A Project Report on

Intelights: Smart Traffic Management System

Submitted in full fulfillment of requirement

for the award of the degree of Diploma Computer Engineering Gujarat Technological University

Submitted By:

Prayag Acharya

(176170307001)

Chaitanya Bhalodia

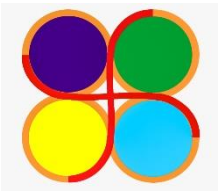
(176170307013)

Dravik Italiya

(176170307039)

Abhishek Dhanani

(176170307027)



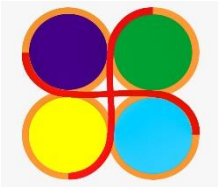
Government Polytechnic Ahmedabad

Computer Department

October 2019

Computer Department

Government Polytechnic Ahmedabad



C E R T I F I C A T E

This is certified that the Project entitled

Intelights: Smart Traffic Management System

Diploma in Computer Engineering

Is a result of the bonafide work carried out by

Prayag Acharya

(176170307001)

Chaitanya Bhalodia

(176170307013)

Dravik Italiya

(176170307039)

Abhishek Dhanani

(176170307027)

Internal Guide	Head of Department
Mrs. Khyati P. Goswami	Prof. H. J. Baldaniya

Government Polytechnic Ahmedabad

ACKNOWLEDGEMENTS

We take this opportunity to express our humble thanks to all those connected with our project entitled "***Intelights: Smart Traffic Management System***".

We express very sincere thanks to Khyati ma'am for providing us an opportunity to work on such an interesting project, and in a congenial environment. Their invaluable guidance has proved to be a key to our success in overcoming challenges that we have faced during the course of the project work.

The knowledge that we have gathered, study of relevant literature for understanding our project, planning of details and selecting the suitable software platform by itself and acknowledgement to the industry, zeal and technical competence of those many individuals who have contributed to it with profound gratitude, we wish to acknowledge all of them.

With Regards,

Prayag Acharya

Chaitanya Bhalodia

Dravik Italiya

Abhishek Dhanani

“ Don’t worry about what anybody else is doing to do. The best way to predict the future is to invent it ”

- Alan Kay

Abstract

Traffic is major problem for many cities of India along with other countries. Failure of signals, poor law enforcement and bad traffic management has led to traffic congestion.

Smart Transportation System must require, Our Software Will Detect flow of vehicles and then for that particular time duration Signal shows Green Light for Vehicles.

Index

1. Introduction

1.1. Project Profile

Project Name:	Intelights: Smart Traffic Management System
Front End:	Python
Back End:	TensorFlow
Internal Guide:	Mrs. Khyati P. Goswami
Team Leader:	Prayag Acharya
Project Duration:	1 year
Team Size:	4 Members

1.2. Hardware Requirements

Processor:	Intel® Core™ i5 processor.
RAM:	Minimum 4 GB
Hard Disk:	Minimum 10 GB
User Interface	(Keyboard, Mouse, Printer)

1.3. Software Requirements

Operating System: Windows 7 or Above

Programming Tools: Back-End (Database) - MySQL Server

Front-End Java, Python

2. Literature Survey

2.1. Existing System

- In Existing System, System will only record the video of each and every cross roads and then operator will see all the recording and when he issues all the challan and print it.
- Current System is not dynamically changing the duration of traffic lights, all the traffic lights' is set as per its place (i.e. place like Kalupur Cross road have heavy traffic in morning time so duration is set to 40 to 45 seconds for green light)
- Cameras are not major the density of traffic all traffic lights duration is set per its time (i.e. Vijay cross road have heavy traffic density in evening time so at evening time signals are set to static duration)

2.1.1. Working of the current system

- Current System is only recording the video and operator will analysis this recording and then operator will check if any vehicle driver broke rule then fined him/her.
- All signal's time Duration is statically set. And Duration is change as per time.

2.1.2. Shortcomings

- In Current System, Traffic Signal Duration will not change as per traffic density.
- Operator must be needed.
- System is not fully automatic.
- There is no proper management for emergency vehicle.

2.1.3. Needs of a new System

- To Enhance the Existing System, New system need not any operator it is fully automatic.
- For better Traffic management.
- For better go through for emergency vehicle.
- Dynamically changing of Traffic Signal duration Using Traffic Density analysis

2.2. Process model

- ITERATIVE WATER FALL MODEL: - A preliminary study was done for the system and was documented as system Project Proposal which was accepted and further development of the system was done with regards to the detail study and preliminary study.

- There are various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as "Software Development Process Models".

- Each process model follows a particular life cycle in order to ensure success in processes of software development. One such approach/process used in Software Development is "The Iterative Waterfall Model".

- In "The Iterative Waterfall" approach, the whole process of software development is divided into separate process phases. The phases in Waterfall model are: Requirement Specifications phase, Software Design, Implementation and Testing& Maintenance.

- All these phases are cascaded to each other so that second phase is started as and when defined set of goals are achieved for first phase and it is signed off, so the name "Waterfall Model".

- Waterfall model has many drawbacks so it is a solution of waterfall model.

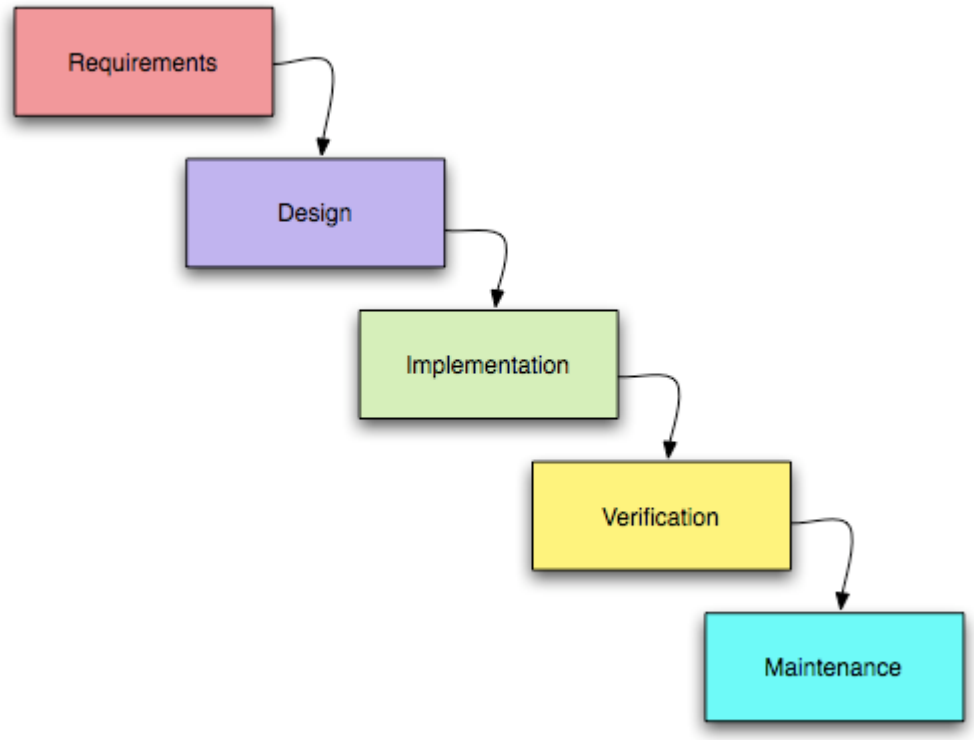


Fig. 2.1 Iterative Waterfall Model

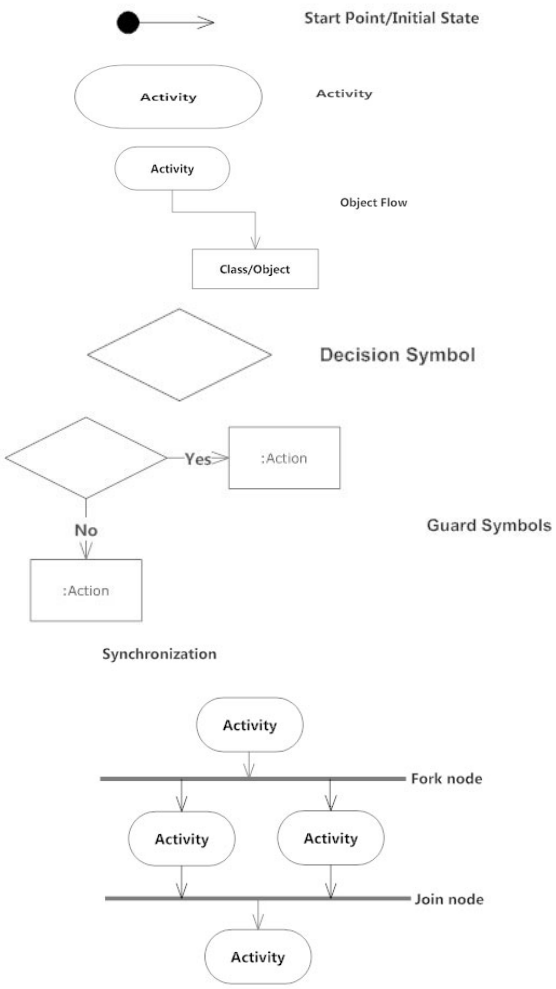
Benefits

- Simple and easy flow.
- Each phase lays out its derivable, followed by a validation process.
- Excellent work progress tracking system.
- Simple and feasible.
- Best option for smaller project.

3. System Design.

3.1. Activity Diagram

- Activity diagram is basically a flowchart to represent the flow from one activity to another activity.
- The activity can be described as an operation of the system. The control flow is drawn from one operation to another.
- The symbols used are:



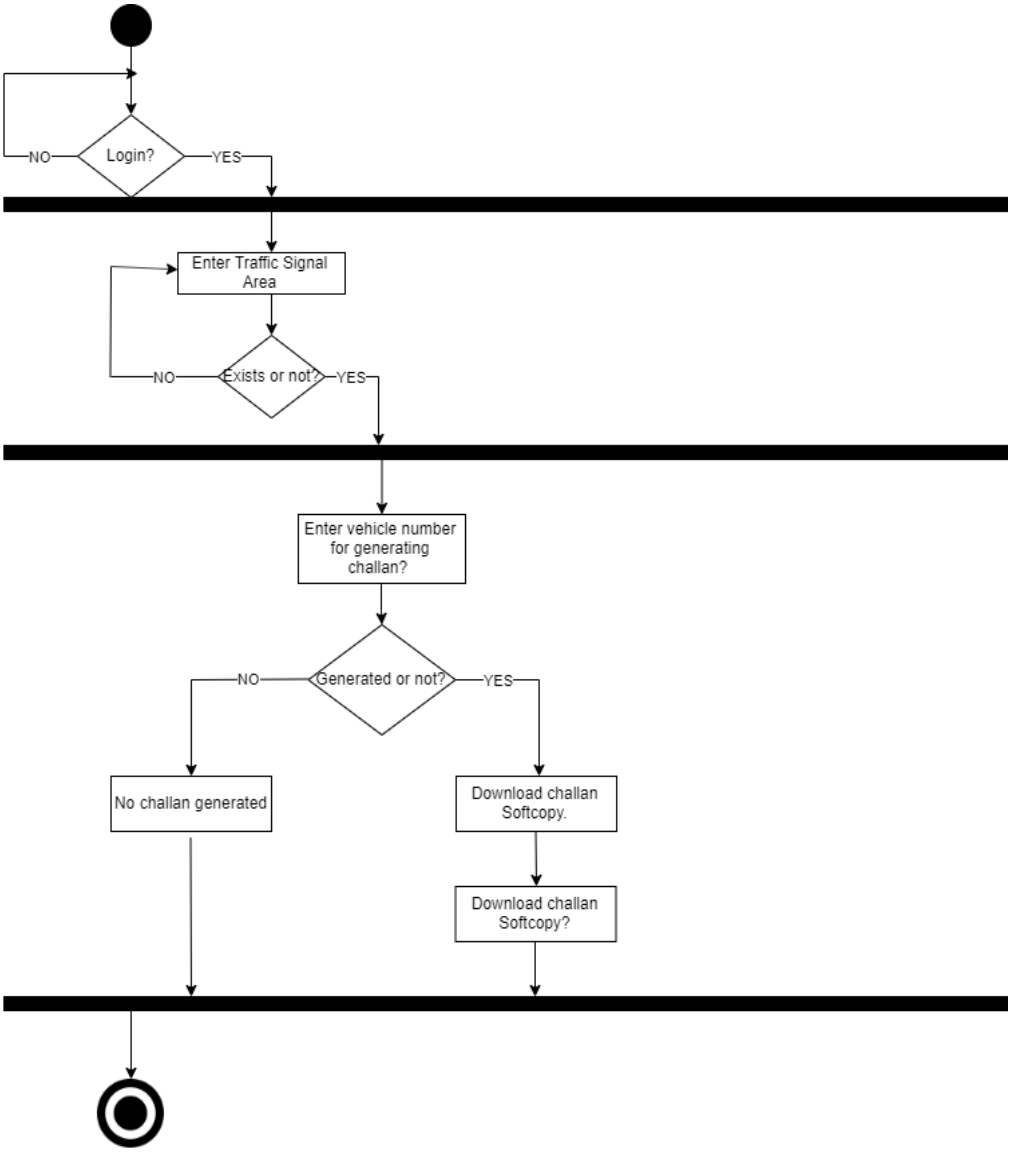


Fig. 3.1.1 System Flow Chart

3.2. Entity Relationship Diagram (ERD)

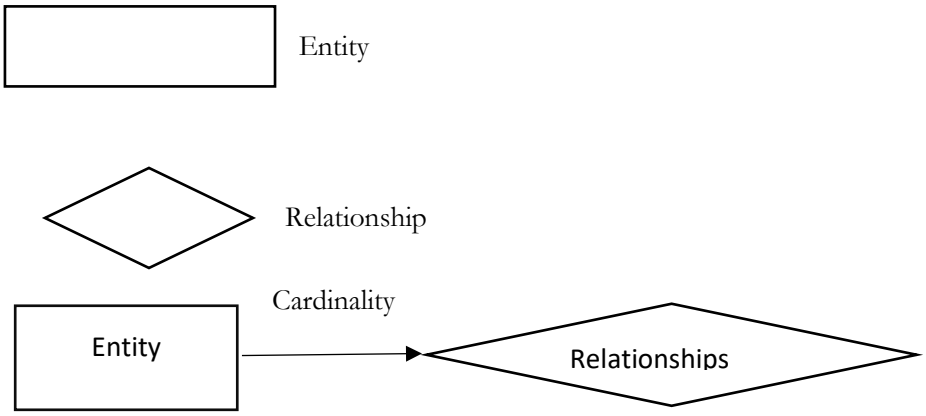
- Entity Relationship Diagram do not depict flow or processing.
- They should not be read the data flow diagrams or flowcharts. Entity Relationship Diagram depicts data at rest, data being stored. They also don not imply how data is implemented, created, modified, used or deleted. ERDs will be easy to read and interpret.
- There are three basic elements in E-R Diagram:

  Data Entity: A Data Entity, which will be referred to as entity flow now on, is the main symbol on an ERD. An entity is anything, real or abstract, about which we went to store data.

  Attribute: A key attribute is the unique, distinguishing characteristic of the entity.

  Relationship: A relationship is a diamond that contains its name. It touches one relationship-entity and optionally some attribute-entity connectors. It is linked with two entities.

➢ Symbols used in E-R diagrams:



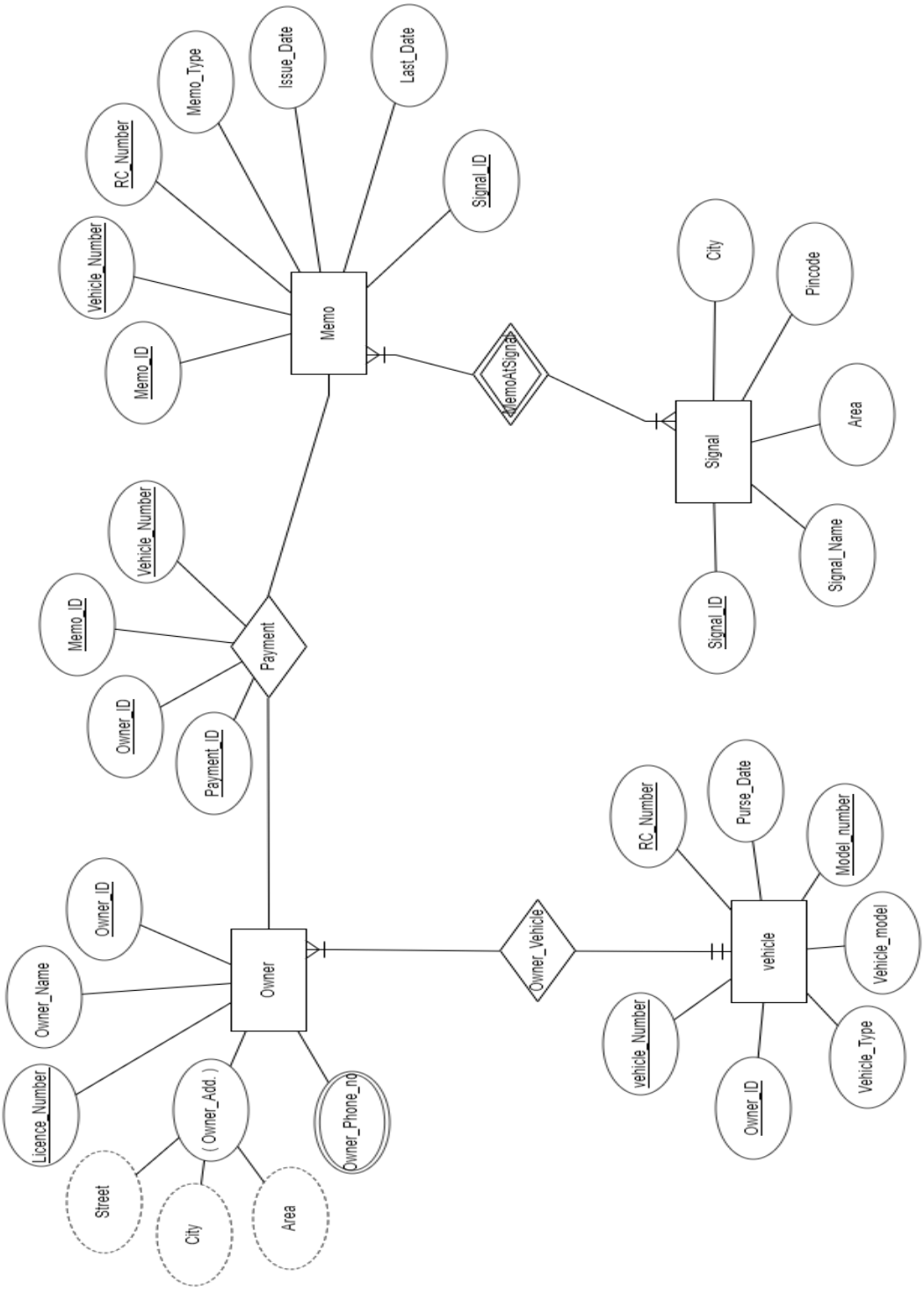
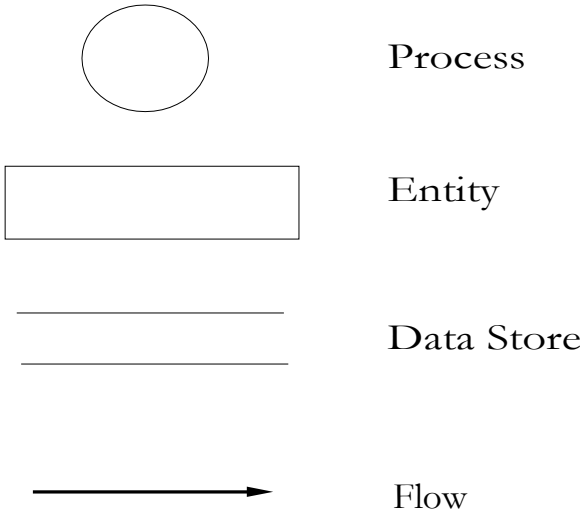


Fig. 3.2.1 System E-R Diagram

3.3. Data Flow Diagram (DFD)

- DFD is a graphical representation of the flow of data through an information system. It differs from the system flowchart as it shows the flowchart as it shows the flow of data through processes instead of hardware.
- A data flow diagram is logical model of the system and shows the flow of the data and the flow of logic so this all thing describes what takes place in a proposed system, not how the activities are accomplished.
- DFD consist of a series of symbols joined together by a line. There may be a single DFD for the entire system or it may be expanded into various levels.
    - Context Level Diagram
    - First Level DFD A Data Flow
    - Second Level DFD



- Context Level DFD

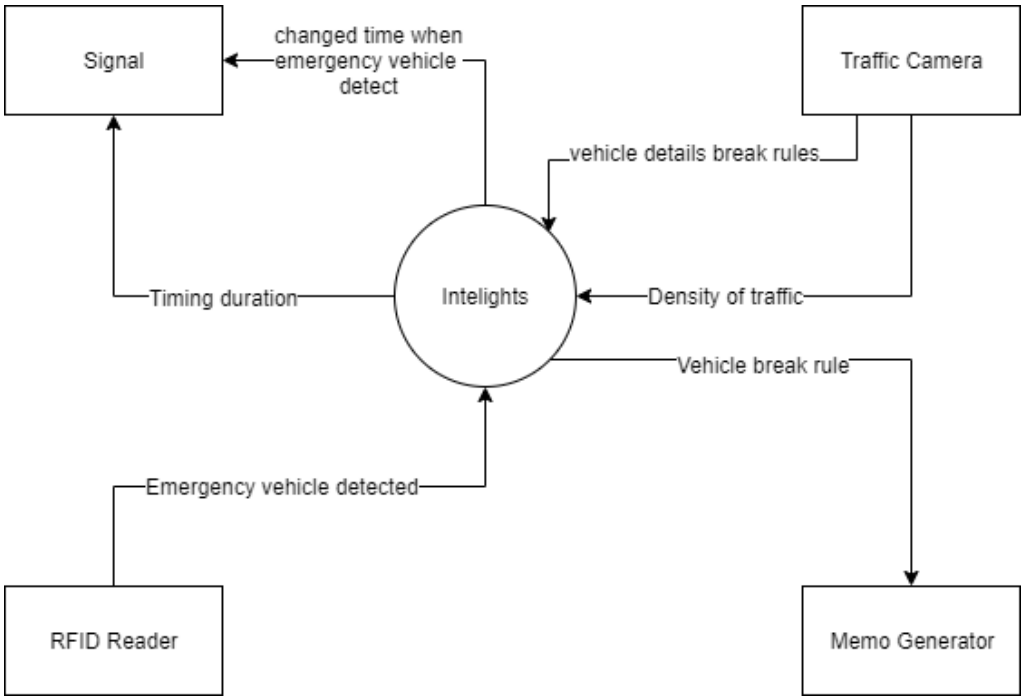


Fig. 3.3.1 Context Level DFD

- Level-1 DFD

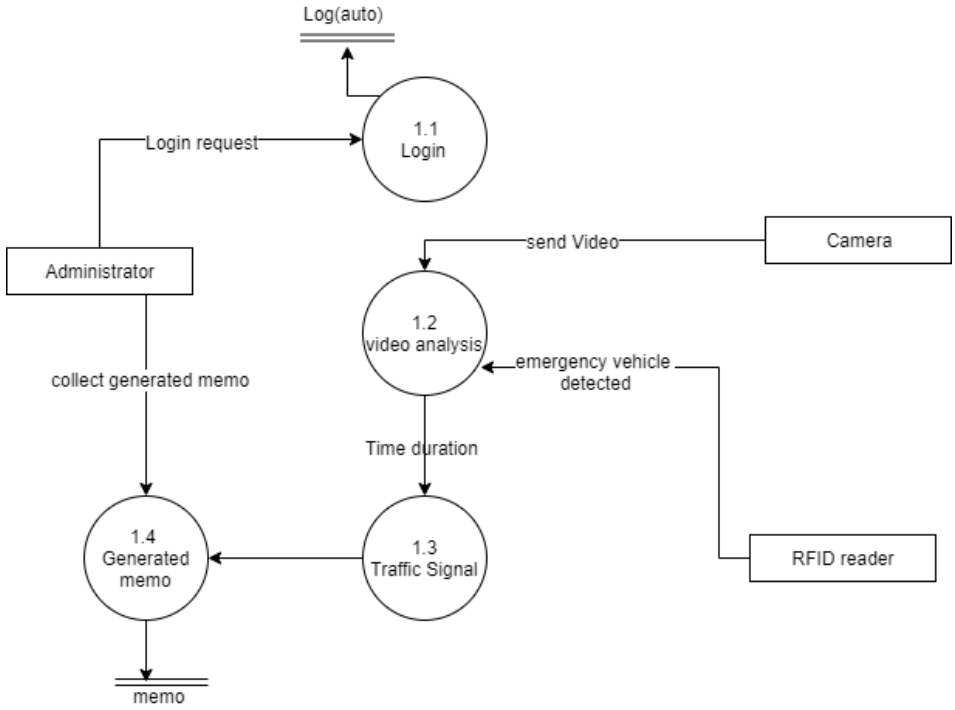


Fig. 3.3.2 Level-1 DFD

3.4. Use Case

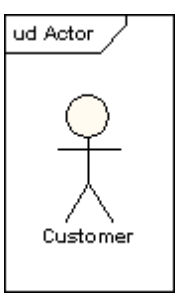
Purpose:

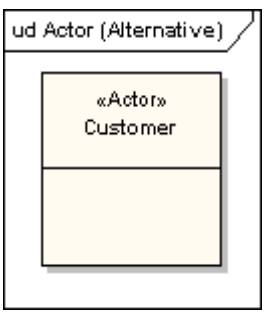
- The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So, we will look into some specific purpose which will distinguish it from other four diagrams.
- Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.
- Now when the initial task is complete use case diagrams are modelled to present the outside view.
- So, in brief, the purposes of use case diagrams can be as follows:

  - Used to gather requirements of a system.
  - Used to get an outside view of a system.
  - Identify external and internal factors influencing the system.
  - Show the interacting among the requirements are actors.

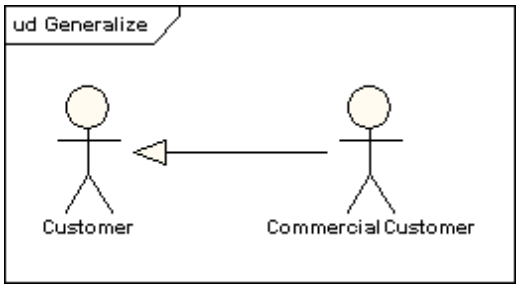
Actors

A use case diagram shows the interaction between the system and entities external to the system. These external entities are referred to as actors. Actors represent roles which may include human users, external hardware or other systems. An actor is usually drawn as a named stick figure, or alternatively as a class rectangle with the «actor» keyword.





- Actors can generalize other actors as detailed in the following diagram:



Use Cases

A use case is a single unit of meaningful work. It provides a high-level view of behavior observable to someone or something outside the system. The notation for a use case is an ellipse.

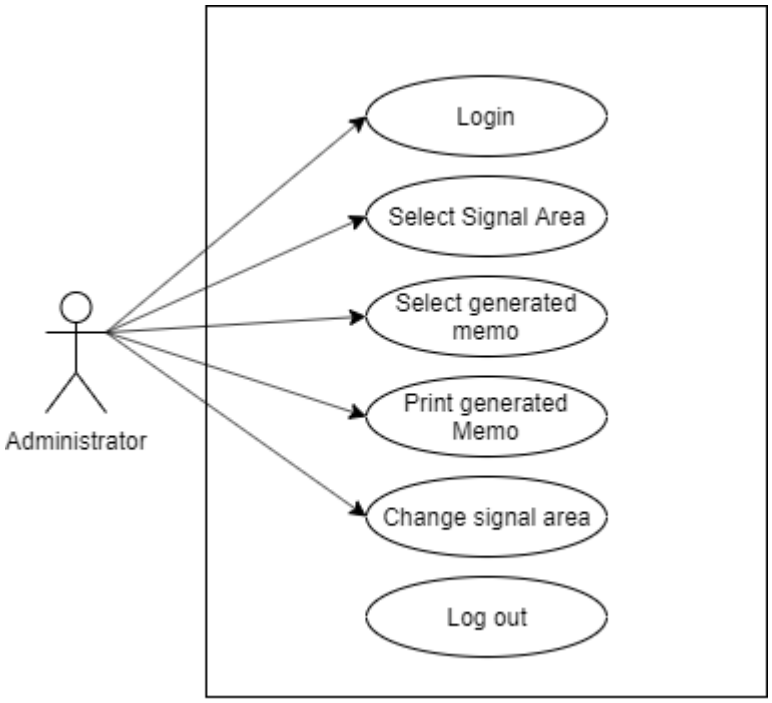


Fig. 3.4.1 Use Case diagram

3.5. Wireframe

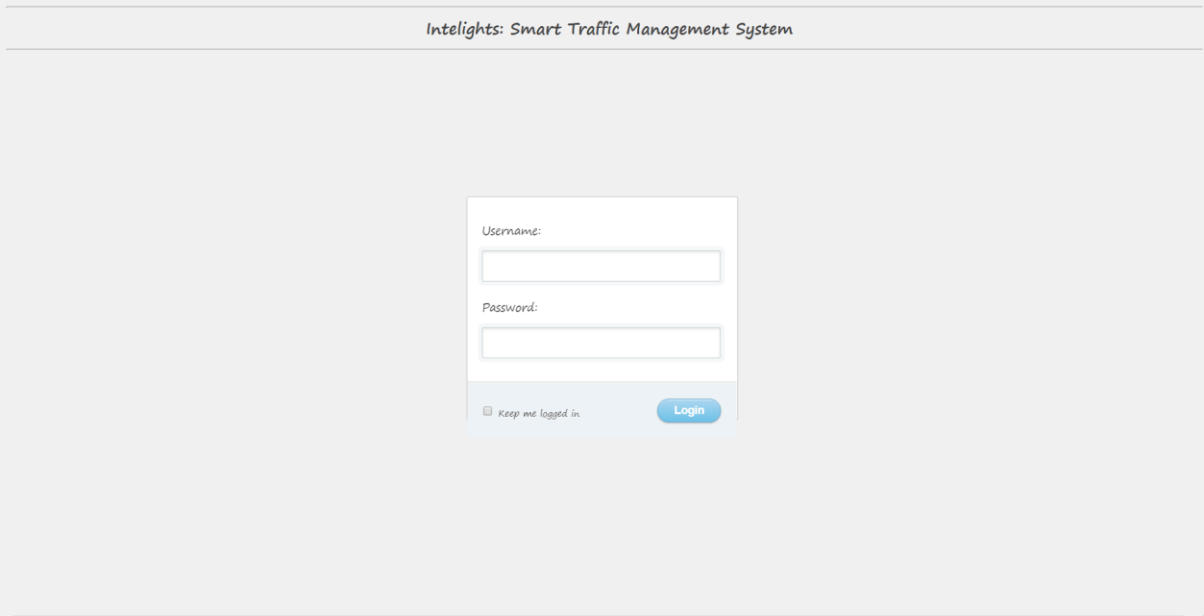


Fig. 3.6.1 Wireframe (Login Page)

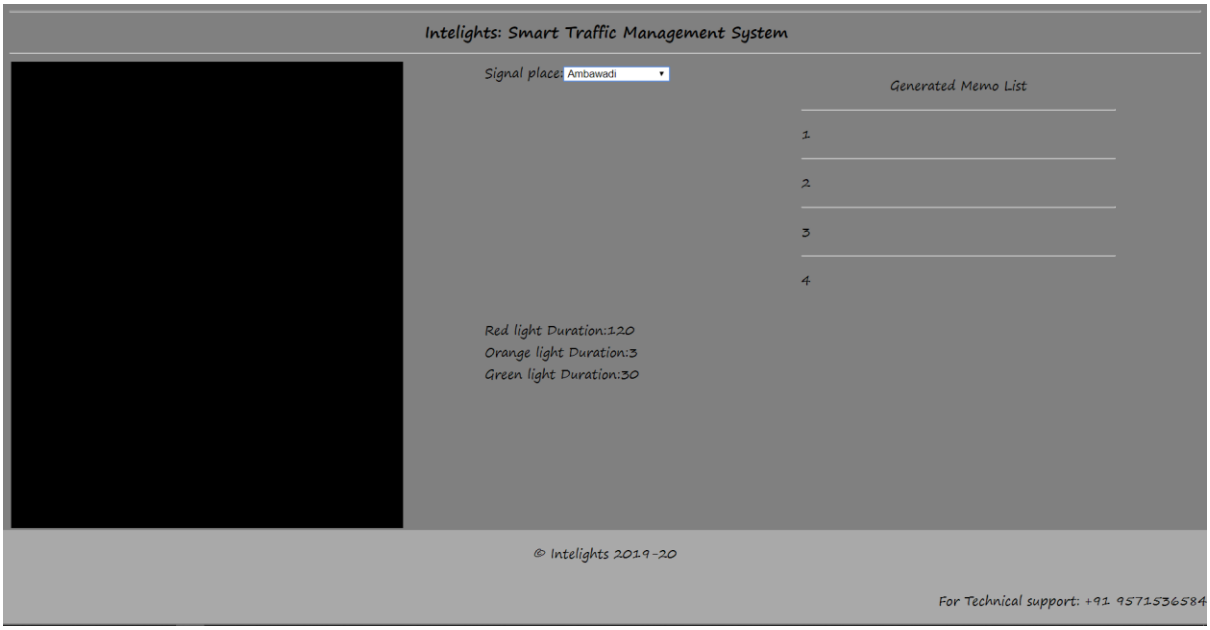


Fig. 3.6.2 Wireframe (Main Page)

4. Screen Shot of System

Fig. 4.1 Initial Window (Main Page)

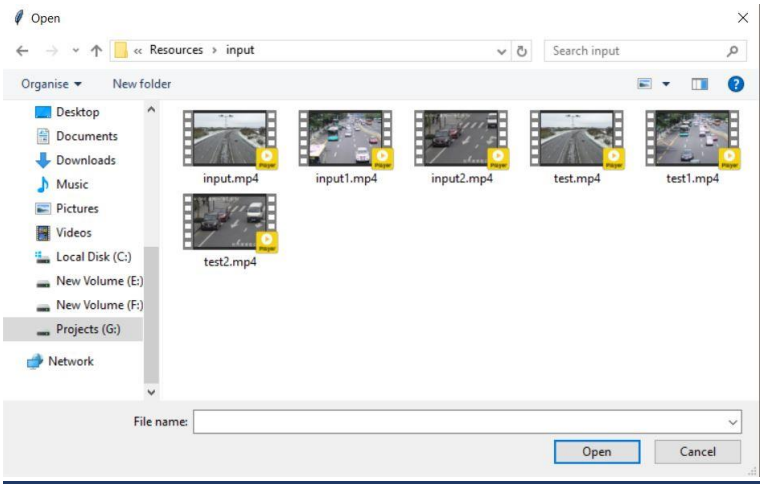


Fig. 4.2 Open Window

Fig. 4.3 Preview Loaded

Fig. 4.4 Select region of interest

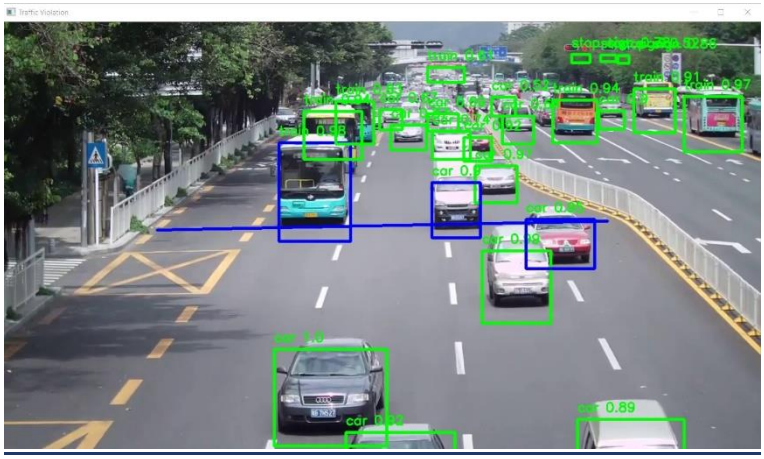


Fig. 4.5 Violation delectation analysis

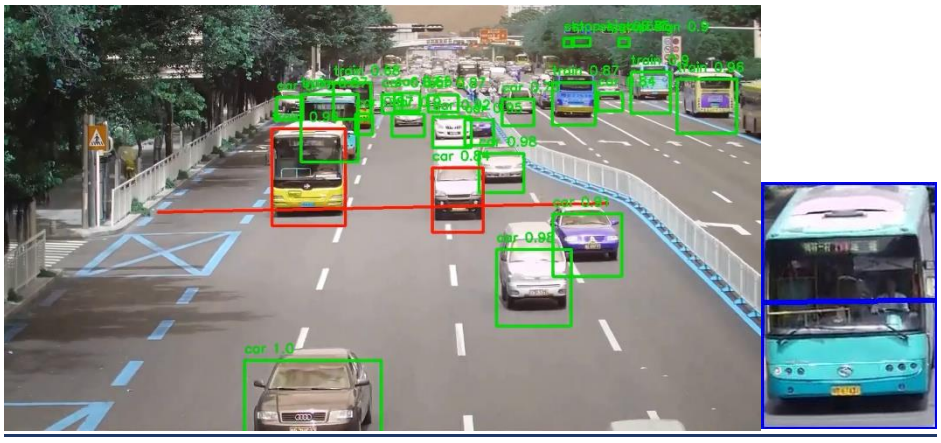


Fig. 4.6 Violation detection

5. Testing

5.1. Testing Planning

Testing planning is carried out in following three stages:

- Design
- Implementation
- Coding

- Design Testing: The design errors are to be rectified at initial stage. Such errors are very difficult to repair after execution of software.
- Implementation Testing: The errors at this stage can't be overlooked because such errors do not allow the further process.
- Coding Testing: The coding procedure plays significant role in software designing. The improper coding of any software can generate inconsistent result. Such error may occur due to incorrect syntax or false logic. If the error at coding stage remain unnoticed may give rise to grave failure to system.

5.2. Testing Strategy

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in successful construction of software. The strategy provides roadmap that describes steps to be conducted as part of testing, then these steps are planned and then undertaken and how much effort, time and resource will require.

6. Conclusion and Future Scope

6.1. Limitation of our project

- In some weather condition our software may not give proper result.
- Costly.
- Maintenance required.

6.2. Conclusion

At the end of this project we design software that vary duration of signal timing as per traffic density and also fined people who break rules. In our software we design better go through for Emergency vehicles and save time of all people and reduce pollution.

6.3. Future Scope

Future Enhancement of This project is:

- We will calculate traffic density using video analysis and hardware like magnetometer also.
- We improve range of RFID reader to detect Emergency vehicles

7. References and bibliography

- ieeexplore.ieee.org