# **DECLARATION**

I, SHEMA Loic, hereby declare that this project entitled “Ibuka E-Library” is the work that I have spent lot of time working on and this work has not received any precious credit at Adventist University of Central Africa or somewhere else in any other institution.

Student Supervisor

SHEMA Loic BATAMULIZA Jennifer

Signature: ……………… Signature: ………………

Date:…../……./……… Date:……/……/………

# **DEDICATION**

To the Almighty God

To my beloved parents

To Ndagijimana Yves

To my supervisor for his guidance

To all my friends and relatives.

# **LIST OF ABBREVIATIONS**

**AUCA:** Adventist University of Central Africa

**DBMS:** Database Management System

**ICT:** Information and Communication Technology

**IDE:** Integrated Development Environment

**ISO:** International Organization for Standardization

**IT:** Information Technology

**OOAD:** Object-Oriented Analysis and Design

**OOD:** Object-Oriented Design

**OOM:** Object-Oriented Methodology

**OOP:** Object-Oriented Programming

**SDLC:** Software Development Life Cycle

**SDP:** Software Development Process

**SEP:** Software Engineering Process

**UML:** Unified Modeling language

**UP:** Unified Process

# **DEFINITION OF TERMINOLOGIES**

The following are the specific terminology in the current system in order to be well understood:

* **System:** set of components that work together to achieve a common goal.
* **Management Information System:** is an organized combination of people, hardware, software, communication networks and data resources that collects, transforms and disseminates information in an organization.

# 

# **ACKNOWLEDGEMENT**

My praise and sincere thanks go to the Almighty God who blessed me with His grace and mercy throughout this whole project.

I sincerely have deep recognition to my supervisor **BATAMULIZA Jennifer** who guided me in the accomplishment of this work, especially for his substantial advice, professional assistance, guidance and precious ideas.

I thank the authorities of Adventist University of Central Africa as well as the staff of department of Information Technology.

Special thanks to my family, colleagues and friends for their support, encouragement, friendship and to those who participated in one way or another in the accomplishment of this work.

God bless you all.

# **Table of Contents**

[**DECLARATION** 1](#_Toc39699249)

[**DEDICATION** 2](#_Toc39699250)

[**LIST OF ABBREVIATIONS** 3](#_Toc39699251)

[**DEFINITION OF TERMINOLOGIES** 4](#_Toc39699252)

[**ACKNOWLEDGEMENT** 5](#_Toc39699253)

[**Table of Contents** 6](#_Toc39699254)

[**List of Figures** 9](#_Toc39699255)

[**List of Tables** 10](#_Toc39699256)

[**CHAPTER I** 11](#_Toc39699257)

[**GENERAL INTRODUCTION** 11](#_Toc39699258)

[**BACKGROUND OF THE STUDY** 11](#_Toc39699259)

[**PROBLEM STATEMENT** 12](#_Toc39699260)

[**MOTIVATION OF THE STUDY** 12](#_Toc39699261)

[**OBJECTIVES OF THE STUDY** 13](#_Toc39699262)

[**GENERAL OBJECTIVE** 13](#_Toc39699263)

[**SPECIFIC OBJECTIVES** 13](#_Toc39699264)

[**SCOPE OF THE STUDY** 13](#_Toc39699265)

[**EXPECTED RESULTS** 13](#_Toc39699266)

[**Methodology and Techniques used in the study** 14](#_Toc39699267)

[**Documentation** 14](#_Toc39699268)

[**Observation** 14](#_Toc39699269)

[**Structure of the work** 14](#_Toc39699270)

[**CHAPTER 2** 16](#_Toc39699271)

[**ANALYSIS OF THE EXISTING SYSTEM** 16](#_Toc39699272)

[**Introduction** 16](#_Toc39699273)

[Description of the Current System Environment 16](#_Toc39699274)

[**Analysis of the Current System** 17](#_Toc39699275)

[**Modeling Current System** 17](#_Toc39699276)

[**Process of the Existing System** 17](#_Toc39699277)

[**PROBLEMS STATEMENT** 17](#_Toc39699278)

[**PROPOSED SOLUTIONS** 18](#_Toc39699279)

[**FUNCTIONAL REQUIREMENTS** 18](#_Toc39699280)

[**NON-FUNCTIONAL REQUIREMENTS** 19](#_Toc39699281)

[**CHAPTER 3** 21](#_Toc39699282)

[**REQUIREMENTS ANALYSIS AND DESIGN OF THE NEW SYSTEM** 21](#_Toc39699283)

[**Introduction** 21](#_Toc39699284)

[**Analysis and Design Methodology** 21](#_Toc39699285)

[**Object-Oriented Methodology (OOM)** 21](#_Toc39699286)

[**Object-oriented analysis and design (OOAD)** 22](#_Toc39699287)

[**Object-oriented analysis** 22](#_Toc39699288)

[**Object-oriented design** 23](#_Toc39699289)

[**Object-oriented programming (OOP)** 23](#_Toc39699290)

[**UML Concepts** 23](#_Toc39699291)

[**Software development process** 27](#_Toc39699292)

[**Unified Process (UP)** 27](#_Toc39699293)

[**Analysis of the new system** 28](#_Toc39699294)

[**Design of the new system** 28](#_Toc39699295)

[**Use Case Diagram** 28](#_Toc39699296)

[**Class Diagram** 35](#_Toc39699297)

[**Class Diagram** 36](#_Toc39699298)

[**Sequence diagram** 36](#_Toc39699299)

[**Database Schema Diagram** 43](#_Toc39699300)

[**Architectural Design** 44](#_Toc39699301)

[**CHAPTER 4** 45](#_Toc39699302)

[**IMPLEMENTATION AND TESTING OF THE NEW SYSTEM** 45](#_Toc39699303)

[**Introduction** 45](#_Toc39699304)

[**Technologies used** 45](#_Toc39699305)

[**NetBeans IDE** 45](#_Toc39699306)

[**Visual Paradigm** 45](#_Toc39699307)

[**Graphical interfaces of the IBUKA E-LIBRARY** 47](#_Toc39699308)

[**Software Testing** 54](#_Toc39699309)

[**CHAPTER 5** 55](#_Toc39699310)

[**CONCLUSION AND RECOMMENDATION** 55](#_Toc39699311)

[**Conclusion** 55](#_Toc39699312)

[**Recommendations** 56](#_Toc39699313)

[**REFERENCES** 57](#_Toc39699314)

# **List of Figures**

[Figure 1: Overview of the current system 13](#_Toc39698321)

[Figure 2: Use case diagram 26](#_Toc39698322)

[Figure 3: Class diagram 32](#_Toc39698323)

[Figure 4 : Create account sequence diagram 34](#_Toc39698324)

[Figure 5: Login sequence diagram 35](#_Toc39698325)

[Figure 6:Registering a victim 36](#_Toc39698326)

[Figure 7: Claiming sequence diagram 37](#_Toc39698327)

[Figure 8: Approving claim sequence diagram 37](#_Toc39698328)

[Figure 9: Viewing sequence diagram 38](#_Toc39698329)

[Figure 10: Search information sequence diagram 38](#_Toc39698330)

[Figure 11: Architectural Design 40](#_Toc39698331)

[Figure 12: Signup page 43](#_Toc39698332)

[Figure 13: Login Page 44](#_Toc39698333)

[Figure 14: Dashboard page 44](#_Toc39698334)

[Figure 15: Add Book Page 45](#_Toc39698335)

[Figure 16: Add Victim page 45](#_Toc39698336)

[Figure 17: Add Testimony page 46](#_Toc39698337)

[Figure 18: Add Video page 46](#_Toc39698338)

[Figure 19: Add Speech page 47](#_Toc39698339)

[Figure 20: Victim List page 47](#_Toc39698340)

[Figure 21: President Dashboard page 48](#_Toc39698341)

[Figure 22: President Claim List page 48](#_Toc39698342)

[Figure 23: Client Read Book page 49](#_Toc39698343)

[Figure 24: Client’s Victim List view page 49](#_Toc39698344)

# **List of Tables**

[Table 1: Use-case diagram elements 26](#_Toc39698430)

[Table 2: Create account use-case description 28](#_Toc39698431)

[Table 3: Login use-case description 29](#_Toc39698432)

[Table 4: Register Data use-case description 29](#_Toc39698433)

[Table 5: Search Data use-case description 30](#_Toc39698434)

[Table 6: Donate funds use-case description 30](#_Toc39698435)

[Table 7: Claim Victims use-case description 31](#_Toc39698436)

[Table 8: Approve Claim use-case description 32](#_Toc39698437)

[Table 9: Sequence diagram elements 35](#_Toc39698438)

[Table 10: Technologies used 42](#_Toc39698439)

# 

# **CHAPTER I**

# **GENERAL INTRODUCTION**

In order to achieve Rwanda’s long term goals for the development of the country, Rwanda needs some improvement in giving good services, innovations, use of technology, and so on. For those reasons, this project will focus on the use of technology by providing good services specifically in the field of history management.

As Rwandans we must learn from our past to build our future, Hence the Ibuka E-Library which will help the whole world to know our Rwandan history about the Genocide against the Tutsi so that it must never happen again elsewhere in the world and to prove for those Genocide denials the facts that it happened.

# **BACKGROUND OF THE STUDY**

Most of the world agree that only 2 genocides happened the Jewish one and the Armenian one, but decline the Genocide against the Tutsi. The issue comes when people start saying that it was a massacre between ethnic groups and also that it was not only the massacre of Tutsi but also Hutu but according to the reality the genocide was intended to eradicate Tutsi.

The genocide victims are killed or eliminated not because they are directly involved in fighting or wars but because of their race and other discriminatory criteria. The perpetrators are often the ones who dictate or decide the criteria on which they based upon when they want to eliminate a certain group of people or ethnic. (CNLG,2019)

IBUKA is a high profile lobby group with a particular interest in addressing justice for survivors and coordinating/leading joint survivors’ projects on a national level

The vision of IBUKA is that Rwandan society should be a place where the memory of the genocide is preserved and where all the genocide survivors are socially included, financially able and live with full dignity. ( Survivors Fund,2008).

# **PROBLEM STATEMENT**

* Because of using papers, books and tapes there is a risk of to be lost, to be burnt, to be stolen or damaged. No enough Security.
* Because of lack of information guides may report false information to visitors of a memorial site.
* Manual system cause loosing data in case of war, fire, accidents etc.
* Many Nations deny the Genocide against the Tutsis because some don’t know Rwanda others because they want to deny their role in it others because they don’t get enough evidence.
* No online archives for the speeches ,events ,testimony for the Genocide against the Tutsis.
* No acting proof for Genocide denials.
* Difficulty in data keeping due to it requires much space for keeping them.
* Difficulty in knowing unidentified victims/bodies because the survivors or family related to the victims maybe abroad.

# **MOTIVATION OF THE STUDY**

The motivation of the study will identify and key out how the advantages of the resulted system will affect development teams positively There will be numerous benefits which must answer some questions like:

* Why is the Study important?
* To whom is it beneficial?
* What are the advantages after the study is conducted?

Those benefits are categorized according to the audience:

**Personal Interest:** it is an opportunity to put in practice the knowledge gained throughout my academic training at the Adventist University of Central Africa by solving problems and improving services and technology.

**For Ibuka:** The developed system will help them in their data manipulation including all archives.

**For my country:** The developed system will make people recognize Rwanda and the Visit Rwanda trademark.

**For my school:** This study will increase the literature of the institution which will be used for the students as a reference for their projects, especially it will serve as a basis to the ones who will deal with related project topics. For academic purposes, it will be used for evaluating if the knowledge provided was well acquired. For researchers visiting from outside the study will provide material and reference for their research projects.

# **OBJECTIVES OF THE STUDY**

## **GENERAL OBJECTIVE**

The general objective to develop this **IBUKA E-LIBRARY** is to develop a web application that will be used as a tool to help Ibuka and their section by resolving the different problems concerning archive management, and it will help the section to carry out its operations quickly.

## **SPECIFIC OBJECTIVES**

The specific objectives of the proposed project are listed below:

The system will help visitors minimize and use their time efficiently.

The system will digitalize the Ibuka Library System.

The system will help to identify unidentified bodies.

The system will help the world to reach all information about Genocide against Tutsis.

# **SCOPE OF THE STUDY**

This project will only contain Ibuka archives like videos, images, and a list of Genocide victims and also Ibuka past events. The system will be accessed using Android Mobile Phones and the internet.

# **EXPECTED RESULTS**

After Solving different problems of Ibuka, these are the expected results:

* Security of Ibuka Archives
* Easier Stock Management
* Decrease in the number of Genocide denials
* Decrease in the number of unidentified bodies
* Increase of Genocide memorials visitors
* A platform where you can have information about all Ibuka upcoming events

# **Methodology and Techniques used in the study**

Research methodology describes how to conduct research. Many facets are involved in conducting research. To understand deeply the requirements and the problem domain, some techniques will help to achieve the aim of this research. Data can be gathered from a number of sources, which include documents, the workplace, the Internet search, field notes, questionnaires, and social interaction or interviews.

To collect data on the existing Ibuka management system, the following techniques were used:

## **Documentation**

Documentation involves systematic data collection from the existing record, such as books, and Websites and so on. This technique was used in this study while consulting a Genocide memorial site.

The analysis of the existing system from information gathered from a review of the documents helped to identify problems and help to assess improvements needed to correct the current system of internal emergency communication.

## **Observation**

Observation is a process of recording the behavior of people, object and occurrences without questioning or communicating with them. This technique was used and found that Ibuka doesn’t have a website for their information.

# **Structure of the work**

This study is composed of the following five chapters:

Chapter oneis the general introduction of the study, highlights in the details the description of the work, it includes an introduction, background of the study, problem statement, motivation, objective of the project, methodology used, requirements collection techniques, scope of the project and expected results.

Chapter twois the requirements analysis of the study, highlights in details the analysis of the existing information system is currently positioned, the descriptions of the models and the weakness of the existing system and define the requirements of the new system, functional and non-functional requirements, use-case specifications, logical data model, system overview and the benefits of the proposed system.

Chapter threeis the system design of the study, highlights in detail the high-level architecture design, describing the modules description and solution design describing the process model, physical data model, class diagram, use-case diagram, and user interface diagram of the new system.

Chapter fouris the system implementation of the study, highlights in the details the tools and technologies used to develop a system, coding, and testing of the new system.

Chapter fiveis the final that holds the recommendations and conclusion related to the results of the project.

# **CHAPTER 2**

# **ANALYSIS OF THE EXISTING SYSTEM**

## **Introduction**

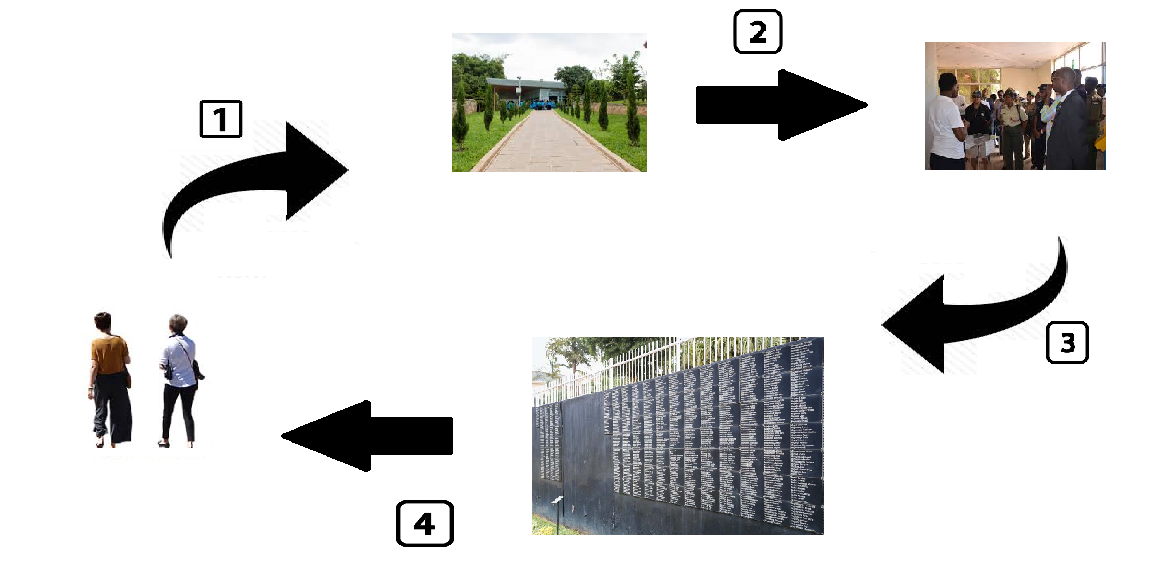
This part is meant to provide a brief description of terms that are being used during the development of the project. After a brief presentation of the **Ibuka E-Library** is given, subsequently, a thorough analysis of the existing system like working principle, problems within the system, all its features and characteristics which are relevant to our upcoming work are also described. Before the end of this chapter, the proposed solution to solve the mentioned problems will be highlighted.

# Description of the Current System Environment

This current system uses guides to explain all informations where by visitors are given a brief description of the genocide against Tutsi 1994 in Rwanda. Mostly, those guides provide brief details such as the theoritical information of what happened within the genocide tragedy. However, they do not provide enough and outstanding details such as proofs including videos and other different materials which are relevant and very crucial to the visitors concerned about the Ibuka memorial site. The visitors mostly get to know few things of what happened during that time and go back with a small package which does not contribute a lot to the changes that they would aspire to make in the country as a whole. Moreover, there are some dead bodies found after a long time by those who are in charge of it, however because the relatives of those people are living outside the country it becomes very hard to honor respect to their beloved ones who have passed away. Citizens living outside the country are still facing this challenge of not being able to communicate and collaborate with the Ibuka memorial site so that their beloved ones possibly receive a kind of respect and love that they deserve. The Ibuka has not yet been able to tackle that hindering problem that the citizens living outside the country are facing. Those who are living outside the country feel helpless with this problem as there is a need to make it work out for them striving for the respect that each Rwandan citizen deserve.

## **Analysis of the Current System**

### **Modeling Current System**

****

**Figure 1: Overview of the current system**

### **Process of the Existing System**

1. A visitor goes to any memorial site that the user wants.
2. A visitor is given an overview explanation on Genocide
3. A visitor is explained how it was planned and how it was put into actions and shown some statements and shown a list of victims that died in Genocide.
4. A visitor goes back.

After a close analysis of the current system description in the section above it is clear that there are problems and inefficiencies in the current system as it is. In the preceding section the current problems discovered from our research are outlined.

# **PROBLEMS STATEMENT**

According to the analysis of the current system the process of keeping information was difficult because there was a room with many videotapes that are not nowadays commonly used. This means there was a lot of damages to information which may be needed for future use.

There was also a failure in recognizing the found bodies because their relatives are abroad or some have been killed or no remaining relatives that can identify them.

There was also a difficulty in proving to the world the history of what happened in Rwanda and the fact of how the Genocide against the Tutsi was planned.

# **PROPOSED SOLUTIONS**

After analysing the Problems IBUKA faces the **IBUKA E-LIBRARY** will implement the

following solutions:

* In the proposed System, there will be safety in storing data for proper keeping and maintaing data because like cds and video tapes, there is a high damage

probability when kept for a long period of time.

* In the Proposed System, there will be an archive of all Kwibuka events, the

testimonies, and list of victims of a memorial site.

* In the Proposed System, Genocide denials will encounter proof that Genocide in

Rwanda happened due to testimonies and viewing all victims of a memorial site

* In the Proposed System, there will be easy data manipulation.
* In the Proposed System, there is an easy way to register victims because they are unidentified victims.

# FUNCTIONAL REQUIREMENTS

* The system must only allow user with valid username and password to enter the system.
* The system must record a list of victims in the database.
* The system must record tapes, speech and testimony in the database.
* System must be able to verify information
* System must be able to delete information if information is wrong or if it is unwanted, and the deleted should be backed up.
* System must be able to search a victim details in the database based on select search type.
* System must be have a form to register a victim information in the database that must be .first verified.
* System must be able to retrieve the information from the database.
* System should be able to add detailed information about any change.
* System should be able to display information on notice board available in the homepage of site.
* System must be able to do a backup.

# NON-FUNCTIONAL REQUIREMENTS

* Maintainability:
  + - The system should be easy to maintain, once it’s needed.
* Security:
  + - The system must be able to hide the users information
    - Only Admin can approve the victim request to be in their database because it may be a spam/false information.
    - Every user can Access with their authority
    - The system includes all available safeguards from viruses, worms and Trojans Etc.
    - The system will close the session when 30 minutes elapse while the logged in user does not perform any action
* Operational:
  + - The system should be able to run on any OS
* User friendly:
  + - The System will be user friendly
    - The system must be easy for a user to use.
* Privacy:
  + - The system shall be able to protect the users privacy.
* Availability:
  + - The system should have high availability
    - The system should not have unexpected downtime
    - The system should have downtime at most 1 hour/month.
* System should be able to display success message when registration is successful or error message when there are some errors.
* System should provide a user friendly environment for the users.

# **CHAPTER 3**

# **REQUIREMENTS ANALYSIS AND DESIGN OF THE NEW SYSTEM**

## **Introduction**

The objective of this chapter is to make retailed review and deepened in the goal to understand the current system and to surround the failing of it in order to propose an adequate computerized solution.

**System analysis** in software development is the interdisciplinary part of science, dealing with the analysis of sets of interacting entities, the systems, often prior to their automation as computer systems, and the interactions within those systems. Thus, System Analysis is the process of investigating a system, identifying problems, and using the information to recommend improvement to the system. (Dennis Alan, 2012)

**System design** in software development is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specific requirements. System design is done from the study of the existing system in order to determine what changes will be needed to incorporate the user needs that were not met by the existing one. The output of this phase will consist of the specifications, which must describe both what the proposed system will do and how it will work actually defining the solutions. (Bell, Douglas, 2005)

# **Analysis and Design Methodology**

## **Object-Oriented Methodology (OOM)**

There are many system development approaches that can be used in the development of software. But we, we are going to focus on the Object-Oriented which uses a language called Unified Modeling Language (UML) to describe the system concept as a collection of objects. Incorporating data, process and the Unified Process (UP) will be also used for our software.

The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering. It provides a set of graphic notation techniques to create visual models of object-oriented software-intensive systems. It was developed by Grady Booch, Ivar Jacobson, and James Rumbaugh at Rational Software in the 1990s. It was adopted by the Object Management Group (OMG) in 1997 and has been managed by this organization ever since. In 2000 the Unified Modeling Language was accepted by the International Organization for Standardization (ISO) as a standard for modeling software-intensive systems. (Rumbaugh, James, 2006)

The Unified Modeling Language (UML) offers a standard way to visualize a system's architectural blueprints, including elements such as:

* Activities
* Actors
* Business processes
* Database schemas
* (logical) components
* Programming language statements
* Reusable software components

## **Object-oriented analysis and design (OOAD)**

Object-oriented analysis and design (OOAD) is a popular technical approach to analyzing, designing an application, system, or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to foster better stakeholder communication and product quality. There are a number of different notations for representing these models, but in our case, the Unified Modeling Language (UML) will be used as mentioned above. (Dennis Alan, 2012)

### **Object-oriented analysis**

Used of modeling to [define](http://www.businessdictionary.com/definition/define.html) and [analyze](http://www.businessdictionary.com/definition/analyze.html) the [requirements](http://www.businessdictionary.com/definition/requirements.html) necessary for the [success](http://www.businessdictionary.com/definition/success.html) of a [system](http://www.businessdictionary.com/definition/system.html). Object-oriented analysis is a [process](http://www.businessdictionary.com/definition/process.html) that [groups](http://www.businessdictionary.com/definition/group.html) [items](http://www.businessdictionary.com/definition/item.html) that interact with one another, typically by [class](http://www.businessdictionary.com/definition/class.html), [data](http://www.businessdictionary.com/definition/data.html) or [behavior](http://www.businessdictionary.com/definition/behavior.html), to [create](http://www.businessdictionary.com/definition/create.html) a [model](http://www.businessdictionary.com/definition/model.html) that accurately [represents](http://www.businessdictionary.com/definition/represent.html) the intended purpose of the system as a whole. Object-oriented analysis does not [factor](http://www.businessdictionary.com/definition/factor.html) implementation [limitations](http://www.businessdictionary.com/definition/limitation.html) into the model. (Wixom, Barbara Haley, 2011).

### **Object-oriented design**

Object-oriented design is the process of planning a system of interacting objects for the purpose of solving a software problem. Start with the candidate objects defined during analysis, but add much more rigor to their definitions. Then you add or change objects as needed to redefine a solution. Object-oriented design (OOD) elaborates on the analysis models to produce implementation specifications. While OOA focuses on what the system does, OOD focuses on how the system does it. (Dennis Alan, 2012)

## **Object-oriented programming (OOP)**

Object-oriented programming (OOP) is a programming paradigm that represents concepts as "objects" that have data fields (attributes that describe the object) and associated procedures known as methods. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs: C#, C++, Objective-C, Smalltalk, Java, Perl, Python, Ruby, and PHP are examples of object-oriented programming languages. Object-oriented programming (OOP) is a programming language model organized around objects rather than "actions" and data rather than “logic”. (Dennis Alan, 2012)

Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data. The Unified Modeling Language is a graphical modeling that provides us with a syntax for describing the major elements of software systems. The UML is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development.

# **UML Concepts**

The description of the object-oriented programming has highlighted the extent of the necessary conceptual work: defining classes, relationships, attributes and methods, interfaces, etc.

The following are UML useful notations:

**Class**: is a description of a collection of objects with common attributes and behaviors. A class is divided into three parts as shown below:

|  |
| --- |
| **Class Name** |
| Attributes |
| Operations() |

* The upper part holds the name of the class.
* The middle part contains the attribute of the class.
* The last part gives the method of operation the class can take or undertake.

An attribute is a named property of a class that describes a range of values that instances of the property may hold. A method is the implementation of a service that can be requested from any object to the class to affect behavior.

* **An attribute** is a named property of a class that describes a range of values that instances of the property may hold.
* **A method** is the implementation of a service that can be requested from any object to the class to affect behavior.

**Relationships:** A relationship is a connection between things.

The three most important relationships are association, generalization, and dependency

A primary purpose of a class diagram is to show the relationships, or associations, that classes have with one another (Stumptner, Springer Berlin Heidelberg, 2007). These are depicted in the diagram by drawing lines between classes.

* **Association**

Use cases are connected to actors through association relationships; these relationships show with which use cases the actors interact. A line drawn from an actor to a use case depicts an association. The association typically represents two-way communication between the use case and the actor. It is represented by the following line.

****

* **Generalization**

Generalization is a relationship between a general kind of thing (called the superclass or parent) and a more specific kind of thing (called the subclass or child). It is denoted by as the following:



* **Aggregation**

It is a plain association between two classes that represent a structural relationship between peers, meaning that both classes are conceptually at the same level, no one more important than the other.



* **Composition**

This is a form of aggregation, with strong ownership and coincident lifetime as part of the whole.



* **Dependency**

Dependence is a weaker form of relationship which indicates that one class depends on another because it uses it at some point in time.



* **Extend**

In another form of interaction, a given use case (the extension) may extend another. The relationship indicates that the behavior of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label «extend». The notes or constraints may be associated with this relationship to illustrate the conditions under which this behavior will be executed. (Dennis Alan, 2012)

Modelers use the «extend» relationship to indicate use cases that are "optional" to the base use case. Depending on the modeler's approach "optional" may mean "potentially not executed with the base use case" or it may mean "not required to achieve the base use Case goal".

«Extends»

**Actor:** An *actor* is not a specific user, but instead is a role that a user can play while interacting with the system. An actor can also represent another system in which the current system interacts. In this case, the actor optionally can be represented by the following stick figure labeled with the actor name. (Rumbaugh, James, 2006)



**Use Case:** is a major process that the system will perform that benefits an actor or actors in some way and it is labeled using a descriptive verb-noun phrase, it also represents a major piece of system functionality. (Wixom, Barbara Haley, 2011)



**System boundary boxes (optional)**

A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of the system. Anything within the box represents functionality that is in scope and anything outside the box is not. (Rumbaugh, James, 2006)



But in this project, we will focus only on Actor, Use Case, and System boundary as UML notations.

## **Software development process**

In software engineering, the Software development process is a division of software development work into distinct phases or stages containing activities with the intent of better planning and management (Kraemer, 2009). It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application. In my concern, I am going to develop an application that will meet the user’s needs.

Object-oriented approaches to developing information systems, technically speaking, can use any of the traditional methodologies (waterfall development, parallel development, phased development, prototyping, and throwaway prototyping). However, the object-oriented approaches are most associated with a phased development RAD methodology. (Dennis Alan, 2012)

The primary difference between a traditional approach like structured design and an object-oriented approach is how a problem is decomposed. In traditional approaches, the problem decomposition process is either process-centric or data-centric. However, processes and data are so closely related that it is difficult to pick one or the other as the primary focus. Based on this lack of congruence with the real world, new object-oriented methodologies have emerged that use the RAD-based sequence of SDLC phases but attempt to balance the emphasis between process and data by focusing the decomposition of problems on objects that contain both data and processes. Both approaches are valid approaches to developing information systems. In this book, we focus only on object-oriented approaches. (Rumbaugh, James, 2006)

## 

## **Unified Process (UP)**

Unified process (UP) is a software development process (SDP) also known as software engineering process (SEP), SDP is the process in which we turn user requirements into software. (Wixom, Barbara Haley, 2011) It tells the workers, activities, and artifacts that are needed to utilize, perform or create in order to model a software system (It defines the Who, What, When, and How of software development).

One of its characteristics is that UP is an iterative and incremental development process. The iterative aspect means that the project has to be broken into small subprojects (called iterations) which are easier to manage and to complete successfully. UP is incremental because each iteration generates a baseline that comprises a partially completed version of the final system and any associated project documentation. Baselines build on each other over successive iterations until the final finished system is achieved. (Rumbaugh, James, 2006)

UP phases (Inception, Elaboration, Construction, and Transition) are divided into a series of time-boxed iterations. Each iteration results in an increment, which is of the system that contains added or improved functionality compared with the previous release. (Dennis Alan, 2012)

# **Analysis of the new system**

The analysis phase answers the question of who will use the system, what the system will do, and where it will be used.

Systems analysis is the dissection of a system into its component pieces to study how those component pieces interact and work. Those pieces are called use cases.

A very powerful UML tool is the Use Case. A Use Case is simply a description of a set of interactions between a user and the system. (Rumbaugh, James, 2006)

## 

## **Design of the new system**

### **Use Case Diagram**

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal. (Eileen M., Trauth, 2006)

**The symbols below are used in the use case diagram:**

|  |  |
| --- | --- |
| **Description** | **Shape** |
| **An actor:** Is a Person/System that derives benefit from and is external to the subject.   * It is depicted as either a stick figure (default) or if a nonhuman Action is involved, as a rectangle with « actor » in it (alternative). * It can be associated with other actors using a specialization/superclass Field, denoted by an Arrow with a hollow arrowhead. |  |
| **A use case:** Represents a major piece of system functionality**.**   * Can extend another use case. * It can include another use case. * Is placed inside the system boundary**.** * Is labeled with a descriptive verb-noun phrase. |  |
| **A Field relationship:**   * Links an actor with the use case(s) with which it interacts. |  |
| **A boundary:** It is a box drawn around the use case to denote the edge or boundary of the system being modeled.   * Includes the name of the subject inside or on top. * Represents the scope of the subject, e.g. a system or an individual business process. |  |

Table 1: Use-case diagram elements

The figure below describes the operations of the new system through the use of a case diagram.

**Use Case Diagram**

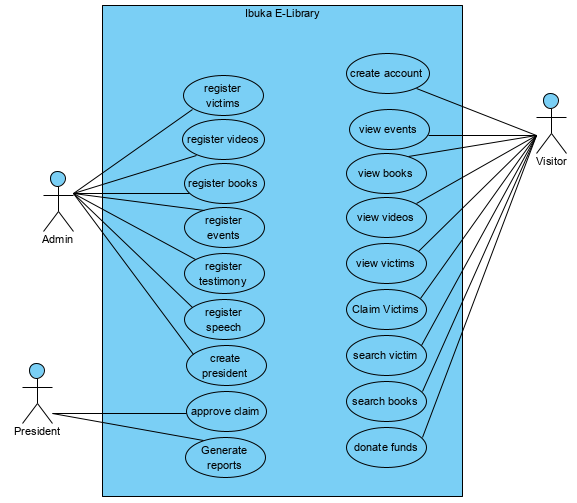


Figure 2: Use case diagram

**Use-case description**

Use Case description details what a use case does, and what it requests in order to be well executed. Each use case looks like this:

* **Name:** A name of a use case.
* **Description:** What a system intends to do.
* **Actor:** The actor involved in the use case.
* **Pre-condition:** The system state before the use case can begin.
* **Post-condition:** The system state when the use case is over.
* **Normal flow:** The actual steps of the use case.
* **Alternative flow:** Steps that may happen in case a normal flow fails.

**Create Account Use Case Description**

|  |  |
| --- | --- |
| **Use Case Number** | **UC-01** |
| **Use Case Name** | **Create Account** |
| **Actor** | **Admin** |
| **Description** | Admin creates an account to access the system. |
| **Pre-Condition** | The Admins must provide valid information. |
| **Post-Condition** | The system should display a message that an account is created. |
| **Normal flow** | 1. Admin run the System. 2. Admin request create account form to the system. 3. System display an account form to fill. 4. Admin fills in the requested data and submit a filled form to the System. 5. System validates data. 6. System save data into the database and display a confirmation message. |
| **Alternative flow** | * 5.a) If there is invalid data filled system shows the error message. * 5.b) Then user fill and submit the information again. * 6.a) If the information are unsaved system display error message”. |

Table 2: Create account use-case description

**Login Use Case Description**

|  |  |
| --- | --- |
| **Use Case Number** | **UC-02** |
| **Use Case Name** | **Login** |
| **Actor** | **Admin** |
| **Description** | Admins login with their credential to access system. |
| **Pre-Condition** | The Admins must use valid username and password to login. |
| **Post-Condition** | The system should display a welcome message to home page. |
| **Normal flow** | 1. An Admin run the system and fill a login form to access information. 2. System validates submitted information of login form. 3. System display a home page(Admin Panel). |
| **Alternative flow** | * 2.If login failed system shows an error and allows to try again. |

Table 3: Login use-case description

**Register Data Use Case Description**

|  |  |
| --- | --- |
| **Use Case Number** | **UC-03** |
| **Use Case Name** | **Register Data** |
| **Actor** | **Admin** |
| **Description** | Admin add informations into database |
| **Pre-Condition** | Admin must login to access the system with a valid credential. |
| **Post-Condition** | The system should display successful message on complete. |
| **Normal flow** | 1. Admin run the System and request data registration form. 2. System display data registration form and Admin fill it. 3. Admin submit a filled form and the system validate it. 4. System save data in database and display confirmation message. |
| **Alternative flow** | * 3.a) If the is invalid data filed system shows an error message. * 3.b) Then admin fill and submit the information again. * 4.a) If data are unsaved system display ”registration failed try again”. |

Table 4: Register Data use-case description

**Search Data Use Case Description**

|  |  |
| --- | --- |
| **Use Case Number** | **UC-04** |
| **Use Case Name** | **Search Victims/ Search Books** |
| **Actor** | **Visitor** |
| **Description** | Visitor can search victims or books and view all details available. |
| **Pre-Condition** | Fill the search form |
| **Post-Condition** | The system should display a searched victims. |
| **Normal flow** | 1. Visitor must run the System. 2. System display a List of victims with a search box on top. 3. Visitor type a name of a victim to search. 4. System validates the searched information. 5. System display the Victims such as: name, gender, materials based on what has been searched. |
| **Alternative flow** | * 5.a) If the information search are not valid the system shows nothing. * 5.b) Then a Visitor search again. |

Table 5: Search Data use-case description

**Donate Use Case Description**

|  |  |
| --- | --- |
| **Use Case Number** | **UC-05** |
| **Use Case Name** | **Donate funds** |
| **Actor** | **Visitor** |
| **Description** | Visitors can donate funds they want. |
| **Pre-Condition** | Fill the donation form |
| **Post-Condition** | The system should display a successful message. |
| **Normal flow** | 1. Visitor must run the System. 2. System display a donate button. 3. Visitors enters amount they want to donate. 4. System validates the entered amount. |
| **Alternative flow** | * 5.a) If the information entered are not valid the system shows error message. * 5.b) Then a Visitor can try again to donate. |

Table 6: Donate funds use-case description

**Claim Victims Use Case Description**

|  |  |
| --- | --- |
| **Use Case Number** | **UC-06** |
| **Use Case Name** | **Claim Victims** |
| **Actor** | **Visitor** |
| **Description** | Visitors can claim a body if not found on our victim’s list. |
| **Pre-Condition** | Fill the claim form |
| **Post-Condition** | The system should display a successful message. |
| **Normal flow** | 1. Visitor must run the System. 2. The visitor login in his/her account. 3. The system show a list of victims with a button to claim. . 4. Visitors enters the details of the victims 5. System validates the entered information. |
| **Alternative flow** | * 5.a) If the information entered are not valid the system shows error message. * 5.b) Then a Visitor can try again to claim. |

Table 7: Claim Victims use-case description

**Approve claim Use Case Description**

|  |  |
| --- | --- |
| **Use Case Number** | **UC-07** |
| **Use Case Name** | **Approve claim** |
| **Actor** | **President** |
| **Description** | Ibuka President can approve claims sent by Visitors about their lost victims. |
| **Pre-Condition** | View list of claims. |
| **Post-Condition** | The system should display a successful message. |
| **Normal flow** | 1. The president must run the System. 2. The president login in the system 3. System display a List of claims. 4. The president examines claims. 5. The president approve or deny the claim. |
| **Alternative flow** | * If the claim is a dead end then the visitor receive a decline message. |

Table 8: Approve Claim use-case description

### **Class Diagram**

A class diagram is the structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

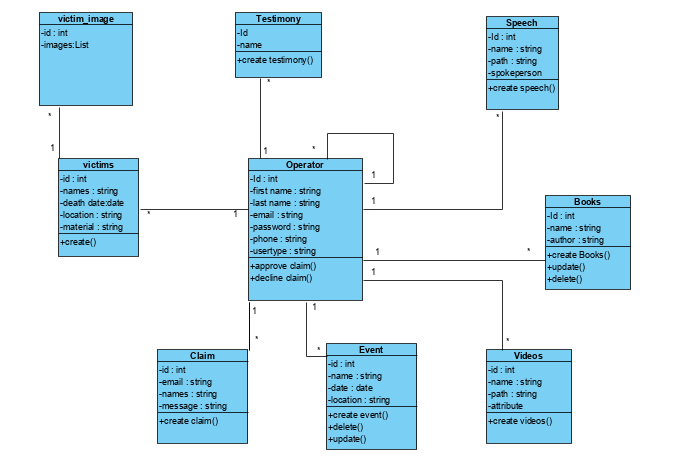
A class diagram is an illustration of the relationships and source code dependencies among classes in Unified Modeling Language (UML), in this context; a class defines the methods and variables in an object, which is a specific entity in a program or the unit of representing that the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP).in a class diagram; the classes are arranged in groups that share common characteristics.

#### **Benefits of class diagrams**

Class diagrams offer a number of benefits for any organization. Use UML class diagrams to:

* Illustrate data models for information systems, no matter how simple or complex.
* Better understand the general overview of the schematics of an application.
* Visually express any specific needs of a system and disseminate that information throughout the business.
* Create detailed charts that highlight any specific code needed to be programmed and implemented to the described structure.
* Provide an implementation-independent description of types used in a system that is later passed between its components.

# **Class Diagram**



**Figure 3: Class diagram**

# **Sequence diagram**

A sequence diagram is a form of interaction diagram which shows objects as lifelines running down the page, with their interactions over time represented as messages drawn as arrows from the source lifeline to the target lifeline. A sequence diagram shows object interactions arranged in time sequence.

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

#### **Purpose of Sequence Diagram**

* Model high-level interaction between active objects in a system.
* Model the interaction between object instances within a collaboration that realizes a use case.
* Model the interaction between objects within a collaboration that realizes an operation.
* Either model generic interactions (showing all possible paths through the interaction) or specific instances of an interaction (showing just one path through the interaction).

**The notations and their definitions that are used in a sequence diagram:**

|  |  |
| --- | --- |
| **Term and definition** | **Symbol** |
| **An actor:**   * It can be a person or system that derives benefit from and is external to the system. * It participates in a sequence by sending/receiving messages. * It is placed across the top of the diagram. |  |
| **An object:**   * It participates in a sequence by sending/receiving messages. * It is placed across the top of the diagram. |  |
| **A lifeline:**   * Denotes the life of an object during a sequence. |  |
| **An activation:**   * It is a long narrow rectangle placed on top of a lifeline. * It denotes when an object is sending or receiving messages. |  |
| **A message:**   * It conveys information from one object to another one. * An operation call is labeled with the message being sent and a solid arrow, whereas a return is labeled with the value being returned and shown as a dashed arrow. | **Message send**    **Return Message**    **Self-Message**    **Asynchronous** |

Table 9: Sequence diagram elements

**Create Account Sequence Diagram**

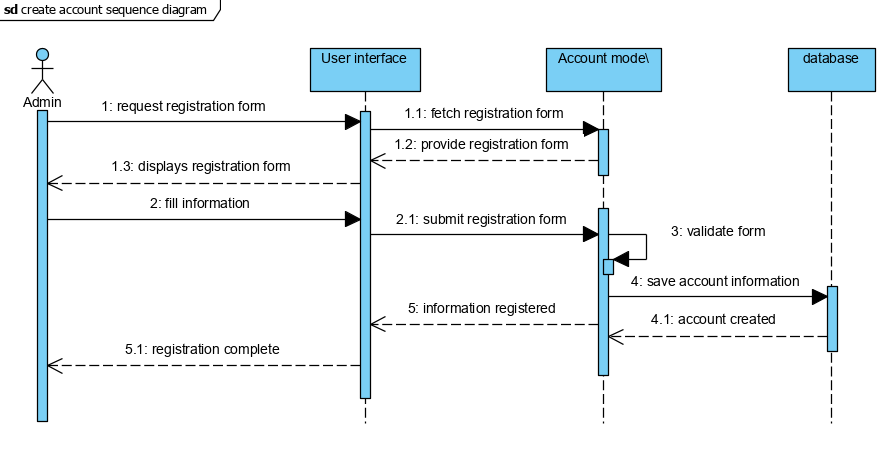


Figure 4 : Create account sequence diagram

**Login Sequence Diagram**

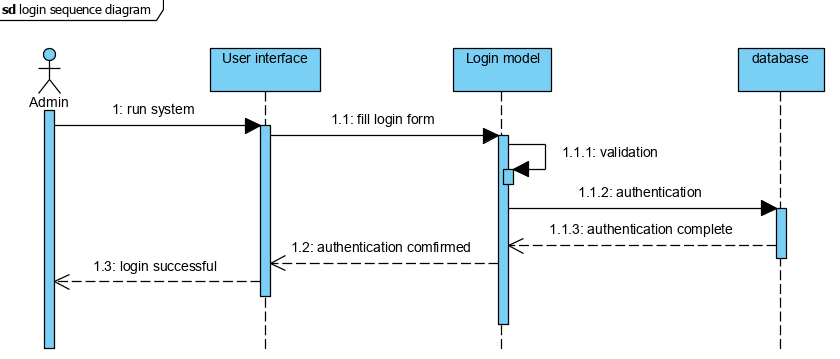
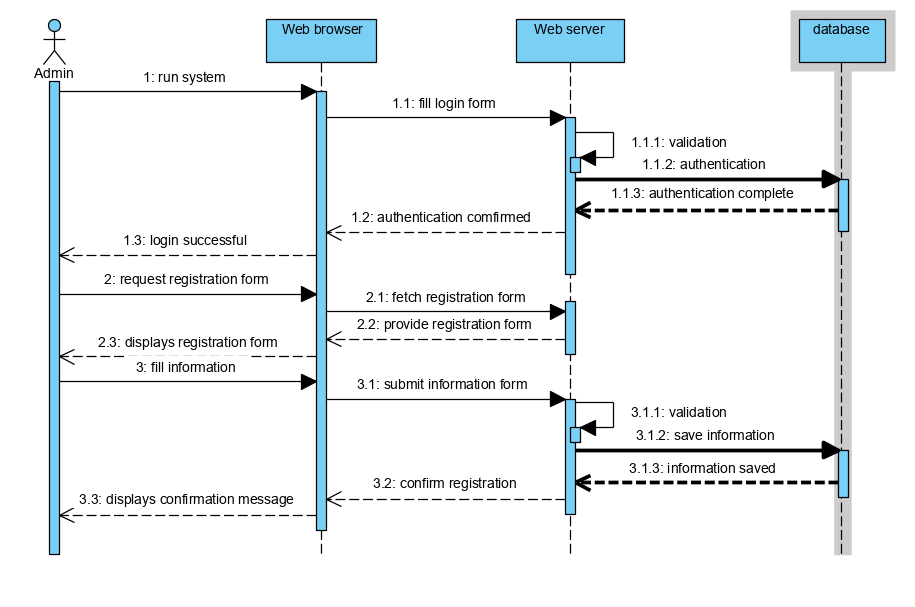


Figure 5: Login sequence diagram

**Registering a victim, Event, Testimony, speech, videos**

Figure 6:Registering a victim

**Claiming sequence diagram**

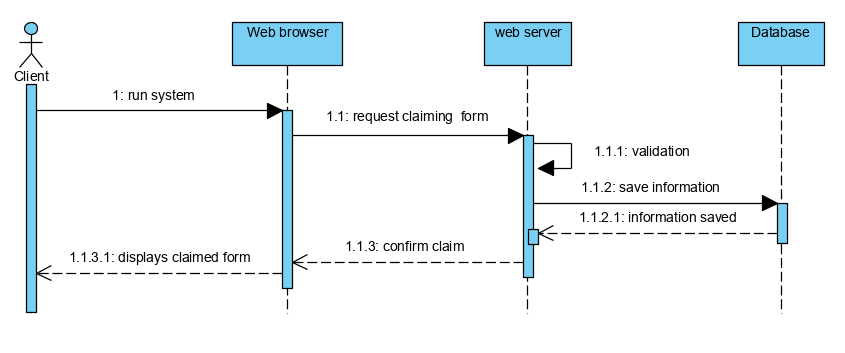


Figure 7: Claiming sequence diagram

**Approving claim sequence diagram**

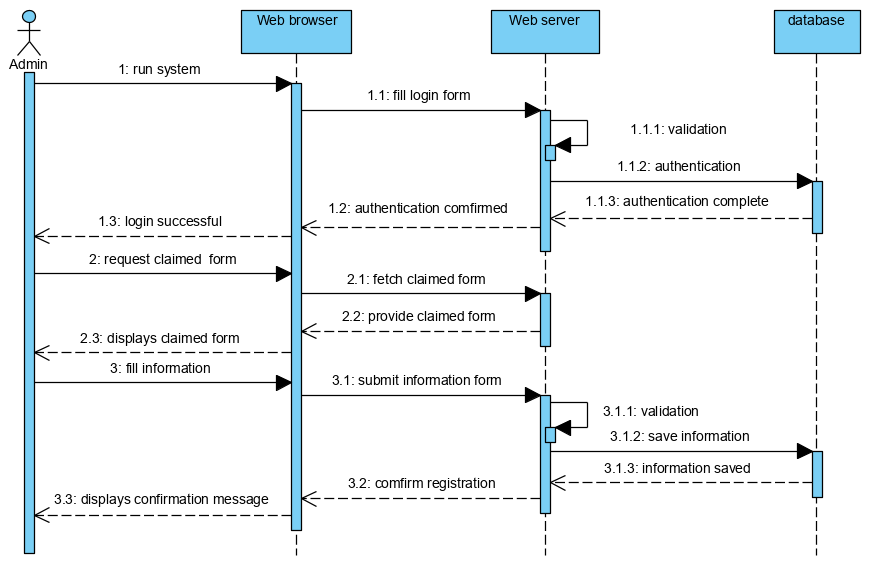


Figure 8: Approving claim sequence diagram

**Viewing event ,videos , testimonies sequence diagram**

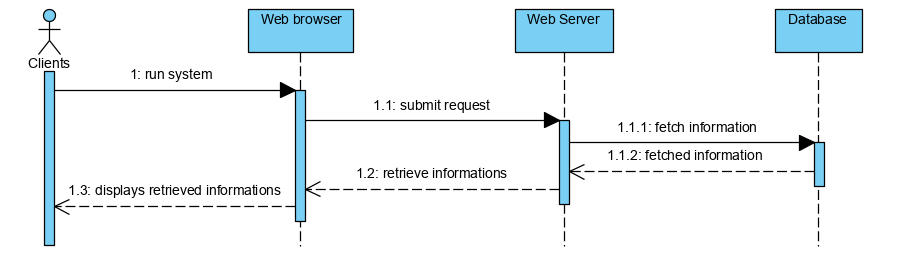


Figure 9: Viewing sequence diagram

**Search information sequence diagram**

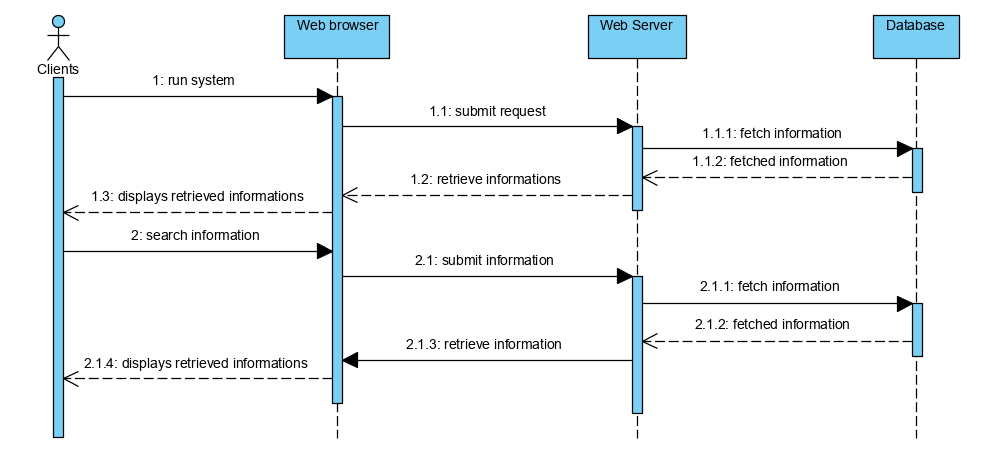


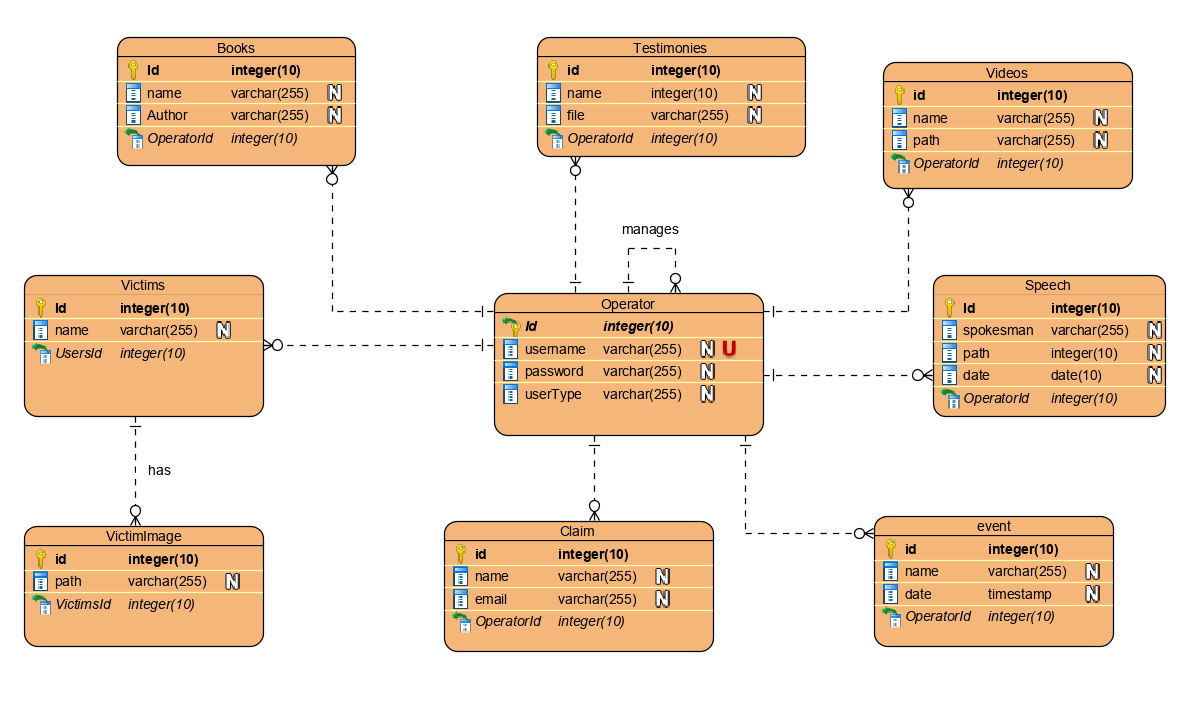
Figure 10: Search information sequence diagram

### 

# 

# **Database Schema Diagram**

The database schema diagram is a set of collection of information that is organized so that it can easily be accessed, managed, and updated. Database Management System (DBMS) is referred to as database software tools that are primarily used for storing, modifying, extracting, and searching for information within a database.



# **Architectural Design**

System architecture design is a diagram of a system, in which the principal parts or functions are represented by blocks connected by lines that show the protocols between the blocks. System Architecture translates the logical design of an information system into a physical structure that includes hardware, software, network support, and processing methods. The following figure displays the sample system architecture of the new application.

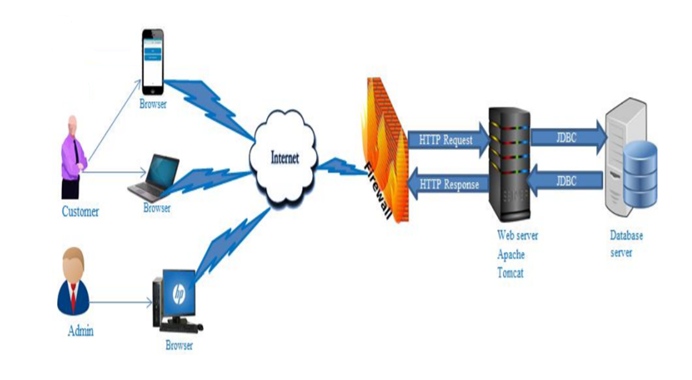


Figure 11: Architectural Design

# 

# **CHAPTER 4**

# **IMPLEMENTATION AND TESTING OF THE NEW SYSTEM**

## **Introduction**

This chapter describes the development of “**IBUKA E-LIBRARY SYSTEM**”. It encloses a brief overview of the technologies used to make this system, operation, tests that have been applied. And also specify software and hardware compatibility requirements.

## **Technologies used**

To develop this System, I used the following different technologies:

| Technology Stack Software | Product |
| --- | --- |
| Integrated development environment | NetBeans IDE |
| Database design | Visual Paradigm |
| Programming languages | JSF(java), CSS, JavaScript |
| Database | MySQL |

Table 10: Technologies used

### **NetBeans IDE**

NetBeans is an integrated development environment (IDE) for java. NetBeans allows applications to be developed from a set of modular software components called modules. NetBeans runs on Windows, mac OS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

### **Visual Paradigm**

Visual Paradigm is a software tool designed for software development teams to model business information system and manage development processes. Visual Paradigm supports key industry modeling languages and standards such as Unified Modeling Language (UML), SysML, SoaML, BPMN, XMI, etc. It offers complete tool-set software companies need for requirements capturing, process analysis, system design, database design, and etc.

**Java Server Faces**

is a framework that allows Web developers to build user interfaces for JavaServer applications. It is supported by Web servers running Java Enterprise Edition (Java EE).

**CSS**

are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

**JavasScript**

is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without communicating with the server.

**MySQL**

MySQL is a software development tool used to develop a database as collection of data. The purpose of a database is to store and retrieve related information.

## 

# **Graphical interfaces of the IBUKA E-LIBRARY**

**Signup Page**

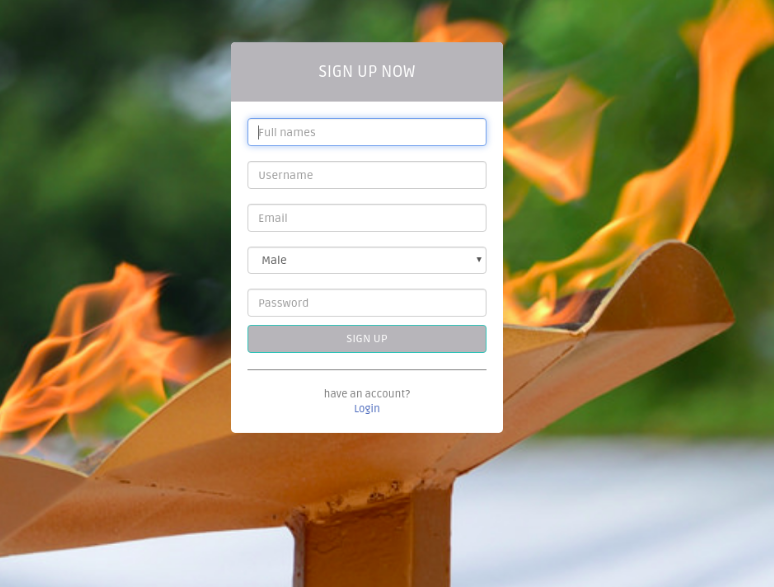


Figure 12: Signup page

**Login Page**

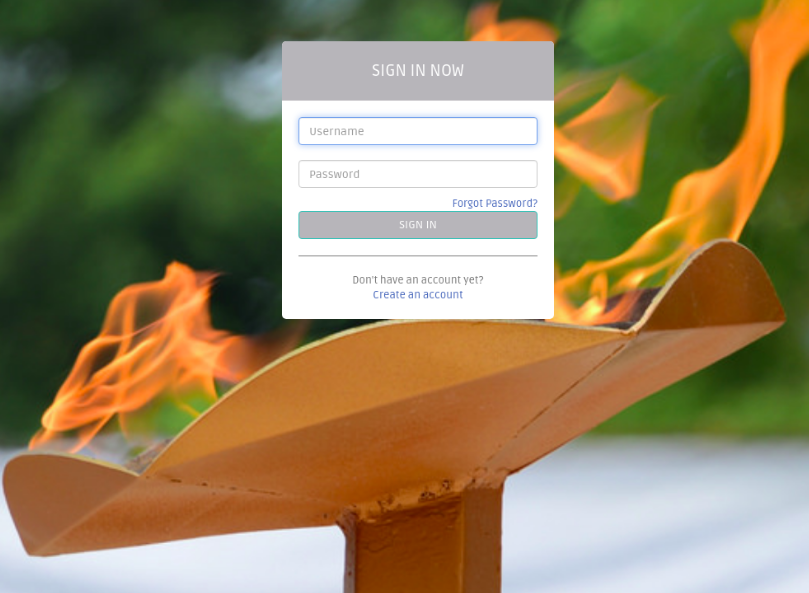


Figure 13: Login Page

**Admin System – Dashboard Page**

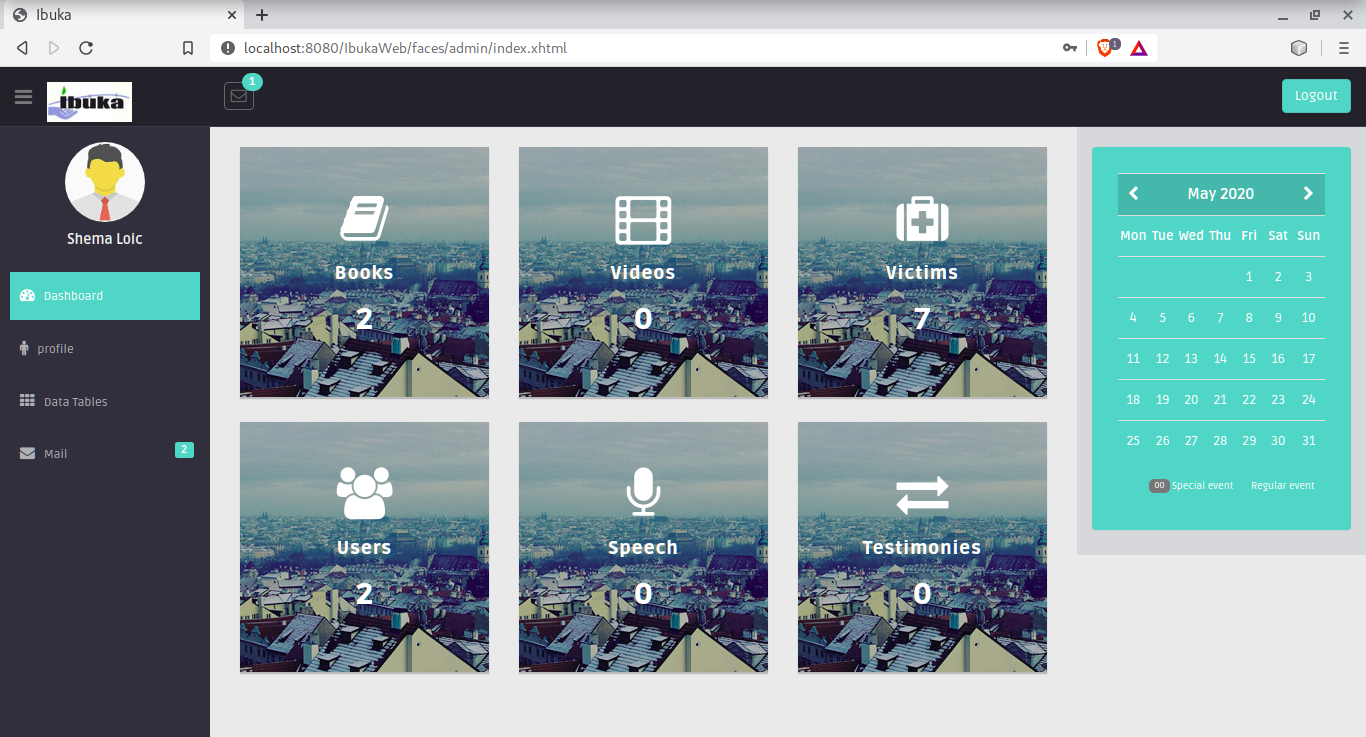


Figure 14**:** Dashboard page

**Admin System – Add Book Page**

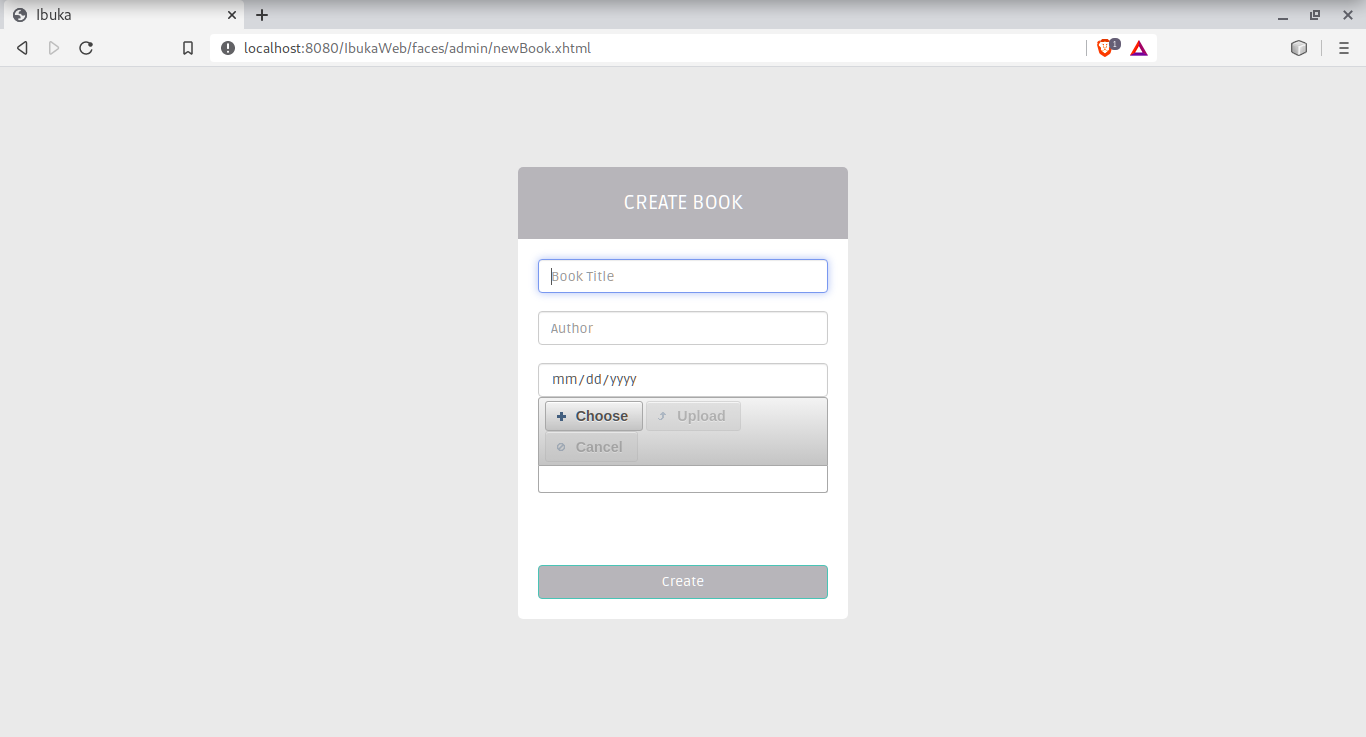
****

Figure 15: Add Book Page

**Admin System – Add VictimPage**

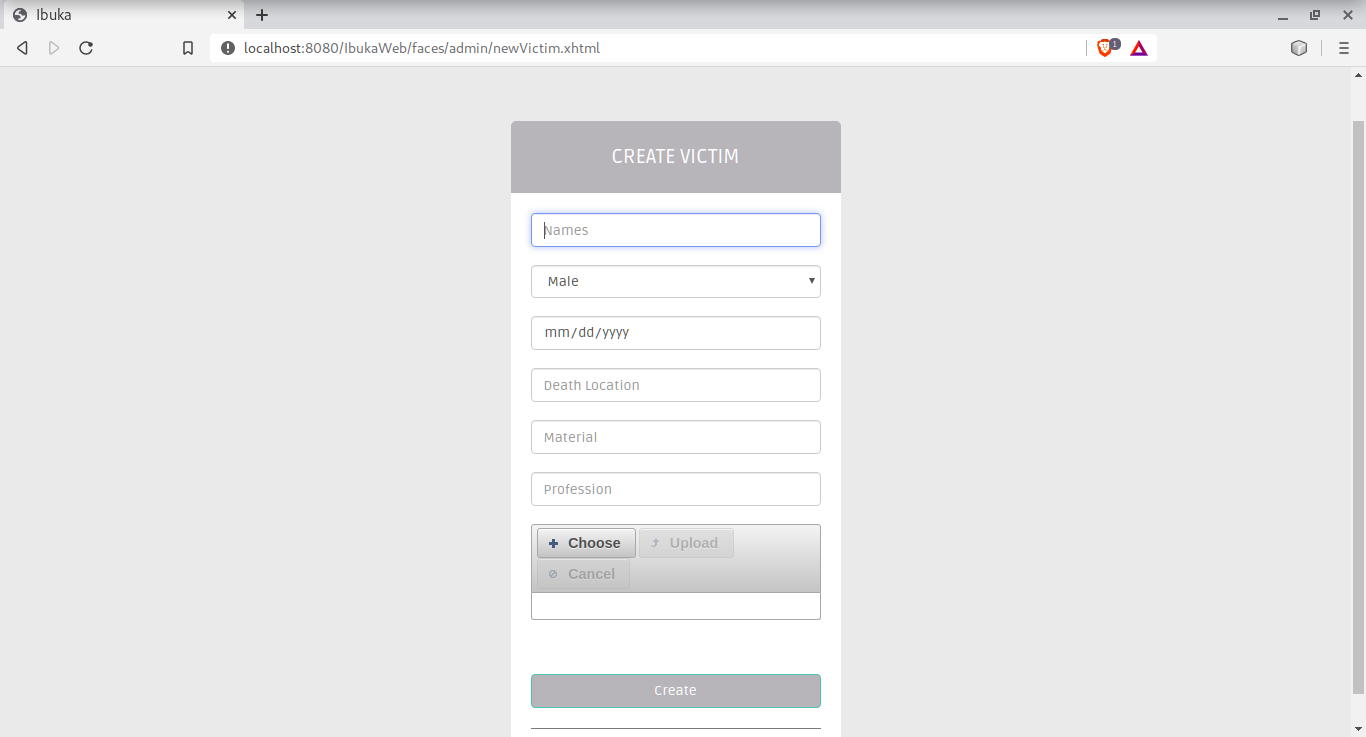
****

Figure 16: Add Victim page

**Admin System – Add Testimony Page**

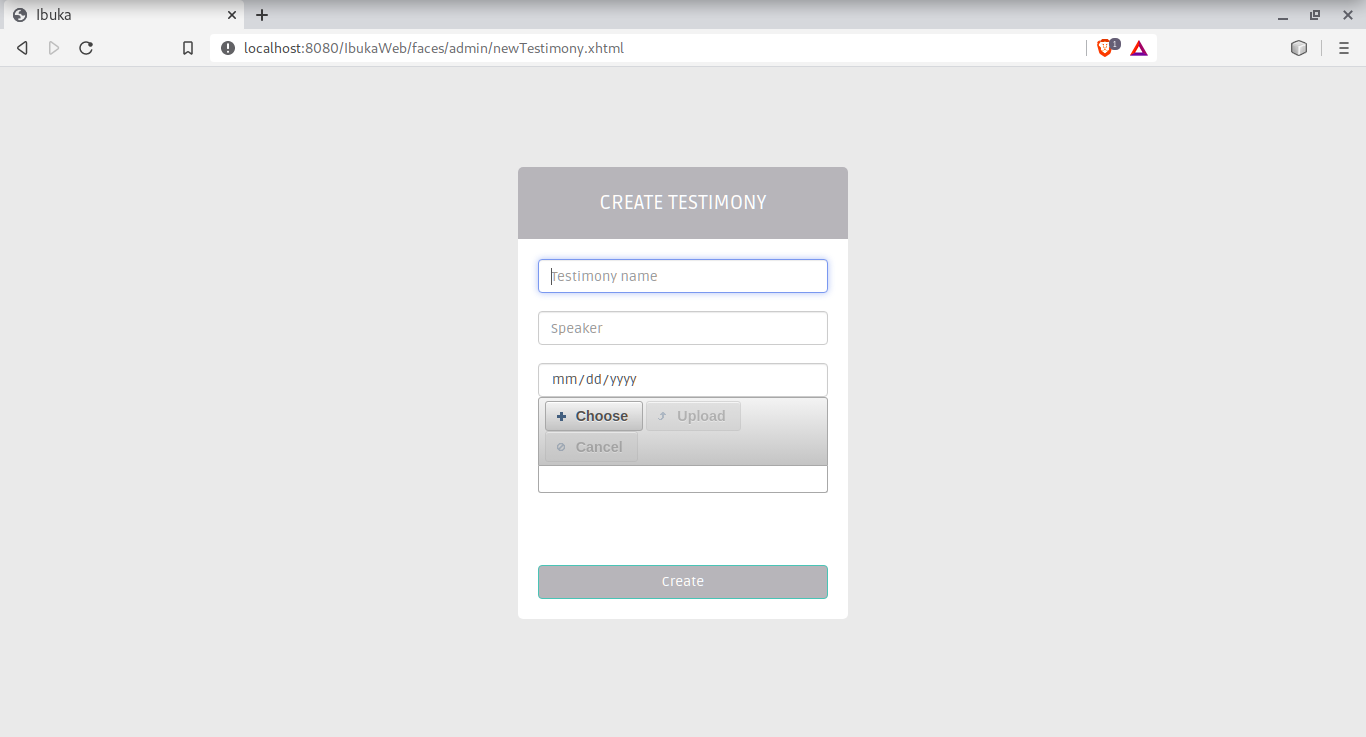
****

Figure 17: Add Testimony page

**\***

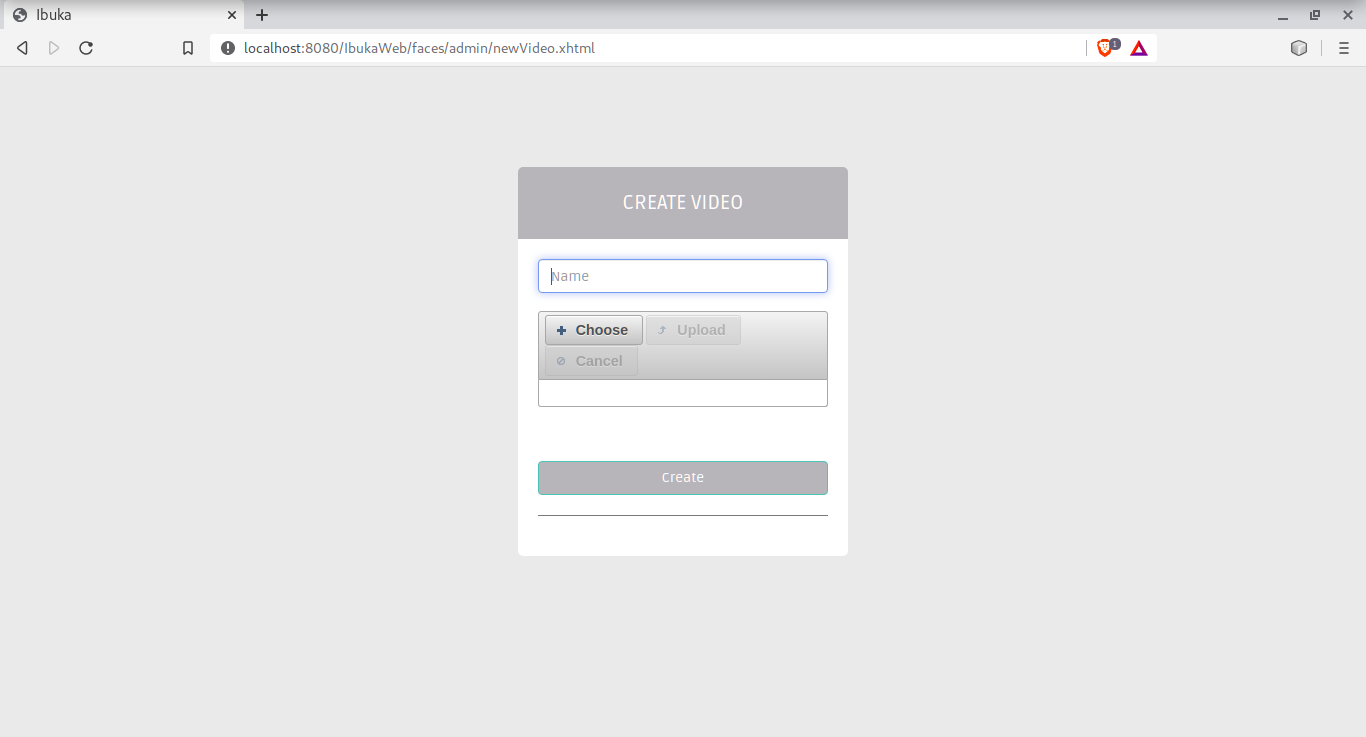
****

Figure 18: Add Video page

**Admin System – Add Speech Page**

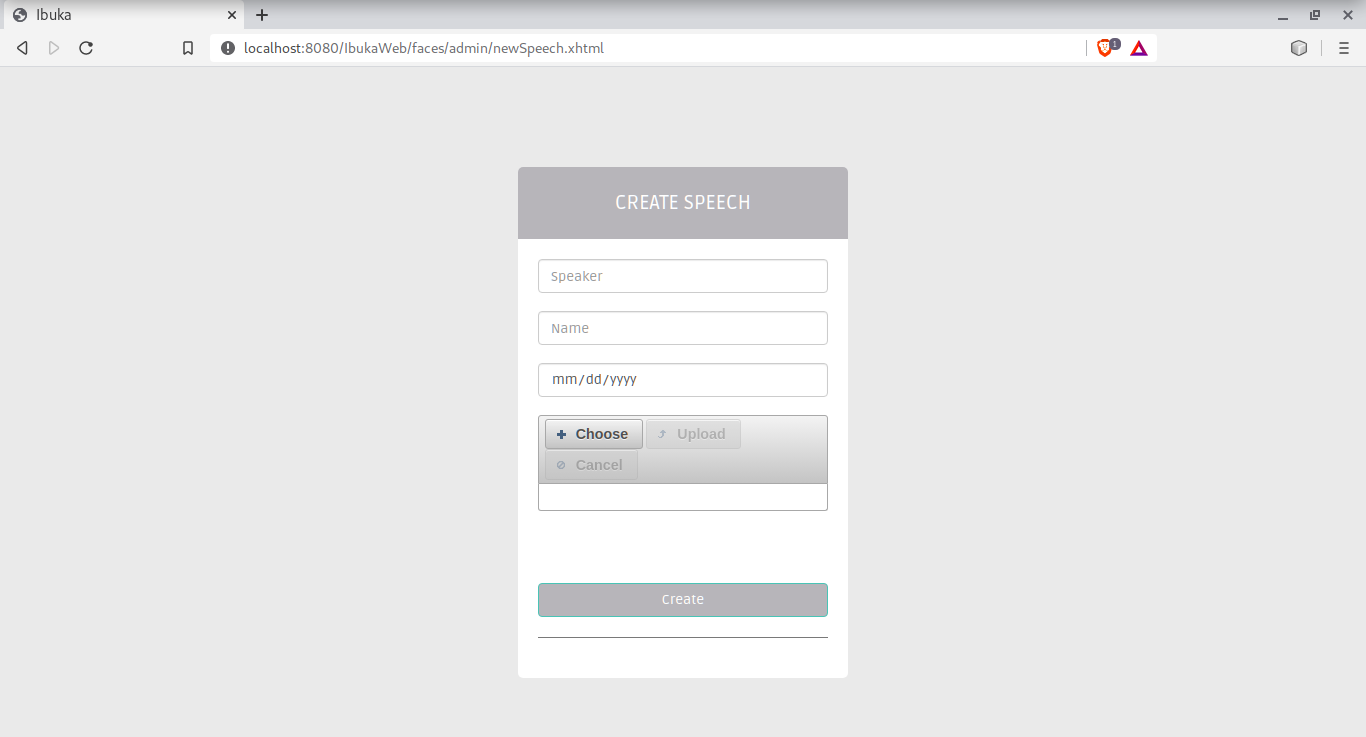
****

Figure 19: Add Speech page

**Admin System – Victim List Page**

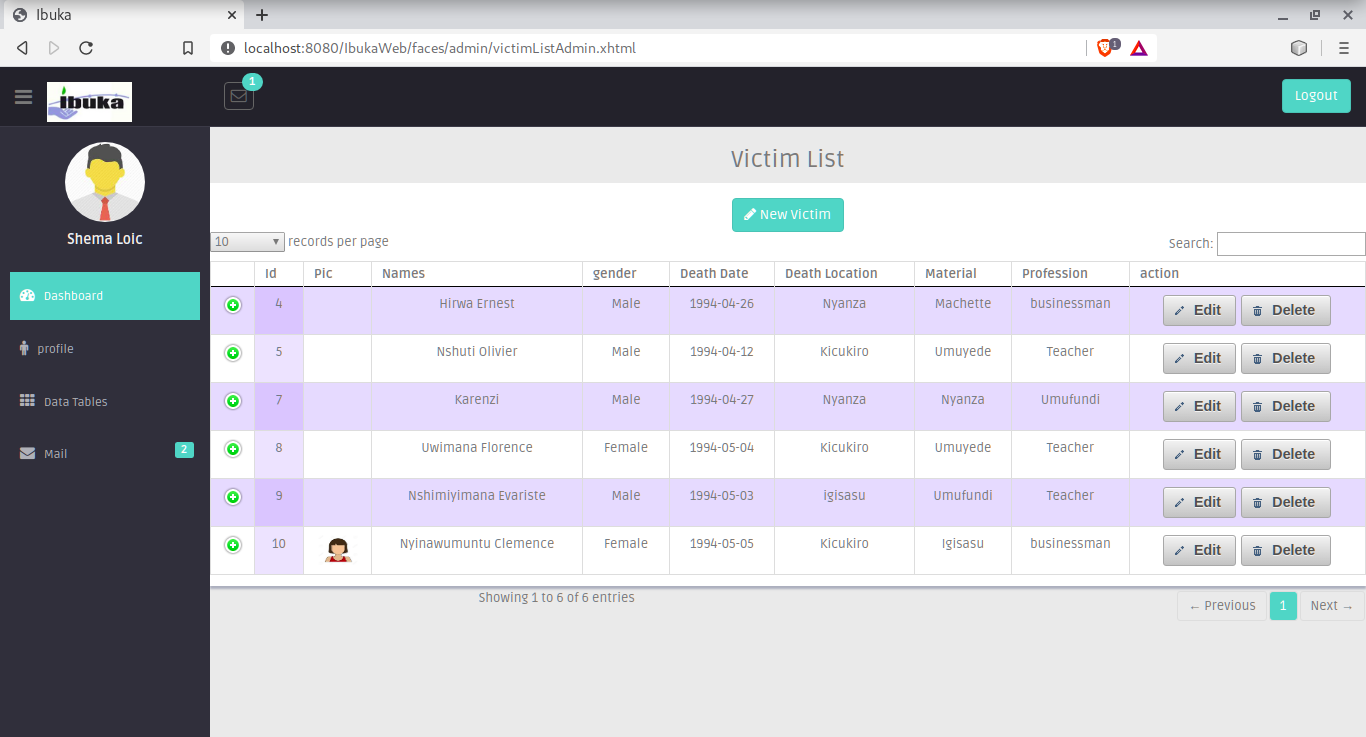
****

Figure 20: Victim List page

**President System – Dashboard Page**

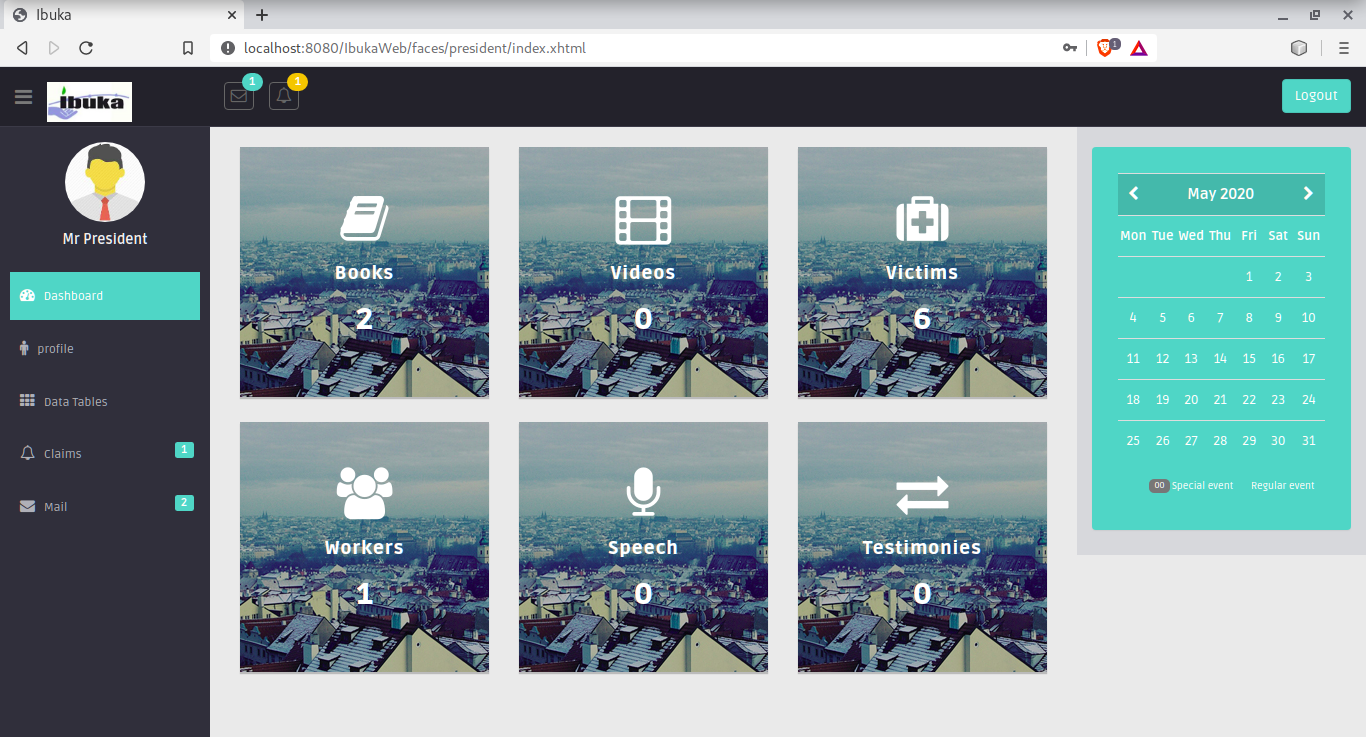
****

Figure 21: President Dashboard page

**President System – Claim List**

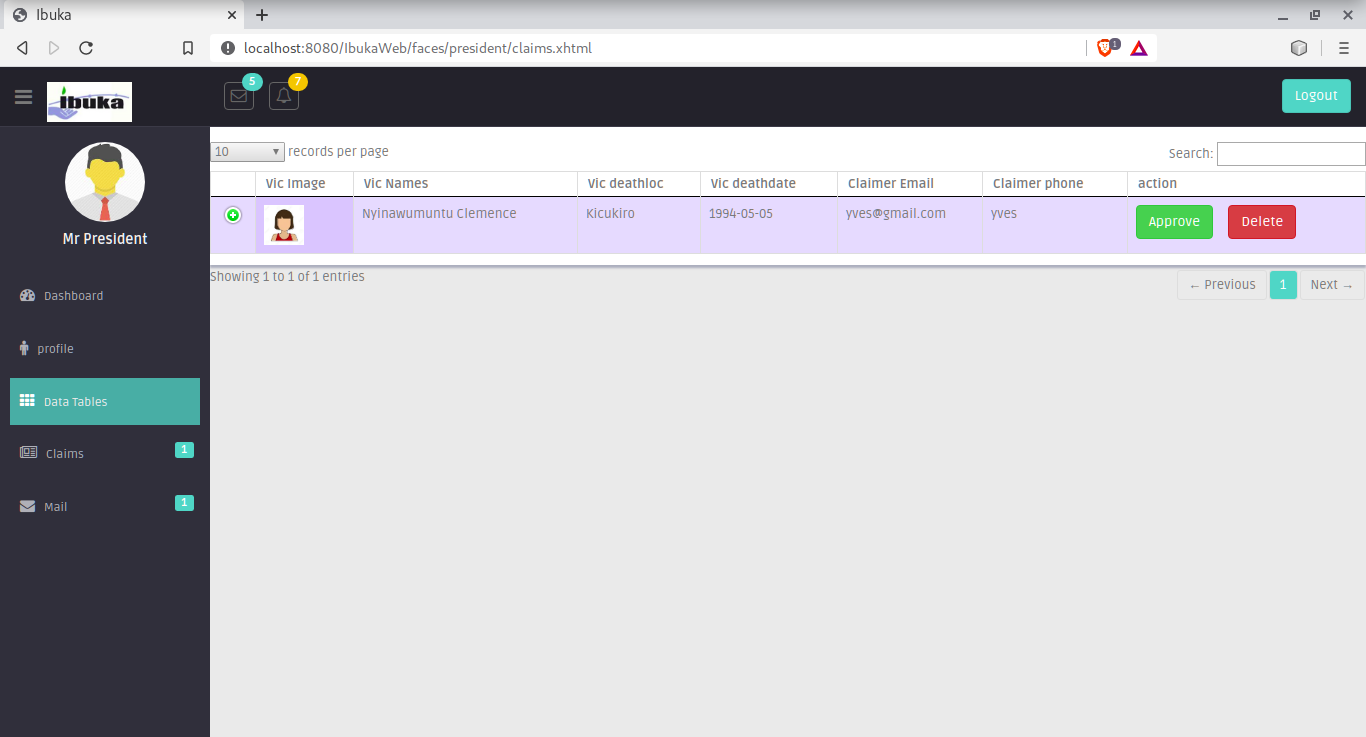
****

Figure 22: President Claim List page

**Client System – Read Book Page**

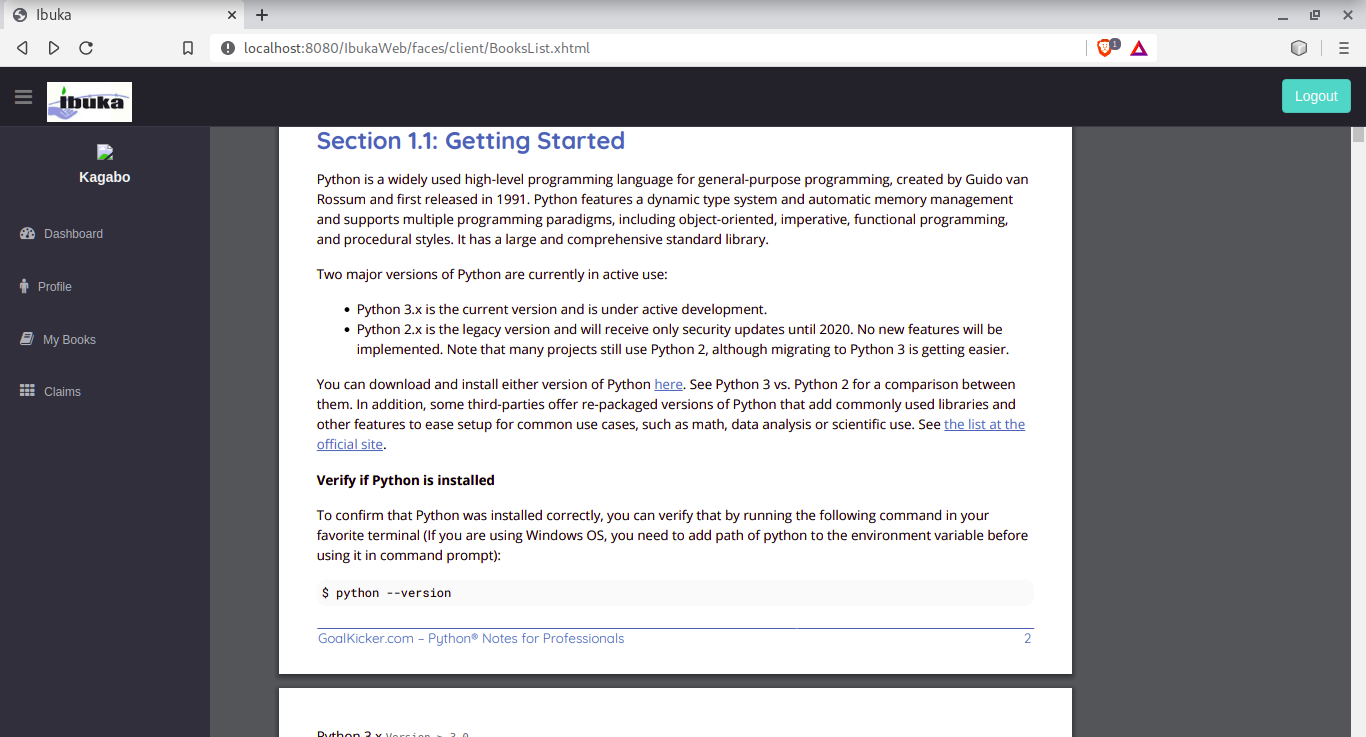
****

Figure 23: Client Read Book page

**Client System – View Victims Page**

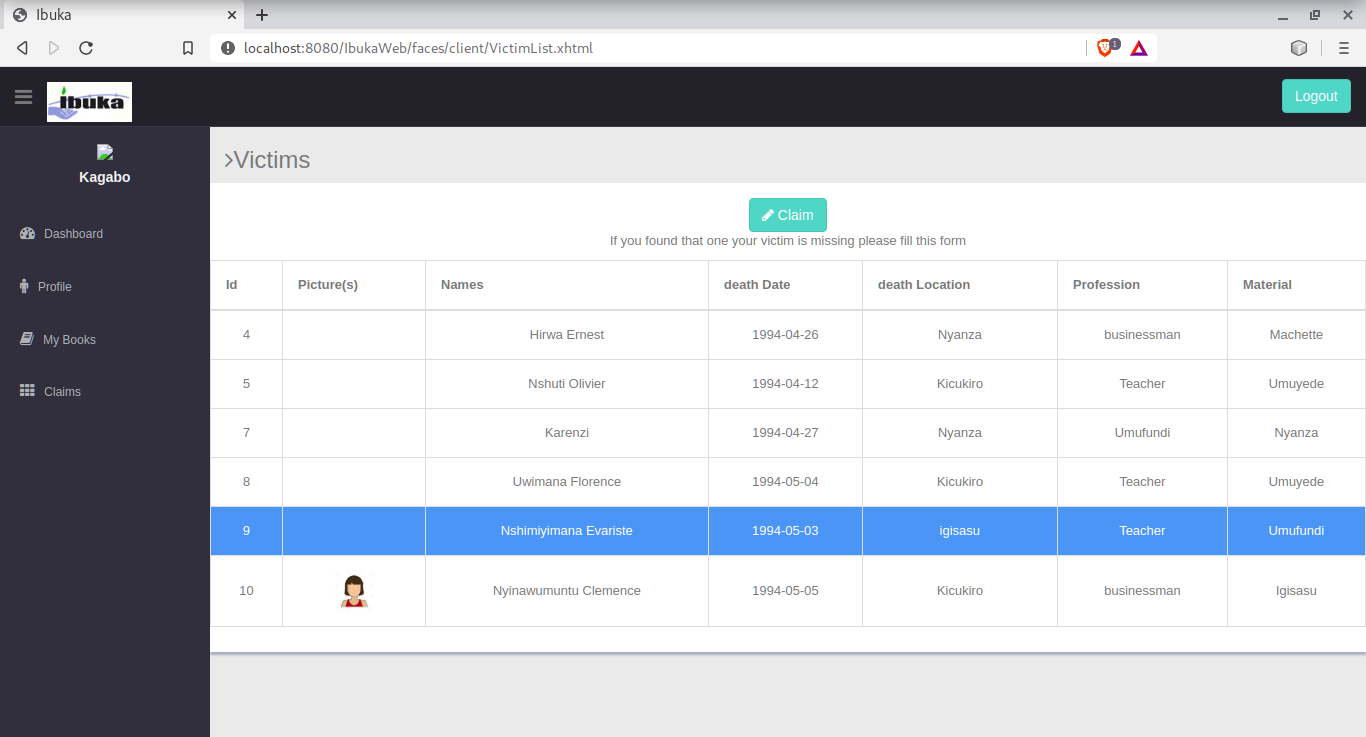
****

Figure 24: Client’s Victim List view page

# **Software Testing**

Software tests play an important role in the software designing. They help to verify the effectiveness of the software to see if it actually does what it was supposed to solve.

Listed are key aspects to take into consideration in software testing:

* Does the application meet the requirements that guided its design and development?
* Does the application works as expected?
* Can the application be implemented with the same characteristics and satisfies the needs of the stakeholders?

The following are some software testing:

**The Unit Test:** Is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use. In other words every small component that can be compiled with the goal to know that every unit matches to its specifications, and to know if there are logical mistakes. Indeed, the unit test is an efficient means that permits to detect the maximum possible mistakes. The application has been checked with the unit test at each piece of the code written.

**The Integration test**: is a level of software testing where individual units are combined and tested as group. the purpose of this level of testing is to expose faults in the interaction between integrated units.

**The Validation test:** The last test phase has the role of validating the software in its external environment. The product has been put in final situation in order to verify if it perfectly answers to the needs expressed in the first phase. The validation test is important, since it is necessary to verify if the setting up of the application corresponds to the expressed needs. The application has been tested in its entirety, and it is in this way that we noticed that the progress of operations done corresponds to the functional specifications.

# **CHAPTER 5**

# **CONCLUSION AND RECOMMENDATION**

## **Conclusion**

The main objective of this project was to design and implement The **IBUKA E-LIBRARY** system that would help all Rwandans and foreigners to understand well our history and decrease the number of all Genocide denials.

During this research project different methods and tools have been used. The observation and documentation have been used as tools for a good understanding of the existing system. After getting the problems in our scope, analysis was done, using UML in order to find the adequate solution by developing the new system using Java programming language in Java Server Faces(JSF).

The expected results realized are the following: Within the new system the citizen can access service of reading a book, viewing a video, listening to speeches, reading given testimonies, viewing victims, and claiming if a user didn’t find his/her relative body using a web application.

The visitor can search a Book, Video, Speech, Victim, Testimony based on their needs, search box of the system allows them to search a Book by using the book name or Author, searching a video using the video name, searching a speech using speech name or spokesman, searching a testimony using the testimony name or spokesman.

As conclusion, the Ibuka E-Library website was successfully implemented as described in previous chapters and is running as expected.

# 

# **Recommendations**

I would like to recommend all Rwandans to use this system to well understand our country’s history and to encourage foreigners to visit as well and stop declining that the Genocide happened, as our President H.E Paul Kagame always says we must learn from our past to build our future.

Finally, I finish my work by welcome and encourage whoever wants to contribute to the improvement of this work.

# 

# **REFERENCES**

Books

- GEARY, D & HORSTMANN, C. (April 2007). Core java server faces, Second Edition, Indiana-USA.

- Bell, Douglas. (2005). *Software Engineering for Students,a Progrmming Approach,Fourth Edition.* USA.

-Dennis Alan. (2012). *System Analysis and Design.* India: John Wiley & Sons, Inc.

-Rumbaugh, James. (2006). *Unified Modeling Language.* USA: Rational software Library.

- Stumptner, Springer Berlin Heidelberg. (2007). *Behavior consistent inheritance in UML.* Mcgrill inc.

-Wixom, Barbara Haley, Roberta M Roth. (2011). *System Analysis and Design Fifth Edition.* USA: Don Fowley.

- Alan Dennis, Barbara Halley Wixom, David Tegardern (2015) System Analysis and Design. An object oriented approach with UML Wiley

- Kraemer. (2009). *Automated encapsulation of UML activities for incremental development and verification.* USA.

Websites

-Kigali Genocide Memorial

-cnlg.gov.rw

- survivors-fund.org.ukz