

An Automated Crowd Behavior Analysis Tool for Real-Time Surveillance



Session: BSSE Spring 2017-2021

Project Advisor: Dr. Sabeen Javaid

Submitted By

Abdul Wahab Najeeb 17241598-203

Muhammad Ahsan 17241598-202

Muhammad Nabeel Bashir 17241598-006

Hamza Ali Anwar 17241598-091

Department of Software Engineering
University of Gujrat, Sialkot Sub Campus

STATEMENT OF SUBMISSION

This is certifying that this project titled “An Automated Crowd Behavior Analysis Tool for Real-Time Surveillance” by Abdul Wahab Najeeb S/O Najeeb-ur-Rehman Roll No. 17241598-203, Muhammad Ahsan S/O Muhammad Asghar Roll No. 17241598-202, Muhammad Nabeel Bashir S/O Muhammad Bashir Khokhar Roll No. 17241598-006 and Hamza Ali Anwar S/O Muhammad Anwar Roll No. 17241598-091, students of Bachelor of Science (Software Engineering), Department of Software Engineering, University of Gujrat, Sialkot Campus, Pakistan, contains sufficient material required for the award of above said degree.

Project Supervisor

Project Coordination Office
Faculty of C&IT -UOG

Head of the Department

Acknowledgment

We would like to express our special thanks of gratitude to our supervisor Dr. Sabeen Javaid who gave us the golden opportunity to do this wonderful project on the topic (An Automated Crowd Behavior Analysis Tool for Real-Time Surveillance) and helped us throughout the whole course of this Project. And, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame. We are using this opportunity to express our gratitude to everyone who supported us throughout the project and we are thankful for their aspiring guidance, invaluable constructive criticism, and friendly advice during the project work. We are sincerely grateful to them for sharing their truthful and illuminating views on several issues related to the project.

- 1- Abdul Wahab Najeeb
- 2- Muhammad Ahsan
- 3- Muhammad Nabeel Bashir
- 4- Hamza Ali Anwar

Date: _____

Abstract

Countries across the globe have been the victim of terrorist attacks over the past decade specifically. The mosque attacks in New Zealand and school attacks in the US and Pakistan are few examples. Providing automated real-time video surveillance is the need of the hour. It will help to minimize related crimes. The system will analyze the crowd behavior by assessing the video data using various dimensions such as density and motion. It will then alert the concerned authorities if inappropriate behavior is identified. Video surveillance plays a significant role in detecting and identifying the existence or emergence of improper behavior in public and private places. It allows the concerned authorities to observe and monitor the activities of an individual or a group through security cameras. Ultimately, this observation contributes to minimizing or preventing many crimes. The Crowd Analyzer analyzes the behavior of any individual or group gathered anywhere. Using Image Processing and Machine Learning based techniques, it detects inappropriate behavior and alerts the concerned authorities in real-time to prevent unfortunate and undesirable incidents (such as terrorist attacks, and other heinous crimes).

TABLE OF CONTENTS

CHAPTER 1: PROJECT FEASIBILITY REPORT.....	1
1.1. INTRODUCTION	2
1.2. PROJECT/PRODUCT FEASIBILITY REPORT.....	2
1.2.1. <i>Technical Feasibility</i>	2
1.2.2. <i>Operational Feasibility</i>	2
1.2.3. <i>Economic Feasibility</i>	2
1.2.4. <i>Schedule Feasibility</i>	3
1.2.5. <i>Specification Feasibility</i>	3
1.2.6. <i>Information Feasibility</i>	3
1.2.7. <i>Motivational Feasibility</i>	3
1.2.8. <i>Legal & Ethical Feasibility</i>	4
1.3. PROJECT/PRODUCT SCOPE	4
1.4. PROJECT/PRODUCT COSTING	4
1.5. TASK DEPENDENCY TABLE	6
1.6. CPM - CRITICAL PATH METHOD.....	7
1.7. GANTT CHART	9
1.8. INTRODUCTION TO TEAM MEMBERS AND THEIR SKILL SET	9
1.9. TASK AND MEMBER ASSIGNMENT TABLE.....	9
1.10. TOOLS AND TECHNOLOGY WITH REASONING	10
1.11. VISION DOCUMENT	10
1.12. RISK LIST	11
1.13. PRODUCT FEATURES/ PRODUCT DECOMPOSITION	11
CHAPTER 2: SYSTEM REQUIREMENT SPECIFICATION.....	12
2.1 INTRODUCTION:	13
2.2. SYSTEM SPECIFICATIONS:	13
2.2.1. <i>Introduction</i>	13
2.2.2. <i>Existing System</i>	13
2.2.3. <i>Scope of the System</i>	14
2.2.4. <i>Summary of Requirements (Initial Requirements)</i>	14
2.2.5. <i>Identifying External Entities:</i>	14
2.2.6. <i>Capture "shall" Statements:</i>	15
2.2.7. <i>Allocate Requirements:</i>	16
2.2.8. <i>Priorities Requirements:</i>	17
2.2.9. <i>Requirements Traceability Matrix:</i>	20
2.2.10. <i>High-Level Usecase Diagram:</i>	21
2.2.11. <i>Analysis Level Usecase Diagram:</i>	23
2.2.12. <i>Use case Description</i>	24
CHAPTER 3: DESIGN DOCUMENT.....	32
3.1. INTRODUCTION:	33
3.2. DOMAIN MODEL	33
3.4. DESIGN CLASS DIAGRAM.....	33
3.5. ENTITY RELATIONSHIP DIAGRAM	34
3.3. SYSTEM SEQUENCE DIAGRAM	35
CHAPTER 4: USER INTERFACE DESIGN	36
4.1 INTRODUCTION.....	37

4.2. SITE MAPS	37
4.3. STORY BOARDS	38
CHAPTER 5: (SOFTWARE TESTING)	43
5.1 INTRODUCTION:	44
5.2. TEST PLAN.....	44
5.2.1. PURPOSE	44
5.2.2. OUTLINE	44
5.3 TEST DESIGN SPECIFICATION	49
5.4. TEST CASE SPECIFICATION.....	53
5.5. TEST PROCEDURE SPECIFICATION	56
5.6. TEST ITEM TRANSMITTAL REPORT	66
5.7. TEST LOG	67
5.8. TEST INCIDENT REPORT	69
5.9. TEST SUMMARY REPORT.....	70

LIST OF TABLES

Table 1: Basic COCOMO	5
Table 2: Intermediate COCOMO.....	6
Table 3: Task Dependency Table	6
Table 4: Sequence of the Activities	7
Table 5: Estimated Activity Completion Time	8
Table 6: Task and Member Assignment Table	10
Table 7: Shall Statement	15
Table 8: Requirements Allocation	16
Table 9: Prioritize Requirements	17
Table 10: Requirements Traceability Matrix	20
Table 11: Use Case for STAFF_REGISTRATION.....	24
Table 12: Use Case for LOGIN	24
Table 13: Use Case for LOGOUT	24
Table 14: Use Case for EDIT_PROFILE	25
Table 15: Use Case for DISPLAY_CAMERAS	25
Table 16: Use Case for SEARCH_VIDEO	25
Table 17: Use Case for PLAYBACK_VIDEO.....	26
Table 18: Use Case for DELETE_VIDEO	26
Table 19: Use Case for DELETE_MEMBER	27
Table 20: Use Case for SEARCH_MEMBER.....	27
Table 21: Use Case for DELETE_ACCOUNT	27
Table 22: Use Case for REGISTER_NEW_MEMBER	28
Table 23: Use Case for READ_MEMBER_LIST	28
Table 24: Use Case for NOTIFY_AUTHORITIES.....	28
Table 25: Use Case for MANAGE_MEMBER_AUTHORITIES	29
Table 26: Use Case for CHANGE_ALERT_STATUS	29
Table 27: Use Case for UPDATE_MEMBER_PROFILE	29
Table 28: Use Case for APPROVE_NEW_MEMBER_REQUEST	30
Table 29: Use Case for DECLINE_NEW_MEMBER_REQUEST	30
Table 30: Use Case for WRITE_SHORT_REPORT.....	31
Table 31: Items to be Tested.....	45
Table 32: Items not to be Tested.....	46
Table 33: Items to be tested	46
Table 34: Features not to be tested	47
Table 35: Responsibility	48
Table 36: Items to be tested	50
Table 37: Items not to be tested	51
Table 38: Features to be Tested	51
Table 39: Features not to be Tested	52
Table 40: Responsibilities.....	53
Table 41: Test Case Identifier.....	54
Table 42: Items to be tested	55
Table 43: Input Specifications	55
Table 44: Output Specifications.....	55
Table 45: Correct Credentials while Login.....	57
Table 46: Incorrect Credentials while Login	57
Table 47: Empty Fields while Login	57
Table 48: The correct contact while verification	58

Table 49: The Incorrect contact while verification	58
Table 50: Forget Password.....	58
Table 51: Logout.....	59
Table 52: Correct Credentials while Registration.....	59
Table 53: Incorrect Credentials while Registration	60
Table 54: Empty Fields while Registration	60
Table 55: Admin Approve Staff Member Registration Request	61
Table 56: Successfully Register	61
Table 57: Admin Decline Staff Member Registration Request	61
Table 58: Registration Declined	62
Table 59: Generate Alert Automatically on Unusual Behavior.....	62
Table 60: Inform Concerned Authorities	62
Table 61: Change Alert Status back to normal and write a short report.....	63
Table 62: Generate Alert Manually	63
Table 63: Delete Staff Member Account	63
Table 64: Search Staff Member	64
Table 65: Read Staff Member List.....	64
Table 66: Correct Credentials while Registration.....	64
Table 67: Correct Credentials while Registration.....	65
Table 68: Description.....	67
Table 69: Procedure results.....	68
Table 70: Incident description	69
Table 71: Summary Report identifier	70
Table 72: Summary of activities	71

LIST OF FIGURES

Figure 1: Network Diagram	8
Figure 2: Gantt chart	9
Figure 3: Use Case Diagram	22
Figure 4: Analysis Level Usecase Diagram	23
Figure 5: Domain Model	33
Figure 6: Class Diagram	34
Figure 7: Entity Relationship Diagram	34
Figure 8: Sequence Diagram	35
Figure 9: Site Map	37
Figure 10: Home Screen	38
Figure 11: Login Screen	38
Figure 12: Registration Screen	39
Figure 13: Registration Screen	39
Figure 14: Profile Screen	40
Figure 15: Logout Screen	40
Figure 16: Alert Screen	41
Figure 17: Live Streams Screen	41
Figure 18: Contact Screen	42
Figure 19: Services Screen	42

Chapter 1: Project Feasibility Report

1.1. Introduction

An Automated Crowd Behavior Analysis Tool for Real-Time Surveillance analyzes the behavior of an individual or group. It will allow concerned authorities to observe the activities through security cameras. Providing automated real-time video surveillance is the need of the hour. It will help to minimize such terrorist attacks and other heinous crimes. The system will analyze the crowd behavior by assessing the video data using various dimensions such as density and motion. It will then alert the concerned authorities if inappropriate behavior is identified.

1.2. Project/Product Feasibility Report

“An Automated Crowd Behavior Analysis Tool for Real-Time: Web Application” is feasible legitimately and technically as well as economically defensible. We are providing a detailed portrayal of the project, tracked by a set of different feasibility areas. The attainment of our project can be driven by these feasibility traits. This feasibility study is being done to determine possible constructive and destructive consequences of a project before devoting a significant amount of time to it. The sustainability of the project can test through several feasibility zones as follows:

1.2.1. Technical Feasibility

An Automated Crowd Behavior Analysis Tool for Real-Time will be a web application and it will be developed using Pycharm (python), the world's most widespread web platform technology, which gives a superlative platform for building applications for web customers. Pycharm also gives tools for forming applications that look great and take benefit of the hardware competencies accessible on each device. It spontaneously adjusts the current device UI to appear its best. Any person who is using a web platform can avail of our application. Moreover, we owe basic web application development knowledge and skills required to build this application.

1.2.2. Operational Feasibility

We are going to make this application easy to use and fault-prone. We will try our best to produce such a product that users can easily operate without facing any technical fault. For assisting our users, in case they don't understand the objective of the specific functionality of the application, the initial guidelines will also be provided along with the application. In case of any issue, the whole team will be readily available to solve the issue as soon as possible. Assisting our users will be the foremost objective. We consider the team has enough capability of tackling all technical issues encountered by users and a quick response will be provided. But again, it rests on a kind of issue as it occurs.

1.2.3. Economic Feasibility

Cost Estimates

- **Development Cost**

In the first version of our application, there is not very much need for money. We need proper devices such as good systems with higher processing capabilities as well as updated Android devices to develop and test the application. This will cost money to every individual. We can develop it on our own so there is no further cost to pay unless in the case of buying any service from an external organization. External help would cause us at least 0.1 million.

- **Maintenance Cost**

Maintenance is an important task for the progress of the application. Maintenance would cost us 0.2 million annually. In case of external help, it would be raised to double.

Benefits Estimates

- **Tangible Benefits**

Working in a team and can develop the first version of our application is our first and foremost benefit. Self-development will reduce our many financial expenditures and ultimately and indirectly, it will increase our revenue. Moreover, we have decided to use the following procedure for revenue generation for our project in future versions.

1. Education Institute
2. Industrial Sector

- **Intangible Benefits**

Our project is mostly based on intangible benefits, as we are developing it as a platform of awareness for the common individual. It will create consciousness among consumers and they will have our system as their partner on their building. Therefore, eventually, it will our first profit and internal satisfaction as well.

1.2.4. Schedule Feasibility

The project is realistic concerning the schedule. Gantt chart has already been prepared and the time agreed has been divided. Everyone in the team is mindful of chores that are to be executed and their deadlines. If all activities go as scheduled in CPM and Gantt, we are the certain project will be concluded on time. Still, there will be a reevaluation of the amount and scope of work lying ahead at different steps so that the project goes as planned.

1.2.5. Specification Feasibility

The requirement elicitation phase will be given extreme importance. This phase is going to have a lion's share in the accomplishment of the project. This phase will be given high prominence, requirement elicitation will be done persistently. All functional and nonfunctional requirements will be gathered. All the requirements will be wrapped up after seeing them convincing and practicable for the project.

1.2.6. Information Feasibility

We find it a good opportunity to develop an application as a Crowd Analyzer system in the middle of the community that has no idea about Crowd Analyzer. The rate of Criminals Activities has been increasing much faster. There is a need for awareness and knowledge. Therefore, it is a step towards educating common people, not only about the Crowd Analyzer but also reduces criminal activities.

1.2.7. Motivational Feasibility

The whole team is compassionate and corporative. Everyone will corporate with each other in a responsive environment. Tasks will be reported daily and discussions will be done. Complications will be tackled with the mutual corporation. Every fellow will share his work understandings with other fellows.

1.2.8. Legal & Ethical Feasibility

As our project is fully self-based and we are not using the services of any other organization, therefore it has no issue legally or ethically. Even if we would need any service from outsourcing media, we will go through the legal channel and pay the cost if required. Although we have an immense global support internet society.

1.3. Project/Product Scope

The Scope of our system is to provide a web application to different organizations is to reduce or detect heinous crimes. It will be designed to provide a platform that reduces human effort. Multiple Cameras are connected. If running and fighting can take place our system detects and show the status to the user. The user generates an alarm inform to concerned authorities about unusual activities. The user also playback recorded video using the search function. The user also deletes the recorded video.

- Admin and Staff members can log in themselves by entering username and password.
- Staff Members can log in themselves by entering their correct credentials.
- Admin and Staff members can display camera views.
- Admin and Staff Member generate an alert.

1.4. Project/Product Costing

A metric is some measurement we can make of a product or process in the overall development process. Metrics are split into two broad categories:

- Knowledge-oriented metrics: these are oriented to tracking the process to evaluate, predict or monitor some part of the process.
- Achievement-oriented metrics: these are often oriented to measuring some product aspect, often related to some overall measure of the quality of the product.

1.4.1. Project Cost Estimation by Function Point Analysis

A function point analysis is a way of attempting to analyze the complexity and effort required to develop software based on "function points". The idea is to characterize a software application in terms of "function points" and attempt to develop an estimate of the effort required based on the number of function points required. Function points are a measure of the size of computer applications and the projects that build them. The size is measured from a functional or user, point of view. Function Point counts at the end of requirements; analysis, design, code, testing, and implementation can be compared.

Total Project Cost and Total Project Effort are calculated given the average productivity parameter for the system.

The formula are given as follow:

$$\text{Cost / FP} = \text{labor rate / productivity parameter}$$

$$\text{Total Project Cost} = \text{FP est.} * (\text{cost / FP})$$

$$\text{Total Estimated Effort} = \text{FP est.} / \text{productivity parameter}$$

$$\text{Labor rate} = 10,000/\text{month} , \text{Productivity} = \text{Nominal (13)}$$

$$\text{Cost / FP} = 10,000 / 13 = 769.231$$

1.4.2. Project Cost Estimation by using COCOMO'81 (Constructive Cost Model)

Boehm's COCOMO model is one of the most used models commercially. The first version of the model delivered in 1981 and COCOMO II is available now. COCOMO 81 is a model that allows one to estimate the cost, effort, and schedule when planning a new software development activity, according to software development practices that were commonly used in the 1970s through the 1980s. It exists in three forms, each one offering greater detail and accuracy the further along one is in the project planning and design process. Listed by increasing fidelity, these forms are called Basic, Intermediate, and Detailed COCOMO. However, only the Intermediate form has been implemented by USC in a calibrated software tool.

Three levels:

- a) **Basic:** This is used mostly for rough, early estimates.
- b) **Intermediate:** This is the most commonly used version, includes 15 different factors to account for the influence of various project attributes such as personnel capability, use of modern tools, hardware constraints, and so forth.
- c) **Detailed:** Accounts for the influence of the different factors on individual project phases: design, coding/testing, and integration/testing. Detailed COCOMO is not used very often.

Each level includes three software development types:

1. **Organic:** Relatively small software teams develop familiar types of software in an in-house environment. Most of the personnel have experience working with related systems.
2. **Embedded:** The project may require new technology, unfamiliar algorithms, or an innovative new method
3. **Semi-detached:** This is an intermediate stage between organic and embedded types.

a) Basic COCOMO

Table 1 shows the formula of Basic COCOMO:

Table 1: Basic COCOMO

Type	Effort	Schedule
Organic	$PM = 2.4 (KLOC)^{1.05}$	$TD = 2.5(PM)^{0.38}$
Semi-Detached	$PM = 3.0 (KLOC)^{1.12}$	$TD = 2.5(PM)^{0.35}$
Embedded	$PM = 2.4 (KLOC)^{1.20}$	$TD = 2.5(PM)^{0.32}$

PM= person-month (effort)

KLOC= lines of code, in thousands

TD= number of months estimated for software development (duration)

b) Intermediate COCOMO

Table 2 shows the formula of Intermediate COCOMO:

Table 2: Intermediate COCOMO

Type	Effort	Schedule
Organic	PM= 2.4 (KLOC) ^{1.05}	TD= 2.5(PM) ^{0.38}
Semi-Detached	PM= 3.0 (KLOC) ^{1.12}	TD= 2.5(PM) ^{0.35}
Embedded	PM= 2.4 (KLOC) ^{1.20}	TD= 2.5(PM) ^{0.32}

PM= person-month

KLOC= lines of code, in thousands

M.- reflects 15 predictor variables, called cost drivers

The schedule is determined using the Basic COCOMO schedule equations.

$$\text{People Required} = \text{Effort} / \text{Duration}$$

Our project “Crowd Analyzer” has similar specifications and characteristics as the organic level of the COCOMO model.

Effort

$$\text{Effort} = a (\text{KLOC})^b \text{ Person/Month}$$

$$\text{Effort} = 2.4(13)^{1.05}$$

$$\text{Effort} = 35.46$$

Development Time

$$\text{Development Time} = c (\text{Effort})^d \text{ Months}$$

$$\text{Development Time} = 2.5(35.46)^{0.38}$$

$$\text{Development Time} = 9.70$$

Average Staff size

$$\text{Average Staff size} = \text{Effort} / \text{Development Time}$$

$$\text{Average Staff Size} = 35.46 / 9.70$$

$$\text{Average Staff Size} = 3.656$$

Productivity

$$\text{Productivity} = \text{KLOC} / \text{Effort}$$

$$\text{Productivity} = 13 / 35.46$$

$$\text{Productivity} = 0.36$$

1.5. Task Dependency Table

Table 3 shows the Task Dependency.

Table 3: Task Dependency Table

T-ID	Tasks	Dependencies	Duration (Weeks)
T1	Feasibility Study	-	2
T2	Software Requirement Document	T1	4
T3	Design Document	T2	3
T4	Interface Design	T2, T3	3
T5	Database Development	T2, T3	1

T6	Backend Development	T2, T3, T5	5
T7	Application Development	T4, T6	7
T8	Testing	T7	2

1.6. CPM - Critical Path Method

The critical path method (CPM), or critical path analysis (CPA), is used for scheduling a set of project activities. CPM calculates the longest path of planned activities to logical endpoints or the end of the project and the earliest and latest that each activity can start and finish without making the project longer. This process determines which activities are "critical" (longest path) and which can be delayed without making the project longer. A critical path is the longest sequence of activities in a project plan, which must be completed on time for the project to complete on the due date. Activity on the critical path cannot be started until its predecessor activity is complete; if it is delayed for a day, the entire project will be delayed for a day unless the activity following the delayed activity is completed a day earlier.

1. Specify the individual activities.
2. Determine the sequence of those activities.
3. Draw a network diagram.
4. Estimate the completion time for each activity.
5. Identify the critical path (longest path through the network)
6. Update the CPM diagram as the project progresses.

1. Specify the Individual Activities

A critical path is the sequence of project network activities, which add up to the longest overall duration, regardless if that longest duration has a float or not. This determines the shortest time possible to complete the project.

In the "Crowd Analyzer" Web Application the activities are:

- Feasibility Study
- Software Requirement Specification (SRS)
- Software Design Document (SDD)
- Interface Design
- Database
- Backend Development
- Application Development
- Testing

2. Sequence of the Activities

Table 4 shows the Sequence of the Activities:

Table 4: Sequence of the Activities

Activity	Name(Activity)	Predecessor	Weeks
A	Feasibility study	None	2
B	Software Requirement Document	A	4
C	Design Document	B	3
D	Interface Design	B,C	3

E	Database Development	B,C	1
F	Backend Development	B,C,E	5
G	Application Development	D,F	7
H	Testing	G	2

3. Network Diagram

Once the activities and their sequencing have been defined, the CPM diagram can be drawn. The network diagram shows the network of activities and their connections along with other activities. Network Diagram of E-Physio is shown in figure 1:

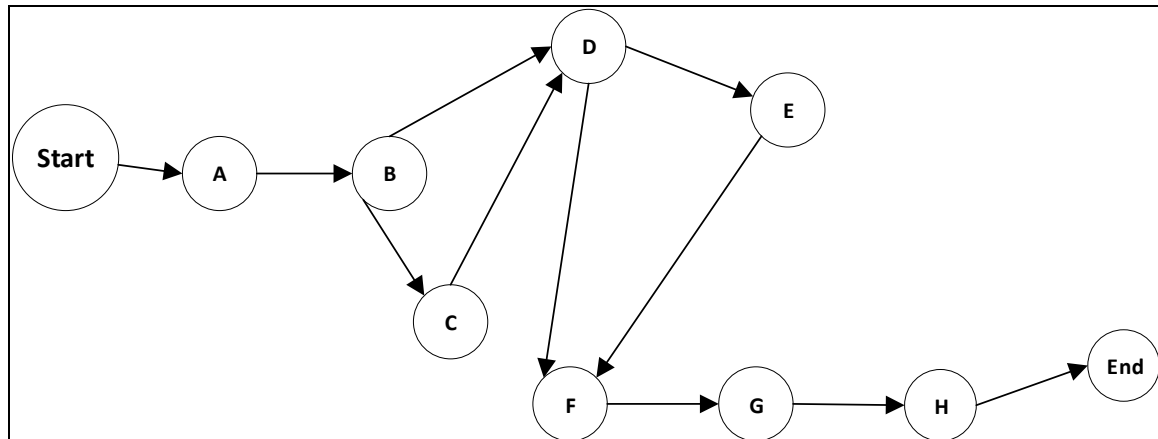


Figure 1: Network Diagram

4. Estimate Activity Completion Time

The time required to complete each activity can be estimated using experience or the estimates of knowledgeable persons. The estimated activity completion time of our project is shown in table 5

Table 5: Estimated Activity Completion Time

Activity	Duration	ES	EF	LS	LF	TS	FS
A	2	0	2	4	6	4	4
B	4	2	5	6	9	4	4
C	3	5	8	6	9	1	1
D	3	5	8	11	14	6	4
E	1	8	9	8	9	0	0
F	5	9	14	9	14	0	0
G	7	14	21	14	21	0	0
H	2	21	23	21	23	0	0

5. Identify the Critical Path

The critical path is the longest-duration path through the network. The significance of the critical path is that the activities that lie on it cannot be delayed without delaying the project. Because of its impact on the entire project, critical path analysis is an important aspect of project planning.

As the critical path is the path with the highest value of duration so:

The critical path of Our Project is:

E→F→G→H=15

1.7. Gantt chart

Here is a brief Gantt chart that describes all tasks is shown in Figure 2:

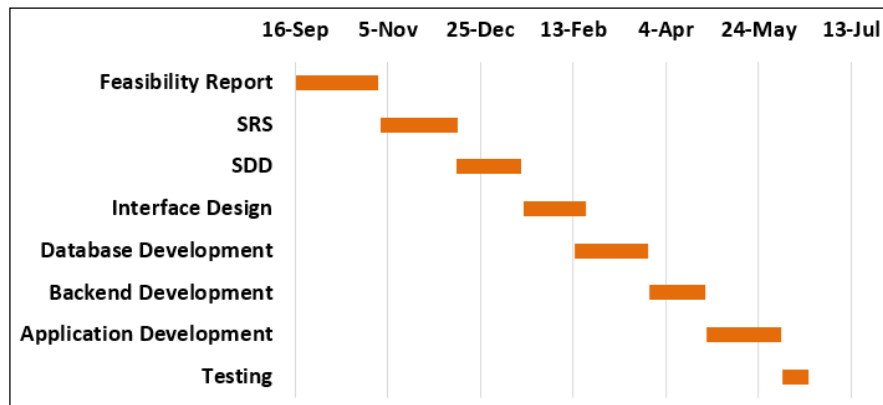


Figure 2: Gantt chart

1.8. Introduction to Team Members and their skill set

a) Abdul Wahab Najeeb

Abdul Wahab Najeeb is performing the following activities in the project:

1. Feasibility Study
2. Design Document
3. Web Application Development
4. Database Handling

b) Muhammad Ahsan

Muhammad Ahsan is performing the following activities in the project:

1. Web Application Development
2. Database Handling
3. Feasibility Study
4. User Interface Design

c) Muhammad Nabeel Bashir

Muhammad Nabeel Bashir is performing the following activities in the project:

1. SRS and Design Document
2. Web Application Development
3. Feasibility Study
4. Test Document

d) Hamza Ali Anwar

Hamza Ali Anwar is performing the following activities in the project:

1. User Interface Design
2. Feasibility Study
3. Test Document

1.9. Task and Member Assignment Table

The task and member assignment table refine the responsibilities of all members concerning their roles in the project. For the ease of traceability characters assigned to every member are starting from M1-M4. Whereas, duration and dependencies are defined to make excellent use of resources are shown in Table 6.

1. Abdul Wahab Najeeb = M1

2. Muhammad Ahsan = M2
3. Muhammad Nabeel Bashir = M3
4. Hamza Ali Anwar = M4

Table 6: Task and Member Assignment Table

Task	Duration (Weeks)	Dependencies	Members
T1	6	None	M1, M2, M3, M4
T2	6	T1	M3
T3	5	T2	M1, M3
T4	5	T2, T3	M2, M4
T5	6	T2,T3	M1, M2
T6	4	T2,T3,T5	M1, M2, M3
T7	6	T4,T6	M2,
T8	2	T7	M3, M4

1.10. Tools and Technology with reasoning

The application tools, which are to be used on the front and back end of the system to be developed are listed below:

Languages:

1. Python

We are using python language because these days most machine learning activities are developed in python. We use Django for web development.

2. HTML

HTML is a markup language, which will be used to design the front end of our web application. Therefore, we will design our interface in HTML language and the pycharm.

3. SQLite

SQLite is an open-source SQL database that stores data to a text file on a device. Django comes in with built-in SQLite database implementation. SQLite will be used for our database creation and database handling for the web. The reason again to select SQLite that it operates well with the web.

Tools:

1. Pycharm

We will use Pycharm to develop and design web applications. It is the most used IDE for python development around the world.

2. MS Visio

We will use MS Visio in our project for documentation purposes to draw different diagrams.

3. Photoshop

We will use photoshop to design creating image compositions. It is the most used for UI design in web development around the world.

4. Google Colab

We will use google collab for the labeling dataset.

1.11. Vision Document

A vision document is an agreement between the person building the software and all the stakeholders. Once the entire stakeholders agree on the feasibility, it is summarized in the vision document. This helps in the requirement and design phase.

Crowd Analyzer would be a system for a common individual, which encloses a web application to analyze crowd behavior. The system will be designed to detect unusual behavior of one person or group of persons and inform the user. The user can generate alerts and inform concerned authorities.

Some agreed features of the system are listed below:

- Log in System for the Users
- Sign up System for the Staff Member
- System detects Fight
- System detects Running
- System informs the user about Fight
- System informs the user about running
- User Generate Alert
- User Change Alert Status

1.12. Risk List

The Risk List is designed to capture the perceived risks to the success of the project. It identifies, in decreasing order of priority, the events that could lead to a significantly negative outcome. It serves as a focal point for project activities and is the basis around which iterations are organized

Following would be risks for our project.

1. Not any Single Development of a similar idea before our app launching
2. We do not have data set to train our model.
3. Without some experts, it is very difficult to design an web application with multiple layouts that can be responsive to any screen.
4. Without heavy machinery, it is very difficult to develop and run our application.
5. Google services may slow down the applications.
6. The online server maybe slows down sometimes.

1.13. Product Features/ Product Decomposition

Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks, or functions the system is required to perform.

1. Login and registration
2. Manage Profile
3. Detect Fight
4. Detect Running
5. Read Registered User
6. Generate Alert
7. Change Alert Status
8. Logout

Chapter 2: System Requirement Specification

2.1 Introduction:

The Software Requirement Specification (SRS) provides a detailed overview of the Requirements of the proposed system (AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE). This document aims to eliminate the ambiguities that may occur. It is also a reference point during software development, implementation, and maintenance.

2.2. System Specifications:

The following clauses illustrate the system specification of the AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE.

2.2.1. Introduction

Video surveillance plays a significant role in detecting and identifying the existence or emergence of improper behavior in public and private places. It allows the concerned authorities to observe and monitor the activities of an individual or a group through security cameras. This observation eventually helps in minimizing or preventing various crimes. Resultantly, safer places can be provided at public and private premises. Video surveillance can be carried out either by humans or automatic systems. Nowadays making the surveillance real-time and automatic is the focus of law enforcement agencies and private organizations as well.

The Crowd Analyzer is such an effort that analyses the behavior of an individual or a group gathered at any place. Using Image Processing and Machine Learning based techniques identifies the inappropriate behavior and alerts the concerned authorities in real-time so that unfortunate and undesirable incidents (such as terrorist attacks, and other heinous crimes) can be avoided.

2.2.2. Existing System

An automated crowd behavior analysis tool for real-time surveillance deals in the following main areas:

- Staff Registration
- Detect and Identify Human Behavior
- Alert Concerned Authorities on Inappropriate Behavior

- **Staff Registration**

It deals with the registration of staff members. Staff registration is verified by the organization Admin from which the member belongs. However unverified members cannot register.

- **Detect and Identify Human Behaviour**

The system should record the crowd through the camera and analyze the crowd shape, density and motion.

- **Alert Concerned Authorities on Inappropriate Behavior**

The system should generate alerts and inform concerned authorities if inappropriate behavior is identified.

2.2.3. Scope of the System

An automated crowd behavior analysis tool for real-time surveillance contains the following features, describe in detail.

Phase I

The initial phase of the following features.

- User Account Management
- Cameras Management
- Video Recording Management

Phase II

In this phase, there are the following features.

- Monitoring
- Status change with feedback

2.2.4. Summary of Requirements (Initial Requirements)

The following requirements act as future analyses of the proposed system.

Cameras Management

The system shall allow the user to manage the camera's view on the display screen. So, the user easily manages how many cameras display he wants to display on one display screen.

Video Recording Management

The system shall allow the user to search recorded video by date and time. The system shall also allow the user to playback or delete the recorded video.

Communication

To reduce the gap between the user and the specific security person, the system has to provide access to make a call or send private messages between them but before sending they have to send and accept the call or message request.

Status Change with Report

The system shall allow a user to change the status if unusual crowd behavior is identified. But the user must write short why he change the status and what happened in-crowd.

2.2.5. Identifying External Entities:

The Identification of the external entities done in two phases:

a) Over Specify Abstract

Based on the Abstract or initial requirement, one might identify the following entities from an automated crowd behavior analysis tool for real-time surveillance.

- Staff Member

- Admin

b) Perform Refinement

After over-specifying the entities, we will now refine them based on our Business logic. Now after refinement, our entities will be:

- Staff Member
- Admin

2.2.6. Capture "shall" Statements:

Identification of “shall” statements, as they would be all functional requirements shown in Table 7.

Table 7: Shall Statement

Para #	Initial Requirements
1.1	The system “Shall” allow the Admin to login.
1.2	The system “Shall” allow the Admin to logout.
1.3	The system “Shall” allow the Admin to edit his account details.
1.4	The system “Shall” allow the Admin to display views camera.
1.5	The system “Shall” allow the Admin to playback video.
1.6	The system “Shall” allow the Admin to search video clip by date.
1.8	The system “Shall” allow the Admin to delete a specific video from the recorded video.
1.9	The system “Shall” allow the Admin to change alert status.
1.10	The system “Shall” allow the Admin to register new Staff Member.
1.11	The system “Shall” allow the Admin to search Staff Member.
1.12	The system “Shall” allow the Admin to delete Staff Member.
1.13	The system “Shall” allow the Admin to display the Staff Member list.
1.14	The system “Shall” allow the Admin to approve new Staff Member request.
1.15	The system “Shall” allow the Admin to decline new Staff Member request.
1.16	The system “Shall” allow the Admin to update the Staff Member profile.
1.17	The system “Shall” allow the Admin to manage Staff Member authorities.
1.18	The system “Shall” allow the Admin to notify concerned authorities.
2.1	The system “Shall” allow the Staff Member to register.
2.2	The system “Shall” allow the Staff Member to login.
2.3	The system “Shall” allow the Staff Member to logout.
2.4	The system “Shall” allow the Staff Member to edit his account details.
2.5	The system “Shall” allow the Staff Member to display the views camera.
2.6	The system “Shall” allow the Staff Member to playback video.
2.7	The system “Shall” allow the Staff Member to search video clip by date.

2.8	The system “Shall” allow the Staff Member to delete his account.
2.9	The system “Shall” allow the Staff Member to change alert status.
2.10	The system “Shall” allow the Staff Member to notify concerned authorities.
3.1	The system “Shall” detect the fight of a person in the crowd.
3.2	The system “Shall” detect the running of a person in the crowd.
3.3	The system “Shall” generate an alarm for the authorities to alert them about unusual behavior.
3.4	The system “Shall” send the location of the unusual behavior to concerned authorities.

2.2.7. Allocate Requirements:

Requirements Allocation shown in Table 8:

Table 8: Requirements Allocation

Para #	Initial Requirements	Use Case Name
1.1	The system “Shall” allow the Admin to login.	UC_LOG_IN
1.2	The system “Shall” allow the Admin to logout.	UC_LOG_OUT
1.3	The system “Shall” allow the Admin to edit his account details.	UC_EDIT_PROFILE
1.4	The system “Shall” allow the Admin to display views camera.	UC_VIEWS_CAMERA
1.5	The system “Shall” allow the Admin to playback video.	UC_PLAYBACK_VIDEO
1.6	The system “Shall” allow the Admin to search video clip by date.	UC_SEARCH_VIDEO
1.7	The system “Shall” allow the Admin to delete the specific video from the recorded video.	UC_DELETE_VIDEO
1.8	The system “Shall” allow the Admin to change alert status.	UC_CHANGE_ALERT_STATUS
1.9	The system “Shall” allow the Admin to register new Staff Member.	UC_REGISTER_MEMBER
1.10	The system “Shall” allow the Admin to search Staff Member.	UC_SEARCH_MEMBER
1.11	The system “Shall” allow the Admin to delete Staff Member.	UC_DELETE_MEMBER
1.12	The system “Shall” allow the Admin to read the Staff Member list.	UC_READ_MEMBER_LIST
1.13	The system “Shall” allow the Admin to approve new Staff Member request.	UC_APPROVE_NEW_MEMBER_REQUEST

1.14	The system “Shall” allow the Admin to decline new Staff Member request.	UC_DECLINE_NEW_MEMBER_REQUEST
1.15	The system “Shall” allow the Admin to update the Staff Member profile.	UC_UPDATE_MEMBER_PROFILE
1.16	The system “Shall” allow the Admin to manage Staff Member authorities.	UC_MANAGE_MEMBER_AUTHORITIES
1.17	The system “Shall” allow the Admin to notify concerned authorities.	UC_NOTIFY_CONCEREND_AUTHORITIES
2.1	The system “Shall” allow the Staff Member to register.	UC_REGISTER
2.2	The system “Shall” allow the Staff Member to login.	UC_LOG_IN
2.3	The system “Shall” allow the Staff Member to logout.	UC_LOG_OUT
2.4	The system “Shall” allow the Staff Member to edit his account details.	UC_EDIT_PROFILE
2.5	The system “Shall” allow the Staff Member to display the views camera.	UC_VIEWS_CAMERA
2.6	The system “Shall” allow the Staff Member to playback video.	UC_PLAYBACK_VIDEO
2.7	The system “Shall” allow the Staff Member to search video clip by date.	UC_SEARCH_VIDEO
2.8	The system “Shall” allow the Staff Member to delete his account.	UC_DELETE_ACCOUNT
2.9	The system “Shall” allow the Staff Member to change alert status.	UC_CHANGE_ALERT_STAT US
2.10	The system “Shall” allow the Staff Member to notify concerned authorities.	UC_NOTIFY_CONCERNED_AUTHORITIES

2.2.8. Priorities Requirements:

Prioritization of Requirements are shown in Table 9:

Table 9: Prioritize Requirements

Para #	Rank	Initial Requirements	Use Case ID	Use Case Name
1.1	Highest	The system “Shall” allow the Admin to login.	UC_2	UC_LOGIN
1.2	Highest	The system “Shall” allow the Admin to logout.	UC_3	UC_LOG_OUT

1.3	Medium	The system “Shall” allow the Admin to edit his account details.	UC_9	UC_EDIT_ACCOUNT
1.4	Highest	The system “Shall” allow the Admin to display views camera.	UC_4	UC_DISPLAY_CAMERA
1.5	Lowest	The system “Shall” allow the Admin to playback video.	UC_10	UC_PLAYBACK_VIDEO
1.6	Medium	The system “Shall” allow the Admin to search video clip by date.	UC_11	UC_SEARCH_VIDEO
1.8	Medium	The system “Shall” allow the Admin to delete a specific video from the recorded video.	UC_7	UC_DELETE_VIDEO
1.9	Highest	The system “Shall” allow the Admin to change alert status.	UC_5	UC_CHANGE_ALERT_STATUS
1.10	Highest	The system “Shall” allow the Admin to register new Staff Member.	UC_9	UC_REGISTER
1.11	Medium	The system “Shall” allow the Admin to search Staff Member.	UC_12	UC_SEARCH_MEMBER
1.12	Medium	The system “Shall” allow the Admin to delete Staff Member.	UC_13	UC_DELETE_MEMBER
1.13	Medium	The system “Shall” allow the Admin to read the	UC_14	UC_READ_MEMBER_LIST

		Staff Member list.		
1.14	Highest	The system “Shall” allow the Admin to approve new Staff Member request.	UC_15	UC_APPROVE_MEMBER_REQUEST
1.15	Highest	The system “Shall” allow the Admin to decline new Staff Member request.	UC_16	UC_DECLINE_MEMBER_REQUEST
1.16	Lowest	The system “Shall” allow the Admin to update the Staff Member profile.	UC_17	UC_UPDATE_MEMBER_PROFILE
1.17	Medium	The system “Shall” allow the Admin to manage Staff Member authorities.	UC_18	UC_MANAGE_MEMBER_AUTHORITIES
1.18	Highest	The system “Shall” allow the Admin to notify concerned authorities.	UC_6	UC_NOTIFY_AUTHORITY
1.19	Lowest	The system “Shall” allow the Staff Member to register.	UC_1	UC_REGISTER
1.20	Medium	The system “Shall” allow the Staff Member to login.	UC_2	UC_LOGIN
1.21	Medium	The system “Shall” allow the Staff Member to logout.	UC_3	UC_LOGOUT
1.22	Medium	The system “Shall” allow the Staff Member to	UC_9	UC_EDIT_ACCOUNT

		edit his account details.		
1.23	Highest	The system “Shall” allow the Staff Member to display the views camera.	UC_4	UC_DISPLAY_CAMERA
1.24	Lowest	The system “Shall” allow the Staff Member to playback video.	UC_10	UC_PLAYBACK_VIDEO
1.25	Lowest	The system “Shall” allow the Staff Member to search video clip by date.	UC_11	UC_SEARCH_VIDEO
1.26	Medium	The system “Shall” allow the Staff Member to delete his account.	UC_13	UC_DELETE_ACCOUNT
1.27	Medium	The system “Shall” allow the Staff Member to change alert status.	UC_5	UC_CHANGE_ALERT_STATUS
1.28	Medium	The system “Shall” allow the Staff Member to notify concerned authorities.	UC_6	UC_NOTIFY_AUTHORITY

2.2.9. Requirements Traceability Matrix:

Requirements Traceability Matrix is shown in Table 10:

Table 10: Requirements Traceability Matrix

Sr#	Para #	System Specification Text	Build	Use Case Name	Category
1	1.1	The system “Shall” allow the Admin to login.	B1	UC_LOGIN	Business

2	1.2	The system “Shall” allow the Admin to logout.	B1	UC_LOGOUT	Business
3	1.3	The system “Shall” allow the Admin to edit his account details.	S1	UC_EDIT_ACCOUNT	System
4	1.4	The system “Shall” allow the Admin to display the camera.	S1	UC_DISPLAY_CAMERA	System
5	1.5	The system “Shall” allow the Admin to playback video.	U1	UC_PLAYBACK_VIDEO	User
6	1.6	The system “Shall” allow the Admin to search video clip by date.	U1	UC_SEARCH_VIDEO	User
8	1.8	The system “Shall” allow the Admin to delete the specific video from recorded video	B1	UC_DELETE_VIDEO	Business
9	1.9	The system “Shall” allow the Admin to change alert status	B1	UC_CHANGE_ALERT_STAT US	Business
10	1.10	The system “Shall” allow the Admin to register new Staff Member.	B1	UC_REGISTER	Business
11	1.11	The system “Shall” allow the Admin to search Staff Member.	B1	UC_SEARCH_MEMBER	Business

2.2.10. High-Level Usecase Diagram:

The high-level use case diagram shows the working of the whole system. It includes actors that are involved in the system. The purpose of this diagram is to demonstrate how objects will interact with AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE and map out the basic functionality of the system. The use case of our purposed system involves two actors.

- Admin
- Staff Member

The high-level Use case diagram describes the overall working of the system as shown in Figure 3:

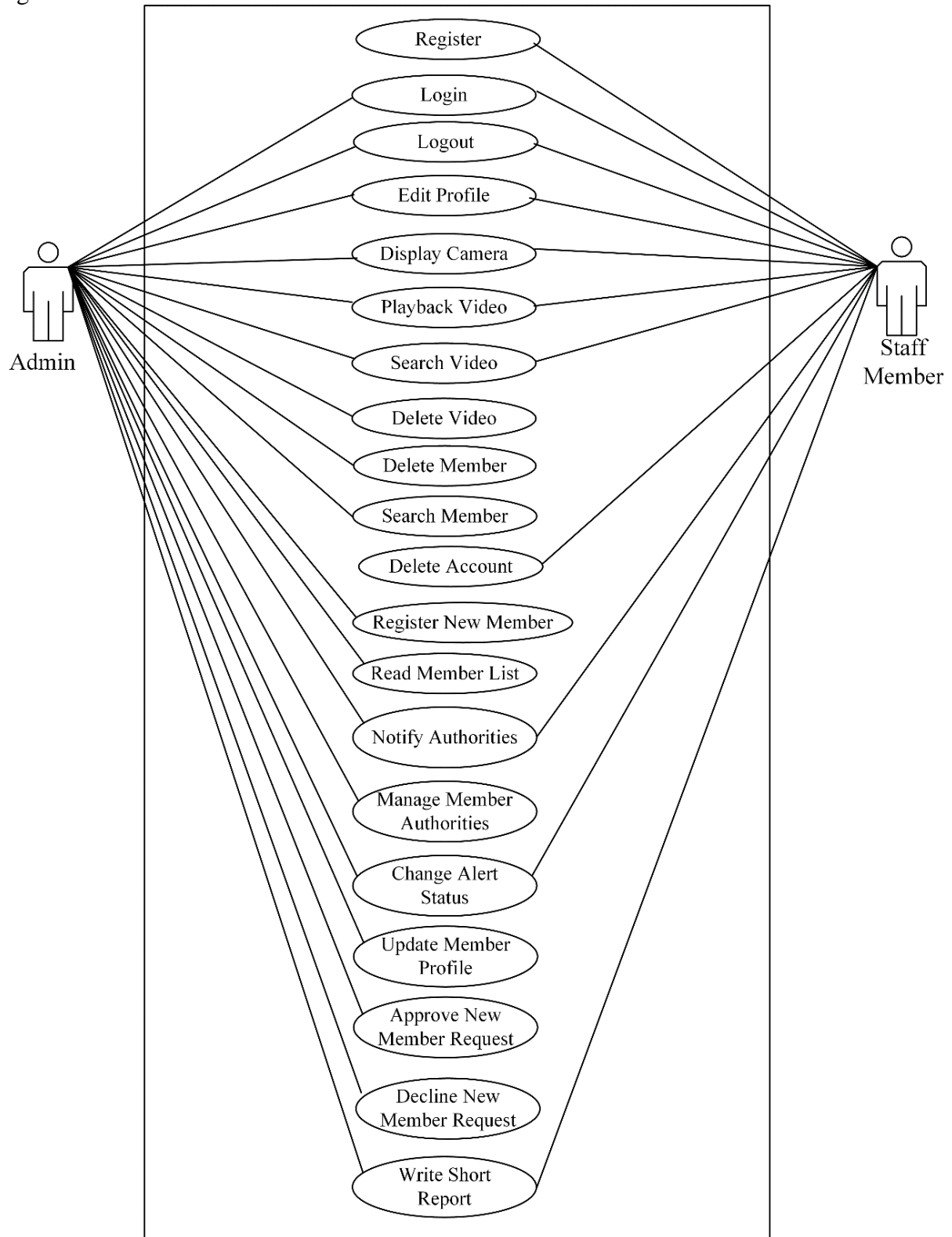


Figure 3: Use Case Diagram

2.2.11. Analysis Level Usecase Diagram:

Analysis Level Usecase Diagram is shown in Figure 4:



Figure 4: Analysis Level Usecase Diagram

2.2.12. Use case Description

Table 11: Use Case for STAFF_REGISTRATION

Use Case ID:	UC_1
Use Case Name:	UC_STAFF_REGISTRATION
Goal/Purpose:	Staff shall register themselves on the system.
Actors:	Staff
Pre-conditions:	Staff shall have installed Chrome Browser.
Post-Conditions:	After registration staff shall have access to login.
Basic Flows:	<ol style="list-style-type: none">1. Staff shall open the browser.2. The login page shall be displayed on the screen.3. It shall have an option for registration.4. Staff shall provide credentials for registration.
Alternative Flows(s):	No
Exception Flows(s):	Display Message to provide credentials again.

Table 12: Use Case for LOGIN

Use Case ID:	UC_2
Use Case Name:	UC_LOG_IN
Goal/Purpose:	Admin/Staff shall login by entering their credentials.
Actors:	Admin/Staff
Pre-conditions:	Admin/Staff should be registered.
Post-Conditions:	<ol style="list-style-type: none">1. Admin/Staff shall login successfully.2. Admin/Staff shall have access to the system.
Basic Flows:	<ol style="list-style-type: none">1. Admin/Staff shall enter information for login.2. Admin/Staff shall click on the login button.
Alternative Flows(s):	No
Exception Flows(s):	Display Message for invalid credentials.

Table 13: Use Case for LOGOUT

Use Case ID:	UC_3
Use Case Name:	UC_LOGOUT
Goal/Purpose:	Admin/Staff shall logout.
Actors:	Admin/Staff
Pre-conditions:	Admin/Staff should be login.
Post-Conditions:	<ol style="list-style-type: none">1. Admin/Staff shall logout successfully.
Basic Flows:	<ol style="list-style-type: none">1. Admin/Staff shall click on the login button.
Alternative Flows(s):	No

Exception Flows(s):	Display Message for not logout.
----------------------------	---------------------------------

Table 14: Use Case for EDIT_PROFILE

Use Case ID:	UC_4
Use Case Name:	UC_EDIT_PROFILE
Goal/Purpose:	Admin/Staff can edit his/her account.
Actors:	Admin/Staff
Pre-conditions:	Admin/Staff should be login.
Post-Conditions:	Admin/Staff shall edit profile.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin/Staff shall go to the profile screen. 2. Admin/Staff shall click to edit account button. 3. Admin/Staff shall edit the account by selecting certain options.
Alternative Flows(s):	No
Exception Flows(s):	Could not edit if any system error.

Table 15: Use Case for DISPLAY_CAMERAS

Use Case ID:	UC_5
Use Case Name:	UC_DISPLAY_CAMERAS
Goal/Purpose:	Admin can manage the camera views on a display screen.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall manage camera views.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to setting. 2. Admin shall click to camera setting. 3. Admin shall change the setting by selecting certain options.
Alternative Flows(s):	No
Exception Flows(s):	Could not edit if any system error.

Table 16: Use Case for SEARCH_VIDEO

Use Case ID:	UC_6
Use Case Name:	UC_SEARCH_VIDEO
Goal/Purpose:	Admin can search the specific recorded video.
Actors:	Admin
Pre-conditions:	Admin should be login.

Post-Conditions:	Admin shall search the specific recorded video.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to setting. 2. Admin shall click to video setting. 3. Admin shall search the specific recorded video.
Alternative Flows(s):	No
Exception Flows(s):	Could not search video if any system error.

Table 17: Use Case for PLAYBACK_VIDEO

Use Case ID:	UC_7
Use Case Name:	UC_PLAYBACK_VIDEO
Goal/Purpose:	Admin can playback the specific video.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall playback the specific video.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to setting. 2. Admin shall click to video setting. 3. Admin shall playback the video.
Alternative Flows(s):	No
Exception Flows(s):	Could not playback the video if any system error.

Table 18: Use Case for DELETE_VIDEO

Use Case ID:	UC_8
Use Case Name:	UC_DELETE_VIDEO
Goal/Purpose:	Admin can delete the specific clip from the recorded video.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall delete the specific clip from the recorded video.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to setting. 2. Admin shall click to video setting. 3. Admin shall delete the specific clip from the recorded video.
Alternative Flows(s):	No
Exception Flows(s):	Could not delete the video if any system error.

Table 19: Use Case for DELETE_MEMBER

Use Case ID:	UC_9
Use Case Name:	UC_DELETE_MEMBER
Goal/Purpose:	Admin can delete the specific member.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall delete the specific member.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to setting. 2. Admin shall click to members. 3. Admin shall delete the specific member.
Alternative Flows(s):	No
Exception Flows(s):	Could not delete member if any system error.

Table 20: Use Case for SEARCH_MEMBER

Use Case ID:	UC_10
Use Case Name:	UC_SEARCH_MEMBER
Goal/Purpose:	Admin can search member by entering the name.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall search member by entering the name.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to setting. 2. Admin shall click to members. 3. Admin shall enter a name to search specific member.
Alternative Flows(s):	No
Exception Flows(s):	Could not search member if any system error.

Table 21: Use Case for DELETE_ACCOUNT

Use Case ID:	UC_11
Use Case Name:	UC_DELETE_ACCOUNT
Goal/Purpose:	Staff Member can delete his account.
Actors:	Staff Member
Pre-conditions:	Staff Member should be login.
Post-Conditions:	Staff Member shall delete his account.
Basic Flows:	<ol style="list-style-type: none"> 1. Staff Member shall go to setting. 2. Staff Member shall delete his account.

Alternative Flows(s):	No
Exception Flows(s):	Could not delete his account if any system error.

Table 22: Use Case for REGISTER_NEW_MEMBER

Use Case ID:	UC_12
Use Case Name:	UC_REGISTER_NEW_MEMBER
Goal/Purpose:	Admin can register new members.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall register new member.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to a staff member. 2. Admin shall click register new member.
Alternative Flows(s):	No
Exception Flows(s):	Could not register a new member if any system error.

Table 23: Use Case for READ_MEMBER_LIST

Use Case ID:	UC_13
Use Case Name:	UC_READ_MEMBER_LIST
Goal/Purpose:	Admin can read the existing members list.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall read the existing member list.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to a staff member. 2. Admin shall click member list.
Alternative Flows(s):	No
Exception Flows(s):	Could not Register read member list if any system error.

Table 24: Use Case for NOTIFY_AUTHORITIES

Use Case ID:	UC_14
Use Case Name:	UC_NOTIFY_AUTHORITIES
Goal/Purpose:	Admin/Staff Member can notify concerned authorities.
Actors:	Admin/Staff Member
Pre-conditions:	Admin/Staff Member should be login.
Post-Conditions:	Admin shall notify concerned authorities.

Basic Flows:	1. Admin shall go to authorities. 2. Admin shall click on notify authorities.
Alternative Flows(s):	No
Exception Flows(s):	Could not notify authorities if any system error.

Table 25: Use Case for MANAGE_MEMBER_AUTHORITIES

Use Case ID:	UC_15
Use Case Name:	UC_MANAGE_MEMBER_AUTHORITIES
Goal/Purpose:	Admin can manage member authorities.
Actors:	Admin
Pre-conditions:	Admin Member should be login.
Post-Conditions:	Admin shall manage member authorities.
Basic Flows:	1. Admin shall go to a staff member. 2. Admin shall click on manage member authorities.
Alternative Flows(s):	No
Exception Flows(s):	Could not manage member authorities if any system error.

Table 26: Use Case for CHANGE_ALERT_STATUS

Use Case ID:	UC_16
Use Case Name:	UC_CHANGE_ALERT_STATUS
Goal/Purpose:	Admin can change the alert system by entering feedback.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall change the alert system by entering feedback.
Basic Flows:	1. Admin shall go to setting. 2. Admin shall click on alert setting. 3. Admin shall change alert status by entering feedback.
Alternative Flows(s):	No
Exception Flows(s):	Could not edit if any system error.

Table 27: Use Case for UPDATE_MEMBER_PROFILE

Use Case ID:	UC_17
Use Case Name:	UC_UPDATE_MEMBER_PROFILE
Goal/Purpose:	Admin can update member profiles.
Actors:	Admin

Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall update member profile.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to a staff member. 2. Admin shall click on the member list. 3. Admin shall click on update.
Alternative Flows(s):	No
Exception Flows(s):	Could not update member profile if any system error.

Table 28: Use Case for APPROVE_NEW_MEMBER_REQUEST

Use Case ID:	UC_18
Use Case Name:	UC_APPROVE_NEW_MEMBER_REQUEST
Goal/Purpose:	Admin can approve new member request.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall approve new member requests.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to a staff member tab. 2. Admin shall click on member request. 3. Admin shall click on approve.
Alternative Flows(s):	No
Exception Flows(s):	Could not approve a request if any system error.

Table 29: Use Case for DECLINE_NEW_MEMBER_REQUEST

Use Case ID:	UC_19
Use Case Name:	UC_DECLINE_NEW_MEMBER_REQUEST
Goal/Purpose:	Admin can decline new member requests.
Actors:	Admin
Pre-conditions:	Admin should be login.
Post-Conditions:	Admin shall decline new member requests.
Basic Flows:	<ol style="list-style-type: none"> 1. Admin shall go to a staff member. 2. Admin shall click on member request. 3. Admin shall click on the decline.
Alternative Flows(s):	No
Exception Flows(s):	Could not decline a request if any system error.

Table 30: Use Case for WRITE_SHORT_REPORT

Use Case ID:	UC_20
Use Case Name:	UC_ WRITE_SHORT_REPORT
Goal/Purpose:	Admin/Staff Member can write a short report after changing alert status.
Actors:	Admin/Staff Member
Pre-conditions:	Admin/Staff Member should be login.
Post-Conditions:	Admin/Staff shall write a short report after changing alert status.
Basic Flows:	<ol style="list-style-type: none">1. The system creates a popup after changing the alert status.2. Admin/Staff Member shall write a report in the text box.3. Admin/Staff Member shall click on submit button.
Alternative Flows(s):	No
Exception Flows(s):	Could not submit a report if any system error.

Chapter 3: Design Document

3.1. Introduction:

A software design document is a written description of a software product, that a software designer writes to give a software development team overall guidance to the architecture of the software project. This document gives a detailed description of the software architecture of AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE system. It specifies the structure of the modules discussed in the SRS.

3.2. Domain Model

Domain Model forms the foundation for the design of the software. In software engineering, a domain model is a conceptual model of the domain that integrates both behavior and data. This model can help to solve domain-related concerns. It is a visual representation of conceptual classes or objects in a domain. Domain Model of AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE is given below in Figure 5.

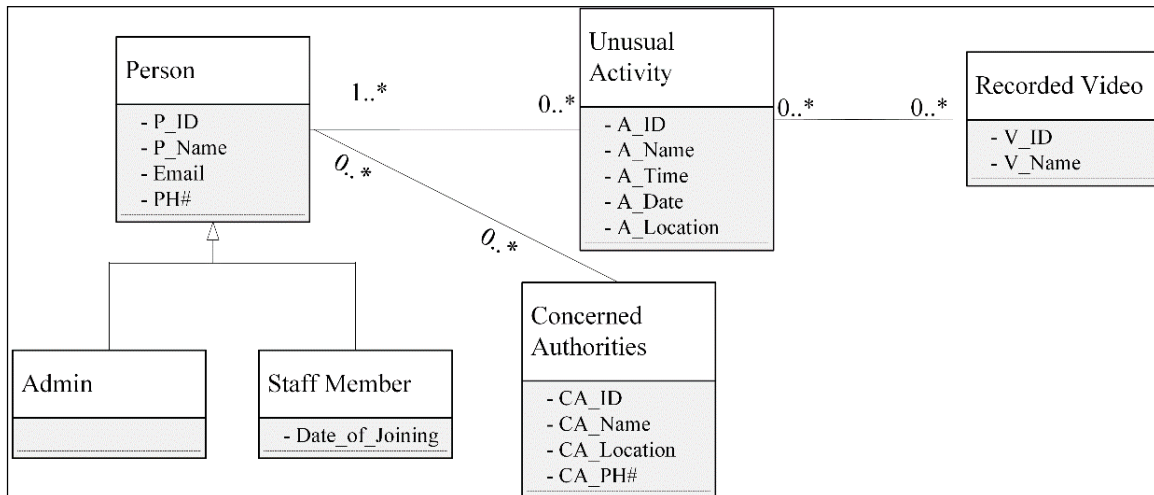


Figure 5: Domain Model

3.4. Design Class Diagram

A class diagram in the (UML) is a type of stagnant structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among those objects. Classes are the work-horses of the design effort—they actually perform the real work of the system. The other design elements—subsystems, packages and collaborations simply describe how classes are grouped or how they interoperate.

The Class Diagram for the “AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE” system is shown in Figure 6:

In a class diagram, classes are symbolized with boxes that contain three partitions:

- The top partition encloses the name of the class.
- The middle partition encompasses the attributes of the class.
- The bottom section holds the operations the class can execute.

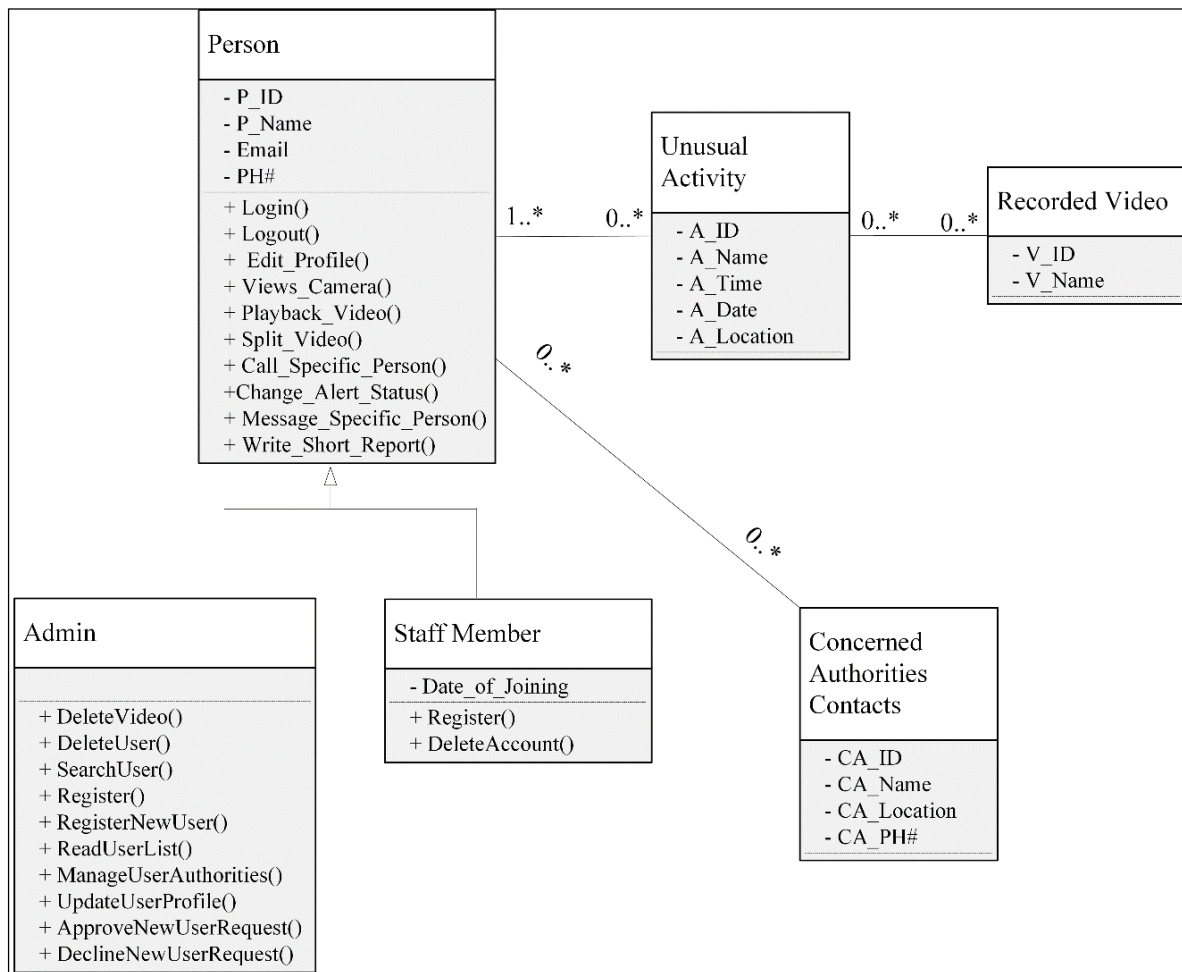


Figure 6: Class Diagram

3.5. Entity Relationship Diagram

The following Figure 7 will show different relationships among entities used in the system:

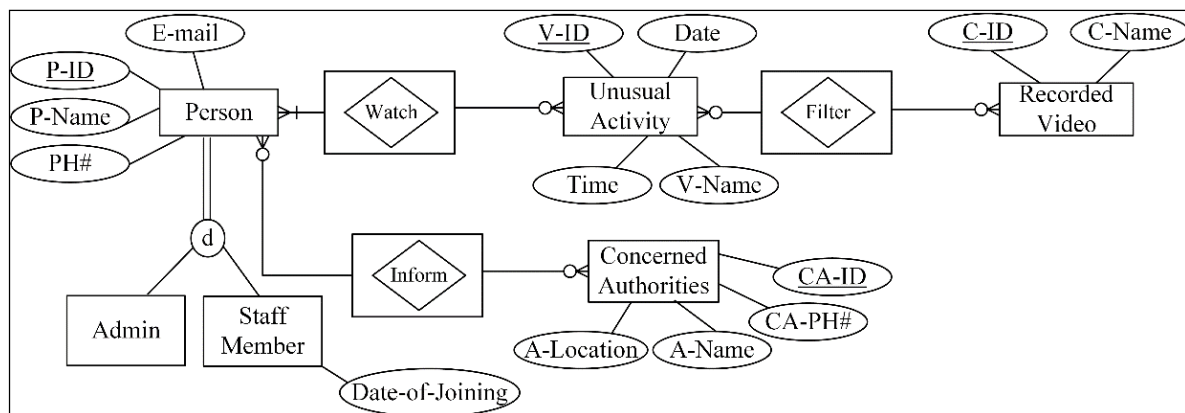


Figure 7: Entity Relationship Diagram

3.3. System Sequence Diagram

The UML system sequence diagram illustrates events sequentially input from an external source to the system. The System Sequence Diagram will define the system events and operations. System sequence diagrams are a timeline drawing of an expanded use case. Events are related by time with the top events occurring first. System events are the important items. These are events that cause a system response. As in the proposed system, there are two actors, Admin and Staff Member. So, we merge both actors into the person. Therefore, there would be the same System Sequence Diagrams for both actors. SSD is shown in Figure 8:

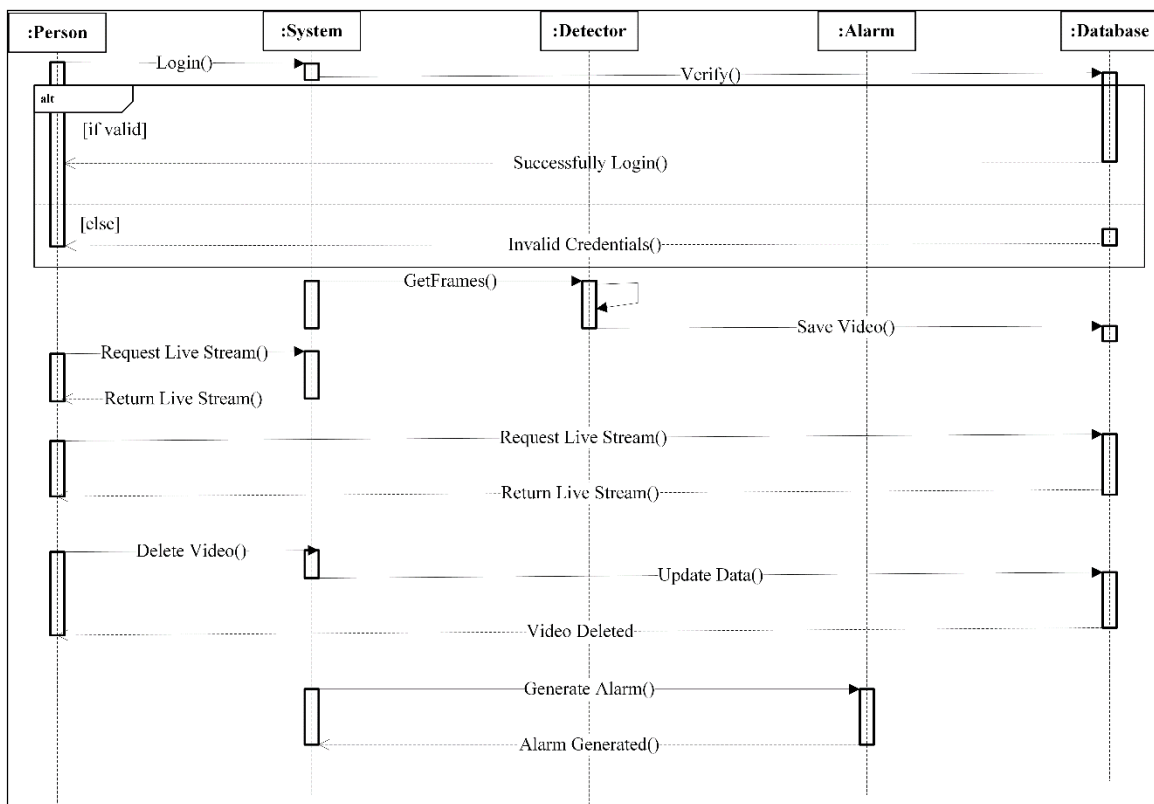


Figure 8: Sequence Diagram

Chapter 4: User Interface Design

4.1 Introduction

User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. User interface design refers to the visual layout of the elements that a user might interact with in a website or technological product. User interface design can affect the usability and user experience of an application.

4.2. Site Maps

A site map's main benefit is to give users an overview of the site's areas in a single glance by dedicating an entire page to a visualization of the information architecture. If designed well, this overview can include several levels of hierarchy, and yet not be so big that users lose their ability to grasp the map. The site map should include

- Home page
- Primary Navigation
- Secondary Navigation
- Tertiary Pages

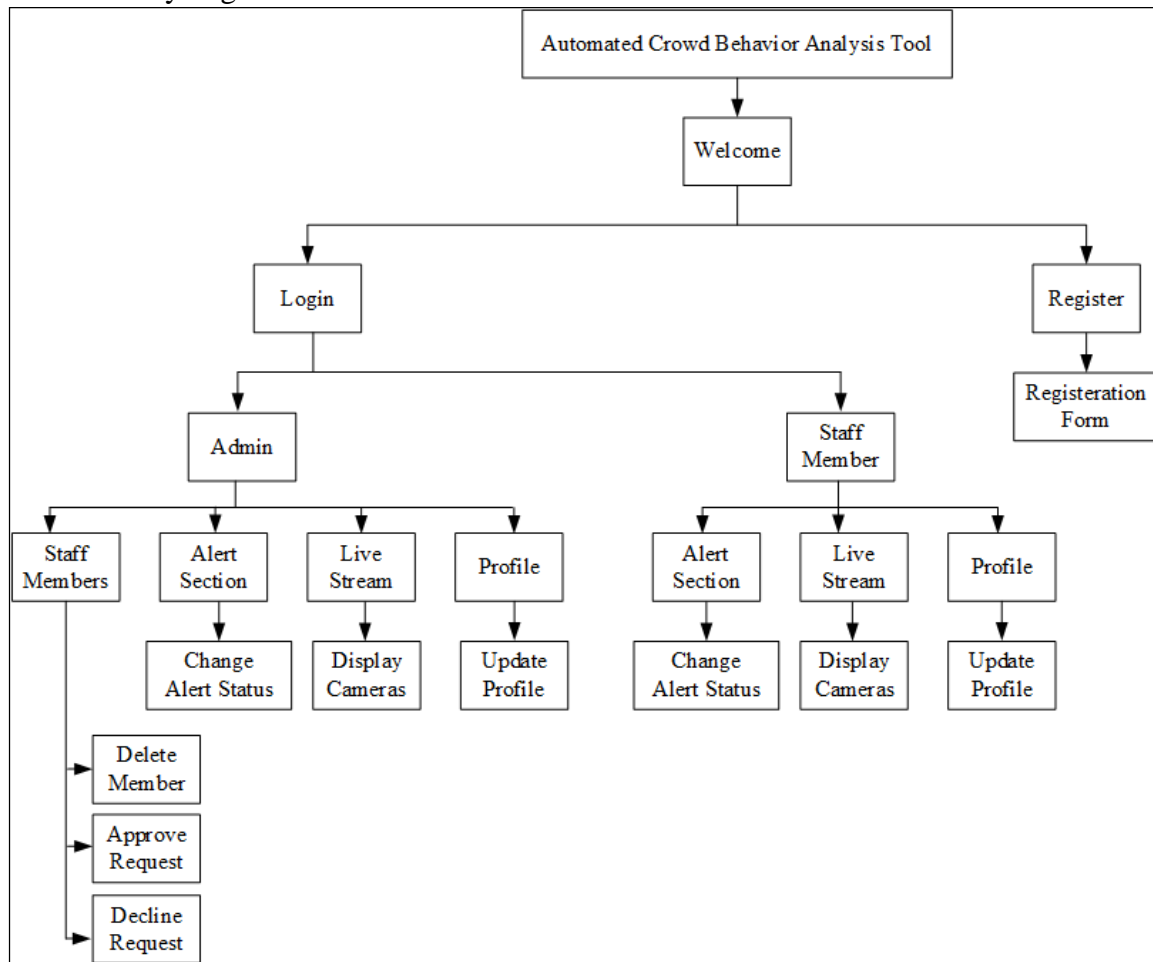


Figure 9: Site Map

4.3. Story boards

A storyboard is a sequence of single images, each of which represents a distinct event or narrative.

UI-1: Welcome Screen

Figure 10 shows the home screen this screen has navigation bar links on it and a decent slider.

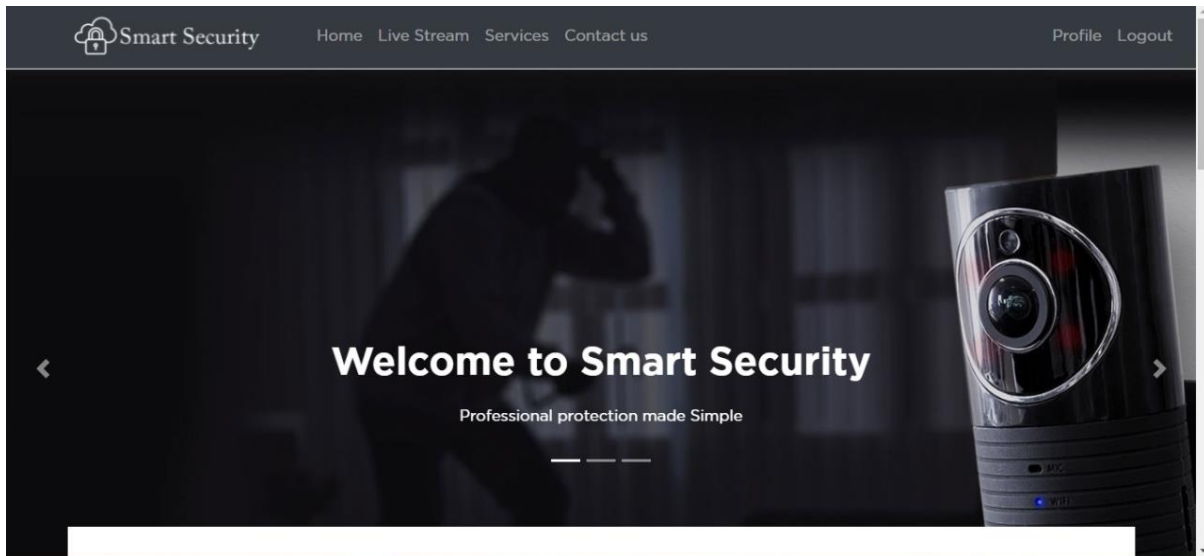


Figure 10: Home Screen

UI-2: Login

Figure 11 shows the Login Screen. This screen will enable the user to enter his/her username and password for successful login.

Figure 11: Login Screen

UI-3: Registration

Figure 12 and Figure 13 show the registration screen. By tapping Register the user will redirect to the registration form.

The user will enter his:

- Username
- E-mail
- Password
- Password Confirmation

If credentials are right the admin will receive the registration request. After the approval from the admin. The user will receive a confirmation message.

So, the user can login by providing the correct username and password.

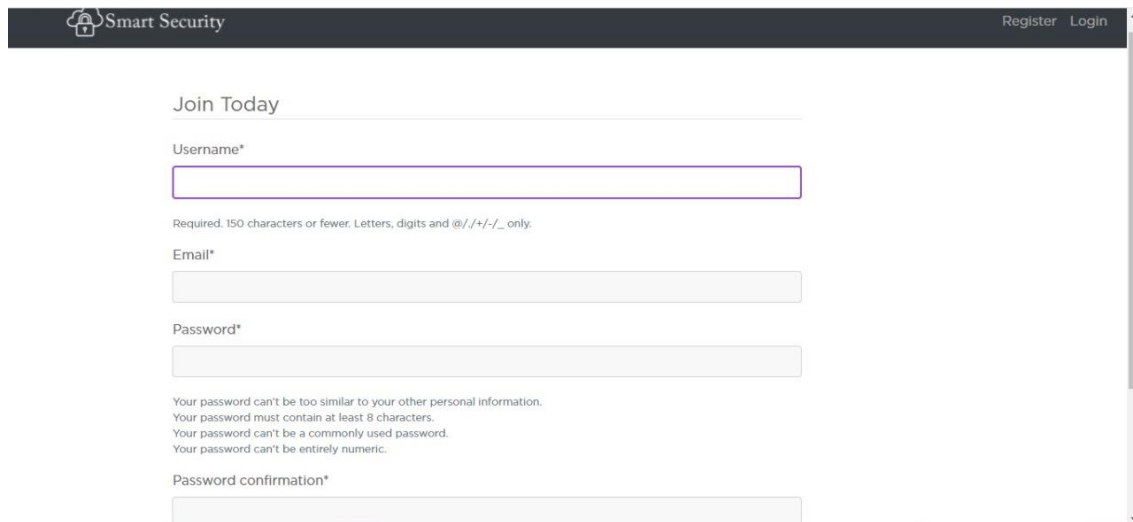
A screenshot of a web application's registration page. The header is dark grey with a logo on the left and 'Register Login' on the right. The main content area is white. It starts with a 'Join Today' heading. Below it are four input fields: 'Username*', 'Email*', 'Password*', and 'Password confirmation*'. The 'Username' field has a purple border. Below the 'Email' field, there is a small text requirement: 'Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.' Below the 'Password' field, there are four lines of password requirements: 'Your password can't be too similar to your other personal information.', 'Your password must contain at least 8 characters.', 'Your password can't be a commonly used password.', and 'Your password can't be entirely numeric.' The 'Password confirmation' field is empty. The page has a vertical scrollbar on the right.

Figure 12: Registration Screen

Users must enter all the required credentials to create his/her account.

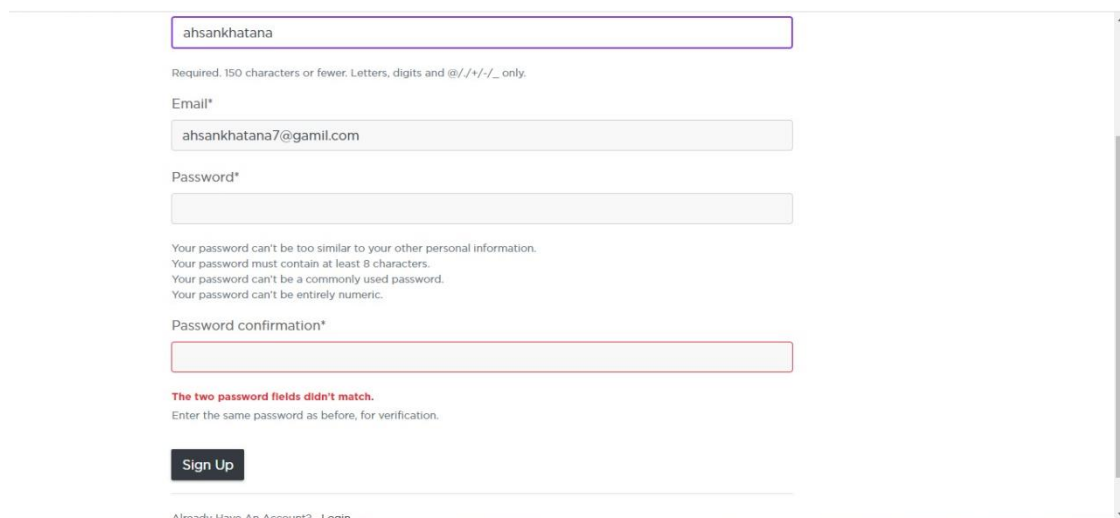
A screenshot of the same registration page as Figure 12, but with some fields filled in. The 'Username' field contains 'ahsankhatana'. The 'Email' field contains 'ahsankhatana7@gamil.com'. The 'Password' field is empty. The 'Password confirmation' field is empty. Below the 'Password' field, there is a red error message: 'The two password fields didn't match.' followed by 'Enter the same password as before, for verification.' At the bottom, there is a 'Sign Up' button. Below the button, there is a link: 'Already Have An Account? Login'. The page has a vertical scrollbar on the right.

Figure 13: Registration Screen

UI-4: Profile

Figure 14 shows the username, profile picture, and email. The user can also update his profile.

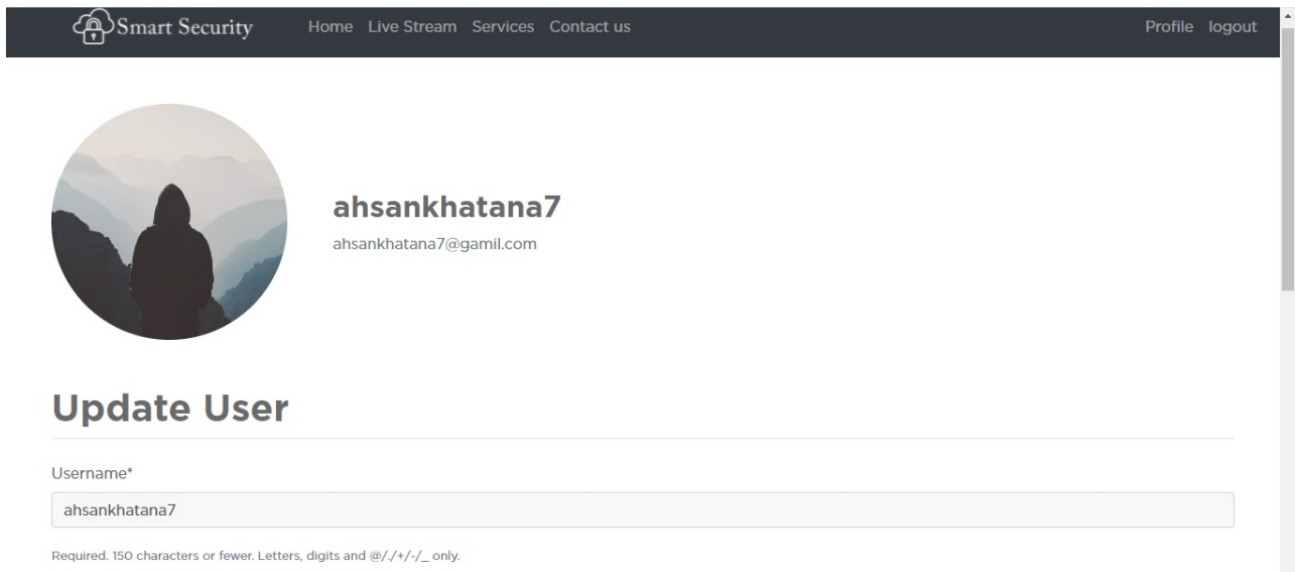


Figure 14: Profile Screen

UI-5: Logout

Figure 15 shows when the user logout from the system this screen will appear.

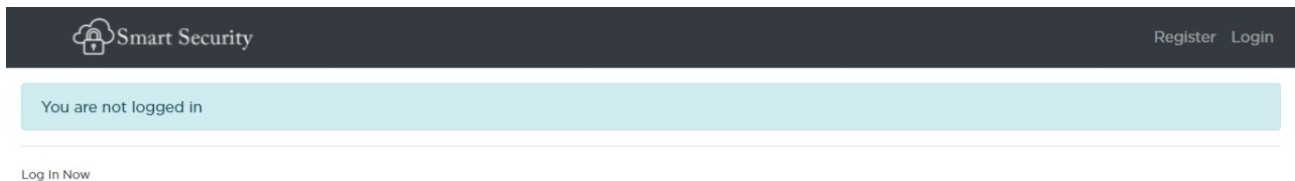


Figure 15: Logout Screen

UI-6: Alert

Figure 16 shows the real-time detection of the system and alert generation button. The user can generate an alert by pressing this button.

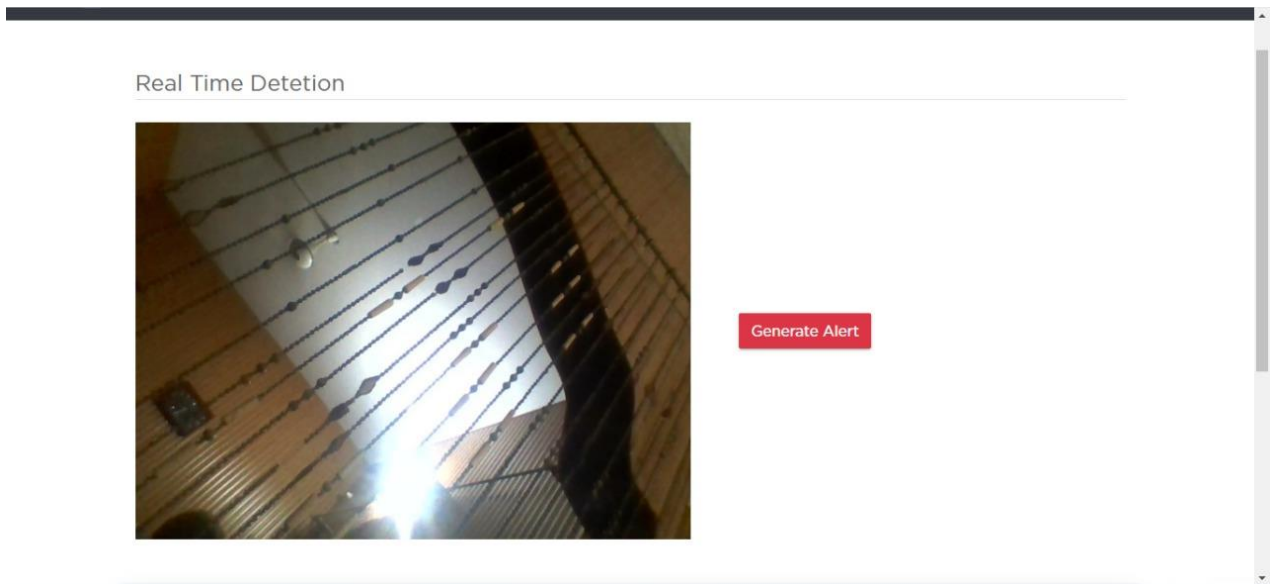


Figure 16: Alert Screen

UI-7: Live Stream

Figure 17 shows the live stream of the video that a system can get through cameras.

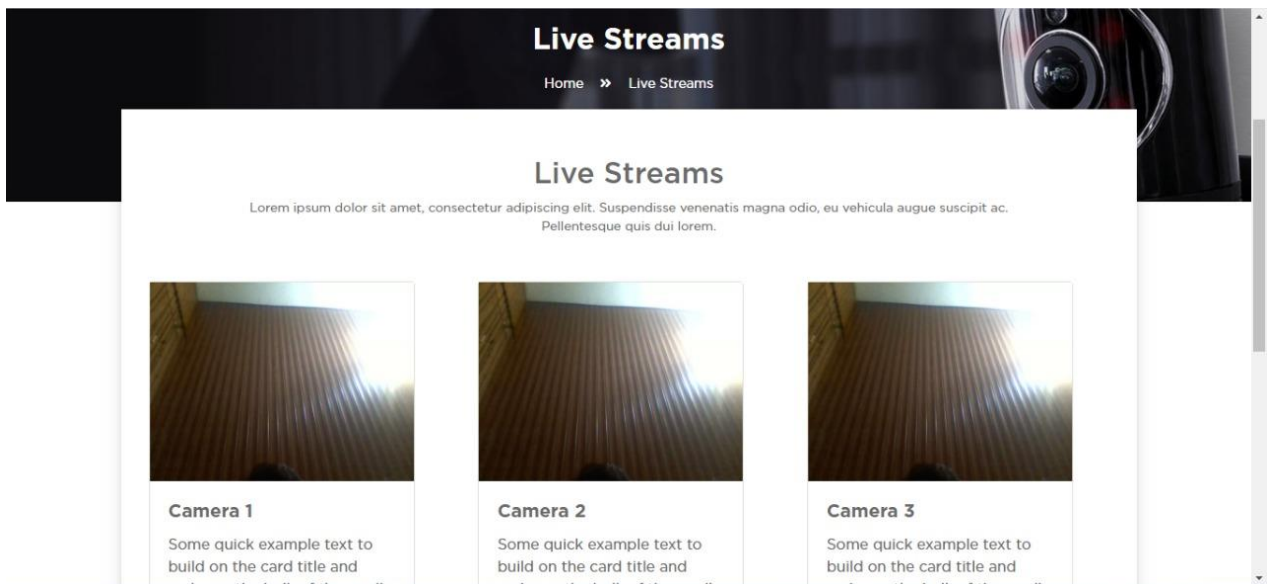


Figure 17: Live Streams Screen

UI-8: Contact

Figure 18 shows the contact screen of the system. The third-party can contact the service provider easily.

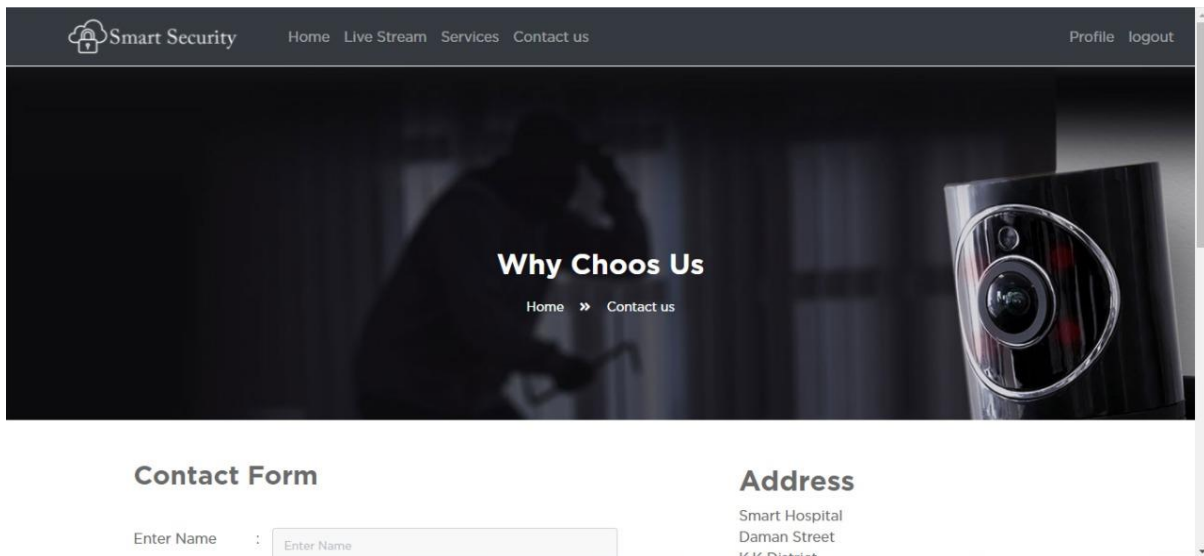


Figure 18: Contact Screen

UI-9: Services

Figure 19 shows the service screen of the system. What type of services a system can provide.

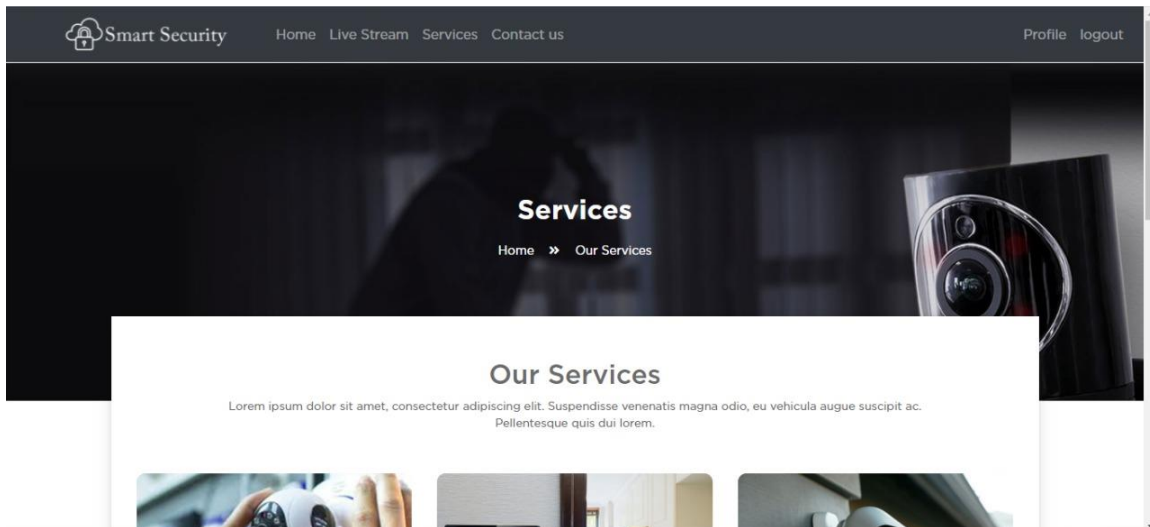


Figure 19: Services Screen

Chapter 5: (Software Testing)

5.1 Introduction:

Testing was carried out following the Software Testing Plan (STP). A software Test Plan will be written to satisfy the requirements. The plan will provide the management and the testing function with an overview of the test activities, schedules, and resources required to perform testing. The plan will describe how the testing specifications will be implemented. Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect-free. Software testing also helps to identify errors, gaps, or missing requirements contrary to the actual requirements. Testing is important because software bugs could be expensive or even dangerous. It makes sure that the system is reliable. This step will determine the quality of the software. Our project is a Web Application, tested on the Web Platform. Hardware resources include our personal computers for compiling and emulating, and for testing on the webserver.

Software tool includes Pycharm.

Following are standard artifacts, which must be included in this deliverable:

1. Test Plan
2. Test Design Specification
3. Test Case Specification
4. Test Procedure Specification
5. Test Item Transmittal Report
6. Test Log
7. Test Incident Report
8. Test Summary Report

5.2. Test plan

The Test Plan describes the scope, approach, resources, and schedule of intended test activities. The test plan will keep track of possible tests that will be performed on the app after coding. It describes the scope of testing, testing techniques to be used, resources required for testing, and the schedule of intended test activities. The scope helps in identifying test items and the features to be tested.

5.2.1. Purpose

The purpose of the test plan is to describe the testing approaches and overall structure of the testing of the application “AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE” We are creating this test plan because it can and will help people outside the team to understand app validations. This plan describes the plans, objectives, scope approach, and focus of the software testing effort.

5.2.2. Outline

A test plan shall have the following structure:

- Test plan identifier
- Introduction

- Test items
- Features to be tested
- Features not to be tested
- Approach
- Item pass/fail criteria
- Suspension criteria and resumption requirements
- Test deliverables
- Testing tasks
- Environmental needs
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies
- Approvals

5.2.2.1. Test plan identifier

As the name suggests, ‘Test Plan Identifier’ uniquely identifies the test plan. Test plan identifier also contains information about the test plan type.

The test plan identifier for our Application is **ACBAT AFTP.1.1**

The abbreviation for HP AFTP.1.1 is

ACBAT	AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL
AFTP	All features test plan
1.1	Version 1 and Revision 1

5.2.2.2. Introduction

A summary of the test plan. We will test all modules of our app. All validation checks will be tested through STLC (Software Testing Life Cycle).

5.2.2.3. Test items

It is a key section for defining the SCOPE of a Test Plan, here we will specify in outline the items, which will be systems and subsystems that are to be tested. It will leave the team with a clear understanding of what will be tested. It includes reports and documentation as well. The test items that we need to test are given in Table 31.

Table 31: Items to be Tested

Login
Registration as Staff Member
Generate Alert
Notify Authorities
Change Alert Status

The items not to be tested are given in Table 32.

Table 32: Items not to be Tested

Number of user login at same time
Third parties libraries
About us

5.2.2.4. Features to be tested

Here we will define which of the features of the app will be tested here we need to identify the functionality that is to be tested in each module or sub-module as these are what deliver value to the team. The features that will be subject to testing can either be a distinct section or a sub-section of the Test Items section shown in Table 33.

Table 33: Items to be tested

Items being Tested	Features being Tested
Login	<ul style="list-style-type: none"> • Correct Credentials while Login. • Incorrect Credentials while Login. • Empty fields while login. • Contact Verification • The incorrect Contact while verification. • Forget Password • Successful logout
Registration as Staff Member	<ul style="list-style-type: none"> • Correct credentials while registration. • Incorrect credentials while registration. • Empty fields while registration. • Admin Approve Registration Request. • Successful Registered.
Generate Alert	<ul style="list-style-type: none"> • Alert Generation on unusual behavior.
Notify Authorities	<ul style="list-style-type: none"> • Inform concerned authorities of specific needs on the specific location.
Change Alert Status	<ul style="list-style-type: none"> • The alert status will be changed back to normal • Generate Alert manually if required.

5.2.2.5. Features not to be tested

The Features not to be tested are shown in Table 34:

Table 34: Features not to be tested

Number of user login at same time
Third parties libraries
About Us

5.2.2.6. Approach

The testing engineer will take over the role of an end-user and will test the app to identify any unexpected behavior or bug. Test Engineer will use test plans, test cases, or test scenarios to test software to ensure the completeness of testing.

The technique we are going to adopt for testing is

- Gray box testing
- Integration Testing
- Unit Testing

Gray box testing is a technique to test the application by having limited knowledge of the internal working of an application. Test Engineer will have the access to documentations and internal working of the system to have a better understanding of the system and to perform testing.

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly.

Developers will perform this type of testing before the setup is handed over to the test Engineer to formally execute the test cases. The goal of **unit testing** is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

5.2.2.7. Item pass/fail criteria

For each test in each category, the operational events should take place in the proper sequence and each of the Application Entities should be left in the appropriate state following the operation. Either pass or fail. If any test case will fail, we will write the code to pass the test and again that test case will be tested.

5.2.2.8. Suspension criteria and resumption requirements

If the number or types of defects reach a point where the follow-on testing has no value, it will make no sense to continue testing. In this case, I will be sending tests for further reviews and development.

In our case we observe suspension

- Database Error:

Resumption Requirements:

Testing of the App will resume from the beginning when the reasons for the suspension of testing have been determined, have been corrected, and new versions of the Application Entities have been submitted to the testing group.

5.2.2.9. Test deliverables

The following documents should be included:

1. Test plan

2. Test design specifications
3. Test case specifications
4. Test procedure specifications
5. Test item transmittal reports
6. Test logs
7. Test incident reports
8. Test summary reports

5.2.2.10. Testing tasks

1. Preparation of the test plan.
2. Preparation of the test design specification.
3. Preparation of test-case specifications.
4. Perform the test procedures.
5. Resolve test incident reports.
6. Repeat tasks (3) - (4) until all test procedures are successful.
7. Prepare the test summary report.

5.2.2.11. Environmental needs

The following elements are required to support the overall testing effort at all levels within the project.

Computer device with the following specifications:

- A high-speed Internet connection
- RAM Minimum of 512 MB, 2 GB is recommended
- 2GB of available hard-disk space for installation extra free space is required during installation.
- Storage Between 850 MB and 1.2 GB

5.2.2.12. Responsibilities

The developing team of “AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE” a Web Application, is responsible for managing, designing, preparing, and executing the test cases. Responsibilities are shown in Table 35:

Table 35: Responsibility

Testing Module	Performed By
Test Planning	Hamza Ali Anwar, Muhammad Nabeel Bashir
Test Specification	Abdul Wahab
Test Case Development	Muhammad Ahsan
Test Writing	Muhammad Nabeel Bashir
Test Execution	Hamza Ali Anwar, Muhammad Nabeel Bashir

5.2.2.13 Staffing and training needs

Testing was performed and executed by Hamza Ali Anwar and Muhammad Nabeel Bashir. The other two members Abdul Wahab and Muhammad Ahsan were also part of the testing

team in one way or another. As testing documents and test plans were done in their presence. Technical assistance was provided by our supervisor, Dr. Sabeen Javaid.

5.2.2.14. Schedule

Testing was conducted on April 29, 2021.

5.2.2.15. Risks and contingencies

- The scope of the plan might be changed.
- The test schedule and development schedule might move out an appropriate number of days.
- Not all stated features might be tested.
- The end-user will reject the system if they found it less interactive and difficult to use after adding a new feature.

5.2.2.16 Approvals

Our supervisor, Dr. Sabeen Javaid, approves this test plan.

Supervisor's Signature

5.3 Test Design Specification

5.3.1. Purpose

Test design specification deals with the designing and predicting of the features, which we are going to test. In this phase, we also prioritize the test cases in their order of importance.

5.3.2. Outline

A test plan shall have the following structure:

- Test plan identifier;
- Introduction;
- Test items;
- Features to be tested;
- Features not to be tested;
- Approach;
- Item pass/fail criteria;
- Suspension criteria and resumption requirements;
- Test deliverables;
- Testing tasks;
- Environmental needs;
- Responsibilities;
- Staffing and training needs;
- Schedule;

- Risks and contingencies;
- Approvals.

5.3.2.1 Test plan identifier

The test Plan identifier for Test design specification is **ACBAT TSD**.

5.3.2.2. Introduction

This test plan describes the testing approach and overall framework that will drive the testing of the “AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE” application. We are creating this test plan because it can and will help people outside the team to understand app validations.

5.3.2.3. Test items

It is a key section for defining the SCOPE of a Test Plan here we will specify in outline the items, which will be systems and subsystems, that are to be tested along with their version numbers. It will leave the team with a clear understanding of what will be tested. It includes reports and documentation as well. The test items that we need to state are given in the Table 36.

Table 36: Items to be tested

Items being Tested	Features being Tested
Login	<ul style="list-style-type: none"> • Correct Credentials while Login. • Incorrect Credentials while Login. • Empty fields while login. • Contact Verification • The incorrect Contact while verification. • Forget Password • Successful logout
Registration as Staff Member	<ul style="list-style-type: none"> • Correct credentials while registration. • Incorrect credentials while registration. • Empty fields while registration. • Admin Approve Registration Request. • Successful Registered.
Generate Alert	<ul style="list-style-type: none"> • Alert Generation on unusual behavior.
Notify Authorities	<ul style="list-style-type: none"> • Inform concerned authorities of specific needs on the specific location.

Change Alert Status	<ul style="list-style-type: none"> • The alert status will be changed back to normal • Generate Alert manually if required.
----------------------------	---

Table 37: Items not to be tested

Number of user login at same time
Third parties libraries
About Us

5.3.2.4. Features to be tested

Table 38: Features to be Tested

Items being Tested	Features being Tested
Login	<ul style="list-style-type: none"> • Correct Credentials while Login. • Incorrect Credentials while Login. • Empty fields while login. • Contact Verification • The incorrect Contact while verification. • Forget Password • Successful logout
Registration as Staff Member	<ul style="list-style-type: none"> • Correct credentials while registration. • Incorrect credentials while registration. • Empty fields while registration. • Admin Approve Registration Request. • Successful Registered.
Generate Alert	<ul style="list-style-type: none"> • Alert Generation on unusual behavior.
Notify Authorities	<ul style="list-style-type: none"> • Inform concerned authorities of specific needs on the specific location.
Change Alert Status	<ul style="list-style-type: none"> • The alert status will be changed back to normal. • Generate Alert manually if required.

5.3.2.5. Features not to be tested

Table 39: Features not to be Tested

Number of user login at same time
Third parties libraries
About Us

5.3.2.6. Approach

We will be using the same strategy and approach as mentioned in the test plan phase.

5.3.2.9. Test deliverables

The following documents should be included:

- Test plan
- Test design specifications
- Test case specifications
- Test procedure specifications
- Test item transmittal reports
- Test logs
- Test incident reports
- Test summary reports

5.3.2.10. Testing tasks

- Preparation of the test plan.
- Preparation of the test design specification.
- Preparation of test-case specifications.
- Perform the test procedures.
- Resolve test incident reports.
- Repeat tasks (3) - (4) until all test procedures are successful.
- Prepare the test summary report.

5.3.2.11. Environmental needs

The following elements are required to support the overall testing effort at all levels within the project.

Computer device with the following specifications:

- A high-speed Internet connection.
- RAM Minimum of 512 MB, 2 GB is recommended.
- 2GB of available hard-disk space for installation extra free space is required during installation.
- Storage between 850 MB and 1.2 GB.

5.3.2.12. Responsibilities

The developing team of “AN AUTOMATED CROWD BEHAVIOR ANALYSIS TOOL FOR REAL-TIME SURVEILLANCE” a Web Application is responsible for managing, designing, preparing, and executing the test cases. Table 40 shows responsibilities:

Table 40: Responsibilities

Testing Module	Performed By
Test Planning	Hamza Ali Anwar, Muhammad Nabeel Bashir
Test Specification	Hamza Ali Anwar
Test Case Development	Muhammad Ahsan, Abdul Wahab
Test Writing	Muhammad Ahsan, Abdul Wahab
Test Execution	Hamza Ali Anwar, Muhammad Nabeel Bashir

5.3.2.13. Staffing and training needs

Testing was performed and executed by Hamza Ali Anwar and Muhammad Nabeel Bashir. The other two members Abdul Wahab and Muhammad Ahsan are also part of the testing team in one way or another. As testing documents and test plans were done in their presence. Our respected supervisor Dr. Sabeen Javaid also provided technical assistance.

5.3.2.14. Schedule

Testing was conducted on April 29, 2020.

5.3.2.15. Risks and contingencies

- The scope of the plan might be changed.
- The test schedule and development schedule might move out an appropriate number of days.
- Not all stated features might be tested.

5.3.2.16. Approvals

Our supervisor, Dr. Sabeen Javaid, approved this test plan

Supervisor's Signature

5.4. Test Case Specification

5.4.1. Purpose

To define a test case identified by a test design specification.

5.4.2. Outline

A test case specification shall have the following structure:

- Test case specification identifier
- Test items
- Input specifications
- Output specifications
- Environmental needs

5.4.2.1. Test case specification identifier

Following are the test case identifiers that will be used in our test. Table 41 shows the Test Case Identifiers.

Table 41: Test Case Identifier

Serial No	Features being Tested	Test Case Identifier
1	Correct Credentials while Login.	TC1
2	Incorrect Credentials while Login.	TC2
3	Empty fields while login.	TC3
4	The correct Contact while verification.	TC4
5	The incorrect Contact while verification.	TC5
6	Forget Password.	TC6
7	Successful logout.	TC7
8	Correct credentials while registration.	TC8
9	Incorrect credentials while registration.	TC9
10	Empty fields while registration.	TC10
11	Admin approves staff member registration request.	TC11
12	Successful Registered	TC12
13	Admin decline staff member registration request.	TC13
14	Registration Declined.	TC14
15	The system generates alert automatically on unusual behavior.	TC15
16	Inform concerned authorities of specific needs on the specific location.	TC16
17	Change alert status back to normal and write a short report.	TC17
18	Generate alert manually if required.	TC18
19	Delete staff member account.	TC19
20	Search staff member.	TC20
21	Read Staff member list.	TC21
22	Edit profile with correct credentials.	TC22
23	Edit profile with wrong credentials.	TC23

5.4.2.2 Test items

Table 42: Items to be tested

Items to be Tested
Login
Registration as Staff Member
Generate Alert
Notify Authorities
Change Alert Status

5.4.2.3. Input specifications

Table 43: Input Specifications

Serial No	Features being Tested	Test Case Identifier
1	Admin approve staff member registration request.	TC1
2	Admin decline staff member registration request.	TC2
3	Successful Registered.	TC3
4	Registration Declined.	TC4
5	Inform concerned authorities for specific need on specific location.	TC5
6	Change alert status back to normal and write short report.	TC6
7	Generate alert manually if required.	TC7
8	Delete staff member account.	TC8
9	Search staff member.	TC9
10	Read Staff member list.	TC10
11	Successful logout.	TC11

5.4.2.4. Output specifications

Table 44: Output Specifications

Serial No	Features being Tested	Test Case Identifier
1	Correct Credentials while Login.	TC1
2	Incorrect Credentials while Login.	TC2
3	Empty fields while login.	TC3
4	The correct Contact while verification.	TC4
5	The incorrect Contact while verification.	TC5
6	Forget Password.	TC6
7	Successful logout.	TC7
8	Correct credentials while registration.	TC8

9	Incorrect credentials while registration.	TC9
10	Empty fields while registration.	TC10
11	Successful Registered.	TC11
12	Edit profile with correct credentials.	TC12
13	Edit profile with wrong credentials.	TC13

5.4.2.5. Environmental needs

5.4.2.5.1. Hardware

Android Supported Mobile Phone or Emulator is required to execute all test cases.

5.4.2.5.2. Software

Following software are required for all test cases to execute successfully.

- (1) Pycharm
- (2) Python
- (3) Operating System
- (4) Internet Browser

5.5. Test procedure specification

5.5.1. Purpose

To specify the steps for executing a set of test cases or, more generally, the steps used to analyze a software item to evaluate a set of features.

5.5.2 Outline

A test procedure specification shall have the following structure:

- Test procedure specification identifier
- Purpose
- Special requirements
- Procedure steps

5.5.2.1. Test procedure specification identifier

The identifier for the test procedure specification is **ACBAT_TPSI**.

5.5.2.2. Purpose

A test case specification is a document that specifies the inputs, predicted results, and set of execution conditions on a test. Test case specification development can begin after the system interface requirements are approved. Here we will describe scenarios that will be tested, how they will be tested. In the test plan, we focused on how the testing for the project will progress, units to be tested and approaches to be used during testing were defined briefly.

5.5.2.3. Special requirements

To proceed with the testing in Web one should know the following techniques:

- (1) Unit testing
- (2) Web testing frameworks

5.5.2.4. Procedure Steps

Table 45: Correct Credentials while Login

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC1
Date:	
Purpose:	To check whether the user successfully login or not.
Pre-Requirement:	The user has a valid username and password.
Test Data:	Username: "nabeel258" Password: "123@abc"
Steps:	Go to the homepage of the system. Enter a valid username. Enter a valid password. Press the Login button.
Expected Result:	The user must be login and access the system.
Annual Result:	The user is login and access the system.
Status:	Pass.

Table 46: Incorrect Credentials while Login

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC2
Date:	
Purpose:	To check the response of the system while entering the incorrect values.
Pre-Requirement:	The system must be running and there would be an internet connection.
Test Data:	Username: "ali258" Password: "123abc"
Steps:	Go to the homepage of the system. Enter a valid username. Enter a valid password. Press the Login button.
Expected Result:	The user must be login and access the system.
Annual Result:	The user gets an error "Invalid Values".
Status:	Fail.

Table 47: Empty Fields while Login

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC3

Date:	
Purpose:	To check the response of the system while entering the no values.
Pre-Requirement:	The system must be running and there would be an internet connection.
Test Data:	Username: “ ” Password: “ ”
Steps:	Go to the homepage of the system. Enter a valid username. Enter a valid password. Press the Login button.
Expected Result:	The user must be login and access the system.
Annual Result:	The user gets an error “Invalid Values”.
Status:	Fail.

Table 48: The correct contact while verification

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC4
Date:	
Purpose:	To verify the user through his contact number.
Pre-Requirement:	The user has entered his contact number.
Test Data:	Contact Number: “+92-306-6633525”
Steps:	Enter contact number in a popup window. Press submit button. Enter in the system.
Expected Result:	The user must be entered into the system.
Annual Result:	The user enters the system.
Status:	Pass.

Table 49: The Incorrect contact while verification

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC5
Date:	
Purpose:	To verify the user through his contact number.
Pre-Requirement:	The user has entered his contact number.
Test Data:	Contact Number: “+92-301-66335”
Steps:	Enter contact number in a popup window. Press submit button. Enter in the system.
Expected Result:	The user must be entered into the system.
Annual Result:	The user gets the error message.
Status:	Fail.

Table 50: Forget Password

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC6

Date:	
Purpose:	To reset the password if the user forgets his password.
Pre-Requirement:	The system must be running and there would be an internet connection user also remember his username.
Test Data:	Username: "nabeel258"
Steps:	Click the link "Forget Password". Enter Correct username. Enter the new password. Re-enter the new password. Press Submit.
Expected Result:	The user must be reset his password.
Annual Result:	The user reset his password.
Status:	Pass.

Table 51: Logout

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC7
Date:	
Purpose:	To logout from the system.
Pre-Requirement:	The user must be login into the system.
Test Data:	
Steps:	Click Logout button
Expected Result:	The user must be logout from the system.
Annual Result:	The user logout from the system.
Status:	Pass.

Table 52: Correct Credentials while Registration

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC8
Date:	
Purpose:	To check whether the user successfully registers or not.
Pre-Requirement:	The user has a valid Name, Contact number, CNIC, username, and password.
Test Data:	Name: "Muhammad Nabeel Bashir" Contact Number: "+92-321-5630930" CNIC: "34603-5825567-1" Username: "nabeel258" Password: "123@abc"
Steps:	Go to the homepage of the system. Press the "Register" link. Enter valid name. Enter valid contact number. Enter valid CNIC. Enter a valid username. Enter a valid password.

	Press Login button.
Expected Result:	The user must be register after admin approval.
Annual Result:	The user register after admin approval.
Status:	Pass.

Table 53: Incorrect Credentials while Registration

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC9
Date:	
Purpose:	To check the response of the system while entering incorrect values.
Pre-Requirement:	The user has a valid Name, Contact number, CNIC, username, and password.
Test Data:	Name: "Muhammad Nabeel Bashir" Contact Number: "+92-321-5630930" CNIC: "34603-5825567-1" Username: "nabeel258" Password: "123@abc"
Steps:	Go to the homepage of the system. Press the "Register" link. Enter valid name. Enter valid contact number. Enter valid CNIC. Enter a valid username. Enter a valid password. Press the Login button.
Expected Result:	The user must be register after admin approval.
Annual Result:	The user gets an error "Invalid values".
Status:	Fail.

Table 54: Empty Fields while Registration

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC10
Date:	
Purpose:	To check whether the user successfully registers or not.
Pre-Requirement:	The user has a valid Name, Contact number, CNIC, username, and password.
Test Data:	Name: "" Contact Number: "" CNIC: "" Username: "" Password: ""
Steps:	Go to the homepage of the system. Press the "Register" link. Enter valid name.

	Enter valid contact number. Enter valid CNIC. Enter a valid username. Enter a valid password. Press the Login button.
Expected Result:	The user must be register after admin approval.
Annual Result:	The user gets an error “Invalid values”.
Status:	Fail.

Table 55: Admin Approve Staff Member Registration Request

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC11
Date:	
Purpose:	To check whether the admin receives staff member registration requests or not.
Pre-Requirement:	The admin must be login into the system.
Test Data:	Registration Request = 1
Steps:	Go to the staff member request page of the system.
Expected Result:	The user must be register after admin approval.
Annual Result:	The user successfully register after admin approval.
Status:	Pass.

Table 56: Successfully Register

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC12
Date:	
Purpose:	To check whether the staff member receives approval message from admin after registration request or not.
Pre-Requirement:	The phone must switch-on of the staff member.
Test Data:	
Steps:	
Expected Result:	The staff member must receive an approval message after admin approval.
Annual Result:	The user successfully receives an approval message after admin approval.
Status:	Pass.

Table 57: Admin Decline Staff Member Registration Request

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC13
Date:	
Purpose:	To check whether the admin receives staff member registration request or not.
Pre-Requirement:	The admin must be login into the system.
Test Data:	Registration Request = 0

Steps:	Go to the staff member request page of the system.
Expected Result:	The user must be register after admin approval.
Annual Result:	The user did not register after the decline registration request by the admin.
Status:	Fail.

Table 58: Registration Declined

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC14
Date:	
Purpose:	To check whether the staff member receive an approval message from the admin after registration request or not.
Pre-Requirement:	The phone must switch-on of the staff member.
Test Data:	
Steps:	
Expected Result:	The staff member must receive an approval message after admin approval.
Annual Result:	The user did not receive an approval message after admin approval.
Status:	Fail.

Table 59: Generate Alert Automatically on Unusual Behavior

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC15
Date:	
Purpose:	To check whether the system generates an alert automatically on unusual behavior or not.
Pre-Requirement:	The system must be running and there would be an internet connection.
Test Data:	Unusual behavior of a person and group of persons.
Steps:	
Expected Result:	The system must be generated an alert automatically on unusual behavior.
Annual Result:	The system generates alert automatically on unusual behavior.
Status:	Pass.

Table 60: Inform Concerned Authorities

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC16
Date:	
Purpose:	To check whether the system informs concerned authorities or not.
Pre-Requirement:	The system must be running and there would be an internet connection and an alert is generated.
Test Data:	Alert is generated on unusual behavior.

Steps:	
Expected Result:	The system must be informed by concerned authorities.
Annual Result:	The system informed concerned authorities.
Status:	Pass.

Table 61: Change Alert Status back to normal and write a short report

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC17
Date:	
Purpose:	To check whether the system change alert status back to normal and get the short report by the user.
Pre-Requirement:	The system must be running and there would be an internet connection and an alert is generated and the user must login to the system.
Test Data:	Alert status = 0 and write a short report.
Steps:	Go to Alert Page. Press the “Normal” Button. Write Short Report in a popup window.
Expected Result:	The system must change the alert status back to normal and get a short report by the user.
Annual Result:	The system changes the alert status back to normal and gets a short report from the user.
Status:	Pass.

Table 62: Generate Alert Manually

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC18
Date:	
Purpose:	To check whether the system generates alerts manually by the user.
Pre-Requirement:	The system must be running and there would be an internet connection and the user must login to the system.
Test Data:	Alert status = 1 by user
Steps:	Go to Alert Page. Press the “Generate Alert” Button.
Expected Result:	The system must be generated an alert manually by the user.
Annual Result:	The system generates alerts manually by the user.
Status:	Pass.

Table 63: Delete Staff Member Account

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC19
Date:	
Purpose:	To check whether the system deletes staff member account by admin.

Pre-Requirement:	The system must be running and there would be an internet connection and the admin must login to the system.
Test Data:	Account Visibility = 0
Steps:	Go to Staff Member Page. Find a specific member you want to remove. Press the “Delete” Button
Expected Result:	The system must delete the staff member account by admin.
Annual Result:	The system deletes staff member account by admin.
Status:	Pass.

Table 64: Search Staff Member

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC20
Date:	
Purpose:	To check whether the system search staff member by admin.
Pre-Requirement:	The system must be running and there would be an internet connection and the admin must log in to the system.
Test Data:	Member Name = “Ahsan”
Steps:	Go to Staff Member Page. Click on the search bar. Write specific member correct name you want to search. Press the “Search” button.
Expected Result:	The system must search staff member account by admin.
Annual Result:	The system search staff member account by admin.
Status:	Pass.

Table 65: Read Staff Member List

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC21
Date:	
Purpose:	To check whether the system show staff member to admin.
Pre-Requirement:	The system must be running and there would be an internet connection and the admin must login to the system.
Test Data:	
Steps:	Go to Staff Member Page.
Expected Result:	The system must show the staff member list to the admin.
Annual Result:	The system shows the staff member list to the admin.
Status:	Pass.

Table 66: Correct Credentials while Registration

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC22
Date:	
Purpose:	To check whether the user successfully updates his profile or not.

Pre-Requirement:	The user has a valid Name, Contact number, CNIC, username, and password, and the user must be login into the system.
Test Data:	Name: "Muhammad Nabeel Bashir" Contact Number: "+92-321-5630930" CNIC: "34603-5825567-1" Username: "nabeel258" Password: "123@abc"
Steps:	Go to the homepage of the system. Press the "Profile" Button. Enter valid name. Enter valid contact number. Enter valid CNIC. Enter a valid username. Enter a valid password. Press the Update button.
Expected Result:	The user's profile must be updated.
Actual Result:	The user's profile is updated.
Status:	Pass.

Table 67: Correct Credentials while Registration

Test Engineer:	Muhammad Nabeel Bashir
Test Case ID:	TC23
Date:	
Purpose:	To check whether the user successfully updates his profile or not.
Pre-Requirement:	The user has a valid Name, Contact number, CNIC, username, and password and the user must be login into the system.
Test Data:	Name: "Muhammad Nabeel Bashir" Contact Number: "+92-321-5630" CNIC: "34603-58255" Username: "na" Password: "12"
Steps:	Go to the homepage of the system. Press the "Profile" Button. Enter valid name. Enter valid contact number. Enter valid CNIC. Enter a valid username. Enter a valid password. Press the Update button.
Expected Result:	The user's profile must be updated.
Actual Result:	The user gets an error "Invalid values".
Status:	Fail.

5.6. Test item transmittal report

5.6.1. Purpose

To identify the test items being transmitted for testing. It includes the person responsible for each item, its physical location, and its status.

5.6.2. Outline

A test item transmittal report shall have the following structure:

- Transmittal report identifier
- Transmitted items
- Location
- Status
- Approvals

5.6.2.1. Transmittal report identifier

The identifier for Transmittal report identifier is **ACBAT_TRI**.

5.6.2.2. Transmitted items

All the items, which are mentioned in the Test Plan, are part of the test transmittal report. We should also provide all those tests, which are failed and again revised and developed for refactoring for further testing.

5.6.2.3. Location

The entire test document including the most important test case specification will be placed in a PDF file and will be transmitted to the test manager and test executor by email. After receiving the email tester will be able to convert test case specifications into written test cases.

5.6.2.4. Status

If there will be any problem in the code after the testing will have been done, a request will be made to the developers to again correct and refactor the code and then again testing will be performed.

5.6.2.5. Approvals

Our supervisor, Dr. Sabeen Javaid approved this test plan.

Supervisor's Signature

5.7. Test log

5.7.2.1. Test log identifier

The identifier for Test Log is **HP_TLI**.

5.7.2.2. Description

Here are all the test items, which are planned to be tested and are transmitted to the test manager.

Table 68: Description

Serial No	Features being Tested	Test Case Identifier
1	Correct Credentials while Login.	TC1
2	Incorrect Credentials while Login.	TC2
3	Empty fields while login.	TC3
4	The correct Contact while verification.	TC4
5	The incorrect Contact while verification.	TC5
6	Forget Password.	TC6
7	Successful logout.	TC7
8	Correct credentials while registration.	TC8
9	Incorrect credentials while registration.	TC9
10	Empty fields while registration.	TC10
11	Admin approves staff member registration request.	TC11
12	Successful Registered	TC12
13	Admin decline staff member registration request.	TC13
14	Registration Declined.	TC14
15	The system generates alerts automatically on unusual behavior.	TC15
16	Inform concerned authorities of specific needs on a specific location.	TC16
17	Change alert status back to normal and write a short report.	TC17
18	Generate alert manually if required.	TC18
19	Delete staff member account.	TC19
20	Search staff member.	TC20
21	Read Staff member list.	TC21
22	Edit profile with correct credentials.	TC22
23	Edit profile with wrong credentials.	TC23

Software Requirements:

Software requirements, which are included as:

1. Pycharm
2. Python

Hardware Requirements:

Hardware requirements are given below:

1. i3 CPU
2. OS.

5.7.2.3. Activity and event entries**5.7.2.3.1. Execution description**

While executing the tests the whole team was present so that all of the team get through an understanding of the testing process.

The tests were executed on **June 17, 2020**, in the Android Studio tool with all the hardware and software specifications already provided.

5.7.2.3.2. Procedure results

Following is the complete result of our test execution based on the pass/fail criteria already mentioned in the planning phase.

Table 69: Procedure results

Serial No	Test Case Result	Test Case	Test Revision	Test Status
1	Correct Credentials while Login.	TC1	1 st	Pass
2	Incorrect Credentials while Login.	TC2	1 st	Fail
3	Empty fields while login.	TC3	1 st	Fail
4	The correct Contact while verification.	TC4	1 st	Pass
5	The incorrect Contact while verification.	TC5	1 st	Fail
6	Forget Password.	TC6	1 st	Pass
7	Successful logout.	TC7	1 st	Pass
8	Correct credentials while registration.	TC8	1 st	Pass
9	Incorrect credentials while registration.	TC9	1 st	Fail
10	Empty fields while registration.	TC10	1 st	Fail
11	Admin approves staff member registration	TC11	1 st	Pass
12	Successful Registered	TC12	1 st	Pass
13	Admin decline staff member registration request.	TC13	1 st	Fail
14	Registration Declined.	TC14	1 st	Fail
15	The system generates alerts automatically on unusual behavior.	TC15	1 st	Pass

16	Inform concerned authorities of specific needs on a specific location.	TC16	1 st	Pass
17	Change alert status back to normal and write a short report.	TC17	1 st	Pass
18	Generate alert manually if required.	TC18	1 st	Pass
19	Delete staff member account.	TC19	1 st	Pass
20	Search staff member.	TC20	1 st	Pass
21	Read Staff member list.	TC21	1 st	Pass
22	Edit profile with correct credentials.	TC22	1 st	Pass
23	Edit profile with wrong credentials.	TC23	1 st	Fail

5.8. Test incident report

5.8.1. Purpose

An incident report can be defined as a written description of an incident observed during testing. To understand better Incident in software testing can be defined as a variation or deviation observed in system behavior from what is expected. It can be a deviation from a functional requirement or the environment setup. The incidents that occur during the testing of our system are explained below.

5.8.2. Outline

A test incident report shall have the following structure:

- Summary
- Incident description
- Impact

5.8.2.2. Summary

When a user logged in and used the application and him exit the application without logging out, he had to log in again to get into the application. By default, his account shouldn't be logged out automatically.

5.8.2.3. Incident description

Table 70: Incident description

Inputs	Expected Result	Actual Result	Anomalies	Date and Time	Procedure Step	Environment	Attempt to Repeats	Testers	Observers
When the user login	The user can get excess	By default, his	Logged out of the	19-Jun-e-	The login module should	No special environment.	2	2	2

to the system and exits the application without logging out.	to the application without login.	account logged out and the user had to login again	account.	2020	be working properly				
--	-----------------------------------	--	----------	------	---------------------	--	--	--	--

5.8.2.4. Impact

This needs to be resolved as soon as possible as the Login module is a major module of the System and its working should be proper without any error.

5.9. Test summary report

5.9.1. Purpose

We conducted all tests and all test cases meet the desired criteria except just one. All test cases were successful and expected outcomes were obtained. We have summarized the results of all the labeled testing activities and evaluated all activities.

5.9.2. Outline

A test summary report shall have the following structure:

- Test summary report identifier
- Summary
- Variances
- Comprehensive assessment
- Summary of results
- Evaluation
- Summary of activities
- Approvals

5.9.2.1. Test summary report identifier

Table 71: Summary Report identifier

Completed By:	Muhammad Nabeel Bashir, Hamza Ali Anwar	Report Date:	5 May 2020
Project ID/Name:	“ AUTOMATED CROWD BEHAVIOR	Testing Name/Event:	Unit Testing Integration Testing Gray Box testing

	ANALYSIS TOOL ” a Web Application		
Total Number of Test Cases	23	Testing Sub-Type	Performance Testing UI Testing: Security Testing

5.9.2.2. Summary

Through testing was performed from test plan to test incident report and all bugs and errors were removed by thorough checking of anomalous behavior shown by the system. Most of the test cases showed desired behavior except for a single test case.

5.9.2.3. Variances

There was no variance from test-plan, test designs, or test procedures. Everything went as projected.

5.9.2.4. Comprehensiveness assessment

In the overall assessment of our testing is we get

- 70% corrected and expected results, and in 30% of cases, we find some contractions.
- Not all modules are testable.
- Testing environment and techniques are already explained in the test plan and testing was conducted following it.
- Anomalous behaviors were checked and all changes were made relative to them.

5.9.2.6. Evaluation

Testing has been done up to user satisfaction criteria. All aspects have been evaluated at their level best to get positive results. All limitations have been eliminated.

5.9.2.7. Summary of activities

Table 72: Summary of activities

Serial No	Test Feature	Test Case Identifier	Status
1	Correct Credentials while Login.	TC1	Succeed
2	Incorrect Credentials while Login.	TC2	Succeed
3	Empty fields while login.	TC3	Succeed
4	The correct Contact while verification.	TC4	Succeed
5	The incorrect Contact while verification.	TC5	Succeed
6	Forget Password.	TC6	Succeed
7	Successful logout.	TC7	Succeed
8	Correct credentials while registration.	TC8	Succeed
9	Incorrect credentials while registration.	TC9	Succeed
10	Empty fields while registration.	TC10	Succeed

11	Admin approves staff member registration request.	TC11	Succeed
12	Successful Registered	TC12	Succeed
13	Admin decline staff member registration request.	TC13	Succeed
14	Registration Declined.	TC14	Succeed
15	The system generates alerts automatically on unusual behavior.	TC15	Succeed
16	Inform concerned authorities of specific needs on a specific location.	TC16	Succeed
17	Change alert status back to normal and write a short report.	TC17	Succeed
18	Generate alert manually if required.	TC18	Succeed
19	Delete staff member account.	TC19	Succeed
20	Search staff members.	TC20	Succeed
21	Read Staff member list.	TC21	Succeed
22	Edit profile with correct credentials.	TC22	Succeed
23	Edit profile with wrong credentials.	TC23	Succeed

5.9.2.8. Approvals

Our supervisor, Dr. Prof. Sabeen Javaid, approved this test report.

Supervisor's Signature