WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION SYSTEM

WEB-BASED ACCIDENT MANAGEMENT SYSTEM FOR WOLKITE CITY POLICE

BY

NAME	ID NO
EBISA EDESA…….	CIR/142/11
DERARO BIREYE	CIR/137/11
ATINAF LAMESSA	CIR/117/11

PROJECT ADVISOR: MR. FASIL YERO

JUN 14, 2022

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION SYSTEM

WEB-BASED ACCIDENT MANAGEMENT SYSTEM FOR WOLKITE CITY POLICE

SUBMITTED TO DEPARTMENT OF INFORMATION SYSTEM

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE IN INFORMATION SYSTEM

BY

NAME	ID NO
EBISA EDESSA	CIR/142/11
DERARO BIREYE	CIR /137/11
ATINAF LAMESSA	CIR /117/11

PROJECT ADVISOR: MR. FASIL YERO

Wolkite University, Wolkite, Ethiopia

JUN 14, 2022

DECLARATION

This is to declare that this project work done under Mr. Fasil Yero's supervision and having the title Accident Management System for Wolkite city police is the sole contribution of Ebisa Edessa, Deraro Biriye, and Atinaf Lamessa.

No part of the project work has been reproduced illegally (copy and paste) which can be considered Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if a violation of this declaration is proven.

Date: _____

Group Members:

Full Name	Signature
1._____	_____
2._____	_____
3._____	_____

APPROVAL FORM

This is to confirm that the project report entitled Web-based Accident Management System for Wolkite city Police submitted to Wolkite University, College of Computing and Informatics, Department of Information systems by Ebisa Edessa, Atinaf Lamessa, and Deraro Biriye is approved for submission.

Advisor Name	Signature	Date
Department Head Name	Signature	Date
Examiner 1 Name	Signature	Date
Examiner 2 Name	Signature	Date
Examiner 3 Name	Signature	Date

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to several individuals for supporting us throughout our Industrial Project. First, we would like to thank God for helping us in many ways. Next, we would like to express our gratitude to our Advisor **Mr. Fasil Yero** for his enthusiasm, patience, insightful comments, helpful information, and practical advice that helped us tremendously at all times throughout our project. We also express our gratitude to our Instructors who support us in different ways by sharing their precious idea and for our university to accept us into the graduate program. Lastly, all those who have contributed to our Industrial Project have a special place in us.

Table of Contents

LIST OF FIGURE

LIST OF TABLES

LIST OF ACRONOMY

AMS…….Accident Management System

ASP……. Active Server Page

BR……..…Business Rule

CSS…….. Cascading Style Sheet

CPU……. Central Processing Unit

GUI…….. Graphical User Interface

HTML…….. Hypertext Mark-up Language

MD5……. Message-Digest Algorithm

MS WORD 2013……..Microsoft Word 2013

MVC…….Model View Controller

OOSAD…….…...Object-Oriented System Analysis and Design

MYSQL……..…. My Structured Query Language

PHP…….. Hypertext Pre-processor

SDLC……....System Development Life Cycle

SNNPR……..…Southern Nation Nationalities Peoples Region

SQL……..….Structured query Language

UCID…….....Use case Identifier

UML……..Unified Modeling Language

WKCP……..… Wolkite City Police

WKCAMS…….……Wolkite city Accident management System

WKU…….. ...Wolkite University

XAMP……..…. Software Stack Windows, Apache, MySQL, PHP

ABSTRACT

The web based accident management system solves and minimizes so many problems on the organization. The system is with the client/server architect configuration. This means that there is a central application database to store data. This system uses SQL database engine to manage the data and it has the front end with the web programming language PHP. This system help for the employee of Wolkite city police to communicate and also access their data easily. In general the system enables the transactions to be easier and faster. When an accident occurs in a society, the society informs the occurrence and other reliable information about the accident for the field officer. Then, the field officer informs the dispatcher by sending message on the form available on the website to detect the accident. After the dispatcher retrieves the detail information from the message sent by the field officer, he/she allocates resource.

CHAPTER ONE

1. INTRODUCTION

This project is developed about a Web-based Accident management system (AMS) for Wolkite city policy (WKCP). An accident management system is a system that used to manage the unexpected event, typically sudden in nature and associated with injury, loss, or harm. Accidents are a common feature of the human experience and result in injury or permanent disability to large numbers of people worldwide every year. Many accidents also involve damage to or loss of property.

Wolkite city is one of the institutions of accident management system to give service to protect people from the accident. Those are protected by accident management stations such as police and society. The accidents that can be reported can be either natural or human-made. Some of these are killing or murder, flood, harassment, etc.

This developed project focused to develop a web-based accident management system. The proposed system applies to accident management institutions all across the country and especially looks at the subject of the accident management system of Wolkite city. It is well understood that accident management system depend on a highly responsive backbone to the information management system. The efficiency of the accident control and the effectiveness with which it deals with accidents depends on what quality of information it can drive or receive from where the accident occurs and how fast it can have responded to it. Therefore Accident management system give service to society and Wolkite city police in efficiency manner.

1.1 Background about the organization

Wolkite city police station is one of the institutions of accident management station that was established 1934 E.C to give services to protect Wolkite city from the accident. Wolkite city police station is located in the southern nation nationalities regional state (SNNPR), in the Gurage zone, 187 kilometers from Addis Abeba. Wolkite city police station has responsible for receiving the accident sound and having a response to the received accident sound and storing the nature of the accident, the location details, the sequence of the accident, information on victims.

Wolkite city police station has vision of being fulfill the interest of society and increase communication with a developed country. Also the organization has the following missions: - To transfer useful and modern technology for society; to give suitable and fast service based on society's request; to increase the acceptance of the police stations; to increase the employee by supporting the technology.

1.2 Background about the project

Now a day the introduction of new information systems is increasing at an alarming rate to bring radical change to the existing manual system, improve the performance of other systems and solve difficulties. Though this technology is new to our country Ethiopia and most other developing countries, it was/is exploited well in developed countries, like Europe, India, and America. This project is developed to advocate for the need of Wolkite city police station to use a facilitated computerized and web-based accident management system. Because even if there is no organized accident management station in the city, the police station that gives this service uses a manual way of information gathering and documenting as well as there is no optimized way to facilitate the service. So we are initiated to develop a web-based automated system. As a result, the team member believed that the user will have the expected satisfaction of the service provided by the accident management system of the Wolkite city police station. Therefore the team member recommends to Wolkite city police station from the manual system to the computerized and web-based accident management system to facilitate the service that they give to society.

1.3 Statement of the problem

In Wolkite city police station accident management apply a manual way of implementing tasks. The police station is the following problem:-

- Difficult to getting information about when and where the accident occurred.
- Difficult to know the previous accident occurred and difficult to identify cause of the accident.
- The accident file control mechanism is very tedious and complicated.
- Customer queries can be difficult to respond to as information is stored in different places and may even require that you find the right person before being able to respond.

- Difficult to generate a report.
- There is a collection of papers to record information: offices like transport, police station and, related offices use paper to record information. This manual recording method exposes information to intrusion, and also retrieving information will take much time.
- Reporting and checking that data is robust can be timely and expensive.
- The officer lost the written information about the accident also lost accident information.
- It takes time to search and record large number of files about accidents.
- Someone coming from another place suddenly dies in an accident the person has no ID or something about himself tells, only thing is to write a report about them.
- Unavailability of the employees in the office when someone wants to report a accident using the office phone: as there is a flow of accident at any time, an accident will be probable. Because of this, there is no way to report accidents that happened at night time. As a result, death and physical injuries may happen, and the criminals may escape.
- Difficult in conducting consistent reports because the record is documented manually and requires much time and human power to search and get wanted information (file).
- There is no fast and efficient way of sharing critical information about the accident.
- Data redundancy and inconsistency
- The current system is very time consume to retrieve accident information.
- Also, there is a security problem as we mentioned before, the existing system is a manual system in which documents are stored in packed paper files so that the file is highly exposed to damage and can be stolen by any other unauthorized person or group.

1.4 Objectives

1.4.1 General Objective

The general objective of this project is to develop a web-based accident management system for Wolkite city police station.

1.4.2 Specific Objective

The specific objective of the accident management system in Wolkite city police station include:-

- Set a plan to achieve the general objective
- To clearly identify and gather user requirements for the proposed system.
- To clearly identify the existing problems
- To make document analysis about traffic information management system.
- To Analyzing the problem of the existing system
- To perform a requirement analysis to find out the system functional and nonfunctional requirements.
- To design the new proposed system
- To Implementation of the new system

1.5 Feasibility Analysis

Most of the information system projects have budgets and deadlines, and analysis of factors for the feasibility forms the business case (analysis of assumptions like resource availability and potential problem and system costs and benefits) that justifies the expenditure on the project.

1.5.1 Technical feasibility

The developed system will be easily maintained/ retained and repaired without requiring high Experts or Technical supports because the system will be installed in adaptable technology's and the employees of the organization have some knowledge about technology by providing training and Help how to use the system and can use the system easily. So the system we developed is technically feasible.

The required technologies such that PHP, MySQL, XAMPP, JAVASCRIPT MS Word 2013 which we will use to work for this project are available, familiar/Popular, free, widely used, maintainable, repair and are compatible with the hardware that we will use. The environment that we deployed the system is available, thus our system is technically feasible.

1.5.2 Economic Feasibility

As part of this, the costs and benefits associated with the developed system are compared and the project is economically feasible only if tangible or intangible benefits outweigh costs. When we analyze our developed system economically, it is cost-effective. In economic analysis, the

procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. So this system can estimate or minimize the cost of materials (paper, pen, pencil) and Cost of employees' time relatively. A small amount of budget is going to be used to build the developed system. . Economically, the system will profit the organization by providing a better means of managing accident management system for Wolkite city police. Generally, the system that we developed Wolkite city police station accident management brought several tangible and intangible benefits.

1.5.3 Operation feasibility

The system we developed will provide accurate, active, secured service and on every page, all the operable components such as links, buttons and so on will be easy, understandable to use by the users meanwhile these components shall operate, would be functional for every required task so, the system is operationally feasible. This system solve the problems of the existing system so the system is operationally feasible.

1.5.4 Political feasibility

The system to be developed does not conflict with any government directly or indirectly, because this web-based accident management gives service for people effectively and efficiently.

1.5.5 Legal and contractual feasibility

This is the process of assessing potential legal and contractual ramifications due to the construction of a system. When we first think to plan and select a system, we have to consider the law, financial reporting, and standard, as well as current contractual obligations. Every activity will be controlled or performed by the policy that governs the Wolkite city police station.

1.6 Scope and Limitation of project

1.6.1 Scope of project

This project developed a web-based system for the Wolkite city police station accident management system. The developed project is:-

- The system can easily retrieve the accident information.
- The system can accurate way of recording and storing information in the database.
- Post information.

- The system allows manages accident files using the web-based system in a manner that is more secure than the previous system.
- The system can post news for the society to make them informed by some additional information.
- Reports cannot be visible to all society.
- The system can search, Insert, delete and update the accident information when it is needed. In the proposed system they can update easily because all data are in the database so we can update easily and in a short time.
- Presence of centralized documented and organized record.
- Enabling the workers of the system to get reliable information where and when occurred as well as the type and level of the accident to give reliable to response on detecting the accident.
- They can use either the Amharic or English language.
- The total number of each accident can be seen easily.
- They can prepare reports easily.
- The system easily manages accident report because everything should be done web-based.

1.6.2 Limitation of the project

There are many constraints within our developed system that limit their effectiveness of performance. Our system is limited only in the process of the web-based accident management system of the Wolkite city police. But we don't include about fire accidents, traffic car accident and insurance payments management system.

1.7 Significance of the project

Web-based accident management is a system that is customized to the Wolkite city police station. Generally, the significance of the new system includes:-

- The most important feature of the developed system is that it is an accurate, easy, and efficient system to detect accidents as well as record accidental information such as the date and time when the accident occurs, location, and level of the accident.
- Also, the cause and effect of the accident are carefully recorded and documented.

- It enabled searching the required information by using keys
- Report generation is done when necessary or required at any time.
- Avoidance of document missing
- Used to record data on the database.
- Used to manage accidents information easily.
- Learn society about the accident and they can protect themself from the accident.
- Ensure data accuracy.
- Used to make recording and retrieving very fast.
- Generally, the purpose of this accident management system is to solve all the problems of the organization and to satisfy the requirement of the people.

1. 8 Beneficiaries of the project

1.8.1 Benefits to the police station

- Avoid improper resource consumption like paper, pens.
- Avoiding data loss because of improper data storage
- Neat investigation files in the station can be transferred from generation to generation
- Enhance security mechanisms to protect crime records

1.8.2 Benefits to Citizens

- Multiple channels to access services from police
- Simplified process for reporting the accident.
- Can view posted information's by the station anywhere at any time
- Faster and assured response from police to an accident.

1.8.3 Benefits to Police Department

- Enhanced tools for investigation
- Facilitates fast and efficient retrieval of data
- Can post necessary information to society easily.
- The reduced workload of the police station back-office activities such as preparation of reports and station records management
- Can easily control the system.

1.8.4 Benefits to Team member

- The team member can get additional knowledge.
- Support the team member for the next working

1.9 Methodology

1.9.1 Data gathering Methodology

The requirement of the system is gathered using primary data collecting techniques. These are listed below.

1.9.1 .1 Interview

By attending at Wolkite city accident management system police station we asked/contact the representative of the organization and then exchange some ideas about their existing, how it has been working and the structure of this organization. As a general, we gathered enough data in order to prepare our project.

1.9.1.2 Observation

To gather relevant information the project team observed how the current system works.

1.9.1.3 Document Analysis

Reading the document available in the organization and document indirectly related to Wolkite city Accident management.

1.9.2 The system analysis and design approaches

The goal of this section is to provide a basic overview of the system that we developed. The system analysis and design approach for this project used object-oriented system analysis & design (OOSAD). Because:-

- It provides code and function reuse through the concepts of inheritance, polymorphism, encapsulation, modularity, coupling, and cohesion.
- To design the system the project team has to choose Object-Oriented Modeling techniques and Unified modeling language (UML) tools.
- Understanding the structure is easy because object-oriented modeling and tools are used to represent real-world entities.
- Modification of the object implementation is easy because objects are loosely coupled.

1.9.3 The system development model

There are various software development life cycle models (SDLC) defined and designed which are followed during the software development process. These models are also referred to as “Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development. But the developed system follows the Iterative Model. Because the Iterative Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete and it can be back to the first phase if it can be error occurs. In this Iterative model, the phases do not overlap. In "The Iterative” approach, the whole process of software development is divided into separate phases and, the outcome of one phase acts as the input for the next phase sequentially and back to the previous phase.

1.9.4 System testing Methodology

Testing is a trial experience in which the deliverables of the project are checked with acceptable standards in the project. We used unit testing, system testing to test the correctness of each module and the compiled program.

1.9.4.1 User Acceptance Testing

The project team has tried to test the system for checking user acceptance that justifies that major components of the system are working as per the requirements of users.

1.9.4.2 Unit testing

In unit testing, each of the data in the database is tested through form or report with specific data. This test helps to trap output errors that occur in the form or report. Each module is tested alone in an attempt to discover any error in its code.

- Register society accident request
- Register accident report
- Insert emergency report
- Display people, unknown people die in an accident.
- Generate report.

1.9.4.3 Performance testing

The system performs well under unfavorable conditions and stresses its functioning as it functions in a normal state for a certain time and may fail if the stress is present for a long time.

1.9.4.4 Security testing

The system has security protection by using authentication. Users of the system will be authorized by the system administrator and get their user name and password to enter into the system. Unauthorized access will be detected when the user wants to access the system. This is done by authenticating the user; a user with a user name and password can only enter into the system. This security action is done in the login form. Another method used to secure the system is by grouping users by privileges. In the system, there are two types of categories; system administrator and other members of the system. Every user of the system will enter the system according to their privileges and access different menus when they enter the system. This security method is tested and the user cannot enter another person's privileges and access data. So the system is secured.

1.9.5 Development Tools and Technologies

1.9.5.1 Frontend Technologies

- JavaScript which is used to code client-side code
- CSS for styling the user interface
- HTML markup language for creating a web page
- Bootstrap to develop a responsive website

1.9.5.2 Back end Technologies

- MySQL Database which is used as storage.
- Apache server let us use client computer as client and server.
- Xammp-Server which function as a server in one computer.
- PHP which used for storing data into a database or server-side code.

1.9.5.3 Documentation and Modeling Tools

- **Microsoft word 2013**: - for documentation purpose
- **Edraw-max: -** for designing of UML diagrams

- **Notepad++**: - for encoding scripting languages

1.9.5.4 Deployment Environment

- **Personal computer and Desktop: -** to write, deploy and store the collected information.
- **Server**: -for deployment purpose

1.10. Document Organization

Chapter One: - This chapter contains a range of parts that are concerned with the developed project such as the introduction of the developed system, background of the organization, statement of the problem, the objective of the project, the feasibility study, scope and limitation, significant, beneficiary, methodology, budget, and schedule.

Chapter Two: - This chapter is mainly concerned with studying the existing system by elaborating its existing system, business rules, main functions, forms and other documents, and bottlenecks.

Chapter Three: - This chapter deals with the proposed system by deeply explaining its functional and non-functional requirements.

Chapter Four: - This chapter deals with the system model, object model, and dynamic model of the proposed system.

Chapter Five: - This chapter deals with the design goals of the proposed system, the proposed system's architecture (including subsystem decomposition, hardware, and software mapping), algorithm design, and user interface of the proposed system.

CHAPTER TWO

2. DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction

In this chapter, we studied the existing system deeply, since it is necessary to know the existing working system of the office to develop a better system. When we studied the existing system we gave emphasis to here under listed questions:

- How the existing system is working?
- What kind of method do they use to handle accident data?
- In what way the police officer is handling accident information managed?
- What is the business rule they use?

After studying the existing system, we also determined the requirement or the feature that must be included in the proposed system. Furthermore, by analyzing the current system, we could also estimate how the developed system solves the setbacks of the existing system.

2.2 Users in the existing system

The main players in the existing system include the following: -

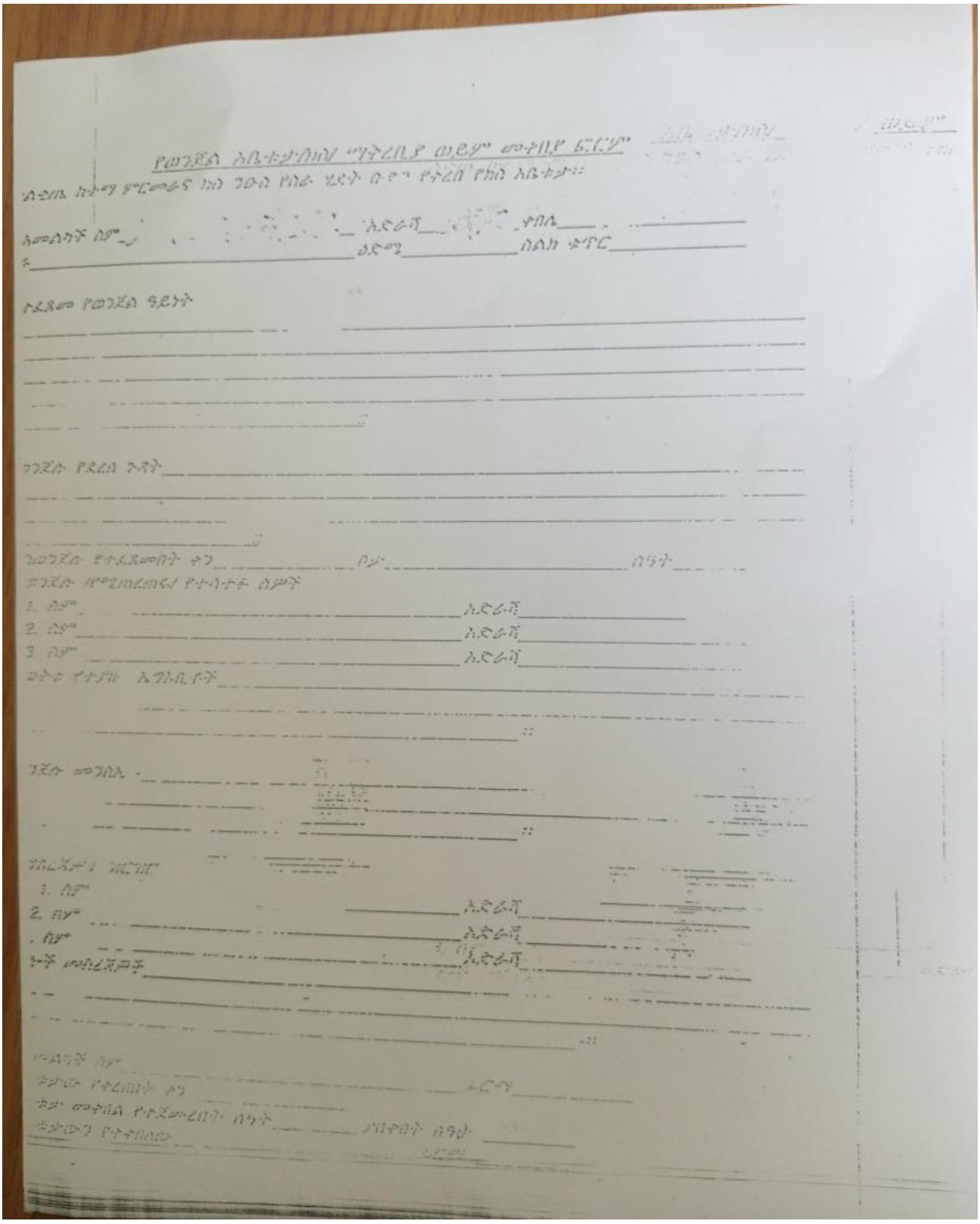
- The existing system enables the **Field officer:-**
  - If the field officer exists in an accident place report accident.
  - Call to police and he/she need additional resource.
  - Directly work with society.
  - Collect all necessary information in his/her notebook.
- The existing system enables the **society**:-
  - If society see the accident happen directly to report the accident
  - He/she report in two way either by telephone or physically going to the police.
  - Send their comment either in oral or written form.
- The existing system enable the **Dispatcher:-**
  - The dispatcher accepts the emergency accident report automatically assign resource and notify the accident control team.
  - View previous accident report
  - Update previous accident report.

- The existing system enables the **Accident controlling team:**
  - Accident report the dispatcher sends the accident report to the go-to place and prepares the report.
  - Prepare the report.
  - Learn how to protect from an accident for society.

2.3 Major functionality of the current system

- **Emergency Accident Report: -** In the existing system the emergency accident is reported in two ways. First, the society calls telephone to Police. Most time the telephone is busy because the police have only one telephone. Second, society comes to police physically.
- **Generate Report: -** The major time to generate reports is classified in daily, weekly, monthly, in three months, in six months, in nine-month, and in a year. All reports generated are very tedious and complex because all thing is done in manual so it is difficult to prepare report, especially the month, three months, six months, year report are more difficult, and additional tasks to the police department to search the file.
- **Store the accident file:** - The existing system stores the accident information manually. This method is not secured, also has no backup. The accident file is only in one document for this reason, if the paper is damaged accident files can also be damaged.

2.4 Forms and Other Documents of the Existing Systems



የወንጀል አቤቱታስለሀ ማጥረቢያ ወይም መተበየ ፎርም

Figure 2.1 forms of existing system

2.5 Drawbacks of the Existing System

In the existing Accident management system, there are several drawbacks that actually cause a problem in the process of operating this system. These problems are listed below:

- **Performance Problem**
  - The response time for performing every process is very slow.
  - Documents are stored physically this increases the space complexity.
  - There is always delay time in information search and retrieval
- **Response time problem**
  - Delay in producing different reports.
  - Delay in identifying employee status.
  - Inaccuracy in updating important information because of manual processing.
  - No centralized database
  - No computerized data gathering, storing, processing, and disseminating facilities
  - To provide information, it may take inevitably long time by searching the necessary facts to which it is intended for
  - Less credible information generated by the reports – The information presented on the report is inconsistent meaning the information is gathered by referring to different documents which are stored in manual or paper form. Therefore, as time goes by, the information on the paper may get lost. Hence, the information is not as trustworthy.
- **Stored data Problem:**
  - Data is stored repeatedly in different files formats – The same information is stored in many copies repeatedly in different forms.
  - Data is not secured. Due to this, some secret information is opened for unauthorized users or agents.
- **Economics Problem:**
  - The economic problem is mainly concerned with cost control and profit incremental issues.
  - Manual handling of data is expensive as compared to an automated system. In general, the cost in terms of time is very high.

- As the business entry increases, the existing manual system will incur costs to handle those requirements.
- As the number of controlling teams to handle the task of manual processing increases, the accident management system will spend a lot of money on its staff.

Efficiency problem

- Waste of time.
- Data is redundantly input or copied.
- Data is redundantly processed.
- Information is redundantly generated

2.6 Business rule of the current system

A business rule of the existing system is successfully an operating standard or policy that will explain the existing system of Wolkite city police.

The existing system has many business rules or principles some of them are:

BR1: New accident reports (first information reports) have to be received and organized by the accident control team.

BR2: Emergency accidents should get a fast response as much as possible.

BR3: accident file should be investigated by the Accident control team.

BR4: The accident should be happened in Wolkite city or around the city, in order to start the investigation process.

BR5: any accidental emergency should be reported in 24 hours

BR6: society had right to report any accident

BR7*:* Dispatcher works to accept report detail information from society (place, location, time, and type of accident)

BR8*:* field officer find a witness to compile a report for dispatcher

BR9*:* dispatcher set order based on the investigation result

CHAPTER THREE

3. PROPOSED SYSTEM

3.1 Introduction

This chapter mainly deals with the description of the proposed system. Through the proposed system, we are going to solve the problems under the existing system. The existing system has its problem and drawbacks. Therefore, the project team tries to develop a system, which solves problems under the existing system.

The proposed system will enable the system to store information on the database. Generally, the new system follows the appropriate system development methodologies as the only option for addressing and solving those problems associated with the current system, and this new system is a web-based application that enables the users to access the services given by the system through the Internet.

3.2. Functional Requirements

The new system will be a networked application that will run on client-server and provide case-to-use user interfaces to the users and window server for the server-side application. The new system will also perform a record management function that the system should record all the modifications on the record management system. All in all the functionalities that will be provided by the system are the following.

- **User account management**

The Dispatcher manage user Account according to the followings:-

- **Create User Account:** the system Dispatcher creates an Account for new users.
- **Update User Profile:** The Dispatcher updates the user profile as needed
- **Delete User Account**: The Dispatcher Delete the user account when it is needed
- **View user Account:** The Dispatcher view the user account information as needed.
- The system allows the dispatcher to post information
- The system allows the dispatcher to view report
- The system allows manage report
- The system allows the dispatcher to open request

- **Accident control team**
- The Accident control team manage files in the following manner:
  - **Add new file:** the system Accident controlling team add new files
  - **Delete file:** The Accident controlling team delete file as needed
  - **Update file:** The Accident controlling team update file as needed
  - **View file:** The Accident controlling team view file as needed
  - The system allows generate report
  - The system allows generate view request
- **Retrieve accident files:** the system can easily retrieve the accident information in easily the accident date, accident weight, and accident type name. Then the system check that accident type, and processes if it is valid displays the information.
- **Manage accident reports:** The system easily manages accident reports because everything should be done web-based. Accident report Date, accident Time, Information Type, Place of an accident, Address of accident, Name of Police, Received Time, Information Received take the system as input.
- **Emergency accident online report:** The system able allows to emergency accidents online. Without coming to the police station office they can able to report online by using the proposed system. The society and the field officer report an emergency accident. They fill in all the necessary information if enter the data correctly the system report to the dispatcher and gets the acknowledgment.
- **Posting an unknown person who died in an accident: -** This is very important for our system because if someone comes from somewhere else and then suddenly dies in an accident and the Person has no id or anything about himself then the system will be able to post this unknown person.
- **Online and offline storage:**-The system allows able to access online storage and offline storage.
- **Generate Report:-** The system allows generate accident reports based on the user's needs
- **Posting news:** the system allows post news for the society.

3.3 Non-Functional Requirements

Non-functional requirements describe user-visible aspects of the system that are not directly related to the functional behavior of the system. These requirements do not directly affect the

Performance of the system but are nonetheless important. They are concerned with security, performance, internationalization, usability, maintainability, reliability, modifiability, efficiency, portability across operating systems, testability, and understandability. Generally, the non-functional requirements of the system are presented the followings:-

3.3.1 User Interface and Human Factors

The System we developed supports both novae and expert users which means a moderate system to support both levels of knowledge. We will do such a thing by providing buttons, menus, and another interactive button in a user-friendly manner. As a human factor, there are many different persons which have different feeling and attitude, so we will develop our systems user interface by selecting the best color which many will people supports and use. This makes our system very interesting because users will use the system without frustration for a long period.

We considered the following list when designing our interface.

- Keep the interface simple
- Be purpose full in page layout
- Strategically use color and texture

3.3.2 Hardware consideration

The system that to be developed should run on the existing standard computers. And the system will be portable that can run on any type of computer that is currently available in the market.

3.3.3 Security Issues

The system is much secured and only authorized persons can access the system by using their username and password. The password with which users are provided will protect them from potential threats, which could jeopardize their responsibility. The system must be secure because the framework we used makes the system itself secure. MVC is a framework in asp.net (Model View Controller), users first interact with a view and no direct communication with the model but between them, there was a controller which is controlling unwanted or invalid requests to the model. Other, we used the MD5 hash algorithm to make passwords encrypted and no one sees or accesses another user's password. These all mechanisms make our system secure.

3.3.4 Performance Consideration

The system performance will be highly responsive because of query optimizing, normalizing tables, and making client-side validation which is very important to check user's data validation before navigating to model or database. Also, ASP.NET has its web optimization for enhancing performance.

3.3.5 Error Handling and Validation

Exception handling is the process of responding to the occurrence of exceptional conditions requiring special processing. Exception handling is important in any application. In ASP.NET we can handle exceptions in the following two ways:

- Try-catch-finally block at method level
- Using Application Error

3.3.6 Quality Issues

MVC is a software architecture pattern that follows the separation of concerns method. In this model .Net, applications are divided into three interconnected parts which are called Model, View, and Controller. So each of these parts can be developed, tested in relative isolation and, also combined to create a very robust application. Generally, Our System quality will be measured through: -

- **Reliability**: The proposed system will minimize crashes during its runtime since more than one user could use the system simultaneously.
- **Availability**: The system shall be available for 24 hours and 7 days a week.

3.3.7 Backup and Recovery

The system includes a secondary database that contains the copy of the original database used for recovery purposes when a problem occurs on the system's original database.

3.3.8 Physical Environment

Since our system is web-based, it is deployed on the server. The server shall be placed at the Wolkite city police data center. Since the data center has moderate temperature and humidity and a fire controller.

3.3.9 Resource Issues

The system needs a resource that has high processor speed and memory mainly for the server. It enables the system to provide the required service efficiently and effectively.

3.3.10 Documentation

Our system have a well-defined document that helps the database maintain the system. We will also prepare a short and precise helping guide for the system users which helps users how to use the system.

CHAPTER FOUR

4. SYSTEM ANALYSIS

4.1 Introduction

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish its purpose. The analysis model contains three models: functional, object, and dynamic models. The functional model can be described by use case diagrams. Class diagrams describe the object model. The dynamic model can also be described in terms of sequence, state chart, and activity diagrams. For the purpose of this project, we have described the analysis model in terms of the functional model and dynamic models using use case, sequence diagrams, Activity diagram, State diagram, and class diagram.

4.2 System Model

System modeling is the process of developing abstract models of a system, so with each model we will present (show) a different view or perspective of our system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML). Models help to understand the functionality of a system.

4.2.1 Use Case Model

The Use Case Model (use case diagram, use case description, use case scenario) is used to define the core elements and processes that make up our system. This Use Case Model captures the functional system components. Because Use Case Models are simple in nature, these Use Case Models are a great way to storyboard flows with users and define the system requirements being modeled and help write the scenarios later used in testing.

Actors	Use Case Under Identified Actor	Use Case ID(UCID)
	Login • Logout	01

Dispatcher	Manage Account ✓ Create Account ✓ Update Account ✓ Delete Account ✓ View user Account	02
	Generate report	03
	Allocate resource	04
	Open request	05
	Post information	06
	Manage report	07
	View report	08
	Manage feedback	09
Accident control team	Login ● Logout	01
	Manage file ✓ Add new file ✓ View accident file ✓ Update file ✓ Delete file	10
	View request	11
	Controlling accident	12
	Identify types of accident	13
	Study cause of accident	14

	Generate report	03
	Post information	06
	View information	15
	Accident detection report	16
Society	Generate report	03
	View information	15
	Report accidents	17
Field officer	Login • Logout	01
	Generate emergency report	18
	Request resources	19
	Generate report	03
	View information	15
	Gather information	20

Table 4.1 Use case Identification

4.2.1.2 Use Case Diagram

A use case diagram is a graphical representation of the interaction among the elements and the system. It describes how a system interacts with outside actors. In this system, we have identified both actors who are anything that interacts with the system and use cases which is the functionality of the system that directly interact with the system.

Actors:

- Dispatcher
- Accident control team
- Field officer
- Society

Use cases: A use case is a software and system engineering term that describes how a user uses a system to accomplish a particular goal. A use case acts as a software modeling technique that defines the features to be implemented and the resolution of any errors that may be encountered.

Use case exists:

- Login
- Allocate resources
- Manage report
- View request
- View information
- Request resource
- Generate emergency report
- Emergency report
- Open request
- Accident detection
- Post information
- Generate report
- Study cause of accident
- Logout

- **Dispatcher**: has the following responsibilities
  - Has direct access to the database
  - Notifying field officer that it provides the report
  - Allocating resources based on the emergency report comes from the field officer
  - Announcing team to detect accident by having allocated resource
  - Managing user account
  - Post information
- **Accident controlling team:** has the following responsibilities
  - Generate accident detection report which includes the process of accident control and effect of the accident.
  - Studying the cause of the accident
  - Control the accidents
  - Identify types of accidents
- **Field officer:** has the following responsibilities
  - Gathering information about the accident from society.
  - Generate emergency report
  - Updating the report if any information is received
- **Society:** has the following responsibilities
  - Informing occurrence and cause of the accident
  - Participating in the detection of the accident
  - Request for support to suffer from the accident
  - Detecting accidents.

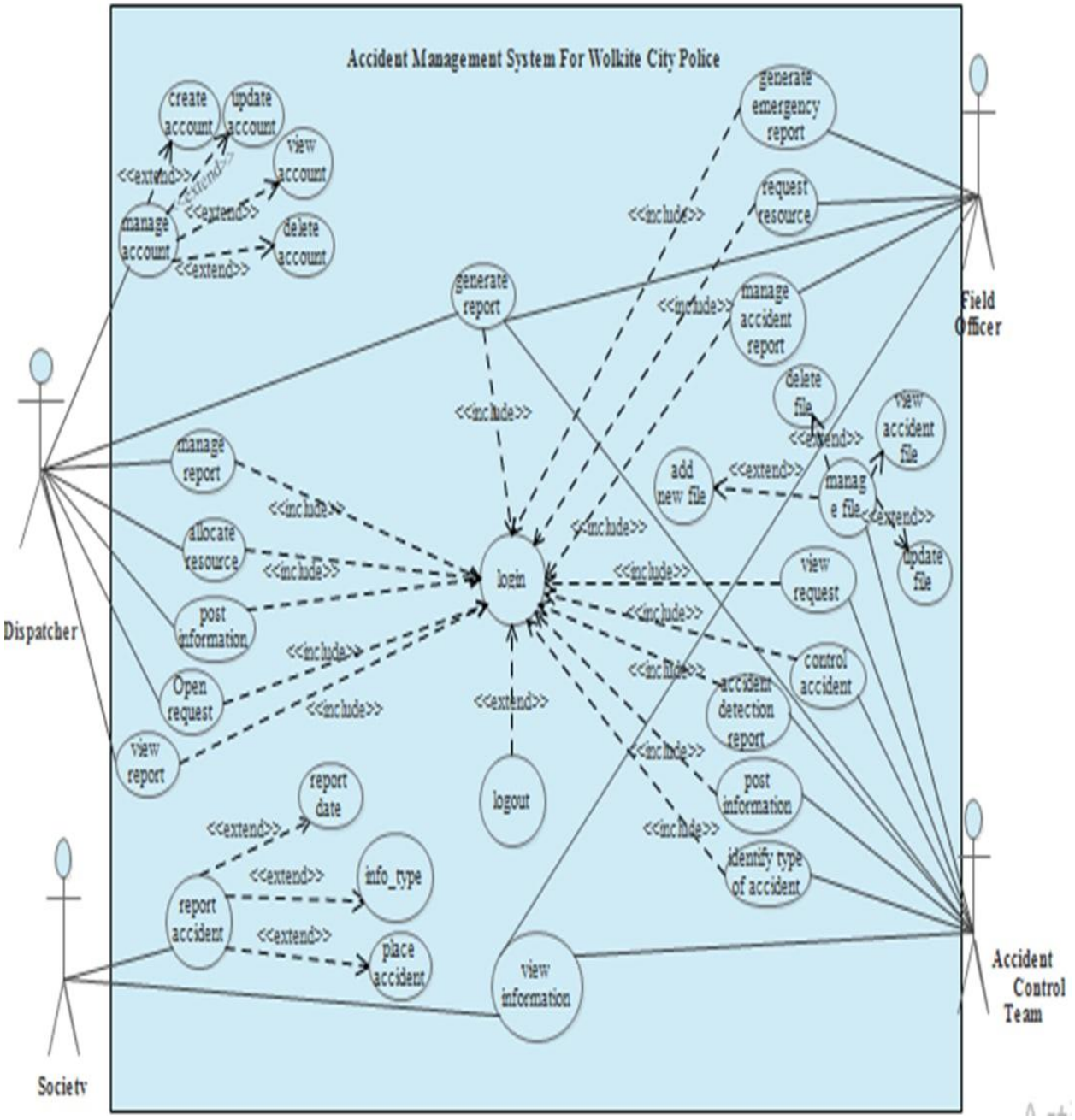


Figure 4.1 Use case diagram

4.2.1.3 Use Case Description

The use case description is used to detail the description of the use case and how the use case works in order to perform user and system functionality. In this project, we discussed a detailed use case description as shown

Use case login

Use case Name	**Login**	
UC_ID:	**UC_01**	
Actor:	Dispatcher, field officer, and accident controlling team, society	
Description:	This use case is used to ensure security for login into the system	
Precondition:	The user must have at least the correct username and password.	
Flow Event:	**Actor action** Step1: user has to open the system	**system response** **step3:** the system will display the login page
	Step2:user clicks on the login button	Step5: system responses to the user username and password form
	Step4: user fill username and password	Step6: system checks his/her username and password if it is correct the system allows the user to access unless it displays a wrong username and password message
	Step7: user clicks on the login button	Step9: use case ends
	Step8:user gets access from the system	
Post condition	The main page will be displayed when the user gets access to its privilege and after finishing his/her work user can logout	
Alternative condition	Step1: If the user enters invalid information in step 4 then display an Error message and return to step 3.	
Exception	condition If the user name does not exist the system stop login	

Table 4.2 use case description of login

Use case description of Adding user

Use case Name	Adding user	
UC_ID:	**UC_02**	
Actor:	Dispatcher	
Description:	It describes how to add a new user account record to the system	
Precondition:	The user should have logged in as a Dispatcher	
Flow Event:	Actor action Step1: the dispatcher clicks on add user button	system response Step2:the system displays the form
	Step3:the dispatcher enters the new user name, account type, and password, and re-enter a password to the system	Step5:the system validates the new user detail
	Step4:the dispatcher submit the news information	Step6:the system saves the user detail into the database
	Step8:the dispatcher tells the category username and password to the user	Step7:the system displays a successful message of adding a new user
		Step9: the system ends
Post condition	The new user information are added and recorded to database	
Alternative condition	if step7 is not successful repeat step3 until it is success	
Exception	if there is the same account with which is to create stop creating	

Table 4.3 use case description of Adding user

Use case description of update user account

Use case Name	Update user account
UC_ID:	**UC_03**
Actor:	Dispatcher, Field officer, and accident controlling team
Description:	It describes how the dispatcher(Field officer or accident controlling team) modify the user database

Precondition:	The user should have logged into the system	
Flow Event:	Actor action Step1: If the Administrator wants to Update an account, then he/she must log in to the system.	system response Step4: the system checks the new account information with the Existing account in the database.
	Step2: the Administrator inserts account _type, username, password, and other useful information.	Step5:the system validates the new user detail
	Step3:the administrator submits the data	Step5: the system saves the new Account to the database.
		Step6: the updating process ends
Post condition	The system modifies the records with the newly entered data.	
Alternative condition	If step5 is not successful repeat step2 until it is success	
Exception	If an account doesn't exist in the system stop updating	

Table 4.4 use case description of update user account

Use case Delete user account

Use case Name	Delete user account	
UC_ID:	**UC_04**	
Actor:	Dispatcher	
Description:	It describes how the dispatcher removes records of a system user in the database	
Precondition:	the user should have logged in as Dispatcher	
Flow Event:	Actor action Step1: if the Dispatcher (administrator) wants to Delete	system response Step3: system displays account type information

	the account, then he/she must log in to the system	
	Step2. the administrator selects the delete account button	Step6: the system deletes the Account from the system.
	Step4: the administrator view the account type and user name of the user	Step7:the use case ends
	Step5: the administrator selects the Delete button.	
Post condition	The system removes the user details from the record	
Alternative condition	if step5 is not successful repeat step2 until it is a success – if the system ends he/she selects no	
Exception	If an account doesn't exist in the system stop deleting	

Table 4.5 use case description of delete user account

Use case updating records

Use case Name	Updating records	
UC_ID:	**UC_05**	
Actor:	Accident controlling team	
Description:	It describes how the dispatcher update records to the database	
Precondition:	the user should have logged in as Dispatcher	
Flow Event:	Actor action Step1: the user opens the system	system response Step4: the system displays accident files stored before
	Step2: the user selects the files button	Step5: the system displays the form based on the selected accident id or number

	Step3: the user selects the update files button after clicking on the n files button	Step8: the system checks the entered information
	Step6: the user fills the displayed form	Step9: if the entered information is correct the system updates the information
	Step7: the user clicks on the update button	Step10: the system displays a successful message
		Step11:the system updates the information
		Step13:the use case ends
Post condition	The system saves the modified record to the database.	
Alternative condition	if step6 is not successful repeat step1 until it is success	
Exception	if the account doesn't exist in the system stop updating records	

Table 4.6 Use case description of updating records

Use case of deleting records

Use case Name	Deleting records	
UC_ID:	**UC_06**	
Actor:	Accident controlling team	
Description:	It is for dropping the content of the database	
Precondition:	The user should have logged in as Dispatcher	
Flow Event:	Actor action Step1: the user login to the system to delete the	system response Step3: the system displays the accident information
	records Step2: the user selects the view button	Step5: the system successfully delete the information based on a selected file id/number

	Step4: the user clicks on the delete button based on the selected file Id	Step6: the use case ends
Post condition	The system removes the recorded information from the database	
Alternative condition	if step5 is not successful repeat step1 until it is success	
Exception	if an account doesn't exist in the system stop deleting records	

Table 4.7 use case description of deleting records

Use case Emergency report

Use case Name	Emergency report	
UC_ID:	**UC_07**	
Actor:	Field officer, society, accident control team	
Description:	It's used to generate a report when an accident happened.	
Precondition:	There must be an accident and collected information about it.	
Flow Event:	Actor action Step1: The user opens the emergency form	system response Step2: system display the emergency report form.
	Step3: fill out the report form.	Step5: the system checks whether the information is correctly filled or not
	Step4:next click the emergency report button	Step6: displays successful message To user.
		Step7: send an emergency report to Dispatcher.
		Step 8:the use case ends
Post condition	The system generates the report and displays a successful message	
Alternative condition	if step6 is not successful repeat step3 until it is success	
Exception	The dispatcher must log in to the system to view the generated report	

Table 4.8 use case description of Emergency report

Use case of Open Request

Use case Name	Open request	
UC_ID:	UC_08	
Actor:	Dispatcher	
Description:	This is used to read the emergency report	
Precondition:	The field officer and society must generate a new emergency report	
Flow Event:	Actor action Step1: the user wants to see the Request and she/he login to the system.	system response Step3: system display request Reports. Step 6:end use case
	Step2: The user selects the request Page.	Step 6:end use case
	Step4: user sees the request.	
	Step5: Notify the accident controlling team of the accident	
Post condition	Acknowledge field officer/society	
Alternative condition	If step3 is not successful repeat step1 until it is success	
Exception	The dispatcher must log in into the system to open the request	

Table 4.9 use case description of open request

Use case accident detection

Use case Name	Accident detection
UC_ID:	**UC_09**
Actor:	Accident controlling team, field officer and society
Description:	This is handling of the accident by using allocated resource.
Precondition:	Emergency report must be generated and resource must be available

Flow Event:	Actor action Step1: officer report the accident to Dispatcher.	system response Step4: system display Request Reports.
	Step2: the user wants to see the Accident detection and she/he login to system.	Step8:end use case
	Step3: The user select the Request Page.	
	Step5: user see the incident.	
	Resource and notify to accident controlling team. Step7: Accident controlling team	
	Protects the accident.	
	Step6: The dispatcher allocate	
Post condition	Accident controlling team protects the accidents	
Alternative condition	If step4 is not successful repeat step1 again	
Exception	The user must login to the system	

Table 4.10 use case description of Accident detection

Use case Accident detection report

Use case Name	Accident detection report
UC_ID:	**UC_10**
Actor:	Accident controlling team
Description:	This is used to know daily, monthly, 3 months, half of year, and 1 year Information's about the accident detections.
Precondition:	Accident must be handled or controlled

Flow Event:	Actor action Step1:Accident control team office click on generate report button	system response Step2:the system displays report form
	Step3:Accident control team fill the report form	Step5:the system checks the filled form
	Step4:Accident control team click on submit button	Step6:the system generate the report to the database
		Step7:the system display the successful message
Post condition	Accident controlling team protects the accidents	
Alternative condition	if step5 is not successful repeat step1	
Exception	The user must login in to the system to generate accident detection report	

Table 4.11 use case description of accident detection report

Use case Accident detection report

Use case Name	Post information	
UC_ID:	**UC_11**	
Actor:	Dispatcher	
Description:	In this use case the dispatcher can post information.	
Precondition:	The dispatcher should have to enter a valid user name and Password to upload information.	
Flow Event:	Actor action Step 1: the dispatcher wants to post unknown persons die in Accident. and clicks on post button	system response Step2:the system request information about unknown people die in accident
	Step3: the dispatcher fills the needed information	Step4: The system checks the filled form

	Step6: the report process Ends	Step5: the system stored in the database
Post condition	if a Police officer entered valid user name and password then he/she can post the information	
Alternative condition	if step5 is not successful repeat step1	
Exception	The user must login in to the system	

Table 4.12 use case description of post information

Use case logout

Use case Name	logout	
UC_ID:	**UC_12**	
Actor:	System Administrator(dispatcher), field officer, and accident controlling team	
Description:	Logout and back to the login page	
Precondition:	The System Administrator, field officer, and accident Controlling team should have Internet connection.	
Flow Event:	Actor action Step 1:The user click logout button	system response Step2: The system returns to login Page
	Step3: end use case	
Post condition	The user will be out of the system or database	
Alternative condition		
Exception		

Table 4.13 use case description of logout

4.2.1.3. Use case Scenario

A use case scenario, or scenario for short, describes a real-world example of how one or more people or organizations interact with a system. They describe the steps, events, and/or actions which occur during the interaction. Use case scenarios can be very detailed, indicating exactly how

someone works with the user interface, or reasonably high-level describing the critical business actions but not the indicating how they're performed.

- **Scenario Name: - Login**

Participant actor:-Dispatcher, Field officer, accident controlling team

Entry condition: -login the system

Flow event: -

- Step1 user has to open the system
- step2: user clicks on login button
- step3: the system will display the login page
- step4: user fill username and password
- Step5: system responses to the user username and password form
- Step6: system checks his/her username and password if it is correct the system allows the user to access unless it displays wrong username and password message
- Step7: user click on login button
- Step8:user gets access from the system
- Use case ends

Alternative event:-

- If step4 is not successful repeat step3 until it is success

- **Scenario name: - Adding user Participant**

Participant actor: - Dispatcher

Entry condition: the user should have logged in as Dispatcher

Flow event: -

- Step1: the dispatcher clicks on add user button
- Step2: the system displays the form
- Step3: the dispatcher enters the new user name, account type, Password and re-enter password to the system
- Step4: the dispatcher submits the new user information.

- Step5: the system validates the new user detail. Step6: the system save the user detail to the database. Step7: the system displays successful message of adding new user
- Step8: the dispatcher tells the category, user name and password to the user
- Step9: the system ends

Alternative event:-

- If step7 is not successful repeat step3 until it is success

Scenario name: -updating user account

Participant actor: - Dispatcher, Field officer and accident controlling team

Entry condition: -the user should have logged to the system

Flow event: -

- Step1: he administrator wants to update an account and he/she login to the system.
- Step2: the administrator inserts account _type, username, password, and other user information.
- Step3: the administrator submits the data.
- Step4: the system checks the new account information with the existing account in the database.
- Step5: the system save the new account to the database.
- Step6: the updating process ends

Alternative event:-

- If step5 is not successful repeat step2 until it is success

Scenario name: - Delete user account

Participant actor: -Dispatcher

Entry condition: -the user should have logged in as Dispatcher

Flow event: -

- Step1: the Dispatcher (administrator) wants to delete the account and he/she login to the system.
- Step2: the Dispatcher selects the view button.

- step3: The system display user account information
- Step4: the administrator view the displayed information
- Step5: the administrator selects deletes button
- Step6: the system deletes the account from the system
- Step7: the use case ends

Alternative: -

- If step6 is not successful repeat step2
- **Scenario name: - updating records**

Participant actor: -Accident controlling team

Entry condition: - the user should have logged in as Accident controlling team

Flow event: -

- Step1: the user opens the system
- Step2: the user select files button
- Step3: the user selects update files button after click on files button
- Step4: the system displays accident files stored before
- Step5: the system displays the form based on selected accident id or number
- Step6: the user fills the displayed form
- Step7: the user clicks on update button
- Step8: the system checks the entered information
- Step9: if the entered information is correct the system updates the information
- Step10: the system displays successful message
- Step11: the system updates the information on a database

Alternative: -

- If step10 is not successful repeat step2

Scenario name: Deleting records Participant

Participant actor: -Accident controlling team

Entry condition: - the user should have logged in as Accident controlling team

Flow event: -

- Step1: the user login to the system to delete the records
- Step2: the user selects the view button
- Step3: the system displays the accident information
- Step4: the user clicks on delete button and select file id
- Step5: the system successfully delete the information based on a selected file id/number

Alternative: -

- If step5 is not successful repeat step2.

Scenario name: - Emergency report

Participant actor: Field officer, society

Entry condition: there must be accident and collected information about it

Flow event: -

- Step1: the user opens emergency form
- Step2: system display the emergency report form
- Step3: fill the report form
- Step4: next click emergency report button
- Step5: the system checks whether the information filled correctly or not
- Step6: display successful message to user
- Step7: send emergency report to dispatcher

Alternative: -

- If step6 is not successful repeat step3

Scenario name: - open Request report

Participant actor: Dispatcher, Field officer

Entry condition: - The field officer and society must generate a new emergency report

Flow event: -

- Step1: the user wants to see the request report and he/she login to the system

- Step2: the user select request report page
- Step3: the system display the request report
- Step4: the user sees the request report
- Step5: notifying accident controlling team about the accident happened

Alternative: -

- If not correct display try again message

- **Scenario name: - Accident detection**

Participant actor: - Accident controlling team, field officer and society

Entry condition: - Emergency report must be generated and resource must be available

Flow event: -

- Step1: officer report the accident to dispatcher.
- Step2: the user wants to see the accident detection and she/he login to system.
- Step3: The user selects the request report page.
- Step4: system displays request report page
- Step5: user sees the request report
- Step6: The dispatcher allocates resource and notify to accident controlling team.
- Step7: Accident controlling team protects the accident.

Alternative: -

- If not correct display try again message.

- **Scenario name: - accident detection report**

Participant actor: -accident controlling team

Entry condition: - Accident must be handled or controlled.

Flow event: -

- Step1: the accident control team office clicks on generate report button
- Step2: the system display the report form
- Step3: the accident control team fills the report form
- Step4: the accident control team officer clicks on submit button

- Step5: the system checks the filled form
- Step6: the system generate the report to the database
- Step7: the system displays successful message
- **Alternative:**
- If the investigative officer enters invalid information then the system will generate an error message in order to fill again the information properly.

- **Scenario name: post information**

Participant actor: Dispatcher

Entry condition: - The dispatcher should have to enter a valid user name and password to upload information.

Flow event: -

- Step 1: the dispatcher wants to post unknown persons die in accident. And clicks on post button
- Step2: the system request information about unknown person die in accident
- Step3: the dispatcher fills the needed information
- Step4: after the filled form system post the information
- Step6: the system display successful message

Alternative: -

- The system display an error message

- **Scenario name of use case: -view posted information**

Participant actor: -society/user

Entry condition: - there should be posted information's

Flow event: -

- Step1: the user wants to view the posted information's
- Step2: the user gets access from the system
- Step3: clicks on view button information

- Step4: the system displays posted information
- Step5: the user views posted information

Alternative:

- **Scenario name: - logout**

Participant actor: -system administrator (dispatcher), field officer and accident controlling team

Entry condition: -the system administrator, field officer and accident controlling team should have internet connection

Flow event: -

- Step1: the user click on logout button
- Step2: the system returns to login page

4.3. Object Model

An object model refers to a visual representation of software or systems' objects, attributes, actions and relationship. It enables the creation of an architectural software or system model prior to the development or programming. In our system we used types of object models like class diagram and data dictionary.

4.3.1 Class Diagram

The purpose of a class diagram is to depict the classes within a model. In an object oriented application, classes have attributes (member variables), operations (member functions) and relation. So in our Accident Management system there will be class that makes the system like Accident control team, Dispatcher, society etc.

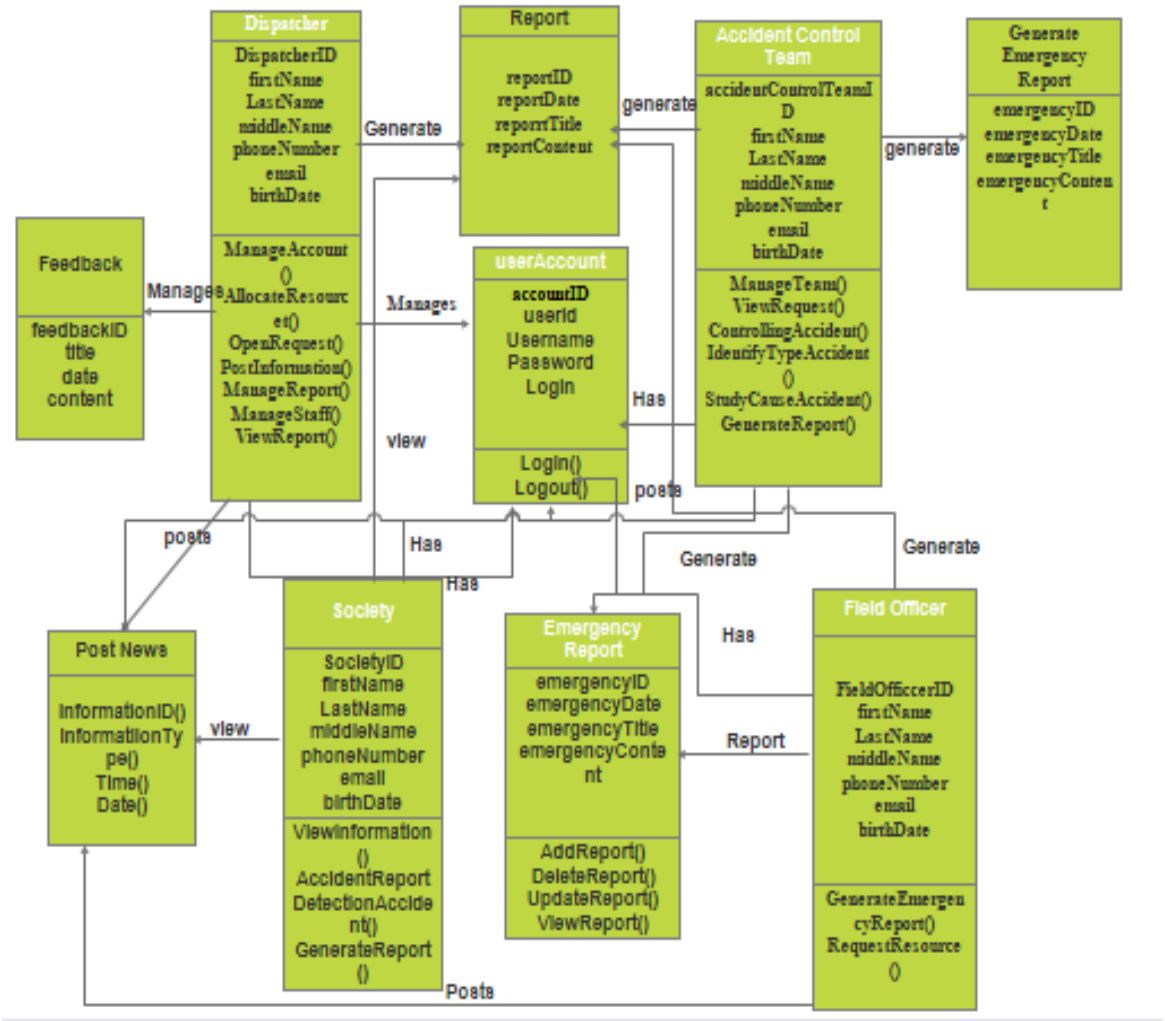


Figure 4.2 class diagram

4.3.2. Data Dictionary

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data.

Data dictionary for Dispatcher

Attributes	Caption	Example	Data Type	Attribute size	Constraint
DispatcherID	ID	Dispatcher/1	varchar	25	Primary Key
firstName	First Name	Eyob	varchar	35	Not null
middleName	Middle Name	Dawit	varchar	35	Not null
lastName	Last Name	Tolosa	Varchar	35	Not null
gender	Gender	Male	Varchar	8	Not null
birthDate	Birth Date	25/4/2013	datetime	10	Not null
phoneNumber	Phone Number	+251945342321	varchar	20	Not null
email	Email	abcde@gmail.com	varchar	30	Not null

Table 4.14 data dictionary for dispatcher

Data dictionary for field officer

Attributes	Caption	Example	Data Type	Attributes size	Constraint
FieldOfficerID	ID	FO/9	varchar	20	Primary Key
firstName	First Name	Ebisa	varchar	30	Not Null
middleName	Middle Name	Edessa	varchar	30	Not Null
lastName	Last Name	Meka	varchar	30	Not Null
gender	Gender	Male	varchar	6	Not Null
birthDate	Birth Date	6/12/1990	datetime	10	Not null
email	E-mail	ebisaedessa@gmail.com	varchar	30	Not Null
phoneNumber	Phone Number	+251947534650	varchar	20	Not Null

Table 4.15 data dictionary for field Officer

Data dictionary for Accident control team

Attributes	Caption	Example	Data Type	Attributes size	Constraint
AccidentControlTeamID	ID	ACT/9	varchar	20	Primary Key
firstName	First Name	Deraro	varchar	30	Not Null
middleName	Middle Name	Biriye	varchar	30	Not Null
lastName	Last Name	Kumure	varchar	30	Not Null
gender	Gender	Male	varchar	6	Not Null
birthDate	Birth Date	4/12/1990	datetime	10	Not null
email	E-mail	Deraro1234@gmail.com	varchar	30	Not Null
phoneNumber	Phone Number	+251940897654	varchar	20	Not Null

Table 4.16 Data Dictionary for Accident Control team

Data Dictionary for Society

Attributes	Caption	Example	Data Type	Attributes size	Constraint
SocietyID	ID	INST/12	varchar	25	Primary Key
firstName	First Name	Ebisa	varchar	35	Not Null
middleName	Middle Name	Edessa	varchar	35	Not Null
lastName	Last Name	Meka	varchar	35	Not Null
gender	Gender	Male	varchar	6	Not Null
birthDate	Birth Date	4/12/1991	datetime	15	Not null
email	E-mail	ebisaedessa@gmail.com	varchar	36	Not Null
phoneNumber	Phone Number	+251947534650	varchar	25	Not Null

Table 4.17 Data dictionary for society

Data Dictionary for User Account

Attributes	Caption	Example	Data Type	Attributes size	Constraint
accountID	Account ID	Acc/35	varchar	25	Primary Key
userID	User ID	Disp/15	varchar	25	Foreign Key
userName	User Name	Atinaf	varchar	35	Not null
password	Password	Password123.	varchar	35	Not null
userRole	User Role	Dispatcher	varchar	25	Not null

Table 4.18 Data Dictionary for User Account

Data dictionary for Report

Attributes	Caption	Example	Data Type	Attributes size	Constraint
ReportID	Report ID	Rep/20	varchar	100	Primary Key
ReportDate	Report Date	21/10/2021	datetime	12	Not null
ReportTitle	Report Title	Written text	varchar		Not null
userRole	User Role	Accident control team	varchar	20	Not null
reportContent	Report Content	Written text	varchar	25	Not null

Table 4.19 Data Dictionary for Report

Data dictionary for feedback

Attributes	Caption	Example	Data Type	Attributes size	Constraint
feedbackID	Feedback ID	FEEDBACK/23	varchar	35	Primary Key
userRole	User Role	Dispatcher	varchar	32	Not null
title	Feedback title	Any written text	varchar		Not null

date	Date	23/11/2020	datetime	12	Not null
content	Feedback content	Any written text	varchar		Not null

Table 4.20 Data Dictionary feedback

Data dictionary for post news

Attributes	Caption	Example	Data Type	Attributes size	Constraint
NewsID	news ID	news/30	varchar	35	Primary Key
informationID	Student ID	INF/23	varchar	54	Foreign Key
informatiionName	Student Name	Title	varchar	90	Not null
InformationType	Course Name	Fire	varchar	45	Not null
time	Time	5:50	datetime	6	Not null
date	Date	12/10/2022	datetime	12	Not null

Table 4.21 Data dictionary of post news

Data dictionary generate emergency report

Attributes	Caption	Example	Data Type	Attributes size	Constraint
EmergencyID	Emergency ID	Emer/20	varchar	100	Primary Key
EmergencyDate	Emergency Date	21/10/2021	datetime	12	Not null
EmergencyTitle	Emergency Title	Written text	varchar		Not null
userRole	User Role	Accident control team	varchar	35	Not null
EmergencyContent	Emergency Content	Written text	varchar	25	Not null

Table 4.22 Data dictionary emergency report

Data dictionary manage file

Attributes	Caption	Example	Data Type	Attributes size	Constraint
FileID	File ID	Acc/35	varchar	50	Primary Key
FileID	User ID	Disp/15	varchar	50	Foreign Key
FileName	File Name	Gizaw	varchar	35	Not null
password	Password	abcd1234	varchar	25	Not null
userRole	User Role	Accident control team	varchar	25	Not null

Table 4.23 Data dictionary for manage file

4.4. Dynamic Model

Dynamic model describes those aspect of the system that are concerned with time and sequencing of the operations. It is used to specify and implement the control aspect of In our system we used sequence diagram, state chart diagram and activity diagram to study the dynamic nature of the system.

4.4.1. Sequence Diagram

Sequence diagrams are used to depict graphically how objects interact with each other via messages in the execution of a use case or operation. They illustrate how the messages are sent and received between objects and in what sequence.

Sequence diagram for login

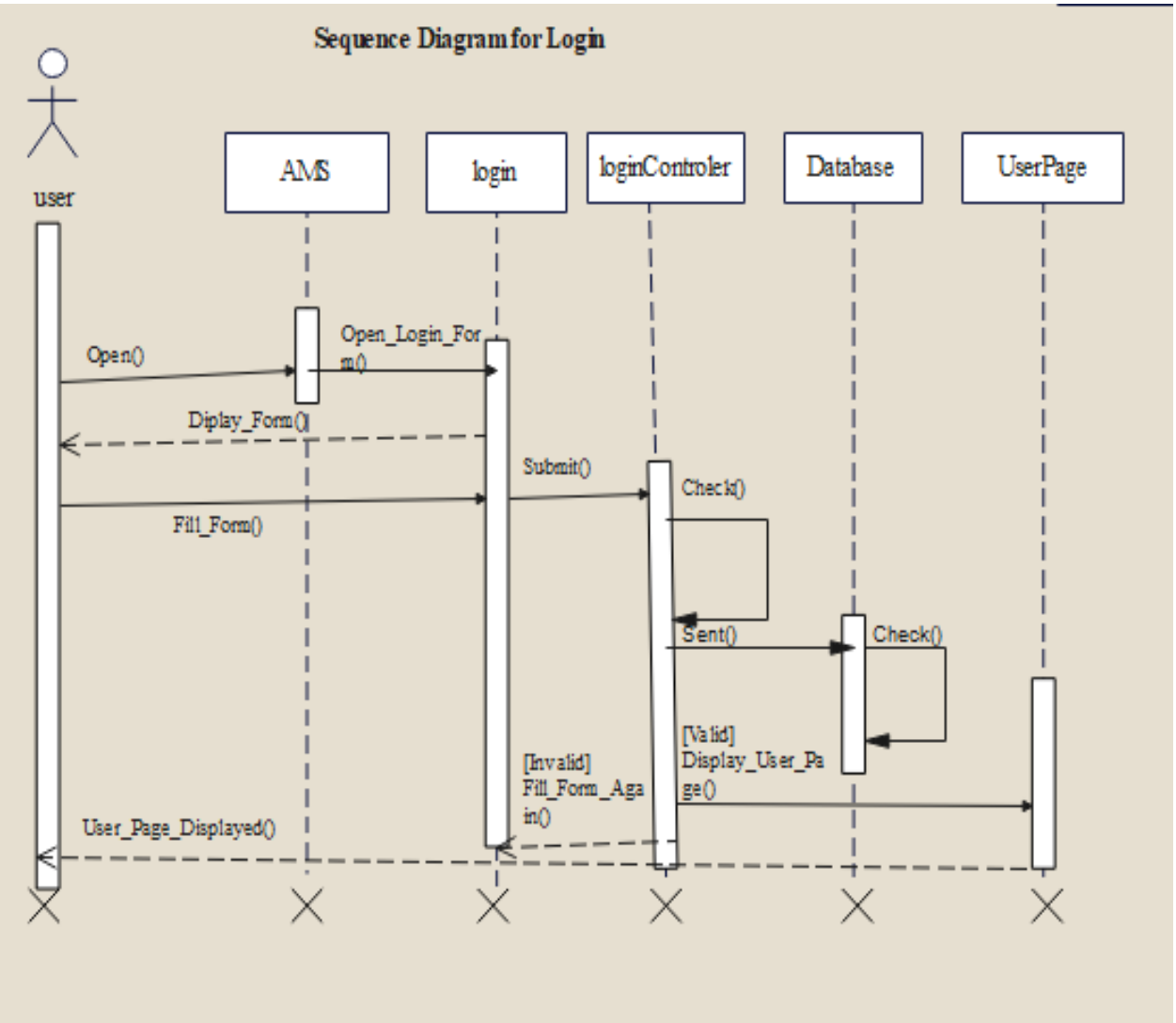


Figure 4.3 sequence for login

Sequence diagram for Updating record

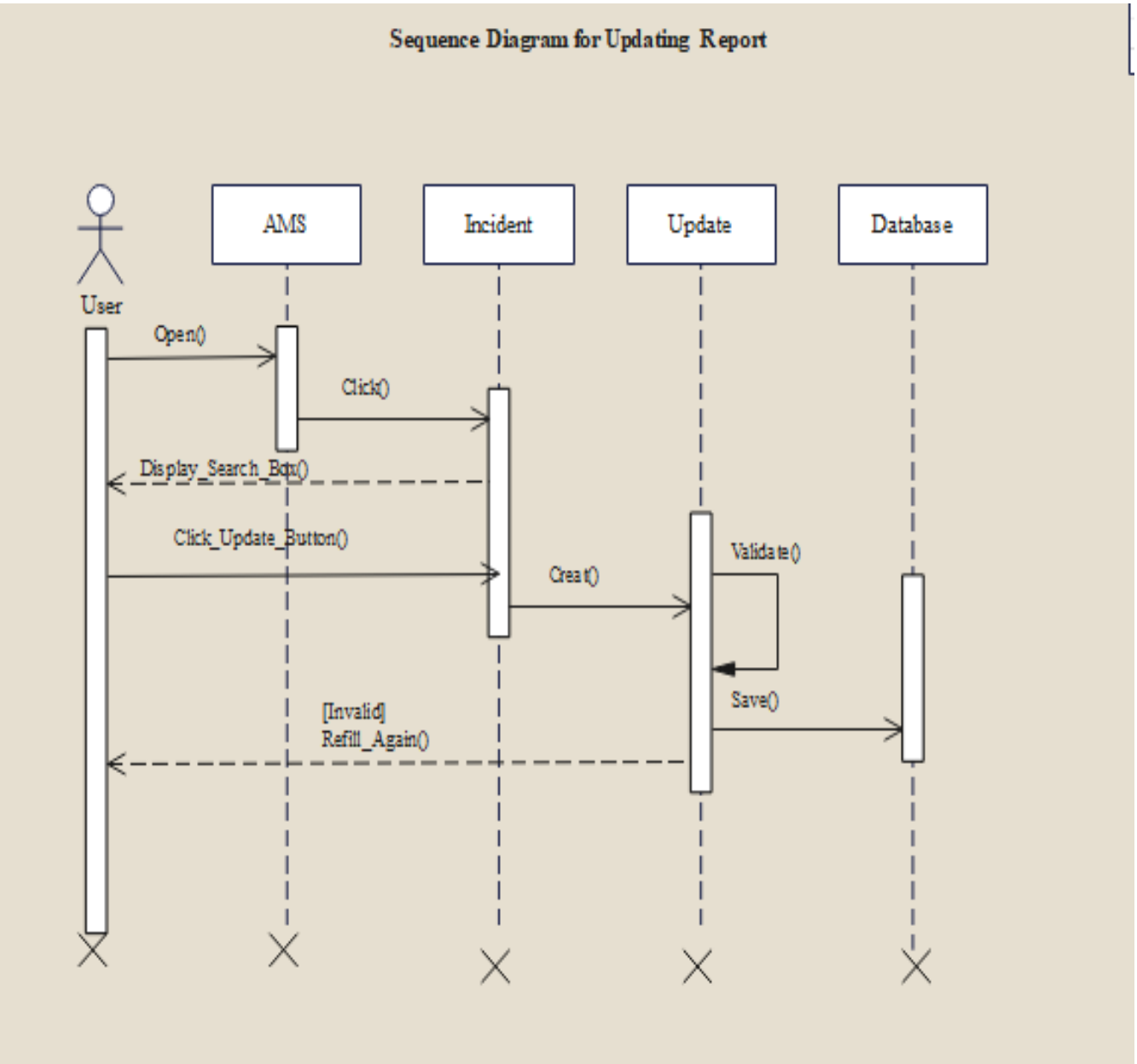


Figure 4.4 sequence for updating report

Sequence diagram for delete report

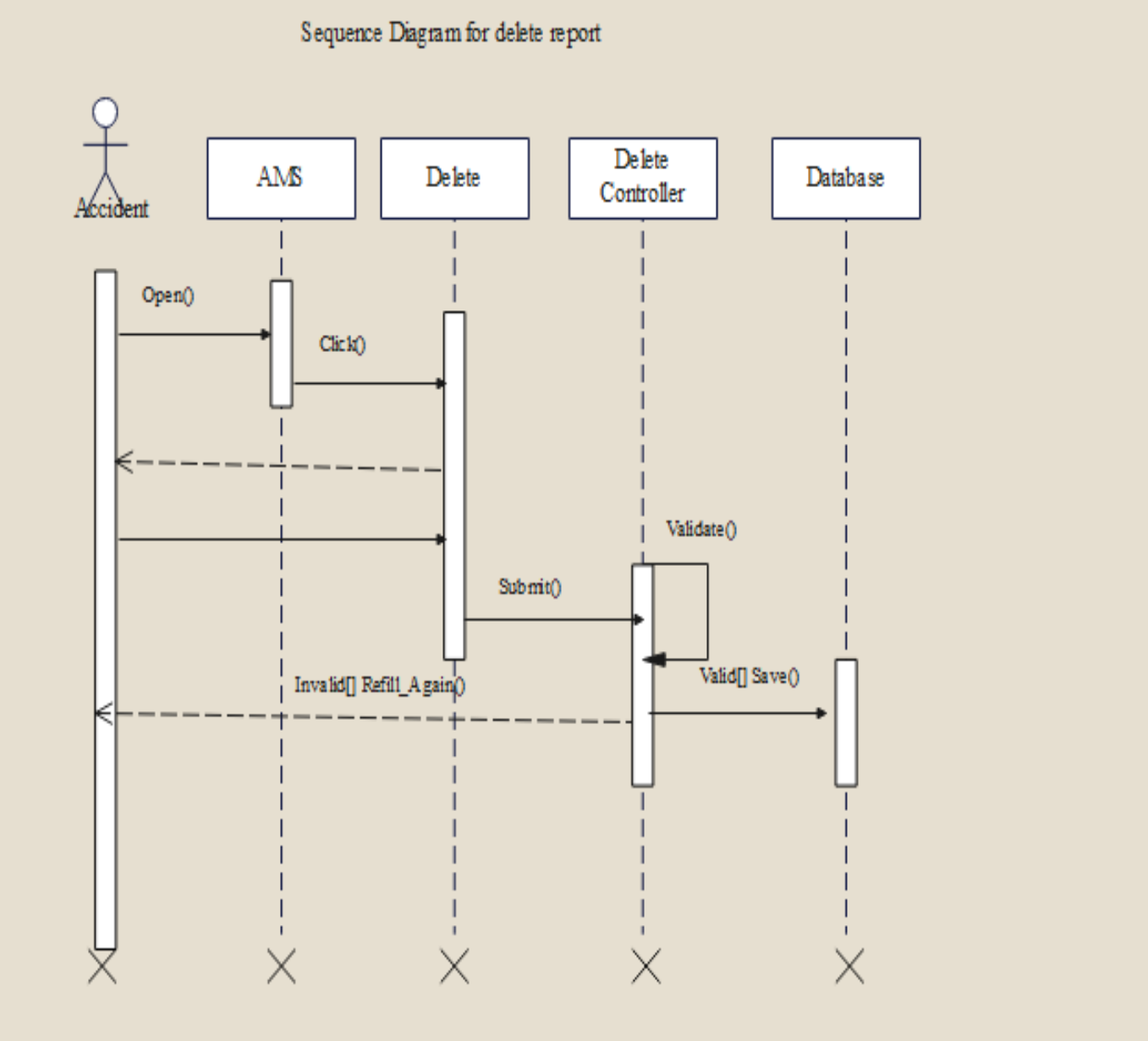


Figure 4.5 sequence for delete report

Sequence diagram for emergency report

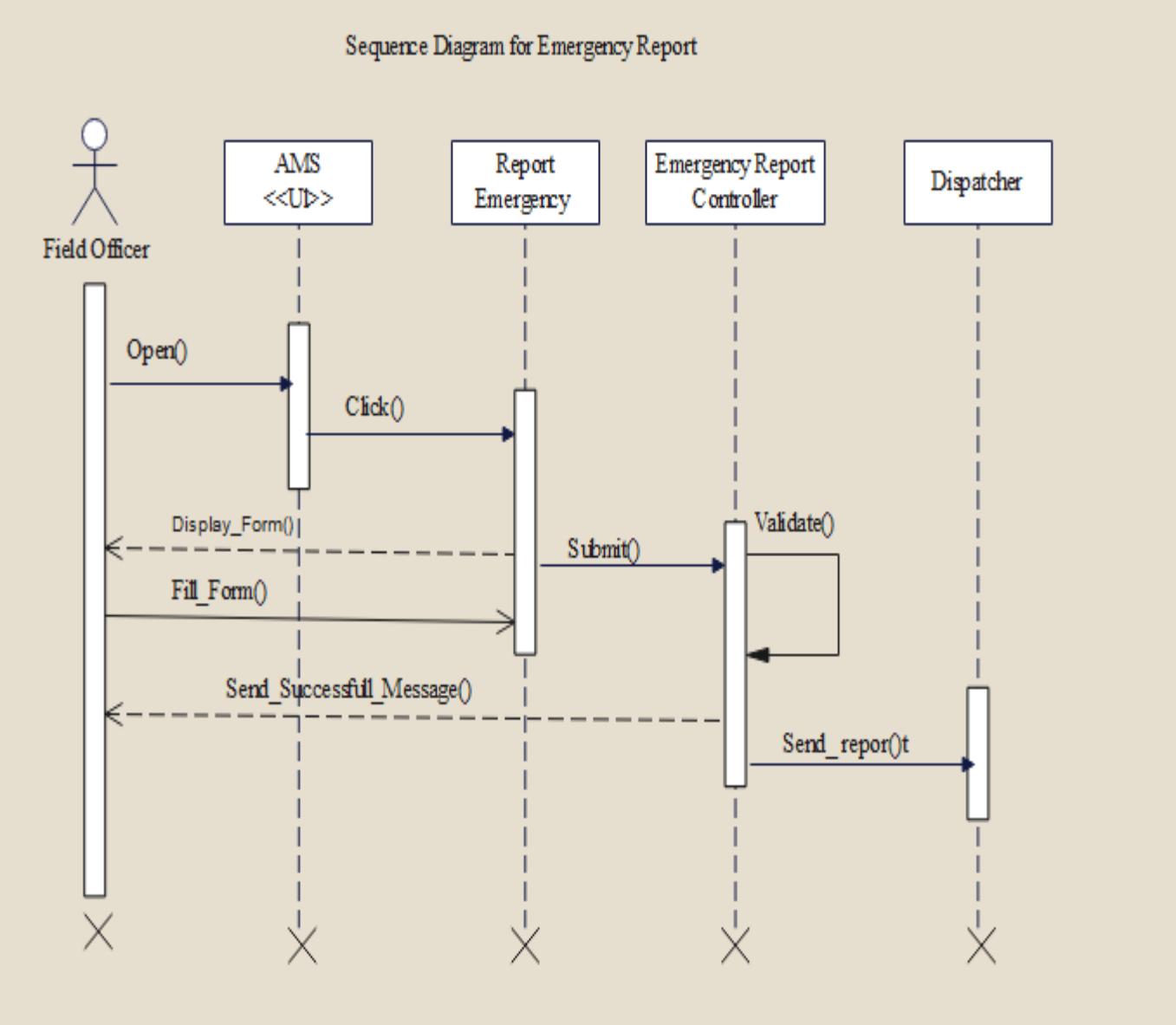


Figure 4.6 sequence for emergency report

Sequence diagram for Open Incident

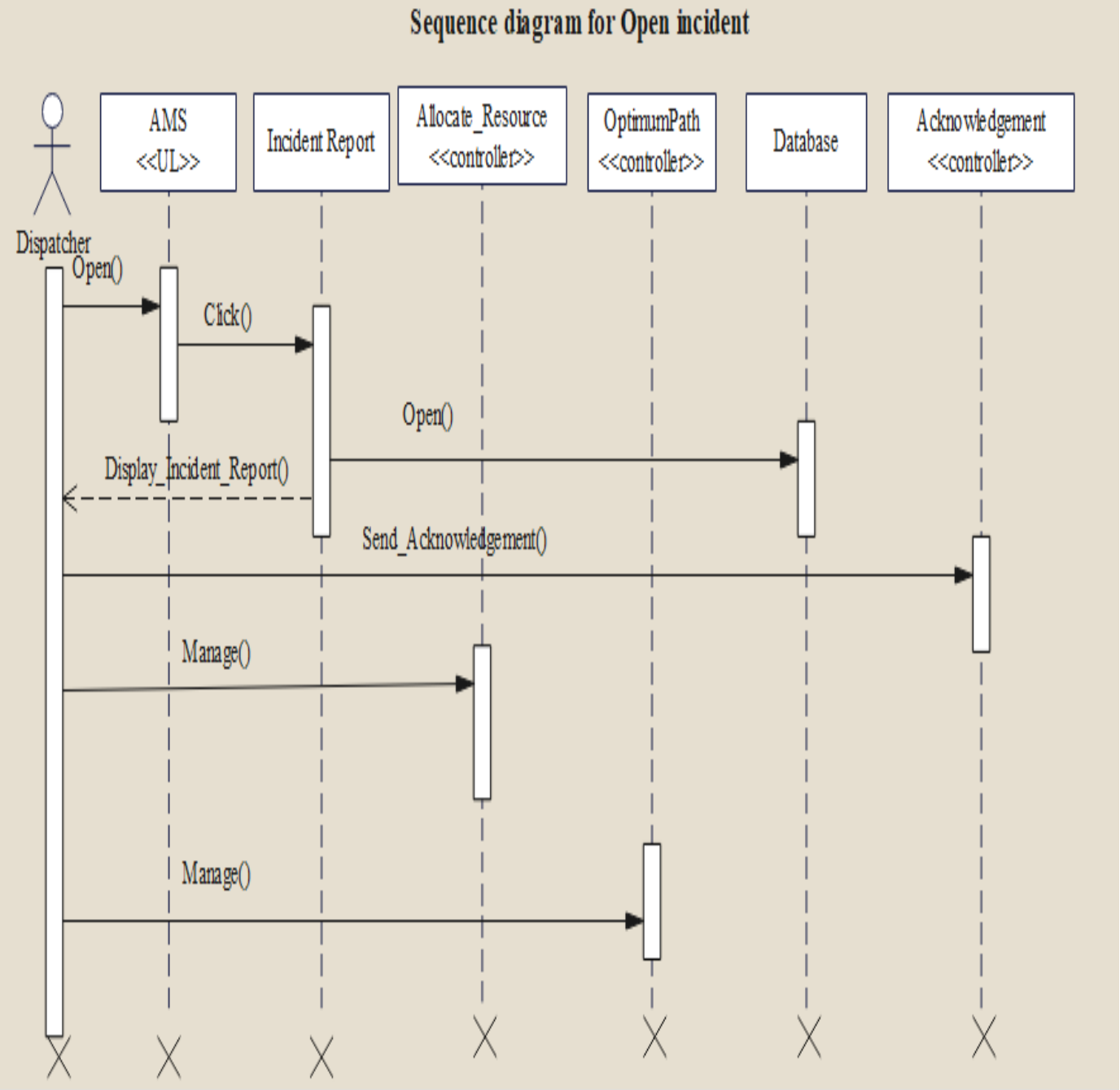


Figure 4.7 sequence for open incident

Sequence diagram for Detection report

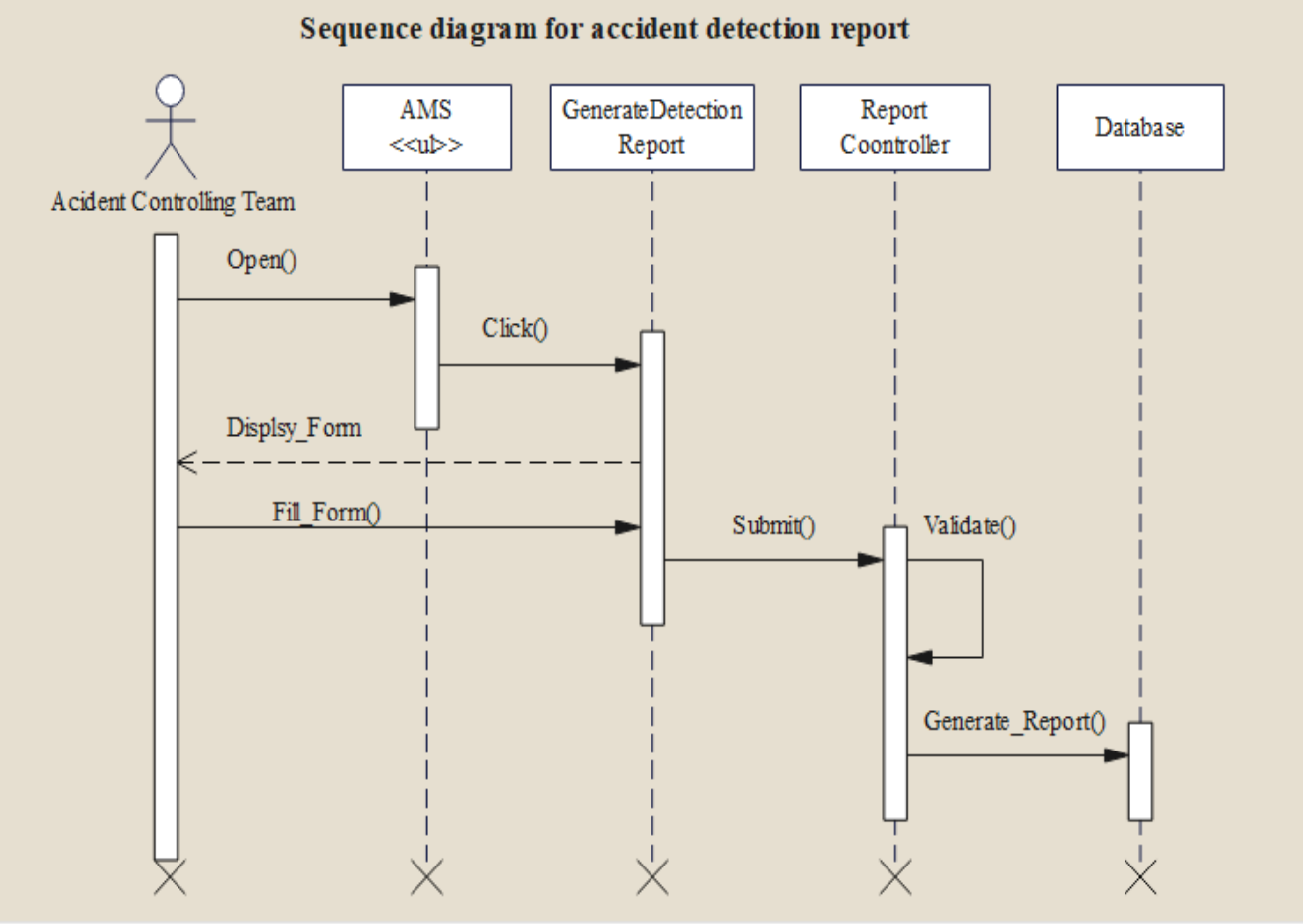


Figure 4.8 Sequence for accident detection report

4.4.2 Activity Diagram

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system in order to perform many tasks there should be one or more activity to be done.

Activity Diagram for Login

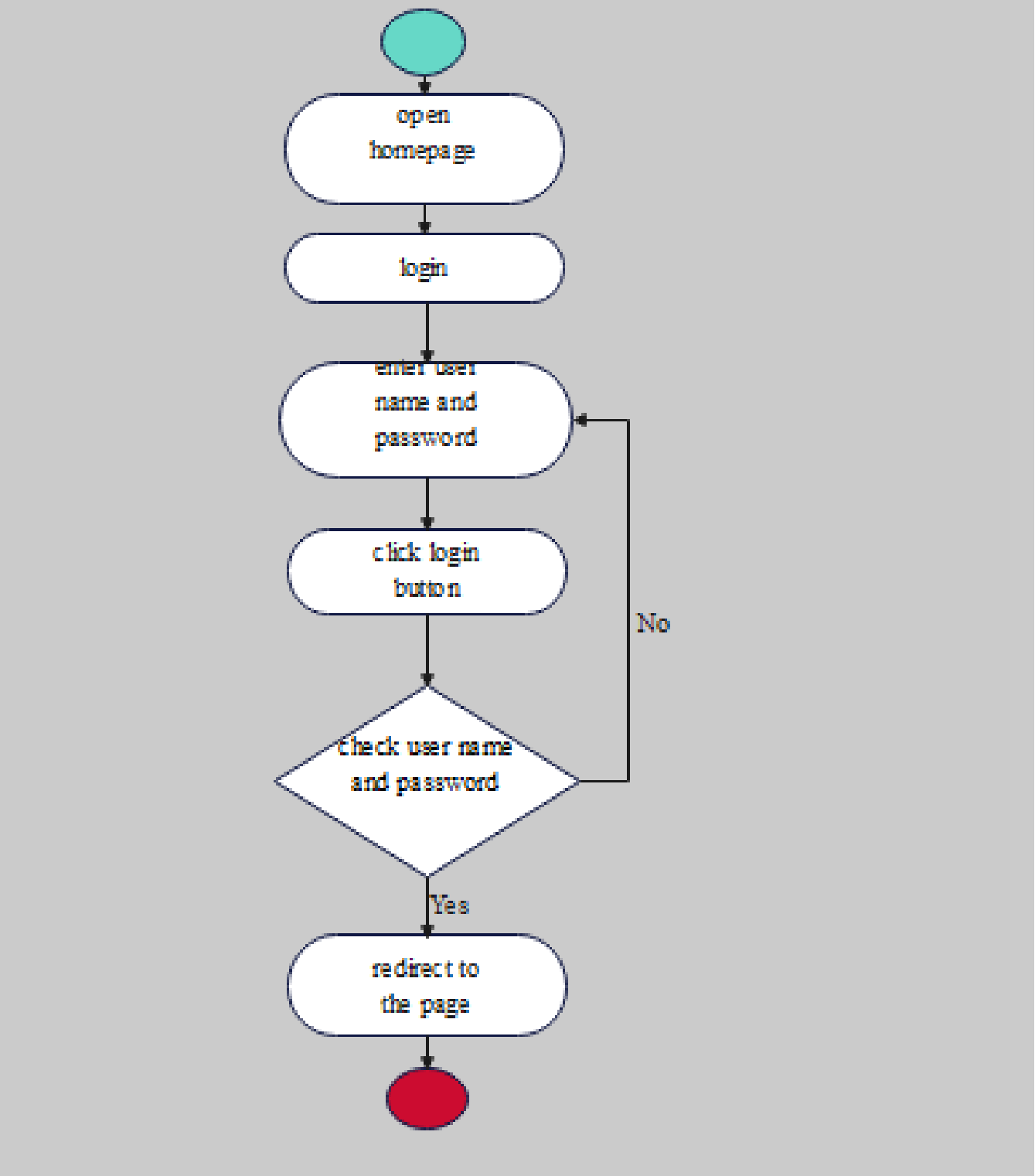


Figure 4.9 activity Diagram for Login

Activity Diagram for accident detection report

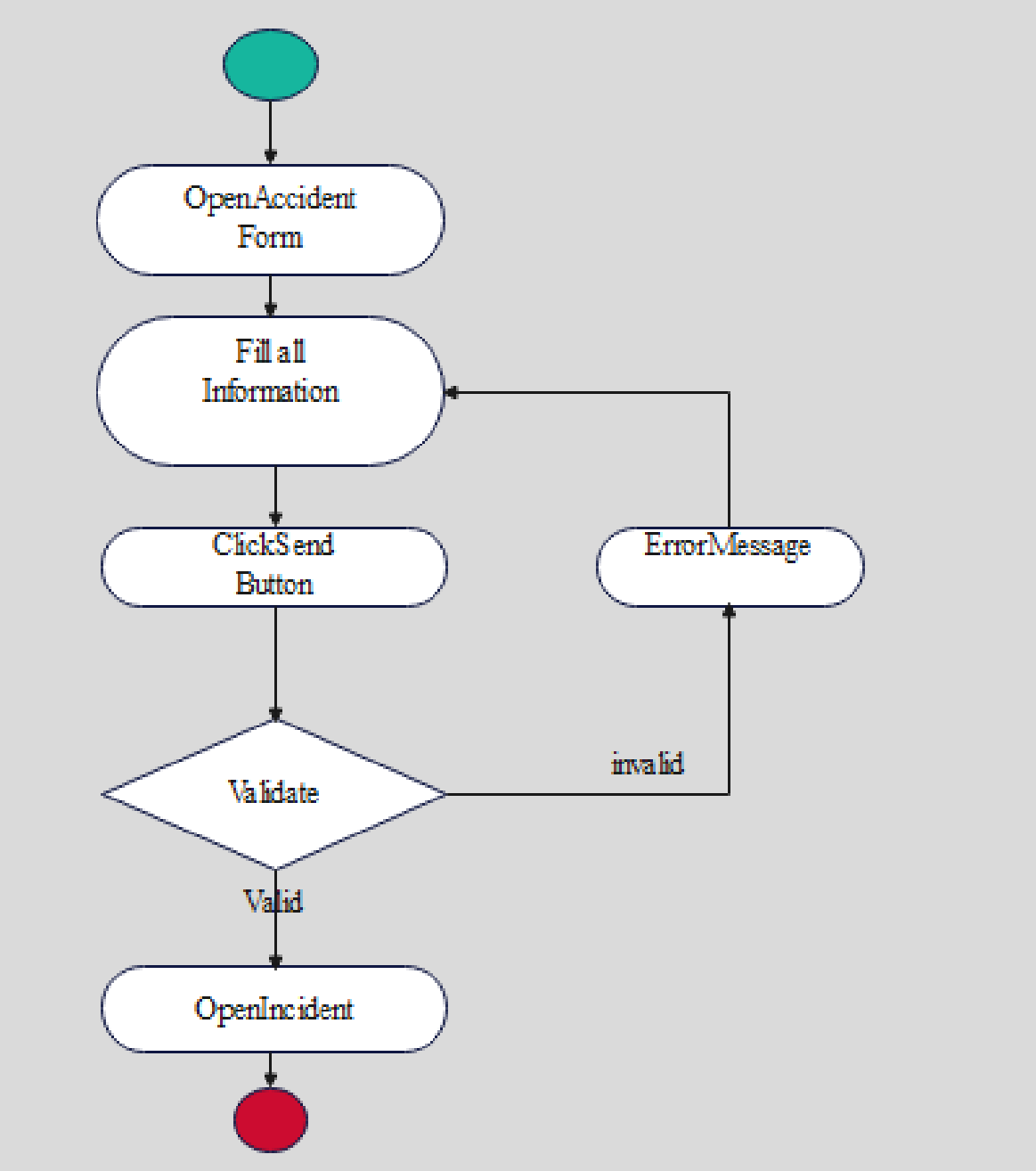


Figure 4.10 activity Diagram for accident detection report

Activity diagram for generate report

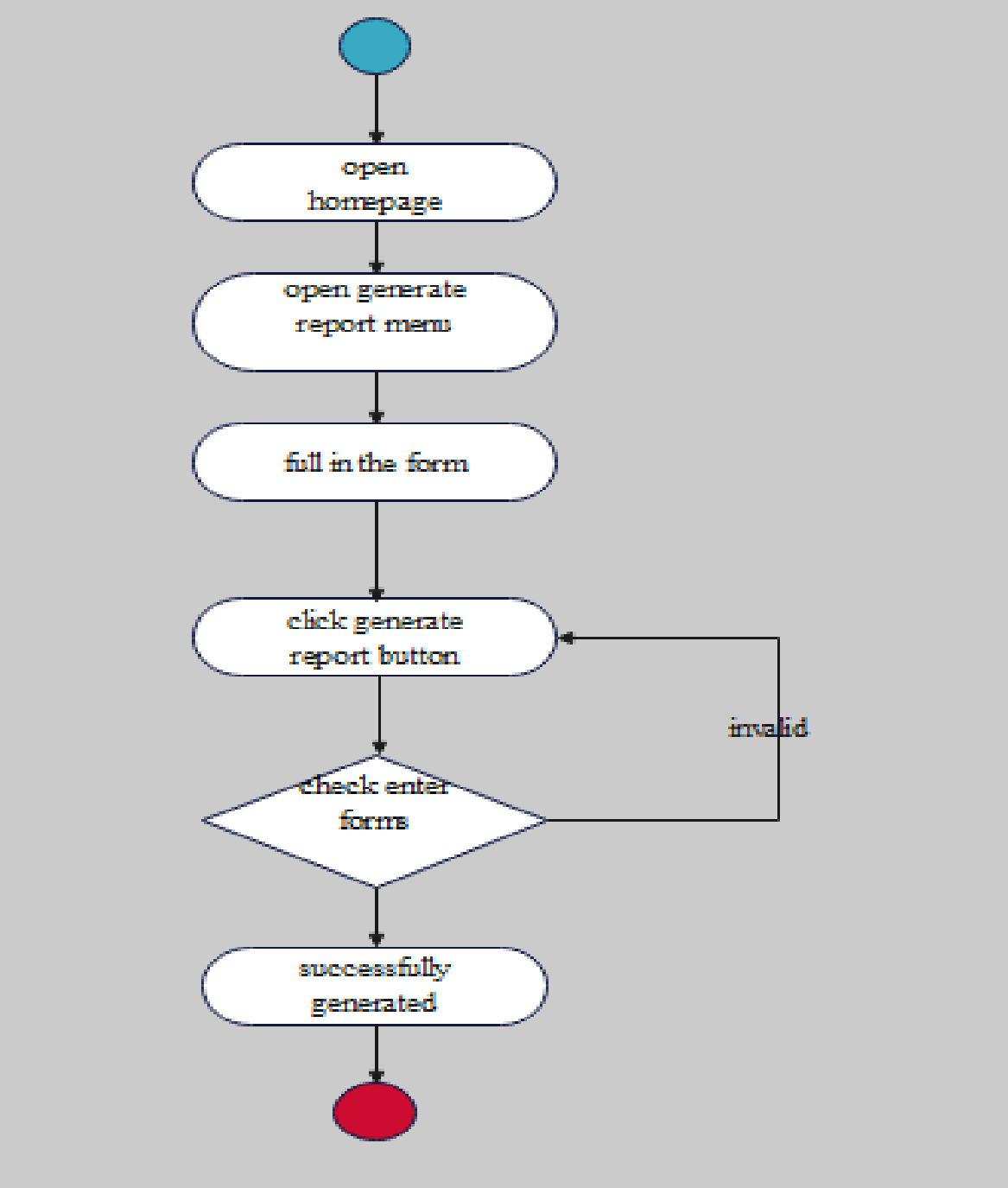


Figure 4.11 activity Diagram for generate report

Activity diagram for update report

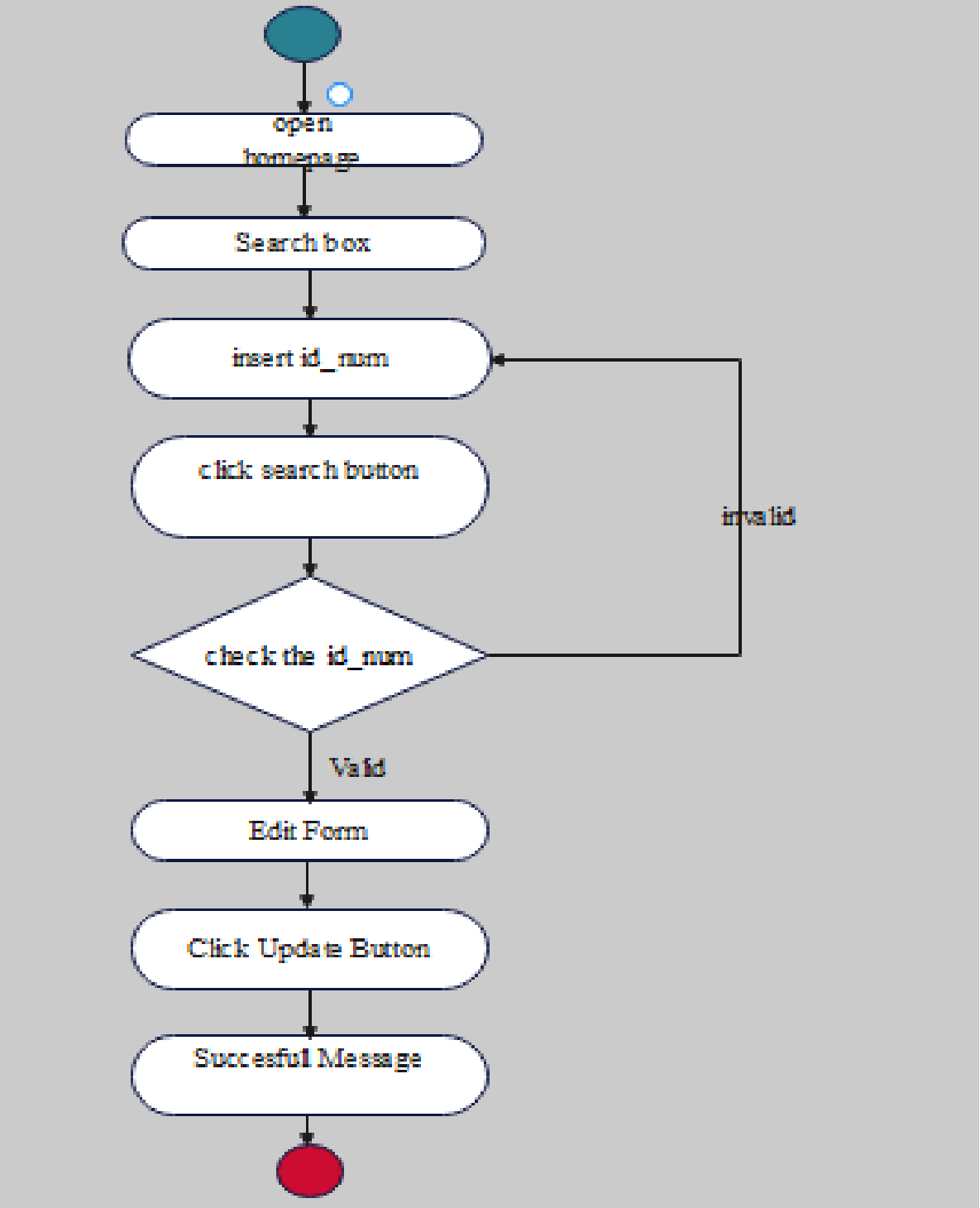


Figure 4.12 activity Diagram for update report

Activity diagram for delete report

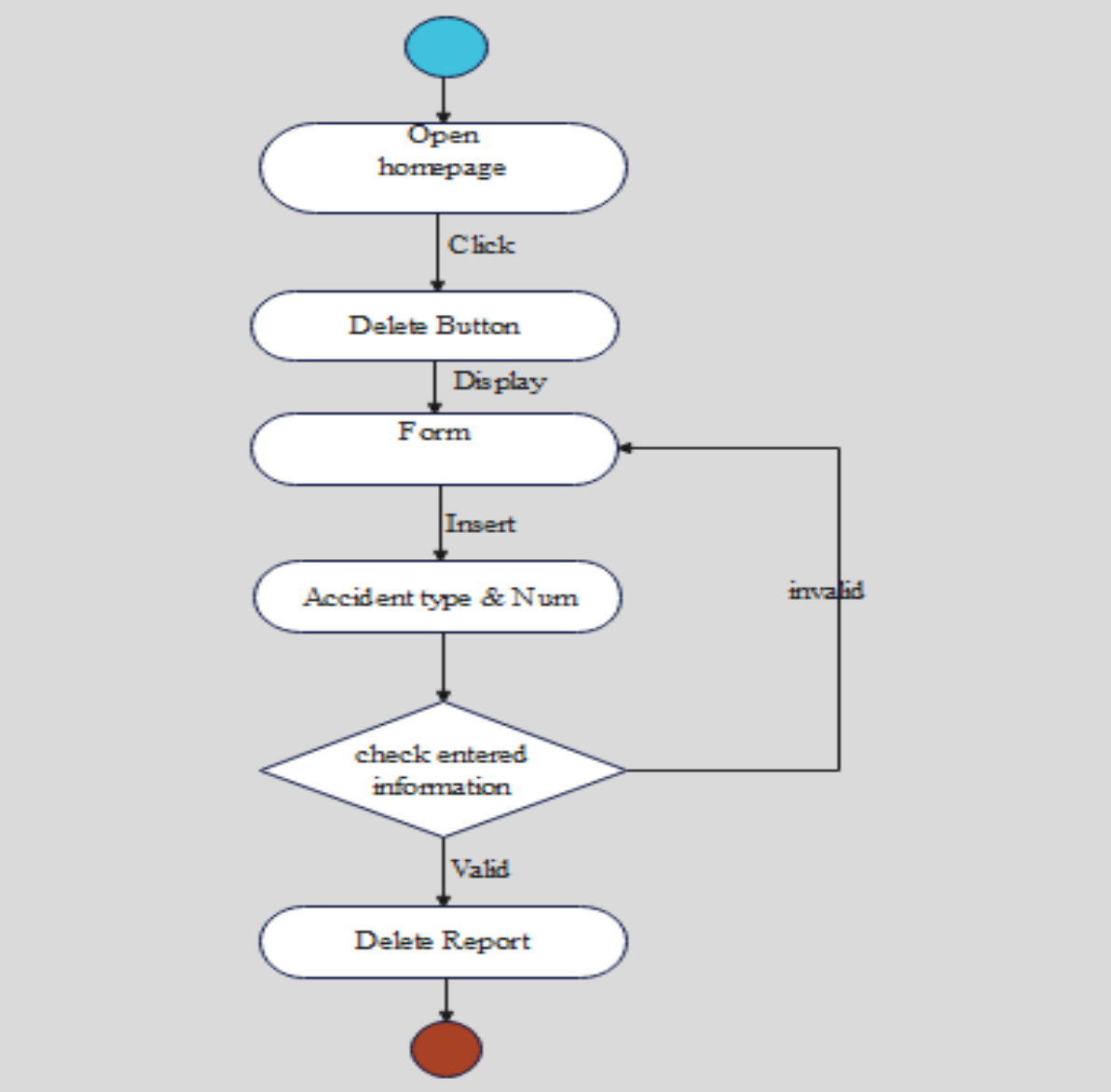


Figure 4.13 activity Diagram for delete report

4.4.3. State Chart Diagram

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of State chart diagram is to model life time of an object from creation to termination. State chart is flow of state to state. State chart modeling is a dynamic modeling

Technique, one that focuses on identifying the behavior within our system, behavior specified to the instances of a single class. It tries to show different state that an object passes through its life span. However, it is not necessary to build state chart for every class in the system; only state charts of complex objects are necessary to be modeled. State chart diagram enables us to observe the state of complex that simplifies implementation.

State chart diagram for login

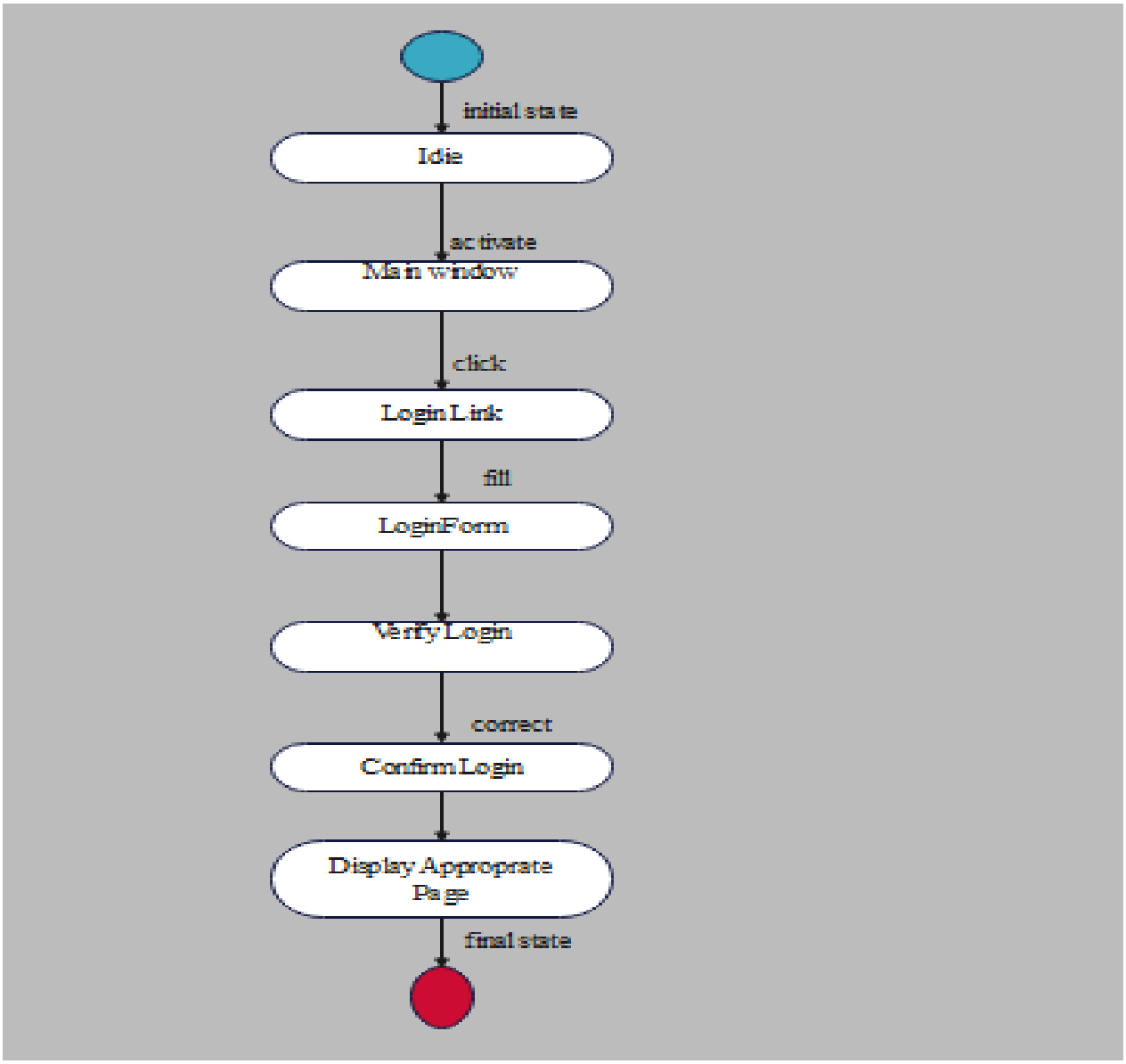


Figure 4.14 state chart diagram login

State chart for view report

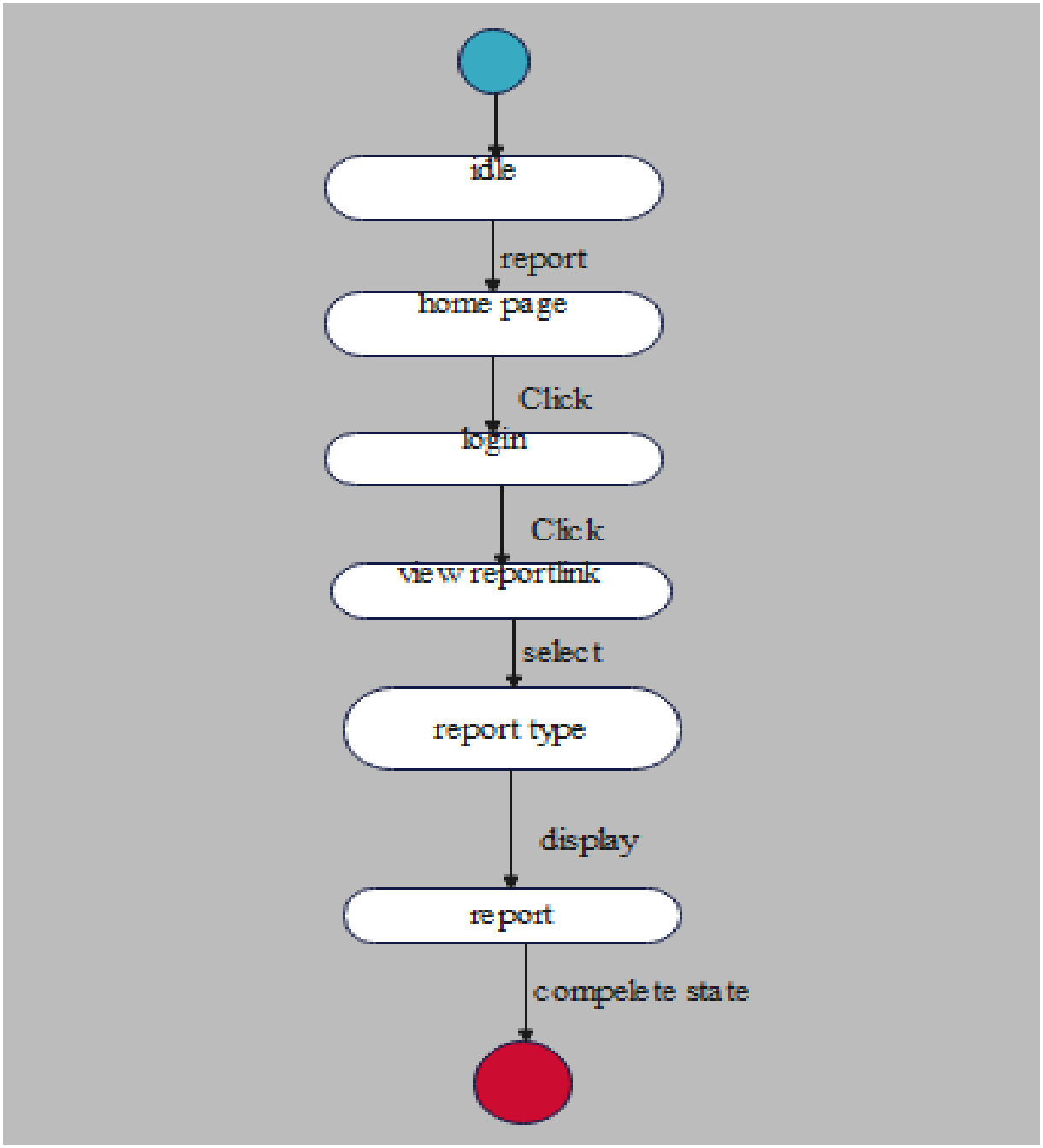


Figure 4.15 state chart diagram view report

State chart diagram for post information

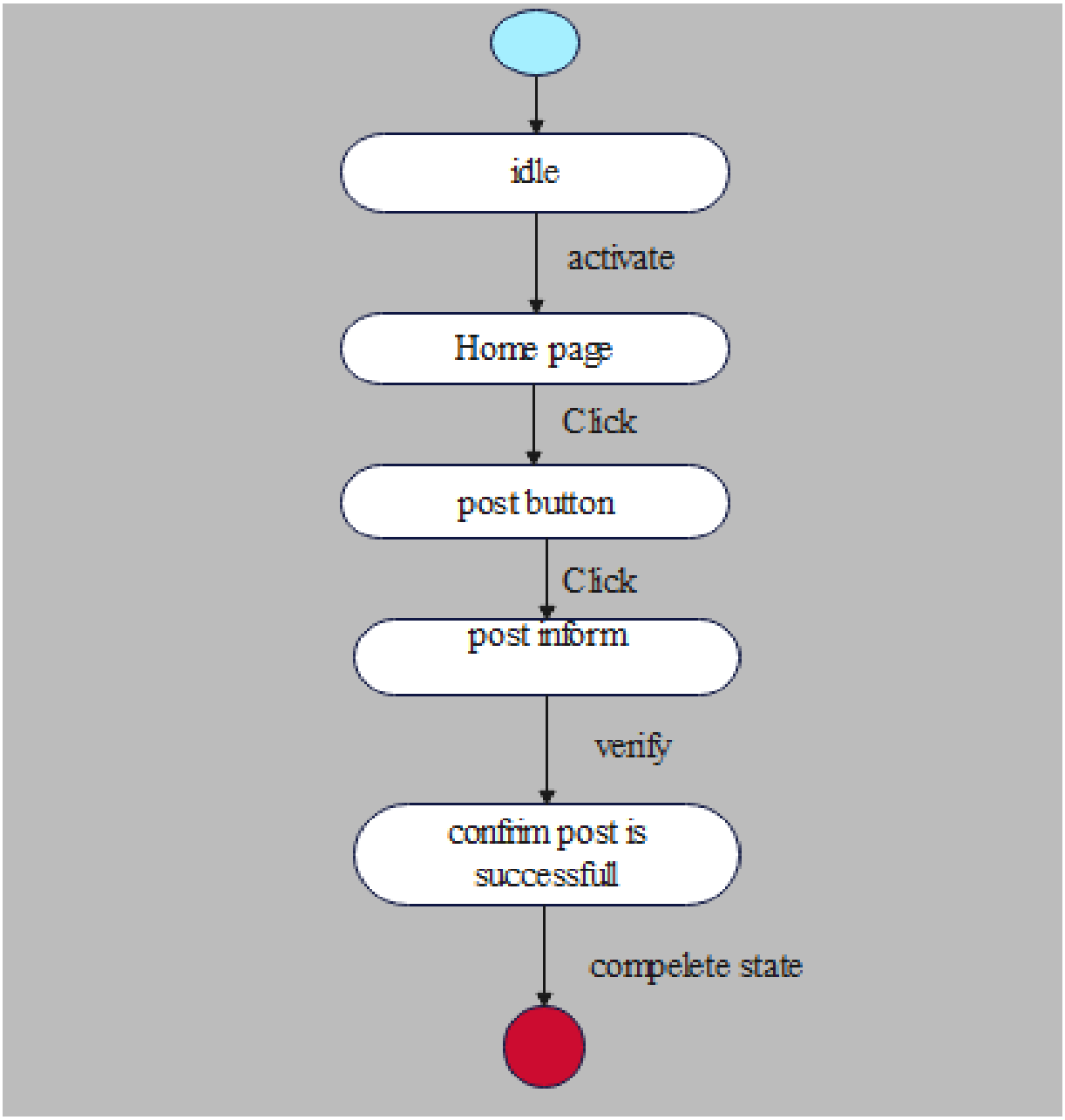


Figure 4.16 state chart diagram post information

State chart diagram for Update report

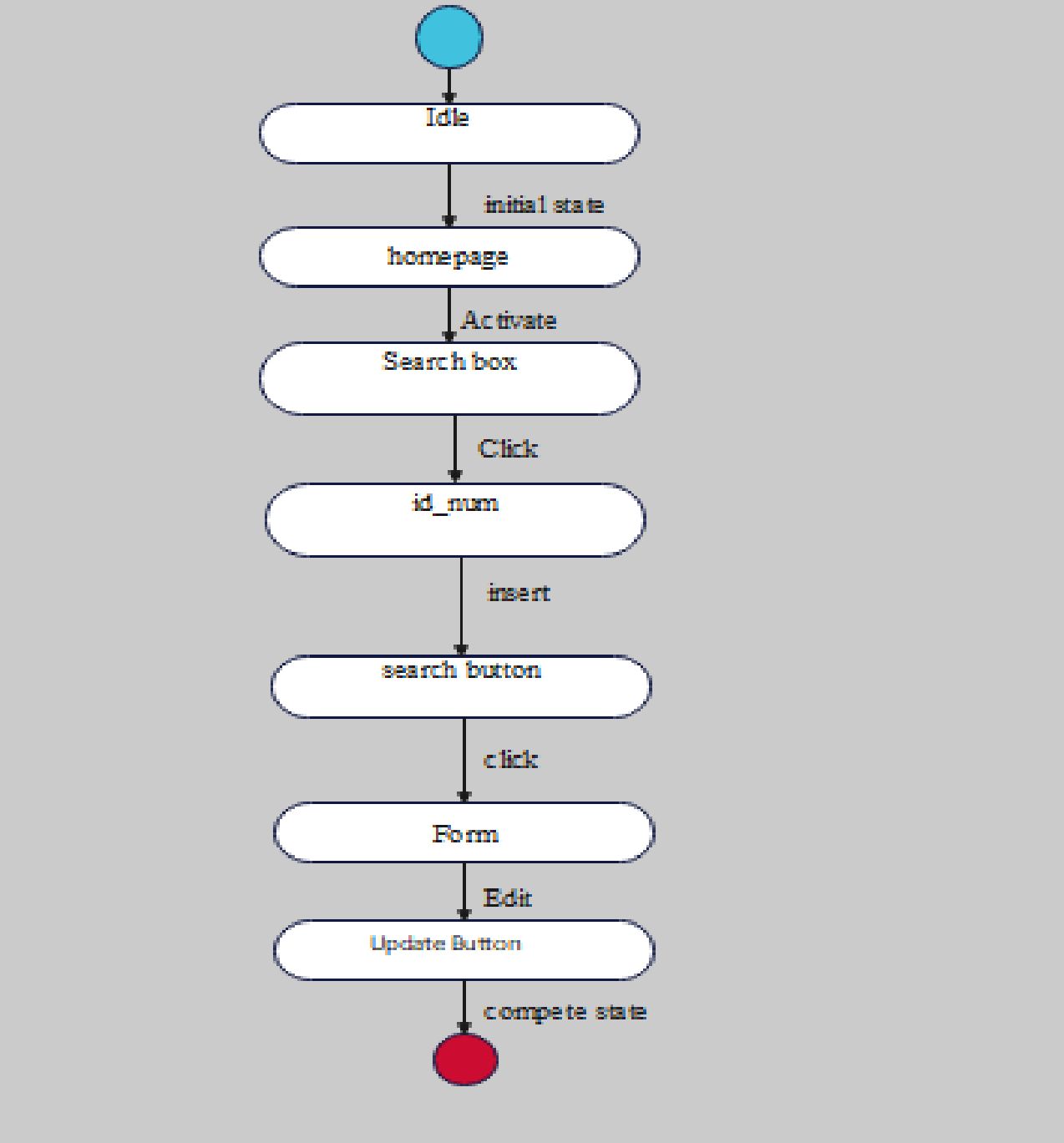


Figure 4.17 state chart diagram update report

State chart diagram for delete report

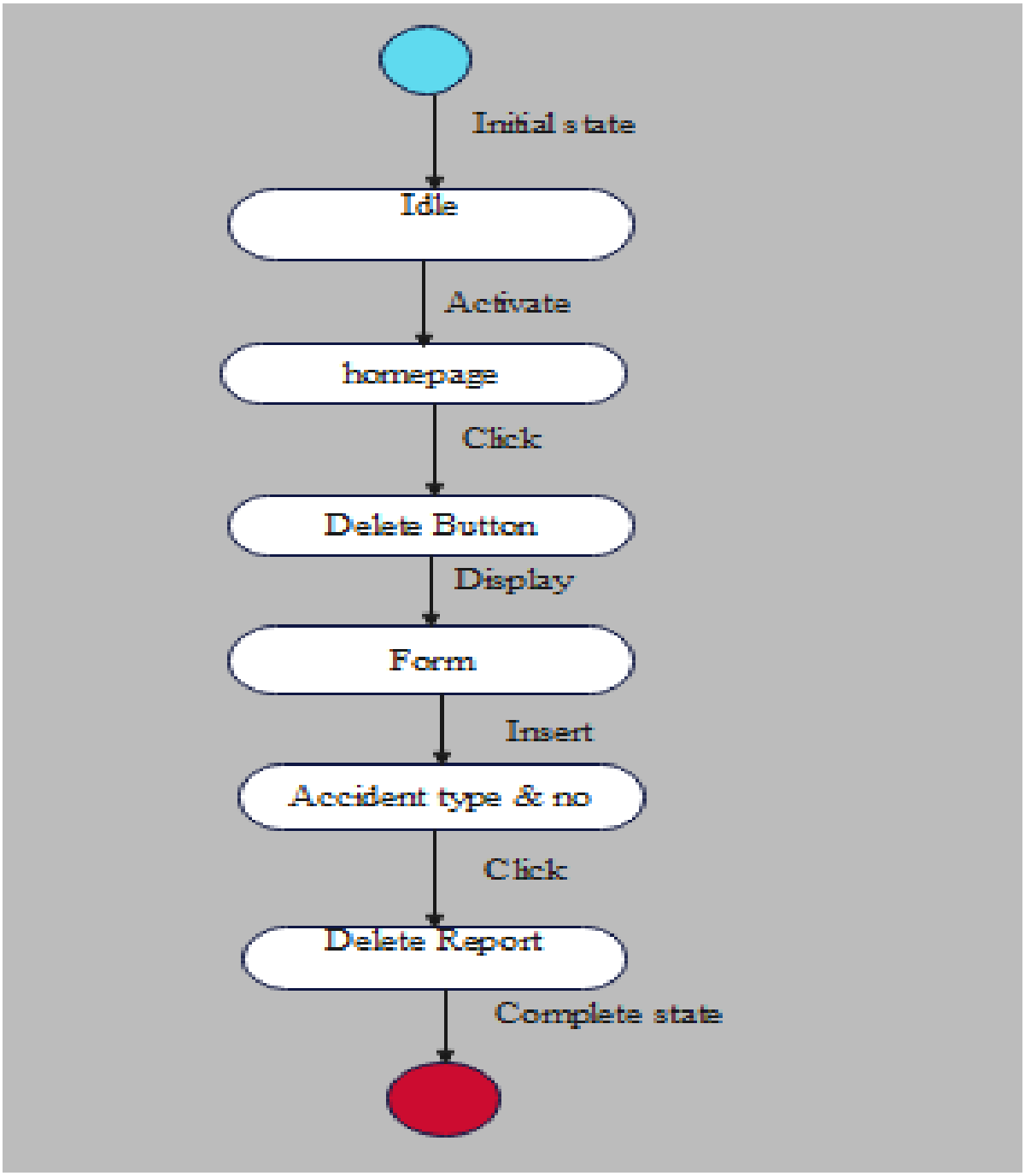


Figure 4.18 state chart diagram delete report

CHAPTER FIVE

5. SYSTEM DESIGN

5.1 Introduction

System design is the transformation of the analysis model into a system design model. System design is the first part to get into the solution domain in software development. This chapter focuses on transforming the analysis model into the design model that takes into account the nonfunctional requirements and constraints described in the problem statement and requirement analysis sections discussed earlier. The purpose of designing is to show the direction of how the system is built and to obtain clear and enough information needed to drive the actual implementation of the system. It is based on the understanding of the model the software is built on. The objectives of the design are to model the system with high quality. Implementing a high-quality system depends on the nature of the design created by the designer. If someone wants to change to the system after it has been put into operation depends on the quality of the system design, so if the system is designed effectively, then it will be easy to make changes to it.

5.2 Design Goal

The design goal of this developed system is to make it easier for the end-users and to give quick and reliable service for users. These describe the qualities of the system that we should optimize. Many design goals are inferred from the non-functional requirements or from the application domain. The developed system is considered to be successful if it meets the following sets of criteria.

- **User Interface**: The user interface of the system allows be easy to use by each user of the system with little training.
- **Documentation**: System administrators and other users are provided with proper documentation about the features of the software.
- **Performance**: The system allows be able to serve a number of users who are expected to access it concurrently.

- **Error Handling and Extreme conditions**: The system allows be robust enough to handle error conditions and continue with normal operations.
- **Availability**: The system availability allows be available most of the time since it is handling emergency situations.
- **Security:** The system allows prevent sensitive data from unauthorized access.
- **Modifiable:** The system allows be designed in Object Oriented language so that modification to some part of the system could not affect other parts.

5.3 Current system architecture

In spite of its organized administrative structure, Wolkite city police station uses accident management in a manual system. In order to handle or process the data for the above-mentioned categories of tasks, there is a manual system. An accident happens then all information is recorded manually. The Wolkite city police station does not use any software. In the existing system, all things done in manual, from collecting the accident information up to documentation are done in manual.

5.4 Proposed system architecture

The proposed subsystem will be implemented in Client/Server architecture. Wherever a user is as long as there is an Internet connection he/she can browse the Wolkite city accident management (WLKCAMS) web page, fill in the required inputs by the web page, and then submit it then the request of the user will be sent to the server. The server will give a response based on the user's request. From this description, the architecture of the system is depicted diagrammatically.

In this developed system we use multi-tier architecture: presentation tier, application tier and database tier.

- **Presentation tier**

The presentation tier is the user interface and communication layer of the application, where the end-user interacts with the application. Its main purpose is to display information and collect information from the user.

- **Application tier**

The application tier, also known as the logic tier or middle tier, is the heart of the application. In this tier, information collected in the presentation tier is processed - sometimes against other information in the data tier - using business logic, a specific set of business rules. The application tier can also add, delete or modify data in the data tier.

- **Database tier**

The data tier, sometimes called the database tier, data access tier, or back-end, is where the information processed by the application is stored and managed.

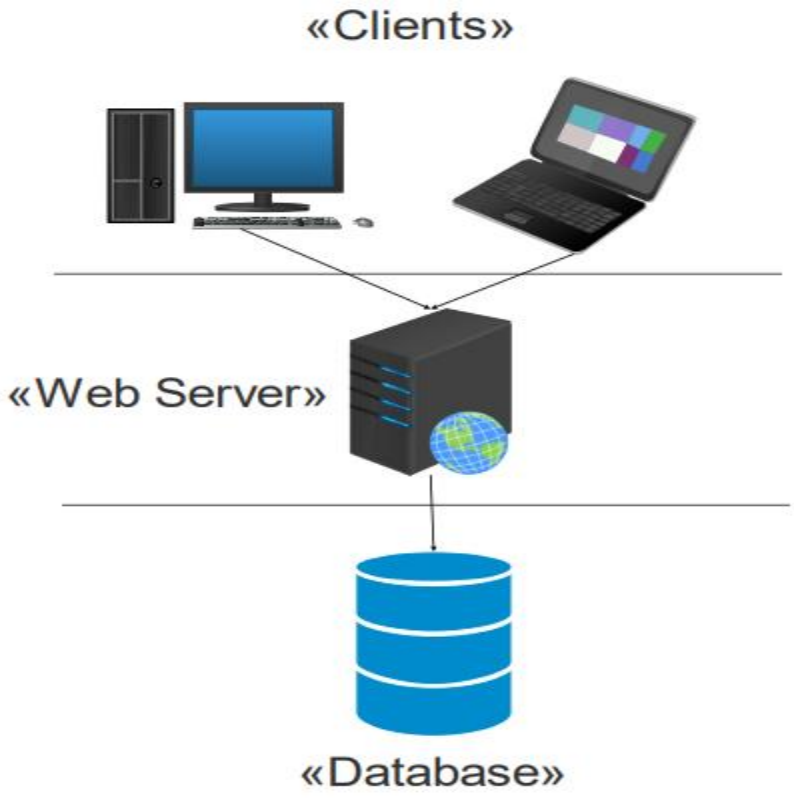


Figure 5.1 general architecture of proposed system

5.4.1 Subsystem decomposition

Subsystem decompositions will help us to reduce/minimize the complexity of the system. So the team identifies the following subsystem from the main system:

Manage User Account Subsystem: The user management subsystem controls the account of the system user.

- Create account
- Update account
- Delete account
- View account

Manage User Subsystem: - for managing user information and performing those operations.

- Add user
- Update user
- Delete user
- View user

Manage file Subsystem: this subsystem manages detail information about file

- Add file
- Update file
- Delete file
- View file

Manage accident information Subsystem

- Record accidents
- View accidents
- Delete accidents
- Update accidents

Manage report Subsystem

- View report
- Generate Report

Feedback Subsystem:

- Give comment
- Delete comment
- Update comment
- View comment

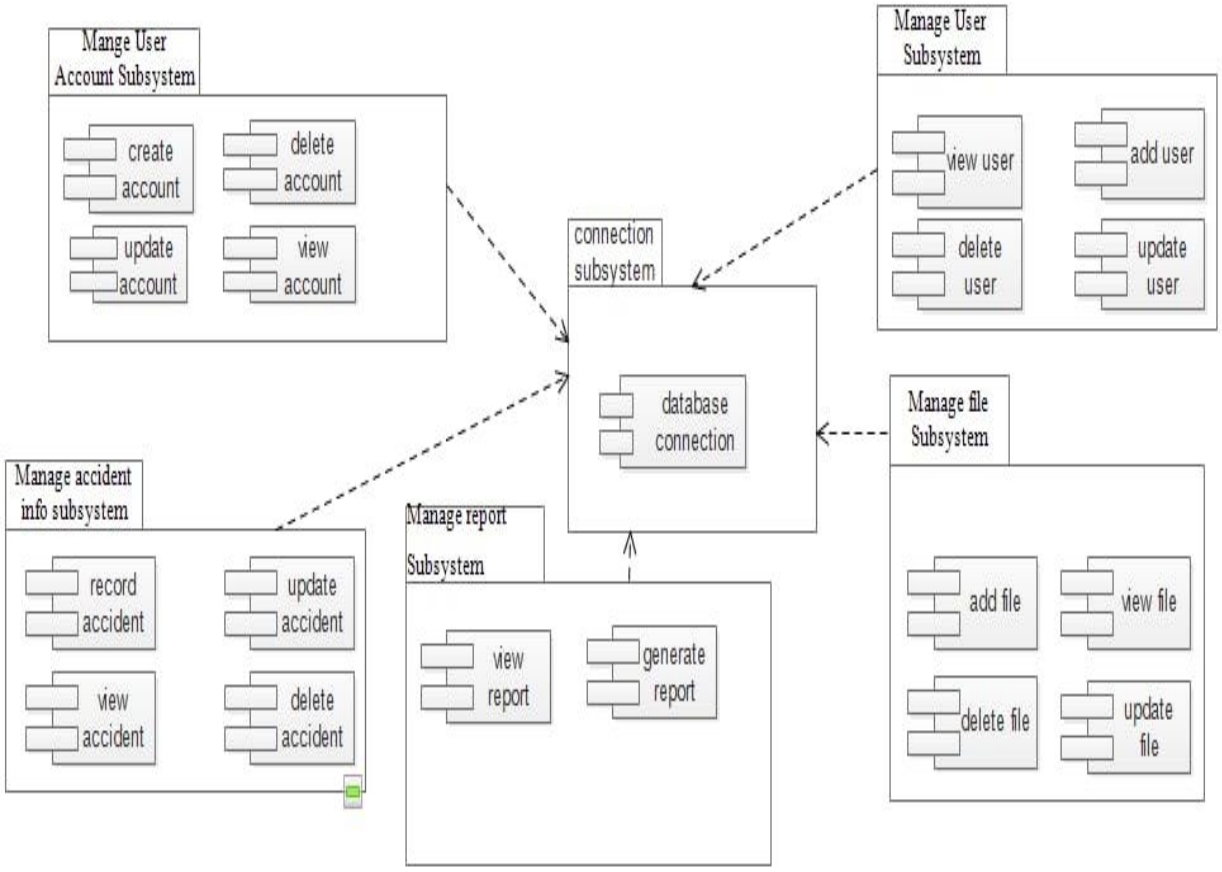


Figure 5.2 subsystem decomposition

5.4.2 Hardware/Software Mapping

Hardware or software mapping describes how subsystems are assigned to hardware and customized components. We, use a UML deployment diagram to diagrammatically illustrate the hardware/software mapping of our proposed system.

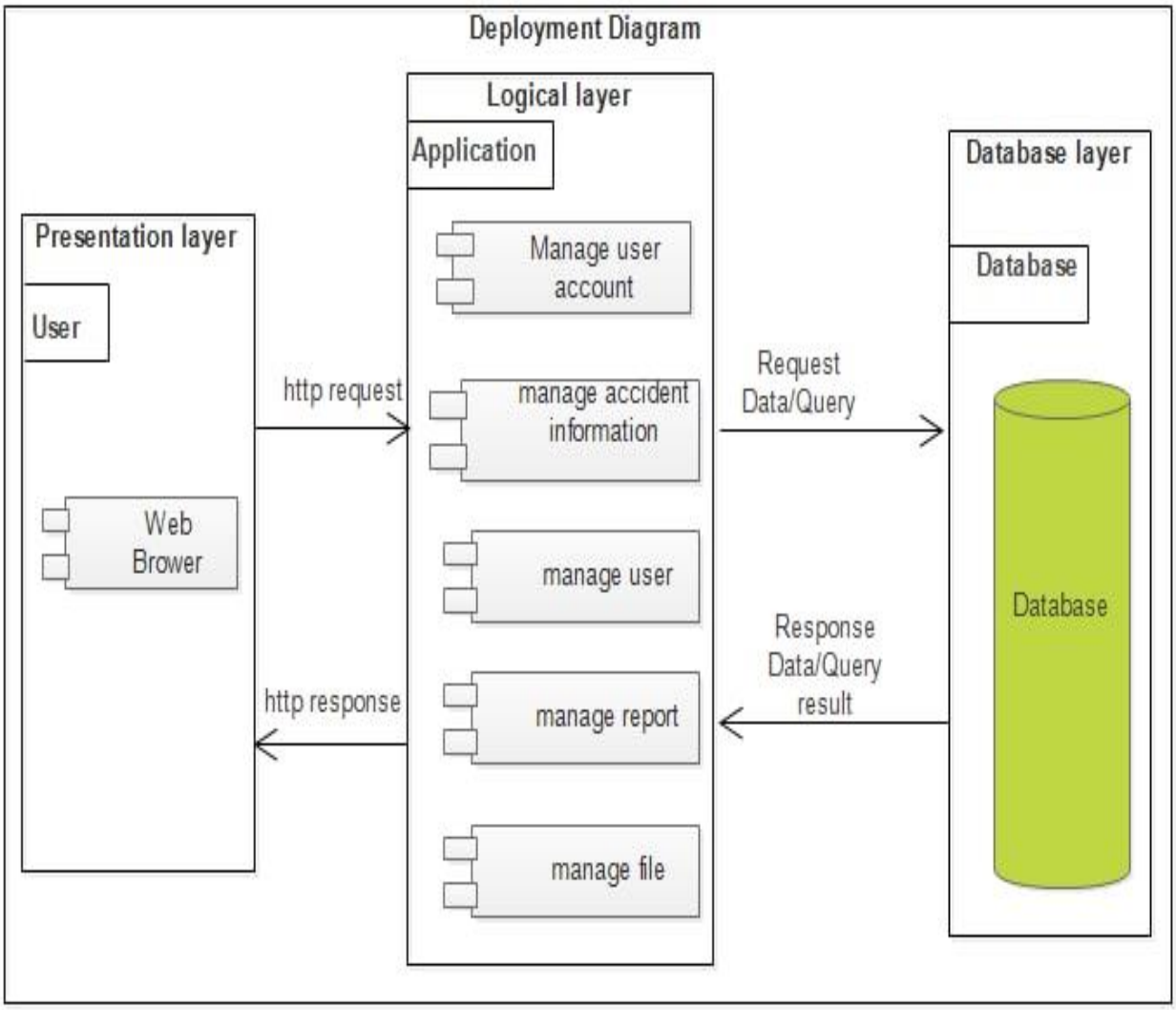


Figure 5.3 Hardware/Software mapping

5.4.3 Detailed Class Diagram

The class diagram shows the static structure of data and the operations that act on the data, i.e. it shows the static structure of an object-oriented model the object class, attribute, methods, and their relationships.

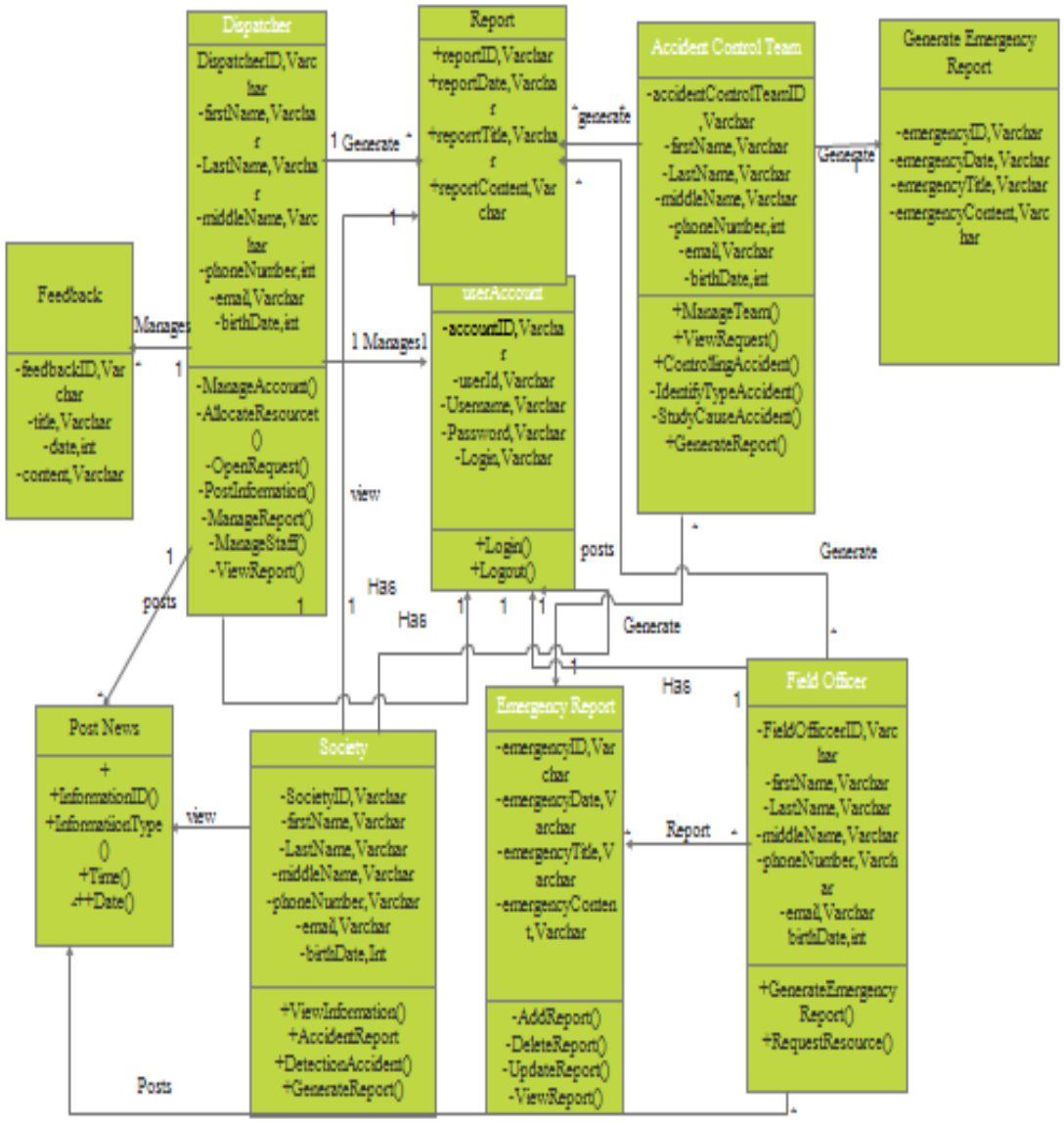


Figure 5.4 detailed class diagram

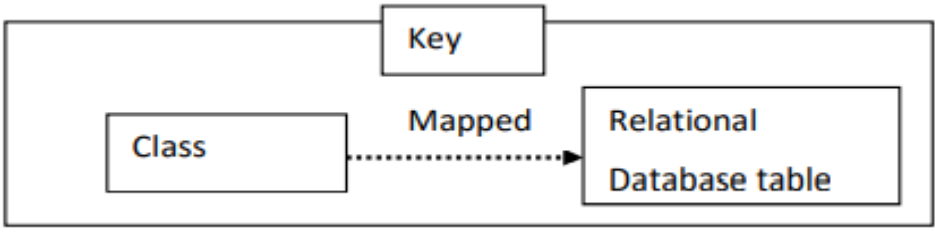
5.4.4 Persistent data management

Persistent data management deals with how the system is going to handle the actual data that needs to be stored on the database of the system. The purpose of persistence modeling is which objects in the system design are required to be stored persistently. Clearly, in a database-driven application like this one, almost all system interactions have to deal with persistent data. The online accident management systems will largely depend on a relational database to perform day-to-day operations and store long data.

Data will be stored in a MySQL Database Management system and manipulated through the Database Subsystem, which will ensure data integrity and consistency. Database Subsystem will contain all necessary SQL queries that will be accessible by the rest of the Subsystems. The data stored in the database will include:

- Emergency accident report
- User account
- Society
- Employ profile

Again, the purpose of this section is to show the mapping of the objects/classes of the system, identified during the analysis stage, into the corresponding relational database.



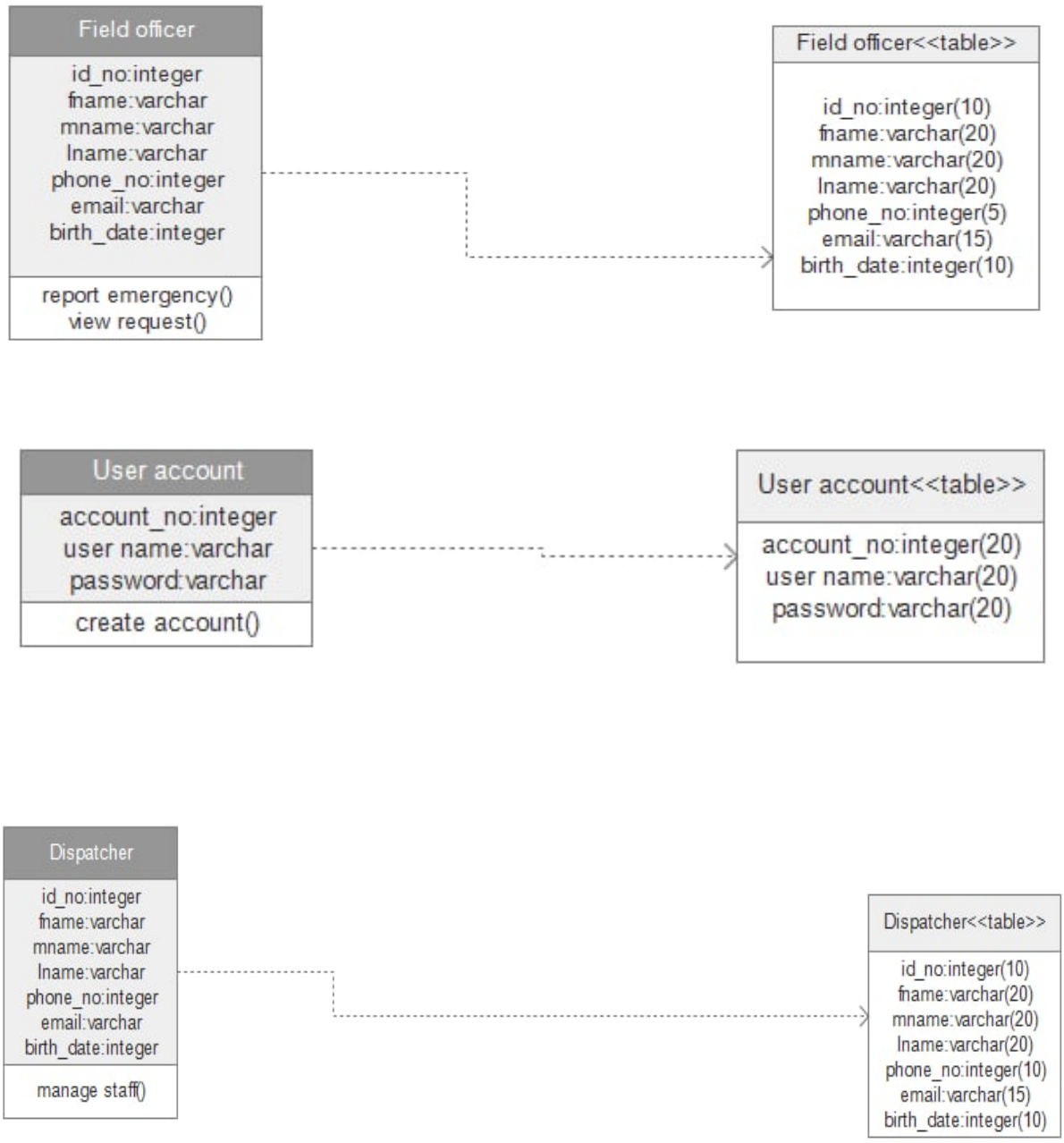


Figure 5.5 persistent data management

5.4.5 Access control and security

In this system, there are multiple actors which have access to different functions and data. The everyday actor may only access the data it creates, whereas a system administrator actor may have unlimited access to system data and to other users' data. Defining access control allows each actor to identify which operations they can access

on each shared object. This mechanism is implemented through username and password in order to restrict an unauthorized person from accessing the entire system.

	Dispatcher	Accident controlling team	Field officer	society
Login	✓	✓	✓	
Create Account	✓			
Update Account	✓			
Delete Account	✓			
View Account	✓			
Allocate resource	✓			
Open request	✓			
Post information	✓	✓	✓	
Manage report	✓			
View report	✓			
Add new file		✓		
View accident file		✓		
Update file		✓	✓	
Delete file		✓		

View request		✓		
Controlling accident		✓		
Identify types of accident		✓		
Study cause of accident		✓		
Generate report	✓	✓	✓	✓
View information	✓	✓	✓	✓
Accident report	✓	✓	✓	✓
Give Feedback	✓			
Generate emergency report		✓		
Change Password	✓	✓	✓	✓
Logout	✓	✓	✓	✓

Table 5.1 access control table and security

5.4 Package

The package is an organized and functionality-based set of related interfaces and classes. Packages organize classes that belong to the same category or provide similar functionality. In our system we category into database management package, manage account subsystem, manage user subsystem, manage account subsystem, manage information package.

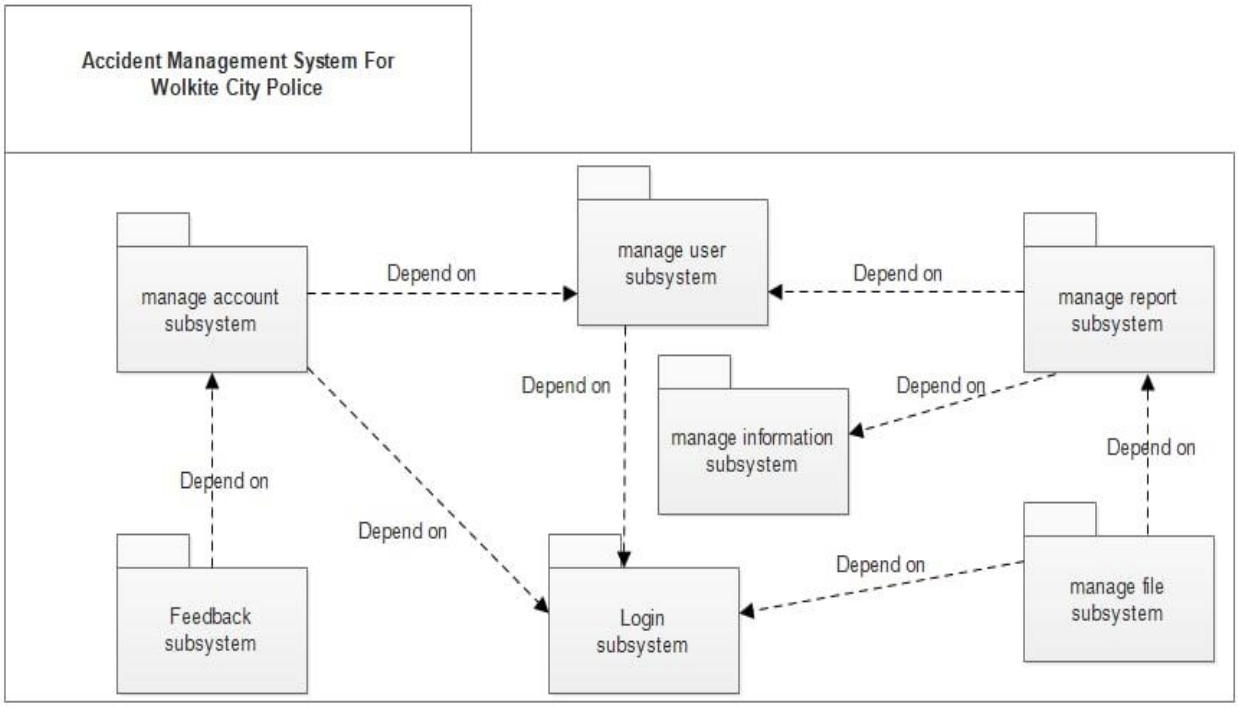


Figure 5.6 packages

5.5. Algorithm Design

Algorithms are designed to show the flow of programs in the system. Pseudo code is a method of planning which enables the programmer to plan without worrying about syntax. Algorithms show the flow and steps of logic in each function.

Pseudo code for Login

step1: BEGIN

step2: enter username and password on the given form.

Step3: IF username and password valid, THEN send to Database.

step4: IF username and password invalid, THEN go to step 2 and re-enter valid username and password.

Step5: Database check entered input

step5.1: IF valid, THEN logged in successfully

step5.2: IF invalid, THEN not logged in successfully and try again with correct username and password.

Pseudo code for Create user account

Step1: BEGIN

Step2: IF user click on create account link THEN Form is displayed

Step3: Fill information for the user

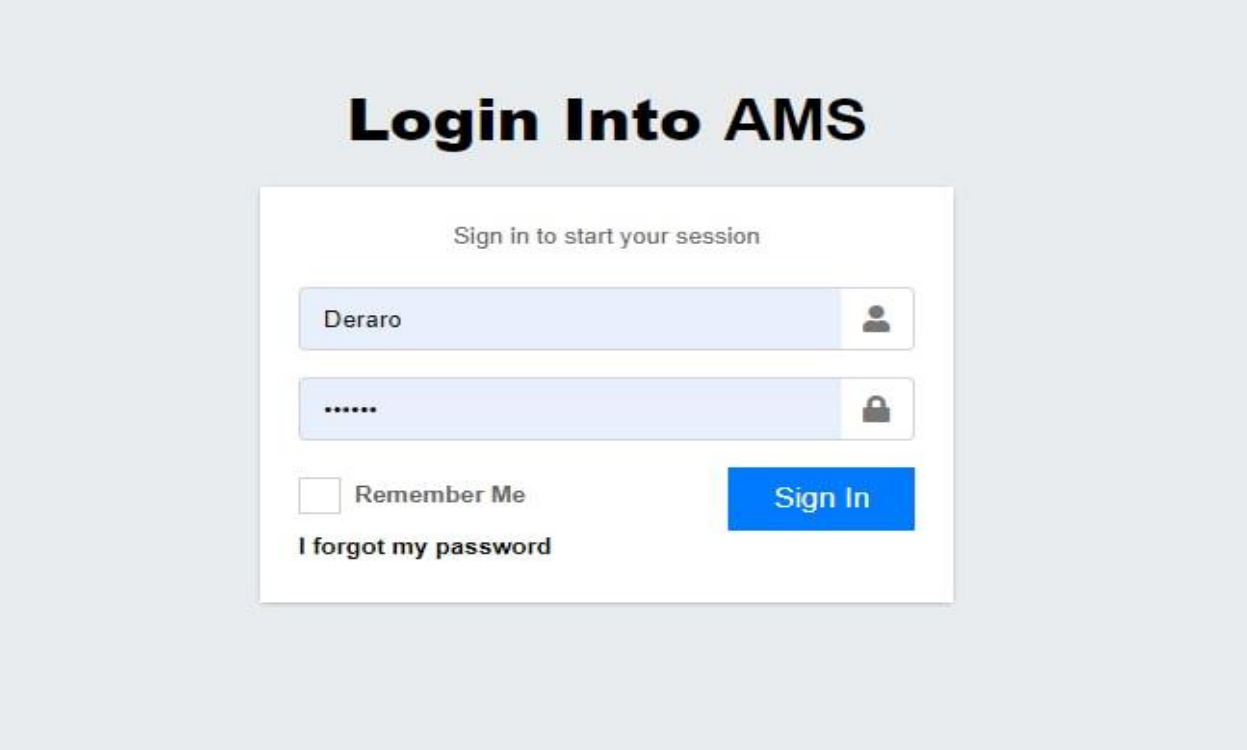
Step4: If filled data valid and user with this account is not exist, then add user account to Database

Step5: Display message user account successfully created.

Step6: IF filled data invalid or the user with this account is preexist, THEN go to step3 and re-enter valid input, or

Step7: ENDIF

5.6. User Interface Design



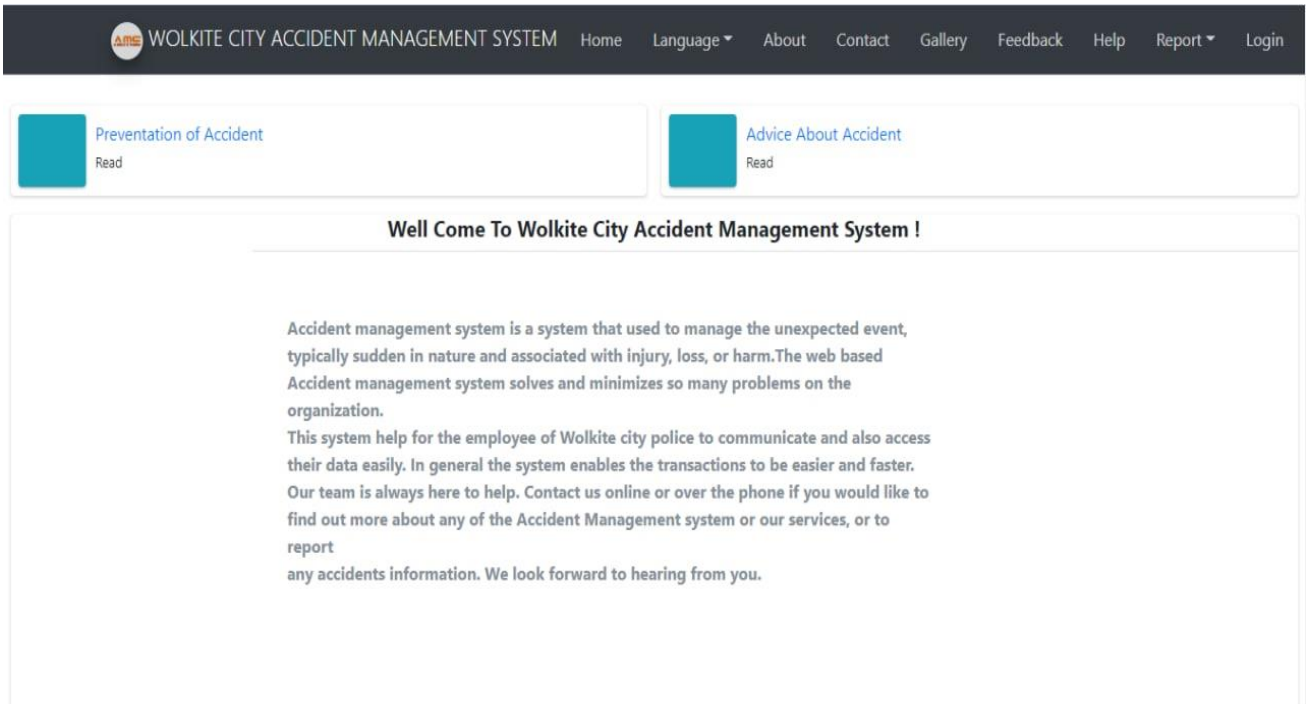


Figure 5.7 user interface design

CHAPTER SIX

6. IMPLEMENTATION AND TESTING

This chapter basically highlights the issues deal with the implementation phases. Implementation is the phase where objectives of physical operations of the system turned into reality i.e. real working model. In this phase the coding convention has made it possible as it's the real phase of objectivity to reality. Then the code is tested until most of the errors have been detected and corrected. The goal of implementation is to introduce our system for the users in real sense that how they use this new system which is developed for their intended objectives

6.1 Implementation of the Database

We have used MYSQL database management system for the implementation of the database. Because MYSQL is most stable secure, reliable and higher performance than the rest; it takes lower time for transactions (accessing and processing) data in a database which means it's faster to process transactions. MYSQL reveals that it can perform efficiently regardless of its workload. After selecting MYSQL database management system, we performed the following activities:

- ✓ We have created all tables which were identified and shown as persistent model. In the design document with their primary keys, foreign keys.
- ✓ We configured a schedule of database backup.
- ✓ We configured WKCAMS end database level security

6.2. Implementation of the Class Diagram

In this class diagram implementation we have implement the class diagram structure. That implements the view of an application, visualizing, describing and documenting different aspect of the system. And also implements the attribute and operation of the class.

6.3. Configuration of the Application Server

We use XAMPP server because XAMPP is simple, lightweight Apache distribution it is extremely easy to create a local web server for testing and deployment purpose. Everything you needed is to set up a web server – server application (Apache), database and scripting language XAMPP works equally well on Linux,Mac and Windows. Since it is suitable and the function we listed above we use XAMPP application server

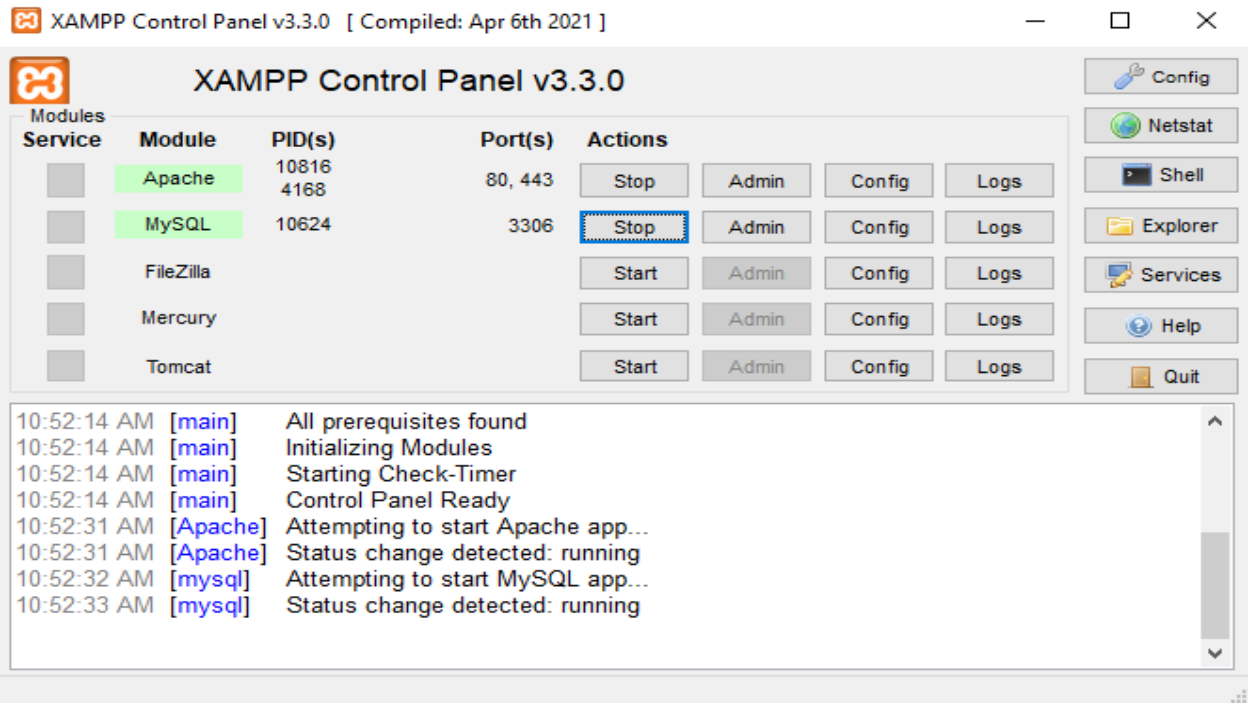


Figure 6.1. Configuration of the Application Server

6.4. Configuration of Application Security

Our system called Accident management system for Wolkite city Police is an issue validates all the inputs by returning error message and suggesting to try again when invalid input occurred. We implement encryption for user password by MD5 when the system admin creates a user account for manager, receptionist and others create their account or the user changes their password system encryption the password. The system has its security principles that control unauthorized authentication.

6.5. Implementation of User Interface

User interface for Accident Management system have been implemented in the following way. The user interface is user centered design (Place users in control of the interface) and the user interface is consistent.

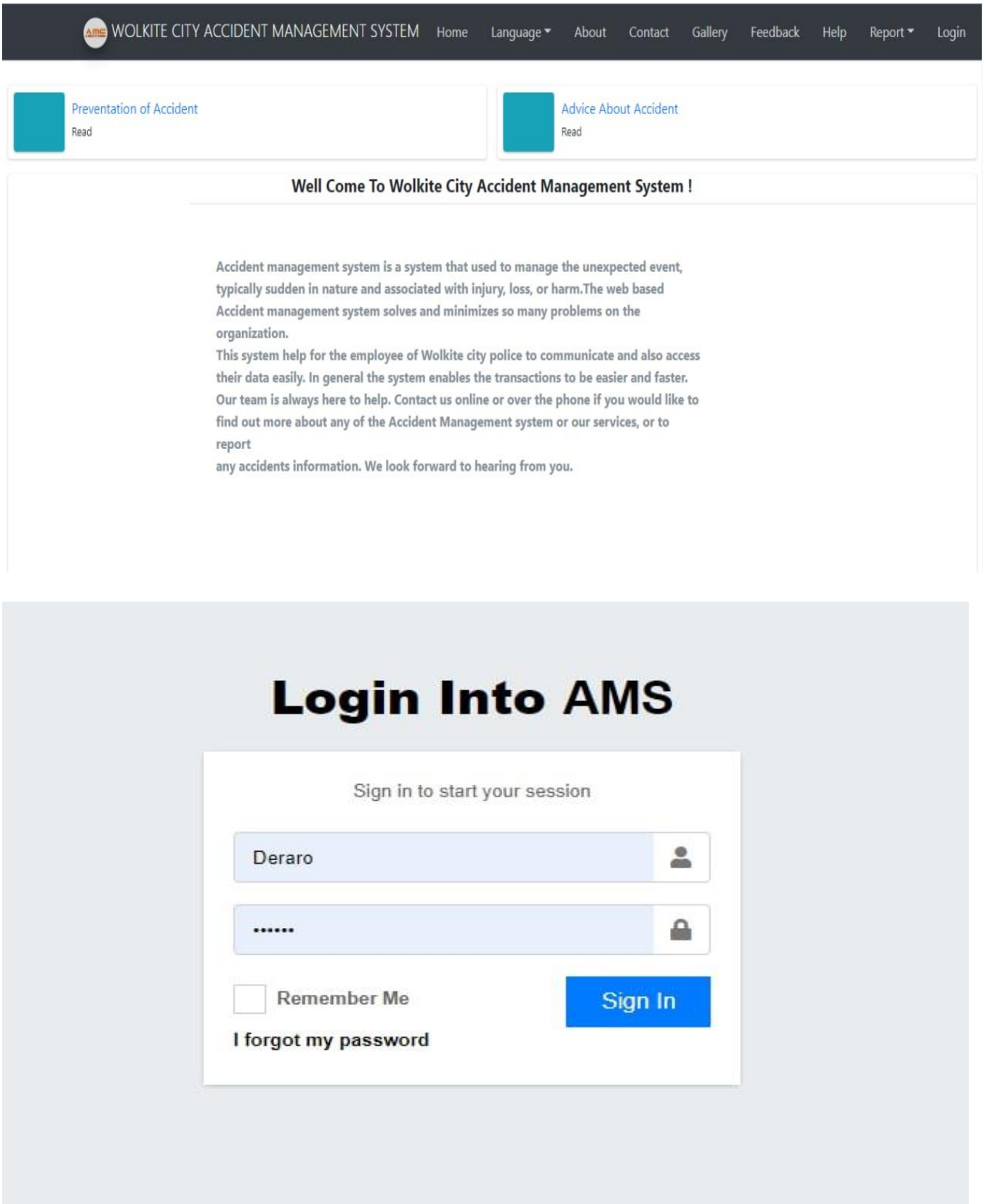


Figure 6. 2. Create User and assign Role

6.6. Testing

Testing evaluates a software product to ensure that it satisfies its planned purpose. A test that is modified to and consistent with development methodologies provides an observable and structured approach to verifying requirements and quantifiable performance

- **Features to be tested**

Criteria are standards by which we evaluate our systems that help us to determine whether a test case passes or fail.

- ✓ Fail Criteria: when the system does not meet the all specific requirements of the system and if the test case is said to fail the expected result is not satisfied by the system that relates with its functionality.
- ✓ Pass Criteria: when the system meets the all specific requirements of the system and if the test case is said to pass the expected result is satisfied by the system that relates with its functionality.

When we test our project, we perform following testing.

- **Sub system communication: -** Our system is decomposed into different module because of module system easily managed and error easily detected and fixed. But communication between them should be tested.
- **Input output functions: -** Check what type of our system should be taken as input and produce as output and checking input is produce expected output.
- **Graphical user interface: -** We consider standard position of interface, standard of interface by comparing with previously performed system and criteria written for graphical user interface.
- **Database transaction: -** Ensuring transaction of our system database.
- **Security: -** Identifying security of our system by identifying only identified user allowed access our system and ensuring password of each individual user not seen by another. This thing is the thing we consider on login capability.
- **User interface and database interaction: -** Database is one that store data and user interface one that user data is entered, so during our test we try to identify data entered on user interface stored in database and other crud.

- **Features not to be tested**

Features not to be tested includes the future of the system which cannot be measured directly or indirectly. There features cannot make serious damage to system but indirectly have an influence on our system performance and acceptance. These features include:

- Determination of exact response time of our system.
- Exact measure of user satisfaction of the system

6.6.1. Test Case

It is a set of actions executed to verify a particular feature or functionality of your software applications.

- ✓ Test Case 1: Check result on entering Valid User ID & Password
- ✓ Test Case 2: Check result on entering Invalid User ID& Password
- ✓ Test Case 3: Check response when a User ID is empty & Login Button is pressed, and many more

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly

Input	Expected Result	Actual outcome	Pass/Fail
Valid User Name and Valid Password	The user logs to the system successfully	The user logs to the system successfully	Pass
If inserted only user name, Without Password	The system displays an error message “please enter password”	The system displays an error message “please enter password”	Fail
If the user name or password Text Box or both are not filled	The system displays an error message “please fill user name and password”	The system displays an error message “please fill user name and password”	Fail

Table 6. 1: Test case

6.6.2 Testing Tools and Environment

Testing tools are important for success of testing phase and naturally the success of product. In our unit testing phase PhpStorm and HeidiSQL are used.

- **PhpStorm Debugger**

PhpStorm provides an editor for PHP, HTML and JavaScript with on-the-fly code analysis, error prevention and automated refactoring's for PHP and JavaScript code. By tracing and correcting bugs and errors our system's stability is increased. Since we use this tool it is easy to keep tracking any mistakes especially in unit testing phase.

- **Hardware testing tools**

✓ Computer with windows platform

- **Software's testing tools**

✓ Operating system
✓ Windows 10
✓ Web browser
✓ Emulator

6.6.3. Unit Testing

The main objective of unit testing to ensure that each individual part is working well and as it's supposed to work. The entire system will only be able to work well if the individual part are working well. Unit testing is performed by software developers themselves. In this level of testing process, the OHR system developers test different sub procedures, functions and tested by applying the black and white box testing.

✓ Cheek whether the return type of the function is correct.
✓ Cheek how the sub procedure or function are call correctly.
✓ Cheek if the correct output is produced for different inputs.

6.6.4. System Testing

The test performs on the whole programs to ensure the whole functional and non-functional requirement specified on the system and also to decide the completeness of the system as a system that full fill perfectly the whole functional and non-functional requirements of the system.

The goal of the system testing phase is to evaluate all what we have been implemented in the previous development phase and to find errors in order to make corrections on the errors and these cannot occur again. But in our case well focus only on function validation and performance.

Sample Test

- ✓ Measure system boundary which is beyond the goal or not.
- ✓ Measure the weakness and the strength of the system using different metrics.
- ✓ Evaluate the functionality of subsystem after combination of each subsystem weather it is works.
- ✓ Check the coherence and coupling of each subsystem.
- ✓ Check the overall functionality of our system that achieves the user's requirement.
- ✓ Check the interaction of each subsystem that performs the specified business process.
- ✓ Verify the system completeness-based customer's requirement.

6.8.5. Integration Testing

In this level of testing, we have examined how the different procedures work together to achieve the goal of the subsystem. The type of integration testing that we followed is bottom up. We integrate each component from single functionality (individual interface) to the main function incrementally step by step.

Our integration testing procedure is given below.

- ✓ Firstly, we created three users who have three role types namely, Field Officer, Accident control Team, and Dispatcher.
- ✓ Then we will log in with the user names and passwords of each user type.

This test whether the authentication mechanism works correctly.

Sample Test

- ✓ Identify the independency of each subsystem with other subsystem.
- ✓ Check the interaction of each functionality that performs the specified task.
- ✓ Evaluate the functionality of subsystem after combination all individual functionality.

6.8.6. Acceptance Testing

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users.

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1 Conclusion

The main purpose of this project is to establish a long lasting and effective communication between different users and also to introduce various features regarding Wolkite City accident management system. An effort has been made to study Wolkite City accident management system as partial fulfillment of BSC degree in computer science. In doing the study the team has tried to follow object oriented system analysis and design methodology. Since the success and failure of any system depends on gathering the right information through different fact-finding techniques and user involvements, the team has made the best effort to gather requirements. After a detail review and study of the existing system of Wolkite City accident management system models have been designed to reflect the new system that are supposed to solve problems. In order to solve different problems existed the team has tried to propose a solution that at least reduce the existed problems and model the proposed system using different tools and methodologies. The team believe the different tools and techniques has helped us a lot in capturing real user requirements and model the right system for the users for their day to day transactions. Thus it should have the precedence in know-how and experience in collecting, processing and utilizing information.

This system offers many advantages for the system users. The main focus of this project is to make Accident Management system for Wolkite city Police computerized and this will reduce happening of accident and any kind of accidents.

7.2. Recommendation

Based on the conclusion our group recommended that the Wolkite city Accident management system will better replace the existing system by new automated system. Because the old SMS and manual method of working is not measurable with today's working condition and it is not efficient. We mention Wolkite city Accident management system to use the developed system in order to facilitate the work process and to reduce todays Accidents and incidents.

References

[1] B. H. W. D. T. Alan Dennis, System Analysis &Design an Object-Oriented Approach with UML, WILEY, 2015.

[2] **The Class Diagram:** Available at: http://creately.com/blog/diagrams/understandingtherelationships-between-classes/ accessed on [Nov-Dec, 2017]

[3] **The Class Diagram**: http://www.leing.com/style/classDiagrmm.htm accessed on [Nov-Dec, 2017]

[4] "https://www.researchgate.net," [Online].

[5] **The Deployment Diagram:** Available at:

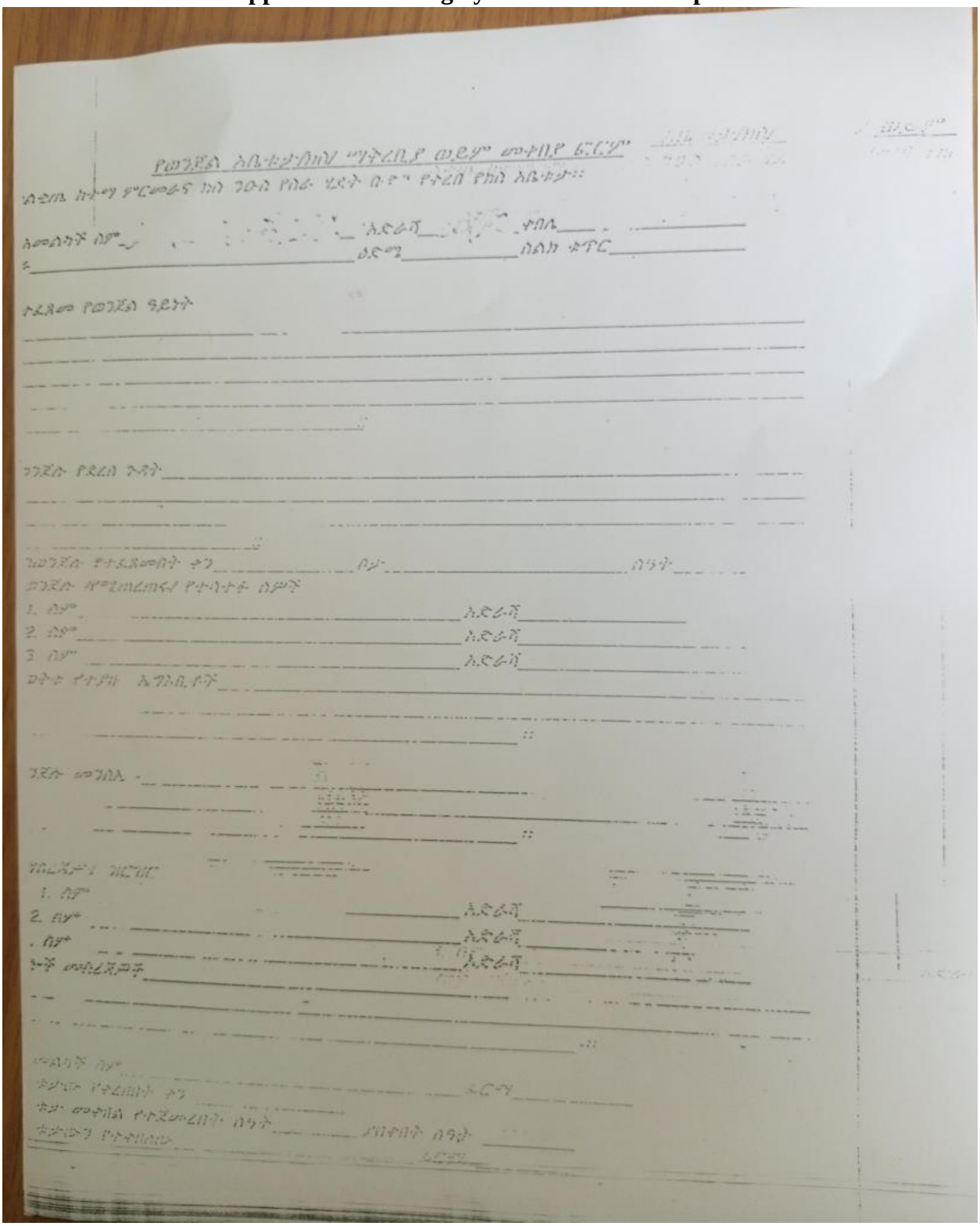
http://www.agilemodeling.com/artifts/deploymentDiagram.html.UBc38rho.dpuf accessed on [Nov-Dec, 2017]

[6] **The subsystem decomposition**: Available at: http://selab.hongik.ac.kr/show_data/education/F-ch06.pdf accessed on [Nov-Dec, 2017]

[7] Final WKU reviewed industrial project Guideline.

APPENDICES

Appendix I: Existing System Forms and Reports



Appendix II: Sample Source Code

```
CREATE TABLE dispatcher (
  Id int(11) NOT NULL,
  Firstname varchar(30) NOT NULL,
  Middlename varchar(30) NOT NULL,
  Lastname varchar(30) NOT NULL,
  Gender varchar(15) NOT NULL,
  Birthdate date NOT NULL,
  Phone int(13)  NULL,
  Email varchar(30)  NULL,
  UserName varchar(30) NOT NULL,
  UserRole varchar(30) NOT NULL,
  Password varchar(30) NOT NULL
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;

INSERT INTO disparcher (Id, Firstname, Middlename, Lastname, Gender, Birthdate, Phone,
Email,UserName,UserRole,Password) VALUES

(5, 'Dararo', 'Biriye', 'Kamure','Male', 12/25/1991, 0924781021,
'dararo@gmail.com','Deraro','dispatcher','dero21');

CREATE TABLE generateemergencyreport (
  Id int(11) NOT NULL,
  SenderName varchar(30) NOT NULL,
  Acidenttype varchar(30) NOT NULL,
  Acclocation varchar(30) NOT NULL,
  Accweight varchar(15) NOT NULL,
  Adate date NOT NULL,
  Atime int(13)  NULL,
  Ainformation varchar(100)  NULL
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;

INSERT INTO generateemergencyreport (Id, SenderName, Acidenttype, Acclocation, Accweight, Adate,
Atime, Ainformation) VALUES

(8, 'Dararo', 'fire', 'kebele 02','simple', 12/25/1991, 5, 'please');
```

```
CREATE TABLE fieldofficer (
  Id int(11) NOT NULL,
  Firstname varchar(30) NOT NULL,
  Middlename varchar(30) NOT NULL,
  Lastname varchar(30) NOT NULL,
  Gender varchar(15) NOT NULL,
  Birthdate date NOT NULL,
  Phone int(13)  NULL,
  Email varchar(30)  NULL
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;

INSERT INTO fieldofficer (Id, Firstname, Middlename, Lastname, Gender, Birthdate, Phone, Email)
VALUES
(1, 'Ebisa', 'Edessa', 'Meka','Male', 12/25/1991,0947534650 , 'ebisaedessa@gmail');

CREATE TABLE accidentcontrolteam (
  Id int(11) NOT NULL,
  Firstname varchar(30) NOT NULL,
  Middlename varchar(30) NOT NULL,
  Lastname varchar(30) NOT NULL,
  Gender varchar(15) NOT NULL,
  Birthdate date NOT NULL,
  Phone int(13)  NULL,
  Email varchar(30)  NULL
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;

INSERT INTO accidentcontrolteam (Id, Firstname, Middlename, Lastname, Gender, Birthdate, Phone,
Email) VALUES
(17, 'Atinaf', 'Lamessa', 'Ayana','Male', 12/25/1991,0947534650 , 'atinaflamessa@gmail');

CREATE TABLE Society (
  Id int(11) NOT NULL ,
  Firstname varchar(30) NOT NULL,
  Middlename varchar(30) NOT NULL,
  Lastname varchar(30) NOT NULL,
```

```
  Gender varchar(15) NOT NULL,
  Birthdate date NOT NULL,
  Phone int(13)  NULL,
  Email varchar(30)  NULL
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
INSERT INTO Society (Id, Firstname, Middlename, Lastname, Gender, Birthdate, Phone, Email)
VALUES
(18, 'Gamshu', 'Satessa', 'Garo','Male', 12/25/1982,094753462 , 'gamshusatessa@gmail');
CREATE TABLE  Useraccount (
  Id int(11) NOT NULL,
  UserName varchar(30) NOT NULL,
  Password varchar(30) NOT NULL,
  UserRole varchar(30) NOT NULL
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
INSERT INTO Useraccount (Id, UserName, Password, UserRole) VALUES
(19, 'Deraro', 'pass123', 'Dispatcher' );
CREATE TABLE  Useraccountfieldoff (
  Id int(11) NOT NULL,
  UserName varchar(30) NOT NULL,
  Password varchar(30) NOT NULL,
  UserRole varchar(30) NOT NULL
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
INSERT INTO Useraccountfieldoff (Id, UserName, Password, UserRole) VALUES
(19, 'Deraro', 'pass123', 'Dispatcher' );
```