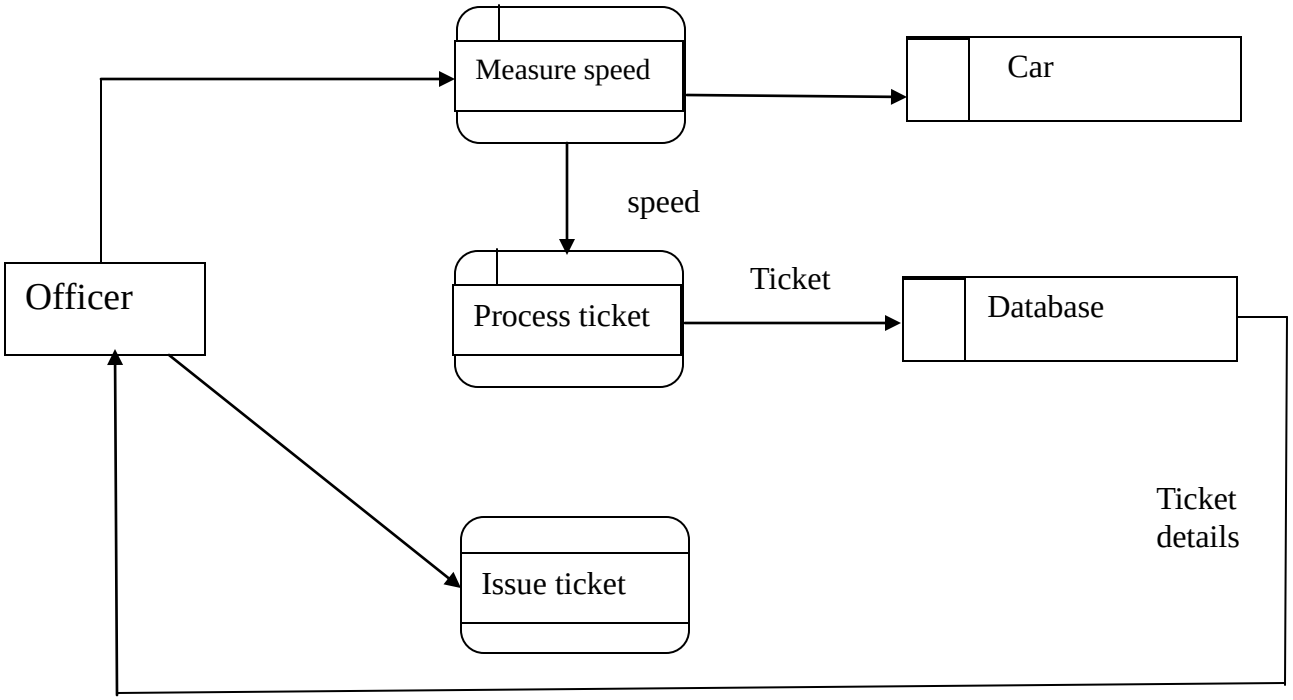

Figure 8 Process flow of the system.

Flow of events in the system

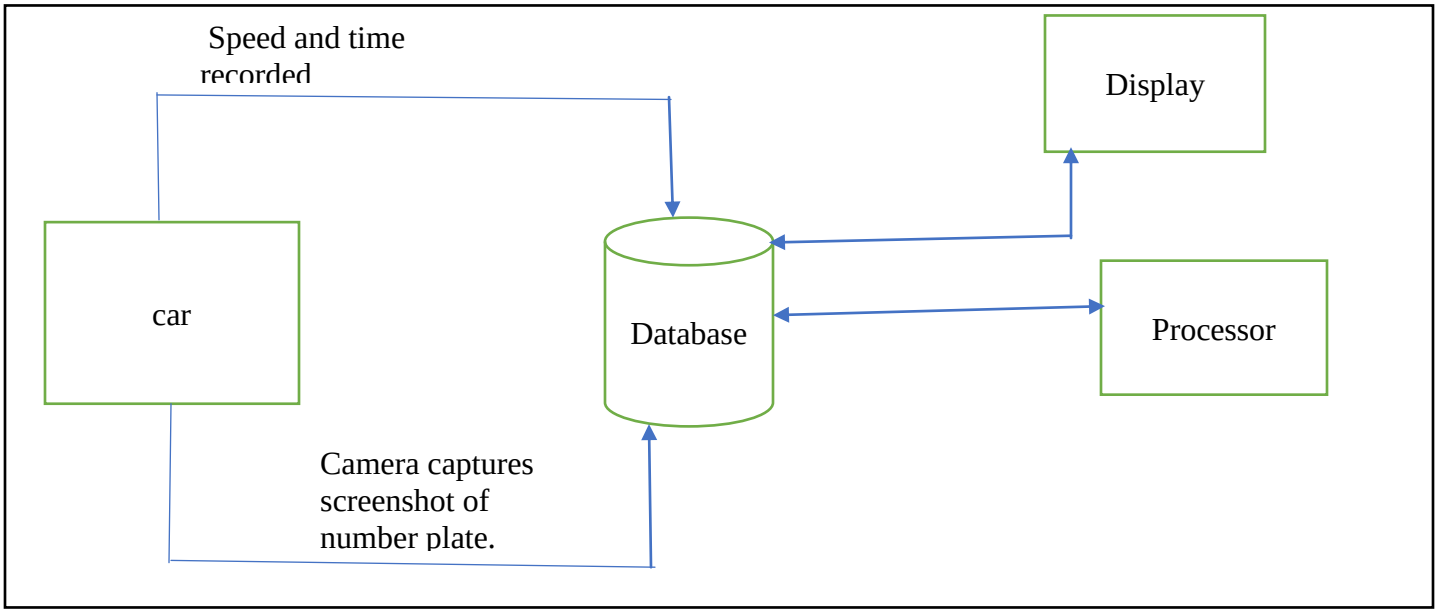

Figure 9 Event flow of the system.

Camera captures screenshot of number plate

Sensor- The sensors capture the car speed.

Memory /Database – the information is sent into the database for processing by the core processor.

User Interface – Once data is processed, it is available for manipulation, analysis and viewing by the user

3.3 Design Rationale.

Architecture choice: Layered Architecture

Rationale

a) The layer of sensors that interacts with the cars.
b) The layer for processing the data. (business logic)
c) The presentation layer for displaying the data at the interface.

Major issues that may arise

- Incompatible server capacity can slow down causing a performance bottleneck.
- Maintenance maybe demanding and expensive.

4 DATA DESIGN

4.1 Data Description

The Data is stored in tables. Numeric data is stored as integers. Texts are stored as character.
Time and date are stored as timestamp. Images/screenshots are stored as BLOB

Data items and their attributes

The data items in the AHSS are car, officer, Offender and offense. The attributes of car are its number plate, its speed and the date on which the speed was captured. Under Officer, the attributes are their user name, password, Identification number. Under Offender are his first name, last name, their identification, penalty committed and the penalty status. Under Offense the attributes are the speed of the car, Highway on which he was, the Ticket number and the time the offense happened.

- Car
  +numberplate
  +speed
  +date captured

- Officer
  +username
  +password
  +idnumber

- Offender
  +fname
  +lname
  +Officer_Id
  +Penalty
  +Penalty_status

- Offense
  +Speed
  +Highway

+TicketNumber

+Time

Entity Relational Diagram.

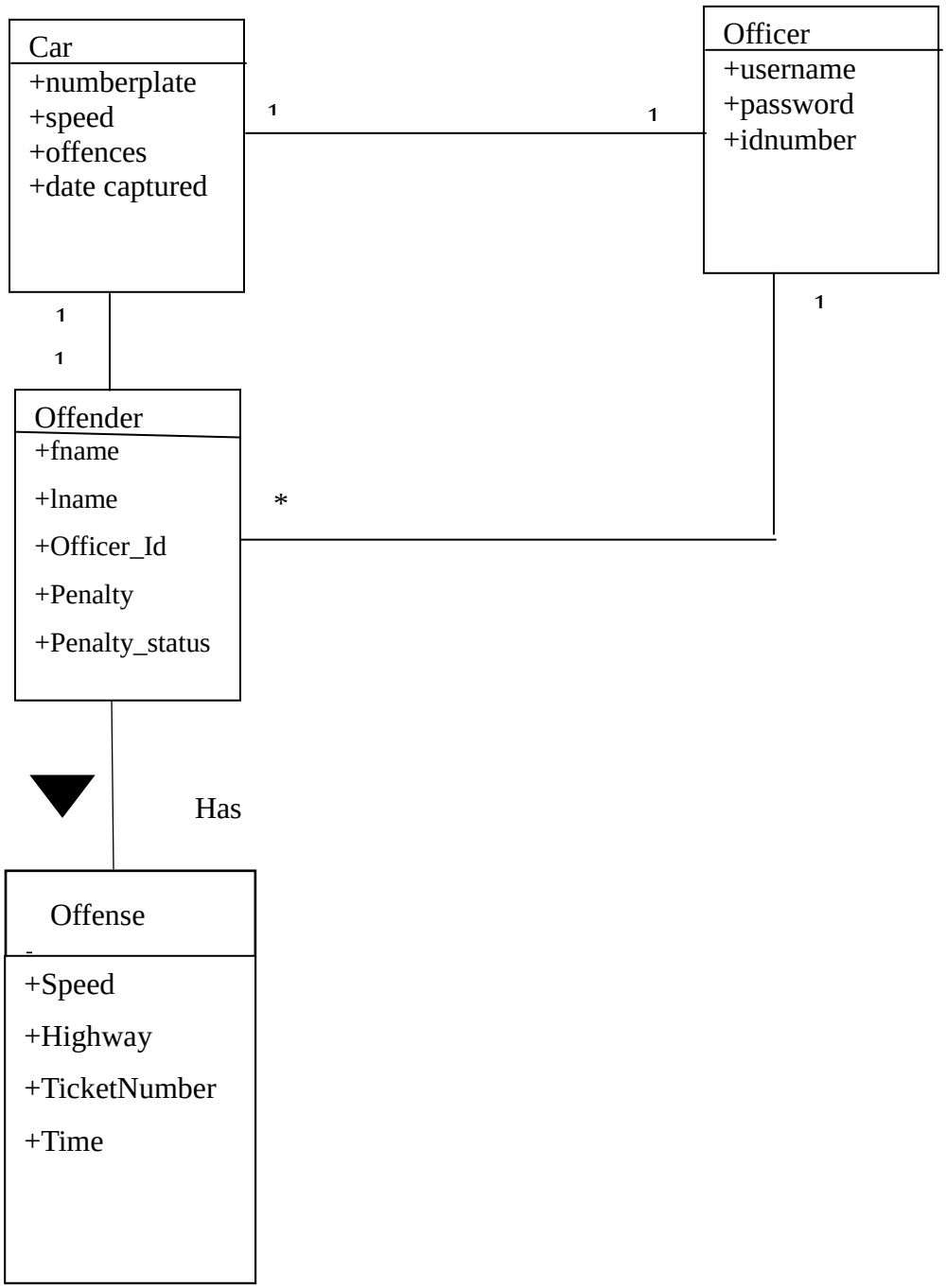

Figure 10 Entity relationship diagram.

4.2 Data Dictionary

Table 1 shows entities , their attributes and data types

Entity	Attribute	Type
Car	 +numberplate +speed +datecaptured	 VARCHAR INT TIMESTAMP
Officer	+username +idnumber	VARCHAR INT
Offender	+name +offenderfine	VARCHAR INT
Camera	+imageId	VARCHAR
Sensor	+speed	INT
Offense	+Speed +Highway +TicketNumber +Time	INT VARCHAR INT TIMESTAMP

5. COMPONENT DESIGN

5.1. Pseudo code

Algorithms

Sensor

Begin

Numeric nNum1, nNum2

Fire radio ray nNum1;

Receive reflected radio ray nNum2

IF nNum1≠ nNum2

Calculate speed of the car

Activate camera screenshots

Record the speed to the database

Else

Do nothing

End

Nature of sensors

Infra-Red Sensors are sensors that allow the device to detect moving vehicles. They can detect distant objects and determine their position and speed of movement.

Camera

Begin

Numeric imageID

Take screenshot

Save imageID to the database

End

UI

Begin

Numeric password, username

Input password

Input username

IF password& username ==TRUE

Display Dashboard

Else

Display "Wrong login, try again

end

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The system shall assist users to perform the following functionalities as specified below;

Login

The system shall allow authorized users to log into their respective accounts after providing valid login credentials which include, User Id and password.

Adding new User

After Logging in, the system shall enable the Head of Traffic police to add new users and assign them highways where they shall be operating from.

Edit profile

Successfully logged in users shall be able to edit their profiles under the profile tab.

Create reports

The system shall allow users to view and create new reports either monthly, daily, or weekly reports as per users' preference. All this shall be found under the Reports tab.

View previous offenses

The system shall allow users to view pervious traffic offenses under the history tab. This data shall then be used to calculate the occurrence of traffic offenses in a day.

Issue tickets

The system shall allow logged in users to view previously issued tickets and also create new ones. Tickets with a penalty shall be issued to traffic offenders and expected to serve it. Un-served tickets shall have a stamp (Pending penalty)

Check roads

The system shall allow users to view a list of roads/highways and the offense occurrence on them. This data shall be used to generate the statistics of roads on which traffic offenses are rampant.

See over speeding as it occurs

The system shall send traffic alerts to users in real time while traffic offenses are being committed.

Feedback

Users shall be able to seek online help and assistance, plus submitting their feedback.

6.2 Screen Images

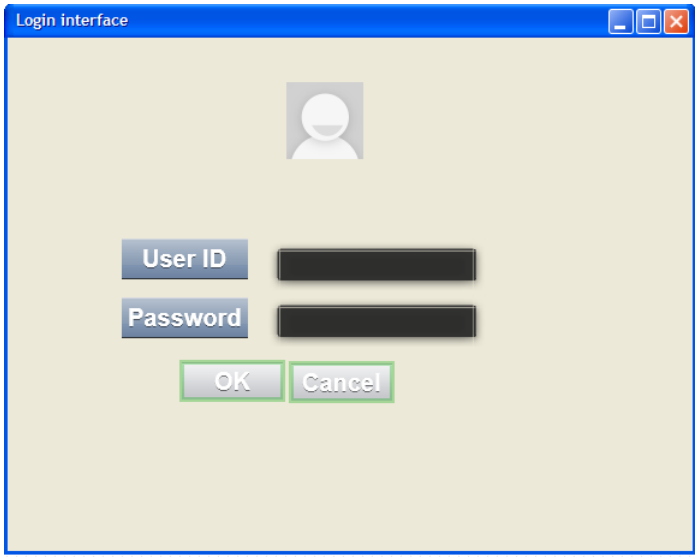

Figure 11 Login interface of the system

Figure 12 Head of Traffic screen and how he shall add new users

Figure 13 Screen where users shall be viewing and editing their profiles.

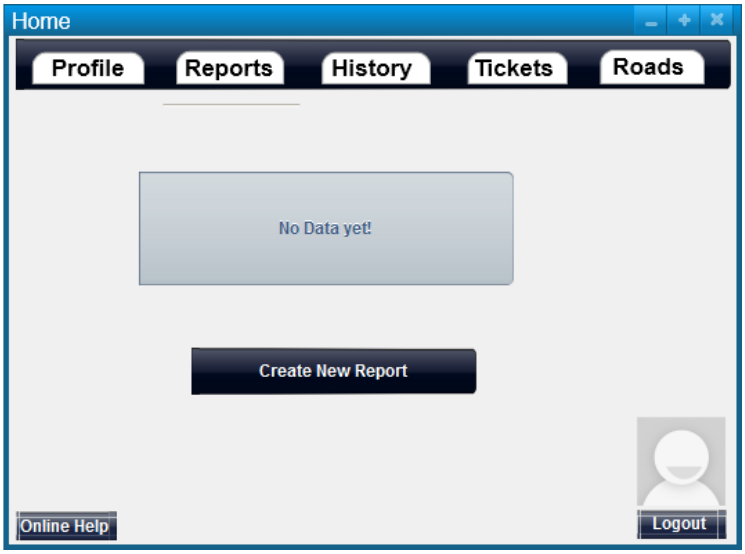

Figure 14 Screen where users shall view and create new traffic reports.

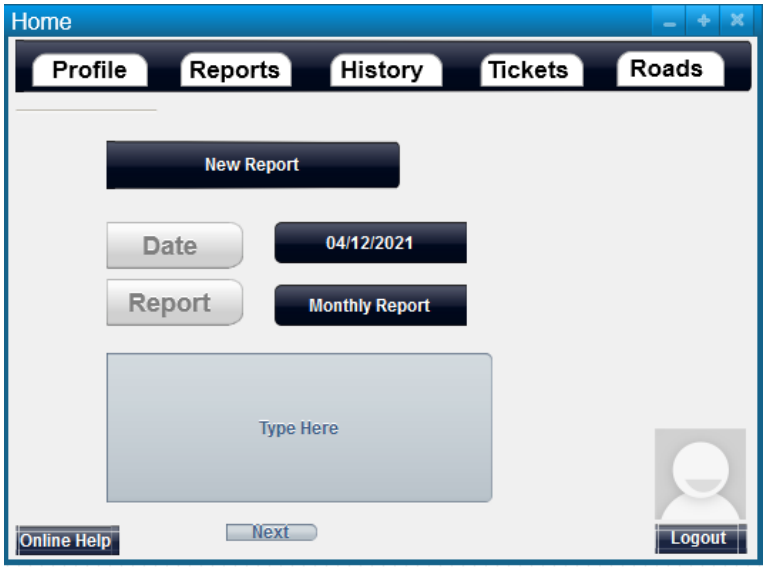

Figure 15 How user shall create traffic reports

Figure 16 Screen where users can view recent offenses under the history tab.

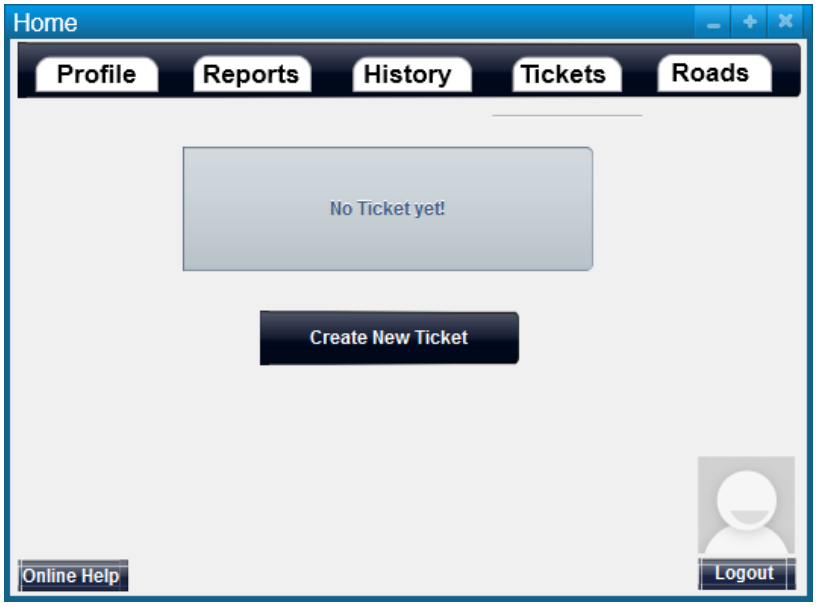

Figure 17 Screen for recent tickets and a button that enables users to create new tickets

Figure 18 How a user can create a new ticket for over speeding and the issue of a penalty.

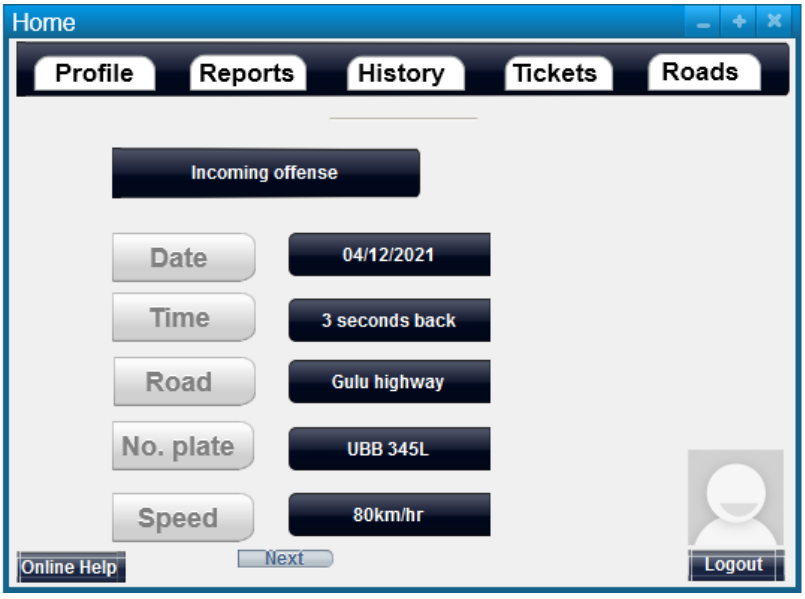

Figure 19 Screen for an incoming offense in real time.

.

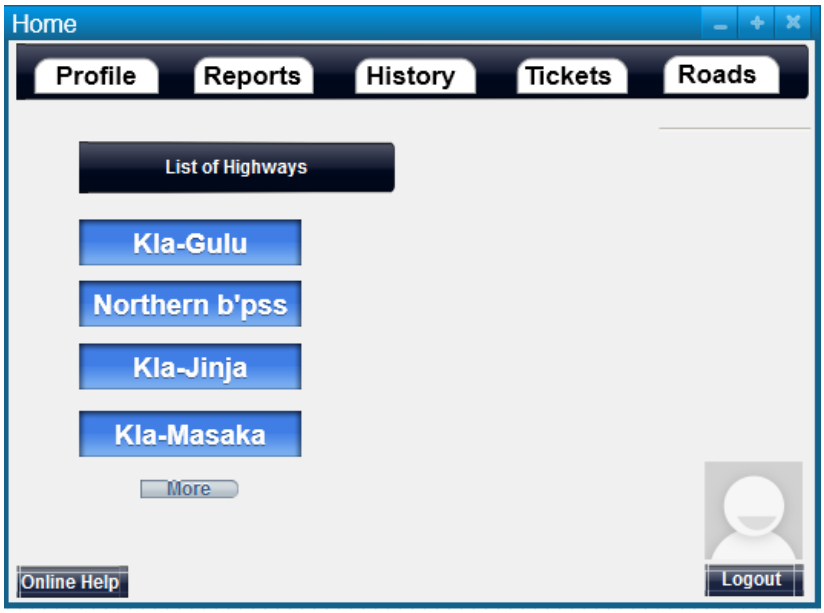

Figure 20 List of highway roads where traffic officers are usually assigned.

6.3 Screen Objects and Actions

1. Login Screen

User Name

User name can be ranged from 6 to 20 letters (numbers), as the industry standard. No special characters and space. The users will use their user ID as their user name for this system.

Password

Password can be ranged from 6 to 20 letters (numbers), as the industry standard. No special characters, space.

OK

If the users enter the right user name with the matching password, it will immediately take them to the main interface.

Cancel

If the user wishes to exit the program, hit the "Cancel" button.

2. Main Interface

- **Menu Items**

The following shows what is included in the main menu the main menu:

- Profile
- Report
- History
- Tickets
- Roads

Profile

- Edit profile
- Add new user

Reports

- Print traffic Report
- Print Blank Checklists
- Efficiency Report

History

- Recent offenses

Tickets

- Recent tickets
- Create new tickets

Roads

- List of roads

7. REQUIREMENTS MATRIX

Functional Requirements

AHSS-01: The system shall allow the traffic officer to log into the system

AHSS-02: The system shall disallow invalid details by the traffic officer.

AHSS-03: The system shall capture car speed and number plate in real time.

AHSS-04: The system shall calculate car speed.

AHSS-05: The system shall send car number plate and speed to the database.

AHSS-06: The system shall allow users to retrieve information from the database.

Table 2 Requirements matrix table.

Functional Requirements	**Roadside sensors**	**Processor**	**System interface**	**Priority**
AHSS-01			✓	High
AHSS-02			✓	High
AHSS-03	✓			High
AHSS-04		✓		High
AHSS-05	✓			High
AHSS-06			✓	High

REPORT

Table of contents

Chapter 1: Introduction

1.1Background and scope of the project

AHSS is a traffic control software system that intends to simplify how over speeding is being monitored by the traffic police in a call to curb over speeding crimes along highways. The system shall monitor the road twenty-four hours, seven days a week for all occurrences of over speeding cars. The system will then log the database with each over speed occurrence that shall be available for follow up and issuing of penalty to the victims.

It is a call to solve the old way of using speed guns that require traffic officers to be staged along highways all the day. With this automatic speed monitoring system, we can then curb down over speeding which is believed to be one of major causes of road accidents along highways.

1.2 Overview of the document

This document describes the implementation, testing and validation findings for the Automatic Highway Surveillance System. AHSS system comes as a substitute for the currently implemented speed control system (Speed guns) that has had a number of challenges such as its lack of agility in curbing and monitoring over speeding occurrences along major highways in Uganda.

The document is divided into the following sections:

Section 1: This section gives an overview of the document

Section 2: This section describes and specifies the system completely

Section 3: This section describes the design inputs (programming languages)

Section 4: This describes the inspection and testing phase of the system

Section 5: This describes the installation and system acceptance testing. All this involves describing input files, supplementary files and installation qualification.

Section 6: It involves describing the performance, servicing, maintenance and phase out process of the system.

CHAPTER 2. System Specifications

2.1 Version of requirements and Version Control

The requirements of the system were changed once which led to the latest version (R1.1) which were used in the final implementation of the system. The first version (1.0) required us to install part of the system inside the target car. However, after thorough investigation, the findings suggest that there were to be conflicts since no driver would like to be tracked. The new requirements (R 1.1) therefore specified that the entire system shall be external and independent of the target car.

2.2 Input

It being a speed control system, the computer receives two types of inputs as specified below. Note that, to save storage space and resources, only cars whose speed is above the speed limit shall be sent and stored on the computer. Illegal values that bypass (For example, when the first senor detects objects twice before the second sensor, we get a certain zero values), shall never be considered as true values. Therefore, shall be ignored.

Input 1: Offensive Speed values/digits. When a car is captured traveling at a speed above 100km/hr, its speed in km/hr is calculated and sent to the computer for storing, follow-ups, and record assessment.

Input 2: In the same case above, a screenshot of the offensive car shall be captured and sent to the system identification (screenshot of the number plate). This shall be sent and stored as BLOB.

2.3 Output

As a way of retrieving data from the system, data formats such as docx, Png, spreadsheet shall be valid. The system shall generate report documents in form of Docx. Using the screenshots captured on the highway, the system shall reproduce these images in Png light version in order to capture the details of the over speeding car. A list of offenders will be documented in form of spreadsheet which will then be used for further analysis to identify the rate at which over speeding occurs on highways. The columns shall contain speeds of the offenders, time of offense and date. All this shall be stored on the computer hard drive for future reference.

2.4 Functionality

The system shall be accurate in the way it measures the speed of the proceeding car. This speed as noted is the basic factor that determines whether or not it’s an offensive vehicle. Therefore, precautions are taken in the code that calculates car speed, to ensure that only accurate and valid speeds are recorded.

The system provides efficiency and reliability in report generation, ticket issuing to the offenders, and offense recording in the computer files. This is also vital since these records and documents are store for further reference.

The system also provides a clear and easy to understand user interface that allows users (traffic officers) to login in with valid credentials, carry out their tasks such as issuing over speeding tickets, generating reports, and registering offenses in a timely and reliable manner.

2.5 Limitations and safety

The system has different limitations in regards to different aspects. Talking about power/energy, the system will not work as required in absence of power supply. Secondly, due to the embedded electronic systems, such as sensors, power consumption of the system is high due to heat dissipation.

Another limitation we faced along the way is; it is difficult to make a back-up file of the system since most of the files are system made rather than user made. This also makes the transfer of files from the system to another Police system very difficult. Lastly, it is also hard to troubleshoot the system in cases of failure.

Data from the sensors shall be fed into the system using input tables of the system. This is to ensure only allowed data types and values in system valid formats are entered. Such format include, all speed is recorded in km, Date in data type date with a format YYYY-MM-DD, Time in data type time and with a format hh:mm:ss[.nnnnnnn], and images in BLOB. Any alteration in respect to the specified data types shall not be recorded in the data tables since the entire process is automated rather than user controlled.

2.6 Default settings

By default, when the system is first installed it contains only one user. That is the system admin with the following details. Username- Admin, Password- Admin@2021. He is the only who can add new users by default. The data input table fields are empty by default and there are no other users registered in the system as yet.

2.7 Special requirements

Since the system deals with Police sensitive records and data, the system is securing every account with passwords to limit access to data by unauthorized personnel. Authorized identities shall access the system using a combination of userId (Police ID) and their valid password. This shall cater for confidentiality and security of data. In case of errors, the system shall be restarted.

2.8 Errors and alarms

Errors may arise in terms of captured speed and captured screenshots. For the speed values, the errors expected are too minimal to affect the real value of the actual speed at which the care is traveling. However, for captured images, it being an object is motion, sometimes the number plate images may be too blur for recognition. In such cases, any other image processing software such as Photoshop maybe used to regenerate the image pixels and produce a clearer image that shows the car number plate.

Infrared sensors can only detect objects that are in a range of 5cm. This makes cars that pass beyond this distance undetectable. However, with bigger IR sensors capable of winder wavelength, we can achieve detection at even more distance away from the sensors.

Chapter 3: Design output

3.1 Implementation (coding and compilation)

The coding and compilation process was done using Arduino environment and Python IDLE. The system uses a number of hardware devices to accomplish its tasks. This includes sensors that detect movement and capture car speeds, and cameras, and Micro sensors. In those categories of hardware used, Infra Red Sensors are responsible for detecting motion and determining the time it takes a car to leave Sensor one to reach sensor 2. The TLL SERIAL CAMERA OV7670 is a microchip board containing a micro camera that captures and sends screenshots of the proceeding car for processing. Lastly, we have the Arduino AT MEGA 2560 with USB Cable which contains microchips and processors that calculates speed. With the help of an Ethernet shield, W5000 and a crossover Ethernet cable, the Arduino board sends all the received data as input to the computer system. FTD1232 USB is just an interfacing board between Arduino board and the ESP32-CAM.All the above hardware operates independent of the computer system hardware or software; however, the data sent by the combined components can be viewed by any computer system that runs Windows operating system.

3.4 Documentation

Table 3 Shows design details.

Topics	**Design output**	
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

Windows programming	☑ Interface implemented using standard Windows elements ☐ Interface implemented using self-developed Windows elements ☐ Application manages single/multiple running instances Comments:
Dynamic testing	☑ 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

4.1 Introduction

During the testing inspection testing, we were able to test the code before installing it on the Arduino board. Inspection testing was done in order to confirm if the system meets its requirements, the check the acceptance level, its complexity level and the anticipated risks as summarized in the table below.

Table 4 Shows 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: Source code was easy to read	12/11/2021 Kenneth
Documentation *Documentation inspected...*	☑ System documentation, flow charts, etc. ☑ Test results ☑ User manuals, On-line help, Notes, etc. ☐ Contents of user manuals approved Comments: Documentations were easy to understand and clear	12/12/2021 Nelson
Software development environment *Environment elements inspected...*	☑ Data integrity ☑ File storage ☐ Access rights ☐ Code protection ☐ Installation kit, replication and distribution Comments: Data was secure and easy to access through file storage	12/13/2021 Nelson

Topics	**3.3.1 Inspection plan and performance**	*Date / Initials*
Result of inspection	☑ Inspection approved Comments: Meets the intended use	12/13/2021 Nelson & Kenneth

4.2 Test plan and performance

The test plan and performance of AHSS system targeted the following modules of the system. The Infra Red Sensors, TLL SERIAL CAMERA OV767 microchip board, the Arduino AT MEGA 2560 with USB Cable, the user interface, and data repositories of the system.

During the testing phase, different components of the system were configured and interfaced to communicate with each other. Then, they were connected to a central power source to confirm that there wasn't any electrical fault. What we expected was the Arduino board to light immediately power was connected which happened.

The W5000 Ethernet shield was connected on the board and wired to the computer with an ether net cable. What we expected were the LED lights to fire which happened as well.

After that, the next step was to load to code onto the Arduino board. The step was successful using a USB connected to the computer.

Following that was the running the code in the working environment to test the sensors and the camera chips to confirm they do what they are required to do. Here, came the problem. The Ethernet shield was not in a position to link the Arduino board to available MSQL server through port 3306 since it couldn't configure the computer's IP address.

Here was the turnover of the entire process. W5000 Ethernet shield we had purchased was actually a Chinese clone unable to wire us to the server. The testing of the Camera too couldn't proceed without a genuine shield. The other option to transform the project was use of a serial port (COM7) with a python code as a client to the Arduino environment.

This took us to the user interface. What we wanted was to confirm if the user can access data sent by the system.

4.2.1 Test objectives

- To check the power supply and usage of the system components. This was done by connecting the system's components onto a central power source noticing whether the red lights are working.
- To check the compatibility of the components with each other. This was done by connecting the, Ethernet shield, Ethernet cable, Infra-Red Sensors and TLL SERIAL CAMERA OV7670 microchip board to the Arduino AT MEGA 2560 with USB Cable and all these were connected to the computer by the USB.
- To check the functionality and specifications of the system. Apart from the TLL SERIAL CAMERA OV7670 that we failed to test due to the failure of the W5000 Ethernet shield, all other functionalities of the hardware passed the test. The Infra Red Sensors were tested to confirm they were detecting object presence. They continued to blink each time we placed an object before them confirming they were working in real time.
- To check the functionality of the user interface. This was tested to note that data from the system could be retrieved by the user on the other end. If the use can create a report, and also export this data on an external storage.

4.2.2Scope and Relevancy of tests

The scope of the test was both hardware and software used by the system at large and it included the volumes of modules tested, system complexity and functionality of the system.

Software modules tested: Executable code to be installed on the Arduino board was tested whether it runs, if libraries were loaded without any errors, and with the required functionality. The python code was the new bridge between the hardware and the software past (Database and UI), and it was tested to check the connection if is available and working. The user interface code was also compiled and tested whether it has the required functionality.

Hardware modules tested: Infra-Red Sensors, W5000 Ethernet shield, Cross over Ethernet cable, the Arduino AT MEGA 2560 Cable and the USB connector cable.

4.2.3 Levels of tests

Module testing was done for each individual hardware component of the system, first of all, to check whether there was current flow in each one of the component. Infra-Red Sensors, W5000 Ethernet shield, crossover Ethernet cable, and the Arduino AT MEGA 2560 with USB Cable were all tested individually.

Integration testing was done for a joint connection of all these hardware components. The goal was to check the compatibility of the interfaces of all these modules and to confirm they could work together without failure.

4.2.4 Types of tests

We carried out input testing to check if captured data by the sensors and calculated speed was being sent to the data repositories for user access through the Ethernet shield.

Functional testing was done to confirm if each module mentioned above was doing its intended role and if jointly the system was doing what it is supposed to do.

Usability testing was done to check on the levels of usability of the system in the face of the user and the environment in which it will run.

4.2.5 Sequence of tests

A sequence of tests was done to confirm if the sequence of activities achieved the primary functionality of the system.

From the Arduino code, we tested whether it was loadable, and could fit in the byte space of the Arduino microcontroller. If the Libraries used were compatible, together with testing the hardware tools we had.

The python code was tested too to confirm its compatibility with both Xampp MSQL, and the Arduino code using the serial port of the computer.

The database was tested whether it would allow inputs with the created user given privileges to insert and alter the database's tables.

Finally, the user interface was tested to confirm if the functionalities were present.

4.2.6 Configuration and calculation tests

The two infrared sensors were placed 0.2m apart.

An object was moved at a slow from sensor 1 and to sensor two

The readings were taken as 13.4km/hr

The second object was moved slightly faster than the first object in front of the sensors

A slightly higher speed was seen in the serial monitor as 20.43km/hr

In the last step, the same object was moved very fast from sensor 1 to 2.

The readings of the speed was even greater, 79.33km, which confirmed that the faster the object in real time, the greater the speed returned on the serial monitor.

4.3 Precautions

4.3.1 Anomalous conditions

Infra-Red Sensors have frequencies in the range of 0.3–40 GHz which is approximately 1 mm to 1 m of wavelength. We used two Infra-Red sensors denoted as Radar1 and Radar2. There might be obstructions and inconsistencies in case sensor one receives double inputs before sensor two receives any input. In this case, vehicle speeds might not be recorded in their right values.

4.3.2 Precautionary steps taken

In order to thwart the above anomalies, sensor should be positioned in a way that they can capture one vehicle at a time rather than numerous vehicles at ago. In this way, the first sensor should only and only receive other input when the second sensor has finished inputting the results of the first car.

Chapter 5: Installation and system acceptance test

5.1Input files

- **Converted source *code* representation of the embedded software**: This is used in the linking and installation of the code onto the Arduino board.
- **Installation files for the user interface**: This contains the files that install the user interface onto the user's computer.

5.2 Supplementary files

There is an additional text file containing (Readme.txt) that contains help steps on how to log into the system the first time the user is accessing, creating accounts and reading data from the repositories.

5.3 Installation qualification

- Install the system's hardware in an appropriate position where it can detect, measure and capture a proceeding vehicle.
- Ensure that the target computer has a disk space of at least 1GB free.
- Install the system's user interface on the user's machine running windows' operating system.
- Read the Readme.txt file to see how to login into the system the first time it is installed.

Table 5 shows 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: We used manual installation.

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: Source code of the embedded system is already installed on the Arduino board.
Installed files *List of (relevant) installed files, e.g. EXE- and DLL-files, spreadsheet Add-ins and Templates, On-line Help, etc.*	• HEX files • PWI files • BAK files • Opt files • C files

Table 6 Installation Procedure Check

Topics	**Installation procedure**	*Date / Initials*
Authorization *Approval of installation in actual environment.*	Person responsible: Nelson	12/17/2021 Nelson
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:	12/17/2021 Kenneth

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

6.1 Service and maintenance

Once the system has been installed in its working environment, we shall monitor it once every month for five months to ensure it is still in the right functionality. In case of any faults, our team shall rectify the issues at hand in the process of maintaining and supporting the client.

At the moment, there is no other version of the system. However, future updates shall arise in response to the common faults that may arise and were not detected in the testing phase. Secondly, there will be an improvement in the functionalities provided by the system, and the general usability improvement.

6.2 Performance and Maintenance

It should be noted that the system shall be monitored every month for the first five months after installation. Once there are no serious faults, then, it will be left to run for the next seven months.

The requirements for service are; the system should have been in the environment running for the past twelve months before the date of servicing and updating.

At this stage, our team shall update data repositories, check for broken code, and perform data cleaning, hardware troubleshooting, and replacement where necessary.

During upgrading of the system from the older version to the newer version, data shall be moved by the use of external storage media such as HDD or SSD cards. Data repository tables and structures shall always be maintained to support old data formats from the older system version.

Table 7 Performance and maintenance details

Topics	**Performance and maintenance**	*Date / Initials*
Problem / solution	*Detection of system problems causing operating troubles. A first step could be to suggest or set up a well-documented temporary solution or workaround.*	Dates must be filled in

Topics	**Performance and maintenance**	*Date / Initials*
Functional maintenance	The system is committed to its functional specifications and requirements. Any changes should address these functionalities and endeavor to maintain the system's intended use (curbing over speeding along highways), same applies to any updates that will come in later.	12/17/2021 Kenneth
Functional expansion and performance improvement	• Moving from Desktop computers to mobile gadgets to improve portability of the system. • Use of wireless technology instead of wired connections also to support portability and all time access to system data and records.	12/17/2021 Nelson

Chapter 7: Conclusion and Recommendations

This document intended to describe the software design document, System implementation, and testing and validation report for the Automatic Highway Surveillance System.

We recommend that the two sensors are placed in an appropriate distance where it can detect one object at a time without interruptions.

In order to implement the camera module in the system, a genuine W5000 Ethernet shield and crossover Ethernet cable should be used to wire the images direct from camera into the database. Otherwise, the serial port data is only 8bits that cannot support image transfer using python.

The alternative is to go deeper into imaging splitting, and converting the image in bytes, then store it on a memory card, a process which was long and needed much more than to say.

Appendix A: User Manual

Hardware setup

The hardware of the system come assembled; otherwise, Infra-Red Sensors are interfaced to the Arduino AT MEGA 2560 with USB Cable.

Ensure that the components are interfaced properly such that they can communicate with each other without failing.

Now, connect the Arduino board to a power supply (your computer) through a USB cable to one of your computer ports.

Let the board start working.

Then, go to your computer and launch the interface setup.

Login using valid credentials (Username: Admin, Password: Admin@2021, Id_number: Ahs01). Once you are granted access, you can view the home page that contains the menu and buttons for performing tasks.

The user interface

The user interface is a collection of the system functionalities implemented as buttons, forms, pages and data displays. Below is how you may complete tasks with the AHSS system.

The administrator logs in and is able to do the following as designated on the user interface.

- Adding users so that unknown users can not access the system.
- Adding highways on which the system will be used.
- Viewing data. The data part is for viewing data available in the database
- Payment. The user interface has a provision that allows the user to insert payment details of the offender.
- Under Ticket, There is issuing and as well as the provision to upload the receipt.
- Under Paid Tickets; The user is able to view the tickets that have been cleared.
- Under reminder, The user sends an email to the offender to pay in time.

- Under Reports, The user can choose between dates to be able to view the offenses carried out in that time frame.

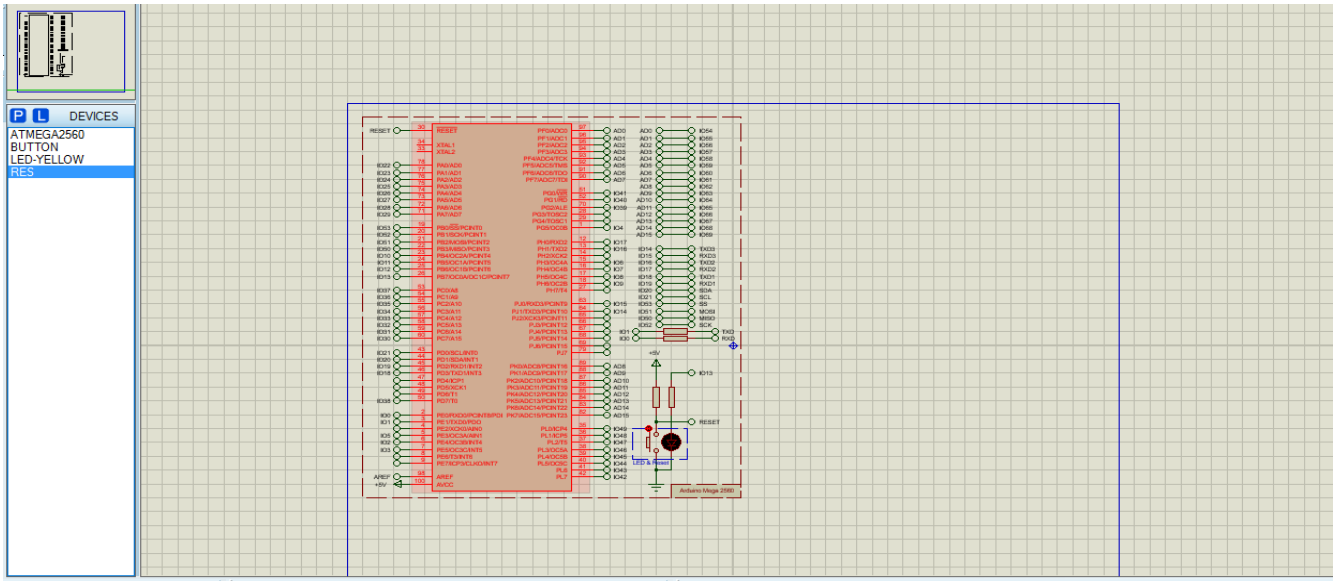

Figure 21shows the Arduino AT MEGA 2560 before connections

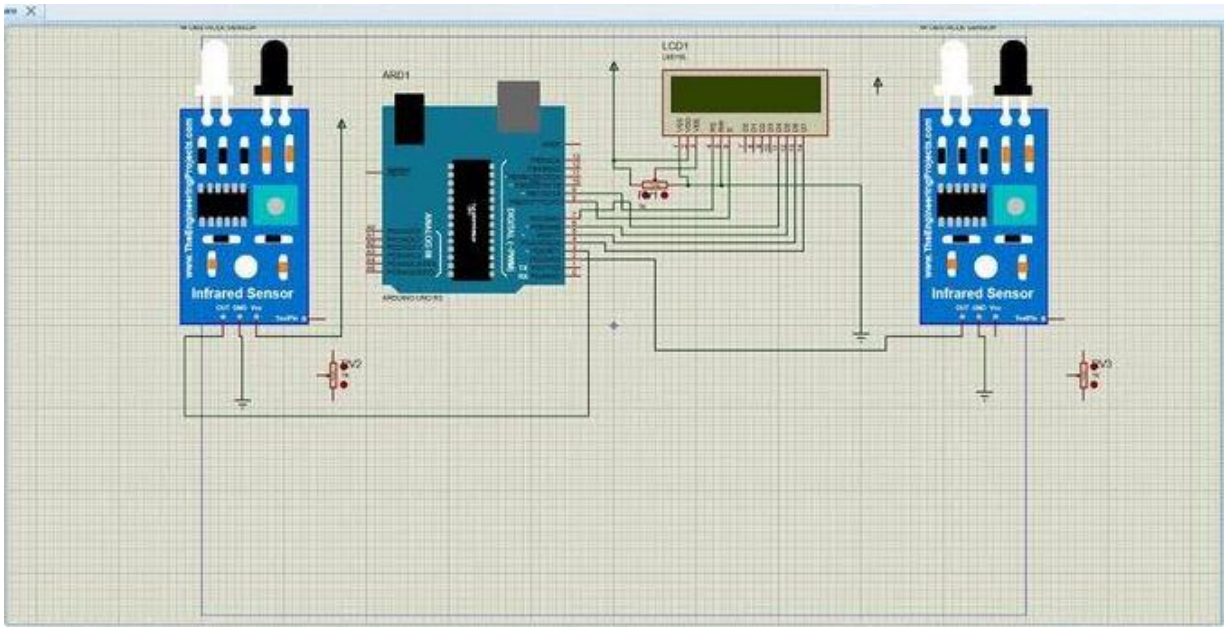

Figure 21 Assembled components with connections

first_code

```
    float velocity;               // This shall be used to measure car velocity
    long RandNumber;


void setup()   {
    // we shall put our setup code here. This is code that shall run only once
      pinMode(Radar1, INPUT);    //This defines the variable inputs for Radar1 using pin
      pinMode(Radar2, INPUT);    //This defines the variable inputs for Radar2 using pin
    //  lcd.clear();     //To clear our screen
    //  lcd.begin(16, 2);  //This shall start our screen
      Serial.begin(9600);  //This starts our computer monitor through the serial port
   //   lcd.setCursor(0,0);    //This shall set the cursor to start at zero row and zero column
   //   lcd.print("Welcome To AHSS");
   //   lcd.print("Speed detector is on!");
        Serial.println("Welcome To AHSS");
        Serial.println("Speed detector is on!");
  }



void loop() {
      // our code that executes in a loop goes here
  if (digitalRead(Radar1)==1)  // if the car is detected at sensor Radar1 then start the timer
      {
```

Done compiling.

```
Sketch uses 11666 bytes (4%) of program storage space. Maximum is 253952 bytes.
Global variables use 1258 bytes (15%) of dynamic memory, leaving 6934 bytes for local variables. Maximum is 8192 bytes.
```

21 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM6

Figure 22 Compiled code of the system

Figure 23 Output on the serial Monitor

Final approval for use	
Identification:	
Responsible for validation:	
Remarks:	
Date:	Signature: