**An Online Library Management System**

**Bradley Aluvala**

**Adm\_no**

**Submitted in partial fulfilment of the requirements of the Diploma of Business in Information Technology at Strathmore University**

**Strathmore Institute of Management and Technology**

**Strathmore University**

**Nairobi, Kenya**

**April 2021**

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a diploma by this or any other University. To the best of my knowledge and belief, the research documentation contains no material previously published or written by another person except where due reference is made in the research documentation itself.

Student: Bradley Aluvala

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_

Supervisor: name of supervisor

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_

Abstract

Borrowing of books by the students can be detrimental in that sometimes a student comes to borrow a certain book to read for a duration of time then ends up not returning the book or even losing the book. This as a result can cause great loss to the institution since books are borrowed and at times never returned back. In addition, students also find it challenging to borrow books from the library due the manual way of storing and lending out books in the book shelves. This further leads to the wastage of time before the student orders a certain book and records it before leaving with the book. In addition, librarians also find it challenging to lend books out to students and at the end of the day he/she cannot know the number of books that he has lend out to students. This later leads to time wastage for him to calculate manually and working with estimates of how many he has lend to students which is very inappropriate. The developed solution is an online library management system which aids in assisting the librarian issue a book to a student recording the time the student has borrowed and returned the book. The system also aids in calculating the charges in-case a student loses a book. The librarian also can be able to see how many books he has lend to students as well as how many books are returned by the students on a particular day. The system is developed using the waterfall methodology where the phases are sequentially implemented.

Table of Contents

[Declaration and Approval ii](#_Toc66806360)

[Abstract iii](#_Toc66806361)

[Table of Contents iv](#_Toc66806362)

[List of Tables viii](#_Toc66806363)

[List of Figures ix](#_Toc66806364)

[List of Abbreviations x](#_Toc66806365)

[Chapter: 1 Introduction 1](#_Toc66806366)

[1.1 Background of Study 1](#_Toc66806367)

[1.2 Problem Statement 2](#_Toc66806368)

[1.3 General Objective 2](#_Toc66806369)

[1.3.1 Specific Objectives 2](#_Toc66806370)

[1.4 Justification 2](#_Toc66806371)

[1.5 Scope and Limitations 3](#_Toc66806372)

[Chapter: 2 Literature Review 4](#_Toc66806373)

[2.1 Introduction 4](#_Toc66806374)

[2.2 Library Management Processes in Kenya 4](#_Toc66806375)

[2.2.1 Challenges Faced by the Technical Stuff in the Library 5](#_Toc66806376)

[2.3 Related Works 5](#_Toc66806377)

[2.3.1 E-Bay 6](#_Toc66806378)

[2.3.2 Evergreen 6](#_Toc66806379)

[2.3.3 BiblioteQ 7](#_Toc66806380)

[2.4 Gaps of Existing Systems 8](#_Toc66806381)

[2.5 Conceptual Framework 9](#_Toc66806382)

[Chapter: 3 Methodology 10](#_Toc66806383)

[3.1 Introduction 10](#_Toc66806384)

[3.2 System Development Approach 10](#_Toc66806385)

[3.2.1 Requirement Analysis 11](#_Toc66806386)

[3.2.2 System Design 11](#_Toc66806387)

[3.2.3 Implementation 12](#_Toc66806388)

[3.2.4 Testing and Integration 12](#_Toc66806389)

[3.2.5 System Deployment 12](#_Toc66806390)

[3.2.6 System Maintenance 12](#_Toc66806391)

[3.3 System Analysis 12](#_Toc66806392)

[3.3.1 Use Case Diagrams 13](#_Toc66806393)

[3.3.2 System Sequence Diagrams 13](#_Toc66806394)

[3.3.3 Class Diagrams 13](#_Toc66806395)

[3.3.4 Tools and Techniques to be Applied 13](#_Toc66806396)

[3.3.5 Hypertext Mark-Up Language 13](#_Toc66806397)

[3.3.6 Cascading Stylesheets 13](#_Toc66806398)

[3.3.7 Hypertext Preprocessor 14](#_Toc66806399)

[3.3.8 My Structured Query Language 14](#_Toc66806400)

[3.3.9 JavaScript 14](#_Toc66806401)

[3.4 System Design 14](#_Toc66806402)

[3.4.1 Database Schema 14](#_Toc66806403)

[3.4.2 Entity Relation Diagram 14](#_Toc66806404)

[3.5 System Deliverables and Milestones 14](#_Toc66806405)

[3.5.1 Students 15](#_Toc66806406)

[3.5.2 Administrator 15](#_Toc66806407)

[Chapter: 4 System Analysis and Design 16](#_Toc66806408)

[4.1 Introduction 16](#_Toc66806409)

[4.2 Requirement Analysis 16](#_Toc66806410)

[4.3 System Requirements 16](#_Toc66806411)

[4.3.1 Functional Requirements 16](#_Toc66806412)

[4.3.2 Non-Functional Requirements 17](#_Toc66806413)

[4.4 System Design 18](#_Toc66806414)

[4.4.1 Use Case Diagram 18](#_Toc66806415)

[4.4.2 Sequence Diagram 19](#_Toc66806416)

[4.4.3 System Sequence Diagram 20](#_Toc66806417)

[4.4.4 Class Diagram 21](#_Toc66806418)

[4.4.5 Entity Relationship Diagram 22](#_Toc66806419)

[4.4.6 Database Schema 23](#_Toc66806420)

[Chapter: 5 System Implementation and Testing 25](#_Toc66806421)

[5.1 Introduction 25](#_Toc66806422)

[5.2 Description of the Implementation Environment 25](#_Toc66806423)

[5.2.1 Hardware Specifications 25](#_Toc66806424)

[5.2.2 Software Specifications 26](#_Toc66806425)

[5.3 Description of Testing 27](#_Toc66806426)

[5.3.1 Testing Paradigm 27](#_Toc66806427)

[5.3.2 Subset of Functional and Non Functional Requirements Tested 27](#_Toc66806428)

[5.4 Testing Results 29](#_Toc66806429)

[5.4.1 Authentication 29](#_Toc66806430)

[5.4.2 Creating a Students Record 30](#_Toc66806431)

[5.4.3 Issuance of a Book 30](#_Toc66806432)

[5.4.4 Availability 31](#_Toc66806433)

[Chapter: 6 Conclusions, Recommendations and Future Works 32](#_Toc66806434)

[6.1 Conclusions 32](#_Toc66806435)

[6.2 Recommendations 32](#_Toc66806436)

[6.3 Future Works 32](#_Toc66806437)

[References 33](#_Toc66806438)

[Appendix 35](#_Toc66806439)

List of Tables

[Table 5.1 Hardware Specifications (Minimal and Recommended) 25](#_Toc66806440)

[Table 5.2 Developer’s Computer Specifications 25](#_Toc66806441)

[Table 5.3 User Interface Dependencies 27](#_Toc66806442)

[Table 5.4 Authentication Test Results 29](#_Toc66806443)

[Table 5.5 New User Creation Test Results 30](#_Toc66806444)

[Table 5.6 Book Issuance Testing and Results 30](#_Toc66806445)

List of Figures

[Figure 2‑1 E-Bay 6](#_Toc49592617)

[Figure 2‑2 Auto Auction Mall 7](#_Toc49592618)

[Figure 2‑3 Salvage Bid 8](#_Toc49592619)

[Figure 2‑4 Conceptual Framework 9](#_Toc49592620)

[Figure 3‑1 Modified Waterfall Methodology 11](#_Toc49592621)

[Figure 4‑1 Use Case Diagram 20](#_Toc49592622)

[Figure 4‑2 Sequence Diagram 21](#_Toc49592623)

[Figure 4‑3 System Sequence Diagram 22](#_Toc49592624)

[Figure 4‑4 Class Diagram 23](#_Toc49592625)

[Figure 4‑5 Entity Relationship Diagram 24](#_Toc49592626)

[Figure 4‑6 Database Schema 25](#_Toc49592627)

List of Abbreviations

CRUD - Create Retrieve Update Delete

CSS - Cascading Style Sheet

DBMS - Database Management System

DFD - Data Flow Diagram

HTML - Hyper Text Markup Language

JS - JavaScript

MySQL - My Standard Query Language

OLMS - Online Library Management System

PHP - Hypertext Preprocessor

USD - Use Case Diagram

# 

# Introduction

## Background of Study

If some small libraries use traditional library management system, they maybe waste resources. According to the present situation of multi-hierarchical architecture development of information system, the author analyzed thoroughly the Flex, Spring and Hibernate frameworks. It integrated the frameworks to design a set of sufficient flexible, loose coupling, expandable and high effective Library Management System by mean of using Flex as presentation layer. Hibernate as enduring layer and in combination with Spring as business layer. (Wang, 2010).

According to Robertson (2004), Library management system (LMS) known as an automated Library System is software that is developed to handle basic functions of a Library, and provides a complete solution for the administration of a library’s technical functions and services to the public. These functions range from; tracking the assets held by the library, managing lending, through to supporting the daily work over. These systems are used in almost all libraries large and a small. (Robertson, 2004)

 According to Kampala International University Dar es Salaam Campus website www.kiu.ac.tz, Kampala International University Dar es Salaam Campus began operations in January 2009 at Quality Plaza along Pugu road. Currently, the university center is situated on a 100-acre piece of land, at Gongo la Mboto area in Ilala District, 7 Km from Mwalimu Julius Nyerere International Airport along Pugu road. The spacious campus offers an ideal university learning atmosphere and plenty of room for further expansion. The first phase of development is almost complete. It is a constituent college of Kampala International University found in Uganda. The University envisions becoming a prominent International in the great lakes’ region and beyond, nurturing talents in multicultural learning environment and advancing market-driven courses. Currently, Kampala International University Dar salaam Campus uses a manual library management system. This wastes a lot of time for students and librarians especially when students are borrowing or returning books. Currently when a student wants to borrow a book, he/she goes to the counter of the reference section and asks the librarian for the book he/she wants. If the book is in the stock, the librarian gets the book, register the details of the book and the student and then give the book to the student. This process, especially during busy hours of the day 11-5, delays students. (Richard, 2012)

## Problem Statement

In many institutions of learning, library management system has become a problem due to continued use of manual supported system. Jomo Kenyatta University is among those affected by the problem due to increased number of students. The current manual system makes it tiresome to serve students adequately. This is seen when a number of student’s queue for not less than 10 minutes to borrow a book from a librarian who has got to register each and every student and the book borrowed manually hence wasting a lot of time. This leads to perceived inefficiency and its associated high cost of labor hiring several librarians to carry out this activity.

## General Objective

The main objective was to develop an online library management system, which will aid in students borrowing books and also when books arrive they are also counted as they go out to shelves. In addition, the system aims to help librarian know how many books have been lend to students as well as how many books have been returned by the students in a particular duration of time.

### Specific Objectives

1. Analyzed the challenges faced by students who want to borrow books and librarians to know how many books have been borrowed and returned.
2. Analyzed the requirement of the online library management system.
3. Designed and developed a library management system that facilitates in issuing books to students.
4. Tested and implemented the OLMS system to ensure that data is validated before entry to the database and issuing and returning books by students are done easily.

## Justification

The developed system facilitates in aiding simplicity as far as issuing a book to a student and returning a book by a student is concerned. The developed system also facilitates in helping librarians to know how many books have been issued in a particular day and how many books have been returned in a particular day. In addition, the librarian can also charge a student if he or she loses a book.

The system was designed in a way that a student can only be able to see the books he has borrowed and returned while the admin/librarian can be able to monitor number of books borrowed and also compute charges in case a student loses a book.

## Scope and Limitations

The system allows registration of new student and after borrowing a book a student can be able to view the time he borrowed and time he returned on the other end the librarian can view books borrowed can add new books and authors and charge a student in case of a lost book. Despite the fact that the system can do all this, it also has some limitations. The system cannot set a deadline for a student who has borrowed a certain book although it can charge the student in case the student loses a book.

# Literature Review

## Introduction

In detailed explanations, this chapter explains how online library management system have been carried out previously and how issuance of books was done, including all the limitations that the existing system used to face. In addition, illustrating the advantages and disadvantages of the developed system.

## Library Management Processes in Kenya

Library is regarded as the brain of any institutes, of course many institutes understand the importance of the library to the growth of the institute and their esteem users which we categorically call the students. An integrated library system, also known as a library management system (Adamson et al., 2008) is an enterprise resource planning system for a library, used to track items owned, orders made, bills paid, and users who have borrowed. The Library Management System is a Library Management software for monitoring and controlling the transactions in a library (Ashutosh and Ashish., 2012). Library Management System supports the general requirement of the library such as the acquisition, cataloguing, circulation and other sections. Before the advent of computer in modern age there are different methods of keeping records in the library. Records are kept in the library on shelves and each shelf are labelled in an alphabetical or numerical order, in which the categories of books available are arranged on different position on the shelves and as well are recorded on the library manuscript and when any book is to be referenced the manuscript is being referred to, to know the position of such required book by the person that requested for the book.

After the invention of computer7 different researchers have carried out various approach on an automated library management system in which this project is as well all about. Another Library Management System is the Capital’s library software with the following benefits Increases support available for staff and users in any modern library service, provides efficiency, innovative system that’s saves library time and improves the user experience.

A library management system usually comprises a relational database, software to interact with that database, and two graphical user interfaces (one for users, one for staff). Most Library Management System separate software functions into discrete program called modules, each of them integrated with a unified interface. Prior to computerization, library tasks were performed manually and independently from one another. Selectors ordered materials with ordering slips, cataloguers manually catalogued items and indexed them with the card cataloguing system (in which all bibliographic data was kept on a single index card), and users signed books out manually, indicating their name on cue cards which were then kept at the circulation desk.

### Challenges Faced by the Technical Stuff in the Library

New tools of information technology have absolutely changed the role & responsibilities of librarians. A number of studies have been conducted to explore the problems faced by librarians. Given section reviews the studies conducted at International level in general and particularly in developing countries to investigate the problems confronted by the librarians. Ado mi and Annie, (2006) in their research on computer literacy skills of professionals in Nigerian University libraries concluded that most of the professionals do not poses high level of computer skill and their use of computer and technology is still maturing. They recommended that library management and leaders should organize and offer in-house20 computer training programmed for librarians and enough computers should be provided in this regard.

Many institutions either have no library or inadequate collection of textbooks. Professional’s status was also found very low, low pay scale and limited opportunities for promotion. In Nepal, Siwakoti, (2008) found that there was no government agency to control, monitor and evaluate the school libraries activities. There was lack of awareness programs, budgetary constraints, inadequate space, inadequate library materials, lack of trained and skilled manpower and lack of appropriate government policy and lack of information literacy. Ademodi and Adepoju, (2009) investigated the computer skill among librarians in academic libraries on Ondo and Ekiti State in Nigeria. It was found the shortage of computers and computer skills among professionals. The study recommended that more attention and funds should be provided for training and procurement of ICT infrastructure in Nigerian University libraries. For computerization purpose, library administration should solicit funds and assistant from foreign agencies and foundations who are interested for the cause.

## Related Works

This are systems advanced in order to conduct the auctioning process easily. In addition, these systems have been implemented and tested to the public in order to solve the problems in which live auctioneers face. The following systems include: E-bay, Auto Auction Mall, Salvage Bid.

### E-Bay

Koha is one of the most advanced, free and open source Integrated library management software (ILMS). Introduced in 1999, Koha has been used by thousands of libraries across the world. Users are impressed with this software simply because of its features. The software is viable, scalable and ideal for all kinds of libraries.

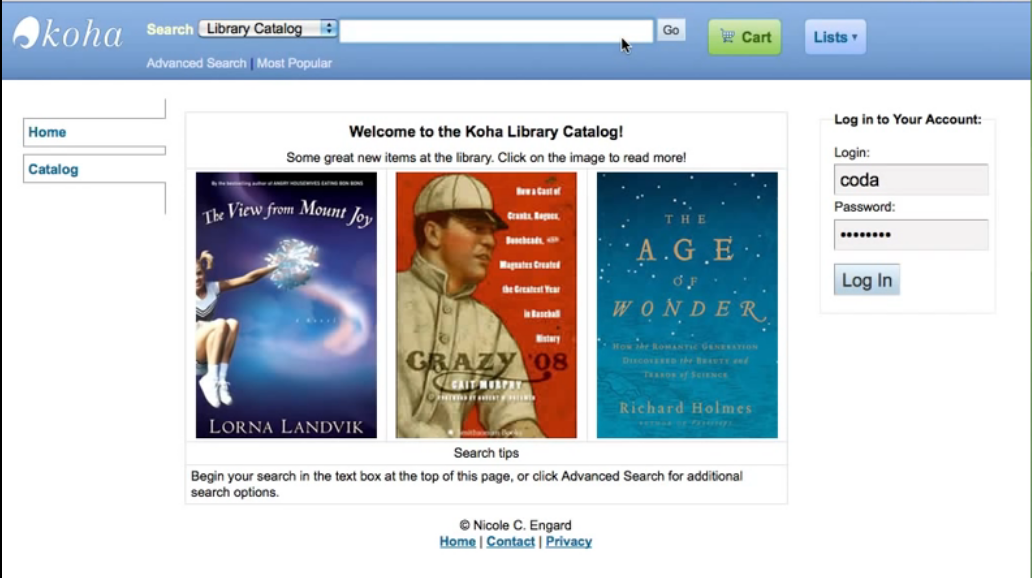


Figure ‑ Koha

### Evergreen

Used by nearly 2000 libraries around the world, this open source and free library management software offers public catalogue interface along with features that help users with the back-end workflow operations too. The software was first developed by the Georgia Public Library system back in 2006, and the community is growing big, and the software has been continuously evolving to meet the needs of users. Some of the features of evergreen include: open scalable request framework, online public access catalogue, powerful search facility and complete tracking of book procured, invoiced etc.

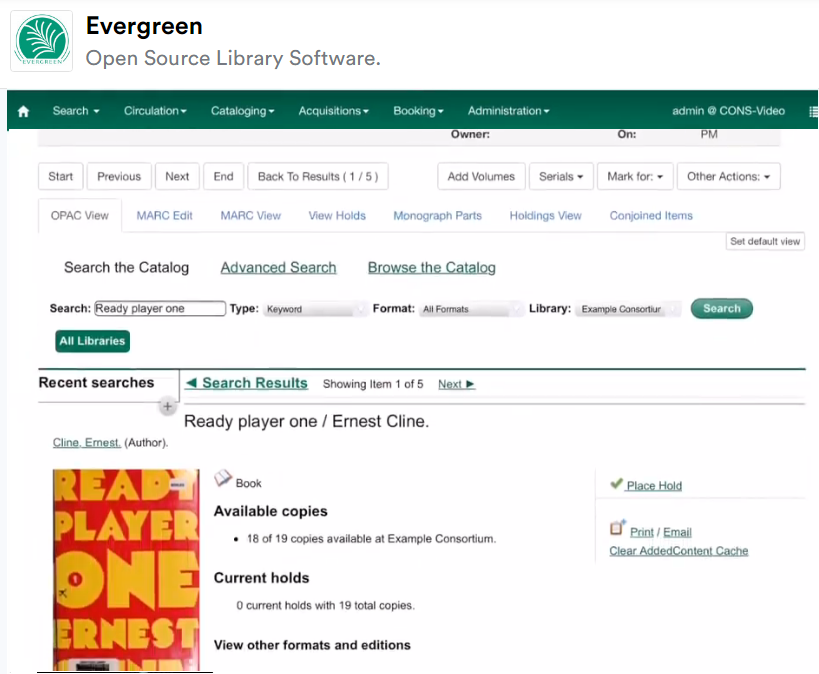


Figure ‑ Evergreen

### BiblioteQ

BiblioteQ is a professional library management solution and cataloguing system adopted by several small, medium and large libraries. The software is compatible with many of the Qt supporting systems. BiblioteQ is a free and Open Source library management software system. Some of the features of BiblioteQ is its user-friendly interface, totally support ARM architecture customized display facility, Language translation and notification of unavailable items.

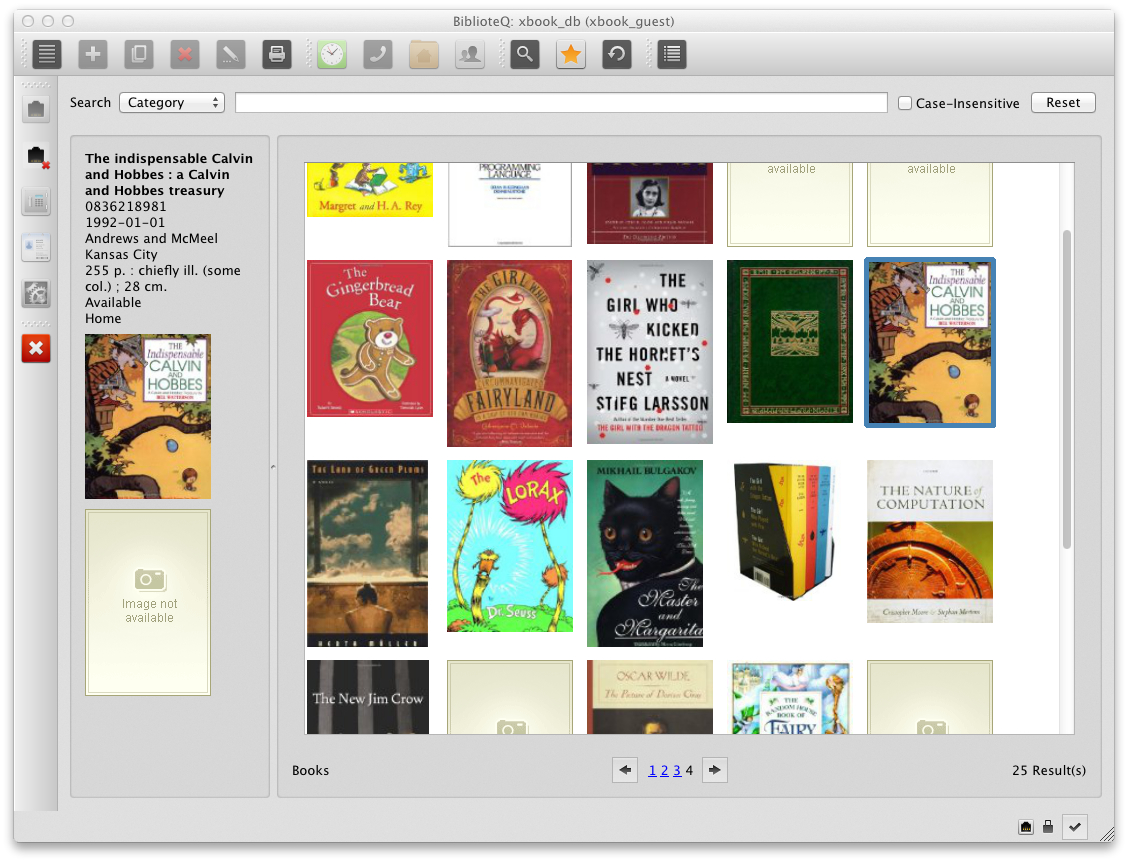


Figure ‑ BiblioteQ

## Gaps of Existing Systems

There are some systems already implemented and deployed in the market that serve different countries in which they facilitate issuance and returning of books by people in an online platform although they lack a number of artefacts, which are going to be resolved with my developed system. There is need for a system to facilitate in an admin adding an author and a certain book in the book shelves. There is need for a system that which will allow the admin to be able to monitor how many books have been borrowed and how many have been returned in a certain day, and there will be also a need for a system which the admin will be able to suspend a student who fails to return a certain book or declines to pay a book within a specified duration of time. The developed OLMS system solves this problem through admin being able to count the number of students who have borrowed a certain book in a particular date and time and also the admin can be able to suspend or block a student who fails to return a book or who also fails to pay for a book yet he wants to be able to borrow more books in the library.

## Conceptual Framework

The figure 2.4 is an explanation of how the OLMS system operates as well as how different operations are carried out. The entities involved the OLMS system are: users and administrators. The student logins to the system only if registered, the student can be able to check the books that he has borrowed at a particular date a time and the books he has returned and if not the fine that is imposed by the admin who is the librarian. An administrator is responsible for managing all students registered to the database, managing all authors, books and managing divisions currently in the system. He can be able to perform CRUD analysis to all the registered books in the database.

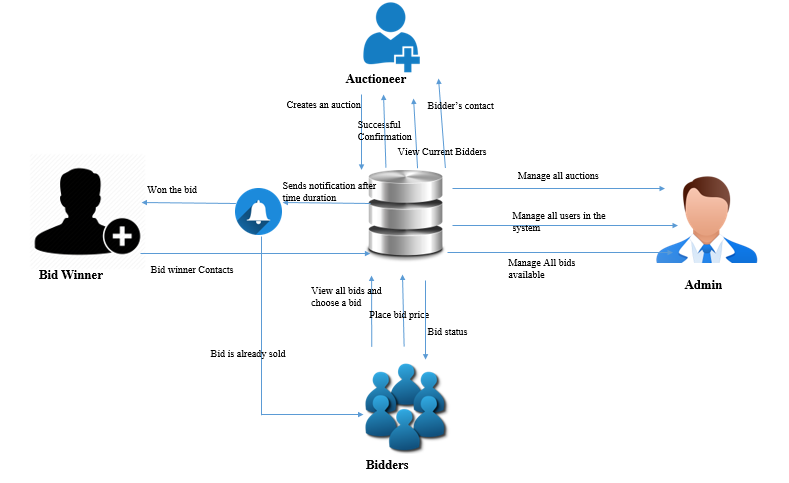


Figure ‑ Conceptual Framework

# Methodology

## Introduction

This chapter illustrates on the different techniques and methodologies that were used in developing of the OLMS system. The developed system is a web-based application in which the student can be able to borrow books in the library and the admin can be able to verify if the student has returned a book or not in which if the student has not returned the book then the admin will pose a charge based on the price of the book. The system methodology to be applied is the object-oriented analysis and design, whose main focus is on capturing the real-world objects in the current scenario that are of importance to the system. This methodology stresses more on data structure, as opposed to procedural structure. In this approach, objects are identified, and their relationships amongst each other, possible states that each object can be in, and finally how all objects collaborate with each other to achieve a broader system goal are identified.

## System Development Approach

The OLMS system was developed using the modified waterfall methodology, in which phases were sequentially implemented in software development. Waterfall model is a system development life cycle whereby the system was implemented sequentially which means one cycle completed followed by the other cycle. Using the waterfall model aids in ensuring data integrity and efficiency.

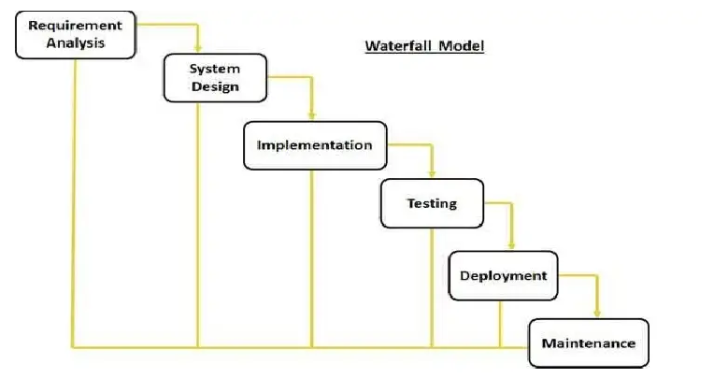


Figure ‑ Modified Waterfall Methodology

### Requirement Analysis

The requirements of the OLMS system are identified. User requirements is evident in the system, whereby a registered student can be able to borrow a book and the system facilitates in returning the specific date and time in which the student borrows a book there after the admin can be able to check the number of registered students as well as the number of registered books or issued books to the students. Security being a nonfunctional requirement is also evident in the system whereby both the student and the admin are prompted to enter their username and password in order to be able to access the system to be able to carry out different operation in their specified accounts.

### System Design

A description of system architecture is explained in logical and physical designs. In logical design, the input like the student’s and admin’s email and password are required by the user for the system to enable the user to perform certain functionality in the system like admin issuing a book to a student and the student being able to check the status of his book either returned or fined if the deadline surpasses. In the physical design, data inputs are verified in that after the student enters the correct credentials, he is authenticated to his user dashboard and can be allowed to make other operations.

### Implementation

Coding takes place in this phase. Information is taken in the previous stage by programmer’s in which they create a functional product. Code is implemented in small pieces in which integration is done at the end of this phase. Various functionalities which require intelligence are met. For example, a student cannot be able to clear or rather be able to change the status to book “returned”. Likewise, a student cannot be able to register an email which has already been registered in the database, it has to be unique.

### Testing and Integration

At the end of the coding process, product testing is formally started. Testers sequentially find and report any problems. In case of a very serious issue, your project may need to return to phase one for reevaluation in order to make corrections for the testing and integration to be done again. Certain functionalities were tested in the system. For example, a student cannot be able to borrow another book if he has lost the book that he or she had previously borrowed. In case the user tried to borrow a book and he had lost previously the admin will notify him to first pay the lost book in order to be able to borrow another book.

### System Deployment

After system testing, the system was taken to a small institute first in order to see its benefit to the few. Thereby after feedback from the user on how the system is, it is further inaugurated to function in all counties in Kenya.

### System Maintenance

A number of dilemmas might arise when released to the market or due to the major arise of the IT experts to conduct an upgrade to the system. Therefore, the system is therefore maintained by creating a newer version each and every time to curb this problem. Refactoring was also implemented whereby, restructuring the existing code was done to improve its performance without altering its external behavior.

## System Analysis

System analysis is the process of data collection, problem identification and recommendation of feasible suggestion of improving the functionality of the problem. The major objectives of system analysis are to find out what is being done, how to do, and how it can be improved. There are various system analysis diagrams, tools and techniques used in implementation of the developed system. They include:

### Use Case Diagrams

A use case diagram models the functionality of the system using actors present in the system and use cases implementation. Use case diagrams was used in order to show who are the actors in the system, their specific roles to perform in the system.

### System Sequence Diagrams

The system sequence diagram shows different scenarios of a use-case, generation of events by external actors. The system sequence diagram was used to elucidate the life line of each entity and the roles to take at specific times within the system.

### Class Diagrams

A class diagram was used to explain the class structure by showing their classes, attributes, and object relationship amongst others. The developed system illustrated different types of objects present in the system and how they relate with each other in order to facilitate in achievement of the objectives required by the buyer and the auctioneer of the product.

### Tools and Techniques to be Applied

The following tools and techniques are to be applied in the full development of the developed system. They include:

### Hypertext Mark-Up Language

HTML is a standard mark-up language designed in order to be displayed in the web browser. They include display of plain text, links, picture elements, sound and videos. Html are designed with special opening and closing tags. HTML as a front-end design was used in full development of the OCAS system, like user login page, home page and other pages.

### Cascading Stylesheets

CSS is a style sheet language designed for presentation of data written in html document in a satisfactory way. Unlike html which is used in creating the skeleton layout of the document, CSS is used for adding styles to the designed html document for example adding color, animation, responsive images to your web document.

### Hypertext Preprocessor

PHP is a server-side scripting language designed for interaction with the server. PHP is embedded with HTML as it is used to fetch data filled in the client side and fill into the database. Information filled by the user in the system was captured and posted to the database with aid of PHP.

### My Structured Query Language

MySQL is a relational database management system embedded with PHP in order to facilitate in storage of user’s information into a database. Information is written on the client-side using HTML and MySQL stores the information in the database.

### JavaScript

JavaScript is a scripting language used to embed the front end to make it look more responsive and proper readability and understandability. It is also embedded with the client side before a record is stored in the database to ensure robust information is fed to the database.

## System Design

This phase elucidates the diagrams to be used in development of the OLMS system and the roles they play respectively.

### Database Schema

A database schema is structured diagrammatically to represent the relations in the database management system. A database schema was used to illustrate how different entity in the system relate to one another and the multiplicities in which all the entities inhibit.

### Entity Relation Diagram

An entity relational diagram illustrates entity stored sets in a database. It is used to sketch out the design of a database. In development of the OLMS system, the student cannot be able to borrow or be issued a certain book without presence of a system. The admin can directly relate with the database as well as the students but the students cannot be able to do a lot of modification in the database, they can only be able to view the status of the operation they had earlier on carried out.

## System Deliverables and Milestones

This entails what the developed system is expected to do, and the different tasks that the entities are to perform within the system.

### Students

The Student can be able to do self-registration and after registration System will issue Student-ID. After login student can view own dashboard. Student can update own profile. Student can view issued book and book return date-time. Student can also change own password. Student can reset or recover own password.

### Administrator

The system admin can be able to do many operations in the system. These include: adding, updating and deleting a category. Adding, updating and deleting an author. Adding, updating and deleting a book. Admin can issue a new book to student and also update the details when student returns a book Admin can search student by using their student ID. Admin can also view Student Details. Admin can change own password.

# System Analysis and Design

## Introduction

This chapter focuses on the analysis and design that was used in development of the OLMS system. It explains in details how the system operates, giving the various analysis techniques used and finding the problem, finding the system’s general and specific designs and a theoretic representation of how the solution is implemented.

## Requirement Analysis

This stage focuses on the analysis techniques that were performed in order to meet the needs and conditions of the users. The requirement analysis technique which is used in development of the OLMS system is observation whereby the analyst generally observes how the current system operates and identifies the enigma experienced in the current system, thereby developing a methodical procedure of how to solve this dilemma.

## System Requirements

This are the systems configuration requirements requisite in order for a hardware or software to run efficiently and smoothly.

### Functional Requirements

#### Universal Login

A universal login is usable for already registered users in the system in which they are prompted to enter their username and user account password. The user and the admin have their own login page in which they are prompted to enter their username and password then the system can be able to facilitate in verification before it redirects to their respective Dashboard.

#### Data Accessibility

The OLMS system is available and accesses in the country with the aid of internet services, in order for students to register and be able to start in borrowing books in the library.

#### Data Integrity

The OLMS system aids in providing accuracy of data in that when a user enters incorrect username and correct password or vice-versa, and error message is displayed showing that the details are incorrect. Likewise, when a user successfully authenticates himself to the system, and goes to update his profile. After updating his details, the system inherits the new credentials and the user can use the new credentials in order to login to the system again, otherwise an error message is resulted.

#### System Security

System Security is met whereby when a student successfully logs in to his student’s dashboard, and after the student logs out the system clears all the sessions in which even when you press the back button on the web browser it will not be able to return to its former page, and therefore for the student to be able to access his or her account, he or she must login again to the system to create another session.

#### History Provision

OLMS system aids in providing history of all the issued books a student has been issued and returned respectively and incase of disputes, the system is also has the exact date and time in which a student borrowed a book from the library.

### Non-Functional Requirements

#### Performance

In the system homepage when the user accesses the webpage, the page takes less than 4 seconds to load. Before redirecting the user to the homepage, a welcome message is displayed then redirects him to his dashboard. The system is available for 24 hours 365 days in a year with scheduled exception and pre notified system maintenance downtime if needed.

#### Interoperability

The system works in different operating systems as from windows 7 onwards to windows 10 and all Linux versions (most preferably in Ubuntu version).

#### Portability

The web-based application is designed in such a way that it is responsive to big devices and smaller devices like it is portable in Laptops, desktops, tablets and mobile phones of different sizes.

#### Reliability

The system software, hardware and software, satisfactorily performs the task designed for within the specified period of time and environment in which it was designed to operate.

## System Design

This phase focuses on the system designs tools and techniques that were used with diagrammatic illustrations, relationships and entity descriptions.

### Use Case Diagram

A use case diagram is a representation of user’s interaction with the system, including all the actor’s roles in the system and entities included in the system. The system has two actors namely: the auctioneer (seller) and the bidder (buyer). OCAS system allows the clients to register, in which they are entitled to a duet option either to auction a car or take part in auctioning an auctioned car. If the user makes a choice to auction a car, he inputs all the integral details of the car, and set a duration of expiry of the auction in which the bid winner is declared immediately the auction expires. In case a user makes a decision to take part in auctioning a car, he is redirected to all current live bids and he makes a choice to bid either product.

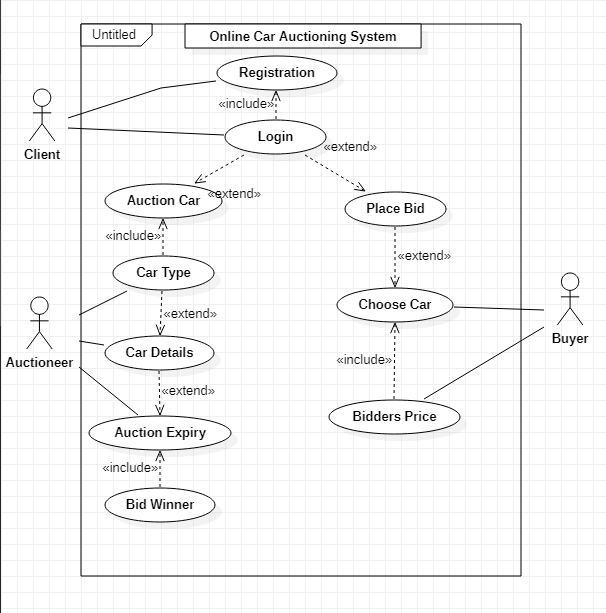


Figure ‑ Use Case Diagram

### Sequence Diagram

A sequence diagram is used to illustrate interactions between parts within a system to carry out a specified task. In relation with the OLMS system, the entities incorporated are as follows: the bidder, the auctioneer, the administrator and the auction database. The auctioneer having the major role in the system, who lists an auction and specifies the duration for expiry in which the information is sent to the auction database, a verification is sent to the auctioneer for a successfully done transaction. The bidder on the other end searches for available auction in which he makes a choice of the car he wants to participate in bidding that car. A successful confirmation is sent to the user when all legal conditions are met, otherwise an error message is sent to the user. After the bid expiry the bid becomes unavailable as a live bid and system facilitates in determining the highest bidder to be the bid winner.

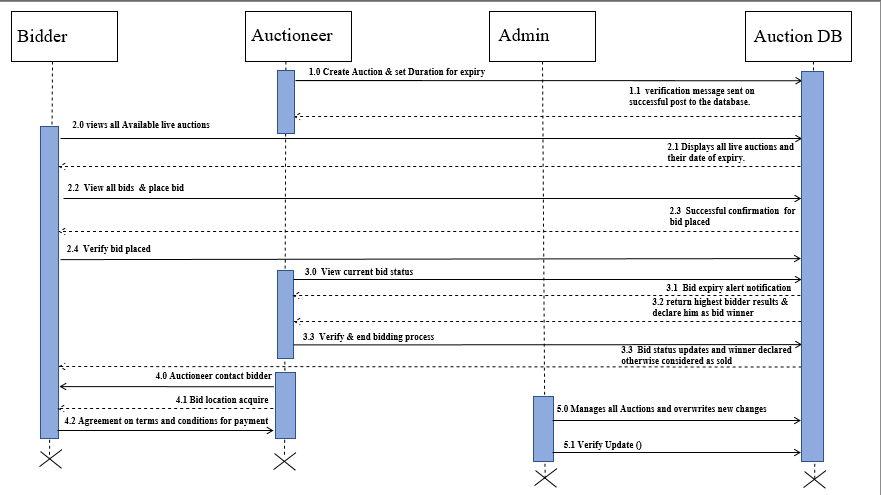


Figure ‑ Sequence Diagram

### System Sequence Diagram

A system sequence diagram is used to illustrate the relationship and interaction between the external entities of the system and the main system which facilitates main interaction.

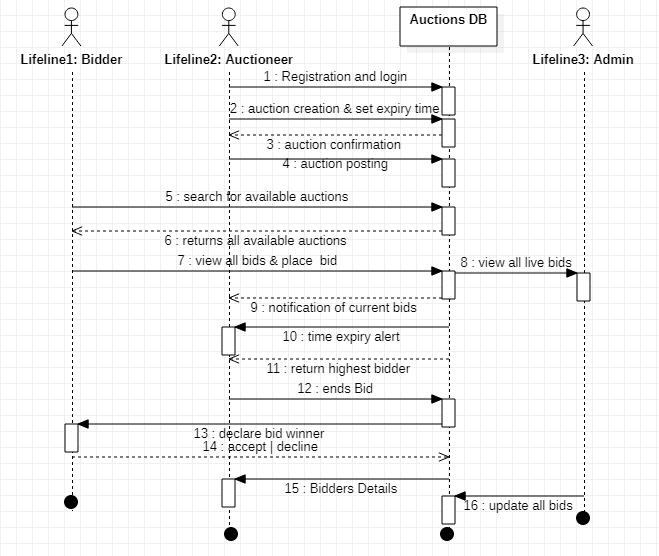


Figure ‑ System Sequence Diagram

### Class Diagram

A class diagram is a static structure diagram describes the system’s structure showing its classes their operation and attributes and relationship among other objects. In relation with the OCAS system, the attributes involved in the system are the auctioneers, the bidders, the admin, and the products.

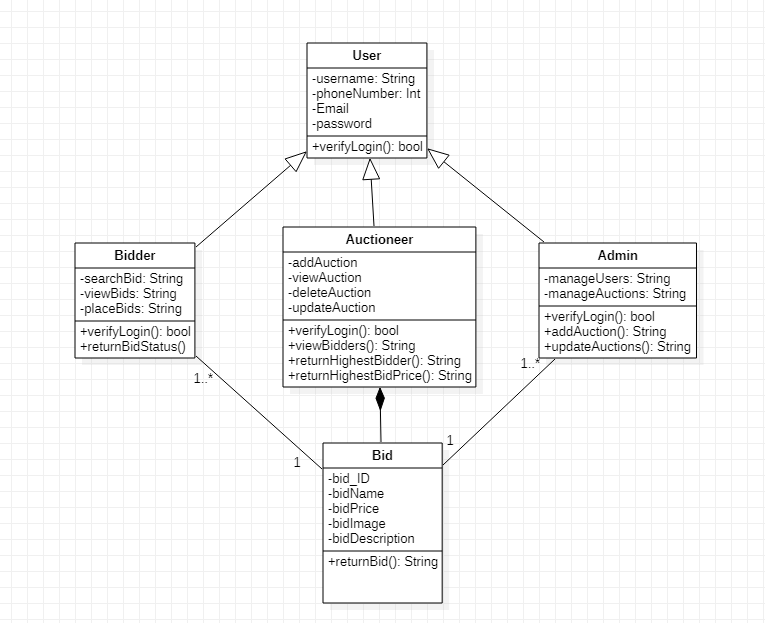


Figure ‑ Class Diagram

### Entity Relationship Diagram

An ERD show the relationship of entities set stored in the database. Each entity is enticed to an attribute showing relationships between each other within the system.

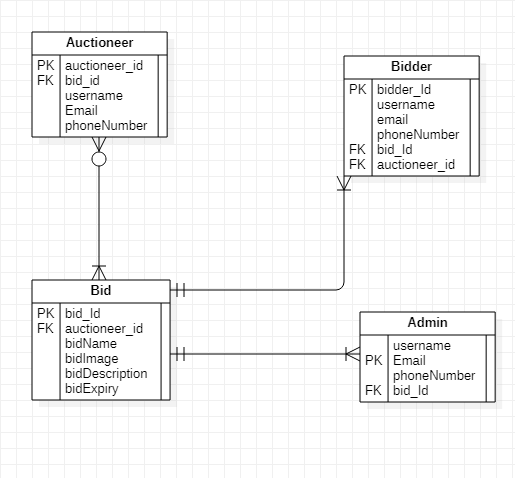


Figure ‑ Entity Relationship Diagram

### Database Schema

A database schema is a structural delineation of the entire database with all the defined relations involved in the system. The OCAS system has four relations which include: the administrator, the bidders, the products and the orders.

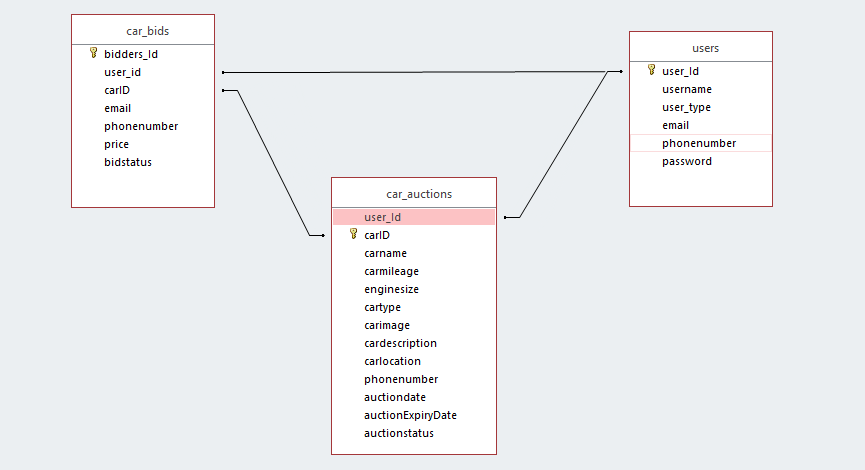


Figure ‑ Database Schema

# System Implementation and Testing

## Introduction

This chapter focuses on how the system development was done, the physical design, hardware specifications, software specifications and the system dependency. In addition, the chapter explains how system was carried out in order to meet high standards.

## Description of the Implementation Environment

This section explains in precisely on the systems hardware and software specifications required for the system to fully and functionally operating.

### Hardware Specifications

Table 5.1 below, shows an explanation of how the system’s minimal and recommended requirements to aid in running the online car auctioning system.

Table 5.1 Hardware Specifications (Minimal and Recommended)

|  |  |  |
| --- | --- | --- |
| **Item** | **Minimal Specifications**  **(Both the Web and The Database Server)** | **Recommended Specifications (Both the Web and Database Server)** |
| **Processor** | 2 x 1.6GHZ CPU | 4 x 1.6GHZ CPU |
| **RAM** | 2GB | 4GB |
| **Hard-Disk Storage** | At least 40 GB free space is the most recommended | At least 40 GB free space is the most recommended |

The table 5.2 below, is used to show the hardware specifications in development of the Online Library Management System.

Table . Developer’s Computer Specifications

|  |  |
| --- | --- |
| **Item** | **Specifications** |
| **Processor** | 3.80GHZ |
| **RAM** | 8GB RAM |
| **Hard-Disk Storage** | 500GB SSD |

The developer’s computer meets all the minimum recommended specifications in order to develop and test the system.

### Software Specifications

The system was implemented using the LAMP (Linux, Apache, MySQL, PHP) environment.

#### Operating System

The Online Car auctioning System was built to operate in Ubuntu Linux most preferably the latest version (Ubuntu 19.10) Long Term Support (LTS). It can also operate in a Windows Operating system most preferably Windows 10 64bit OS. However, the system can also run in other Linux distro’s like Linux mint as well as other versions of Windows.

#### Web Server

OLMS was designed with configuration of an Apache Web Server version 2.4. It is compatible with other versions of web servers.

#### Database Server

OLMS was designed and operates with MySQL database management system. Other related versions are the Maria DB version 10.4.11. It is also compatible with other versions of MySQL. However, for the system to run efficiently, some certain aspects in the data manipulations like SELECT, INSERT, UPDATE and DELETE syntax have to be functional for the system to perform its operations.

#### Dependencies

OLMS utilize the (SS) dependency which means than for activity B to start A must have already started in order for B operation to start. For example, for you to view your bidders you must have an auction in the system. In addition, the OCAS system utilizes AdminLTE a popular open source web application template for administrators control panel and dashboard. Features like HTML5 &CSS3 are also available. Table 5.3 are required for Admin LTE to function appropriately to produce a rich user experience.

Table 5.3 User Interface Dependencies

|  |  |
| --- | --- |
| **Plug-in** | **Description** |
| Date picker | A JavaScript library with supporting CSS styling that provides a flexible date picker widget in the Bootstrap style. |
| Time picker | A JavaScript plug-in that provides a simple time picking component in Bootstrap style. |
| I-Check | A JavaScript plug-in that provides identical HTML inputs across all major browsers and devices on both desktop and mobile. |
| Data Tables | A plug-in for the jQuery JavaScript library that provides highly advanced features for HTML tables from any client or server-side data source such as search functionality, pagination and multi-column ordering. |

## Description of Testing

Software testing is done in order to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements. It is also done in order to validate and verify if the software solution solves the problems which were earlier on established and meets the user’s expectations. The goals for testing the software testing is also to find out the system faults and bugs which may result to system failure to operate as expected.

### Testing Paradigm

The testing paradigm used in the system was black box testing. In this case, the user tests the system’s functionality without going into further details about how the system was developed. The user inputs a simple data to the system and checks if the results displayed by the system is the correct details of what it is expected to produce as the outcome.

This testing method was put into consideration in order to suit the people who don’t really need to know the inner working of the system but just need the system to aid in helping them trading their items. Although some white-box testing was involved in order to find out the system functionality, data structures and system design which aids in future system maintenance.

### Subset of Functional and Non Functional Requirements Tested

This testing technique was undertaken in order to find out the functional requirements which entails the system and its components as well as the system non-functional requirements.

#### Authentication

Valid and Invalid credentials were inputted in order to check the login functionality. The systems’ logout functionality was tested in order to find out whether the sessions were cleared and deleted when the user logs out of the system. The system Authorization roles were also tested in order to ensure that different users logging in to the system are redirected to their specific pages for example and admin is redirected to his dashboard as well as the student’s, via a universal login.

#### Creating a User Record

Validation in various forms like user phone numbers were checked in order to ensure that the user enters a valid email and also the system checks to ensure that the email provided is also not registered in the database otherwise it will throw an error of “email already exists.”

#### Search

OLMS system provides search functionality in order for an to search a particular book or author and incase a student wants to borrow another book the admin also can be able to check if the student has an outstanding fine and in case he has a fine he must pay the fine before being allowed to borrow another book.

#### Reports

OLMS system also facilitates in generating of reports of a student who wants to see the transaction of books that he has done in the library and the books borrowed and the books returned with the exact date and time issued and returned respectively.

#### Security

Security being a non-functional requirement. System security was tested in basis of the system database, to ensure it is not attacked by SQL injection as well as a user account confidentiality to access user auction history in that a user is prompted to enter his password in order to access his full modification property of his auction.

#### Availability

Availability being a non-functional requirement was implemented to ensure that the user can access the site in different platforms like in a laptop or pc, a tablet, or a mobile phone, responsively.

## Testing Results

Testing results are shown in terms of functional and non-functional requirements as previously outlined.

### Authentication

The table 5.4 illustrates the tests that were carried out during authentication and the results which were obtained afterwards.

Table 5.4 Authentication Test Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **#** | **Description** | **Test Data** | **Expected Outcome** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Verification to ensure that the user enters the correct username and correct password | Username is ‘Bradley’ password is ‘brad123’ | A success message, displayed and user is directed to the user-homepage. | As Expected | Pass |
| 2. | Verification to ensure that user cannot login with a valid username and invalid password | Username is ‘Bradley’ and password ‘brad’ | An incorrect username or password message displayed in the login page and the user is prompted to enter valid credentials | As Expected | Pass |
| 3. | Verification to ensure that user cannot login with an invalid username and valid password | Username is ‘Bradleyy’ and password ‘brad123’ | An incorrect username or password message displayed in the login page and the user is prompted to enter valid credentials | As Expected | Pass |
| 4. | Login Verification when the username and password are left blank and submit button clicked | Username and password fields empty | Text tool tip alerts user to input a username | As Expected | Pass |

### Creating a Students Record

The table 5.5 shows the tests which were carried out and the results obtained in creating a new auction by a user.

Table 5.5 New User Creation Test Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **#** | **Description** | **Test Data** | **Expected Outcome** | **Actual Result** | **Status (Pass/Fail)** |
| 9 | Verify that the email address entered must be unique. | ‘bradleyaluvala @gmail.com’ | User alerted that the email entered already exists. | As Expected | Pass |

### Issuance of a Book

Table 5.6 illustrates the test carried out to ensure that user auctioning functionality works as intended to work.

Table 5.6 Book Issuance Testing and Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Description** | **Test Data** | **Expected Outcome** | **Actual Result** | **Status (Pass/Fail)** |
| 9. | Verify to ensure that the admin has not entered invalid student ID | ‘1000000000’ | Error message is sent to the admin of invalid Student ID. Please enter valid student ID. | As Expected | Pass |
| 10. | Verify to ensure the admin enters correct ISBN Number. | 333020 | Error message sent to the admin of invalid ISBN number. | As Expected | Pass |

### Availability

Table 5.8 illustrates the testing undertaken to prove that the user interface of the system fits the viewport size and responsiveness when operating in different platforms.

Table 5.8 User Interface Responsiveness Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Description** | **Test Data** | **Expected Outcome** | **Actual Result** | **Status (Pass/Fail)** |
| 11. | Verification of the user interface to fit the screen in small devices whose viewport size is 576 and less. | Browser size scaled down to represent small device with width of 576px | Navigation-Menus, Tables, Images and other user interface elements arranged in a stack to allow for neat presentation of information. | As Expected | Pass |
| 12. | Verification to ensure the user interface scales to fit screen of medium devices with a viewport size of 768 pixels or less | Browser size scaled down to represent small device with width of 768px | User interface elements stacked or scaled to a minimum of 60% of actual size | As Expected | Pass |

# Conclusions, Recommendations and Future Works

## Conclusions

Most of the Online Library Management System which are prevalent allows a student to be able to borrow books even after the student has lost a book. OLMS system has addressed this issue in a way that a student can be allowed to borrow more than one book but if he does not return one of the book he is fined and cannot be allowed to borrow another book. In addition, other systems implemented allows a student to check the status of the book if returned or not returned but this OLMS allows the student to be able to monitor his exact time in which he was issued a certain book and the time that he has returned a book for personal records. In addition, the OLMS also allow the admin to be able to block a user to be unable to access the system in case a student is defiant in paying a book which he lost.

## Recommendations

OLMS was implemented and deployed an apache HTTPS web server which is probably the most used and PHP 7.0 although I would recommend running the web server with the latest version of PHP for also better performance and efficiency.

For optimum usage, I would strongly recommend to run this system in Ubuntu 20.04LTS version or a windows 10 operating system of at least 4GB RAM processor speed.

## Future Works

Although there are other similar systems designed for Online Library Management System, I would recommend to the future programmers, who may want to develop a similar project to implement other added functionality like in case a student loses a book then he can then be able to do online M-PESA payment to instantly pay for the book and the system automatically removes the fine for the student in order for the student to be able to borrow another book in the library again. I would also recommend that the system should be in a way that also the student can be able to check how many people have borrowed a certain book in order to see if the book is a good book or not that good by judging on how many people have borrowed that book.

References

Bidsuite. (2018, October 02). *Bidsuite*. Retrieved from Traditional and modern auctions: https://www.bidsuite.app/articles/traditional-vs-modern-auctions/

Bidsuite. (2018, October 02). *traditional and modern auctioning* . Retrieved from Traditional and Modern Auctions: https://www.bidsuite.app/articles/traditional-vs-modern-auctions/

Business, T. (2012, April 4). *Business Today*. Retrieved from online auction gaining ground in Kenya: https://businesstoday.co.ke/online-auction-gaining-ground-in-kenya/

Charu Aggarwal, P. S. (2009, July). *research gate*. Retrieved from Online auctions: There can only be one.: https://www.researchgate.net/publication/221542999\_Online\_Auctions\_There\_Can\_Be\_Only\_One

Computer Hope. (2018, November 12). *https://www.computerhope.com/basic.htm: Computer Hope*. Retrieved from Computer Hope Web Site: https://www.computerhope.com/basic.htm

Institute, R. (2018, June 6). *How internet technology is changing auctions*. Retrieved from auctions over the internet: https://www.rliland.com/online-auctions-internet-technology-changing-auctions/

Muneeswaran Packiyaraj. (2016, August). *research gate*. Retrieved from An effective online auctions: https://www.researchgate.net/publication/310615150\_An\_effective\_online\_auction\_system

Parsonburg. (2020, January). *Auctioneers and Appraisers*. Retrieved from Online bidding: http://www.amauctions.com/live-online-auctions/online-bidding-explained/

Richard, A. (2012). *Electronic Library Management System*. Retrieved from https://www.grin.com/document/205391

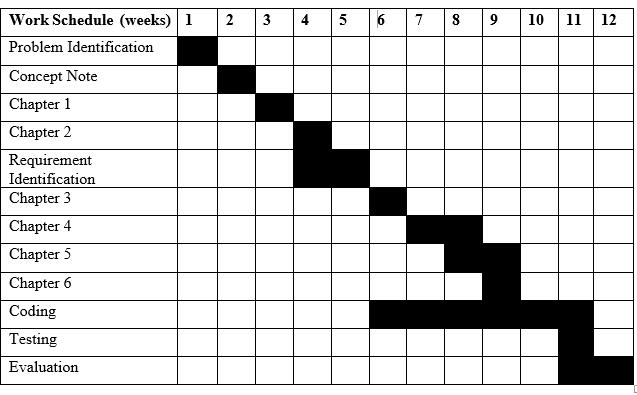
Robertson. (2004). Retrieved from Electronic library management system: grin.com/document/205391

Today Business. (2012, April 04). *Business Today*. Retrieved from Online auction gaining ground in Kenya: https://businesstoday.co.ke/online-auction-gaining-ground-in-kenya/

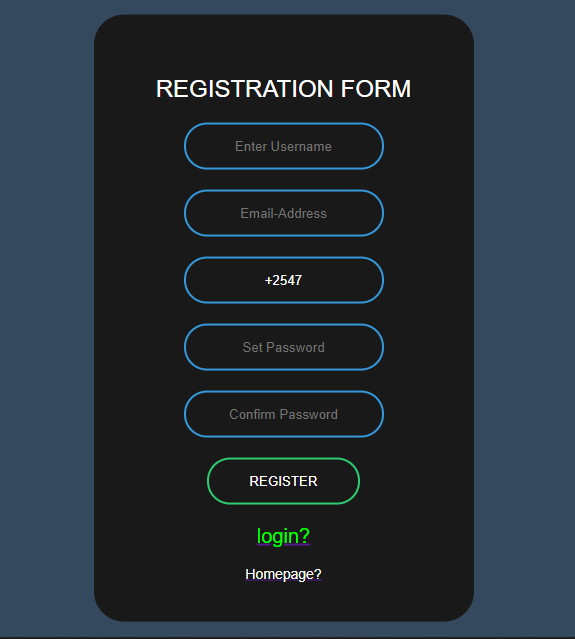
Wang, M. (2010, March). *research gate*. Retrieved from The building of Library Management System Based on Hibernate Model: https://dl.acm.org/doi/10.1109/CESCE.2010.255

Appendix

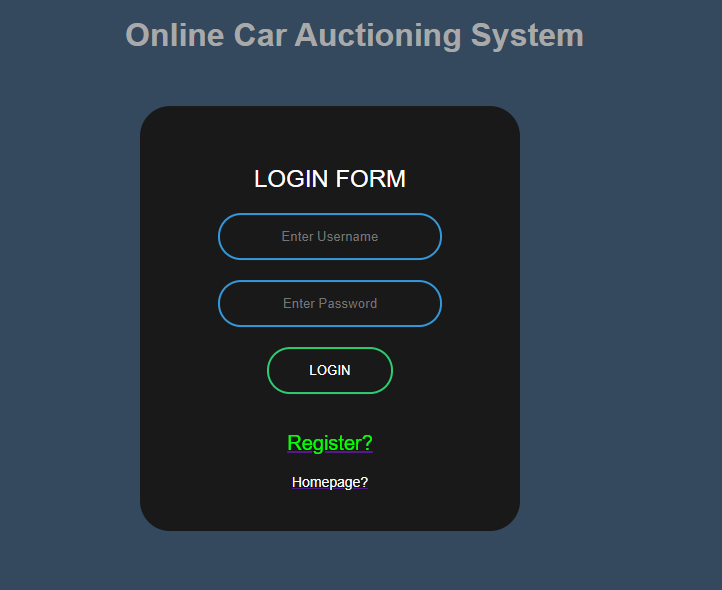
Appendix A: Gantt Chart



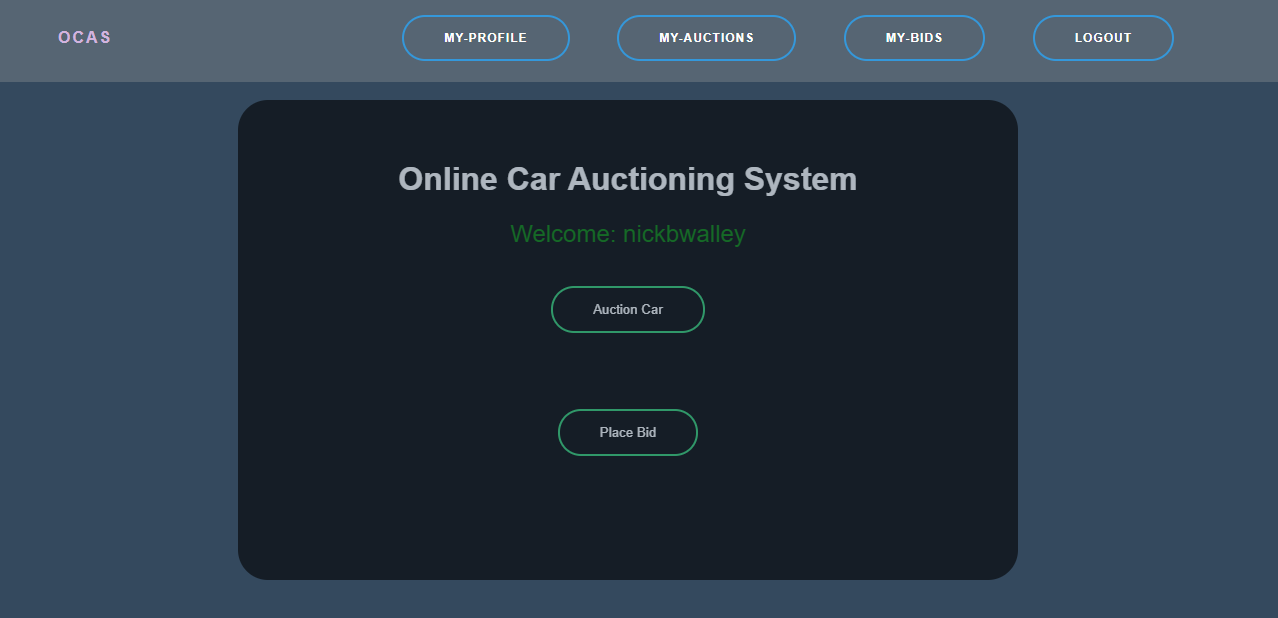
Appendix B: Registration Page



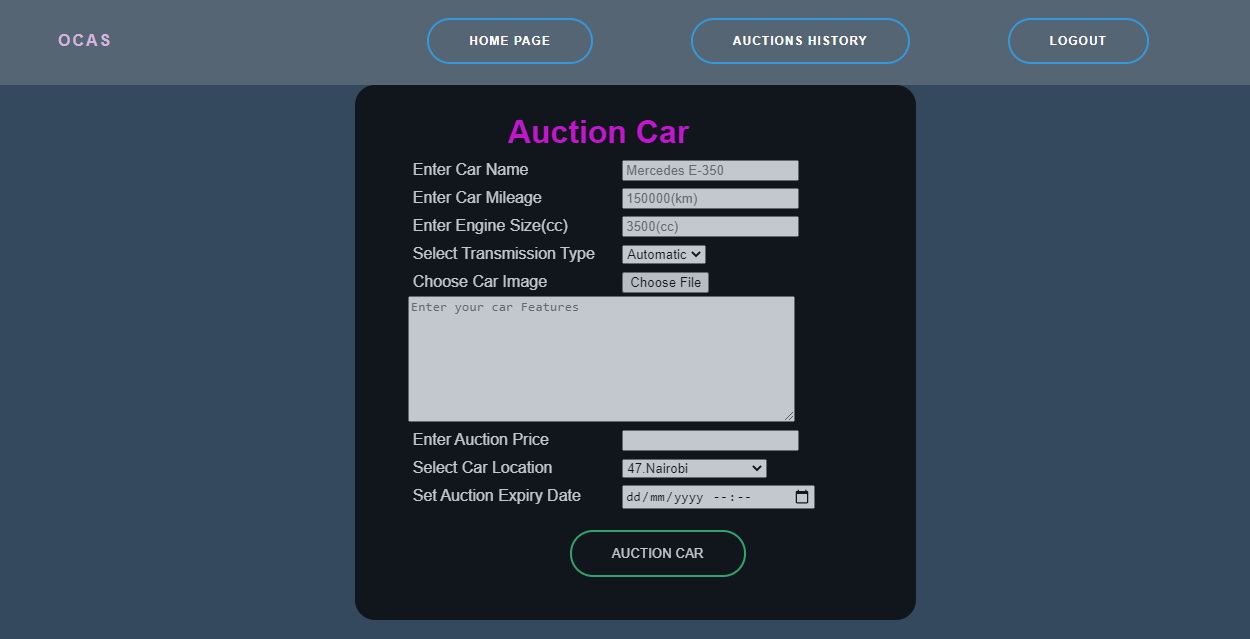
Appendix C: Login Page



Appendix D: Home Page



Appendix E: Auctions Page



Appendix F: Bid Page

