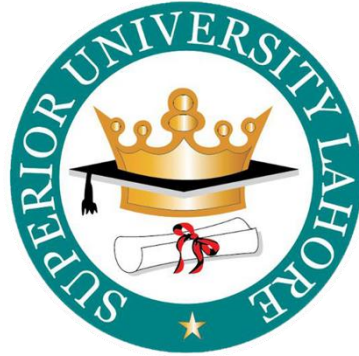


SUPERIOR UNIVERSITY LAHORE



Faculty of Computer Science & IT

Final Year Project PROJECT REPORT

Explosive Detector

Project ID: **BSCS-F14-011**

Project Team

Student Name	Student ID	Program	Contact Number	Email Address
Muhammad Khuzema Ch	BCSM-F14-079	BSCS	+92 300 4254535	m_khuzema@yahoo.com
Hammad Saeed	BCSM-F14-100	BSCS	+92 331 7085577	hammadsaeed415@gmail.com
Hafiz Junaid Jaffar	BCSM-F14-077	BSCS	+92 331 4952078	Stuborn.junaid@gmail.com

Khurrum Jilani

Assistant Professor

Project Report

[Title of Project]

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Muhammad Khuzema	1.0	28-08-2018	<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
Muhammad Khuzema	2.0	07-09-2018	<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: **Khurrum Jilani**

Date: _____ Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____ Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____ Signature: _____

Dedication

*This work is dedicated to **Allah Almighty** Our Creator Who taught Human and parents of all members, who taught us that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to best supervisor **Prof Khurram Jilani** under whose supervision we did all this work, who taught us that even the largest task can be accomplished if it is done one step at a time.*

Acknowledgments

We are thankful to Allah Almighty who gave us health, thoughts and the opportunity to complete this work. We feel great pleasure in expressing our deep and sincere gratitude to our knowledgeable and dedicated supervisor, Mr. Khurram Jilani, Assistant Professor of Computer Science & IT Department who supervised our project lightheartedly and proficiently made the dispatch of intimidating workload possible by persistent guidance communicated to us. We are also thankful to have such a supporting project group Mr. Khuzema Ch, Mr. Hammad Saeed, and Mr. Junaid Jaffar; our coordination helped us a lot in our project work. May ALLAH Almighty grant great rewards to all of the above-mentioned personalities here and hereafter who played their significant roles in enabling us to dispatch our humble efforts in form of this project.

Executive Summary

In this competitive world, the nation's fate was changed because of the terrorism and the war between the two countries becoming frequent. The method of war between countries also improved apart from the fighting of soldiers of two countries such as bio-war, bomb traps in borders may catch many lives of both soldiers and ordinary people. In addition to this, the terrorist of the opponent country and anti-social elements will make the bomb blast in public places like malls, theatres, markets, temples, etc., could kill a lot of people.

During those cases, the bombs are detected by the bomb squad. In such condition, the squad people may also get died due to the wrong operation or shortage of time. However, loss of lives will remains a burden to the country. To prevent this accidental loss of life, the metal detecting robots can be used in such domains. Therefore, we can implement these metal detecting robots in national bomb squad service and also in military defense system. Elimination of terrorism in every nation is a very tedious process. It could not be done with one movement of the defense system. Since it has lots of loss of lives, some immediate and technological activities should be taken for anti-terrorism. In technological wise, this can be overcome with the help of engineers by improving the equipment used in military defense system. This is because nowadays every nation's army face is changing towards the developed technology for ease in identifying and defending of terrorist activity. Robots play an important role in recent technological development. In this emerging technology it could be used fruitfully in the defense system with this the efficiency can be improved.

The main aim of this project is to differentiate between explosive materials and metals and also determine the location of explosives in the terrorist targeting areas. Every type of the bombs used to explode is must chemical made. But unfortunately, existing solutions are not enough to detect accurately for this we're embedding sensing technique of animals with computing unit to enhance the capability of detecting explosive substances.

Table of Contents

Dedication	iv
Acknowledgements.....	v
Executive Summary.....	vi
Table of Contents	vii
List of Figures	x
List of Tables	xi
Chapter 1.....	1
Introduction	1
1.1. Background.....	3
1.2. Motivations and Challenges.....	3
1.3. Goals and Objectives.....	3
1.4. Literature Review/Existing Solutions	3
1.5. Gap Analysis	5
1.6. Proposed Solution	6
1.7. Project Plan	6
1.7.1. Work Breakdown Structure	7
1.7.2. Roles & Responsibility Matrix.....	8
1.7.3. Gantt chart.....	9
Chapter 2.....	10
Software Requirement Specifications	10
2.1. Introduction.....	11
2.1.1. Purpose.....	11
2.1.2. Document Conventions	11
2.1.3. Intended Audience and Reading Suggestions	12
2.1.4. Product Scope.....	12
2.2. Overall Description.....	13
2.2.1. Product Perspective.....	13
2.2.2. Product Functions.....	13
2.2.3. User Classes and Characteristics	13
2.2.4. Operating Environment	13
2.2.5. Design and Implementation Constraints.....	14
2.2.6. User Documentation	15
2.2.7. Assumptions and Dependencies	15
2.3. External Interface Requirements	15
2.3.1. User Interfaces.....	15
2.3.2. Hardware Interfaces	16
2.3.3. Software Interfaces	16
2.3.4. Communications Interfaces.....	18
2.4. System Features	18
2.4.1. Smart IOT Robot	19
2.4.1.1. Description and Priority	19
2.4.1.2. Functional Requirements.....	19

2.4.2.	Smart Detection.....	19
2.4.2.1.	Description and Priority	19
2.4.2.2.	Functional Requirements.....	19
2.4.3.	UBidots Cloud	20
2.4.3.1.	Description and Priority	20
2.4.3.2.	Functional Requirements.....	20
2.5.	Other Nonfunctional Requirements	20
2.5.1.	Performance Requirements	20
2.5.2.	Safety Requirements	20
2.5.3.	Security Requirements	20
2.5.4.	Hardware Quality Attributes	20
2.5.5.	Business Rules.....	20
	Management Team and Company Structure:	21
	Chapter 3.....	22
	Use Case Analysis.....	22
3.1.	Use Case Model.....	23
3.2.	Use Case Descriptions	24
	Chapter 4.....	32
	System Design	32
4.1.	Architecture Diagram (Hardware).....	33
4.2.	Domain Model.....	33
4.3.	Entity Relationship Diagram.....	34
4.4.	Class Diagram	35
4.5.	Sequence / Collaboration Diagram	35
4.5.3	Add User: UC_03.....	37
4.5.4	Delete User: UC_04.....	37
4.5.5	Monitor Record: UC_05	38
4.6.	Operation contracts	38
4.7.	Activity Diagram	38
4.8.	State Transition Diagram.....	47
4.9.	Component Diagram	48
4.10.	Deployment Diagram.....	48
	Chapter 5.....	49
	Implementation	49
5.1.	Important Flow Control/Pseudo codes.....	50
5.2.	Components, Libraries, Web Services and stubs	75
5.3.	Deployment Environment.....	75
5.4.	Tools and Techniques.....	75
5.5.	Best Practices / Coding Standards.....	76
5.6.	Version Control	77
	Chapter 6.....	78
	Testing and Evaluation.....	78
6.1.	Use Case Testing.....	79

6.2. Equivalence partitioning	89
6.3. Boundary value analysis.....	90
6.4. Data flow testing	90
6.5. Unit testing.....	92
6.6. Integration testing.....	92
6.7. Performance testing.....	92
6.8. Stress Testing	93
Chapter 7.....	94
Summary, Conclusion and Future Enhancements.....	94
7.1. Project Summary	95
7.2. Achievements and Improvements	95
7.3. Critical Review	96
7.4. Lessons Learnt.....	96
7.5. Future Enhancements/Recommendations	96
Appendices.....	97
Appendix A: User Manual	98
Appendix B: Administrator Manual	98
Appendix C: Information / Promotional Material	99
Reference and Bibliography.....	103
Index.....	106

List of Figures

1.1 WBS	7
1.2 Gantt Char	9
2.1 Mobile Application Design	15
2.2 Hardware Design	16
2.3 Cloud Design	17
2.4 Communication Interface	18
2.5 IOT Robot Communication	18
3.1 Use Case Model	23
3.2 UC-Admin/User Login	24
3.3 UC-Add User	25
3.4 UC-Delete User	26
3.5 UC-Sensors	27
3.6 UC-Wireless Communication	28
3.7 UC-GPS Tracking	29
3.8 UC-Surveillance	30
3.9 UC-Explosive Detection	31
4.1 Architecture Diagram	33
4.2 Entity Relationship Diagram	34
4.3 Class Diagram	35
4.4 Sequence Diagrams	36
4.4.1 System Functionality	36
4.4.2 Admin Login	36
4.4.3 Add User	37
4.4.4 Delete User	37
4.4.5 Monitor Data	38
4.5 Activity Diagrams	39
4.5.1 Admin Login	39
4.5.2 Add User	40
4.5.3 User Login	41
4.5.4 Delete User	42
4.5.5 User Data Monitor	43
4.5.6 Security Agency Data Monitor	44
4.5.7 Explosive Detection	45
4.5.8 Surveillance	46
4.6 State Transition Diagram	47
4.7 Component Diagram	48
6.4 Data Flow Testing	91

List of Tables

1.1 Roles and Responsibility	8
2.1 Team Member	21
3.1 UC- Admin/User Login	24
3.2 UC- Add User	25
3.3 UC- Delete User	26
3.4 UC- Sensors	27
3.5 UC- Wireless Communication	28
3.6 UC- GPS Tracking	29
3.7 UC- Surveillance	30
3.8 UC- Explosive Detection	31
6.1 Use Case Testing	79
6.1.1 Poistive UC Admin/User Login Testing	79
6.1.2 Negative UC Admin/User Login Testing	79
6.2.1 Positive UC Add User Testing	80
6.2.2 Negative UC Add User Testing	81
6.3.1 Positive UC Delete User Testing	82
6.3.2 Negative UC Delete User Testing	82
6.4.1 Positive UC Sensor Testing	83
6.4.2 Negative UC Sensor Testing	83
6.5.1 Positive UC GPS Tracking Testing	84
6.5.2 Negative UC GPS Tracking Testing	85
6.6.1 Positive UC Surveillance Testing	85
6.6.2 Negative UC Surveillance Testing	86
6.7.1 Positive UC Wireless Communication Testing	87
6.7.2 Negative UC Wireless Communication Testing	87
6.8.1 Positive UC Eplosive Detection Testing	88
6.8.2 Negative UC Eplosive Detection Testing	88
6.9 Equilance Partitioing	89
6.2.1 Add User	90
6.2.2 Login	90

Chapter 1

Introduction

Chapter 1: Introduction

A positive indication of an explosive threat from a perfect sensor does not mean that an explosive is present. In reality, neither a perfect sensor nor a signal that is completely noise-free exists. Standoff explosives detection must take into account more than the single sensor indication because a system that relies on only a single source of information is too likely to make decisions with excessive false positives (false alarms).

Global terrorism a big issue for every country no one can prove that they can provide full proof security. As security is the first need and necessity for life and for this our military and other Security agencies are working day and night. But if they don't have proper and efficient equipment than terrorists easily bypass security hurdles and achieve their target. As we all know Explosive detection is a non-destructive inspection process to determine whether a container contains explosive material or not. For this, we'll make a small gadget which can detect explosives by sensing. We're integrating honey bees electronically because honeybee's success criteria of sniffing are 98% accurate and most probably a unique way of detecting explosives. [1][2][3]

The main aim of this project is to differentiate between explosive materials and metals and also determine the location of explosives in the terrorist targeting areas. Every type of the bombs used to explode is must chemical made. But unfortunately, existing solutions are not enough to detect accurately for this we're embedding sensing technique of animals with computing unit to enhance the capability of detecting explosive substances.

We're working to incorporate intelligent systems with animal's senses to produce smart handheld systems, which can also be accessed remotely, for detection of explosives and other hazardous materials at places where human life can be endangered and also introducing surveillance system which includes face recognition system.

1.1. Background

Detection of explosives is based on a wide variety of technologies that focus on either bulk explosives or traces of explosives. Bulk explosives can be detected indirectly by imaging characteristic shapes of the explosive charge, detonators, and wires or directly by detecting the chemical composition or dielectric properties of the explosive material. Trace detection relies on vapors emitted from the explosive or on explosive particles that are deposited on nearby surfaces. Explosive detection is a very challenging task, and combinations of the various techniques offer increased sensitivities and selectivities. There are numerous technologies for detecting explosives that have been proposed, are in the research stage, or are currently in use. A recent review of technologies and products being used today is available on the web [4]. A more general review of explosives detection can be found in reports [5]

1.2. Motivations and Challenges

Because being Pakistani in the whole world our country is being blamed of promoting terrorism and as in Pakistan terrorism activities are on its peak so to overcome all these issues and give a message to the whole world that we're also against terrorist activities and we're doing our best to overcome this challenge.

1.3. Goals and Objectives

- To introduce innovation in security
- To reduce death rates in case of terrorist attacks.

1.4. Literature Review/Existing Solutions

Direct Competitors:

- Sniffing Dogs.

Indirect Competitors:

- Explosive Detector Devices(using other techniques)
- Metal Detectors

Sniffing Dogs [6]:

Dogs have a superior olfactory sense to humans and can detect small amounts of odor. For this reason, detection dogs are used to detect a lot of different odors both non-biological like accelerants, narcotics, explosives, mines and pollution and biological like cancer, criminals, various species, cows in oestrus or bacteria. Detection in different applications has often been portrayed in the media (newspapers, TV, and internet) but not much has been tested using scientific methods. Therefore, the aim of this literature study is to investigate what dogs can detect, with a focus on examples reported in the peer-reviewed scientific literature, and assess both the difficulties and possibilities with canine odor detection. In the nasal cavities, all species have a sensory epithelium, folded like a labyrinth (ethmoidal turbinate bones). Species with a good olfactory sense have a larger surface on the sensory epithelium because of increased folding and more bones. In humans, the area of the epithelium is 5 cm² whereas the dog has a surface of 150 cm² and 250 million olfactory cells (Sjaastad et al., 2010).

Drawback:

Dogs are not efficient and reliable because many times they failed to sniff and even they aren't carried out everywhere for detection people easily make them fool [7]

Metal Detectors:

The feeling of safety can be created while in an airport, college, office building, or any other area where these detectors are being used. People that are using them can control who enters a specific area and what he/she brings inside it.

There are different types and models of metal detectors, but the most commonly used detectors are the handheld models and walk through models.

With the help of a handheld detector, security personnel can physically detect if a person is carrying a metallic object. On the other hand, walk through metal detectors are much quicker, and they alert the security personnel of dangerous objects such as knives or weapons. [8]

Drawback:

As for the disadvantages of using any type of metal detector, they cause privacy concerns. A lot of people can find it uncomfortable while passing through a detector scanning them for potentially dangerous metallic objects. In some cases, it is possible that they ask to be manually searched when the rules of safety allow such actions.

Also, in this advantageous medical era, we can find a lot of medical reasons why some person might have metal on his/her body. Sometimes, metal devices can be surgically installed inside the person's body which will cause the alarm to be set off each time that person walks through a metal detector. In many cases, it is possible for the security personnel to see on their computers that the metal object that caused the alarm is actually a harmless tool and not a weapon.

As already said, metal detectors create electromagnetic fields. This can cause electrical interference to electronic devices that one person might have in its possession. This includes medical devices such as pacemakers. Some metal detectors can harm pacemakers because they have a very strong electromagnetic field. In this case, such detectors can be fatal for those people, and it is strongly advised that they avoid passing through them.

In many situations, metal detectors can give a false alarm if people are wearing some kind of jewelry, watch, phone, loose change or similar. This can make some people feel embarrassed and can cause panic for no reason. A false alarm caused by detectors can be inconvenient both for the security personnel and the person to whom the alarm turned on. [8]

1.5. Gap Analysis

Global terrorism is a big issue for every country no one can prove that they can provide full proof security. As security is the first need and necessity for life and for this our military and other Security agencies are working day and night. But if they're late to respond in any tragedy, it causes the loss of many lives. Our main concern is to reduce the smuggling of explosive material because if reduce this then easily overcome terrorism. Explosive detection is a non-destructive inspection process to determine whether a container contains explosive material or

not. For this, we'll make a small gadget which can detect explosives by sensing. We're integrating honey bees electronically because honeybee's success criteria of sniffing are 98% accurate and most probably a unique way of detecting explosives.

1.6. Proposed Solution

Whenever humans find difficulty in doing any works then it leads to the path of invention. As we come to know that sniffing success criteria of dogs are 71% accurate so we inspired from this and try to enhance our research regarding the senses of animals. We're trying to incorporate intelligent systems with animal's senses to produce smart handheld systems, which can also be accessed remotely, for detection of explosives and other hazardous materials at places where human life can be endangered.

Our current product is a smart solution which embeds living honey bees with smart computing unit which translates senses of the bee into digital signals when the bees in the unit sense explosives. We're embedding honeybees because of their sniffing success criteria is 98% accurate as compared to sniffing of dog criteria whose success criteria is 71% and for honeybees, we don't need a huge amount and time to train them.

1.7. Project Plan

The project plan of the explosive detector is as follows:

- Planning
- Designing and development of website and application
- Designing hardware.
- Implementation of Coding
- Development of hardware.
- Integration with application and cloud.
- Testing and Debugging.

1.7.1. Work Breakdown Structure

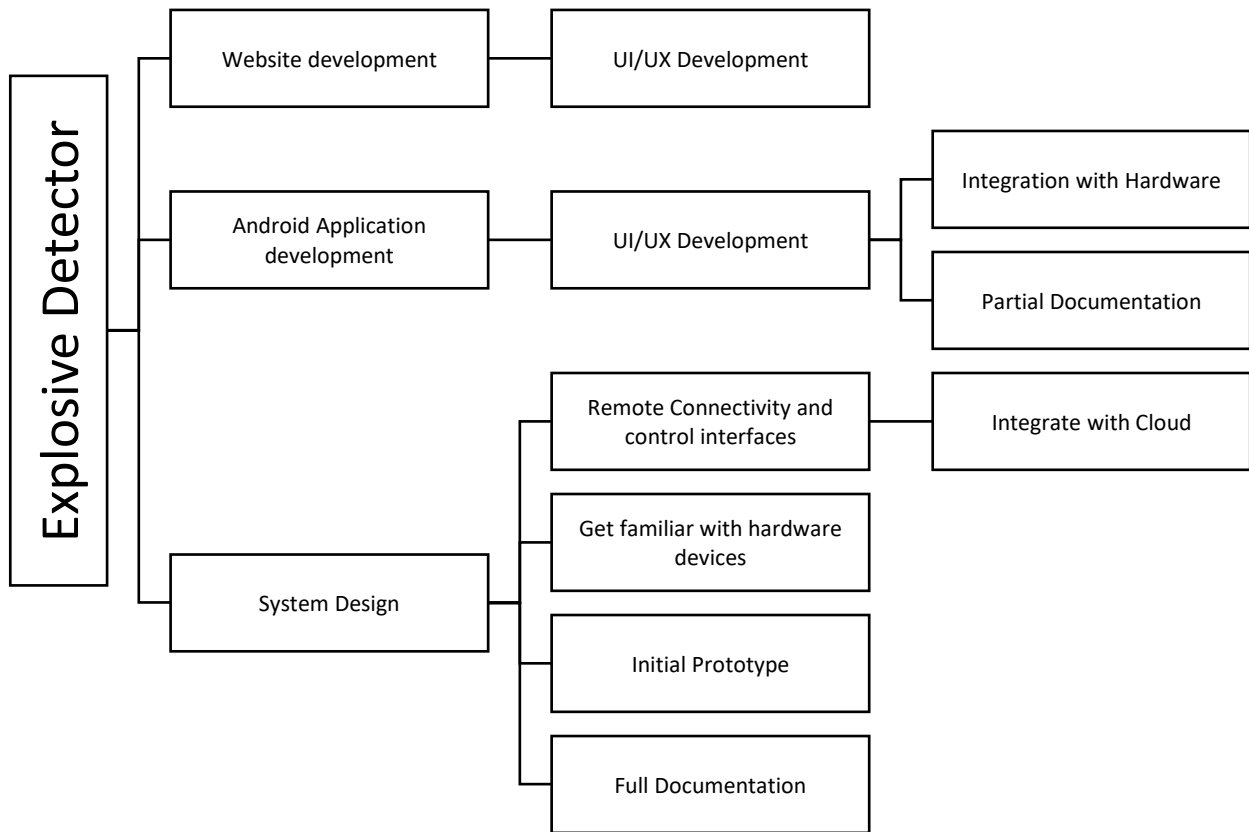


Fig 1.1-WBS

1.7.2. Roles & Responsibility Matrix

The purpose of the roles & responsibility matrix is to identify who will do what.

Table 1.1-Roles & Responsibility

Tasks and Milestone	Due Date	Due date type	Percent Complete	Responsible Team Member(s) & Role(s)
Project Choice	11/12/2017	Assigned	100%	Hammad Saeed, Muhammad Khuzema
Team Name and Facilities	11/12/2017	Assigned	100%	All
Scope Definition	Constant Adjustments	On-going	40%	All
Presentations and Reports	20/12/2017	Assigned	50%	Muhammad Khuzema (Report) Hammad Saeed & Junaid (Presentation)
Hardware Devices selection	4/1/2018	Milestone	50%	Muhammad Khuzema
Initial Prototype	30/12/2017	Milestone	50%	All
Website Development	20/1/2018	Milestone	100%	Junaid(Front-end)
Android App working	25/1/2018	Milestone	100%	Hammad & Khuzema
Final Report	28/8/2018	Assigned	100%	Khuzema
Prototype Completion	28/8/2018	Estimated	100%	All

1.7.3. Gantt chart

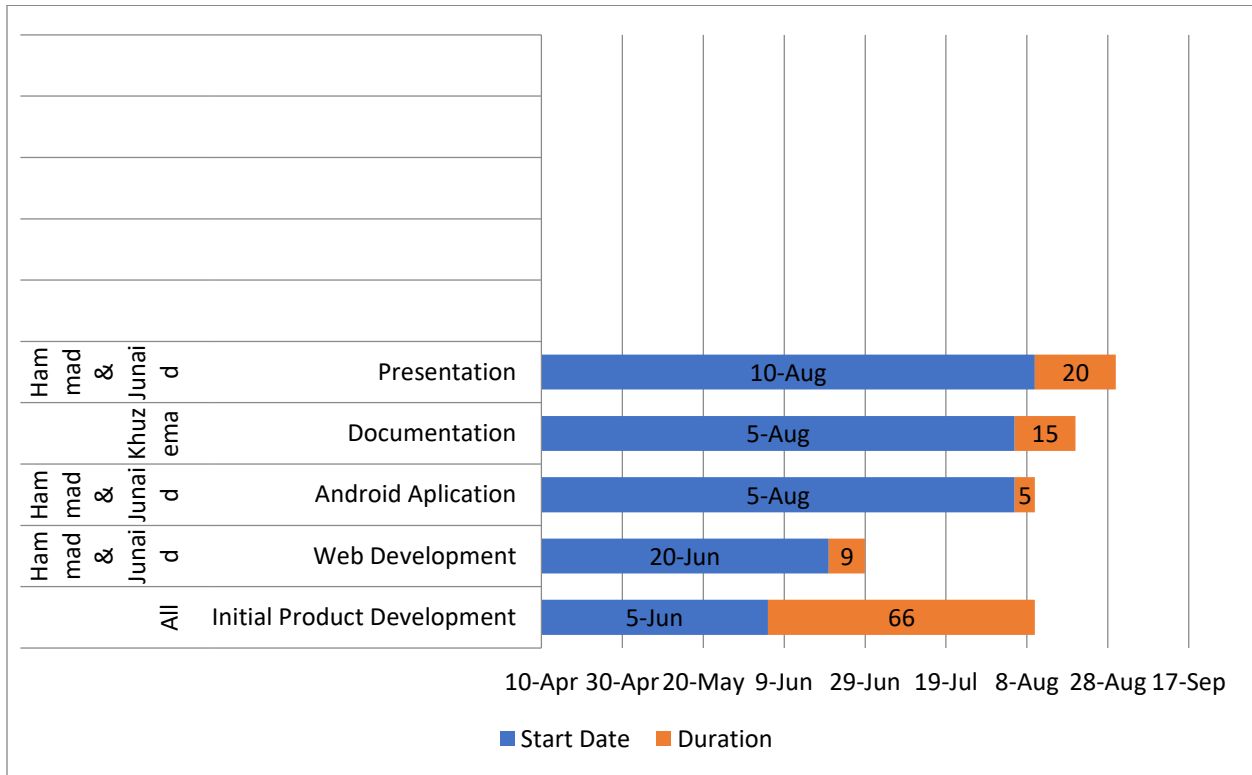


Fig 1-2 Gantt chart

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, document conventions, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight into the complete **Explosive Detector Project** by defining the problem statement in detail. The detailed requirements of the **Explosive Detector** are provided in this document.

2.1.1. Purpose

The purpose of the document is to collect and analyze all assorted ideas that have come up with to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters, and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team, and the audience see the product and its functionality.

2.1.2. Document Conventions

This SRS Document follows MLA Format. Bold text has been used to highlight section and sub-section headings. Highlighting is to point out words in the glossary and italic text is used to label and recognize diagrams. Technical details related to hardware and software systems and interfaces are not included in this document in detail. Regarding information and sources will be provided in the appendix sections at the end of the document.

2.1.3. Intended Audience and Reading Suggestions

This Software Requirements document is intended for:

- Developers who can review project's capabilities and more easily understand where their efforts should be targeted to improve or add more features to it (design and code the project – it sets the guidelines for future development).
- Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. This way testing becomes more methodically organized.
- End users of this product who wish to read about what this product can do.

2.1.4. Product Scope

Global terrorism is a big issue for every country no one can prove that they can provide full proof security. As security is the first need and necessity for life and for this our military and other Security agencies are working day and night. But if they don't have proper and efficient equipment than terrorists easily bypass security hurdles and achieve their target.

Because being Pakistani in the whole world our country is being blamed of promoting terrorism and as in Pakistan terrorism activities are on its peak so to overcome all these issues and give a message to the whole world that we're also against terrorist activities and we're doing our best to overcome this challenge.

The main aim of this project is to differentiate between explosive materials and metals and also determine the location of explosives in the terrorist targeting areas. As security is the first need and necessity for life and for this our military and other Security agencies are working day and night. But if they don't have proper and efficient equipment than terrorists easily bypass security hurdles and achieve their target. Every type of the bombs used to explode is must chemical made. But unfortunately, existing solutions are not enough to detect accurately for this we're embedding sensing technique of animals with computing unit to enhance the capability of detecting explosive substances.

2.2. Overall Description

2.2.1. Product Perspective

Explosive Detector is a very useful product due to working phenomena and in a domain where it needs to be used. As it is the need of the native market and with its more efficiency and accuracy it'll boost gradually.

2.2.2. Product Functions

Explosive Detector (Robot) provides the following features:

- GPS Tracking: With GPS tracking we can easily monitor the movement of our robot.
- IOT Robot: We're introducing an IOT based remotely controlled the robot by embedding a GPS tracker in it to monitor it from anywhere.
- Face Detection and Recognition: With this feature, we can cater the question mark every time raised on different incidents; we can detect faces of peoples and recognize the suspicious person.
- Explosive Detection: As explosive detection is our main purpose of this project by integrating honey bees with our computing unit.

2.2.3. User Classes and Characteristics

Explosive Detector is a powerful robot, yet easy product to use so, but as it is defensive product and we don't want to use it by everyone openly so we provide proper licensing and for respective users with some basic training are able to use it efficiently.

- Security Agencies: Any security providing department who are responsible for providing security in different areas can easily get familiar with our product and utilize it in the best way.

2.2.4. Operating Environment

Explosive Detector is defensive hardware product it can operate only by security agents within their respective domain with little bit experience. We have used Arduino as a controller to control the process of our project. The Arduino Uno Wi-Fi is an Arduino Uno with an integrated Wi-Fi module. The board is based on the ATmega328P with an ESP8266 Wi-Fi

module integrated. The ESP8266 Wi-Fi module is a self-contained SoC with integrated TCP/IP protocol stack that can give access to your Wi-Fi network (or the device can act as an access point). One useful feature of Uno Wi-Fi is support for OTA (over-the-air) programming, either for transfer of Arduino sketches or Wi-Fi firmware. We are using laser sender/receiver to detect the cut off between laser sender and receiver due to the tongue of honey bee which will sense respective parameters and that values will be sent to the Arduino board and cloud. Then according to the received values from sensors Arduino will take the particular action specified in the program stored inside the memory of Arduino. The output devices like LED and Buzzer are used to alert the user as well as Buzzer will be blown so that user can understand warning from the robot. Android app is made to remotely control the robot.

2.2.5. Design and Implementation Constraints

Design section addresses the issues that need to be resolved before attempting to develop a complete design solution.

- Work products such as documents will be in compliance with IEEE standards.

- General Constraints

There are three stages of the development of our product.

- IOT based remotely controlled Robot.
 - Wi-Fi should be connected to any router to transmit data in the cloud.
 - The Application must be connected to Bluetooth.
- Trained Honeybees.
 - Honeybees must be trained perfectly for every type of explosive detection.
- Placement of Honeybee into Box.
 - In this stage, honeybee must be placed perfectly inside the box.
 - The User must check that laser sender/receiver is in front of honeybee's mouth.

Training and placement of honeybee can be done only by respective persons.

2.2.6. User Documentation

As the product is under the R & D stage and requires a complete implemented prototype to explain the user documentation. Once the prototype is designed and implemented online manuals, user manuals can be provided to the concern persons upon request.

2.2.7. Assumptions and Dependencies

As we already mentioned above in this document that this product is under the Research & Development phase and we're working on that project where we need more time and resources to enhance the efficiency of our product. Moreover, our product results are efficient if the user uses in the way we recommended.

2.3. External Interface Requirements

2.3.1. User Interfaces

All hardware products have its own interfaces which can be user-friendly enough and users can manipulate with it. The Menu is also an important thing while making come into existence the SRS document part. In this SRS document part; we have used the menu quick-shot in the user done. These quick-shots are based on the menu of the product.

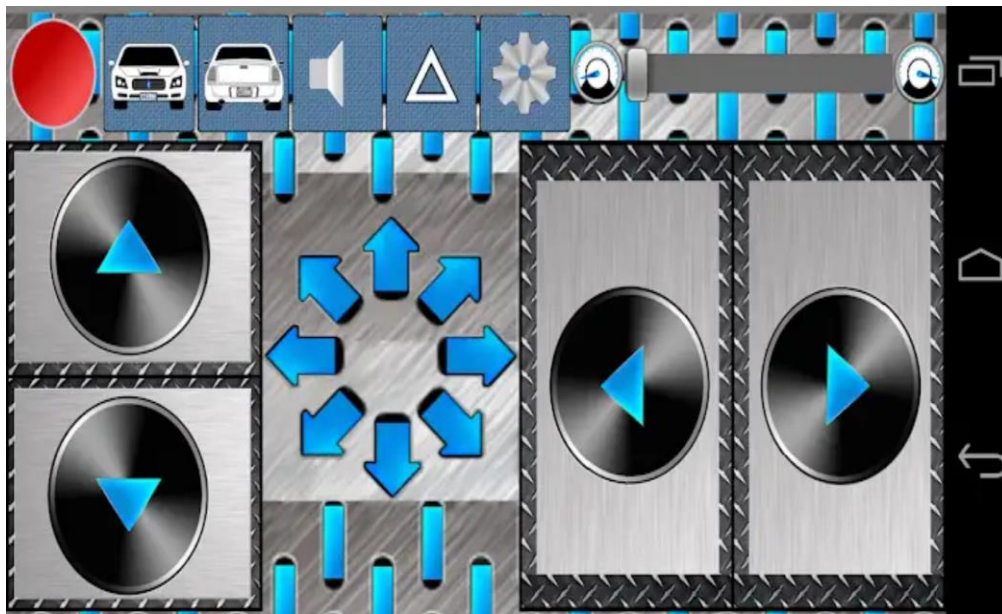


Fig 2.1 Mobile App

2.3.2. Hardware Interfaces

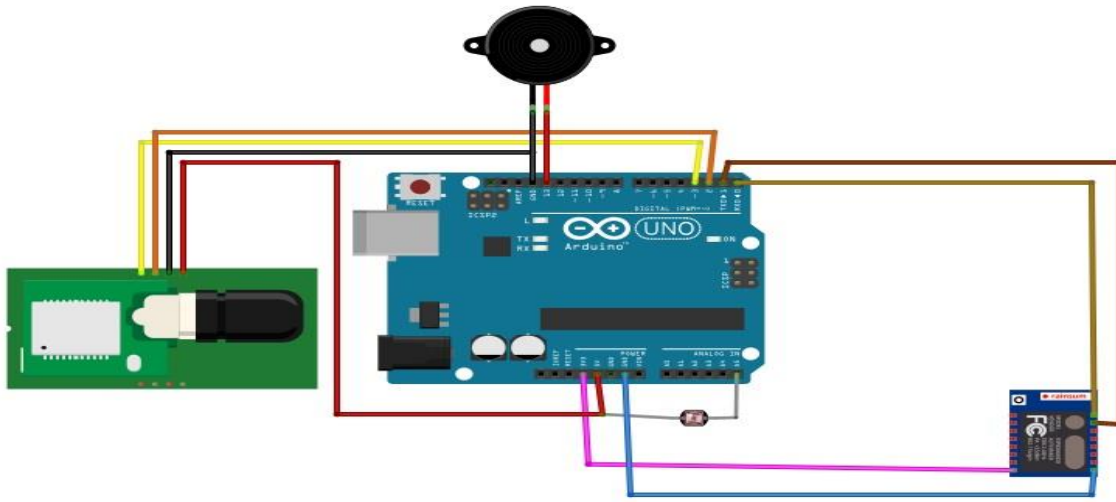


Fig 2.2 Hardware Design

2.3.3. Software Interfaces

The software we're using for our cloud is ubidots where the user can easily monitor data from anywhere. Here is the demo screen.

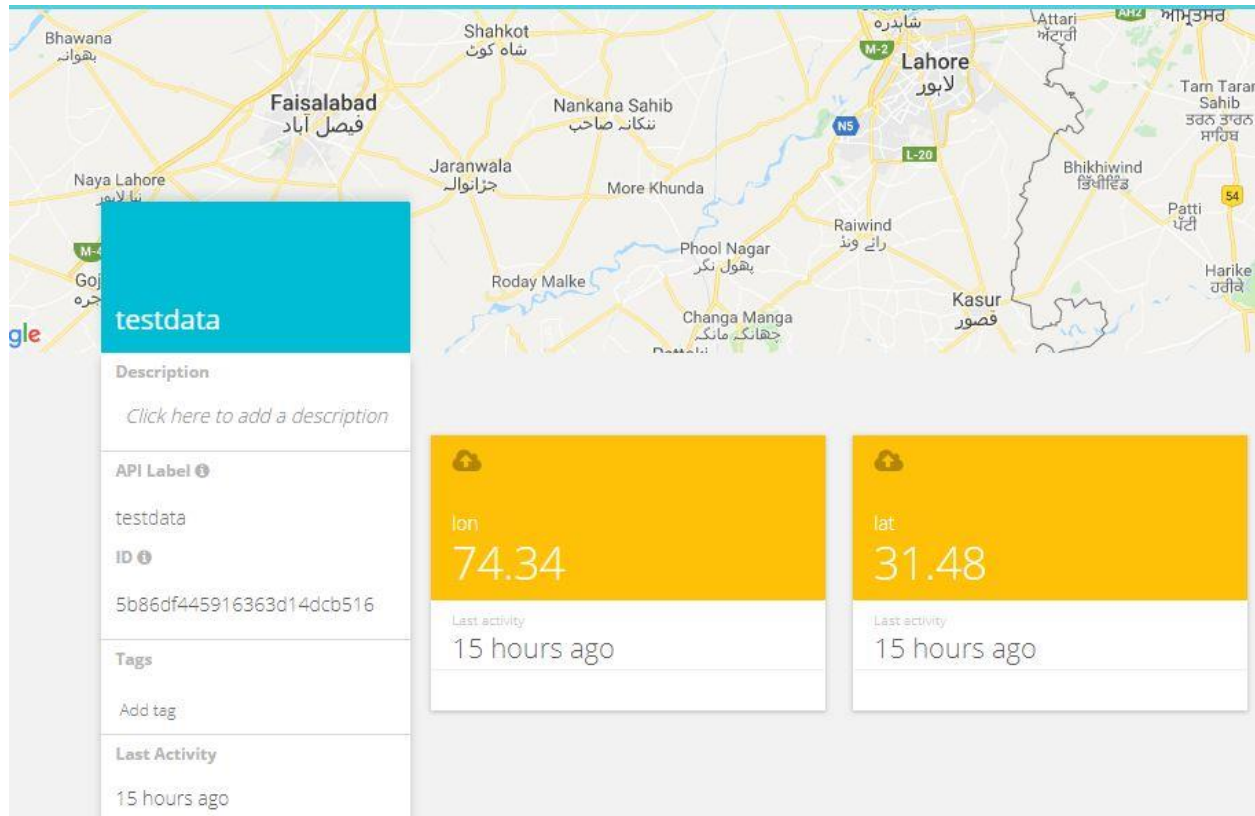


Fig 2.3 Cloud

1) Development Tools

Programming for Arduino may be written in any programming language with a compiler that produces binary machine code like C and C++ programming. The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming code language Java. It originated from the IDE for the languages Processing and Wiring. It was created for people with no profound knowledge of electronics. The Arduino IDE supports the programming languages C and C++ using special rules to organize code. The Arduino IDE also supplies a software library from the Wiring project that gives many common input and output procedures. The user-written code only requires two functions, for starting the sketch and the main programs loop, that is compiled and then linked with a program stub `main()` into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program air dude to convert from executable code into a text file in hexadecimal coding that is loaded into the Arduino board by a loader program in the board's firmware.

2) Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android application development, based on the IntelliJ IDEA. On top of IntelliJ's, there is a powerful code editor and developer tool; Android Studio offers even more advantageous features that enhance your productivity when building Android applications.

2.3.4. Communications Interfaces

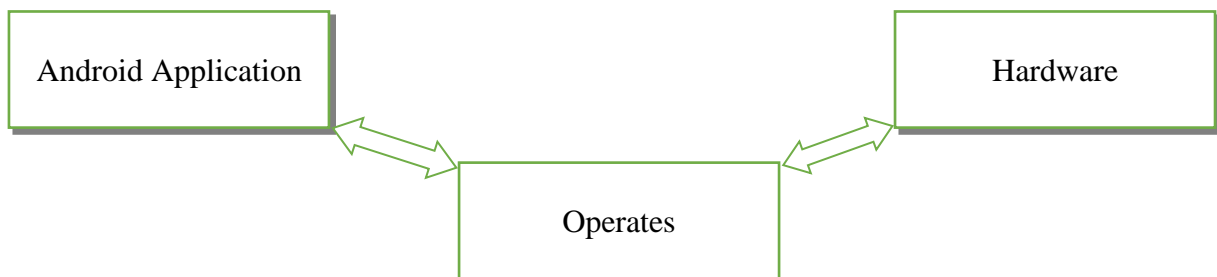


Fig 2.4 communication interface

2.4. System Features

Explosive Detector Robot features are divided into two main categories: functional features and non-functional features. Functional features form the body of the application and include any features that are essential to the functionality of the order to have a fully-functioning application. Additional features, however, are not required for the app to function. The smart Sensing Unit for fridge, freezers or any food storing appliances which can:

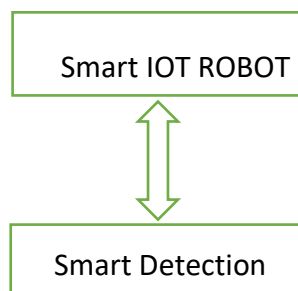


Fig 2.5 IOT Robot

2.4.1. Smart IOT Robot

Here in this feature IOT based robot is discussed which will be accessed remotely with the help of the mobile application.

2.4.1.1. Description and Priority

- WI-FI of robot needs to be connected to the router for transmission of data to the cloud.
- GPS tracker will send coordinates values of the robot at Cloud.
- The camera will be sent live stream at Cloud where all the data should be saved in the form of Images.

2.4.1.2. Functional Requirements

REQ-SF1-1: WI-FI module needs to be connected.

REQ-SF1-2: GPS tracker works perfectly.

REQ-SF1-3: Camera records images and sends it to the cloud for monetarization.

2.4.2. Smart Detection

Here in this feature Detection of explosive materials will be done with the help of living honey Bees.

2.4.2.1. Description and Priority

Italian Honey Bee with proper training of explosive detection needs to be placed perfectly. Laser sender/receiver needs to be placed perfectly. If honey bee sniffs any explosive materials then Green Led blinks and if within the 20-sec connection between laser sender/receiver breaks more than five times than we've positive response and that point is declared as suspicious.

2.4.2.2. Functional Requirements

REQ-SF2-1: Italian Honey Bee with proper training of explosive detection

REQ-SF2-2: Honey Bee within the box and a laser sender/receiver placed properly.

REQ-SF2-3: Wi-Fi module needs to be connected.

2.4.3. UBidots Cloud

2.4.3.1. Description and Priority

The data collected from Robot can be monitored from anywhere and can be used only by higher management through which they can easily find out suspicious persons and if some incident occurs then they can find out reasons behind the incident.

2.4.3.2. Functional Requirements

REQ-SF3-1: Login

REQ-SF3-2: Add User

REQ-SF3-3: Delete User

2.5. Other Nonfunctional Requirements

2.5.1. Performance Requirements

Performance of our gadget is depended upon the accuracy of our gadget if our gadget is accurate then it'll perform well otherwise maintenance is required.

2.5.2. Safety Requirements

For safety purpose initially, we're providing services when with the passage of time and fully training of security departments we'll sell our products only to security departments.

2.5.3. Security Requirements

For security, we'll manage all backend functionalities of our product whether it is detection or visualizing through image processing.

2.5.4. Hardware Quality Attributes

This System is flexible, maintainable and usable in this sense that it is simple to use and low cost is required for its development.

2.5.5. Business Rules

We are a team of three members currently working on this idea to start as a fyp project and in future to run a new business which will help to enhance security more powerfully. We will be able to generate revenue as soon our product will be introduced in the concerned market. We are working on the development of explosive detector which will help the agencies to provide full proof security.

ED Tech is a small company developed by a team. We believe in this quotation “all for one, one for all”. Our all team work together dividing our tasks. Our company’s formworks will offer major advances, complete adaptability, and high strength-to-weight ratio, and all at cost-effective prices. During form use, maintenance will be minimal.

Management Team and Company Structure:

Table 2.1 Team Members

M Khuzema Ch	Responsible for all day-to-day management decisions and for implementing the Company's long and short-term plans. Also, responsible for team management.
Hammad Saeed	Responsible for Technical Management. And all hardware working.
Junaid Jaffar	Responsible for the daily operation of the company,

Chapter 3

Use Case Analysis

Chapter 3: System Analysis

System analysis can look at the end-user implementation of a software package; it looks in depth to explain the techniques used in Software and Hardware.

3.1. Use Case Model

Use case analysis is a method used to classify the requirements of a system and the information used to both define processes used and classes, which are set of actors and processes which will be used both in the use case diagram and the overall use case in the development or redesign of the product. The use case analysis is the base upon which the system will work.

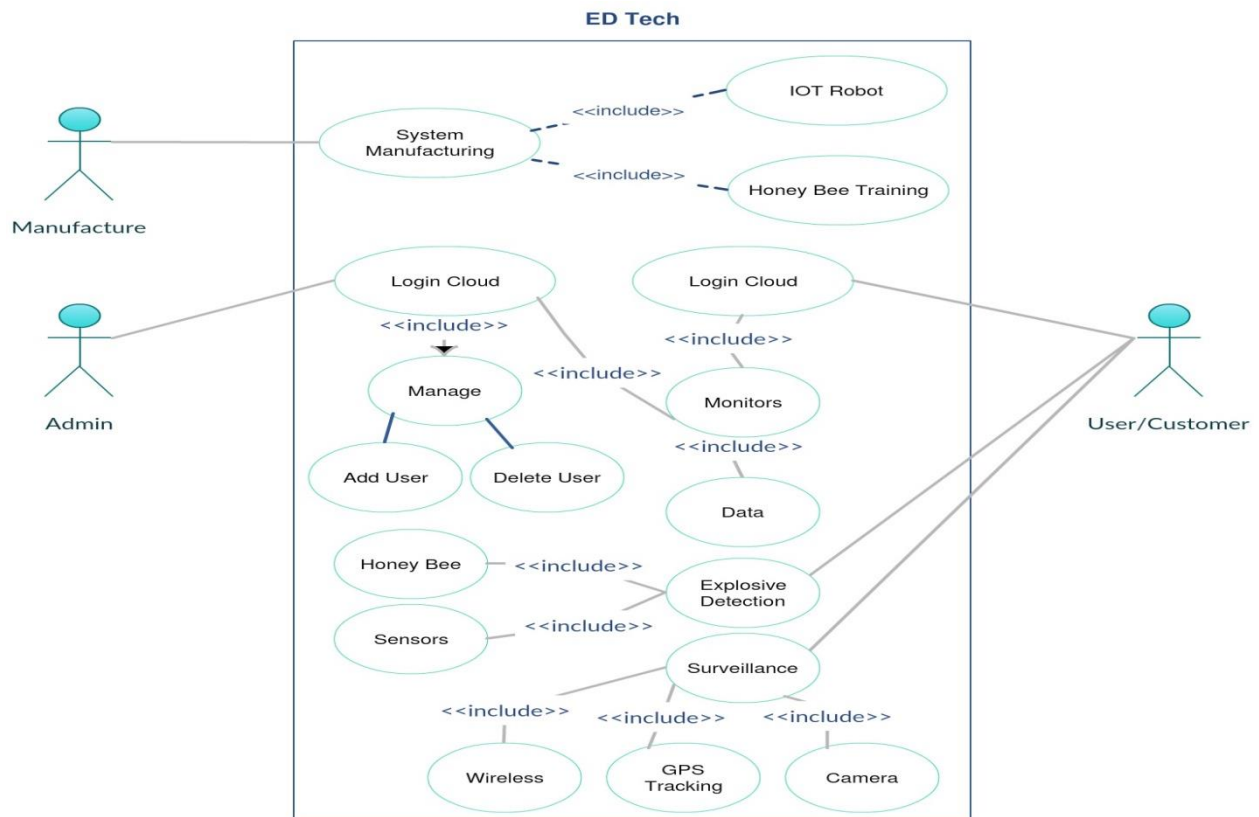
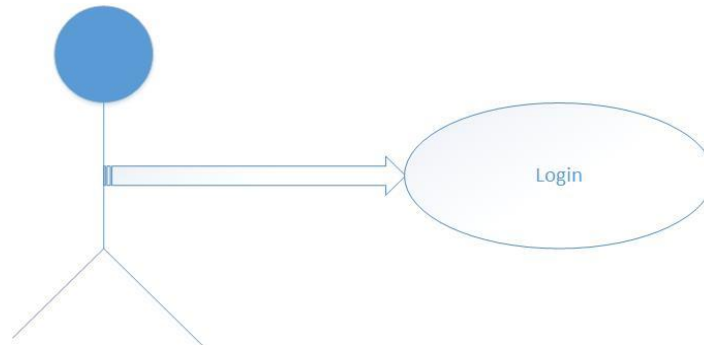


Fig 3.1 Use case Model

3.2. Use Case Descriptions

3.2.1. Use-case: Admin/User log-in



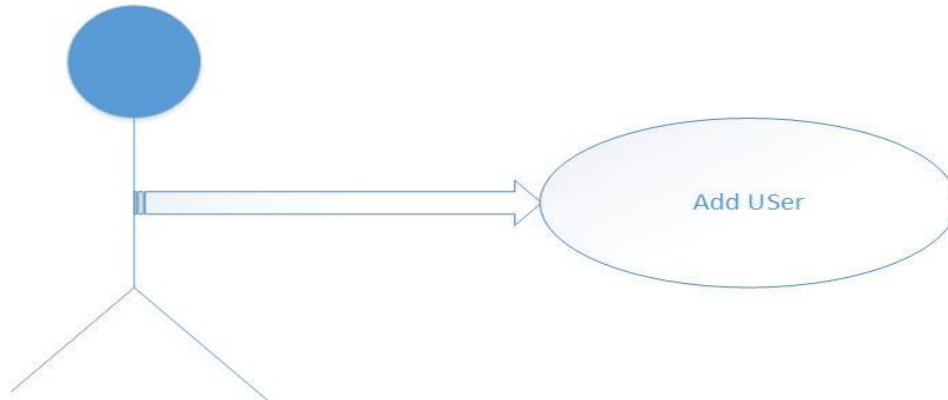
Actor: Admin/User

Fig 3.2 Use-case: Admin/User log-in

Table 3.1 Use-case: Admin/User log-in

USE CASE	UC_01	
FR #	FR_01	
Name	Admin /User Login	
Primary Actor	Admin, User	
Brief description	Step	Action
	1	Admin Must be Registered
	2	Admin must provide correct parameters
Pre-condition	Admin provide email & password	
Post-conditions	Successfully logged in to the system	
Basic flow	Step	Action
	1	Open the login section
	2	Provide Correct parameters
	3	Logged in Successfully

3.2.2. Use-case: Add User



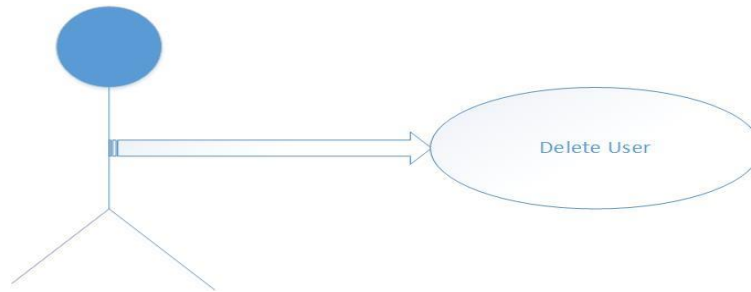
Actor: Admin

Fig 3.3 Use-case: Add User

Table 3.2 Use-case: Add User

USE CASE	UC_02	
FR #	FR_02	
Name	Add User	
Primary Actor	Admin	
Brief description	Step	Action
	1	Open the registered User request
	2	Click the add button against the User that should be active
Pre-condition	Admin must be logged in	
Post-conditions	Selected User has been added	
Basic flow	Step	Action
	1	Login to admin panel
	2	Open the registered User request
	3	Click the add button against the User that should be active
	4	Selected User has been added

3.2.3. Use-case: Delete Users



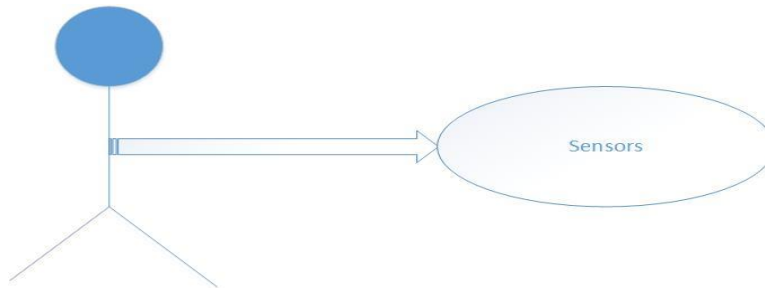
Actor: Admin

Fig 3.4 Use-case: Delete User

Table 3.3 Use-case: Delete User Account

USE CASE	UC_03	
FR #	FR_03	
Name	Delete User account	
Primary Actor	Admin	
Brief description	Admin wants to Delete any User's account	
Pre-conditions	1	The user must be active
	2	Admin must be logged in
	3	There has to be some legit reason
Post-conditions	User's account Deleted	
Basic flow	Step	Action
	1	Admin must be logged in
	2	Admin selects the User to be Deleted
	3	Admin click the delete button
	4	Deleted

3.2.4. Use-case: Sensors



Actor: User

Fig 3.5 Use-case: Sensors

Table 3.4 Use-case: Sensors

USE CASE	UC_04	
FR #	FR_04	
Name	Sensors	
Primary Actor	User	
Brief description	The User wants to detect.	
Pre-condition	All sensors should be placed properly.	
Post-conditions	NONE	
Basic flow	Step	Action
	1	The User checks the sensors.
	2	All sensors will be generate output and sends feedback to the user and cloud.

3.2.5. Use-case: Wireless Communication

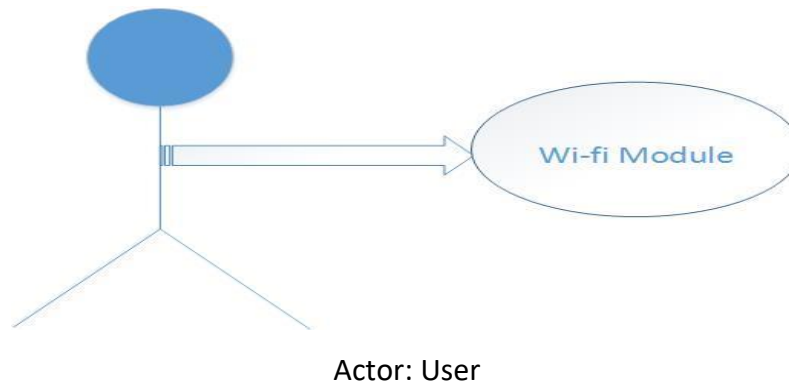
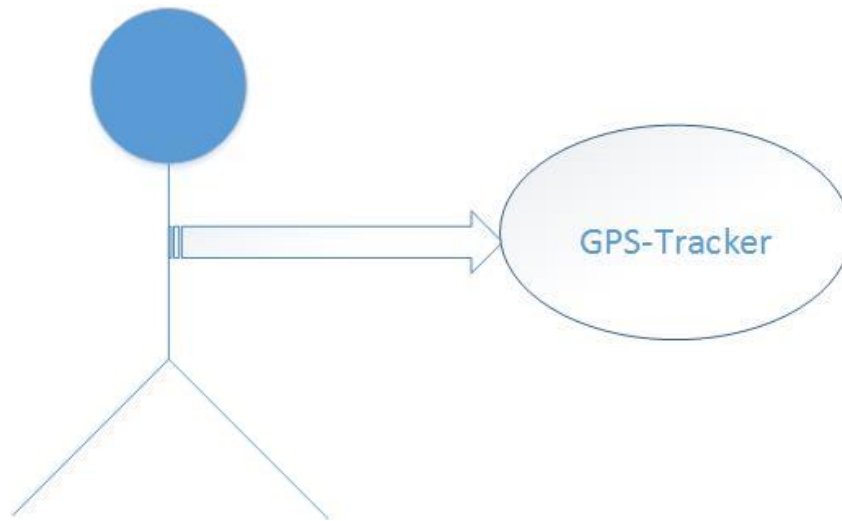


Fig 3.6 Use-case: Wireless Communication

Table 3.5 Use-case: Wireless Communication

USE CASE	UC_05	
FR #	FR_05	
Name	Wireless Communication	
Primary Actor	User	
Brief description	The User wants to check to communicate wirelessly.	
Pre-condition	WI-FI must be connected	
Post-conditions	NONE	
Basic flow	Step	Action
	1	Detection process should be completed.
	2	All values should be sent to Cloud

3.2.6. Use-case: GPS Tracking



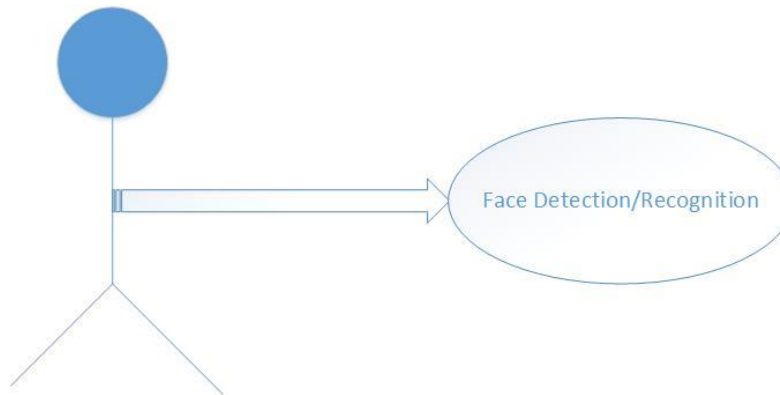
Actor: User

Fig 3.7 Use-case: GPS Tracking

Table 3.6 Use-case: GPS Tracking

USE CASE	UC_06	
FR #	FR_06	
Name	GPS Tracking	
Primary Actor	User	
Brief description	The User wants to monitor GPS Tracking	
Pre-condition	The user must be logged in	
Post-conditions	NONE	
Basic flow	Step	Action
	1	The user selects the "GPS" option
	2	All GPS values will be displayed.

3.2.7. Use-case: Surveillance



Actor: User

Fig 3.8 Use-case: Surveillance

Table 3.7 Use-case: Surveillance

USE CASE	UC_07	
FR #	FR_07	
Name	Surveillance	
Primary Actor	User	
Brief description	The User wants to Investigate through using Face Recognition	
Pre-condition	The user must be logged in The camera must be connected to WI-FI	
Post-conditions	NONE	
Basic flow	Step	Action
	1	The camera will transmit video on Cloud
	2	The user can monitor data

3.2.8. Use-case: Explosive Detection

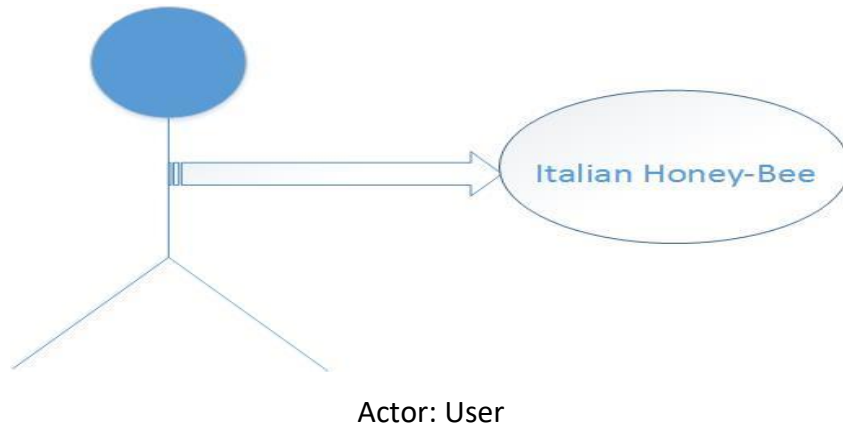


Fig 3.9 Use-case: Explosive Detection

Table 3.8 Use-case: Explosive Detection

USE CASE	UC_08	
FR #	FR_08	
Name	Explosive Detection	
Primary Actor	User	
Brief description	Honey Be to be used for Explosive Detection	
Pre-condition	Honey Bee must be trained Sensors must be placed correctly	
Post-conditions	NONE	
Basic flow	Step	Action
	1	Honey Bee Sniffs
	2	Signal will be generated

Chapter 4

System Design

Chapter 4: System Design

In system design diagrams related to a product's designing and the functions of each module will display using diagrams which while enhancing the functionality of the system's working.

4.1. Architecture Diagram (Hardware)

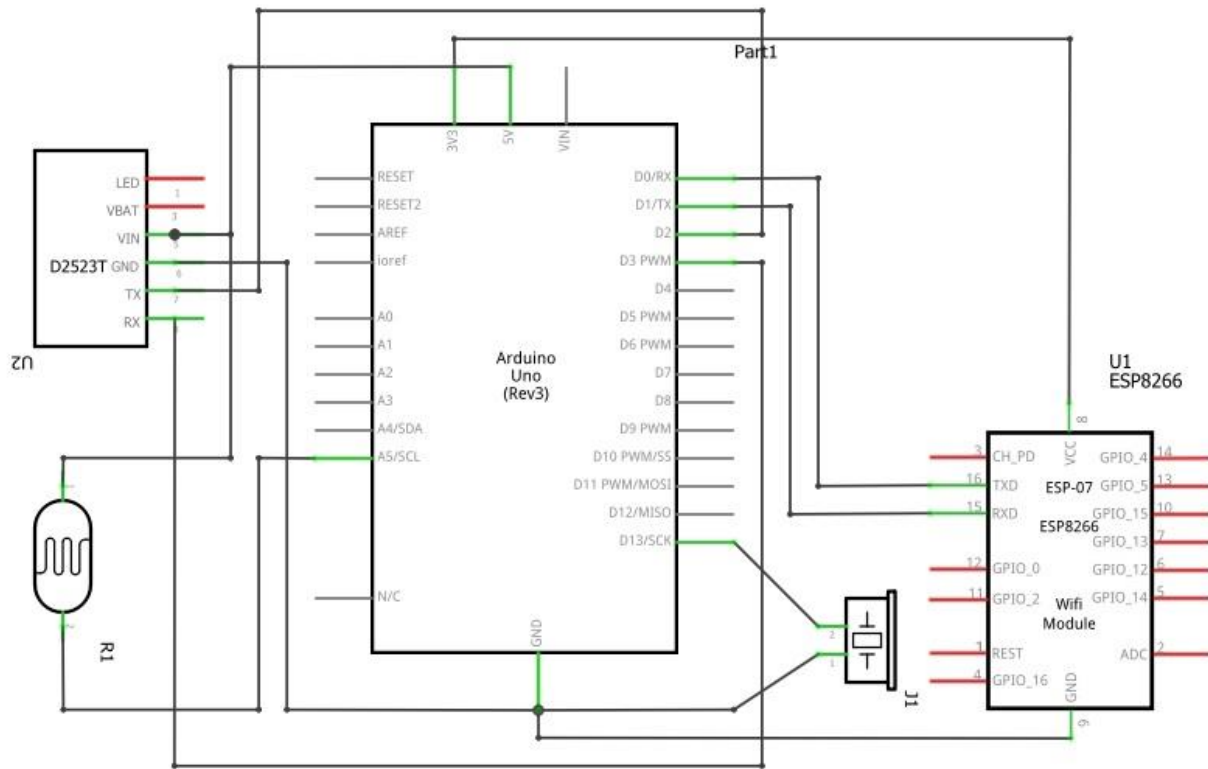


Fig 4.1 Architecture Diagram: AD_1

4.2. Domain Model

N/A

4.3. Entity Relationship Diagram

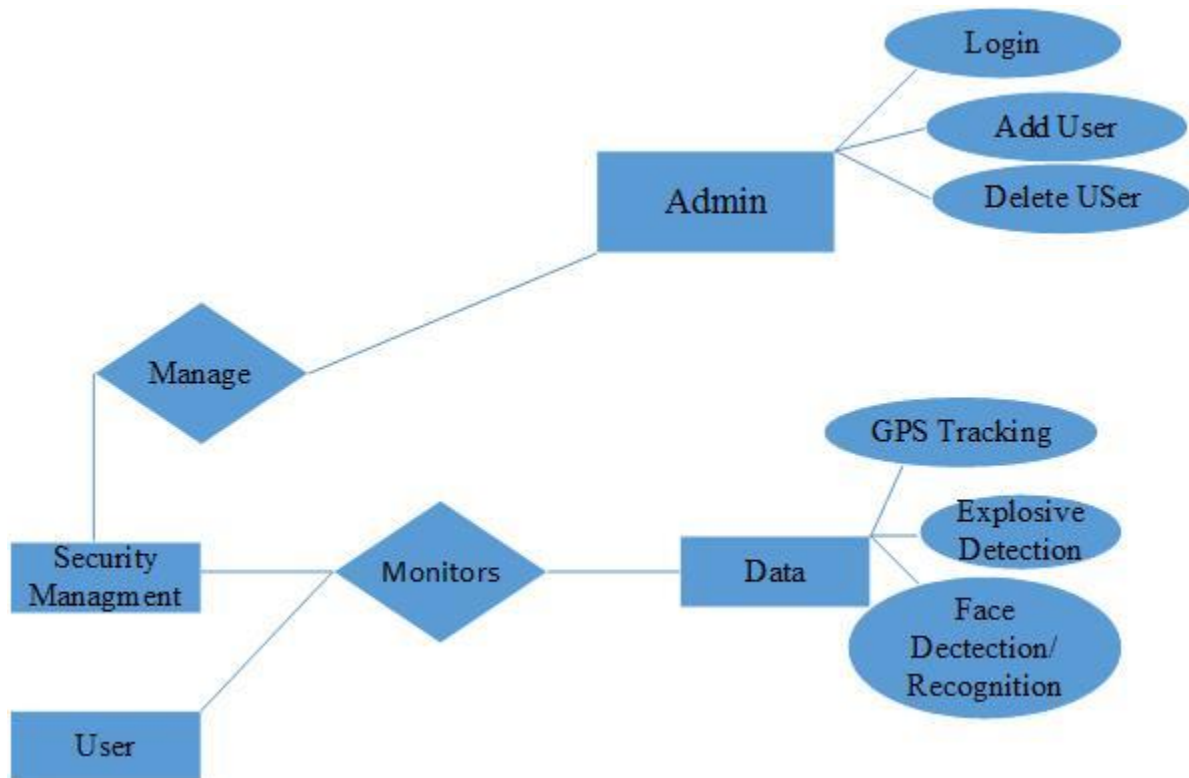


Fig 4.2 Entity Relationship Diagram: ERD_1

4.4. Class Diagram

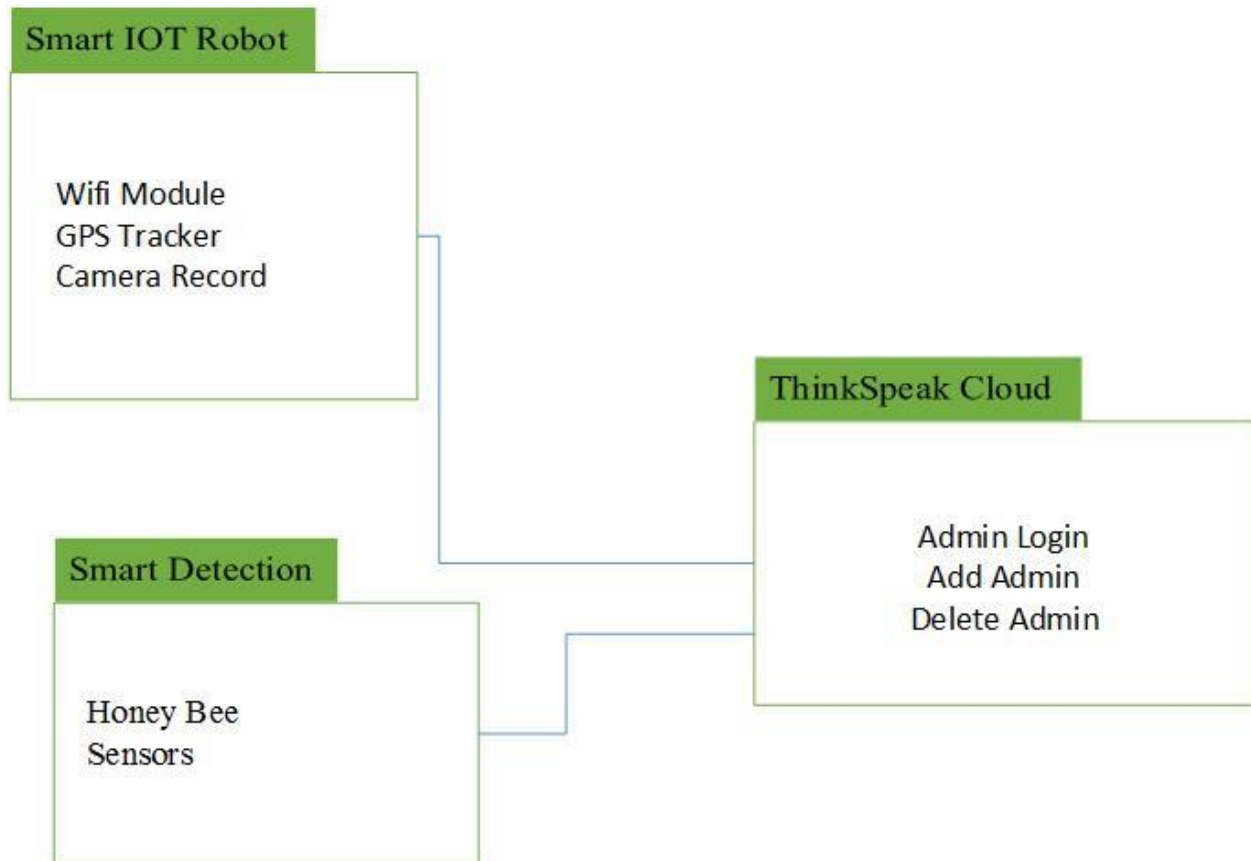


Fig 4.3 Class Diagram: CD_1

4.5. Sequence / Collaboration Diagram

4.5.1 System Functionality: UC_01

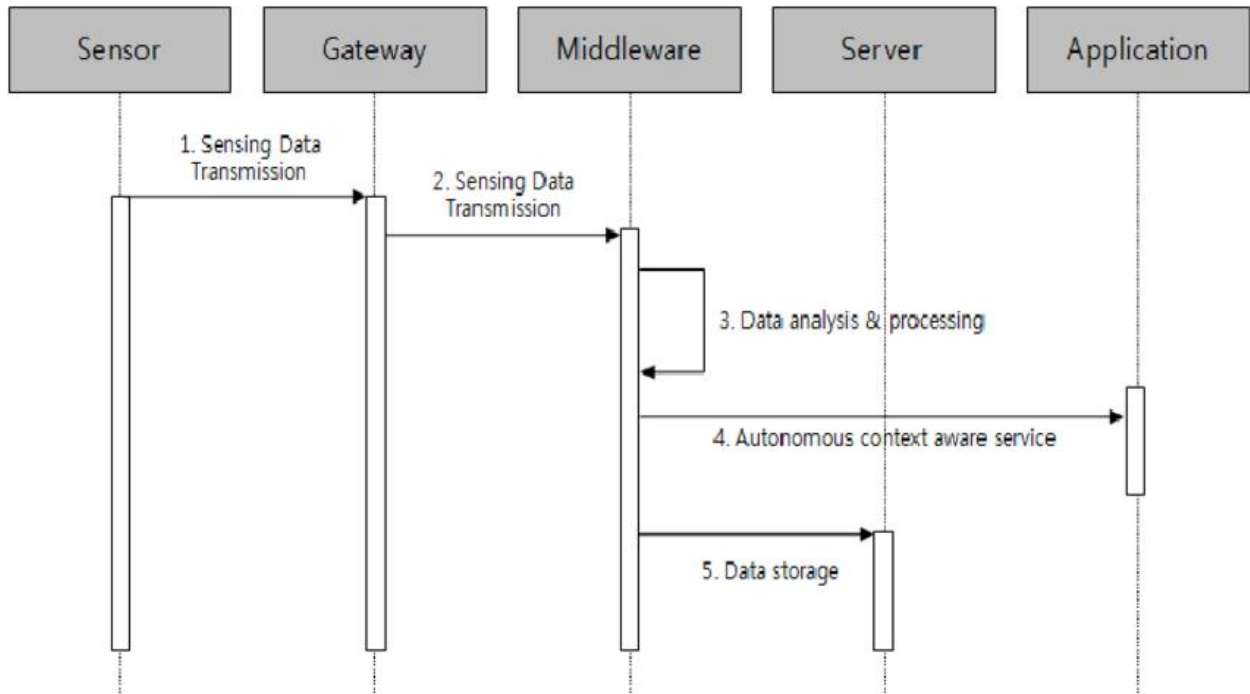


Fig 4.4.1 System Functionality: UC_01

4.5.2 Admin login: UC_02

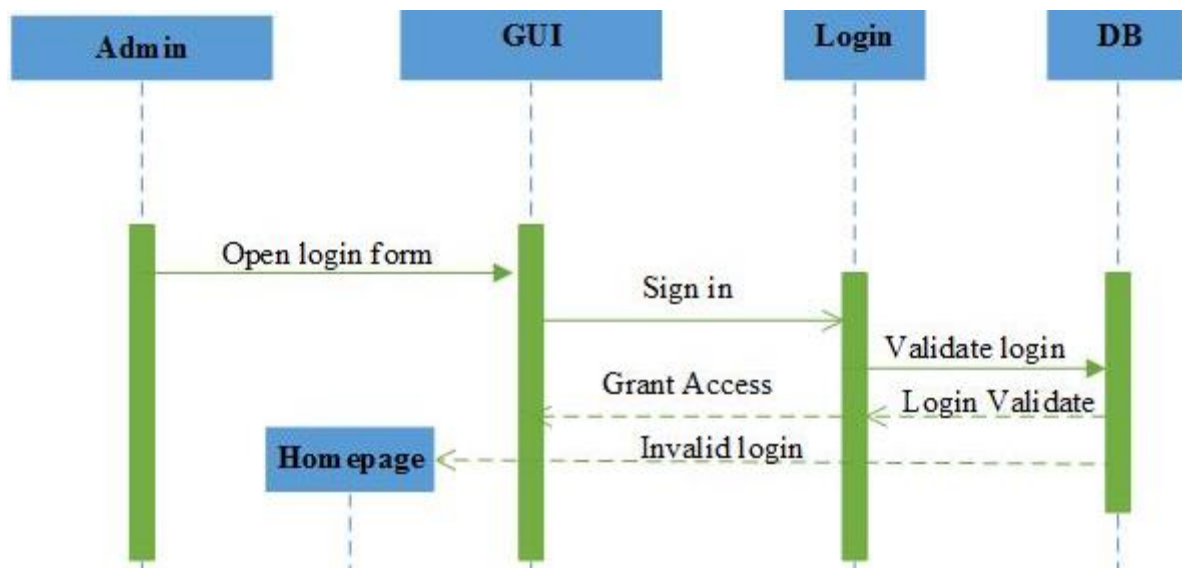


Fig 4.4.2 Admin login: SD_02

4.5.3 Add User: UC_03

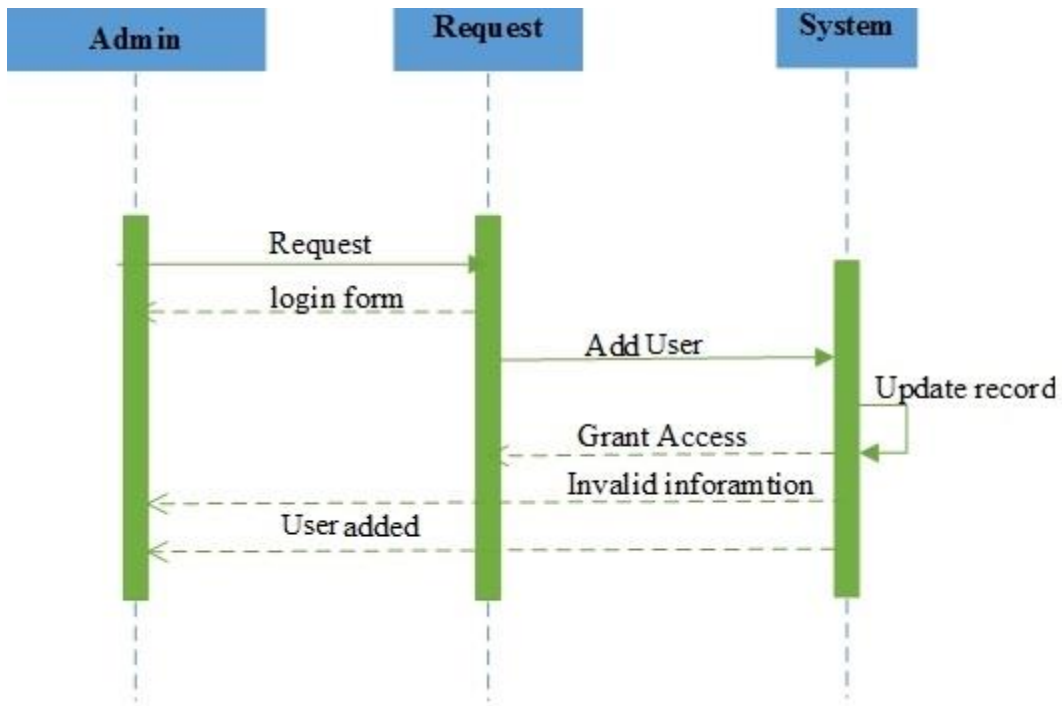


Fig 4.4.3 Add User: UC_03

4.5.4 Delete User: UC_04

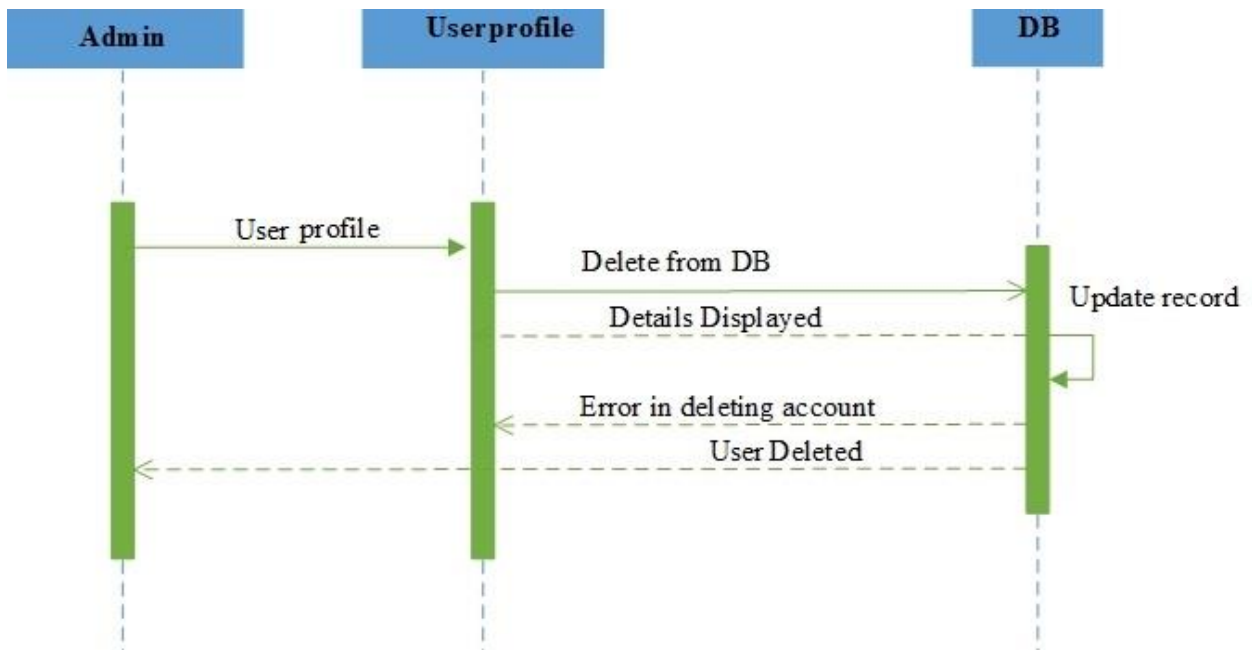
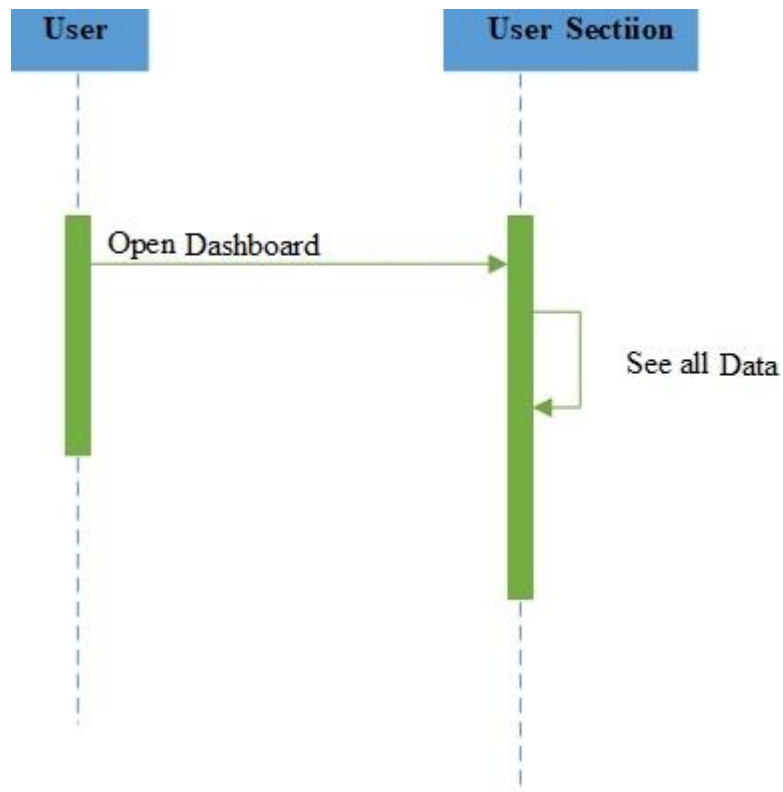


Fig 4.4.4 Delete User: UC_04

4.5.5 Monitor Record: UC_05



4.4.5 Monitor Record: UC_05

4.6. Operation contracts

N/A

4.7. Activity Diagram

4.7.1. Admin login: UC_01

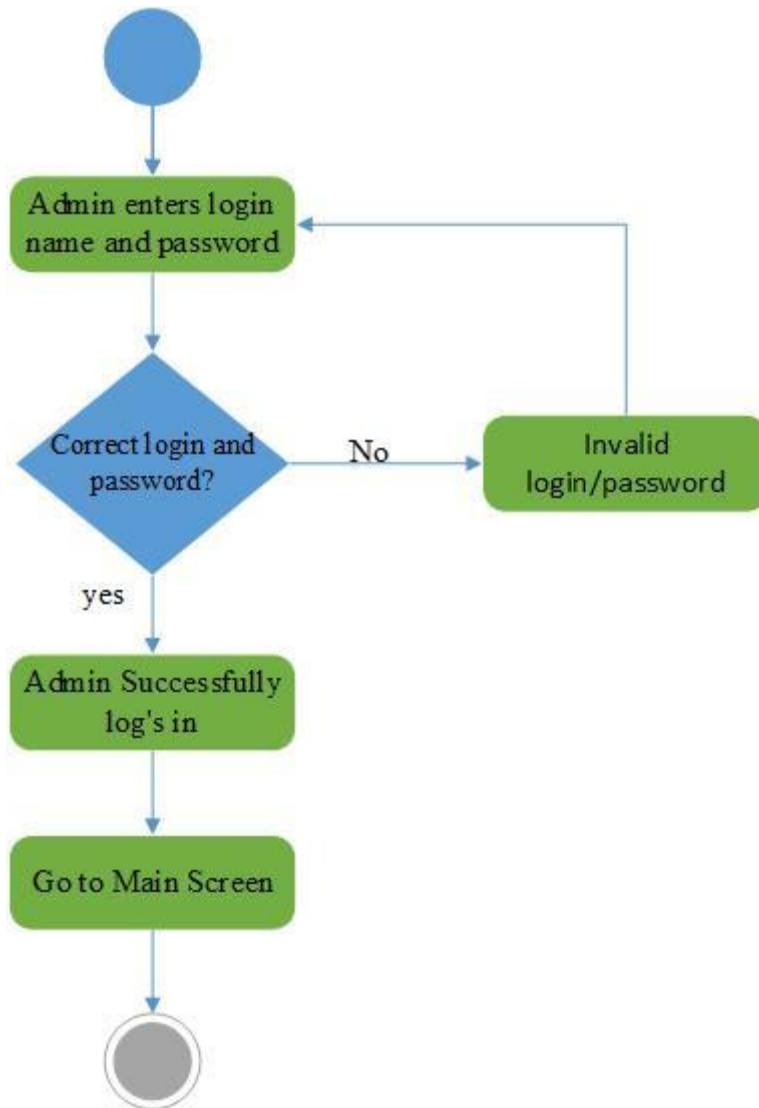


Fig 4.5.1 Admin login: UC_01

4.7.2. Add User: UC_02



Fig 4.5.2 Add User: UC_02

4.7.3. User login: UC_03

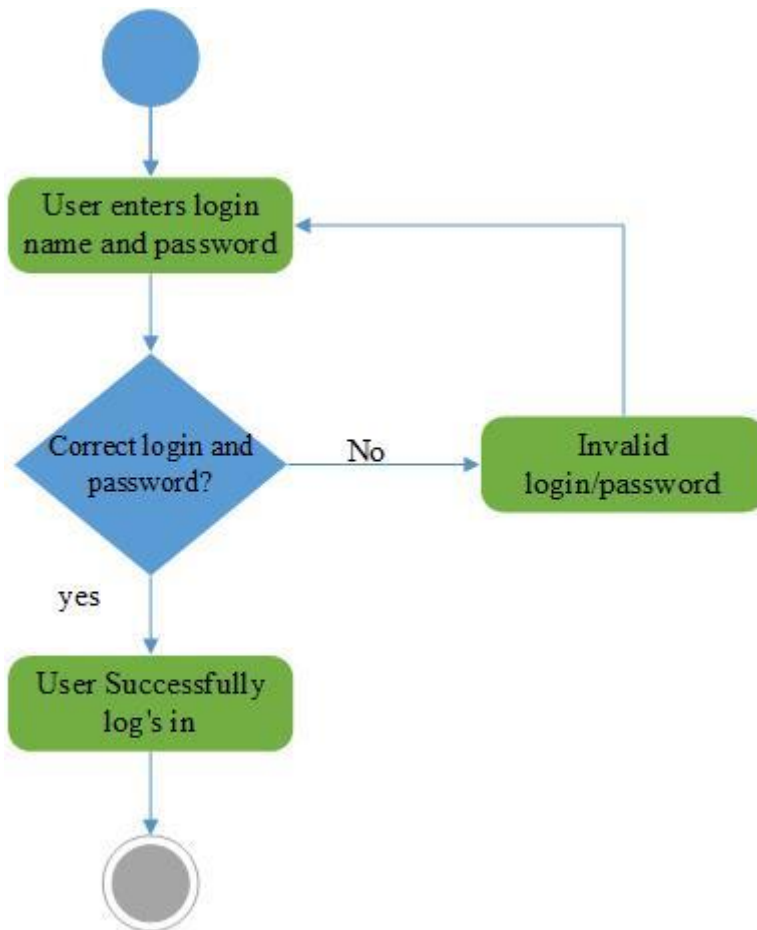


Fig 4.5.3 User login: UC_03

4.7.4. Delete User: UC_04

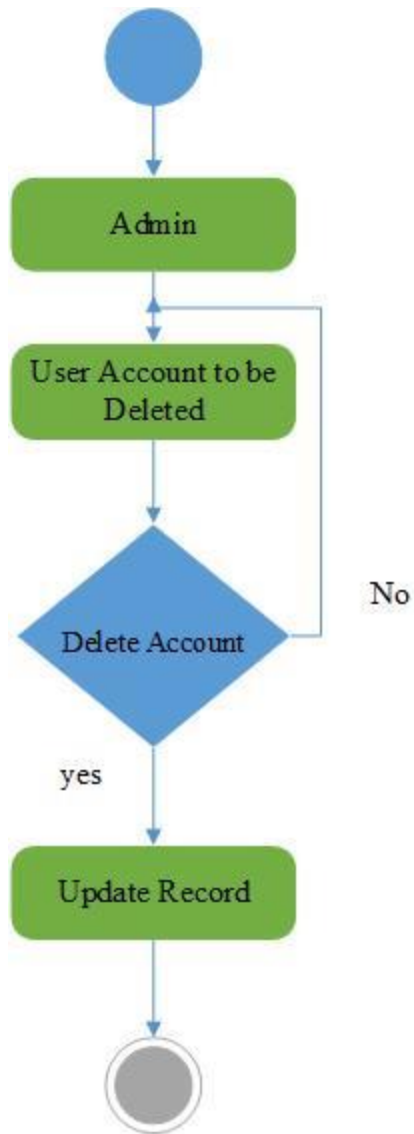


Fig 4.5.4 Delete User: UC_04

4.7.5. User Data Monitor: UC_05

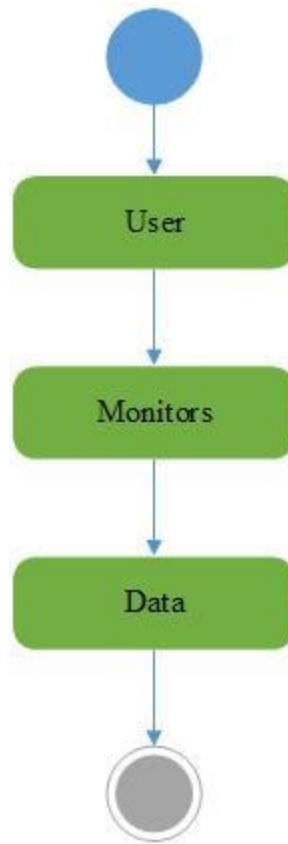


Fig 4.5.5 User Data Monitor: UC_05

4.7.6. Security Agency Monitor Data: UC_06

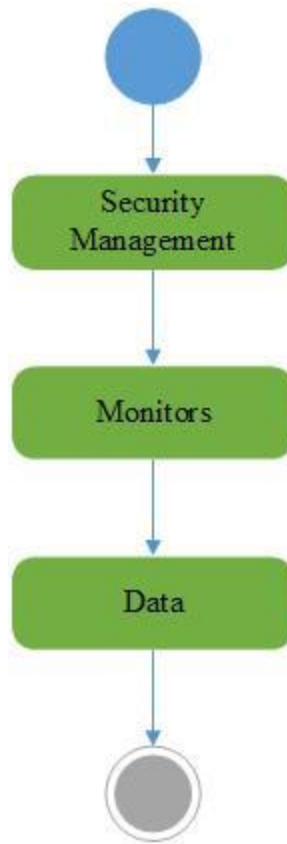


Fig 4.5.6 Security Agency Monitor Data: UC_06

4.7.7. Explosive Detection: UC_07

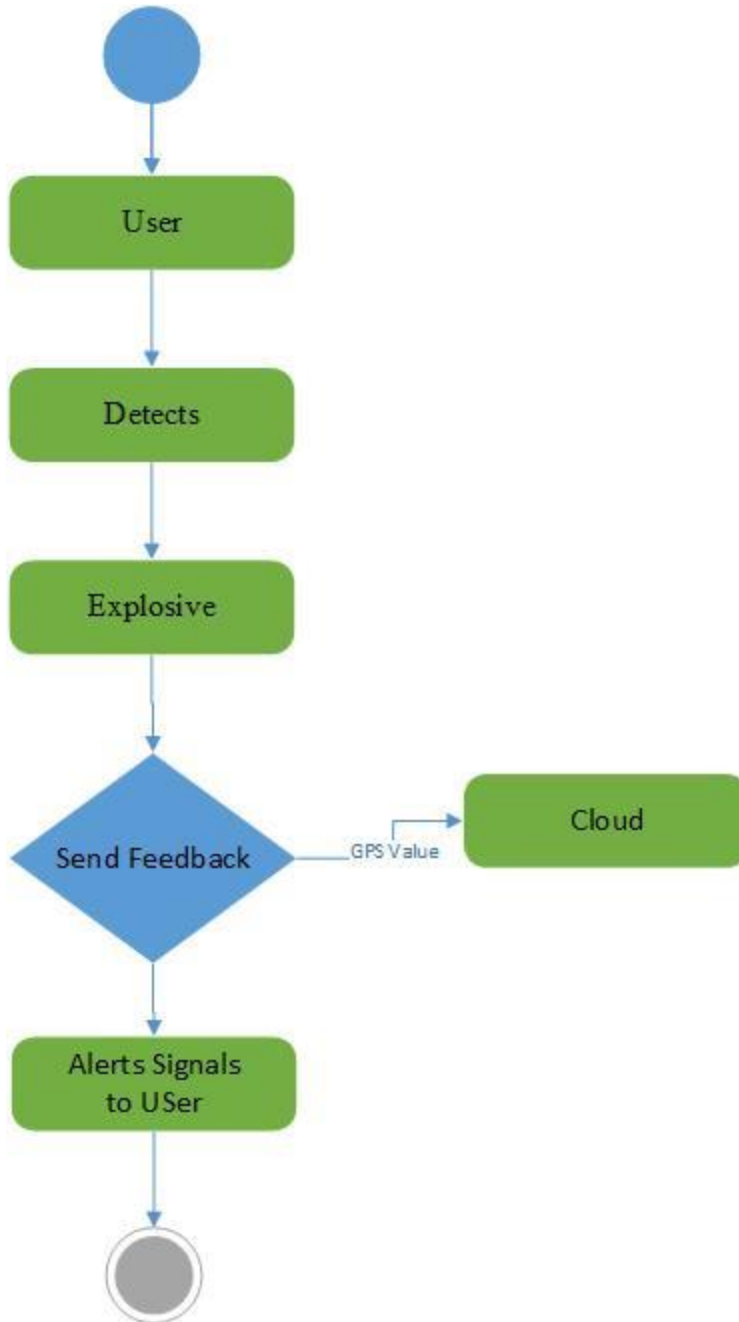


Fig 4.5.7 Explosive Detection: UC_07

4.7.8. Face Detection/Recognition : UC_08

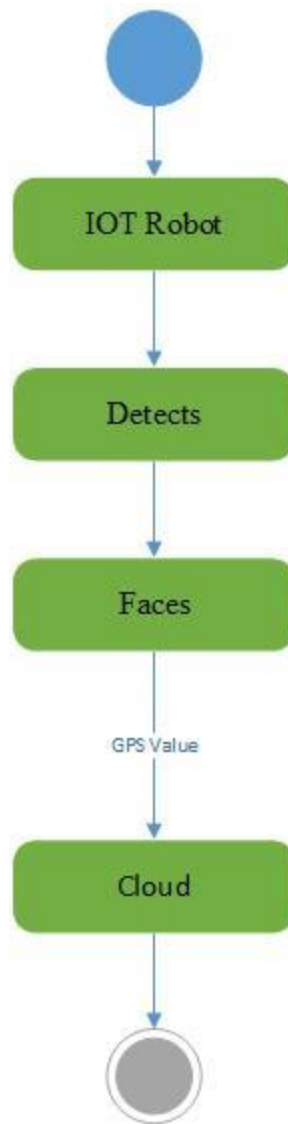


Fig 4.5.8 Face Detection/Recognition: UC_08

4.8. State Transition Diagram

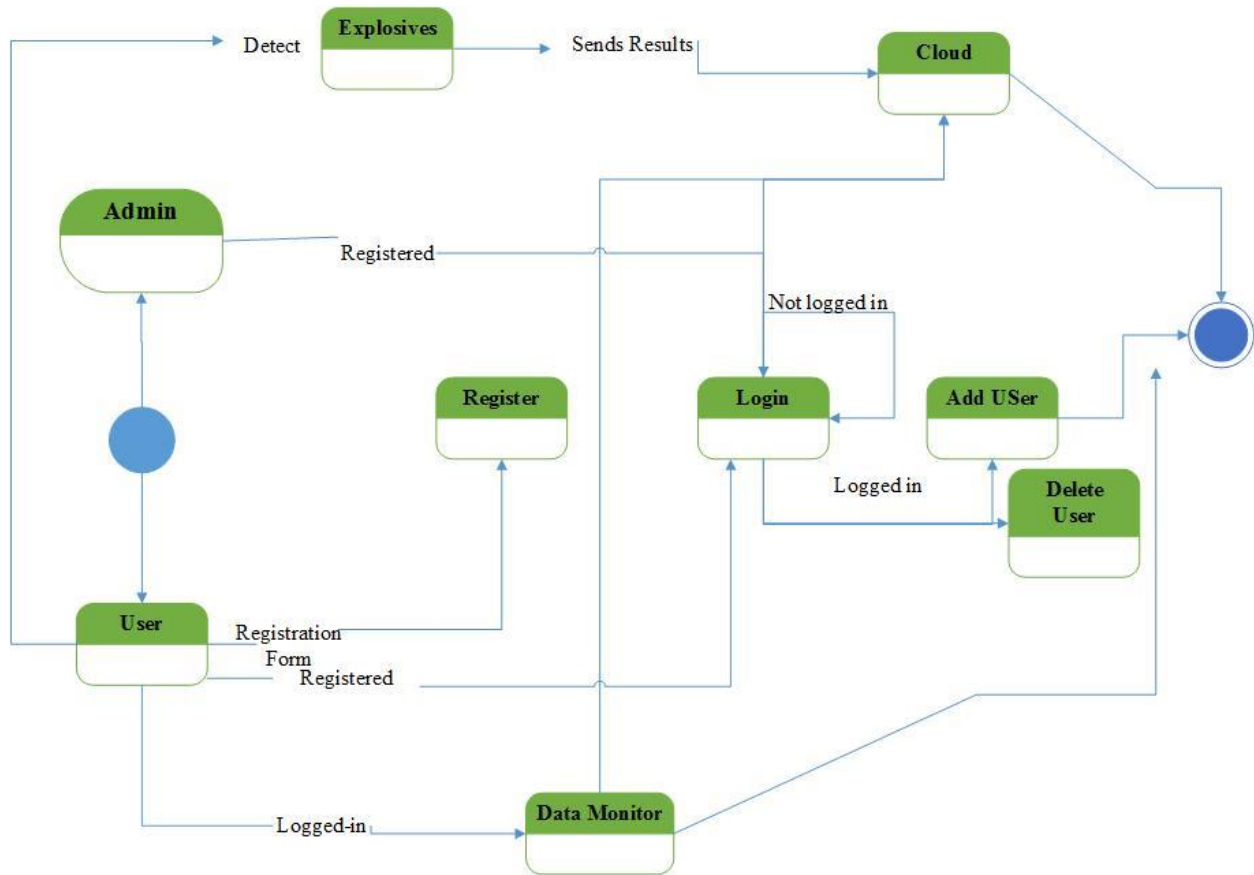


Fig 4.6 State Transition Diagram_01

4.9. Component Diagram

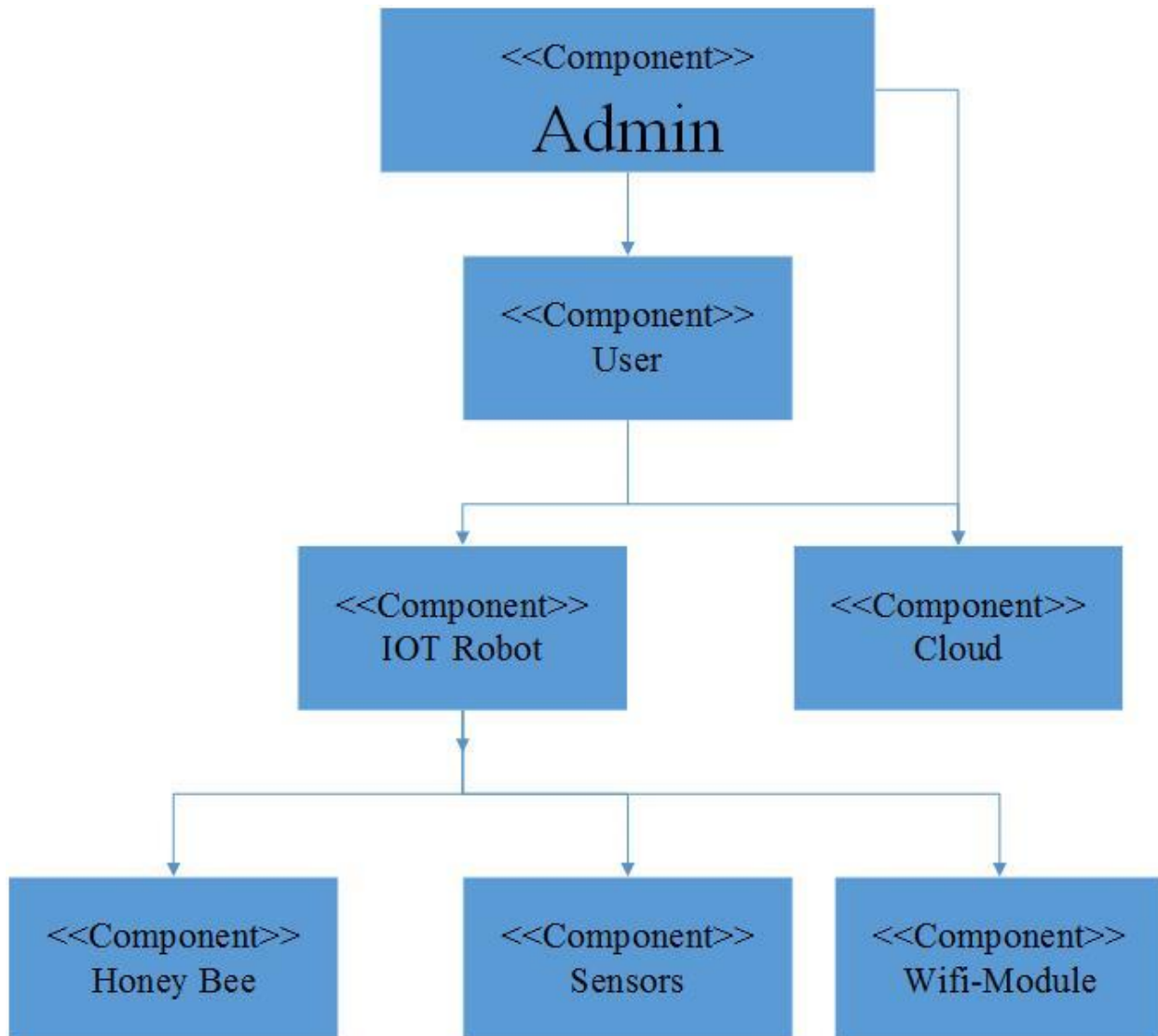


Fig 4.7 Component Diagram_01

4.10. Deployment Diagram

N/A

Chapter 5

Implementation

Chapter 5: Implementation

This chapter deals with the detailed hardware and software description and implementation of the various sensors and components.

5.1. Important Flow Control/Pseudo codes

Explosive Detector:

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial BTserial(10, 11);
```

```
extern volatile unsigned long timer0_millis;
```

```
unsigned long new_value = 0;
```

```
int pushButton = 8;
```

```
unsigned long time;
```

```
int check=0;
```

```
int res_check=0;
```

```
int breaks=0;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    BTserial.begin(9600);
```

```
    pinMode(pushButton, INPUT);
```

```
    pinMode(13,OUTPUT);
```

```
    pinMode(53,OUTPUT);
```

```
}
```

```
void loop() {
```

```
if(breacks==0)
{

    setMillis(new_value);
}
Serial.print("Time: ");
time = millis();
Serial.println(time);

int buttonState = digitalRead(pushButton);

if(buttonState==HIGH)
{
    Serial.println("ok");
    digitalWrite(13,LOW);
    check=0;
}
else if(check==0)
{
    breacks++;
    BTserial.print("Break");
    BTserial.print(breacks);
    BTserial.print(",");
    digitalWrite(13,HIGH);
    check=1;
```

```
}

if(breacks>4&&time<20000)
{
    BTserial.print("Result Positive");
    digitalWrite(53,HIGH);

}

if(time>20000){
    setMillis(new_value);
    breacks=0;
    digitalWrite(53,LOW);

}

delay(1);
}

void setMillis(unsigned long new_millis){
    uint8_t oldSREG = SREG;
    cli();
    timer0_millis = new_millis;
    SREG = oldSREG;
}
```

Human Tracker Arduino:

```
int m1f = 10;
int m1r = 9;
//left motor
int m2f = 6;
int m2r = 5;
int led = 13;
char value;
int val[3];
int len;
int direc;

void setup()
{
  //start serial communication at Baud rate of 9600
  Serial.begin(9600);
  pinMode(m1f, OUTPUT);
  pinMode(m1r, OUTPUT);
  pinMode(m2f, OUTPUT);
  pinMode(m2r, OUTPUT);
  pinMode(led, OUTPUT);
  digitalWrite(led, HIGH);
}

void execute()
{
  int no;
  int i = 0;
```

```
//convert ASCII value from serial buffer into int
```

```
no = value - '0';
```

```
//Serial.println(no);
```

```
int a, b, c, d;
```

```
/*
```

```
****
```

Direction control Index:

'<' , '>' are the frame check bits for serial communication

Numbers represent the direction to be moved as per their position on numpad

1: Back Left

2: Back

3: Back right

4: Left

5: Stay still

6: Right

7: Front Left

8: Forward

9: Forward right

```
****
```

```
*/
```

```
switch (no)
```

```
{
```

```
case 0:
```

a = 0; b = 100; c = 100; d = 0;

break;

case 3:

a = 0; b = 255; c = 0; d = 100;

break;

case 2:

a = 0; b = 255; c = 0; d = 255;

break;

case 1:

a = 0; b = 100; c = 0; d = 255;

break;

case 4:

a = 255; b = 0; c = 0; d = 0;

break;

case 5:

a = 0; b = 0; c = 0; d = 0;

break;

case 6:

a = 0; b = 0; c = 255; d = 0;

break;

case 7:

a = 255; b = 0; c = 100; d = 0;

break;

case 8:

a = 150; b = 0; c = 150; d = 0;

break;

case 9:

a = 100; b = 0; c = 255; d = 0;

break;

```
}

analogWrite(m1f, a);
analogWrite(m1r, b);
analogWrite(m2f, c);
analogWrite(m2r, d);
//delay(255);
}

void loop()
{
  if (Serial.available())
  {
    //Check for frame control bits
    char ch = Serial.read();
    if (ch == '<')
    {
      len = 0;
    }
    else if (ch == '>')
    {
      execute();
      //Serial.println(value);
      len = 1;
    }
    else if (len == 0)
      value = ch;
  }
}
```

Python OpenCv Code:

```
# -*- coding: utf-8 -*-
```

```
import numpy as np
```

```
import sys
```

```
import time
```

```
"""
```

```
PySerial library required for Arduino connection
```

```
OpenCV library required for face tracking
```

```
"""
```

```
import serial
```

```
import cv2
```

```
"""
```

```
Arduino connected at port No. COM28,
```

```
Confirm and change this value accordingly from the control panel
```

```
Baud Rate = 9600
```

```
"""
```

```
arduino = serial.Serial('COM28', 9600)
```

```
time.sleep(2)
```

```
# waiting the initialization...
```

```
print("initialised")
```

```
#gets the direction for Arduino serial
```

```
def direction(bound, initArea=40000):
```

```
    """
```

```
        Direction control Index:
```

```
'<' , '>' are the frame check bits for serial communication
```

```
    Numbers represent the direction to be moved as per their position on numpad
```

```
1: Back Left
```

```
2: Back
```

```
3: Back right
```

```
4: Left
```

```
5: Stay still
```

```
6: Right
```

7: Front Left

8: Forward

9: Forward right

""""

#anchor the center position of the image

center=(320, 240)

#current rectangle center

curr = (bound[0] + bound[2]/2, bound[1]+bound[3]/2)

out=0

flag=0

fb = 0

#0-stay 1-fwd 2-bwd

lr = 0

#0-stay 1-left 2-right

#if the object is coming closer

i.e. it's size is increasing then move bwd

```
if bound[2]*bound[3] > (initArea+5000) or bound[1]<50 :
```

```
    fb = 2
```

```
#if the object os moving away i.e. its size is decreasing then move towards it
```

```
    elif bound[2]*bound[3] < (initArea-5000) or (bound[1]+bound[3])>430 :
```

```
        fb = 1
```

```
    else :
```

```
        fb = 0
```

```
#move right
```

```
if curr[0] > (center[0] + 100):
```

```
    lr = 2
```

```
#move left
```

```
elif curr[0] < (center[0] - 100):
```

```
    lr = 1
```

```
else:
```

```
    lr = 0
```

```
if lr == 0 and fb == 0:
```

```
    out = 5
```

```
    print "stay"
```

```
elif lr == 0 and fb == 1:
```

```
    out =8
```

```
    print "fwd"
```

```
elif lr == 0 and fb == 2:
```

```
    out = 2
```

```
    print "back"
```

```
elif lr == 1 and fb == 0:
```

```
    out = 4
    print "left"
elif lr == 1 and fb == 1:
    out = 7
    print "fwd left"
elif lr == 1 and fb == 2:
    out = 1
    print "left back"
elif lr == 2 and fb == 0:
    out = 6
    print "right"
elif lr == 2 and fb == 1:
    out = 9
    print "fwd right"
elif lr == 2 and fb == 2:
    out = 3
    print "bwd right"
else :
    out = 5
    print "Stay Still"

#Write the encoded direction value on the serial communication line
print out
arduino.write('<')
arduino.write(str(out))
arduino.write('>')

def detectAndDisplay(frame):
    #use OpenCV HAAR face detetcion algorithm to detect faces
```

```

faces = cascade.detectMultiScale(frame, scaleFactor=1.1, minNeighbors=3,
                                minSize=(30, 30),maxSize=(500,500),
                                flags=cv2.cv.CV_HAAR_SCALE_IMAGE)

#if any face is detected then process else continue searching
if(len(faces)!=0):
    #If number of faces in the image is more than 1
    #Then choose the one with maximum size
    max_area=-1
    i=0
    for (x,y,w,h) in faces:
        if w*h > max_area:
            max_area=w*h
            pos=i
        i=i+1

    RECT=faces[pos]
    #Mark the face being tracked on the image display
    cv2.rectangle(frame, (RECT[0], RECT[1]), (RECT[0]+RECT[2], RECT[1]+RECT[3]), (0, 255, 0),
2)
    #draw_str(frame, (RECT[0], RECT[3]+16), 'x: %.2f y: %.2f size: %.2f' % (RECT[2]-RECT[0])/2
% (RECT[3]-RECT[1])/2 % RECT[2]*RECT[3])

    #Put the text details about the ROI on imdisplay
    cv2.putText(frame, `RECT[0] + RECT[2]/2`+' '+'RECT[1]+RECT[3]/2`+' '+'RECT[2]*RECT[3]`,
(RECT[0],RECT[1]+RECT[3]), cv2.FONT_HERSHEY_SIMPLEX , 1, (0,0,255));

    #compute direction for the arduino bot to be moved.
    direction(RECT)

```

```
else:
```

```
    print 'Search...'
```

```
    arduino.write('<')
```

```
    arduino.write(str(5))
```

```
    arduino.write('>')
```

```
cv2.imshow('frame',frame)
```

```
cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

```
#cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')
```

```
cap = cv2.VideoCapture(1)
```

```
cap.grab()
```

```
ret, frame = cap.retrieve()
```

```
cv2.namedWindow('frame')
```

```
#Run the tracker in infinite loop
```

```
while(1):
```

```
    #grab the frames from web camera
```

```
    ret, frame = cap.retrieve()
```

```
    if ret ==0:
```

```
        print "frame not loaded"
```

```
    if ret==True:
```

```
        #Resize the frame for faster computation
```

```
        #cv2.resize(frame,(240,320))
```

```
#Process the frame and pass data to arduino
```

```
detectAndDisplay(frame)
```

```
#cv2.imshow('input',frame)
```

```
#press ESC to exit program
```

```
ch = cv2.waitKey(1)
```

```
if ch==27:
```

```
    break
```

```
#Free up memory on exit
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

```
arduino.close()
```

Python Face Detection:**Face Detection:**

```
from __future__ import print_function
```

```
import numpy as np
```

```
import cv2 as cv
```

```
from video import create_capture
```

```
from common import clock, draw_str
```

```
def detect(img, cascade):
```

```
    rects = cascade.detectMultiScale(img, scaleFactor=1.3, minNeighbors=4, minSize=(30, 30),  
                                     flags=cv.CASCADE_SCALE_IMAGE)
```

```
    if len(rects) == 0:
```

```
        return []
```

```
    rects[:,2:] += rects[:,2:]
```

```
    return rects
```

```
def draw_rects(img, rects, color):
```

```
    for x1, y1, x2, y2 in rects:
```

```
        cv.rectangle(img, (x1, y1), (x2, y2), color, 2)
```

```
if __name__ == '__main__':
```

```
    import sys, getopt
```

```
    print(__doc__)
```

```
args, video_src = getopt.getopt(sys.argv[1:], '', ['cascade=', 'nested-cascade='])
try:
    video_src = video_src[0]
except:
    video_src = 0
args = dict(args)
cascade_fn = args.get('--cascade',
"../data/haarcascades/haarcascade_frontalface_alt.xml")
nested_fn = args.get('--nested-cascade', "../data/haarcascades/haarcascade_eye.xml")

cascade = cv.CascadeClassifier(cascade_fn)
nested = cv.CascadeClassifier(nested_fn)

cam = create_capture(video_src, fallback='synth:bg=../data/lena.jpg:noise=0.05')

while True:
    ret, img = cam.read()
    gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
    gray = cv.equalizeHist(gray)

    t = clock()
    rects = detect(gray, cascade)
    vis = img.copy()
    draw_rects(vis, rects, (0, 255, 0))
    if not nested.empty():
        for x1, y1, x2, y2 in rects:
            roi = gray[y1:y2, x1:x2]
            vis_roi = vis[y1:y2, x1:x2]
            subrects = detect(roi.copy(), nested)
```

```
    draw_rects(vis_roi, subrects, (255, 0, 0))
dt = clock() - t

draw_str(vis, (20, 20), 'time: %.1f ms' % (dt*1000))
cv.imshow('facedetect', vis)

if cv.waitKey(5) == 27:
    break
cv.destroyAllWindows()
```

```
Python Face Recognition: import face_recognition
import cv2

# 1. Process each video frame at 1/4 resolution (though still display it at full resolution)
# 2. Only detect faces in every other frame of video.

# Get a reference to webcam #0 (the default one)
video_capture = cv2.VideoCapture(0)

# Load a sample picture and learn how to recognize it.
Hammad_image = face_recognition.load_image_file("Hammad.JPG")
Hammad_face_encoding = face_recognition.face_encodings(Hammad_image)[0]

# Load a second sample picture and learn how to recognize it.
Khuzema_image = face_recognition.load_image_file("Khuzema.JPG")
Khuzema_face_encoding = face_recognition.face_encodings(Khuzema_image)[0]

# Load a third sample picture and learn how to recognize it.
Junaid_image = face_recognition.load_image_file("Junaid.JPG")
Junaid_face_encoding = face_recognition.face_encodings(Junaid_image)[0]

# Create arrays of known face encodings and their names
known_face_encodings = [
    Hammad_face_encoding,
    Khuzema_face_encoding,
    Junaid_face_encoding
]
known_face_names = [
    "Hammad Saeed",
```

```
"Muhammad Khuzema",
"Junaid Jaffar"
]

# Initialize some variables
face_locations = []
face_encodings = []
face_names = []
process_this_frame = True

while True:
    # Grab a single frame of video
    ret, frame = video_capture.read()

    # Resize frame of video to 1/4 size for faster face recognition processing
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

    # Convert the image from BGR color (which OpenCV uses) to RGB color (which
    face_recognition uses)
    rgb_small_frame = small_frame[:, :, :-1]

    # Only process every other frame of video to save time
    if process_this_frame:
        # Find all the faces and face encodings in the current frame of video
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)

        face_names = []
        for face_encoding in face_encodings:
```

```
# See if the face is a match for the known face(s)
matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
name = "Unknown"

# If a match was found in known_face_encodings, just use the first one.
if True in matches:
    first_match_index = matches.index(True)
    name = known_face_names[first_match_index]

face_names.append(name)

process_this_frame = not process_this_frame

# Display the results
for (top, right, bottom, left), name in zip(face_locations, face_names):
    # Scale back up face locations since the frame we detected in was scaled to 1/4 size
    top *= 4
    right *= 4
    bottom *= 4
    left *= 4

    # Draw a box around the face
    cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

    # Draw a label with a name below the face
    cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
    font = cv2.FONT_HERSHEY_DUPLEX
    cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)
```

```

# Display the resulting image
cv2.imshow('Video', frame)

# Hit 'q' on the keyboard to quit!
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Release handle to the webcam
video_capture.release()
cv2.destroyAllWindows()

UBIDots Cloud:
/*****
* Include Libraries
*****/

#include "UbidotsESPMQTT.h"

#include <SoftwareSerial.h>
#include <TinyGPS.h>
TinyGPS gps;
SoftwareSerial ss(13, 15, false, 256);
/*****
* Define Constants
*****/

#define TOKEN "A1E-Ti4fjn6vatGj2XvZEEGAJGSiHyaLly" // Your Ubidots TOKEN
#define WIFINAME "ITU" //Your SSID
#define WIFIPASS "ITU6THF*" // Your Wifi Pass

#define analogInput A0 //Attach analog input at pin A0 of ESP8266

```

```
#define digitalInput D1 //Attach digital nput at pin D1 of ESP8266
```

```
/******
```

```
* Define variable
```

```
*****/
```

```
int sensor_value = 0;
```

```
int sensor1_value = 0;
```

```
Ubidots client(TOKEN);
```

```
/******
```

```
* Auxiliar Functions
```

```
*****/
```

```
void callback(char* topic, byte* payload, unsigned int length) {
```

```
  Serial.print("Message arrived [");
```

```
  Serial.print(topic);
```

```
  Serial.print("] ");
```

```
  for (int i=0;i<length;i++) {
```

```
    Serial.print((char)payload[i]);
```

```
  }
```

```
  Serial.println();
```

```
}
```

```
/******
```

```
* Main Functions
```

```
*****/
```

```
void setup() {
```

```
// put your setup code here, to run once:
client.ubidotsSetBroker("things.ubidots.com"); // Sets the broker properly for the business
account
client.setDebug(true); // Pass a true or false bool value to activate debug messages

Serial.begin(115200);
ss.begin(9600);
Serial.print("Simple TinyGPS library v. "); Serial.println(TinyGPS::library_version());
Serial.println();

pinMode(analogInput, INPUT);
pinMode(digitalInput, INPUT);

client.wifiConnection(WIFINAME, WIFIPASS);
client.begin(callback);
}

void loop() {
// put your main code here, to run repeatedly:
bool newData = false;
unsigned long chars;
unsigned short sentences, failed;

// For one second we parse GPS data and report some key values
for (unsigned long start = millis(); millis() - start < 1000;)
{
while (ss.available())
{
char c = ss.read();
```

```
//Serial.write(c); // uncomment this line if you want to see the GPS data flowing
if (gps.encode(c)) // Did a new valid sentence come in?
    newData = true;
}
}

if(!client.connected()){
    client.reconnect();
}

if (newData)
{
    float flat, flon;
    unsigned long age;
    gps.f_get_position(&flat, &flon, &age);
    Serial.print("LAT=");
    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
    flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6;
    client.add("lat", flat); //Insert your variable Labels and the value to be sent
    client.ubidotsPublish("testdata");
    Serial.print("LON=");
    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
    flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6;
    client.add("lon", flon); //Insert your variable Labels and the value to be sent
    client.ubidotsPublish("testdata");

/*
    Serial.print(" SAT=");
    Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 : gps.satellites());
```

```
Serial.print(" PREC=");  
Serial.print(gps.hdop() == TinyGPS::GPS_INVALID_HDOP ? 0 : gps.hdop());*/  
}  
  
delay(1000);  
client.loop();  
}
```

5.2. Components, Libraries, Web Services and stubs

Libraries and Web Services:

- WI-FI ESP-8266
- Open CV
- NumPy
- Cv2
- UBIDOTS

Components:

- Arduino UNO WIFI
- Sensor
- Bluetooth HC-05
- GPS

5.3. Deployment Environment

N/A

5.4. Tools and Techniques

During the development of this project, we used the following tools and technologies:

- Corel Draw, Adobe Photoshop & Adobe Illustrator used for making things just like images, posters, logos etc.

- Arduino software Arduino software used to compile your program and then upload it on the microcontroller.
- Python with open CV used for face detection and recognition.
- The Arduino Uno Wi-Fi is an Arduino Uno with an integrated Wi-Fi module. The board is based on the ATmega328P with an ESP8266 Wi-Fi Module integrated. The ESP8266 Wi-Fi Module is a self-contained SoC with integrated TCP/IP protocol stack that can give access to your Wi-Fi network (or the device can act as an access point). One useful feature of Uno Wi-Fi is support for OTA (over-the-air) programming, either for transfer of Arduino sketches or Wi-Fi firmware.
- Microsoft

5.5. Best Practices / Coding Standards

Naming Conventions

The naming conventions in Arduino IDE we used:

Variables: begins with the lowercase letters.

Classes: Classes are declared with the uppercase letter. The classes are the collection of functions.

Functions: Functions are starts with uppercase letters. Functions are a collection of code and they are written once and they can be used as they need in the program.

Inline Comments: Inline comments explaining the functionalities.

Use of Braces

- Braces are used to delimit the bodies of conditional statements, control constructs, and blocks of scope.
- Braces shall be used even when there is only one statement in the control block.

Compiler Warnings

- Compiler and linker warnings shall be treated as errors and fixed because they often indicate problems which may affect the behavior, reliability, and portability of the code.
- The options fully enable the compiler's code-checking features.

5.6. Version Control

Arduino provides us services to build any project as a team in different ways but rather than using this we individually complete our tasks in our separate machines and the finally joined all this and burn code from one single machine in which our complete project is up and burned on board. For further any update we fix this issue in the same way we did before.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

6.1. Use Case Testing

6.1.1. Admin login

Table no. 6.1.1

Positive Test Case	
ID	TC_01
Priority	High
Description	Admin should be logged in after entering the correct email and password if the user is registered.
Reference	Functional Requirement reference.
User	Admin
Pre-Conditions	<ul style="list-style-type: none"> (i) System is online (ii) Login form should be working (iii) The user should have internet access
Steps	<ul style="list-style-type: none"> (i) Open the web link to the system (ii) Enter username (iii) Enter the password (iv) Press login
Input Criteria	Username: xyz Password: *****
Expected Result	Admin panel will be open.
Status	Pass

Table no. 6.1.2

Negative Test Case	
ID	TC_01
Priority	High

Description	Admin should not be logged in after entering the incorrect email and password if the user is registered.
Reference	Functional Requirement reference.
User	Admin
Pre-Conditions	(i) System is online (ii) Login form should be working (iii) The user should have internet access
Steps	(i) Open the web link to the system (ii) Enter username (iii) Enter the password (iv) Press login
Input Criteria	Username: abc Password: ****
Expected Result	Admin panel will not be opened.
Status	Pass

6.1.2. Add User

Table no. 6.2.1

Positive Test Case	
ID	TC_02
Priority	High
Description	Admin will add new Users
Reference	Functional Requirement reference
User	Admin
Pre-Conditions	(i) The system should be working (ii) Admin should be logged in

Steps	(i) Admin will be logged in to the system (ii) Admin will check new authenticated Users (iii) Click on Add User button.
Input Criteria	Click the “Add User” button
Expected Result	User added in the system
Status	Pass

Table no. 6.2.2

Negative Test Case	
ID	TC_02
Priority	High
Description	Admin will add new Users
Reference	Functional Requirement reference
User	Admin
Pre-Conditions	(i) The system should be working (ii) Admin should be logged in
Steps	(i) Admin will be logged in to the system (ii) Admin will check new authenticated Users (iii) Click on Add User button.
Input Criteria	Not Click on the “Add User” button
Expected Result	User not added in the system
Status	Pass

6.1.3. Delete User account

Table no. 6.3.1

Positive Test Case	
ID	TC_03
Priority	High
Description	Admin delete a User account
Reference	Functional Requirement reference
User	Admin
Pre-Conditions	(i) Admin should be logged in (ii) There should be an existing account which admin wants to delete
Steps	(i) Admin will be logged in to the system (ii) Admin will check whether there is an existing account or not (iii) Click on the “Delete User” button
Input Criteria	Click the “Delete User” button
Expected Result	User deleted from the system
Status	Pass

Table no. 6.3.2

Negative Test Case	
ID	TC_03
Priority	High
Description	Admin delete a User account
Reference	Functional Requirement reference
User	Admin
Pre-Conditions	(i) Admin should be logged in (ii) There should be an existing account which admin wants to delete

Steps	(i) Admin will be logged in to the system (ii) Admin will check whether there is an existing account or not (iii) Click on the “Delete User” button
Input Criteria	Not Click the “Delete User” button
Expected Result	User not deleted from the system
Status	Pass

6.1.4. Sensors

Table no. 6.4.1

Positive Test Case	
ID	TC_04
Priority	High
Description	The User wants to detect.
Reference	Functional Requirement reference
User	User
Pre-Conditions	(iii) The user should know how to use sensors. (iv) All sensors should be placed properly.
Steps	(iv) The user checks the sensors. (v) All sensors will be generate output and sends feedback to the user
Input Criteria	Placed honey bee between sensors
Expected Result	sensors will be generate output and sends feedback to the user
Status	Pass

Table no. 6.4.2

Negative Test Case

ID	TC_04
Priority	High
Description	The User wants to detect.
Reference	Functional Requirement reference
User	User
Pre-Conditions	(v) The user should know how to use sensors. (vi) All sensors should be placed properly.
Steps	(vi) The user checks the sensors. (vii) All sensors will be generate output and sends feedback to the user
Input Criteria	Placed honey bee between sensors
Expected Result	Honeybee or sensor may not place correctly
Status	Pass

6.1.5. GPS Tracking

Table no. 6.5.1

Positive Test Case	
ID	TC_05
Priority	High
Description	The User wants to monitor GPS Tracking
Reference	Functional Requirement reference
User	User
Pre-Conditions	(vii) The user must be logged in
Steps	(viii) The user selects the "GPS" option (ix) All GPS values will be displayed.

Input Criteria	User Selects GPS from Dashboard
Expected Result	The user will locate the current location of Robot
Status	Pass

Table no. 6.5.2

Negative Test Case	
ID	TC_05
Priority	High
Description	The User wants to monitor GPS Tracking
Reference	Functional Requirement reference
User	User
Pre-Conditions	(viii) The user must be logged in
Steps	(x) The user selects the "GPS" option (xi) All GPS values will be displayed.
Input Criteria	User Selects GPS from Dashboard
Expected Result	The user will locate the current location of Robot
Status	Pass

6.1.6. Surveillance

Table no. 6.6.1

Positive Test Case	
ID	TC_02
Priority	High
Description	The User wants to Investigate through using Face Recognition

Reference	Functional Requirement reference
User	User
Pre-Conditions	(ix) The user must be logged in (x) The camera must be connected to WI-FI
Steps	(xii) The camera will transmit video on Cloud (xiii) The user can monitor data
Input Criteria	Click the “Camera” button
Expected Result	The user will see the face detection and recognition process
Status	Pass

Table no. 6.6.2

Negative Test Case	
ID	TC_06
Priority	High
Description	The User wants to Investigate through using Face Recognition
Reference	Functional Requirement reference
User	User
Pre-Conditions	(xi) The user must be logged in (xii) The camera must be connected to WI-FI
Steps	(xiv) The camera will transmit video on Cloud (xv) The user can monitor data
Input Criteria	Click the “Camera” button
Expected Result	Camera or waif may not work.
Status	Pass

Table no. 6.6.2

6.1.7. Wireless Communication

Table no. 6.7.1

Positive Test Case	
ID	TC_07
Priority	High
Description	The User wants to <u>check to</u> communicate wirelessly.
Reference	Functional Requirement reference
User	User
Pre-Conditions	(xiii) WI-FI must be connected
Steps	(xvi) Detection process should be completed. (xvii) All values should be sent to Cloud
Input Criteria	Connect the WI-FI for communication with the cloud
Expected Result	Data will be sent
Status	Pass

Table no. 6.7.2

Negative Test Case	
ID	TC_07
Priority	High
Description	The User wants to check to communicate wirelessly.
Reference	Functional Requirement reference
User	User
Pre-Conditions	(xiv) WI-FI must be connected

Steps	(xviii) Detection process should be completed. (xix) All values should be sent to Cloud
Input Criteria	Connect the WI-FI for communication with the cloud
Expected Result	Data may not be sent
Status	Pass

6.1.8. Explosive Detection

Table no. 6.8.1

Positive Test Case	
ID	TC_08
Priority	High
Description	Honey Be to be used for Explosive Detection
Reference	Functional Requirement reference
User	User
Pre-Conditions	(xv) Honey Bee must be trained (xvi) Sensors must be placed correctly
Steps	(xx) Honey Bee Sniffs (xxi) Signal will be generated
Input Criteria	Move Robot in multiple directions
Expected Result	Results will be sent an alert generated
Status	Pass

Table no. 6.8.2

Negative Test Case

ID	TC_08
Priority	High
Description	Honey Be to be used for Explosive Detection
Reference	Functional Requirement reference
User	User
Pre-Conditions	(xvii) Honey Bee must be trained (xviii) Sensors must be placed correctly
Steps	(xxii) Honey Bee Sniffs (xxiii) Signal will be generated
Input Criteria	Move Robot in multiple directions
Expected Result	Results may not be sent and no alert generated
Status	Pass

6.2. Equivalence partitioning

Add User

Table no. 6.9.1

Field Name	Data type	Invalid	Valid	Invalid
First Name	String	Less than 1, no numeric, special characters	Character 1-15	More than 15 not accept
Last Name	String	Less than 1, no numeric, special characters	Character 1-15	More than 15 not accept
Email	Email	Missing '.'	Must be valid email type	Missing '@'
Password	Varchar	Less than 8 not accepted	Alphanumeric characters >=8	Less than 8 not accepted

Phone	Digits	Less than and greater than 13 not accepted	13 number digits	First, 2 digits should be '03'
CNIC	Digits	Less than 15 not accepted	CNIC Number pattern in numeric	Greater than 15 not accepted

Login

Table no. 6.9.2

Field Name	Data type	Invalid	Valid	Invalid
Email	Email	Missing '.'	Must be valid email type	Missing '@'
Password	Varchar	Less than 8 not accepted	Alphanumeric characters >=8	Less than 8 not accepted

Table no. 6.9.2

6.3. Boundary value analysis

N/A

6.4. Data flow testing

Most of the product uses the variables to make the data flow within the program. The program may use multiple variables to calculate values for different variables. Now, these variables get some values from the user interface or sometimes from different sources. This complexity makes a program full of risk. So testing these variable data flow makes sense and Dataflow testing concept can be used to test all the variables from the program.

Levels of Dataflow Testing:

- **Static data flow testing**
 - o Identifying Potential Defects within the program.
 - o Analyzing code for defects

- o The code will not be executed
- **Dynamic data flow testing**
 - o We will execute the program
 - o Looks like a control flow testing
 - o List out the path need test

About Variables:

There are two types of variable in a program called use and definition occurrence.

Definition Occurrence: Values are assigned to variables.

Use Occurrence: Values are used in the program from other variables.

Predicate Use Occurrence: Value of this type of the variable will be used to an execution path.

Computational Use Occurrence: Value of variable will be used to calculate the value of another variable.

Def-Use Pair Variable: It is a pair of Definition Occurrence and Uses Occurrence.

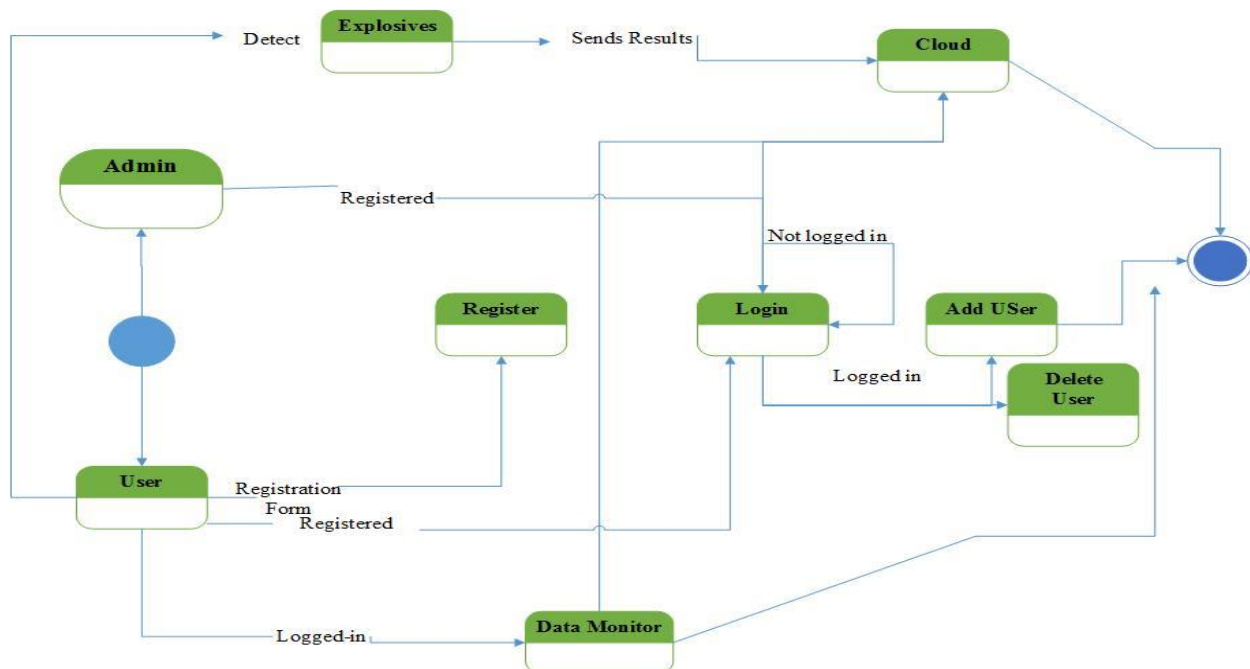


Fig Data Flow Testing 6.4

6.5. Unit testing

A unit testing framework for Arduino platforms inspired by ArduinoUnit and Google Test. The unit tests run in the embedded controller, not in a simulator or emulator. It is almost a drop-in replacement of ArduinoUnit with some advantages. AUnit supports timeouts and test fixtures. It sometimes consumes 50% less flash memory on the AVR platform, and it has been tested to work on the AVR, ESP8266, ESP32 and Teensy platforms. The sister AUniter project provides command line tools to verify upload and validate the unit tests. The AUniter tools can be used in a continuous integration system like Jenkins.

6.6. Integration testing

This project has three parts.

- Smart IOT Robot
- Smart Detection
- Ubidots Cloud

Smart IOT robot moves according to the instructions of the user and inside the robot, the detection unit has honey bee which is used for detection of explosive material and send feedback at UBIdots cloud and also alert to the user.

6.7. Performance testing

Stress testing should also be carried out after the implementation of the product. We shall test the product's robustness, availability, and reliability under extreme conditions. The goal of stress testing is to identify application issues that arise or become apparent only under extreme conditions. We have tested our system by simultaneously using our system as a different user and performing their associated tasks in a continuous manner. Following steps have been followed:

- We have analyzed system behavior under high traffic.
- We have gathered statistics reviews about crashing app, time taking operations and shall resolve those issues in later revisions.

6.8. Stress Testing

Performance testing should be carried out after the implementation of the product. We shall test the speed, accuracy, and effectiveness of functions as well as quantitative test e.g. response time of any function. We have tested the performance of our system by checking the working of every complex operation, again and again, multiple times. We have followed the following steps.

- List of all complex functions especially database related function.
- Assess response time and accuracy of core functions.
- Risk assessment of error or exception code (if any).

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

We as a company wants to be the top producers of systems to explore and detect hazardous material which can harm living being in a way where it's safe for the operators of the devices by incorporating handheld systems which can also be accessed and controlled remotely. One of the most desired milestones of the company is to create a hybrid of living being senses and modern computing to produce intelligent, safe and economic systems. As Vision is to be known as one of the best company in the world as the producers of intelligent products which can detect hazardous materials and react rationally to help save living beings and explore the potential of living senses.

7.2. Achievements and Improvements

During this process of product development we learned a lot of skills:

- Hardware Development
- Working on IOT Robotics.
- Best Coding practices.
- Integration of different objects on a single platform.
- Creative Thinking
- Familiar with animal senses.

With respect to the product, the biggest obstacle we have faced is the awareness of such products in the native market; we have overcome this hinder to some extent by showcasing our idea and prototype at various events where we have found the very encouraging response by the audience.

Moreover, we're going to continue our research on this idea and further working on animal senses under the umbrella of the world's best research centers. Soon we'll launch our product with the best benefits.

7.3. Critical Review

We are awesome as we are persistent that we will aware the globe about the potential of this hybrid of living beings and machines we don't need to replace living beings with machines but we need to incorporate living senses with better computing systems to fully explore the horizons of nature.

It is the demand of the native market and in the whole world, no such product is officially working to reduce such terrorist activities so we are hoping that it'll have the best impact on the market.

7.4. Lessons Learnt

We absorb very much from this project. This project sharpens our skills in hardware development, designing tools, and many management concepts as well as how to deal with a problem and how to stick for finding the solution to any problem until you find. As well as technical skills this project also enhances our personal development skills such as team working, dedication. Our passion for making a unique product that contains innovation and nationwide best achievement and we succeeded it. We continue our work after this to make it through worldwide. This project is a stepping stone for us, in the beginning, we don't even know that if we accomplished it or not but due to our hardworking and research we do it and it's an achievement for us that motivates us to do better and more.

7.5. Future Enhancements/Recommendations

So generally, there are many enhancements we want to put in our product i.e.

- Proper Testified Detection of drugs and all hazardous materials.
- Proper linking of the database with a robot so that face detection and recognition will take place immediately with the best GPS tracking system.
- Create a similar silicon chip which will work same as honey bee's senses working.

Appendices

Appendix A: User Manual

In the appendix section, we describe the different phases of the user interface and also describe how a user can use our project. Secondly, we describe our promotional plan and promotional materials like broacher, banner, standee, and other marketing materials. We also describe other interfaces of our project.

Appendix A: User Manual

There is no user manual for our project at this time.

A.1. First Level Heading

N/A

A.1.1. Second level heading

N/A

A.1.1.1. Third level heading

N/A

Appendix B: Administrator Manual

There is no administration manual for our project.

B.1. First Level Heading

N/A

B.1.1. Second level heading

N/A

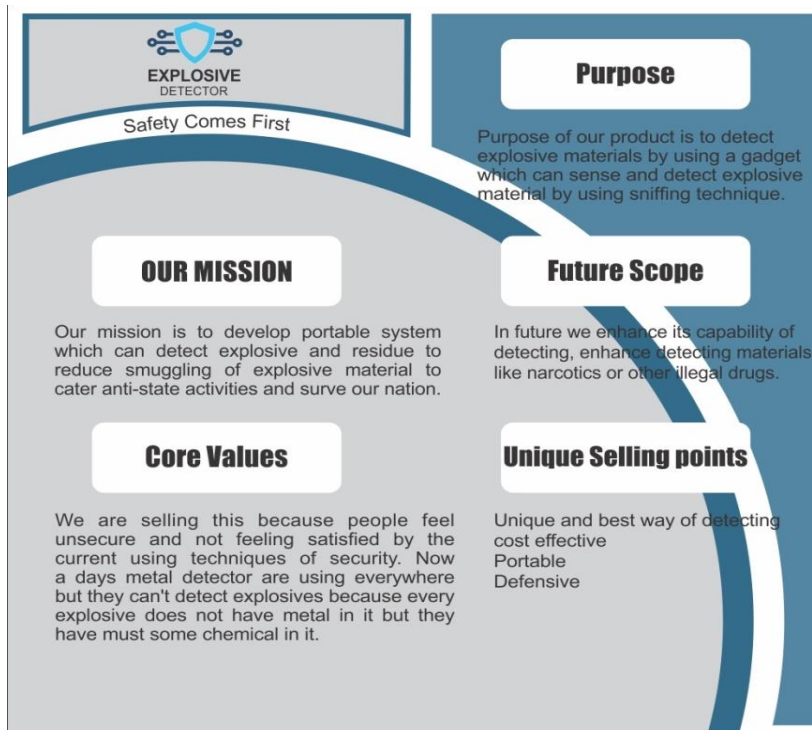
B.1.1.1. Third level heading

N/A

Appendix C: Information / Promotional Material

This chapter describes the promotional material that we used for the promotion of our Project “ED Tech”

C.1. Broacher





C.2. Flyer



C.3. Standee

Dr. Ch. Abdul Rehman
Chairman

Dr. Sumaira Rehman
Pro-rector

ED Tech

Humanity-a safe word in an unsafe world!

VISION | MISSION | CORE VALUES

EXPLOSIVE DETECTOR

VISION
To be the best company for providing services and solutions for explosive detection and anti-explosion system

MISSION
Our mission is to develop portable system which can detect explosive and residue to reduce smuggling of explosive material to cater anti-state activities and serve our nation.

CORE VALUES
Our product is portable system which can :
• Detect explosive residue through RF beam
• Detect explosive sensitivity

Zain-ul-Abideen
Instructor

Waqas Arshad
Coach

Taimoor Shakeel
Coach

Khurram Jilani
Mentor

M. Khuzema CH

Hammad Saeed

M. Zaid

Ali Akhtar

C.4. Banner

Humanity-a safe word in an unsafe world!

VISION | MISSION | CORE VALUES



Dr. Ch. Abdul Rehman
Chairman



ED Tech



**EXPLOSIVE
DETECTOR**



Dr. Sumaira Rehman
Pro-rector

· VISION
To be the best company for providing services and solutions for explosive detection and anti-explosion system

· MISSION
Our mission is to develop portable system which can detect explosive and residue to reduce smuggling of explosive material to cater anti-state activities and serve our nation.

· CORE VALUES
Our product is portable system which can :

- Detect explosive residue through RF beam
- Detect explosive sensitivity





Zain Ul Abidin
Instructor



Waqif Arshad
Coach



Khuram Jilani
Mentor



Taimoor Shaker
Coach



M.Khazema Ch
CEO



Hamad Saeed
CTO



M.Zaid Shahid
CFO




Ali Akhmer
COO

Reference and Bibliography

Reference and Bibliography

- [1] Arner, L.D., Johnson, G.R., Skovronek, H.S. 1986. Delineating Toxic Areas by Canine Olfaction. *Journal of Hazardous Materials* 13, 375-381.
- [2] Boomers, M.K., van Agtmael, M.A., Luik, H., van Veen, M.C., Vandenbroucke-Grauls, C.M.J.E., Smulders, Y.M. 2012. Using a dog's superior olfactory sensitivity to identify *Clostridium difficile* in stools and patients: proof of principle study. *British Medical Journal* 345, 7396.
- [3] Cablk, M.E., John, C., Sagebiel, J.C., Heaton, J.S., Valentin, C. 2008. Olfaction-based Detection Distance: A Quantitative Analysis of How Far Away Dogs Recognize Tortoise Odor and Follow It to Source. *Sensors* 8, 2208-2222.
- [4] DeGreeff, L.E., Weakley-Jones, B., Furton, K.G. 2012. Creation of training aids for human remains detection canines utilizing a non-contact, dynamic airflow volatile concentration technique. *Forensic Science International* 217. 32–38.
- [5] *Sumers, Brian*. Bomb-sniffing dogs at airports a reliable tool but not perfect. *Daily Breeze*. 2014. <http://www.morningjournal.com/article/MJ/20140217/NEWS/140219380.27-08-18>
- [6] Revkin, Andrew C. "Bees Learning Smell of Bombs With Backing From Pentagon." *The New York Times on the Web*. May 13, 2002. <http://www.cyberbee.net/news/beemine.html>
- [7] Vergano, Dan. "Honeybees join the bomb squad." *USA Today*. Nov. 27, 2006. http://www.usatoday.com/tech/science/2006-11-26-bees- bomb-sniffing_x.htm
- [8] "New homeland security buzz: Bomb-sniffing bees." *CNN.com*. Nov. 28, 2006. <http://www.cnn.com/2006/TECH/science/11/28/bombsniffing. bees.reut/index.html>
- [9] Appel, Adrienne. [Drug-Sniffing Wasps May Sting Crooks](#). *National Geographic*. 27 October 2005. Accessed 6 October 2011.
- [10] ["sniffer bees"](#). *CNet News*. Cnet. Retrieved 3 March 2012.
- [11] ["Inscentinel trains sniffer bees to detect bombs"](#). *ICIS website*. Retrieved 3 March 2012.
- [12] ["Pearson Matthews Research & Development"](#). *PMUK Website*. PMUK. Retrieved 3 March 2012.
- [13] ["Inscentinel Develops Bee-Based Sensor Device to Detect Explosives"](#). *AZO Sensors website*. AZO sensors. Retrieved 3 March 2012.
- [14] ["Sniffer bees set to snare suicide bombers"](#). *This is London Website*. *This Is London*. Retrieved 3 March 2012.
- [15] ["Inscentinel Honeybees Sniff For Explosives: Science Fiction in the News"](#). *Science Profiles Website*. Science Profiles. Retrieved 3 March 2012.
- [16] Khot, Anna. ["Humble Honey Bee Helping National Security"](#). *Naked Scientists Website*. Naked Scientists. Retrieved 3 March 2012
- [17] Will Knight (10 August 2006). ["Analysis: Explosive detection technologies"](#). *New Scientist.com news service*.
- [18] *Dossey, Aaron T. (December 2010)*. ["Insects and their chemical weaponry: New potential for drug discovery"](#). *Natural Prodts Reports*. *Royal Society of Chemistry (RSC Publishing)*. 27 (12): 1737–1757. [doi:10.1039/c005319h](#). [PMID 20957283](#).

- [19] J.R. Riley, U. Greggers, A.D. Smith, D.R. Reynolds, and R. Menzel (12 May 2005) "Letters to Nature: The flight paths of honeybees recruited by the waggle dance," *Nature*, vol. 435, pages 205-207.
- [20] (Newlon, Clark (1974). *Police Dogs in Action*. New York, Dodd, Mead, and company).
- [21] *'Practical Guide for Sporting & Working dogs. Royal Canin page 4*
- [22] "Hot picks: UK tech start-ups". BBC.co.uk. 2007-09-05.
- [23] Gubbi, Jayavardhana; Buyya, Rajkumar; Marusic, Slaven; Palaniswami, Marimuthu (1 September 2013). "[Internet of Things \(IoT\): A vision, architectural elements, and future directions](#)". *Future Generation Computer Systems*. Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services & Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond. **29** (7): 1645–1660. [arXiv:1207.0203](#) . [doi:10.1016/j.future.2013.01.010](#)
- [24] Hodgson, Martin. [Sniffer bees: New flying squad in war against terror Archived](#) July 9, 2012, at the [Wayback Machine](#).. [The Independent](#). 7 May 2006. Accessed 6 October 2011.
- [25] [Trained Wasps May Be Used To Detect Bombs, Bugs, Bodies And More](#). [Science Daily](#). 21 October 2005. Accessed 6 October 2011.

Index

Index

[A]

[B]

[C]