
**COMSATS University Islamabad**

**Abbottabad, Pakistan**

# Leading Emergency Service Rescue 1122 Web and Android application

***By***

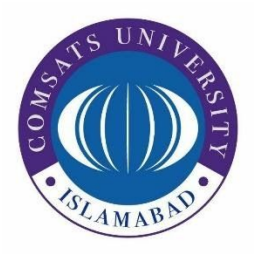
| | |
|---|---|
| **Saaf Farooq** | **CIIT/FA16-BSE-138/ATD** |
| **Afnan Arshad** | **CIIT/FA16-BSE-126/ATD** |
| **Hassan Ali** | **CIIT/FA16-BSE-140/ATD** |

***Supervisor***

**Dr Faisal Rehman**

***Bachelor of Science in Computer Science (2016-2020)***

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




**COMSATS University Islamabad,**

**Abbottabad Pakistan**

# Leading Emergency Service Rescue 1122 Web and Android application

**A project presented to**

**COMSATS Institute of Information Technology, Islamabad**

**In partial fulfillment**

**of the requirement for the degree of**

***Bachelor of Science in Computer Science (2016-2020)***

***By***

| | |
|---|---|
| **Saaf Farooq** | **CIIT/FA16-BSE-138/ATD** |
| **Afnan Arshad** | **CIIT/FA16-BSE-126/ATD** |
| **Hassan Ali** | **CIIT/FA16-BSE-140/ATD** |

## DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

| **Saaf Farooq** | **Afnan Arshad** | **Hassan Ali** |
| --- | --- | --- |
| --- | --- | --- |

## CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS) "**Leading Emergency Service Rescue 1122 Web and Android application**" was developed by **Saaf Farooq (CIIT/FA16-BSE-138/ATD), Afnan Arshad (CIIT/FA16-BSE-126/ATD) and Hassan Ali (CIIT/FA16-BSE-140/ATD)** under the supervision of "**Dr Faisal Rehman**" that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

**Dr Faisal Rehman**

---

**Supervisor**

---

**External Examiner**

---

**Head of Department**

**Dr Imran Ali Khan**

**(Department of Computer Science)**

# EXECUTIVE SUMMARY

In public places, there is often a need for monitoring people and different activities going on, which can be referred later for many reasons including security. Appointing humans for this task involves many problems such as increased employee hiring, accuracy problem, trust, no proof for later use, and also the fact that a human can remember things till a certain time limit. Talking about the current security system, they use dumb still cameras with a continuous recording facility ir-respective of the fact that any event may happen or not. Moreover they are usually pointing at a specific user defined locations so more than one cameras are required to cover the entire region.

To prevent all these problems from prevailing, the CSCS is developed. It is a surveillance system, which provides solution to many of these problems. It is a stand-alone application which doesn't require any computer to operate. It monitors different situations using a camera which is able to rotate intelligently based on sensor messages and captures the scene in the form of video or photos later reference as well.

**C**ustomizable **S**urveillance **C**ontrol **S**ystem **(CSCS)** is a surveillance system that can be assigned a sensor type as in our case a heat sensor is used, it works accordingly, rotates the camera upon event detection and perform user defined actions like capturing video and stores them, for the future use.

It is an embedded system consisting of Linux fox kit with embedded a running server application also a camera, USB storage device and a sensor node base station is attached with fox kit. LAN communication is used by user to download the videos and to operate the system manually.

# DEDICATION

It is my genuine gratefulness and warmest regard that I dedicate this work to my parents without whom it was difficult for me to complete my thesis work. Thank you for all the guidance and support that you people have given me and helping me to succeed.

Afnan Arshad

---

Dedicated to my beloved parents and teachers for their love, endless support, encouragement & sacrifices

Saad Farooq

---

This piece of work is dedicated to my parents, it's impossible to thank you adequately for everything you've done, from loving me unconditionally to raising me in an environment, where you instilled traditional values and taught your children to celebrate and embrace life.

Hassan Ali

---

## ACKNOWLEDGEMENT

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Majid Iqbal Khan” and our Co-Supervisor “Mr. Mukhtar Azeem”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

**Saaf Farooq** **Afnan Arshad** **Hassan Ali**

--- --- ---

# ABBREVIATIONS

| | |
|---|---|
| **SRS** | Software Require Specification |
| **Hardware platform** | Cell phones/Laptop/Desktop |
| **SSD** | System Sequence Diagram |
| **API** | Application Programming Interface |
| **OS** | Operating System **(**Android/Windows/Os') |
| **Browser** | Google chrome, Mozilla Firefox etc. |
| **OOP** | Object Oriented Programming |
| **SDLC** | Systems Development Life Cycle |

# TABLE OF CONTENTS


**1 Introduction** 11

1.1 Brief 11
1.2 Relevance to Course Modules 11
1.3 Project Background 11
1.4 Literature Review 11
1.5 Analysis from Literature Review (in the context of your project) 11
1.6 Methodology and Software Lifecycle for This Project 11
1.6.1 Rationale behind Selected Methodology 11

**2 Problem Definition** 12

2.1 Problem Statement 12
2.2 Deliverables and Development Requirements 12
2.3 Current System (if applicable to your project) 12

**3 Requirement Analysis** 13

3.1 Use Cases Diagram(s) 13
3.2 Detailed Use Case 13
3.3 Functional Requirements 13
3.4 Non-Functional Requirements 13

**4 Design and Architecture** 14

4.1 System Architecture 14
4.2 Data Representation [Diagram + Description] 14
4.3 Process Flow/Representation 14
4.4 Design Models [along with descriptions] 14

**5 Implementation** 15

5.1 Algorithm 15
5.2 External APIs 15
5.3 User Interface 15

**6 Testing and Evaluation** 16

6.1 Manual Testing 16
6.1.1 System testing 16

6.1.2 Unit Testing 16
6.1.3 Functional Testing 17
6.1.4 Integration Testing 17
6.2 Automated Testing: 18
Tools used: 18
**7 Conclusion and Future Work** 19
7.1 Conclusion 19
7.2 Future Work 19
**8 References** 20

## LIST OF FIGURES

Fig 1.1 Block Diagram 8
Fig 2.1 Use Case Diagram 9

## LIST OF TABLES

Table 1.1 Table1 8
Table 2.1 Table 2 9

# Chapter No. 1: Introduction

This project consists of creating a Leading Emergency Service Recue1122 web and android application – an updated version of Rescue1122.
. In an emergency situation, People provide precise information about location. So through this App and Website, they will provide their location to one of the given departments (**Ambulance service, Fire service, natural disaster service**).

## 1.1 Brief Introduction

### Brief overview of the document chapters

**Chapter no. 1:** This chapter provides an introduction and specifies the relevance of the project
**Chapter no. 2:** This chapter describes the real problem, current system and the possible solution for the problem.
**Chapter no. 3:** This chapter provides use-cases of the system, detailed use cases, functional requirements and non-functional requirements.
**Chapter no. 4:** This chapter provides the detailed system design i.e package diagram, class diagram, domain model etc.
**Chapter no. 5:** This chapter provides the details about the API's used and User interface of the system.
**Chapter no. 6:** This chapter has details about all types of testing which has been performed.
**Chapter no. 7:** This chapter has the conclusion and future work which can be done on this project.with the courses studied throughout the degree.

## 1.2 Relevance to Course Modules

The ongoing project "**Leading Emergency Service Rescue 1122 Web and Android application** " is developed using android studio and Netbeans. In whole tenure of **BSE** following courses we studied are relevant to our project:

- Software Requirement Engineering
- Software engineering 1
- Software engineering 2
- Software quality engineering
- Software design and architecture
- Database system

- Software construction
- Advanced OOP
- Web Engineering
- Software project management

## 1.3 Project Background

Leading Emergency Service Recue1122 web and android application – an updated version of Rescue1122.
. In an emergency situation, People provide precise information about location. So through this App and Website, they will provide their location to one of the given departments (**Ambulance service, Fire service, natural disaster service**).

- **Android app** for users will be developed where they can give their location to Service groups in emergency situations. They have to select the department they are seeking to avail service and provide location. The nearby service group gets the notification and response as fast as possible. Workers profile is developed in the app.

- **Web application** will be developed for admin. Users can also visit the site. Workers from three different departments have their profiles in the android app . They can get all the working schedules there. Admin can communicate with all workers. All departments are connected with each other through chat rooms.

## 1.4 Literature Review

Mobile apps and websites are under huge revolution. With the start of latest technologies like the growing performance enhancement of smartphones, mobile app development companies have to remain on their toes to develop different apps for which people want,need or interested. It has to keep track of the latest trends emerging in mobile apps.
Every second and minute, a lot of incidents happen in the whole world.Many people don't know what they do in emergency situations. Also in emergency situations, People don't know how they provide precise information about location or incident. So to solve this problem, most of the countries made different emergency apps for their citizens.

## 1.5 Analysis from Literature Review (in the context of your project)

Our app and website is unique in its own way. In an emergency situation, People provide precise information about location. So through this App and Website, they will provide their location to one of given departments (**Ambulance service, Fire service, natural disaster**

**service**).The app will have the capability to release on play/app store once all the modules are completed and can make it for people's used in all over Pakistan.

## 1.6 Methodology and Software Lifecycle for This Project

For any project to be completed, it has to go through stages called Development Life Cycles. SDLC is the process of understanding how an information system can support business needs, designing the system, building it and delivering it to the users. SDLC comprises four phases: Planning, Analysis, Design and Implementation.
In order for this project to be developed, the methodology that we are using is “Object Oriented approach “and software lifecycle is “Incremental Development “.

## 1.7 Rationale behind Selected Methodology

Reason behind selected methodology and SDLC is: -
Object-oriented programming provides several useful software development innovations.

- First, it provides a useful operational definition of module as a class.
- Second, it establishes an organizing principle for task decomposition, and minimization of communication and dependencies among modules.
- Third, it enforces a useful formal separation between architecture, implementation, and realization. There is separation between public and private members of a class,and the separation of class declaration from class realization.
- Object-oriented design simplifies the structure of large software systems

**Incremental Model**
As the requirements of the complete system are clearly defined and understood therefore this model is advantageous to use.
Generates working software quickly and early during the software life cycle.

# CHAPTER NO. 2: Problem Definition

This application will solve some important issues regarding providing precise information and location to one of the given departments (**Ambulance service, Fire service, natural disaster service**) in any emergency situations. Users need to send precise information about location to any emergency departments.

## 2.1 Problem Statement:

Mobile apps and websites are under huge revolution. With the start of latest technologies like growing performance enhancement of smartphones, mobile app development companies have to remain on its toes to develop different apps for which people wants,needs or interested. It has to keep track of the latest trends emerging in mobile apps. The current apps on online stores are either too fancy or simple. Some people like to use a app's which is not too colourful or fancy but made simple and easy to use . Any app's that is developed must have capability to keep user engaged , but most of the app's lack this feature. And users end-up uninstalling the app's with leaving a negative review. It is important to bring something new with each new upgradation which does not end up being any issues.

Every second or minute, a lot of incidents happen in the whole world.Many people don't know what they do in emergency situations.Also in an emergency situation, People don't know how they provide precise information about location or incident. So to solve this problem, most of the countries made different emergency app for their citizens. Our ongoing project "**Leading Emergency Service Rescue 1122 Web and Android application** " will solve some important issues regarding providing precise information and location to one of the given departments (**Ambulance service, Fire service, natural disaster service**) in any emergency situations. Users need to send precise information about location to any emergency departments.

## 2.2 Deliverables and Development Requirements

At the end, we will get a Web application as well as an Android application. Users can easily use this application. This application will solve some important issues regarding providing precise information and location to one of the given departments (**Ambulance service, Fire service, natural disaster service**) in any emergency situations.

- **Android app** for users will be developed where they can give their location to Service groups in emergency situations. They have to select the department they are seeking to avail service and provide location. The nearby service group gets the notification and response as fast as possible. Workers profile is developed in the app.
- **Web application** will be developed for admin. Users can also visit the site. Workers from three different departments have their profiles in the android app . They can get all the working schedules there. Admin can communicate with all workers. All department are connected with each other through chat room

### 2.2.1 Admin:

- Can login.
- Can add, edit, delete, view workers.
- Can receive records from workers and save into databases.
- Can give complaints from users.
- Can get alerts to users.
- Can see the location of any worker.
- Admin communicate with workers through chat rooms.

### 2.2.2 Worker:

- Can Signup.
- Can Login
- Can accept/reject user emergency requests.
- Can visit their portal profile on their website.
- Can provide information about current emergency situations to admin.
- Worker's have a unique profile.
- Provides services in emergency situations only.

### 2.2.3 User:

- Can Signup.
- Can login.
- Can Visit Website
- Can complain.
- Can press the Emergency Button.
- Can view tips of emergency situations.
- Can get alerts from Admin.
- Can provide location in emergency situation

# CHAPTER NO. 3 Requirement Analysis

Functional and nonfunctional requirement of system. This is the way to describe customer desires for an application that will be manufactured or modified. The prerequisite research includes each of the tasks that led to the realization of the needs of different stakeholders.

## 3.1 Use Cases Diagram

It is a form of software requirement. Use cases refer to normal behavior, not the exact strategy to accomplish them. Use of the cases referred to once can mean both literary and visual photography

The Figure 3.1 shows how the primary and secondary actor's(users) interact with the system.

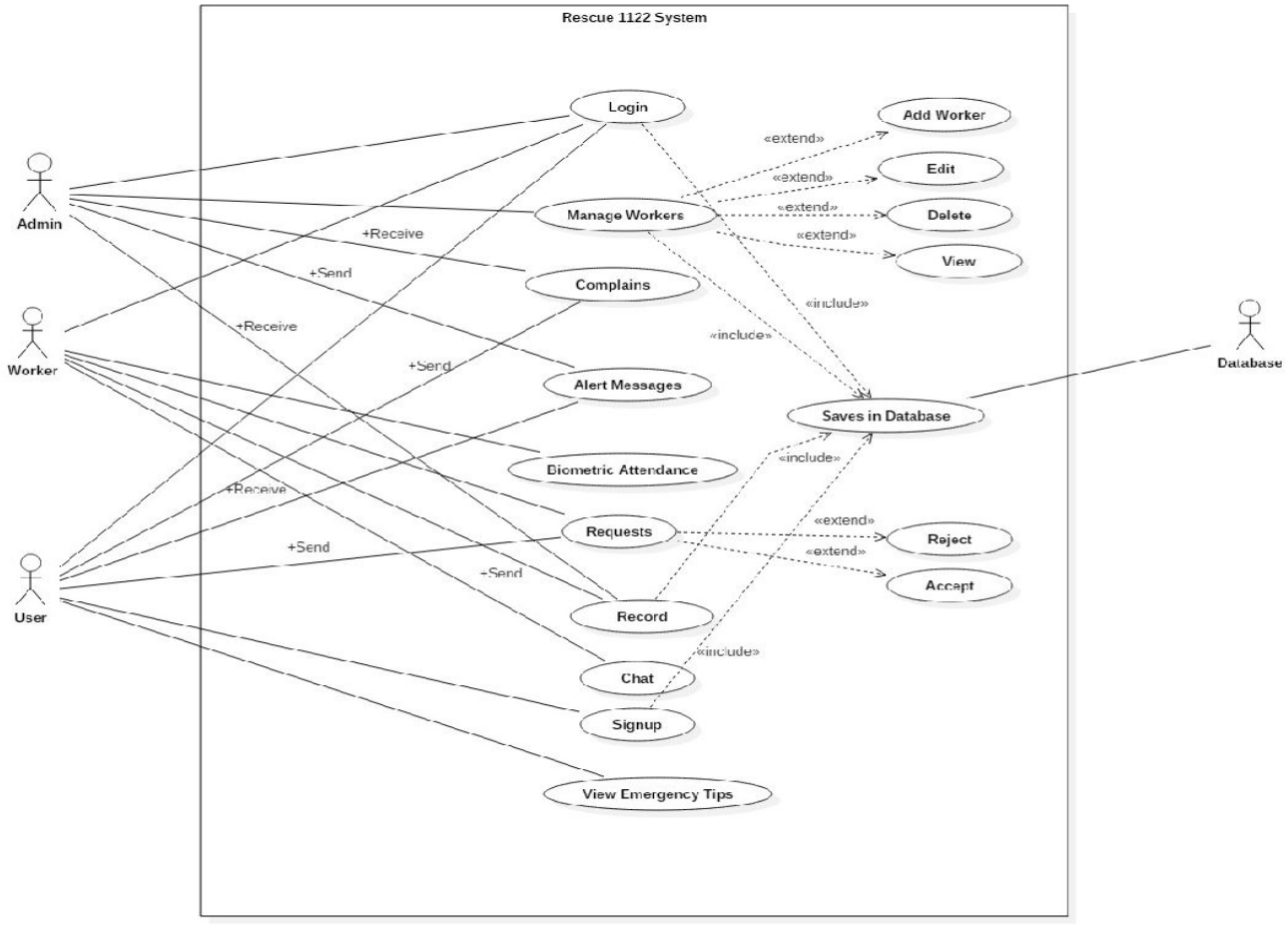



Figure 3.1 Usecase Diagram

## 3.2 Detailed Use Case

This session tells us about the details of the use case in our system. Also tells us about the details of the use case of the system.

**Use Cases:**

1. Login
2. SignUp
3. Add Worker
4. Edit Worker
5. Delete Worker (Record)
6. View Worker (Record)
7. Alerts Message
8. Biometric attendance
9. Requests(Accept, Reject)
10. Chat
11. Signup
12. View Emergency Tips

### 3.2.1 Login

Table 3.2.1 shows the use case of "Login"

**Table 3.2.1**

| | |
|---|---|
| **Use Case ID:** | UC-1 |
| **Use Case name:** | Login |
| **Actors:** | **Primary Actor:** Admin, Worker, User<br>**Secondary Actor:** Database |
| **Description:** | Primary actors' login in order to use the website, for which they fill signup form providing username and password then clicks login's button. Then their data will be authenticated from database after authentication and they are redirected to website homepage |
| **Trigger:** | User wants to login the application. Admin, Worker and user indicates he wants to login to their id. |
| **Preconditions:** | PRE-1. Working internet connection is required.<br>PRE-2. Admin, Worker and User should have an account.<br>PRE-3. Admin, Worker and User must have valid username, password.<br>PRE-4. Admin, Worker and User must have his own password, username. |

| | |
|---|---|
| **Post conditions:** | POST-1. Admin, Worker and User fills the login form and clicks the login button.<br>POST-2. Admin, Worker and User data authenticated.<br>POST-3. Application will provide access to the Admin, Worker andUser.<br>POST-4 Admin, Worker and User is redirected to homepage |
| **Normal Flow:** | • User clicks the login button.<br>• User fills the login form and clicks the login button.<br>• User's data will be authenticated from database User types the code.<br>• After successful authentication "login success" message displayed.<br>• User is redirected to the homepage. |
| **Exceptions:** | 1- User fills invalid information.<br>- warning message will be displayed to the user highlighting the mistake.<br>2- user inputs wrong password<br>- warning alert will be shown<br>3- if user forgets password or username<br>3a- user clicks forgot password button<br>3b- password reset link will be send to user email0<br>3c- user access the link<br>3d- user sets up new password |
| | |
| | |

### 3.2.2 Signup

Table 3.2.2 shows the use case of "Signup"

**Table 3.2.2**

| | |
|---|---|
| **Use Case ID:** | UC-2 |
| **Use Case Name:** | Signup |
| **Actors:** | **Primary Actor:** User<br>**Secondary Actor:** Database |
| **Description:** | Primary actors' signup in order to use the application, for which they fill signup form providing authentic credentials and click submit button. Then their data will be saved in database after authentication and they are redirected to login page |
| **Trigger:** | User wants to use the application. |

| | |
|---|---|
| **Preconditions:** | PRE-1. Working internet connection is required.<br>PRE-2 User must have knowledge about browsing<br>PRE-3 Users must have a working mail address.<br>PRE-4 Users should have a valid CNIC number. |
| **Post conditions:** | POST-1. User fills the form and clicks the submit button.<br>POST-2. User's data is stored in a database.<br>POST-3 User is redirected to the login page. |
| **Normal Flow:** | • User clicks the signup button.<br>• User fills the form and clicks the register button.<br>• Users type the CNIC.<br>• User typed CNIC is authenticated<br>• "Signup success" message displayed.<br>• User's data stored in the database.<br>• User is redirected to Login page |
| **Exceptions:** | 1- User fills invalid information.<br>- Warning message will be displayed to the user highlighting the mistake.<br>2- if email is invalid<br>2a- warning alert will be shown<br>2b- user will renter email |

### 3.2.3 Add Worker

Table 3.2.3 shows the use case of "Add Worker"

**Table 3.2.3**

| | |
|---|---|
| **Use Case ID:** | UC-3 |
| **Use Case Name:** | Add worker |
| **Actors:** | **Primary Actor:** Admin<br>**Secondary Actor:** Database |
| **Description:** | Admin add a new worker record. |
| **Trigger:** | Admin wants to add a new worker. |
| **Preconditions:** | PRE-1. Admin should have login.<br>PRE-2. Working internet connection is required.<br>PRE-3 Admin must have knowledge about browsing |
| **Post conditions:** | POST-1. Admin Add the worker record.<br>POST-2. The record save in database. |

| | |
|---|---|
| | POST-3 Record Show in the list. |
| **Normal Flow:** | • Admin clicks login button.<br>• Admin Fill the login form And click login button.<br>• Admin data will be authenticated from the database.<br>• After successful authentication “login success” message displayed.<br>• Admin is redirected to homepage.<br>• Admin browse to Employee Record.<br>• Admin Clicks the create button.<br>• Admin Fill The form Click The create button.<br>• Data is saved in the Database. |
| **Exceptions:** | 1- Admin fills invalid information.<br>- Warning message will be displayed to the user highlighting the mistake.<br>2- Admin inputs wrong password<br>- warning alert will be shown<br>3- Worker Record already existed.<br>3a- Message display “Already Existed”.<br>3b- Redirect to create View. |
| | |

### 3.2.4 Delete Worker Record

Table 3.2.4 shows the use case of “Delete Worker Record”

**Table 3.2.4**

| **Use Case ID:** | UC-4 |
|---|---|
| **Use Case Name:** | Delete worker record |
| **Actors:** | **Primary Actor:** Admin<br>**Secondary Actor:** Database |
| **Description:** | Admin deletes existing worker records. |
| **Trigger:** | Admin wants to delete a worker record from the database. |
| **Preconditions:** | PRE-1. Admin should have login.<br>PRE-2. Working internet connection is required.<br>PRE-3 Admin must have knowledge about browsing |
| **Post conditions:** | POST-1. Add put worker name or ID in search option.<br>POST-2. Admin deletes the worker record. |

| | |
|---|---|
| | POST-2. The existing record deletes from the database.<br>POST-3 Then deleting the Worker Record not Show in the list. |
| **Normal Flow:** | • Admin clicks login button.<br>• Admin Fill the login form And click login button.<br>• Admin data will be authenticated from the database.<br>• After successful authentication "login success" message displayed.<br>• Admin is redirected to homepage.<br>• Admin browse to Employee Record.<br>• Admin Clicks the delete worker record button.<br>• Add put worker name or ID in search option.<br>• Admin deletes the worker record. button.<br>• The existing record deletes from the database.<br>• Then deleting the Worker Record not Show in the list. |
| **Exceptions:** | 1- Admin fills invalid information.<br>- Warning message will be displayed to the user highlighting the mistake.<br>2- Admin inputs wrong password<br>- warning alert will be shown<br>3- Worker Record does not already exist.<br>3a. Message display "Not Existed".<br>3b. Redirect to delete record page. |

### 3.2.5 Edit Worker

Table 3.2.5 shows the use case of "Edit Worker"

**Table 3.2.5**

| | |
|---|---|
| **Use Case ID:** | UC-5 |
| **Use Case Name:** | Edit Worker |
| **Actors:** | **Primary Actor:** Admin<br>**Secondary Actor:** Database |
| **Description:** | Admin Edit already Existed worker record. |
| **Trigger:** | Admin want to Edit a worker record. |
| **Preconditions:** | PRE-1. Admin should have login.<br>PRE-2. Working internet connection is required.<br>PRE-3 Admin must have knowledge about browsing |
| **Post conditions:** | POST-1. Admin Edit the worker record.<br>POST-2. The record save in database.<br>POST-3 Record Show in the list. |

| | |
|---|---|
| **Normal Flow:** | • Admin clicks login button.<br>• Admin Fill the login form And click login button.<br>• Admin data will be authenticated from the database.<br>• After successful authentication “login success” message displayed.<br>• Admin is redirected to homepage.<br>• Admin browse to Employee Record.<br>• Admin Clicks the Edit button.<br>• Admin Fill The form Click The Edit button.<br>• Data is saved in the Database. |
| **Exceptions:** | 1- Admin fills invalid information.<br>- Warning message will be displayed to the user highlighting the mistake.<br>2- Admin inputs wrong password<br>- warning alert will be shown<br>3- If Admin Submit Empty form .<br>3a. Message display “Form is empty”.<br>3b. Refresh the Edit View. |

### 3.2.6 View

Table 3.2.6 shows the use case of “View Worker Record”

**Table 3.2.6**

| | |
|---|---|
| **Use Case ID:** | UC-6 |
| **Use Case Name:** | View |
| **Actors:** | **Primary Actor:** Admin<br>**Secondary Actor:** Database |
| **Description:** | Admin View the List of worker records. |
| **Trigger:** | Admin wants View The List of worker records. |
| **Preconditions:** | PRE-1. Admin should have login.<br>PRE-2. Working internet connection is required.<br>PRE-3 Admin must have knowledge about browsing.<br>PRE-4 Worker List should not be empty. |
| **Post conditions:** | POST-1. Admin click the view button.<br>POST-2. The list of Worker records Appears to Admin Along With Three buttons edit, delete, details. |
| **Normal Flow:** | • Admin clicks login button.<br>• Admin Fill the login form And click login button.<br>• Admin data will be authenticated from the database. |

| | |
|---|---|
| | • After successful authentication “login success” message displayed.<br>• Admin is redirected to homepage.<br>• Admin browse to Employee Record.<br>• Admin Clicks the View button.<br>• List Of worker record appears to admin. |
| **Exceptions:** | 1- Admin fills invalid information.<br>- Warning message will be displayed to user highlighting the mistake.<br>2- Admin inputs wrong password<br>- warning alert will be shown<br>3- If list is Empty.<br>- .A blank List appears.<br>4- Worker Record does not exist.<br>4a. Message display “Not Existed”.<br>4b. Redirect to view page. |

### 3.2.7 Alert Messages

Table 3.2.7shows the use case of “Alerts Messages”

**Table 3.2.7**

| | |
|---|---|
| **Use Case ID:** | UC-7 |
| **Use Case Name:** | Alert Messages |
| **Actors:** | **Primary Actor:** Admin,User<br>**Secondary Actor:** Database |
| **Description:** | Users send emergency alerts or inform about incident location and Admin view emergency alert and send it to nearby workers. |
| **Trigger:** | User wants to acknowledge the Admin or worker about any incident. |
| **Preconditions:** | PRE-1. Users should have login.<br>PRE-2. Working internet connection is required.<br>PRE-3 Users must have knowledge about browsing.<br>PRE-4 User selects the emergency department and provides location or emergency alert. |
| **Post conditions:** | POST 1. Admin receive the complaint.<br>POST-2. The emergency alert Appears to Admin.<br>POST-3. Admin click the view alert.<br>POST-4. Admin send emergency alerts to those areas available near the worker. |
| **Normal Flow:** | • User clicks the login button. |

| | |
|---|---|
| | • User Fill the login form And click login button.<br>• User data will be authenticated from the database.<br>• After successful authentication “login success” message displayed.<br>• User is redirected to the homepage.<br>• User clicks the emergency button.<br>• Users select one emergency department.<br>• Admin receive the complaint.<br>• Admin click the view alert.<br>• The emergency alert Appears to Admin.<br>• Admin send emergency alerts to those areas available near workers.<br>• Worker response as fast as possible and available on emergency location. |
| **Exceptions:** | 1- Admin fills invalid information.<br>- Warning message will be displayed to the user highlighting the mistake.<br>2- User inputs wrong password<br>- warning alert will be shown<br>3- If the emergency alert is Empty.<br>- Message appears ”Text box is empty”.<br>4- Admin could not receive the emergency alert<br>- A text message appears to the user “alert failed”.<br>5- Admin inputs wrong password<br>- warning alert will be shown<br>6- If the list is Empty.<br>- .A blank List appears. |

### 3.2.8 Requests(Accept, Reject)

Table 3.2.8 shows the use case of “Request Accept or Reject”

**Table 3.2.8**

| | |
|---|---|
| **Use Case ID:** | UC-8 |
| **Use Case Name:** | Requests(Accept,Reject) |
| **Actors:** | **Primary Actor:** Worker, User<br>**Secondary Actor:** Database |
| **Description:** | Users send emergency requests or inform about incident location . Admin sends those requests to the worker and worker rejects or accepts the request and performs the operation as fast as possible. |
| **Trigger:** | User wants to acknowledge the Admin or worker about any incident. |

| | |
|---|---|
| **Preconditions:** | PPRE-1. Users should have login.<br>PRE-2. Working internet connection is required.<br>PRE-3 User must have knowledge about browsing.<br>PRE-4 User selects the emergency department and provides an emergency request. |
| **Post conditions:** | POST 1. Admin receive the request.<br>POST-2. The emergency request Appears to Admin.<br>POST-3. Admin click the view alert.<br>POST-3. Admin send emergency alerts to those areas available near workers.<br>POST-4. Workers accept or reject emergency requests. |
| **Normal Flow:** | • User clicks the login button.<br>• User Fill the login form And click login button.<br>• User data will be authenticated from the database.<br>• After successful authentication “login success” message displayed.<br>• User is redirected to the homepage.<br>• User clicks the emergency button.<br>• Users select one emergency department.<br>• Admin receive the complaint.<br>• Admin click the view alert.<br>• The emergency alert Appears to Admin.<br>• Admin send emergency alerts to those areas available near the worker.<br>• Workers reject the emergency request because of some other issues.<br>• If a Worker accepts a request then Worker response as fast as possible and available on emergency location. |
| **Exceptions:** | 1- Admin fills invalid information.<br>- Warning message will be displayed to the user highlighting the mistake.<br>2- User inputs wrong password<br>- warning alert will be shown<br>3- If the emergency alert is Empty.<br>- Message appears ”Text box is empty”.<br>4- Admin could not receive the emergency alert<br>- A text message appears to the user “alert failed”.<br>5- Admin inputs wrong password<br>- warning alert will be shown<br>6- If the list is Empty.<br>- A blank List appears. |

### 3.2.9 Chat

Table 3.2.9 shows the use case of "Chat"

**Table 3.2.9**

| | |
|---|---|
| **Use Case ID:** | UC-9 |
| **Use Case Name:** | Chat |
| **Actors:** | **Primary Actor:** Admin, Worker<br>**Secondary Actor:** Database. |
| **Description:** | Admin communicates workers through chat rooms. |
| **Trigger:** | Admin wants to Acknowledge worker's about any issue or other activities. |
| **Preconditions:** | PRE-1. Admin should have login.<br>PRE-2. Admin internet connection is required.<br>PRE-3 Admin must have knowledge about browsing.<br>PRE-4 Chatroom is in working condition. |
| **Post conditions:** | POST-1 Admin communicates with workers through chat rooms.<br>POST-2. Admin inform Workers about any emergency situation |
| **Normal Flow:** | • Admin clicks login button.<br>• Admin Fill the login form And click login button.<br>• Admin data will be authenticated from the database.<br>• After successful authentication "login success" message displayed.<br>• Admin is redirected to homepage.<br>• Chatroom is in working condition.<br>• Admin communicate with workers through chat room.. |
| **Exceptions:** | 1- Admin fills invalid information.<br>- Warning message will be displayed to the user highlighting the mistake.<br>2- Admin inputs wrong password<br>- warning alert will be shown<br>3- Chatroom is not in working condition. |

### 3.2.10 Complain

Table 3.2.10 shows the use case of "Complain"

**Table 3.2.10**

| | |
|---|---|
| **Use Case ID:** | UC-10 |
| **Use Case Name:** | Complain |

| | |
|---|---|
| **Actors:** | **Primary Actor:** Admin, User.<br>**Secondary Actor:** Database. |
| **Description:** | Users send complaint messages to the admin and complain further saves in the database. |
| **Trigger:** | User wants to Acknowledge the admin about any issue. |
| **Preconditions:** | PRE-1. Users should have login.<br>PRE-2. Working internet connection is required.<br>PRE-3 Users must have knowledge about browsing.<br>PRE-4 User Enter the Complaint View Tab. |
| **Post conditions:** | POST-1 Complaint sent to admin via message.<br>POST-2. Admin reads the notifications.<br>POST-3. Complaints will be saved in the database.<br>POST-4 Admin replies to user complaints. |
| **Normal Flow:** | • User clicks the login button.<br>• User Fill the login form And click login button.<br>• User data will be authenticated from the database.<br>• After successful authentication “login success” message displayed.<br>• User is redirected to the homepage.<br>• User browse to Complain Tab.<br>• User Submits the Complain.<br>• Admin receive the complaint.<br>• Admin reply the users complain.<br>• Complain save in Database |
| **Exceptions:** | 1- User fills invalid information.<br>- Warning message will be displayed to the user highlighting the mistake.<br>2- User inputs wrong password<br>- warning alert will be shown<br>3- If the Complaint box is Empty.<br>- Message appears ”Text box is empty”.<br>4- Admin could not receive the complaint<br>- A text message appears to the user “Complain failed”. |
| | |

### 3.2.11 View Emergency Tips

Table 3.2.11 shows the use case of “View Emergency Tips”

**Table 3.2.11**

| | |
|---|---|
| **Use Case ID:** | UC-11 |

| | |
|---|---|
| **Use Case Name:** | View Emergency Tips |
| **Actors:** | **Primary Actor:** User<br>**Secondary Actor:** Database. |
| **Description:** | When User sends an emergency request or alert then the system automatically generates emergency tips for Users. |
| **Trigger:** | Users want to know about Emergency Tips. |
| **Preconditions:** | PRE-1. Users should have login.<br>PRE-2. Users internet connection is required.<br>PRE-3. Users must have knowledge about browsing.<br>PRE-4. Users send Emergency requests or alert. |
| **Post conditions:** | POST-1. System get or receive Emergency request<br>POST-2. System generates Emergency Tips<br>POST-3. Users receive Emergency Tips from the system about the incident. |
| **Normal Flow:** | • User Fill the login form And click login button.<br>• User data will be authenticated from the database.<br>• After successful authentication “login success” message displayed.<br>• User is redirected to the homepage.<br>• Users send Emergency requests or alerts.<br>• System get or receive Emergency request<br>• System generates Emergency Tips<br>• User receives Emergency Tips from the system about the incident. |
| **Exceptions:** | 1- User fills invalid information.<br>- Warning message will be displayed to the user highlighting the mistake.<br>2- User inputs wrong password<br>- Warning alert will be shown<br>3- Users do not receive emergency tips.<br>- System Error. |

## 3.3 System Sequence Diagram

It is a summary of use cases. System sequence diagrams have external actors, messages, return values, indication of loop.

### 3.3.1 Admin Login

Figure 3.1 shows that System will allow admin to login. Admin login the system by entering user name and password. System validates their username and password and displays the Home page.

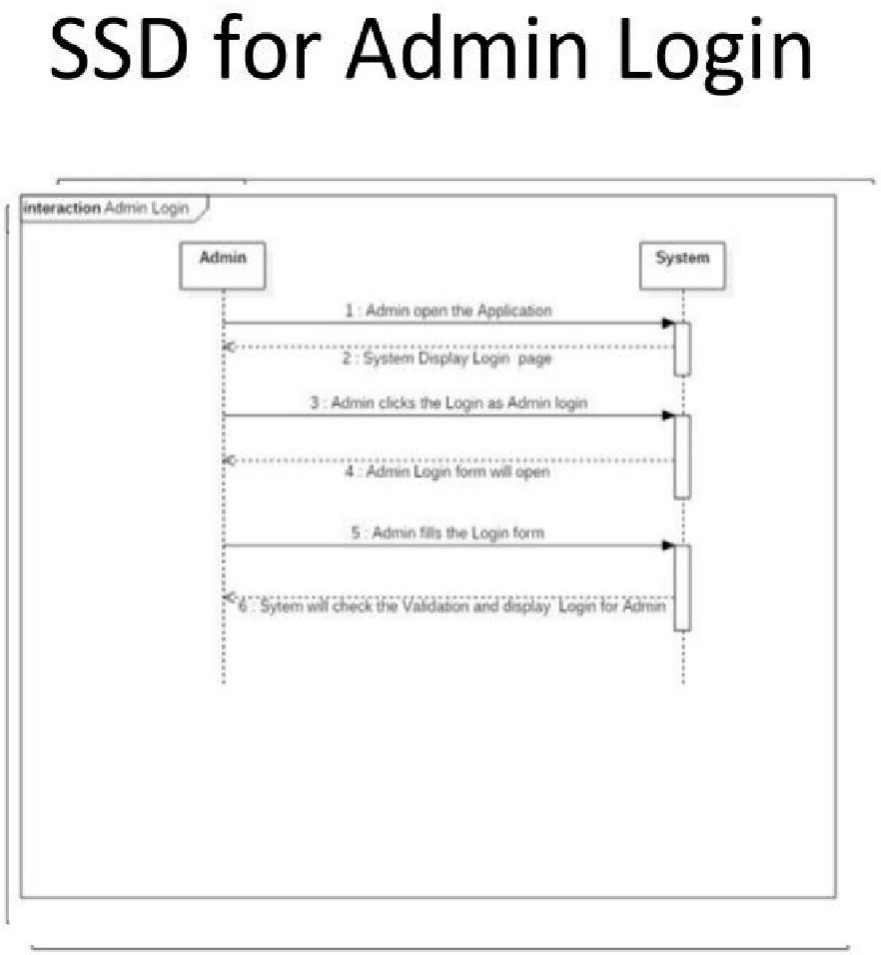



Figure 3.1 System Sequence Diagram (Admin Login)

### 3.3.2 Worker Login

Figure 3.2 shows that System will allow workers to login. Worker's login to the system by entering username and password. System validates their username and password and displays the Home page.

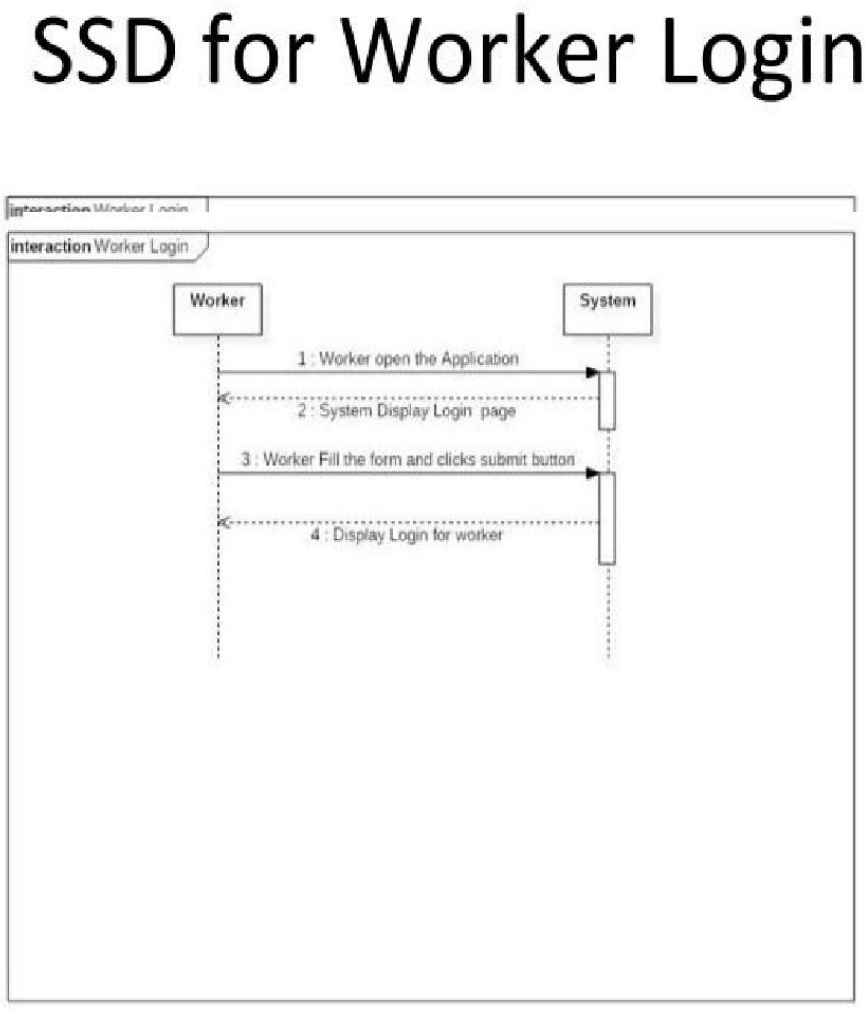



Figure 3.2 System Sequence Diagram (Worker's Login)

### 3.3.3 Add New Worker Record

Figure 3.3 shows that the System will allow the admin to add a new worker record. Admin wants to add a new worker record in the database. Admin open add new worker tab by clicking on add new worker option, then enter biometric data with new name and phone number and then forward or save record in database. System generates "new worker add successfully" message to Admin. If the same record is already present in the database the system generates a "worker already exists" message to Admin.

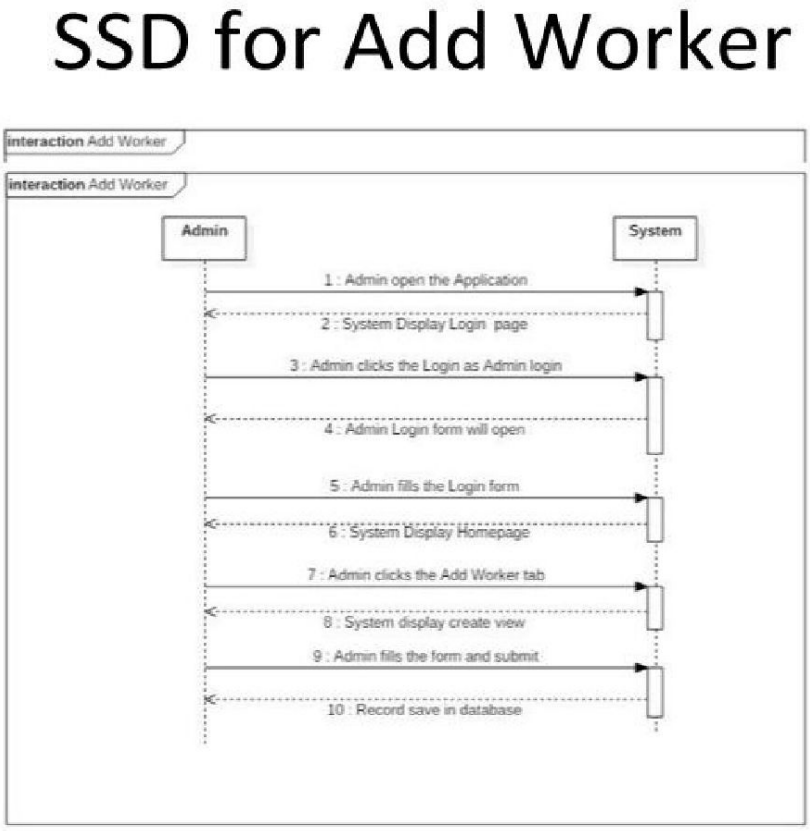



Figure 3.3 System Sequence Diagram (Add New Worker Record)

### 3.3.4 Delete Worker Record

Figure 3.4 shows that System will allow the admin to delete worker records. Admin wants to delete the worker record from the database. Admin open delete worker tab by clicking on delete worker option, then enter name or email and then forward to database. System finds those records and generates "worker record deleted successfully" message to Admin. If the same record does not exist or present in the database the system generates a "worker doesn't exist" message to Admin.

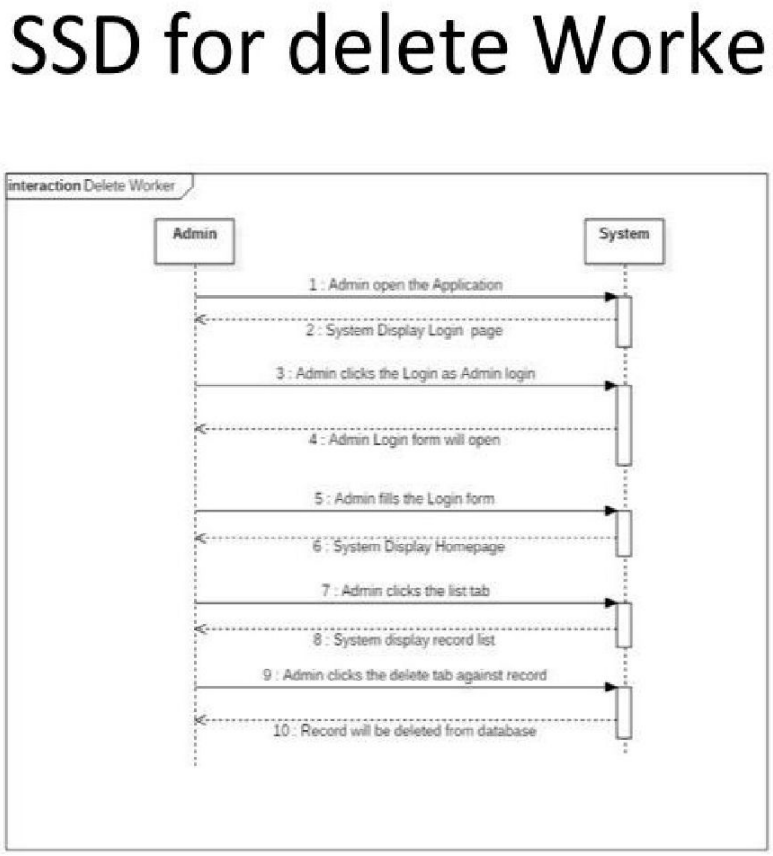



Figure 3.4 System Sequence Diagram (Delete Worker Record)

## 3.3 System Sequence Diagram

It is a detailed summary of use cases. Sequence diagrams have external actors, messages, return values, indication of loop.

### 3.4.1 Login

Figure 3.5 shows that Admin,Workers and Users login the system by entering username and password. System validates their username and password and displays the Home page.

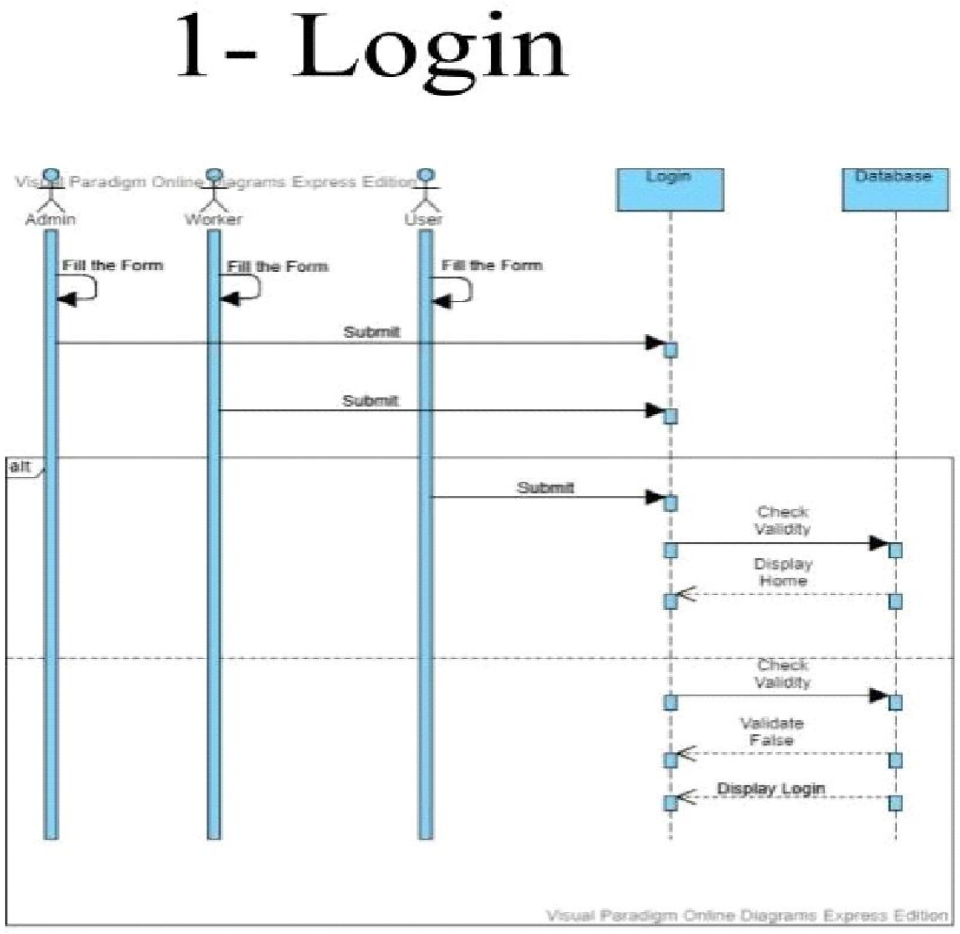



Figure 3.5 Sequence Diagram (Login)

### 3.4.2 Signup

Figure 3.6 shows that User opens the Signup page and enters email and password. System validates their email and password and then will be User sign in.

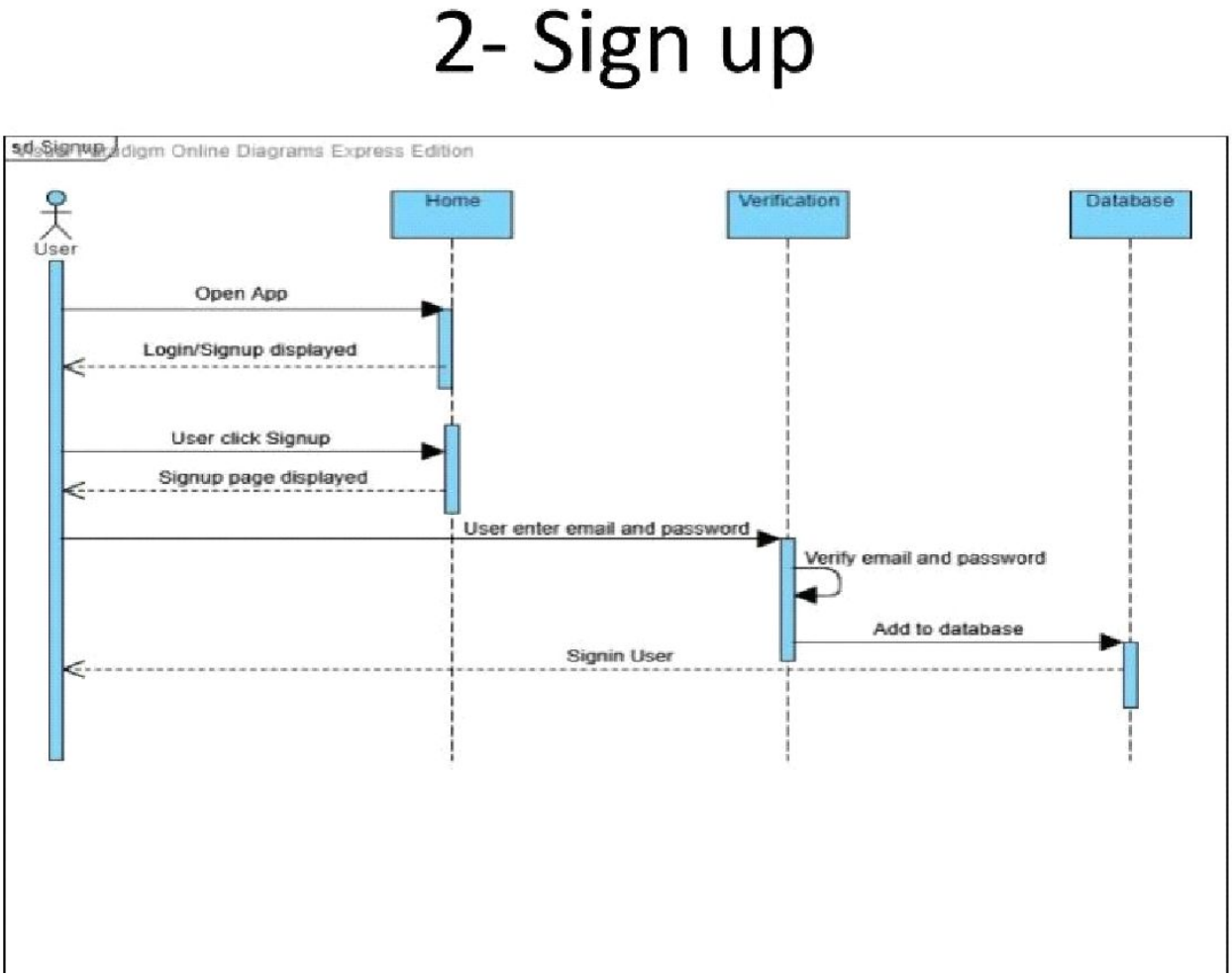



Figure 3.6 Sequence Diagram (Signup)

### 3.4.3 Complains:

Figure 3.7 shows that Users want to send complaints to Admin about any worker's or anything else. Users open the complaint tab by clicking on the complaint tab option and then write and send a complaint. System receives any complaints from users and forwards it to the Admin.

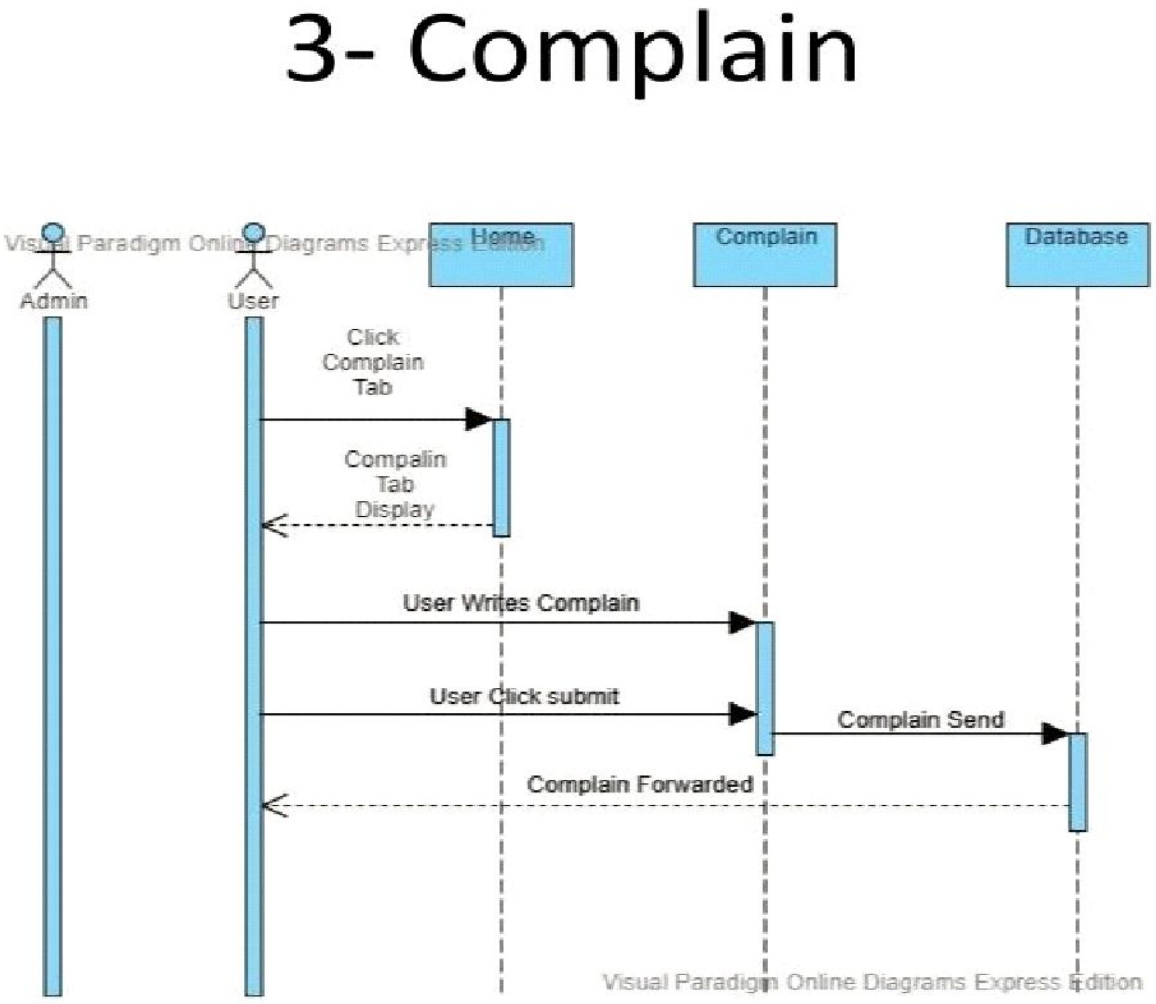



Figure 3.7 Sequence Diagram (Complains)

### 3.4.4 Alert Message:

Figure 3.8 shows that Admin want to send emergency alerts or tips for users. Admin open Alert tab by clicking on alert tab option and then generate and post any alert. System receives any alerts from Admin and forwards it to User.

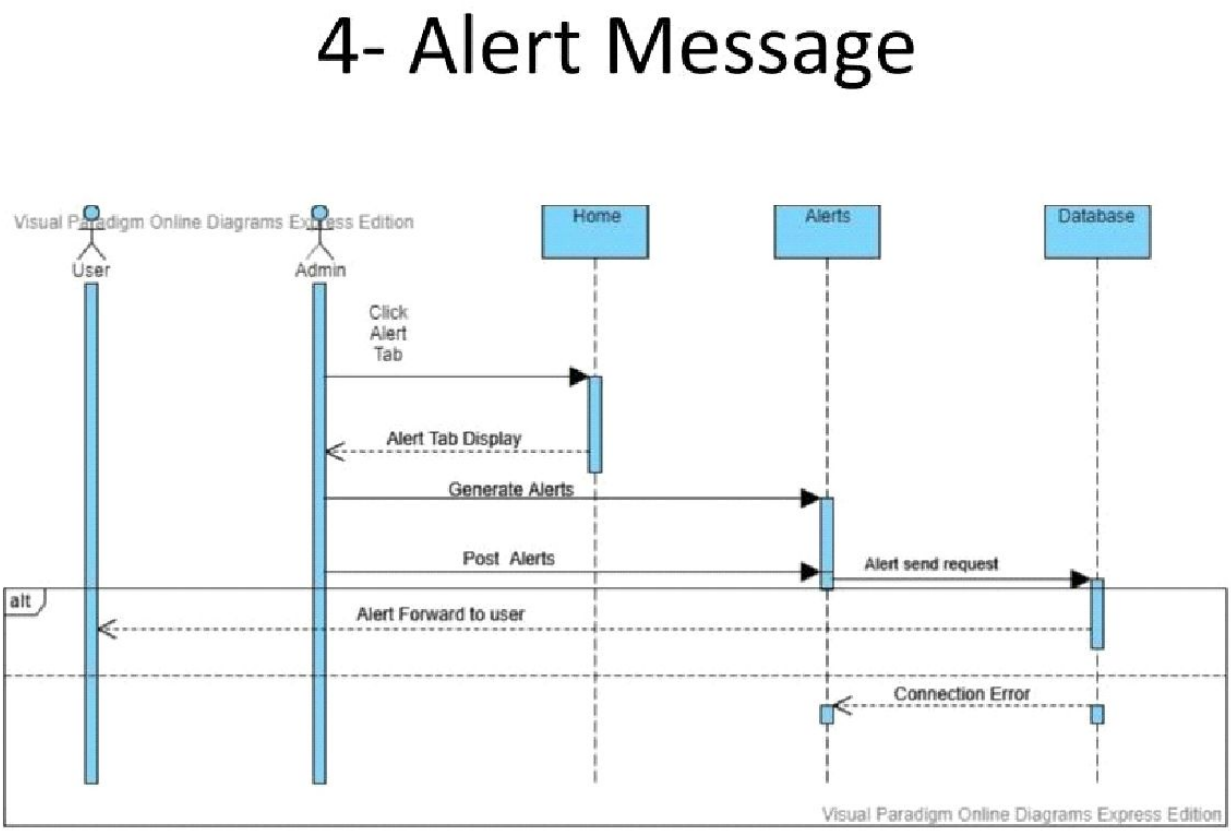



Figure 3.8 Sequence Diagram (Alert Messages)

### 3.4.6 Manage Worker(Add Worker)

Figure 3.9 shows that Admin want to add a new worker record in the database. Admin open add new worker tab by clicking on add new worker option, then enter biometric data with new name and phone number and then forward or save record in database. System generates "new worker add successfully" message to Admin. If the same record is already present in the database the system generates a "worker already exists" message to Admin.

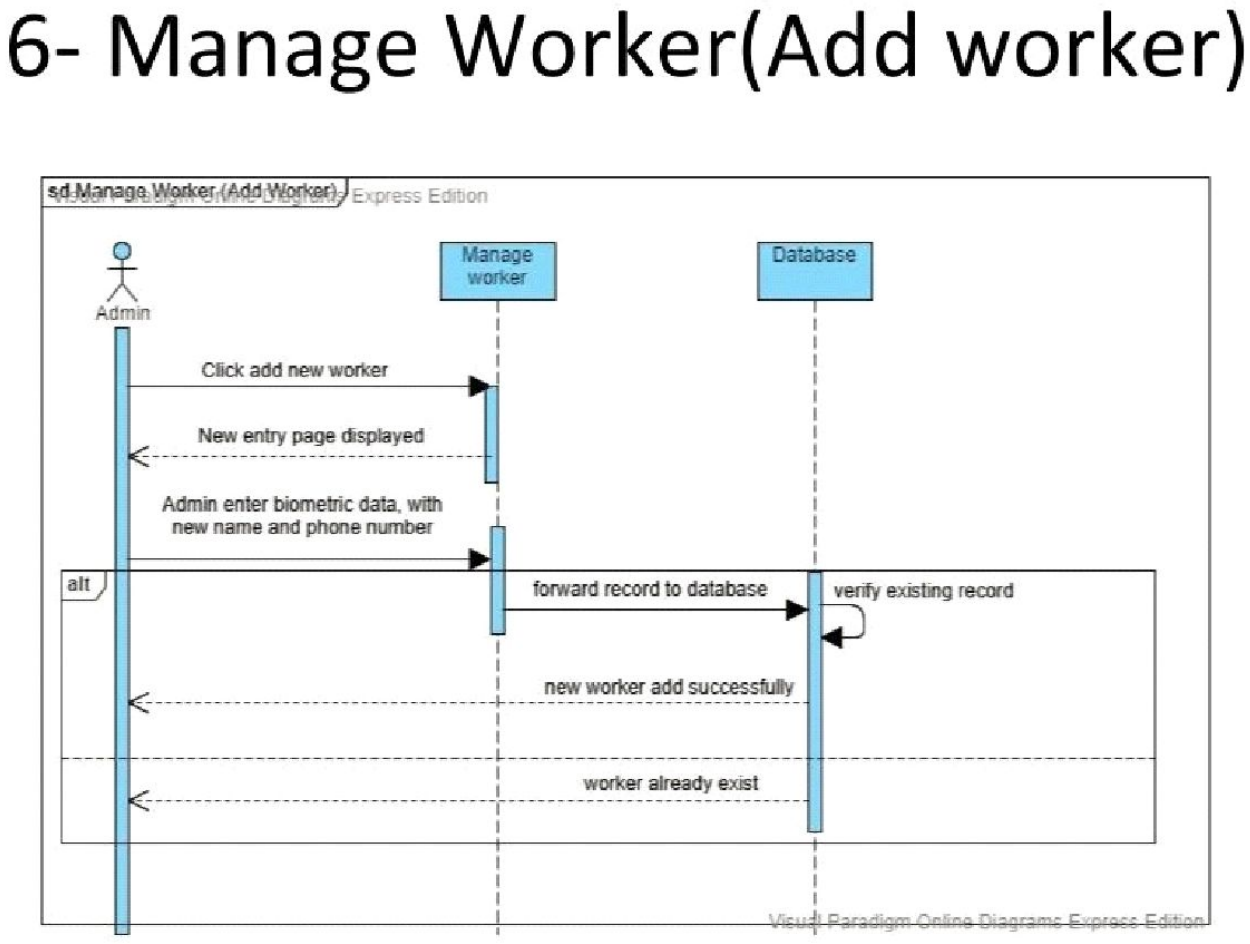



Figure 3.9 Sequence Diagram (Add Worker)

### 3.4.7 Manage Worker(Edit Worker):

Figure 3.10 shows that Admin want to edit worker records in the database. Admin open edit worker tab by clicking on edit worker option, then enter biometric data with new name and phone number and then forward or save record in database. System generate "record update successfully" message to Admin. If the same record is already present in the database the system generates an "already existing worker" message to Admin.

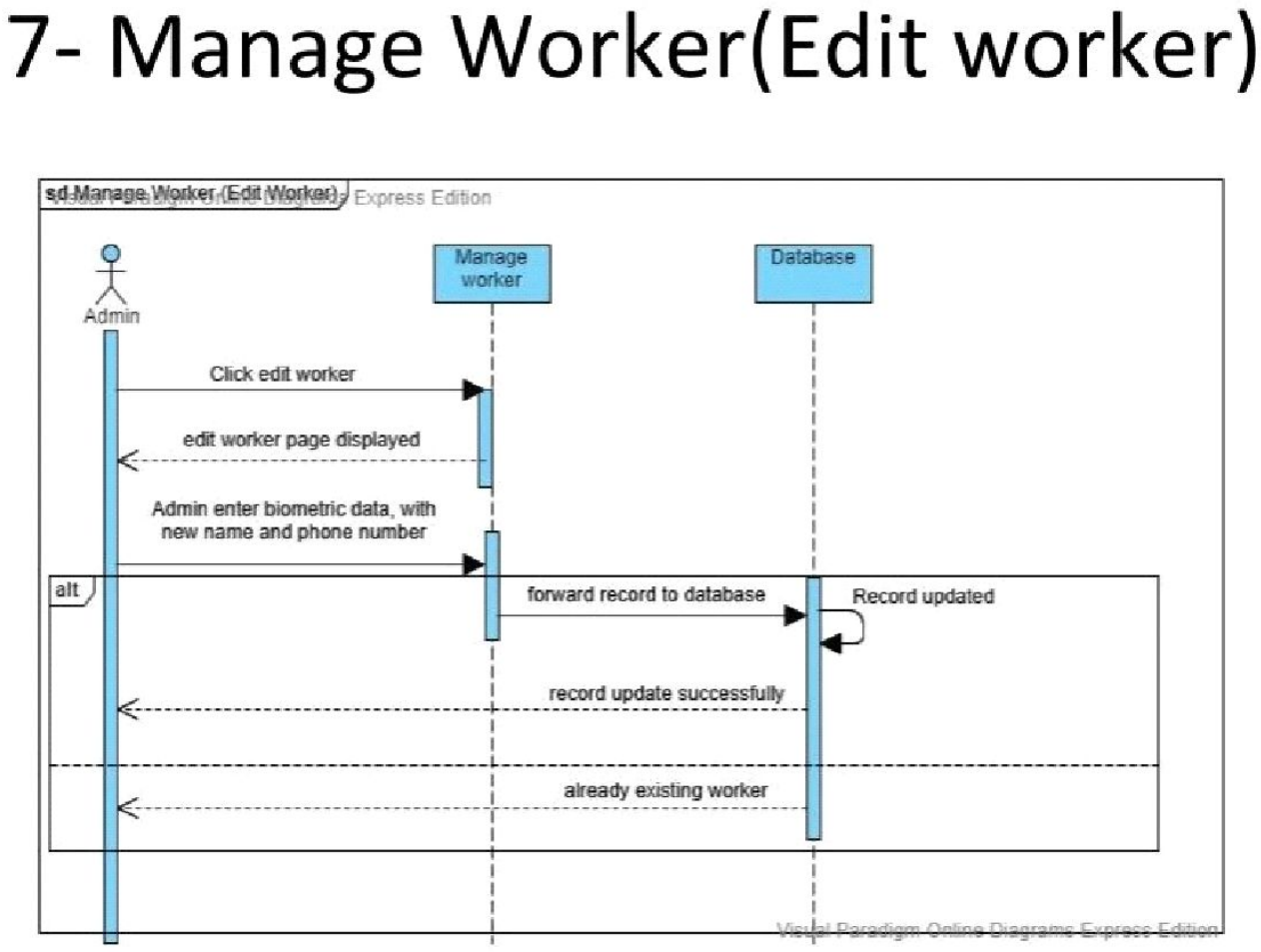



Figure 3.10 Sequence Diagram (Edit Worker)

### 3.4.8 Manage Worker(Delete Worker)

Figure 3.11 shows that Admin wants to delete worker records from the database. Admin open delete worker tab by clicking on delete worker option, then enter name or email and then forward to database. System finds those records and generates "worker record deleted successfully" message to Admin. If the same record does not exist or is present in the database the system generates a "worker doesn't exist" message to Admin.

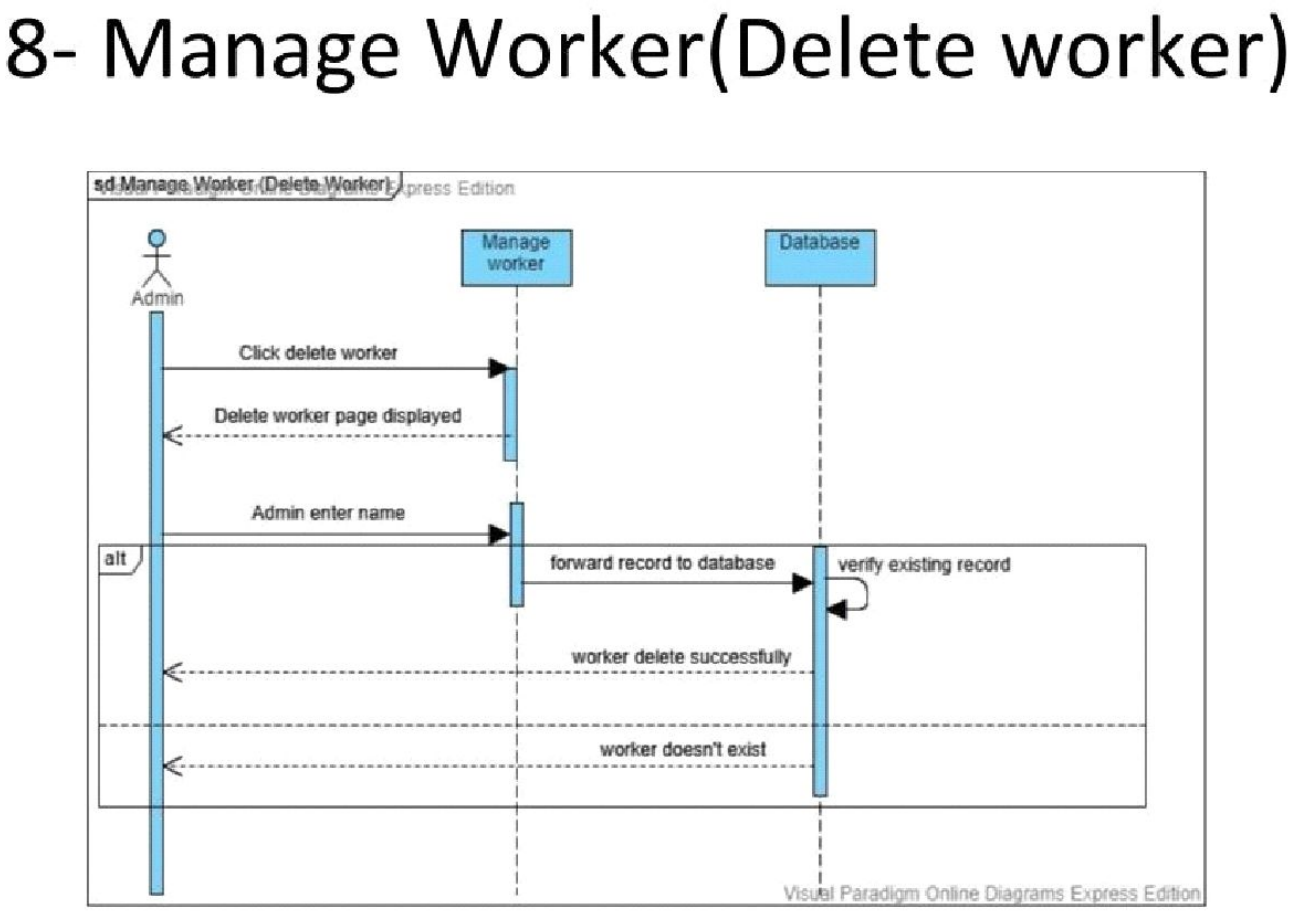



Figure 3.11 Sequence Diagram (Delete Worker)

### 3.4.9 Manage Worker(View Worker)

Figure 3.12 shows that Admin want to view worker's record from database. Admin open view worker's tab by clicking on view worker's record option. System finds those records and displays the worker's record to the Admin.

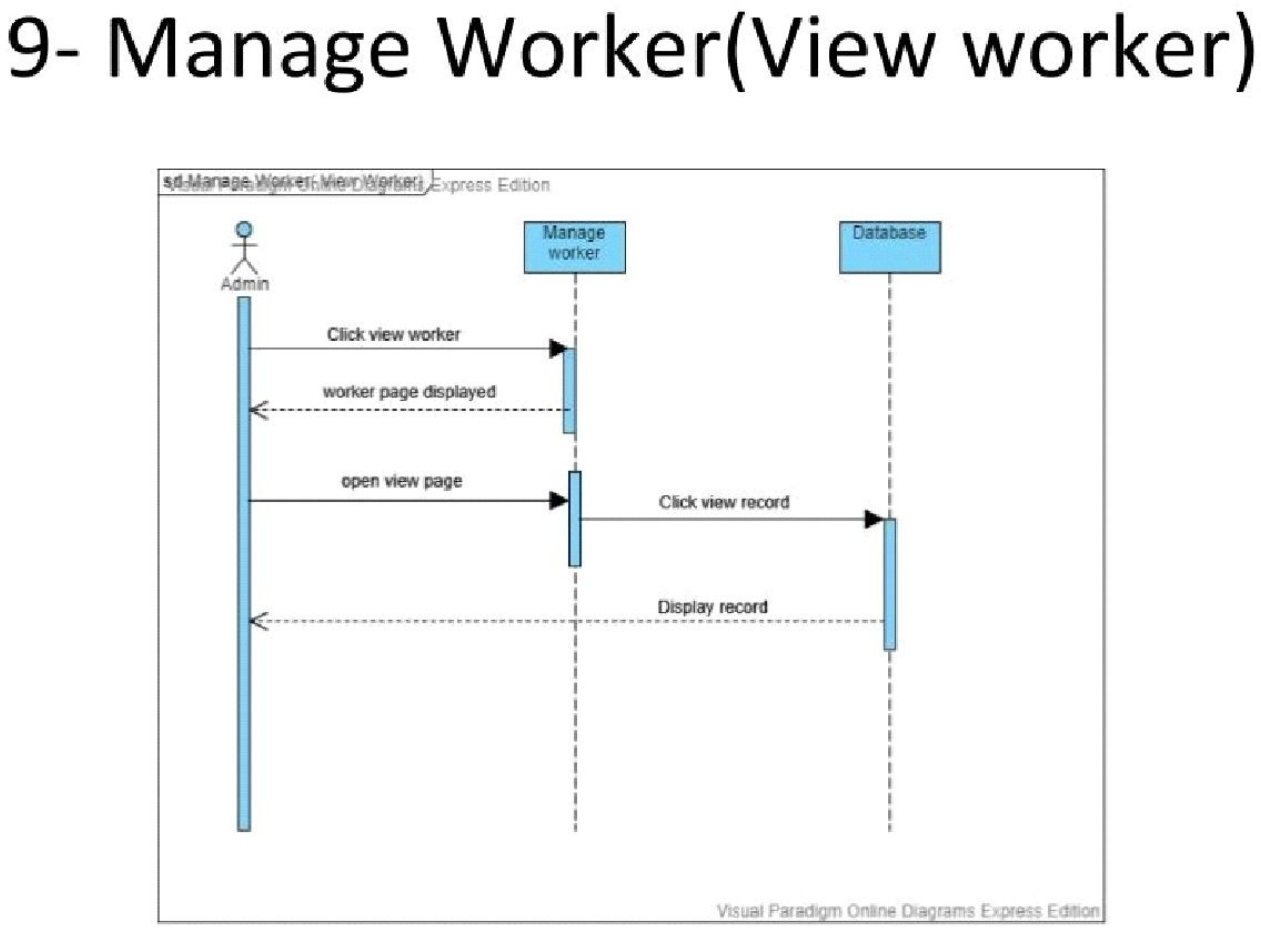



Figure 3.12 Sequence Diagram (View Worker)

### 3.4.10 Biometric Attendance

Figure 3.13 shows that worker's perform biometric attendance. Worker's place their thumb on a biometric module. Biometric module verifies worker's and forward worker's attendance to the database.

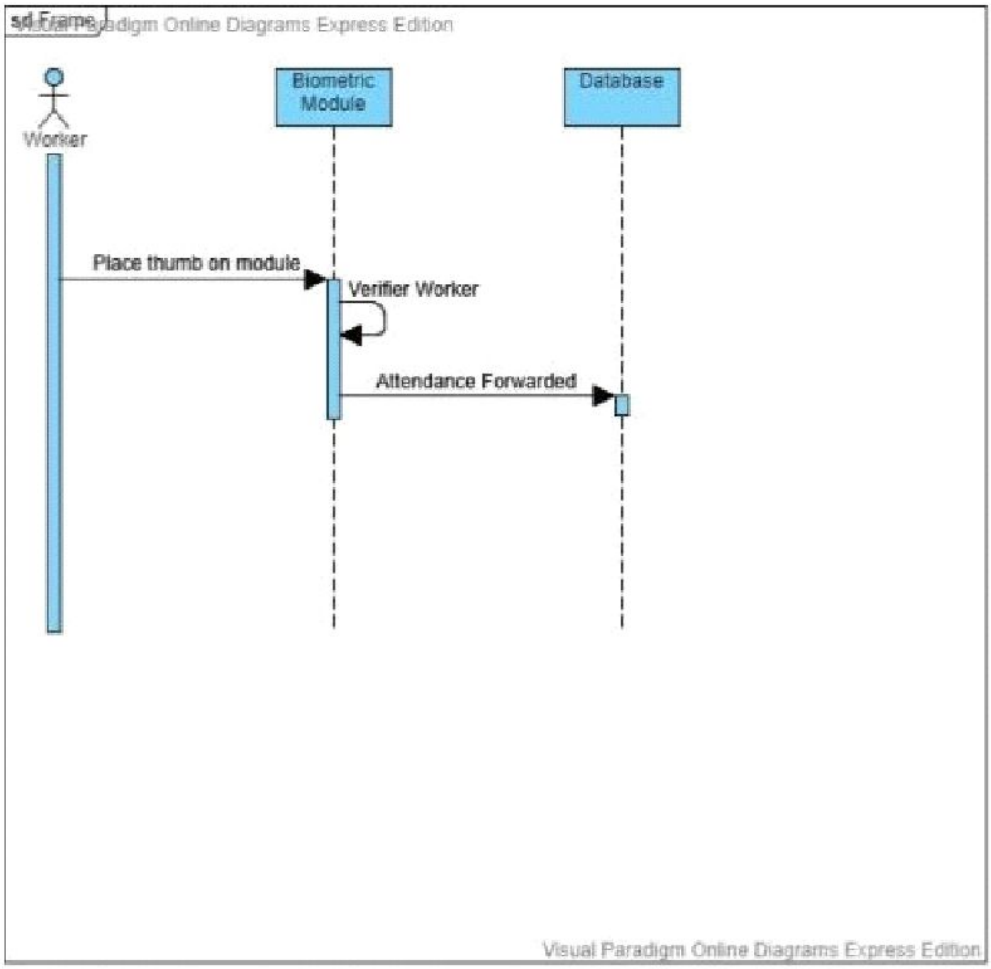



Figure 3.13 Sequence Diagram (Biometric attendance)

## 3.5 Functional Requirement

Need of the customer. Functional requirements characterize a framework or a part of it. It depicts the capabilities a product should perform. Capacity is only the input, its behavior, and yield.

- Admin
- Workers
- User

### 3.5.1 Admin:

- Can login.
- Can add, edit, delete, view workers.
- Can receive records from workers and save into databases.
- Can give complaints from users.
- Can get alerts to users.
- Can see the location of any worker.

### 3.5.2 Worker:

- Can login.
- Can accept/reject user emergency requests.
- Can visit their portal profile on their website.
- Can provide information about current emergency situations to the admin.
- Worker's have a unique profile.
- Provides services in emergency situations only.

### 3.5.3 User:

- Can Signup.
- Can login.
- Can Visit Website
- Can complain.
- Can press the Emergency Button.
- Can view tips of emergency situations.
- Can get alerts from Admin.
- Can provide location in emergency situation

### 3.5.4 Login:

**Description and Priority**

Login is a mandatory requirement for an Admin, User and Worker to use this application. Users must have to provide email, password and username for login.

**Functional Requirements**

Feature Requirements: Login (Login-V)

| **Features Name** | Description |
|---|---|
| **Manual input** | App must provide a form to the User,Worker and Admin to enter the email and password. |
| **Authentication** | System must authenticate the email and password. |
| **Allow Access** | System must give access to the application if the profile is authenticated. |
| **Unauthorize Message** | System must display an error for an unauthenticated profile. |
| **Invalid input Message** | System must display a message for the invalid input. |
| | |
| | |

Login is a use case for functional requirement , in which Admin,Worker and User will provide a registered account information then system will authenticate it, if user is authenticated then access to application will be provided else denied.

### 3.5.5 Signup:

**Description and Priority**

Signup is also a mandatory requirement for a User in order to use the application, for which they fill signup form providing authentic credentials and click submit button. Then their data will be saved in the database.

**Functional Requirements**

Feature Requirements: Signup (Signup-V)

| **Features Name** | Description |
|---|---|
| **Manual input** | App must provide a form to User to enter the name, email and password and other information. |
| **Authentication** | System must authenticate or save its details.. |

| | |
|---|---|
| **Allow Access** | System must give access to the application if the profile is authenticated. |
| **Unauthorize Message** | System must display an error for an unauthenticated profile. |
| **Invalid input Message** | System must display a message for the invalid input. |
| | |
| | |

### 3.5.6 Manage Worker(Add Worker):

**Description and Priority**

Add Worker is a requirement for an Admin in order to use the application, for which they fill a Login form providing authentic credentials and click Add New Worker and submit a new worker record. Then their data will be saved in the database.

**Functional Requirements**

Feature Requirements: Add Worker (Add Worker-V)

| **Features Name** | Description |
|---|---|
| **Manual input** | App must provide a form to the Admin to enter the email and password . |
| **Authentication** | System must authenticate its details.. |
| **Allow Access** | System must give access to the application if profile authenticated, to submit a new Worker record. |
| **Unauthorize Message** | System must display an error for unauthenticated profile. |
| **Invalid input Message** | System must display message for the invalid input. |
| | |
| | |

### 3.5.7 Manage Worker(Edit Worker):

**Description and Priority**

Edit Worker is another option for an Admin in order to use the application, for which they fill a Login form providing authentic credentials and click Edit Worker Record and submit a new worker record. Then their data will be saved in the database.

**Functional Requirements**

Feature Requirements: Edit Worker (Edit Worker-V)

| **Features Name** | Description |
|---|---|
| **Manual input** | App must provide a form to the Admin to enter the email and password . |
| **Authentication** | System must authenticate its details.. |
| **Allow Access** | System must give access to the application if the profile is authenticated, to edit Worker records. |
| **Unauthorize Message** | System must display an error for an unauthenticated profile. |
| **Invalid input Message** | System must display a message for the invalid input. |
| | |
| | |

### 3.5.8 Manage Worker(Delete Worker):

**Description and Priority**

Delete Worker from a Worker list is another option for an Admin in order to use the application, for which they fill a Login form providing authentic credentials and click delete Worker Record from a Worker's List. Then their data will be deleted in the database.

**Functional Requirements**

Feature Requirements: Edit Worker (Edit Worker-V)

| **Features Name** | Description |
|---|---|
| **Manual input** | App must provide a form to the Admin to enter the email and password . |
| **Authentication** | System must authenticate its details.. |
| **Allow Access** | System must give access to the application if the profile authenticated, to delete Worker records from a Worker's List. |
| **Unauthorize Message** | System must display an error for an unauthenticated profile. |
| **Invalid input Message** | System must display a message for the invalid input. |
| | |

| | |
|---|---|

### 3.5.9 Manage Worker(View Worker):

**Description and Priority**

View Worker list and Worker's record is another option for an Admin in order to use the application, for which they fill a Login form providing authentic credentials and click View Worker List .

**Functional Requirements**

Feature Requirements: Edit Worker (Edit Worker-V)

| **Features Name** | Description |
|---|---|
| **Manual input** | App must provide a form to the Admin to enter the email and password . |
| **Authentication** | System must authenticate its details.. |
| **Allow Access** | System must give access to the application if profile authenticated, to view Worker records. |
| **Unauthorize Message** | System must display an error for an unauthenticated profile. |
| **Invalid input Message** | System must display a message for the invalid input. |
| | |
| | |

### 3.5.10 Authenticates Admin,User,Worker's:

| **Identifier** | FR9 |
|---|---|
| **Title** | Authenticates Admin, User and Worker's. |
| **Requirement** | Primary actors' login in order to use the website, for which they fill signup form providing username and password then clicks login's button. Then their data will be authenticated from database after authentication and they are redirected to website homepage |
| **Source** | Validation and authorization system. |

| | |
|---|---|
| **Rationale** | It is required because it is important for Admin, users and workers should be properly authenticated in order to avail the services of websites. |
| **Business Rule** | Users must be have Pakistan cnic holder. |
| **Priority** | High |
| | |

### 3.5.11 User's Access:

| | |
|---|---|
| **Identifier** | FR9 |
| **Title** | User’s access |
| **Requirement** | Users should be able to send emergency requests and also access to details and data that is entered by Admin etc. So should be able to view their profile and give feedback, contact them or report them |
| **Source** | User |
| **Rationale** | This is necessary because users have to be sure that they are informed the right way in order to acquire the services. |
| **Business Rule** | User must properly login in the app and send an emergency request and location about the incident. |
| **Dependencies** | Emergency Request(Location), feedback, recommendation system, rating system. |
| **Priority** | High |

### 3.5.12 Worker's Communication:

| | |
|---|---|
| **Identifier** | FR10 |
| **Title** | Worker's communication |
| **Requirement** | Worker's should be able to communicate with each other in order to avail their services, they should be able to send them texts to provide details about their any incident etc. |
| **Source** | User |
| **Rationale** | The motivation behind this is that direct communication is always important to present information clearly without any ambiguity or confusion. |

| | |
|---|---|
| **Business Rule** | User must properly Login the app and then send an emergency request. |
| **Dependencies** | Chat system, feedback, recommendation system, rating system. |
| **Priority** | High |

### 3.5.13 Password Recovery:

| | |
|---|---|
| **Identifier** | FR11 |
| **Title** | Password recovery |
| **Requirement** | The system shall be able to provide Admin, Worker's the facility to recover the forgotten password or username. Admin and Worker's could not access the account if they don't remember the password or user name. |
| **Source** | Password recovery system |
| **Rationale** | The motivation behind this is that Admin and Worker's sometimes forgets their password or user names this will helps them to gain access to the account |
| **Business Rule** | Admin and Workers must have registered their number or email address while making the account. |
| **Dependencies** | feedback, recommendation system, rating system. |
| **Priority** | High |

## 3.6 Non-Functional Requirements

Wants of customers. It tells us quality attributes of software such as availability, robustness, reliability, maintainability etc.

### 3.6.1 Usability:

As the system will be easy to handle and navigates in the most expected way with no failure. This application will be built on latest GUI and design patterns, e.g. Minimum taps to achieve a function.

The system should have a help portal which provides the user the orientation about the website, how to use it or how to utilize the site properly.

There will be privacy and terms and guidelines for the user, there he can easily understand.

### 3.6.2 Performance Requirement:

The system must be interactive and the delays involved must be less so, in every actionresponse of the system, there are no immediate delays. In case of opening windows forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds, in case of opening databases, sorting and evaluation there are no delays and the operation is performed in less than 2 seconds for opening, sorting, computing, posting for greater than 95% of the files. Also, when connecting to the server the delay is based on editing on the distance of the 2 systems and the configuration between them so there is high probability that there will be or not a successful connection in less than 15 seconds for sake of good communication.

The system should work on all platforms (liek Android) seamlessly, it should show responsive behavior to users across different platforms. System should fetch all the details from the database within 5 seconds.

As this is also a web application, the performance of any changes depends mainly on the internet connection and host on which application is deployed.

#### 3.6.2.1 Start-up Time:

Application will start in seconds but it depends upon the internet connection because it is a web application and client-server android application.

#### 3.6.2.2 Response Time:

Application response time should be fast so that the information will be quickly seen within a limited time.

#### 3.6.2.3 Workload:

System should be able to deal with multiple users at the same time.

### 3.6.3 Safety and Security Requirement:

- Security requirements are important factors in this system as classified data will in the database.
- Admin and Worker's validation will be done during login to insure that the Admin and Worker's is valid and that the Admin, Worker's user only has access to his or her permission data
- Users will only have access through the user interface.
- The system shall permit only Admin, to add, edit, delete and view records.
- The system shall permit Workers who are not Admin to view only their own information, not the information of other Workers.

- Application will ensure that the data of Admin, Worker's and user should not be compromised.
- Information transmission should be securely transmitted to the server without any changes in information.

## 3.6.4 Software Quality Attributes

### 3.6.4.1 Availability:

- With internet availability, the website and application will perform without a single second of delay.
- The Application will be providing services 24/7.
- If the internet service gets disrupted while sending information to the server, the information can be sent again for verification

### 3.6.4.2Reliability:

System should provide reliability because all the information is fetched from the Server. Database information will be checked for any mistake or duplication error. To ensure reliability and correctness, it will be made sure to zero tolerance for errors.

### 3.6.4.3 Security:

The main security concern is for users' accounts hence proper login mechanism should be used to avoid hacking. The Smart phones id registration is a way to spam check for increasing the security. Hence, security is provided from unwanted use of recognition software

### 3.6.4.4 Usability:

As the system will be easy to handle and navigates in the most expected way with no failure or delays. In that case the system program reacts accordingly and transverses quickly between its states. This application will be built on latest GUI and design patterns, e.g. Minimum taps to achieve a function.

### 3.6.4.5 Maintainability:

The website or app is designed with a way that it is easily maintainable if any error or bug occurs after deployment. The administrative team will be available to fix the error or bug which is reported by the user.

#### 3.6.4.6 Extendibility

The application should be easy to extend. The code should be written in a way that it favours implementation of new functions. In order for future functions to be implemented easily to the application.

#### 3.6.4.7 Testability

Test environments should be built for the application to allow testing of the application's different functions. Application will be tested on a real device with all required features.

#### 3.6.4.8 Adaptability:

Since it is simple to use for any non-computer background user. This software product shall not require the acute knowledge of computers and it will be user friendly.

# Chapter no 4: Design and Architecture

The following parts of Software Design Description (SDD) report should be included in this chapter.

## 4.1 Class Diagram

Figure 4.1 shows the static structure diagram. It is used for the generally calculated presentation of the application structure, and for each point after another which indicates making the interpretation of models to the programming code.

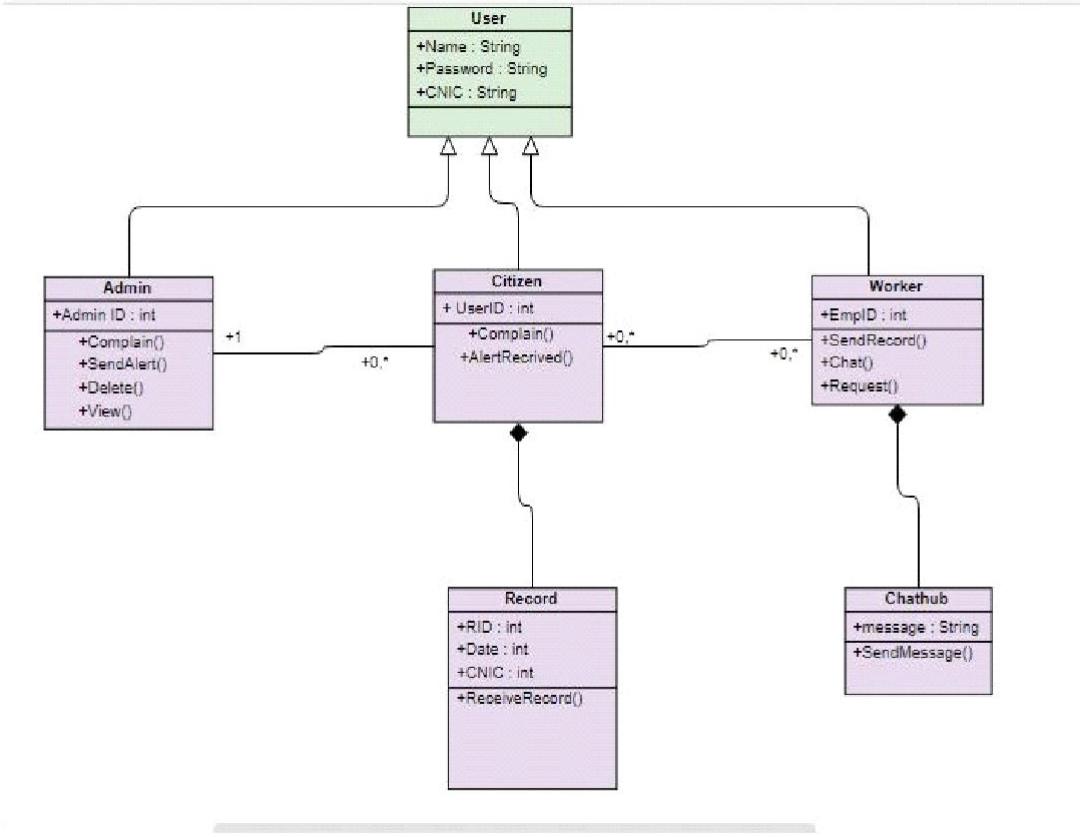



Figure 4.1 Class Diagram

## 4.2 ERD Diagram

Figure 4.2 shows the entity relationship between actors or modules. An Entity Relationship Diagram shows entities (tables) in a database and relationships between tables within that database.

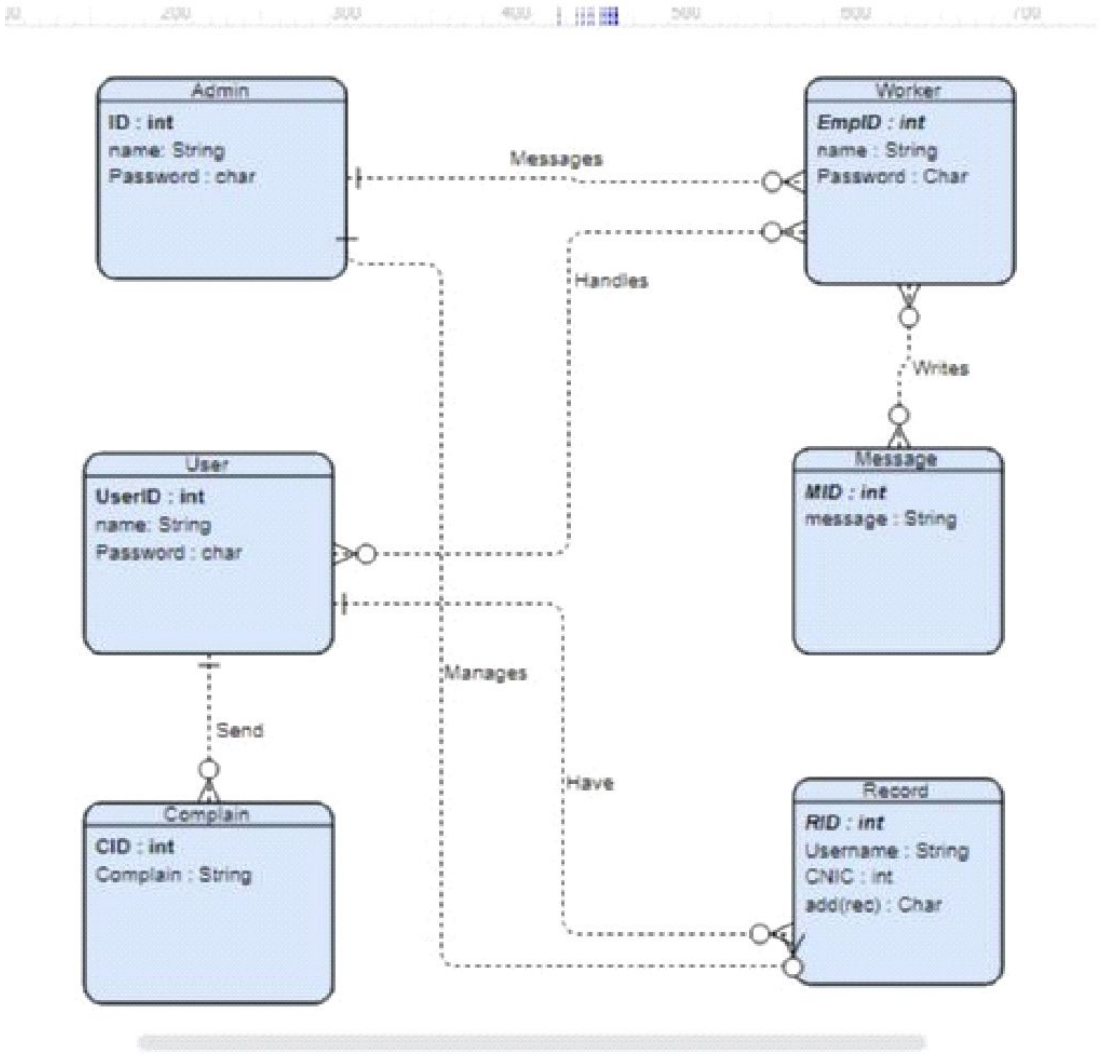



Figure 4.2 ERD Diagram

## 4.3 System Architecture

Figure 4.3 shows it is a structure of software. In addition, this structure has components, relations among them, and properties of the two components and relations.

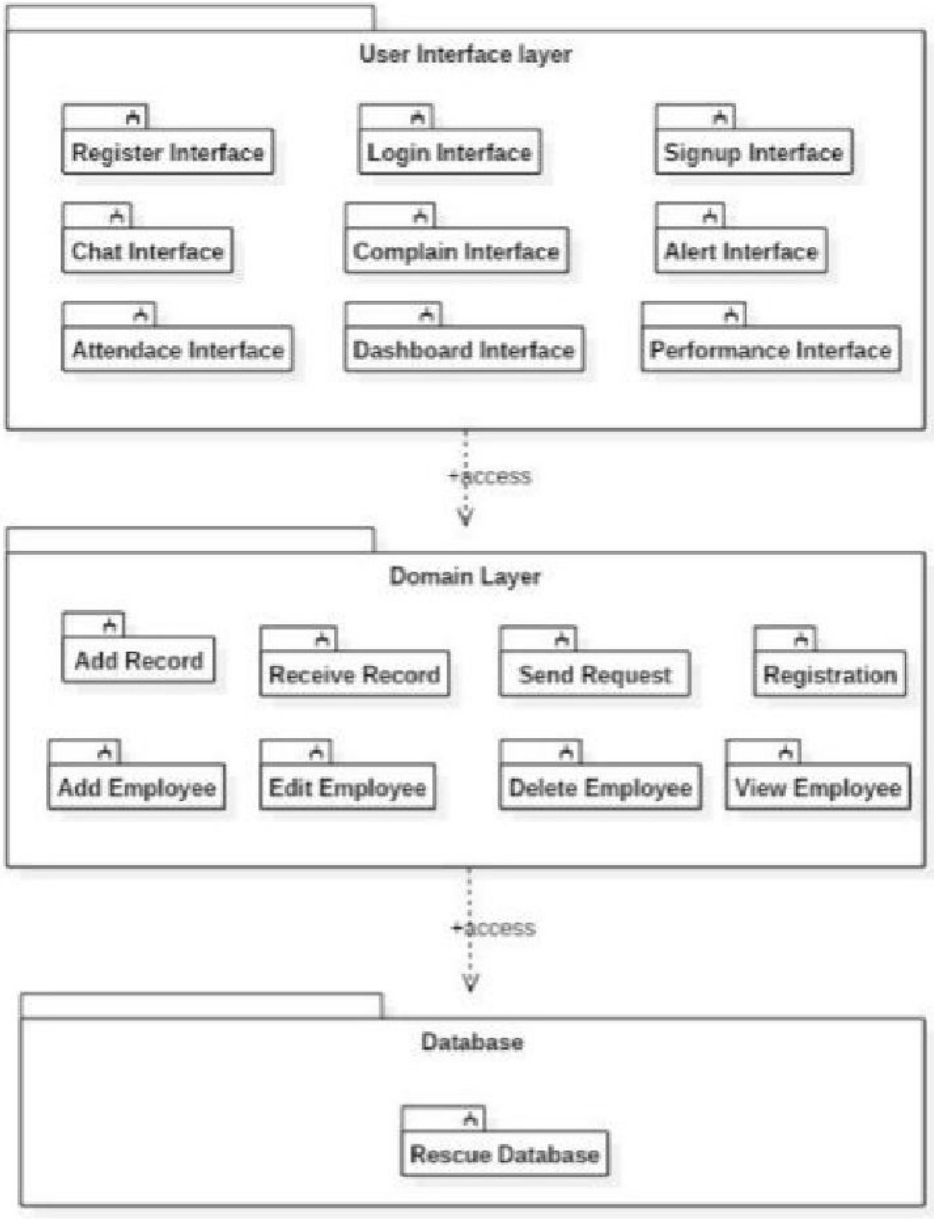
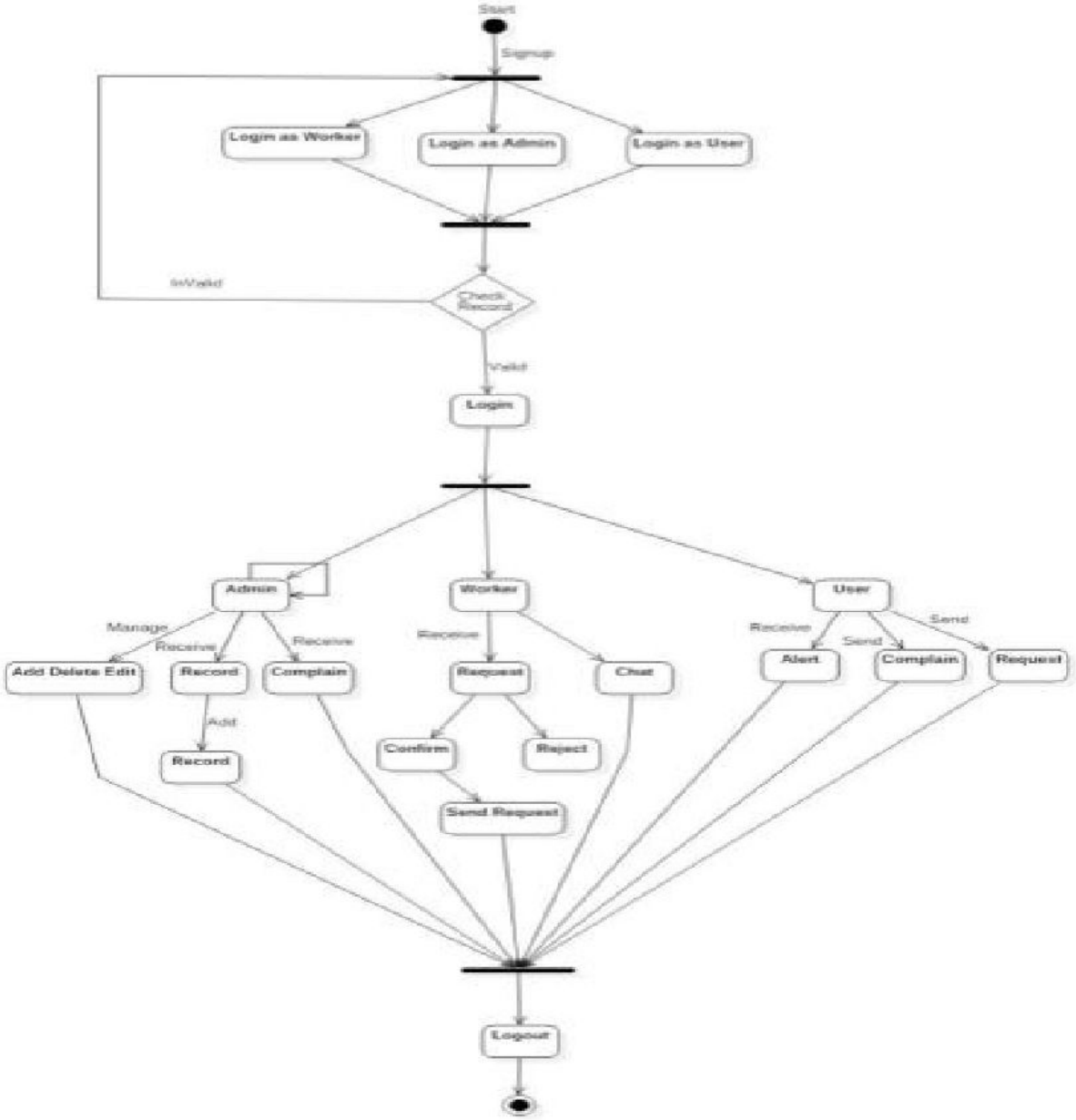



Figure 4.3 System Architecture

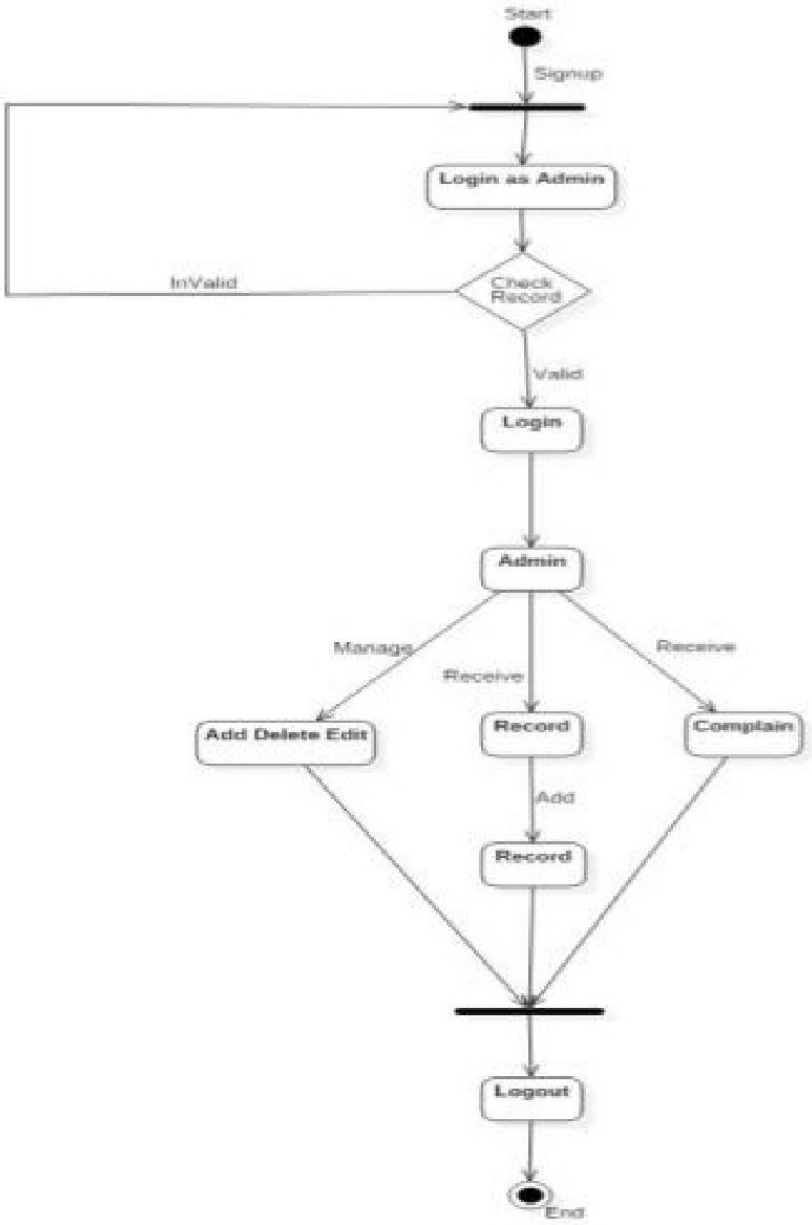
### 4.4 Activity Diagram

Figure 4.4.1, 4.4.2, 4.4.3 and 4.4.4 shows the flow of tasks performed by the system. The control current is drawn from one activity to the next. This flow can be consecutive, extended or synchronous



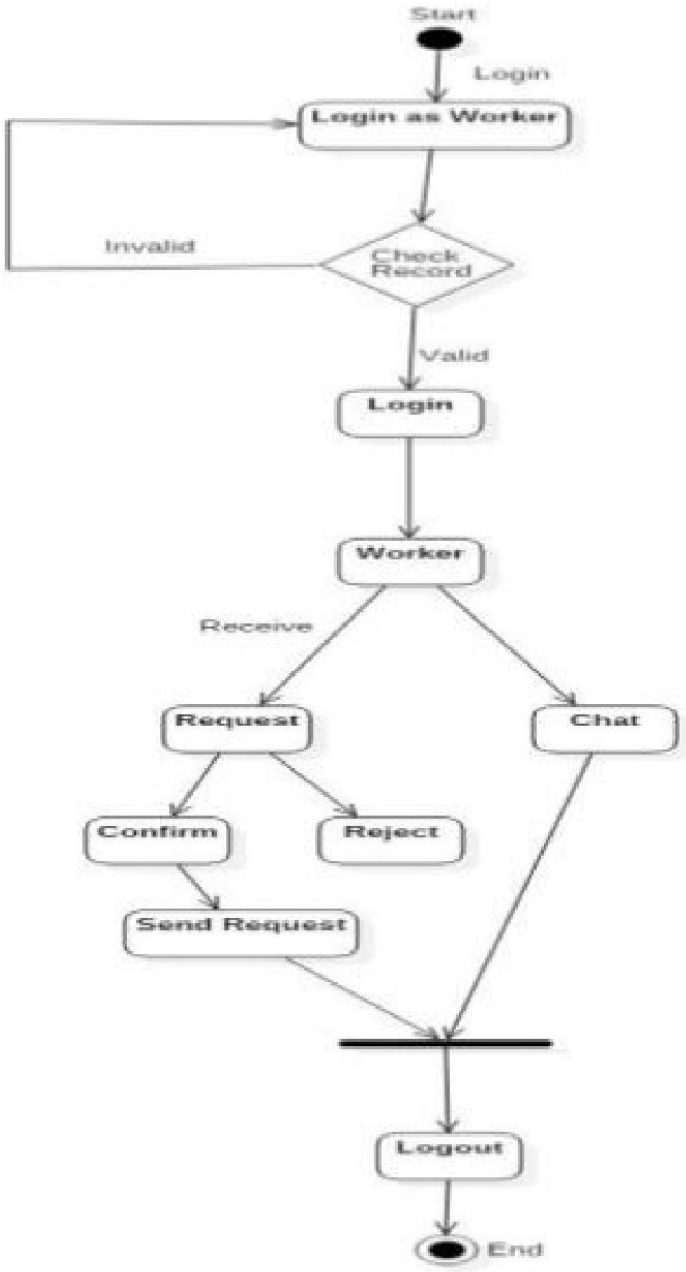

Figure's 4.4.1 Activity Diagram For Whole System

### 4.4.2 Activity Diagram For Admin:



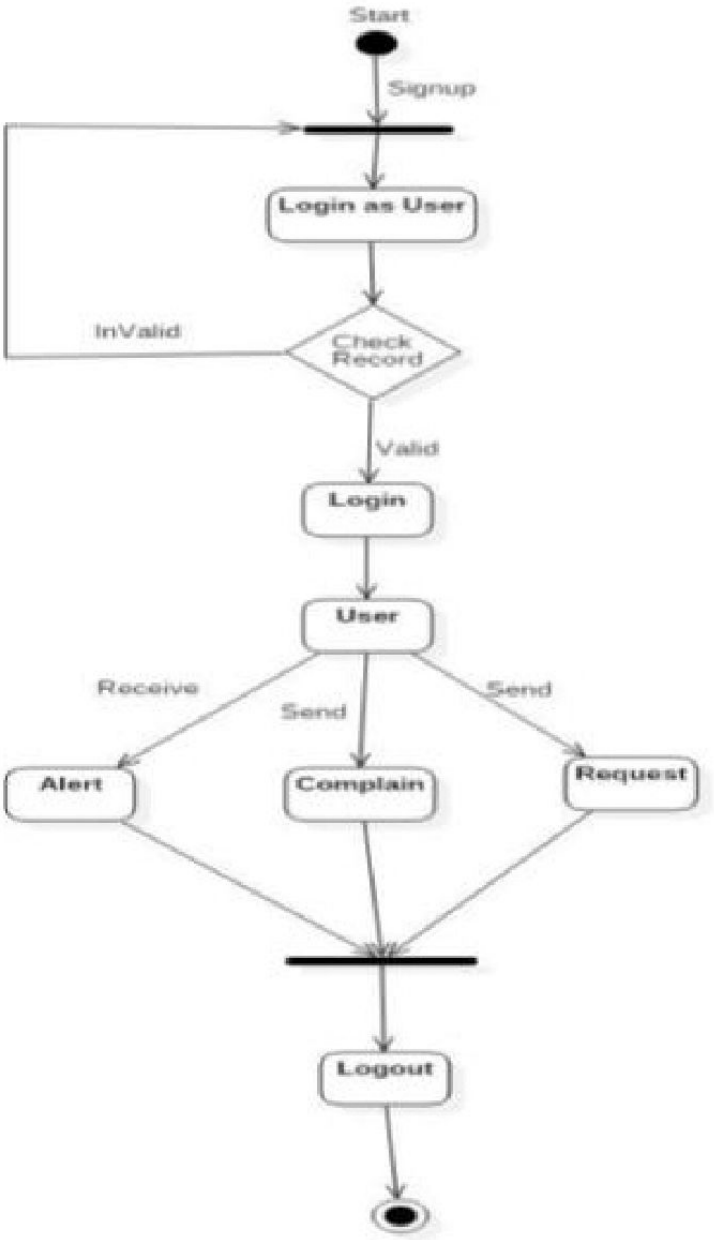

Figure's 4.4.2 Activity Diagram For Admin

### 4.4.3 Activity Diagram For Workers:




Figure's 4.4.3 Activity Diagram For Worker

### 4.4.4 Activity Diagram For User:




Figure's 4.4.4 Activity Diagram For User's

### 4.5 Package Diagram

Figure 4.5 shows packages. In addition, show how packages depend on each other. It can be utilized to disentangle complex class diagram, it can bunch classes into bundles (packages).

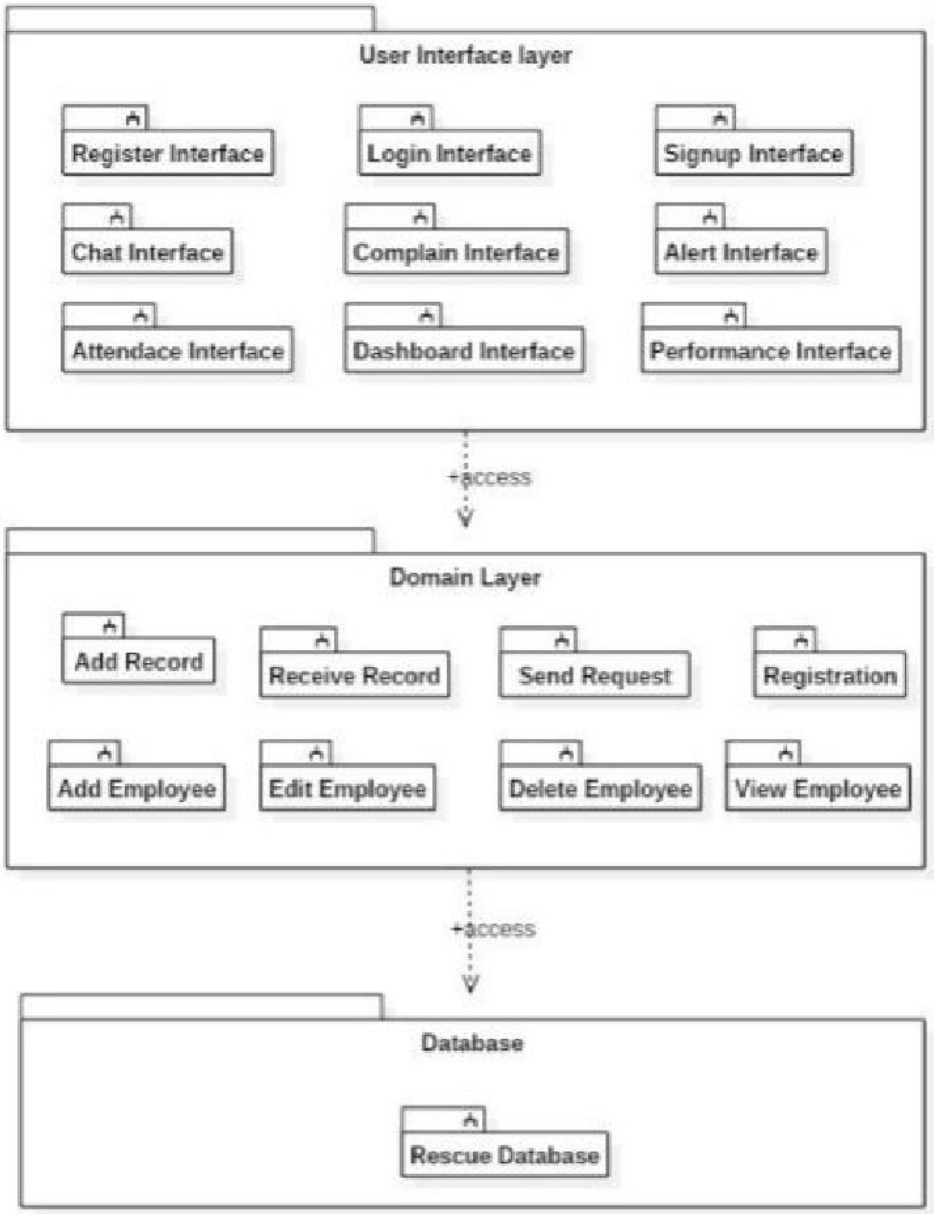



Figure 4.5 Package Diagram

## 4.6 Design Model

### Cyclic Iterative Model

Figure 4.6 shows object-based images that speak to the use cases for a framework. On the other hand, it is the way to portray a framework's usage and source codes.

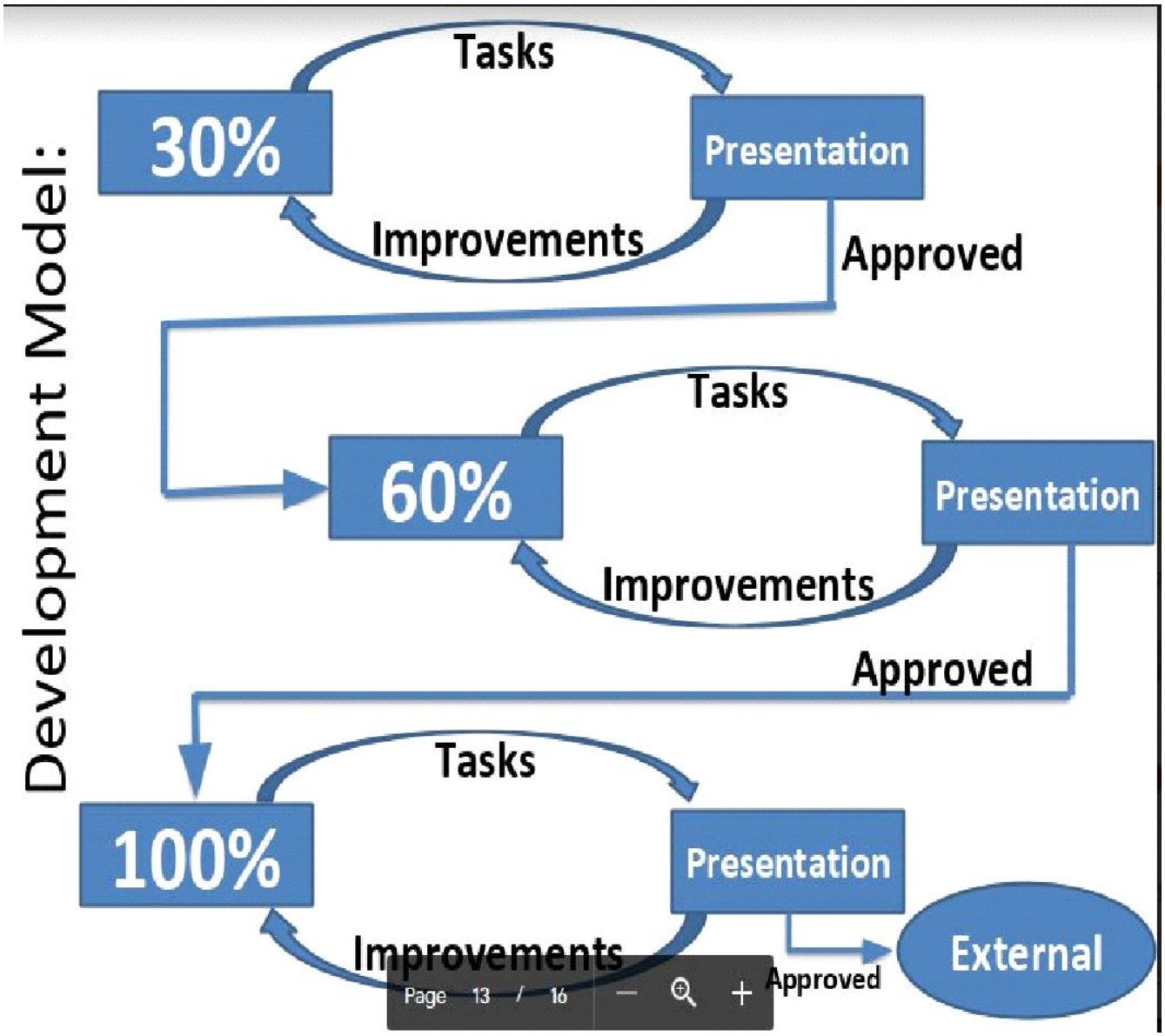



Figure 4.6 Design Model

## 4.7 Deployment Diagram

The below 2 Figure(Figure 4.7.1 and Figure 4.7.2) shows structure diagram. It is utilized in displaying the physical parts of an object-oriented system.

### 4.7.1 Deployment Diagram for Android app:

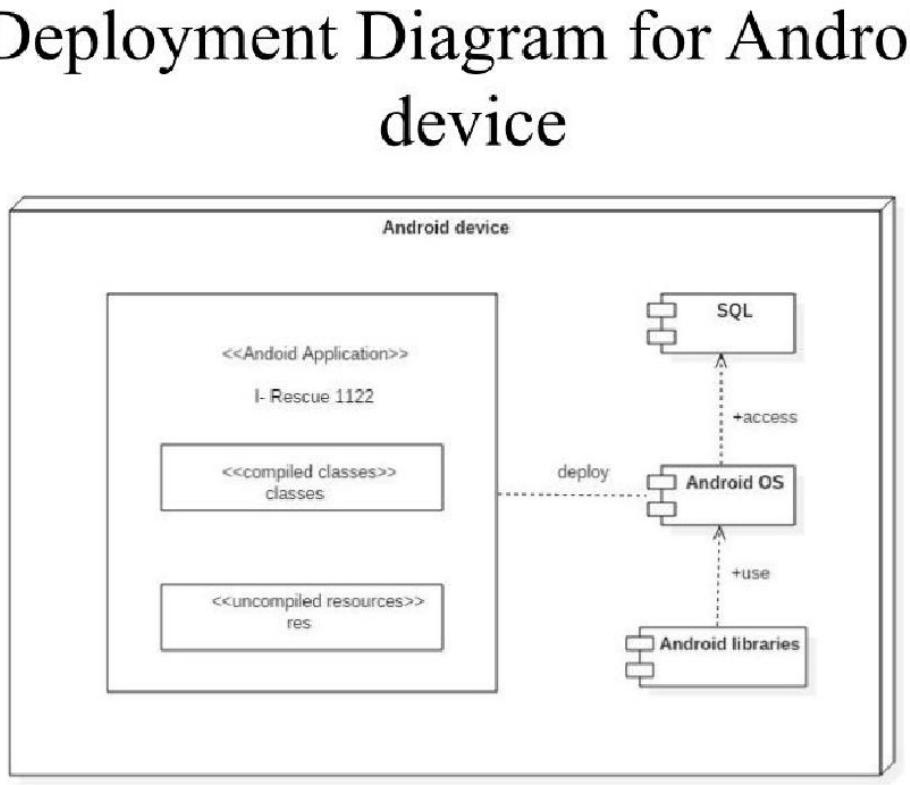





Figure 4.7.1 Deployment diagram for Android App

### 4.7.2 Deployment Diagram for Web Application:

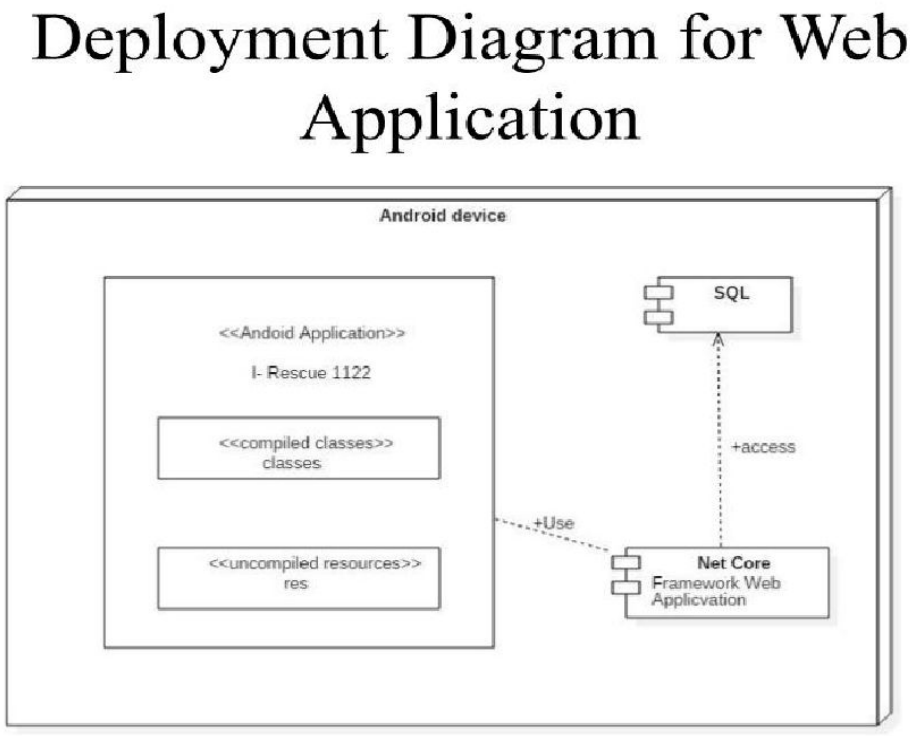





Figure 4.7.2 Deployment diagram for Web App

# Chapter 5: External Interface Requirements

This chapter will discuss implementation details supported by UML diagrams (if applicable). You will not put your source code here. Any of the following sections may be included based on your project.

## 5.1 User Interface

Application will contain following Interfaces for which Admin, Workers and User's interact with the system.

| Interfaces | Description |
|---|---|
| **Registration(Signup)** | This will provide a form to users for registration. Primary actors' signup in order to use the application, for which they fill signup form providing authentic credentials and click submit button. Then their |

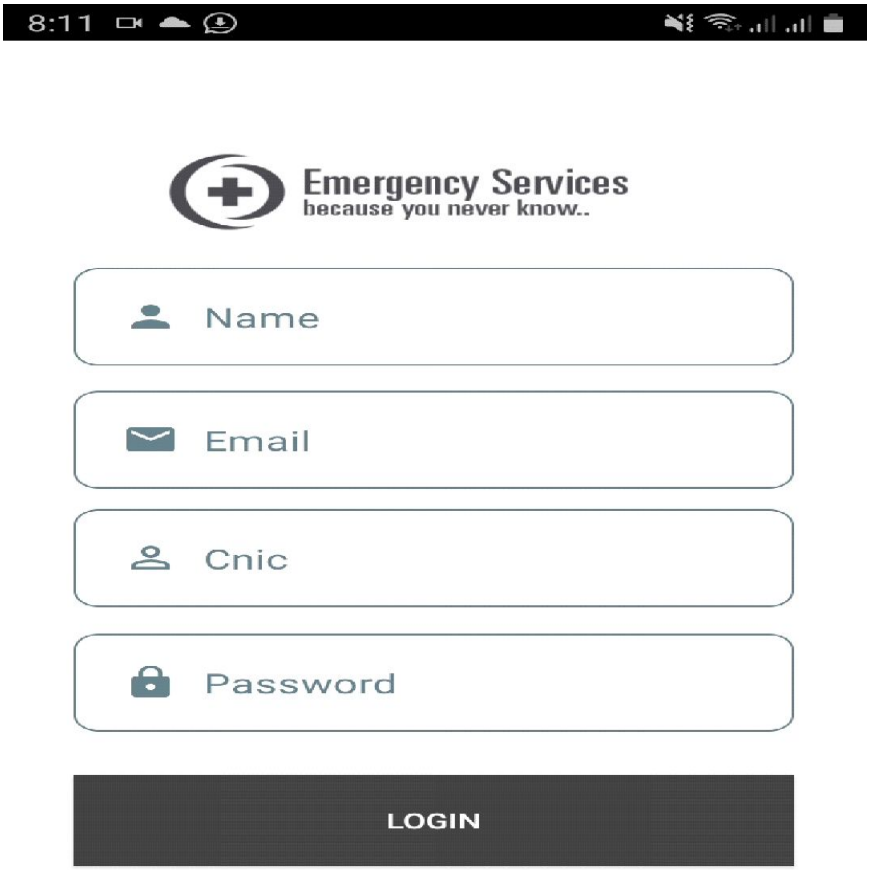
| | |
|---|---|
| | data will be saved in the database after authentication and they are redirected to the login page. |
| **Login** | This is a button which will open the Login Screen. User's, Admin or Worker's put Username and password and then Homepage is open. |
| **Add Worker** | This will open the Add Worker Record Screen in which Admin add a new worker record. |
| **Edit Worker Record** | This will open the edit Worker Record Screen in which Admin Edit already Existed worker record. |
| | |
| **Delete Worker's Record** | This will open the Delete Worker Record Screen in which Admin deletes existing worker records. |
| **View Worker's Record** | This will open the View Worker Record Screen in which Admin View the List of worker records. |
| **Alerts Messages** | This will open the View Alerts Messages Screen. User send emergency alerts or inform about incident location and Admin view emergency alert and send it to nearby workers. |
| **Complains** | This will open the User's complaints Screen. Users send complaint messages to the admin and complain saves in the database. |
| **Request(Accept,Reject)** | This will open the User's Request Screen. Users send emergency requests or inform about incident location . Admin send those requests to worker and worker ject or accept requests and perform operation as fast as possible. |
| **Chat** | Admin communicate with workers through chat rooms. |
| **View Emergency Tips** | This will open the Emergency Tips Screen. When User sends an emergency request or alert then the system automatically generates emergency tips for Users. |
| **Logout** | This is a button through which Admin,Worker's can logout. |

## Android App Main Interface :

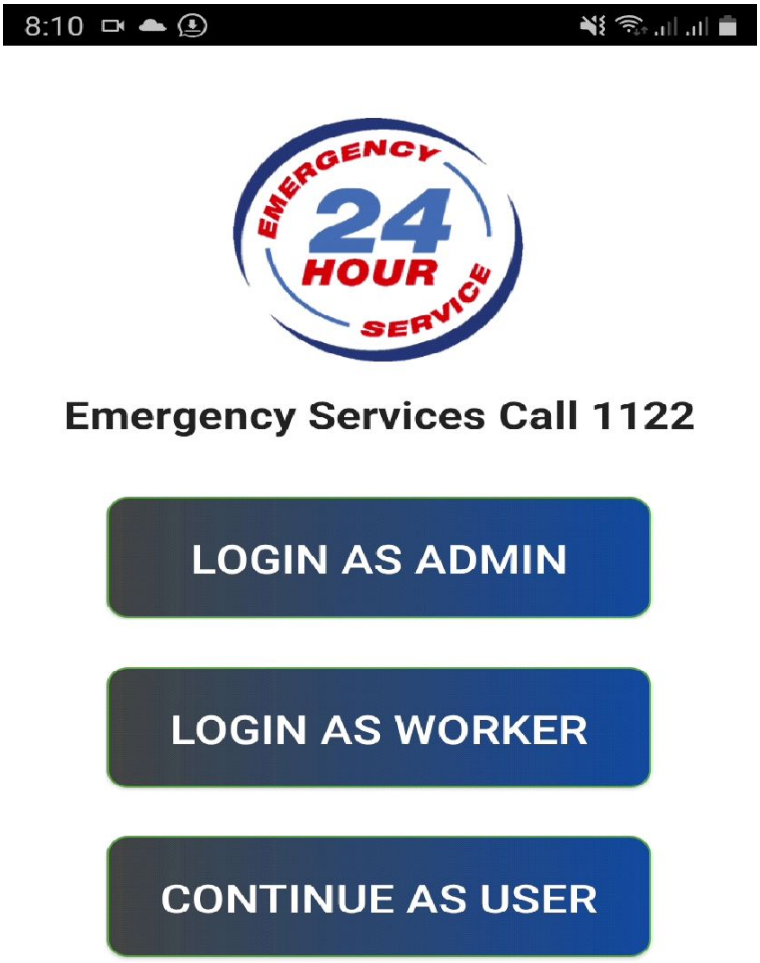


**Registration (Signup) Interface:**

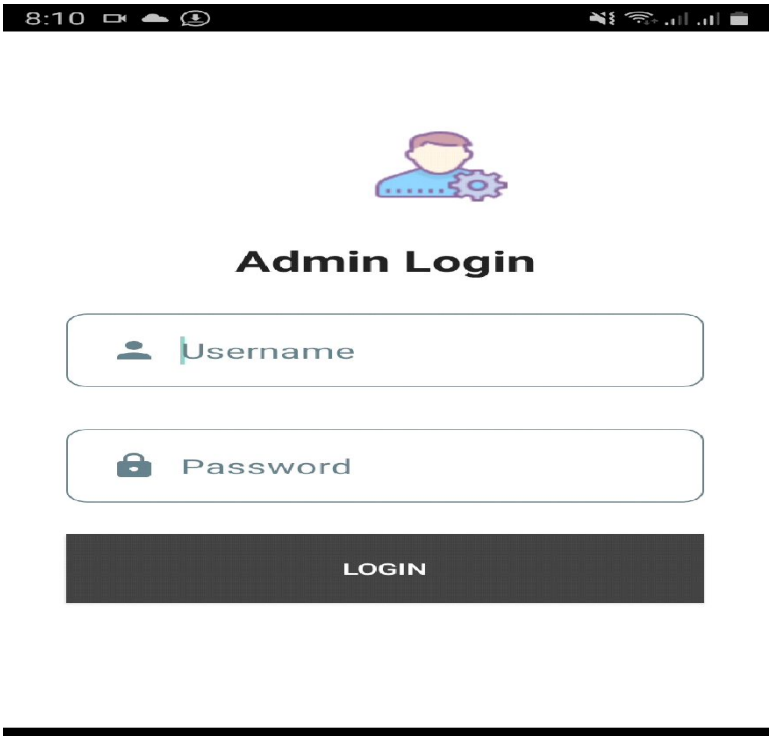
**Registration Form:**



**Login Interface:**

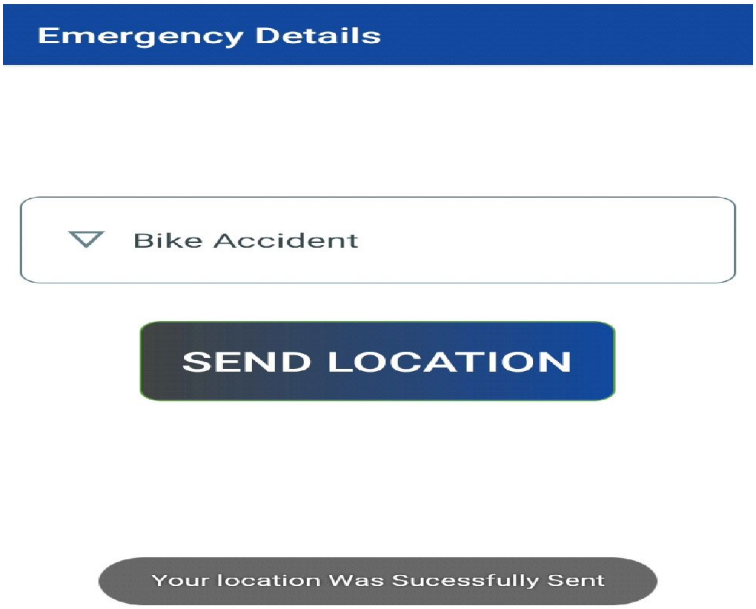


**Admin Login:**



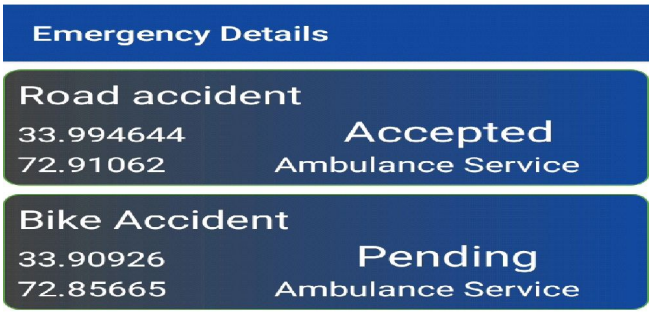

**Emergency Services Interface:**

**Emergency Alert(User send Location) Interface:**

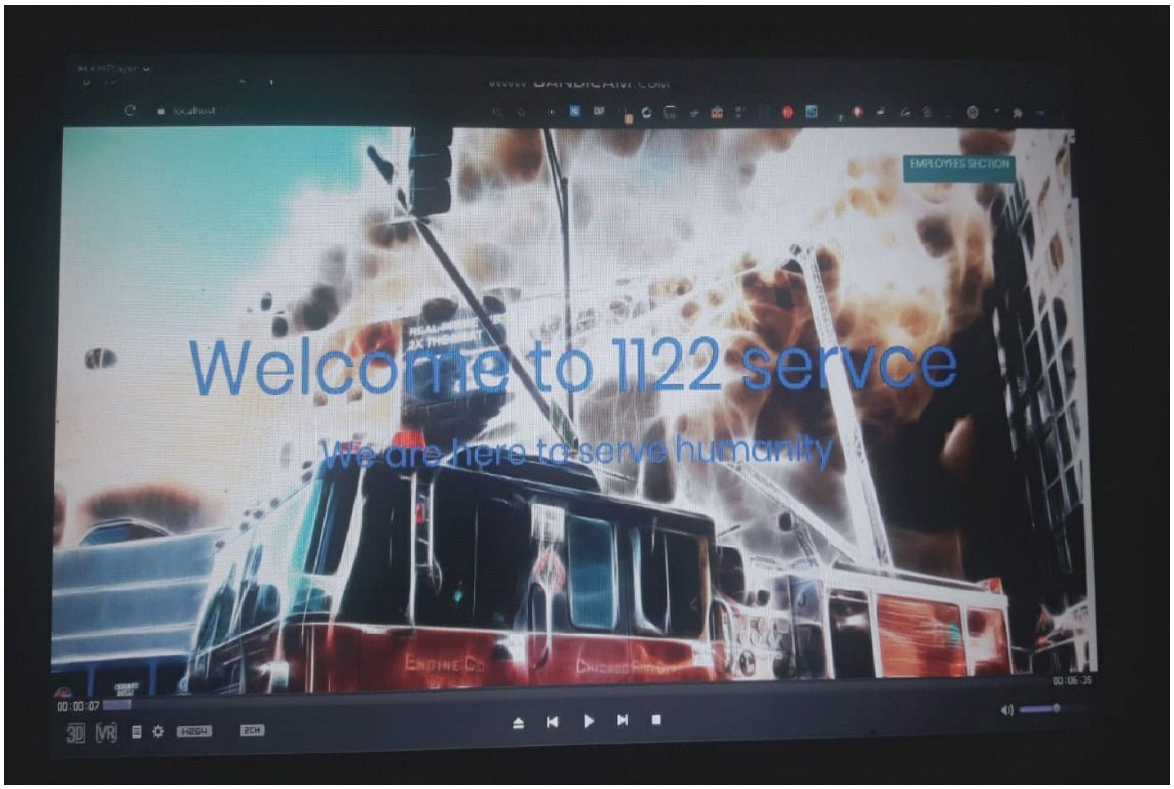



**Check Emergency Requests Interface:**

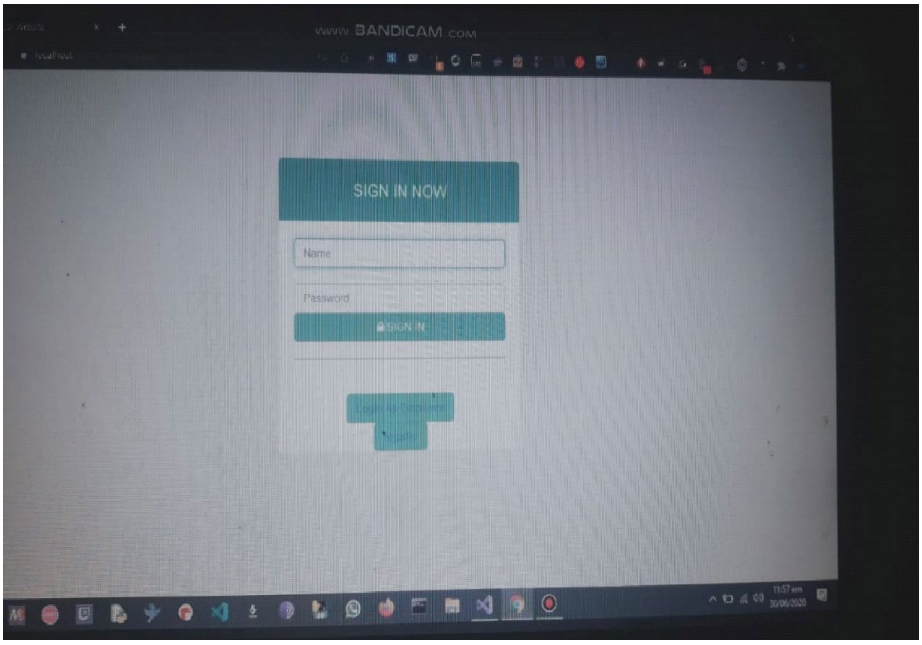
**Requests (Accept Or Reject):**




## Web App Interface

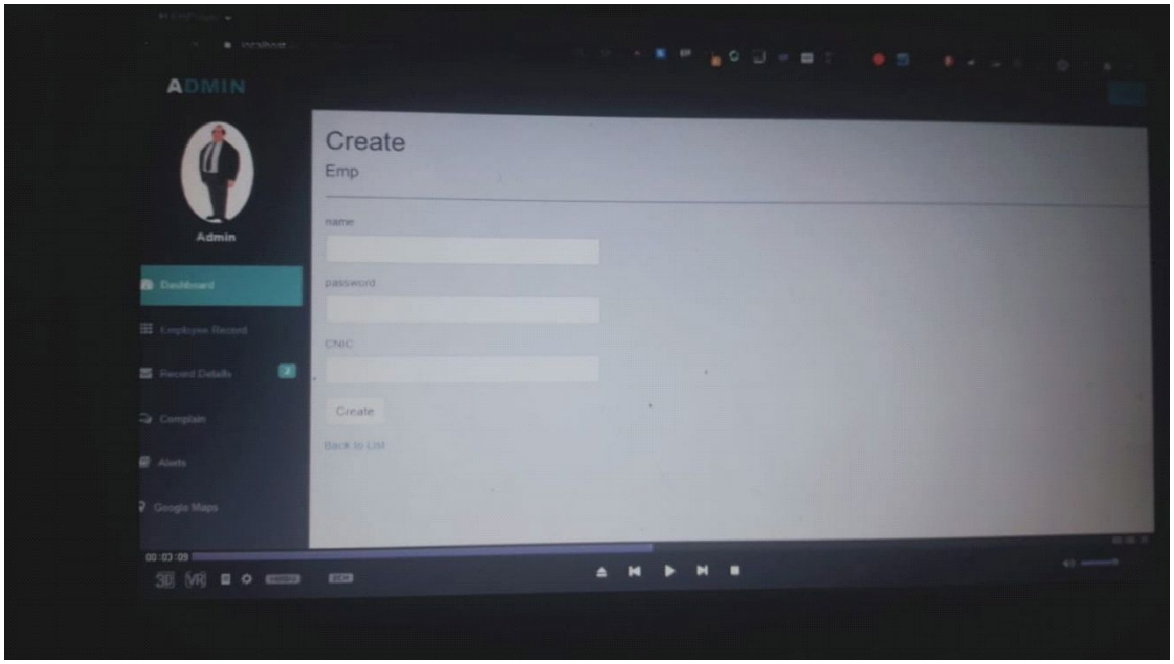


**Login Interface:**

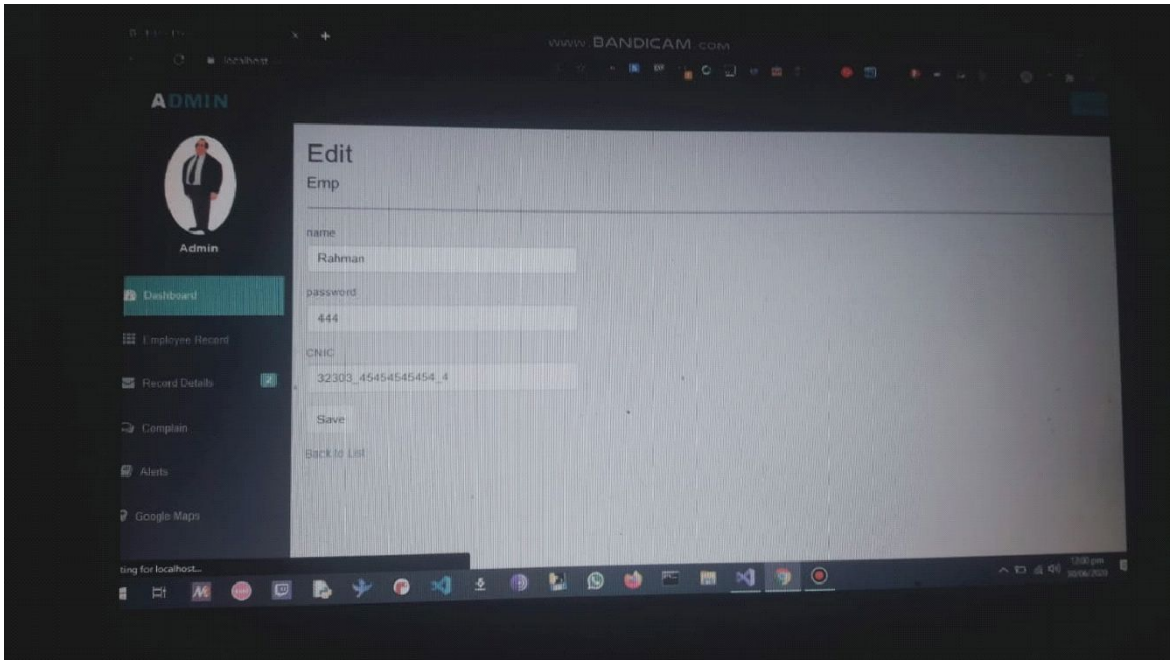



**DashBoard Interface:**

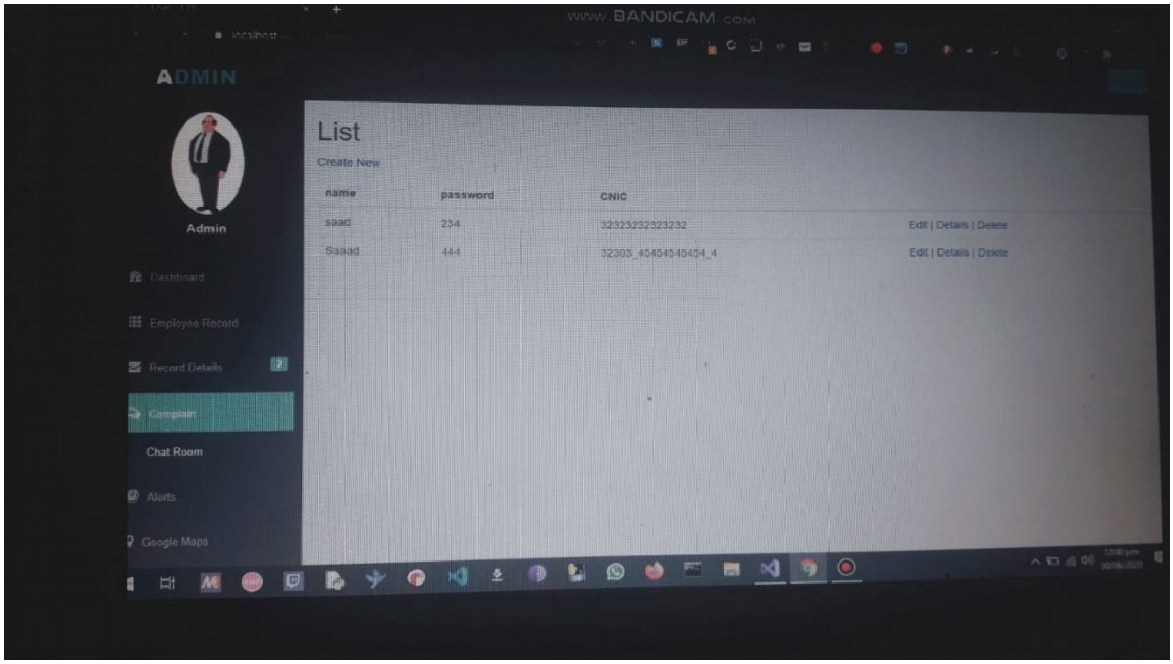
**Create or Add Worker Interface:**



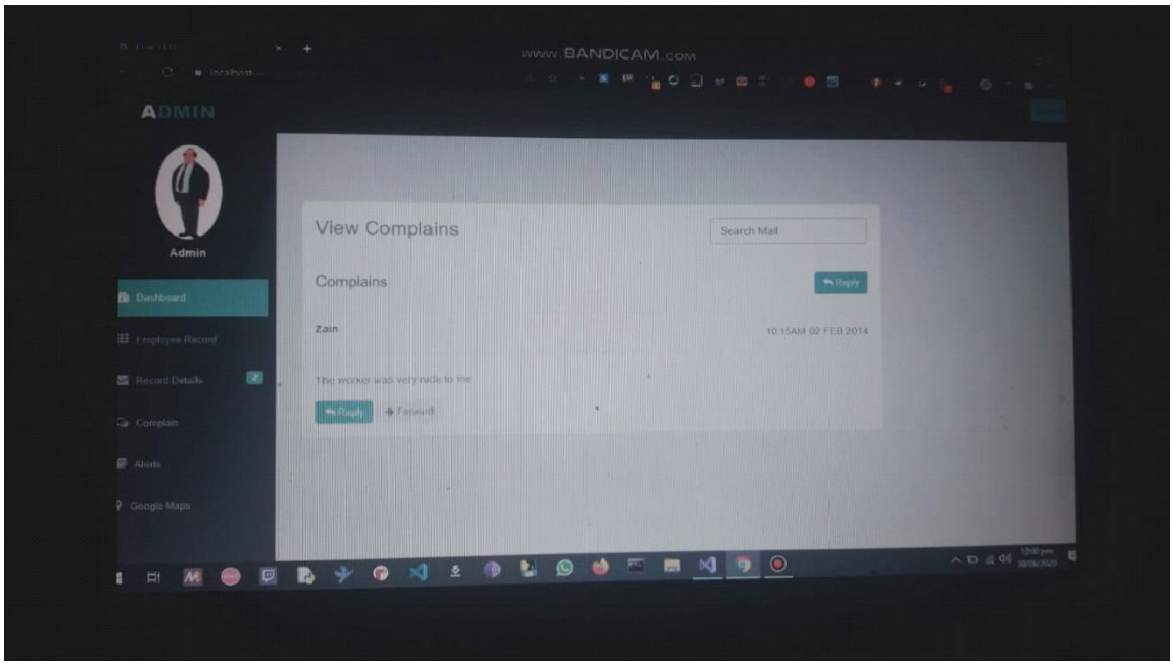

**Edit Worker Interface:**



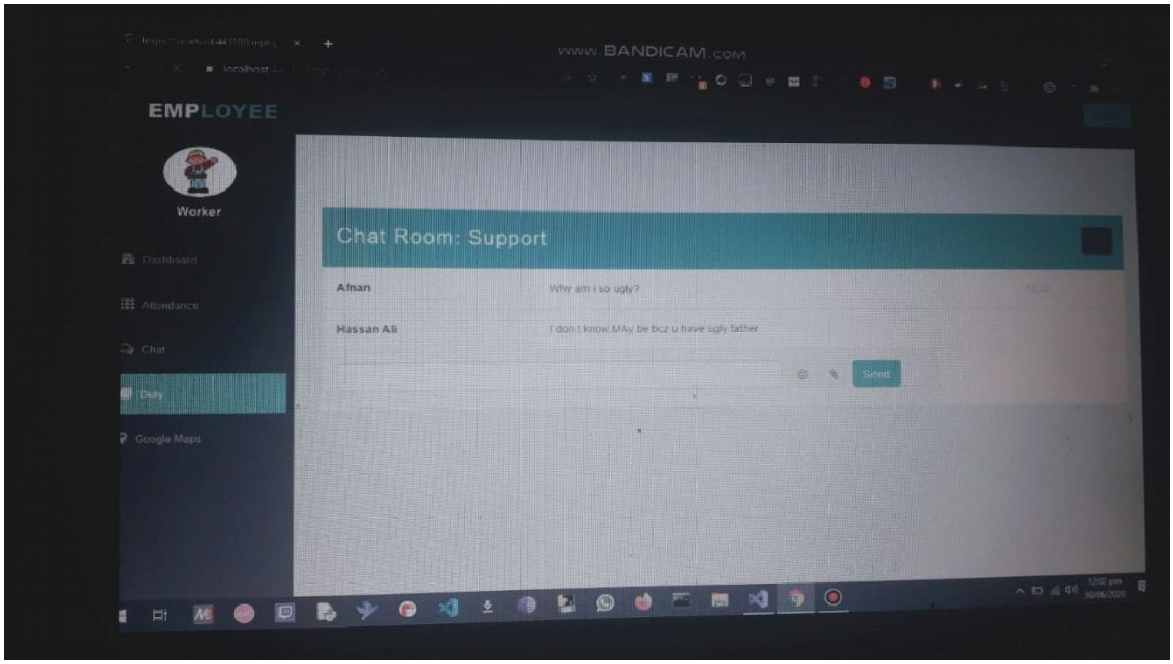
**View Worker Record Interface:**




**View Complains Interface:**



**Chatroom Interface:**




## 5.2 Hardware Interfaces

- Mobile Phones (Android)
- laptop or PC
- Operating System **(**Android/Windows/Os')

## 5.3 Software Interfaces

- Unity 3D
- Android operating system
- Firebase/XAMPP
- Android Software Development Kit

# 5- Implementation

## 5.2 Algorithm:

**Admin:**

- Can login.
- Can add, edit, delete, view workers.
- Can receive records from workers and save into databases.
- Can give complaints from users.
- Can get alerts to users.
- Admin communicate with workers through chat room.
- Can see the location of any worker.

**Worker:**

- Can Signup.
- Can Login.
- Can accept/reject user emergency requests.
- Can visit their portal profile on their website.
- Can provide information about current emergency situations to the admin.
- Worker's have a unique profile.
- Provides services in emergency situations only.

**User:**

- Can Signup.
- Can login.
- Can Visit Website
- Can complain.
- Can press the Emergency Button.
- Can view tips of emergency situations.
- Can get alerts from Admin.
- Can provide location in emergency situation

# chapter 6: Testing and Evaluation

## System testing

Once the system has been successfully developed, testing has to be performed to ensure that the system works as intended. This is also to check that the system meets the requirements stated earlier. Besides that, system testing will help in finding the errors that may be hidden from the user. There are few types of testing which include the unit testing, functional testing, and integration testing. The testing must be completed before it is being deployed for a user to use.

**1: Registration(Signup):**

1- Test Case Objective: To ensure the Registration form is working correctly

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify user registration after click on the 'Registration' button on registration form with correct input data | Username: Afnan Arshad Email: afnan@gmail.com CNIC: 13302-3356096-3 Password: 123 | Successfully registered and will show Home screen | Pass |
| | | | | |

2- Test Case Objective: To verify that if the player is already registered, then system should not registered them to same account

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check user cannot register an email that is already registered | Username: Saad Farooq Email: afnan@gmail.com CNIC: 13302-3356096-3 Password: 123 | The System will not registered player through already existing email | Pass |
| | | | | |

3-Test Case Objective: To verify that system is showing error message for invalid email and password

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether system generates error message for invalid email and password | Email: afnan<br>Password: 126 | The System generates error messages | Pass |

4 Test Case Objective: To verify that system is showing error message for empty email and password fields

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether system generates error message for empty email or password fields after click on “Registration “ button | Username:<br>Afnan Arshad<br>Email:<br>CNIC:<br>13302-3356096-3<br>Password: | TThe System generates error messages. | Pass |

**2: Login:**

1- Test Case Objective: To ensure the login form is working correctly.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify Admin Worker's and user login after click on the ‘Login ’ button on login form with correct input data | Email:<br>Hassan@gmail.com<br>Password:<br>123 | Successfully login and will show Home screen | Pass |
| | | | | |

2- Test Case Objective: To verify that system is showing error message for invalid email and password

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify Admin Worker's and user login after click on the 'Login ' button on login form with invalid input data | Email: Hassan Password: khan | The System generates error messages | Pass |
| | | | | |

3- Test Case Objective: To verify that system is showing error message for unregistered user who are logging.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify user cannot login after click on the 'Login ' button on login form unregister input data | Email: unregister@gmail.com Password: unregister | The System generates error messages | Pass |

4- Test Case Objective: To verify that system is showing error message for empty email and password fields

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether system generates error message for empty email and password fields after click on "Login " button | Email: password: | The System generates error messages | Pass |

**3: Add new Worker's Record:**

1- Test Case Objective: To ensure the Adding new Worker's information form is working properly.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify Workers information after click on the Add Worker Record' button on Adding new Worker's information form with correct input data | Workername: Afnan Arshad Email: afnan@gmail.com Password: 123 | Successfully add new Worker's information and system will show message "Worker record saved Successfully" | Pass |
| 2. | | | | |

2- Test Case Objective: To verify that Adding new Worker's information form with invalid input data in any of the fields generating an error message

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether worker's with invalid input data cannot add worker information after click on 'Add Worker Record' button | Workername: Email: afnan Password: 126 | The System generates error messages | Pass |
| | | | | |

**4: Edit Worker's Record:**

1- Test Case Objective: To ensure the change of worker's information form is working properly.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify Worker's personal information editable after click on the 'Edit Worker's information' button | Email: afnan@gmail.com<br>New Password: 126<br>Old Password: 123 | Successfully changed personal information and system will show messag“Changes saved Successfully" | Pass |
| | | | | |

2- Test Case Objective: To verify that Editing of Worker's record form with invalid input data in any of the fields generating an error message

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether worker with invalid input data cannot change the worker information after click on 'Edit Worker's information ' button | Email:<br>Old Password: khan<br>New Password: khan | The System generates error messages | Pass |
| | | | | |

3- Test Case Objective: To verify that system is generating error message when user entered wrong old password

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether worker with incorrect password cannot able | Email:<br>Saad<br>Old Password: | The System generates error messages | Pass |

| | | | | |
|---|---|---|---|---|
| | to edit personal information. | wrong<br>New Password: ab126 | | |

**5: Delete Worker Record:**

1- Test Case Objective: To ensure the Delete Worker's Record form is working properly.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify Workers record deletes after click on the 'Delete Worker's record' button | Admin Click on 'Delete Worker's record ' button | Successfully deletes worker's record and system will show message “delete Successfully” | Pass |
| | | | | |

2- Test Case Objective: To verify that the delete worker's record option is showing to only admin.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether only Admin can see the delete worker record option. | Admin must login | The system showing delete worker record option to only login Admin | Pass |
| | | | | |

**6: View Worker Record:**

1- Test Case Objective: To ensure the View Worker's Record form is working properly.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify Workers record show after click on the 'View Worker's record ' button | Admin Click on 'View Worker's record ' button | System successfully shows the Worker's record. | Pass |
| | | | | |

2- Test Case Objective: To verify that the view worker's record option is showing to only admin.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether only Admin can see view worker record option | Admin must login | The system showing view worker record option to only login Admin. | Pass |
| | | | | |

**7: Complains:**

1- Test Case Objective: To ensure that the complaint form is working correctly.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify User complain send after click on the 'complain ' button | User Click on complaint button | Successfully send user complain and system will show message “complain send Successfully” | Pass |

| | | | | |
|---|---|---|---|---|
| | | | | |

2- Test Case Objective: To verify that the send complaint option is showing to only login users.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether only login user can see send complain option | User must login | The system showing send complain option to only login user | Pass |
| 2. | | | | |

3- Test Case Objective: To verify that the Admin receives user complaints.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify users send complaints and Admin receives User complains. | Admin must login | Admin receives user complaints. | Pass |
| | | | | |

**8: View Emergency Tips:**

1- Test Case Objective: To ensure that the Emergency Tips form is working correctly.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1. | Verify emergency tips form open after click on the 'View Emergency Tips' button | User Click on 'View Emergency Tips' button | Successfully open Emergency Tips Form. | Pass |

2- Test Case Objective: To verify that the send view emergency tips form is showing to only login users.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Check whether only login user can view emergency tips form | User must login | The system showing view emergency tips form to only login user | Pass |

**9: Request Accept or Reject:**

1- Test Case Objective: To ensure that the user request form is working correctly.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify User request send after click on the 'send request ' button | User Click on 'send request ' button | Successfully send user request and system will show message “Request send Successfully” | Pass |

2- Test Case Objective: To verify that the Worker receives a user request.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1. | Verify user send request and Worker's receives User request. | Workers must login | Admin receives user requests. | Pass |

3- Test Case Objective: To ensure that the Worker accepts or rejects user requests.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify Workers receive a User request and then accept it or not. | Workers must login | Admin accept or reject user requests. | Pass |

**10: Logout:**

1 Test Case Objective: To ensure the logout is working correctly

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|
| 1. | Verify user logout after click on the ‘ logout ’ button | Click on logout button | Successfully logout and will show login screen | Pass |
| | | | | |

2- Test Case Objective: To verify that the logout option is showing to only login users.

| No. | Test case/Test script | Attribute and value | Expected result | Result |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1. | Check whether only login user can see logout option | User must login | The system showing logout option to only login user | Pass |
| | | | | |

# Chapter 7: Conclusion and Future Work

## 7.1 Conclusion

Every second and minute, a lot of incidents happen in the whole world.Many people don't know what they do in emergency situations. Also in emergency situations, many People don't know how they provide precise information about location or incident. So to solve this problem, **Leading Emergency Service Rescue 1122 Web and Android application** is a prototype developed by the students of Department of Computer Science COMSATS University Islamabad, Abbottabad Campus. This application is used in emergency situations, in which People provide precise information about location. So through this App and Website, they will provide their location to one of given departments (**Ambulance service, Fire service, natural disaster service**).The app will have the capability to release on play/app store once all the modules are completed and can make it for people's used in all over Pakistan.

- **Android app** for users will be developed where they can give their location to Service groups in emergency situations. They have to select the department they are seeking to avail service and provide location. The nearby service group gets the notification and response as fast as possible. Workers profile is developed in the app.
- **Web application** will be developed for admin. Users can also visit the site. Workers from three different departments have their profiles in the android app . They can get all the working schedules there. Admin can communicate with all workers. All departments are connected with each other through chat rooms.

## 7.2 Future Work

We have developed this system for emergency situations, in which People provide precise information about incident location. So through this App and Website, they will provide their

location to one of given departments (**Ambulance service, Fire service, natural disaster service**).The app will have the capability to release on play/app store once all the modules are completed and can make it for people's used in all over Pakistan.

# Chapter 8: References


Android connection with MySQL using php.
https://www.tutorialspoint.com/android/android_php_mysql.htm
Java MySQL connection.
https://www.javatpoint.com/example-to-connect-to-the-mysql-database
Create Android App and get other helping materials links:
https://www.youtube.com/watch?v=EOfCEhWq8sg
https://www.youtube.com/watch?v=roDz8mMvbIg&list=PLknSwrodgQ72X4sKpzf5vT8kY80HKcUSe
Save image to server using php files.
https://www.codexworld.com/how-to/save-image-from-url-using-php/


**********

**END**