**DESIGN AND IMPLEMENTATION OF A VISITOR MANAGEMENT SYSTEM**

By

**Ozoilo Johnbosco Chibunna**

Registration no:

**U14/NAS/CSC/079**

Being a BSc report submitted in partial fulfillment of the requirements for the award of a Bachelor’s degree in Computer Science of the Godfrey Okoye University.

**Supervisor: Mrs. M.0 Ezugorie**

**Department of Mathematics/Computer Science Godfrey Okoye University, Enugu**

**………, 2018**

**CERTIFICATION**

This is to certify that this research titled VISITOR MANAGEMENT SYSTEM is an original work of OZOILO JOHNBOSCO CHIBUNNA under the supervision of Mrs. Ezegorie.

## DEDICATION

I dedicate this project to God Almighty my creator, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. I also dedicate this work to my family and friends, for the continuous love they showed me during the time of this write up and also my parents Mr. & Mrs. Ozoilo together with my lovely aunty Mrs. Njideka Onos.

## APPROVAL PAGE

This is to certify that this research titled **VISITOR MANAGEMENT SYSTEM** was carried out by **OZOILO JOHNBOSCO CHIBUNNA,** Registration No: **U14/NAS/CSC/079** of the department of Computer Science in partial fulfillment of the requirements for the award of Bachelor of Science in Computer Science.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 **Mrs. Ezugorie**  Date/sign

 Project Supervisor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 **Mrs. Monica Agu**  Date/sign

 Head of Department

## ACKNOWLEDGMENT

My gratitude goes to God Almighty who made it possible for me to come this far in my Academic pursuit. Without the guidance of my ever listening supervisor,Mrs EZEUGORIE this work would not have been this successful. I appreciate your support, encouragement towards me. Also wish to appreciate my Head of Department for the help rendered and all her advises to the students

My regard also goes to MRS. NJIDEKA ONOS, MR.BENSON, and Dr OZORFOR for their support and advises. I wish to thank my parents Mr & Mrs OZOILO for their immense support. To all my relatives and siblings.

My gratitude also goes to all my friends David, Sampson, Augustine, Ifeanyi, Hector, Peter, Maxwell, goodness, Blessing and all others who assisted me in one way or another to make my project successful and also made my stay in Godfrey Okoye University peaceful.

## TABLE OF CONTENTS

Title page

Certification

Dedication

Approval page

Acknowledgement

Table of Contents

Abstract

**CHAPTER ONE:**

1.0 Background of study

1.1 Statement of the problem

1.2 Objective of study

1.3 Significance of study

**CHAPTER TWO:**

1. Introduction

2.1 Background of study

2.2 Literature Review

2.2.1 Information and Reviews

2.2.2 Product Perspective

2.2.3 Product Function

# 2.2.4 Product Selection/Payment

2.2.5 User charactersitics

2.2.6 General Constraints

2.2.7 Assumptions and Dependencies

2.3 Summary

**CHAPTER THREE:**

3.0 Introduction

3.1 Description of the Existing System

3.1.1 Problems and Weakness of the Existing System

3.2 Analysis of the Proposed System

3.3 Design of the Proposed System

* + 1. Use case Diagram

3.3.2 Database Design

3.4 System Architecture

**CHAPTER FOUR:**

4.0 Introduction

4.1 Choice of Development Environment

4.2 Implementation Architecture

4.3 Software Testing

4.4 Documentation

**CHAPTER FIVE:**

5.0 Introduction

5.1 Summary

5.2 Conclusion

5.3 Recommendation

Reference

Appendix

## ABSTRACT

In this computer technology world the impact of IT contributes major roles in all real time system. Various management systems implemented for achieving the business organization towards profit, standards, and further business enhancement. The main aim of this research is to implement a web based system that can secure the information’s of visitors to a company. This system is a well-structured system and easy to use. The languages used to develop this system are PHP (Personal Home Page), HTML (Hyper Text Markup Language), CSS (Cascading Style Sheet), and Database is MySQL. The website has a large memory of storing all the information of the visitors to a company and also keeping record, it is highly effective and accurate. I will recommend that if there is going to be any modification the new writer should endeavor to improve on the limitations, such as making it possible for visitors to book appointment online and fill the necessary information before coming down for visit or an appointment.

**CHAPTER ONE**

**INTRODUCTION**

**1.0 BACKGROUND OF STUDY**

A Visitor Management System (VMS) is an evolving concept defined as a systematic collection of visitor’s information. It is a record in digital format that is theoretically capable of keeping track of list of visitors and their personal details for effective service from both clients and the company.

This system tracks the usage of a public office by gathering increasing amounts of information. A visitor management system can record the usage of the facilities by specific visitors and provide documentation of visitor’s whereabouts. Since a visitor management system provides a record of office used and the person visited, these systems are frequently used to complement office security systems and access control systems.

As electronic visitor management systems become more common and more powerful, these systems are taking over many of the functions of office security and access control like gathering information manually.

Managing visitors is an essential access control activity traditionally handled through paper logs or guest books, where visitors sign in at the front desk. Synergis’ Visitor Management module allows you to modernize your approach, go paperless, and become more efficient. Enroll visitors, assign physical access rights, and track their movements with ease. A digital visitor management system does the core job of checking in visitors. But it also does many things pen and paper cants which include:

* Maintain a cloud-based visitor log.
* Take photos of your visitors to help you identify them.

And so on.

* 1. **STATEMENT OF PROBLEM**

After carefully analysis of the project the researcher observed that the following problems are mainly encountered by different organization

1. The present method of visitors management system is mainly based on manual recording,
2. The present system makes use of small or no database for recording of visitor’s information.
3. Due to lack of comprehensive database, accountability is usually very difficult,
4. The present system is time consuming,
5. Owing to the paper and pen method of recording information visitors may not give out its full information needed at a particular time,
6. Queuing may occur at the cause of entering information needed.

**1.2 OBJECTIVE OF STUDY**

The aim of this project is to design and implement an online visitors management system.

The objectives include:

1. **To build a web database for storing visitor’s information:** to setup a strong relational database for storing visitor’s information for easy retrieval and annual report analysis.
2. **To develop a web base which helps to insert and retrieve visitors information:** retrieval of information stored in a database is vital and must be ready to get back from the database when needed so that the purpose of storing the data must be accomplished.
3. **To give account of visitors that visited the organization:** the system that is setup must be ready to give total number of visitors that visited the organization per day, week, yearly etc.
4. **To ascertain the reasons why the visitor visited:** the system will also track the reasons why the visitors visited, whether the message is relevant or not.
5. **To provide easy access of searching and navigation for time saving:** the system have a special way of retrieving data from the database like *search* the visitor using its identification number.
6. To provide adequate assessment of staff base on the amount of visitor (Client) and their reason.
	1. **SIGNIFICANCE OF STUDY**

With growth of information technology, the study offers numerous values in business premises and revenue payment:

1. **Visitors get high quality service:** since the recording requires human information, with use of computer more concrete information will be provided.
2. **It will be easier to generate overall report:** comprehensive and relational database will make the report generation easy and faster.
3. **High Security Environments:** Assign and print credentials for your visitors like any other employee, define detailed access rights, specify visitor escorts, and give your staff full control over access to your facilities.
4. It will provide efficient, effective and timely services,
5. It will reduce corruption,
6. It contributes to the entire organization security.

**CHAPTER TWO**

**Literature Review**

**2.0: Introduction**

Organizations need to take care of security measures concerning visitor’s management and tracking. Corporate offices will get too many number of visitors every day for many reasons such as interviews, parcel deliveries. By introducing a visitors management system that allows organizations to track visitors on daily basis by introducing procedures such as ID registration, visitor approvals, process management, pass or badge and record keeping.

The disadvantages of this system, is centered on the area of illiteracy and availability of materials needed for, by the visitors to register with the security. This problem can be encountered mainly in rural areas where internet services are still limited.

**2.1 Theoretical Background**

The project was built with web technologies which are HTML (Hyper Text Markup Language), CSS (Cascading Style Sheet), PHP (Hypertext preprocessor), and wamp server, where relational database was integrated (MySQL and phpMyAdmin), having Apache as the engine controller. And it was made with a user friendly capability, with the help of an application program known as a “web browser”, example Mozilla Firefox, Opera mini, Internet Explorer, Google Chrome, and Maxton Cloud etc.

HTML: This is an acronym for Hyper Text Makeup Language; it is a language for describing the web pages. Html form elements are elements that allow the user to enter information (like text fields, text area fields, drop-down menus etc.) in a form. It also have hyperlinks which is used to connect two or more HTML pages together to make navigations easy from one page to another. HTML uses a tag to define the function of the text written inside it.

Every html tag in the web document has a function it does on the web browser, example if I write

<html>

<head>

<title>Computerized Visitors Management System</title>

</head>

<body bgcolor=”green”>

<h2 align="center" >Computerized Visitors Management System <br /><br /> by <br /><br />Ozoilo Johnbosco Chibunna</h2>

</body>

</html>

The above pseudo-code will display a title Computerized Visitors Management System at the top of the web browser and the main page; Computerized Visitors Management System by Ozoilo Johnbosco Chibunna, all will be displayed at the centered of the web page. A form was designed to collect data from the visitors which involves: surname, first name, middle name, etc which when entered and submitted with the submit button that sends the data to a browser, and into the relational database to hold the data.

HTML is the major tag, which forms the visual website that is used to create forms and specify a location onto which particular portion is mapped inside a web page e.g. <div id=”form”>form location

<form action=”” method=”post”>

surname<input type=”text” name=”surname”/>

first name<input type=”text” name=”first name”/>

Middle name<input type=”text” name=”middle name”/>

<input type=”submit” name=”submit” value=”Submit”/>

</form>

</div>

The HTML tag above when previewed on the browser appears on the screen as follows:

Surname

First name

Middle name

**Submit**

The division (div) positions and held every site layout.

.

MySQL: This is my serial query language; it uses a standard form of well-known SQL data language. Different data types were used in this project. It was used aligns side with PHP to create and submit forms used in the software.

Cascading Style Sheets 3 (CSS3) is a style sheet language used for describing the look and formatting of a document written in a mark-up language these codes are used to style html elements, when html tags are written and viewed on a web browser, it doesn’t really appear in animated format, so we used CSS to style the html tags. It is also used for describing the look and formatting of a document written in a mark-up language.

PHP is a server-side scripting language created in 1995 and designed for web development, but also used as a general-purpose programming language. Php code where used to send the collected data from the user to MySQL relational database. It is a widely used scripting language that was originally designed for web development, to produce dynamic web pages. These codes can be embedded into html tags

**2.2 Review of related Literature**

Charles O’Malley and Eric Turner (2008) The Visitor Registration System is a web-based application used by the Board to register, screen and manage visitors to buildings and other locations owned or leased by the Board.

Rick Hagan (2012) visitor management system refers to the gathering of information from visitors and then tracking their whereabouts within the school. The purpose is to document the information and limit visitor access to only persons who have a need to be there. There are different methods for documenting this information including manual logs and computerized systems. For many schools, this process is manual and consists of a clipboard and hand written visitor stickers. While this is better than nothing, there are technologies today that offer much more in an effort to keep children safe and enhance productivity.

Visitor management is the recording and collection of visitor data, either manually or through automated check in software, for the purpose of knowing who is in the facility, who they are visiting and how long they spent there.

**Anna Spenceley, Jon Kohl, Simon Mcarthur, And Peter MYL** says that, there are many different types of visitors to protected areas. They may be official guests to a national park; researchers working in a strict nature reserve; volunteers assisting with a national park work program; educational groups learning about special natural or cultural heritage; or people who conduct their business within a protected area including contractors and shop owners. Importantly, visitors also include tourists and recreationists. In this chapter, we briefly examine the types of visitors protected area managers may need to deal with and management considerations associated with such visitor use. We, however, provide a focus on tourism and its management in this chapter. Depending on the International Union for Conservation of Nature (IUCN) protected area category, tourism and recreation are common visitor uses of most protected areas and important contributors to local and national economies. As part of managing protected areas for tourists, we describe a management framework for providing a range of recreation opportunities within reserves, the provision of visitor services and facilities and management responses to visitor impacts.

**Andrews International (AI)** (2014) developed Visitor Management System to provide security managers with an efficient tool to reduce delays in logging and badging visitors, distinguish special consideration visitors, and identify individuals “not to be admitted”. The system enables notes and name tag creation – all through one easy-to-use program.

**Joan Engebretson and Whelan** (2009). The new visitor management system also offers a higher level of assurance that visitors will not re-enter the premises at a future date without checking in again, as badges are programmed to automatically expire after a certain number of hours. Other important elements of the new system can be found in the parking garage, which is connected via a Wi-Fi wireless link to the AMAG system. Like the guard at the check-in desk, a security guard in the garage also can scan visitors’ credentials and print out a bar code badge. The guard does this using a wireless handheld device that has a built-in scanner.

As Whelan explains, “It’s set up like a gun.” The guard scans a bar code on

the back of the visitor’s driver’s license by pulling a trigger. Alternatively, the

guard can type in the visitor’s information. A badge is printed out via a wireless link sends the visitor’s data to the AMAG system about 100 to 200 feet away, where the record is added to the visitor database. The most challenging part of the installation was integrating three separate systems based on Internet protocol communications, including the visitor management system and two different access control systems, comments Whelan. “The Easy Lobby system had to populate the access control system once it enters a visitor and we had to manage that data so it’s deleted after x amount of time,” he explains.

**Veristream** (2012) iVisitor by Veristream provides a secure visitor management system for multi-tenant facilities of any size looking for easier visitor control, increased visitor security and simplified visitor badging. No more maintaining written visitor logs, struggling with software or on-site servers. iVisitor is the simplest, hassle-free way to control visitors and property in a multi-tenant setting.

In typical electronic visitor management situations, the software is installed on site. For a multi-tenant facility, this can cause major headaches when the server goes down or the software crashes. No local IT? You are left trying to solve the problem yourself. iVisitor is different because it is online and managed by Veristream. A simple Internet connection brings you Veristream’s secure, encrypted and feature-rich visitor management system. Your visitor management is worry free. • iVisitor is designed for multi-tenant situations. • There are no servers or software to maintain and no need for IT support. • There are an unlimited number of user authorities, including tenant, management and security users. • Uptime rate is 99.995%

**2.3 Summary**

The researcher, cannot but have the same idea with the scholars, that our computerized visitor’s management system needs total restructuring before effective and accurate means of recording information could be conducted, by using specialized method as check to ensure that data collected are channeled to the proper hands.

In most developing countries revenue collection and occupant willing to pay the levy imposed on the government property he/she is using, is what prompted this idea of business premises payment system. Occupants (landlords) must ensure that their tenants register all types of businesses established within the street or area as the case may be.

**CHAPTER THREE**

**System Analysis and Design**

**3.0 Introduction**

In this chapter, the analysis and choice of method used in this research work, will be analyzed from the existing system and design. It includes specific methods which were used in order to achieve the aim of the research work, particular requirements for implementation of the project and clear explanation of reasons why such method were used for design and implementation of the proposed system, also included is a brief description of the current system of visitors management.

The methodology used is Unified Modeling Language (UML). Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software. In this article we will give you detailed ideas about what is UML, the history of UML and a description of each UML diagram type, along with UML examples.

UML is designed to enable users to develop an expressive, ready to use visual modeling language. In addition, it supports high level development concepts such as frameworks, patterns and collaborations.

UML diagrams can be divided into two categories. The first type includes six diagram types representing structural information. The second includes the remaining seven representing general types of behaviour. Structure diagrams are used in documenting the architecture of software systems and are involved in the system being modeled.

Different structure diagrams are:

* Class Diagram: represents system class, attributes and relationships among the classes.
* Component Diagram: represents how components are split in a software system and dependencies among the components.
* Deployment Diagram: describes the hardware used in system implementations.
* Composite Structure Diagram: describes internal structure of classes.
* Object Diagram: represents a complete or partial view of the structure of a modeled system.
* Package Diagram: represents splitting of a system into logical groupings and dependency among the grouping.

Behavior diagrams represent functionality of software system and emphasize on what must happen in the system being modeled.

Different behaviour diagrams are:

* Activity Diagram: represents step by step workflow of business and operational components.
* Use Case Diagram: describes functionality of a system in terms of actors, goals as use cases and dependencies among the use cases.
* UML State Machine Diagram: represents states and state transition.
* Communication Diagram: represents interaction between objects in terms of sequenced messages.
* Timing Diagrams: focuses on timing constraints.
* Interaction Overview Diagram: provides an overview and nodes representing communication diagrams.
* Sequence Diagram: represents communication between objects in terms of a sequence of messages.

UML diagrams represent static and dynamic views of a system model. The static view includes class diagrams and composite structure diagrams, which emphasize static structure of systems using objects, attributes, operations and relations. The dynamic view represents collaboration among objects and changes to internal states of objects through sequence, activity and state machine diagrams. A wide variety of UML modeling tools are available to simplify the modeling process, including IBM Rational Rose, Rational Rhapsody, MagicDraw UML, StarUML, ArgoUML, Umbrello, BOUML, Power

Class Diagram of UML



Fig 3.1 Class Diagram

Component diagram UML



Fig 3.2 Component diagram

**3.1 Analysis of the Existing System.**

The existing visitor’s management system is mainly manual (paper and pen) registration. The manual method/registration is done by given the visitor a visitor register to register his or her information.

After registering with the security officer he or she will be allowed to enter inside the organization. Later the visitor will return and sign out, to show that he/she have left the environment

**Figure 3.2: Sample of register used visitors.**

During analysis, the following problems were discovered in the above system:

* Cost oriented and time consuming,
* No good information flow,
* Inadequate archiving place.

**3.2 Analysis of the Proposed System**

The proposed system is a web based visitors management system, were the security officers will be using a computer system as a major tools for registering visitors.

The system have both admin end and the user end the:

The admin end is where the system are been controlled, granting access to the users which is the which is the security people in the duty post to control and manage visitors for proper management of data and information flow of any visitors.

At the user end, this is where direct interaction with the visitors occur. The registration is done by the security on duty post by telling the visitor to give him/ her the required information to be filled on the system.

**3.3 Design of the Proposed System**

The subheadings below will illustrate how the system will be operated.

**3.3.1 Database Design**

* Vms\_Admin
* Access\_tab
* Visitors\_in\_tab
* Visitors\_out\_tab

Table 1: Vms\_Admin

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field  | Data type | Size | Attributes | Null | Default | Extra |
| admin\_id | Int | 11 |  | No | None | AUTO\_INCREMENT |
| User\_name | Varchar | 100 |  | No | None |  |
| password | Varchar | 100 |  | No | None |  |

Table 2: Access\_tab

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field  | Data type | Size | Attributes | Null | Default | Extra |
| Access\_id | Int | 11 |  | No | None | AUTO\_INCREMENT |
| User\_n | Varchar | 100 |  | yes | Null |  |
| Pass\_d | Varchar | 100 |  | yes | Null |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field  | Data type | Size | Attributes | Null | Default | Extra |
| Visitors\_in\_id | Int | 11 |  | No | None | AUTO\_INCREMENT |
| F\_name | Varchar | 255 |  | yes | Null |  |
| Contact\_add | Varchar | 255 |  | yes | Null |  |
| Contact\_num | Varchar | 255 |  | yes | Null |  |
| Gender | Varchar | 25 |  | yes | Null |  |
| Purpose\_visiting | Varchar | 255 |  | yes | Null |  |
| Who\_to\_visit | Varchar | 255 |  | yes | Null |  |
| Tag\_no | Varchar | 25 |  | yes | Null |  |
| Time\_in | Varchar | datetime |  | yes | Null |  |

Table 3: Visitors\_in\_Tab

Table4: Visitors\_out\_tab

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field  | Data type | Size | Attributes | Null | Default | Extra |
| Visitors\_out\_tab\_id | int | 11 |  | no | none | AUTO\_INCREMENT |
| Visitors\_in\_id | varchar | 4 |  | yes | Null  |  |
| F\_name | Varchar | 255 |  | yes | Null |  |
| Contact\_add | Varchar | 255 |  | yes | Null |  |
| Contact\_num | Varchar | 255 |  | yes | Null |  |
| Gender | Varchar | 25 |  | yes | Null |  |
| Purpose\_visiting | Varchar | 255 |  | yes | Null |  |
| Who\_to\_visit | Varchar | 255 |  | yes | Null |  |
| Tag\_no | Varchar | 25 |  | yes | Null |  |
| Time\_in | datetime |  |  | yes | Null |  |
| Time\_out | datetime |  |  | yes | Null  |  |

**3.3.2 System Architecture**

HOME PAGE

ADMIN LOGIN

USER LOGIN

REGISTER VISITOR

GRANT USER ACCESS

SEARCH VISITORS

DATABASE

**Figure 3.3: System architecture**

|  |
| --- |
| REGISTRATION FORMSUBMITTAG NOWHOM TO VISITPURPOSE OF VISITINGGENDER CONTACT NUMBERCONTACT ADDRESSFULL NAME |

**3.3.3 Input Design**

**3.3.4 Output Design**

**TALLY CARD**

NAME

CONTACT NUMBER

YEAR OF ADMISSION

YEAR OF ADMISSION

YEAR OF ADMISSION

YEAR OF ADMISSION

**Figure3.5: Visitors Tally Card**

TALLY NUMBER

TIME IN

ALGORITHM FOR LOGIN PAGES

Step 1: Start

Step 2: Declare a variable username and password

Step 3: Read variable username and password

Step 4: if username= password

 Open a new page

 Else

 Display error message

 step 5: stop

**CHAPTER FOUR**

**System implementation**

**4.0 Introduction**

In this chapter we did a general introduction of the implementation of visitors management system how security will make efficient use of the newly developed system of visitors management form. The system is user friendly, easy to understand and operate.

**4.1 Choice of Development Environment.**

The system was developed as an interactive mechanism between the user at the interface, and the database using the web browser. It was designed using notepad++ and wamp server. These tools enable the admin and the user through a browser to interact with MySQL database to enter, edit, view, and retrieve data, as privilege granted. These activities were achieved using: HTML forms which offer the best layout to enter, change and view from database. This form was also kept as short and simple as possible to suit the individual who will provide data.

The choice of programming used is top-down approach. This approach was used because the project arrangement was in linear format and linear format is always from top to-down to achieve a goal.

**4.1.1 Admin Login Page**



**Figure 4.1: Screen shot of admin login page.**

This is where the administrator login is located. The username and the password provided here cannot be easily changed, except the administrator opens the MySQL database to make the change. Administrator is the only one who has the right to grant access to different units that will make use the system.

**4.1.2 Visitors Registration Form**



. **Figure 4.2: Screen shot of visitors registration form.**

 **4.2 Implementation Architecture**

HOME PAGE

ADMIN LOGIN

USER LOGIN

REGISTER VISITOR

SEARCH VISITOR

GRANT USER ACCESS

INPUT TAG ID NO

INSERT USERNAME & PASSWORD

REGISTER VISITOR

SUBMIT

PRINT

REGISTER VISITOR

LOGOUT

DATABASE

Figure 4. 3: Implementation architecture

**4.3 Software Testing**

During the development of the system, the application undergoes two phases of testing:

Firstly: Testing done during development phase. This testing includes:

* Syntax error testing: this method of testing is used to check all the syntax errors during the development. It also checks the programming pattern of the language, for instance dollar ($) is used to donate variable name in php while in java the variable name will just appear as ordinary text, in javascript it will use the name “var” to show variable name.
* Compatibility testing: this testing is used to test if all the languages used during the programming were compatible with each other.
* Logical testing: this is the argument involved during the programming. This will check whether the argument is accepted by the system or not.

Secondly, testing done by running the software on realistic data samples. This testing includes:

* Running the application in its fullness using a local server such as wamp server, xamp server, lamp server etc that have local host which is using Apache as engine and MYSQL as the database.
* Browser testing: this testing is used to test the appearance and shape of the software on the browser. This testing also replaces system testing in desktop application.

**4.3.1.1 INPUT SPECIFICATION**



**4.3.1.2 INPUT SPECIFICATION**



**Figure 4.4: Screen shot of the input form**

Once all the requested information are provided and click on submit the data will immediately send to database for storage and retrieval.

**4.3.2 Output Specification**



Figure 4.5: Clearance form for graduating students

**4.4 Documentation**

As said earlier, that this system was built with basic HTML tags, PHP, CSS, and MySQL language, they are open source program that allows modification to occur in future.

HTML: This is an acronym for Hyper Text Makeup Language; it is a language for describing the web pages. Html form elements are elements that allow the user to enter information (like text fields, text area fields, drop-down menus etc.) in a form

CSS: This is the acronym for cascading style sheets, these codes are used to style html elements, when html tags are written and viewed on a web browser, it doesn’t really appear in animated format, so we used CSS to style the html tags.

PHP: Hypertext preprocessor is widely used scripting language that was originally designed for web development, to produce dynamic web pages. its codes can be embedded into html tags.

MySQL: This is my serial query language; it uses a standard form of well-known SQL data language. Different data types were used in this project. It was used alongside with PHP to create and submit forms that are used in the software.

The developed system is packaged in a CD drive in a directory named *visitors management system.*

**4.4.1 User Manual**

Procedure on how to execute the program:

1. Boot the system to windows
2. Install notepad++ and wampserver 3.0.6.
3. Launch notepad++ and wampserver.
4. Open the index page with notepad++ and run the software using a browser by locating the file using your server name *localhost* and then point the index page.

The system has the following menu commands for the admin:

Staff section

1. *Grant Access to user.* On clicking this menu it will display a form that the admin will use to give username password to the users.
2. *Search*. On clicking this menu it will display search box where the visitors tag no will be entered and sign him or her out.

User section

1. *Register*. On clicking this menu it will open form that will request for visitors data.
2. *Login*. When the argument in this page is true it open a registration form for the user.

4.4.2 Source Code

The source code of this system will be attached in the appendix.

**CHAPTER FIVE**

SUMMARY, RECOMMENDATIONS AND CONCLUSION

**5.1 Summary**

The web base visitor’s management system will help manage visitor’s information, time, period and purpose of visiting an institution or organization.

Is also a means through which the security personnel determines the number of visitors visited in a particular period of time, it help them during report writing.

 Visitors management in protected areas and has become increasingly important in a world where business is flourishing, the area of the organization is growing, as the organization is demand for remote, unique destinations. Visitor management used to be focused on carrying capacity and minimizing the environmental effects of the organization, but evolved to adopt many different approaches in order to achieve the goal of environmental stewardship and visitor satisfaction

**5.1 Conclusion**

I concluded that using web based management system to control visitors flow will be beneficial for an organization that buy the idea because of its security purpose and loss of data are drastically reduces since the database is online and cannot be interred with physical damages.

**5.2 Recommendation**

After carefully observed the research project, I found out that the web based visitors management system will be best practice to put in use, especially in this era of computer technology where human beings embrace any method that involves the use of computer system.

**REFERENCE**

[1] Drumm, Andy, and Moore, A. MANAGING VISITOR IMPACTS. 1st ed, 2005 [ebook] pp.4-22. Available at: http://sanctuaries.noaa.gov/management/international/pdfs/day3\_impacts\_manual.pdf

[2] Hagan, R. New school visitor management technologies help keep school safe, 2012. [Online] Available at: http://www.seenmagazine.us/articles/article- detail/articleid/2010/smid/403/articlecategory/56/new-school-visitor-management-technologies-help-keep-school-safe.aspx.

[3] Morrissey, P. The 5 Problems with Manual Visitor Management | Articles. [online] Magic-card.com.au, 2015. Available at: <http://www.magic-card.com.au/articles/the-5-problems-with-manual-visitor-management.html>.

[4] McArthur,S. Beyond carrying capacity: introducing a model to monitor and manage visitors activity in forests. In:Font. X. AND Tribe,J. (Editors).Forest Tourism and Recreation: Case studies in environmental management. Pp: 259-278. CABI Publishing, New York, 2000.

[5] Sampada Khot, T. P. R. a. M. R. Visitor Management System Using GSM. International Journal of Modern Trends in Engineering and Research (IJMTER), 2(4), p. 1, 2015.

[6] Rick Hagan. New school visitor management technologies help keep school safe, 2012. [online] <http://www.seenmagazine.us/Articles/Article-Detail/ArticleId/2010/smid/403/ArticleCategory/56/New-school-visitor-management-technologies-help-keep-school-safe>

[7] Joan Engebretson and Whelan. Visitor Management Adds Efficiencies for Busy High Rise, 2009. [Online] <http://www.easylobby.com/CustomerSuccess/One%20Gateway%20Center%20-%20Visitor%20Management.pdf>

[8] Andrews International (AI) Process Visitors Efficiently With Visitors Management System, 2014. [Online] <http://www.andrewsinternational.com/technology/documents/technology-visitor-management.pdf>

[9] Veristream. Secure Visitor Management System For Multi-Tenant Facilities, 2012. [Online]

<https://www.anixter.com/content/dam/Suppliers/Veristream/Veristream_iVisitor_MutiTenant.pdf>

APPENDIX 1

The codes would be broken down into section for the different page on the web application.

Code for: ***Connect.php Page***

*<?php*

*$vms = mysqli\_connect("localhost","root","")or die('could not connect to servre'.mysqli\_error());*

*$vms\_select = mysqli\_select\_db($vms, "vms\_db")or die('could notselect server'.mysqli\_error());*

*?>*

Code for: ***db.php Page***

*<?php*

*include('connect.php');*

*$vms\_db = mysqli\_query($vms, "create database if not exists vms\_db")or die('could not create databse'.mysqli\_query($vms));*

*?>*

Code for: ***tab.php Page***

*<?php*

*include('connect.php');*

*$vms\_tab = mysqli\_query($vms, "create table if not exists vms\_admin(*

 *admin\_id int not null primary key auto\_increment,*

 *user\_name varchar(100) not null,*

 *password varchar(100) not null*

 *)*

*")or die('could not create table vms\_admin'.mysqli\_error($vms));*

*$vms\_check\_admin = mysqli\_query($vms,"select \* from vms\_admin*

 *where*

 *user\_name ='vms' and password = 'vms'*

*")or die('could not check the vms\_admin table'.mysqli\_error());*

*$vms\_row = mysqli\_num\_rows($vms\_check\_admin);*

*if($vms\_row<1)*

*{*

 *$vms\_insert = mysqli\_query($vms, "insert into vms\_admin values(*

 *'0',*

 *'vms',*

 *'vms'*

 *)")or die ('could not insert into admin\_table'.mysqli\_query($vms));*

*}*

*$acess = mysqli\_query($vms, "create table if not exists access\_tab(*

 *access\_tab\_id int not null primary key auto\_increment,*

 *user\_n varchar(25),*

 *pass\_d varchar(25))*

*")or die('could not create access\_tab table'.mysqli\_error($vms));*

*$visitors\_tab = mysqli\_query($vms,"create table if not exists visitors\_in\_tab(*

 *visitors\_tab\_id int not null primary key auto\_increment,*

 *f\_name varchar(255),*

 *contact\_add varchar(255),*

 *contact\_num varchar(255),*

 *gender varchar(25),*

 *purpose\_visiting varchar(255),*

 *whom\_to\_visiting varchar(255),*

 *tag\_no varchar(25),*

 *time\_in datetime)*

*")or die('could not craete visitors table'.mysqli\_error($vms));*

*$visitors\_tab\_2 = mysqli\_query($vms,"create table if not exists visitors\_out\_tab(*

 *visitors\_out\_tab\_id int not null primary key auto\_increment,*

 *visitors\_tab\_id varchar(4),*

 *f\_name varchar(255),*

 *contact\_add varchar(255),*

 *contact\_num varchar(255),*

 *gender varchar(25),*

 *purpose\_visiting varchar(255),*

 *whom\_to\_visiting varchar(255),*

 *tag\_no varchar(25),*

 *time\_in datetime,*

 *time\_out datetime)*

*")or die('could not craete visitors table'.mysqli\_error($vms));*

*?>*

Code for: ***index.php Page***

*<?php*

*include('../connect/connect.php');*

*$error = array();*

*if(isset($\_POST['sub\_vms']))*

 *{*

 *$user\_name = $\_POST['u\_name'];*

 *$pasw = $\_POST['p\_word'];*

 *if($user\_name =='')*

 *{*

 *$error[] = "eneter user name<br>";*

 *}*

 *if($pasw == '')*

 *{*

 *$error[] = "enter password<br>";*

 *}*

 *$user\_check = mysqli\_query($vms, "select \* from access\_tab*

 *where*

 *user\_n = '$user\_name' and*

 *pass\_d = '$pasw'*

 *")or die('could not select from access\_tab table'.mysqli\_error());*

 *$user\_check\_num = mysqli\_num\_rows($user\_check);*

 *if($user\_check\_num<1)*

 *{*

 *$error[] = "user name or password is not correct<br>";*

 *}*

 *if(sizeof($error)<1)*

 *{*

 *$user\_sele = mysqli\_query($vms, "select \* from access\_tab*

 *where*

 *user\_n = '$user\_name' and*

 *pass\_d = '$pasw'*

 *")or die('could not select from access\_tab table'.mysqli\_error());*

 *$user\_sele\_num = mysqli\_num\_rows($user\_sele);*

 *if($user\_sele\_num>0)*

 *{*

 *$fetcher = mysqli\_fetch\_assoc($user\_sele);*

 *$access\_tab\_id = $fetcher['access\_tab\_id'];*

 *session\_start();*

 *$\_SESSION['vms\_u'] = $access\_tab\_id;*

 *echo $access\_tab\_id;*

 *header("location:reg.php");*

 *}*

 *}*

 *foreach($error as $show)*

 *{*

 *//echo $show;*

 *}*

 *}*

*?>*

*<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">*

*<html xmlns="http://www.w3.org/1999/xhtml">*

*<head>*

*<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />*

*<title>VMS</title>*

*<link rel="shortcut icon" href="../image/logo\_1.fw.png">*

*<link href="../css/css.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<!--wrapper-->*

*<div id="vms\_wrapper">*

*<!--heading-->*

 *<div class="vms\_head">*

 *<div class="vms\_h\_center">*

 *<div class="vms\_head\_inside">*

 *<!--site logo-->*

 *<div class="logo"><img src="../image/logo\_1.fw.png" class="logo\_real" /></div>*

 *<!--site title-->*

 *<div class="vms\_coine">VISITORS MANAGEMENT SYSTEM</div>*

 *<a class="vms\_admin" href="../owner/admin\_index.php">Admin Login</a>*

 *</div>*

 *</div>*

 *</div>*

 *<!--body-->*

 *<div id="vms\_wrapper\_in">*

 *<div class="vms\_guide\_form">*

 *<!--form guider-->*

 *<div class="form\_div">*

 *<div class="form\_div\_f">*

 *<label class="vms\_label">Login</label>*

 *<span class="eroor"><?php foreach($error as $show){echo $show;}?></span>*

 *<form action="" method="post" enctype="multipart/form-data" class="vms\_former">*

 *<input type="text" name="u\_name" class="vms\_f" placeholder="User name"/>*

 *<input type="password" name="p\_word" class="vms\_f" placeholder="Password"/>*

 *<input type="submit" name="sub\_vms" class="vv\_sub" value="Submit" />*

 *</form>*

 *</div*

 *</div>*

 *</div>*

 *</div>*

 *<div class="footer">*

 *<div class="footer\_c">Visitors Management System by Ozoilo Johnbosco Chibunna</div>*

 *</div*

 *</div>*

*</div>*

*</body>*

*</html>*

Code for: ***logout.php page***

*<?php*

*session\_start();*

 *if(isset($\_SESSION['vms\_u']))*

 *{*

 *session\_destroy();*

 *header("location:index.php");*

 *}*

*?>*

Code for: Reg***.php page***

*<?php*

*include('../connect/connect.php');*

*$error = array();*

*session\_start();*

*(isset($\_SESSION['vms\_u']))? $access\_tab\_id = $\_SESSION['vms\_u']:header("location:index.php");*

*if(isset($\_POST['reg\_sub\_vms']))*

*{*

 *$f\_name = $\_POST['f\_name'];*

 *$contact\_add = $\_POST['contact\_add'];*

 *$contact\_num = $\_POST['contact\_num'];*

 *$gen = $\_POST['gender'];*

 *$p\_of\_v = $\_POST['p\_of\_v'];*

 *$w\_to\_v = $\_POST['w\_to\_v'];*

 *$tag\_no = $\_POST['tag\_no'];*

 *if($f\_name == '')*

 *{*

 *$error[] = "Enter Full Name<br>";*

 *}*

 *if($contact\_add == '')*

 *{*

 *$error[]= "Enter Address<br>";*

 *}*

 *if($contact\_num == '')*

 *{*

 *$error[] = "Enter Phone number<br>";*

 *}*

 *if($gen =='')*

 *{*

 *$error[] = "select gender<br>";*

 *}*

 *if($p\_of\_v == '')*

 *{*

 *$error[] = "Enter the purpose of visiting<br>";*

 *}*

 *if($w\_to\_v == '')*

 *{*

 *$error[] = "Enter Whom you are visiting<br>";*

 *}*

 *if($w\_to\_v == '')*

 *{*

 *$error[] = "Enter tag number<br>";*

 *}*

 *$checker = mysqli\_query($vms, "select \* from visitors\_in\_tab*

 *where*

 *f\_name = '$f\_name' || contact\_num = '$contact\_num'*

 *")or die('could not select from visitors\_in\_tab'.mysqli\_error());*

 *$checker\_num = mysqli\_num\_rows($checker);*

 *if($checker\_num>0)*

 *{*

 *$error[] = "The Visitor Have Visited Before Use AdminAcount to update his/her bio-data<br>";*

 *}*

 *if(sizeof($error)<1)*

 *{*

 *$insert\_visitors = mysqli\_query($vms, "INSERT INTO visitors\_in\_tab values(*

 *'0',*

 *'$f\_name',*

 *'$contact\_add',*

 *'$contact\_num',*

 *'$gen',*

 *'$p\_of\_v',*

 *'$w\_to\_v',*

 *'$tag\_no',*

 *now())*

 *")or die('could not insert into visitoes\_in\_tab table'.mysqli\_error($vms));*

 *if($insert\_visitors)*

 *{*

 *$success = "Successfully Submitted<br>";*

 *}*

 *}*

 *foreach($error as $show)*

 *{*

 *//echo $show;*

 *}*

*}*

*?>*

*<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">*

*<html xmlns="http://www.w3.org/1999/xhtml">*

*<head>*

*<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />*

*<title>VMS</title>*

*<link rel="shortcut icon" href="../image/logo\_1.fw.png">*

*<link href="../css/css.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<!--wrapper-->*

*<div id="vms\_wrapper">*

*<!--heading-->*

 *<div class="vms\_head">*

 *<div class="vms\_h\_center">*

 *<div class="vms\_head\_inside">*

 *<!--site logo-->*

 *<div class="logo"><img src="../image/logo\_1.fw.png" class="logo\_real" /></div>*

 *<!--site title-->*

 *<div class="vms\_coine">VISITORS MANAGEMENT SYSTEM</div>*

 *<a class="vms\_admin" href="logout.php"> Logout</a>*

 *</div>*

 *</div>*

 *</div>*

 *<!--body-->*

 *<div id="vms\_wrapper\_in">*

 *<div class="vms\_guide\_form">*

 *<!--form guider-->*

 *<div class="form\_div\_reg">*

 *<div class="form\_div\_reg">*

 *<label class="vms\_label">Register Visitors</label>*

 *<form action="" method="post" enctype="multipart/form-data" class="vms\_former">*

 *<span class="eroor"><?php foreach($error as $show){echo $show;}?></span>*

 *<span class="success"><?php if(isset($\_POST['reg\_sub\_vms']))*

*{ if($insert\_visitors){echo $success;}}?></span>*

 *<input type="text" name="f\_name" class="vms\_f" placeholder="Full Name"/>*

 *<input type="text" name="contact\_add" class="vms\_f" placeholder="Contact Address"/>*

 *<input type="text" name="contact\_num" class="vms\_reg" placeholder="Contact Number"/>*

 *<input type="text" name="gender" class="vms\_reg" placeholder="Gender"/>*

 *<input type="text" name="p\_of\_v" class="vms\_f" placeholder="Purpose of Visiting"/>*

 *<input type="text" name="w\_to\_v" class="vms\_reg" placeholder="Whom to Visit"/>*

 *<input type="text" name="tag\_no" class="vms\_reg" placeholder="Tag No"/>*

 *<input type="submit" name="reg\_sub\_vms" class="vv\_sub" value="Submit" />*

 *</form>*

 *</div>*

 *</div>*

 *</div>*

 *</div>*

 *<div class="footer">*

 *<div class="footer\_c">Visitors Management System by Ozoilo Johnbosco Chibunna</div>*

 *</div>*

 *</div>*

*</div>*

*</body>*

*</html>*

Code for: ***Visitor\_search.php page***

*<?php*

*include('../connect/connect.php');*

*$error = array();*

*session\_start();*

*(isset($\_SESSION['vms']))? $admin\_id = $\_SESSION['vms']:header("location:admin\_index.php");*

 *if(isset($\_POST['sms\_ser']))*

 *{*

 *$sera = $\_POST['vms\_search'];*

 *if($sera == '')*

 *{*

 *$error[]= "input Correct data for searching <br>";*

 *}*

 *$checko = mysqli\_query($vms, "select \* from visitors\_in\_tab*

 *where*

 *tag\_no ='$sera' || contact\_num = '$sera'*

 *")or die('could not select from visistors\_in\_tab table'.mysqli\_error());*

 *$checko\_num = mysqli\_num\_rows($checko);*

 *if($checko\_num<1)*

 *{*

 *$error[]="Check the input supplied<br>";*

 *}*

 *if(sizeof($error)<1)*

 *{*

 *$main\_select = mysqli\_query($vms,"select \* from visitors\_in\_tab*

 *where*

 *tag\_no ='$sera' || contact\_num = '$sera'*

 *")or die('could not select from visitors\_in\_tab table'.mysqli\_error());*

 *$main\_select\_rows = mysqli\_num\_rows($main\_select);*

 *if($main\_select\_rows>0)*

 *{*

 *$fetcher\_array = mysqli\_fetch\_assoc($main\_select);*

 *$visit\_id = $fetcher\_array['visitors\_tab\_id'];*

 *$f\_n = $fetcher\_array['f\_name'];*

 *$con\_no = $fetcher\_array['contact\_num'];*

 *$tag = $fetcher\_array['tag\_no'];*

 *$time\_in = $fetcher\_array['time\_in'];*

 *}*

 *}*

 *}*

*?>*

*<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">*

*<html xmlns="http://www.w3.org/1999/xhtml">*

*<head>*

*<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />*

*<title>VMS</title>*

*<link rel="shortcut icon" href="../image/logo\_1.fw.png">*

*<link href="../css/css.css" rel="stylesheet" type="text/css" />*

*</head>*

*<body>*

*<!--wrapper-->*

*<div id="vms\_wrapper">*

*<!--heading-->*

 *<div class="vms\_head">*

 *<div class="vms\_h\_center">*

 *<div class="vms\_head\_inside">*

 *<!--site logo-->*

 *<div class="logo"><img src="../image/logo\_1.fw.png" class="logo\_real" /></div>*

 *<!--site title-->*

 *<div class="vms\_coine">VISITORS MANAGEMENT SYSTEM</div>*

 *<div class="vms\_admin\_2"><a href="grant\_access.php">Grant Access</a></div>*

 *<div class="vms\_admin\_2"><a href="logout.php">Logout</a></div>*

 *</div>*

 *</div>*

 *</div>*

 *<!--body-->*

 *<div id="vms\_wrapper\_in">*

 *<div class="vms\_guide\_view">*

 *<form action="" method="post" enctype="multipart/form-data" class="ss\_xx">*

 *<span class="eroor"><?php foreach($error as $show){echo $show;}?></span>*

 *<input type="search" name="vms\_search" placeholder="Search by tag Number or Phone Number" required="required" class="ser"/>*

 *<input type="submit" value="Search" name="sms\_ser" class="s\_ns"/>*

 *</form>*

 *<div class="display\_all">*

 *<div class="display"><?php if(isset($\_POST['sms\_ser'])){if($main\_select\_rows>0){echo*

 *'<span class="coco">Name:</span> '. $f\_n.'<br>'.*

 *'<span class="coco"> Contact Number:</span> '.$con\_no.'<br>'.*

 *'<span class="coco"> Tag Number:</span> '.$tag.'<br>'.*

 *'<span class="coco"> Time In:</span> '.$time\_in.'<br>'.*

 *$visit\_id*

 *;}}?>*

 *<a href="sign\_out.php?ou=<?php if(isset($\_POST['sms\_ser'])){if($main\_select\_rows>0){echo $visit\_id;}}?>">*

 *<?php echo "Sign\_out"?></a>*

 *</div>*

 *</div>*

 *</div>*

 *</div>*

 *</div>*

 *<div class="footer">*

 *<div class="footer\_c">Visitors Management System by Ozoilo Johnbosco Chibunna</div>*

 *</div>*

 *</div>*

*</div>*

*</body>*

*</html>*

Code for: ***sign\_out.php page***

*<?php*

*include('../connect/connect.php');*

*$error = array();*

*session\_start();*

*(isset($\_SESSION['vms']))? $admin\_id = $\_SESSION['vms']:header("location:admin\_index.php");*

*if(isset($\_GET['ou']))*

 *{*

 *$out= $\_GET['ou'];*

 *$select\_in = mysqli\_query($vms, "select \* from visitors\_in\_tab*

 *where*

 *visitors\_tab\_id = '$out'*

 *")or die('could not select from visitors\_in\_tab table'.mysqli\_error());*

 *$select\_in\_num = mysqli\_num\_rows($select\_in);*

 *if($select\_in\_num>0)*

 *{*

 *$feter\_in = mysqli\_fetch\_assoc($select\_in);*

 *$in\_tab\_id = $feter\_in['visitors\_tab\_id'];*

 *$f\_nm = $feter\_in['f\_name'];*

 *$con\_add = $feter\_in['contact\_add'];*

 *$cont\_num = $feter\_in['contact\_num'];*

 *$gn = $feter\_in['gender'];*

 *$pp =$feter\_in['purpose\_visiting'];*

 *$wtv =$feter\_in['whom\_to\_visiting'];*

 *$tg = $feter\_in['tag\_no'];*

 *$tm\_in = $feter\_in['time\_in'];*

 *$insert\_out = mysqli\_query($vms, "INSERT INTO visitors\_out\_tab values(*

 *'0',*

 *'$in\_tab\_id',*

 *'$f\_nm',*

 *'$con\_add',*

 *'$cont\_num',*

 *'$gn',*

 *'$pp',*

 *'$wtv',*

 *'$tg',*

 *'$tm\_in',*

 *now())*

 *")or die('could not insert into visitors\_out\_tab'.mysqli\_error($vms));*

 *if($insert\_out)*

 *{*

 *header("location:visitor\_search.php");*

 *}*

 *}*

 *}*

*?>*