

**University of Mumbai**

**LAN MONITORING**

Submitted in partial fulfillment of requirements

For the degree of

**Bachelors in Technology**

by

**Rushabh Bid  
1514009**

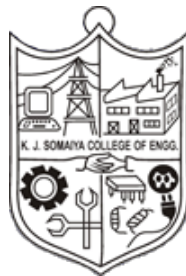
**Yash Kandalam  
1514023**

**Pratyusha Reddy  
1514041**

**Yash Jain  
1514017**

Guide

**Prof. Ravindra Divekar**



**Department of Information Technology  
K. J. Somaiya College of Engineering, Mumbai-77  
(Autonomous College Affiliated to University of Mumbai)**

**Batch 2015 -2019**

# **K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

## **Certificate**

This is to certify that the dissertation report entitled **LAN MONITORING** is bona fide record of the dissertation work done by

1. **Rushabh Bid**
2. **Yash Kandalam**
3. **Pratyusha Reddy**
4. **Yash Jain**

in the year 2018-19 under the guidance of **Prof. Ravindra Divekar** of Department of Information Technology in partial fulfilment of requirement for the Bachelors in Technology degree in Information Technology of University of Mumbai.

---

Guide

---

Head of the Department

---

Principal

Date:

Place: Mumbai-77

## **K. J. Somaiya College of Engineering, Mumbai-77**

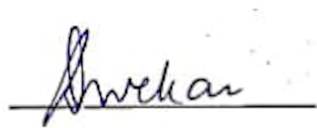
(Autonomous College Affiliated to University of Mumbai)

### **Certificate of Approval of Examiners**

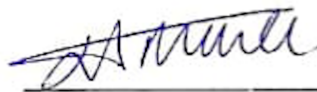
We certify that this dissertation report entitled **Implementation of Supervised Learning Techniques for Text Classification** is bona fide record of project work done by

1. **Rushabh Bid**
2. **Yash Kandalam**
3. **Pratyusha Reddy**
4. **Yash Jain**

This project is approved for the award of Bachelors in Technology Degree in Information Technology of University of Mumbai.



Internal Examiner



External Examiner

Date: 08/05/19

Place: Mumbai-77

# **K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

## **DECLARATION**

We declare that this written thesis submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in our submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

<div>Signature of the Student</div> <div>1514009</div> <div>Roll No.</div>	<div>Signature of the Student</div> <div>1514023</div> <div>Roll No.</div>
<div>Signature of the Student</div> <div>1514041</div> <div>Roll No.</div>	<div>Signature of the Student</div> <div>1514017</div> <div>Roll No.</div>

**Date:**

**Place: Mumbai-77**

## **Abstract**

Network surveillance has become a necessity with the increase in cybercrime. Monitoring the activities of the devices on a network is paramount to the security of an organization. It is difficult for a novice user to understand the root levels of LAN monitoring through packets sniffing. In order to remove this complexity, a simple UI driven application which will enable the administrator to observe, manage and analyze the activities of the client machines is proposed. While preventing cybercrime the system also aims to create a discipline in the activities of the users working in the organization.

**Key words:** LAN, monitoring, networks, OCR, administration.

# Contents

<b>List of Figures</b>	<b>i</b>
<b>List of Tables</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Definition . . . . .	1
1.2 Motivation . . . . .	1
1.3 Scope . . . . .	2
1.3.1 Functional Requirements: . . . . .	2
1.3.2 Non Functional Requirements: . . . . .	2
1.4 Salient Contribution . . . . .	2
1.5 Organization of the Synopsis . . . . .	2
<b>2 Literature Survey</b>	<b>4</b>
2.1 Introduction . . . . .	4
2.2 Methods . . . . .	4
2.3 Discussion . . . . .	5
2.3.1 Android Based LAN Monitoring . . . . .	5
2.3.2 GPRS Based LAN Monitoring and Controlling . . . . .	5
2.3.3 LAN Monitoring using Remote Method Invocation(Java RMI) . . . . .	5
2.3.4 Comparison of different LAN Monitoring software . . . . .	6
2.4 Conclusion . . . . .	6
<b>3 Software Project Management Plan</b>	<b>8</b>
3.1 Introduction . . . . .	8

3.1.1	Project Overview	8
3.1.2	Project Deliverables	9
3.2	Project Organization	10
3.2.1	Software Process Model	10
3.2.2	Roles and Responsibilities	11
3.2.3	Tools And Techniques	11
3.3	Project Management plan	12
3.3.1	Tasks	12
3.3.1.1	Task Description	12
3.3.1.2	Deliverable and Milestones	12
3.3.1.3	Resources Needed	13
3.3.1.4	Dependencies and Constraints	13
3.3.1.5	Risk And Contingencies	13
3.3.2	Timetable	14
<b>4</b>	<b>Software Requirement Specification</b>	<b>15</b>
4.1	Introduction	15
4.1.1	Product Overview	15
4.2	Specification Requirement	15
4.2.1	External Interface Requirements	16
4.2.1.1	User Interfaces	16
4.2.1.2	Hardware Interfaces	16
4.2.1.3	Software Interfaces	17
4.2.1.4	Communication Protocols	17
4.2.2	Software Product Features	17
4.2.2.1	Monitoring Through Screen Shots	17
4.2.2.2	Request Live Screen Shot	17
4.2.2.3	Shutdown Client	17
4.2.2.4	Shutdown All	18
4.2.2.5	Message Individual Client	18
4.2.2.6	Broadcast Message	18

4.2.2.7	Enter Keywords	18
4.2.2.8	Calculate Relevance	18
4.2.2.9	Disconnect Client	18
4.2.2.10	Reconnect Client	18
4.2.2.11	Storage of Screenshots in Folder	19
4.2.3	Software System Attributes	19
4.2.3.1	Reliability	19
4.2.3.2	Availability	19
4.2.3.3	Security	19
4.2.3.4	Maintainability	19
4.2.3.5	Portability	19
4.2.3.6	Performance	20
4.2.4	Software Quality Attributes	20
<b>5</b>	<b>Software Design Description</b>	<b>21</b>
5.1	Introduction	21
5.1.1	Design overview	21
5.1.2	Requirements Traceability Matrix	22
5.2	System Architectural Design	22
5.2.1	Chosen System Architecture	22
5.2.2	Discussion of alternate Design	24
5.2.3	System Interface Description	24
5.2.3.1	Use case diagram	24
5.2.3.2	Sequence Diagram	25
5.2.3.3	State chart diagram	25
5.2.3.4	Class Diagram	26
5.2.3.5	Deployment diagram	26
5.2.3.6	Software Component Diagram	27
5.2.3.7	Object diagram	27
5.2.3.8	Level Zero DFD	28
5.2.3.9	Level One DFD	28



<b>6</b>	<b>Implementation</b>	<b>29</b>
6.1	Technologies Used	29
6.1.1	Socket Programming	29
6.1.2	Tesseract - OCR	29
6.1.3	Flask - Web Framework	30
6.1.4	Python	30
6.2	Algorithm	31
6.2.1	Score Generation	31
6.3	Implementation	32
<b>7</b>	<b>Software Test Document</b>	<b>35</b>
7.1	Introduction	35
7.1.1	System Overview	35
7.1.2	Test Approach	35
7.2	Introduction	36
7.2.1	Features to be tested	36
7.2.2	Features not to be tested	36
7.2.3	Testing Tools and Environment	36
7.2.3.1	Testing Tools	36
7.2.3.2	Testing Environment	36
7.3	Test Cases	37
7.3.1	Test case ID: TC-0001	37
7.3.1.1	Purpose	37
7.3.1.2	Input	37
7.3.1.3	Expected outputs and Pass/Fail criteria	37
7.3.1.4	Test procedure	37
7.3.2	Test case ID: TC-0002	37
7.3.2.1	Purpose	37
7.3.2.2	Input	37
7.3.2.3	Expected outputs and Pass/Fail criteria	37
7.3.2.4	Test procedure	38

7.3.3	Test case ID: TC-0003	38
7.3.3.1	Purpose	38
7.3.3.2	Input	38
7.3.3.3	Expected outputs and Pass/Fail criteria	38
7.3.3.4	Test procedure	38
7.3.4	Test case ID: TC-0004	38
7.3.4.1	Purpose	38
7.3.4.2	Input	39
7.3.4.3	Expected outputs and Pass/Fail criteria	39
7.3.4.4	Test procedure	39
7.3.5	Test case ID: TC-0005	39
7.3.5.1	Purpose	39
7.3.5.2	Input	39
7.3.5.3	Expected outputs and Pass/Fail criteria	39
7.3.5.4	Test procedure	39
7.3.6	Test case ID: TC-0006	40
7.3.6.1	Purpose	40
7.3.6.2	Input	40
7.3.6.3	Expected outputs and Pass/Fail criteria	40
7.3.6.4	Test procedure	40
7.3.7	Test case ID: TC-0007	40
7.3.7.1	Purpose	40
7.3.7.2	Input	40
7.3.7.3	Expected outputs and Pass/Fail criteria	40
7.3.7.4	Test procedure	41
7.3.8	Test case ID: TC-0008	41
7.3.8.1	Purpose	41
7.3.8.2	Input	41
7.3.8.3	Expected outputs and Pass/Fail criteria	41
7.3.8.4	Test procedure	41

<b>8 Conclusion and Future work</b>	<b>42</b>
8.1 Conclusion . . . . .	42
8.2 Scope for Future Work . . . . .	43
<b>References</b>	<b>44</b>
<b>Author's Publication</b>	<b>45</b>
<b>Acknowledgements</b>	<b>50</b>

# List of Figures

3.1 Waterfall Life Cycle . . . . .	10
3.2 Gantt Chart . . . . .	14
5.1 Client-Server Architecture . . . . .	23
5.2 Use Case Diagram . . . . .	24
5.3 Sequence Diagram . . . . .	25
5.4 State Chart Diagram . . . . .	25
5.5 Class Diagram . . . . .	26
5.6 Deployment Diagram . . . . .	26
5.7 Component Diagram . . . . .	27
5.8 Object Diagram . . . . .	27
5.9 Level Zero DFD . . . . .	28
5.10 Level One DFD . . . . .	28
6.1 Icons and Functions . . . . .	32
6.2 Main Page . . . . .	33
6.3 Single Client View . . . . .	33
6.4 Message Modal . . . . .	34
6.5 Broadcast Message Modal . . . . .	34
6.6 Enter Keywords Modal . . . . .	34

# List of Tables

2.1 Comparison of network monitoring and administration projects . . . . .	7
3.1 Project Deliverables . . . . .	9
3.2 Roles and Responsibilities . . . . .	11
3.3 Tools and Techniques . . . . .	11
5.1 Requirements Traceability Matrix . . . . .	22
6.1 Score generation based on different Keywords for different screenshots .	32
7.1 Testing . . . . .	35

# Chapter 1

## Introduction

*This chapter presents introduction to the proposed system and discusses about the problem definition and gives a brief overview of how this project aims to solve the same.*

### 1.1 Problem Definition

Monitoring Student's PCs in a Lab is a tedious task and requires constant attention and hence engages the Lab Authority thus making their time unproductive. Further evaluation of the work done is even more time consuming. The Processes that are running on a PC can only be viewed by going to the individual's PC and opening the Task Manager. If the Internet is Down it is difficult to send a Broadcast Message or any message.

### 1.2 Motivation

- There is no efficient way to track the activity of the student throughout the session.
- This leaves no room for continuous evaluation, hence it is based only on final output.
- After every lab session there is a possibility that multiple machines are yet to be shut down. Shutting down every machine individually wastes a lot of time which can rather be used to serve better purposes.

## **1.3 Scope**

### **1.3.1 Functional Requirements:**

- Administrator server can remotely monitor the activities of client machines through the screenshots received
- Administrator can demand for a screenshot of the client machine at any time
- Administrator can remotely shutdown any connected client machine
- Administrator can send a message to any connected client machine
- Administrator can broadcast message to all the connected client machines
- Administrator will be able to analyze the relevance of the client's work with respect to a topic.

### **1.3.2 Non Functional Requirements:**

- Reliability
- Availability
- Maintainability
- Portability
- Performance(Depends on the number of clients connected)

## **1.4 Salient Contribution**

- Relevance

## **1.5 Organization of the Synopsis**

- Chapter 1 gives the overview of the project which includes the problem definition, motivation and scope of the project.

- Chapter 2 contains the literature survey of the project which tells us about the research done on the existing systems and the technology used to build the project.
- Chapter 3 briefs us about the software project management plan(SPMP) in which the tasks and the timetable for the completion of those tasks is given.
- Chapter 4 includes the specification required for the project as well as the functional and non-functional features of the project.
- Chapter 5 contains the software design description which deals with the design of the project.
- Chapter 6 includes the pseudo code and technologies used in the project
- Chapter 7 gives information about the whole testing process of the project
- Chapter 8 concludes the project and also gives information about the future scope of the project



# Chapter 2

## Literature Survey

*The purpose of this chapter is to cover background research and study of existing systems and references in order to improve the project scope.*

### 2.1 Introduction

The need for LAN monitoring arises when there is a need to keep the activities of the users on the network under surveillance. LAN monitoring basically is keeping a check on the activities of the individual machines on the network. There are different methods through which LAN Monitoring can be established, viz. Android Based LAN Monitoring, GPRS Based LAN Monitoring, LAN Monitoring using Remote Method Invocation (RMI), etc. For monitoring all the machines in the LAN, a remote access connection has to be established between the host and the machines. This connection will enable the admin to observe, manage and analyze the activities of the client.

### 2.2 Methods

The LAN monitoring is a compelling prevalent system. There exist many commercial products, technical papers exploring different scopes and information on blogs. Internet was researched for conducting the literature survey. Different online resources like IEEE Digital Library and Google search engine were significant for identifying the existing research and application.

## **2.3 Discussion**

From the literature survey we discovered various research papers and software applications that provide remote access to the computers connected in LAN. Each of these vary in many ways but the basic system of LAN monitoring is constant. Following are the most recent information collected:

### **2.3.1 Android Based LAN Monitoring**

The paper [android] represents a method to monitor local area network comprising of multiple computers from a remote place using an Android device with the help of Internet. An Android application is developed to keep an eye on targeted systems in a network. The processes being run on a computer can be monitored by the hand held device. The administrator can see the log files of targeted machine from any location that has internet facility.

### **2.3.2 GPRS Based LAN Monitoring and Controlling**

The paper [gprs] proposes a method for an administrator to monitor a Local Area Network from a distant location. A system is proposed to aim at the expansion of network services and facilities which are essential for monitoring a LAN by the use of GPRS. A network is created by bringing multiple workstations together. A GPRS enabled device will fulfill the purpose of monitoring the LAN whilst the administrator is not physically present. This method interacts with the admin only through sending and receiving emails via the central monitoring server.

### **2.3.3 LAN Monitoring using Remote Method Invocation(Java RMI)**

In the paper [rmi], a software which is able to monitor the whole network is being proposed. The whole software is developed in Java RMI [Remote Method Invocation]. The proposed software is being made to provide the live details of the users by taking the screen-shots of their desktops and sending them to main machine. The admin can view the image and can take action according to the image received. The admin can warn the

user by sending a message to the client machine. The admin can also remove the user remotely.

### **2.3.4 Comparison of different LAN Monitoring software**

Numerous software applications have been developed on the concept of LAN Monitoring. Each has its own features focusing on different aspects of monitoring. There are a lot of commercial software available for network monitoring, free and paid like LogicMonitor, Paessler PRTG Network Monitor, Spiceworks Network Monitor, Datadog, LabTech, ManageEngine OpManager, Ruckus Wireless ZoneDirector 1200, Idera Uptime Infrastructure Monitor, Vallum Halo Manager. Few of the software have features like mobile device support, real-time alerting, Application Programming Interface (API), custom dashboards, etc. These software focus on keeping the company's data centre healthy and functioning. Table 2.1 shows comparison of few features of some of the LAN monitoring software.

## **2.4 Conclusion**

The currently available LAN monitoring products suitable for company and home usage. But there is no software that has been manufactured which analyses the information on the screen of the connected monitors using image processing. This will be of great use in schools and colleges to control the activities of students on the computer during laboratories, tests and exams.

Table 2.1: Comparison of network monitoring and administration projects

Features	Network monitoring and administration projects					
	Master BLACK (proposed s/w)	ActivTrak	Amelia	SolarWinds - Network Performance Monitor	Wireshark	PRTG
Open source	-	-	✓	-	✓	-
Free	-			-		
UI based on	Web	Cloud	Windows Application	Web		Windows /Mac /Android application
Screenshot sharing	✓	✓	✓	-	-	-
View Client IP, port	✓	-	✓	-	-	-
View running applications	✓	-	-	-	✓	✓
Video Playback, Alarms, Website blocking	-	✓	-	-	-	-
Network performance	-	-	✓	✓	-	-
Traffic, packets, bandwidth	-	-	-	-	✓	✓
Cloud services, Databases	-	-	-	-	-	✓
Client work-relevancy score	✓	-	-	-	-	-
Method	Socket Programming, OCR	-	java.net API via TCP	SNMP	Packet Sniffing	SNMP version 1, 2c, 3
Use Case	Laboratory Session Monitoring	Insider Threat Detection, Organizational Efficiency, etc.	General Network Monitoring	Advanced Network Alerting	Network protocol analyzer	Infrastructure Management, Network Monitoring

# Chapter 3

## Software Project Management Plan

*This document is the fully developed Software Project Management Plan for LAN Monitoring. This document highlights the deliverable, roles, tasks and probabilistic schedule for the team and this project.*

### 3.1 Introduction

#### 3.1.1 Project Overview

The main purpose of the software is to monitor all the clients connected to LAN. First of all, the user has to download the software. After that there is a registration process to be completed and with that you can use the software. The log in credentials are used to log in in the software. The admin can view the screen shots of the machines which are connected to the LAN. The admin can also request the screen shot of any machine. The features to view and manage the processes and sending the messages to individual clients as well as broadcasting them have also been provided.

### 3.1.2 Project Deliverables

Project Deliverable	Description of the Project Deliverable	Delivery Date
Problem Definition	It will include the requirements and parameters of the project and methods to solve the problems.	30/08/2018
Software Requirement Specification(SRS)	Requirement identification of the project as requested by the client	14/09/2018
Software Project Management Plan(SPMP)	The objectives, dependencies, constraints, deliverables, and timetable of the project will be specified	21/09/2018
Software Design Document (SDD)	GUI and UML diagrams are designed	26/09/2018
Software Test Document (STD)	Details the types of tests to be carried out on the system to ensure the system meets requirements and maintains integrity	05/10/2018
Front-End (GUI)	The user interface of the software	31/12/2018
Connectivity	Establishing connection between the systems and achieving other functionalities	31/12/2018
Processing	Use image processing to calculate relevance of client's work	31/12/2018
Integration	Use image processing to calculate relevance of client's work	31/01/2018
Testing	All the modules of software will be tested	29/01/2019
Deployment of final product	Submission of final software with documents	30/04/2019

Table 3.1: Project Deliverables

## 3.2 Project Organization

### 3.2.1 Software Process Model

The software process model used for this project is waterfall model. Waterfall process model is selected for this project because of the following reasons:

- The requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- Ample resources with required expertise are available freely
- The project is short.

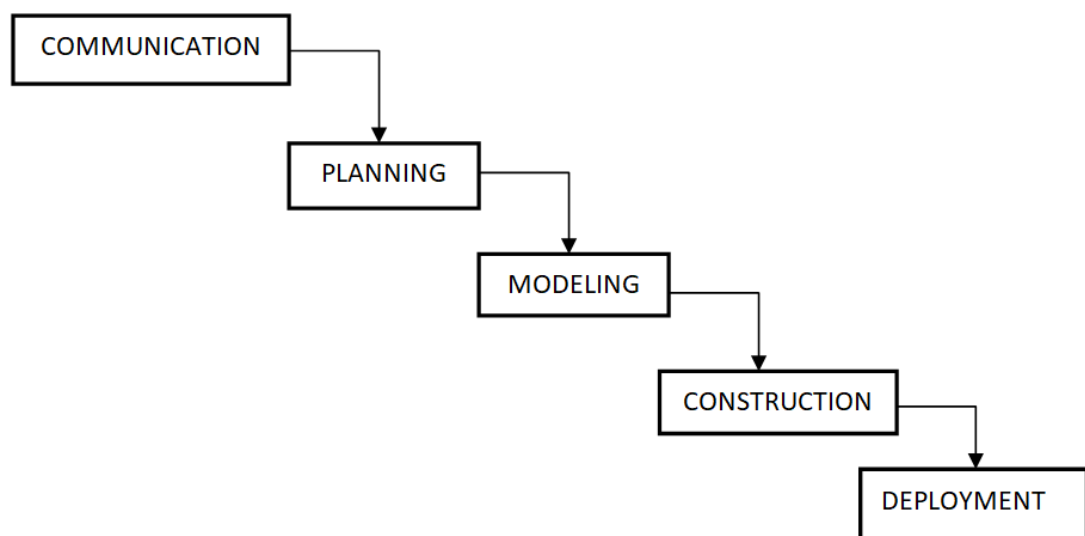


Figure 3.1: Waterfall Life Cycle

In this model, each module passes through the following process

- **Communication:** Project initiation and requirement gathering
- **Planning:** Estimation, scheduling and documentation
- **Modelling:** Analysis of project and designing

- **Construction:** Writing Code and testing
- **Deployment:** Deliver the project and maintain

### 3.2.2 Roles and Responsibilities

Table 3.2: Roles and Responsibilities

Sr No	Name	Roles	Responsibilities
1	Rushabh Bid	Strategist	Formulates, analyses and creates goals for back-end coding
2	Yash Jain	Tester	Tests the codes, documentation
3	Yash Kandalam	Developer	Builds and debugs the codes
4	Pratyusha Reddy	Analyst	Supervises the work-flow, front-end coding

### 3.2.3 Tools And Techniques

Table 3.3: Tools and Techniques

Deliverable	Tools	Techniques
Requirement	Latex	Interview
Design	Lucidchart	UML 2.0
Project Plan	Latex	Discussion and Research
Front-end	PYCHARM/ANACONDA	Coding PYTHON
Back-end	PYCHARM/ANACONDA Text	Coding PYTHON
Connectivity	Flask	Coding PYTHON
Processing	PYCHARM/ANACONDA	Coding PYTHON



## **3.3 Project Management plan**

### **3.3.1 Tasks**

#### **3.3.1.1 Task Description**

T1.1 - Requirement Gathering

T1.2 - Prepare SRS

T1.3 - GUI

T1.4 - UML

T2.1 - Design software GUI

T2.2 - Connectivity

T2.3 - Processing

T3.1 - Integrate

T4.1 - Testing

T4.2 - Deployment

#### **3.3.1.2 Deliverable and Milestones**

T1.1 - Project plan

T1.2 - Understand requirements of project, Overleaf

T1.3 - Architecture diagram featuring specifications

T1.4 - Architecture diagram featuring specifications, Delivery of final SDD

T2.1 - Deliver demo

T2.2 - Connect systems, live monitoring, message broadcast

T2.3 - Display work relevance of client

T3.1 - Integration of the connectivity and processing modules into the GUI

T4.1 - Delivery of the code

T4.2 - Website goes live

#### **3.3.1.3 Resources Needed**

- T1.1 - Time for meetings
- T1.2 - Delivery of SRS
- T1.3 - Delivery of GUI
- T1.4 - Overleaf, Rational Rose
- T2.1 - PyCharm
- T2.2 - PyCharm, Anaconda
- T2.3 - PyCharm, Anaconda
- T3.1 - PyCharm, Anaconda
- T4.1 - LAN connected computers
- T4.2 - Delivery of executable website URL

#### **3.3.1.4 Dependencies and Constraints**

- T1.1 - Frequent Communication
- T1.2 - The project must be approved by the assigned panel
- T1.3 - Development member has to be included in brainstorming sessions
- T1.4 - Development member has to be included in brainstorming sessions
- T2.1 - SRS, SDD must be ready
- T2.2 - SRS, SDD must be ready, Compatibility with OS
- T2.3 - SRS, SDD must be ready. Compatibility with OS
- T3.1 - SRS, SDD must be ready
- T4.1 - The project should be completed
- T4.2 - LAN connected computers

#### **3.3.1.5 Risk And Contingencies**

- T1.1 - Panel fails to convey requirements
- T1.2 - Panel changes requirements
- T1.3 - Panel changes GUI. Wrong GUI designed

- T1.4 - Human-error or software crashes
- T2.1 - Coding of demo in-feasible. Change demo
- T2.2 - Poor coding. Unsupported system
- T2.3 - Poor coding. Unsupported system
- T3.1 - Compatibility with modules. Unsupported code
- T4.1 - Incomplete coding. Errors in final product. Find and eradicate the problem
- T4.2 - Unfinished product. Extend the deadline

### 3.3.2 Timetable

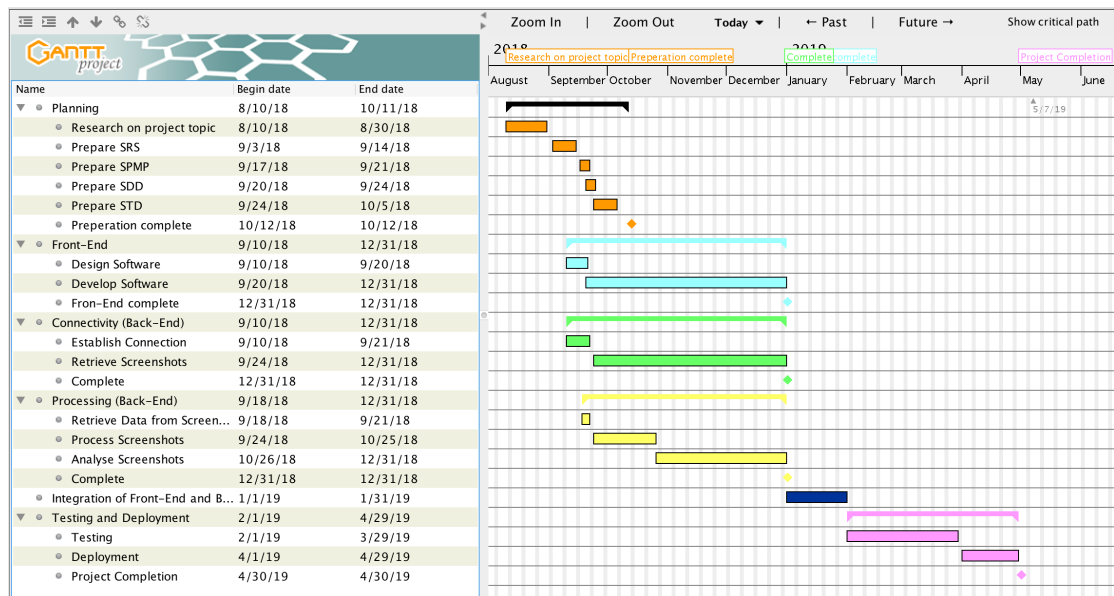


Figure 3.2: Gantt Chart

# Chapter 4

## Software Requirement Specification

*Software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software underdevelopment. The SRS fully describes what the software will do and how it will be expected to perform.*

### 4.1 Introduction

#### 4.1.1 Product Overview

The main purpose of the software is to monitor all the clients connected to LAN. First of all, the user has to download the software. After that there is a registration process to be completed and with that you can use the software. The log in credentials are used to log in in the software. The admin can view the screen shots of the machines which are connected to the LAN. The admin can also request the screen shot of any machine. The features to view and manage the processes and sending the messages to individual clients as well as broadcasting them have also been provided.

### 4.2 Specification Requirement

This section includes all the specific requirements of the systems. The detailed description of the software is given.

## **4.2.1 External Interface Requirements**

This section includes the detailed specifications of all inputs and outputs by the user. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

### **4.2.1.1 User Interfaces**

This section gives a brief idea about the user interface. When the application is opened, the homepage is loaded. The options on the homepage are "*CONNECTED CLIENTS*", "*VIEW DESKTOPS*", "*BROADCAST MESSAGE*". When the admin clicks on "*CONNECTED CLIENTS*", he/she views the list of connected clients to the LAN. When the admin clicks on "*VIEW DESKTOPS*", he/she sees a page containing a grid of images which are the screen shots of the client machines connected to the LAN. The admin has multiple options to choose from on this screen. They are "*REQUEST SCREEN SHOT*", "*MESSAGE*", "*SHUTDOWN*" and "*RELEVANCE*". The options request the screen shot of the particular client desktop, message a particular client, shutdown a client machine and calculate the relevance of the client's work with respect to the task assigned respectively.

### **4.2.1.2 Hardware Interfaces**

This section gives a brief idea about the external hardware used in the software. Following are the hardware interfaces used:

1. PCs/workstations
2. Network Interface Card(NIC)
3. Cabling and connectors
4. Hub, concentrator, Bridge, LAN Switch and Router
5. 500MB free memory
6. 2GB RAM (minimum)

#### **4.2.1.3 Software Interfaces**

This section gives an idea about the software interfaces used in the software. Following are the software interfaces used:

1. Windows 10
2. PyCharm
3. Python
4. Tesseract

#### **4.2.1.4 Communication Protocols**

FTP - File Transfer Protocol will be used for implementation of the software. This section gives an idea about the communication protocols followed in the software. They are as follows:

1. FTP - file transfer protocol
2. IPv4
3. IPv6

### **4.2.2 Software Product Features**

#### **4.2.2.1 Monitoring Through Screen Shots**

Administrator server can remotely monitor the activities of client machines through the live screen shots received.

#### **4.2.2.2 Request Live Screen Shot**

Administrator can demand for a screen shot of the client machine at any time.

#### **4.2.2.3 Shutdown Client**

Administrator can remotely shutdown any selected client machine.

#### **4.2.2.4 Shutdown All**

Administrator can remotely shutdown all connected client machine.

#### **4.2.2.5 Message Individual Client**

Administrator can send a message to any connected client machine.

#### **4.2.2.6 Broadcast Message**

Administrator can broadcast message to all the connected client machines.

#### **4.2.2.7 Enter Keywords**

Administrator has to enter keywords according to which the relevance score is generated for each client machine. The keywords can be changed at any given time.

#### **4.2.2.8 Calculate Relevance**

Administrator will be able to analyze the relevance of the client's work. The relevance will be displayed in the form of color coded border around each screenshot of the machine. It will be calculated according to the keywords given by Administrator and the screenshots retrieved from client machine. Green means relevant work is being done. Yellow means fair amount of relevant work is being done. Red means client is not working on the given keywords.

#### **4.2.2.9 Disconnect Client**

If there is a need to stop the live feed of a client machine, Administrator can do so by selecting the disconnect button.

#### **4.2.2.10 Reconnect Client**

If there is a need to restart the live feed of a client machine, Administrator can do so by selecting the connect button.

#### **4.2.2.11 Storage of Screenshots in Folder**

The images captures in a session are stored in folders according to client machines. The images will be saved with the relevance score color.

### **4.2.3 Software System Attributes**

#### **4.2.3.1 Reliability**

The system should maintain consistency. The accuracy of "relevance of users' work" should be maintained.

#### **4.2.3.2 Availability**

System will be available for 7 days and 24 hours. In the application side, system will be tested against probable failures before publishing the first version or updated versions of application. Published version will be error free. In database side, in case of a failure, system will recover any information of user and system.

#### **4.2.3.3 Security**

The concerns with security are not much as the software does not store important credentials of the users.

#### **4.2.3.4 Maintainability**

All code shall be fully documented. Each function shall be commented with pre-conditions and post-conditions. All program files shall include comments concerning authorship and date of last change. The code shall be modular to permit future modifications.

#### **4.2.3.5 Portability**

The software is fully portable as it does not have any physical object connected to it. It can be used on any computer with required OS.



#### **4.2.3.6 Performance**

The performance of the software depends on the connection of client machines to the administrator computer.

#### **4.2.4 Software Quality Attributes**

Adaptability, availability, correctness, consistency, flexibility, portability, reliability, re-usability, robustness, secure, test-ability, usability, scalability, user-friendliness.

# Chapter 5

## Software Design Description

*This chapter gives a description of the design approach, highlighting the essential features that allow the design to meet the stated requirements.*

### 5.1 Introduction

#### 5.1.1 Design overview

The project is designed with an interface that will allow the admin to monitor all the users connected to the network. The user interface is developed using web technology. The aim is to achieve clean user friendly software. Administrator's user interface will consist of a web page which will stream live images of all the connected client machines. This will enable the Administrator to observe all the machines remotely at once and to analyse their work with the help of efficiency score. The web page has a list of twenty images for each connected client machine. Each image will have a colored border which represents the relevance of the work being done by the user. The architectural design used for this project is discussed in this document.

### 5.1.2 Requirements Traceability Matrix

Requirements	Components					
	User Interface	Server	File System	Client	Image Processing	File Sending
Live View	X	X	X	X		X
Message	X	X		X		
Manage Processes	X	X		X		
Check Relevance	X	X	X	X	X	X
Shutdown Machines	X	X		X		

Table 5.1: Requirements Traceability Matrix

## 5.2 System Architectural Design

### 5.2.1 Chosen System Architecture

This project uses a client-server architecture model. The software will be set up on the server and many users can communicate over a computer network on separate hardware.

Its basic principles are:

- There is at least one component that has the role of server, waiting for and then handling connections.
- There is at least one component that has the role of client, initiating connections in order to obtain some service.

An important variant of the client-server architecture is the three-tier model under which a server communicates with both a client (usually through the Internet) and a database server (usually within an intranet, for security reasons). The server acts as a client when accessing the database server.

Following are the advantages of client server architecture :

Centralization : Unlike P2P, where there is no central administration, here in this architecture there is a centralized control. Servers help in administering the whole set-up. Access rights and resource allocation is done by Servers

Proper Management : All the files are stored at the same place. In this way, management of files becomes easy. Also it becomes easier to find files

Back-up and Recovery possible : As all the data is stored on server its easy to make a back-up of it. Also, in case of some break-down if data is lost, it can be recovered easily and efficiently. While in peer computing we have to take back-up at every work-station.

Technical Risks	Contingency Plans
Server down to due heavy load of large data	Allow download of limited data at a time, then remaining shall be downloaded in next part of the request When large number of users are using server, restrict the number to 20 at a time
Change in functionality leading to too many alterations in the other part of code	In case of such occurrence, use alternative architecture: MVC, for logical separation of Model, View and Component.

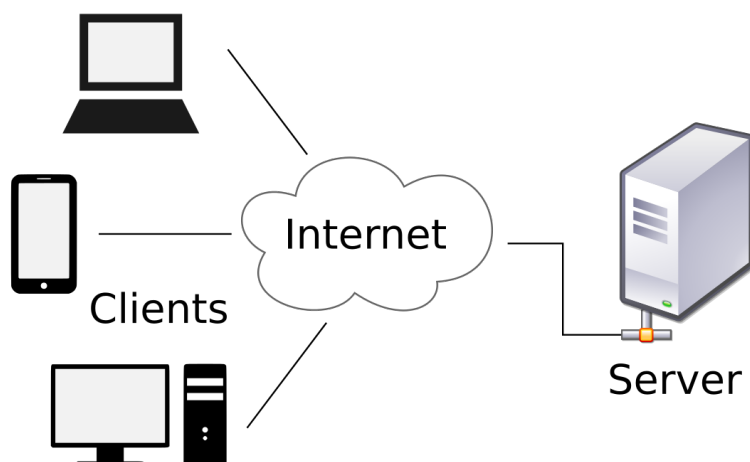


Figure 5.1: Client-Server Architecture

## 5.2.2 Discussion of alternate Design

The alternate is Component-based architecture, it focuses on the decomposition of the design into individual functional or logical components that represent well-defined communication interfaces containing methods, events, and properties. It provides a higher level of abstraction and divides the problem into sub-problems, each associated with component partitions. The primary objective of component-based architecture is to ensure component re-usability. This architecture is not used because client-server model is more suitable.

## 5.2.3 System Interface Description

### 5.2.3.1 Use case diagram

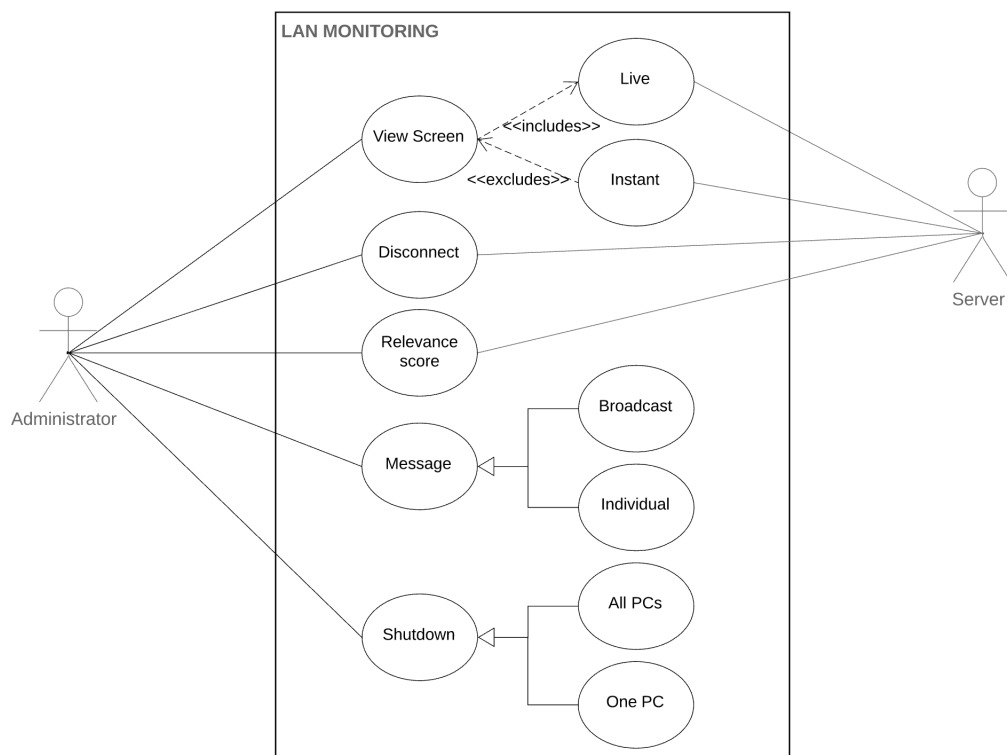


Figure 5.2: Use Case Diagram

### 5.2.3.2 Sequence Diagram

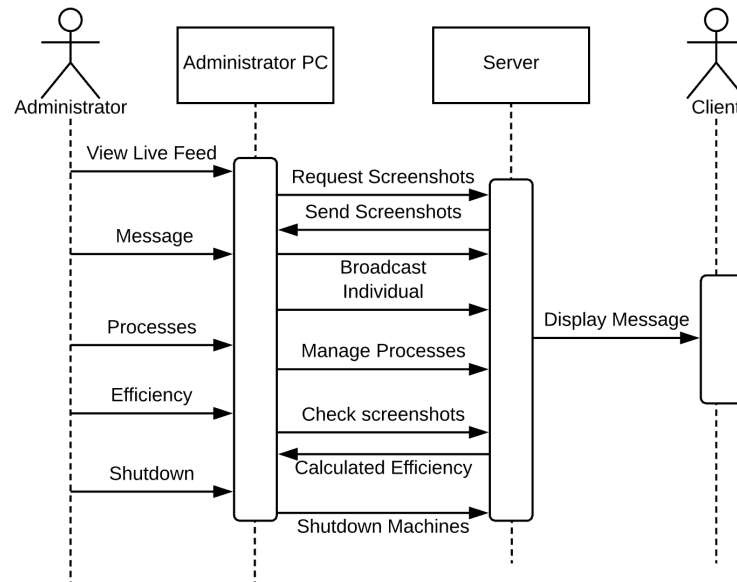


Figure 5.3: Sequence Diagram

### 5.2.3.3 State chart diagram

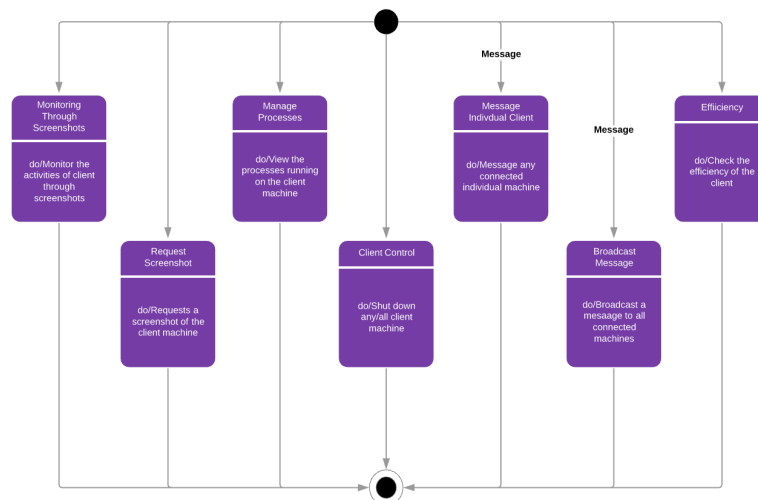


Figure 5.4: State Chart Diagram

#### 5.2.3.4 Class Diagram

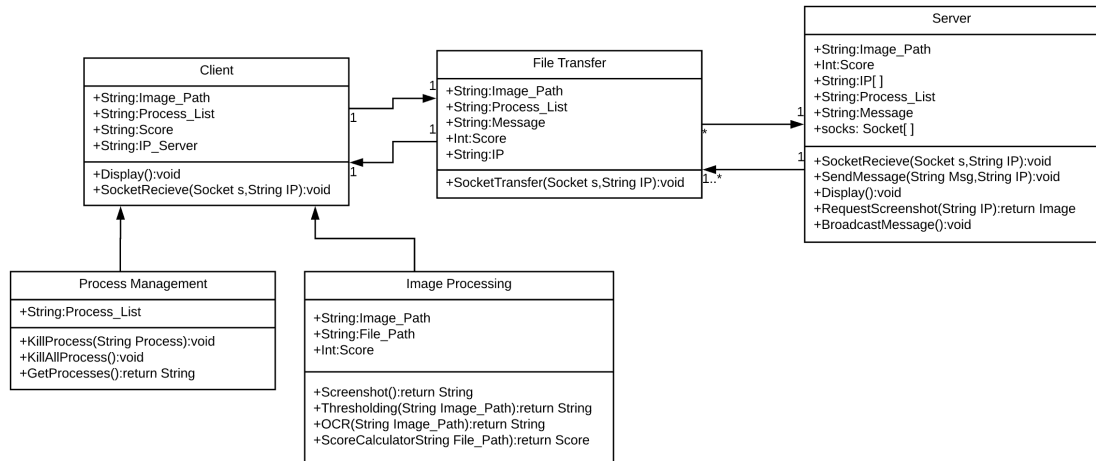


Figure 5.5: Class Diagram

### 5.2.3.5 Deployment diagram

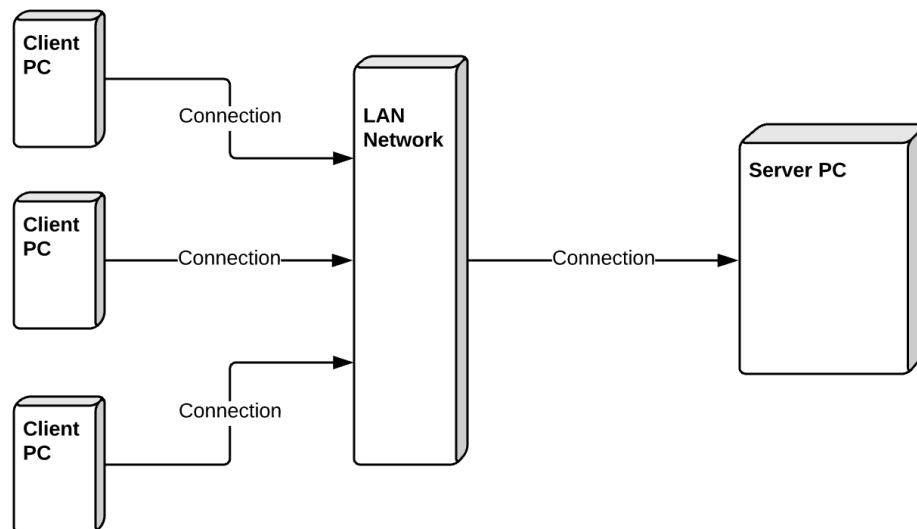


Figure 5.6: Deployment Diagram

### 5.2.3.6 Software Component Diagram

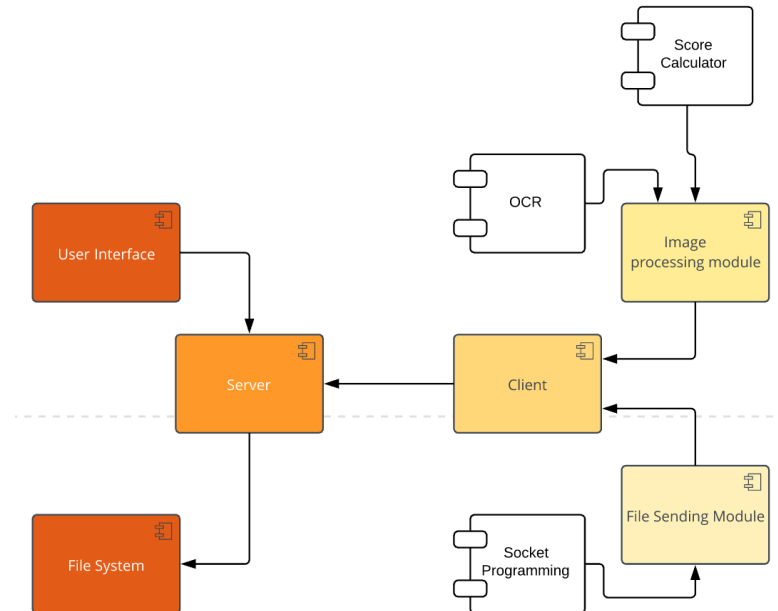


Figure 5.7: Component Diagram

### 5.2.3.7 Object diagram

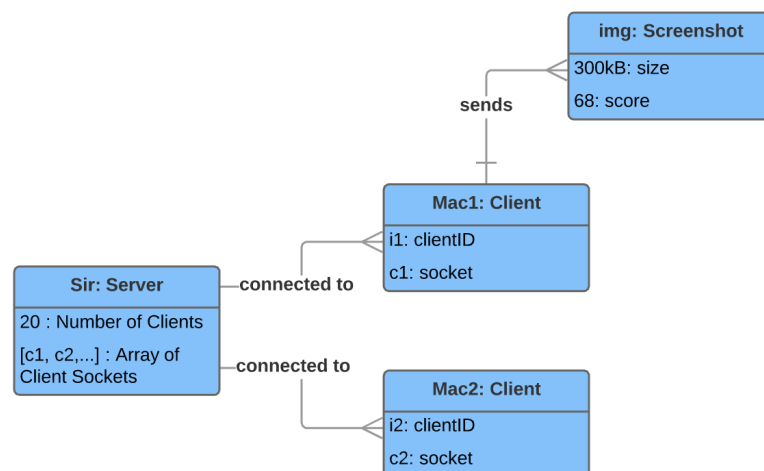


Figure 5.8: Object Diagram



### 5.2.3.8 Level Zero DFD

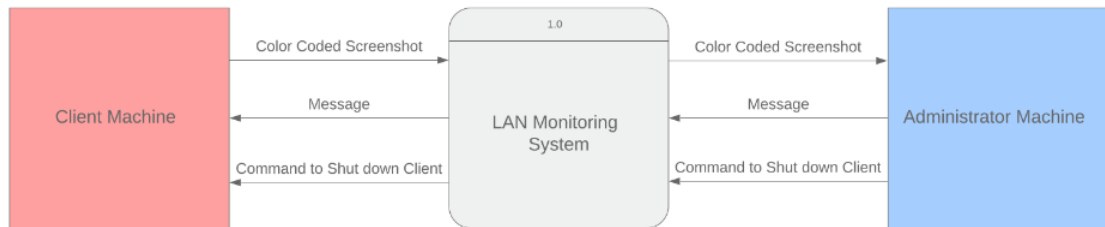


Figure 5.9: Level Zero DFD

### 5.2.3.9 Level One DFD

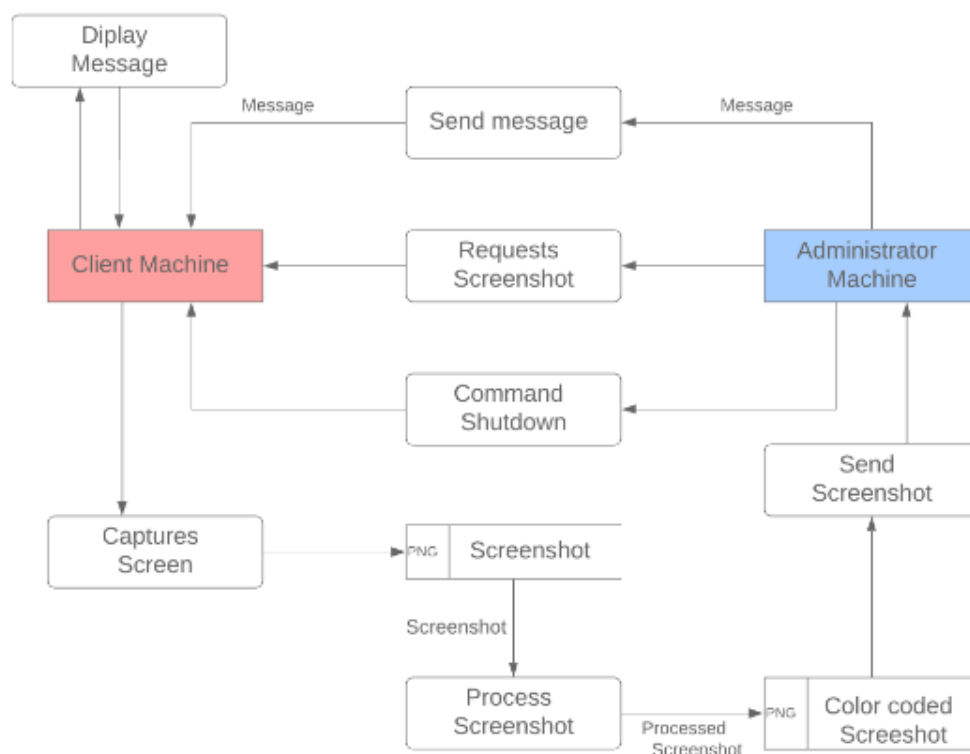


Figure 5.10: Level One DFD

# **Chapter 6**

## **Implementation**

### **6.1 Technologies Used**

#### **6.1.1 Socket Programming**

Socket Programming is a method of establishing a connection between two nodes (machines) on a network and establishing communication between them. One node listens on a particular port at an IP (server), while the other reaches out to the former to create the connection (client). As the name suggests, sockets are used at both the nodes, i.e. Server and Client. The server first binds a socket to a specific IP and port to receive requests for connection. Then it goes into listening mode and accepts/closes requests from clients. All the communication between the server and clients is backed by sockets. This communication includes requesting and sending screenshots, sending commands to stop applications and shut down machines, and sending messages.

#### **6.1.2 Tesseract - OCR**

Tesseract is an open source OCR engine which was released under Apache license and sponsored by Google. It is a powerful tool which can be easily used with a variety of programming languages and also has strong community support. Optical Character Recognition(OCR) is the technology that is used for recognition of text in an image. The following preprocessing is proposed to get the optimum efficiency from the Tesseract

OCR. The computer first converts the screenshot image into a grayscale image. The image undergoes appropriate pre-processing to remove any noise present in the image. After the image is pre-processed the text in the image gets photo scanned character by character. This data is then analyzed and translated to ASCII character code. The text data that is produced has a lot of junk symbols which is cleaned using regular expressions. The final text data contains the text recognized by the Tesseract OCR which is free of junk values and is stored in a list format.

### **6.1.3 Flask - Web Framework**

The proposed system is developed as a web application. This results in a multiplatform, highly flexible, readily deployable and intuitive application. The web application is built to work without the internet which decreases the number of requirements to install and use the software. On running the application, a web browser opens up, with a predefined address, comprising the UI. An HTTP request will be sent and the server is started. Flask is used to manage this request of the administrator. Flask is a BSD licensed microframework for Python. It is based on Werkzeug and Jinja2. It figures out what the requests are, what is being asked and what response is needed to send back to the administrator. Flask framework was selected because of its simple process of web application development. It allows focus on the administrator request and response in a seamless manner.

### **6.1.4 Python**

The entire software is built on Python. Python is a high-level, general-purpose programming language. The development process was easy and efficient due to Python's large and robust library, code readability, programming paradigms, and language features.

## 6.2 Algorithm

### 6.2.1 Score Generation

The process of monitoring each screenshot manually is a bit tedious in a large LAN. A scoring system is used to score each client machine thus removing the need to check all the screenshots manually. These scores are calculated based on the work performed on the client machines in comparison with the work assigned to them by the administrator. In order to calculate the relevance of the data, the administrator inputs keywords related to the assigned task. These keywords are then matched with the text data generated after applying OCR to the client machine's screenshot. The score is generated by counting the number of keywords that occur in the text data generated from the screenshot. Higher the occurrence, the higher the score. The screenshots of the client machines when some software applications are running in the foreground, do not produce meaningful text data which can be matched with the keywords. The list of running applications is checked for such applications and changes in the score are made accordingly. In order to give the administrator complete information regarding the client's work, the scores from multiple screenshots of the client machine over a period of time are calculated and stored. Weights are assigned to the scores based on the time it arrives at. The weight is higher for scores generated from the latest screenshots.

The formula of score calculation is given by: where,

$$\text{Net Score} = \frac{\sum_{i=0}^n W_i * S_i}{n}$$

W = weight assigned to the score

S = Score generated from screenshots

n = Number of screenshots sent till calculating the net score

The clients are classified using a color-coded interface which exploits the ease of understanding of colors by a human brain thus making a clean and convenient UI.

Table 6.1 shows the scores generated for each keyword set against a particular screenshot. The screenshots used are of the following: Idle Desktop, Google search

Table 6.1: Score generation based on different Keywords for different screenshots

List of Keywords	Screenshots				
	Desktop	Google	YouTube	Wikipedia	Google homepage
Google, Wikipedia, YouTube, machine, learning	0	40	36	37	10
neural, network, regression, machine, learning	0	44	41	43	0
neural, network, regression, epoch, tensorflow	0	0	0	2	0
neural, network, regression, epoch, tensorflow, machine, learning, unsupervised, supervised	0	40	36	45	0
machine, learning	0	33	35	37	0

for machine learning, YouTube search for machine learning, Wikipedia search for machine learning, Google search for Wikipedia Thus from the following observations we can conclude that the relevancy score is most reliable for a rich set of keywords that are closely related to the task assigned. Moreover, it is observed that generic words like Google, YouTube, and Wikipedia might lead to unreliable higher scores for irrelevant images. Using larger sets of keywords won't necessarily increase the score as all of the text might not be captured through the screenshots

## 6.3 Implementation

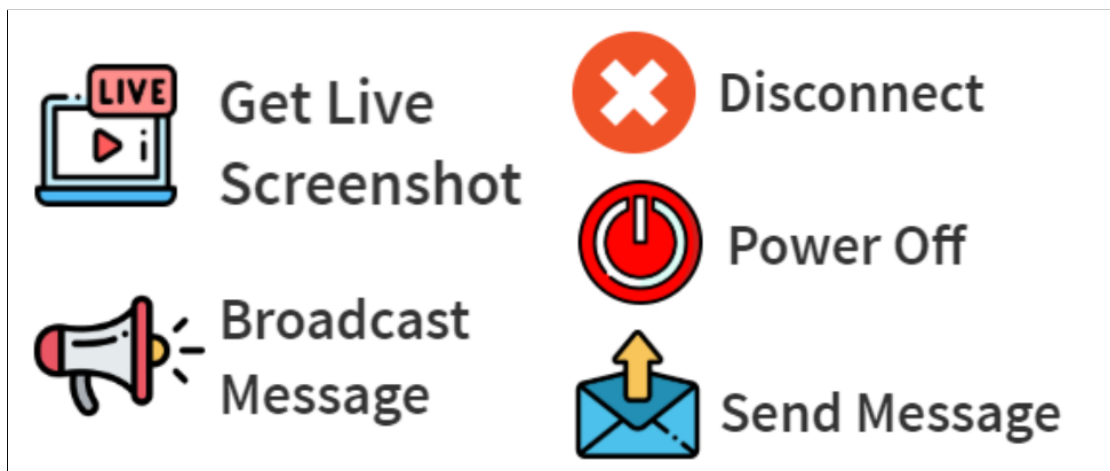


Figure 6.1: Icons and Functions

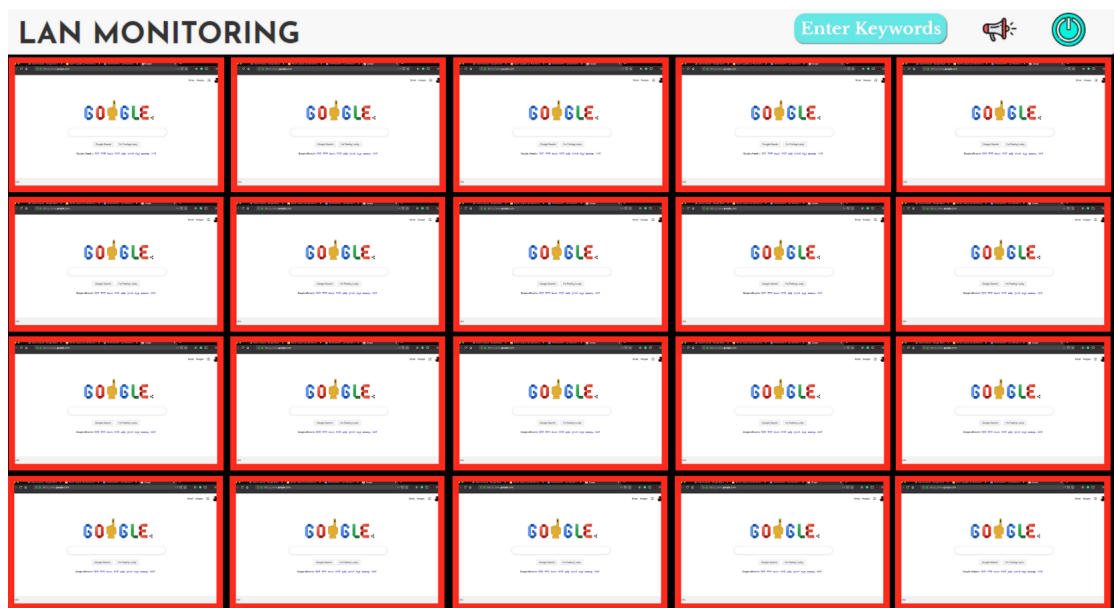


Figure 6.2: Main Page

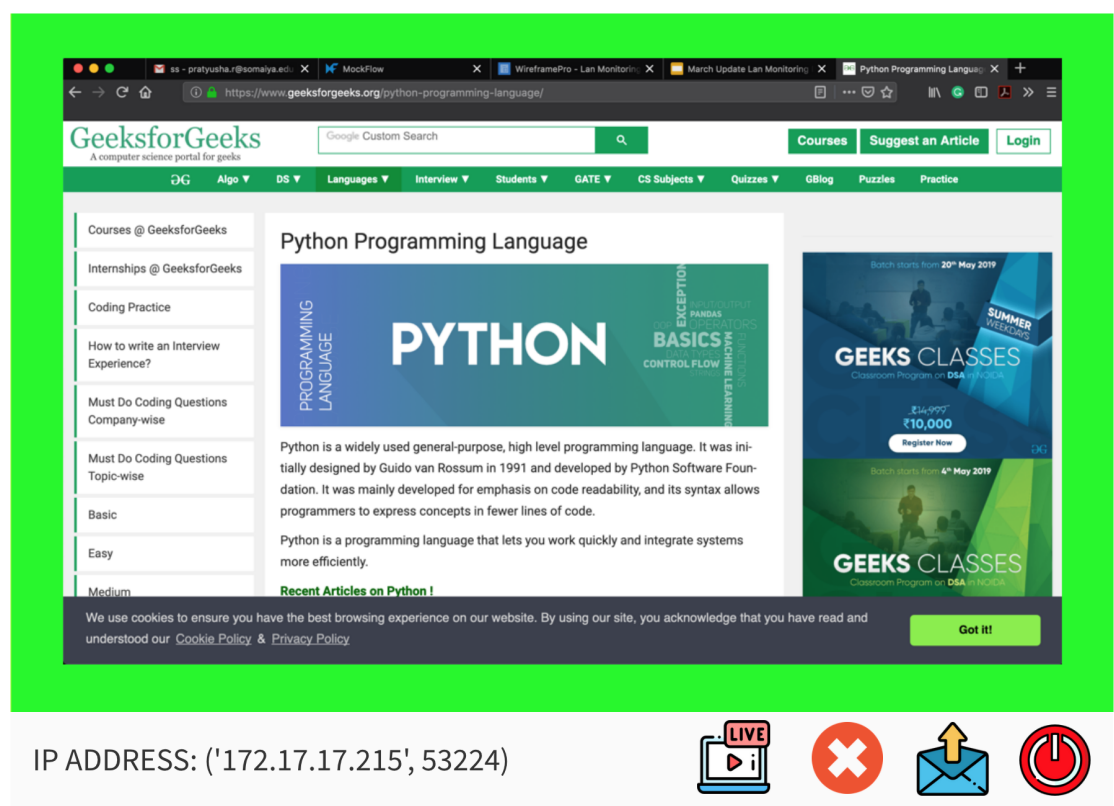
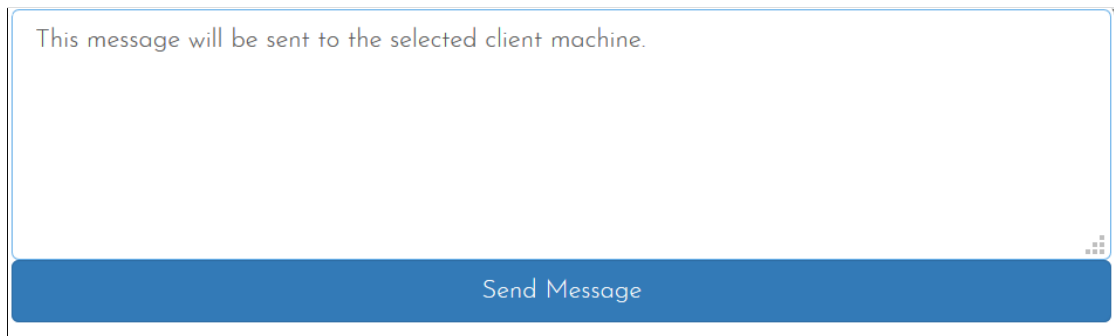


Figure 6.3: Single Client View

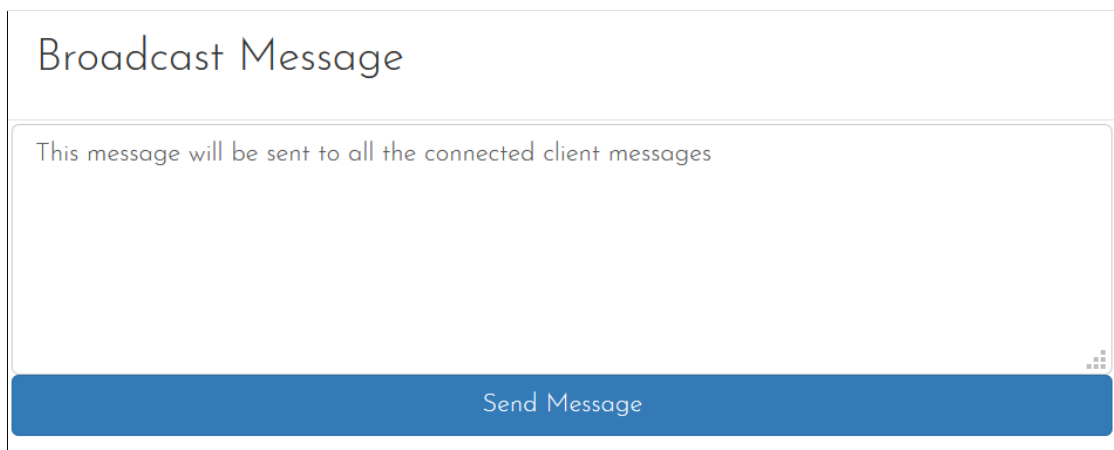


This message will be sent to the selected client machine.

Send Message

The modal is a light gray rectangle with a thin border. It contains a text area with the message "This message will be sent to the selected client machine." and a blue button labeled "Send Message" at the bottom right.

Figure 6.4: Message Modal



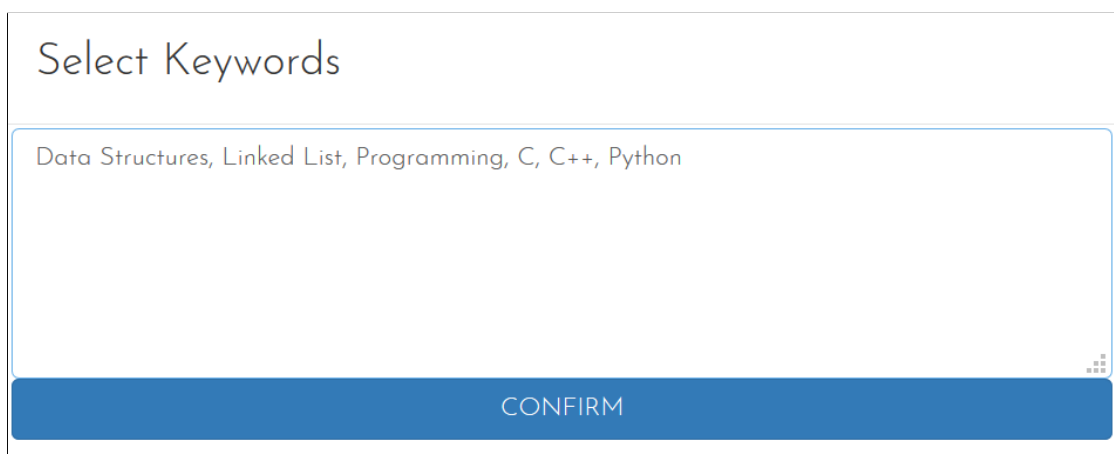
Broadcast Message

This message will be sent to all the connected client messages

Send Message

The modal is a light gray rectangle with a thin border. It has a title "Broadcast Message" at the top. Below it is a text area with the message "This message will be sent to all the connected client messages". At the bottom right is a blue button labeled "Send Message".

Figure 6.5: Broadcast Message Modal



Select Keywords

Data Structures, Linked List, Programming, C, C++, Python

CONFIRM

The modal is a light gray rectangle with a thin border. It has a title "Select Keywords" at the top. Below it is a text area containing the keywords "Data Structures, Linked List, Programming, C, C++, Python". At the bottom right is a blue button labeled "CONFIRM".

Figure 6.6: Enter Keywords Modal

# Chapter 7

## Software Test Document

*This chapter discusses about all the test scenarios such as testing approaches and type of testing used. It also shows the functionalities which are to be tested.*

### 7.1 Introduction

#### 7.1.1 System Overview

Black master will assist the administrator to monitor all the client PCs which share the same LAN. Client PCs will send regular Screenshots to the Admin. These screenshots will also be processed with OCR and generate a color coded efficiency score. Admin can also request for screenshots. Unit testing will be carried out throughout the development process and System Testing will be carried out on the final product.

#### 7.1.2 Test Approach

White-box testing will be used in Bottom-up approach.

Unit Testing	All the individual components will be tested
Integration Testing	Testing will be carried out on how all the Front-end and Back-end integrate.
Acceptance Testing	This testing will be carried out to check if software matches the clients demands.

Table 7.1: Testing



## **7.2 Introduction**

### **7.2.1 Features to be tested**

1. Display of Screenshots
2. Requesting of Screenshots by admin and its processing.
3. Shutting down of client machines.
4. Messaging feature.
5. Relevance of Image Processing.

### **7.2.2 Features not to be tested**

The software feature which will not be tested include:

1. LAN Network of the connected clients.

### **7.2.3 Testing Tools and Environment**

#### **7.2.3.1 Testing Tools**

No special testing tools are required as the testing will be carried out manually.

#### **7.2.3.2 Testing Environment**

Hardware

- LAN Network
- Computer
- Internet
- No. of testers: 2

## **7.3 Test Cases**

### **7.3.1 Test case ID: TC-0001**

#### **7.3.1.1 Purpose**

Monitoring through screenshots.

#### **7.3.1.2 Input**

Screenshot of client machine.

#### **7.3.1.3 Expected outputs and Pass/Fail criteria**

- Screenshot received by admin without errors - Pass
- Screenshot not received by admin - Fail
- Screenshot received does not match the sent image - Fail

#### **7.3.1.4 Test procedure**

Send image files through sockets and test if the images are sent and received without errors.

### **7.3.2 Test case ID: TC-0002**

#### **7.3.2.1 Purpose**

Request Screenshot.

#### **7.3.2.2 Input**

Request by admin.

#### **7.3.2.3 Expected outputs and Pass/Fail criteria**

- Screenshot received by admin - Pass

- Screenshot not received by admin - Fail
- Screenshot received does not match the sent image - Fail

#### **7.3.2.4 Test procedure**

Send request from admin machine and test if the current screenshot is received by the admin without errors.

### **7.3.3 Test case ID: TC-0003**

#### **7.3.3.1 Purpose**

Message Individual Client.

#### **7.3.3.2 Input**

Demo message from admin.

#### **7.3.3.3 Expected outputs and Pass/Fail criteria**

- The message is only received by a particular client - Pass
- The message is not received by that particular client - Fail
- The message is received by all the clients - Fail

#### **7.3.3.4 Test procedure**

Send a demo message from the admin machine to a client machine and test if the message is received by the client and not received by all the clients.

### **7.3.4 Test case ID: TC-0004**

#### **7.3.4.1 Purpose**

Broadcast Message.

#### **7.3.4.2 Input**

Demo message from Administrator.

#### **7.3.4.3 Expected outputs and Pass/Fail criteria**

- The message is received by all the clients - Pass
- The message is not received by at least one client - Fail

#### **7.3.4.4 Test procedure**

Broadcast a demo message from the admin machine to all the client machines and test if the message is received by all the clients.

### **7.3.5 Test case ID: TC-0005**

#### **7.3.5.1 Purpose**

Shutdown individual client machine.

#### **7.3.5.2 Input**

Administrator request.

#### **7.3.5.3 Expected outputs and Pass/Fail criteria**

- Correct machine turned off: Pass
- Incorrect machine turned off: Fail

#### **7.3.5.4 Test procedure**

Shutdown from Admin machine and check the client machines.

### **7.3.6 Test case ID: TC-0006**

#### **7.3.6.1 Purpose**

Shutdown all client machines.

#### **7.3.6.2 Input**

Administrator request.

#### **7.3.6.3 Expected outputs and Pass/Fail criteria**

- All machines turned off: Pass
- All machines not turned off: Fail

#### **7.3.6.4 Test procedure**

Shutdown from Admin machine and check the client machines.

### **7.3.7 Test case ID: TC-0007**

#### **7.3.7.1 Purpose**

Test Relevance.

#### **7.3.7.2 Input**

Screenshot of client machine.

#### **7.3.7.3 Expected outputs and Pass/Fail criteria**

- The client screenshot is marked green if the allotted work is being done - Pass
- The client screenshot is marked red if the allotted work is not being done - Pass
- The client screenshot is marked green if the allotted work is being done - Pass
- The client screenshot is marked green if the allotted work is not being done - Fail

#### **7.3.7.4 Test procedure**

Analyse screenshots displayed on Admin GUI

### **7.3.8 Test case ID: TC-0008**

#### **7.3.8.1 Purpose**

Check screenshots storage.

#### **7.3.8.2 Input**

Screenshots of client machine.

#### **7.3.8.3 Expected outputs and Pass/Fail criteria**

- The client screenshots are stored in their respective folders - Pass
- The client screenshots are not stored in their respective folders - Fail

#### **7.3.8.4 Test procedure**

Check if the screenshots are being saved in the correct folder in Administrator machine.

# Chapter 8

## Conclusion and Future work

*This chapter provides the conclusion driven from all the work done on the project. It also includes the possible future scope for the project.*

### 8.1 Conclusion

Our project 'LAN Monitoring' will provide any administrator to monitor the clients on through LAN network. In the beginning of the year, we formulated a timetable for the project. The basic requirements documentation was completed in the first month and was later updated as the project proceeded. We compared the existing software and consulted with various faculty to understand the needs of we formulated a timetable for the project. The basic requirements documentation was completed in the first month and was later updated as the project proceeded. We compared the existing software and consulted with various faculty to understand the needs of administrator. After gathering necessary data we started the designing phase. We assigned tasks to each member and divided the work equally. Two of as the project proceeded. We compared the existing software and consulted with various faculty to understand the needs of administrator. After gathering necessary data we started the designing phase. We assigned tasks to each member and divided the work equally. Two of our members designed the User Interface for the system. We used HTML, Bootstrap and CSS for styling. Simultaneously, the other members started with the connectivity between the client and server. This was coded using socket programming in python. The initial code was used to

send messages from server to client. After adding the necessary logic, the client can now send screenshots at regular interval to the server. We have also integrated Image Processing in our project. We can extract the data from the screenshots using Tesseract OCR library. This will help in calculating the score of client machine's screenshot. This is the total progress that our team has achieved during this year.

## **8.2 Scope for Future Work**

The software can be hosted on a mobile platform to introduce portability. Basically, an application can be developed to keep an eye on devices over a network from any remote GPRS enabled device. An additional feature that can be added is installing software concurrently on all connected machines. In order to further improve the efficiency of the scoring system, documents related to the work to be performed can be used to extract a rich list of keywords.



# References

- Karishma Gidge, Kalyani Patil, Priyanka Wadnere, “Android Based LAN Monitoring”, Journal for Research in Applied Science Engineering Technology (IJRASET) 3 (January 2015).
- Meghana Sapkal, Shekhar Patil, Leesensa Vispute, Santosh Jagtap, “GPRS Based LAN Monitoring and Controlling”, IOSR Journal of Computer Engineering 16 (May-Jun 2014), pp. 09–15.
- Harsh Mittal, Manoj Jain, Latha Banda, “Monitoring Local Area Network Using Remote Method Invocation”, International Journal of Computer Science and Mobile Computing (IJCSMC) 2 (May 2013), pp. 50–55.
- Paul Ferrill. The Best Network Monitoring Software of 2018.  
URL: <https://in.pcmag.com/cloud-services/97759/the-best-network-monitoring-software>
- ActivTrak, <https://support.activtrak.com/hc/en-us>
- Amelia, <https://github.com/Obsidiam/amelia>
- SolarWinds - Network Performance Monitor, <https://support.solarwinds.com/@api/deki/files/10232/NPMAAdministratorGuide.pdf?revision=22>
- Wireshark, <https://www.wireshark.org/docs/wsdghtml/>
- PRTG, <https://www.paessler.com/manuals/prtg>

## **Authors' Publication**

# **OCR Enabled Monitoring and Administration of Computers Over LAN**

**Rushabh Bid**  
Department of Information  
Technology  
K. J. Somaiya College of  
Engineering  
Mumbai, India

**Yash Kandalam**  
Department of Information  
Technology  
K. J. Somaiya College of  
Engineering  
Mumbai, India

**Pratyusha Reddy**  
Department of Information  
Technology  
K. J. Somaiya College of  
Engineering  
Mumbai, India

**Yash Jain**  
Department of Information  
Technology  
K. J. Somaiya College of  
Engineering  
Mumbai, India

**Ravindra Divekar**  
Department of Information  
Technology  
K. J. Somaiya College of  
Engineering  
Mumbai, India

## **ABSTRACT**

Network surveillance has become a necessity with the increase in cybercrime. Monitoring the activities of the devices on a network is paramount to the security of an organization. It is difficult for a novice user to understand the root levels of LAN monitoring through packets sniffing. In order to remove this complexity, a simple UI driven application which will enable the administrator to observe, manage and analyze the activities of the client machines is proposed. While preventing cybercrime the system also aims to create a discipline in the activities of the users working in the organization.

## **Keywords**

LAN, monitoring, networks, OCR, administration

## **1. INTRODUCTION**

There is a plethora of monitoring software applications that work on the concept of screenshot sharing. In this paper, an approach of blending the administration and monitoring of devices on a local network with a clean user interface is discussed. Further, the idea to generate a score of the client machine based on screenshots using optical character recognition (OCR) is proposed to reduce the need for human attention and interruption.

The primary requirement of the system is the connection of each client to the admin. The administrator will be able to remotely monitor the applications running on client computers and also terminate them if needed. The administrator will be able to send personal and broadcast messages and shut down machines as well [3].

## **2. PROPOSED SYSTEM**

The system proposed in this paper requires a physical connection between clients and the administrator, making it suitable for Local Area Networks. On start-up, the client machines connect to the administrator machine. For every connected machine, the administrator machine requests screenshots at a defined interval of time.

The administrator also sets keywords for sessions, and they are sent to the appropriate clients. Based on the running applications, and the comparison of the text in the screenshots with the set keywords, a relevance score is generated for the client's work over the session. The UI is responsive to this score and is color-coded according to it.

On noticing ill behaviour, the administrator can message the client to alert the client. The administrator can view the applications running on the client and depending on the severity, stop specific applications and even shut the machine if required [3]. To communicate general information, the administrator can broadcast a message to all the clients. If there is a need to ignore a client machine for security reasons, the administrator can disconnect it after which the screenshots of that client machine will no longer be stored or sent. Lastly, on session end, the administrator can check for connected clients and shut all the connected machines to securely close the session.

## **3. FEATURES OF THE SYSTEM**

The application is lightweight and easy to install. The use of multiple threads makes the processing faster.

Following are the features:

- Remotely monitor the activities of client machines through the screenshots received
- Generation of the score for client machines
- Demand a live screenshot of the client machine at any instance
- Send a message to any connected client machine
- Broadcast message to the connected client machines
- Remotely shutdown any connected client machine
- Disconnect a particular client machine
- View and manage processes running on the client machine

## 4. IMPLEMENTATION TOOLS

### 4.1 Tesseract - OCR

Tesseract is an open source OCR engine which was released under Apache license and sponsored by Google. It is a powerful tool which can be easily used with a variety of programming languages and also has strong community support.

Optical Character Recognition (OCR) is the technology that is used for recognition of text in an image. The following pre-processing is proposed to get the optimum efficiency from the Tesseract OCR. The computer first converts the screenshot image into a gray scale image. The image undergoes appropriate pre-processing to remove any noise present in the image. After the image is pre-processed the text in the image gets photo scanned character by character. This data is then analyzed and translated to ASCII character code. The text data that is produced has a lot of junk symbols which is cleaned using regular expressions. The final text data contains the text recognized by the Tesseract OCR which is free of junk values and is stored in a list format.

Getting clean text data using OCR is the first step of generating a relevancy score for the client machine.

### 4.2 Flask - Web Framework

The proposed system is developed as a web application. This results in a multiplatform, highly flexible, readily deployable and intuitive application. The web application is built to work without the internet which decreases the number of requirements to install and use the software. On running the application, a web browser opens up, with a predefined address, comprising the UI. An HTTP request will be sent and the server is started. Flask is used to manage this request of the administrator.

Flask is a BSD licensed microframework for Python. It is based on Werkzeug and Jinja2. It figures out what the requests are, what is being asked and what response is needed to send back to the administrator. Flask framework was selected because of its simple process of web application development. It allows focus on the administrator request and response in a seamless manner.

## 5. SCORE GENERATION

The process of monitoring each screenshot manually is a bit tedious in a large LAN. A scoring system is used to score each client machine thus removing the need to check all the screenshots manually. These scores are calculated based on the work performed on the client machines in comparison with the work assigned to them by the administrator.

In order to calculate the relevance of the data, the administrator inputs keywords related to the assigned task. These keywords are then matched with the text data generated after applying OCR to the client machine's screenshot. The score is generated by counting the number of keywords that occur in the text data generated from the screenshot. Higher the occurrence, the higher the score. The screenshots of the client machines when some software applications are running in the foreground do not produce meaningful text data which can be matched with the keywords. The list of running applications is checked for such applications and changes in the score are made accordingly.

In order to give the administrator complete information regarding the client's work, the scores from multiple screenshots of the client machine over a period of time are

calculated and stored. Weights are assigned to the scores based on the time it arrives at. The weight is higher for scores generated from the latest screenshots.

The formula of score calculation is given by:

$$\sum_{i=0}^n W_i * S_i$$
$$\text{Net Score} = \frac{\quad}{n}$$

Where,

W = weight assigned to the score

S = Score generated from screenshots

n = Number of screenshots sent till calculating the net score

The clients are classified using a color-coded interface which exploits the ease of understanding of colors by a human brain thus making a clean and convenient UI.

Table 1 shows the scores generated for each keyword set against a particular screenshot. The screenshots used are of the following:

Idle Desktop, Google search for machine learning, YouTube search for machine learning, Wikipedia search for machine learning, Google search for Wikipedia

Thus from the following observations we can conclude that the relevancy score is most reliable for a rich set of keywords that are closely related to the task assigned. Moreover, it is observed that generic words like Google, YouTube, and Wikipedia might lead to unreliable higher scores for irrelevant images. Using larger sets of keywords won't necessarily increase the score as all of the text might not be captured through the screenshots.

## 6. USE CASES

### 6.1 Educational Institutions

This application is targeted to be used by college faculty while conducting laboratory sessions. This will provide an easy alternative to the manual assessment of students' performance in the laboratory. This application can be used to send and install software in student's computers through LAN, thus saving the time of individual manual installation. Online attendance systems can also be incorporated, thus automating the laboratory experience further.

### 6.2 Corporations

This application can be used to monitor the activities of multiple employees. It can be useful for ensuring that they are committed to their work. The software can also be tweaked to detect any unauthorized external devices connected to the employee machines.

Generally, in corporations, most of the suspicious services are denied access. In some exceptional cases, such services are needed. The proposed system provides an alternate approach to such situations. The corporations can block the surely harmful services and allow access to some suspicious services that are needed, as the administrator holds the power to terminate them for someone who tries to abuse the access.

### 6.3 Online Examination Portals

This application can be used as a plug-in for online examination portals to ensure that students attempting the exams do not cheat hence providing a fair environment for all the exam takers.

**Table 1: Score generation based on different sets of Keywords for different screenshots**

List of Keywords	Screenshots				
	Desktop	Google	YouTube	Wikipedia	Google homepage
Google, Wikipedia, YouTube, machine, learning	0	40	36	37	10
neural, network, regression, machine, learning	0	44	41	43	0
neural, network, regression, epoch, tensorflow	0	0	0	2	0
neural, network, regression, epoch, tensorflow, machine, learning, unsupervised, supervised	0	40	36	45	0
machine, learning	0	33	35	37	0

**Table 2: Comparison of network monitoring and administration projects [4]**

Features	Network monitoring and administration projects					
	Master BLACK (proposed s/w)	ActivTrak [5]	Amelia [6]	SolarWinds - Network Performance Monitor [7]	Wireshark [8]	PRTG [9]
Open source	-	-	✓	-	✓	-
Free	-	✓	✓	-	✓	✓
UI based on	Web	Cloud	Windows Application	Web	Window / Mac application	Windows/Mac/Android application
Screenshot sharing	✓	✓	✓	-	-	-
View Client IP, port	✓	-	✓	-	-	-
View running applications	✓	-	-	-	✓	✓
Video Playback, Alarms, Website blocking	-	✓	-	-	-	-
Network performance	-	-	✓	✓	-	-
Traffic, packets, bandwidth	-	-	-	-	✓	✓
Cloud services, Databases	-	-	-	-	-	✓
Client work-relevancy score	✓	-	-	-	-	-
Method	Socket Programming, OCR	-	java.net API via TCP	SNMP	Packet Sniffing	SNMP version 1, 2c 3
Use Case	Laboratory Session Monitoring	Insider Threat Detection, Organizational Efficiency, etc.	General Network Monitoring	Advanced Network Alerting	Network protocol analyzer	Infrastructure Management, Network Monitoring

## **7. COMPARISON WITH EXISTING PRODUCTS**

Numerous software applications have been developed on the concept of LAN Monitoring. Each has its own features focusing on different aspects of monitoring. Table 2 shows a comparison of a few features of some of the existing LAN monitoring and administration products.

## **8. CONCLUSION**

Software projects for monitoring devices over a network are available, but they tend to occupy human resources for constant monitoring. The score generation technique has proved to be quite accurate when given a smart choice of keywords. This allows the administrator to check the system at any point in time and obtain a concise observation of the relevance of every client's work throughout the time span. Thus, the application saves the time of monitoring each client's machine individually.

Furthermore, the current administration methodologies rely heavily on denial of access to suspicious services and generally provide information that is difficult to comprehend for a naive user. The features provided by the proposed system are intuitive and straightforward providing a raft of useful options for monitoring as well as administering devices over LAN. The system stands suitable for educational institutions and can be readily scaled for deployment in the corporate sphere.

## **9. FUTURE SCOPE OF APPLICATION**

The software can be hosted on a mobile platform to introduce portability. Basically, an application can be developed to keep an eye on devices over a network from any remote GPRS enabled device [1][2]. Additional features that can be added are storing the live screenshots to examine them if required and installing software concurrently on all connected machines.

In order to further improve the efficiency of the scoring system, documents related to the work to be performed can be used to extract a rich list of keywords.

## **10. REFERENCES**

- [1] Karishma Gidge, Kalyani Patil, Priyanka Wadnere, "Android Based LAN Monitoring", Journal for Research in Applied Science & Engineering Technology (IJRASET) 3 (January 2015).
- [2] Meghana Sapkal, Shekhar Patil, Leesensa Vispute, Santosh Jagtap, "GPRS Based LAN Monitoring and Controlling", IOSR Journal of Computer Engineering 16 (May-Jun 2014), pp. 09–15.
- [3] Harsh Mittal, Manoj Jain, Latha Banda, "Monitoring Local Area Network Using Remote Method Invocation", International Journal of Computer Science and Mobile Computing (IJCSMC) 2 (May 2013), pp. 50–55.
- [4] Paul Ferrill. The Best Network Monitoring Software of 2018. URL: <https://in.pcmag.com/cloud-services/97759/the-best-network-monitoring-software>
- [5] ActivTrak, <https://support.activtrak.com/hc/en-us>
- [6] Amelia, <https://github.com/Obsidiam/amelia>
- [7] SolarWinds - Network Performance Monitor, [https://support.solarwinds.com/@api/deki/files/10232/NP\\_M\\_Administrator\\_Guide.pdf?revision=22](https://support.solarwinds.com/@api/deki/files/10232/NP_M_Administrator_Guide.pdf?revision=22)
- [8] Wireshark, [https://www.wireshark.org/docs/wsdg\\_html/](https://www.wireshark.org/docs/wsdg_html/)
- [9] PRTG, <https://www.paessler.com/manuals/prtg>

# Acknowledgement

We take this opportunity to express our profound gratitude and deep regards to our guide Prof. Ravindra Divekar for his exemplary guidance, monitoring and constant encouragement throughout the course of this project.

We thank Dr Irfan Siddavatam, Head of Department of Information Technology, of KJSCE, Mumbai for encouragement and providing the necessary infrastructure for pursuing the project.

We are obliged to staff members of K. J. Somaiya College of Engineering, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.