INDOOR DISASTER ANALYZER

By

NC Umer Farooq
PC M.Abubakar
PC Muhammad

Submitted to Faculty of Department of Computer Software Engineering, National University of Sciences and Technology, Islamabad, in partial fulfilment for the requirements of a B.E Degree in Computer Software Engineering

June, 2018

In the name of Allah, the Most Beneficent, the Most Merciful

ABSTRACT

Indoor Disaster Analyzer!

In minor to moderate natural and man-made disasters, such as earthquakes and fires, people may be trapped inside buildings and hurt by the disaster. Considering that trapped victims may be severely injured or unconscious, there is a high demand by emergency responders to get information on the locations and physical statuses of trapped victims inside a building during a disaster.

In this paper, a smartphone controlled robotic vehicle, indoor disaster emergency response assistance system, named Indoor Disaster Analyzer, is presented. The system is comprised of two subsystems: a Robotic vehicle and a smartphone. Robotic vehicle will be sent to disastrous environment to get live feed and other environmental data via different sensors installed on vehicle. Image processing algorithm Convolutional Neural Network (CNN) is employed to classify victim bodies in disastrous environment.

An analyzed summary of disastrous environment will be presented on android device enabling rescue team to plan their rescue operations in well-organized way.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis – Indoor Disaster Analyzer carried out by Umer Farooq, Muhammad Abubakar, and Muhammad under supervision of Dr. Naima Iltaf for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved By

_____

Assistant Professor Dr. Naima Iltaf

Computer Software Department

Military College of Signals (NUST)

Dated: _____

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To great Muslim Scientists and Researchers of the history whose accomplishments have greatly contributed to the progress of this modern world and who are of great inspiration to us to excel in the fields of Science and Technology.

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH Almighty. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are also grateful to our parents and family and well-wishers for their admirable support and their critical reviews. We would like to thank our supervisor. Dr. Naima Iltaf, for her continuous guidance and motivation throughout the course of our project. Without their help we would have not been able to accomplish anything.

Table of Contents

Chapter#01

System Introduction

Chapter 1. Introduction

1.1 Overview

The project is intended to collect information of disastrous environment, so If disaster occurs at any place some people may evacuate and some may not, so robot will be used to detect the human body in disasteristic environment and if it detects anybody in that environment, robot will send relative coordinates to android smartphone. So rescue team will get the information and plan their rescue operation according to received information from robot.

1.2 Problem Statement

When natural disasters are occurred, a lot of injured victims are found and need medical treatment as soon as possible. However, looking for the victims is not easy due to destroyed infrastructures such as partially or completely buildings, roads, coal mines etc. Also a lot of rescue team members face death every year due to collapse of buildings while carrying out their rescue operations.

1.3 Approach

Indoor Disaster Analyzer system offers an alternative to deal with the above stated problem statement. An android controlled robotic vehicle will be sent to disastrous environment to get live feed and other environmental data. Image processing algorithm Convolutional Neural Network (CNN) is employed to classify victim bodies in disastrous environment. An analyzed summary of disastrous environment will be presented on android device for planning the rescue operations.

1.4 Scope

Robotic Vehicle will be used to analyze the disasteristic environment before resuce operation so that rescue team will plan their rescue operation according to the information provided by the robot. Using this technique rescue team will get maximum output in minimum time.

1.5 Objectives

The main objective of this software system is:

- To develop a device which will assist rescue team to carry out their rescue operations in disastrous buildings more efficiently and effectively.
- To mitigate life threat to rescue team members in disastrous environment.

1.6 Deliverables

Sr	Tasks	Deliverables
1	Literature Review	Literature Survey and Feasibility Analysis
2	Requirements Specification	Software Requirements Specification document (SRS)
3	Detailed Design	Software Design Specification document (SDS)
4	Implementation	Project demonstration
5	Testing	Evaluation plan and test document
6	Training	Deployment plan
7	Deployment	Complete application with necessary documentation

1.7 Overview of document

1.7.1 Purpose

This document covers the software requirement specifications for project Indoor Disaster Analyzer! The idea of the project Indoor Disaster Analyzer! Is to ensure that the rescue team who intend to rescue the human bodies in disastrous to know that is any human body is present in environment by using robotic vehicle. This document is meant to outline the features and requirements of Indoor Disaster Analyzer, to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

1.7.2 Document Conventions

Headings

Heading are prioritized in a numbered fashion, the highest priority heading having a single digit and subsequent headings having more numbers, per their level.

All the main headings are titled as follows: single digit number followed by a dot and the name of the section (All bold Calibri Light, size 18, Centered).

All second level sub headings for every sub section have the same number as their respective main heading, followed by one dot and subsequent sub heading number followed by name of the sub section (All bold Calibri Light, size 16).

Further sub headings, i.e. level three and below, follow the same rules as above for numbering and naming, but different for font (All bold Calibri, size 14).

Figures

All figures in this document have captions, and are numbered. Context and flow diagrams are based on UML standards.

Reference

All references in this document are provided where necessary, however where not present, the meaning is self-explanatory. All ambiguous terms have been clarified in the glossary at the end of this document.

Links to web pages

All links have been provided with underlined font, the title of the web page or e-book is written at the top of the link and the title may be searched on google to pinpoint to the exact address.

Basic Text

All other basic text appears in regular, size 12 Calibri Light. Every paragraph explains one type of idea.

1.7.3 Intended Audience and Reading Suggestions

This requirements document contains general information about Indoor Disaster Analyzer, use cases, functions, features and special technologies. It describes in detail all that and system needs to do in order to correctly convey the available parking slot. Functional and non-functional requirements are addressed separately. System features with use cases and constraints are discussed in detail. System interfaces are also discussed in detail.

For better understanding, the document is divided into chapters

- In chapter 1 & 2, an overall description of Application is provided. First product perspective is presented with product features and main functions. A complete research in project's domain is documented as a part of chapter 2. Then follow user classes and characteristics, operating environments that Application supports as well as design and implementation constrains. After all that, user documentation is presented and will provide you with more details about each feature's technology.

- In chapter 3 & 4 most important features are presented with detailed description, use cases and requirements.
- In chapter 5 testing and evaluation of the parking system is documented with detailed test cases for unit and integration testing.

This document is intended for:

- **Developers:** (Project Group)
  To be sure that they are developing the right project that fulfills the requirements provided in this document.
- **Testers:** (Project Group, Supervisor)
  To have an exact list of the features and functions that must respond according to requirements.
- **Users:** (Public/Drivers)
  To get familiar with the idea of the project and how to use/respond in failure situations and suggest other features that would make it even more functional.
- **Project Supervisor:** (Dr. Naima Iltaf)
  This document will be used by the project supervisor to check and guide the group about the understanding and implementation of the requirements properly and completely during the development lifecycle.
- **Project Evaluators:** (CSE Dept. MCS)
  To know the scope of the project and evaluate the project throughout the development for grading.

References

More about application can be retrieved from project development team.
Umer Farooq
BESE 20 MCS NUST
umerfarooq492@gmail.com
Group Leader – Dev. Team

Chapter#02

Literature review

Chapter 2. Literature review

2.0 Challenges during rescue operation in collapsed buildings

A tragic disaster occurred on April 24, 2013, in Bangladesh, when a nine storied building in a suburban area collapsed and killed 1115 people and injured many more. In an interview, one of the rescue team member described the difficulties faced by them while carrying out the rescue operations as "Crowd management was crucial; there were thousands of people who came to see what was going on; many of them were relatives of the victims and they were waiting with apprehension to see if their relatives were alive and needed to be rescued or not. Moreover, another mass of people came to help the rescuers and rescued persons by providing food, medicine, and other necessities. Bangladesh Armed forces, Civil Aviation, Rapid Action Battalion, and Police played a tremendous role in managing the crowd. However, there was always a risk of mishap or an accident at any time. The majority of the public did not have any idea or training on how to rescue people from inside the building. Rescue operations had proven to be thoroughly difficult. The pillars and ceilings had collapsed at so many angles and in such precarious ways that any wrong move could cause a fresh tragedy. The army had brought in huge cranes to pull the concrete blocks apart. But these could not be used earlier for the fear of further collapse [1]."

2.1 Existing Related System's Literature Review

A related system is already implemented in US but that system basically outdoor disaster analyzer and used android mobile controlled drone technology. According to literature "Natural disasters such as earthquakes and volcanic events destroy infrastructure such as roads and buildings, making search and rescue very challenging. In such cases, unmanned aerial vehicles (UAVs, or drones) can be used to reach victims. Increasingly people are wearing or implanting biometric and medical sensors with wireless interfaces. For example a wrist-band, heart monitor, foot-mounted accelerometer and a smart phone could be interconnected wirelessly to form a Wireless Body Area Network (WBAN). Collectively these can measure and process vital signals to help infer the physiological and psychological state of the wearer. This can be used to provide vital health and fitness information and for many other applications. The future ubiquity of such WBANs creates an opportunity to improve response to wide-scale incidents such as earthquakes, tsunamis, landslides, or building collapse where search, rescue and paramedic assistance are imperatives [2]."

2.2 Localization of victim body

About discovery of victim body, literature says “the presence of a victim can be discovered by detection of their wireless signature. The discovery problem is particularly difficult, because typically WBAN devices are low power, as they are battery operated and designed to only operate over short ranges. The victim may be inside structures such as buildings, or buried under rubble. Furthermore, there are many potential signal types which could be used, corresponding to different standards (e.g. Wi-Fi, Bluetooth, 802.15, etc.) and/or proprietary transmission protocols (many sport-watch manufacturers use their own signal designs). Thus, we choose to base victim discovery upon detection of an arbitrary cyclostationary signal.”

2.3 Conclusion

Unfortunately, no indoor disaster analyzer system existed prior to our work. Only related system we found was an outdoor disaster analyzer that used mobile controlled drone. Since when a building collapses, the pillars and ceilings are collapsed at so many angles that no proper space is available for drone to fly in collapsed building and analyze the disaster. Also this system hadn’t used image processing technique for detection of victim bodies

Chapter#03.

Software Requirement Specification

Chapter 3. Software Requirement Specification

3.1 Introduction

This chapter translates the project objectives in to functionalities by specifying the requirements of project. First, project is explored in detail by discussing perspective, functions and operating environment. Then approach for tackling the problem statement is explored by stating the function of system. These functions are then converted into precisely written Functional and Non-functional requirements. Moreover, this chapter also states the design and implementation constraints that should be kept in mind.

3.2 Overall Description

3.2.1 Product Perspective

The project is intended to collect information of disastrous environment, so If disaster occurs at any place some people may evacuate and some may not, so robot will be used to detect the human body in disasteristic environment and if it detects anybody in that environment, robot will send relative coordinates to android smartphone. So rescue team will get the information and plan their rescue operation according to received information from robot.

3.2.2 Product Functions

Main features of the product are given below:

- Robotic vehicle will be navigated via android device
- Network connection between Raspberry Pi with Android device through Wi-Fi.
- Android application will display the live feed from Pi camera.
- Android application will display the relative coordinates of human bodies.
- Android application will display environment Humidity level, Temperature, Carbon Dioxide level using sensors connected on Arduino mega.

3.3 User Classes and Characteristics

Following are user classes and their brief description.

3.3.1 Tester (occasional user)

Tester will use this project to check for finding bugs. They will also use the project to check if it's in accordance to the Software Requirements Specification document.

3.3.2 Project Supervisor (occasional user)

Project supervisor will use the product to evaluate. They will use this product to find the accuracy and error in the output.

3.3.3 Rescue Team (Regular user)

Rescue team will send robotic vehicle in the disastrous environment and analyze the environment on android device and plan their rescue operation accordingly.

3.3.4 Operating environment:

Required operating environment for the application is listed below.

Hardware Requirements:

- Android device: To install software for communication and visualization of data.
- Raspberry PI: To connect smartphone through Wi-Fi along with camera to get live feed. It will also be connected with Arduino.
- Arduino Mega: To connect different sensors (IR Sensor, Humidity Sensor, Gas Sensor and Temperature Sensor).
- Vehicle: To carry Raspberry Pi, Arduino and Camera.
- Camera: To get live feed.
- Sensors: To get different environment data.

Software Requirements:

- Operating System: Android
- Image Processing API: OpenCV

3.3.5Design and Implementation Constraints:

Constraints of the product are given below:

Indoor Disaster Analyzer!

- Robotic vehicle will not work without Wi-Fi connection.
- Robotic vehicle will detect only one object (Human being) at a time.
- Robotic vehicle will work till its battery life.
- Robotic vehicle will work only in fired places.
- Robotic vehicle will detect objects (Human being) from at least 1m distance.

3.3.6 User Documentation

A user manual will be provided to the users in which separate instructions will be given according to the particular user i.e. Regular users, developers and testers. It will include the details of the system's working. Help documents will also be a part of the system.

The project report will also be available for the users which will highlight the system features, working and procedures.

3.3.7 Assumptions and Dependencies

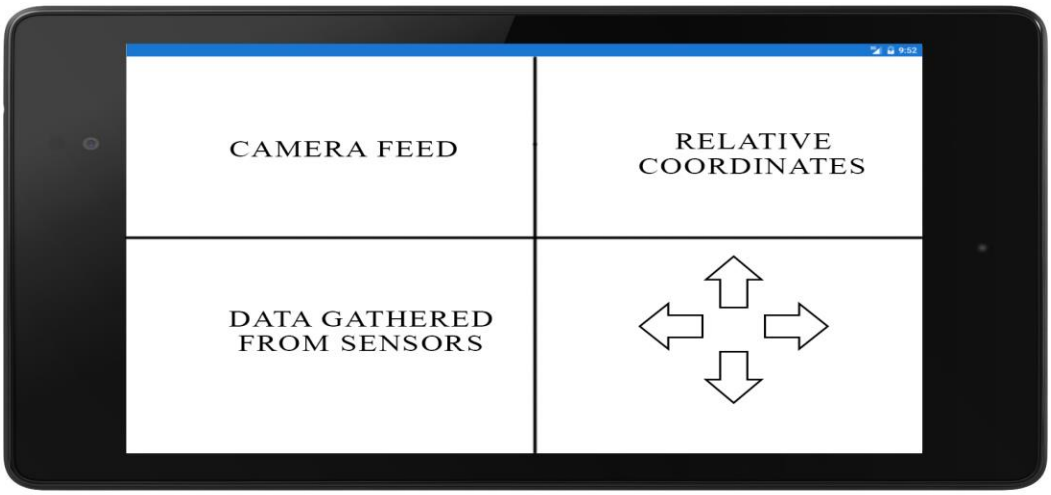
- Overall performance of the product will depend on the hardware infrastructure and network speed.
- User must know about the Interface for the better performance of the product.
- Limitations of the product must be kept in mind by the user.

3.4 External Interface Requirements

User Interfaces

Responsive graphical user interfaces must be provided to user to work with the application. User will see the following screen.

Expected screen:



Hardware Interfaces

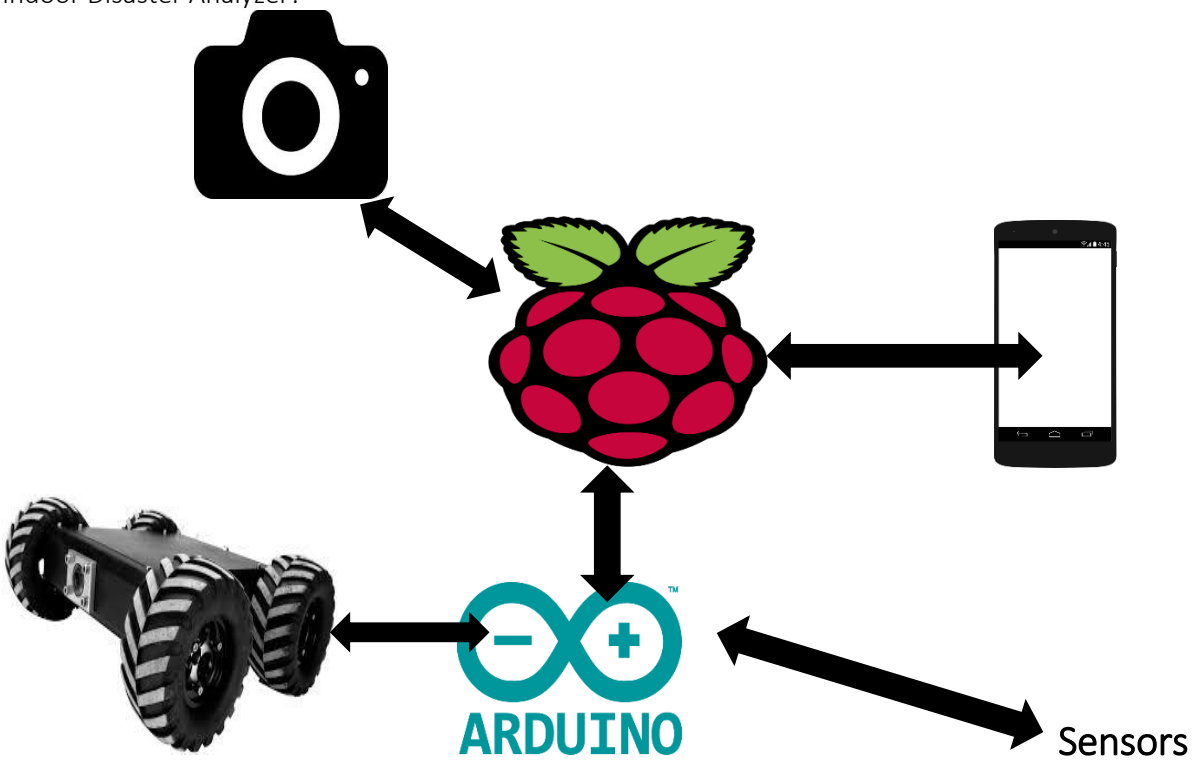
- Data will be gathered from different sensors.
- Live feed will be collected from camera integrated on Raspberry Pi.

Software Interfaces

- Android OS for running the android application.
- Raspbian OS which will be installed on the Raspberry PI.

Communications Interfaces

- Wi-Fi will be used as medium of communication between android devices and raspberry pi



3.4 System Features

System features are organized by use cases and functional hierarchy so that the main functions of the system will be understandable.

3.4.1 Wi-Fi Connection

This feature allows robotic vehicle and Android device to establish a connection.

Description

When rescue team want to send a robotic vehicle inside disastrous environment. They must have to establish a connection between android device and vehicle.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User will open the application.

2. Application will create network hotspot.
3. Raspberry Pi will find network hotspot and connect with application.

Alternative Data Flows

Alternative Data Flow 1

1. User opens application and is not connected to any network.
2. Error message is displayed to user.

Alternative Data Flow 2

1. On connection lost application will display an error message.
2. Application will recreate connection.

Alternative Data Flow 2

1. User's device is not connected with raspberry pi.
2. User is displayed with an error message.

Functional Requirements

1. Application must be installed and connected to a network.
2. Raspberry pi and user's smartphone shall be visible to each other for network connection.
3. Raspberry pi and user's android smartphone shall be able to connect with each other.

3.4.2 Navigation

This feature provides navigation/movement of robotic vehicle.

Description

Rescue team will control the navigation of robotic vehicle. When user press navigation buttons application will send navigation commands to robotic vehicle.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User press navigation button and commands will send to Raspberry Pi.
2. Raspberry Pi will commands to vehicle's rotating motor.

Alternative Data Flows

Alternative Data Flow 1

1. User send navigation command and robotic vehicle is not moving.
2. Error message will display to user.

Functional Requirements

1. Application must be installed and connected to a network.
2. User shall be able to navigate robotic vehicle through application.

3.4.3 Camera Feed and Coordinates

This feature will provide live camera feed of disastrous environment and relative coordinates of objects (human beings).

Description

User will be able to see live camera feed from camera installed on raspberry pi. Camera feed will be shown on android application. Android application will draw relative coordinates according to objects (human beings) detection.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User will navigate the robotic vehicle.
2. Camera will be able to capture live feed and send it to mobile application.
3. Live feed will be processed and relative coordinates will be displayed on mobile application.

Alternative Data Flows

Alternative Data Flow 1

1. User opens the app and start navigating but camera doesn't respond.
2. Error message will be displayed.

Functional Requirements

1. Application must be installed and connected to the network.
2. User shall be able to see live feed.
3. User shall be able to see objects (Human Beings) relative coordinates.

3.4.4 Displaying Data from Sensors

This feature will provide gathered data from sensors and display it on mobile application.

Description

User must be able see information (Temperature, Humidity, CO_2 level) from different sensors integrated on Arduino Mega.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. Raspberry pi will get data from sensors integrated on Arduino mega.
2. Raspberry pi will send data to mobile application.
3. Mobile application will display the received data.

Alternative Data Flows

Alternative Data Flow 1

1. User opens the app and navigate the robotic vehicle and application is unable to display data.
2. Error message will be displayed.

Alternative Data Flow 2

Indoor Disaster Analyzer!

1. Sensors are unable to get data.
2. Error message will be displayed.

Functional Requirements

1. System shall be able to get data from sensors.
2. User shall be able to see the gathered data.

3.5 Other Nonfunctional Requirements

Performance Requirements

Application shall run on a minimal amount of memory and take up a small amount of disk space after install. Depending on the performance of the user's device or amount of data returned from the robotic vehicle, the communication might slow down the Application.

Security Requirements

Application should not need any additional information other than the collected data from the robotic vehicle. There are no connections to third party devices or severs so no data will be sent or received or used in any way.

Software Quality Attributes

- Reliability

  The application should run perfectly with all the features mentioned above available. It should be tested and debugged completely. All exceptions should be well handled.

- User Friendliness/Simplicity

  Application should have a graphical user interface with user friendly menu and options.

- Ease of Access

  The user shall be able to install the application with just one click/tap from their local server.

Chapter#04

Software Design & Implementation

Chapter 4. Software Design & Implementation

4.1 Introduction

World is moving towards automation and using Internet of Things for Automating their daily tasks. Similarly, some tasks in disastrous environment are also regularly performed such as disaster occurs at any place some people may evacuate and some may not, so robot will be used to detect the human body in disasteristic environment and if it detects anybody in that environment, robot will send relative coordinates to android smartphone. So rescue team will get the information and plan their rescue operation according to received information from robot.

4.1.1 Purpose

This document includes software design for **Indoor Disaster Analyzer** release number 1.0. This document specifies the detailed architectural design of **Indoor Disaster Analyzer** which is being developed. This document will act as a guideline for developers and all the other stakeholders throughout the development. Document includes classes and their inter-relationships, use cases with detailed descriptions, sequence diagrams, activity diagrams and various others.

4.2 Project Scope

Robotic Vehicle will be used to analyze the disasteristic environment before resuce operation so that rescue team will plan their rescue operation according to the information provided by the robot. Using this technique rescue team will get maximum output in minimum time

4.3 Definitions

OS: Operating System

API: Application Programming Interface

Raspbian OS: A flavor of Linux OS used in raspberry pi

CO_2: Carbon Dioxide

CO: Carbon Monoxide

4.4 References

The reading material and some guides are available in the form of web links.

1. Sequence Diagram
   https://en.wikipedia.org/wiki/Sequence_diagram
2. Block Diagram
   https://en.wikipedia.org/wiki/Block_diagram
3. Use case modeling guidelines
   http://ieeexplore.ieee.org/document/787548/?reload=true&arnumber=787548

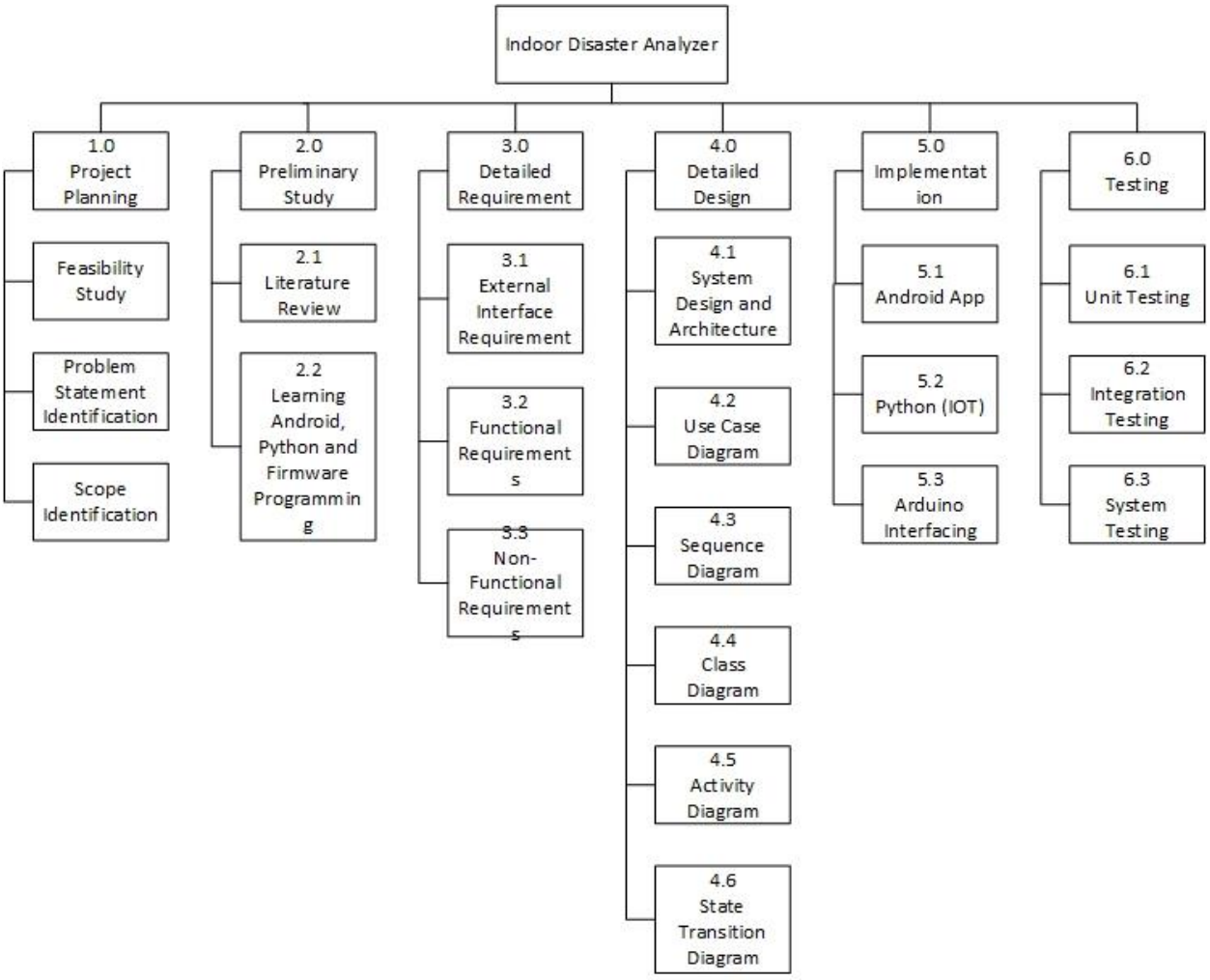
4.5 Overview of Document

This document is about the detailed architectural design of **Indoor Disaster Analyzer** powered by IoT. For simplicity, the document is divided into various sections. Section 1 introduces the document and provides overview for executive purposes. Section 2 includes detailed description of the system with various diagrams and charts. This section includes all the architectural details of system under development. Section 3 describes all the modules and components of the system in detail one by one. Section 4 compares this product to various other similar products available in market. Section 5 throws light on the design decisions and tradeoffs.

This design document contains detailed information about **Indoor Disaster Analyzer** powered by IoT. All modules, use cases, functions, features and special technologies and their inter-relationship is discussed in detail and also assisted with diagrams where required.

This document is intended for developers, testers, users, documentation writers, project supervisor, and project evaluators. A copy of this document will be made available to all stakeholders.

4.6 Work Breakdown Structure



5 System Architecture Description

This section provides detailed system architecture of **Indoor Disaster Analyzer** powered by IoT. Overview of system modules, their structure and relationships are described in this section. User interfaces and related issues are also discussed

5.1 Overview of Modules

This Indoor Disaster Analyzer System has following required modules. Here we give a brief overview of all these modules. Detailed descriptions of these modules are presented in section 3.

1. Communication:

   Wi-Fi will be used as medium of communication between android devices and raspberry pi.

2. Arduino Interfacing:

   This module will provide connection between Arduino to Raspberry Pi via Nanpy. Nanpy is a library that use Arduino as a slave, controlled by a master device (Raspberry Pi) where you run your scripts. The main purpose of Nanpy is making Arduino programming easier.

3. Camera Feed:

   This feature will provide live camera feed of disastrous environment. User will be able to see live camera feed from camera installed on raspberry pi. Camera feed will be shown on android application.

4. Display Gathered Data:

   This feature will provide gathered data from sensors and display it on mobile application. User must be able see information (Temperature, Humidity, CO_2 level) from different sensors integrated on Arduino Mega.

5. Display Processed Information(Graph/Map):

   This feature will provide a graph/map showing relative coordinates of objects (human beings). Image Processing will be done on live feed provided by camera, to detect human bodies and draw relative coordinates of detected objects (human beings).

6. Gather Data from Sensors(Arduino):

   Environmental parameters will be captured using various sensors such as temperature, humidity, CO_2 level sensors etc. These different sensors will be integrated on Arduino Mega.

7. Environment Analysis:

   Arduino will send gathered environmental data from different sensors to microprocessor (Raspberry Pi). Microprocessor will then analyze input data from Arduino. Based on this analysis, microprocessor will recommend/not recommend user to enter in disastrous environment.

8. Navigation:

   This feature provides navigation/movement of robotic vehicle. Rescue team will control the navigation of robotic vehicle. When user press navigation buttons application will send navigation commands to robotic vehicle.

9. User Interface:

   User interface is one of the ways to interact with application. It packages all those screens that are visible to user.

5.2 Structure and Relationships

This section covers the overall technical description of Indoor Disaster Analyzer. It shows the working of application in perspective of different point-of-views and also shows relationship between different components.

5.2.1 System Block Diagram

This diagram shows the higher-level description of the application. It shows all the modules of the system and their associations and flow of data between modules.

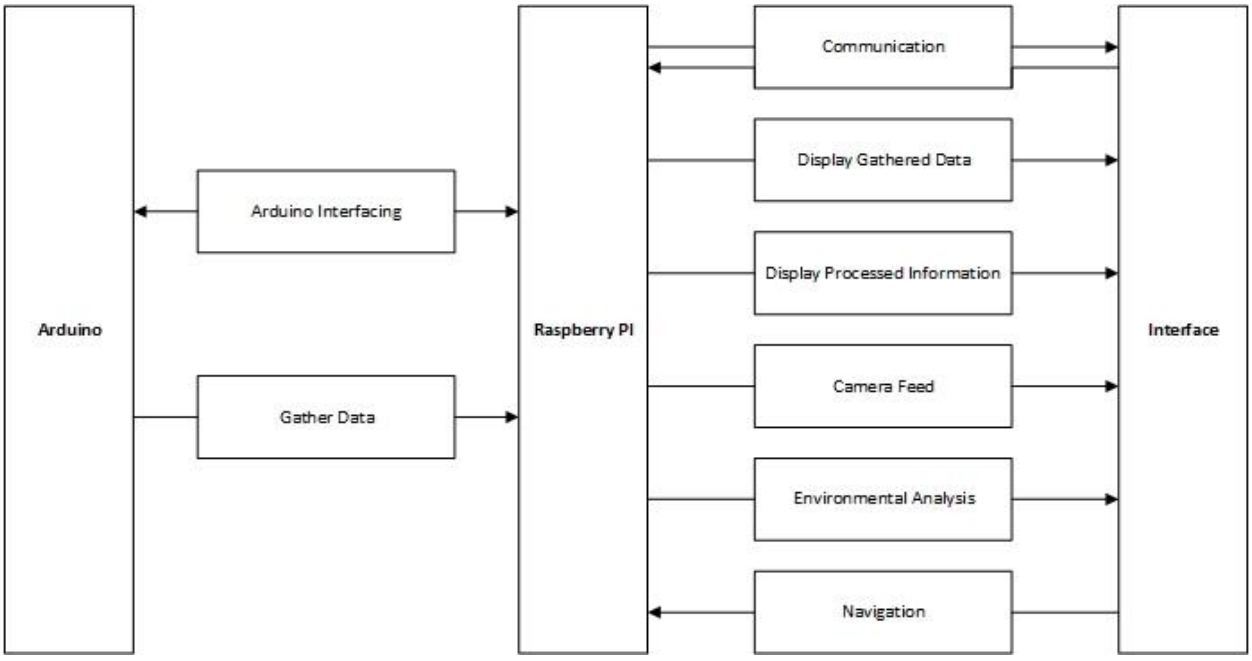


Fig 2.2.1.1 (System Block Diagram)

5.2.2 User View (Use case diagram)

Following diagram shows course of events that take place when an actor (user and other allowed interactions) interact with the system.

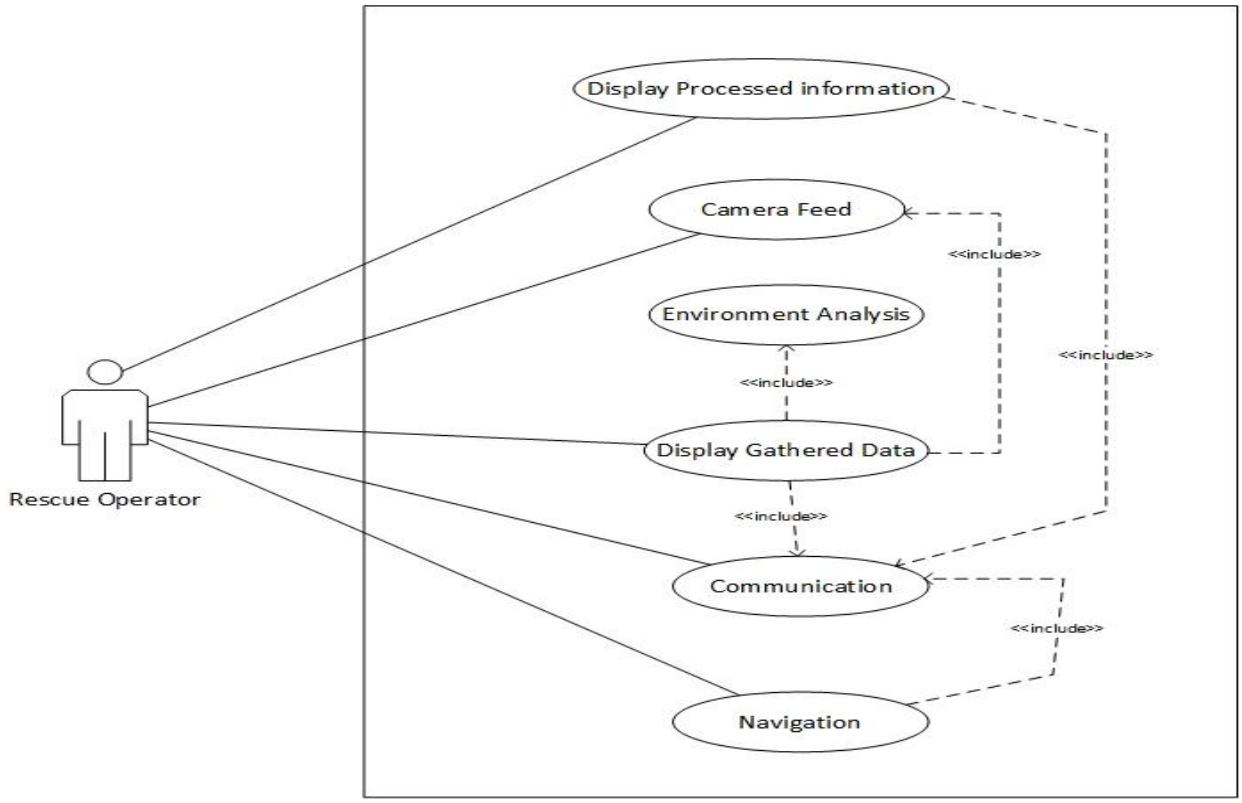


Fig 2.2.2.1 (Use Case Diagram)

5.2.2.1 Actors

- Rescue Operator

5.2.2.2 Use Cases

- Communication

Indoor Disaster Analyzer!

- Display Gathered Data
- Environment Analysis
- Navigation
- Camera Feed
- Display Processed Information

Use cases shown in the figure above are described below.

Use Case 1: Connection

Use Case Requirement: Wi-Fi will be used as medium of communication between android devices and raspberry pi. Wi-Fi (Raspberry pi) and android device are configured properly to run the application.
Use Case Paths: • Connection established. • Connection failed.
Normal Path: Connection Established
Preconditions: Wi-Fi SSID is visible and user entered correct password at android device.
Interactions: Entered password is verified.
Post conditions: Connection is established and user is notified about successful establishment of connection.
Exceptional Path: Connection Failed
Preconditions: Wi-Fi SSID is not visible or user entered incorrect password at android device.
Interaction: Entered password is not verified.
Post conditions: Connection is not established and user is notified about unsuccessful connection establishment.

Use Case 2: Display Gathered Data From sensors

Use Case Requirement: All sensors (temperature, humidity, CO_2 level sensors etc.) are working properly and are integrated on Arduino properly.
Use Case Paths: • Successfully gathered environmental data • Unsuccessful to gather environmental data
Normal Path: Successfully gathered environmental data
Preconditions: Sensors are working properly
Interactions: Environmental parameters are captured using various sensors such as temperature, humidity, CO_2 level sensors etc.
Post conditions: Arduino received sensors data
Exceptional Path: Unsuccessful to gather environmental data
Preconditions: Sensors are not working properly
Interaction: N/A
Post conditions: Pi didn't received sensors data

Use Case 3: Environment Analysis

Use Case Requirement: Arduino will send gathered environmental data from different sensors to microprocessor (Raspberry Pi). Microprocessor will then analyze input data from Arduino.
Use Case Paths: • Recommended / Not Recommend

• No input data received from Arduino
Normal Path: Recommended/ Not Recommend
Preconditions: Sensors are working properly
Interactions: Microprocessor (Raspberry Pi) will analyze input data provided by Arduino.
Post conditions: Microprocessor (Raspberry Pi) will send Recommendation/Not Recommendation notification to android device
Exceptional Path: No Input data from Arduino
Preconditions: Sensors are not working properly
Interaction: An error notification is sent to android device
Post conditions: No input data received by Arduino

Use Case 4: Navigation

Use Case Paths: • Navigation working properly. • Navigation not working properly.
Normal Path: Navigation working properly
Preconditions: Connection should be established between Pi and Android device
Interactions: A notification is sent to the raspberry pi (local network) for processing.
Post conditions: Robotic vehicle moves accordingly.
Exceptional Path: Navigation not working properly
Preconditions: The android device is not connected to raspberry Pi.
Interaction: An error notification is sent to the android device.

Post conditions: Android device is being notified that connection not established.

Use Case 5: Camera Feed

Use Case Requirement: User will be able to see live camera feed from camera installed on raspberry pi. Camera feed will be shown on android application.
Use Case Paths: • Camera connection successful. • Camera connection failed.
Normal Path: Camera Connection Successful
Preconditions: Camera working properly
Interactions: Live view of disastrous environment is display at android application
Post conditions: User able to see live view provided by camera
Exceptional Path: Camera Connection Failed
Preconditions: Camera not working properly
Interaction: An error notification of “camera not working properly” is displayed at android device
Post conditions: User not able to see live view of disastrous environment at android device

Use Case 6: Display Processed Information

Use Case Requirement: Camera is working properly and providing graph/map view of disastrous environment to user on android device.
Use Case Paths: • Graph/Map drawn. • No graph/map drawn.
Normal Path: Graph/Map drawn
Preconditions: Camera is working properly and capturing live view of disastrous environment
Interactions: Image Processing is done on live feed provided by camera, to detect human bodies and draw relative coordinates of detected objects (human beings).
Post conditions: Graph/Map showing relative coordinates of detected objects is displayed to user
Exceptional Path: No graph/map drawn
Preconditions: Camera is not working properly and not capturing live view of disastrous environment
Interaction: An error notification is displayed at android device and no Image Processing is done.
Post conditions: No Graph/Map shown to user on android device

5.2.3 Sequence Diagrams

Following sequence diagrams show the sequence of activities performed in application.

5.2.3.1 Communication

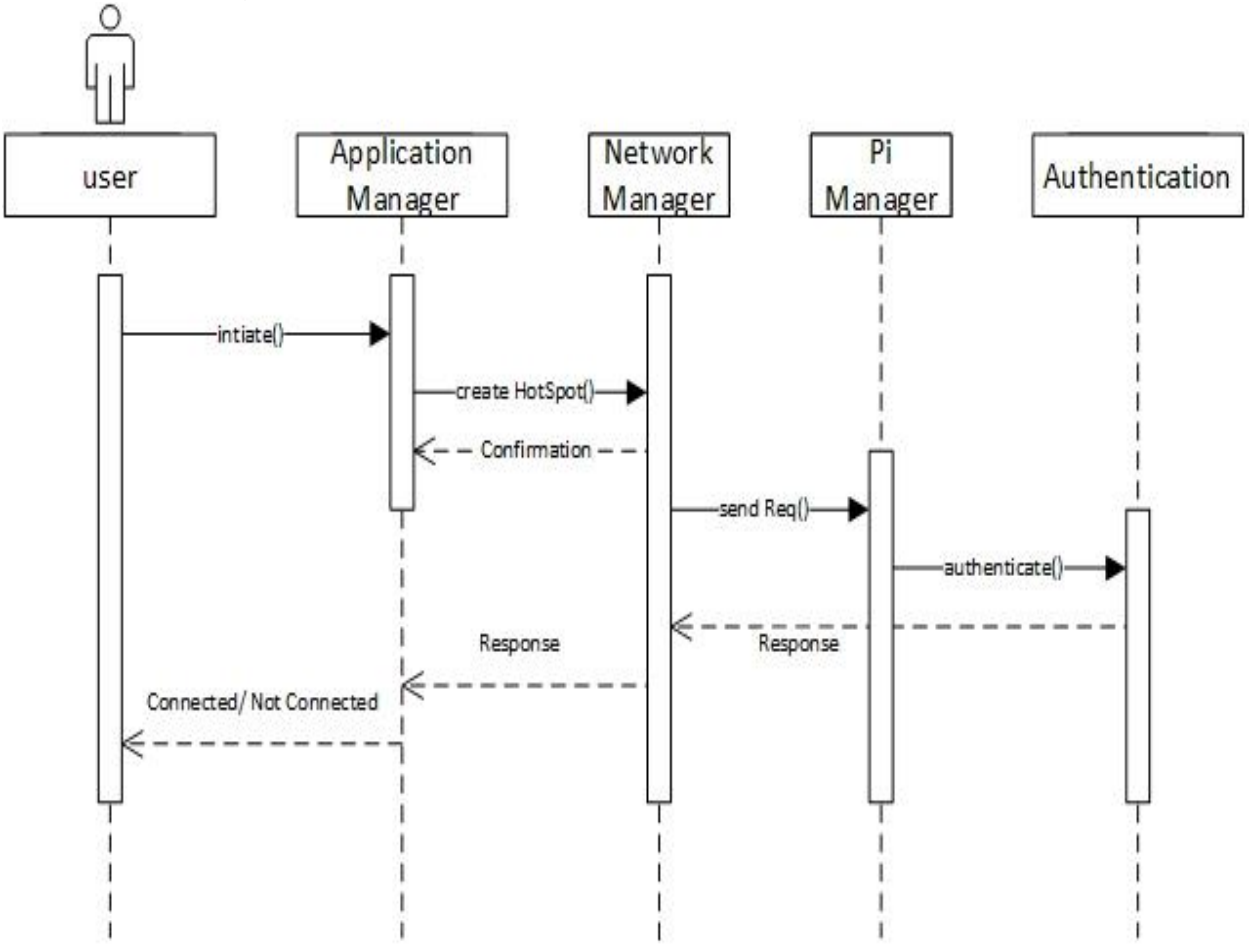


Fig 2.2.3.1 (Communication Sequence)

5.2.3.2 Display Gathered Data

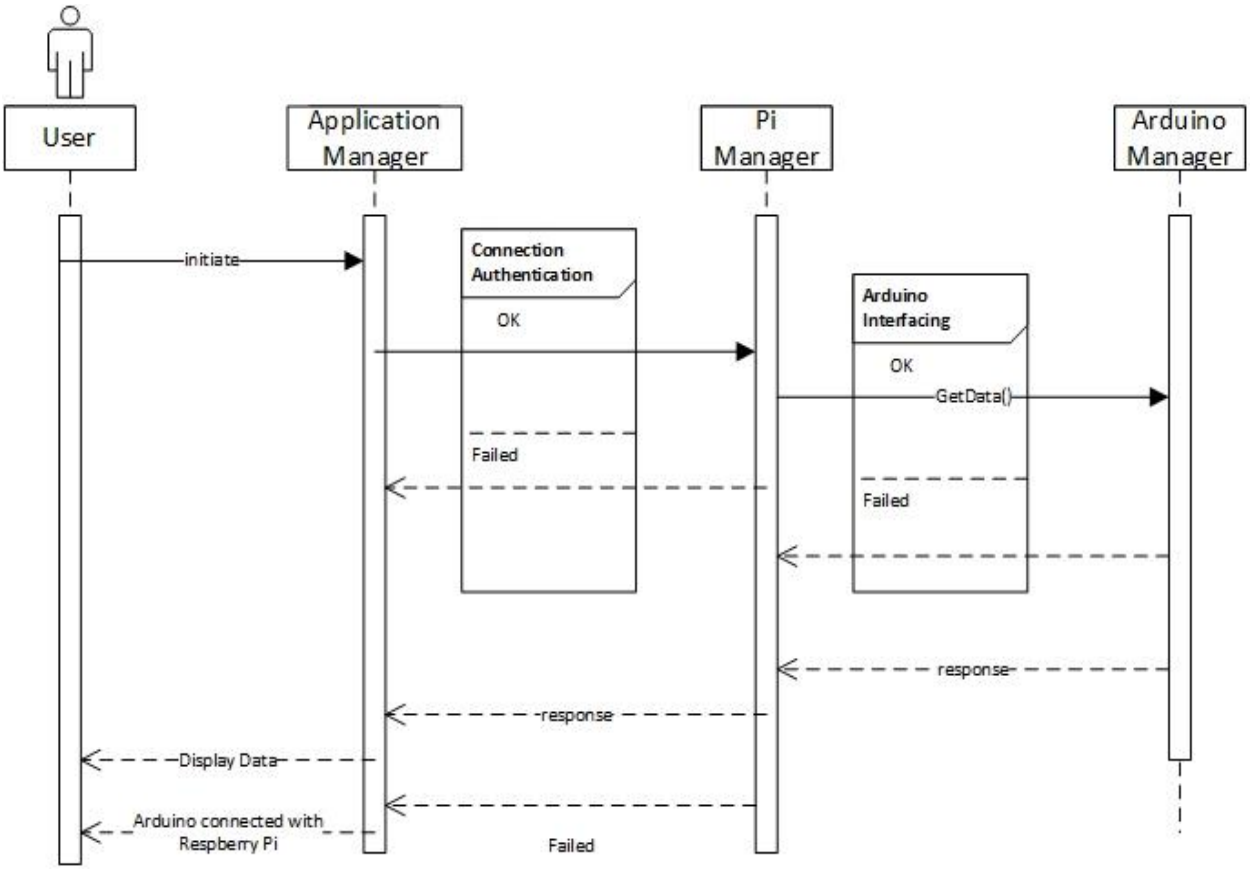


Fig 5.2.3.2 (Display Gathered Data Sequence)

5.2.3.3 Display Processed Information

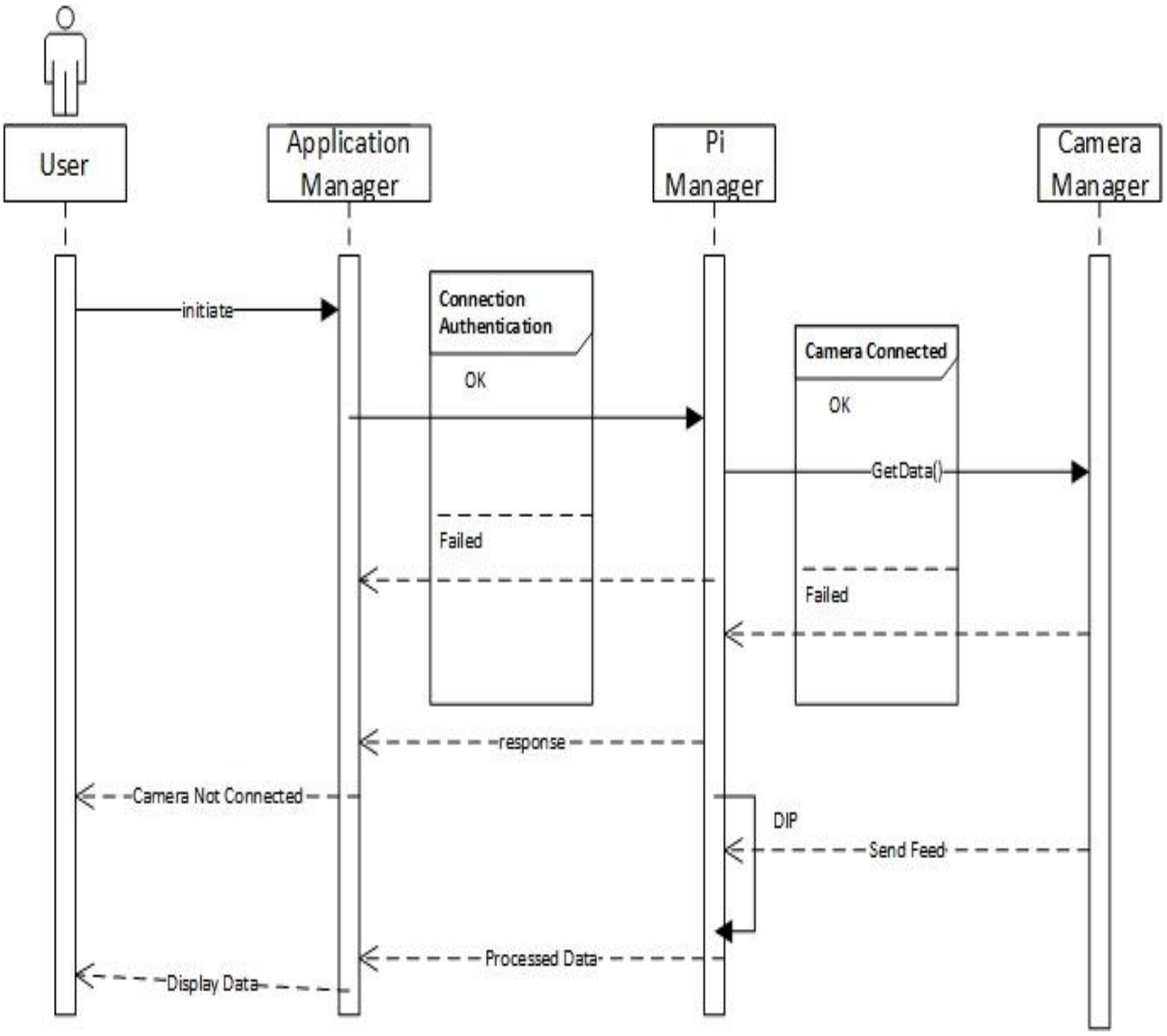


Fig 5.2.3.3 (Display Processed Information Sequence)

5.2.3.4 Environment Analysis

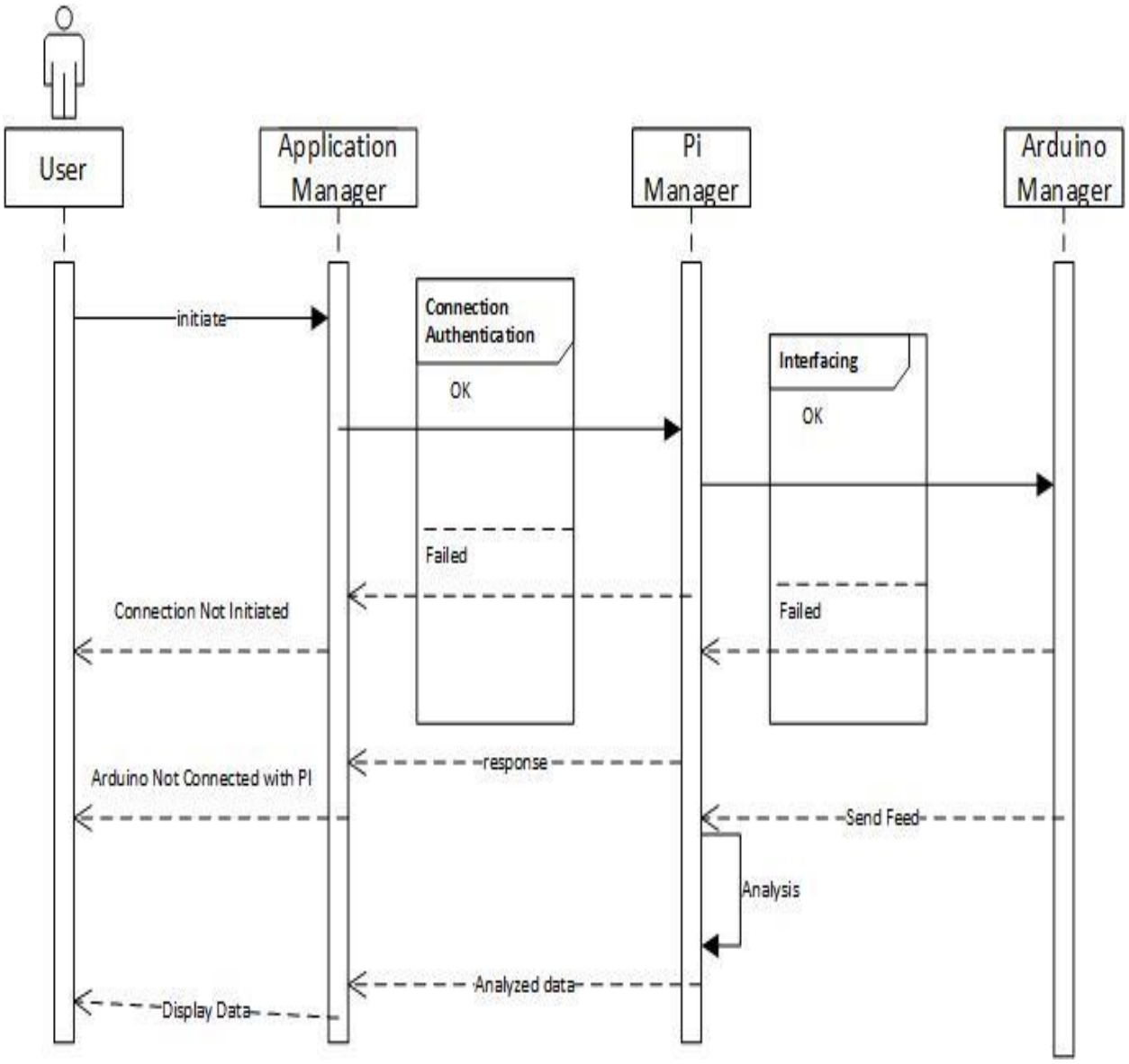


Fig 5.2.3.4 (Environment Analysis Sequence)

5.2.3.5 Camera Live Feed

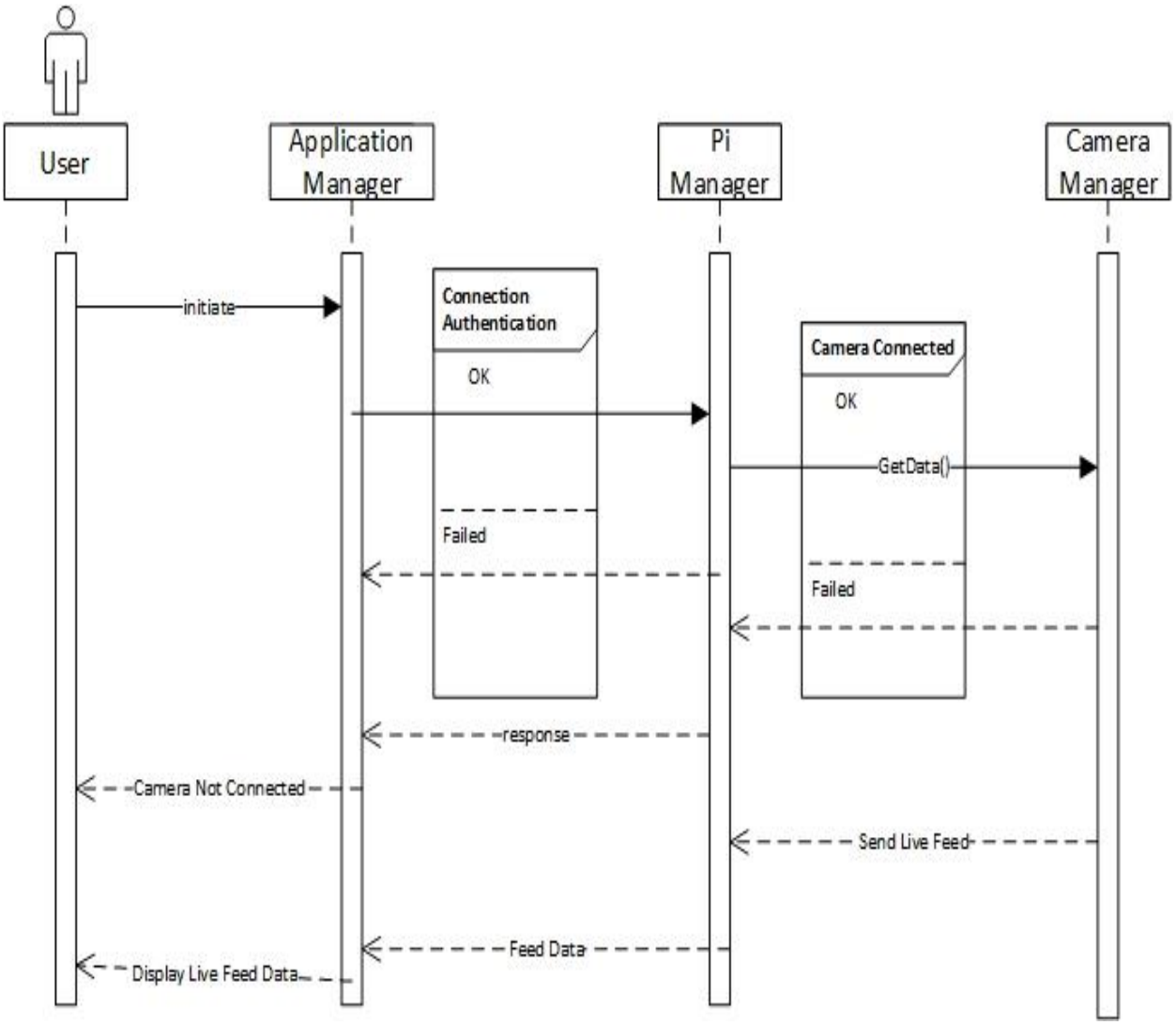


Fig 5.2.3.5 (Camera Live Feed Sequence)

5.2.3.6 Navigation

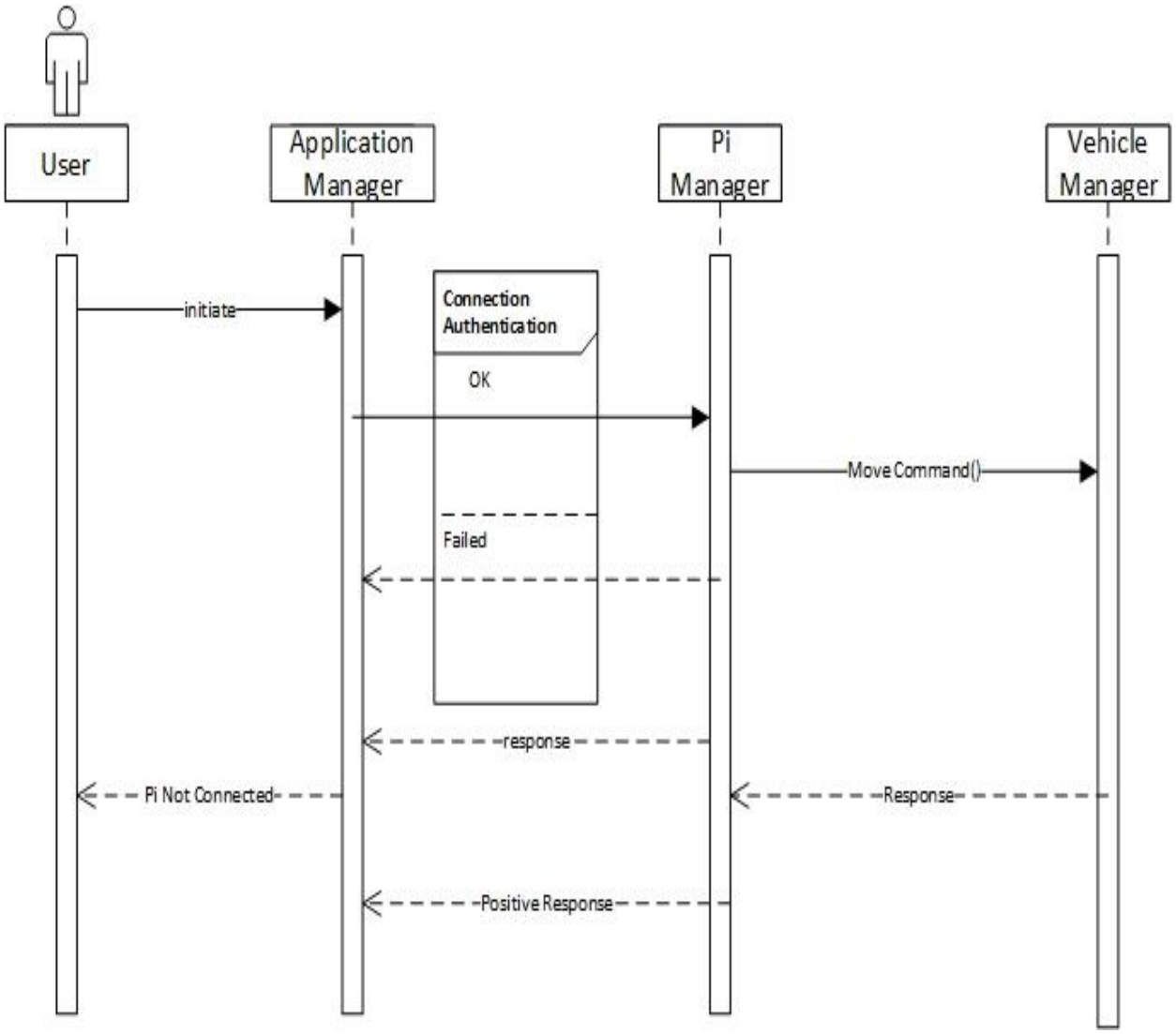


Fig 5.2.3.6 (Navigation Sequence)

5.2.4 Implementation View (Class Diagram)

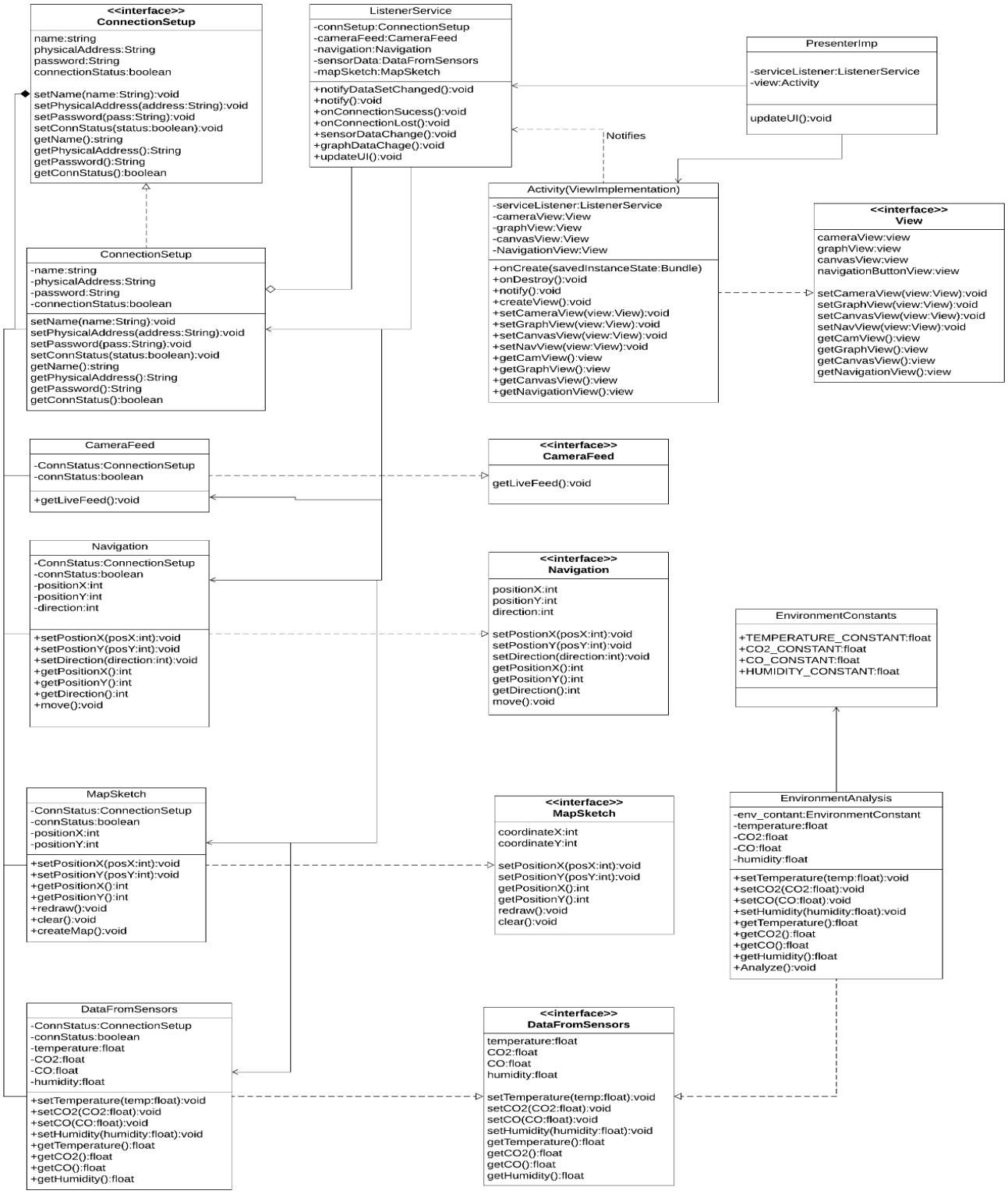


Fig 5.2.4.1 (Class Diagram for Android App)

Indoor Disaster Analyzer Class Name	Description
ConnectionSetup	This is the class that handles Connection services for a smartphone and raspberry Pi. This class will implement ConnectionSetup interface.
CameraFeed	This is a class that handles the live camera feed obtained from camera. This class will implement the CameraFeed interface
Navigation	This is a class that handles the navigation commands initiated by the rescue operator. This class will implement the Navigation Interface.
MapSketch	This is a class that will generate Maps on user interface screen by getting data from raspberry pi. This class will implement the MapSketch interface.
DataFromSensors	This class will manage all data obtained from sensors that are connected on Arduino. This class will implement the DataFromSensors interface.
EnvironmentAnalysis	This class will provide analysis on Data obtained from sensors and give recommendation for rescue operation. This class will implement the DataFromSensors interface.
EnvironmentConstants	This class contains all threshold constant values that will be used Environment Analysis.
Activity	This class will display whole view of user interface and notifies the listener service whenever user request change. This class will implement the View interface.

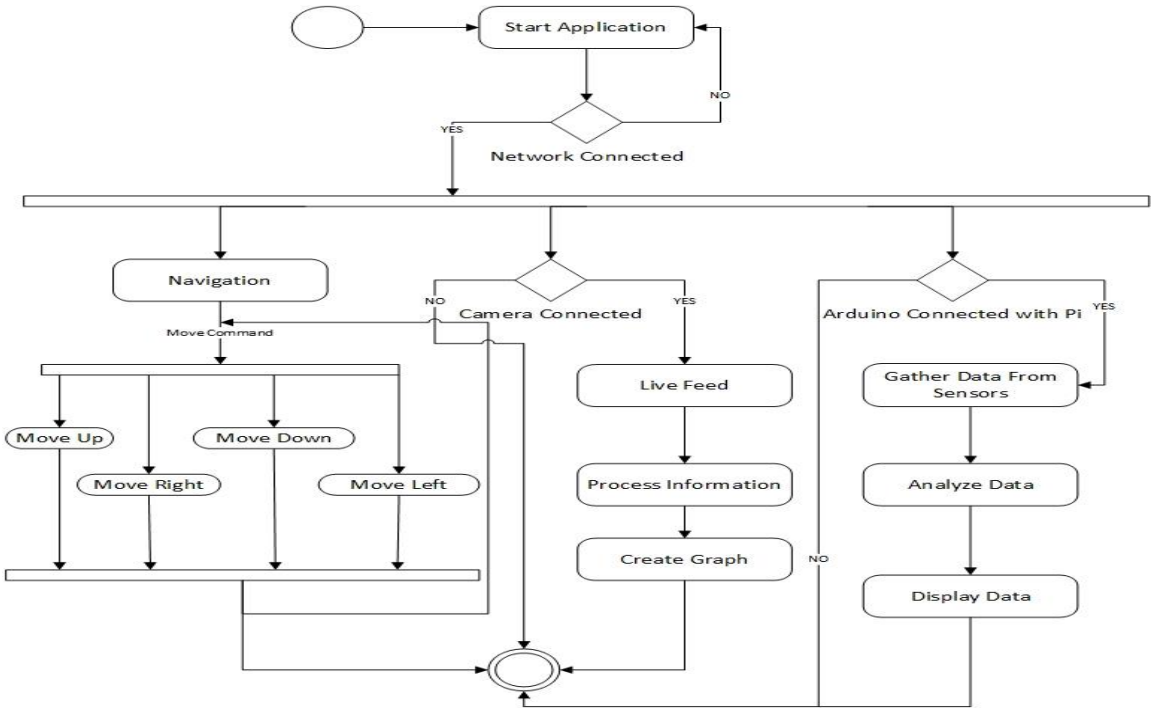
ListenerService	This class always listen all changes occurred in the system
PresenterImp	This is the concrete class which will commit the changes occurred in the model classes and view classes.

5.2.5 Dynamic View (Activity Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown.

5.2.5.1 Rescue Operator View

Fig 2.2.5.1.1 (Rescue Operator Based Activity Diagram



5.3 User Interfaces

Following diagrams show tentative user interfaces and screens for **Indoor Disaster Analyzer** powered by IoT application.

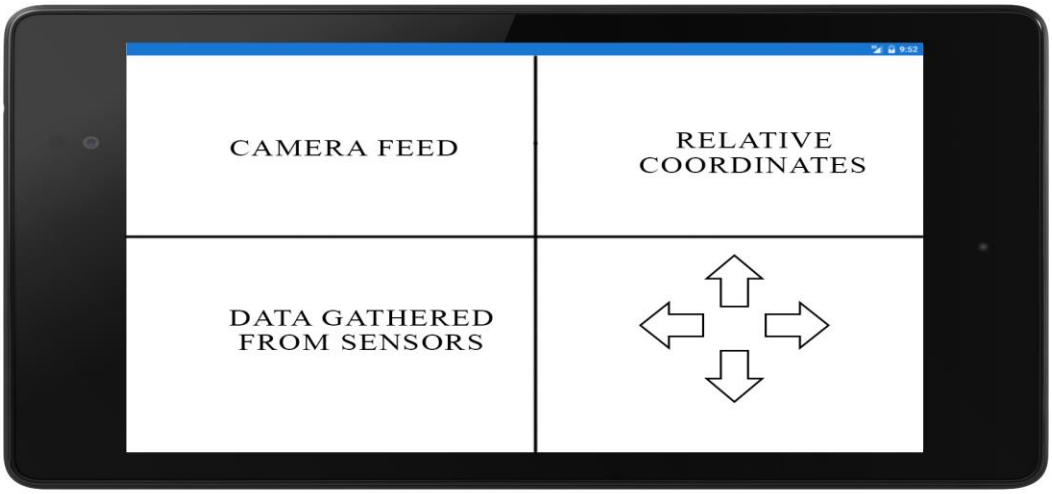


Fig 5.3 (Rescue Operator Screen)

6.0 Detailed Description of Components

This section describes in detail all components of Indoor Disaster Analyzer powered by IoT application.

6.1 Communication

Identification	Name: Communication
Type	Service

Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Communication** This feature allows robotic vehicle and Android device to establish a connection. **Description:** When rescue team want to send a robotic vehicle inside disastrous environment. They must have to establish a connection between android device and vehicle.
Function	This module provides connection between android device and robotic vehicle. Medium used for this module is Wi-Fi
Subordinates	It has one subordinate: 1. Wi-Fi Connection: Requirement 4.1 in SRS
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	This component has following interfaces: 1. **Communication Interface:** To establish connection between android device and camera on Raspberry pi
Resources	**Hardware:** Android smartphone, raspberry pi, Computer **Software:** Open source android libraries, Android SDKs

Processing	Wi-Fi connection between android device and Raspberry pi should remain established. When user presses Live Feed button, live view of disastrous environment will start displaying on the android device screen.
Data	This component uses following information of the application: 1. Device Info 2. Raspberry pi Info

6.2 Camera Feed

Identification	Name: Camera feed
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Camera feed** This feature will provide live camera feed of disastrous environment and relative coordinates of objects (human beings). **Description:** User will be able to see live camera feed from camera installed on raspberry pi. Camera feed will be shown on android application. Android application will draw relative coordinates according to objects (human beings) detection.
Function	This module shows the live view of disastrous environment to user on android device
Subordinates	It has one subordinate: 1. Camera feed: Requirement 4.3 in SRS

Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	This component has following interfaces: 1. **Communication Interface:** To establish connection between android device and camera on Raspberry pi 2. **Application UI Interface:** To get live feed from camera
Resources	**Hardware:** Android smartphone, Raspberry pi, Camera, Computer **Software:** Open source android libraries, Android SDKs
Processing	Wi-Fi connection between android device and Raspberry pi should remain established. When user presses Live Feed button, live view of disastrous environment will start displaying on the android device screen.
Data	This component uses following information of the application: 1. Wi-Fi Connection

6.3 Display Processed Information

Identification	Name: Display Processed Information
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document:

	Display Processed Information This feature will provide processed information which will be gathered from Camera Feed. **Description:** User will be able to see the processed information which will be gathered from camera then system will process on gathered data and will perform some functions.
Function	This module allows the rescue operator to see processed data gathered from camera and take decision according to obtained data.
Subordinates	It has one subordinate: 1. Display processed Information gathered from camera
Dependencies	This component is dependent module on camera feed.
Interfaces	This component has following interfaces: **Network Interface: :** To establish connection between android device and camera on Raspberry pi **Application UI Interface:** to visualize the processed data
Resources	**Hardware:** Smartphone, Raspberry pi, Camera **Software:** Open source android libraries, Android SDKs, Python libraries,

Processing	Wi-Fi connection between android device and Raspberry pi should remain established. System will start processing on gathered data which will be gathered from camera and act according to the visualized data.
Data	This component uses following information of the application: 1. Wi-Fi Connection 2. CameraFeed

6.4 Gather Data from Sensors:

Identification	Name: Gather Data From Sensors
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Gather Data From Sensors** This feature will gather data from sensors connected with Arduino. **Description:** System will be able to get the data from Sensors connected with Arduino
Function	This module allows the system to get data from sensors. The data will be Temperature, CO,CO2 level and humidity of the disastrous environment
Subordinates	It has one subordinate: 1. Gather Data from Sensors

Dependencies	This component is dependent module on Connection and runs in parallel to entire application.
Interfaces	This component has following interfaces: 1. **Network Interface:** To interact with Arduino for receiving the data from sensors
Resources	**Hardware:** Smartphone, Raspberry pi, Arduino **Software:** Open source android libraries, Android SDKs, Python libraries,
Processing	Wi-Fi connection between android device, Raspberry pi and Arduino should remain established. System will start processing on gathered data which will be gathered from Sensors.
Data	This component uses following information of the application: 1. Wi-Fi Connection

6.5 Display Gathered Data

Identification	Name: Display Gather Data
Type	Component
Purpose	This service fulfils following requirement from Software Requirements Specification Document:

	Display Gather Data This feature allows android device to display data gathered from raspberry pi. **Description:** System will be able to display the gathered data from sensors on user interface screen
Function	This module allows the users to visualize the gathered data from sensors on user interface.
Subordinates	It has one subordinate: 1. Display the gathered data
Dependencies	This service is dependent module on gather data from sensors
Interfaces	This service has following interfaces: 1. **UI Interface:** To visualize the data to user.
Resources	**Hardware:** Smart phone, Arduino, Pi **Software:** OpenCV
Processing	User opens the application and system starts to gather data from sensors and display on user interface screen
Data	This component uses following information: 1. WiFi Connection 2. Data from sensors

6.6 Environment Analysis

Identification	Name: Environment Analysis
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Environment Analysis** This feature provides analysis of the disastrous environment **Description:** System will be able to analyze the disastrous environment and notifies the user to visualize the data on screen to plan their operation according to that data.
Function	This module allows to analyze the gathered data from sensors and compare with their threshold data and recommend or not according to analyzed data.
Subordinates	It has one subordinate: 1. Environment Analysis
Dependencies	This component is dependent module on gather data from sensors
Interfaces	This component has following interfaces: 1. **Network Interface:** To interact with Arduino to get data and to perform analysis on data.

	2. **Application UI Interface:** To display Recommendation or Not Recommendation message on user interface.
Resources	**Hardware:** Smartphone, Raspberry pi, Arduino **Software:** Open source android libraries, Android SDKs, Python libraries
Processing	System will automatically perform analysis after getting the data from sensors and starts to compare functions with threshold values of constants.
Data	This component uses following information of the application: 1. Wi-Fi Connection 2. Data from sensors

6.7 Navigation

Identification	Name: Navigation
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Navigation:** This feature allows rescue operator to navigate the robotic vehicle using android phone user interface

	Description: A rescue operator will be able to navigate the robotic vehicle using controlling modes of the robotic vehicle displaying on user interface screen
Function	This module allows the rescue operator to control the robotic vehicle using controlling modes listed on user interface screen.
Subordinates	It has one subordinate: 1. Navigation
Dependencies	This component is depended module on connection module.
Interfaces	This component has following interfaces: 1. **Network Interface:** To interact with raspberry pi where Pi will interact with Arduino for sending command for moving vehicle 2. **Application UI Interface:** To tap on controlling mode of navigation
Resources	**Hardware:** Smartphone, Raspberry pi, Arduino **Software:** Open source android libraries, Android SDKs, Python libraries, Firebase
Processing	This module works when rescue operator send commands using controlling modes of navigation.
Data	This component uses following information of the application: 1. Wi-Fi Connection 2. Move Command

6.8 User Interface

Identification	Name: User Interface
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **4.2 Navigation** This feature provides navigation/movement of robotic vehicle. **Description:** Rescue team will control the navigation of robotic vehicle. When user press navigation buttons application will send navigation commands to robotic vehicle. **4.3 Camera Feed and Coordinates** This feature will provide live camera feed of disastrous environment and relative coordinates of objects (human beings). **Description:** User will be able to see live camera feed from camera installed on raspberry pi. Camera feed will be shown on android application. Android application will draw relative coordinates according to objects (human beings) detection. **4.3 Display Data from Sensors** This feature will provide gathered data from sensors and display it on mobile application. **Description:** User must be able see information (Temperature, Humidity, CO_2 level) from different sensors integrated on Arduino Mega.
Function	User interface is one of the ways to interact with application. It packages all those screens, dialogs and forms that are visible to user. User interface is user friendly and easy to understand.

Subordinates	This component has following subordinates: 1. Navigation 2. View Map/Graph 3. View Live Feed from Camera 4. View Data gathered from Sensors
Dependencies	NIL
Interfaces	This component has following interfaces: 1. **Network Interface:** To interact with raspberry pi to get info about gathered data from sensors and get live feed from camera.
Resources	**Hardware:** Smartphone, Computer **Software:** Open Source Android libraries, Android SDKs
Processing	User interface displays info to user passed by other components
Data	This component uses following information of the application: 1. Network Status 2. Camera Feed 3. Sensors Data 4. Navigation Commands

7.0 Reuse and Relationships to other Products

Indoor Disaster Analyzer is the robotic vehicle controlled through Smartphone. It's not an extension of any application at any level. Robotic vehicle is an autonomous system designed for indoor rescue operations. Various system for disaster analysis have been proposed using sensors and camera based system but these are for outdoor disaster analysis, these systems are also difficult to install for indoor disaster analysis. Indoor Disaster Analyzer is easy to install and make rescue operations easy and help to judge the indoor environment through visualization.

This system is composed of many modules and each module is independent to other, so the modules are highly reusable.

8.0 Design Decisions and Tradeoffs

Our project requires client-server architecture. The smartphone app acts as a client and raspberry pi (local server) is acting as server, both are connected through Wireless network.

We are using Model View Presenter (MVP) design pattern because user always request for some change and notify the ListenerService and delegate directly to the presenter, then presenter change the model and then update the view. And when model data set changed it will be listened by ServiceListener, then ServiceListener tells the presenter and presenter will commit the change.

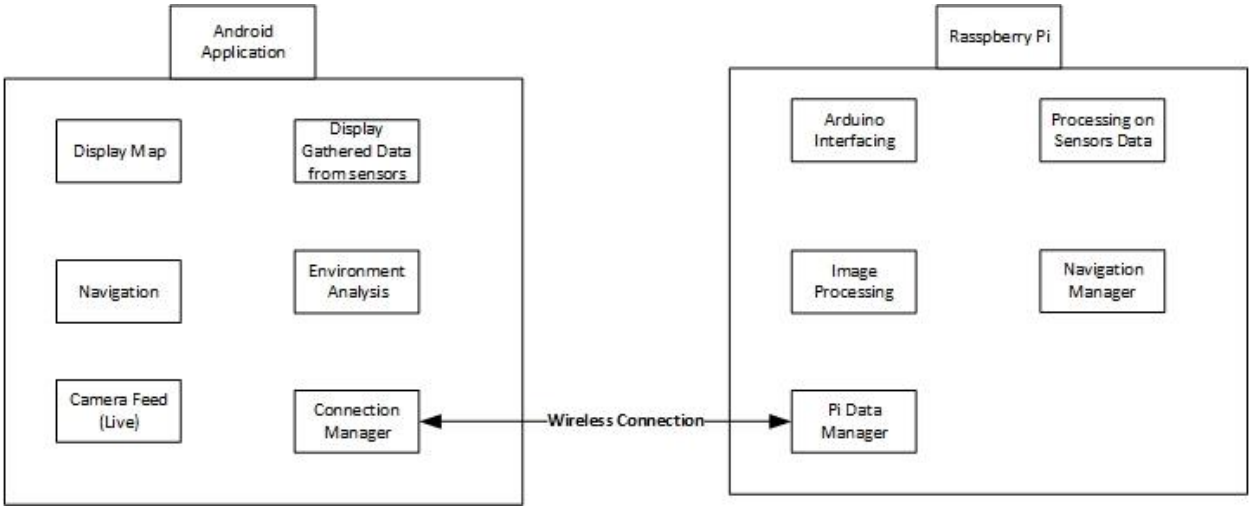


Fig (System Architectural Diagram)

Chapter#05

Project Test and Evaluation

Chapter 5.0 Project Test and Evaluation

5.1 Introduction

This document provides the test documentation for project Indoor Disaster Analyzer, version 1.0 that will facilitate the technical tasks of testing including the detailed test cases for black box testing. Each test case specifies who will be performing the test, the preconditions required to execute each test case, the specific item to be tested, the input, expected output or results, and procedural steps where applicable.

5.2 Test Objective

The objective of this document is to expand on the test plan and provide specific information needed to actually perform the necessary tests. By providing detailed test information, we hope to reduce the probability of overlooking items and improve test coverage. Testers will be able to use each test case provided in this document to move forward and begin testing.

5.3 Test Items

Based on the requirements of project Indoor Disaster Analyzer following are the major modules/ functionalities that should be taken into account during the testing process: -

1. Connection between raspberry pi and android device
2. Video Acquisition
3. Image Acquisition
4. Video Processing
5. Detection of Bodies
6. Acquisition of Environmental Data via Sensors
7. Robotic Vehicle Navigation
8. Mapping the Detected Bodies
9. Environmental Analysis

5.4 Features to Be Tested

Following features are being tested:

Indoor Disaster Analyzer!

1. The system shall be able to connect raspberry pi and Android device through Wi-Fi.
2. The system shall be able to connect raspberry pi and Arduino through Nanpy.
3. The system shall be able to acquire live feed (video) from camera of disastrous environment.
4. System shall be able to get frame from live feed (video).
5. System shall be able to do image processing on selected frame and detect human bodies.
6. System shall be able to get coordinates of human bodies.
7. System shall be able to map the coordinates on graph.
8. System shall be able to get the disastrous environment data of (CO_2, CO, temperature, humidity) from sensors.
9. System shall be able provide Graphical User Interface (to display the environment data, Live feed, Navigation controls) on Android device.
10. System shall enable user to navigate Robotic vehicle via Android device
11. System shall be able to analyze the disastrous environmental data (CO_2, CO, temperature, humidity) from sensors and recommend/not recommend rescue team to enter in disastrous environment

5.5 Detailed Test Strategy

The project Indoor Disaster Analyzer is a computationally intensive system that is why systems modules should be developed independently and then these modules should be integrated. Overall strategy comprises of Unit Testing using White Box and Black box testing. Integration testing is performed in order to successfully integrate the system.

5.5.1 Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

5.5.2 White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level in functions and the results are compared according to requirements. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. The test cases that have

been generated shall cause each condition to be executed at least once. To ensure this happens, we are applying Basis (alternative) Path Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

5.5.3 Black Box Testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would.

5.5.4 Integration Testing

Integration testing is the part where we will test all the previous tested modules in a way that they are functioning normally when they are combined together.

5.5.5 Incremental Testing

There are seven primary modules that are required to be integrated. These components, once integrated, will form the complete application testing. The following describes these modules as well as the steps that will need to be taken to achieve complete integration. We will be employing an incremental testing strategy to complete the integration. The integration testing will be performed by the development team.

Connection

This is the major and basic module of the project that is used to communicate between depended modules as between Android device and raspberry so after connection successfully established then rest modules will initiate to perform their functionalities.

Video Acquisition

This is the module from where the major functioning of application initiates. This module takes live feed (video) from camera of the disastrous environment and will be the shown on android device.

Video Processing

This module takes a frame from live feed (video) and it depends on video acquisition module as its takes a frame from a video will send to the next integrated module to perform analysis on captured frame.

Image Acquisition

This module will get frame captured during video processing module and will start to enhance quality of the captured frame and will send to the next integrated module for further processing.

Human Body Detection

This is the major integrated module which will be used to perform analysis on the enhanced image received from previous module and start to detect the human bodies from frame and counts the detected bodies.

Acquisition of Environment Data via Sensors

This module will be used to gather environment data from disastrous environment like Gas leakage, humidity, room temperature and motion detection.

Robotic Vehicle Navigation

This module is related to navigate the robotic vehicle through android device. User can control movement of the vehicle through forward, backward, left and right navigation buttons.

Mapping of Detected Bodies

This is the major module of the project which is used to map the coordinates of the detected bodies and after mapping the coordinates will be shown on android device as in graph format.

5.6 System Testing

In the end, system testing will ensure that all the modules are working, separately and together combined. Then only the final outcome of the program will decide the correctness of whole system.

Performance testing

This test will be conducted to evaluate the fulfillment of a system with specified performance requirements. It will be done using black-box testing method. And this will be performed by:

1. Checking out the response time of the system
2. Memory management of the program

5.7 Item Pass/Fail Criteria

Details of the test cases are specified in the section Test Deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

1. Preconditions are met
2. Inputs are carried out as specified
3. The result works as what specified in output => Pass
4. The system doesn't work or not the same as output specification => Fail

5.8 Suspension Criteria and Resumption Requirements

Testing will be suspended when a defect is introduced/found that cannot allow any further testing. Testing will be resumed after defect removal.

5.9 Test Deliverables

Following are the test cases:

Test case name	Connection between Android device and Raspberry Pi
Test Case Number	1
Description	This feature enables the system to connect raspberry pi and Android device through Wi-Fi.
Testing Technique used	White Box Testing
Preconditions	Wi-Fi and raspberry pi should be configured properly
Input	Wi-Fi credentials

Steps	Nil
Expected output	Raspberry pi and android device are connected successfully
Alternative path	**Cause:** Function returned False **Corresponding Output:** Error Message Displayed
Actual output	Confirmed

Test case name	**Arduino interfacing with Raspberry Pi**
Test Case Number	2
Description	This feature enables the system to connect raspberry pi and Arduino through Nanpy.
Testing Technique used	White Box Testing
Preconditions	Raspberry pi should be configured properly
Input	Nil
Steps	Nil
Expected output	Raspberry pi and Arduino are connected successfully
Alternative path	**Cause:** Function returned False **Corresponding Output:** Error Message Displayed

Actual output	Confirmed

Test case name	**Acquire video**
Test Case Number	3
Description	This feature enables the system to acquire video of disastrous environment from camera mounted Raspberry pi. This video will be fed into the system for further processing.
Testing Technique used	White Box Testing
Preconditions	Camera should be configured properly with the raspberry pi.
Input	Camera's video feeding into this module
Steps	Open the video capture mode using OpenCV function
Expected output	Function returned True and populate the list with frames provided as argument
Alternative path	**Cause:** Function returned False **Corresponding Output:** Error Message Displayed
Actual output	Confirmed

Test case name	**Image Acquisition**

Test Case Number	4
Description	This feature extracts one image from acquired video as input for further operations.
Testing Technique used	White Box Testing
Preconditions	Video have been successfully fed in to this function, Test Case 3 satisfied
Input	Acquire video module as input
Steps	Nil
Expected output	One image is selected and returned
Alternative path	N/A
Actual output	Confirmed

Test case name	**Human body detection**
Test Case Number	5
Description	Image processing will be applied on image obtained from previous stage to identify human body.
Testing Technique used	White Box Testing
Preconditions	Image(frame) is extracted from video feed

Input	Frame
Steps	1) Convolutional Neural Networks(CNN) Algorithm used to detect human body
Expected output	Body detected or Not detected
Alternative Path	**Cause:** Function returned False **Corresponding Output:** Error Message Displayed
Actual output	Confirmed

Test case name	**Get Coordinates of detected Human bodies**
Test Case Number	6
Description	System shall be able to get coordinates of human bodies
Testing Technique used	White Box Testing
Preconditions	Image processing applied successfully and human body detected. Test Case 1-5 satisfied
Input	Human body detected
Steps	1) Get current location of robot 2) Measure distance from robot to detected body 3) Calculate coordinates on basis of point (1 and 2)
Expected output	Coordinates of detected human body are returned.

Alternative Path	N/A
Actual output	Confirmed

Test case name	**Graph/Map Generation**
Test Case Number	7
Description	Graph/Map of detected Human bodies using their coordinates will be drawn.
Testing Technique used	White Box Testing
Preconditions	Coordinates of detected Human body are extracted successfully Test Case 1-6 satisfied
Input	Coordinates of detected Human body
Steps	1) Draw coordinates on android device
Expected output	Graph/Map should be generated
Alternative Path	N/A
Actual output	Confirmed

Test case name	Environmental Data
Test Case Number	8
Description	This feature enables the system to get the disastrous environmental data (CO_2, CO, temperature, humidity) from sensors
Testing Technique used	Black Box Testing
Preconditions	Test Case 1 and 2 are already satisfied
Input	Nil
Steps	……..
Expected output	Environmental data from sensors returned successfully
Alternative Path	**Cause:** Function returned False **Corresponding Output:** Error Message Displayed.
Actual output	Confirmed

Test case name	Robotic Vehicle Navigation
Test Case Number	9
Description	This feature enables Robotic vehicle to be Navigated via Android device

Indoor Disaster Analyzer!

Testing Technique used	Black Box Testing
Preconditions	Test Case 1 and 8 are already satisfied
Input	Nil
Steps	Press navigation buttons provided on Android device
Expected output	Vehicle navigated successfully
Alternative Path	**Cause:** Function returned False **Corresponding Output:** Error Message Displayed.
Actual output	Confirmed

Test case name	**Environmental Analysis**
Test Case Number	10
Description	This feature enables the system to analyze the disastrous environmental data (CO_2, CO, temperature, humidity) from sensors and recommend/not recommend rescue team to enter in disastrous environment
Testing Technique used	Black Box Testing
Preconditions	Test Case 1,2,8 and 9 are already satisfied
Input	Environmental data from sensors.(Output of Test case 9)

Steps	1) System will compare each inputs with their threshold values.
Expected output	System recommended/not recommended rescue team to enter in disastrous environment
Alternative Path	**Cause:** Function returned False **Corresponding Output:** Error Message Displayed.
Actual output	Confirmed

Test case name	**Graphical User Interface**
Test Case Number	11
Description	This module is related to the design of dynamic and responsive user interface. System shall be able provide Graphical User Interface (to display the environment data, Live feed, Navigation controls and mapping graph) on Android device.
Testing Technique used	Black Box Testing
Preconditions	Nil
Input	Program Initiation will be treated as input to the system
Steps	Create and layout the widgets in the root window
Expected output	Main Screen displayed on screen
Alternative Path	N/A

Actual output	Confirmed

5.10 Environmental Needs

Hardware

1. Raspberry Pi 3 Model B
2. Camera
3. Arduino
4. Android device

Software

1. Raspbian Operating System
2. Python 3.4.0 installed
3. OpenCV 3.0.0

5.11 Responsibilities, Staffing and Training Needs

Responsibilities

All developers of the project are responsible for the completion of all units testing and integration testing tasks.

Staffing and Training Needs

Basic knowledge of testing strategies and techniques is needed for the testing of project.

Techniques such as Black Box testing, integration testing should be known to developers.

All the developers will be testing each other's work and will be actively participating in the development and testing of the project simultaneously.

Risk and Contingencies

Efforts have been made to remove all and every chance of failure but there are certain unpredictable factors such as network issues, corrupt input data, or system failure that may lead to some issues. Error handling will be applied more deeply to cover all these issues but unforeseen circumstances may happen.

Schedule Risk

The project might get behind schedule. So, in order to complete the project on time, we will need to increase the hours/day.

Budget Risk

The budget will be compensated by using less costly alternatives to fit the budget requirements.

Chapter#06

Future Work

Future Work

Indoor disaster analyzer is just a prototype which designed for rescue team to help them for rescue operation but robotic vehicle is used to navigate in disastrous environment is not feasible for real disastrous environment because it can moves on flat surface and cross the hurdles of 40-50 degree angles.

We will design an autonomous robotic vehicle which would be used in any disastrous environment without any user control and will be fire proof. We will also use laser detector to detect presence of human body in disastrous environment.

Chapter#07

Bibliography

Bibliography

i. https://www.hindawi.com/journals/tswj/2015/136434/

ii. https://ieeexplore.ieee.org/document/8301846/

iii. https://www.researchgate.net/publication/283744256_Characteristics_of_indoor_disaster_environments_for_small_UASs

iv. https://ieeexplore.ieee.org/document/7485825/

v. *https://ieeexplore.ieee.org/document/7017661/*

Chapter#08

Glossary

Glossary

- **OS:** Operating System
- **API:** Application Programming Interface
- **Raspbian OS:** A flavor of Linux OS used in raspberry pi
- **CO_2:** Carbon Dioxide
- **CO:** Carbon Monoxide
- **SLAM:** Simultaneous Localization and Mapping