Style Guidelines for Final Year Project ReportsSurveilia

Final Year Project – Mid Report

Session 2017-2021

A 4th Year Student

A project submitted in partial fulfilment of the

COMSATS University Degree

of

BSc. (Hons.)BS in Computer Science (CUI)



Department of Computer Science

COMSATS University Islamabad, Lahore Campus

31 July 2020

# Evaluation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Group Members** (To be filled by students) | | | | (To be filled by supervisor) |
| Sr.# | Reg. # | Student Name | \*Signature | Obtained Marks  (Total Marks: 10) |
| (i) | SP17-BCS-109 | IFRAH TEHLEEL | A picture containing sitting, air  Description automatically generated |  |
| (ii) | SP17-BCS-145 | NAUMAN AKRAM | A picture containing drawing  Description automatically generated |  |
| (iii) | SP17-BCS-028 | JAN MUHAMMAD MIRZA | A picture containing light  Description automatically generated |  |

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others.

Supervisor Name: **DR.USAMA IJAZ BAJWA** Supervisor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Remarks (if any): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

Surveillance cameras are gradually being used at every place that catches the anomalous event, yet the checking capacity of security agencies has not met the level. To overcome this problem, we proposed the idea to develop an automated anomaly detection application by using deep learning through activity recognition algorithms and video analysis. The proposed application will detect anomalies such as burglary, theft, vandalism, etc., and its location in surveillance CCTV and alarm the security team, instantly. It would be able to differentiate between the abnormal and normal events in the live stream. It would detect anomalies in both environments; outdoor and indoor. As the matter of security is very important in any organization, so the main purpose is to automate the security. With the utilization of the proposed project, one can achieve the novel solution of its security issues.

**Acknowledgement**

We have taken efforts in this project and are highly indebted and grateful to COMSATS University Islamabad, Lahore for their guidance and constant supervision of Dr. Usama Ijaz Bajwa. We are grateful to Dr. Usama for choosing us for this project. He provided us with constant help, support, and guidance. It would have not been possible to start this project without his precious support and help.

**Table of Contents**

[Surveilia 1](#_Toc47101616)

[Final Year Project – Mid Report 1](#_Toc47101617)

[Session 2017-2021 1](#_Toc47101618)

[Evaluation 2](#_Toc47101619)

[\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others. 2](#_Toc47101620)

[1 Introduction 12](#_Toc47101621)

[1.1 Introduction 12](#_Toc47101622)

[1.2 Objectives 13](#_Toc47101623)

[1.3 Problem statement 13](#_Toc47101624)

[1.4 Assumptions & Constraints 13](#_Toc47101625)

[1.4.1 Assumptions 13](#_Toc47101626)

[1.4.2 Constraints 14](#_Toc47101627)

[1.5 Project scope 14](#_Toc47101628)

[2 Requirements Analysis 16](#_Toc47101629)

[2.1 Literature review 16](#_Toc47101630)

[2.1.1 Research-Based Related work 16](#_Toc47101631)

[1) Advance intelligent video surveillance system (AIVSS) 16](#_Toc47101632)

[2) Anomaly Localization in Topic-based Analysis of Surveillance Videos 17](#_Toc47101633)

[2.1.2 Application-Based Related Work 18](#_Toc47101634)

[1) Surveillance App 18](#_Toc47101635)

[2) iCentana 19](#_Toc47101636)

[3) Dahua Security 19](#_Toc47101637)

[4) Hikvision 19](#_Toc47101638)

[5) Mobotix 19](#_Toc47101639)

[2.2 Stakeholders list 19](#_Toc47101640)

[2.3 Requirements elicitation 20](#_Toc47101641)

[2.3.1 Functional requirements 20](#_Toc47101642)

[2.3.2 Non-functional requirements 22](#_Toc47101643)

[2.3.3 Requirements traceability matric 24](#_Toc47101644)

[2.4 Use case descriptions 25](#_Toc47101645)

[2.4.1 Use case-01: Login 25](#_Toc47101646)

[2.4.2 Use case-02: Logout 25](#_Toc47101647)

[2.4.3 Use case-03: Anomaly Detection 26](#_Toc47101648)

[2.4.4 Use case-04: Re-Training the Model 27](#_Toc47101649)

[2.4.5 Use case-05: To View Profile 27](#_Toc47101650)

[2.4.6 Use case-06: View anomaly detected clips 28](#_Toc47101651)

[2.4.7 Use case-07: View Users 29](#_Toc47101652)

[2.5 Use case design 30](#_Toc47101653)

[2.5.1 Use case-01: User and Admin Login 30](#_Toc47101654)

[2.5.2 Use case-02: Logout 31](#_Toc47101655)

[2.5.3 Use case-03: Anomaly Detection 32](#_Toc47101656)

[2.5.4 Use case-04: Re-Training the model 33](#_Toc47101657)

[2.5.5 Use case-05: View Profile 34](#_Toc47101658)

[2.5.6 Use case-06: View Anomaly Detected Clip 35](#_Toc47101659)

[2.5.7 Use case-07: View User 36](#_Toc47101660)

[2.5.8 Use case-08: SURVEILIA (Complete System) 37](#_Toc47101661)

[2.6 Software development life cycle model 38](#_Toc47101662)

[2.6.1 Model Used in our project: 38](#_Toc47101663)

[2.6.2 Why? 38](#_Toc47101664)

[3 System Design 41](#_Toc47101665)

[3.1 Work breakdown structure 41](#_Toc47101666)

[3.2 Activity diagram 42](#_Toc47101667)

[3.2.1 Login 42](#_Toc47101668)

[3.2.2 Anomaly Detection 43](#_Toc47101669)

[3.2.3 Passing through the model 44](#_Toc47101670)

[3.3 Sequence diagram 45](#_Toc47101671)

[3.3.1 Sequence Diagram: Login 45](#_Toc47101672)

[3.3.2 Sequence Diagram: Create a New User 46](#_Toc47101673)

[3.3.3 Sequence Diagram: Anomaly Detection 47](#_Toc47101674)

[3.4 Software architecture 48](#_Toc47101675)

[3.5 Network diagram 49](#_Toc47101676)

[3.6 Collaboration diagram 50](#_Toc47101677)

[4 System Testing 52](#_Toc47101678)

[4.1 Unit Testing 52](#_Toc47101679)

[4.1.1 Test Case-01 52](#_Toc47101680)

[4.1.2 Test Case-02 52](#_Toc47101681)

[4.1.3 Test Case-03 53](#_Toc47101682)

[4.1.4 Test Case-04 54](#_Toc47101683)

[4.2 Integration Testing 54](#_Toc47101684)

[4.3 Acceptance testing 54](#_Toc47101685)

[5 Conclusion 56](#_Toc47101686)

[5.1 Problems faced and lessons learned 56](#_Toc47101687)

[5.1.1 Size of Dataset: 56](#_Toc47101688)

[5.1.2 Google Colab Limitations: 56](#_Toc47101689)

[5.2 Project summary 56](#_Toc47101690)

[5.3 Future work 57](#_Toc47101691)

[6 References 59](#_Toc47101692)

**List of Tables**

[Table 1: Functional Requirement-01: Login 20](#_Toc47101100)

[Table 2: Functional Requirement-02: Dashboard 21](#_Toc47101101)

[Table 3: Functional Requirement-03: Anomaly Detection 21](#_Toc47101102)

[Table 4: Functional Requirement-04: Video Database 21](#_Toc47101103)

[Table 5: Functional Requirement-05: Register New User 22](#_Toc47101104)

[Table 6: Non-Functional Requirement-01: Security 22](#_Toc47101105)

[Table 7: Non-Functional Requirement-02: Performance 23](#_Toc47101106)

[Table 8: Non-Functional Requirement-03: Usability 23](#_Toc47101107)

[Table 9: Non-Functional Requirement-04: Completeness 23](#_Toc47101108)

[Table 10: Non-Functional Requirement-05: Robustness 24](#_Toc47101109)

[Table 11: Requirement Traceability Matrix 24](#_Toc47101110)

[Table 12: Use case description-01: Login 25](#_Toc47101111)

[Table 13: Use case description-02: Logout 25](#_Toc47101112)

[Table 14: Use case description-03: Anomaly Detection 26](#_Toc47101113)

[Table 15: Use case description-04: Re-Training the model 27](#_Toc47101114)

[Table 16: Use case description-05: View Profile 28](#_Toc47101115)

[Table 17: Use case description-06: View Anomaly Detected Clip 28](#_Toc47101116)

[Table 18: Use case description-07: View Users 29](#_Toc47101117)

[Table 19: Gantt Chart 49](#_Toc47101118)

[Table 20: Test Case-01 52](#_Toc47101119)

[Table 21: Test Case-02 52](#_Toc47101120)

[Table 22: Test Case-03 53](#_Toc47101121)

[Table 23: Test Case-04 54](#_Toc47101122)

**List of Figures**

[Figure 1: Architecture of AIVSS [6] 16](#_Toc47101078)

[Figure 2: Performance of AIVSS [6] 17](#_Toc47101079)

[Figure 3: Localization of Anomaly [7] 17](#_Toc47101080)

[Figure 4: Dense Sampling [9] 18](#_Toc47101081)

[Figure 5: Use case-01: Login 30](#_Toc47101082)

[Figure 6: Use case 02- Logout 31](#_Toc47101083)

[Figure 7: Use case-03: Anomaly Detection 32](#_Toc47101084)

[Figure 8:Use case-04: Re-Training the model 33](#_Toc47101085)

[Figure 9: Use case-05: View Profile 34](#_Toc47101086)

[Figure 10: Use case-06: View Anomaly Detected Clip 35](#_Toc47101087)

[Figure 11: Use case-07: View User 36](#_Toc47101088)

[Figure 12: Use case: Surveilia 37](#_Toc47101089)

[Figure 13: Incremental Model 39](#_Toc47101090)

[Figure 14: Work Breakdown Structure 41](#_Toc47101091)

[Figure 15: Activity Diagram-01: Login 42](#_Toc47101092)

[Figure 16: Activity Diagram-02: Anomaly Detection 43](#_Toc47101093)

[Figure 17: Activity Diagram-03: Passing through the model 44](#_Toc47101094)

[Figure 18: Sequence Diagram: Login 45](#_Toc47101095)

[Figure 19: Sequence Diagram: To Create New User 46](#_Toc47101096)

[Figure 20: Sequence Diagram: Anomaly Detection 47](#_Toc47101097)

[Figure 21: System Architecture 48](#_Toc47101098)

[Figure 22: Collaboration Diagram 50](#_Toc47101099)

**Chapter 1**

**Introduction**

# Introduction

## Introduction

Human Activity Recognition also termed as HAR, is the process of recognizing the human movement in any area that can be for say indoor/outdoor. It revolves around the recognition of goals and activities of agents/s from a series of observations on the actions of agents and environmental conditions.

As we have seen numerous cameras located at every street, malls, organizations and even now houses have surveillance cameras that continuously monitor the movement of people. Even though they are placed to detect human activities, there are groups of people sitting behind those multiple cameras who are constantly monitoring and analyzing the video stream to see what activities are being done by people and whether there is any unusual activity or not. It is exhausting work. Any sudden or minor deviation from the screens, where the live feed is being displayed, can result in various consequences. In the past, there were several incidents in buildings that included CCTVs, but the system is inefficient, and traditional couldn’t detect them. Puyallup South Hill Mall Burglary [1] and Westfield Century City Mall Robbery [2] and many others are an example of such incidents. The anomaly detection in surveillance videos is important because surveillance cameras are rapidly being used in public places that capture a variety of realistic anomalous events, but the monitoring abilities of law enforcement agencies have not kept the speed. There is a need for the system to automatically detect any kind of abnormal human activities so that complete safety is guaranteed.

Our proposed project would perform the task of identifying unexpected incidents by using Video Analytics. Video Analytics has become a hot topic in the world and by using the techniques of Video Analytics plenty of work has been done in detecting human activities. We will initially train and test our system using the Something-Something V1 dataset [3], Something-Something V2 dataset [4] , and UCF Crime dataset [5].

There are many solutions present to detect human activity in surveillance videos, but the main issue is that most of them require the users to change their system architecture and system hardware, due to which it turns out to be very costly. The proposed project will provide the solution by not requiring any update in their systems or usage of any new, advanced, and expensive technology.

## Objectives

The objectives for the proposed project are as follows:

* To build a product that can detect humans’ activities and their interpretation through automated video analysis using deep learning knowledge.
* The product should assist the security personnel so that the process of handling anomalies can be improved in real-time.
* To automate the anomaly detection process likely to be as human
* Reducing the “False Alarm” rate to the minimum to increase the consistency of the product

## Problem statement

The present world has advanced but still one does not feel secure at the malls and other public places. We hear news about theft, vandalism, and other street crimes, daily. Though the surveillance cameras are increasingly being placed at almost all public places still some of the events get unnoticed or are noticed when it’s too late because the monitoring skills of law enforcement have not kept up the pace. The surveillance is limited to the human operators which limits the multi-screen monitoring and the high cost of labour is required. It is almost impossible for a human eye to concentrate and go through all the screens, thoroughly, when multiple actions are taking place at the same time on each screen. Due to this reason, anomalous events get missed from the eyes of law enforcement and crimes still exist.

To solve this problem, there is a need for an automated system that will differentiate between normal and abnormal events, and in case of any abnormal event, it would notify the security team to take appropriate actions, immediately. The proposed project is an automated system that would spot abnormal activities in real-time and will alarm the security team to take appropriate actions against the anomalous event, immediately.

## Assumptions & Constraints

### Assumptions

The proposed project is assumed to be a user-friendly application that will help manage surveillance tasks by automatically detecting anomalies and resultantly, the security system will be improved. The product will benefit through:

* Reducing human efforts as it would save the exhaustive task of 24/7 watch
* Real-time spotting anomalies and other related events
* Identifying the suspects and real victims
* Reducing crime rates by introducing strict checks everywhere
* Live to monitor of targeted area/location

### Constraints

Although there are a lot of pros in the proposed application, there are some small but technical cons; the detection of abnormal and normal activities as separate is doubtful, especially in crowded areas. Moreover, the unavailability of labelled datasets makes the system to be inefficient as its learning is not enough. As the project aims to deal with live camera feed, the time will be consumed while pre-processing the live feed and passing it to the inferencing model. There will be a delay of a few milliseconds in informing the user about the abnormal event, resultantly. Moreover, our project will not give the option to the user to choose between which suspicious activity he/she wants to detect at a camera located at a specific location. Another major issue is maintaining the privacy of the public while detection anomaly.

## Project scope

Video analytics has become an emerging topic in the modern era. A lot of modification is being done daily in Computer Vision. As a subfield, understanding human activities is very demanding and we are keen to work in this domain to meet the concerns of security issues.

Security is one of the main concerns for every person, either at home or in public places. Nowadays traditional CCTVs are used at almost every place, be it home or public area. Security guards sit behind those CCTV screens, inspecting the behaviour of people, and monitoring their activities. But unfortunately, it is out of the capacity of the human brain to focus on multiple screens at a single time. Multiple applications are being built to automatically detect any abnormal activity so the security can be ensured, and people can move without concerns of safety. We are developing an application that will automatically detect any kind of abnormality in surveillance CCTV streams. The proposed application will include,

* Detection of abnormal events such as burglary, theft in real-time
* Detecting such activities which require the attention of security personnel so that such an event can be dealt on time and can be minimized.

**Chapter 2**

**Requirements Analysis**

# Requirements Analysis

## Literature review

A lot of work has been done in the field of anomaly detection through various approaches to detect abnormal activities. A few of the working systems are discussed below.

### Research-Based Related work

### Advance intelligent video surveillance system (AIVSS)

AIVSS [6] is a research-based article describing the need for intelligent video surveillance systems as surveillance cameras are increasing day by day. The paper proposes the idea to develop the surveillance cameras based on the internet protocol (IP).

The author explains that the IP videos are much better in terms of processing power and improved detection. Figure 1 below depicts the architecture of AIVSS. It shows that all the elements of the system are connected using IP cameras.

A close up of a device

Description automatically generated

Figure 1: Architecture of AIVSS [6]

Figure 2 below depicts the system’s performance. It not only depends on the hardware and software but the privacy, too. The author explains that the privacy issue is a crucial factor in many countries. On the other side, the author explains that the network performance is also important.

A screenshot of a cell phone

Description automatically generated

Figure 2: Performance of AIVSS [6]

### Anomaly Localization in Topic-based Analysis of Surveillance Videos

The paper [7] in the subject here recommends a technique to spot the location of an anomaly in the video with a large number of agents actively pursuing different tasks, and the size information by integrating Spatio-temporal gradient descriptors. The author has discussed the framework to be three-tier for the analysis of abnormal videos. The flow of the application is; modelling, detection, and then localization of anomaly. Figure 3 below depicts the localization of anomaly using different techniques.

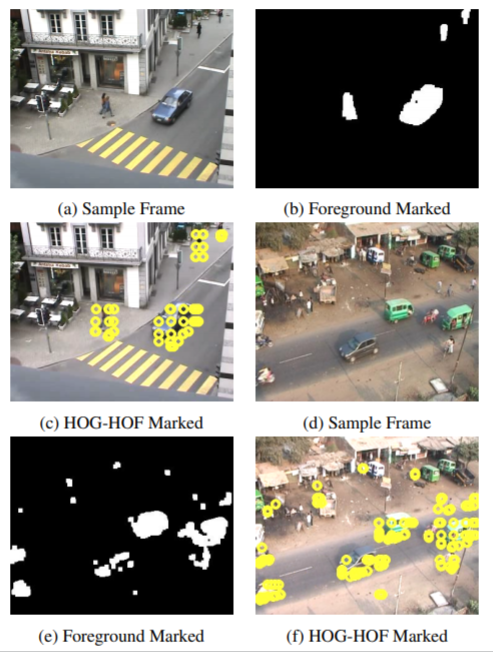


Figure 3: Localization of Anomaly [7]

1. **Video Anomaly Detection in Confined Areas**

The paper [8] focuses on the detection of an anomaly in confined areas. It proposes a new technique and algorithm for detection at such places. According to this research paper, the event is said to be abnormal when the speed of the object, the motion of the object in forbidden time, or the path of the object is different from what it is trained with. The system converts the video into grayscale first and then it samples the frames into Spatio-temporal volumes using dense sampling as depicted in the Figure 4. The events with relatively lesser occurrence are detected as anomalous.

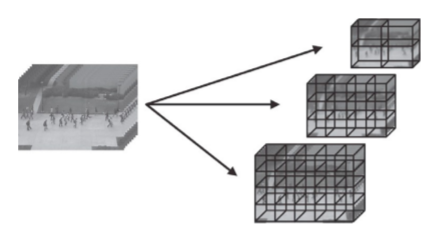


Figure 4: Dense Sampling [9]

1. **Survey Paper on Anomaly Detection in Surveillance Videos**

In paper [9], the author describes the need for automated anomaly detection in surveillance videos. The method used in this paper is to train the neural network using deep multiple instance learning. But it only detects the presence of the anomaly. The location and time cannot be detected. This paper motivated us to create an application that not only detects the anomaly but also the location and time.

### Application-Based Related Work

### Surveillance App

Surveillance App [10] is a real-time wireless surveillance CCTV home system app developed by Reservoir Dev, which detects noise or any movement in the live stream and notifies the viewer. It allows the user to monitor his/her home and speak through the user’s device microphone to scare off the thief/intruder. But it is limited to indoor or specifically, home of the user.

### iCentana

iCentana [11] is AI video analytics for automated real-time identification of critical events. It monitors cameras and automatically detects any abnormal activity. It identifies precursor events, learns and adapts automatically, rapidly reviews recorder video, and detects abnormal precursor events. They are costly and lack in localization of anomaly.

### Dahua Security

A Chinese CCTV company [12] launched cameras to detect anomalies in public areas. Their mission is “Enabling a safer society and smarter living”. They have cameras with integrated chips to be able to run deep neural networks right on board. Key technologies used are HDCVI Technology, Predictive Focus Algorithm, EPOE, and ANPR. Though, it is much accurate but requires special CCTV cameras with integrated chips, which are very costly.

### Hikvision

Hikvision [13] provides digital surveillance and is also Chinese based, CCTV cameras embedded with intelligent video analytics, company. But people report that it is poor efficiency, does not train properly and causes much delay in alarming the security person which is a huge problem.

### Mobotix

Mobotix [14] provides a surveillance system for both indoor and outdoor. It is secure to use. But it is very much complex in terms of its user interface. Most of the important features are hidden behind the toolbars.

## Stakeholders list

Stakeholders of our project will be

* Project team
  + Ifrah Tehleel
  + Jan Muhammad Mirza
  + Nauman Akram
* Project Supervisor
  + Dr. Usama Ijaz Bajwa
* Security Firm
* Anomaly Detection Researchers
* Data Analyst
* Law Enforcement
* Final Year Project Jury
* CUI Lahore

## Requirements elicitation

### Functional requirements

A functional requirement is a description of the facility that the software should perform. It defines a system or its components. It benefits you to capture the planned performance of the system.

#### Functional Requirement-01: Login

In Table 1 below, the functional requirements of the login page are mentioned.

Table 1: Functional Requirement-01: Login

|  |  |
| --- | --- |
| **ID** | **Functional Requirements** |
| FR11 | The application shall permit the user to log in. |
| FR12 | The application shall maintain a separate account for each user. |
| FR13 | The application shall grant the admin access after verification of login credentials. |
| FR14 | The application shall permit the admin to update his/her username and password |
| FR15 | The application shall only permit the admin to change the login credentials of other users. |
| FR16 | The application shall maintain a database of all the user accounts. |

#### Functional Requirement-02: Dashboard

In Table 2 below, the functional requirements of the dashboard are mentioned.

Table 2: Functional Requirement-02: Dashboard

|  |  |
| --- | --- |
| **ID** | **Functional Requirements** |
| FR21 | The application shall permit the user to choose between options displayed on the dashboard. |

#### Functional Requirement-03: Anomaly Detection

In Table 3 below, the functional requirements of anomaly detection are mentioned.

Table 3: Functional Requirement-03: Anomaly Detection

|  |  |
| --- | --- |
| **ID** | **Functional Requirements** |
| FR31 | The application shall alert the user/admin when an anomaly is detected. |
| FR32 | The application shall permit the admin and the user to view the clipped videos. |

#### Functional Requirement-04: Video Database

In Table 4 below, the functional requirements of the video database are mentioned.

Table 4: Functional Requirement-04: Video Database

|  |  |
| --- | --- |
| **ID** | **Functional Requirements** |
| FR41 | The application shall have a database of anomaly clips called anomaly history. |
| FR42 | The application must maintain a database of all the live feeds. |
| FR43 | The application shall allocate a unique ID to each clip in anomaly history. |
| FR44 | The application shall permit the admin to view any video clip from the anomaly history. |
| FR45 | The application shall permit the admin to remove or delete any video clip from the anomaly history. |

#### Functional Requirement-05: Register New User

In Table 5 below, the functional requirements of adding the new user are mentioned.

Table 5: Functional Requirement-05: Register New User

|  |  |
| --- | --- |
| **ID** | **Functional Requirements** |
| FR51 | The application shall only permit the admin to add a new user. |
| FR52 | The admin must add the information of the new user. |

### Non-functional requirements

Non-functional requirements describe the performance quality of the software system. It permits you to impose limitations on the design of the system.

#### Non-Functional Requirement-01: Security

The non-functional requirements regarding security are mentioned in Table 6.

Table 6: Non-Functional Requirement-01: Security

|  |  |
| --- | --- |
| **ID** | **Non-Functional Requirement** |
| NFR11 | Only the Logged in/Signed up user shall be allowed to access the functionalities. The application must only allow authorized users with correct login credentials to access the system. |
| NFR12 | The application shall be secure. |

#### Non-Functional Requirement-02: Performance

The non-functional requirements regarding the performance of the application are mentioned in Table 7.

Table 7: Non-Functional Requirement-02: Performance

|  |  |
| --- | --- |
| **ID** | **Non-Functional Requirement** |
| NFR21 | The start-up time of the application must not be more than 20 seconds. |
| NFR22 | The application must alert the user when an anomaly has detected, and the time required for it must be no more than 10 seconds. |
| NFR23 | The response time between click and reaction must be less than two seconds. |

#### Non-Functional Requirement-03: Usability

The non-functional requirements regarding the usability are mentioned in Table 8.

Table 8: Non-Functional Requirement-03: Usability

|  |  |
| --- | --- |
| **ID** | **Non-Functional Requirement** |
| NFR31 | The interface of the application must be easy to understand. |
| NFR32 | The user must get familiar with it in no more than 20 seconds. |

#### Non-Functional Requirement-04: Completeness

The non-functional requirements regarding the completeness are mentioned in Table 9.

Table 9: Non-Functional Requirement-04: Completeness

|  |  |
| --- | --- |
| **ID** | **Non-Functional Requirement** |
| NFR41 | The application must always be consistent and efficient in detecting the anomaly. |
| NFR42 | The application must always generate the same result for a user’s task. |

#### Non-Functional Requirement-05: Robustness

The non-functional requirements regarding the robustness are mentioned in Table 10.

Table 10: Non-Functional Requirement-05: Robustness

|  |  |
| --- | --- |
| **ID** | **Non-Functional Requirement** |
| NFR51 | The application's average time to crash should not be more than 10 minutes within a month. |

### Requirements traceability matric

Table 11 displays the requirements traceability matrix.

Table 11: Requirement Traceability Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **FR\_ID** | **Description of Requirement** | **Objective** | **Priority** |
| 01 | FR-05 | The system will facilitate the users with 2 types of accounts i.e. Admin and User. | Register a new user. | High |
| 02 | FR-01 | The system will facilitate the user with a login system. | Logging user account. | High |
| 03 | FR-02 | The system allows the user to choose between the items on the dashboard. | Choosing options on Dashboard. | High |
| 04 | FR-03 | The system shall not generate false alarms. | False alarms | High |
| 05 | FR-03 | The system shall detect the anomaly. | Anomaly Detection | High |

## Use case descriptions

### Use case-01: Login

The use case description for the login process is shown in Table 12.

Table 12: Use case description-01: Login

|  |  |
| --- | --- |
| **ID:** | 1 |
| **Name of Use Case:** | Login |
| **Actors:** | Admin, User |
| **Description:** | The admin/user can log in to the application. |
| **Pre-Condition:** | Admin/User should already have an account. Users must know his/her login credentials of his/her account. |
| **Post-Condition:** | Admin/User has been signed into the application, successfully. |
| **Events:** | 1. Admin/user opens the application. 2. Admin/user clicks on the login button. 3. Admin/user fills the username and password fields. 4. On authentication from the database, the user has been signed in. |
| **Alternatives Flow:** | In case of forgetting the password, the user must contact the admin. |
| **Exceptions:** | None |

### Use case-02: Logout

The description of the use case of the logout process is shown in Table 13.

Table 13: Use case description-02: Logout

|  |  |
| --- | --- |
| **ID:** | 2 |
| **Name of Use Case:** | Logout |
| **Actors:** | Admin, User |
| **Description:** | The user can logout from the application. |
| **Pre-Condition:** | The user should be signed in his/her account. |
| **Post-Condition:** | Admin/user has been successfully logged out from the application. |
| **Events:** | 1. The user presses the logout button. 2. The user logs out and the login page appears. |
| **Alternatives Flow:** | The user has already logged out. |
| **Exceptions:** | None |

### Use case-03: Anomaly Detection

The use case description for the detection of anomaly is shown in Table 14.

Table 14: Use case description-03: Anomaly Detection

|  |  |
| --- | --- |
| **ID:** | 3 |
| **Name of Use Case:** | Anomaly Detection |
| **Actors:** | Admin, User |
| **Description:** | The application shall detect the anomaly, clip the video, store it, and alert the user via the desktop application. |
| **Pre-Condition:** | There should be a live feed. |
| **Post-Condition:** | Anomaly has been detected. |
| **Events:** | 1. The camera captures the live stream. 2. The application pre-processes the stream. 3. The application detects the anomaly. 4. The application clips the relevant part. 5. The application alerts the user and shows the video. |
| **Alternatives Flow:** | No anomaly has been detected. |
| **Exceptions:** | None |

### Use case-04: Re-Training the Model

The use case description for retraining the model is shown in Table 15.

Table 15: Use case description-04: Re-Training the model

|  |  |
| --- | --- |
| **ID:** | 4 |
| **Name of Use Case:** | Re-training the model |
| **Actors:** | Database |
| **Description:** | The application shall get the model from the database and retrain it. |
| **Pre-Condition:** | The database must exist with the model. |
| **Post-Condition:** | The model is retrained. |
| **Events:** | 1. The application gets the live stream from the CCTV. 2. The application passes it through the model. 3. The model is retrained by new training examples i.e. the live stream. |
| **Alternatives Flow:** | There is no live feed. |
| **Exceptions:** | None |

### Use case-05: To View Profile

The description for the use case to view profile is shown in Table 16.

Table 16: Use case description-05: View Profile

|  |  |
| --- | --- |
| **ID:** | 5 |
| **Name of Use Case:** | View Profile |
| **Actors:** | Admin, User |
| **Description:** | The application shall permit the user/admin to view the profile and update it. |
| **Pre-Condition:** | The user/admin must be signed in with correct sign-in credentials. |
| **Post-Condition:** | The user has viewed and updated his profile. |
| **Events:** | 1. The user/admin presses the account button. 2. The user/admin clicks the view profile button. 3. The user/admin updates the profile by entering updated information. 4. The user/admin changes his/her password if required. |
| **Alternatives Flow:** | The user/admin does not update his profile and returns. |
| **Exceptions:** | None |

### Use case-06: View anomaly detected clips

The use case description to view the anomaly detected clip is shown in Table 17.

Table 17: Use case description-06: View Anomaly Detected Clip

|  |  |
| --- | --- |
| **ID:** | 6 |
| **Name of Use Case:** | View anomaly detected clip |
| **Actors:** | User and Admin |
| **Description:** | The application shall permit the admin to view the anomaly detected clip log. |
| **Pre-Condition:** | The anomaly detected clip must be present in the log. |
| **Post-Condition:** | The clip is found and successfully played. |
| **Events:** | 1. The admin/user logs in to the application. 2. The admin/user clicks the button of anomaly clips 3. The admin/user searches a video from the clip history 4. The admin clicks the video 5. The video starts to play. |
| **Alternatives Flow:** | There is no video in the anomaly detected clip log. |
| **Exceptions:** | None |

### Use case-07: View Users

The description of the use case to view users is shown in Table 18.

Table 18: Use case description-07: View Users

|  |  |
| --- | --- |
| **ID:** | 7 |
| **Name of Use Case:** | View Users |
| **Actors:** | Admin |
| **Description:** | The application shall permit the admin to view, edit, and update the information of the other users. |
| **Pre-Condition:** | The admin must be signed into the system with the correct sign-in credentials. |
| **Post-Condition:** | The admin has viewed, updated, deleted, or added users. |
| **Events:** | 1. The admin clicks the account button. 2. The admin clicks the view users’ button. 3. The admin views, updates, deletes, or adds new users. |
| **Alternatives Flow:** | The user does not exist. |
| **Exceptions:** | None |

## Use case design

### Use case-01: User and Admin Login

The use case for the login process for the user and admin is shown in Figure 5.

A close up of text on a white background

Description automatically generated

Figure 5: Use case-01: Login

### Use case-02: Logout

The options for the logout process of the admin and the user are shown in Figure 6.

A close up of text on a white background

Description automatically generated

Figure 6: Use case 02- Logout

### Use case-03: Anomaly Detection

The options for anomaly detection are represented in Figure 7 in form of a use case diagram.

A close up of text on a white background

Description automatically generated

Figure 7: Use case-03: Anomaly Detection

### Use case-04: Re-Training the model

The use case for retraining the model is displayed in Figure 8.

A close up of a logo

Description automatically generated

Figure 8:Use case-04: Re-Training the model

### Use case-05: View Profile

The option to view the profile is shown in Figure 9.

A close up of text on a white background

Description automatically generated

Figure 9: Use case-05: View Profile

### Use case-06: View Anomaly Detected Clip

The use case to view the anomaly detected clip is represented in Figure 10.

A close up of a map

Description automatically generated

Figure 10: Use case-06: View Anomaly Detected Clip

### Use case-07: View User

The options for viewing the user and its details are shown in Figure 11.

A close up of text on a white background

Description automatically generated

Figure 11: Use case-07: View User

### Use case-08: SURVEILIA (Complete System)

The use case in Figure 12 depicts the overview of the entire system by combining all modules in one diagram for a better understanding of the reader.

A picture containing text

Description automatically generated

Figure 12: Use case: Surveilia

## Software development life cycle model

The software development life cycle is a conceptual outline that describes all the events in the development of a project from for planning and creating to maintenance and testing of the system. The basic stages are initiation, planning, designing, building, coding, testing, and deployment. Some of the most important models of this cycle:

* Incremental model
* Waterfall model
* Agile model
* Spiral model
* Iterative model
* RAD

### Model Used in our project:

We are choosing an **Incremental Model** for the development process of our system. The incremental model is one of the most important models of SDLC. In this process, requirements are broken down into multiple separate modules of the cycle. And it is iterative.

### Why?

* This model will be feasible for us if there is any alternation required throughout the project. So, it would be more reasonable.
* It is easier to identify the risks and handle them separately in the iterations.
* Our project would require revisions until we receive our final project.
* Testing and debugging during the smaller iterations would be a better option. Each iteration is an easily managed milestone.

A picture containing screenshot

Description automatically generated

Figure 13: Incremental Model

Figure 13shows how incremental works in different builds. Each build is a little more progressed at the beginning which makes the work increment in small patches and provides flexibility to the model for risk analysis.

**Chapter 3**

**System Design**

# System Design

## Work breakdown structure

The overall work breakdown structure is shown in Figure 14. It describes different phases such as the project management, design document, and so on until the testing and deployment phase.

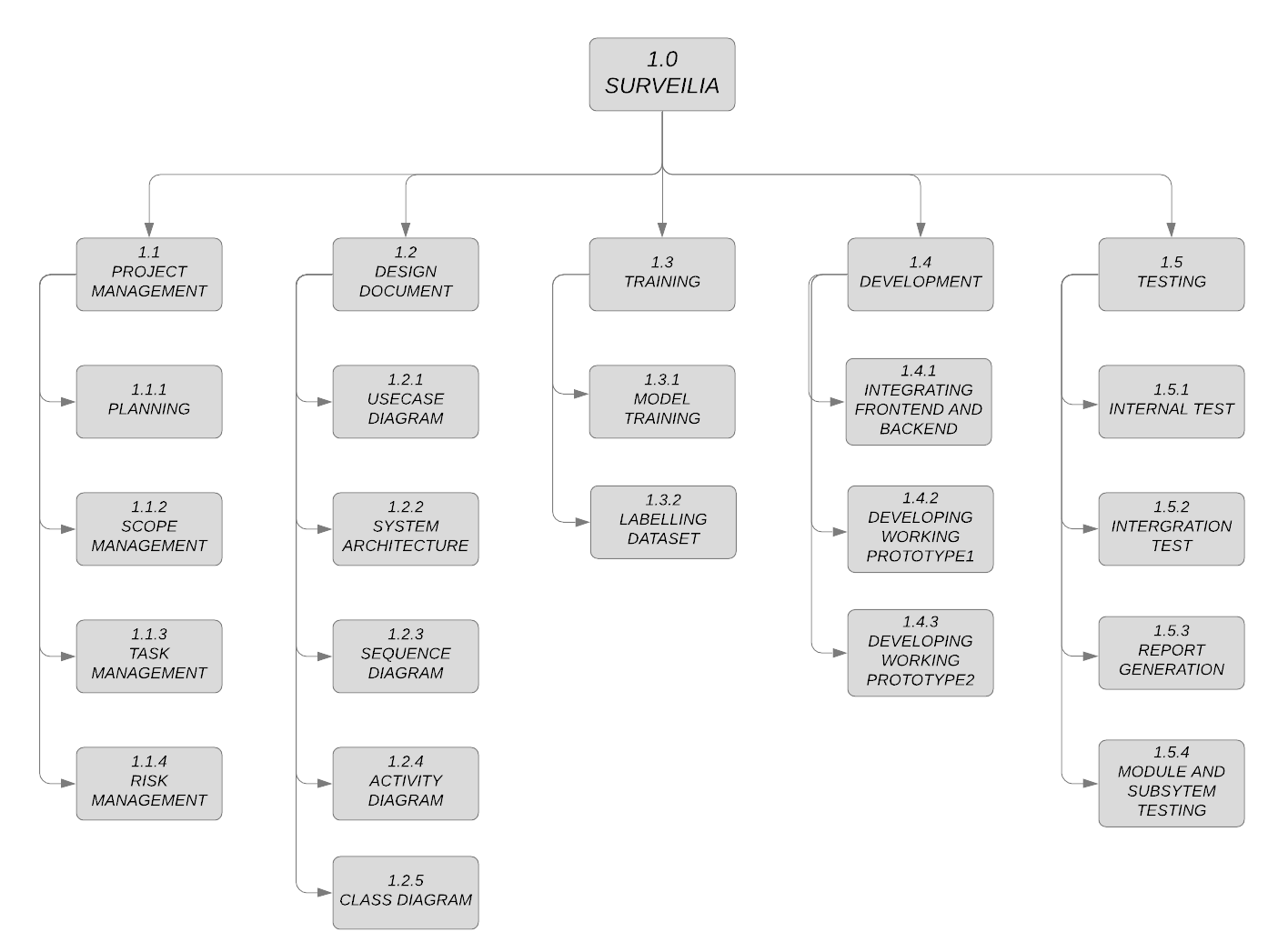


Figure 14: Work Breakdown Structure

## Activity diagram

### Login

The flow of activities of login is shown in Figure 15.

A picture containing clock

Description automatically generated

Figure 15: Activity Diagram-01: Login

### Anomaly Detection

The activity diagram of the flow of activities anomaly detection is shown in Figure 16.

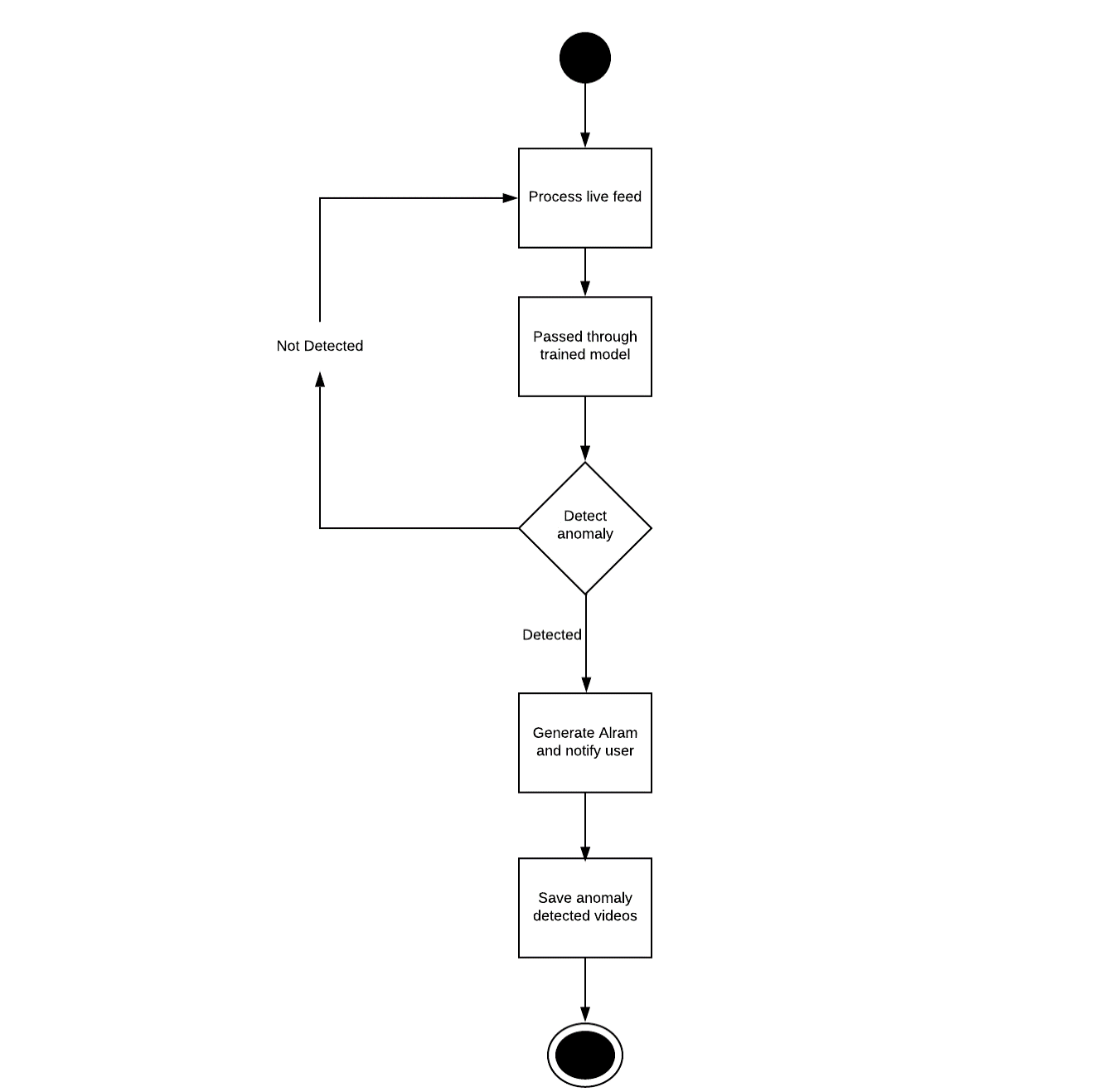


Figure 16: Activity Diagram-02: Anomaly Detection

### Passing through the model

The activity diagram of the flow of activities of passing videos through the model is shown in Figure 17.

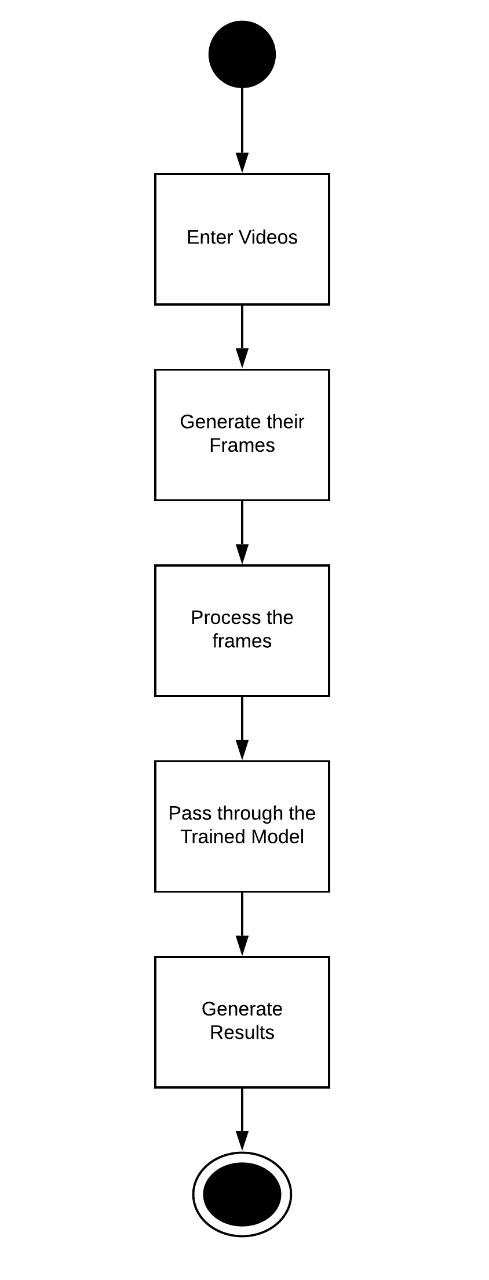


Figure 17: Activity Diagram-03: Passing through the model

## Sequence diagram

### Sequence Diagram: Login

Figure 18 shows the sequence diagram for the login. It displays the sequence of modules and their actions for login.

A screenshot of a cell phone

Description automatically generated

Figure 18: Sequence Diagram: Login

### Sequence Diagram: Create a New User

Figure 19 shows a sequence diagram to create a new user. It displays the sequence of modules and their actions to create a new user.

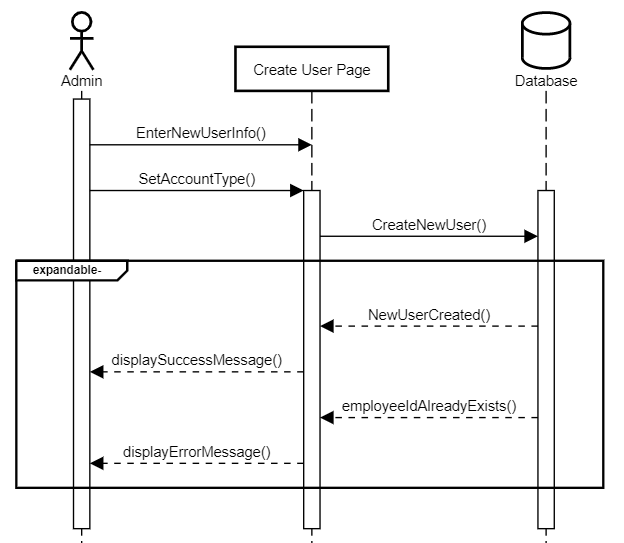


Figure 19: Sequence Diagram: To Create New User

### Sequence Diagram: Anomaly Detection

Figure 20 shows a sequence diagram for anomaly detection. It displays the sequence of modules and their actions for anomaly detection.

A screenshot of a cell phone

Description automatically generated

Figure 20: Sequence Diagram: Anomaly Detection

## Software architecture

The live video stream will be passed through a less computation-intensive model such as a temporal shift module [15]. We shall be using models of python like PyTorch, TensorFlow, etc. If any kind of anomaly is detected, the alarm is generated as shown in Figure 21 and the position of the anomaly detected will be highlighted.

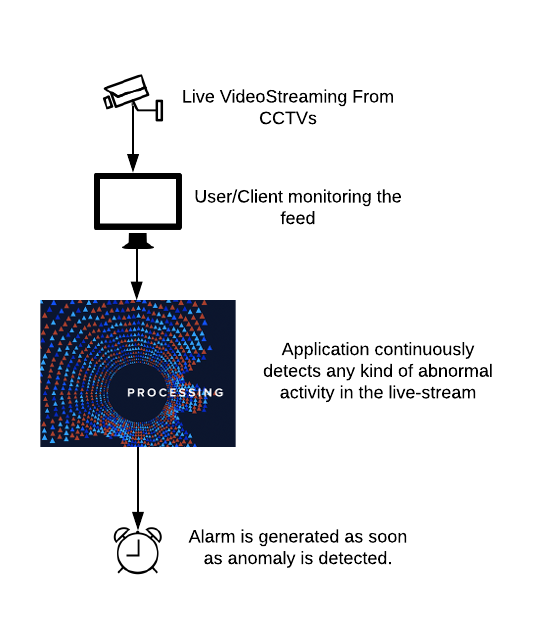


Figure 21: System Architecture

## Network diagram

Table 19 is the Gantt chart, it graphically represents which tasks would be done in which duration, what tasks will be done in parallel, and which tasks will be done in series.

Table 19: Gantt Chart

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ACTIVITIES | Feb 2020 | Mar 2020 | April-May 2020 | June-July 2020 | Aug 2020 | Sept 2020 | Oct 2020 | Nov 2020 | Dec 2020 |
| Proposal submission |  |  |  |  |  |  |  |  |  |
| Project Planning |  |  |  |  |  |  |  |  |  |
| Training the model with the dataset |  |  |  |  |  |  |  |  |  |
| Prototype1 |  |  |  |  |  |  |  |  |  |
| Prototype 2 |  |  |  |  |  |  |  |  |  |
| Finalizing the prototype |  |  |  |  |  |  |  |  |  |
| Creating the application |  |  |  |  |  |  |  |  |  |
| Testing & bug fixing |  |  |  |  |  |  |  |  |  |
| Beta release |  |  |  |  |  |  |  |  |  |

## Collaboration diagram

Figure 22 is the collaboration diagram for Surveilia. It shows flow from the user to the backend processes of the application.

A close up of text on a white background

Description automatically generated

Figure 22: Collaboration Diagram

**Chapter 4**

**System Testing**

# **System Testing**

## Unit Testing

In unit testing, all the components of the application are individually tested. It is the first and most fundamental part of testing. We will utilize the beta analyzers for the issue of this venture. We haven’t performed the unit testing, yet but, for future, the test cases for the unit testing are:

### Test Case-01

This test case Table 20 is for the Functional Requirement FR-01: Login. It explains what users can do and what can go wrong if not tested. This test case aims to log in with the correct login credentials.

Table 20: Test Case-01

|  |  |
| --- | --- |
| **Test case Name** | TC-01 |
| **Application Name** | SURVEILIA |
| **Use case** | Login |
| **Input Summary** | The application has a button for logging in. |
| **Output Summary** | Success:  The user has been logged in. |
| **Pre-Conditions** | Users must enter correct login credentials. |
| **Post-Conditions** | User logs in and dashboard screen appears. |

### Test Case-02

This test case Table 21 is for the Functional Requirement FR-02: Anomaly detection.

Table 21: Test Case-02

|  |  |
| --- | --- |
| **Test case Name** | TC-02 |
| **Application Name** | SURVEILIA |
| **Use Case** | Anomaly Detection |
| **Input Summary** | The live feed coming from the camera. |
| **Output Summary** | Success:  The anomaly is detected, and the video clip is generated. |
| **Pre-Conditions** | Live Feed from camera |
| **Post-Conditions** | The video clip is sent from the database to the user interface. |

### Test Case-03

This test case Table 22 is for the Functional Requirement FR-03: Video Database

Table 22: Test Case-03

|  |  |
| --- | --- |
| **Test case Name** | TC-03 |
| **Application Name** | SURVEILIA |
| **Input Summary** | The live feed coming from the camera. |
| **Output Summary** | Success:  The live feed and all the anomaly detected clips are stored in the database. |
| **Pre-Conditions** | Live Feed from camera |
| **Post-Conditions** | Videos are stored in the database. |

### Test Case-04

This test case Table 23 is for the re-training of the model.

Table 23: Test Case-04

|  |  |
| --- | --- |
| **Test case Name** | TC-04 |
| **Application Name** | SURVEILIA |
| **Input Summary** | The live feed coming from the camera and the trained model from the database. |
| **Output Summary** | Success:  The model is continuously training the model by using the live feed as training examples. |
| **Pre-Conditions** | Trained model from the database and live feed from the camera. |
| **Post-Conditions** | The model is trained, continuously. Hence, it is updated and generates better results. |

## Integration Testing

Integration testing is the second step of the software testing procedure. At this level of testing, the system is tested after combining the distinct units into clusters. It is to test the faults and errors in the interaction between the combined units. We haven’t done the integration testing, yet.

## Acceptance testing

Acceptance testing is defined as the final step for the software testing procedure. It is to govern whether the required specifications of the system are met. At this step, we estimate whether the system under test is complete with the basics and necessities for final processing. We would be doing this step at the end of all the iterations to provide a system with proper specifications.

**Chapter 5**

**Conclusion**

# Conclusion

## Problems faced and lessons learned

### Size of Dataset:

The first problem that we faced was the size of the dataset. So, when training the first Neural Network then one must not move towards using the complete dataset. Instead, the division of the dataset into two or more halves depending on the size, classes of dataset, or system’s computational power, is required. Once, the neural network is trained with a smaller dataset and all the errors are resolved, one can alter that code to go for the complete dataset. This approach worked for us.

### Google Colab Limitations:

The second problem we faced was Google Colab. We failed in reading the complete terms and conditions, FAQs, and guidelines related to it. The main issue we faced while dealing with Google Colab was “I/o error 5” or “Google timeout errors”. Colab has a window of 90minutes timeout of inactivity and 12hrs maximum timeout which passes very quickly when training a model as it is itself a time taking process.

## Project summary

Due to cheaper technology, surveillance cameras are installed everywhere such as streets, shopping malls, hospitals, schools, banks, and even residential areas to observe human activities. The basic purpose of keeping track of human activity is to handle abnormal events that may require the attention of the public or maybe only of authorities. The traditional watch system is human-based so is full of errors, biases, and more importantly exhaustive. In the past, there were a lot of anomalous events which could have been avoided but traditional surveillance system failed to do the job such as Puyallup South Hill Mall burglary [1] and Westfield Century City Mall robbery [2] and many others of similar nature. Pakistan is an underdeveloped country and needs attention in several departments and surveillance system is one of them.

The proposed solution is a complete surveillance system that aims at providing maximum accuracy with minimum false alarms. The process of building the application involves incorporating the deep learning algorithms for activity recognition by taking advantage of a huge amount of data produced by CCTV cameras to provide a fully trained machine. The main purpose of the product is to assist humans in handling surveillance-based issues, and not to replace them completely. This product will be trained by using activity recognition dataset “Something-Something v1” [3] “Something-Something v2” [4] and “UCF Crime” [5] to perform deep video analysis, which detects the anomalies such as burglary, theft, vandalism, etc., in live streaming CCTV and alarms the security instantly. The most important aspect of the proposed product is its usability in every kind of sector i.e. Indoor/Outdoor and Public/Private.

## Future work

Discussed below are the ideas for expansion of the project in the future:

* In the future, this project can be extended by adding more classes of anomaly detection such as detection of camera tampering.
* Another extension would be, providing the user to choose between the classes of the anomaly which he/she wants to detect at the camera instead of all the classes to be detected at the same time. Such as, if the camera is placed in the meeting room; the user may only want to detect any missing object from the room.

**Chapter 6**

**References**

# References

|  |  |
| --- | --- |
| [1] | “KOMO NEWS,” [Online]. Available: https://komonews.com/news/local/security-video-shows-violent-south-hill-mall-burglary. [Accessed 30 July 2020]. |
| [2] | “Mall Robbery,” [Online]. Available: https://patch.com/california/centurycity/century-city-mall-robbery-triggers-panic. [Accessed 30 July 2020]. |
| [3] | “Something-Something V1,” [Online]. Available: https://20bn.com/datasets/something-something/v1. [Accessed 31 July 2020]. |
| [4] | “Something-Something V2,” [Online]. Available: https://20bn.com/datasets/something-something. [Accessed 30 July 2020]. |
| [5] | “UCF Crime Dataset,” [Online]. Available: https://www.crcv.ucf.edu/projects/real-world/. [Accessed 29 July 2020]. |
| [6] | “Advance Intelligent video surveillance system,” [Online]. Available: https://www.intechopen.com/books/intelligent-video-surveillance/advance-intelligent-video-surveillance-system-aivss-a-future-aspect. [Accessed 31 July 2020]. |
| [7] | “Anomaly Localization in Topic-based analysis of Surveillance videos,” [Online]. Available: https://www.cs.cmu.edu/~dpathak/papers/wacv15.pdf. [Accessed 29 july 2020]. |
| [8] | “Video Anomaly Detection in Confined Areas,” [Online]. Available: https://pdf.sciencedirectassets.com/280203/1-s2.0-S1877050917X00148/1-s2.0-S1877050917319294/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEGQaCXVzLWVhc3QtMSJHMEUCIQDqWRiDU9zfneyjU41BFQ8eUwWlBFoZGnA4UCRrwRtZXwIgPq63Vw2oNx5DBjX5NoJLv%2B%2FXED1GRIuvirKGlAz0. [Accessed 30 July 2020]. |
| [9] | “Anomaly Detection,” [Online]. Available: https://www.irjet.net/archives/V7/i1/IRJET-V7I1207.pdf. [Accessed 30 July 2020]. |
| [10] | R. Dev, “Surveillance App,” [Online]. Available: https://play.google.com/store/apps/details?id=com.reservoirdev.surveillance&hl=en. [Accessed 30 July 2020]. |
| [11] | “iCetana,” [Online]. Available: https://icetana.com/. [Accessed 30 July 2020]. |
| [12] | D. Security. [Online]. Available: https://www.dahuasecurity.com/. [Accessed 31 July 2020]. |
| [13] | “Hikvision,” Hikvision, [Online]. Available: https://www.hikvision.com/. [Accessed 27 July 2020]. |
| [14] | “Mobotix,” [Online]. Available: https://www.mobotix.com/en/unique-quality. [Accessed 27 July 2020]. |
| [15] | “Temporal Shift Module,” [Online]. Available: https://github.com/mit-han-lab/temporal-shift-module. [Accessed 30 July 2020]. |

A screenshot of a cell phone

Description automatically generated