

# **Real Time Traffic Problem Detection Using CCTV Footage**

**UCS503 Software Engineering Project Report**

**Mid-Semester Evaluation**

**Submitted by:**

(101903504) KARTIK PANDEY(Online)  
(101903522) Mehul Sah (Online)  
(101903500) Chirag Singla (Online)  
(101903498) Satyajeet Singh (Online)

**BE Third Year, CoE**

**Group No: 1**

**Submitted to:**

**Arwinder Dhillon**



**Computer Science and Engineering Department**

**TIET, Patiala**

**October 2021**

## TABLE OF CONTENTS

<b>S.No.</b>	<b>Assignment</b>	<b>Page No.</b>
<b>1.</b>	<b>Project Selection Phase</b>	
1.1	Software Bid	
1.2	Project Overview	
<b>2.</b>	<b>Analysis Phase</b>	
2.1	Use Cases	
2.1.1	Use-Case Diagrams	
2.1.2	Use Case Templates	
2.2	Activity diagram and Swimlane Diagrams	
2.3	Data Flow Diagrams (DFDs)	
2.3.1	DFD Level 0	
2.3.2	DFD Level 1	
2.3.2	DFD Level 2	
2.4	Software Requirement Specification in IEEE Format	
<b>3.</b>	<b>Design Phase</b>	
3.1	Class Diagram	
3.2	Sequence Diagram	
3.3	Collaboration Diagram	
3.4	State Chart Diagram	
<b>4.</b>	<b>Implementations</b>	
4.1	Component Diagrams	
4.2	Deployment Diagrams	
4.3	Screen Shot	

**5. Testing**

- 5.1 Test Plan
- 5.2 Test Cases
- 5.3 Test Reports

# **1. Project Selection Phase**

## **1.1 Software Bid**

# **Software Bid / Project Team**

## **UCS 503- Software Engineering Lab**

Group : 3CO19

Dated: 13/08/2021

**Team Name:** Kapie

**Team ID:** 1

Name	Roll No	Project Experience	Programming Language used
KARTIK PANDEY	101903504	Cassava Plant Disease Detection ATM Banking System Real Time Object Detection for Flying Objects	Python/C++/ C/HTML/CSS/Javascript/Flutter
Chirag Singla	101903500	Portfolio Website Flutter based Calculator	Python/C/C++/HTML/CSS
Mehul Sah	101903522	Text based game Robotic arm	Python/ C++/ C
Satyajeet Singh	101903498	The Simon Game, Tip Calculator	HTML,CSS, Javascript

### **Programming Language / Environment Experience**

1. Python
2. C++
3. C
4. HTML/ CSS
5. Javascript / Flutter
6. Java

### **Choices of Projects:**

First Choice	Real Time City Problem Detection using CCTV footage( This project would make the process of dispatch of fire, health, police forces automated and could help in CCTV Surveillance and other traffic problem )
Second Choice	Skin cancer segmentation and classification (automated diagnostics methods with the ability to classify multiclass skin lesions with greater accuracy)
Third Choice	Malware Detection by using Malware as Images (Convolutional Neural Networks (CNN) against redundant API injection)
Fourth Choice	Fish Segmentation and Classification (systems based on machine learning and image processing, which present fast and precise results)

## **Additional Remarks/ Inputs**

---

1. One of the reasons for choosing this project is that this idea is unique. And we believe that we can greatly increase our understanding of the topic by implementing this project. Moreover, this project sounds interesting and has a lot of potential for a real time-real world setting, mostly traffic surveillance is not automated and there is a huge potential for automation in this sector.
  2. Manual diagnosis of skin cancer is time-consuming and expensive; therefore, it is essential to develop automated diagnostics methods with the ability to classify multiclass skin lesions with greater accuracy.
  3. Driven by economic benefits, the number of malware attacks is increasing significantly on a daily basis. Malware Detection Systems (MDS) is the first line of defense against malicious attacks, thus it is important for malware detection systems to accurately and efficiently detect malware. Traditional MDS typically utilizes traditional machine learning algorithms that require feature selection and extraction, which are time-consuming and error-prone. Conventional deep learning based approaches typically use Recurrent Neural Network (RNN) which can be vulnerable to redundant API injection. Thus, we investigate the effectiveness of Convolutional Neural Networks (CNN) against redundant API injection.
  4. Assessing the quality of seafood both in retail and during packaging at the production side must be carried out minutely in order to avoid spoilage which causes severe human health problems and also economic loss. Since the illnesses and decay in seafood presents distinct symptoms in different species, primarily the classification of species is required. In this field, the inadequacy of the current laborious and slow traditional methods can be overcome with systems based on machine learning and image processing, which present fast and precise results.
-

## 1.2 Project Overview

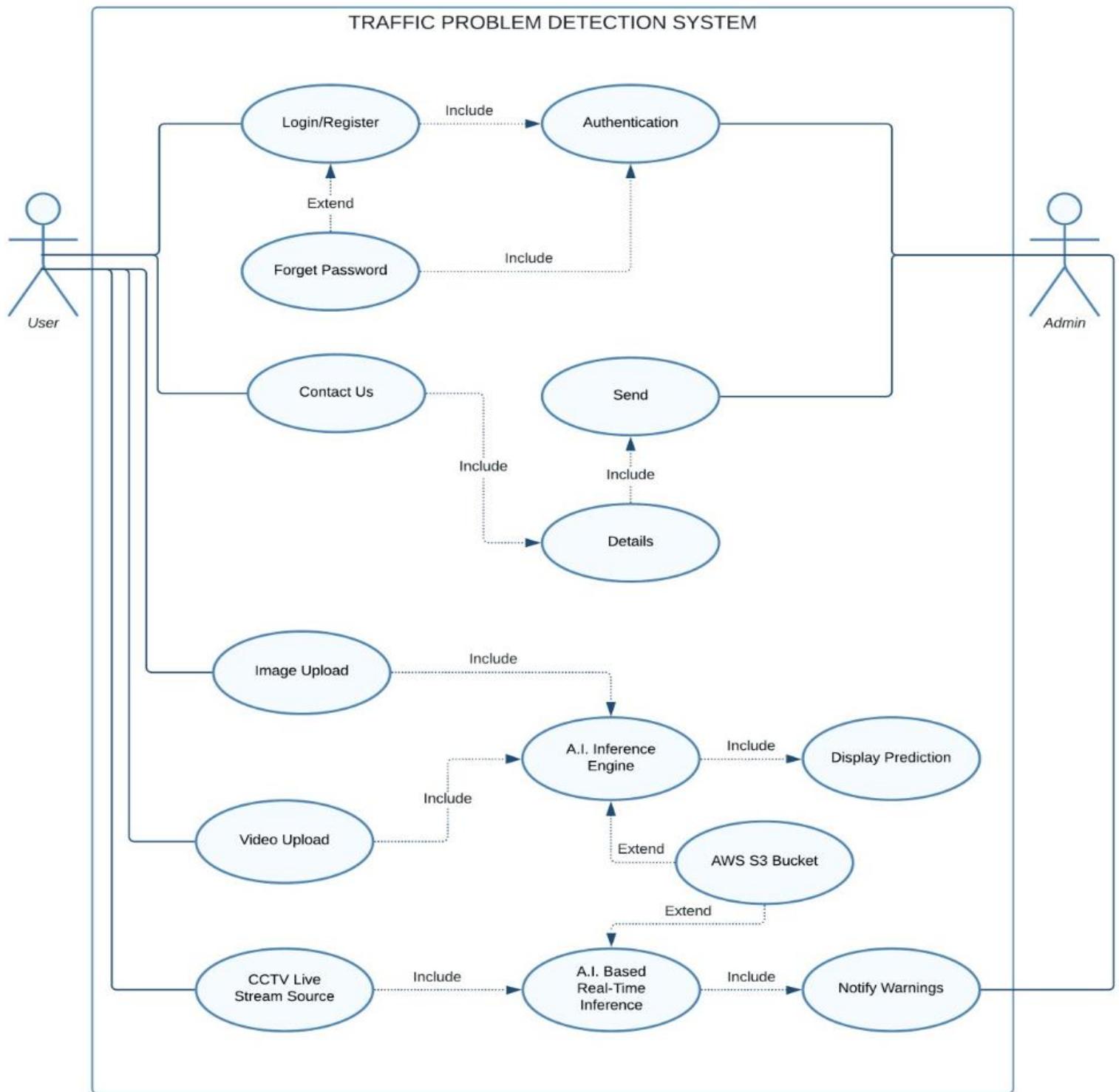
This project aims at developing an A.I. Model to detect various traffic related problems on roads via multiple CCTV footages, and alert the admin of warnings so that the required services can be dispatched accordingly. Moreover, we can also test the model on random images and videos to evaluate its performance or rung some analysis.

This project also aims at developing a real time system to detect traffic problems using previously deployed CCTV cameras on roads. The benefit of such a software is that it can significantly reduce the wait time for required traffic services to be sent at the particular location. The objective of this project is to develop a lightweight A.I. system that can monitor multiple CCTVs across the whole city/state/country which would incur very less computing cost. Our secondary focus in this project is to make the system more secure because it would be dealing with sensitive information like CCTV footage.

## 2. Analysis Phase

### 2.1 Use Cases

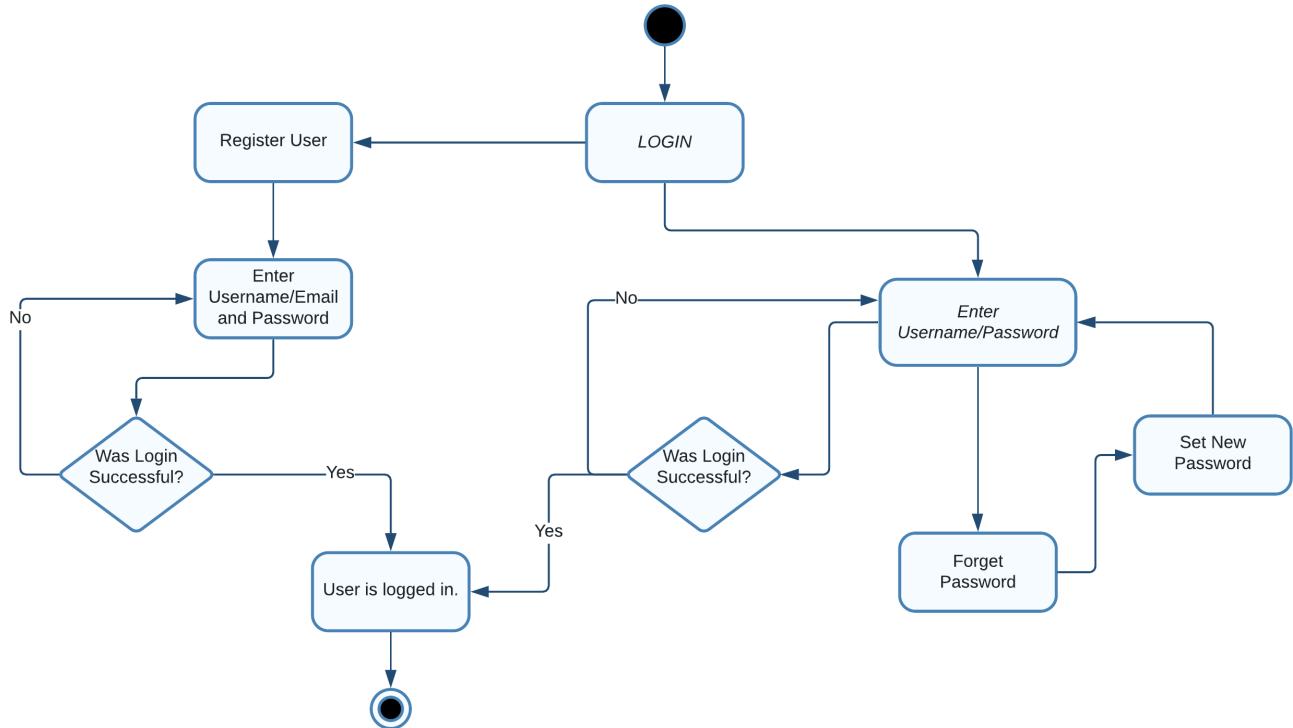
#### 2.1.1 Use-Case Diagram



The above use case diagram shows the interactions of actors like user and admin with other processes of the system including the login system, contact us page and image, video, CCTV footage. The above interactions shows us how the user can login as admin and contact with admin along with how the core functionality of the project works like image, video and CCTV livestream traffic problem detection.

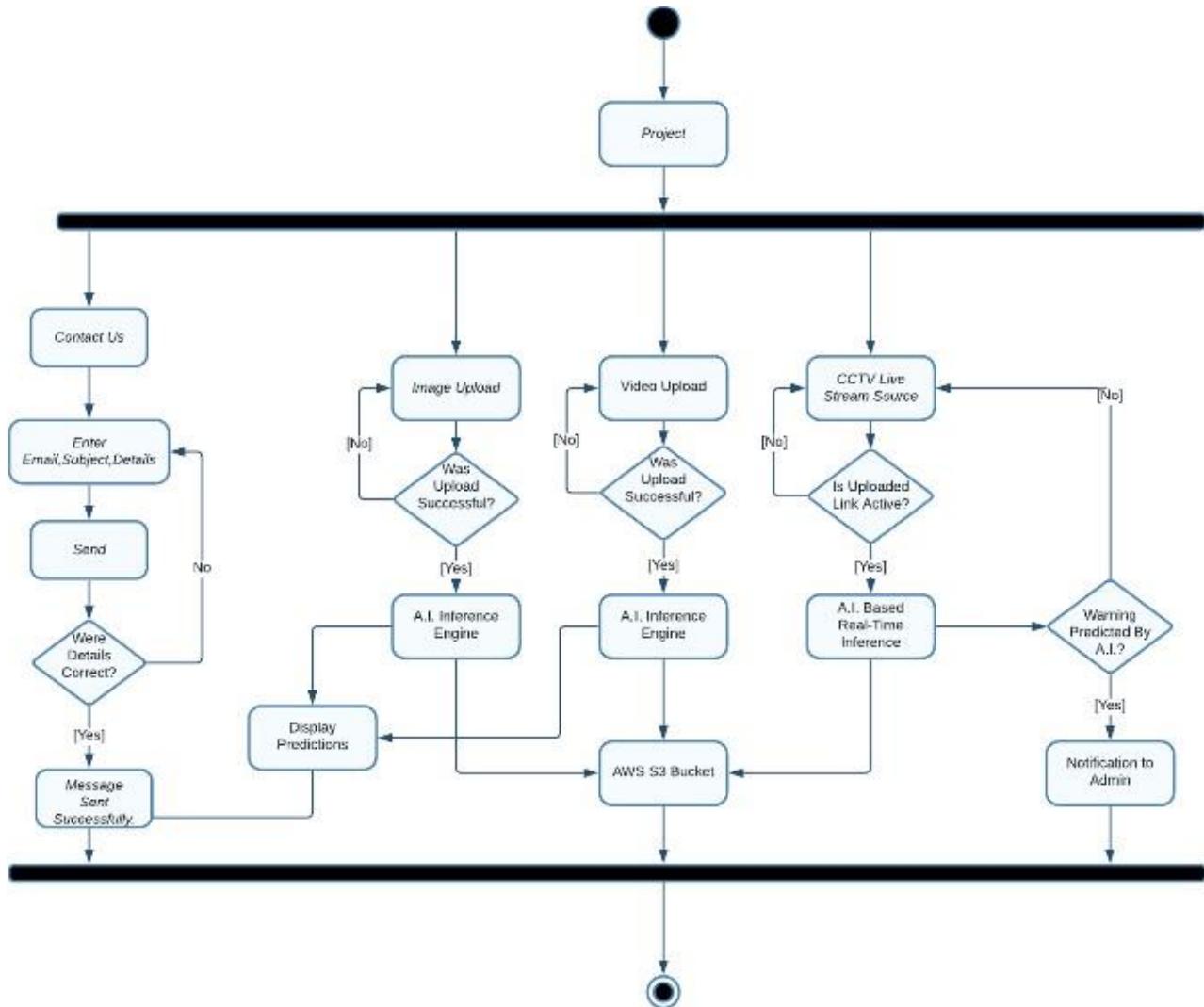
## 2.2 Activity diagram

### 2.2.1 Login Page



This activity diagram is regarding the login page and shows the flow of program at a higher level and it also lets us know about the constraints and conditions that are involved during the login process. For example decision like ‘was login successful’ and the flow of program if the given condition is true or false.

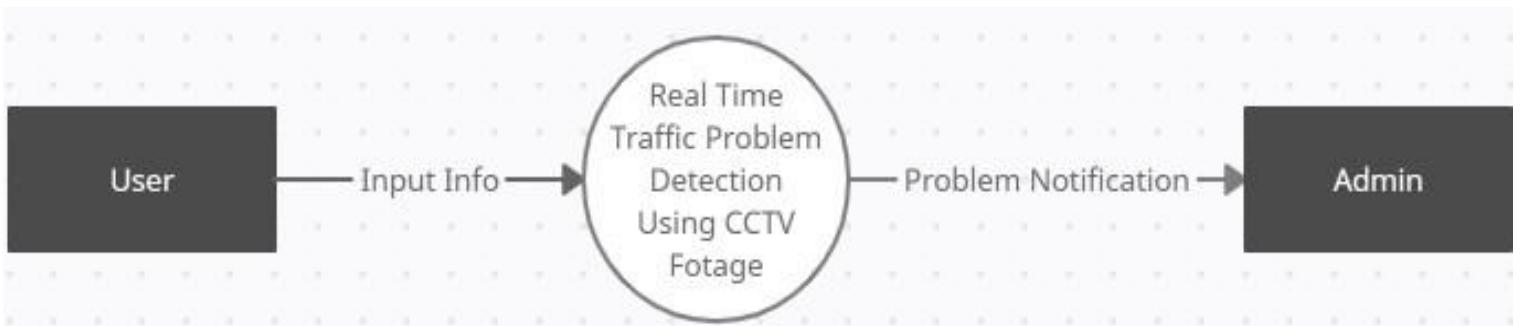
## 2.2.2 Website



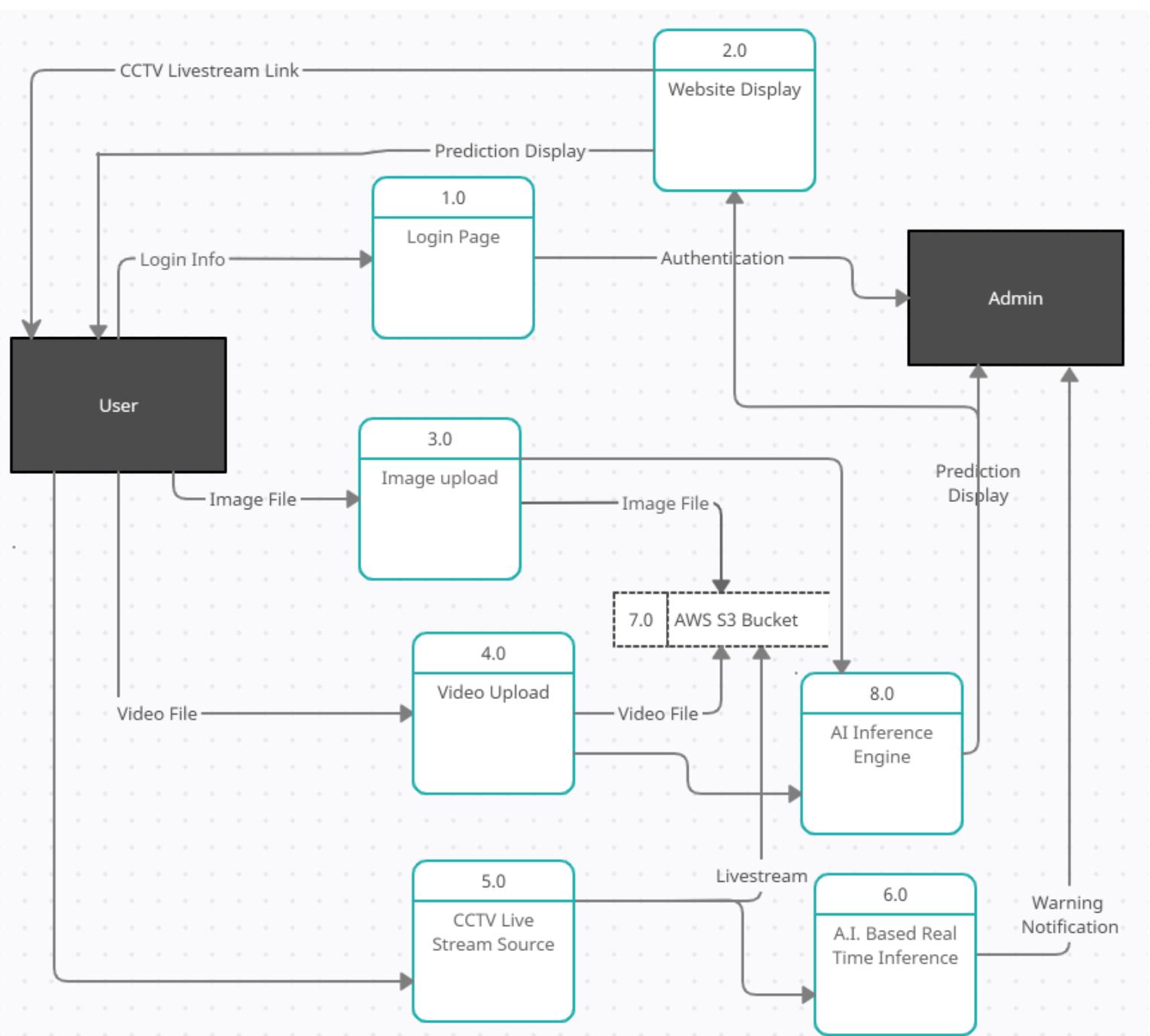
This activity diagram is regarding website except the login page and shows the flow of program at a higher level and it also lets us know about the constraints and conditions that are involved in using the website. For example decision like ‘was details correct’, ‘was upload successful’, ‘is link active’, etc and the flow of program if the given condition is true or false.

## 2.3 Data Flow Diagrams (DFDs)

### 2.3.1 DFD Level 0



### 2.3.2 DFD Level 1



The above level 1 data flow diagram highlights main functions of a system. Visualize incoming data flow, processes and output data flows. It notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an “exploded view” of the context diagram.

---

# **Software Requirements Specification**

**For**

**Real Time Traffic Problem Detection Using CCTV  
Footage**

**Version 1.0 approved**

**Prepared by**  
**KARTIK PANDEY - 101903504**  
**Mehul Sah - 101903522**  
**Chirag Singla - 101903500**  
**Satyajeet Singh - 101903498**

**Thapar Institute of Engineering and Technology**

**13/09/2021**

## Table of Contents

<b>Table of Contents .....</b>	<b>ii</b>
<b>Revision History .....</b>	<b>ii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Project Scope .....	1
1.5 References.....	1
<b>2. Overall Description.....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Features.....	2
2.3 User Classes and Characteristics .....	2
2.4 Operating Environment.....	2
2.5 Design and Implementation Constraints .....	2
2.6 User Documentation .....	2
2.7 Assumptions and Dependencies.....	3
<b>3. System Features .....</b>	<b>3</b>
3.1 System Feature 1 .....	3
3.2 System Feature 2 (and so on).....	4
<b>4. External Interface Requirements.....</b>	<b>4</b>
4.1 User Interfaces .....	4
4.2 Hardware Interfaces .....	4
4.3 Software Interfaces .....	4
4.4 Communications Interfaces .....	4
<b>5. Other Nonfunctional Requirements.....</b>	<b>5</b>
5.1 Performance Requirements .....	5
5.2 Safety Requirements .....	5
5.3 Security Requirements .....	5
5.4 Software Quality Attributes .....	5
<b>6. Other Requirements .....</b>	<b>5</b>
<b>Appendix A: Glossary.....</b>	<b>5</b>
<b>Appendix B: Analysis Models.....</b>	<b>6</b>
<b>Appendix C: Issues List.....</b>	<b>6</b>

## Revision History

Name	Date	Reason For Changes	Version

## 1. Introduction

### 1.1 Purpose

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionalities. This project aims at developing an A.I. Model to detect various traffic related problems on roads via multiple CCTV footages, and alert the admin of warnings so that the required services can be dispatched accordingly. Moreover, we can also test the model on random images and videos to evaluate its performance or run some analysis.

### 1.2 Document Conventions

The following are the list of conventions and acronyms used in this document:

- Font Style – Times New Roman
- Paragraph spacing – 1.5
- Left and Right Margin – 1 Inch
- Font Size of Heading – 18
- Font Size of Subheading – 14
  
- Admin - Used to represent a person with administrative privileges
- FPS – Frame Per Second
- A.I. Model – Artificially Intelligent Model
- RAM – Random Access Memory
- RSA – Rivest Shamir Adleman algorithm
- GPU – Graphical Processing Unit
- OS – Operating System

### 1.3 Intended Audience and Reading Suggestions

This document is made keeping in mind that it could be read by anyone interested in this domain, with some expertise in the field of A.I., web development, and app development. Although intended readers could be developer, officials, students, researchers, lawmakers. We recommend starting with introduction then reading the hardware/software requirements and seeing the use case diagrams to get a fair idea of the project, after this the reader can go through the whole document according to his/he needs.

### 1.4 Project Scope

This project aims at developing a real time system to detect traffic problems using previously deployed CCTV cameras on roads. The benefit of such a software is that it can significantly reduce the wait time for required traffic services to be sent at the particular location. The objective of this project is to develop a lightweight A.I. system that can monitor multiple CCTVs across the whole city/state/country which would incur very less computing cost. Our secondary focus in this project is to make the system more secure because it would be dealing with sensitive information like CCTV footage.

## **1.5 References**

- [\*A3D Benchmark \(Traffic Accident Detection\) | Papers With Code\*](#)
- [\*SA Benchmark \(Traffic Accident Detection\) | Papers With Code\*](#)
- [\*custom Benchmark \(Traffic Accident Detection\) | Papers With Code\*](#)
- [\*City Problems dataset\*](#)

## 2. Overall Description

### 2.1 Product Perspective

*With the advent of concept of digital India, we have seen a significant increase in surveillance devices across the whole country. But unfortunately, these devices are only used for minor tasks like number plate detection and certain other scenarios. What we propose is to an infrastructure above the current infrastructure, which would the currently deployed CCTV cameras on roads and neighborhoods to detect certain situations so that required service can be sent automatically without any human intervention. This product aims at collecting various links of livestream CCTV footages and using A.I. base deep learning model on them to monitor them for any certain event and sends warning to the admin in case any such event occurs.*

### 2.2 Product Features

- A secure system to hold sensitive information.
- Precise detection of trigger events real time.
- Storage of the images, videos, and live CCTV footages.
- Automatic notifications to admin.
- Security access rules for different accounts (read-only, review only, admin, etc).
- A way to contact the admin.

### 2.3 User Classes and Characteristics

*Users of this system would fall majorly into the following categories – Person monitoring emergency services, developers, researchers and lawmakers. The users of this system must be restricted because this system deals with sensitive information that everyone should not have access to. The technical experience required for the users is very minimal and is limited to knowing how to upload an image link or video on website. The security and privilege level required depends on the lawmakers who decide on how the project could be implemented. But on a higher level there would be a person who would be given admin rights and he/she would be the one to get the notifications/warnings from the live CCTV footage.*

### 2.4 Operating Environment

*The website could be easily accessible through the internet and there are no such restrictions on the type of web browser the user can use. Although we recommend to use the latest version of the web browser. In case GPU servers are to be deployed locally then there continuous cooling should be monitored and temperatures should not rise too high, as it might damage the GPUs.*

### 2.5 Design and Implementation Constraints

*Items or issues that would limit the options available to the developers could be:*

- The GPU requirements in order to implement the project.
- Development of an extremely lightweight A.I. model.
- Security standards that would be deployed to maintain confidentiality of sensitive data.
- Requirement of high storage space.
- Acquiring permission from government for deploying the project.

## **2.6 User Documentation**

*Various tutorials will be provided along with the project in order to teach the users how to use the system along with a ‘contact us’ feature to talk directly to the developers or the admin. And various user manuals would also be created and provided.*

## **2.7 Assumptions and Dependencies**

- *Acquiring permission from the lawmakers regarding the use of live CCTV footage.*
- *Acquiring the required resources like GPU servers and storage space.*

### 3. System Features

This section illustrates the features in the system and explains how they are used and what will be the output to the users/admin.

#### 3.1 Image/Video Upload

##### 3.1.1 Description and Priority

This feature gives user the control to upload sample images/videos in order to perform analysis or to check the accuracy of the A.I model. The priority level of this feature is medium. The cost level of this function is around 4. And the benefit level is around 5.

##### 3.1.2 Stimulus/Response Sequences

The user first has to upload the images/videos and after uploading the system will authenticate those images/videos and after authentication they would be sent to the A.I. model after which the corresponding predictions would be displayed to the user.

##### 3.1.3 Functional Requirements

It includes the requirement of storage space (REQ-1) and GPUs (REQ-2) in order to store the images and videos and perform computation on them respectively. A server (REQ-3) would also be required to authenticate the uploaded images or videos.

#### 3.2 CCTV Link Upload

##### 3.2.1 Description and Priority

This feature gives user the control to upload CCTV link which would be used to stream realtime footage. The priority level of this feature is high. The cost level of this function is around 9. And the benefit level is around 9.

##### 3.2.2 Stimulus/Response Sequences

The user first has to upload the CCTV link and after uploading the system will authenticate those CCTV links and after authentication, they would be sent to the A.I. model after which the corresponding real time altered footage would be displayed to the user. In case of trigger event, a warning would be sent to the admin.

##### 3.2.3 Functional Requirements

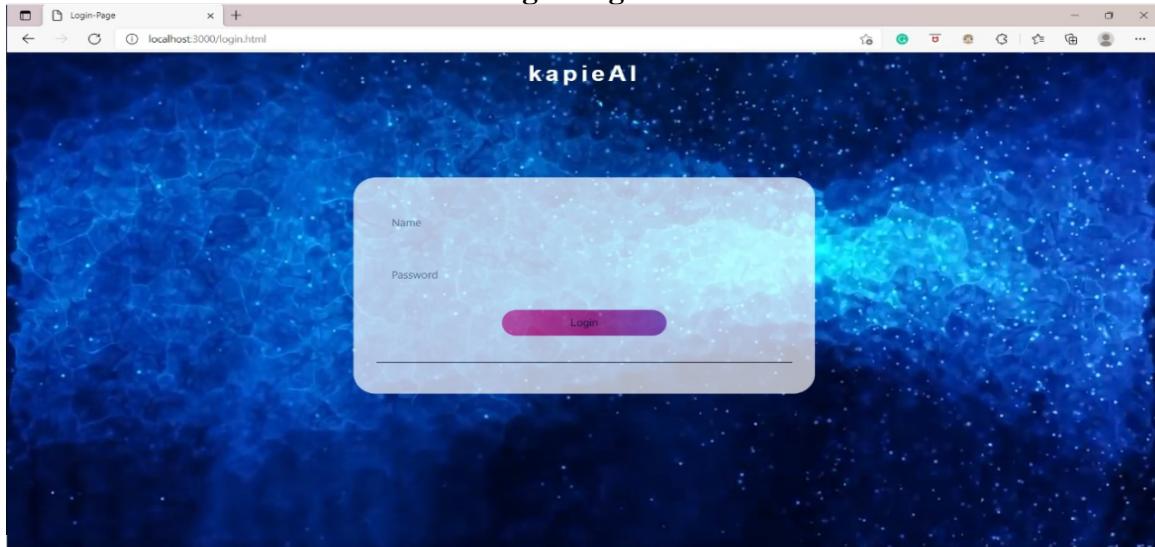
It includes the requirement of storage space (REQ-1) and GPUs (REQ-2) in order to store the link and real time footage and to perform computation on them respectively. A server (REQ-3) would also be required to authenticate the uploaded link.

## 4. External Interface Requirements

### 4.1 User Interfaces

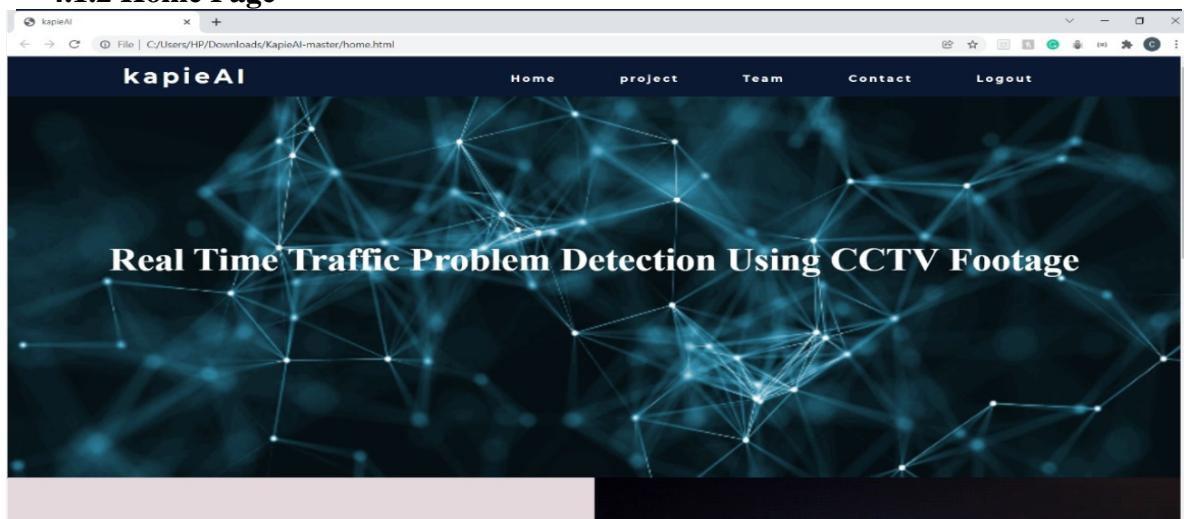
The website should have authentic feel to it with hues of purple and classic fonts. The UI design is made by using Google's Material UI. The website should be responsive and would be using a top navigation bar for navigating to different pages.

#### 4.1.1 Login Page



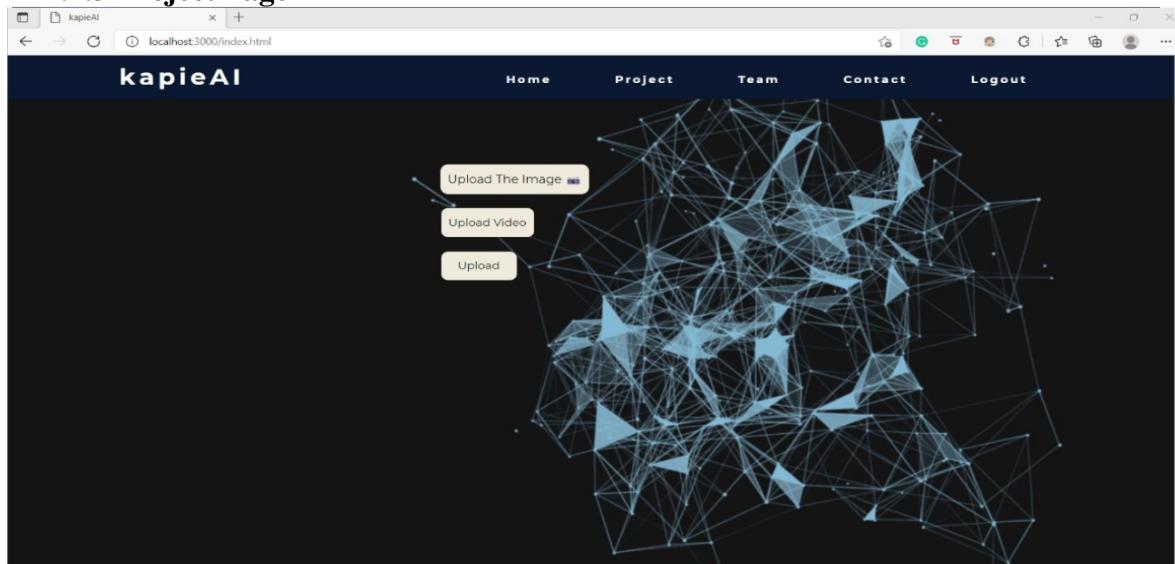
The login page consists of two user details: Name and Password along with login button (TBD).

#### 4.1.2 Home Page



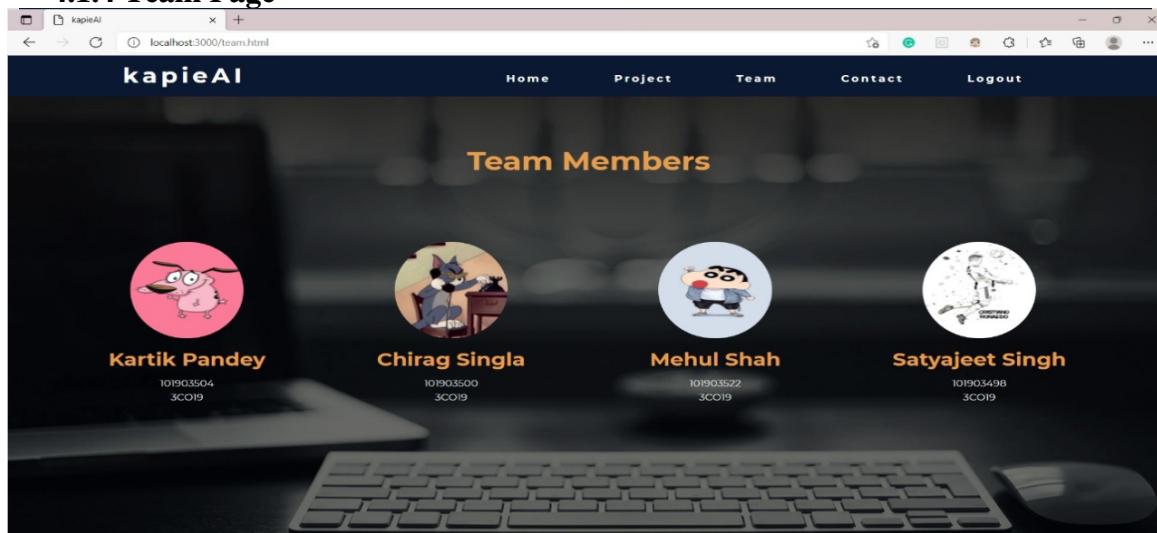
The home page consists of all details regarding the project along with tutorial and user manuals (TBD).

#### 4.1.3 Project Page



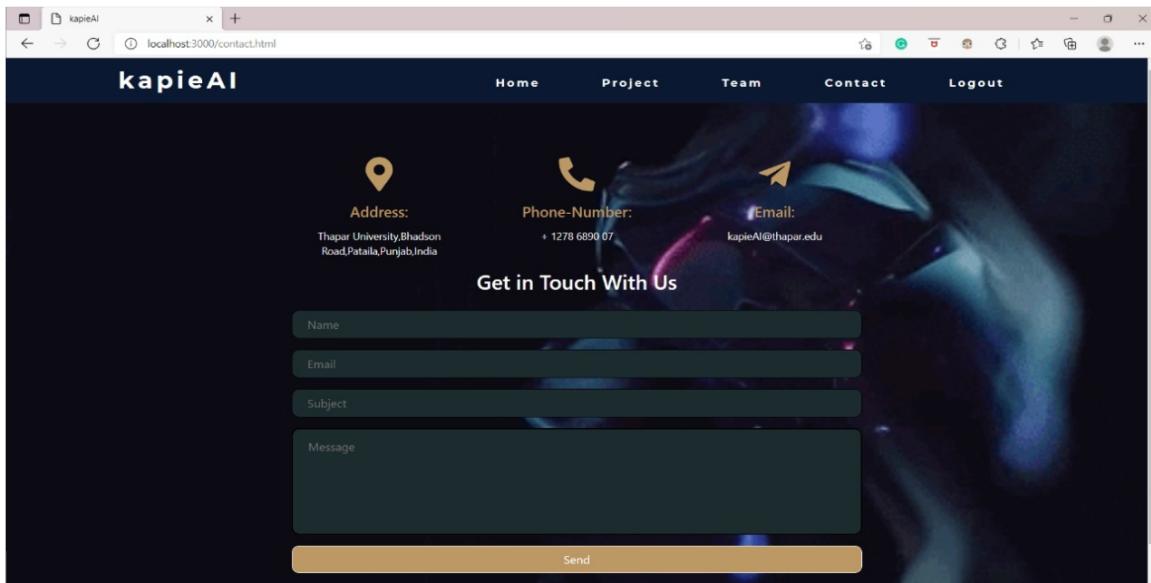
The project page is under development and will eventually consist of three upload buttons. First two for image and video upload and third for CCTV link upload (TBD).

#### 4.1.4 Team Page



The team page consists of details of all the team members (TBD).

#### 4.1.5 Contact Us Page



The contact us page is present to help user to send any query or message to the admin or developers (TBD).

#### 4.2 Hardware Interfaces

The system will require CCTV footage with the help of deep learning. Real time location data would also be required through GPS which would be inside CCTV cameras. The user can access this website via any mobile, tablet or pc with a supported web browser.

Requirements for client:

- Mobile, tablet or pc with a supported web browser.
- Minimum RAM requirements (TBD).

Requirements for server:

- Fault tolerant, reliable, energy efficient.
- Minimum storage space of 100gb.
- High end GPUs.

#### 4.3 Software Interfaces

Requirements for client:

- Mobile, tablet or pc with a supported web browser.

Requirements for server:

- LINUX/UNIX based OS.
- All outbound requests allowed.
- Required A.I./Web packages installed.

#### 4.4 Communications Interfaces

The only requirement for communication interface is to maintain the encryption and security standards for login credentials and uploaded file media respectively.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

The performance requirements for this project include:

- Real time predictions with an FPS of around 30.
- Per instance RAM requirement of around 10GB.
- Accuracy of A.I. model above 90%.

### 5.2 Safety Requirements

The safety requirements for this project include:

- Accurate predictions from A.I. model

### 5.3 Security Requirements

The safety requirements for this project include:

- Encryption of login credentials and uploaded file media using RSA encryption.
- Acquiring permission from law makers.

### 5.4 Software Quality Attributes

The software quality attributes for this project include:

- Accuracy of model should be high.
- Model should be saleable and robust.
- Model must support online learning.
- Model must be adaptive under different traffic environment.

## Appendix A: Glossary

FPS – Frame Per Second

A.I. Model – Artificially Intelligent

ModelRAM – Random Access

Memory

RSA – Rivest Shamir Adleman

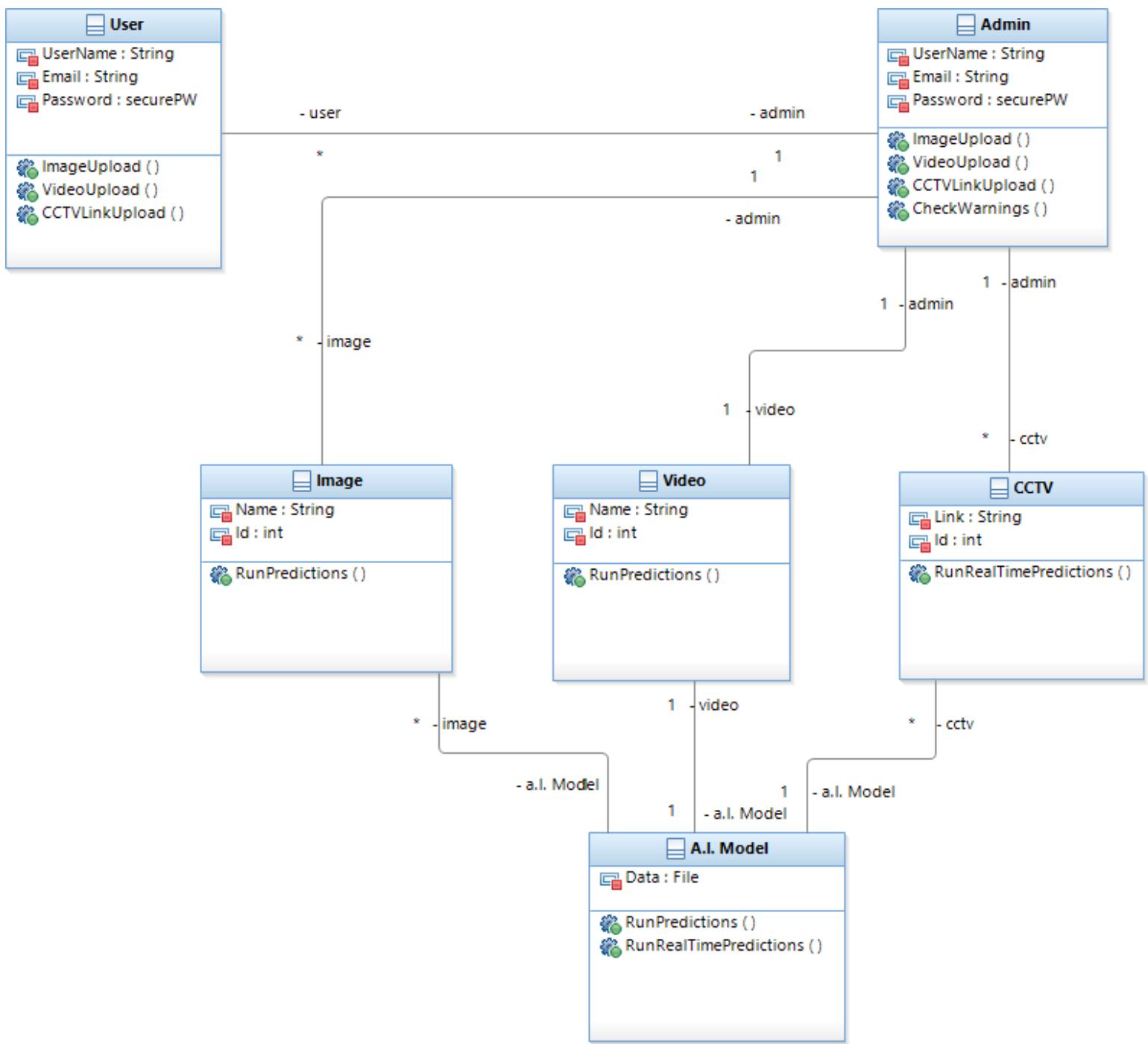
algorithmGPU – Graphical

Processing Unit

OS – Operating System

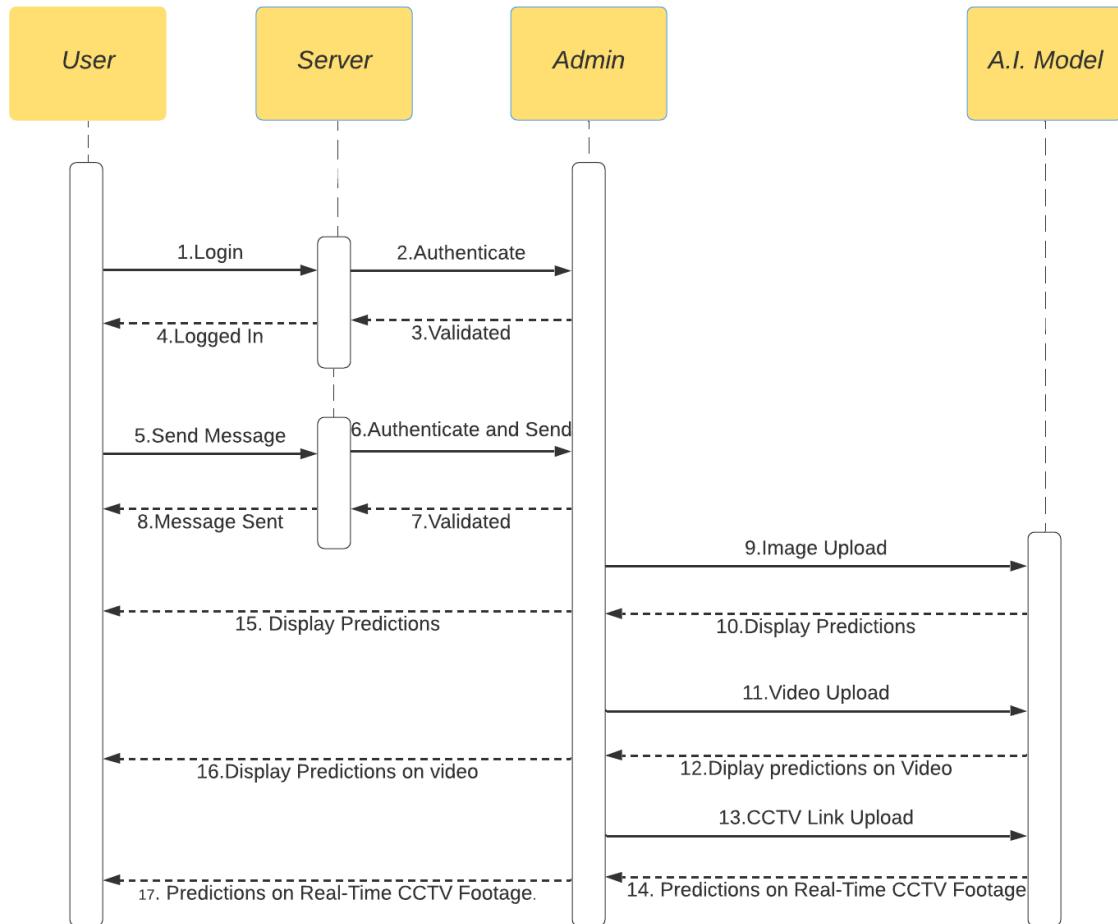
### 3. Design Phase

#### 3.1 Class Diagram



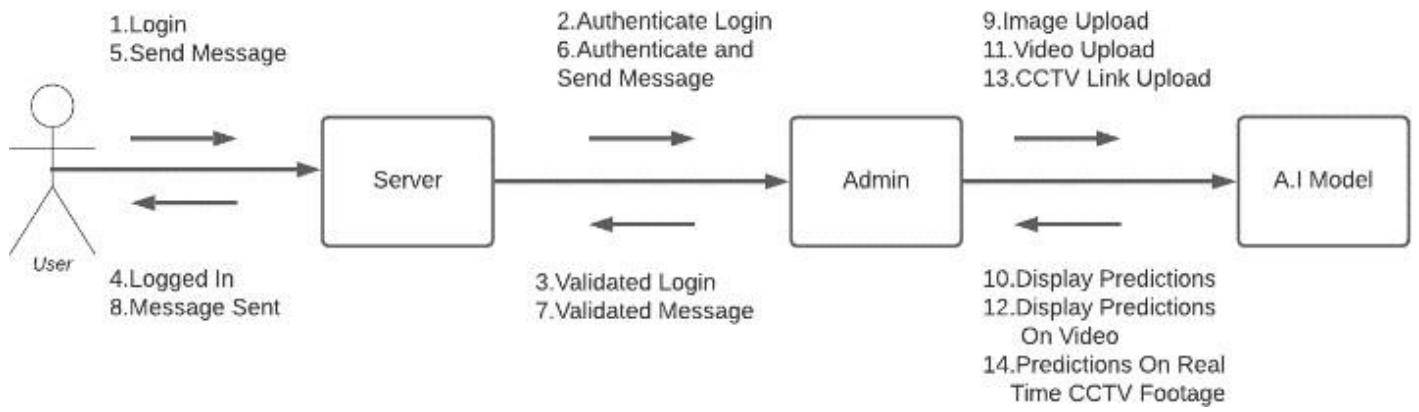
The above class diagrams act as the blueprints of your system or subsystem. Class diagrams are used to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. Class diagrams are useful in many stages of system design.

### 3.2 Sequence Diagram



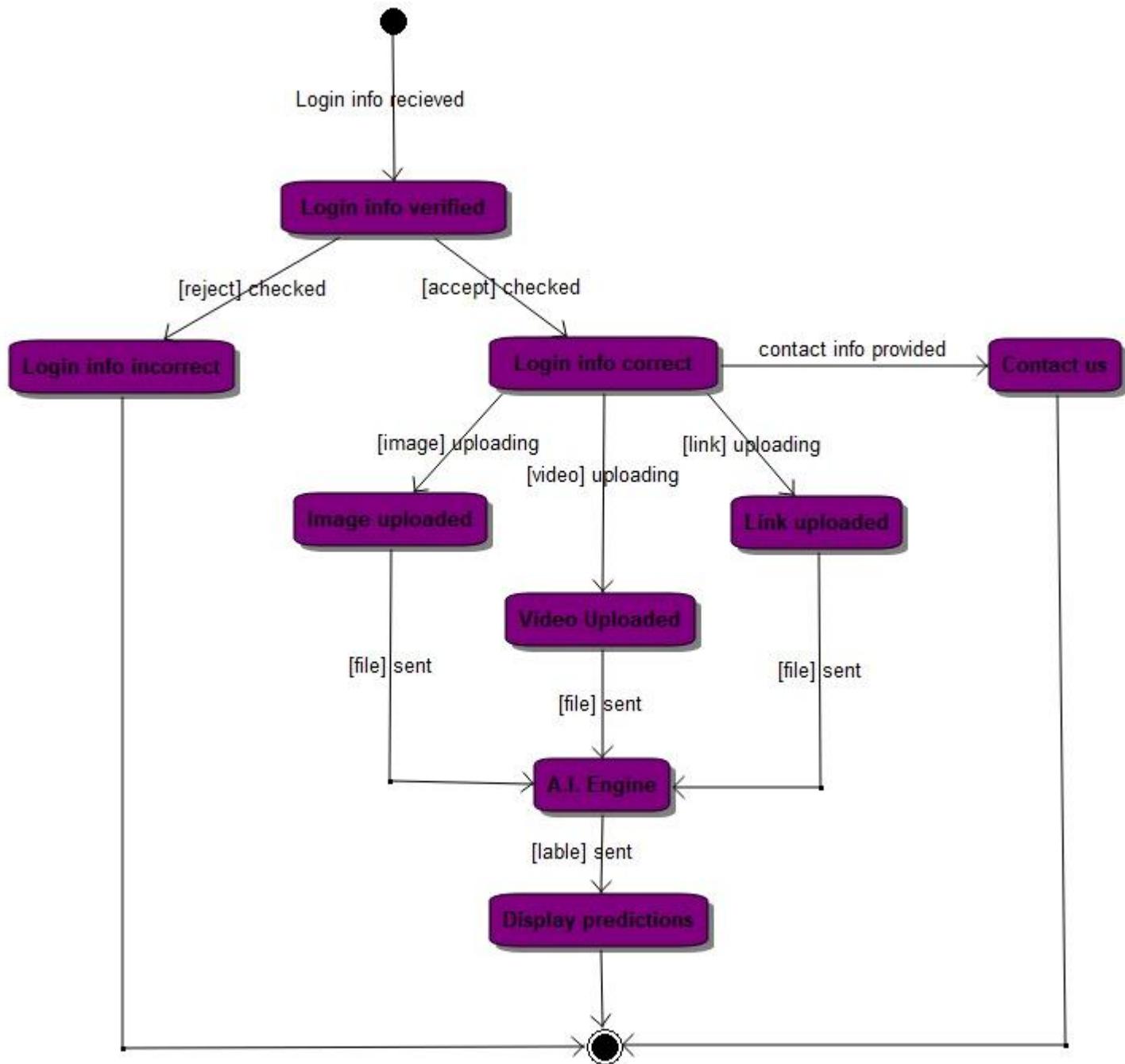
The above sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects.

### 3.3 Collaboration Diagram



The above collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

## 3.4 State Chart Diagram

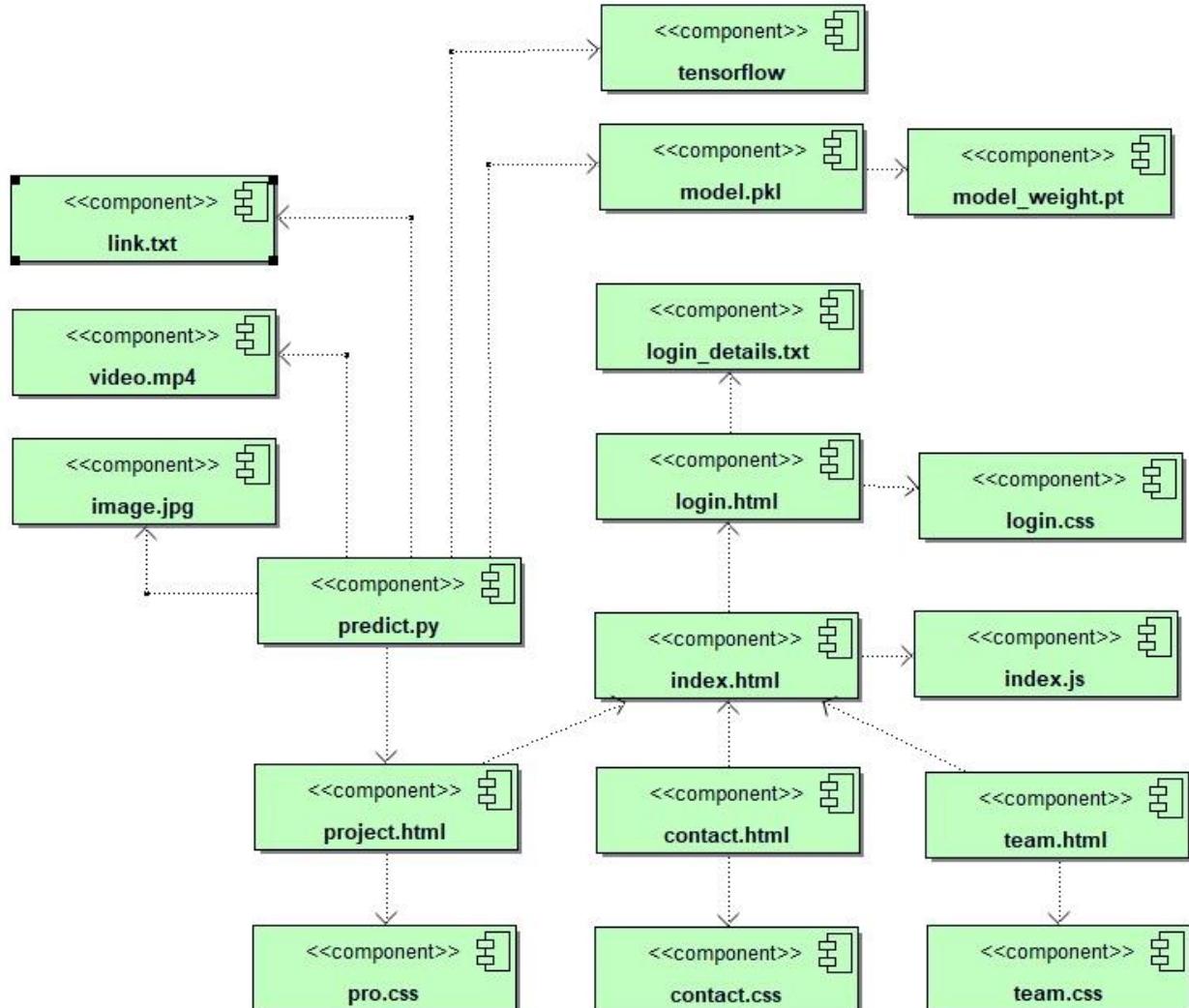


State chart diagram is used to model the dynamic nature of a system. It defines different states of an object during its lifetime and these states are changed by events. State chart diagram is useful for modelling the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

It also describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

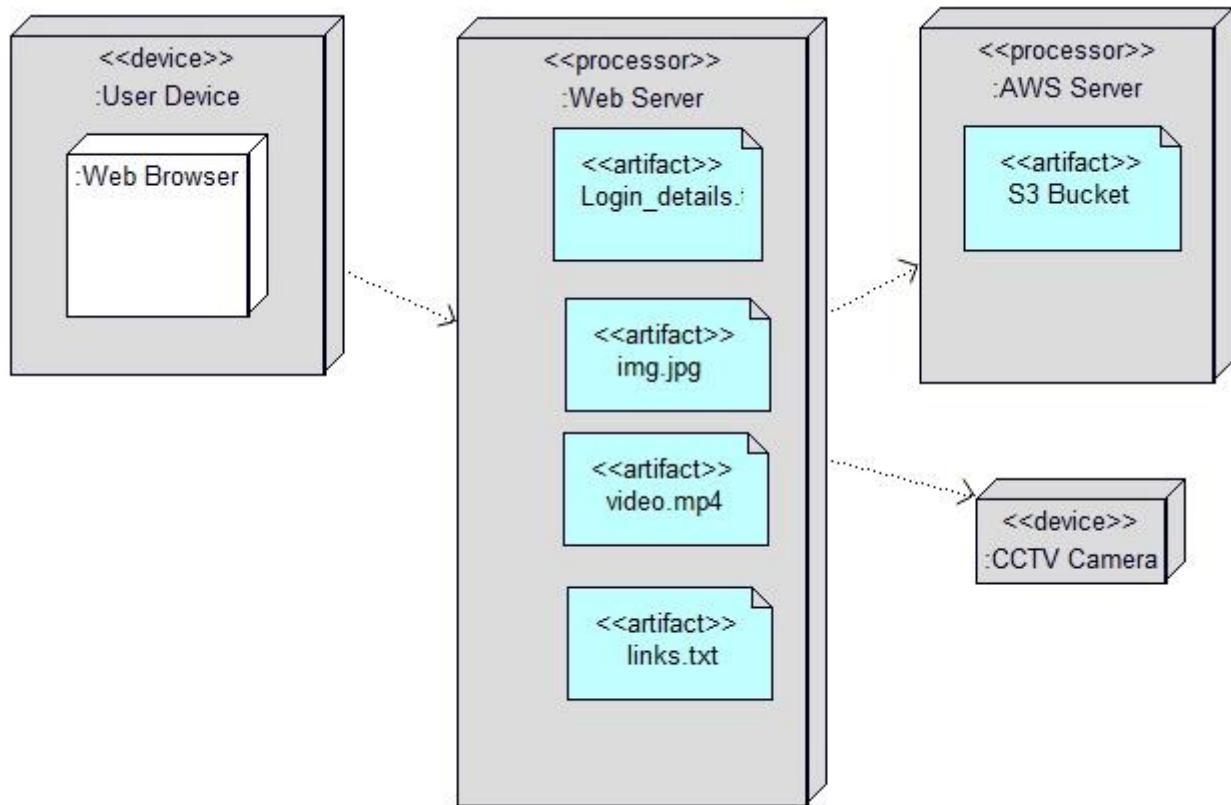
## 4. Implementation

### 4.1 Component Diagrams



Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

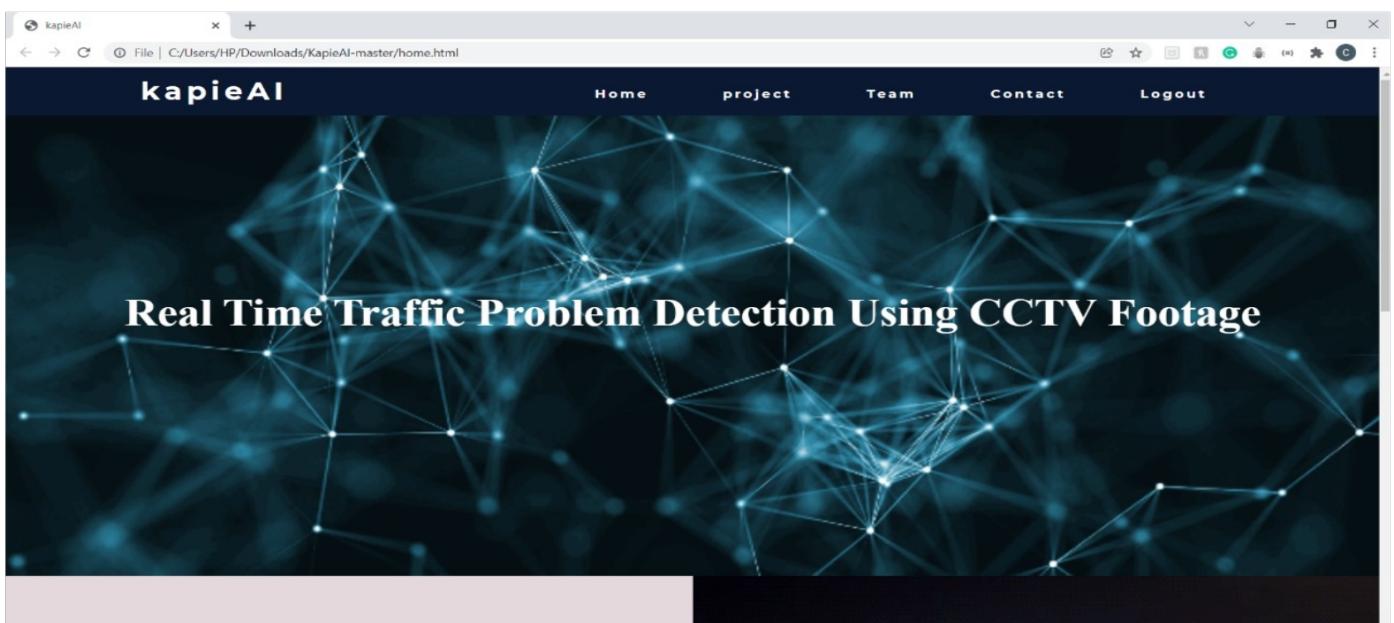
## 4.2 Deployment Diagrams



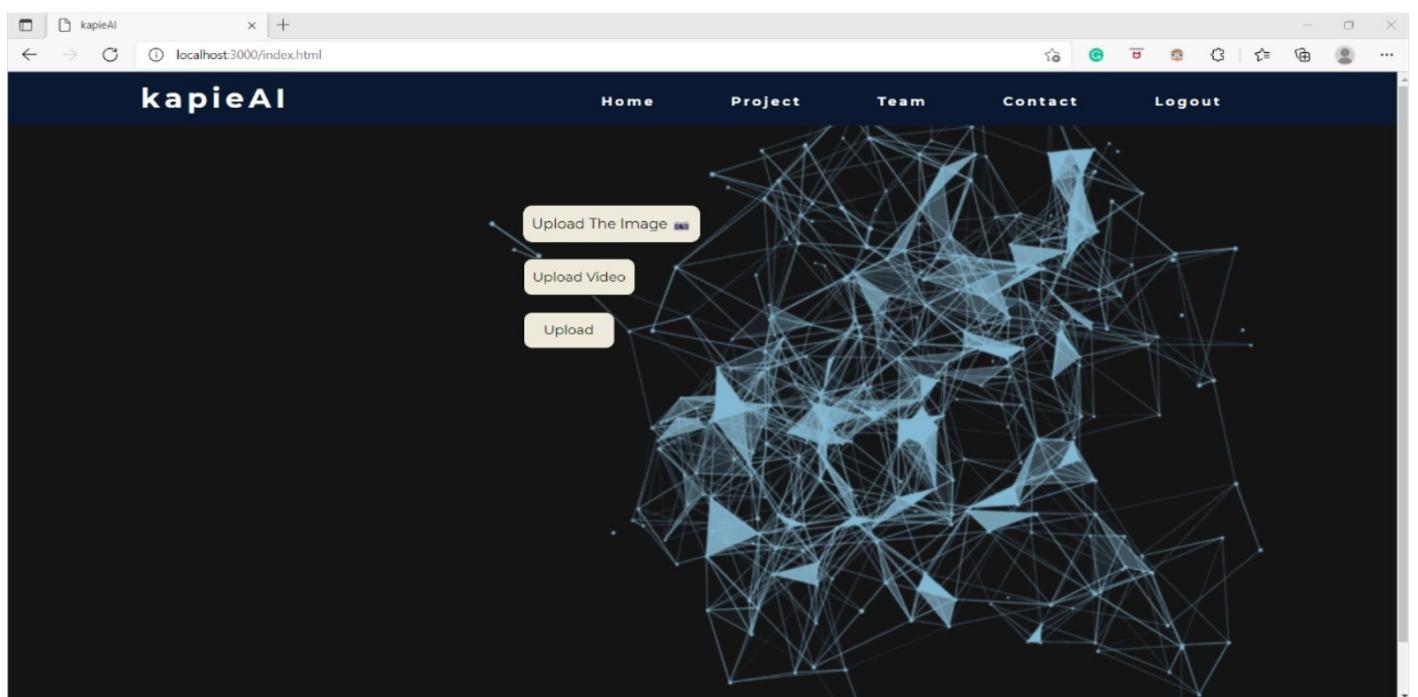
Deployment diagram shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.

## 4.3 Screen Shots

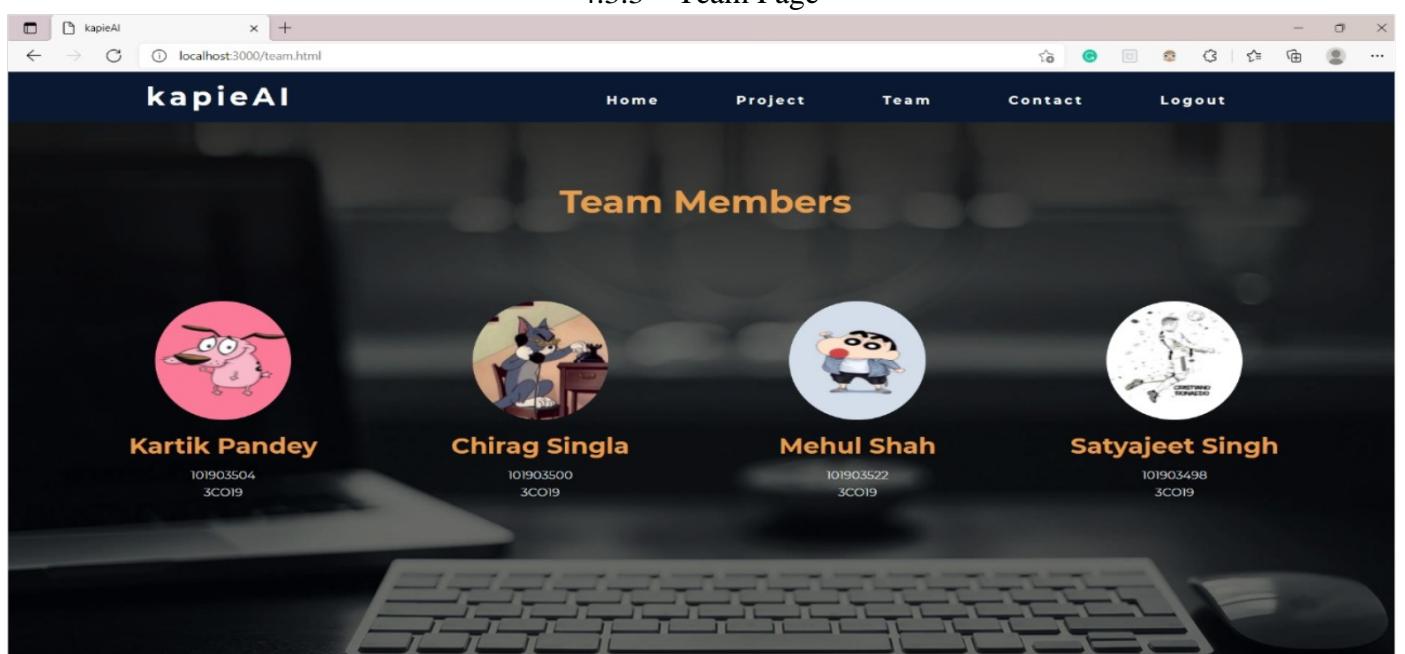
### 4.3.1 Home Page



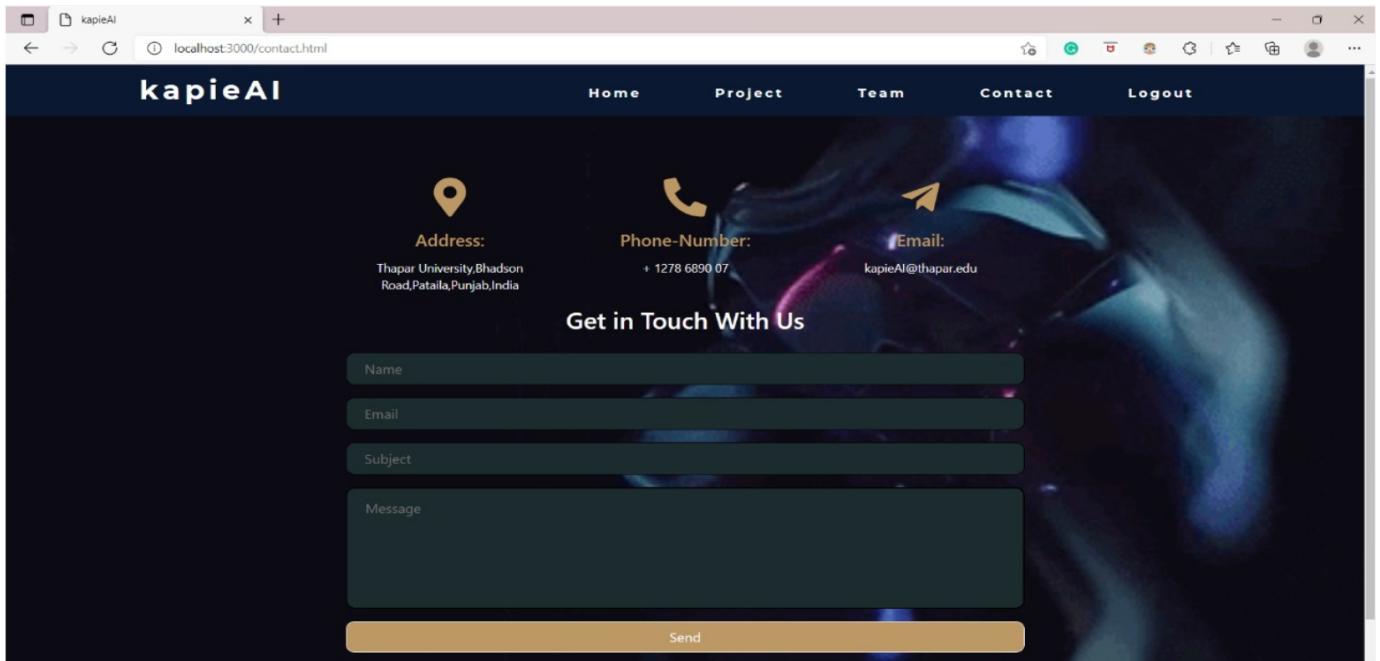
#### 4.3.2 Project Page



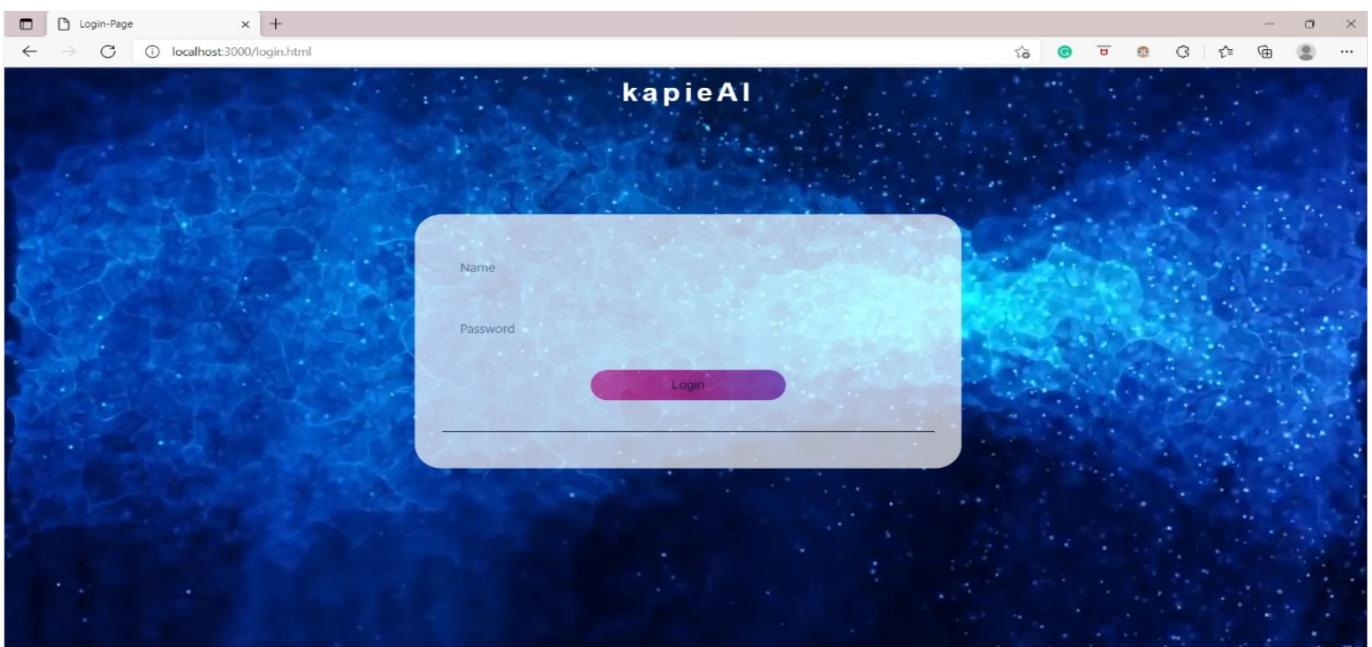
#### 4.3.3 Team Page



#### 4.3.4 Contact Page



#### 4.3.5 Login Page



## 5. Testing

### Test Case (Doc:T\_01)

**Test Case #:** 1

**System:** Real time city problem detection

**Designed by:** KAPIE

**Executed by:** KARTIK PANDEY

**Short Description:** Testing the login page

**Test Case Name:** Login user

**Page:** 1 of 5

**Subsystem:** Login

**Design Date:** 10/09/2021

**Execution Date:** 20/10/2021

#### Pre-conditions

User has valid login credentials

Step	Action	Expected System Response	Pass/ Fail	Comment
1	User enters login credentials	Details are correct is displayed	Pass	
2	User presses login button	The user logs in successfully if correct details are provided, otherwise user has to re-enter login credentials	Pass	
3	Homepage displayed	The homepage of website is displayed after successful login	Pass	

#### Post-conditions

If login details were not previously saved, then it is now saved

# Test Case (Doc:T\_02)

**Test Case #:** 2

**System:** Real time city problem detection

**Designed by:** KAPIE

**Executed by:** KARTIK PANDEY

**Short Description:** Testing the contact us page

**Test Case Name:** Contacting admin

**Page:** 2 of 5

**Subsystem:** Contact

**Design Date:** 11/09/2021

**Execution Date:** 21/10/2021

## Pre-conditions

User should be logged in

Contact details provided by user should be correct

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Provide contact details and message	N.A.	Pass	
2	Press the send button	Message sent successfully is displayed if provided details were correct, otherwise please provide correct details is displayed	Pass	

## Post-conditions

Message and contact details are saved and sent to admin

# Test Case (Doc:T\_03)

**Test Case #:** 3

**System:** Real time city problem detection

**Designed by:** KAPIE

**Executed by:** KARTIK PANDEY

**Short Description:** Testing the images upload

**Test Case Name:** Images uploading

**Page:** 3 of 5

**Subsystem:** Images

**Design Date:** 12/09/2021

**Execution Date:** 22/10/2021

## Pre-conditions

User should be logged in

Images should not be corrupt

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Press upload button	Opens browse dialogue box	Pass	
2	Browse and select the images	Images are uploaded message is displayed	Pass	
3	Press test button	Corresponding predictions are displayed	Pass	

## Post-conditions

N.A.

# Test Case (Doc:T\_04)

**Test Case #:** 4

**System:** Real time city problem detection

**Designed by:** KAPIE

**Executed by:** KARTIK PANDEY

**Short Description:** Testing the video upload

**Test Case Name:** Video uploading

**Page:** 4 of 5

**Subsystem:** Video

**Design Date:** 13/09/2021

**Execution Date:** 23/10/2021

## Pre-conditions

User should be logged in

Videos should not be corrupt

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Press upload button	Opens browse dialogue box	Pass	
2	Browse and select the video	Video is uploaded message is displayed	Pass	
3	Press test button	Corresponding predictions are displayed	Pass	

## Post-conditions

N.A.

# Test Case (Doc:T\_05)

**Test Case #:** 5

**System:** Real time city problem detection

**Designed by:** KAPIE

**Executed by:** KARTIK PANDEY

**Short Description:** Testing the links upload

**Test Case Name:** Links uploading

**Page:** 5 of 5

**Subsystem:** Links

**Design Date:** 14/09/2021

**Execution Date:** 24/10/2021

## Pre-conditions

User should be logged in

Link should be correct

CCTV camera should support live feed through some webpage

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter the link for live CCTV footage	Links are uploaded message is displayed	Pass	
2	Press test button	Corresponding predictions are displayed	Pass	

## Post-conditions

Links are saved in a file