Department of Information Technology

University of Gujrat

SMART FIRE RESCUER

Session: BS-SE Fall 2017-2021

Project Advisor: Dr. Sabeen Javed

Submitted By

Abdul Rehman	**17241598-017**
Muhammad Qasim	**17241598-090**
Ameer Hamza	**17241598-086**
Uzair Jameel	**17241598-084**

Department of Software Engineering

University of Gujrat, Sialkot Sub Campus (PPP)

STATEMENT OF SUBMISSION

This is certify that Abdul Rehman 17241598-017, Muhammad Qasim 17241598-090, Ameer Hamza 17241598-086, Uzair Jameel 17241598-084 has successfully completed the final year project named as Smart Fire Rescuer at the Department of Software Engineering, University of Gujrat, Sialkot Sub Campus (PPP), to fulfill the requirement of the degree of BS **in Software Engineering**.

______________________ ______________________

Project Supervisor Project Coordination Office

Dr. Sabeen Javed
Department of Software Engineering
Faculty of Computing & Information Technology
University of Gujrat, Sialkot sub-campus, Punjab, Pakistan.
Email: sabeen.javaid@uogsialkot.edu.pk

____

Head of the Department
Assistant Professor Sabeen Javaid
Department of Software Engineering
Faculty of Computing & Information Technology
University of Gujrat, Sialkot sub-campus, Punjab, Pakistan.
Email: sabeen.javaid@uogsialkot.edu.pk

Acknowledgment

We truly acknowledge the cooperation and help by our supervisor, Dr. Sabeen Javed, Faculty member of the Department of Software Engineering, University of Gujrat, Sialkot sub-campus (PPP). He has been a constant source of guidance throughout the course of this project. We are also thankful to our friends and families whose silent support led us to complete our project. We are also thankful to our friends and families whose silent support led us to complete our project.

1. Abdul Rehman
2. Muhammad Qasim
3. Ameer Hamza
4. Uzair Jameel

Date: July 11, 2021

Abstract

The early fire detection is very difficult to task and many rescue teams feels pressure to go the exact location where the large fire comes out. Fixed camera and systems cannot detect fire from some meter range therefore there should be some solution to overcome to this situation.

This work shows the effective way to reduce the outcome of fire in different situations using the drone to monitor the location and make the effective and automatic alert predictions about the situations the deep learning model is connect to the drone camera Feed and and make the predictions after the predictions collected it send it to the android application used by the rescue teams to go and visit the locations under the incident happen. It saves human time and effort to search the hotpot areas under fire and detect human seeking help. the sign language deep learning model also increase the accuracy as the human wanted help by showing international help sign in front of drone using hand gestures . The system will work runtime as no need to refresh and reload the screen. The good graphic card and camera also increase the performance of the work.

TABLE OF CONTENTS

Chapter 1: Project Feasibility Report

1.1 Introduction

The Management of disaster is a very big challenging task. During the disaster situation the communication channels may be affected and it is difficult to alert the rescue teams. During a disaster like if there is a fire in a specific place it becomes very difficult to detect that humans are stuck in the fire or not. Heat makes it impossible to go near the fire to check the presence of humans. For this reason, a remote control solution will facilitate the detection of the humans. Additionally, this application is used to detect fire like if an earthquake happens and some roads and buildings are destroyed so due to blockage of roads and other hurdles it is difficult for the team to go there so they use this device to detect humans that are stuck and also the fire and hotspots areas of fire. The device consists of a drone and sensors which are connected to microcontroller, and a supportable battery is needed for Arduino. A wifi module is also needed to communicate with the central system on which the drone can send the message that fire or humans are detected.

1.2. Project/Product Feasibility Report

When a project is started the first matter to establish is to assess the feasibility of a project or product. Feasibility means the extent to which appropriate data and information are readily available or can be obtained with available resources such as staff, expertise, time, and equipment. It is basically used as a measure of how practical or beneficial the development of a software system will be to you (or organization). This activity recurs throughout the life cycle.

There are many types of feasibilities:

- Technical

- Operational
- Economic
- Schedule
- Specification
- Information
- Legal and Ethical

1.2.1. Technical Feasibility

The Smart fire rescuer will be a desktop and android application and it will be developed using Android Studio IDE and other platforms or tools, Android gives tools for forming apps that look great and take benefit of the hardware competencies accessible on each device. The world's most widespread mobile platform technology, which gives a superlative platform for building apps and games for Android customers. It spontaneously adjusts current device UI to appear its best. Any person who is using a smartphone with Android OS Android 6.0.1 (Marshmallow or newer) can avail our app placed on Play Store. Moreover, we owe basic mobile application development knowledge and skills required to build this app.

1.2.2. Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems. We are going to make this app easy to use. We will try our best to produce such a product that users can easily operate without facing any technical fault. 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 with all the technical issues encountered by users and quick response will be provided. But again it rests on kind of issue as it occurs.

1.2.3. Economic Feasibility

Our Desktop application is clearly economically feasible as it provides the facility to save the items which is going to expire and the users can buy the products on a discounted rate that helps poor and needy people or common people to buy high amount products on a discount rate.

1. **Cost Estimates:**

(a) Development Cost

In the first version of our application, we will not have any charges for the development of our app. The reason for this is that we have a Desktop and mobile application developer and a Backend developer in our team.

(b) Maintenance Cost

Maintenance is a very important job for the success of software. In near future for the maintenance purpose, we have to hire a special developer and It can cost up to 0.4

Million annually.

2. **Benefit Estimates:**

There are two types of benefit estimates

(a) Tangible Benefit

Our first and foremost benefit is that we are a working in a team and has the capacity for building initial phase of the project which will reduce a lot of money and this will indirectly increase our revenue. Moreover, we have decided to use the following procedure for revenue generation for our project.

1. Brand Advertisement
2. Google AdSense

(b) Non-Tangible Benefit:

Non-tangible benefits of our project are that it provides complete satisfaction to our customers and users which will lead to an increase in the use of our project and that will eventually increase our profits.

1.2.4. Schedule Feasibility

Schedule Feasibility is one of the most important things in timely completion of the project. For timely completion of our project, we have developed a Gantt chart which can be seen at the specific heading of the Gantt chart. We are working 5-6 hours per day in our project and according to our Gantt chart we will be completing our project 15 days earlier than the evaluation date and after completing the app we will use remaining 15 days for the testing purpose. Our schedule is quite flexible and it is least probable that we will be late in achieving our milestones.

1.2.5. Specification Feasibility

All requirements are completed within time. All requirements are clear. We have acquired requirements for the project by keeping the fact in view the user priorities and their demands and the problem they faced. These requirements can be found in Requirement Elicitation section of the documentation.

We have mentioned the constraints for the project and also constraint and conditions are lucid and can be understood easily. All Requirements are the features that our system must have or a constraint that must be accepted for end users. Specification feasibility also included hardware and software specification required for completion of our project.

1.2.6. Information Feasibility

All given information is reliable and meaningful. No confusion about the requirement. Information quality is very good and true information is given. Hence making our software feasible.

As the scope of online shopping applications in all over the world balloons up, we find it a good opportunity to launch our app at that point so it can facilitate users to understand tour application and how this application will going to be very helpful to everyone.

1.2.7. Motivational Feasibility

Our supervisor motivates our group regularly so that's why the members of our group are fully motivated to perform the daunting task of developing this system. We realize that how big this is going to be and we as a team are the source of motivation for ourselves.

1.2.8. Legal & Ethical Feasibility

This project is exclusively being developed for learning and providing a platform to students, teachers and common people to enhance their communication skills. The services provided by "Smart fire rescuer" are not available on any single platform. This project is feasible legally and ethically.

1.3. Project/Product Scope

The document only covers the requirements specifications for the Smart Fire Rescuer

Table 1: Scope of Project

For	Rescue teams (Govt. or privately operated)
What	Prevent wasting man power on low risk areas with drone movements ensuring rapid actions needed over more affected areas due to natural or any unexpected disasters.
The	Smart fire rescuer (SFR)
Is	Desktop Application Android Mobile Application
That	Helps saving time discovering highly damaged areas having more people stuck due to destruction and alarming fire fighting and rescue teams.

In the 2nd 3rd phase the application will contain more type of products not just Mart items and etc.

1.4. Project/Product Costing

This portion provides total costing of the project. We will make costs evaluation using different formulas estimating all inputs and finding out required output. It allows understanding whether the required capital is available to support the project. At the same time Project Costing serves as a parameter to help ensure that there are sufficient funds to complete the project. By understanding the degrees of cost appraising project forecasts can be improved to a great deal.

1.4.1. Project Cost Estimation by Function Point Analysis

Following are five information domain characteristics are determined and counts are provided in here as

Count of total external Inputs = 4
Count of total external Outputs = 7
Count of total user inquiries = 3
Count of total internal logical files = 1
Count of total external interface files = 1

Table 2: Total Count

Information Domain Value	Low values	Average Values	High values	FP Count
External Input	0*3=0	4*4=16	0*6=0	12
External Output	0*4=0	7*5=20	0*7=0	35
User Inquiries	0*3=0	3*4=16	0*6=0	12
Internal Logical Files	1*7=7	0*10=0	0*15=0	7

External Interface Files	1*5=5	0*7=0	0*10=0	5
Total Count	**-**	**-**	**-**	**71**

Complexity Adjustment Factors:

The value adjustment factor (VAF) is based on 14 general system characteristics (GSC's) that rate the general functionality of the application being counted.

For each of the GSC a rating of 0 to 5 can be provided which represents

- 0=Not present, or no influence
- 1= Incidental influence
- 2=Moderate influence
- 3=Average influence
- 4= Significant influence
- 5=Strong influence throughout

Table 3: Complexity Adjustment factor

S#	Factor	Value (0-5)
1	**Installation Ease**	**5**
2	**Operational Ease**	**5**
3	**Multiple Installation**	**4**
4	**Heavily Used Configuration**	**3**
5	**Transaction Rate**	**4**
6	**On-Line Data Entry**	**0**
7	**End User Efficiency**	**4**
8	**On-Line Update**	**0**
9	**Complex Processing**	**3**

10	**Reusability**	**4**
11	**Data Communication**	**4**
12	**Distributed Data Processing**	**4**
13	**Performance**	**4**
14	**Facilitate Change**	**4**
	Total =	**48**

CAF = [0.65 + 0.01 * ( )]
= [0.65 + 0.01 * (48)]
= 1.13
FP est. = Count Total * CAF
FP est. = 71*1,13
FP est. = 80.23
For Our Project
Average Productivity = 26 FP/PM
Labor Rate = 36,400 RS/Month
Total estimated Effort = FP est. / productivity
= 80.23/ 26 = **3.085** pm
Cost/FP = Labor Rate/productivity
= 36400/26 = **1400** Rs/Fp
=
Total Project Cost = FP est. * (Cost/FP)
= 80.23* (1400)
=**112,322 RS.**

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

In order to Calculate cost estimation using the COCOMO model, we will first have to choose that which mode of the COCOMO model to choose.
There are three Modes of COCOMO model as:

1. Organic
2. Semi-Detached
3. Embedded

According to the below table, our software project has similar specifications and characteristics as an organic mode of the COCOMO model. So in order to evaluate cost estimation, we will be using the parameters of the COCOMO model.

Modes of COCOMO Model

Table 4: Models of COCOMO

Parameters	Organic	Semi-Detached	Embed
Size	2-50 KLOC	50-300 KLOC	300KLOC or above
Team	Small	Medium	Large
Developer Experience	Experienced Dev	Average	Little Experience
Environment	Familiar	Less familiar	Changed
Innovation	Little	Medium	Major Innovation
Deadline	Flexible	Medium	Tight deadline

Parameters of Different Modes

Table 5: Parameters of different Modes

Mode	A	B	C	D
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embed	3.6	1.20	2.5	0,32

Effort:

E = A(KLOC)^B Person/Month
E = 2.4(4) ^1.05
E = Person/Month

Development Time:

Dev Time = C(Effort)^D Months
Dev Time = 2.5(10.289) ^0.38 Months
Dev Time = 6.062

Average Staff Size = Effort/ Dev Time
Average Staff Size = 10.289 / 6.062
Average Staff Size = 1.697

Productivity = KLOC/Effort
Productivity = 4/10.289
Productivity = 0.388

1.5. Task Dependency Table

Task Dependency is a relationship in which a task or milestone relies on other tasks to be performed (completely or partially) before it can be performed. This is also referred to as a logical relationship. A logical relationship can be a dependency between project tasks or between tasks and milestones.

Table 6: Task Dependency

Task	Duration (Weeks)	Dependencies
T1	2	None
T2	2	A (Feasibility)
T3	2	B (Requirement Specification)
T4	2	B, C(Requirement Specification, Design Document)
T5	6	C (Design Document)
T6	10	D, E (Interface Design, Data Gathering)
T7	6	F (App Development)
T8	2	G(Implementation)

1.6. CPM - Critical Path Method

1. Specify the Individual Activities

In Smart fire rescuer we have divided our tasks broadly in following activities. These activities are performed on weekly bases and their Gantt chart is also developed along with their Staff allocation

(1) Feasibility
(2) Requirement Specification
(3) Design Document
(4) Interface Design
(5) Data Gathering
(6) App Development
(7) Implementation

(8) Testing

2. Determine the Sequence of the Activities

Sequence Activities is the process of identifying and documenting relationships among the project activities.

These are following activities and duration of activities are following that are shown in table 7.

Table 7: Sequence of the Activities

Activity	Name (Activity)	Immediate Predecessor	Duration (Weeks)
A	Feasibility	None	2
B	Requirement Specification	A	2
C	Design Document	B	2
D	Interface Design	B, C	2
E	Data Gathering	C	6
F	App Development	D, E	10
G	Implementation	F	6
H	Testing	G	2

3. Draw the Network Diagram

A Network Diagram is a graphical way to view tasks, dependencies, and the critical path of your project. Boxes (or nodes) represent tasks, and dependencies show up as lines that connect those boxes. Network Diagram of our project is shown in figure 1

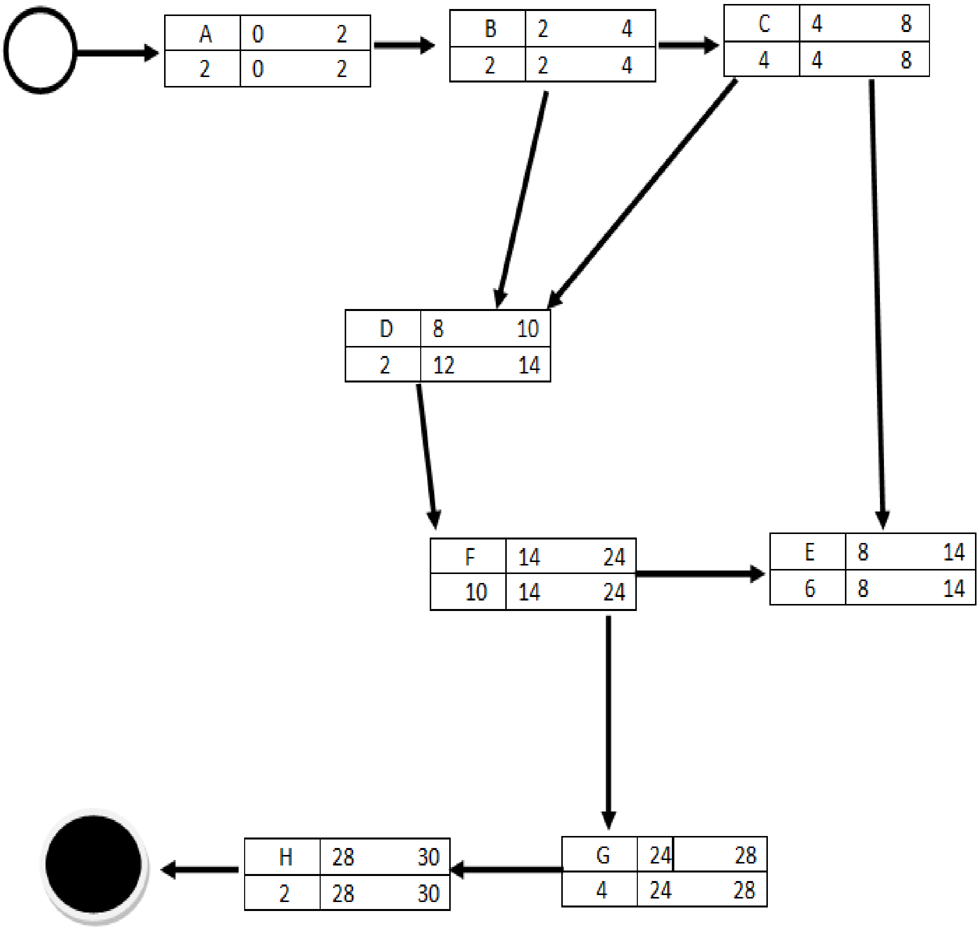

Figure 1: Network Diagram

4. Estimate Activity Completion Time

These are the following Activity Completion Time.

Table 8: Activity Completion Time

Activity	Duration	ES	EF	LS	LF	TS	FS
A	2	0	2	0	2	0	0
B	2	2	4	2	4	0	0
C	4	4	8	4	8	0	0

D	2	8	10	12	14	4	4
E	6	8	14	8	14	0	0
F	10	14	24	14	24	0	0
G	4	24	28	24	28	0	0
H	2	28	30	28	30	0	0

5. Identify the Critical Path

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

The critical path of Our Project is:

A=>B=>C=> E=>F=>G=> H = 30 Weeks

1.7. Gantt chart

The Gantt chart enumerates the activities to be performed on the vertical axis and their corresponding duration on the horizontal axis. It is possible to schedule activities by either an early start or late start logic. In the early start approach, each activity is initiated as early as possible without violating the precedence relations. In the late start approach, each activity is delayed as much as possible as long as the earliest finish time of the project is not compromised. As shown in figure 2

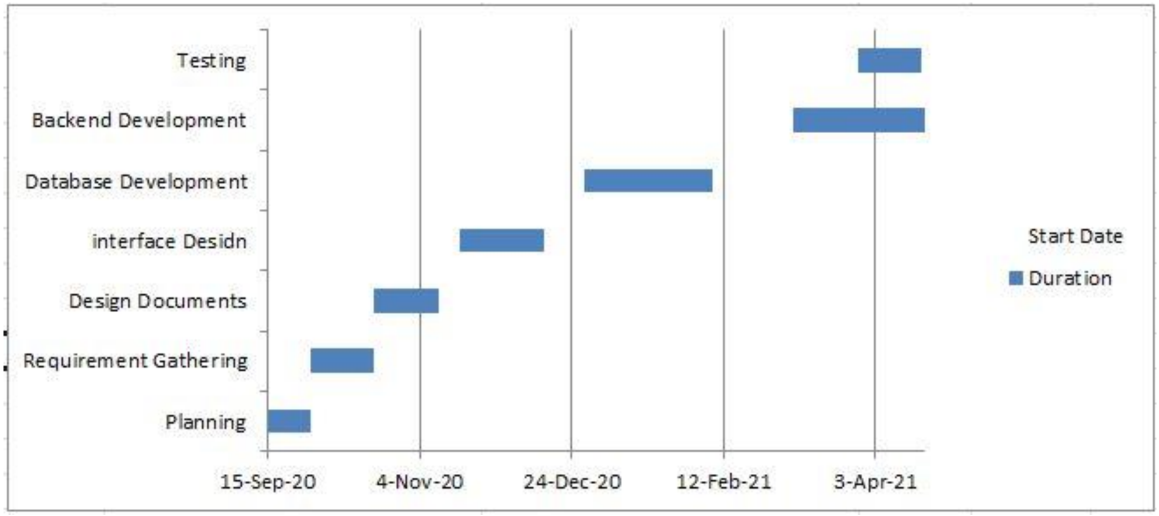

Figure 2: Gantt chart Figure

1.8. Introduction to Team member and their skill set

Abdur Rehman

Mr. Abdur Rehman is performing the following responsibilities in the project:

(1) Documentation
(2) Frontend Development
(3) Desktop Application Development
(4) Data Gathering
(5) Implementation
(6) Testing

Muhammad Qasim

Mr.Muhammad Qasim is performing the following responsibilities in the project:

(1) User interface designing
(2) Desktop Application Development
(3) Graphics Designing
(4) Implementation
(5) Testing

Uzair Jamil

Mr. Uzair Jamil is performing the following responsibilities in the project:

(1) Android development
(2) Feasibility Study
(3) Implementation
(4) Design Document

Ameer hamza

Mr. Ameer Hamza is performing the following responsibilities in the project.

(1) Data Gathering
(2) Feasibility study
(3) Design Document
(4) Android Development

1.9. Task and Member Assignment Table

In this table, tasks are assigned to members.

Table 9: Task Assignment

Task	Engineer
T1	T1(Abdur Rehman)
T2	T2(Uzair Jamil)
T3	T2(Abdur rehman,Ameer)
T4	T4(Uzair,Ameer)
T5	T2(Ameer,Uzair)
T6	T5 (Abdurrehma,Qasim)
T7	T4(Qasim)
T8	T7(Abdurrehma,Qasim)

Task Dependency Table

Task Dependency is a relationship in which a task or milestone relies on other tasks to be performed (completely or partially) before it can be performed. This is also referred to as a logical relationship. A logical relationship can be a dependency between project tasks or between tasks and milestones.

Table 10: Task Dependency

Task	Duration (weeks)	Dependencies
T1	2	None
T2	2	A (Feasibility)
T3	2	B (Requirement Specification)
T4	2	B, C (Requirement Specification, Design Document)
T5	6	C (Android Development, Design Document)
T6	10	D, E (Interface Design, Data Gathering)
T7	6	F (Desktop Application Development)

T8	2	G(Implementation)

Allocation of People to Activities:

Strategic management is only successful when employees have the right tools to perform effectively. To accomplish this, a company needs staff allocation solutions that enable employees to balance different tasks. Coupled with efficient scheduling, workflow runs smoothly from one task to the next.

Staff Allocation:

(1) Feasibility Study (Ameer , Uzair)
(2) Requirement Specification (Abdu Rehman)
(3) Design Document (Uzair, Ameer)
(4) Interface Design (Qasim)
(5) Data Gathering (Abdul Rehman, Ameer)
(6) App Development (Qasim, Abdul Rehman)
(7) Implementation (Abdurehman, Qasim, Uzair)
(8) Testing (Abdul Rehman,Qasim)

1.10. Tools and Technology with reasoning

- **Languages and Tools**

LANGUAGES

1. **Java**

We are using java language because Android applications are mostly developed in Java using the android studio and it has better support to android development than other languages and platform like C# Xamrian and Ionic Framework.

2. **Python**

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability, we have pythons based YOLO v3 algorithm for image processing to identify living entities and fire eruptions if any.

3. **C++**

C++ is a programming language which is used to design the front end of the desktop application and all the design or we can say that UI by this language and to transfer sensor readings from drone to mobile and desktop application.

TOOLS

1. **Android Studio 3.0**

Android Studio will be used to develop and design Android apps. Previously eclipse was used but now the android studio has replaced eclipse because it has more support desktop-based development.

2. **Adobe Photoshop**

We will be using Adobe Photoshop to develop our logo and other different restore graphic files for our android application.

3. **MS Visio**
   We will use MS Visio in our project for documentation purpose to draw use cases, entity diagrams etc.

4. Fire Base

Fire Base is used for database connectivity and for online local host and apache...

1.11. Vision Document

The smart fire rescuer system is very effective to lower the risk of lives of peoples in the disaster situation as it alerts very early to make a strategy against abnormal behavior of nature. The system has the ability to detect specific disaster such as fire. It can also be placed in permanent places. The system is using drone which can be able to move to multiple places. The drone has the ability to detect human being as well as the disaster. After detecting any disaster the drone itself sends the gathered data to the central station by web services and android applications where the Algorithm predicts the type of disaster and its seriousness. If there is any prediction of the disaster, the system will alert to the rescue teams by generating an alert message.

The main aspect of the overview of the project is to overcome the outcomes of Disaster is as following:

1. Fire detection
2. Human detection
3. Generates alerts
4. Inform rescue teams

Assumptions:

- Internet access is necessary for the running of the project.

1.12. Risk List

Following are the risks which can be the biggest blow to the Smart fire rescuer project.

(1) Development of similar app before our app launching
(2) It is difficult to create a Desktop and mobile app for different densities of the screen. So, it is possible that the app may not run gracefully on all desktop and android phones.
(3) This application doesn’t support a wide range of stores that may create a problem in fulfilling the objective of the application.

(4) Lack of proper internet or mobile carriers for smooth communication between desktop and mobile app can risk the whole systematic functioning of this project in remote or affected areas.

Second Deliverable For (For Structured Approach)

Chapter 2: System Requirement Specification

2.1 Introduction:

The early fire recognition is hard to task and many salvage groups feels strain to go the specific area where the huge fire comes out. Fixed camera and frameworks can't recognize fire from some meter range accordingly there ought to be some answer for defeat to the present circumstance.

The previous work on the problem statement is the detect fire and increase the detection accuracy by improved algorithm but problems is the camera cannot detect fire from some meter range as the camera is fixed. When implement to the CCTV camera the detection is on one location only and some previous works cannot detect both fire and human those stuck in that fire.

The previous proposed deep learning models cannot able to detect the human want help or not. Some person go near to the fire location to check out the situation then computer vision applications detect both fire and human and can leads to misunderstanding that if rescue is needed or not.

This work shows the successful method to diminish the result of fire in various circumstances utilizing the robot to screen the area and make the powerful and programmed ready forecasts about the circumstances the profound learning model is associate with the robot camera feed and make the expectations after the forecasts gathered it send it to the android application utilized by the salvage groups to proceed to visit the areas under the episode occur. It saves human time and exertion to look through the hotpot territories enduring an onslaught and distinguish human looking for help. the communication through signing profound learning model additionally increment the exactness as the human needed assistance by giving worldwide assist indication with fronting drone utilizing hand signals. The framework will work runtime as no compelling reason to invigorate and reload the screen. The great realistic card and camera likewise increment the exhibition of the work

Here, requirements specification is to be discussed. Requirements specification would lead to the following four steps:

- Identify external interfaces
- Development of context diagram
- Capture “shall statements

- Allocate requirements
- Prioritize requirements
- Development of requirements traceability matrix

2.1.1 Systems Specifications

The following are the clauses that must be included while describing the system specifications.

Introduction

The early fire detection is very difficult to task and many rescue teams feels pressure to go the exact location where the large fire comes out. Fixed camera and systems cannot detect fire from some meter range therefore there should be some solution to overcome to this situation.

This work shows the effective way to reduce the outcome of fire in different situations using the drone to monitor the location and make the effective and automatic alert predictions about the situations the deep learning model is connect to the drone camera Feed and make the predictions after the predictions collected it sends it to the android application used by the rescue teams to go and visit the locations under the incident happen. It saves human time and effort to search the hotpot areas under fire and detect human seeking help. the sign language deep learning model also increase the accuracy as the human wanted help by showing international help sign in front of drone using hand gestures . The system will work runtime as no need to refresh and reload the screen. The good graphic card and camera also increase the performance of the work

As the forest fire is a very big issue to the humans and wild life infrastructure and also it is a very big challenge to rescue the people stuck in the fire, like in fire situation the buildings destroyed badly in some areas, so it is impossible to find people and do rescue operation due to the destruction and blockage, so it is a very difficult task to find humans that are stuck in the debris. we introduces a smart fire rescuer system which works in every situation it is a networked system which is connected to a central system it send all the gathered information to the central system.

Existing System

The existing system is research base much research work not implemented in real life fire and human is detected either by sensors or by camera (computer vision). The system also fixed such as CCTV camera or fixed sensors.

Scope of the System

In this project the system detect fire and people stuck in the fire through deep learning models based on computer vision techniques as well as hardware sensor to detect both fire and human. After detection the real time detection alerts will be sent to the android application used by rescue teams.

Table 11: Scope of the System

For	Rescue team, Drone operator/User
What	Fire Detection Human Detection Send Alert message
The	Smart Fire Rescuer
Is	Desktop Application Android Application
That	Help to find out people stuck in fire in less time. Help to point out the location of fire place even in smoky environment.

Summary of Requirements: (Initial Requirements)

Functional Requirements:

The purposed system must fulfill the following requirements as follow:

A System shall provide a clear and easy interface for user to interact with the system.

The system shall detect human and detect fire though camera. Moreover, the system shall detect human and fire place though sensor. System shall detect human movement the system send data asynchronously and update information without refresh. The User shall receive notification without opening Android application and the user of mobile application shall check the current location of drone.

Non-Functional Requirements:

Performance Requirements

- **User Satisfaction:**

The application is according to the average user and meets the user expectations.

- **Response Time:**

Response time of every operation is good. This has been made possible by careful programming

- **Error Handling:**

Response to user errors and undesired situations has been taken care of to ensure that the application operates without uncertainty.

- **User-Friendly:**

This application is easy to use and understandable for user. An average user can also use the system effectively, without any difficulties.

- **Other non-functional Requirements:**

All requirements are not covered in this document. Requirements might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project etc.

2.1.2. Identifying External Entities

The Identification of External Interfaces is done in two phases.

Over Specify Entities from Abstract:

- Rescue Team users
- Drone Operator

Perform Refinement:

- **User:**
  -Users shall register or login with android application or desktop applications.
  -Users shall buy the soon to expire products on a discount rate.
  -Users send request for items deliveries.

2.1.3. Capture "shall" Statements:

The System shall statement would be all functional requirements.

Para#	Initial requirements
1	The system shall detect human though camera.
2	The system shall detect fire though camera.
3	The system shall detect human though sensor.
4	The system shall detect fire though sensor.
5	System shall detect human movement.
6	The users shall register through mobile application.
7	The system shall work on windows application.
8	The system sent data asynchronously.
9	System shall update information without refresh.
10	Android application shall receive notification without opening.

11	Android notification shall on mobile screen.
12	User of mobile application shall check drone current location.
13	Sensors data shall represent in digital format.
14	The system shall allow user to get registered.
15	The system shall allow user to log in.

2.1.4. Allocate Requirements:

Allocate the requirements in the use cases.

Table 12: Requirements Allocation

Para#	Initial requirements	Use Case Name
1	The system shall detect human though camera	UC_Detect_Human_Cam
2	The system shall detect fire though camera	UC_Detect_Fire_Cam
3	The system shall detect human though sensor	UC_Detect_Human_Sensor
4	The system shall detect fire though sensor	UC_Detect_Fire_Sensor
5	System shall detect human movement	UC_Detect_Movement
6	The users shall register through mobile application	UC_User_Register
7	The system shall work on windows application	UC_WorkOn_WindowApp
8	The system send data asynchronously	UC_Send_Data
9	System shall update information without refresh	UC_Update_Info
10	Andriod application shall receive notification without opening	UC_Receive_Notification
11	Android notification shall on mobile screen	UC_Screen_On
12	User of mobile application shall check drone current location	UC_Check_Drone_Location
13	Sensors data shall represent in digital format	UC_Display_Digital_Data
14	The system shall allow user to get registered.	UC_Reg_Desk_App
15	The system shall allow user to log in.	UC_Login_Desk_App

2.1.5. Prioritize Requirements:

Requirements are prioritized as this will help achieve tasks easily. Rank them as “highest, medium, and lowest”.

Table 13: Prioritize Requirements

Pa ra#	Rank	Initial requirements	Use case ID	Use Case Name
1	HIGH	The system shall detect human though camera	UC_1	UC_Detect_Human_Cam
2	HIGH	The system shall detect fire though camera	UC_2	UC_Detect_Fire_Cam
3	HIGH	The system shall detect human though sensor	UC_3	UC_Detect_Human_Sensor
4	HIGH	The system shall detect fire though sensor	UC_4	UC_Detect_Fire_Sensor
5	HIGH	System shall detect human movement	UC_5	UC_Detect_Movement
6	HIGH	The users shall register through mobile application	UC_6	UC_User_Register
7	HIGH	The system shall work on windows application	UC_7	UC_WorkOn_Window App
8	HIGH	The system send data asynchronously	UC_8	UC_Send_Data
9	HIGH	System shall update information without refresh	UC_9	UC_Update_Info
10	HIGH	Andriod application shall receive notification without opening	UC_10	UC_Receive_Notification
11	MEDIUM	Android notification shall on mobile screen	UC_11	UC_Screen_On
12	MEDIUM	User of mobile application shall check drone current location	UC_12	UC_Check_Drone_Location
13	MEDIUM	Sensors data shall represent in digital format	UC_13	UC_Display_Digital_Data

14	HIGH	The system shall allow user to get registered.	UC_14	UC_Reg_Desk_App
15	HIGH	The system shall allow user to log in.	UC_15	UC_Login_Desk_App

2.1.6. Requirements Traceability Matrix:

The requirements traceability matrix is a table used to trace project lifecycle activities and work products to the project requirements. The matrix establishes a thread that traces requirements from identification through implementation.

Table 14: Requirements Traceability Matrix

Sr#	System Specificaion Text	Build	Use Case Name	Category
1	The system shall detect human though camera	B1	UC_Detect_Human_Cam	Business
2	The system shall detect fire though camera	B2	UC_Detect_Fire_Cam	Business
3	The system shall detect human though sensor	B3	UC_Detect_Human_Sensor	Business
4	The system shall detect fire though sensor	B4	UC_Detect_Fire_Sensor	Business
5	System shall detect human movement	B5	UC_Detect_Movement	Business
6	The users shall register through mobile application	B6	UC_User_Register	Business
7	The system shall work on windows application	B7	UC_WorkOn_WindowApp	Business
8	The system send data asynchronously	B8	UC_Send_Data	Business
9	System shall update information without refresh	B9	UC_Update_Info	Business

10	Andriod application shall receive notification without opening	B10	UC_Receive_Notific ation	Business
11	Android notification shall on mobile screen	B11	UC_Screen_On	Business
12	User of mobile application shall check drone’s current location	B12	UC_Check_Drone_ Location	Business
13	Sensors data shall represent in digital format	B13	UC_Display_Digital _Data	Business
14	The system shall allow user to get registered.	B13	UC_Reg_Desk_App	Business
15	The system shall allow user to log in.	B15	UC_Login_Desk_A pp	Business

2.2.1. High-Level Usecase Diagram:

High level visual representations of user requirements and they can act into the use cases. The use cases are the ovals with which the actor interacts. The box is separating some top-level internals of the system from the external actors. High level of our project is shown in figure 1.

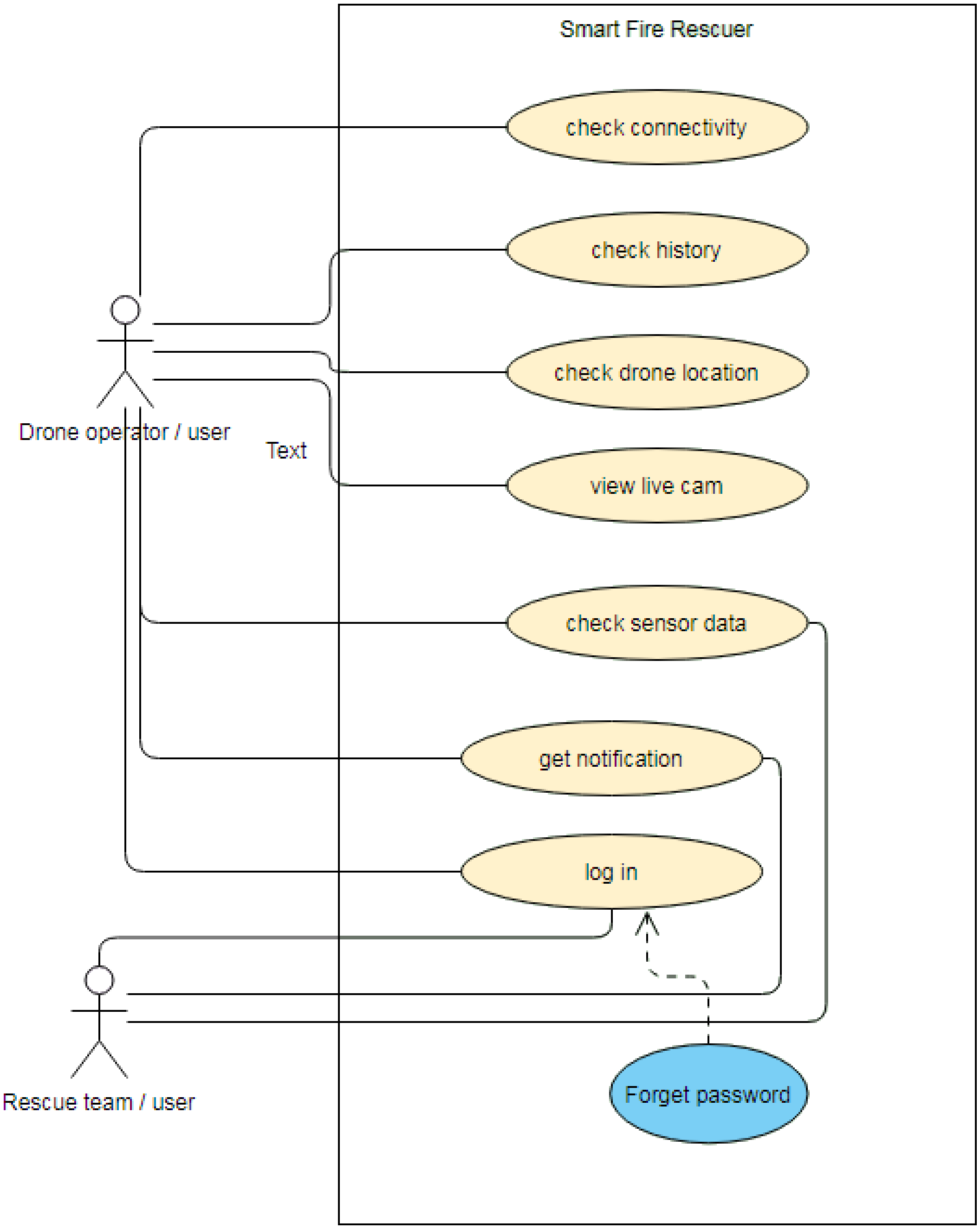

Figure 3: High-Level Usecase Diagram

2.2.2. Analysis Level Usecase Diagram:

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

2.2.3. Usecase Description

While technically not part of UML, use case documents are closely related to UML use cases. A use case document is a text that captures the detailed functionality of a use case. Such documents typically contain the following parts:

- **UC_1** describes what is considered the happy path the functionality that occurs when use case Detect Human through camera executes without errors. It can include critical conditions when the execution of detection occurs.

Table 15: Use Case Description

Use Case Description	
Use Case ID:	UC_1
Use Case Name:	UC_Detect_Human_Cam
Goal/Purpose:	To detect human by using camera in order to find human stuck inside the fire, building or debris.
Actors:	Drone Operator/user
Pre-Conditions	Live screen
Post-Conditions:	Person detected
Basic Flow:	Shows live video in order to detect any person
Alternate Flow(s):	Detect human by using PIR sensor

- **UC_2** describes what is considered the happy path the functionality that occurs when use Detect Fire through camera executes without errors. It can include critical conditions when the execution of detection occurs.

Table 16: Use Case Description

Use Case Description	
Use Case ID:	UC_2
Use Case Name:	UC_Detect_Fire_Cam
Goal/Purpose:	To detect fire by using camera in order to find the location of the fire.
Actors:	Drone Operator/user
Pre-Conditions	Live screen
Post-Conditions:	Fire Detected
Basic Flow:	Shows live video in order to detect fire
Alternate Flow(s):	Detect fire by using fire sensor sensor

- **UC_3** describes what is considered the happy path the functionality that occurs when use Detect Human through sensor executes without errors. It can include critical conditions when the execution of detection occurs.

Table 17: Use Case Description

Use Case Description	
Use Case ID:	UC_3
Use Case Name:	UC_Detect_Human_Sensor
Goal/Purpose:	To detect human by using sensor in order to find the person
Actors:	Drone Operator/user
Pre-Conditions	Live screen
Post-Conditions:	Human detected
Basic Flow:	Sent notification on android application when sensor detects human
Alternate Flow(s):	Detect human by using camera

- **UC_4** describes what is considered the happy path the functionality that occurs when use Detect Fire through Sensor executes without errors. It can include critical conditions when the execution of detection occurs.

Table 18: Use Case Description

Use Case Description	
Use Case ID:	UC_4
Use Case Name:	UC_Detect_Fire_Sensor
Goal/Purpose:	To detect fire by using sensor in order to find the location of the fire.
Actors:	Drone operator/user
Pre-Conditions	Live screen
Post-Conditions:	Fire detected
Basic Flow:	Sent notification on android application when sensor detects fire
Alternate Flow(s):	Detect fire by using camera

- **UC_5** describes what is considered the happy path the functionality that occurs when use case Detect Human Movement execute without errors. It can include critical conditions when execution of detection occurs.

Table 19: Use Case Description

Use Case Description	
Use Case ID:	UC_5
Use Case Name:	UC_Detect_Movement
Goal/Purpose:	To detect Movement of a person by using PIR sensor in order to find the person.
Actors:	Drone operator/user
Pre-Conditions	Live screen
Post-Conditions:	Motion detected
Basic Flow:	Sent notification on android application when sensor detects movement
Alternate Flow(s):	Detect person by using camera
Exception Flow(s):	Detect person by using sensor

- **UC_6** describes what is considered the happy path the functionality that occurs when use case Register execute without errors. It can include critical conditions when execution of registration occurs.

Table 20: Use Case Description

Use Case Description	
Use Case ID:	UC_6
Use Case Name:	UC_User_Register
Goal/Purpose:	Fill the registration form for users to get register with the android appliction for using the all functionality of the application.
Actors:	Rescue team/user
Pre-Conditions	Home page
Post-Conditions:	Login
Basic Flow:	Show the accessibility for user.
Alternate Flow(s):	Get register with the application by verifying Identity.
Exception Flow(s):	Internet error

- **UC_7** describes what is considered the happy path the functionality that occurs when use case Working with Desktop application execute without errors. It can include critical conditions when execution of registration occurs.

Table 21: Use Case Description

Use Case Description	
Use Case ID:	UC_7
Use Case Name:	UC_WorkOn_WindowApp
Goal/Purpose:	Fill the registration form for users to get register with the desktop appliction for using the all functionality of the application.
Actors:	Drone operator/user
Pre-Conditions	Home page
Post-Conditions:	Login
Basic Flow:	Show the accessibility for user.
Alternate Flow(s):	Get register with the application by verifying Identity.
Exception Flow(s):	Internet error

UC_8 describes what is considered the happy path the functionality that occurs when uses case Sent Data to execute without errors. It can include critical conditions when execution sending data occur.

Table 22: Use Case Description

Use Case Description	
Use Case ID:	UC_8
Use Case Name:	UC_Send_Data
Goal/Purpose:	To send data asynchronously for immediate action
Actors:	Drone operator
Pre-Conditions	Live screen
Basic Flow:	Sent notification on android application when sensors detects human, fire or movement

- **UC_9** describes what is considered the happy path the functionality that occurs when use case Update Info execute without errors. It can include critical conditions when execution of update info occurs.

Table 23: Use Case Description

Use Case Description	
Use Case ID:	UC_9
Use Case Name:	UC_Update_Info
Goal/Purpose:	To update information without refresh in order to get instant/immediate alert.
Actors:	Rescue team/user
Pre-Conditions	Live screen
Basic Flow:	Sent notification on android application when sensors detects human, fire or movement
Alternate Flow(s):	refresh

- **UC_10** describes what is considered the happy path the functionality that occurs when use case Receive notification without errors. It can include critical conditions when executions of receiving notification occur.

Table 24: Use Case Description

Use Case Description	
Use Case ID:	UC_10
Use Case Name:	UC_Receive_Notification
Goal/Purpose:	To receive information without opening the android application as pop up notification in order to get instant/immediate alert.
Actors:	Rescue team/user
Pre-Conditions	Live screen
Post-Conditions:	Pop-up notification
Basic Flow:	Receive notification on mobile without opening android application
Alternate Flow(s):	Open android application

- **UC_11** describes what is considered the happy path the functionality that occurs when uses case Screen on to execute without errors. It can include critical conditions when execution of screen on on notification to occur.
-

Table 25: Use Case Description

Use Case Description	
Use Case ID:	UC_11
Use Case Name:	UC_Screen_On

Goal/Purpose:	Mobile screen turns on when notification comes in order to get instant/immediate alert.
Actors:	Rescue team/user
Pre-Conditions	Live screen
Post-Conditions:	Screen on
Basic Flow:	Mobile screen turns on for notification to get instant alert.
Alternate Flow(s):	Notification sound alert

- **UC_12** describes what is considered the happy path the functionality that occurs when use case Check Drone Location execute without errors. It can include critical conditions when execute the drone location check.

Table 26: Use Case Description

Use Case Description	
Use Case ID:	UC_12
Use Case Name:	UC_Check_Drone_Location
Goal/Purpose:	To check the current location of drone in order to reach to in emergency situation
Actors:	Drone operator
Pre-Conditions	Location screen
Post-Conditions:	Get location by coordinates (longitude and latitude)
Basic Flow:	Get location by clicking on location button
Alternate Flow(s):	None

- **UC_13** describes what is considered the happy path the functionality that occurs when use case Display Data execute without errors. It can include critical conditions when execution of display data occurs.

Table 27: Use Case Description

Use Case Description	
Use Case ID:	UC_13
Use Case Name:	UC_Display_Digital_Data
Goal/Purpose:	Sensors data represents in digital format for better understanding

Actors:	Drone operator
Pre-Conditions	Live screen
Post-Conditions:	Display data on screen
Basic Flow:	Sensors data display on the screen in digital format
Alternate Flow(s):	None
Exception Flow(s):	None

- **UC_14** describes what is considered the happy path the functionality that occurs when uses case Register desktop application to execute without errors. It can include critical conditions when execution of registration occurs.

Table 28: Use Case Description

Use Case Description	
Use Case ID:	UC_14
Use Case Name:	UC_Reg_Desk_App
Goal/Purpose:	Fill the registration form for user to get register with desktop application for using all functionality of the application.
Actors:	Admin
Pre-Conditions	Home screen
Post-Conditions:	View live cam/video
Basic Flow:	Show the accessibility for user.
Alternate Flow(s):	Get register with the desktop application by verifying.
Exception Flow(s):	Internet error

- **UC_15** describes what is considered the happy path the functionality that occurs when uses case log in desktop application to execute without errors. It can include critical conditions when log in occur.

Table 29: Use Case Description

Use Case Description	
Use Case ID:	UC_14
Use Case Name:	UC_Reg_Desk_App
Goal/Purpose:	Fill the log in form for user to get start with desktop application to use all functionalities of the application.
Actors:	Drone operator
Pre-Conditions	Home screen
Post-Conditions:	View live cam/video
Basic Flow:	Show the accessibility for user.
Alternate Flow(s):	Get login with the desktop application by verifying.
Exception Flow(s):	Internet error

Chapter 3: Design Document (For Structured Approach)

3.1. Introduction:

The early fire recognition is hard to task and many salvage groups feels strain to go the specific area where the huge fire comes out. Fixed camera and frameworks can't recognize fire from some meter range accordingly there ought to be some answer for defeat to the present circumstance.

The previous work on the problem statement is the detect fire and increase the detection accuracy by improved algorithm but problems is the camera cannot detect fire from some meter range as the camera is fixed. When implement to the CCTV camera the detection is on one location only and some previous works cannot detect both fire and human those stuck in that fire.

The previous proposed deep learning models cannot able to detect the human want help or not. Some person go near to the fire location to check out the situation then computer vision applications detect both fire and human and can leads to misunderstanding that if rescue is needed or not.

This work shows the successful method to diminish the result of fire in various circumstances utilizing the robot to screen the area and make the powerful and programmed ready forecasts about the circumstances the profound learning model is associate with the robot camera feed and make the expectations after the forecasts gathered it send it to the android application utilized by the salvage groups to proceed to visit the areas under the episode occur. It saves human time and exertion to look through the hotpot territories enduring an onslaught and distinguish human looking for help. the communication through signing profound learning model additionally increment the exactness as the human needed assistance by giving worldwide assist indication with fronting drone utilizing hand signals. The framework will work runtime as no compelling reason to invigorate and reload the screen. The great realistic card and camera likewise increment the exhibition of the work.

Analysis & Design Model for structured approach must contain following artifacts:

1. Entity Relationship Diagram
2. Data Flow Diagram (Functional Model)
3. State Transition Diagram (Behavioral Model)
4. Architecture Design

3.2. Entity Relationship Diagram (ERD)

In the analysis model, Entity Relationship Diagram is used to understand the system under consideration with respect to entities involved and their relationships. Each entity is documented by extracted its attributes, cardinality, and modality.

Entity relationship diagram consists of three things

- **Entities** in system
- **Relationship** between entities
- **Attributes** of entities

Entities in our system

1. Drone Operator / User
2. Rescue team

Associated Entities:

1. Operating drone
2. Receive alert notification

Relationships:

1. Registration

3.3. Data flow diagram (Functional Model)

DFD is all about to identify the major processes in your system and develop Data Flow Diagram up to required level.

3.3.1. Context Level DFD

A context diagram shows the context into which the business process fits. It also shows the overall business process as just one process and shows all the outside entities that receive information from or contribute information to the system.

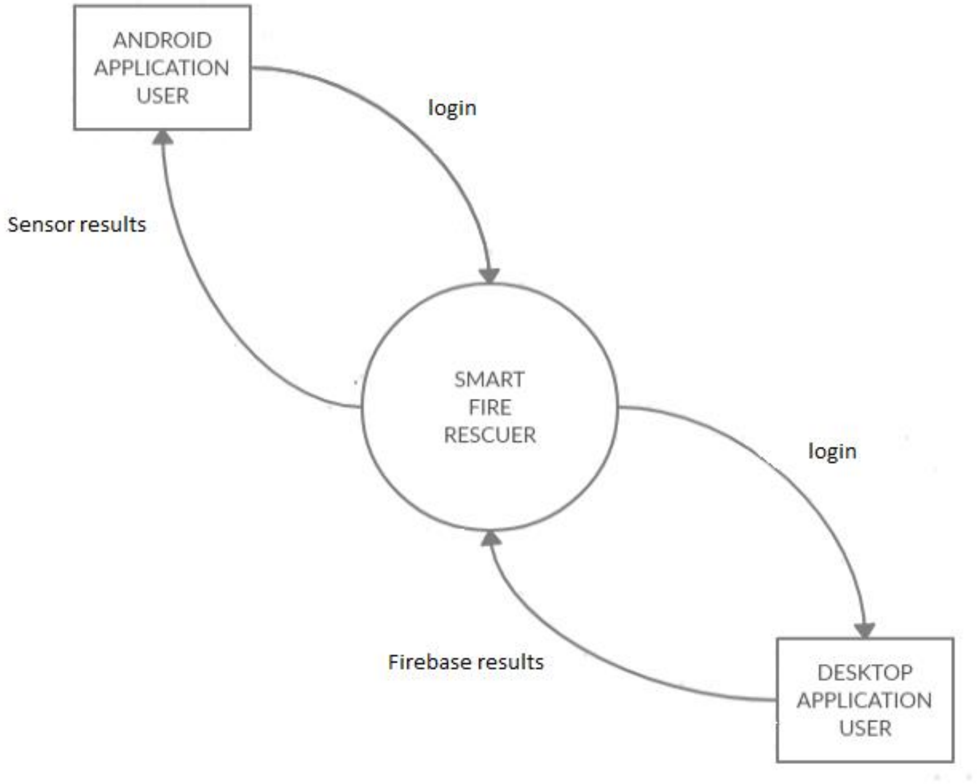

Figure 4: Context Level DFD

3.3.2. Level 1 Diagram

This diagram shows all the processes that comprise the overall system and how information moves from and to each process. Data stores are added to it.

3.4. State Transition Diagram

State Transition Diagram is developed to represent the behavior of the system under consideration. The constructs of STD are as follows:

In Smart Fire Rescuer desktop application there are some states and actions going to perform and some transition happened the STD of the desktop application is given below.

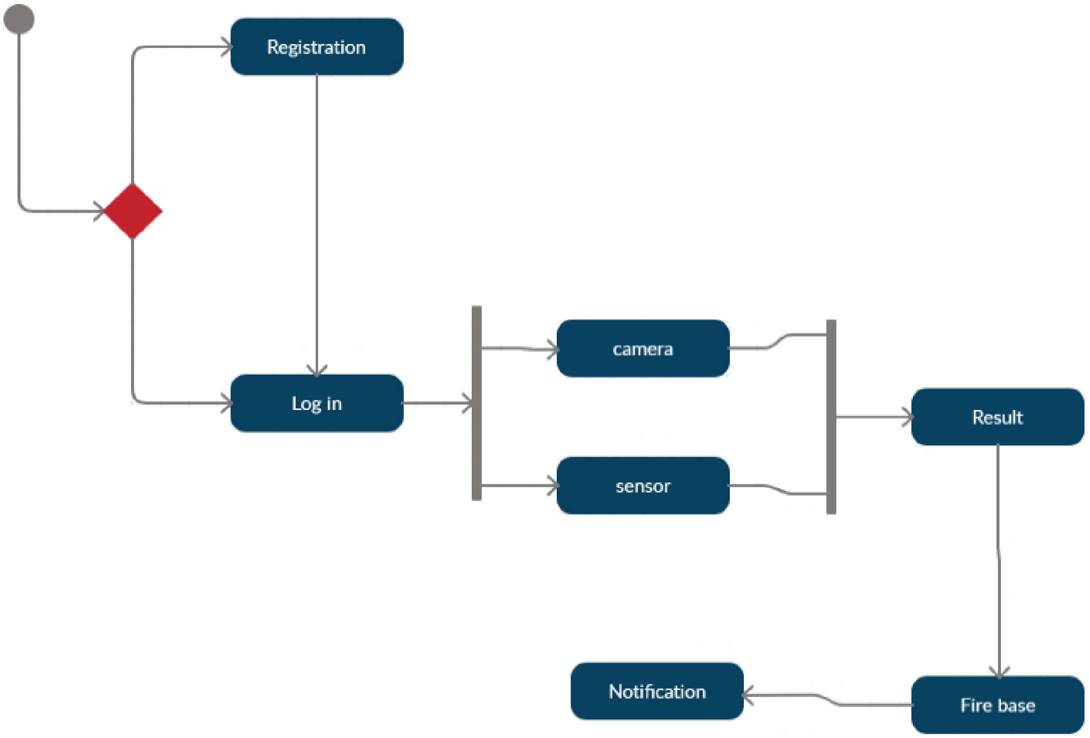

Figure 5: State Transition Diagram

3.5. Architectural design

The Focus of architecture design is the Mapping of Requirements into Software Architecture. DFD prepared in analysis model is analyzed to do it.

Two major structural patterns or two major alternatives are Transform (Flow) Analysis and Transaction (Flow) Analysis.

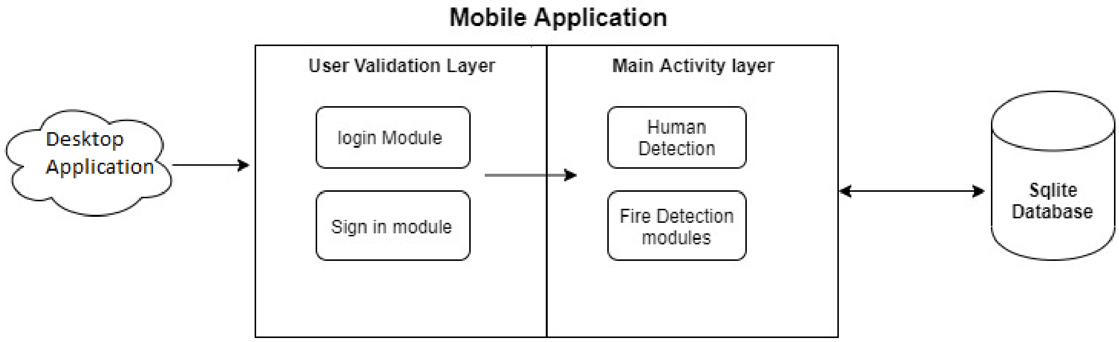

Figure 6 Application Archtecture

2.6. Component Level Design

Every component, which is appearing in program structure/design architecture, and there flow of process.

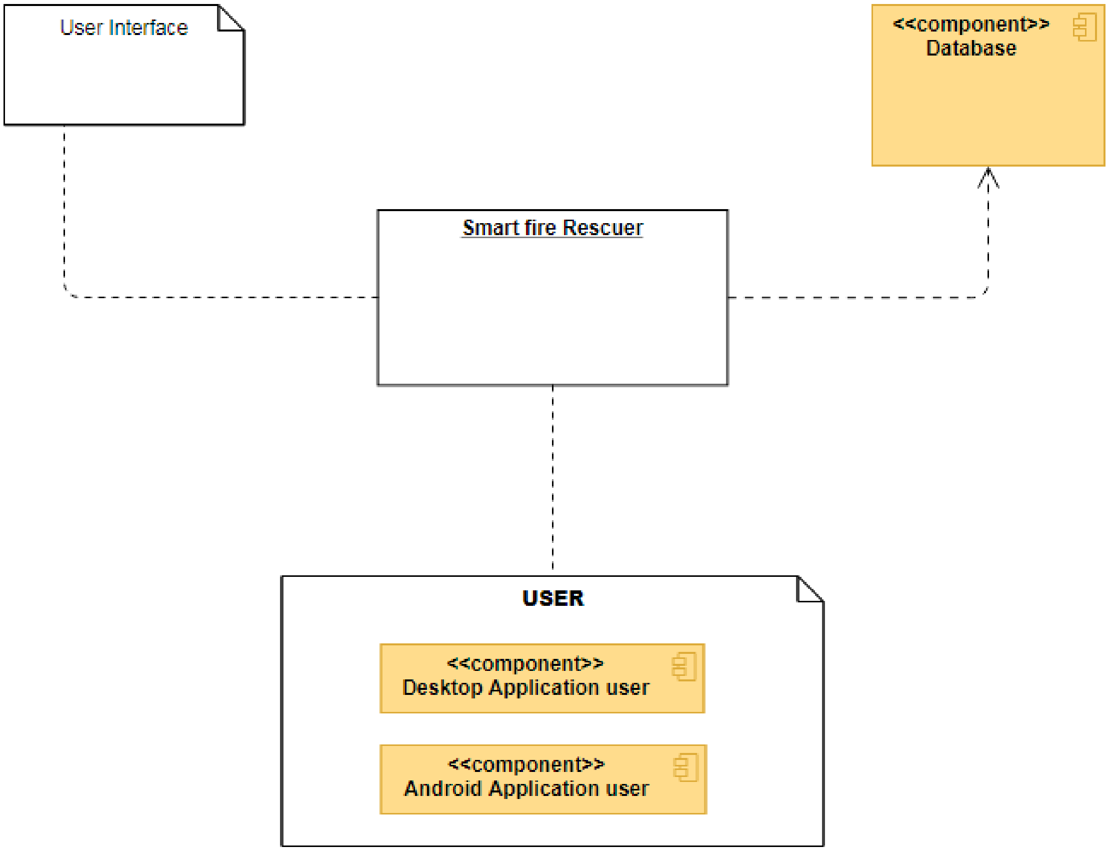

Figure 7: Component Design Diagram

Chapter 4: User Interface Design

4.1. Introduction

A user interface design consists of three main parts:

Page elements should be visualized on paper before building them in the computer. Just as you draw a sitemap to plan the site, use cartoons and storyboards to begin blocking out the site's appearance and navigational scheme.

1. Sitemaps
2. Storyboards
3. Navigational maps
4. Traceability Matrix

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. Site map's of our project is shown in figure 11

4.3. Story boards

A storyboard is a sequence of single images, each of which represents a distinct event or narrative. It is also a visual representation of the script illustrating the interaction between the user and the machine. It can also be imagined as a film in visual-outline form.

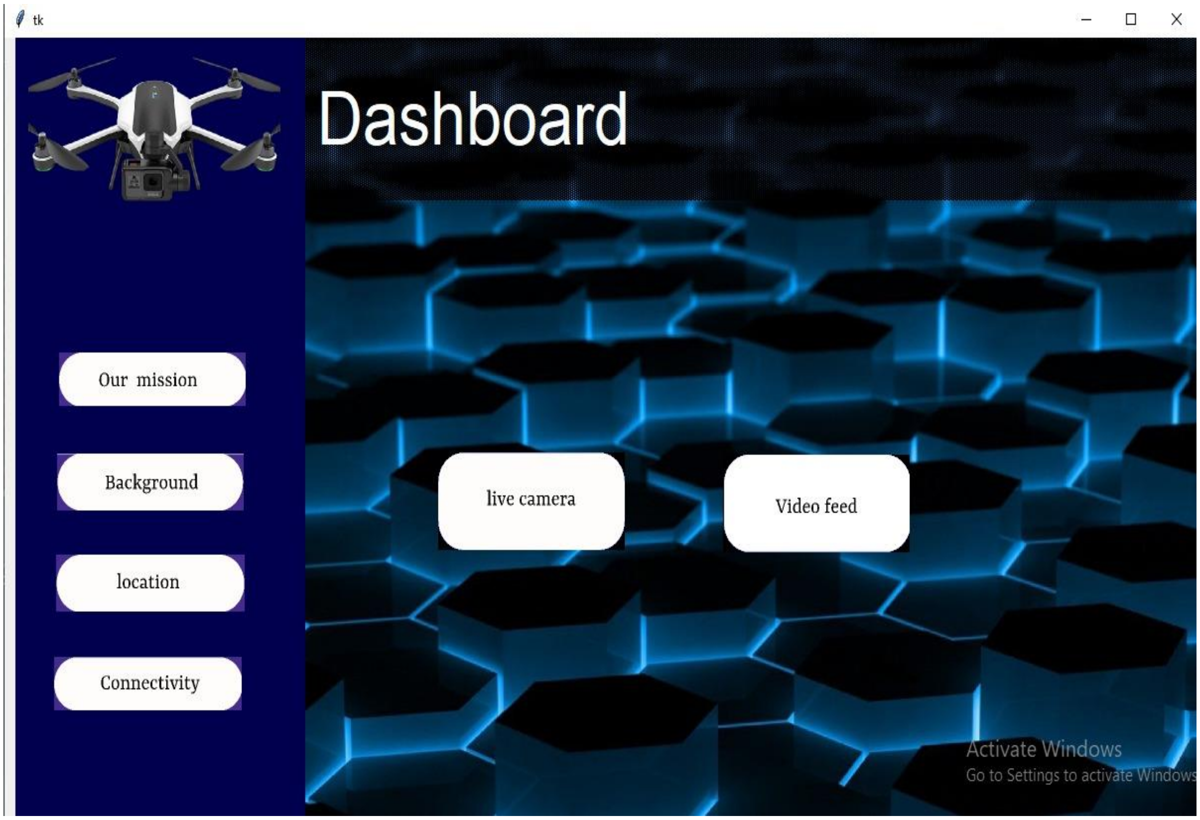

Figure 8: Home Screen

- Environment: Desktop App
- Visual cues: User can view home screen
- Audible cues: No Audio
- User input: User will select which page to open
- Machine output: The Desired page will be open

- User's emotions: User touch the desired feature
- Technology: Python with open CV and tkinter.

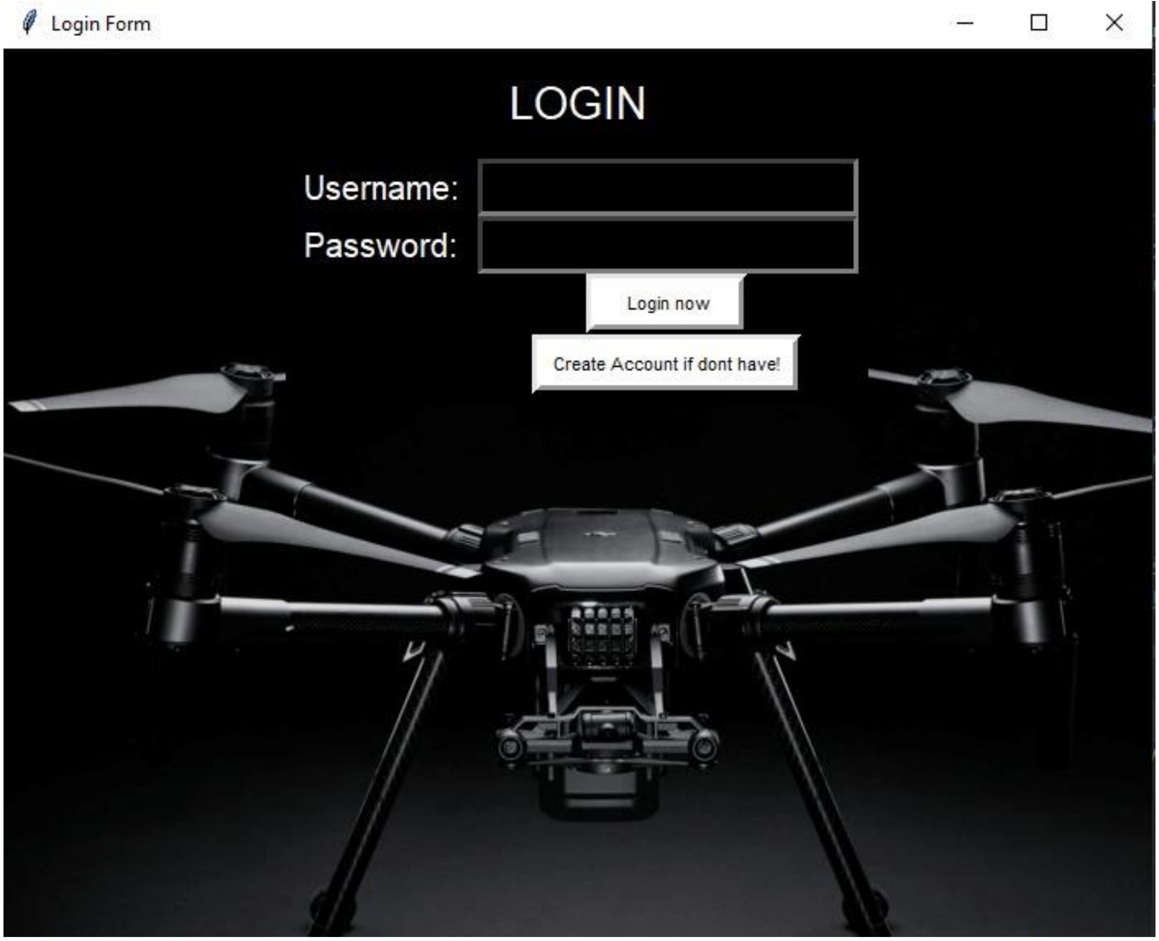

Figure 9: Login Screen

- Environment: Desktop App
- Visual cues:

  (1) The user can enter the username and password for login.
  (2) The user can fill the information form for registration
  (3) The user can registered by verify email address.
- Audible cues: No Audio
- Tactile cues: User can edit textbox for input
- User input: User enter text in text fields for completing the information
- Machine output: user registered
- User's emotions: User get the login access
- Technology: Python with open cv,tkinter and sqlite.

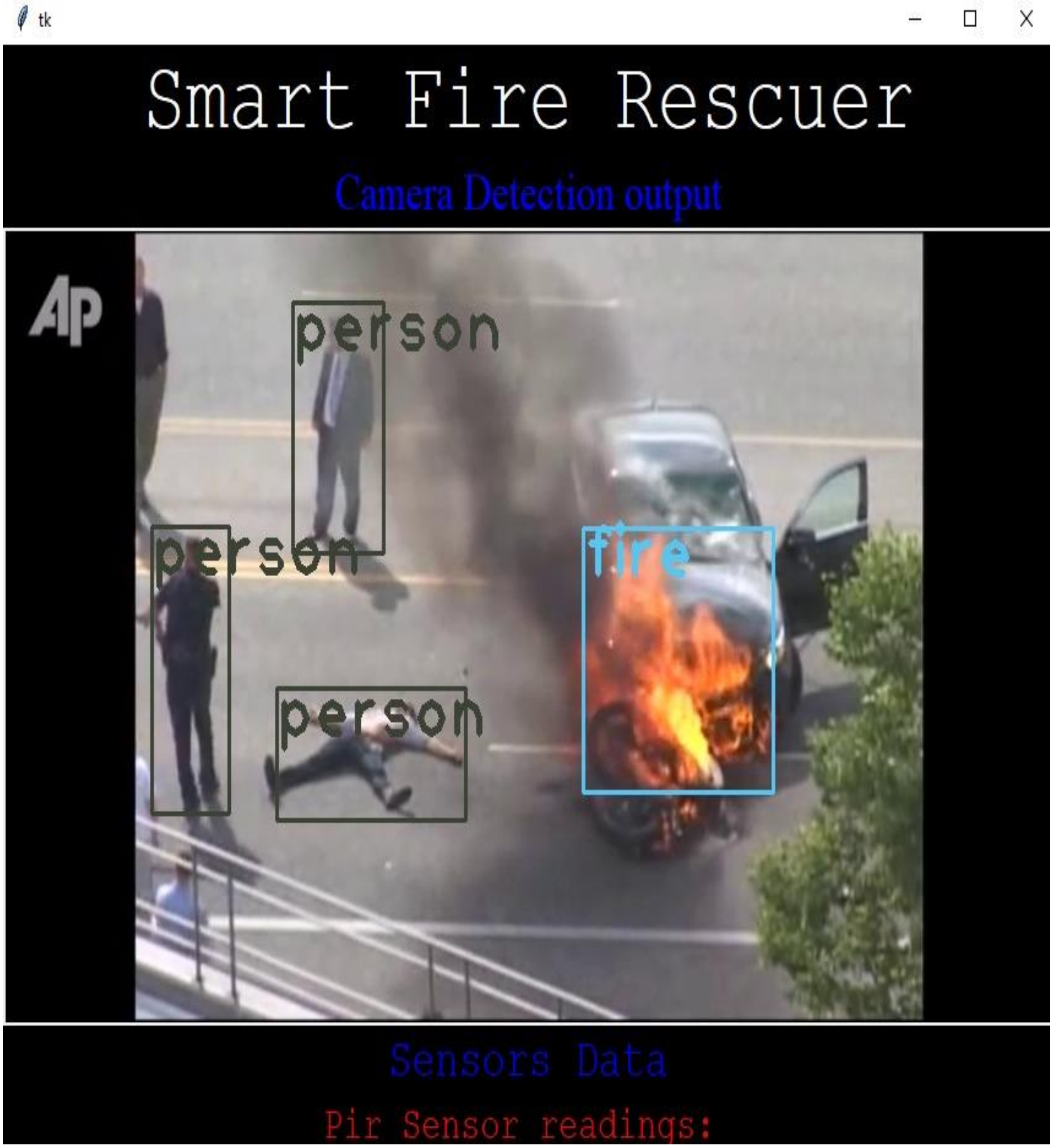

Figure 10: **Seller Screen**

- Environment: Desktop App
- Visual cues: User can see the video and live camera results
- Audible cues: No Audio
- User input: Select the camera option.
- Machine output: Show video
- User’s emotions: User will get all detail about the camera and sensors.
- Technology: Python with open cv,tkinter and sqlite.

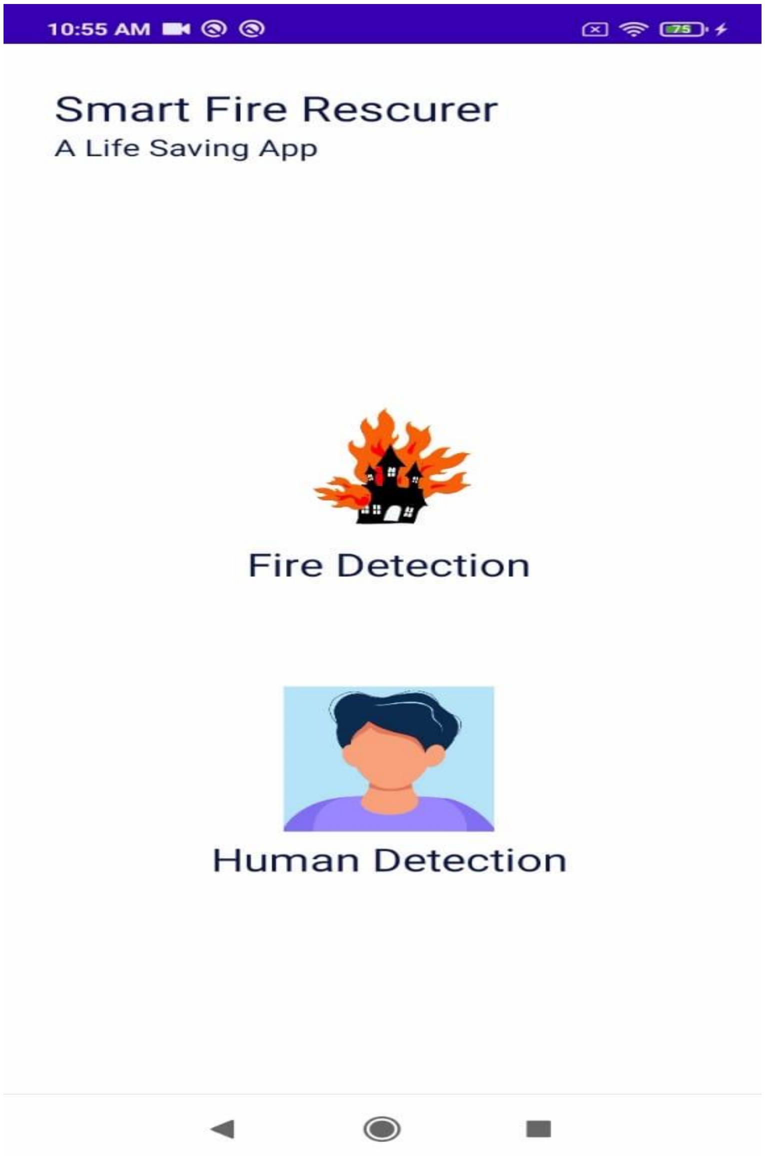

Figure 11: Mobile Application

- Environment: Andriod Studio
- Visual cues: User can see the video and live camera results
- Audible cues: No Audio
- User input: Select the fire detection or human detection module.
- Machine output: Show results
- User’s emotions: Seller will get have all detail about the products and orders
- Technology: Java and firebase.

Chapter 5: Software Testing

5.1 Introduction:

From the start of software engineering different software testing schemes and testing, templates have invented. These testing templates are used to test both UI and unit testing.
Every credible software house developed their testing documentation for the purpose of developing efficient systems.

To provide a common set of standardized documents the IEEE developed the 829 Standard for Software Test Documentation for any type of software testing, including User Acceptance Testing.

In our testing module, we are going to test different features and modules of the system using a documentation template supported by IEEE829.

Test Cases:

5.1.1 Test Case: Connection Successful with a Sqlite database

Table 30: Test Case

Test Engineer:	Muhammad Qasim
Test Case ID:	TC1
Related UC/FR:	UC_1
Date:	14-4-2021
Purpose:	To fetch the data from the database
Pre-Req:	Set up a database
Test Data:	Connection strings
Steps:	• Open database connection • Check connection strings • Check whether the connection is established successfully

Expected Result:	Connection with the Sqlite database is successful.
Actual Result:	Connection established successfully
Status:	Pass

5.1.2 Test Case: Connection Unsuccessful with a local database

Table 31: Test Case

Test Engineer:	Muhammad Qasim
Test Case ID:	TC1
Related UC/FR:	UC_1
Date:	14-4-2021
Purpose:	Connection Successful with a local database
Pre-Req:	Set up a database
Test Data:	Connection strings
Steps:	• Run connection strings • Database connection not established. • Sqlite database not available
Expected Result:	Connection with the Sqlite database is successful.
Actual Result:	Connection not established
Status:	Fail

5.1.3 Test Case: Data Retrieval from database

Table 32: Test Case

Test Engineer:	Abdul rehman
Test Case ID:	TC2
Related UC/FR:	UC_2
Date:	14-4-2021
Purpose:	Data Retrieval from the database
Pre Req:	Database connection must be established

Test Data:	Sqlite database data
Steps:	• Establish a database connection • Run database table quires to fetch data • Selects the demanded data from the database. • There is no ambiguity in data. • The retrieved data may be stored in a file, printed, or viewed on the screen
Expected Result:	Data is displayed according to the database quires.
Actual Result:	Data is retrieved from the database
Status:	Pass

5.1.4 Test Case: Fail to Retrieve Data from database

Table 33: Test Case

Test Engineer:	Abdul rehman
Test Case ID:	TC2
Related UC/FR:	UC_2
Date:	14-4-2021
Purpose:	Data Retrieval from the database
PreReq:	A database connection must be established
Test Data:	Sqlite database data
Steps:	• Establish a database connection • Run database table quires to fetch data • Selects the demanded data from the database. • There is no such table present in the database
Expected Result:	Data is displayed according to the database quires.
Actual Result:	Data is not retrieved from a database table
Status:	Fail

5.1.5 Test Case: Check machine learning model results

Table 34: Test Case

Test Engineer:	Ameer hamza
Test Case ID:	TC3
Related UC/FR:	UC_8
Date:	23-6-2021
Purpose:	Check models output
Pre-Req:	Updated model not uploaded
Test Data:	Yolo model for deep learning
Steps:	Browser updated model Add on the code Run the compiler
Expected Result:	Models provides excellent results
Actual Result:	Models provides excellent results
Status:	Pass

5.1.6 Test Case: Check live camera is working

Table 35: Test Case

Test Engineer:	Ameer hamza
Test Case ID:	TC4
Related UC/FR:	UC_9
Date:	14-4-2021
Purpose:	Run the live camera
Pre-Req:	Select the camera for running
Test Data:	Camera detail for running
Steps:	Open compiler Give camera path Run the code Check the results
Expected Result:	Live video is playing
Actual Result:	Live video is playing
Status:	Pass

5.1.7 Test Case: check sensor data is uploading

Table 36: Test Case

Test Engineer:	Abdul rehman
Test Case ID:	TC4
Related UC/FR:	UC_3
Date:	14-4-2021
Purpose:	check sensor data is uploading
Pre-Req:	Select sensor to check
Test Data:	Sensors data reading
Steps:	• Connect sensor to nodemcu • Connect sensor to nodemcu • Run wizard • Check result
Expected Result:	Sensor is sending data to firebase successfully
Actual Result:	Sensor is sending data to firebase successfully
Status:	Pass

i. **Test Case: Sensor data not going to firebase**

Table 37: Test Case

Test Engineer:	Uzair Jameel
Test Case ID:	TC5
Related UC/FR:	UC_10
Date:	14-4-2021
Purpose:	Check sensor reading to firebase
Pre-Req:	Select sensors
Test Data:	Sensor Reading data
Steps:	• Select sensor data • Push to firebase • Open firebase to check
Expected Result:	Sensor data is going to firebase
Actual Result:	Sensor datais not going to firebase
Status:	Fail

5.1.9 Test Case: camera results uploading to firebase

Table 38: Test Case

Test Engineer:	Muhammad Qasim
Test Case ID:	TC6
Related UC/FR:	UC_11
Date:	14-4-2021
Purpose:	Check camera results is uploading
Pre-Req:	Identify camera ip address
Test Data:	Camera live videos
Steps:	• Open camera in compiler • Give camera path • Check the firebase connection • Upload the result • Check camera status
Expected Result:	Camera data sent successfully
Actual Result:	Camera data sent successfully
Status:	Pass

5.1.10 Test Case: getting alert message successfully

Table 39: Test Case

Test Engineer:	Abdul rehman
Test Case ID:	TC8
Related UC/FR:	UC_12
Date:	14-4-2021
Purpose:	Android application get alert message successully
Pre-Req:	Sign in on andriod application
Test Data:	Firebase data readings
Steps:	• Sign in on andriod application • Check sensor reading • Open firebase and check sensor reading.

Expected Result:	Getting alert message successfully
Actual Result:	Alert message is sent successfully
Status:	Pass

5.1.11 Test Case: To check desktop application fetch data from firebase

Table 40: Test Case

Test Engineer:	Abdul rehman
Test Case ID:	TC9
Related UC/FR:	UC_13
Date:	14-4-21
Purpose:	Check desktop application fetch sensor reading
Pre-Req:	Login to desktop application
Test Data:	Firebase sensor reading
Steps:	• Login to desktop application • Check live and video feed tabs • Check sensor results
Expected Result:	desktop application fetch data from firebase succesfully
Actual Result:	desktop application fetch data from firebase successfully
Status:	Pass

5.1.12 Test Case: Connection Successful Andriod with a Sqlite database

Table 41: Test Case

Test Engineer:	Muhammad Qasim
Test Case ID:	TC10
Related UC/FR:	UC_15
Date:	14-4-2021
Purpose:	To fetch the data from the database

Pre-Req:	Set up a database
Test Data:	Connection strings
Steps:	• Open database connection • Check connection strings • Check whether the connection is established successfully
Expected Result:	Connection with the Sqlite database is successful.
Actual Result:	Connection established successfully
Status:	Pass

5.1.13 Test Case: check sensor output is showing

Table 42: Test Case

Test Engineer:	Muhammad qasim
Test Case ID:	TC11
Related UC/FR:	UC_16
Date:	14-4-21
Purpose:	check sensor output is showing
Pre-Req:	Run arduino uno with nodemcu
Test Data:	Sensor reading data
Steps:	• Open arduino uno compiler • Connect circuit to laptop • Upload code • Check output wizard
Expected Result:	check sensor output is showing successfully
Actual Result:	check sensor output is showing successfully
Status:	Pass

5.1.15 Test Case: Check system run on other ip camera address

Table 43: Test Case

Test Engineer:	Abdul rehman
Test Case ID:	TC12
Related UC/FR:	UC_17
Date:	14-4-21
Purpose:	Check system run on other ip camera address
Pre-Req:	Login to desktop application
Test Data:	Device wifi ip addresss
Steps:	• Login to desktop application • Give ip address of camera • Check machine learnig results
Expected Result:	system run on other ip camera address succesfully
Actual Result:	system run on other ip camera address successfully
Status:	Pass

5.1.16 Test Case: To check desktop application runs video feed

Table 44: Test Case

Test Engineer:	Abdul rehman
Test Case ID:	TC16
Related UC/FR:	UC_14
Date:	14-4-21
Purpose:	check desktop application runs video feed
Pre-Req:	Login to desktop application
Test Data:	Video data for detection
Steps:	• Login to desktop application • Check video feed tabs • Check machine learning results
Expected Result:	check desktop application runs video feed successfully
Actual Result:	check desktop application runs video feed successfully

Status:	Pass

5.1.17 Test Case: To check nodemcu connected to internet

Table 45: Test Case

Test Engineer:	Abdul rehman
Test Case ID:	TC15
Related UC/FR:	UC_19
Date:	14-4-21
Purpose:	Check nodemcu connected to internet
Pre-Req:	Connect nodemcu to internet
Test Data:	Nodemcu and wifi credentials
Steps:	• Connect nodemcu to lptop • Verify credentials • Open run tab • Check nodemcu status
Expected Result:	nodemcu connected to internet succesfully
Actual Result:	nodemcu connected to internet successfully
Status:	Pass

Appendixes:

Appendix 1: Final Documentation Format

Typographical Format and Binding

Page Format:

Page size:	A4
Top margin:	1.00 inch
Bottom margin:	1.00 inch
Left margin:	1.25 inch
Right margin:	1.00 inch

Page numbering:
Bottom right - part of the footnote
Title page not numbered
All other pages before the page of chapter one numbered in lower roman numerals (i, ii, iii,)
All other pages starting from first page of chapter one to last page of the report numbered in integers (1, 2, 3,)

Footer:
Each page shall have a footnote “Division of Science & Technology, University of Education, Lahore”
Left aligned
In case of long titles shorter versions should be used.
There shall be a line over the footnote.

Header:
Each page shall have a header “Project Name”
Left aligned
In case of long titles shorter versions should be used.
There shall be a line under the footnote.

Chapter Startup:
Each chapter shall be numbered as Chapter 1, Chapter 2, etc. The name of the chapter shall be written immediately below. Both shall be centered horizontally as well as vertically.
The actual chapter content shall start from the next page.7

Text:
Only one side of the paper shall be used.
The other side shall be blank.
When a report is opened the right side would contain text, figures, or tables and the left side would be blank.

Tables and Figures: Tables and figures shall be placed on one side only
Separate pages shall be used for figures and tables.
One page may contain more than one figure or table but text will not be combined or interlaced with figure or table.
Each table / figure shall be numbered.
For example "Table 1.2: Population distribution in Asia" or "Figure 3.2: Temperature distribution"
The table number or figure number shall be placed as normal text centered at the bottom of the table or figure or sideways with table / figure title coming on the opening side of the paper and note on the binding side.

Paragraph:

Single-spaced.
Line entered paragraph.
DONOT put indents at the beginning of the paragraph.
Left aligned or justified.

Text Format

Normal and plain text:

Font Type:	Times New Roman
Font Size:	12

Headings:

Chapter Heading:	Times New Roman Bold Size 16 Title Case normal
Heading 1:	Times New Roman Bold Size 14 Title Case normal
Heading 2:	Times New Roman Bold Size 12 Title Case normal
Heading 3:	Times New Roman Bold Size 12 Title Case italic

Sections and Subsections

In case of sections and subsections follow this format:

```
1       Section
1.1     Sub Section
1.1.1   Nested Sub Section
            A
            B
                i
                ii
```

The subsequent reference to a any section shall be made using the section and its number. For example **section 2.1.3** means chapter 2 section 1 subsection 3.

Mathematical Equations

The following numbering scheme should be used to number the equations:

$$f(x) = x+3 \qquad \text{(XX:YY)}$$

Where XX is the chapter number and YY is the sequence number of that equation in that chapter.

If an equation is previously quoted in an earlier chapter, say as equation 4:5 and need to be re-quoted in chapter 5, its number will remain as equation 4:5.

References

References are to be placed in square brackets and interlaced in the text. For example "A comprehensive detail of how to prevent accidents and losses caused by technology can be found in the literature [1]. A project report / thesis cannot be accepted without proper references. The references shall be quoted in the following format:

The articles from journals, books, and magazines are written as:

[1] Abe, M., S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. *Journal of the Acoustical Society of Japan*, April 1990, E-11 pp 71-76.

[2] Hermansky, H. Perceptual linear predictive (PLP) analysis for speech. *Journal of the Acoustical Society of America*, January 1990, pp 1738-1752.

The books are written as:

[1] Nancy G. Leveson, Safeware System Safety and Computers, A guide to preventing accidents and losses caused by technology, Addison-Wesley Publishing Company, Inc. America, 1995.

[2] Richard R. Brooks, S. S. Iyengar, *Multi-Sensor Fusion Fundamentals and Applications with Software*, The Prentice-Hall Inc. London, 1998.

Binding

All reports shall be bounded with an appropriate print on the backbone.

Two copies should be submitted.

Color of the binding:

• BSc project / thesis reports:	black
• MSc project / thesis reports:	blue

Contents of the CD Attached

All reports / theses must accompany a CD whose contents will have the following:

Top-level directories:

Doc All documents related to the project

- Instructions how to access the CD to the point to running the project
- All reports already submitted
- The final project report in thesis form
- Installation instructions
- Trouble shooting instructions in case of problems
- User manual
- Research material including URLs
- Papers consulted / referred to
- Slides of the presentations

Source All source files that will be needed to compile the project.
Further subdirectories can be used.
This must include sample data files as well.

Project The running project including sample data files as well as sample output.
This should be in a form that if copied to a machine runs without errors.
This may an exe file of an entire project, an installer depending on the project or simply a running project.

You can have sub directories with appropriate names.

Length

The length of your dissertation depends on the type of project you have selected. An excellent dissertation will often be brief but effective (its author will have said a lot in a small amount of space). Voluminous data can be submitted electronically on CD.