DESIGN AND IMPLEMENTATION OF AN ONLINE TESTING SYSTEM FOR LAUTECH MIT STUDENTS

BY

ONUORAH OSONDU BENJAMIN
(EAA1502734)

EAA1502734

A PROJECT REPORT SUBMITTED
TO

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY
LADOKE AKINTOLA UNIVERSITY OF TECHNOLOGY,
OGBOMOSO, NIGERIA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER IN INFORMATION TECHNOLOGY (MIT) DEGREE

MARCH, 2018

CERTIFICATION

This is to certify that this project titled: DESIGN AND IMPLEMENTATION OF AN ONLINE TESTING SYSTEM FOR LAUTECH MIT STUDENTS was carried out by ONUORAH, OSONDU BENJAMIN with matriculation number EAA1502734 of the department of Computer Science and Engineering, Faculty of Engineering and Technology, Ladoke Akintola University of Technology, Ogbomoso, Oyo state, Nigeria, and supervised accordingly.

_____ _____

Engr, Dr. A.O. Oke
Supervisor

Date

_____ _____

Dr. A.O. Afolabi

Date

Head of Department

DEDICATION

This project is dedicated to Almighty God, my wife Blessing Ebele Onuorah and the entire members of the Onuorah's family.

ACKNOWLEDGEMENTS

My gratitude goes to Almighty God for his protection and guidance during the course of this project. My gratitude also goes to my supervisor Engr. Dr. Mrs. A. O. Oke for her advice, attention and openness and experience shared during the course of this program and project; may God bless you Ma. My thanks also goes to the Head of department, Dr. A.O. Afolabi and also the entire lecturers of Computer Science and Engineering Department.

My warmest appreciation goes to my caring and loving wife Blessing Ebele Onuorah for her moral and emotional support; may God bless you.

ABSTRACT

Technology has supported and enhanced online testing process successfully over the years. Online Testing System is a web-based software application, which allows an organization or institutions to organize, conduct and manage test via an online platform. It can be done through the Internet or Local Area Network. Some of the challenges faced during manual testing are the delays occurred in result processing and difficulty in filtering of records. The chance of loss of records is high also record searching is difficult. Online Testing is one of the crucial parts for online education system. It is efficient, fast enough and reduces the large amount of material resource. This project describes the principle of the system, presents the main functions of the system, analyzes the auto-generating test algorithm and discusses the security of the system with Ladoke Akintola University of Technology, Masters of Information Technology program as a cast-study. The project proposed a Web-based Online Testing System, to handle testing operations such as test questions and answers entry, student authentication, test attempting and result processing. The two main research sub-domains investigated during the project are Human-Computer Interaction (HCI) and Software Engineering (SE). The project demonstrated SE methodologies from the initial requirement gathering phase to the software testing and validation phase. Some notable practices include establishing software architecture that could promote separation of concern and reusability, designing essential data structures and algorithms for test question randomization, presentation and result processing, and adopting web technology for real-time communication.

TABLE OF CONTENTS

CHAPTER THREE: METHODOLOGY

CHAPTER FOUR: DEVELOPMENT AND IMPLEMENTATION

CHAPTER FIVE: CONCLUSION AND RECOMMENDATION

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background of Studies

Education is about teaching, learning skills and knowledge. Education also means helping people to learn how to do things and encouraging them to think about what they learn. It is also important for educators to teach ways to find and use information. Through education, the knowledge of society, country, and of the world is passed on from generation to generation. In democracies, through education, children and adults are supposed to learn how to be active and effective citizens. Formal education is usually in school, where a person may learn basic, academic, or trade skills. Small children often attend a nursery or kindergarten but often formal education begins in elementary school and continues with secondary school. Post-secondary education (or higher education) is usually at a polytechnic, college or university which may grant an academic diploma or degree, often tests and examinations is administer in formal school to grade and qualify the learner to proceed to next academic stage unit graduation. Written tests are tests that are administered on paper or on a computer (as an eExam). A test taker who takes a written test could respond to specific items by writing or typing within a given space of the test or on a separate form or document.

A test developer's choice of which style or format to use when developing a written test is usually arbitrary given that there is no single invariant standard for testing. Be that as it may, certain test styles and format have become more widely used than others. Multiple choice tests, essay test are the two common style or format administer in the universities and can be conducted user paper and pen or computer-based tests (online testing system) Online Testing

System is a software solution or application, which allows any industry or institute to arrange, conduct and manage test via an online environment. It can be done through the Internet/Intranet and/ Local Area Network environments. The organization or Institution is in charge of the creating or setting up the test questions and answers. Online testing system gives remotely access to learner or student. It assists the organization or Institution with reducing the work of leading test, checking answer sheets and producing result. All these work is finished by the machine. All the information is put away on the server.

1.2 Problem Statement

The MIT students of LAUTECH had have little or no assessment or test prior to the semester examination, test are conducted by higher institution of learning to prepare student for the final examination, however because most of the MIT Programme's students at LAUTECH are fully employed and reside off the host state, conducting a physical test (either scored or not) may not be very feasible, this problem is what this project aim to solve by designing and implementing an Online Testing web-based Solution that can allow the Student to login and practice some test prior to the examination date from the comfort of their homes or offices. This approach of supporting Distance Education can be as effective as traditional education when the methods and technologies are used appropriately. With the introduction of technologies for learners and teachers, most universities have introduced distance learning/distance education to meet the need of part-time and distance learning student.

1.3 Aim and Objectives

1.3.1 Aim

This project aims at designing and implementing an Online Testing System, for LAUTECH MIT students so as to give them an opportunity to practice and prepare for their examinations.

1.3.2 Objectives

The objective of this project is to:

- Design a web-based online testing system
- Implement a web-based online testing system that allows the learner to attempt or take test online remotely.
- Evaluate the system performance

1.4 Significance of Study

Online Testing System is one of the distance learning tools that assists universities today to carry out assessment or test their students or prepare them for examination, the Online Testing System is much faster in collecting and computing test results, it also saves time and cost for both distance learning student from coming to the universities to take test, as well as the university in terms of logistics such as print of question papers, classroom space and personnel to supervise, mark and collect the test result.

1.5 Scope of Study

The focus of this research is to design and implement an online testing system using standard and open source web technology, framework and programming language for the Master in Information Technology students of Ladoke Akintola University of Technology, Ogbomoso, Oyo state, Nigeria.

1.6 Definition of Terms

- **Apache Web Server:** It is a public domain and the world most popular open source web server, as web server allows a user to access (serve) a web page via a network either locally or on the internet
- **Cascading Style Sheets** (**CSS**): It is a style sheet language used for describing the look and formatting of a document written in a markup language.
- **Database Administrator (DBA):** This is someone (IT professional) who has central control of both the data and the programs that access those data.
- **Database Management System (DBMS):** This is a collection of interrelated data and a set of programs to access those data.
- **Database:** This is simply a collection of related data or record, an a logical or well organized manner
- **Database:** The collection of data containing information relevant to an enterprise.
- **HyperText Markup Language (HTML):** This defines how messages are formatted and transmitted, and what the Web servers and browsers shall do in response to various commands. This is a core technology markup language of the Internet used for structuring and presenting content for the World Wide Web.

- **LAUTECH:** Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria
- **MIT:** Master in Information Technology degree
- **MySQL:** (as of March 2014) the world's second most widely used open-source relational database management system (RDBMS).
- **PHP:** A server-side scripting language designed for web development but also used as a general-purpose programming language. While PHP originally stood for *Personal Home Page*, it now stands for *PHP: Hypertext Preprocessor*, which is a recursive acronym.
- **Query:** This is a statement requesting the retrieval of information.
- **Relational Database Management System (RDBMS):** A database product, like those from Oracle or Sybase that stores its data in tables of rows and columns. Usually speaks SQL.
- **Structured Query Language (SQL):** A special-purpose programming language designed for managing data held in a relational database management system (RDBMS).
- **Web Application:** A web Application is a collection of web pages that interact with the user, with each other and with various resources on a web server, including the databases.
- **Web Browser:** This is a software makes requests from the web server and the server sends back a response.
- **Web Server:** A web server is a software program (middleware) that serves web pages in response to request from web browsers.
- **XP:** extreme programming
- **OO:** Object Oriented

CHAPTER TWO

LITERATURE REVIEW

2.1 Test

A test or examination (informally, exam or evaluation) is an assessment intended to measure a test-taker's knowledge, skill, aptitude, physical fitness, or classification in many other topics, for example beliefs). A test may be administered verbally, on paper, on a computer, or in a confined area that requires a test taker to physically perform a set of skills. Tests vary in style, rigor and requirements. For example, in a closed book test, a test taker is often required to rely upon memory to respond to specific items whereas in an open book test, a test taker may use one or more supplementary tools such as a reference book or calculator when responding to an item. A test may be administered formally or informally. An example of an informal test would be a reading test administered by a parent to a child. An example of a formal test would be a final examination administered by a teacher in a classroom or an I.Q. test administered by a psychologist in a clinic. Formal testing often results in a grade or a test score. A test score may be interpreted with regards to a norm or criterion, or occasionally both. The norm may be established independently, or by statistical analysis of a large number of participants. A standardized test is any test that is administered and scored in a consistent manner to ensure legal defensibility. Standardized tests are often used in education, professional certification, psychology, the military, and many other fields.

Ancient China was the first country in the world that implemented a nationwide standardized test, which was called the imperial examination. The main purpose of this examination was to select able candidates for specific governmental positions. The imperial

examination was established by the Sui dynasty in 605 AD and was later abolished by the Qing dynasty 1300 years later in 1905. England had adopted this examination system in 1806 to select specific candidates for positions in Her Majesty's Civil Service, modeled on the Chinese imperial examination. This examination system was later applied to education and it started to influence other parts of the world as it became a prominent standard of delivering standardized tests.

2.2 Online Testing System

Online Testing System is often a web based software solution or application, which allows any industry or institute to arrange, conduct and manage test via an online environment. It can be done through the Internet/Intranet and/ Local Area Network environments. Some of the problems faced during manual testing systems are the delays occurred in result processing, filing poses a problem, filtering of records is difficult. The chance of loss of records is high also record searching is difficult. Maintenance of the system is also very difficult and takes lot of time and effort. Online examination is one of the crucial parts for online education system. It is efficient, fast enough and reduces the large amount of material resource. A testing system is developed based on the web.

2.3 Web Application

World Wide Web (WWW) application, or *web application*, is any software application that is executed on the web. Originally, the web functioning as an information medium and most of its content remained static. *Web application* evolved though, from statistic textual content with limited interactivity, to rich interfaces with dynamic content and responsive interaction, known as *Web 2.0*. The role of the web has transformed from simple information publication to

distributed enterprise-scale workflow systems. *Web application* has proven that web technology could help in software development. Three basic elements of *WWW* that are found useful to software application development are highlighted by as the following:

- *Uniform Resource Locators* (URLs), is a naming scheme to identify computer location, the requested resource in the file system and a protocol to communicate with the resources. The requested resource is not limited to file document, instead, developers may also use an URL to access a particular software service. Modern software applications enable communication across server boundaries by pin pointing the remote resources with the URL.
- *Hypertext Markup Language* (HTML), refers to a formatted document containing content, styling and structure to present the content of web pages [20]. The recent evolution to *HTML5* 1promised even greater flexibility in visual presentation and responsive interaction. The potential of HTML is further enhanced by a combination of client side scripting (e.g. JavaScript) and dynamic content generation with server based programming language (e.g. PHP). In software development, designers could work on HTML presentation while back-end developers could focus on server side algorithms.
- *Hypertext Transfer Protocol* (HTTP), is a main communication protocol of the Web [20]. The protocol formulates the request data operation in GET method and sends data operation in POST operations. This allows modern *web application* to utilize a single URL for information request and data manipulation by specifying the desired operation in a request packet. The widely adopted protocol improves inter-operability of software application in fetching dynamic content. There are also increasing attempts to embed a web browser within a native software application to present information.

Web technology emerges as one of the popular choices to develop an *online testing system*. One of the major advantages of *web application* is its accessibility. This connects to Gellersen & Gaedke's statement,

"Applications that use HTML-based front ends benefit from the pervasive distribution of Web browsers for universal, cross-platform access,"

In other words, *web application* could allow users to access remote resources (e.g. data and services) that are distributed across an enterprise network or the internet. Besides, the nature of ubiquitous clients and centralized maintenance in *web application* has also enabled instantaneous deployment of software updates at minimal cost. Compared to the conventional patch and update approach, updating *web applications* could happen in real-time. The update also would have minimum impact on the client side as most changes are processed on server side. Both specified advantages make web technology an excellent candidate for inclusion in Online Testing System – in which remote data access and information sharing, is crucial to operations efficiency.

2.4 Review of Existing Work

Delaware is one of a handful of states that has moved all of its testing online. Some education reformers and technology experts are hailing the move, which has the backing of the Obama administration, as a revolution. They are promising more well-rounded tests, less frequent cheating and immediate feedback for both students and teachers as students' answers are transmitted quickly over the Internet to states and the results are then sent back to districts. But other educators and experts point to a host of potential problems. Shrinking school budgets

could make it difficult for districts to purchase new equipment, and states that pioneered online tests have dealt with network meltdowns. Some worry that the move to online testing could take time away from learning.

Wyoming switched from paper-and-pencil to online tests in 2010, but technical problems popped up everywhere. Online testing was such a debacle that voters threw the state superintendent out of office and the state sued NCS Pearson, the company hired to design and administers the test. The state went back to old-fashioned paper exams. In Virginia, the switch to online tests went more smoothly. Over a decade, Virginia expanded online testing incrementally, starting in high school and moving down to earlier grades. The state also invested nearly $650 million in new technology. But despite its careful rollout, in 2007, nearly 10,000 students were unable to complete online exams — administered by Pearson Educational Management — after a series of technical glitches. Bryan Bleil, Pearson's vice president for online and technology implementation, says the company is working with states and districts to help them make the transition to computer-based testing — ensuring they have enough Internet bandwidth, for example, to handle the online traffic during testing times. The shift to computer-based testing also corresponds with a push to make students digitally literate. And instantaneous scoring by computers will allow teachers, students and parents to see test results right away, rather than having to wait weeks or months after the school year has ended. Don Davis, principal of Brick Mill Elementary, in Delaware's Appoquinimink district, has mixed feelings about the tests, including whether they might widen the achievement gap for low-income students who don't have computers at home. But, he said, "It's better than what we used to have."

CHAPTER THREE

METHODOLOGY

3.1 Software Development Methodology

Agile Development is the software development methodology used for the Online Testing System. Design process in traditional software development follows a series of planned design activities or steps, or phases which produces design models that become guidelines to the developers – for the implementation phase. However, this can conflict with agile methods, as if considering an interesting fact:

"…agile practitioners believe that design is not only highly iterative, but emergent, and models often lie. Thus, only coding, running tests, and refactoring the code reveal the truth about a design,"

In order to understand this conflict, two styles of design in software development will be explained, namely: i) planned design; and ii) evolutionary design.

Evolutionary design means the design for a particular system grows as the system is being developed. However, evolutionary design is a disaster in common usages because:

- Aggregate of ad-hoc tactical decisions lead to code base that hard to change;
- It leads to poor design when ability to make changes deteriorates; and
- Bugs become exponentially expensive to fix.

In contrast, **planned design** is closer to the engineering metaphor – where designer produces blueprint that consists of fundamental rules and structures to build the software. This design style had been applied widely in early software development, yet it has several drawbacks:

Impossible to think through all the issues and new questions that emerge during programming; Technology changes rapidly and initial design concept may not match with the latest tools and materials; and It deals poorly with unforeseen changes of requirements. The conflict of planned design and evolutionary design is widely known as “code is my design” versus “big design upfront”. Nevertheless, both design styles may not be ideal to software development due to their problems mentioned above. This was until the emerge of Agile methods such as XP – that enabled evolutionary design by adopting set of principles to address changes and effects of changes in design. It pointed out that the core of enabling principles of XP is testing and continuous integration. Testing ensures that design decisions and changes are safety verified while continuous integration is essential to keep team in sync and not worry about new changes will break existing system. Design process in the online testing system attempts to embrace changes with agile methods while encourage important design decisions that could be done with planned modeling. It needs to be aligned to several principles of Agile Modeling. They are described as following:

- **Model with a purpose.** It is important to know the purposes and for which stakeholder a particular model was intended when creating a model. The model is intended to keep the developer focus on solving particular problems rather than figuring out whether it is sufficiently detailed and accurate.
- **Incremental changes.** Models do not need and are unlikely to meet all needs when they are first implemented. Modelling should start from high-level details and evolve over time in an incremental manner. The model should contain just enough details to solve current goals and may require refactoring.

- **Assume simplicity.** Modelling should assume the simplest solutions are the best solutions and hence avoid unnecessary efforts spent on features that users do not need or optimizations that are not necessary – or focus on issues that may take long to materialize and are not essential. This is highlighted in the quotations: "Do the Simplest Thing that Could Possibly Work" and "You are not Going to Need It" which are manifestation of Simple Design principle in XP.
- **Working software is your primary goal.** The main objective in software development is to produce high quality software. Every documentation, artifact or model that does not directly contribute to this goal should be questioned and avoided if it cannot be justified.

3.2 Architecture Design

The design process in the Online testing system started with an initial architecture design. The architecture of software system is described by TOGAF as:

> *"1) A formal description of a system, or a detailed plan of the system at component level to guide its implementation; and 2) The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time,"*

It acts as the important entry point to the design process in software development. This can be described as "architecture envisioning" and it is considered particularly important in scaling software development as it gradually become large and complex. As the project progresses through the various decision-makings, it needs to be well-established architecture that acts as a baseline to such activities. Thus, it is concern about the evolution of the system as specified by the second meaning in TOGAF.

Architecture design is also connected to design goal of achieving separation of concern by the decomposition of functional elements into different subgroups. The mechanism of decomposition (or division) will affect every functional module including the structural definition and their interaction. Thus, the architecture should be planned carefully to ensure that the system and its sub-modules align to the project objectives. This is again linked to the first meaning in the TOGAF definition. The next sections will explain two key design decisions in architecture design: software architecture and physical architecture (deployment).

3.2.1 Software Architecture

As one of the major concerns in software engineering, the term software architecture could be understood through various definitions. It has been defined as the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security and manageability. These definitions mean that the software architecture phase represents the important design decision phase required to build a stable and scalable system. This is crucial for a successful system because it helps to address the quality and risk factors aligned to the project. Moreover, design concerns such as selection of algorithm(s), business logics and data structures often overlap with architecture decisions. Thus, the vision of architecture needs to be established to provide direction of development style. This project is developing a web application; hence, the following sections will explore several software architecture styles for web application.

3.2.1.1 Client-Server Architecture

The most common architecture style for a distributed system such as web application is the conventional client-server architecture. It divides the system into server components that offer services and client components that provide user interface to consume the services through

connected networks. The server could serve request from multiple clients. This architecture style is also known as 2-tier architecture. Clients are often represented by range of applications with GUI to capture user inputs and transform them into requests to the server. The server contains data storage to collect, modify and distribute data. Client could be either a thin client if application processing logic is located at server side or a fat client if it is embedded with client application (see Figure 4.3). However, this architecture style lacks scalability because both the client and the server have limited resources. Reusing the application logic for different modules is also increasingly complex as systems expand due to their tight coupling between either data tier or presentation logic. In addition, it is difficult to maintain because it is hard to distribute new changes to system users.

3.2.1.2 N-tier and Layered Architectural Style

The N-tier (multi-tier) or most popular 3-tier architecture style addresses conventional 2-tier architecture issues by placing the application logic at an additional tier (see Figure 4.4). Layered architecture shares the common goal with multi-tier architecture style – separation of functionalities into segments to improve scalability and maintainability. N-tier architecture is often used together with layered architecture style and they addressing following concerns:

i) Presentation logic (user interface);

ii) Business and application logic (domain process); and

iii) Data accessing logic (database communication).

The presentation tier is at the top most level of the system and exposes visual representation to allow users to interact with the system. It gathers the user intention in the form of commands and inputs and forwards them to the business tier. The business tier is responsible for processing data between presentation and data tier. It contains important application logic

performs calculations, evaluation and makes logical decisions. The result of the logic execution will either become responses to the presentation tier request or forwarding to data tier for persistent. At the bottom, the data tier handles create, update, read, and delete (CRUD) operations for the data sources. It contains various information to form appropriate query for data retrieval. In an OO environment, it also in charge of handle mapping between information from data source to domain entity.

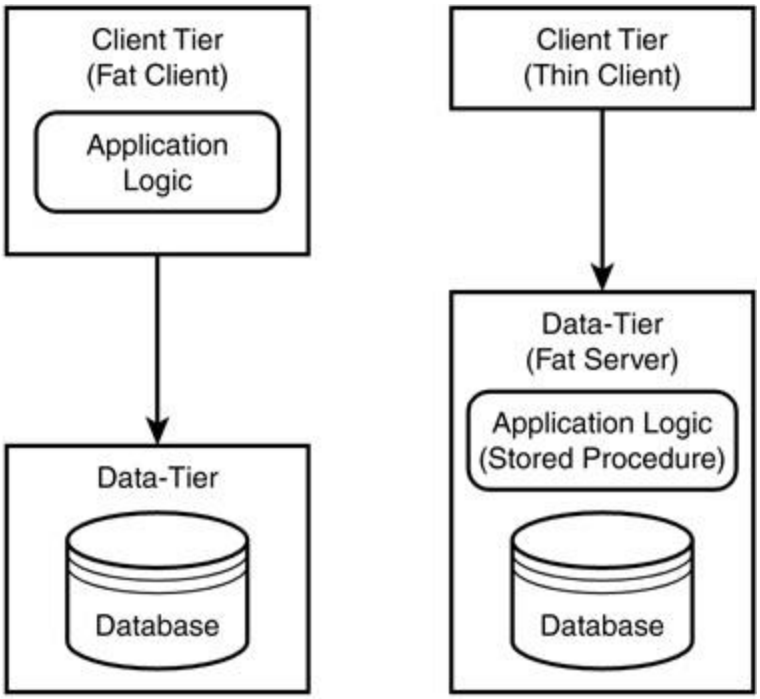


Figure 3.1: 2-Tier Architecture (Client/Server)

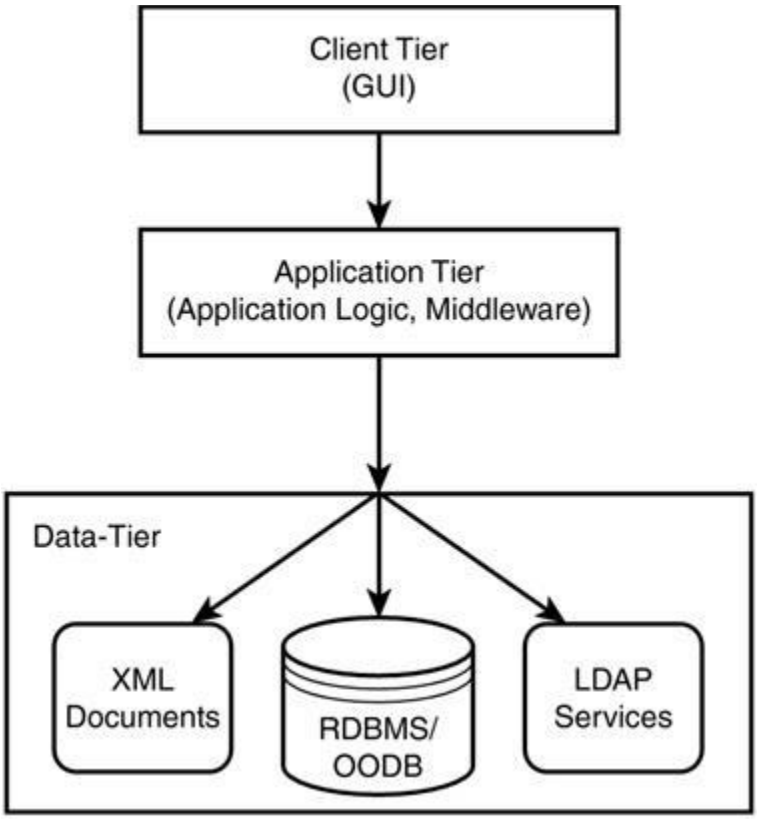


Figure 3.2: 3-Tier Architecture

The layered architecture style focuses on grouping related functionality into distinct layers and stacked vertically. Layered architecture model system structures in layers and each layer will have a collection of functions with similar concerns. Each layer can grow independently due to its loose coupling to upper and bottom layer. In theory, multi-tiered and layered architecture styles have a number of similarities. However, layered architecture has adopted OO design concepts into its core principles. Each layer promotes abstraction and encapsulation by providing just enough details to the dependent. It should also have high cohesion and clearly defined responsibility for maximized reuse. In contrast, multi-tiered architecture styles are concerned with the deployment of each segment into distributed environment. For this project implementation we adopted the 3-Tier Architecture.

3.3 Physical System Architecture

Once the software architecture is established, the next step is to consider a deployment scenario and the infrastructure of system. This explains the components of the system, their intercommunication, and the location of where they physically run. When deploying the system, most software packages are deployed into Business tier or the web server. However, the web browser of each client device will fetch some of HTML views and scripts when connected to the web server. This implies presentation components are partially deployed to Presentation tier. In addition to that, database schema in DAL will be used to construct the database instance in database server of Data tier.

3.4 System Analysis

System Analysis is a two-stage process. During the first stage, requirements determination, appropriate fact-finding techniques are used to determine detailed requirements of the proposed information system. The second stage is the requirements analysis, during which the information system's requirements are analyzed. The product of the analysis phase is the specification, an abstract, precise and high-level definition of the desired information system, the aims of the analysis are to produce a specification that is:

- suitable for communication between users and developers
- suitable for use as a basis for mapping through lower levels to an implemented system

Modern approaches to the analysis phase are found on the **conceptual modeling approach**. The approach distinguishes three levels of description of an information system: **external, conceptual and internal**. While the internal level is the level of logical and physical

design, the notions of external and conceptual levels are found in the specification which is the product of the analysis phase. For the purpose of the analysis phase, the conceptual level of description will be the focus which will model the organization precisely, on an abstract level and naturally. The conceptual model is on the conceptual level and it represents the part of the organization that will be modeled in the computer system. It consists of three components: structure, process and rule.

3.4.1 Structure Component

The structure component typically consists of entities, attributes and relationships, which will be specified diagrammatically using a method called **entity modeling**. The term "structure" is used, as these elements are the basis for other parts of the conceptual model. They are refined from the requirements model, so as to obscure its basic nature as a system overview, contains only the most important objects and properties.

3.4.2. Rule Component

Rules are restrictions in an organization which are modeled in terms of restrictions on entities, attributes and relationships in the structure component. It is a restriction on the systems states (expressed in terms of objects and properties) that may exist. Rules define allowed system states. At any given instant, a system is said to exist in a certain state. This state is defined or characterized by the relevant objects that exist in the system at that instant. Rules are specified non-procedurally, that is, declaratively, with the elements of the structure components to which they refer using a precise language usually based on logic or sets.

3.4.3. Process Component

Processes from the requirements specification will be refined into more detail, down to the level of primitive processes operating on elements of the structure component. All **events** will be specified. An event is an important occurrence that provides information or initiates a process. In contrast to process, which takes place during a time interval, an event is instantaneous. Events and processes are closely related to each other in a system and they jointly constitute the system's behavior. Process control structure will also be modeled, as well as the structure elements operated on. There are variety of forms for specifying processes, for example, **process decomposition diagram** for showing detailed process refinement and events and **data flow diagrams** for process sequence and entity life histories showing processes which operate on objects.

3.5 Requirements Determination

This is the preliminary investigation phase in systems analysis. It is used to determine and document what takes place in the current system with a view to making recommendations for the proposed system. Procedures and documents used and the people involved in various operations are analyzed as well as the transactions involved in such operations and the information generated within the system. The strengths of the current system as well as its weakness are analyzed. In addition to this, the strengths and demands of proposed systems are highlighted.

3.5.1 Analysis of Existing System

The analysis of the existing system (manual testing process) will describe the various strengths and weaknesses involved in the existing system.

3.5.2 Strengths of Existing System

- **Lower Cost of Implementation:** The existing system is relatively easy to implement using pen and paper, since it just involves skilled human resources only, that is, man-power. It does not involve any form of electronic gadgets like computers, networking equipment and wires.
- **Lower Maintenance Cost:** The existing system which is current manual administration of test is cheaper to maintain compared to the proposed system. Furthermore, no cost is incurred in the maintenance of any program or software and cost of maintaining uninterrupted power supply is totally avoided.
- **Less Need for Skilled Personnel in Information Technology:** The existing system can still perform optimally without the presence of personnel that are skilled in information technology or its variant. This saves the management from the hassles of acquiring skills in information technology know-how in form of in-house training.
- **Less Effect of Power Outage:** The existing system can still carry out its functions uninterrupted even during periods of epileptic power supplies.

3.5.3 Problems of the Existing System

- **Speed in Information Processing:** The existing system takes a longer time in information processing and retrieval due to human factor, such as fatigue from administration of test, scoring and grading most especially when the number of test participant is large.
- **Error-Prone:** The existing system is prone to human errors; marking of result, storing result and result sheets.

- **Security:** The existing system lacks security, in terms of disasters and confidentiality. Fire outbreaks can render information stored in files and cabinets useless. Also, intruders can gain access to records by just opening a chest of drawers, which breeches confidentiality.
- **Data Redundancy and Inconsistency:** The existing system encourages information to be duplicated in several places which leads to redundancy. Also, changes made on a particular item of data has to be made across all files, else data inconsistency sets in.
- **Difficulty in Accessing Data:** The existing system makes access to data very difficult. Volumes of papers accrued over the years can be difficult to sort or retrieve in a convenient and efficient manner.
- **Durability:** The system for keeping records in the existing system is mainly the use files and cabinets. This is prone to intrusion by rodents. This makes information not durable over a long period of time.
- **Result access by student:** The existing system will require the student to wait for days or even weeks before seeing the result of their test.

3.5.4 Analysis of the Proposed System

The process activities and component of the proposed system will be analyzed using use case model and its strengths and weaknesses will be discussed.

3.5.5 Use Case Model for the Proposed System

Uses cases are a software modeling technique that helps developers determine which features to implement and how to gracefully resolve errors. It defines a goal-oriented set of interactions between external actors and the system under consideration. Actors are parties outside the system that interact with the system. An actor may be a class of users, roles users can

play, or other systems. A use case describes the steps or actions between a user and a software system which lead the user towards something useful.

A Use Case Diagram has four major elements:

- The system described
- The actors that the system interacts with
- The use cases, or service, that the system knows how to perform
- The relationships between the above elements

3.5.5.1 System

As part of use-case modelling, the boundaries of the system developed must be defined. Defining the boundaries of the system is not trivial. The following questions are asked.

- Which tasks are automated and which are manual?
- Which tasks are performed by other systems?

The entire solution that we supply should be included in the system boundaries. A system in a Use Case Diagram is represented as a box. The name of the system appears above or inside the box.

3.5.5.2 Actors

A use case defines the interactions between external actors and the system under consideration to accomplish a goal. An actor specifies a role played by a person or thing when interacting with the system. The same person using the system may be represented as different actors because they are playing different roles. It is important to note that an actor and a user are not the same thing. A typical user may play a number of different roles when using a system, whereas an actor represents a class of external entities (often, but not always, people) that play just one role. There exist the primary actor and the secondary actor in a use case.

A ***primary* actor** is one having a goal requiring the assistance of the system while a ***secondary* actor** is one from which the system needs assistance. A use case is initiated by a user with a particular goal in mind, and completes successfully when that goal is satisfied. It describes the sequence of interactions between actors and the system necessary to deliver the service that satisfies the goal. It also includes possible variants of this sequence, e.g., alternative sequences that may also satisfy the goal, as well as sequences that may lead to failure to complete the service because of exceptional behavior, error handling, etc. The system is treated as a "black box", and the interactions with system, including system responses, are as perceived from outside the system. Thus, use cases capture *who* (actor) does *what* (interaction) with the system, for what *purpose* (goal), without dealing with system internals, that is it does not specify how that goal is achieved at the tail end. A complete set of use cases specifies all the different ways to use the system, and therefore defines all behavior required of the system, bounding the scope of the system. Generally, use case steps are written in an easy-to-understand structured narrative using the vocabulary of the domain. This is engaging for users who can easily follow and validate the use cases, and the accessibility encourages users to be actively involved in defining the requirements.

3.5.5.3 Use case scenario

A scenario is an instance of a use case, and represents a single path through the use case. Thus, one may construct a scenario for the main flow through the use case, and other scenarios for each possible variation of flow through the use case (e.g., triggered by options, error conditions, security breaches, etc.).

3.5.5.4 Use case focus

Each use case focuses on describing how to achieve a goal or a task. For most software projects, this means that multiple, perhaps dozens of use cases are needed to define the scope of the new system. The degree of formality of a particular software project and the stage of the project will influence the level of detail required in each use case. Use cases should not be confused with the system requirements. System requirements describe the functionality needed to meet a stakeholder request or user need. Each feature or requirement can be analyzed into one or more use cases, which detail cases where an actor uses the system. Each use case should be traceable to its originating feature, which in turn should be traceable to its originating stakeholder/user request. A use case should:

- Describe what the system shall do for the actor to achieve a particular goal.
- Include no implementation-specific language.
- Be at the appropriate level of detail.
- Not include detail regarding user interfaces and screens. This is done in user-interface design, which references the use case and its business rules.

3.5.5.5 Benefits of use cases

- It captures operational requirements from user's perspective.
- Gives a clear and consistent description of what system should do.
- A basis for performing system tests.
- Provides the ability to trace functional requirement into actual classes and operations in the system.
- Uncover and describe all tasks that need doing in a system of both human and system actors.

- To analyze what functionality that needs developing for the system.
- The use of use cases must mean that the right functional requirements are made of the IT system (the requirement of the business).

3.5.5.6 Use case: Include

In one form of interaction, a given use case may include another. Include is a Directed Relationship between two use cases, implying that the behavior of the included use case is inserted into the behavior of the including use case. The first use case often depends on the outcome of the included use case. This is useful for extracting truly common behaviors from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "«**include**»". This usage resembles a macro expansion where the included use case behavior is placed inline in the base use case behavior. There are no parameters or return values. To specify the location in a flow of events in which the base use case includes the behavior of another, you simply write ***include*** followed by the name of use case you want to include. The diagram below shows the use case for this web application:

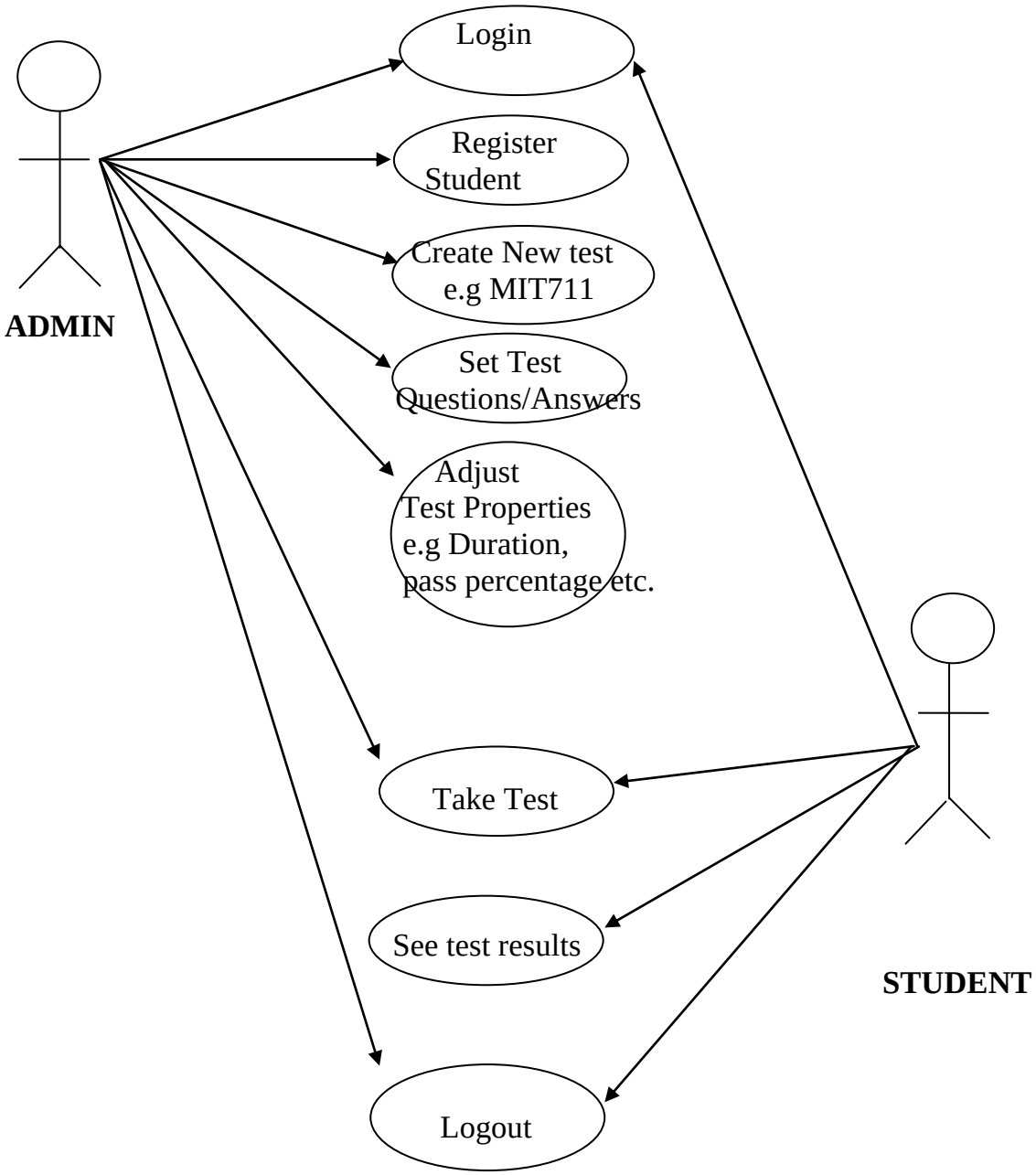


Figure 3.3: Use Case Diagram

3.5.6 Strengths of the Proposed System

- **Increased Throughput:** Throughput is the work done per unit time performance measurement. With the physical implementation of a well detailed logical design of the proposed system, information concerning the Online Testing processes can be easily retrieved from the database that stores data in relational database management system (RDBMS) which is a suite of software programs for creating, maintaining, modifying and manipulating a relational database.
- **Turnaround time is minimized:** Turnaround time is the time from which a job is submitted until the job is returned to the user. It is a performance measure used in Data processing. The proposed system will minimize turnaround time such as test submission time and result access time.
- **Improved storage technology:** The proposed system will make use of an enterprise database technology, MySQL Server, which can store 1,048,516 terabyte of data.
- **Easy accessibility to information:** With the proposed system, Learner can easily logon to the system to take test, rather than coming to the school campus, and access his/her immediately after completion of test.
- **Less Prone to Error:** The proposed system will be less prone to human and malicious errors. Since it is an automated system, errors in the computation of test records are highly minimized.
- **Security:** In contrast to the existing system that offers little, the proposed system will have a high level of security. Information will be stored in a central location (the database server) and this will prevent unauthorized access to information. Passwords will also be implemented to ensure that only authorized personnel gain access to information.

- **Software Scalability:** The proposed system will be scalable to meet future needs of the institution. As the student increases, the number of processes to be carried out by the system increases. The technologies adopted make it possible to increase the capabilities of the software as demand increases. New modules can be written to cater for new requirements in the system.

3.5.7 Demand of the Proposed System

Irrespective of the numerous benefits of the proposed system, it has its own demands and challenges. These challenges can be in form of what users are required to do to get the best from the system. The demands of the system being proposed include:

- **Computer / Internet or network Access:** as required of any Online web solution, the users of the proposed system will require a computer system as well as reliable internet access to use the Online Test Management system.
- **Maintenance:** Any work done to change the system after it is in operation is considered to be maintained. For the proposed system to always meet the present needs of its users, it has to be maintained. Maintenance cost has to be incurred from time to time, though it is minimal.
- **Training:** Competent personnel or admin will be needed to operate the proposed system once it is deployed. These may cost some training fee
- **Operational Expenses:** The smooth running of the proposed system is totally dependent on the availability of power supply. This implies that logistics have to be put in place to ensure that there is no power outage most especially during the test period. This can be in form of the purchase of stand-by generating units and uninterrupted power supply units.

3.6 System Design

Software design comprises a set of principles, concepts and practises to build high quality system. It is intended to form a solution by appropriate consideration of requirements and technical issues. Software design can be defined as:

> *"Software design is a systematic, intelligent process in which designers generate, evaluate, and specify concepts for a software system whose structure and function achieve clients' objectives or users' needs while satisfying a specified set of constraints,"*

Further, gives another view that describes the design space as focused on attaining the stakeholders' goals by adapting inner environments (means) to the outer environments (tasks). The outer environments refers to requirements, goals and need; while inner requirements is the set of software, languages, components and tools used to build software (see Figure 3.2). Software design may have broad spectrum of meanings and objectives based definitions above. Nevertheless, the primary goal of the design process in online testing system is to develop concepts and ideas that could answers our research questions while satisfying project requirements. There are four key considerations when performing design activities:

- Manage problem complexity through separation of concerns;
- Produce abstract representation of design decision;
- Provide unambiguous meaning to the concepts and terms used in the design models; and
- Establish guided paths to achieve specific end-user task.

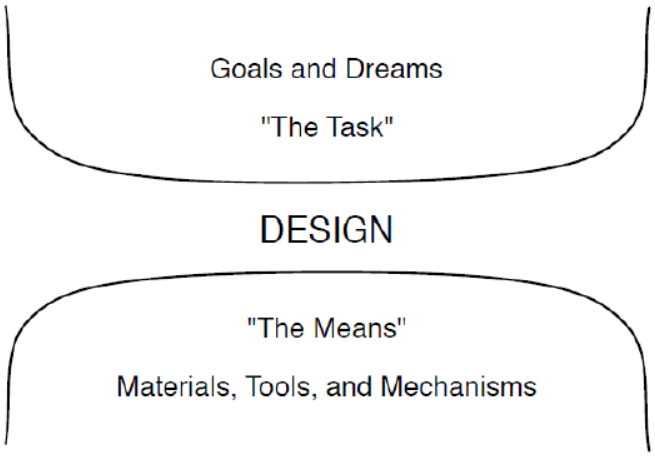


Figure 3.4: The continuing place of design

3.6.1 Entity Modeling

The entity modeling method is widely used for modeling the static part of an organization, using the basic concepts of entity, attribute and relationship to build the structure component. Underlying the structure of a database is the data model: a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints. There are various types of data models. They include:

- The entity-relationship model
- The relational model
- The object-oriented data model
- The object-relational data model
- The network data model
- The hierarchical data model

The entity-relationship model will be used for the analysis of the proposed system while the relational model will be used in the logical design phase. The entity-relationship diagram

expresses the overall logical structure of a database graphically. Figure 3.5 gives the entity-relationship model of the proposed system.

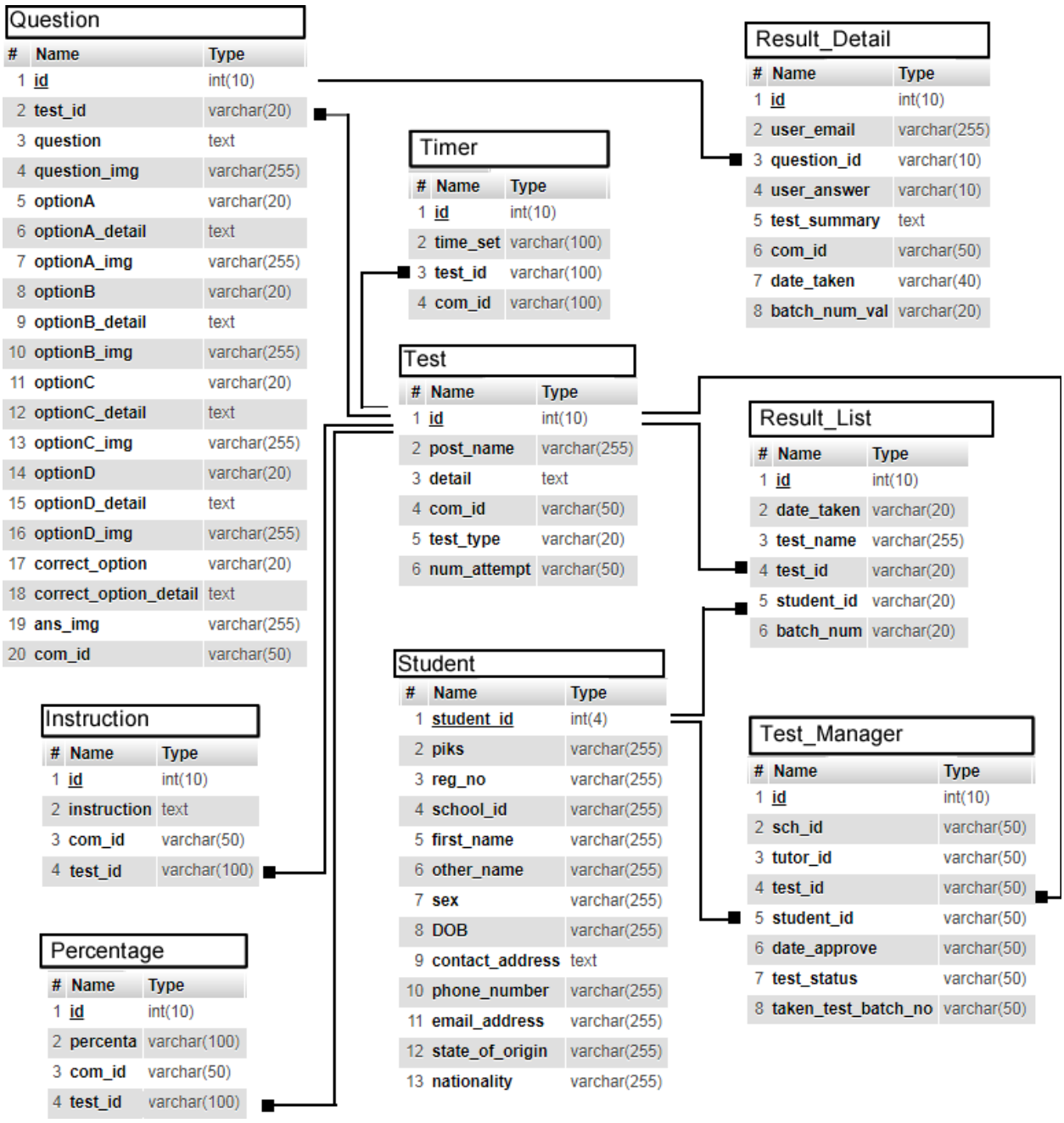


Figure 3.5: Entity-Relationship Model of the Proposed System

3.6.2 Relational Modeling

A collection of tables can be used to represent a database that conforms to an E-R database schema. For each entity set and for each relationship set in the database, there is a unique table to which is assigned the name of the corresponding entity set or relationship set. Each table has multiple columns, each of which has a unique name. Both the E-R model and the relational database model are abstract, logical representations of the real-world enterprise because the two models use similar design principles, the E-R model can be converted into a relational design. Converting a database representation from an E-R diagram to a table format is the way to arrive at a relational-database design from an E-R diagram. Although differences exist between a relation and a table, informally, a relation can be considered to be a table of values.

A relational database consists of tables, each of which is assigned a unique name. A relational schema consists of a list of attributes and their corresponding domains. The term domain refers to the group of values that are allowed for a column. It can also be used to describe business rules that govern each column. Business rules are organizational standards that require certain policies to be followed to ensure that business functions are carried out effectively. Business rules ensure that the database is consistent with business policies. At the simplest level, the domain represents the data type assigned to a column. The notion of keys (super key, candidate key and primary key) is also applicable to the relational model. A key is a property of the entity set rather than of the individual entities. Any two individual entities in the set are prohibited from having the same value on the key attributes at the same time. The designation of a key represents a constraint in the real-world enterprise being modeled. A super key is a set of one or more attributes that, taken collectively, allows us to identify uniquely an entity in the entity set. The concept of super key is not sufficient since a super key may contain extraneous

attributes. If K is a super key, so is any superset of K. Super keys for which no proper subset is a super key will contain no such extraneous attributes. Such minimal keys are called candidate keys. It is possible that several distinct sets of attributes could serve as candidate key. A primary key is a candidate key that is chosen as the principal means of identifying entities within an entity set. The primary key is chosen such that its attributes are never, or very rarely change.

3.6.3 Schema Diagram

A database schema, along with the primary key and foreign key dependencies, can be depicted by a relation schema. The database schema is given in the figure below.

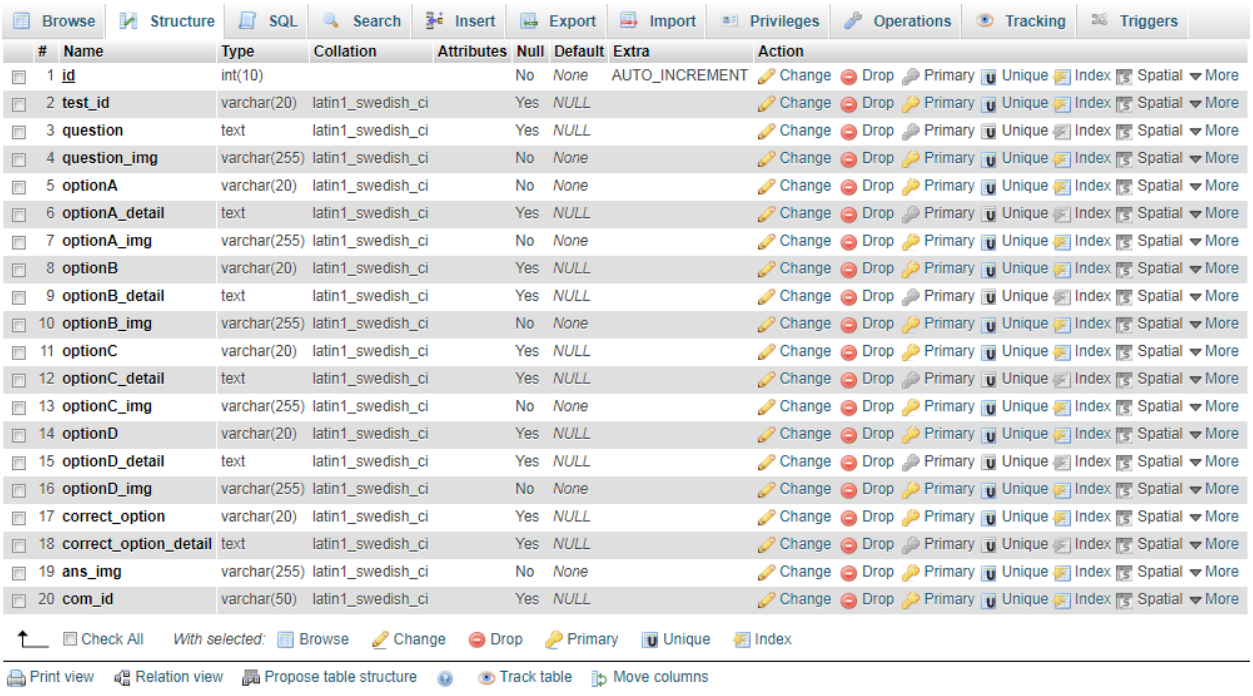


Figure 3.6: Database Schema

3.6.4 Data-Definition Language

A database system provides a database definition language to specify the database schema. A database schema is specified by a set of definitions expressed by a special language called the data-definition language (DDL). The execution of DDL statements updates a special

kind of tables called the data dictionary or data directory. A data dictionary contains metadata, that is, data about data. The schema of a table is an example of metadata; a database system consults the data dictionary before reading or modifying actual data. The storage structure and access methods used by the system are specified in a special type of DDL called a data storage and definition language. These statements define the implementation details of the database schema which is usually hidden from the users. The DDL also provides facilities to specify consistency constraints and it checks these constraints every time the database is updated. One of the SQL data definitions for the proposed information system database is given below:

```
CREATE TABLE IF NOT EXISTS `j_nems_question` (
`id` int(10) NOT NULL,
  `test_id` varchar(20) DEFAULT NULL,
  `question` text,
  `question_img` varchar(255) NOT NULL,
  `optionA` varchar(20) NOT NULL,
  `optionA_detail` text,
  `optionA_img` varchar(255) NOT NULL,
  `optionB` varchar(20) DEFAULT NULL,
  `optionB_detail` text,
  `optionB_img` varchar(255) NOT NULL,
  `optionC` varchar(20) DEFAULT NULL,
  `optionC_detail` text,
  `optionC_img` varchar(255) NOT NULL,
  `optionD` varchar(20) DEFAULT NULL,
  `optionD_detail` text,
  `optionD_img` varchar(255) NOT NULL,
  `correct_option` varchar(20) DEFAULT NULL,
  `correct_option_detail` text,
  `ans_img` varchar(255) NOT NULL,
  `com_id` varchar(50) DEFAULT NULL
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO INCREMENT=16 ;
```

CHAPTER FOUR

DEVELOPMENT AND IMPLEMENTATION

4.1. System Development and Implementation

Standards and procedures help in proper documentation of program for easy understanding by prospective users. They also aid in translating design to codes. Standardized documentation also helps in locating errors and in making changes. By structuring codes according to standards, the correspondence between design modules and code modules is maintained. Changes in the design therefore pose no problem to implement in the code. As discussed in chapter three, XP was employed as the software process model. Implementing the project requirements involved iterative and incremental release, a core agile practice.

4.2 Client-Side Development Language

The client-side of the web application refers to the web browser. Regardless of whichever technology is employed to construct the UI, the web browser will only process the received view as HTML. Hence, the client-side UI of the online testing system is mainly written in HTML5. HTML5 is the new standard of HTML and currently supported by a wide range of web browsers. Some HTML5 characteristics desired by the project are:

- Adding new functionality is purely based on HTML, CSS, Document Object Model (DOM), and JavaScript. This avoids unnecessary installation of external software plugins to the user device; and

- HTML5 prefers more mark-up to replace scripting. Many common features (e.g. validation, UI elements) are included in the standard of HTML5, hence the project can avoid reinventing the wheel.

However, HTML is static and its presentation unlikely to change once rendered by the web browser. Therefore, the project also utilizes JavaScript to introduce dynamic behaviours to the HTML pages; For example, the floating timer use by the testing system. JavaScript has become the dominant client side of scripting technology in recent years. It is open platform and has numerous libraries that enable the streamlining of modern web development. Most web browsers today support JavaScript execution, thus allowing more interactive and responsive experiences built into web applications. The most notable JavaScript library is JQuery and its validation plugin. JQuery provides API14 for HTML document traversal and manipulation, event handling, animation, and Asynchronous JavaScript and XML (Ajax). It allows the developer to create rich and dynamic UIs at the client side. Besides, the validation plugin contains the common validation functionality for HTML input such as textbox. This could significantly reduce data entry error and improve data integrity; plain HTML is not appealing to users and is likely to provide a poor user experience. Hence, HTML often uses CSS to define its look and feel.

4.3 Server-Side Programming Language

The underlying server side language for developing the system in PHP, In many ways the PHP language is representative of the stereotypical open source project, created to meet a developer's otherwise unmet needs and refined over time to meet the needs of its growing community. The origins of PHP date back to 1995 when an independent software development

contractor named Rasmus Lerdorf developed a Perl/CGI script that enabled him to know how many visitors were reading his online résumé. His script performed two tasks: logging visitor information, and displaying the count of visitors to the web page. Because the Web at the time was still a fledgling technology, tools such as these were nonexistent. Thus, Lerdorf's script generated quite a bit of interest. Lerdorf began giving away his toolset, dubbed Personal Home Page (PHP).

The clamor prompted Lerdorf to continue developing the language, with perhaps the most notable early change being a new feature for converting data entered in an HTML form into symbolic variables, encouraging exportation into other systems. To accomplish this, he opted to continue development in C code rather than Perl. Ongoing additions to the PHP toolset culminated in November 1997 with the release of PHP 2.0, or Personal Home Page/Form Interpreter (PHP/FI). The 2.0 release was accompanied by a number of enhancements and improvements from programmers worldwide. The new PHP release was extremely popular, and a core team of developers soon joined Lerdorf. They kept the original concept of incorporating code directly alongside HTML and rewrote the parsing engine, giving birth to PHP 3.0. By the June 1998 release of version 3.0, more than 50,000 users were using PHP to enhance their Web pages. Development continued at a hectic pace over the next two years, with hundreds of functions being added and the user base growing by leaps and bounds. At the beginning of 1999, Netcraft (www.netcraft.com), an Internet research and analysis company, reported a conservative estimate of a user base of more than 1 million, making PHP one of the most popular scripting languages in the world. Its popularity surpassed even the greatest expectations of the developers, and it soon became apparent that users intended to use PHP to power far larger applications than originally anticipated. Two core developers, Zeev Suraski and Andi Gutmans, took the initiative

to completely rethink the way PHP operated, culminating in a rewriting of the PHP parser, dubbed the Zend scripting engine.

4.3.1 Language Summary

- PHP stands for PHP: Hypertext Preprocessor. The product was originally named Personal Home Page Tools, and many people still think that's what the acronym stands for. But as it expanded in scope, a new and more appropriate (albeit GNU-ishly recursive) name was selected by community vote.
- PHP is a server-side scripting language, which is basically embedded into HTML.
- Rasmus Lerdorf: creator and original driving force behind PHP. The first part of PHP was developed for his personal use in late 1994.
- Zeev Suraski and Andi Gutmans, the two Israeli programmers who developed the PHP3 and PHP4 parsers, have also generalized and extended their work under the rubric of Zend.com (Zeev, Andi, Zend, get it?).

4.4 Relational Database Management System (RDBMS)

This project employs RDBMS as its data storage method. The project does not explicitly constrain the RDMBS selection to a particular vendor. In fact, the RDBMS usage could be particularly different when viewed from the development stage and the deployment stage. In the development stage, the data is often dummy and easily generated. The data schema also changes rapidly as design decisions changed. The database's tables need to be constantly dropped, altered, recreated; and facilating testing. Such required functionality would be better implemented by a lightweight RDBMS; that would allow instant creation of a database from scratch. On the other hand, the performance and functionality of RDBMS is more demanding

when the system is deployed into the actual environment. The RDBMS selected for the project is MySQL. MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments. Before You

4.5 Walkthrough

In order to demonstrate the usage of the system, a series of walkthroughs will be presented. Figure 4.1 is the Admin login form, the entry point for the Admin to access the system to create a test, populate test questions and set test parameters.

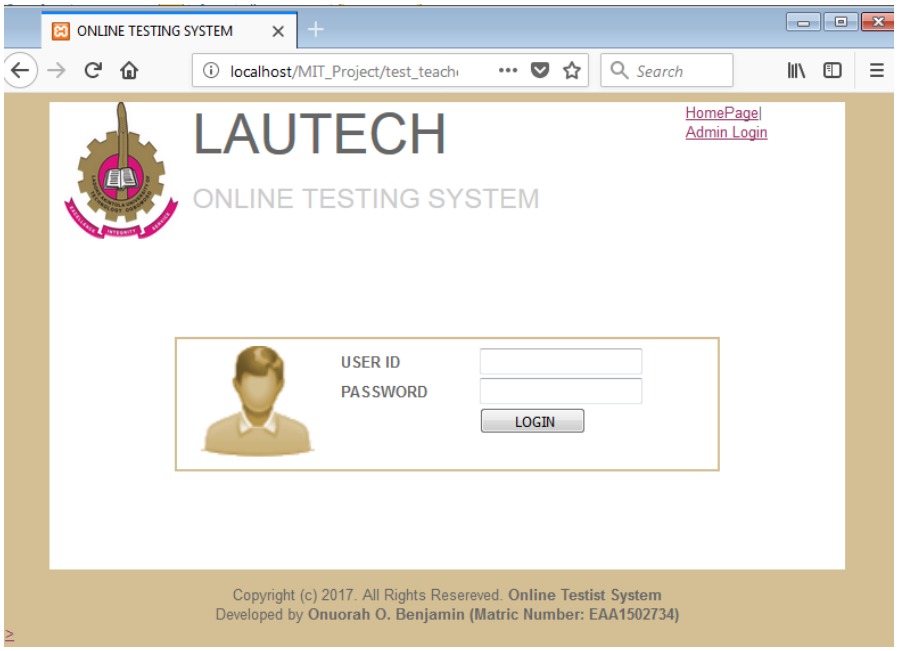


Figure 4.1: Admin Login page

4.5.1 Creating a Test

When the admin successfully login, the first page shown is the **Create Test** page as shown in Figure 4.2 this page allows the Admin to setup or create a new test, for example "MIT 709 Software Engineering Methodologies".

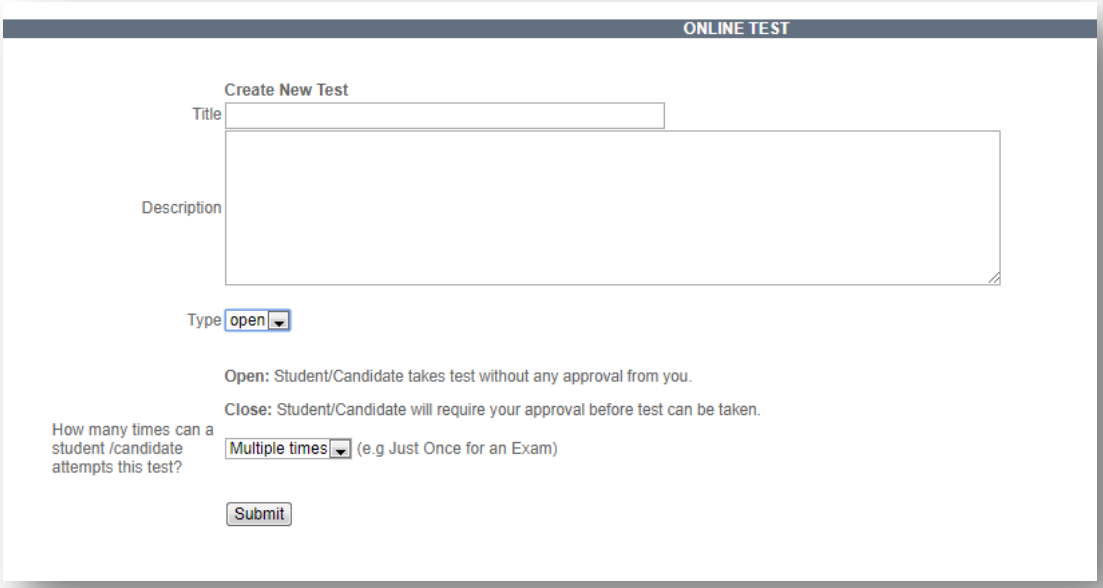


Figure 4.2: Create New Test page

4.5.3 Adjusting the Test properties

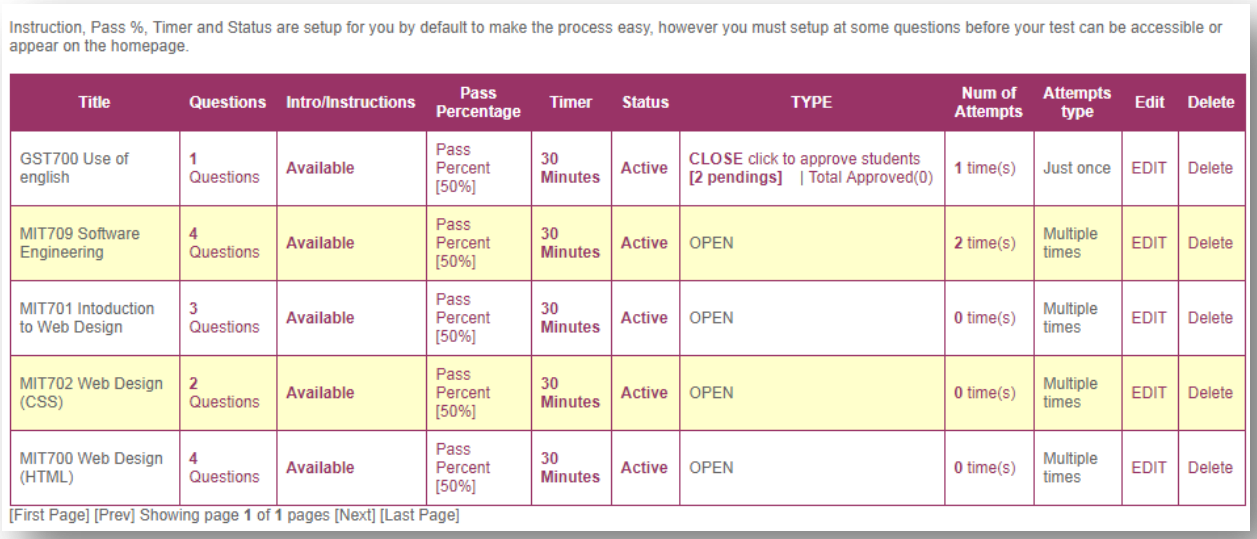
The record of created tests also appears just below the "Create Test" test page, see Figure 4.3, this record also presents adjustable information or properties of the tests. Below are the available properties options.

- **Questions:** this column displays the number of questions available under each Test, for example MIT709 Software Engineering has 4 questions setup by the Admin, the Admin clicks on a specify Questions of a particular test to setup the questions. (This will be discussed next)
- **Intro/Instructions:** Introduction/instructions that appears as the student is about taking the test
- **Pass Percentage:** Admin sets the minimum pass mark in percentage for each test. For example, the default is 50% for a pass mark.

- **Timer:** This is the duration or time allotted for a test. For example, 30 minutes is the default. This will appears as a countdown when student is writing the test.
- **Status:** This is made active by default, but you can turn it to be inactive if you are done with the test and don't want student to keep attempting it.
- **Type:** when creating a test, the admin specifies the type to be either open or close test

  Open: Student/Candidate takes test without any approval from Admin.

  Close: Student/Candidate will require Admin's approval before taking the test.
- **Num of Attempts:** This displays the total number of candidates that have taken a particular test, and from here the admin can see their result as well.
- **Attempts type:** How many times a student/candidate can attempts the test, it can either be "Just once" or "Multiple times".



Instruction, Pass %, Timer and Status are setup for you by default to make the process easy, however you must setup at some questions before your test can be accessible or appear on the homepage.

Title	Questions	Intro/Instructions	Pass Percentage	Timer	Status	TYPE	Num of Attempts	Attempts type	Edit	Delete
GST700 Use of english	1 Questions	Available	Pass Percent [50%]	30 Minutes	Active	CLOSE click to approve students [2 pendings] | Total Approved(0)	1 time(s)	Just once	EDIT	Delete
MIT709 Software Engineering	4 Questions	Available	Pass Percent [50%]	30 Minutes	Active	OPEN	2 time(s)	Multiple times	EDIT	Delete
MIT701 Intoduction to Web Design	3 Questions	Available	Pass Percent [50%]	30 Minutes	Active	OPEN	0 time(s)	Multiple times	EDIT	Delete
MIT702 Web Design (CSS)	2 Questions	Available	Pass Percent [50%]	30 Minutes	Active	OPEN	0 time(s)	Multiple times	EDIT	Delete
MIT700 Web Design (HTML)	4 Questions	Available	Pass Percent [50%]	30 Minutes	Active	OPEN	0 time(s)	Multiple times	EDIT	Delete

[First Page] [Prev] Showing page 1 of 1 pages [Next] [Last Page]

Figure 4.3: List of Test and properties

4.5.4 Populating the Questions

Test without a question is useless since no student will be able to attempts it, therefore this is always the first activity to be carried out, immediately after creating a test, as a New test will show “0 Questions” under the question column on the test properties, the admin will click the question link to load the Questions page see Figure 4.4.

- From this page the Admin will be able to enter the Question and optionally upload an image file with it if necessary (note: some questions come with drawings, formulas, picture, tables or diagrams).
- Enter option A, option B, option C, option D and optionally upload an image file with it if necessary.
- Select the correction answer
- Optionally explain the Answer
- Clicks submit to save the newly created question.
- See all created question for the selected test at the right corner of the page
- Delete existing or created questions from the list of the right corner of the page

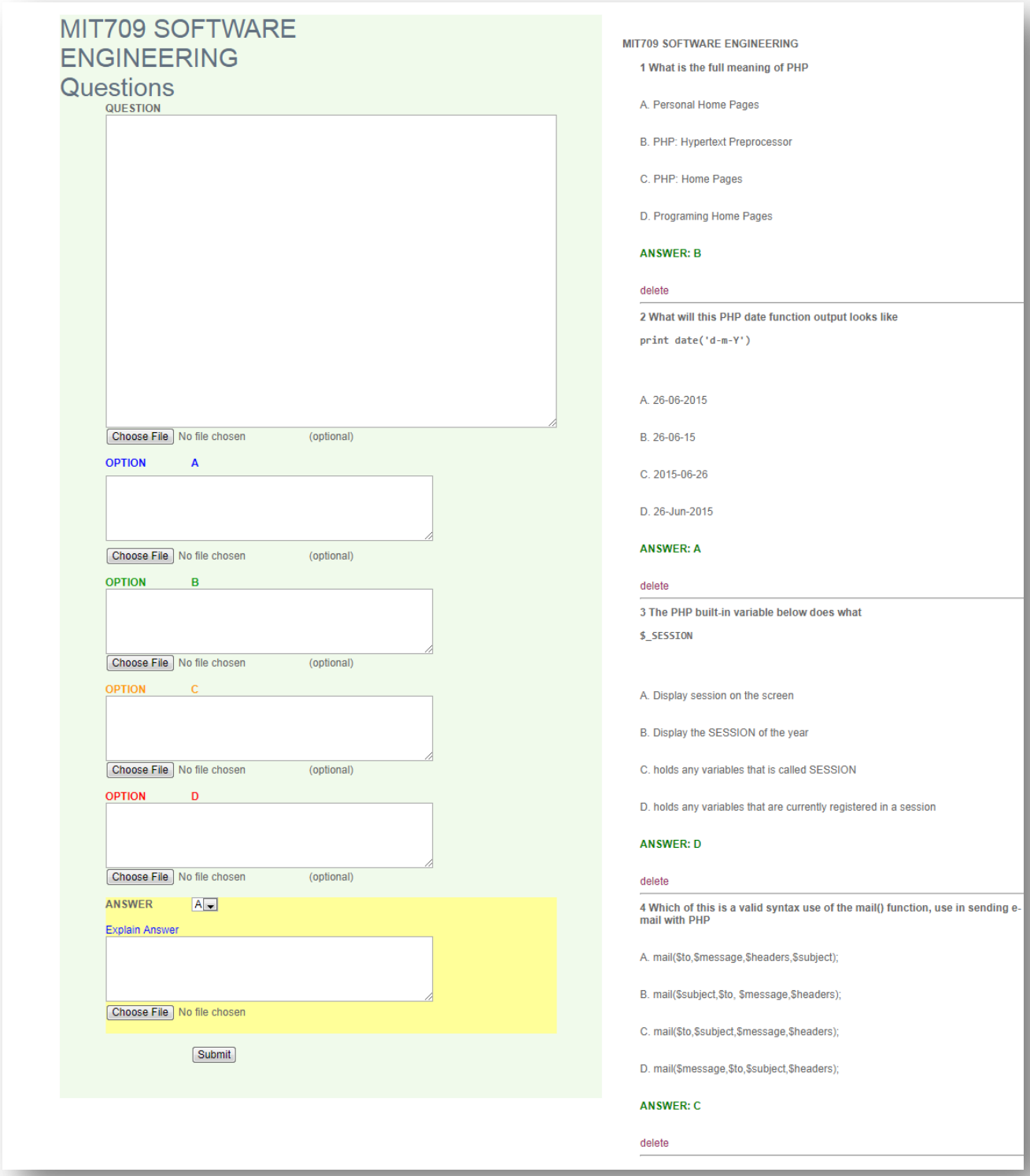


Figure 4.4: Adding or populating and managing test questions page

4.5.5 Student Login

Figure 4.5 represent the student login page, the access point for the student to access the online test.

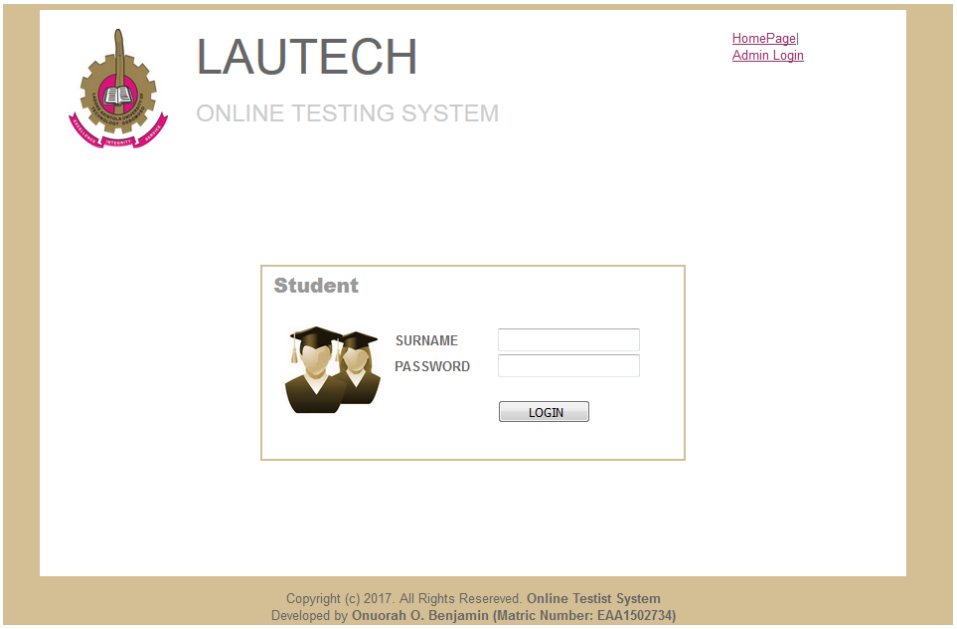


Figure 4.5: Student login page

Figure 4.6 represent the first page the student will see after a successful login, this page contains the student main menus at the left side and the student personal data at the right.

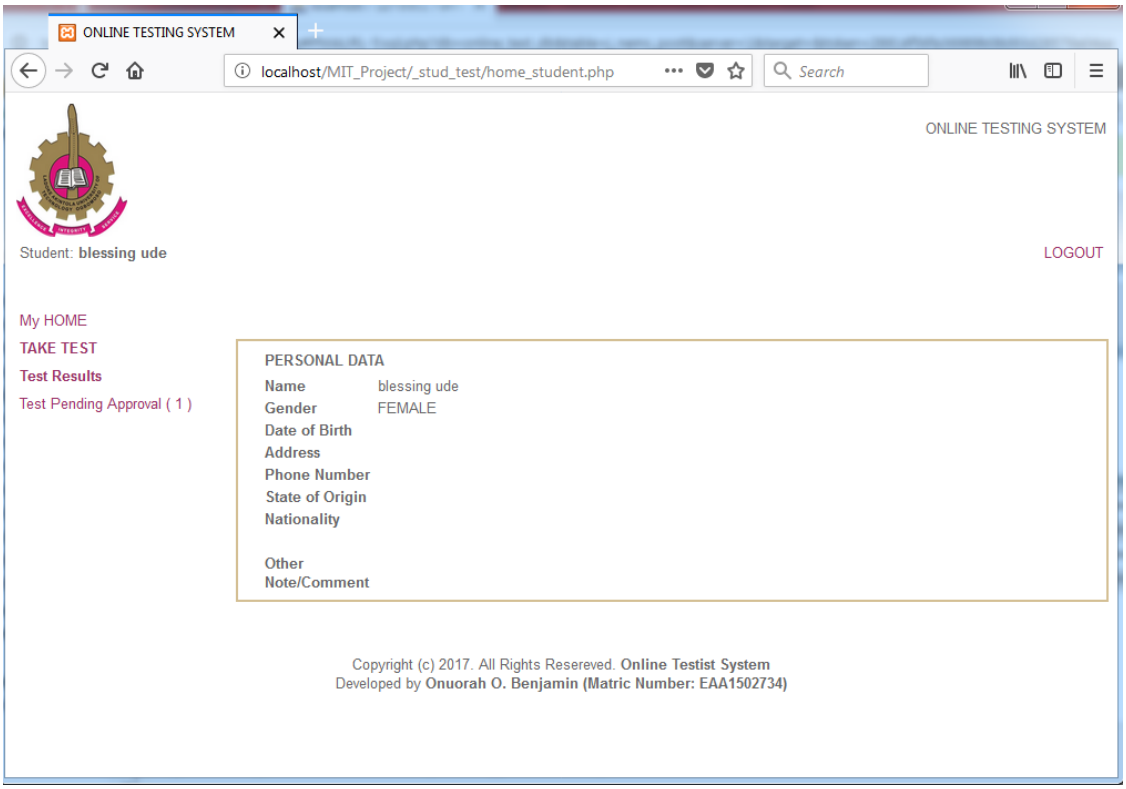


Figure 4.6: Student personal data page

4.5.5 Student Attempting the Test

After a successful login attempt by the student the student can then click on the "TAKE TEST" link from the student dashboard, to load the test page as shown in Figure 4.7

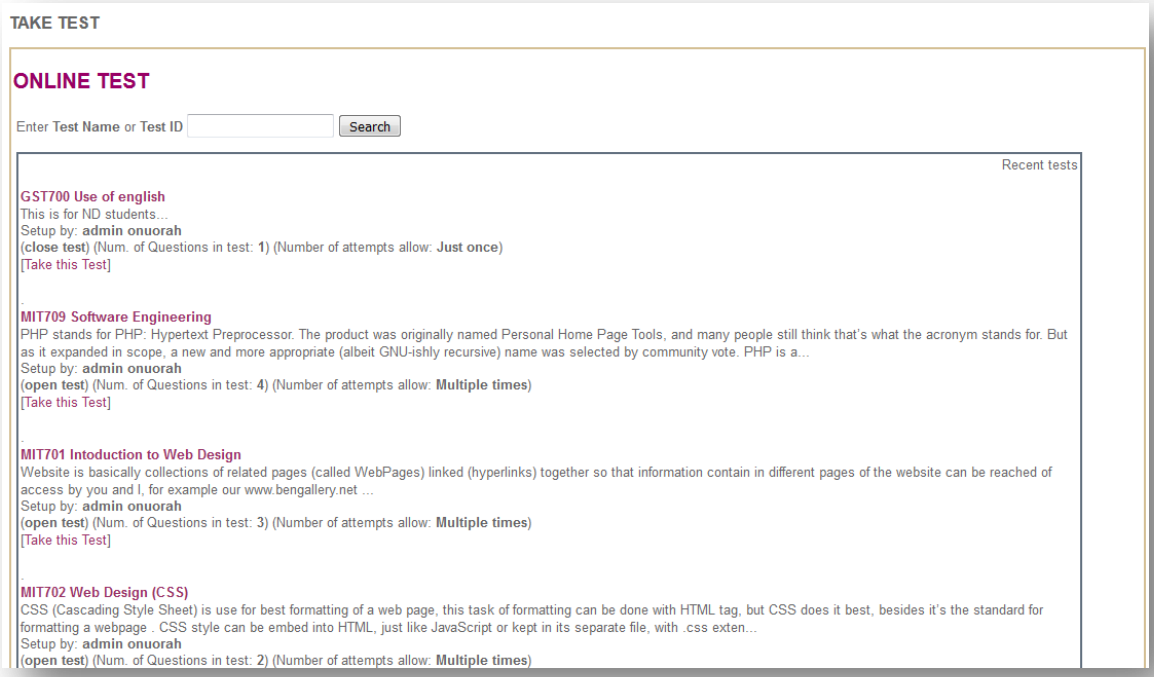


Figure 4.7: Take test page

Note: the test page above shows all the list and details of created tests, the student can then click on the Title or [Take this test] link to attempt a particular test. This often starts by loading the instruction page of the test as shown in Figure 4.8.

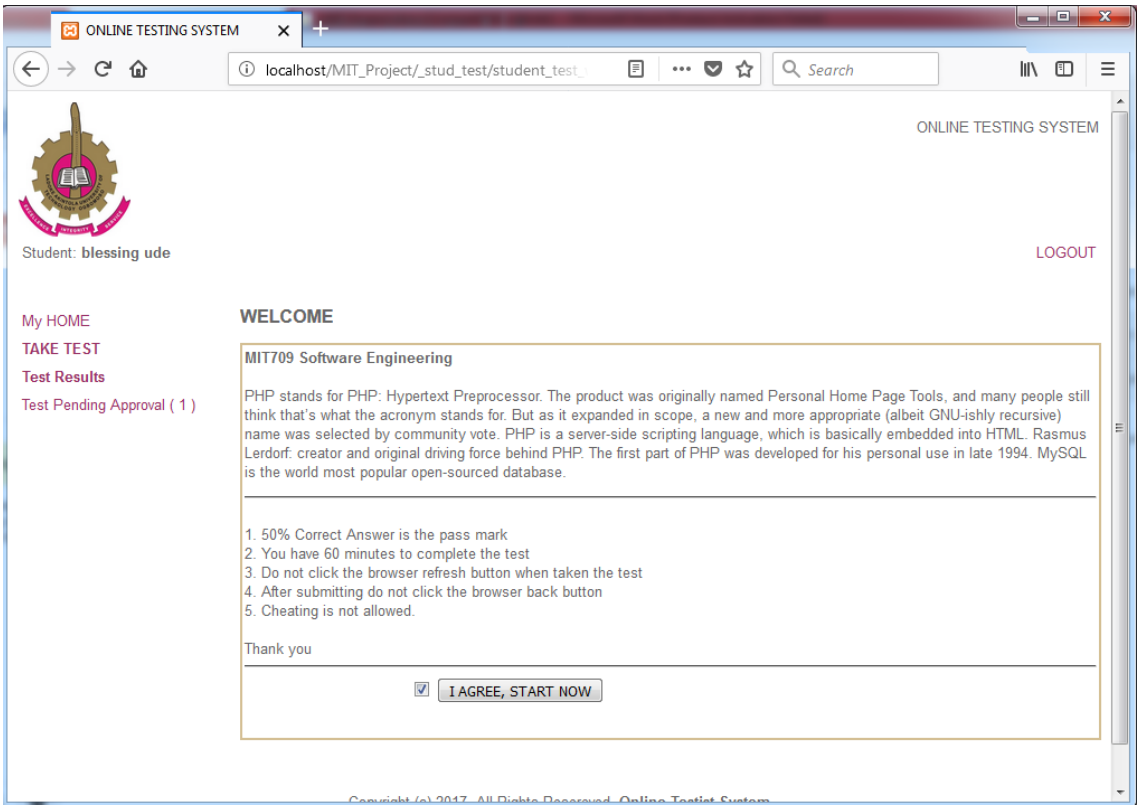


Figure 4.8: Test instruction page

The student is expected to click the "I AGREE, START NOW" to commence the test. As soon as this is done the test question loads and the timer start counting down. As shows in Figure 4.9

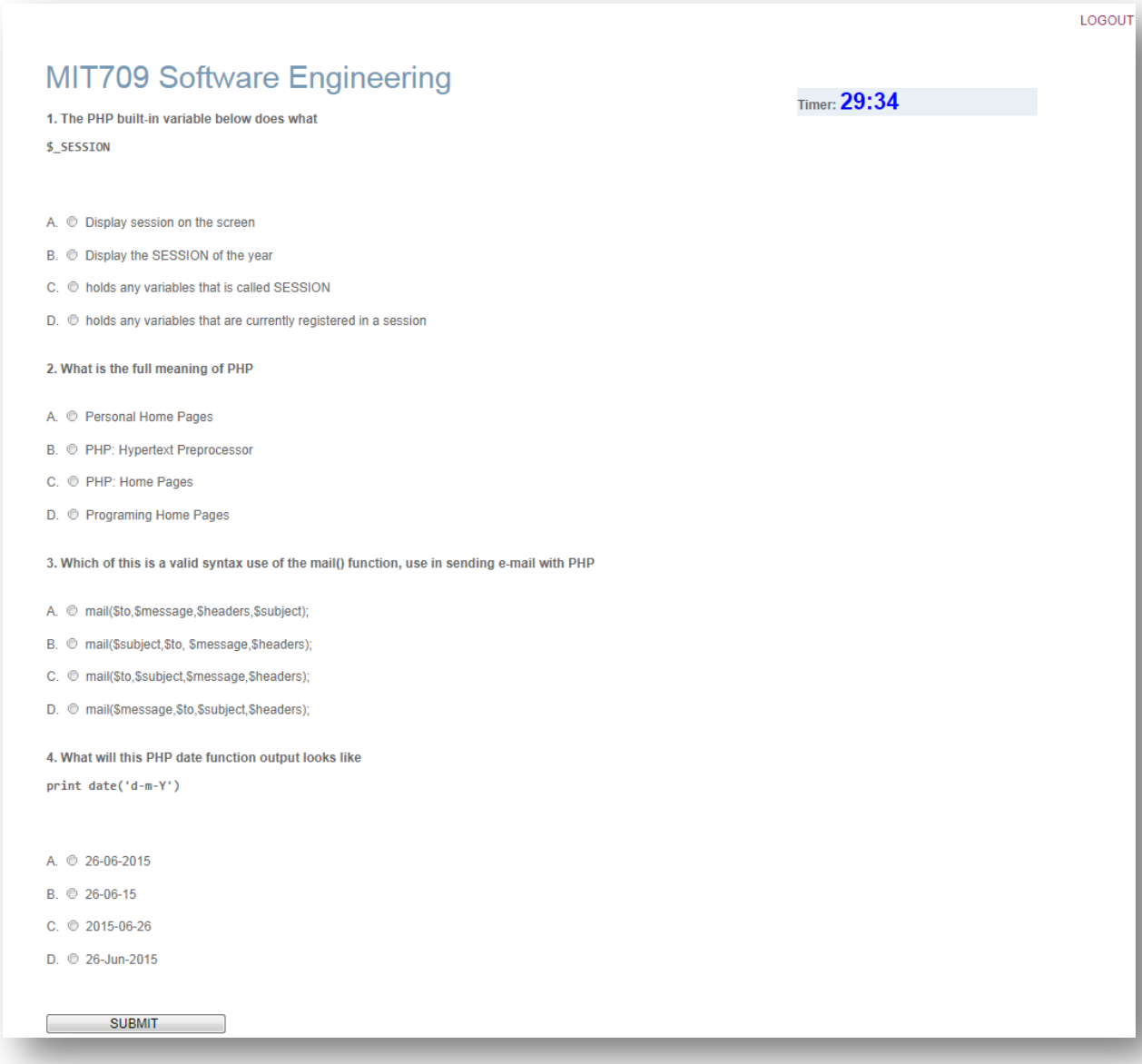


Figure 4.9: Testing page

After completing the test the student is expected to click the "SUBMIT" button and get instant result of pass or fail, based on percentage. Note: if the student is unable to complete the test during the specified time, the application does auto submission and all unattempt questions are assumed or recorded as failed attempts

4.5.6 Online Test Result

The student can click on the "Test Results" link from the menu to see all the record of past attempted Test as shown in Figure 4.10 and click "Result Detail" of a particular test to see result detail as shown in Figure 4.11

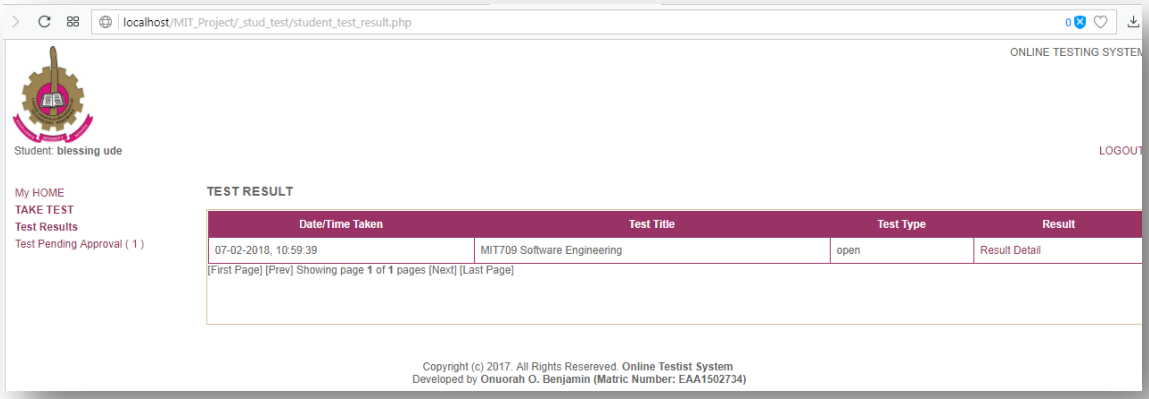


Figure 4.10: Result list page

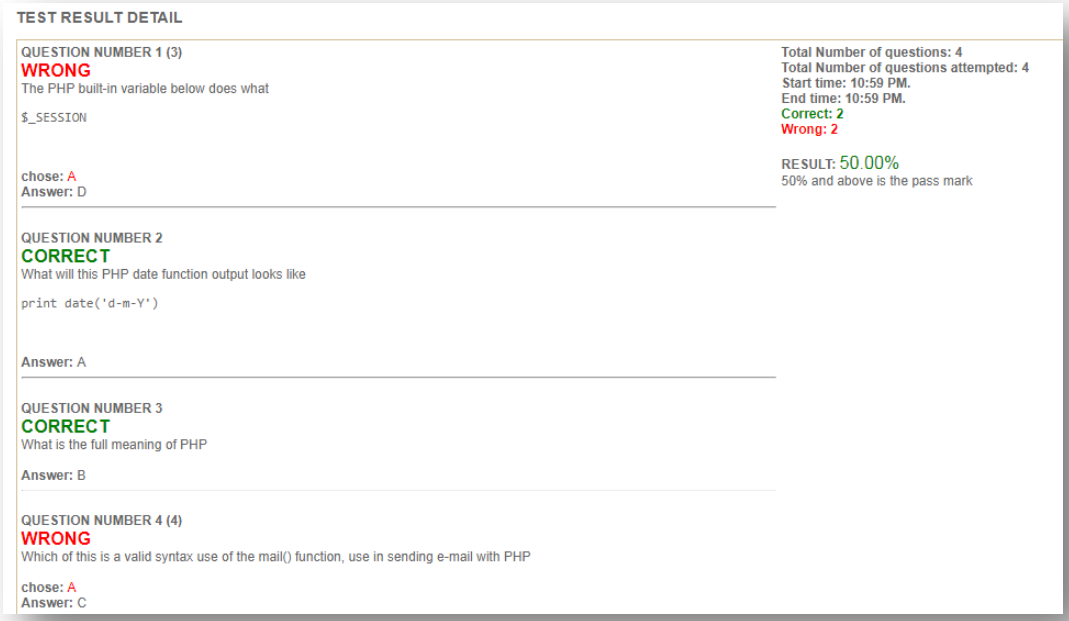


Figure 4.11: Result detail page

4.6. Limitations

The Online Testing System only supports objective questions with four (4) answer options to pick from. The Online Testing System will require the School management to setup computer and network infrastructure to support the successful implementation of the proposed system.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The project undertakes a viable solution to the problem of testing, preparing and evaluating students, most especially distance learning students. The integration of technology into education termed Educational technology, as given birth to many interesting and very useful education solutions, among them is an online testing system. The Online Testing System is a Web Application that provides access for Masters in Information Technology of Ladoke Akintola University as a case study to be able to attempt tests online regardless of their location. Administrator has a privilege to create test, add test questions and manage test properties as well as register students. Students can login, take multiple choice test and see the results instantly. It saves time and other resources for student, lecturer and the university management.

5.1 Recommendation

The design and implementation of the Online Testing System shows that it is very possible for the MIT students of LAUTECH to have a platform where thet can attempt tests online, so as to prepare them for their semester examinations. Furthermore in future it could be more economical to actually conduct examinations online using the system.

REFERENCES

Agrawal R., (2008). "The Claremont report on database research" (PDF). SIGMOD

Alexander I., (2003); "Stakeholders: who is your system for?," Computing & Control Engineering Journal, vol. 14, pp.

Allen G.T. (2006); SQL For Dummies, 6th Edition; Wiley Publishing, Inc., Indianapolis, Indiana.

Ambler S. W., (2004); The object primer : Agile Modeling-driven development with UML 2.0, 3rd ed. Cambridge: Cambridge University Press.

Ambler S. W., (2002); Agile modeling: effective practices for eXtreme programming and the unified process. New York: J. Wiley.

Ambler S., (2003); Agile Database Techniques: Effective Strategies for the Agile Software Developer: Wiley.

Barker R., (2002); Embedded Systems Development Guide: PHASE 1. Manchester: UMIST.

Beck K., (1999) "Embracing change with extreme programming," Computer, vol. 32, pp. 70-77.

Beck K., Beedle M., *et al*., (2001); Manifesto for Agile Software Development," in Manifesto for Agile Software Development.

Ben O. (2015); Web Design and Programming; Source: www.bengallery.net/ben/wdp.php

Billsus, D. and M. Pazzani (2000). User modeling for adaptive news access. *User Modeling*

Buxton J. N. and Randell B. (1969); Software engineering techniques: report on a conference sponsored by the NATO Science Committee, Rome, Italy NATO.

Chin L. T, (2013); "implementing a web-based computerized restaurant system".

Chris K, (2018) "EnterpriseDB Adding New Cloud Option for PostgreSQL Database.

Connolly T. M., Begg C. E., (2005); Database Systems: A Practical Approach to Design, Implementation, and Management: Addison-Wesley.

Diomidis S.. (2012) Git. 100-101. Available: http://doi.ieeecomputersociety.org/10.1109/MS.2012.61

Dix A. J., Finlay J. E., *et al* (2006) Human Computer Interaction: Pearson Education Canada.

Fowler M., (2012); Patterns of Enterprise Application Architecture: Pearson Education.

Fowler M., Foemmel M., (2006); "Continuous integration," Thought-Works) http://www. thoughtworks. com/Continuous Integration. pdf.

Gellersen H.W. *et al*., (1999); "Object-oriented Web application development," Internet Computing, IEEE, vol. 3, pp. 60-68.

Hornsby P.. (2010). Hierarchical Task Analysis [Online]. Available: http://www.uxmatters.com/mt/archives/2010/02/hierarchical-task-analysis.php

Hu C., (2013) "The nature of software design and its teaching: an exposition," ACM Inroads.

Jazayeri M., (2007); "Some Trends in Web Application Development," presented at the 2007 Future of Software Engineering.

Ken North, "SQL, NoSQL or SomeSQL?", Dr. Dobb's, Retrieved 2011-11-9.

Minh L. Ha, Wettengel T., *et al*, (2011) "Empowering UML application design with task models," in Information Technology and Multimedia (ICIM), International Conference on, pp. 1-6.

Mitra T. (2008). Documenting software architecture, Part 2: Develop the system context [Online]. Available: http://www.ibm.com/developerworks/library/ar-archdoc2/

Paul D. (2007); MySQL Cookbook, Second Edition; O'Reilly Media, Inc.

Pressman R. S (2010); Software Engineering, a Practitioner's approach.

Radeke F. and Forbrig P., (2007); "Patterns in task-based modeling of user interfaces," presented at the Proceedings of the 6th international conference on Task models and diagrams for user interface design, Toulouse, France.

Reenskaug T., (1979); "Models-views-controllers," Technical note, Xerox PARC, vol. 32, p. 55.

Roger . S. (2010); Software engineering : a practitioner's approach, 7th international

Royce W. W. (1970); "Managing the development of large software systems," in proceedings of IEEE WESCON.

Schach S. R., (2005); Object-oriented and classical software engineering, 6th ed ed. New York.

Sommerville I. (2011); Software engineering, 9th ed. Boston, MA: Pearson.

Swoyer, S. (2012) "NuoDB: A Database for the Cloud." TDWI. Nov. 13.

Taiwo O. (2015); Cloud Computing: Concepts, Technologies and Business Implications.

Taylor R. N., Hoek A., (2007); "Software Design and Architecture the once and future focus of software engineering,".

Tim C. *et al* (2004); PHP5 and MySQL Bible.

Tina C. (1996); Information Technology.

APPENDIX

APPENDIX A: Screenshots

This section presents the screenshots of remaining features. This report only selects several significant screenshots to represent the features. The features are quite self-expressive once the user has followed the application UI design. They are consistent and mapped to particular tasks of the user.

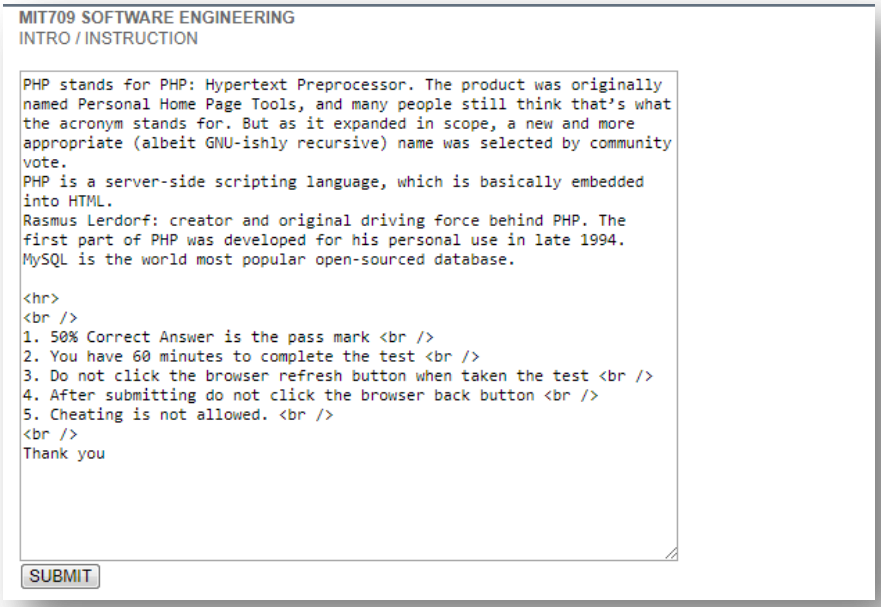


Figure A.1: Intro/Instructions

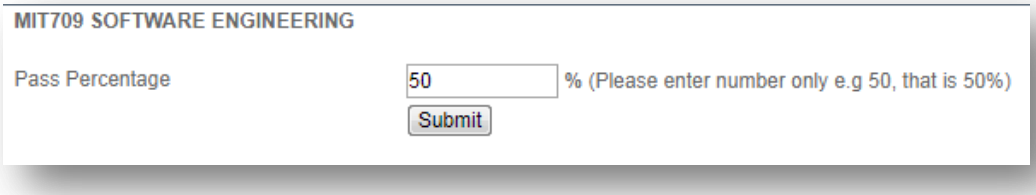


Figure A.2: Pass Percentage

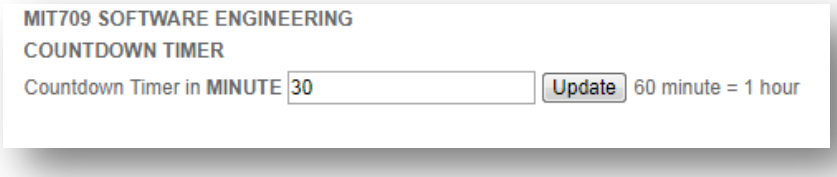


Figure A.3: Test Timer

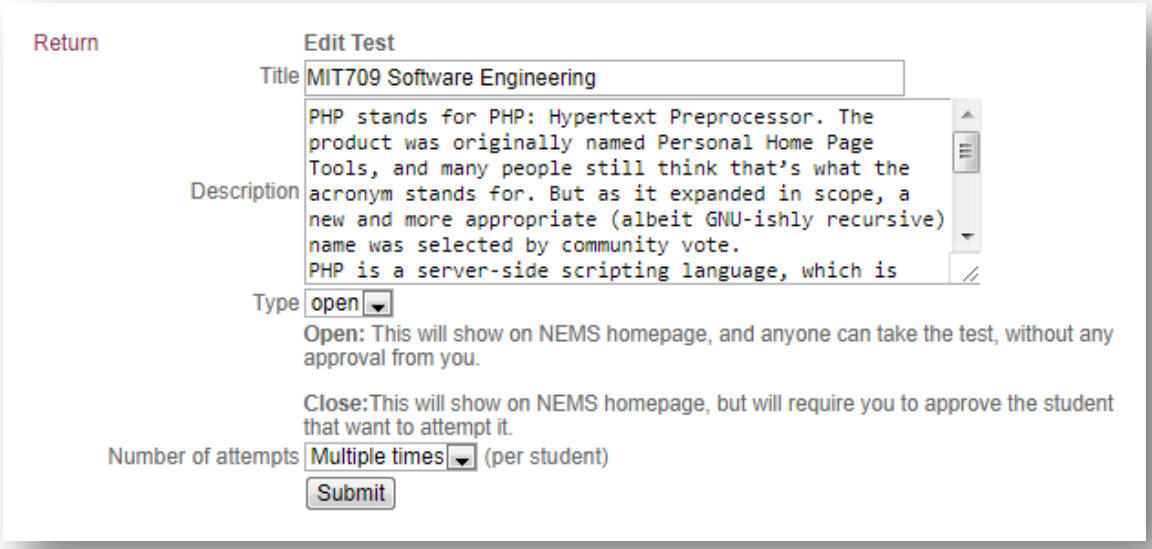


Figure A.4: Edit Test

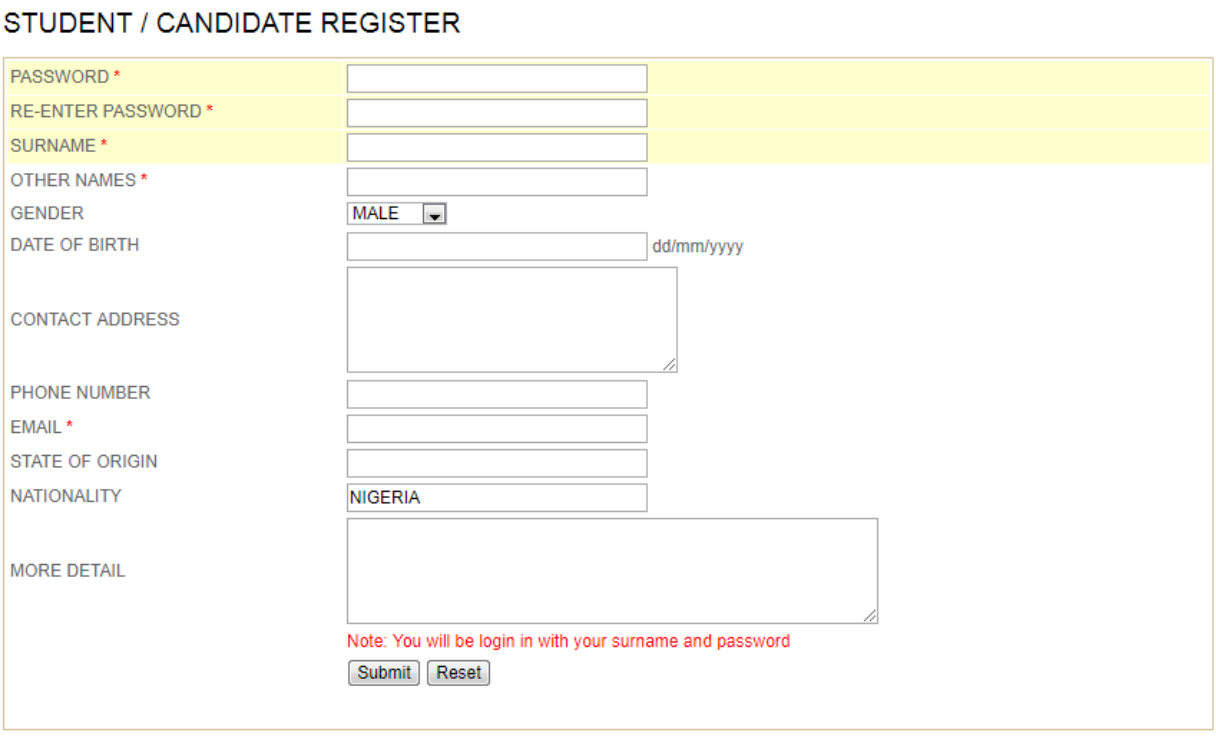


Figure A.5: Register Student

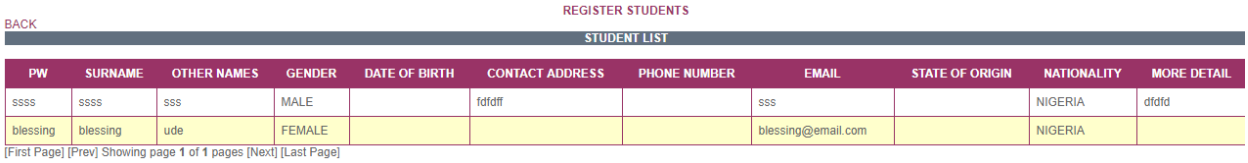


Figure A.6: List of registered students

APPENDIX B: JavaScript and PHP Source Code

This section presents some of the key or important PHP and JavaScript code

conn.php

```
<?php
define ('DB_USER', 'root');
define ('DB_PASSWORD', '');
define ('DB_HOST', 'localhost');
define ('DB_NAME', 'online_test_db');

$conn = mysqli_connect (DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
if(!$conn){
 die('Failure: ' . mysqli_error());
}
print "
<script language=\"JavaScript\" type=\"text/javascript\"
src=\"../ui/script.js\"></script>
<link href=\"../ui/style.css\" rel=\"stylesheet\"
type=\"text/css\" />
";

//Priviledge monitor
session_start();
?>
```

Test_dashboard.php

```
<?php
session_start();
error_reporting(E_ERROR | E_PARSE);
?>
<?php
include("1validate.php");
?>
<?php
//create and issue query
include("../conn.php");

if($_POST['ready']=="go"){
     if (($_POST['pst']!="")||($_POST['detail']!="")){



        $pst = mysqli_escape_string($conn, $_POST['pst']);
```

```
        $type = mysqli_escape_string($conn, $_POST['type']);
        $detail = mysqli_escape_string($conn, $_POST['detail']);
        $attempt = mysqli_escape_string($conn,
$_POST['attempt']);

                $sql11="INSERT INTO test VALUES ('','$pst',

     '$detail','$_SESSION[teacher_id]','$type','$attempt')";
                mysqli_query($conn, $sql11) or
die(mysqli_error());
                $post_id = mysqli_insert_id($conn);


// Setup instruction by default
$instruction = "$detail
<hr>
<br />
1. 50% Correct Answer is the pass mark <br />
2. You have 60 minutes to complete the test <br />
3. Do not click the browser refresh button when taken the test
<br />
4. After submitting do not click the browser back button <br />
5. Cheating is not allowed. <br />
<br />
Thank you";
                        $sql="INSERT INTO instruction
                        VALUES
('','$instruction','$_SESSION[teacher_id]','$post_id')";
                        mysqli_query($conn, $sql) or
die(mysqli_error());


// Setup pass percentage
                    $sql2="INSERT INTO limit
                    VALUES
('','50','$_SESSION[teacher_id]','$post_id')";
                    mysqli_query($conn, $sql2) or
die(mysqli_error());

// Setup pass percentage
                    $thirty_min = (30 * 60); //convert 30 min to
seconds
                    $sql2="INSERT INTO timer
                    VALUES
('','$thirty_min','$post_id','$_SESSION[teacher_id]')";
                    mysqli_query($conn, $sql2) or
die(mysqli_error());
```

```
// Setup the status
                    $sql2="INSERT INTO deadline
                    VALUES ('','1','$post_id',
'$_SESSION[teacher_id]')";
                    mysqli_query($conn, $sql2) or
die(mysqli_error());


               $msg = "<font color=\"blue\">Done!</font>";

     }else{
          $msg = "<font color=\"red\">Please enter value for
Title and Description field !</font>";
     }
}
?>
```

Student_test_start.php

```
<?php
session_start();
error_reporting(E_ERROR | E_PARSE);
?>

<?php
if (!$_SESSION['student_id']){
     print "

     <p>
     Sorry your session has ended</font></b></p>
         <a href=\"../index.php\">Click here</a> and  Login
again.";
     exit;
     }
include("../conn.php");

if(isset($_GET['pid'])){
     $test_id = mysqli_escape_string($conn, $_GET['pid']);
     $_SESSION['post_id']=$test_id;
}
$_SESSION['selected_options'] = "";
$_SESSION['testStartTime'] = new DateTime;
$_SESSION['another_time_start'] = date("h:i A.", time());

//  Retrive the  timer for the test, for a particular post
$sql_reg = "SELECT * FROM timer WHERE post_id =
'$_SESSION[post_id]'";
```

```
$sql_reg_query = mysqli_query($conn, $sql_reg) or
die(mysqli_error());
$data = mysqli_fetch_array($sql_reg_query);
$timer = $data['time_set'];
$sql2 = "SELECT * FROM test WHERE id = '$_SESSION[post_id]'";
          $result2 = mysqli_query($conn, $sql2) or
die(mysqli_error());
          $rec = mysqli_fetch_array($result2);
          $post_name = $rec['post_name'];
          $num_attempt = $rec['num_attempt'];

if($num_attempt == "Just once"){
     $sql2 = "SELECT * FROM result_list WHERE test_id =
'$_SESSION[post_id]' AND student_id='$_SESSION[student_id]'";
          $result2 = mysqli_query($conn, $sql2) or
die(mysqli_error());
          if (mysqli_num_rows($result2) == 1) {
               include("../banner_login.php");
               print "
               <link href=\"../images/style.css\"
rel=\"stylesheet\" type=\"text/css\" />

               <p align=\"center\">
               <br /><br /><br />
                    <b><font color=\"red\">Sorry this test is to
be taken once AND
                    record shows that you have taken it once
already!</font></b>
                    <br />
                    <a href=\"student_test.php\">CLICK HERE to
Return</a>
               </p>
               ";
               exit();
          }
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />
<title>ONLINE TESTING SYSTEM</title>

<style type="text/css">
```

```
<!-- @import "../_stud - Copy/css/jquery.countdown.css"; -->
#defaultCountdown { width: 240px; height: 45px; }
</style>

<link href="../images/style.css" rel="stylesheet"
type="text/css" />
<style type="text/css">
</style>
</head>

<body>
<table width="100%" border="0" align="center" cellpadding="0"
cellspacing="0">
  <tr bgcolor="#FFFFFF">
   <td colspan="3"><?php include("../banner_login.php"); ?></td>
  </tr>
  <tr bgcolor="#FFFFFF">
    <td width="17%"><a href="../logout.php"></a></td>
    <td width="53%"> </td>
    <td width="30%"><div align="right"></div></td>
  </tr>
  <tr>
    <td><div align="center"></div></td>
    <td> </td>
    <td> </td>
  </tr>
  <tr>
    <td height="104" valign="top">

<?php include("navigation_student.php"); ?>  </td>
    <td colspan="2" valign="top">
          <table width="100%" border="0" align="center"
cellpadding="0" cellspacing="0">
            <tr>
              <td width="66%" height="318" valign="top"><table
width="95%" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#E8EFF5">
                <tr>
                  <td width="100%" height="43" valign="top"
bgcolor="#FFFFFF"><form id="form1" name="form1" method="post"
action="student_test_start_complete.php">

                    <table width="100%" border="0"
align="center" cellpadding="0" cellspacing="0">
                      <tr>
                        <td>
<?php
```

```
          print "<span class=\"style2\">$post_name</span><br
/><br />";
?>
                            <?php
$sql2 = "SELECT * FROM question  WHERE
test_id='$_SESSION[post_id]'";
$result2 = mysqli_query($conn, $sql2) or die(mysqli_error());
$num_row = mysqli_num_rows($result2);


$sql = "SELECT * FROM question WHERE
test_id='$_SESSION[post_id]'
 ORDER BY rand() limit 0,$num_row";
$result = mysqli_query($conn, $sql) or die(mysqli_error());

//get the number of rows in the result set; should be 1 if a
match

print "

<table width=100% border=0>

";



$counter = 0;

while($info = mysqli_fetch_array($result))

     {
     $counter = $counter + 1;

     //if authorized, get the values of f_name n l_name
           $id=$info['id'];
          $question=$info['question'];
           $question=stripslashes($question);
           $optionA=$info['optionA'];
           $optionA_detail=$info['optionA_detail'];
$optionA_detail=stripslashes($optionA_detail);
           $optionB=$info['optionB'];
           $optionB_detail=$info['optionB_detail'];
$optionB_detail=stripslashes($optionB_detail);
           $optionC=$info['optionC'];
           $optionC_detail=$info['optionC_detail'];
$optionC_detail=stripslashes($optionC_detail);
```

```
            $optionD=$info['optionD'];
            $optionD_detail=$info['optionD_detail'];
$optionD_detail=stripslashes($optionD_detail);
            $correct_option=$info['correct_option'];
//----------------IMAGE PROCESS------------
            $question_img=$info['question_img'];
            $optionA_img=$info['optionA_img'];
            $optionB_img=$info['optionB_img'];
            $optionC_img=$info['optionC_img'];
            $optionD_img=$info['optionD_img'];
            $ans_img=$info['ans_img'];
$im1=$im2=$im3=$im4=$im5=$im6="";
if($question_img != ""){
     $im1="<img src=\"admin/piks/$question_img\" width=500>";
}
if($optionA_img != ""){
     $im2="<img src=\"admin/piks/$optionA_img\" width=500>";
}
if($optionB_img != ""){
     $im3="<img src=\"admin/piks/$optionB_img\" width=500>";
}
if($optionC_img != ""){
     $im4="<img src=\"admin/piks/$optionC_img\" width=500>";
}
if($optionD_img != ""){
     $im5="<img src=\"admin/piks/$optionD_img\" width=500>";
}
if($ans_img != ""){
     $im6="<img src=\"admin/piks/$ans_img\" width=500>";
}
     print "
<tr>
<td>
<b> $counter. $question </b>
<br />
$im1
<br /><br />
A. <input name=\"$id\" type=\"radio\" value=\"A\" />
$optionA_detail
<br />
$im2
<br />
B. <input name=\"$id\" type=\"radio\" value=\"B\" />
$optionB_detail
<br />
$im3
<br />
```

```
C. <input name=\"$id\" type=\"radio\" value=\"C\" />
$optionC_detail
<br />
$im4
<br />
D. <input name=\"$id\" type=\"radio\" value=\"D\" />
$optionD_detail
<br />
$im5
<br /><br />
</td>
</tr>
<tr>
<td >
  <div align=\"right\"> </div>
 </td>
</tr>
<tr>
<td bgcolor=\"#FFFFFF\"></td>
</tr>
     ";
     $_SESSION['num_of_question']=$counter;
     // store the id in and array
     //$id_arr[] = $id;
     $_SESSION['selected_options'][$id] = $info;
}
print "</table>";
          ?></td> </tr>
                      <tr>
                        <td><div align="center">
                        </div></td>
                      </tr>
                    </table>
                    <p>
                      <input name="test_id" type="hidden"
id="test_id" value="<?php print $_GET['id'] ?>" />
                    </p>
                    <input type="submit" name="Submit" value="
SUBMIT                    " />
                  </form>
                  <p> </p></td>
                </tr>
                <tr>
                  <td valign="bottom"
bgcolor="#FFFFFF"><p> </p></td>
                </tr>
              </table>
```

```
                <p> </p>
                <p> </p>
                <p class="style11"> </p></td>
              <td width="34%" align="center"
valign="top"><p> </p>
                <table width="80%" border="0" cellspacing="0"
cellpadding="0">
                  <tr>
                    <td align="center" valign="middle"
bgcolor="#E8EFF5"><div align="left" style="position: fixed;
width: 250px; background-color: #E8EFF5"> <strong>Timer:
</strong> <span id="countdown" class="timer"></span>
                      <script>
//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+
function Redirect()
{
    //window.location="http://www.bengallery.net";
     document.getElementById("form1").submit();
}

var seconds = <?php print $timer; ?>;
function secondPassed() {
    var minutes = Math.round((seconds - 30)/60);
    var remainingSeconds = seconds % 60;
    if (remainingSeconds < 10) {
        remainingSeconds = "0" + remainingSeconds;
    }
    document.getElementById('countdown').innerHTML = "<font
color=blue size=5><b>"+minutes + ":" +
remainingSeconds+"</b></font>";
    if (seconds == 0) {
        clearInterval(countdownTimer);
        document.getElementById('countdown').innerHTML = "<font
color=red size=5><b>TIME UP</b></font><font color=red> Please
submit: else you will be logout automatically in the next few
seconds</font>";

          //------------ SENT OUT AFTER 10 SEC. -----------
          setTimeout('Redirect()', 8000);


    } else {
        seconds--;
    }
}
```

```
var countdownTimer = setInterval('secondPassed()', 1000);
                </script>
                    </div></td>
                  </tr>
                </table>
                <p> </p></td>
            </tr>
          </table>
          <p> </p>
    <p> </p></td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
    <td> </td>
  </tr>
</table>

  <?php include("../footer.php"); ?>


</body>
</html>
```

Note: below is a sample login details for the Online testing system
Admin login id: admin, password:admin
Student login id: blessing, password: blessing