**DEVELOPMENT OF A COMPUTER BASED TESTING APPLICATION USING THE CODEIGNITER FRAMEWORK**

**BY**

**UKEH VICTOR UGOCHUKWU**

**16CJ020768**

**A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE BACHELOR OF ENGINEERING DEGREE IN COMPUTER ENGINEERING. COVENANT UNIVERSITY.**

**SUPERVISOR(S):**

**PROF. E. ADETIBA**

**ENGR (MRS) AYODELE IFIJEH**

**AUGUST 2021**

# **DECLARATION**

I hereby declare that I carried out the work reported in this report in the Department of Electrical and Information Engineering, Covenant University, under the supervision of Prof. E. Adetiba and Engr (Mrs) A.H Ifijeh. I also solemnly declare that to the best of my knowledge, no part of this report has been submitted here or elsewhere in a previous application for award of a degree. All sources of knowledge used have been duly acknowledged.

……………………………………

UKEH VICTOR UGOCHUKWU

16CJ020768

# **CERTIFICATION**

This is to certify that the project titled “Development of a Computer Based Testing Application using CodeIgniter Framework” by Ukeh Victor Ugochukwu, meets the regulations and requirements governing the award of the Bachelor of Engineering (Computer Engineering) degree of Covenant University and is approved for its contribution to knowledge and literary presentation.

**Supervisor 1:** Sign: …………………… …………………

Name: Prof. Emmanuel A. Adetiba Date:

**Supervisor 2:**  Sign: …………………… …………………

Name: Engr. Ifijeh Ayodele Date:

**Head of Department:** Sign: ………………………. …………………

Name: Prof. Emmanuel A. Adetiba Date:

**Internal Examiner:** Sign: …………………… ………………… Name: ……………………. Date:

# 

# **DEDICATION**

This project is dedicated to the Mighty God, for His guidance and grace throughout the duration of this project. I also dedicate this project to my Parents, family members and friends who have contributed towards the accomplishment of this project.

# **ACKNOWLEDGEMENTS**

My sincere gratitude goes to the Almighty God for helping me to accomplish this work. I would love to show gratitude to my parents, friends and colleagues for their assistance both mentally and financially. I would also love to show my appreciation and gratitude to my project supervisors, Prof. E. Adetiba and Engr. Ayodele Ifijeh, for all their assistance during the course of this project.

# **ABSTRACT**

Several studies have provided evidence that Computer Based Testing is far more reliable than the common Paper Based Testing. The paper based method of testing is known to be characterized by leakage of questions, bribery and even human errors which occur during the marking of scripts and recording of scores. Due to the advancement in technology, Computer-based Testing (CBT) was born to reduce and even eliminate the problems plaguing Paper-based testing and other traditional forms of testing. CBT provides a wide service for online testing which is free from irregularities while being fair to examinees, being cheaper, and providing instant feedbacks. Despite the numerous gains, the cost of procuring CBT systems from foreign software vendors is highly prohibitive. The design and implementation of a CBT system towards an affordable solution is presented in this report. This system was developed based on the model-view-controller design pattern leveraging the CodeIgniter framework, which is known for its great structure, great functionalities and ease of use. This system provides examiners (lecturers) with the ability to create examinations which would be taken by the examinees (students) and have examinee scores immediately recorded and accessible to the examiners. It also tracks the examination history of the examinations that have formerly been created and administered. The system provides many innovative features including real time data collection, easy management and distributed and interactive assessments towards encouraging the use of online assessments. This project provides a proper structure for CBT systems and also serves as a cheaper alternative to present day CBT applications.

**TABLE OF CONTENTS**

[**DECLARATION** ii](#_Toc82708194)

[**CERTIFICATION** iii](#_Toc82708195)

[**DEDICATION** iv](#_Toc82708196)

[**ACKNOWLEDGEMENTS** v](#_Toc82708197)

[**ABSTRACT** vi](#_Toc82708198)

[**LIST OF TABLES** xi](#_Toc82708199)

[**LIST OF FIGURES** xii](#_Toc82708200)

[**CHAPTER ONE** 1](#_Toc82708201)

[**a.** **BACKGROUND OF STUDY** 1](#_Toc82708202)

[**b.** **SIGNIFICANCE OF STUDY** 3](#_Toc82708203)

[**c.** **PROBLEM STATEMENT** 4](#_Toc82708204)

[**d.** **AIM AND OBJECTIVES** 4](#_Toc82708205)

[**i. AIM** 4](#_Toc82708206)

[**ii. OBJECTIVES** 5](#_Toc82708207)

[**e.** **METHODOLOGY** 5](#_Toc82708208)

[**f.** **SCOPE OF STUDY** 5](#_Toc82708209)

[**g.** **LIMITATION OF STUDY** 6](#_Toc82708210)

[**h.** **PROJECT ORGANIZATION** 6](#_Toc82708211)

[**CHAPTER TWO** 8](#_Toc82708212)

[**2.1. INTRODUCTION** 8](#_Toc82708213)

[**2.2. COMPUTER BASED TESTING SYSTEM** 8](#_Toc82708214)

[**2.3. MODEL-VIEW-CONTROLLER** 11](#_Toc82708215)

[**2.4. SOFTWARE DEVELOPMENT LIFE CYCLE MODELS** 12](#_Toc82708216)

[**2.4.1 Waterfall model** 13](#_Toc82708217)

[**2.4.2 Prototyping Model** 14](#_Toc82708218)

[**2.4.3 Iterative and Incremental Model** 15](#_Toc82708219)

[**2.5. CODEIGNITER FRAMEWORK** 17](#_Toc82708220)

[**2.5.1 Application Flow in CodeIgniter** 19](#_Toc82708221)

[**2.6. COMPONENT THEORIES** 20](#_Toc82708222)

[**2.6.1 System Backend** 21](#_Toc82708223)

[**2.6.2. System Frontend** 24](#_Toc82708224)

[**2.7. REVIEW OF RELATED WORKS** 26](#_Toc82708225)

[**2.7.1. Review of Studies Undertaken on CBT** 26](#_Toc82708226)

[**2.7.2. Review of the Applications of CBT** 28](#_Toc82708227)

[**2.7.3. Review of the Technology Used in the Development of Web Applications in The Industry.** 30](#_Toc82708228)

[**2.8. SUMMARY** 32](#_Toc82708229)

[**CHAPTER THREE** 34](#_Toc82708230)

[**3.1. INTRODUCTION** 34](#_Toc82708231)

[**3.2. SYSTEM ANALYSIS** 34](#_Toc82708232)

[**3.2.1. The CBT Application Architecture** 35](#_Toc82708233)

[**3.3. SOFTWARE DEVELOPMENT LIFECYCLE** 37](#_Toc82708234)

[**3.4. SYSTEM REQUIREMENT ANALYSIS** 39](#_Toc82708235)

[**3.4.1. System Hardware Requirements** 39](#_Toc82708236)

[**3.4.2. System Software Requirements** 40](#_Toc82708237)

[**3.4.3. Functional Requirements** 40](#_Toc82708238)

[**3.4.4. Non-Functional Requirements** 44](#_Toc82708239)

[**3.5. SOFTWARE DESIGN** 45](#_Toc82708240)

[**3.5.1. Class Diagram** 47](#_Toc82708241)

[**3.5.2. Activity Diagram** 49](#_Toc82708242)

[**3.5.3. The Sequence Diagram** 52](#_Toc82708243)

[**3.6. DATABASE DESIGN** 54](#_Toc82708244)

[**3.7. SOFTWARE IMPLEMENTATION TOOLS** 59](#_Toc82708245)

[**3.7.1. Software Frontend** 59](#_Toc82708246)

[**3.7.2. Software Backend** 60](#_Toc82708247)

[**3.8. SUMMARY** 61](#_Toc82708248)

[**CHAPTER 4** 62](#_Toc82708249)

[**4.1. INTRODUCTION** 62](#_Toc82708250)

[**4.2. WEB APPLICATION IMPLEMENTATION** 62](#_Toc82708251)

[**4.2.1. CodeIgniter Framework** 63](#_Toc82708252)

[**4.2.2. Student Section of the CBT system Web Application** 64](#_Toc82708253)

[**4.2.3 Lecturer Section of the CBT system Web Application** 67](#_Toc82708254)

[**4.2.4. Administrator Section of the CBT system Web Application** 70](#_Toc82708255)

[**4.3. SOFTWARE TESTING** 71](#_Toc82708256)

[**4.3.1. Unit Testing** 72](#_Toc82708257)

[**4.3.2. Deployment Testing** 74](#_Toc82708258)

[**4.4. SUMMARY** 76](#_Toc82708259)

[**CHAPTER 5** 77](#_Toc82708260)

[**5.1. INTRODUCTION** 77](#_Toc82708261)

[**5.2. KNOWLEDGE GAINED** 77](#_Toc82708262)

[**5.4. ACHEIVEMENTS** 78](#_Toc82708263)

[**5.4.1. Challenges Encountered** 78](#_Toc82708264)

[**5.4.2. Limitations** 79](#_Toc82708265)

[**5.5. CONCLUSION** 79](#_Toc82708266)

[**5.6. RECOMMENDATIONS** 79](#_Toc82708267)

[**REFERENCES** 80](#_Toc82708268)

[**APPENDIX** 87](#_Toc82708269)

# **LIST OF TABLES**

[Table 2.1: Advantages and Disadvantages of Waterfall Model 13](#_Toc82515593)

[Table 2.2 Advantages and Disadvantages of Prototyping Model 15](#_Toc82515594)

[Table 2.3: Advantages and Disadvantages of Iterative and Incremental Model 16](#_Toc82515595)

Table 3.1: Non-Functional Requirements………………………………………………45

Table 3.2 Users Table…………………………………………………………………..55

Table 3.3 Examinations Table………………………………………………………….57

Table 3.4 Questions Table……………………………………………………………...58

Table 3.4 Scores Table….……………………………………………………………...59

# **LIST OF FIGURES**

[Figure 2.1: Multi-stage Form Diagram [11] 10](#_Toc82708270)

[Figure 2.2: MVC Reference Architecture 11](#_Toc82708271)

[Figure 2.3: Waterfall Model [13] 14](#_Toc82708272)

[Figure 2.4: Iterative and Incremental Model 17](#_Toc82708273)

[Figure 2.5: CodeIgniter Framework Contents 18](#_Toc82708274)

[Figure 2.6: CodeIgniter Application Flow Chart [14] 19](#_Toc82708275)

[Figure 2.7: Structure of MySQL table 22](#_Toc82708276)

[Figure 2.8: Block Diagram of MySQL Query Processing [25] 23](#_Toc82708277)

Figure 3.1: System Block Diagram [45] 35

Figure 3.2: Client-Server Architecture 36

Figure 3.3: Waterfall Model 37

Figure 3.4: Laptop used for development 40

Figure 3.5: Use Case Diagram for Student and Faculty Members 42

Figure 3.6: Use Case Diagram for an Administrator 43

Figure 3.7: The CBT system Class Diagram 48

Figure 3.8: CBT System Functional Diagram 49

Figure 3.8: CBT System Full-scale Activity Diagram 51

[Figure 3.10: CBT Sequence Diagram 53](#_Toc82708278)

Figure 4.1: The Orion Web Application Login Page 65

Figure 4.2: VSCode view of the CodeIgniter Framework 66

Figure 4.3: The Orion Student Dashboard 67

Figure 4.4: The Orion Student Assessment page 68

Figure 4.5: The Orion Examination Page 68

Figure 4.6: Lecturer Dashboard 69

Figure 4.7: View Created Exams Page 70

Figure 4.8: Create Exam Page 70

Figure 4.9: Add Question Page 71

[Figure 4.10: User Registration Page 71](#_Toc82708279)

[Figure 4.11: PHP Sandbox Interface 72](#_Toc82708280)

[Figure 4.12: Error Result 73](#_Toc82708281)

[Figure 4.13: Correct Result 74](#_Toc82708282)

[Figure 4.14 Accessing the CBT application on Google Chrome. 75](#_Toc82708283)

[Figure 4.15 Accessing the CBT application on Firefox. 75](#_Toc82708284)

[Figure 4.16 Accessing the CBT application on Microsoft Edge. 76](#_Toc82708285)

# **CHAPTER ONE**

**INTRODUCTION**

## **BACKGROUND OF STUDY**

Computer based testing commonly abbreviated as CBT is the utilization of ICT for any type of assessment. It allows educators to administer various types of assessments such as tests, exams, assignments, quizzes and even surveys, within a time limit. The responses are then automatically marked and recorded electronically. The intention is to ensure the assessment process is fair, quicker, secure and more reliable.

In the world today, technology has come far from what it was in the 20th century. Earlier methods of writing objective exams and tests such as hand written essays and Optical Mark Recognition (OMR), have proven to have a very high level of susceptibility to examination malpractice, bribery and cheating over the years[1]. Because of the rise in technology advancement in recent years, the world has come up with CBT, which has proven to be more reliable. The application and utilization of CBT is growing increasingly. The rise in popularity has been aided by the development of improved interfaces, the drastic increase in the amount of hardware and the accessibility to the hardware [2][3].

Due to online testing usually moderated by ICT becoming more prevalent, there is need for schools and other organizations to consider using these tools. CBT is a proven remedy for both test and examination assessments rather than previously tried methods. Methods such as OMR, essay examinations and other paper based testing methods have proven to have many limitations such as inability to log off students after time has expired or examination has been concluded, lack of flexibility, lack of scalability, lack of robustness, delay in delivering results of assessments, high risk of human error, delay in release of students results, subjective scoring and cases of missing results amongst others [4].

CBT has a lot of benefits, ranging from increase in test creation efficiency, to quicker delivery of assessments and higher engagement of students. CBT can be very helpful for high stake exams and continuous assessment systems where instant response to students is required [5]. CBT has been adopted by some organizations who either sell this services or use them for their own businesses. Some of this organizations include Moodle, Canvass, Google classroom and so many others.

In 2012, CBT was tested in the FUOYE (Federal University Oye Ekiti State), Nigeria [4]. The assessments of performances were carried out by about two hundred and fifty volunteers including programmers, students, engineers and lecturers. The average score rating was very good and proved the system had a high level of robustness, scalability, flexibility and quality Graphical User Interface (GUI). The test proved the CBT very reliable for mass testing. But unfortunately, Computer-based testing has not become the new normal in Nigeria. Schools and Universities have just started getting accustomed to this technology and as a result of that, there is no nationwide accepted structure or framework for CBT.

Hence, Universities and other organizations that need this service have to pay huge amounts to international companies to make use of their technology. However, since there is no widely accepted structure and framework for computer based testing in Nigeria, the earlier stated problems still persist. In Nigeria, most of the Universities and many other organizations conducting electronic examination and tests rely on the software vendor for the administration of their exams and tests. This has led to such organizations spending a humongous amount of money just on computer based testing. This major problem can be rectified by creating a web based application that can serve as a cheaper alternative to present CBT applications while also serving as a future reference to the development of better systems.

In this project, the codeIgniter framework was used for the development of the application. CodeIgniter is a PHP framework, which comes as a toolset with required tools and libraries required to build a web application. CodeIgniter is a popular framework, which is known for its definite structure, functionalities and ease of use. The framework is therefore a great choice for building of this application. Therefore, the goal is to show the efficiency of computer based testing by learners in Secondary Schools, Universities and other organizations [6].

## **SIGNIFICANCE OF STUDY**

The project supplies a more effective method for the conduction of assessments as opposed to the more common paper based assessments. The CodeIgniter framework ensures that the application is secure, fast and super responsive therefore, improving the overall experience of the students. The project will improve the general administration of tests and examinations in schools while reducing vices such as examination malpractice, lack of adherence to time and bribery. Furthermore, the project prevents cases of academic dishonesty and also helps programmers to develop future projects with better structure and CBT functionality.

## **PROBLEM STATEMENT**

As a result of technological advancement over the last two decades, Computer Based Testing is gradually replacing the former methods of taking assessments. A key reason for this is as a result of CBT providing more security and reliability. Unfortunately, not all Nigerian schools have adopted this technology.

Considering this, the major problems include:

1. Lack of a widely accepted CBT framework and structure.
2. High rate of examination malpractice, cheating and other vices
3. Unavailability of locally based applications with this technology
4. Huge cost is spent acquiring this services.

Provision for future technology innovation can be made in these areas, if the necessary work and effort is put in to enable proper handling of these problems, hence the need for developing a computer based testing application.

## **AIM AND OBJECTIVES**

## **i. AIM**

Theaim of this project, is to develop a web application for Computer Based Testing using Code Igniter Framework.

## **ii. OBJECTIVES**

The objectives to realize this aim are to:

1. Elicit the requirements for the application.
2. Design of the System’s interface, classes and databases.
3. Implementation of the application’s front-end and back-end.
4. Testing of the System to eliminate errors.
5. Online deployment of the Web application.

## **METHODOLOGY**

The following methodology will be adopted in developing the CBT application

1. Elicitation and analysis of requirements based on existing system functionality.
2. Design and development of software requirements using UML, PHP, HTML, CSS and JavaScript.
3. Database creation and implementation using MariaDB and MySQL.
4. Hosting on the internet using Cpanel.

## **SCOPE OF STUDY**

The large sums of money spent by Universities and other tertiary institutions can be greatly reduced by having the technology that can provide computer based testing services for a smaller amount as compared to the pricing offered by international organizations. The CBT application employed in this project will go a long way in reducing the financial pressure faced by both parents and guardians. The software helps not only schools and parents but also provides aid to programmers and researchers.

## **LIMITATION OF STUDY**

The web application does not make use of component based rendering therefore for every page refresh, all components of the application are going to be refreshed making the system a little slower than it could possibly be. Also, CodeIgniter has limitations such as not supporting modular separation of code by default and requiring libraries with limited capacity [7].

## **PROJECT ORGANIZATION**

This chapter gives an Introduction to the project. It contains the background information of the project being carried out, the motivation, the problem statement, the aim and objectives of the project, methodology being used and finally it concludes with the significance and limitations of the project.

Chapter Two is the Literature review which gives more depth into the processes that make up the intended system while summarizing already existing projects and related systems done by engineers. It also contains the review of existing literatures related to the project at hand.

Chapter Three is the Methodology which involves the system analysis and design. It presents the Computer Based Testing Web application and all its system designs which consists of the physical and logical design of the different modules that make up the system then presents the system architecture and conceptual design. The design of the database is also included in this chapter.

Chapter Four is the Implementation which contains the essential outline of the system implementation which consists of the tools and the frameworks used, and also the operation of the sub-modules of the system.

Finally, the Chapter Five closes the project and makes suggestions and recommendations for further future works and also for projects which could be developed in the future in the area of computer based testing systems.

# **CHAPTER TWO**

**LITERATURE REVIEW**

## **2.1. INTRODUCTION**

CBT involves the use of computers to enable lecturers administer tests. CBT has grown over the years since it was conceived in the 1970s. It has become more secure, more resistant to theft and hacks and has become acceptable world-wide. It contains automatic marking and printing of results in record time.

A lot of Engineers, software developers and researchers around the globe, have contributed to the ethics, architecture, structure and solutions in the Information and Communication Technology (ICT) field with respect to CBT. This chapter gives a background literature of CBT and also review several works and publication in this domain.

As CBT is a web based activity, it can be developed using a lot of web technologies including frameworks like CodeIgniter, CakePHP, React, Express, Vue.js and so many more. As older technologies get updated and better technologies come about, the applications and efficiency of CBT systems keeps on growing while getting better over time.

## **2.2. COMPUTER BASED TESTING SYSTEM**

CBT application also known as electronic assessment refers to carrying out assessments such as tests and examinations on organized systems on computers. It provides quick automotive marking, time duration and quick solution questions to students. It is a broad industry made up of various types of assessments, purpose, design which are appropriate for certification testing, achievement testing and a lot more other types of testing [8].

There are four main models for the administration of computer based testing according to [9] they are:

1. **Linear or fixed form**: This involves examining students with the same set of questions and items either in the same format or a randomly shuffled format. The scoring is done conventionally, either by summing up the total number of correct answers selected or by using the popular item response theory (IRT) methods.
2. **Random form**: This model involves drawing of questions and other related items from a pool to ensure statistical rules are met. The pool contains more items than required. This is to ensure that students get different question which are equal in reliability and difficulty.
3. **Multi-stage form**: This method involves a “routing test”. Students take a routing test which contains about ten questions. After the test is done, a decision is taken based on the students score in the test. The decision lets students with scores above the cut-off mark take a harder second test while those who fell short take an easier second test. Upon ending the second test the students’ total scores from totaling both their tests is provided.

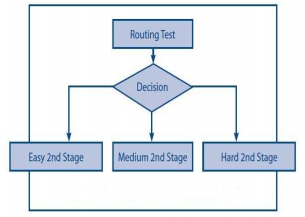


Figure 2.1: Multi-stage Form Diagram [11]

1. **Item-adaptive form:** This is a more intense form of the Multi-stage form. This model provides a question from a pool of questions based on the student’s score after each question. This is done to complete four objectives which are:
   * 1. To maximize testing efficiency.
     2. To provide a balanced test based on content.
     3. To protects some questions from overexposure-e while encouraging other question with minimal exposure.

## **2.3. MODEL-VIEW-CONTROLLER**

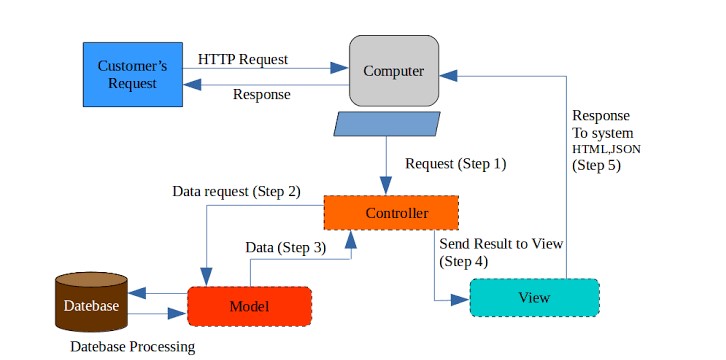


Figure 2.2: MVC Reference Architecture

The architecture for the CBT application uses a Model-View-Controller (MVC) architecture as shown in figure 2.2. The architecture is divided into three parts which work together to achieve the desired results. They each have different characteristics to help carry out their different duties to perform in application development in order to obtain the end goal.

1. **Model:** The model consists of classes that contains essential information that support a particular problem which an application accesses and manipulates while encapsulating storage and various types of validation issues for an application. This is tone of the cores of an application [10]**.**
2. **View:** The view is a set of classes that contains templates and data forms which provides responses through a browser. The controller provides the view with required data to display on a browser based on the type of request. There are three types of view namely:
3. The Graphical User Interface (GUI/widget) view
4. The Command Line Interface (CLI) view
5. The Application Program Interface (API) view
6. **Controller:** Controllers are responsible for managing the execution of applications. A controller returns responses of a request that has been made. A controller can return various actions depending on the type of request. It coordinates the flow between the model and the view while also controlling the logic of the application [11].

The user sends a request to the system which can be either be a PUT, POST, GET or DELETE request through the views they can relate with on their browser. The browser then sends the request to the controller and fetches the specific API. Based on the type of request, the controller sends a data request to the database according to the request of the customer/client. The database then processes the request by checking the classes in the model. After processing, the views the sends feedback either through JSON or html responses to the computer browser the client is using.

## **2.4. SOFTWARE DEVELOPMENT LIFE CYCLE MODELS**

Software development models are the different set of processes that are chosen for the development of a software system based on the aim and goals of a project [12].The selection of models is highly important as it determines what testing techniques to use. There are various software techniques and some of them are:

1. Waterfall model
2. V model
3. RAD model
4. Agile model
5. Iterative and Incremental model
6. Spiral model
7. Prototype model

Some of the software techniques are built on other software techniques. Therefore, the main types of software models which will be explained include the Waterfall model, Prototyping model and Iterative and Incremental model.

### **2.4.1 Waterfall model**

Waterfall is the oldest software development model. It is a direct sequential flow whereby progress flows downwards as shown in figure 2.3. This means the next process can only begin when the previous process in completed. It provides a lot of restrictions to changes during development. It is used when a project does not focus on changing the system requirements and for large complicated systems.

Table 2.1: Advantages and Disadvantages of Waterfall Model

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| It is easy to understand | The requirements of the software system are frozen |
| Structural approach. | Returning to a prior stage after delivery is difficult |
| Stages and activities are well defined. | Adjusting scope of the project is difficult |
| Planning and scheduling are made easy | Expensive and time-consuming |
| Verification at each stage ensures timely detection of problems and errors. |  |
| Deliverables are specified for each stage |  |

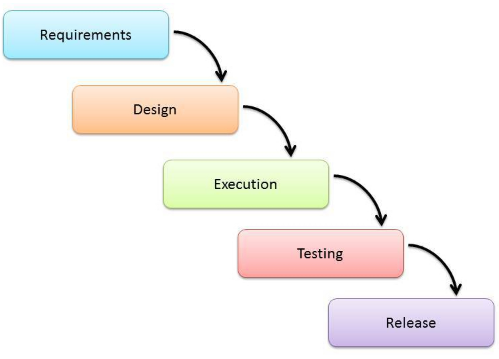


Figure 2.3: Waterfall Model [13]

### **2.4.2 Prototyping Model**

This is the creating of prototypes of software applications. It is done by developing some of the software content in order to reduce misunderstanding between the customer’s system requirements and the creators. This type of model is normally used when a system contains user-interactions. It can also be used in conjunction with any other software development model. There are three types of prototyping models. There are:

1. Throw-away prototyping
2. Evolutionary prototyping
3. Incremental prototyping
4. Extreme prototyping

Table 2.2 Advantages and Disadvantages of Prototyping Model

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Reduces expenses and maximizes time. | It can lead to user confusion due to insufficient analysis |
| Allows increased user involvement | Misunderstanding between creator and customer |
|  | A lot of time is spent developing the prototype. |
|  | It is expensive to implement the prototype. |

### **2.4.3 Iterative and Incremental Model**

This model was developed to eliminate the weaknesses of the waterfall model. The model involves developing the system through repeated cycles and in minor portions at a time. It normally begins with the initial plan and ends with the deployment of the application. It can be used for developing three types of systems which include shrink-wrap applications, large applications built in segments and for systems with separate components.

Table 2.3: Advantages and Disadvantages of Iterative and Incremental Model

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Business value is developed early | It involves a lot of documenting work |
| Scarce resources are managed better due to good increment definition | Processes are defined and are followed accordingly |
| Some change requests can be allowed in between increments | Increments are defined according to their functions and feature dependencies. |
| Focus is on customer value. | A lot of customer involvement |
| Issues and related problems can be detected earlier | Separating the different functions and features of the software application may be problematic. |
|  | Integration between the iterations can be an issue if it is not considered during the development and project planning. |

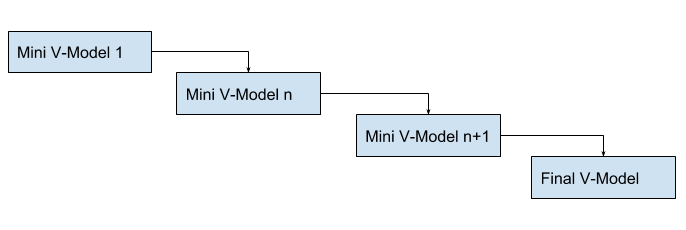


Figure 2.4: Iterative and Incremental Model

## **2.5. CODEIGNITER FRAMEWORK**

CodeIgniter is a PHP based framework. Itis a free non-proprietary software which has a small footprint aimed at programmers who require a rich and simple set of tools to develop full featured applications [14]. It is a Model-View-controller framework which comes with many core libraries that enable applications to be created. The toolset is divided into 4 parts which are:

1. Application
2. System
3. User guide
4. Others

The application folder is the part of the CodeIgniter framework where development is done. It consists of the cache, config, controllers, core, helpers, hooks, language, libraries, logs, models, third-party and views as shown in fig 2.4. This is the area where the engineer mostly interacts with to build the application of choice [15]. The system folder consists of files that help make programming easier such as the core folder which contains CodeIgniter’s core class, the database, the fonts, the helpers, the language and the libraries folders. The User guide contains an offline CodeIgniter guide which explains the functions, libraries and so on, in case a need arises. Finally, others or other files consists of the engineer’s license and other dependencies added to the framework such as composer and Git.

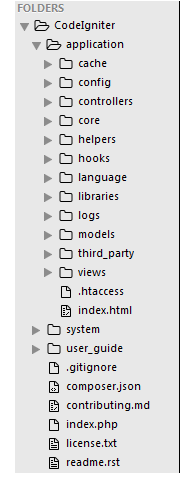


Figure 2.5: CodeIgniter Framework Contents

### **2.5.1 Application Flow in CodeIgniter**

The application flow in the CodeIgniter framework in the system starts from the controller which is can be named according to how the user sees fit. By default the controller is named index.php. The flow of data in the framework is shown in the diagram below [16].

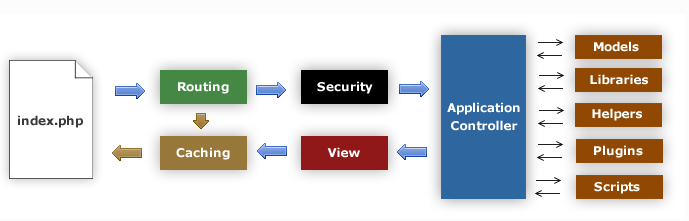


Figure 2.6: CodeIgniter Application Flow Chart [14]

1. The controller which is the index (index.php) file in this case initiates the resources from the base controller needed to run the application.
2. The router then determines what would be done with the HTTP request after inspection.
3. In the presence of a cache document, the request is sent directly to the browser.
4. Before the controller is initialized, The HTTP request and any other data to be submitted is checked to ensure the pass the requirements for security.
5. The controller then loads all resources to process the particular request and sends the result to the view. The resources include models, helpers, libraries, plugins and scripts.
6. **Models:** These are PHP classes that have been designed to work and interact with information in your database.
7. **Helpers:** Helper files are a collection of functions in a specific category that aid the CodeIgniter framework to perform different tasks. There are various types of helpers such as Text Helpers, URL Helpers, Cookie Helpers and File Helpers etc. For example, Text Helpers aid in formatting different types of text routines, URL helpers help in developing links and so on. [14]
8. **Libraries:** These are collections of non-volatile resources that indirectly increases the speed of developing and designing an application. The CodeIgniter library folder is located at system/libraries as shown in fig 2.3.
9. **Plug-ins:** Plug-ins are (sometimes essential) piece of software code that enables an application or program to carry out certain jobs it could not do otherwise.
10. **Scripts:** Scripts are sets of instructions that are interpreted or carried out by another program or class instead of the system processor.
11. The final view is then rendered to the web browser to be seen by the end user. If cache has not been disabled, the view is cached first so that other incoming requests can be served.

## **2.6. COMPONENT THEORIES**

In recent times, software systems have become more complex and extensive due to a plethora of factors such as increasing client’s demands and advancements in technology [17]. This has led to the change from the use of older components and technologies, to the use of more recent components and technologies which are better and more efficient. The components of this system are separated into two. Some components are used for the backend while some are used for the frontend. This components used for both the design and implementation of this system are explained and summarized here.

### **2.6.1 System Backend**

#### **2.6.1.1 PHP**

PHP is widely known as the most popular server-side programming language due to its record share of 79.10% [18] as of 2021 being used for developing most of the recognized websites including Facebook and Wikipedia and other management systems. Due to its weakly and dynamically typed syntax and a large number of built-in features, the language is easy to learn for beginners. It is therefore the quickest developing script language for web based applications [19]. It is a dynamic language with a lot of complicated semantics [20].

#### **2.6.1.2 MySQL**

MySQL isa free, open source, relational, structured database management system which was created in Sweden in the year 1995 and is currently owned by Oracle [21]. SQL is an abbreviation for Structured Query Language. MySQL works with tables where by a created database consists of tables based on the data to be collected by the engineer. The tables as shown in fig 2.5 below, consists of both rows and columns [22]. The first row contains the type of data to be collected by that field. It allows linking of data in separate tables together. Each table consists of different fields, which represents bits of information [23].

[24] The advantages of MySQL include:

1. Security of data.
2. Ability to scale database on demand
3. High performance
4. 24/7 Uptime
5. Total control of workflow
6. Open source flexibility
7. Portability

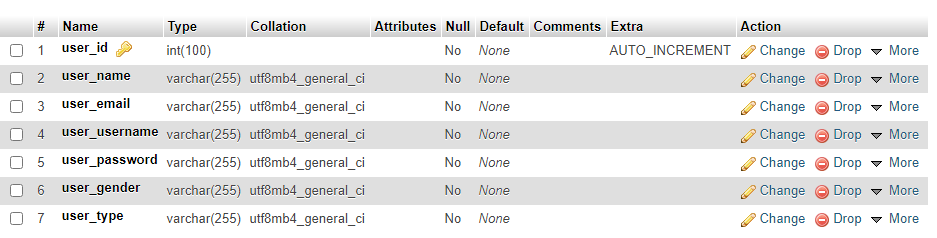


Figure 2.7: Structure of MySQL table

The MySQL query processing works by first connecting to MySQL after running the PHP script. The PHP script provides instructions in which MySQL interprets and goes on to specify the name and columns identified in the script from the desired table of the specific database. It then performs the required function either posting, getting, updating, deleting and so on. Then finally, it converts the result back into required data type accepted by HTML and closes the connection,

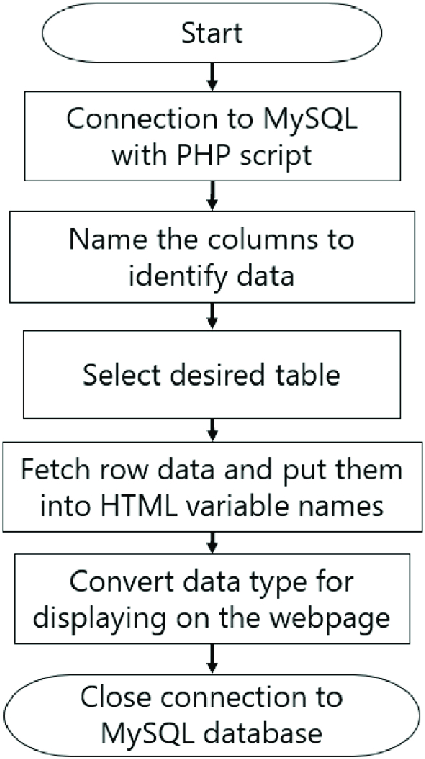


Figure 2.8: Block Diagram of MySQL Query Processing [25]

#### **2.6.1.3 XAMPP**

XAMPP is the abbreviation for Cross-Platform, Apache, MySQL, PHP and Perl. It is a basic, cross-platform and lightweight Apache distribution that makes it easy for programmers to create a local web server for testing purposes. It also makes transition from a local server to a live server relatively easy [23]. XAMPP has four main components which are:

1. **Apache**: Apache is the main web server application that performs the processing and delivery of data on a web to the web browser. It runs an about sixty-seven percent (67%) of all web servers worldwide.
2. **MySQL**: XAMPP uses MySQL as its relational database to query data into and from it no matter the size of the dataset.
3. **PHP**: Hypertext Preprocessor commonly abbreviated as PHP. It is a free, open source software that allows the execution of scripts on a web server. It generates dynamic web-pages known as PHP scripts to reside between reserved PHP tags. This allows developers to embed PHP scripts in HTML pages.
4. **PERL**: Perl is a stable multi-platform programming language formally used for manipulating texts but presently used for system administration, web development and network programing etc.

### **2.6.2. System Frontend**

#### **2.6.2.1. Hyper Text Mark-up Language**

Hyper Text Markup Language (HTML), is a language used for standard marking up of documents to be displayed on a website browser. It serves as a formatting system for presenting information gotten from the Internet. Each unit used for the retrieval is known as a Web page (from www known as World Wide Web). Web pages usually contain hypertext links that allow related pages to be fetched. Thereby, providing html to describe the structure of Web pages. HTML allows software engineers to publish documents online with headings, paragraphs, text, tables, lists, photos and videos while providing the ability to access online data and information via hypertext links. HTML elements are usually put in tags.

#### **2.6.2.2 Cascading Style Sheet**

Cascading Style Sheet (CSS), is a style sheet language used for describing the presentation of a document written in HTML or any other markup language. It allocates the style of an HTML manuscript and determines how HTML elements will be presented [26]. CSS contains both rules and properties by which a software engineer must follow in order to achieve the desired result. The can either be put directly into a HTML document or through another file with a .css extension.

#### **2.6.2.3 Bootstrap**

Bootstrap is a free open-source front-end framework used for the development of both web applications and websites. Bootstrap is agile and faster than many other frontend frameworks. It contains a lot of HTML and CSS templates which are used as User Interface (UI) interface elements such as headers, buttons and forms. It allows developers to customize their application or website to the result the desire.

[27] The advantages of Bootstrap include:

1. Bootstrap is Easy to use
2. Easy Integration unlike many other frontend frameworks
3. It supports JavaScript components
4. Bootstrap is totally customizable
5. A lot of frequent updates and so many more others.

#### **2.6.2.3 JavaScript**

JavaScript also known as JS, is known to be the world’s most popular programming language It is high-level language, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. It provides functionality in the frontend aspect of an application. It can also be used to load embedded resources. By the year 2012, about fifty-five (54.5%) of Web pages make use of JavaScript to load embedded resources [28].

## **2.7. REVIEW OF RELATED WORKS**

### **2.7.1. Review of Studies Undertaken on CBT**

Jimoh et al [29] took on a research to determine the learners’ perception of Computer Based Testing for Chemistry Courses offered by undergraduates. They were able to deduce that if quality of CBT design is improved to match difficulty of an assessment, the examination experience will be more favorable to students. The sample space which was studied was too small to judge general perception of chemistry students worldwide. Samson et al[30] studied the problems with CBT regarding both security and result uprightness. They deduced that a new method of developing such a system would give better results. The chosen method was to develop the system using Unified Modelling Language (UML) which was executed using common technologies such as PHP, HTML and MySQL. The results of the study were positive, showing improved security and result uprightness.

Ejim et al[9] conducted a study to provide an overview on CBT. The book provided both the history of Computer Based Testing (CBT) and the advantages of CBT over the common PPT assessment. However the book failed to provide a more in-depth explanation of the workability and interconnectivity in the design of CBT examinations. Stoyanov et al[31] conducted a study to compare and distinguish between Computer Based Testing and eLearning. The concluded that CBT is utilized as a beginning stage for the advancement of a means to support learning while eLearning is used as a point of reference for future R&D in education and learning while maximizing the use of media and technology.

Abubakar and Adebayo [1] conducted a study to highlight the “prospects and challenges” that would be a source of concern by and threat to the success of adopting Computer Based Testing in all kinds of examinations whilst highlighting how to eliminate, reduce or arrest them. They concluded that ten challenges affects the adoption and utilization of CBT in Nigeria which include the financial factor, security, Software, Poor ICT culture, policy and implementation and finally, power failure. They finally concluded that the approaches to eliminate these challenges are to recognize educational presentation on Computer Based Testing, public connection crusade, web campaign, post-test criticism, adequate power supply, execution of ICT strategy etc.

According to [32] CBT has made an outstanding progress over the previous two decades. They conducted a research to explore the validity problems in CBT. They found out that due to a high number of issues, CBT had yet not been able to live up to expectations. The problems included discovering better approaches to more readily guarantee test security, creating strategies for utilizing CAT procedures to evaluate multidimensional language constructs and finally making scoring frameworks fit for estimating importance and sensation of composed and spoken talk. They concluded by stating that such problems will most surely continue but the benefits of CBT will definitely be worth the hassle.

An interactive CBT system was also created and implemented at the University of Luton by Zakrzewski and Bull [33]. Their resulted research showed that the benefits of CBT are counterbalanced against the time taken to plan and produce objective tests. The responses to the CBT from both academic staff and students were very favorable thereby making this form of assessment to be profoundly attractive both educationally and monetarily. They finally go on to explain that both the investment of resources in in staff advancement and focal help are a determining factor in the resulting viability and productivity of the system. Also, Moe [34] took on a research in 2009 to know the reaction of Norwegian schools to the introduction of CBT on a large scale among the children. The result showed that students and staff preferred this kind of testing. A limitation on the research was that the tests were conducted only on English Language rather than diverse subjects. Therefore, the general reaction to using CBT in various subjects are unknown.

### **2.7.2. Review of the Applications of CBT**

Clariana and Wallace et al [35] undertook a study to affirm a few key factors in CBT versus Paper Presentation Testing evaluation. Elements of the examination were content commonality, PC familiarity, competitiveness and sexual orientation. The investigation utilized a post-test which worked with one factor test mode. The dependent variables for the study included a 100-question multiple choice type examination and a personal report distance learning study which was undertaken by the undergraduates. Results showed that CBT conveyance positively affected the students’ scores when contrasted with the results from PPT.

Also, Ayo et al [36] conducted a research on the prospects of Online Examination implementation in Nigeria. They developed a model as a method of controlling the problems found amongst various examinations conducted in Nigeria. The model included a one hour examination which consisted of 120 questions gotten from various subjects including English, Mathematics, General Science, Commercial Courses and Religious Knowledge. Results showed that the model had the ability to curb the problems associated with PPT testing such as impersonation. They concluded that the journey for e-Examination could accelerate the improvement in e-Learning in Nigeria as access to education becomes easier and improved.

Ejim et al [37] carried out a project in 2013 to create an online examination system with message notification and multifactor authentication features. He designed his CBT system making use of the OOAD (Object Oriented Analyzing and Designing) methodology whereby he utilized technologies such as PHP, html, Cascading Style Sheet and MySQL. He used the Apache Server for running his application. The result of the projects was effective and reusable for modern applications. Therefore, it could be expanded and a lot of other features could be added to it. Langer and Bodendorf et al [38] took on a project to develop a flexible system architecture for Multimedia Computer Based Testing software and applications. It gives authors the ability to reconfigure modules for various educational settings. The architecture was a success, it made testing easier and secure.

### **2.7.3. Review of the Technology Used in the Development of Web Applications in The Industry.**

A project to design and implement a web based application for Intermediate Online Shop with Laravel Framework, was carried out by Mahmood et al [39]. They made use of MVC pattern frameworks instead of other prior methods to develop a web application due to the fact they had enormous impediments, very time consuming and a lot of startling mistakes. The result was that the improvement was normalized and non-business rationale connections automatically prepared therefore, making the application more scalable and efficient than it would have been using older technologies.

Also, Hidayat and Utomo et al [40] created an open source mobile phone learning application which made use of the famous CodeIgniter framework. They made use of CodeIgniter because it supports MVC coding architecture which helps greatly in reducing app development duration. For their project, they also made use of CodeIgniter because it comes with inbuilt libraries that gives essential usefulness to tackle regular problems. They concluded by explaining that codeIgniter enables application developers to focus on coding application or plugin functionality rather than wasting time building the whole application from scratch.

Furthermore, Wibawa et al [41] conducted a study to evaluate the level of user acceptance of an online CBT application they built by using the powerful PHP framework known as day CodeIgniter. The concluded that based on the online application testing and User Acceptance Testing (UAT), the type of application was very acceptable for a school environment and was also remarkably received by the body of students. Therefore, online testing was given the go ahead due to CBT already having fruitful procurement of a category.

Kuyoro et al [3] designed and executed a Computer Based Testing System to decrease the postponement in the notice of students’ final assessment score as well as other areas of appraisals such as Tests and Assignments. They made use of technologies such as WAMP for local hosting, MySQL database, PHP: Hypertext Preprocessor, JavaScript, Cascading Style Sheet (CSS) and Hypertext Markup Language (HTML). From their study the figured that the utilization of CBT can increase the proficiency and productivity of schools and other institution as it lessens the expenses of stationeries and work associated with conducting assessments physically.

In Gunadarma University in Indonesia, a research was conducted by Hustinawati et al [42] who were in the Information System Department, to compare the performance between the CodeIgniter and CakePHP frameworks when used for website design and development. From their result they were able to deduce that although CakePHP has a larger number of features and utilities, CodeIgniter was more preferable in website development because it was lighter and simpler to learn, change, and coordinate Libraries and Helpers. More so, Sujatmiko et al [43] took on a study to determine the way of creating CBT equipment and the level of validity of the equipment. The developed Moodle type web system using the R&D process with CodeIgniter. The result proved that with a score of 90.65 percent, that the CBT based Moodle was valid. The limitation is that the program is only suited to the Moodle platform.

Abass et al [22] conducted a project to develop a web-based examination system using open source programming model. The project was conducted to create a system devoid of irregularities as were found among the PPT assessment. The system was developed using CSS, html, PHP, Dreamweaver and MySQL technologies. He achieved efficient release of results without error but was unable to achieve collation of results from the different courses taken by a student. Furthermore, Kaparang et al [44] carried out a research to produce object data modelling for e-assessment systems based on indicating the level of achievement competency. The system was implemented using both the MVC model and Object Oriented Programming (OOP) concepts. It was created with the PHP language whilst using CodeIgniter framework. The result enabled automatic result viewing while making it easier for teachers to set questions. A limitation was the unavailability of security techniques for protection from exploitation.

## **2.8. SUMMARY**

This chapter showed a brief overview of the history of online testing, applications of CBT in various aspects and technologies used in building CBT systems to provide a thorough understanding of the workings of CBT as a whole. From the various reviews above, it has been shown that online testing is continuously improving as the technology continues to improve. Productivity and efficiency has risen in the area of assessing students’ knowledge through tests, examinations and assignments. It is fast eliminating vices in schools thereby making quality education possible. The present study aims at improving the quality of computer based testing solutions.

# **CHAPTER THREE**

**SYSTEM ANALYSIS AND DESIGN**

## **3.1. INTRODUCTION**

This chapter discusses the process involved in the development of a CBT application using the CodeIgniter framework. It shows the step-by-step process involved in the development of the application including the integrations of the various components required in establishing the system. The necessary diagrams which include the various UML diagrams and other schematics are also provided and explained.

Some of the tasks that were carried out in this project includes setting up the MySQL database, building the backend with the CodeIgniter Framework using PHP, developing the views using HTML, CSS, Bootstrap, JavaScript and setting up local hosting on Xampp.

## **3.2. SYSTEM ANALYSIS**

The system deals with the general overview including the structure, functions and the various activities performed by the system and its components. The system consists of three roles which include the administrator, the teacher (lecturer) and the student. The administrator is the general overseer of the system. The admin is allowed access to manage the system, manage the users and also manage examinations which are currently being administer. The admin is also able to view examination history from all examinations created. On the other hand, the lecturer is able to create and administered examinations, manage users taking the examinations and also view student reports after the examination. The Student has a limited role wherein he/she is only allowed to take examinations and view his present and past scores.

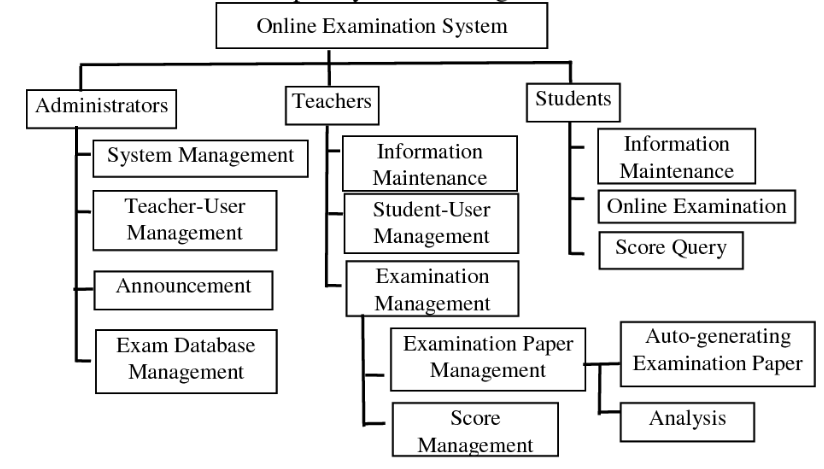


Figure 3.1: System Block Diagram [45]

### **3.2.1. The CBT Application Architecture**

The CBT application architecture was determined based on the functional and non-functional requirements of the system. As a result of these requirements, the chosen architecture for the software system is the client-server architecture. This architecture is a distributed computing system where system operations are split between the Client and the Server. The Client is the client computer and the Server is the server computer from where the software is hosted. This is used where the client requests information from the server through a browser or any other means. The properties of this architecture are:

1. Sharing of resources among computers on a network.
2. Use of one database regardless of the number of users.
3. Easy addition to hardware power.
4. Provision of security clearance to the Human Resource (HR) from a client computer.
5. The database is held by the Human Resource Information System (HRIS) server computer used by the organization providing the services.
6. Data access is impossible if server is not working.

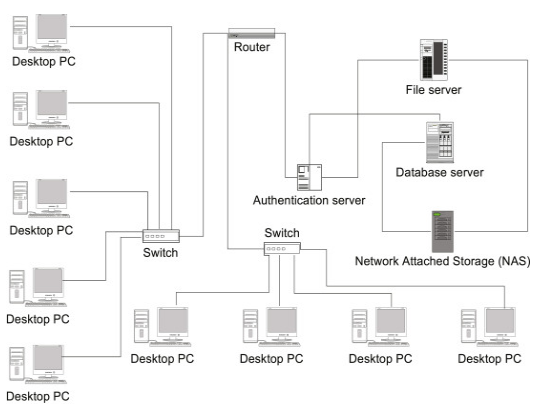


Figure 3.2: Client-Server Architecture[46]

In the system, the server computer is responsible for the processes that require requesting information from the database while, the client computers are used to display requested information to the end user.

## **3.3. SOFTWARE DEVELOPMENT LIFECYCLE**

After careful consideration of the type of software development model to use, the conclusion to the search was the waterfall model. The first person to deliver a proper description of the waterfall model was Royce in the year 1976. The waterfall model is used in this project instead of other models such as the agile model and the prototyping model because of the following reasons.

1. The system requirements are already known.
2. The technology to be used is well understood (CodeIgniter).
3. The project is developed using a milestone-focused development strategy.

The waterfall has an inherently linear structure, which contains about 5 or 6 phases. The diagram in figure 3.3 shows the development process of the waterfall model.

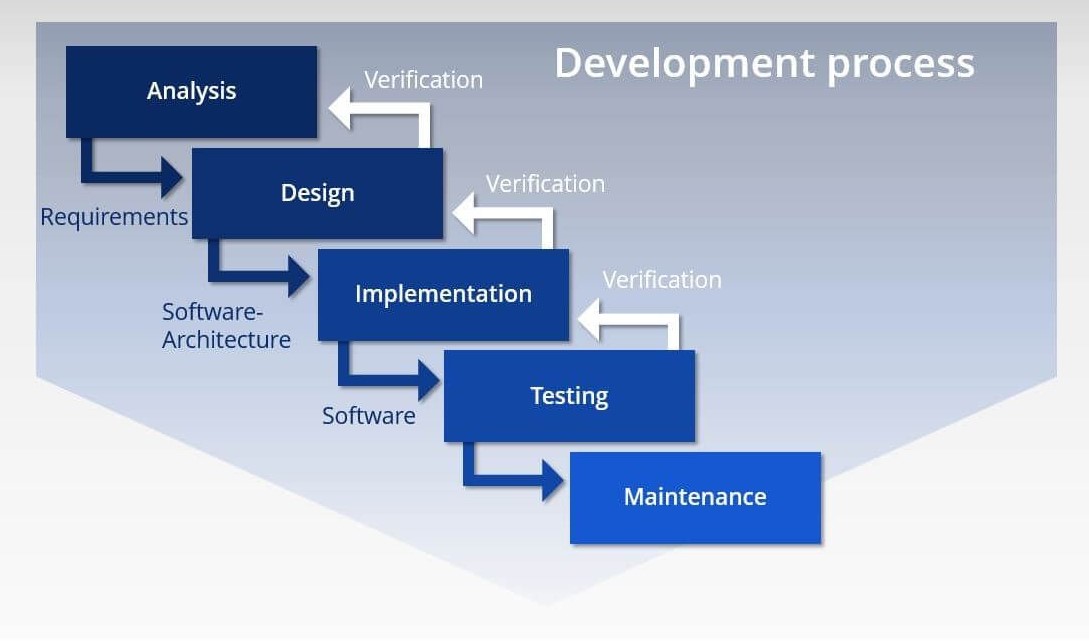


Figure 3.3: Waterfall Model

The phases of the waterfall model consists of analysis, design, implementation, testing and maintenance phase.

1. **Analysis phase:** A requirement specification document is created where the requirements of the system to be created are documented. In this phase, the software requirements of the CBT system were set. The requirements included the system software and hardware requirements and, the functional and non-functional requirements.
2. **Design phase:** The system is designed after carefully studying the requirements specifications. Then the system architecture of the system was designed while putting into consideration the hardware requirements and the user requirements such as the type of laptop to develop this system and the type of users using the system.
3. **Implementation phase**: The system is broken down into units which are developed and tested for functionality. This includes coding using PHP in the CodeIgniter framework. The software is fully developed in this phase and the database is also created and connected.
4. **Testing phase**: The entire system is then integrated together and tested again but for failures, errors, and faults this time. This phase is known as the verification and validation phase. The CBT system is also checked to ensure system and user requirements are met while providing the necessary functional and non-functional requirements.
5. **Maintenance phase:** This phase is done after deployment of the software system. In this phase of the system development, mistakes are corrected, releasing patches to the system and enhancing the products of the system are done. This phase helps to ensure the deliverable satisfies the users while maintaining the quality and performance of the CBT software system.

## **3.4. SYSTEM REQUIREMENT ANALYSIS**

The section states and explains the system requirements for system efficiency. The system requirements include the hardware requirements, software requirements and functional and non-functional requirements.

### **3.4.1. System Hardware Requirements**

The hardware required for the implementation of this project is a laptop or PC system. The system should have the following requirements in order to perform optimally.

1. The Random Access Memory (RAM) of the system should be no less than 3 GB.
2. A system Hard Drive (HDD) or Solid State Drive (SDD) of no less than 25GB.
3. A colored display screen.
4. A minimum processor speed of 2.0 GHz.
5. An efficient mouse, keyboard and power supply.



Figure 3.4: Laptop used for development

### **3.4.2. System Software Requirements**

The software requirements include:

1. A Windows 7, 8,9,10 OS or a Mac OS or a Linux OS.
2. Visual Studio Code Integrated Development Environment (IDE).
3. A XAMPP Local hosting server.
4. PHP, HTML, JavaScript and CSS programming languages.
5. A CodeIgniter 3.0 framework.

### **3.4.3. Functional Requirements**

These requirements include the services that the software system must provide. It involves user interaction, business process and data manipulation. The functional requirements of this system include:

1. Login is based on the user role whether student, admin or lecturer.
2. Students should possess the ability to view assessments and attempt them.
3. Students should be able to see their results and previous assessment histories.
4. Assessments should contain automatic marking and countdown timing.
5. The lecturer should be able to create assessments and add questions to them.
6. The lecturer should be granted the ability to view student attempts on assessment created.
7. Administrator should be able to manage users and courses.
8. Administrator should be able to view all attempts across all assessments.
9. Students, lecturers and administrators should possess dashboards according to their user roles.

These functional requirements can then be represented graphically using the use case diagram of the UML. A use case diagram captures a specific functionality of a system. This UML diagram captures the static parts of the software system, internal and external influences and the requirements of a system (mainly design requirements). These set of diagrams contain the specific use cases, the actors, subsystems and the relationship connections. In this project, the use case diagram is used to share the activities of the application according to the three user roles. The use case diagram helps show the following in a software application.

1. The scope of the software application.
2. The different scenarios whereby the software application interacts with people, an organization or any other external systems.
3. The goals of the system

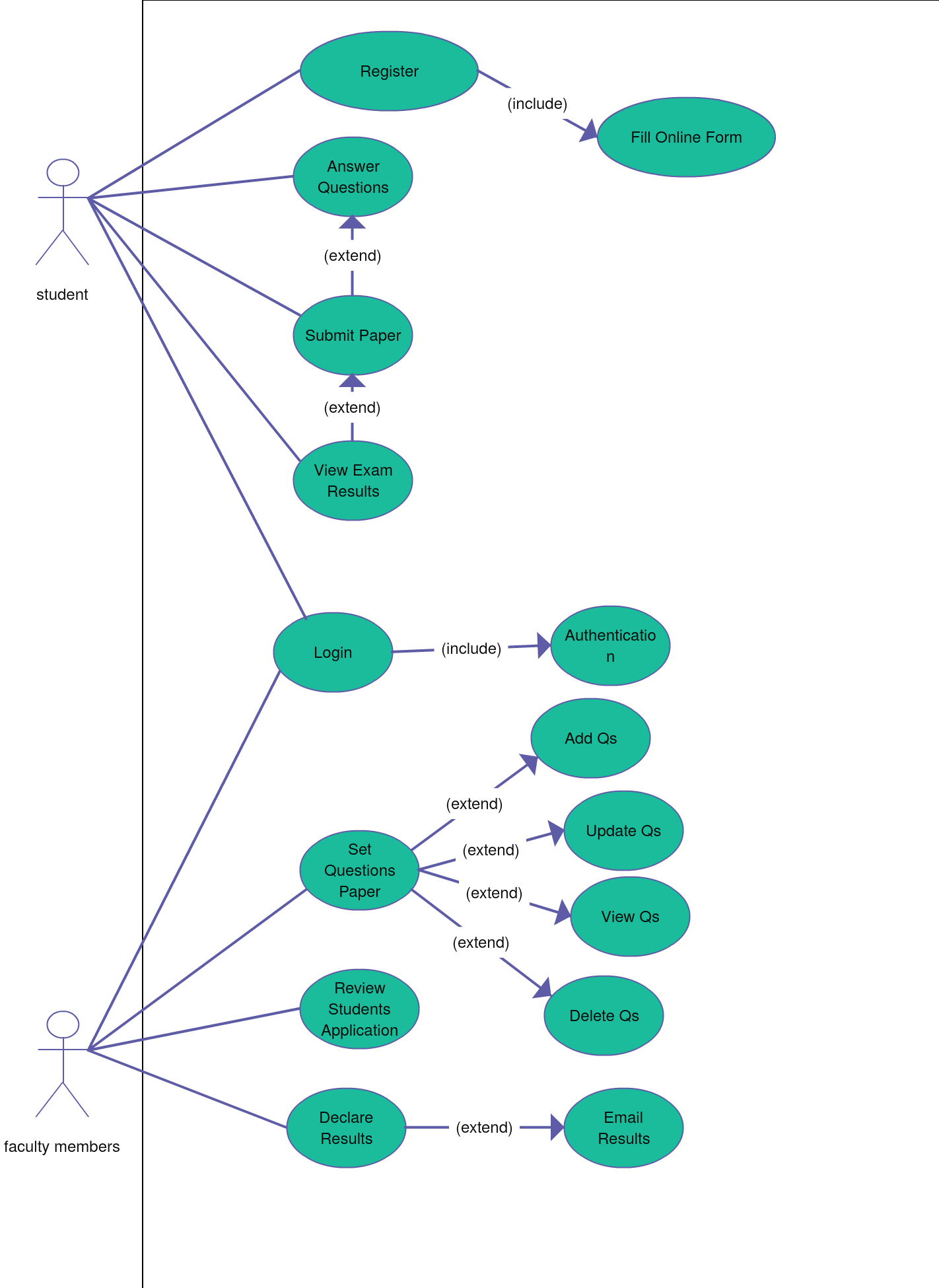
****

Figure 3.5: Use Case Diagram for Student and Faculty Members

The diagram in figure 3.5 represents the use Case diagram of both the students and faculty members who pass through the login page. Their details are ensured to be authentic before being allowed into the system. The student is allowed to take the exam, answer questions and view results. While faculty members are allowed to set question papers, review students and declare results.

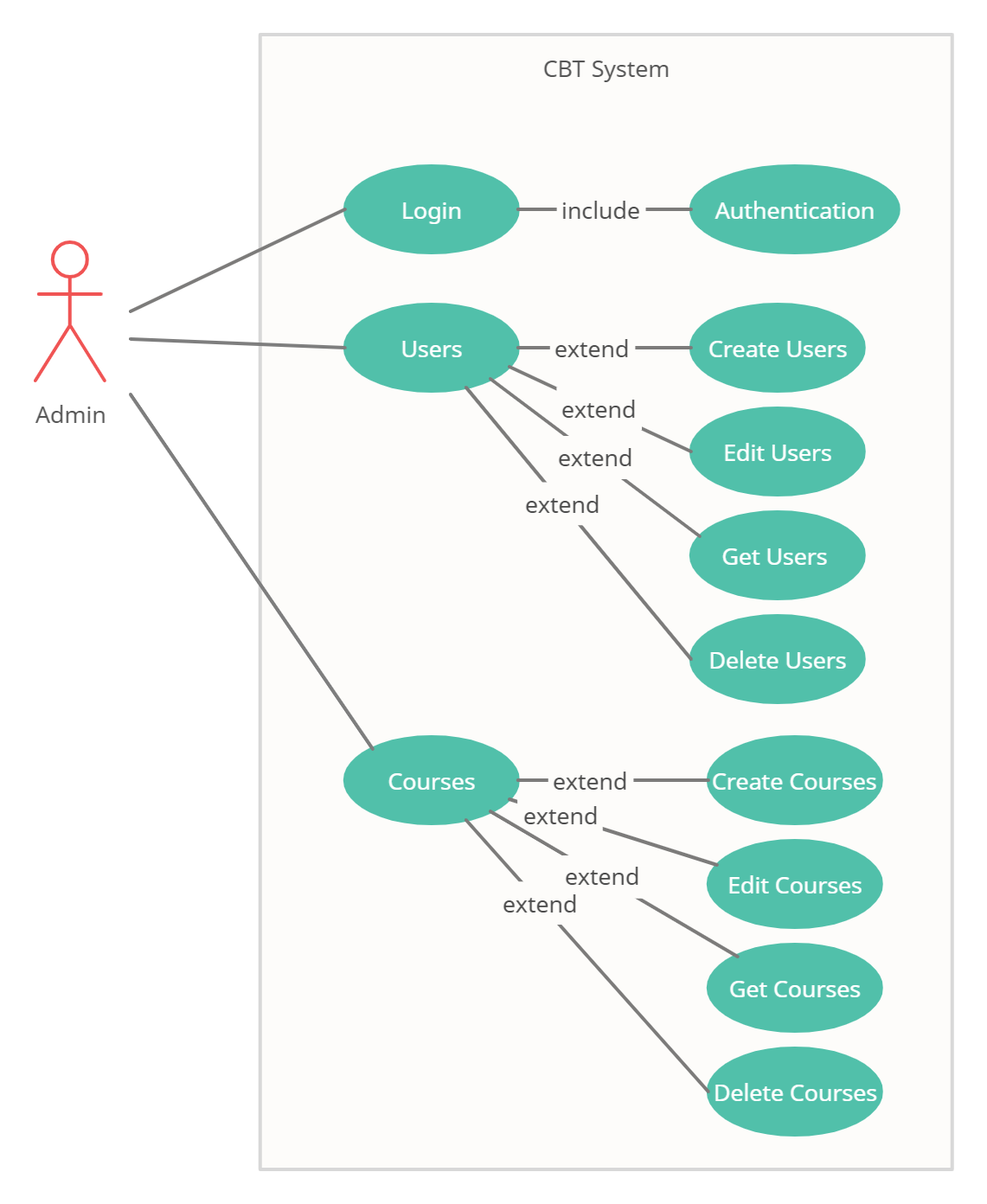


Figure 3.6: Use Case Diagram for Administrator

**-**

The diagram in figure 3.6, is the Use Case diagram for administrators. The administrator passes through the login page where the details are authenticated. If they are valid, the admin is allowed into the system where he/she can manage both users and courses.

### **3.4.4. Non-Functional Requirements**

These requirements represent the criteria that will be used to evaluate the performance of the software system. The non-functional requirements determine the usability and efficiency of the system. The requirements are shown in table 3.1.

Table 3.1: Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| S/N | Requirement | Description |
| 1 | System Security | 1. Only users who are registered are allowed to use the system. 2. Each user’s operation is limited by the role they possess whether lecturer, student or admin. 3. Personal information should be kept private. |
| 2 | System Performance | 1. Students result should be authentic, avoiding any mix ups. 2. Lecturers should be able to view reports of only students that attempted their examination. 3. When timer is up, the examination should be submitted automatically. |
| 3 | System Operation | 1. It is a web based application. 2. It should run on both intranet and Internet. 3. It should be a multi-platform application. |
| 4 | System Capacity | 1. The plugin should be capable of handling a large number of users. |

## **3.5. SOFTWARE DESIGN**

UML provides software architectures, which aids in reducing large and complex systems into smaller components [17]. Krutchen [47] described software architectures to have four various perspectives in which they can be viewed from. They include:

1. Physical
2. Logical
3. Development
4. Process

The physical perspective focuses on the integration and interconnection of the hardware components, which are run by the software. From the logical perspective, a software application or system is divided into components based on their relationships with one another. Also, the development perspective focuses on problems that are related to the implementation of a system. Finally, the process perspective deals with the interaction of system elements in other to achieve the desired result of the system. Software can be modelled with various perspectives in mind however, there are various system modelling languages. [48] Types of modelling languages include the UML (Unified Modelling Language), BML (Business Modelling Language), EPC (Event-driven Process Chain) and the NDL (Network Description Language). The more popular one is the Unified Modelling Language commonly known as (UML).

Unified Modelling Language is a third generation, general-purpose, object-oriented modelling language which is used in the area of software engineering to provide a standard method to design and envisage a software system. The UML consists of two main modelling categories which are the dynamic and the static diagrams. While static diagrams are concerned with the system structure, functional diagrams are concerned with the changes that occur in a system. Examples of static diagram are object, class, deployment and profile diagrams. While examples of dynamic diagrams are sequence, state machine and activity diagrams.

The UML also consists of three main diagram categories. They are the structure diagrams, behavior diagrams and interaction diagrams. They all work together to provide a detailed description of the structure, behavior and functionality of a software system. The diagrams are explained further below.

1. **Structure diagrams:** These diagrams represent the static parts of a system. The represent the structure and the software architecture of a system. Examples of structure diagrams are the Class diagrams and the Component diagrams.
2. **Behavior diagrams:** Behavior diagrams show the behavior and functionality of a software system. Examples of these diagrams are the use case diagrams and the activity diagrams.
3. **Interaction diagrams:** Interaction diagrams explain the flow of data and control of things in a software system. Examples of these diagrams are the communication diagrams and the sequence diagrams.

Furthermore, from across all UML diagram categories, complex or large systems can however be modelled using four major diagrams which are:

1. The use-case diagram.
2. The class diagram.
3. The activity diagram.
4. The sequence diagram.

There are numerous advantages of the UML diagrams in software engineering. They include;

1. It is easy to learn especially due to its graphical nature.
2. It is a properly defined language
3. It provides support from extensive tools.
4. The UML notation is represented in a graphical nature.

### **3.5.1. Class Diagram**

Class diagrams are very popular UML diagrams. They are used for describing, visualizing and also documenting various aspects of a system. They are also used to show the construction of existing executable code on the software application. These diagrams describe the operations of a class, attributes of a class, and the constraints of a system. Class diagrams consist of classes, collaborations, constraints, interfaces, and associations. The class diagram for the CBT application is shown in figure 3.7.

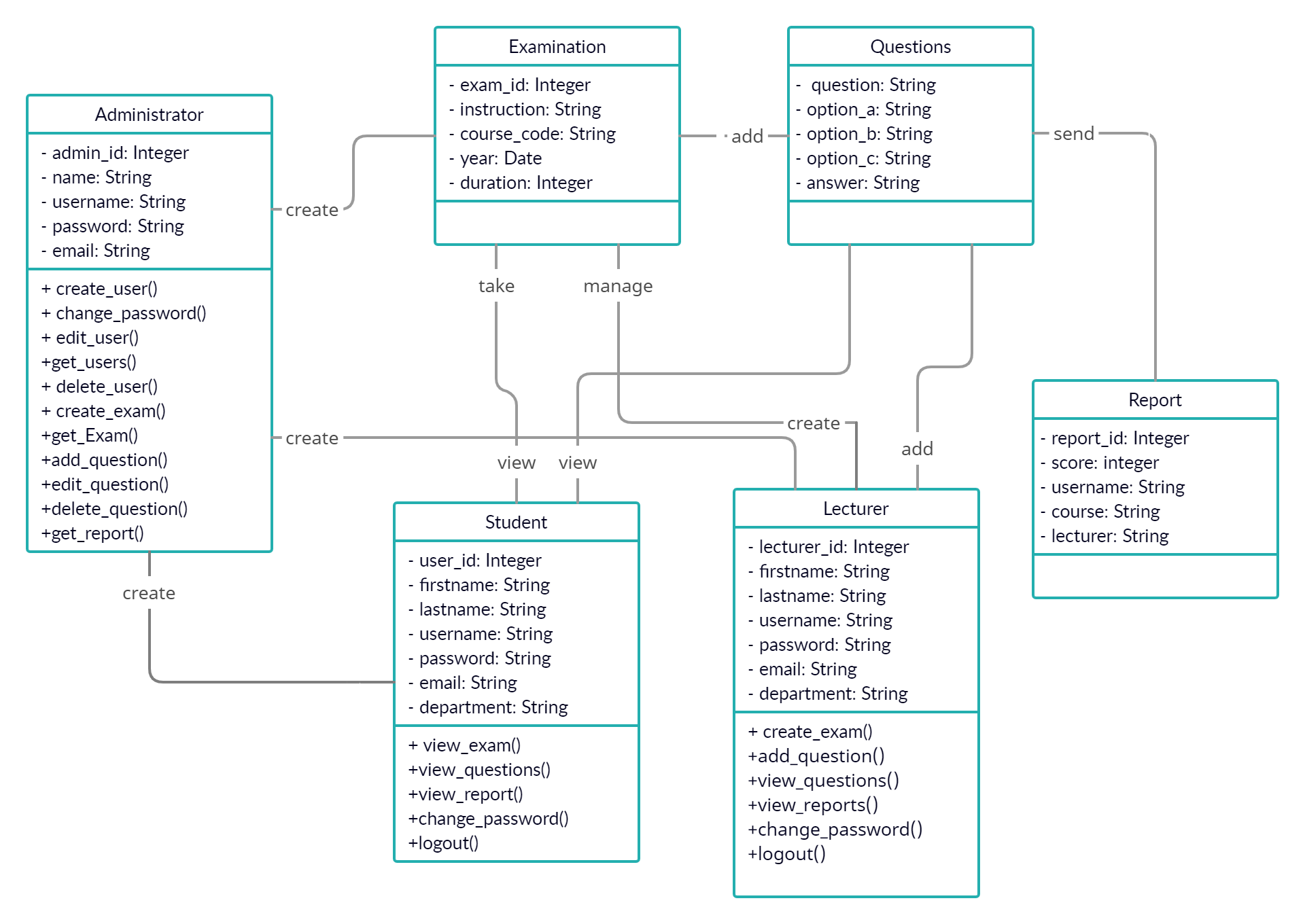


Figure 3.7: CBT system Class Diagram

In the diagram, it can be seen that the admin is allowed to create users and even examination if the need arises. The admin has a total number of endpoints that he has access to including getting and deleting users and same for examinations. On the other hand, the student is allowed to take examinations and answer the questions. After the activity, a report is then generated by the system which can be accessible to the student, the admin and the lecturer who administered it. Finally, the lecturer is in charge of creating and administering examinations and setting questions.

### **3.5.2. Activity Diagram**

Activity diagrams are in the form of flow charts. They show the flow from one activity to another. Activity diagrams represent all types of flow control. They describe dynamic aspects of a system and provide an executable system the forward and reverse engineering technique.

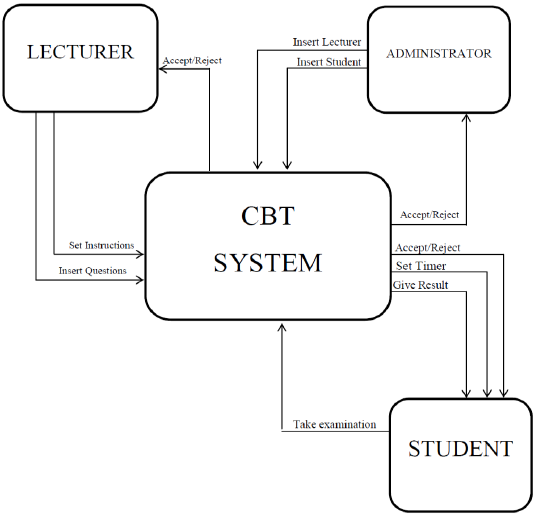


Figure 3.8: CBT System flow of commands.

The first activity diagram in Figure 3.8 shows the flow of activities which can be done by the different user types in the system. The administrator manages both lecturer and students that can access the system. The lecturer can set instructions and questions which the student will take in the form of an examination. The student views his/her result after the examination and the results are sent to both the lecturer and admin to be recorded.

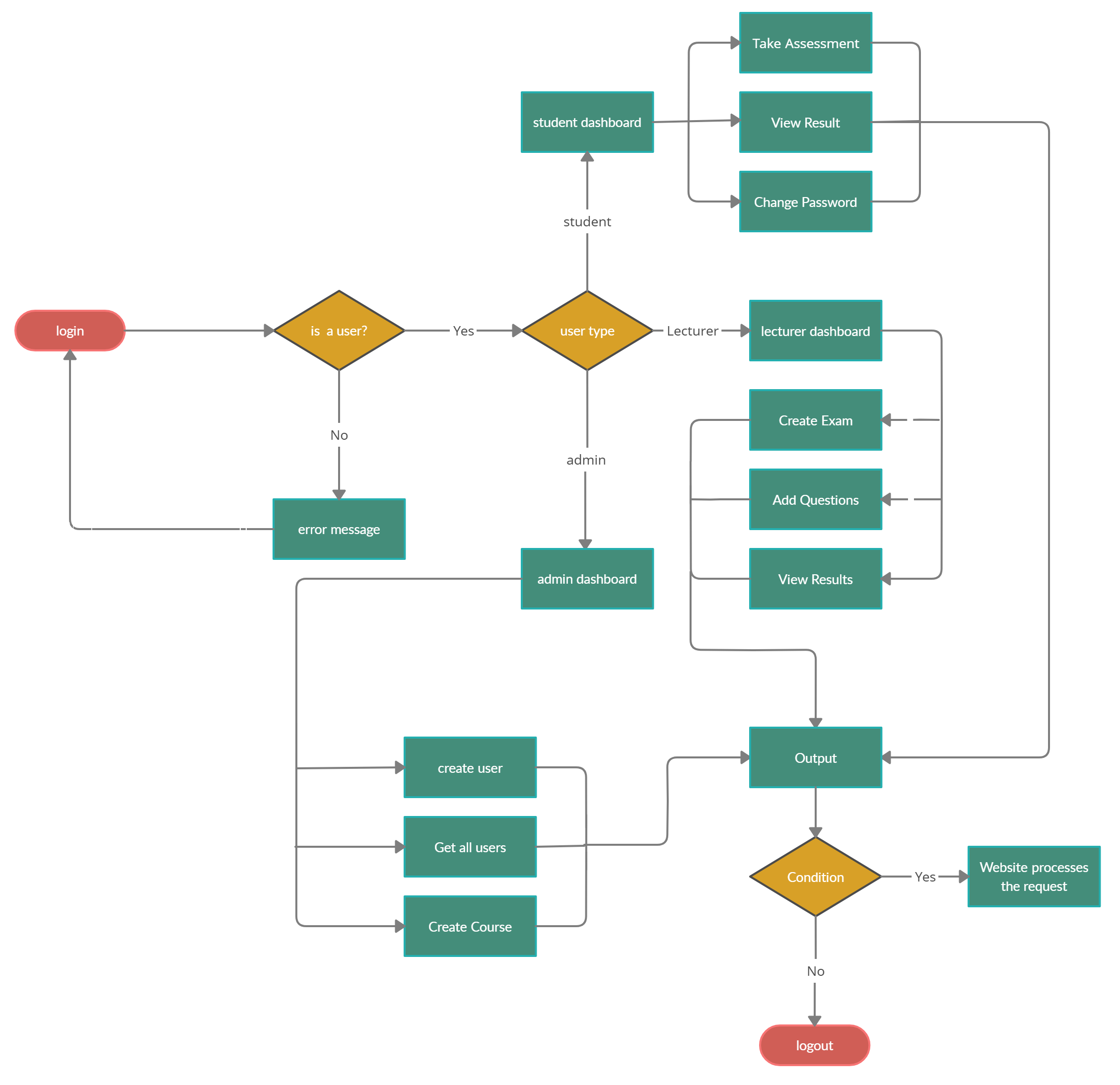


Figure 3.9: The CBT Full-scale Activity Diagram

The second activity diagram shows how the system works from login in. A user is registered by an administrator. If the user is not registered the system throws an error message. If he/she is registered then the type of user is checked. He/she is either an administrator, lecturer or student. After the check, the user is then redirected to the dashboard of their respective roles. According to their different roles, they are allowed to perform different functions. When a function request is sent the request is checked to be valid. If the request is valid the website processes the request. If not valid, the website logs the user out because the session has expired.

### **3.5.3. The Sequence Diagram**

The sequence diagram is another dynamic modelling UML diagram that represents the flow of events or messages in a software system. The processes, objects and actors interact to perform the required function before the end of the lifeline. In unified modelling, lifelines are represented by vertical bars and message flows are represented by arrow lines. These diagrams make use of both iterations and branching. Figure 3.9 represents the sequence diagram of the CBT web application.

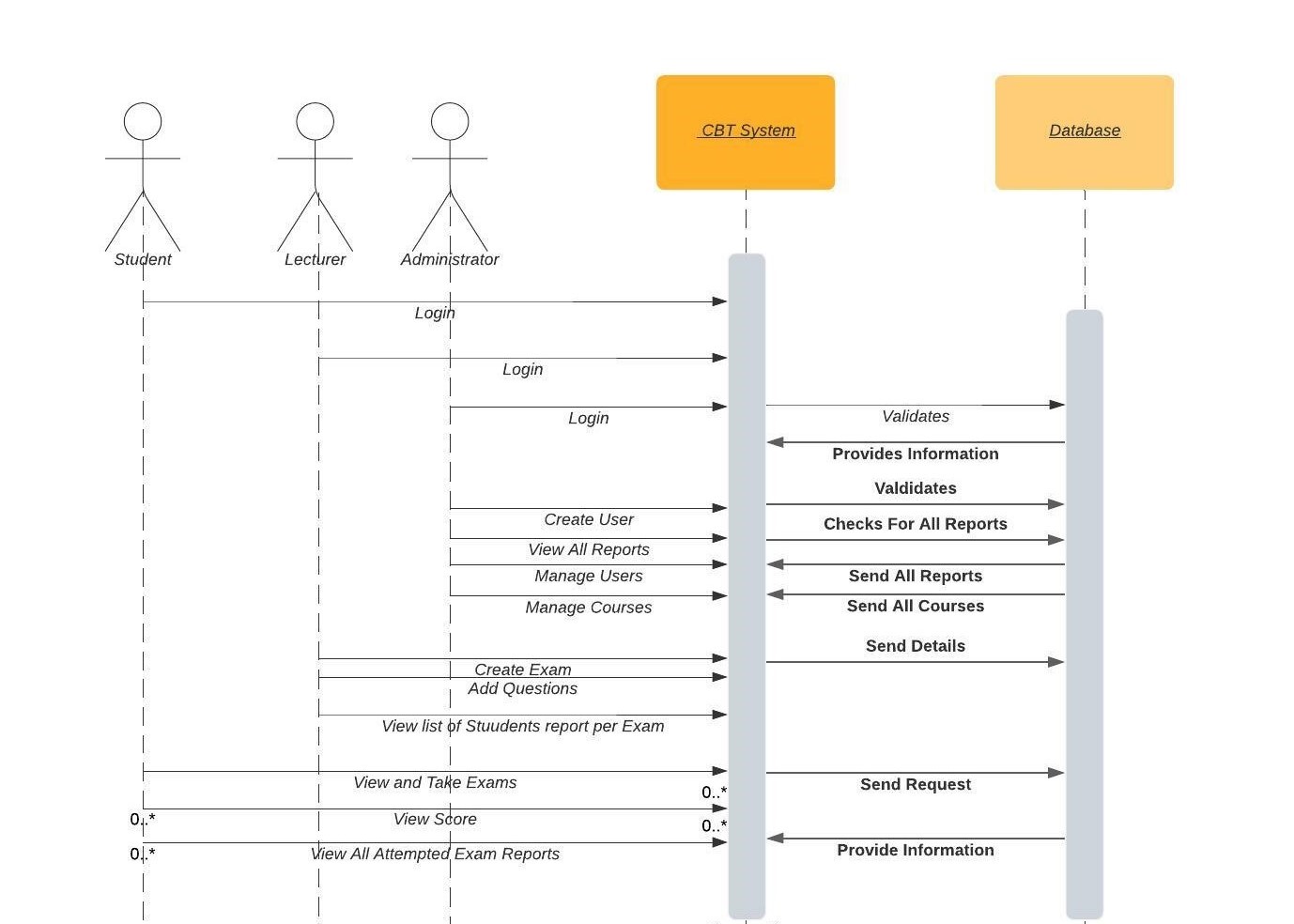


Figure 3.10: CBT Sequence Diagram

In the figure above, the actors in the system are the student, the lecturer and the administrator. The objects are the CBT system and the database. When any of the actors logs in, their credentials are validated with the database. If they are valid, they are permitted entry into the system. According to their level of authorization, they are then allowed to perform various actions which request resources from the databases.

## **3.6. DATABASE DESIGN**

A database is a collection of data that is organized so that it could be accessed, edited and managed with ease. It contains records and files of various data types such as integer, string, double, date and so many others. The database being employed for this project is the MySQL database which is a structured database involving tables with rows and columns. It is accessed locally through the XAMPP server where it can be accessed by clicking the phpMyAdmin link from the dashboard. The database being used for this project is the Orion database. It contains 4 tables which are the examinations, scores, questions and users’ tables. These tables are further explained below:

1. **Users Table:** Table 3.2 contains all the data of users that have access to the software system. It contains names, user names, passwords, emails, ids, genders and roles.

Table 3.2 Users Table

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Field** | **Data type** | **Description** |
| 1 | User\_id | Int | This field is the unique identification factor of the table and the primary key by which incrementing is done. |
| 2 | User\_name | Varchar | This field contains the name a user that has access to the system. |
| 3 | User\_email | Varchar | This field contains the unique email of the user. |
| 4 | User\_username | Varchar | This field holds the user name of the user. It could be a matric number, Staff ID or admin. It is the main means of identification in the software system. |
| 5 | User\_password | Varchar | This field holds the user’s password for accessing the system. It holds the hashed format of the password. |
| 6 | User\_gender | Varchar | This field holds the gender of the user whether male or female. |
| 7 | User\_role | Varchar | This field contains the role of the user which determines the type of access into the system. |

1. **Examinations Table:** The examinations table in Table 3.3 contains the assessments that have been created and can be attempted by a student. It contains fields such as exam\_ids, user\_ids, courses, examination\_name, instruction, duration, password and date.

Table 3.3 Examinations Table

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Field** | **Data Type** | **Description** |
| 1 | Exam\_id | Int | This field is the unique identification factor of the examinations table and the primary key by which incrementing is done. |
| 2 | User\_id | Int | This field contains the id of the user that created the examination. |
| 3 | Course | Varchar | This field contains the Course in which the examination is under. |
| 4 | Examination\_name | Varchar | This field contains the name of the examination to be attempted. |
| 5 | Instruction | Varchar | This field contains the instruction for the assessment to be attempted. |
| 6 | Duration | Int | This field contains the time duration in which the assessment must be completed. |
| 7 | Password | Varchar | This field contains the assessment password which grants user access to the assessment. |
| 8 | Date | Datetime | This field contains the date in which the examination was created. |

1. **Questions Table:** The questions table in Table 3.4 contains all the questions to the various examinations that have been created. It contains fields such as question\_ids, examination\_ids, option\_As, option\_Bs, option\_Cs, option\_Ds, answers and questions.

Table 3.4 Questions Table

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Field** | **Data Type** | **Description** |
| 1 | Question\_id | Int | This field is the unique identification factor of the scores table and the primary key by which incrementing is done. |
| 2 | Exam\_id | Int | This field contains the id of the examination that the question belongs to. |
| 3 | Examination\_name | Varchar | This field contains the name of the examination to be attempted. |
| 4 | Question | Varchar | This field contains a question in the examination being attempted. |
| 5 | Option\_A | Varchar | This field contains the first option of the question to be attempted. |
| 6 | Option\_B | Varchar | This field contains the second option of the question to be attempted. |
| 7 | Option\_C | Varchar | This field contains the third option of the question to be attempted. |
| 8 | Option\_D | Varchar | This field contains the fourth option of the question to be attempted. |
| 9 | Answer | Varchar | This field contains the correct answer to the question. |

1. **Scores Table:** The Scores table in Table 3.5 contains all the scores of users on the various examinations that have been attempted. It contains fields such as ids, user\_usernames, scores, dates, times and intTimes.

Table 3.5 Scores Table

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Field** | **Data Type** | **Description** |
| 1 | Ids | Int | This field is the unique identification factor of the scores table and the primary key by which incrementing is done. |
| 2 | User\_username | Varchar | This field holds the user name of the user that attempted the examination. |
| 3 | Score | Int | This field contains the score or result of a user on the examination attempted. |
| 4 | Date | Varchar | This field contains the date in which the examination was attempted. |
| 5 | Time | Varchar | This field contains the Time in which the result was generated successfully. |
| 6 | Examination\_name | Varchar | This field contains the name of the examination to be attempted. |
| 7 | intTime | Int | This field contains the time duration in which the assessment was completed. |

## **3.7. SOFTWARE IMPLEMENTATION TOOLS**

### **3.7.1. Software Frontend**

**HTML:** HTML is used to develop the view of the application. It was used to develop each page of the views of the application. It is used to structure and display the web page of the project. The PHP functional values and results are pass directly to the html tags to render the specific data to the user.

**CSS:** CSS is used to style the html tags of the web pages. It is also used to provide certain media functionalities like printing. CSS is also used for animating HTML tags, which is made possible by the media CSS attribute.

**JavaScript:** JavaScript is used for the countdown functionality of the system. It is also used for the refreshing animation and for generating various other functionalities throughout the web app.

**Bootstrap:** Bootstrap is used to create an overlay template in which the rest of the frontend is built upon. It also helps with the sizing and general dimensioning of the HTML tags.

### **3.7.2. Software Backend**

**PHP:** PHP is used to develop the backend of the system in the CodeIgniter framework. PHP is used to develop all the functions in both the model and the controller. It is also used to collect data from MySQL queries. It is also used in passing values from the database to the views directly through the html tags.

**MySQL:** It is used to read data from the database and pass it into the backend of the application to make use of the respective database values.

**XAMPP:** XAMPP is used for hosting the application locally on a system before final deployment of the system.

**Git:** Git is used for version control of the system to manage changes that occur during development of the CBT application.

## **3.8. SUMMARY**

The various software components including those used for the frontend and the backend of the application have been presented in this section. The backend of the software explains the database, database integration, MySQL tables and the use of MVC architecture. The frontend explains the integration of the UI in the application making use of HTML, CSS, JavaScript and the Bootstrap framework. The workflow of the application is then explained using different UML diagrams including the Class diagram, Activity diagram and the Use Case diagram.

# **CHAPTER 4**

**SYSTEM IMPLEMENTATION AND RESULT**

## **4.1. INTRODUCTION**

This chapter presents the work done during the development of the CBT application using the CodeIgniter Framework. It explains the integration of the frontend with the backend and the integration of the backend with the database. The implementation of the various building blocks involved in the software system and the performance and quality of the system and components that have been tested are also presented. Finally, this chapter shows the results that have been achieved across the various sections of this project.

## **4.2. WEB APPLICATION IMPLEMENTATION**

The CBT web application developed in this work is named Orion. It provides lecturers with the resources to administer tests, examinations and other assessments to be taken by students while under the management and supervision of an administrator. User accounts can only be created by an administrator. The application has an authentication system which allows users to login with a username and password and also logout when required. As shown in figure 4.1, Users can only be allowed access into the application after being authenticated from the login page. Based on their given roles the users are redirected to either the admin dashboard, student dashboard or lecturer dashboard. Once logged in, students can then view their profiles, change their passwords, view their prior score histories or take new assessments. Lecturers can then change their passwords, view results from assessments and create new assessments while the Admin is in charge of managing both users and courses.

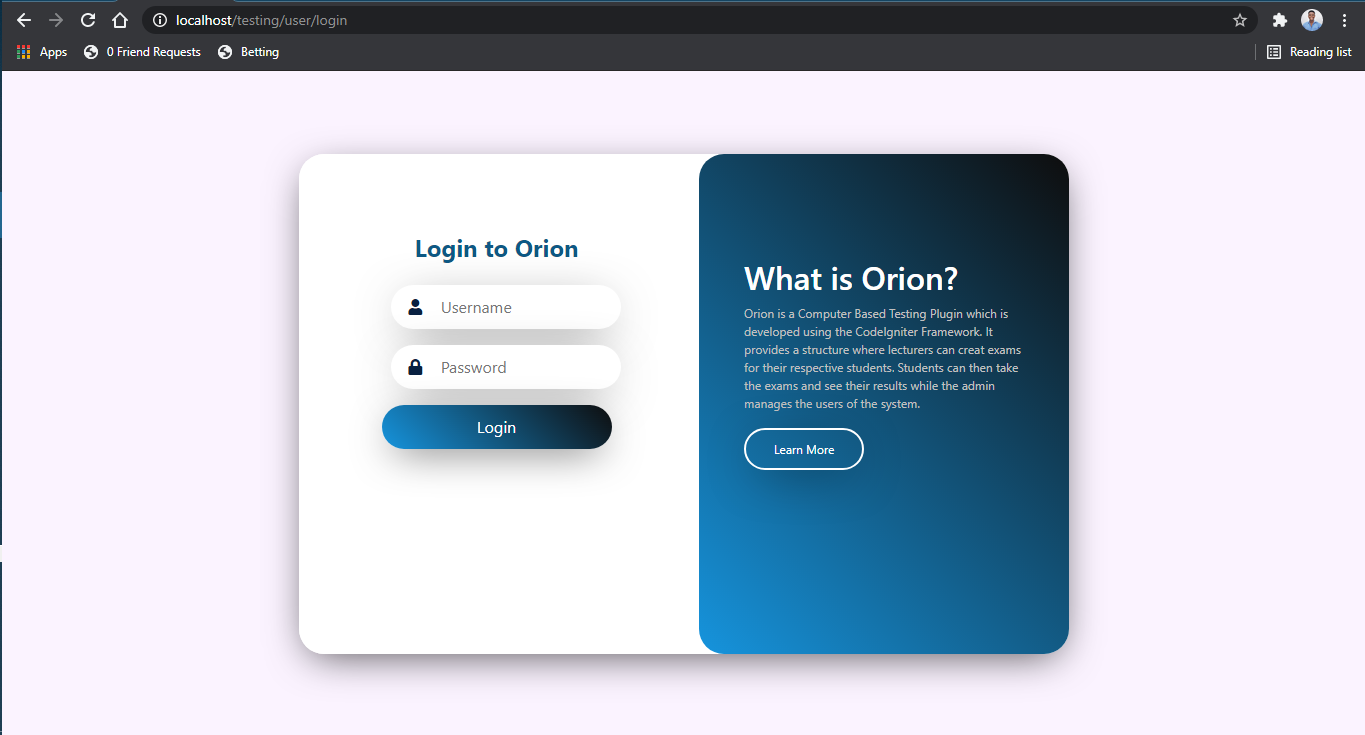
****

Figure 4.1: The CBT system Web Application Login Page

### **4.2.1. CodeIgniter Framework**

The CodeIgniter Framwork was used to develop this system. The framework made use of the following software in order to work efficiently:

1. Visual Studio Code
2. PHP version 8.0
3. XAMPP
4. PHP Intelliphense

Visual Studio Code commonly known as VSCode as shown in figure 4.2 provided formatting and software running functionalities to help detect errors while the PHP intelliphense was the extension used by VSCode in order to run and execute PHP code effectively.

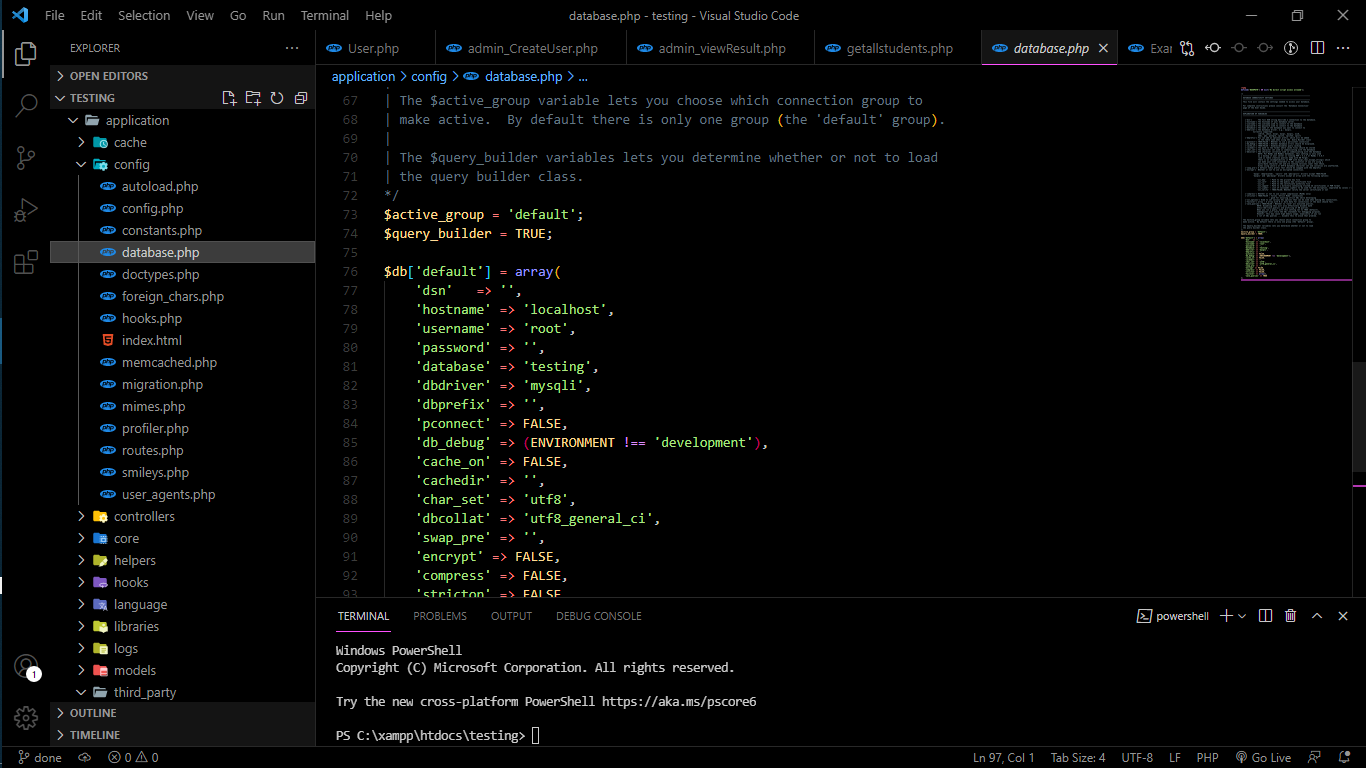


Figure 4.2: VSCode view of the CodeIgniter Framework

The system was developed using the Form validation, Session and Pagination Libraries. They were loaded through the **autoload.php** file in the Config folder under the Application folder.This was done through the following command:

$autoload['libraries'] = array('form\_validation', 'session', 'pagination');

Errors were displayed in HTML format, which was accessible when the webpage was refreshed. The errors were thrown by the inbuilt CodeIgniter standalone index.php file.

### **4.2.2. Student Section of the CBT system Web Application**

The first page of this section is the student dashboard in Figure 4.3, which provides user details and the total exams attempted by the user. The second page is the Assessments page in Figure 4.4, which shows all the assessments to be accessed by the student. Finally, the fourth page in Figure 4.6, is the page that shows the student attempting the examination.

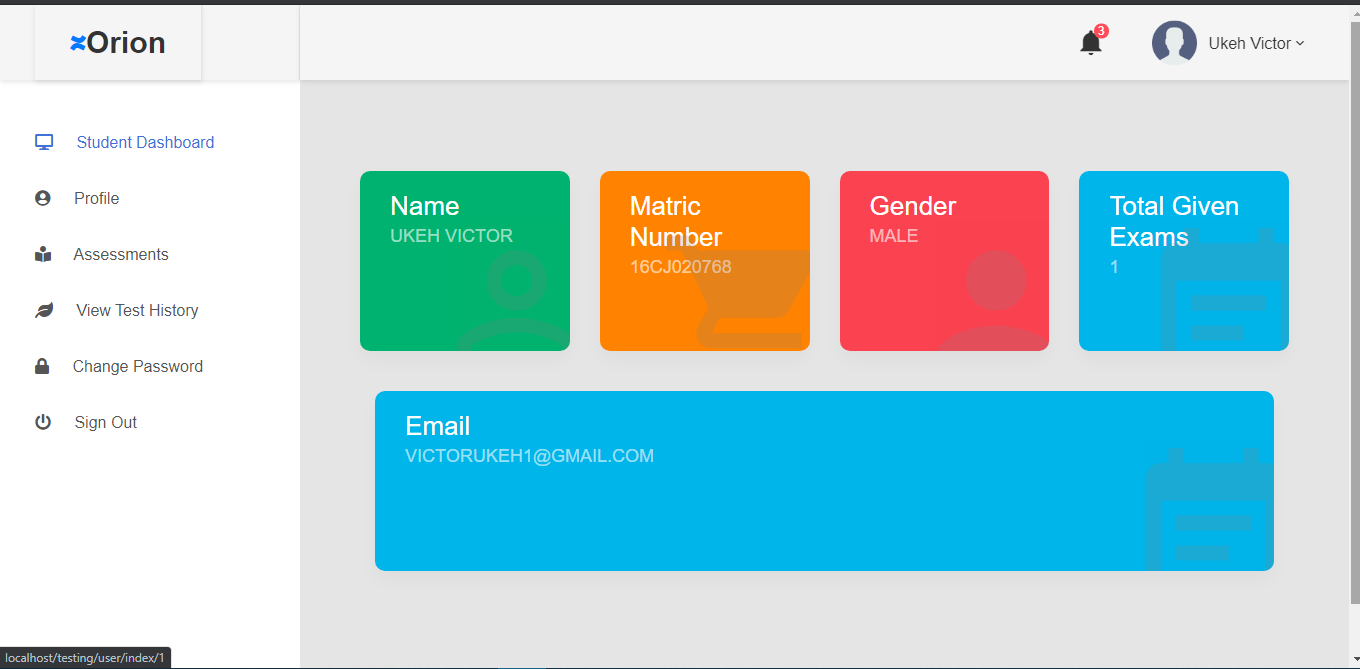


Figure 4.3: Orion Student Dashboard

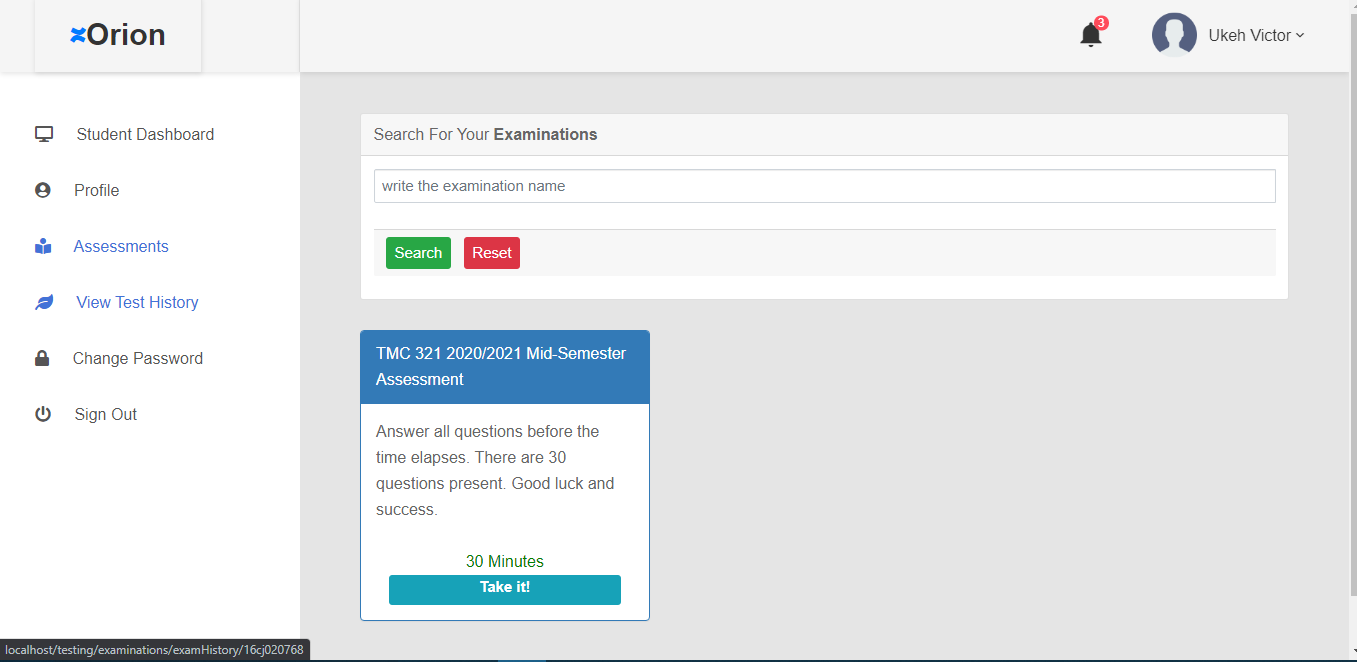


Figure 4.4: Orion Student Assessment page

**`**

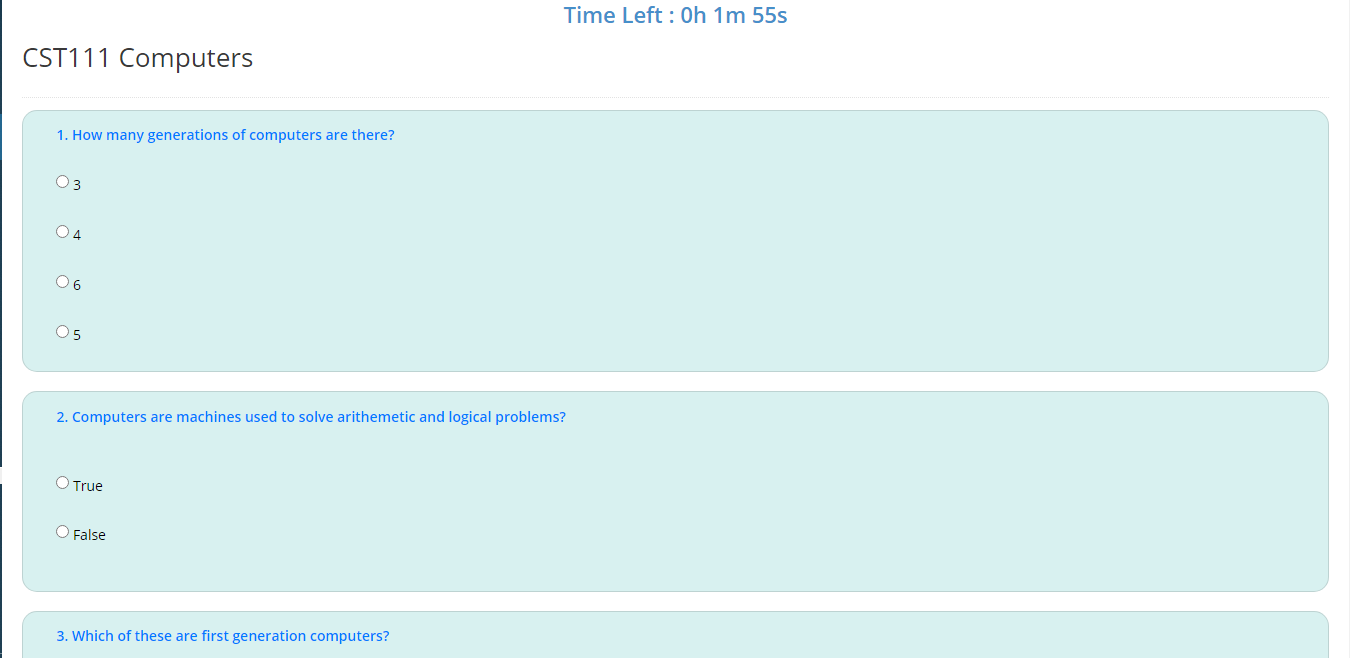


Figure 4.5: Orion Examination Page

### **4.2.3 Lecturer Section of the CBT system Web Application**

The first page of this section is the lecturer dashboard in Figure 4.7, which shows the details of the lecturer and the total examinations they have created. The second page in Figure 4.8 of this section is the “created exams” page which shows the assessments created by a lecturer and includes add question, delete and edit functionalities. The third page in Figure 4.9 of this section is the “create exam” page that provides the lecturer with the input to create an examination. The diagrams in Figures 4.10 and 4.11 are from the “add question” page that provides the lecturer with the ability to add any type of question. Finally, the diagram in Figure 4.12 shows the scores of students that have attempted the examination.

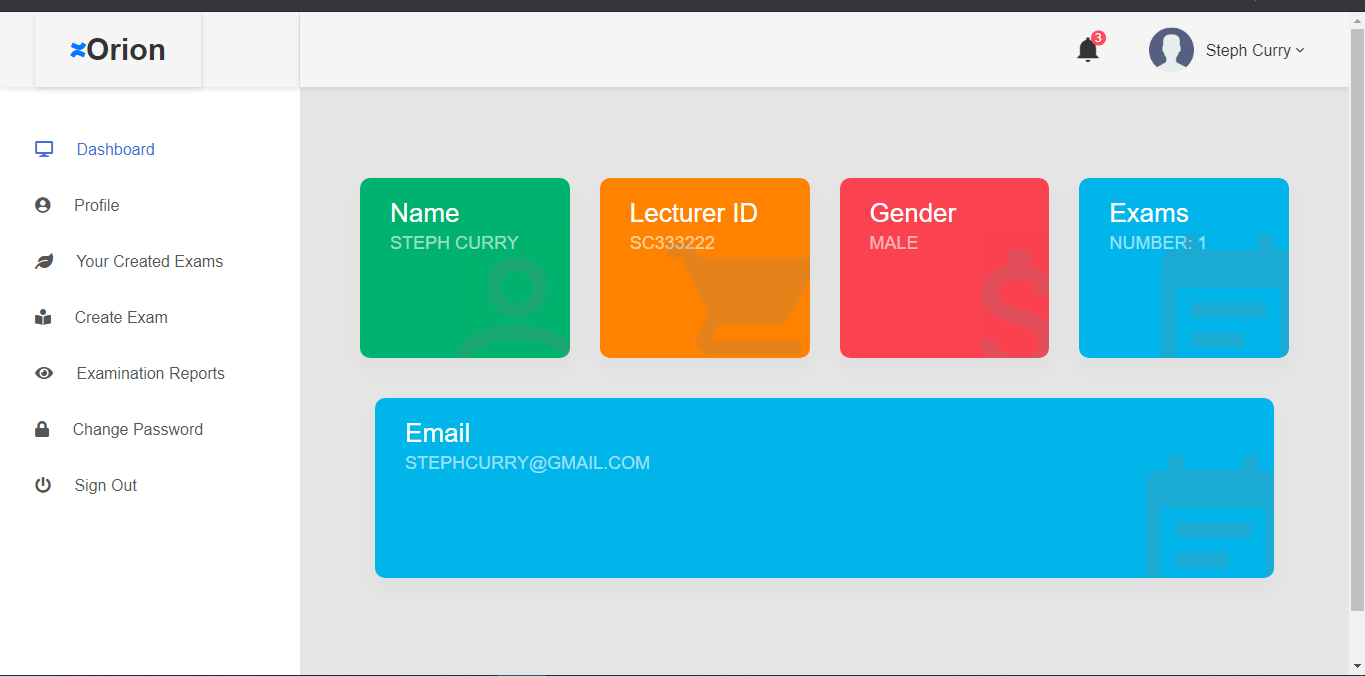


Figure 4.6: Lecturer Dashboard

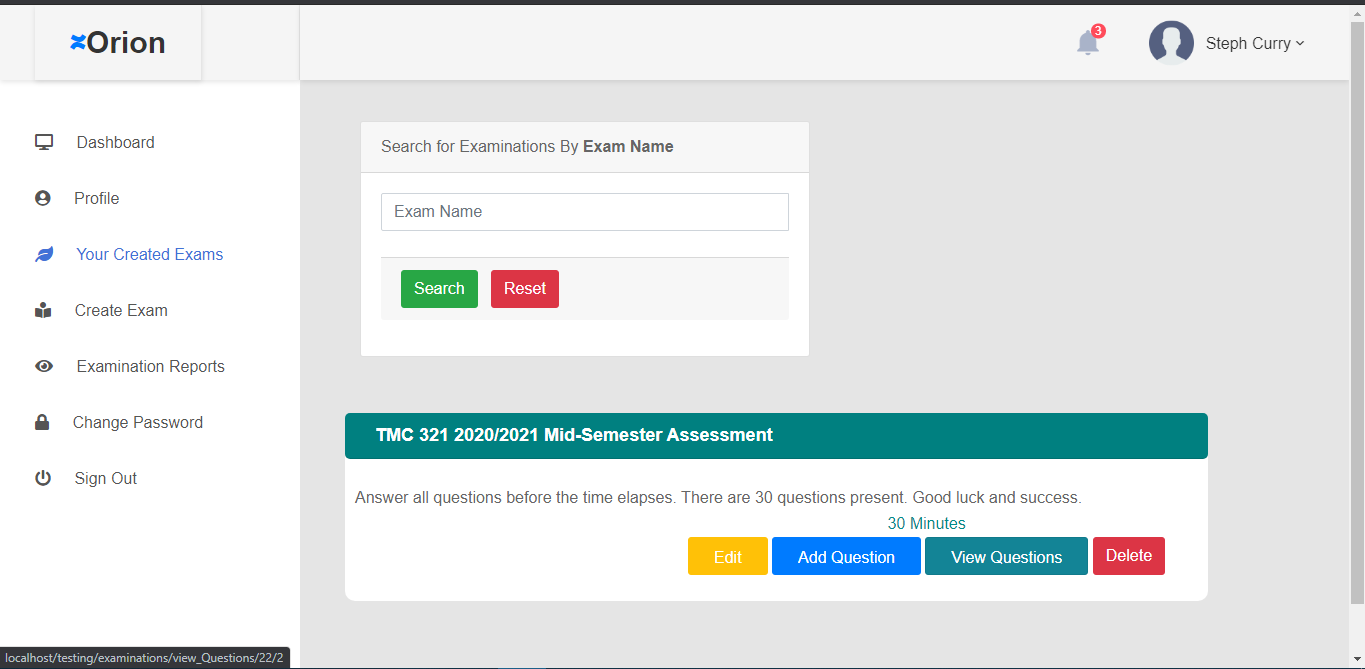


Figure 4.7: View Created Exams Page

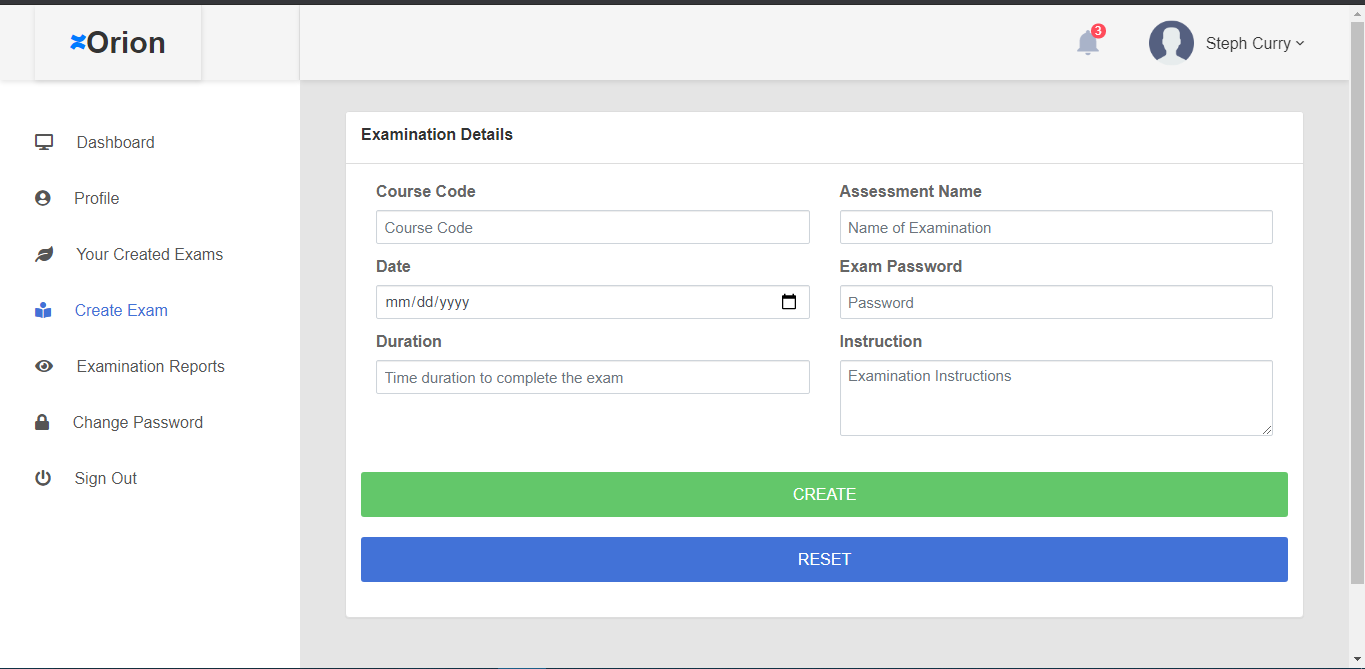


Figure 4.8: Create Exam Page

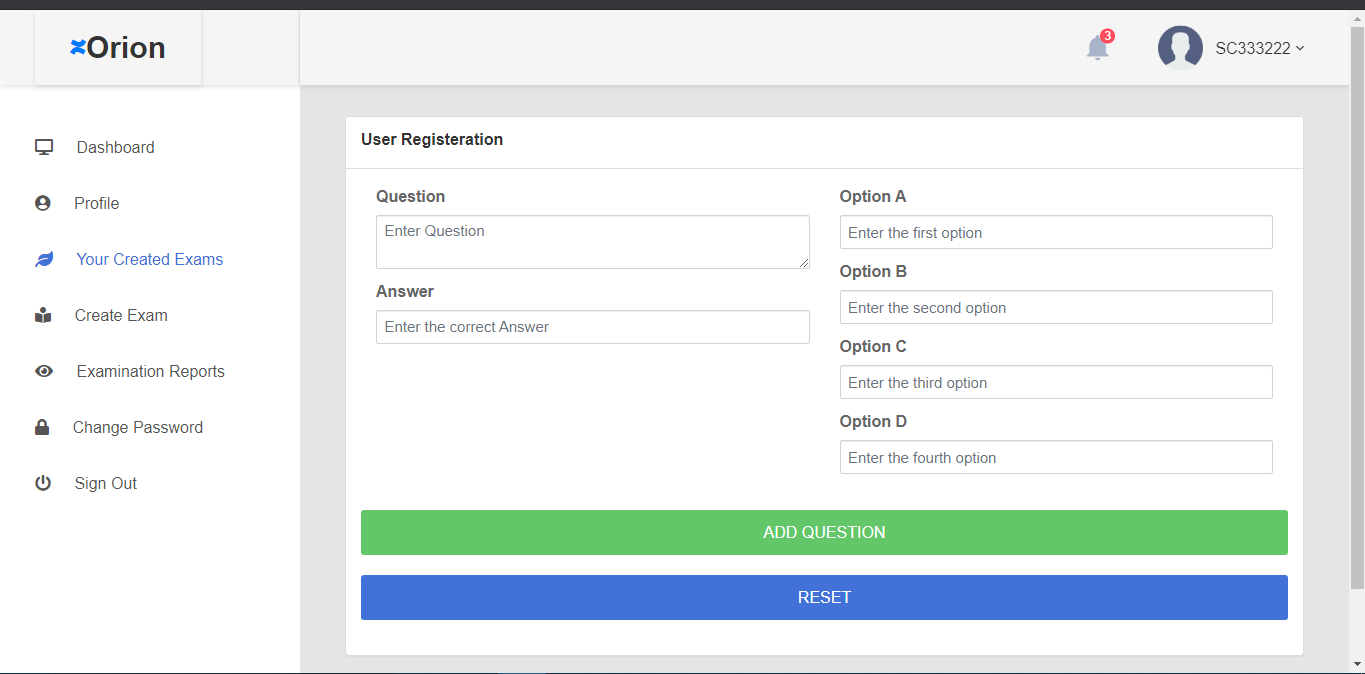


Figure 4.9: Add Question Page

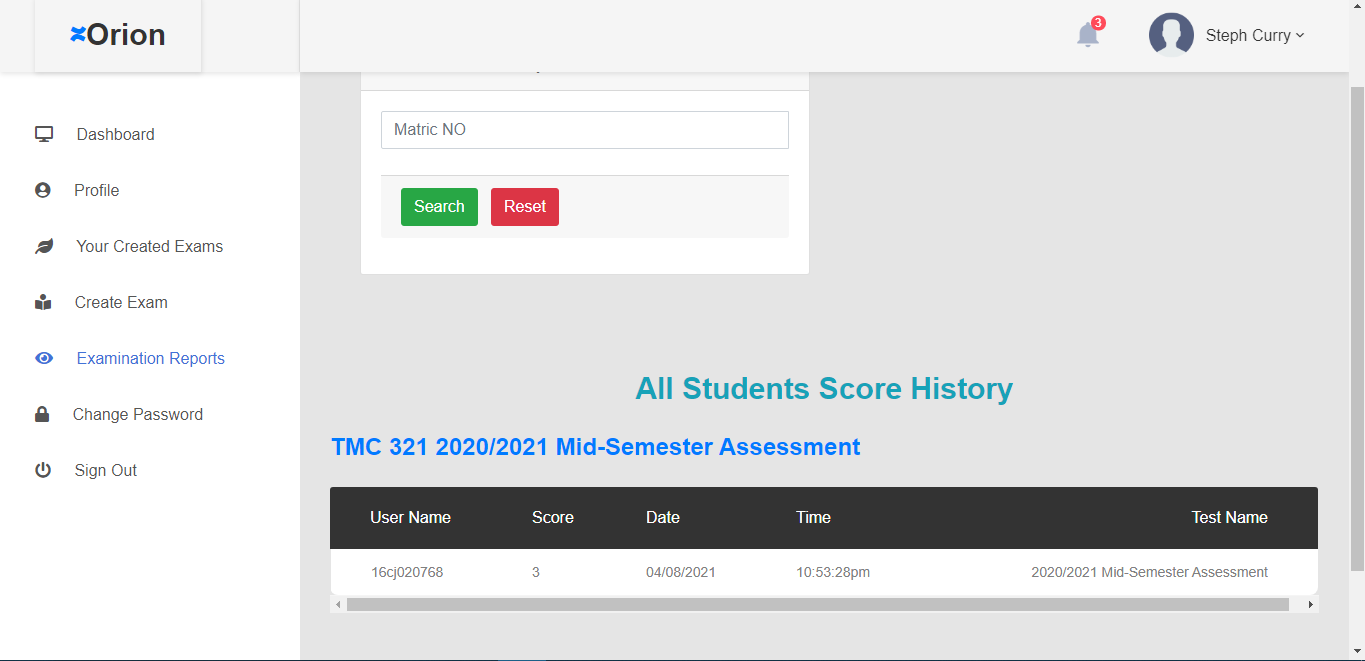


Figure 4.10: Student Examination History

### **4.2.4. Administrator Section of the CBT system Web Application**

The administrator has a dashboard as shown in Figure 4.13 that shows the total number of users using the system, the total number of lecturers and students and finally the total number of examinations. The second page in Figure 4.14 shows the page for registering a user. The administrator is also able to perform other functions like view lecturers and students, managing courses and view all student reports amongst others.

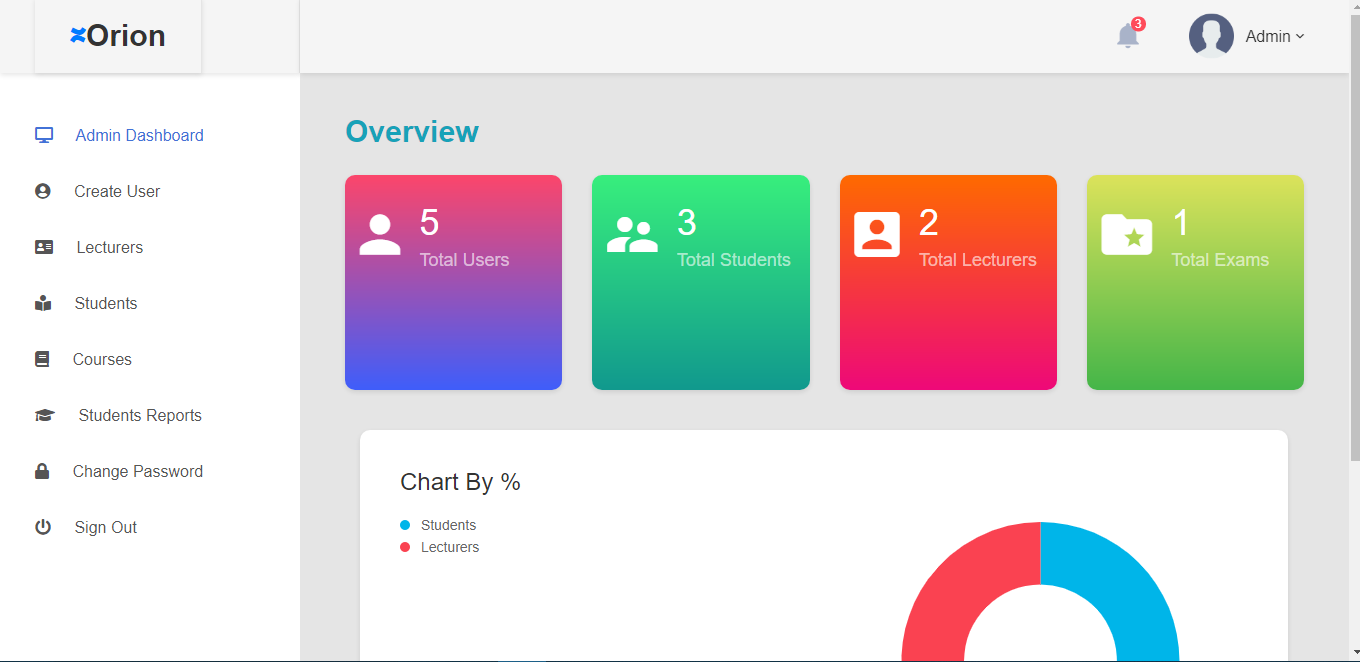


Figure 4.11: Administrator Dashboard

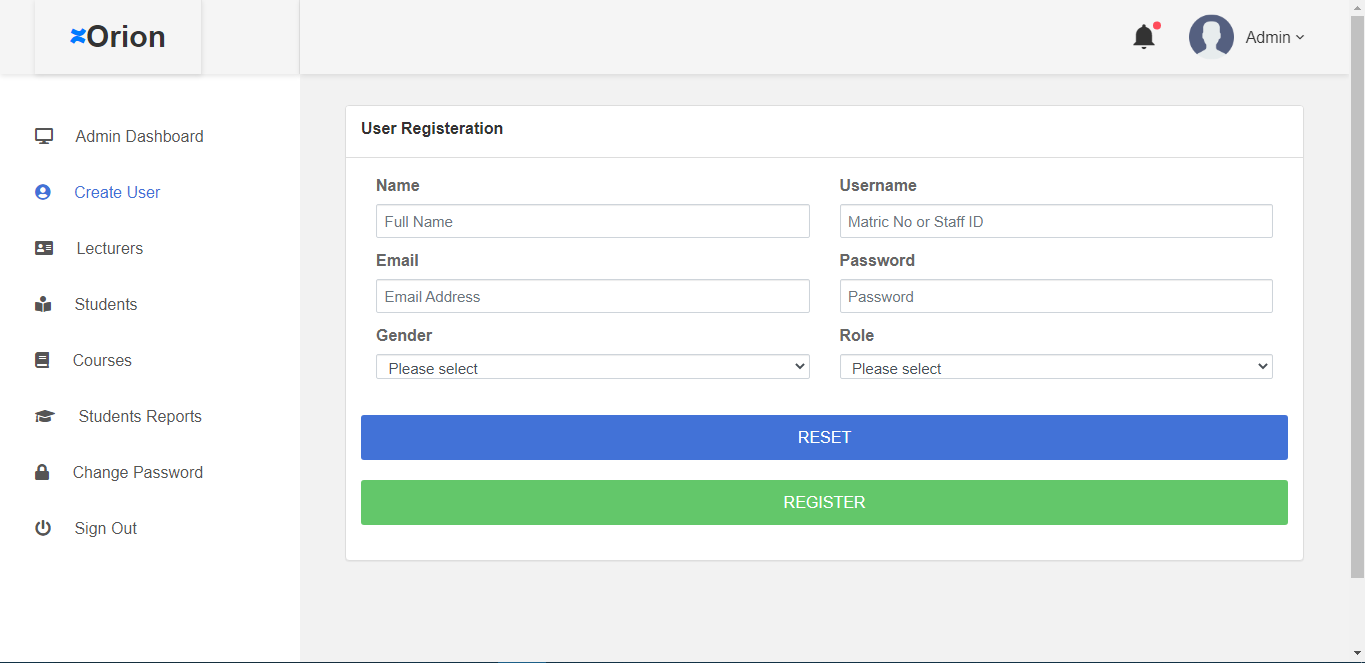


Figure 4.10: User Registration Page

## **4.3. SOFTWARE TESTING**

The software system was passed through various tests to ensure that the software system works as it was intended to. Software testing involves verifying the system quality and ensuring the system meets pre-defined requirements. The tests carried out on the system include unit testing and deployment testing. The reasons for testing are:

1. To check the performance of the software system.
2. To check the reliability of the system
3. Assurance checks.

### **4.3.1. Unit Testing**

The unit testing of the system involved testing each line of the code and each unit of the system to ensure optimal efficiency. The testing was done manually due to size of the system as this approach was more convenient. The entire system was tested from the login page to the entire system. The tool used for testing is the free online tool known as PHP sandbox as shown in figure 4.17. The PHP sandbox performs checks on PHP code to check if it meets required standards and checks if there are errors. Errors were found and corrections were made as shown in Figure 4.18. Due to the errors, bug fixes were done and the result passed as shown in Figure 4.19.

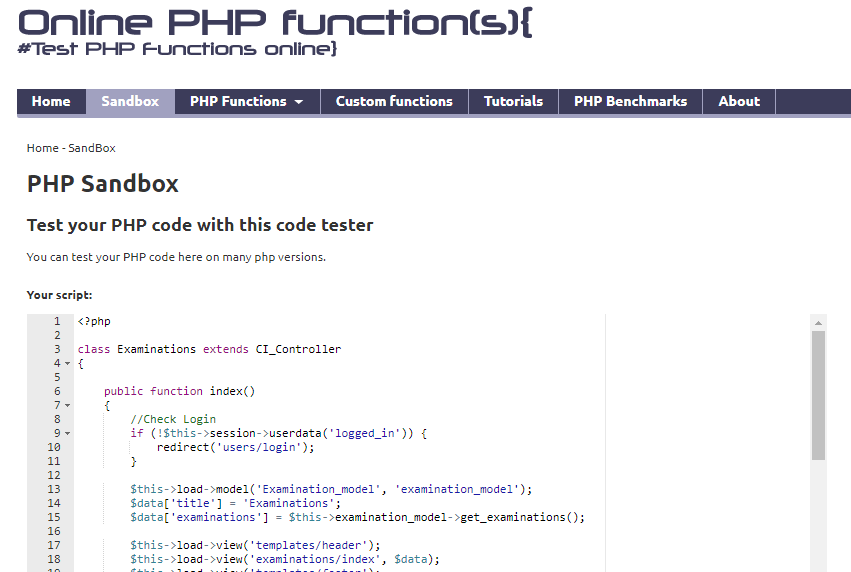
****

Figure 4.11: PHP Sandbox Interface

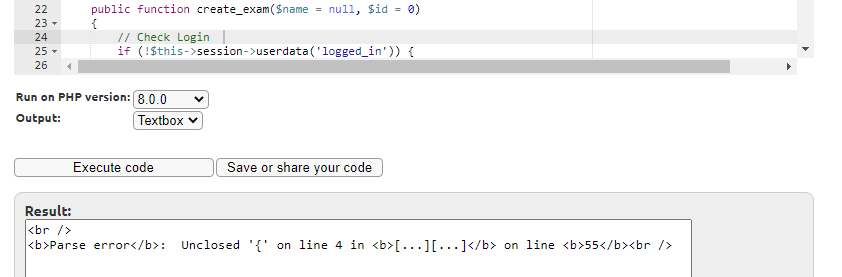


Figure 4.12: Error Result

****

Figure 4.13: Correct Result

### **4.3.2. Deployment Testing**

This testing was done to ensure the application worked the way it ought to work when hosted online on a hosting platform just as it was with local hosting. The site was tried on various browsers including Firefox, Google Chrome and Microsoft Edge to ensure it works regardless of the user’s browser. Other minor tests such as spelling tests were also done. It can currently be accessed via <http://orion.com.ng> link.

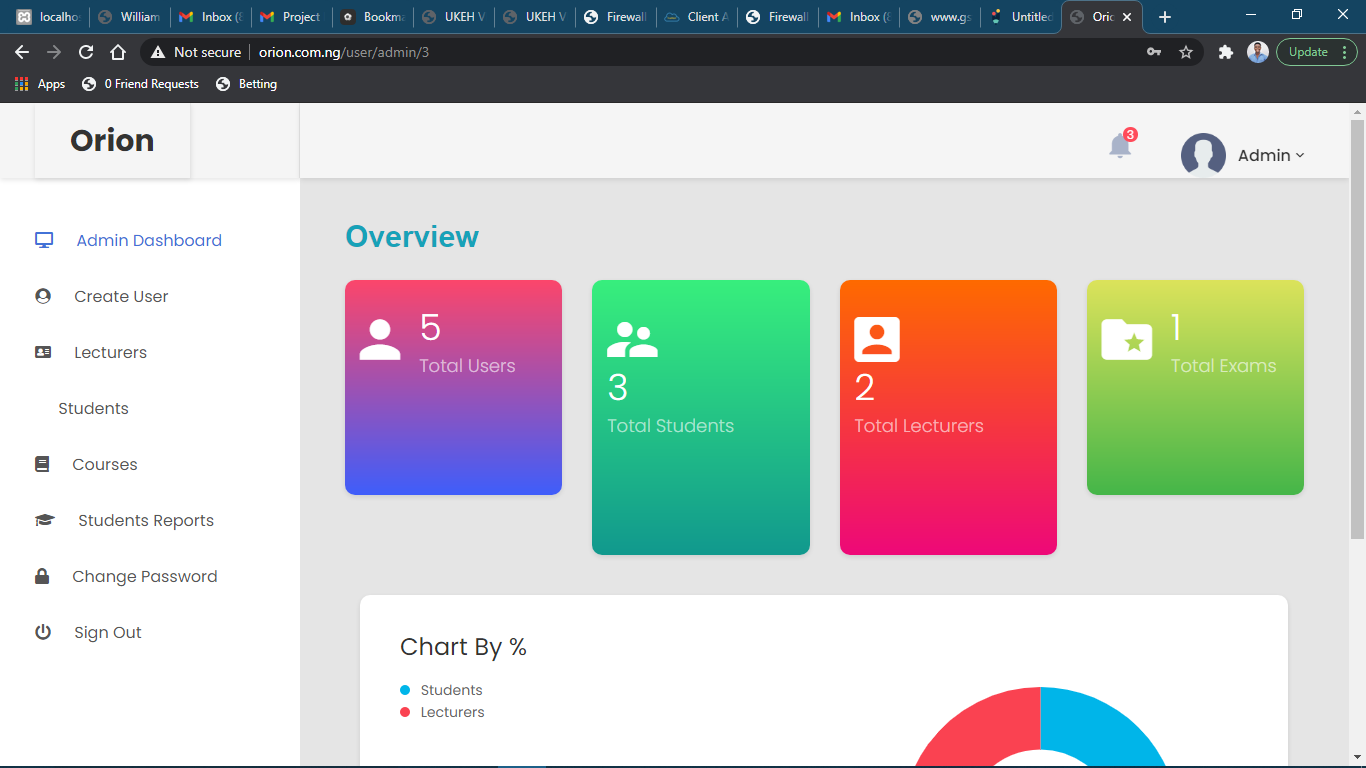


Figure 4.14 Accessing the CBT application on Google Chrome.

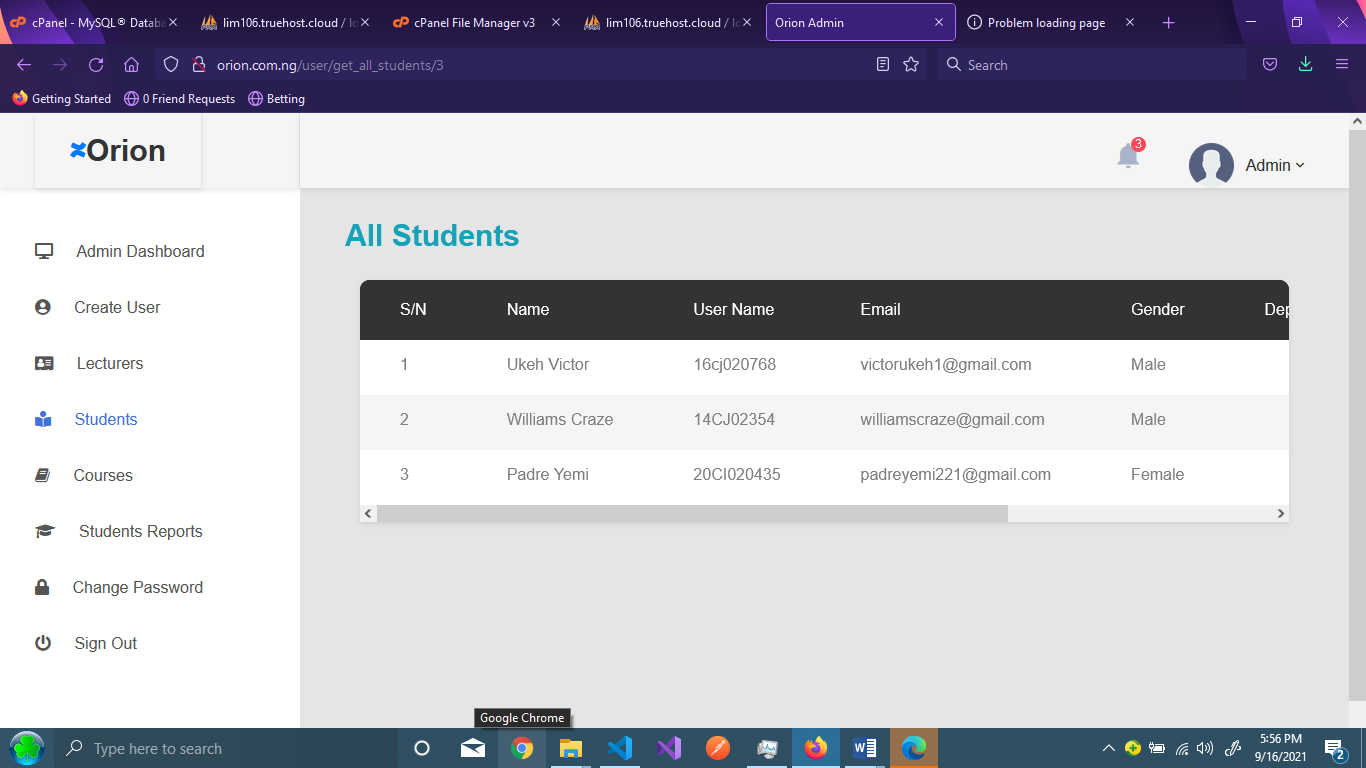


Figure 4.15 Accessing the CBT application on Firefox.

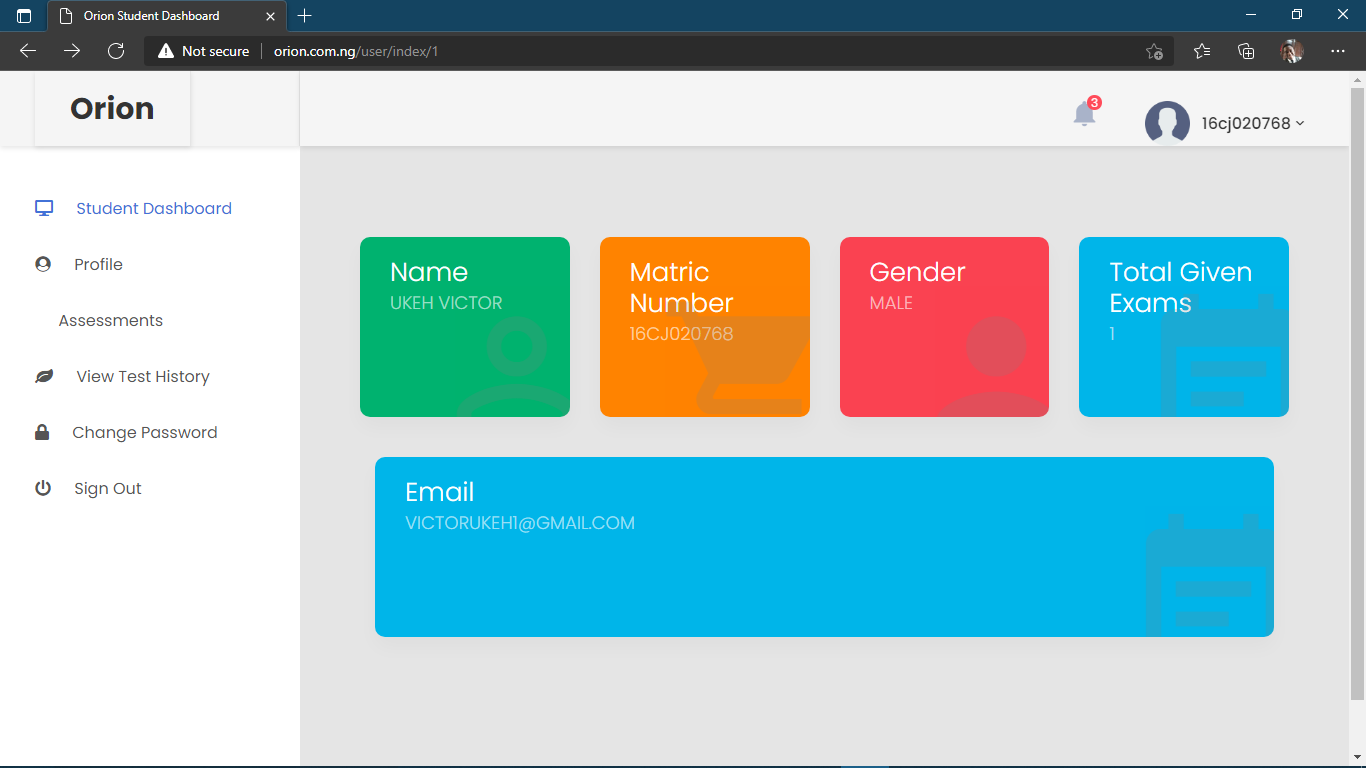


Figure 4.16 Accessing the CBT application on Microsoft Edge.

## **4.4. SUMMARY**

The chapter describes the testing and implementation of the software system. It shows how the system was implemented using CodeIgniter, Bootstrap, CSS, HTML and JavaScript to develop the software system why testing locally with XAMPP and finally hosting the application online. The chapter also expatiates on the different tests the system went through to ensure the system gave the best possible performances and the actualization of both functional and non-functional requirements.

# **CHAPTER 5**

**CONCLUSION AND RECOMMENDATIONS**

## **5.1. INTRODUCTION**

This chapter provides the summary of the project as a whole including the summary of previous chapters provided, recommendations to aid improvement and the knowledge gained during the timeline of this project. Finally, it states the challenges encountered during the course of this project.

The CBT system which was developed in this study was designed to aid lecturers conduct efficient CBT examinations. This software system enables lecturers to create assessments, set questions and generate reports of students that have undertaken an exam. It also helps students to take examinations in a user-friendly environment where they are allowed to see the score of their assessments immediately after attempting it. The software system can be accessed through a browser on most popular operating systems. After successfully completing the project it is deduced that the project satisfies the aim and objectives, which had been set out earlier during the analysis phase.

## **5.2. KNOWLEDGE GAINED**

During the course of this project, I learnt a number of different software technologies and methods of designing software systems including, things to take into consideration. The following are a few things learnt during the course of this study:

1. **Software Development Models:** The different models were learnt in order to choose the right model for the project.
2. **UMLs:** The use of UMLs were crucial in modelling the software system using graphical means
3. **System Architectures:** The various system architectures were learnt in order to select the most efficient architecture for the system.
4. **Hosting:** Knowledge on how to host a system software.
5. A deeper knowledge in the use of the CodeIgniter framework and other programming languages needed in developing this project.

## **5.4. ACHEIVEMENTS**

The design and implementation of the CBT application was achieved including core features like countdown timing, automatic marking and viewing of results. Hosting of the application, efficient user authentication and quality system responses to various clients’ requests were also successfully achieved.

### **5.4.1. Challenges Encountered**

Some challenges were encountered during the development of the project. They include:

1. Incorporating PHP code into HTML views
2. Interrupted Power Supply
3. Wi-Fi related issues

### **5.4.2. Limitations**

The project was found to have certain limitations since no software system is 99.9% efficient. The limitations of this system are:

1. The software works faster and better on higher grade operating systems.
2. Not efficient for an organization of more than 10,000 persons.
3. PHP flashdata tends to distort the page for about 0.5s when it is shown.
4. Due to the unavailability of component based rendering there is a slight delay here.

## **5.5. CONCLUSION**

The CBT application was able to meet its aim and objectives. It can be therefore used for schools and other organizations interested in employing CBT technology. The challenges faced from paper based testing, high spending on CBT and other vices can therefore be curbed using this software application.

## **5.6. RECOMMENDATIONS**

To improve the output and effectiveness of the CBT system. The following recommendation should be taken into considerations.

1. Use of either React js, Vue js or Angular js to reduce delay time and avoid repeated code.
2. Automatic SMS messaging in case of new logging in from a different device.
3. Display panel of questions attempted.

# **REFERENCES**

[1] A. S. Abubakar and F. O. Adebayo, “Using Computer Based Test Method for the Conduct of Examination in Nigeria: Prospects, Challenges and Strategies,” *Mediterr. J. Soc. Sci.*, Jan. 2014, doi: 10.5901/mjss.2014.v5n2p47.

[2] D. Bartram and R. K. Hambleton, Eds., *Computer-Based Testing and the Internet*. West Sussex, England: John Wiley & Sons, Ltd, 2005.

[3] S. O. Kuyoro, G. U. Maminor, R. U. Kanu, and O. Akande, “‘The Design and Implementation of a Computer Based Testing System Using Component-Based Software Engineering Michael,’” *J. Appl. Comput.*, vol. 01, no. 01, pp. 1–7, 2016, [Online]. Available: http://www.ijcst.com/vol8/2/13-michael-ajinaja.pdf.

[4] A. T. Alabi, A. O. Issa, and R. A. Oyekunle, “The Use of Computer Based Testing Method for the Conduct of Examinations at the University of Ilorin,” *Int. J. Learn. Dev.*, vol. 2, no. 3, p. 68, May 2012, doi: 10.5296/ijld.v2i3.1775.

[5] H. A. Zubairu, I. O. Oyefolahan, S. O. Etuk, and F. J. Babakano, “A framework for semantic driven electronic examination system for subjective questions,” *Niger. J. Technol.*, vol. 37, no. 1, p. 200, Jan. 2018, doi: 10.4314/njt.v37i1.26.

[6] D. Susanto, S. Irdoni, and M. U. H. Al Rasyid, “Attendance report plugin for E-learning applications in PENS: (Based on moodle),” in *2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, Sep. 2017, pp. 153–160, doi: 10.1109/KCIC.2017.8228579.

[7] “Advantages & disadvantages of CodeIgniter Framework.” https://www.technootech.com/advantages-disadvantages-of-codeigniter-framework/ (accessed Jun. 24, 2021).

[8] R. M. Luecht and S. G. Sireci, *A review of models for CBT*. 2011.

[9] S. Ejim, “An Overview of Computer Based Test,” *Res. Publ.*, no. April, pp. 1–4, 2018, doi: 10.13140/RG.2.2.32040.88326.

[10] “Data Modeling for MVC Applications - Archived Tutorial - Dojo Toolkit.” https://dojotoolkit.org/documentation/tutorials/1.6/data\_modeling/ (accessed Jun. 24, 2021).

[11] “MVC Framework - Controllers - Tutorialspoint.” https://www.tutorialspoint.com/mvc\_framework/mvc\_framework\_controllers.htm (accessed Jun. 24, 2021).

[12] “What are the Software Development Models?” http://tryqa.com/what-are-the-software-development-models/ (accessed Aug. 01, 2021).

[13] “Software Development Life Cycle Models and Methodologies - Mohamed Sami.” https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/ (accessed Aug. 01, 2021).

[14] David Upton, “Codeigniter for Rapid Php Application Development - David Upton - Google Books.” https://books.google.com.my/books?hl=en&lr=&id=RwEBcaYY-i8C&oi=fnd&pg=PT10&dq=what+is+codeigniter&ots=wHnharSuGm&sig=regCGQGoY7RKirnVUYqjT12USbo&redir\_esc=y#v=onepage&q=what is codeigniter&f=false (accessed Jun. 29, 2021).

[15] “File Structure in CodeIgniter - javatpoint.” https://www.javatpoint.com/file-structure-in-codeigniter (accessed Jun. 29, 2021).

[16] codeigniter.com, “Application Flow Chart — CodeIgniter 3.1.11 documentation,” 2020. https://codeigniter.com/userguide3/overview/appflow.html (accessed Jun. 29, 2021).

[17] M. Ozkaya and F. Erata, “A survey on the practical use of UML for different software architecture viewpoints,” *Inf. Softw. Technol.*, vol. 121, p. 106275, May 2020, doi: 10.1016/j.infsof.2020.106275.

[18] “Usage Statistics and Market Share of Server-side Programming Languages for Websites, July 2021.” https://w3techs.com/technologies/overview/programming\_language (accessed Jul. 31, 2021).

[19] J. Dahse and T. Holz, “Simulation of Built-in PHP Features for Precise Static Code Analysis,” 2014, Accessed: Jul. 31, 2021. [Online]. Available: http://dx.doi.org/.

[20] A. S. Christensen, A. Møller, and M. I. Schwartzbach, “Precise Analysis of String Expressions,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2694, pp. 1–18, 2003, doi: 10.1007/3-540-44898-5\_1.

[21] J. Letkowski, “Doing database design with MySQL,” *J. Technol. Res.*, vol. 6, no. I, pp. 1–15, 2014, [Online]. Available: https://pdfs.semanticscholar.org/6dab/89a3e32c994cff67d0c0f40c69ef9446db2c.pdf%0A.

[22] O. A. Abass, S. A. Olajide, and B. O. Samuel, “Development of web-based examination system using open source programming model,” *Turkish Online J. Distance Educ.*, vol. 18, no. 2, pp. 30–42, 2017, doi: 10.17718/tojde.306555.

[23] S. Walia and S. Gill, “A Framework for Web Based Student Record Management System using PHP,” *Int. J. Comput. Sci. Mob. Comput. ISSN 2320–088X*, vol. 3, no. 8, pp. 24–33, 2014, [Online]. Available: https://pdfs.semanticscholar.org/f5e1/b5d218431f89d5ecb05e7d2e23892cfaf042.pdf?\_ga=2.4884785.364962908.1574432524-89530086.1569769324.

[24] “8 Big Advantages of Using MySQL | Datamation.” https://www.datamation.com/storage/8-major-advantages-of-using-mysql/ (accessed Jun. 29, 2021).

[25] C. Vargas-Salgado, J. Aguila-Leon, C. Chiñas-Palacios, and E. Hurtado-Perez, “Low-cost web-based Supervisory Control and Data Acquisition system for a microgrid testbed: A case study in design and implementation for academic and research applications,” *Heliyon*, vol. 5, no. 9, p. e02474, Sep. 2019, doi: 10.1016/j.heliyon.2019.e02474.

[26] M. W. Alawar and S. S. Abu-Naser, “CSS-Tutor: An intelligent tutoring system for CSS and HTML,” *Int. J. Acad. Res. Dev.*, vol. 2, no. 1, pp. 94–99, 2017, [Online]. Available: http://dspace.alazhar.edu.ps/xmlui/handle/123456789/216.

[27] “Top 10 Benefits of Using Bootstrap!” https://www.xeliumtech.com/blog/Top-10-Benefits-of-Using-Bootstrap (accessed Jun. 30, 2021).

[28] J. F. Brunelle, M. Kelly, M. C. Weigle, and M. L. Nelson, “The impact of JavaScript on archivability,” *Int. J. Digit. Libr.*, vol. 17, no. 2, pp. 95–117, Jun. 2016, doi: 10.1007/s00799-015-0140-8.

[29] R. G. Jimoh, K. S. Abduljaleel, and Y. K. Kawu, “Students ’ Perception of Computer Based Test ( CBT ) for Examining Undergraduate Chemistry Courses,” *J. Emerg. Trends Comput. Inf. Sci.*, vol. 3, no. 2, pp. 125–134, 2012.

[30] O. T. Oluwatosin and D. D. Samson, “Computer-Based Test: Security and Result Integrity,” *Int. J. Comput. Inf. Technol.*, vol. 02, no. 02, pp. 2279–764, 2013, [Online]. Available: www.ijcit.com.

[31] S. Stoyanov, I. Ganchev, I. Popchev, and M. O�droma, “From CBT to e-Learning,” *J. Inf. Technol. Control*, vol. 3, no. 4, pp. 2–10, 2005.

[32] K. L. Huff and S. G. Sireci, “Validity Issues in Computer-Based Testing,” *Educ. Meas. Issues Pract.*, vol. 20, no. 3, pp. 16–25, Oct. 2005, doi: 10.1111/j.1745-3992.2001.tb00066.x.

[33] S. Zakrzewski and J. Bull, “The Mass Implementation and Evaluation of Computer‐based Assessments,” *Assess. Eval. High. Educ.*, vol. 23, no. 2, pp. 141–152, Jan. 1998, doi: 10.1080/0260293980230203.

[34] E. Moe, “Introducing large-scale computerised assessment. Lessons learned and future challenges.,” *Transit. to Comput. assessment. New approaches to Ski. Assess. Implic. large-scale testing.*, pp. 51–56, 2009.

[35] R. Clariana and P. Wallace, “Paper–based versus computer–based assessment: key factors associated with the test mode effect,” *Br. J. Educ. Technol.*, vol. 33, no. 5, pp. 593–602, Nov. 2002, doi: 10.1111/1467-8535.00294.

[36] C. K. Ayo, I. O. Akinyemi, A. A. Adebiyi, and U. O. Ekong, “The prospects of E-examination implementation in Nigeria,” *Turkish Online J. Distance Educ.*, vol. 8, no. 4, pp. 125–134, 2007, doi: 10.17718/tojde.49840.

[37] S. Ejim, “Computer based examination system with multi-factor authentication and message notification features,” no. February, pp. 1–58, 2017, doi: 10.13140/RG.2.2.14713.88167.

[38] K. Langer and F. Bodendorf, “A system architecture for flexible, knowledge-based, multimedia CBT-applications,” in *Proceedings IEEE 1st International Conference on Multi Media Engineering Education*, pp. 20–29, doi: 10.1109/MMEE.1994.383195.

[39] M. Thakir Mahmood, O. Ibrahem Ashour, and O. Bayat, “International Journal of Computer Science and Mobile Computing Design and Implementation of Web Based For Intermediate Online Shop with Laravel Framework,” *Int. J. Comp. Sci. Mob. Comput.*, vol. 8, no. 3, pp. 124–133, 2019, [Online]. Available: www.ijcsmc.com.

[40] A. Hidayat and V. Gayuh Utomo, “Implementing Code Igniter Framework in Open Source Mobile Learning Application,” *Int. J. Comput. Appl.*, vol. 108, no. 18, pp. 9–14, 2014, doi: 10.5120/19010-0459.

[41] S. C. Wibawa *et al.*, “Online test application development using framework CodeIgniter,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 296, p. 012041, Jan. 2018, doi: 10.1088/1757-899X/296/1/012041.

[42] H. Hustinawati, A. Kurnia Himawan, and L. Latifah, “Performance Analysis Framework Codeigniter and CakePHP in Website Creation,” *Int. J. Comput. Appl.*, vol. 94, no. 20, pp. 6–11, 2014, doi: 10.5120/16549-5946.

[43] B. Sujatmiko, S. C. Wibawa, D. A. Dermawan, and H. A. Ongge, “Computer Based Test (CBT) using Moodle-based High Order Thinking Skills in network design subjects,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1098, no. 5, p. 052112, Mar. 2021, doi: 10.1088/1757-899X/1098/5/052112.

[44] D. R. Kaparang, A. Mewengkang, C. P. Munaiseche, and I. D. Pesik, “Object Data Modelling for Online Test Systems Based on Indicator of Competency Achievement Approach,” 2019, doi: 10.2991/ictvet-18.2019.105.

[45] Q. JIA, “Design of a Student’s Online Examination System Based on Web,” *DEStech Trans. Soc. Sci. Educ. Hum. Sci.*, no. emass, Mar. 2017, doi: 10.12783/dtssehs/emass2016/6761.

[46] J. L. Harrington, *Relational Database Design and Implementation*, Fourth Edi. 2009.

[47] P. Krutchen, “The ‘4 + 1’ View of Software Architecture,” *IEEE Softw.*, vol. 12, no. 6, pp. 42–50, 1995.

[48] E. Söderström, B. Andersson, P. Johannesson, E. Perjons, and B. Wangler, “Towards a Framework for Comparing Process Modelling Languages,” 2002, pp. 600–611.

# **APPENDIX**

**<?php**

**class Examinations extends CI\_Controller**

**{**

**public function index()**

**{**

**//Check Login**

**if (!$this->session->userdata('logged\_in')) {**

**redirect('users/login');**

**}**

**$this->load->model('Examination\_model', 'examination\_model');**

**$data['title'] = 'Examinations';**

**$data['examinations'] = $this->examination\_model->get\_examinations();**

**$this->load->view('templates/header');**

**$this->load->view('examinations/index', $data);**

**$this->load->view('templates/footer');**

**}**

**public function create\_exam($name = null, $id = 0)**

**{**

**// Check Login**

**if (!$this->session->userdata('logged\_in')) {**

**redirect('user/login');**

**}**

**$this->load->model('User\_model', 'user\_model');**

**$this->load->model('Examination\_model', 'examination\_model');**

**$asdata = array('udata' => $this->user\_model->getById($id));**

**$data['udata'] = $this->user\_model->getById($id);**

**// print\_r($data);**

**$data['title'] = 'Create Exam';**

**$this->form\_validation->set\_rules('course', 'Course', 'required|min\_length[4]|is\_unique[examinations.course]');**

**$this->form\_validation->set\_rules('test\_name', 'TestName', 'required');**

**$this->form\_validation->set\_rules('date', 'Date', 'required');**

**$this->form\_validation->set\_rules('duration', 'Duration', 'required');**

**$this->form\_validation->set\_rules('instruction', 'Instruction', 'required');**

**$this->form\_validation->set\_rules('password', 'Password', 'required');**

**if ($this->form\_validation->run() == FALSE) {**

**$this->load->view('templates/lecturer\_createExam\_navbar', $asdata);**

**$this->load->view('examinations/create\_exam', $data);**

**$this->load->view('templates/user\_navbars\_footer');**

**} else {**

**$this->examination\_model->create\_exam($id);**

**$this->session->set\_flashdata('exam\_created', 'Exam has been successfully registerd');**

**redirect('examinations/viewExamsCreated/' . $name);**

**}**

**}**

**public function edit\_exam($id = 0)**

**{**

**if (!$this->session->userdata('logged\_in')) {**

**redirect('user/login');**

**}**

**// else if ($id == 0 || $id != $this->session->userdata('usernames')) {**

**// redirect('user/login');**

**// }**

**$data['title'] = 'Edit Exam';**

**$this->load->model('User\_model', 'user\_model');**

**$this->load->model('Examination\_model', 'examination\_model');**

**$data['post'] = $this->examination\_model->getexambyId($id);**

**$sname = $this->examination\_model->getexambyId($id);**

**$sdata = array('udata' => $this->examination\_model->getbyId($sname['user\_id']));**

**$this->form\_validation->set\_rules('course', 'Course', 'required|min\_length[4]|is\_unique[examinations.course]');**

**// $this->form\_validation->set\_rules('date', 'Date', 'required');**

**$this->form\_validation->set\_rules('duration', 'Duration', 'required');**

**$this->form\_validation->set\_rules('instruction', 'Instruction', 'required');**

**$this->form\_validation->set\_rules('password', 'Password', 'required');**

**if ($this->form\_validation->run() == FALSE) {**

**$this->load->view('templates/lecturer\_createExam\_navbar', $sdata);**

**$this->load->view('examinations/edit\_exam', $data);**

**$this->load->view('templates/user\_navbars\_footer');**

**} else {**

**$this->load->model('Examination\_model', 'examination\_model');**

**$this->examination\_model->update(**

**['examid' => $id],**

**[**

**'instruction' => $this->input->post('instruction'),**

**'course' => $this->input->post('course'),**

**'duration' => $this->input->post('duration'),**

**'password' => $this->input->post('password')**

**]**

**);**

**}**

**}**

**public function delete\_exam($exam\_id)**

**{**

**if (!$this->session->userdata('logged\_in')) {**

**redirect('user/login');**

**}**

**$data = array();**

**// $name = $this->usermodel->($exam\_id);**

**$this->load->model('User\_model', 'user\_model');**

**$data = $this->user\_model->getByExamID($exam\_id);**

**$this->load->model('Examination\_model', 'examination\_model');**

**$this->load->model('Question\_model', 'question\_model');**

**$data = $this->examination\_model->delete(['exam\_id' => $exam\_id]);**

**$query = $this->question\_model->delete(['question\_id' => $exam\_id]);**

**if ($query && $data) {**

**redirect('examinations/viewExamsCreated' . $exam\_id);**

**}**

**}**