# UNIVERSITY OF CAPE COAST

# FACULTY OF PHYSICAL SCIENCE

# DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

# ELECTRONIC EXAMINATION TIMETABLE FOR UCC

# BY

# DAVID TWUMASI

# MARY JONAH

# RASHIDAT NOIBI

# SUPERVISED BY: AKUA GYANBA BINEY

# JUNE 2014

# UNIVERSITY OF CAPE COAST

# FACULTY OF PHYSICAL SCIENCE

# DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

# ELECTRONIC EXAMINATION TIMETABLE FOR UCC

# BY

# DAVID TWUMASI SC/ITC/10/0005

# MARY JONAH SC/ITC/10/0009

# RASHIDAT NOIBI SC/ITC/10/0024

A project work submitted to the Department of Computer Science and Information Technology, School of Physical Sciences, University of Cape Coast, in partial fulfillment of the requirement for the award of a B SC. Information Technology degree.

**JUNE 2014**

# Declaration

## Candidate’s Declaration

We hereby declare that this project work is the result of our own original research and no part of it has been presented before for another degree in this university or elsewhere.

Candidate’s Signature: ……………………… Date: ……………

Name: David Twumasi

Index Number: SC/ITC/10/0005

Candidate’s Signature: ……………………… Date: ……………

Name: Mary Jonah

Index Number: SC/ITC/10/0009

Candidate’s Signature: ……………………… Date : ……………

Name: Rashidat Noibi

Index Number: SC/ITC/10/0024

**Supervisor’s Declaration**

I hereby declare that the preparation and presentation of this project work was supervised in accordance with the guidelines on supervision of project work laid down by the University of Cape Coast.

Supervisor’s Signature: ……………………… Date: ……………

Name: Mrs. Akua Gyanba Biney

# Abstract

The purpose of our project was to develop an electronic examination timetable generator for university of Cape coast. The manual way of drafting the examination timetable has been the trend in university of Cape coast for some number of years now. This has led to many problems and disadvantages which include a class writing two papers on the same day and at the same time, assignment of bigger classes to smaller classrooms and also the consumption of resources e.g. the examination board receives sitting allowances when they sit to draft the timetable.

We intended as a group to help the university in moving from the manual to electronic or automated way of drafting her timetable. This will automate the whole system by reducing the number of people who sit to draft the timetable to one. A single individual can generate the timetable by some few clicks and less time will be spent as compared to the manual way. This reduces the number of resources used in drafting the timetable.

Basically, there will be an Admin who generates the timetable and add or delete users. The users of the system will be the HODs who can also view their departmental timetable through search by department name. The application which is the electronic examination timetable will take certain inputs from the Admin e.g. start and end date for the examination, examination times and courses upload. After these inputs, the application using these information will generate the timetable when the Admin clicks on the generate button. Some software tools used in developing this application includes HTML5, CSS, PHP, and JavaScript.

# Dedication

This work is dedicated to the Almighty God for guiding and protecting us throughout our four years of study at the University of Cape Coast.

Thank you to our entire family especially our parents and siblings for their prayers, encouragement and financial support throughout our undergraduate degree programme. To our supervisor, Mrs.Akua Gyanba Biney, we are most grateful.

# Acknowledgement

We would like to express our profound gratitude to the Almighty God for His grace, mercies and guidance throughout our stay on the University of Cape Coast campus.

We would like to acknowledge all the individuals that assisted and contributed immensely to the completion of this project.

We are very grateful to Mrs. Akua Gyanba Biney (our supervisor) for her relentless advice, guidance and follow up throughout the project span.

Special thanks go to Mr. Charles Haruna and the entire staff of the department of Computer Science and Information Technology.

Special gratitude goes to our parents for the motivation and financial support.

# TABLE OF CONTENTS

**1 Introduction** ……….…………………………………………………… 1

1.1 Introduction .…………………………………………………….. 1

1.2 Background .…………………………………………………….. 2

1.3 Motivation/Objectives ….……………………………………….. 3

1.4 Methodology ……………………………………………………. 4

1.5 Schedule and Timeline ..………………………………………… 7

1.6 Conclusion ...…………………………………………………….. 7

1.7 Thesis Organization .…………………………………………….. 7

**2 Background and Literature Review ……………………………….…. 9**

2.1 Introduction ..……………………………………………………. 9

2.2 Structure of related Automated Timetable generator ………….. 10

2.3 Genetic Algorithm ….………………………………………….. 11

2.4 Conclusion ...…………………………………………………… 12

**3 Design and Implementation .................................................................. 12**

3.1 Introduction .……………………………………………………. 14

3.2 System Overview ...…………………………………………….. 14

3.3 Methodology …………………………………………………… 15

3.4 System Architecture .…………………………………………… 19

3.5 Structure of the electronic examination timetable ...…………… 20

3.6 Modules purpose and functionality ……………………………. 20

3.6.1 Database ..……………………………………………… 21

3.6.2 Login/Authentication ..………………………………… 21

3.6.3 Administrator module …………………………………. 21

3.6.4 Member module ……………………………………….. 21

3.6.5 Search and lookup .…………………………………….. 22

3.7 User classes and characteristics ……………………………….. 22

3.7.1 Administrator ………………………………………….. 22

3.7.2 Members …………………………………………….… 22

3.8 Use case Diagrams for electronic examination timetable …….. 24

3.9 External and Interface requirement ..………………………….. 25

3.9.1 User Interface requirements ..…………………………. 25

3.9.2 Sitemap ..……………………………………………… 29

3.9.3 Database Design ……………………………………… 29

3.9.4 Data dictionary ……………………………………….. 31

**4 Testing, Evaluation and Maintenance ..……………………………. 31**

4.1 Introduction …………………………………………………... 36

4.2 Order of testing .………………………………………………. 36

4.3 Resource requirement for Testing .…………………………… 37

4.3.1 Hardware requirement .……………………………….. 37

4.3.2 Software requirement ...………………………………. 37

4.4 Testing Strategy .……………………………………………... 37

4.5 Testing Methodology ...………………………………………. 38

4.6 Testing Process ...…………………………………………….. 38

4.6.1 Functionality Testing ………………………………… 38

4.6.2 Usability Testing ..……………………………………. 41

4.6.3 Interface Testing ...…………………………………… 42

4.6.4 Database Testing .……………………………………... 43

4.6.5 Compatibility Testing .………………………………… 43

4.6.6 Security Testing ..……………………………………… 44

4.7 Dependencies ...………………………………………………... 46

4.7.1 Hardware dependencies .………………………………. 46

4.7.2 Software dependencies ...………………………………. 46

4.8 Maintenance ..………………………………………………….. 46

**5 Conclusion and Recommendations .…………………………………. 47**

5.1 Introduction ..………………………………………………....... 47

5.2 Conclusion ...……………………………………………..…….. 47

5.3 Recommendation ………………………………………………. 48

**References ……………………………………………………………………... 50**

**Appendix ………………………………………………………………………. 52**

# LIST OF FIGURES

Figure1 Modules and their functionalities ……………………………… 20

Figure2 Electronic Examination Timetable generator use case diagram .. 24

Figure3 User Interface for Index Page ……..…………………………… 25

Figure4 Homepage for Admin ………………………………………….. 26

Figure5 Members Page …………………………………………………. 27

Figure6 Course Page ……………………………………………………. 28

Figure7 Sitemap ...………………………………………………………. 29

Figure8 Admin Internal links ...…………………………………………. 39

Figure9 HODs Internal links ...………………………………………….. 39

Figure10 Login Page ……………………………………………………... 55

Figure11 Course Page ...………………………………………………….. 56

Figure12 Course Upload Dialog box …………………………………….. 57

Figure13 Course Page ...………………………………………………….. 57

Figure14 Course Page ..…………………………………………………... 58

Figure15 Add Member page ……………………………………………... 59

Figure16 Delete and Edit Member details Page ..………………………... 59

Figure17 Admin Homepage (Date and Time) ..………………………….. 60

Figure18 Delete Date Page ………………………………………………. 61

Figure19 Timetable Page ………………………………………………… 62

Figure20 Search Page ..…………………………………………………... 64

Figure21 Member Reset Password Page ………………………………… 65

# LIST OF TABLES

Table1 Schedule and Timeline ………………………………………. 7

Table2 Use case Scenario for Admin ..……………………………… 23

Table3 Use case Scenario for Members ..………………………....… 23

Table4 Course Table ...……………………………………………..... 32

Table5 Venue Table ……………………………………………….… 33

Table6 Date Table ...……………………………………………….… 33

Table7 Duration Table …………………………………………….… 33

Table8 Login Table ..…………………………...………………….… 34

Table9 Users Table ..………………………………………………..... 34

Table10 FinalTimetable Table ……………………………………….... 35

# CHAPTER 1

# INTRODUCTION

## 1.1. INTRODUCTION

In every institution, students are assessed based on what they are taught at the end of every term or semester. To assess the students, there is a need for examination which is conducted using a timetable. Examination as defined by **Wikipedia** is “an assessment intended to measure a test-takers knowledge, skill, aptitude, physical fitness or classification in many other topics (example beliefs.)”. The assessment of students through the writing of examination is an activity that needs planning due to some factors which includes the population of students (i.e. due to the population of students, all students cannot write examination on the same day), availability of rooms for the examination, etc. In resolving these problems, an examination timetable is drawn by a selected committee who is responsible for organizing and conducting this exercise. According to Wikipedia’s encyclopedia, timetable is a basic [time management](http://en.wikipedia.org/wiki/Time_management) tool consisting of a list of times at which possible tasks, events, or actions are intended to take place or a [sequence of events](http://en.wikipedia.org/wiki/Sequence_of_events) in the chronological order in which such things are intended to take place. Due to the growth of technology and its importance, many institutions now apply technology in their everyday activities (e.g. electronic examination timetable, online registration. etc.). We as group are aiming at eliminating the manual way of generating the examination timetable by introducing an electronic examination timetable generator. An electronic examination timetable is a timetable which is automatically generated by the use of software, based on the user’s inputs. Using the electronic examination timetable by the university’s examination board will portray the advancement in technology by the university.

## 1.2. BACKGROUND

The University of Cape Coast was established 51 years ago and as an institution for higher studies, it has churned out thousands of certificate holders both graduates and undergraduates. In order for the university to produce these individuals, they have to go through a number of years of study and write series of end of semester exams to prove that they fully comprehend what they have learnt. In the drawing of a timetable, most institutions including the University of Cape Coast (UCC) use the manual way. In UCC, the selected committee who is responsible for organizing and conducting this exercise is called the Examination Board. This committee uses the manual mode for creating the timetable. A lot of universities use the electronic means of generating their examination timetable. Universities like university of Warwick, City university of Hong Kong, university of Auckland; to mention a few, generate their examination timetable electronically. Since UCC generates her timetable manually, we as a group are motivated by the mentioned universities to help UCC generate her examination timetable electronically. It is not advisable for UCC to implement other university’s examination timetable generation software because the two universities might be using different examination structures (e.g. variation in time and day i.e. UCC writes her exams from Monday – Friday whiles the other university might include weekends.). Also, the electronic examination timetable generators used by the various universities may not have been acquired for free but we intend to develop it free for the university. The software will take inputs from the user which includes course name, course title, and class size. With these inputs, it will generate the timetable automatically with course name, course title, venue, time and date the examination will take place. There will also be permission restriction which will be controlled by an administrator. This means editing of the timetable can only be done by those who have the permission.

## 1.3. MOTIVATION/OBJECTIVES

Currently with the growth of technology and its adaption, most institutions are incorporating technology into their activities. With the introduction of electronic examination timetable generator, it will help reduce or eliminate some problems or difficulties encountered when drafting the exams timetable manually. The University of Cape-Coast drafts its timetable manually. The manual way of drawing the timetable has faced many challenges and has consumed many resources. Some of the challenges encountered when the examination timetable is drawn manually includes, having a large class size assigned to a smaller examination center, and also having a class of students who write two papers at the same time and on the same day. This justifies why the Examination Board presents a Provisional Time Table.

* Our goal is to develop an Electronic Examination Timetable generator for the Academic Board of the University of Cape-Coast.
* To meet the aim of our project, we have outlined the following objectives; to assign large classes to larger venues, to eliminate the Provisional timetable, to eliminate the clashing of borrowed and compulsory courses during examination, and also to ensure that any modification to the timetable is only done by authorized persons.

## 1.4. METHODOLOGY

A software development methodology or system development methodology in software engineering is a framework that is used to structure, plan and control the process of developing an information system. According to Schaum’s Outline of Software Engineering, the four most common software lifecycle models are:

* Linear Sequential Model (Waterfall Model)
* Prototyping Model
* Incremental Model
* Boehm’s Spiral Model

The chosen methodology to be used for the development of the Electronic Examination Timetable is Prototyping model with linear sequential model. This is because our system will be developed in modules. We chose the Prototyping model because it engages the client in the development process and also, it helps to avoid developing unnecessary features. The Linear sequential model allows a step by step approach to each phase, whereby each phase must be completed before you can start the next phase and there is no overlapping in the phases. After receiving a positive feedback from the client based on the prototype, we then use the linear sequential model to build the module.

The Life Cycle Activities we chose for our software development includes the following:

* **Feasibility Studies** - After carrying out the feasibility study we found out that there is a need for the software because the system is not in existence, and also because it will help to eliminate the drafting of the timetable manually.
* **Requirement Identification** – the requirements for the development of the software i.e. course code, class size, venue etc. would be obtained from the stakeholders i.e. Examination Board, Student Record and Management Information Section etc. of the University of Cape Coast.
* **Developing a module’s prototype** - Since we stated earlier that the software will be developed in modules e.g. Interface and other features, a prototype for each module is then developed based on the requirements obtained from that module.
* **Prototype Review** – A prototype will be sent to the client to assess and to give us feedback.
* **Revision or Development of Prototype –** Based on the feedback, there would be either a revision of the prototype or the development of the actual module. A revision of the prototype will be done if there is a negative feedback from the client. Else, we proceed with the development of the actual module using the linear sequential model.

The system should have certain functional requirements such as

* The user should be able to input class size to obtain an appropriate venue.
* The user should be able to obtain the generated timetable for the University.
* There would be a search field to help in the easy identification of the timetable for a particular program.
* The administrator would be the only person who can make changes.

Also there should be some technical requirements such as:

* Hardware requirement:
  + Processor: Intel ® Celeron ® CPU B815 @ I.60 GHz
  + RAM: 2.0 GB
  + System Type: 32-bit Operating System
  + Operating System: Windows 7 Ultimate
* Language:
  + HTML5 and CSS
  + PHP
  + MySQL

## 1.5. SCHEDULE AND TIMELINE

|  |  |  |
| --- | --- | --- |
| **DATE (2014)** | **ACTIVITIES** | **DURATION** |
| Jan 27 – Feb 10 | **Feasibility studies.** | **2 weeks** |
| Feb 12 – Feb 26 | **Proposal writing and submission.** | **2 weeks** |
| Feb 28 – Mar 21 | **Sketching and creating of interfaces.** | **3 weeks** |
| Mar 24 – Mar 31 | **Submission and review of Interface prototype.** | **1 week** |
| Mar 27 – Apr 3 | **Database Creation.** | **1 week** |
| Apr 4 – Apr 11 | **Submission and review of Database prototype.** | **1 week** |
| Apr 14 – May 5 | **Designing and implementation of Algorithm.** | **3 weeks** |
| May 6 – May 13 | **Testing.** | **1 week** |
| Mar 3 – May 19 | **Documentation.** | **11 weeks** |
| First Week in June | **Final Defense.** | **-** |
| 20TH June | **Submission of Project work to the department.** | **-** |

*Table1 (****N.B Gathering of information runs through the whole period.)***

## 1.6. CONCLUSION

This proposal captures an insight and a brief description of the Electronic Examination Timetable, the background, the motivation behind the development of the software and also the software engineering framework that we intend to use to structure, plan and control the process of the software development. It also captures the schedule and timeline, i.e. the time or period we intend to spend on each activity of the development process.

## 1.7. THESIS ORGANIZATION

Background and Literature study of this work will be discussed in the chapter two. Chapter three focuses on the design and implementation strategies used in developing the application. Testing, evaluation and maintenance and conclusion and recommendations will also be discussed in chapter four and five respectively.

# CHAPTER 2

# BACKGROUND AND LITERATURE REVIEW

## 2.1. INTRODUCTION

The creating of a university timetable has always been a difficult one. The recent growth in student numbers and the adoption of modular degree structures by many institutions has made scheduling of university courses and examination an even bigger problem than it used to be. University of Cape Coast for the past years has used the manual way of drafting her examination timetable and with the advancement in technology and its associated areas, many universities have moved from the manual to an electronic way of drafting their examination timetable. This transition has been inspired by many factors which include reduced labor cost and reduced time spent on generating the timetable.

If we were asked to determine the most important part of a university life, the one thing we probably would not think of is the timetable. Planning a timetable is one of the most complex and error prone applications. There are still serious problems like generation of high cost timetable which occurs while scheduling and these problems are repeating frequently. [2] Therefore there is a great requirement for an application distributing the course evenly and without collisions. The aim is to develop a simple, easily understandable, efficient and portable application which can within a second generate a good quality timetable. [3]

## 2.2. STRUCTURE OF A RELATED AUTOMATED TIMETABLE GENERATOR

The structure of a timetable generator consists of Input Date Module, relation between the input data module, time interval, time slots module, applying active rules and GA module then extract the reports.

1. **Input Data** [7]

The input data module is described by a type of data from the database. The data contains:

1. **Person:** Data describe the name of lecturers
2. **Subject:** Data describe the name of courses in the class
3. **Room:** Data describe the name of the classes and capacity of each.
4. **Time interval:** It is a time slot with a starting time and duration.
5. **Constraints**

Constraints can be divided in to three parts:

Validity violation constraints

Hard constraints

Soft constraints

1. **Validity violation constraints:** [3] There are the constraints which are needed to be incorporated necessarily otherwise there is no guarantee of valid timetables generated. They are included as part of initial generation of population, as they cannot be violated.
2. **Hard constraints:** [8] Hard constraints are the ones which need to be fulfilled necessarily. These include:

* Classrooms must not be double booked.
* A classroom must be large enough to hold each class booked to it.
* Some examination requires particular rooms.

1. **Soft constraints:** [7] These are constraints that are not that obvious but still demanding. They are not to be really satisfied but the solutions are generally considered good if large numbers of them are taken care. It includes :

* No consecutive papers.
* Courses must be evenly distributed.

## 2.3. GENETIC ALGORITHM

Genetic algorithms have proved to be very useful general purpose optimization tools that may be applied to a wide range of, often very difficult problems. Their development was inspired by the process of biological evolution. A population of possible solutions (which in this case would be timetables) are taken and rated according to how good they are by an evaluation function. The result of this evaluation on any particular population member is called its fitness. In biological model this may be how quick a rabbit can run to escape its predators. Here it would more likely be how many students are expected to take exams in two consecutive periods. A selection function is then used to choose pairs of individuals (timetables) to mate with each other. The pairs are chosen according to their fitness to produce a new population. Finally, after a number of generations, the population converges on a good solution. There are several reasons why genetic algorithms are particularly appropriate for the timetabling problem. Firstly, the evaluation and selection functions are independent from the encoding of, and method of creating new individuals. This means that the evaluation function may be defined independently by the university (or even provided by individual departments) to reflect their particular desires as to what the final result will look like.

Several universities including the University of Edinburgh already use Genetic algorithms to timetable their undergraduate courses [10]. In this case, a simple representation is used where the individual population members contain a list of when each exam is to be held. This has successfully produced timetables of a better quality than were already available. At Napier University, a different scheme is being developed where instead of specifically prescribing when an exam is to be taken; the individual is a list of instructions as to how the timetable is to be created [11].

## 2.4. CONCLUSION

In creating an examination timetable generator, there have been several key requirements which need to be addressed independent of the institution. Some of these requirements include:

* Avoid writing two papers at the same time
  + That is same class of students writing two of their courses on same day and at the same time.
  + Resit courses clashing with current semester courses.
* Assigning a class to a venue that can contain them.
* Avoid assigning two different classes to a venue that cannot contain them.

In reviewing other similar applications, they had almost all the basic requirements. Our application will incorporate an administrator feature. From the above review, it is noticed that more universities have incorporated the use of electronic (automated) electronic timetable generator and they are benefiting from it. In developing an application, one goes through the design and implementation stage where he/she structures, plan and control the process of developing the application. Chapter three discusses into details the design and implementation strategies.

# CHAPTER 3

# DESIGN AND IMPLEMENTATION

## 3.1. INTRODUCTION

A software development methodology or system development methodology in software engineering is a framework that is used to structure, plan and control the process of developing an information system. Software design and implementation are invariably interleaved. Software design is a creative activity in which you identify software components and their relationships, based on the customer’s requirements. On the other hand implementation is the process of realizing the design as a program. Design is about how to solve a problem, so there is always a design process. Design and implementation are closely linked. This chapter contains a comprehensive description of the high level architecture that is being used in the development of the electronic examination timetable. Architectural design is a creative process where you design a system organization that will satisfy the functional and non-functional requirements of a system. It is intended to form the basis of the Software Detailed Design and Implementation and it will also show the overall system design of the methodology, Database system, the user interface design and higher level module design.

## 3.2. SYSTEM OVERVIEW

The Electronic Examination Timetable Software will be a web based application with a database backend that contains the list of courses, levels that take particular courses and which program belong to which department. It also contains venues, venue sizes, class sizes, date, duration and user information. An added functionality to this software is the introduction of an administrator who will upload courses every semester, the administrator also has the authority to add, edit and delete courses in case changes have to be made to the uploaded courses. The administrator can also add, edit and delete users of this software. The system administrator will also select the period at which the examination is to take place and then generate the timetable that will be used for the examination of that semester.

The administrator interface will be built with web development languages but will function differently as compared to that of the member interface. The user interface has been designed in such a way that the administrator and other users have a uniform login page. But the login page has been structured in a manner that the access level of an administrator is different from that of other users. The administrator logs in as admin and has the authority to add, edit and delete courses, add, edit and delete users and finally select a period and generate the time table. While the other users log in as members or users and because they have a different access level they can only view the generated timetable. The Graphical user interface has been built with web development languages including, HTML5, PHP, Java Script and cascading style sheet (CSS).

## 3.3. METHODOLOGY

Sommerville defined a software process as a set of related activities that leads to the production of a software product. Software process provides a fixed generic framework that can be tailored to a specific project. It includes specific parameters involving size, cost and duration. Process models used in software development include Linear Sequential Model (Waterfall Model), Prototyping Model, Incremental Model, Boehm’s Spiral Model etc.

For the development of this electronic examination timetable software, the chosen methodology is the Prototyping model coupled with linear sequential model. We use Prototyping model because it engages the client in the development process and also, it helps to avoid developing unnecessary features. The Linear sequential model allows a step by step approach to each phase, whereby each phase must be completed before you can start the next phase and there is no overlapping in the phases. After receiving a positive feedback from the client based on the prototype, we then use the linear sequential model to build the module.

Prototyping model allows users to see how well the system supports their work. They may get new ideas for requirements, and find areas of strength and weakness in the software. They may then propose new system requirements. Furthermore, as the prototype is developed, it may reveal errors and omissions in the requirements that have been proposed.

Phases or stages used in developing this web application using these process models generally include the following:

* **Information Gathering**

This is the first step in designing a successful application or software. Many things need to be taken into consideration when developing an application. It involves collection of information that is required for the development of the software (electronic examination timetable). It involves a solid understanding of the goals, content and target audience of the application being developed.

* **Requirement Analysis**

Requirements for a system are the descriptions of what the system should do: the services that it provides and the constraints on its operations i.e. determining what functionalities the software should contain and also determining what tasks and structures are common to its development.

* **Planning**

This has to do with using the information gathered to put together a realistic plan in developing the software. It includes describing the required process, the expected product and how it should be implemented. At this stage the software designer will decide on what technologies to implement in the development of the software.

* **Design**

This represents the structure of data and program components that are required to build a computer based system. It considers the architectural style that the system will take, the structure and properties of the component that constitute the system and the interrelationships that occur among all architectural components of a system. At this stage of the development, a prototype of the system will be created. The Prototyping model engages the client in the development process. When the client approves of the prototype, further work can be carried out.

* **Development**

This is a critical stage of the developing process; at this point you create an executable version of the software. After an approval of the prototype by the client, the developer can go ahead to implement the software. The development of both the front end and the backend are done at this stage.

* **Testing**

This stage is intended to show that a program does what it is intended to do and to discover program defects before it is put into use. The first goal leads to validation testing, where you expect the system to perform correctly using a given set of test cases that reflect the system expected use. While the second goal leads to defect testing, here, the test cases are designed to expose defects.

* **Maintenance and updating**

Software development does not stop when a system is delivered, but continues throughout the lifetime of a system. After a system has been deployed it inevitably has to change if it is to remain useful. Software maintenance is the general process of changing a system after it has been delivered. The changes made may be simple changes to correct coding errors more extensive changes to correct design errors or accommodate new requirement.

## 3.4. SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide plan from which products can be procured, and systems developed, that will work together to implement the overall system.

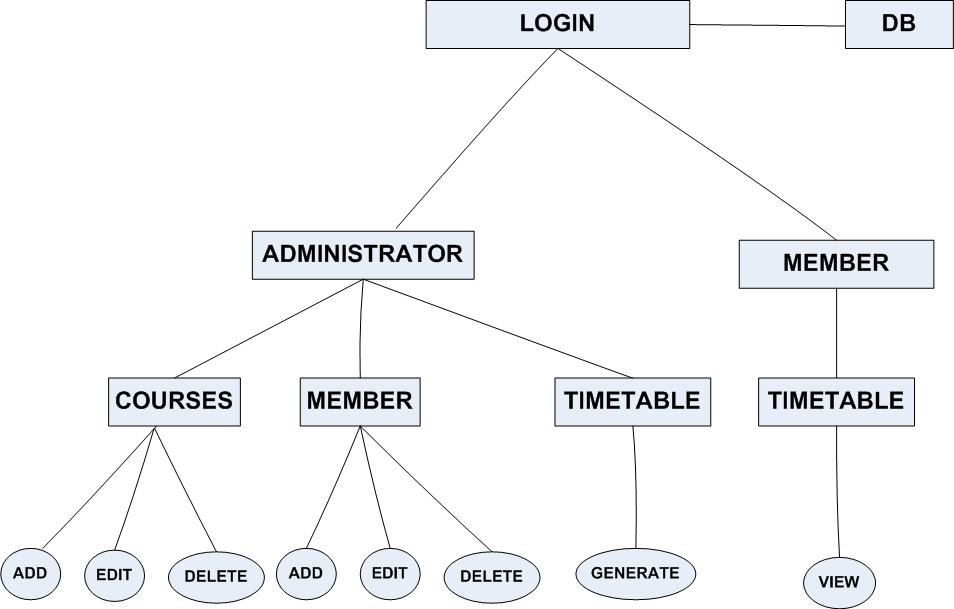
* **Architectural design**- here, one identifies the overall structure of the system, the principal components (sometimes called sub-systems or modules), their relationships, and how they are distributed.
* **Interface Design-** the designing of the interfaces between system components conducted here. This interface specification must be unambiguous. With a precise interface, a component can be used without other components having to know how it is implemented. Once interface specifications are agreed, the components can be designed and developed concurrently.
* **Component Design**- here, one takes each system component and design how it will operate. This may be a simple statement of the expected functionality to be implemented, with specific design left to the programmer.
* **Database Design** – at this stage, one designs the system data structures and how these are to be represented in a database. This depends on whether an existing system needs to be reused or a new database to be created.

## 3.5. STRUCTURE OF THE ELECTRONIC EXAMINATION TIMETABLE

The electronic examination timetable software consists of various architectural styles due to its scope. The architectural design styles used in this system include object-oriented and Database-centric architectural styles.

This figure below shows, in simple terms, how the system will be implemented and the high level architecture it requires.

## 3.6. MODULES PURPOSE AND FUNCTIONALITY

****This system consists of several modules that helped and aid the development and design of this software.

***Figure1***

### 3.6.1. Database

The database stores all information or data pertaining to courses, venues, level, program, administrator information and user information. The data is managed by MySQL server. The database module includes the MySQL database, the connection and the SQL statements.

**3.6.2. Login/Authentication**

This module displays a uniform login screen for both the administrator and the other users. Where the user is directed to after login shows or explains their access level or authentication. It also includes the functionality that authenticates them and grants them access to their respective modules. The administrator can exhibit more features as compared to the other users or members due to the higher authentication level.

* + 1. **Administrator Module**

This module allows the administrator of the system to create user accounts, edit and delete users, and also generate timetable. This can be done by importing spreadsheets containing the courses that have been registered for in a particular semester. The administrator also has the privilege to add, edit and delete courses.

**3.6.4. Member Module**

This module allows users after login to view the generated timetable and also to search for particular programs or courses in order to know what venue, date and time the exams has been fixed for.

**3.6.5. Search and Lookup**

This module provides all the functionality to allow searching and looking up of specific departments, programs and courses to determine the date, time and venue of an exam. This module could also help the administrator in searching for members or users information that has been previously added or created.

## 3.7. USER CLASSES AND CHRACTERISTICS

### 3.7.1. Administrator

Administrator has full administrative right to the system and all other management activities. The administrator will be able to import spreadsheet of courses registered for the particular exam of a semester. He can also add and delete users, and also edit information about a particular user. The administrator can also add and delete courses from the uploaded spreadsheet. The administrator will most likely be a head in the examination board of the University of Cape coast.

**3.7.2. Members**

The users of this system will be added by the administrator. They are most likely faculty heads or department heads. They will be able to view the generated examination timetable and also be able to get a subset of the generated timetable based on the faculty or department. The subset of the generated timetable would now be made available to students in that department or faculty.

**Administrator**

|  |  |
| --- | --- |
| Actor | Administrator |
| Summary | For an administrator to access the system he needs both username and password to access various pages of the system. |
| Flow of Event | 1. Logs in at the login section. 2. With this access level the administrator can view the home page. 3. The administrator then selects the time frame for conducting the exam and then generates the timetable. 4. Administrator can also upload the courses into the system. In addition to that, he can also edit and delete courses. 5. Administrator can also add, edit and delete members allowed to use this system. |

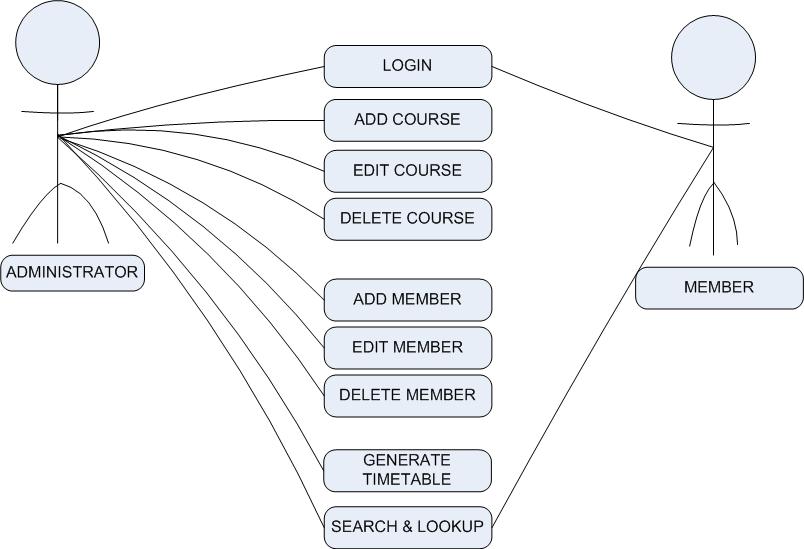
*Table2 (Use case scenarios for Administrator).*

**Members**

|  |  |
| --- | --- |
| Actor | Members |
| Summary | Members that use this system can login using their username and password, as long as they have been added by the administrator. |
| Flow of event | 1. Logs in at the login section. 2. With the access level granted to the member of this system, the member will be directed to the timetable page of the system. 3. This page contains the timetable that has been generated by the administrator. The members can only view, search and lookup courses pertaining to him or her. 4. Members can also reset passwords for security purpose. |

*Table3 (Use case scenario table for Members.)*

## 3.8. USE CASE DIAGRAMS FOR ELECTRONIC EXAMINATION TIMETABLE

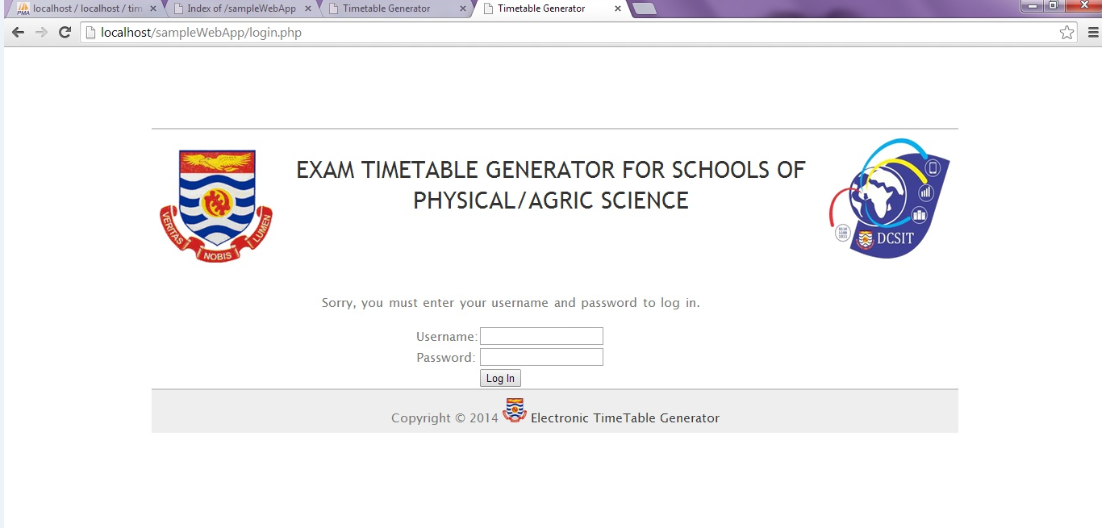
****

*Figure2 (Electronic Examination Timetable Use Case Diagram.)*

## 3.9. EXTERNAL AND INTERFACE REQUIREMENT

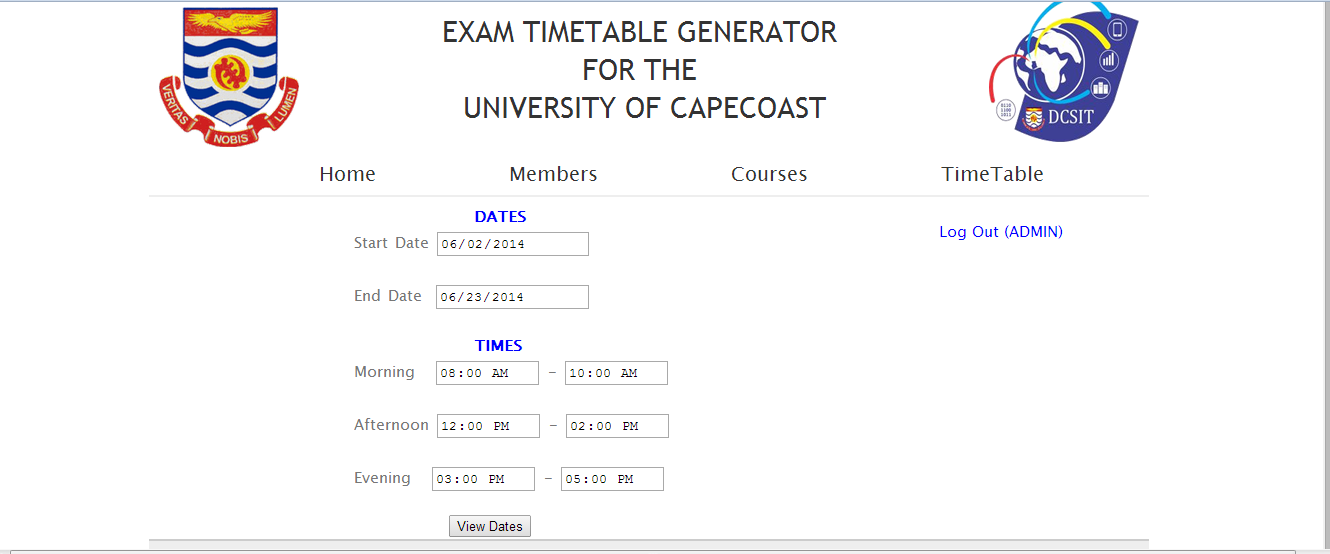
### 3.9.1 User interface requirement

The user interface has been designed and presented to members as a web based application using web technologies such as PHP, HTML 5 and CSS. The application starts with an index page, with a user login interface that requests for both username and password. After being able to login the user is redirected to an interface based on level of access to the application.

* **Administrator interface**
  + **Index page:** This is the login page for all levels of users who are to login to gain access to the various page interface in the system. The administrator and the members have a uniform login interface. Hence, depending on the access level (administrator, member) the user gains access to the next interface of the system.

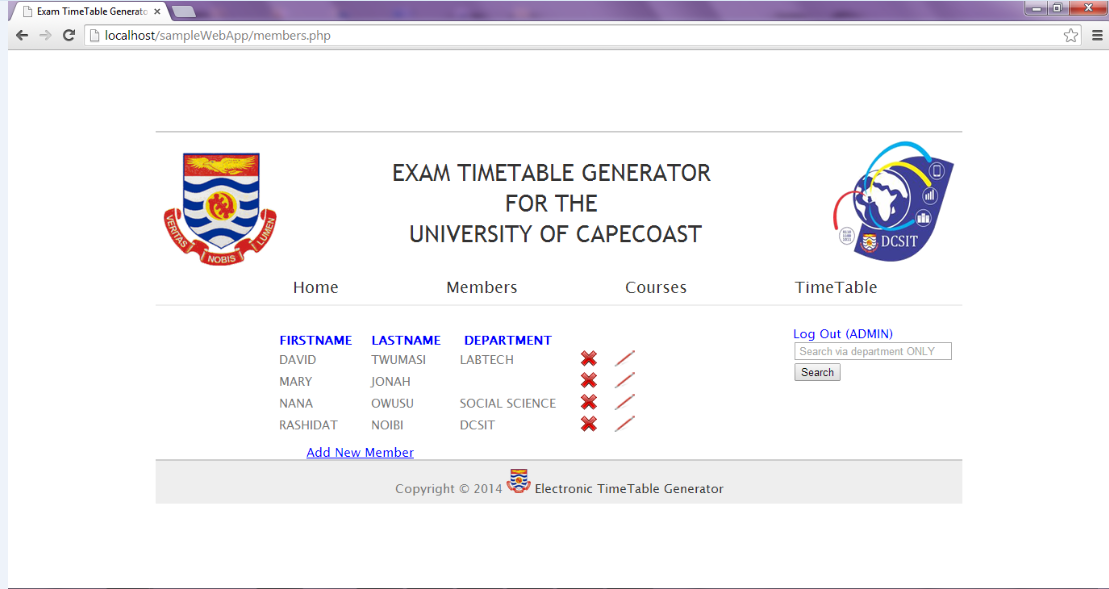
*Figure3 (User interface for the index page*.)

* + **Home page**: This interface allows the administrator of the system to perform system wide functionalities. The administrator can navigate to gain access to other pages from this page. From this page, the administrator selects the time frame for the exam to be conducted. From this page, the Admin can click view date to display the dates that exams will be conducted with attributes to delete particular dates before generating.



*Figure4 (Homepage for Admin)*

* + **Members’ page**: this is the page that lists the added members that can use the system. The administrator can also add new members from this page, using the ‘**add new member’** link. The administrator can also edit member information and also delete members from this page. There is also a search bar to lookup specific members by typing their names or department.



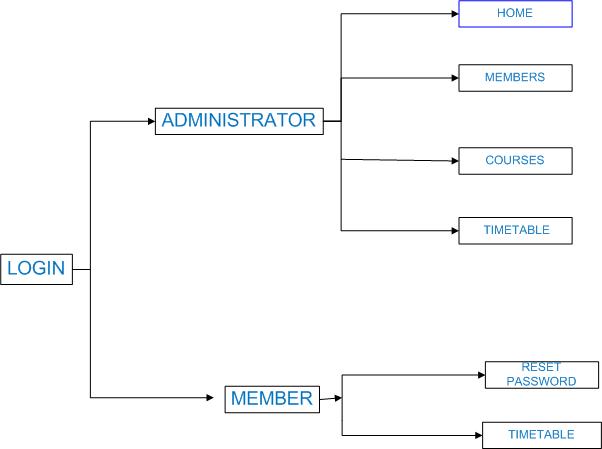
*Figure5 (Member page interface)*

* + **Course page:** on this page the administrator uploads courses using a spread sheet containing courses and their class sizes including the program and department. The administrator can also add, edit and delete courses on this page. There is also an added feature which is helpful in case the administrator needs to search for a particular course.



*Figure6 (Course page)*

### 3.9.2 Site Map



*Figure7 (sitemap)*

### 3.9.3 Database Design

Database design is the process of producing a detailed data model of database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes named relationships. However, the term database design could also be used to apply the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system.

Innumerable types of databases exist for use by either developers or casual users. Keeping various data on the project being worked on, the administrators and the members of the system requires the use of a database. The requirement analysis undertaken shows that the data being stored ought to be devoid of any data redundancy but should be highly related as well. For a perfect relational database design, the researchers used the seven-step algorithm for converting the conceptual design to relational database.

Relational database design model was first introduced by Edgar Frank “Ted” Codd. An English computer scientist who, while working for IBM, invented the relational model for database management, the theoretical basis for relational databases in 1970 in a classic paper (Codd 1970), and attracted immediate attention due to its simplicity and mathematical foundation.

A relational database organizes data in tables (or relations). A table is made up of rows and columns. A row is also called a record (or tuple). A column is also called a field (or an attribute). A database tuple is similar to a spreadsheet. However, the relationships that can be created among the tables enable a relational database to efficiently store huge amount of data, and effectively retrieve selected data. These relationships between tables are defined by the use of primary keys and foreign keys and should be in appropriate normal forms. The relational database tables sometimes suffer from integrity, performance and maintenance. Its major drawback is redundancy. Tables in a relational database may be badly formed with duplicate data that exist in other tables. In order to eliminate insertion, update and deletion anomalies, they should be normalized to make the database operations more robust.

Database normalization is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining the relationships between them. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database using the previously defined relationships. Different normal forms exist including first, second, third, fourth and Boyce-Codd normal forms respectively. The Boyce-Codd normal form is a special type of third normal form.

### 3.9.4 Data Dictionary

The dictionary lists all the tables of the Electronic Examination timetable database. It describes the data contained in each of the tables.

**tbl\_Course**

the course table contains information about the course, courseId, coursecode, name of course, level taking the course, no of students for each course, programme name and department name.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Type | Length | Null | Comment |
| CourseId | Int | 11 | No | Auto increment. Primary key |
| CourseCode | Varchar | 10 | No | Uniquely identifies every course |
| Name | Varchar | 255 | No | Name of each course |
| Level | Int | 4 | No | Student levels |
| No\_of\_students | Int | 255 | No | Number of students taking a course |
| Programme  Deparment | Varchar  Varchar | 255  255 | No  No | Name of programme  Department Name |

*Table4 (Course table)*

Tbl\_Venue

The venue table contains information about the venues, venuecode, venuesize, venuename, venueId.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Type | Length | Null | Comment |
| VenueCode | Varchar | 50 | No | Uniquely identifies every venue |
| VenueSize | Int | 255 | No | Indicates the number of students that can fit into a class |
| VenueName | Varchar | 255 | No | Name of the venue |
| VenueId | Int | 255 | No | Auto increment to number venues |

*Table5 (Venue table)*

Tbl-Date

The date table contains information of every date of an examination, DateId, Date, and Day.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Type | Length | Null | Comment |
| DateId | Int | 255 | No | Auto increment every date specified |
| Date | Varchar | 50 | No | Specifies every date |
| Day | Varchar | 10 | No | The day of each date. |
|  |  |  |  |  |

*Table6 (Date table)*

Tbl\_Duration

The duration table contains information on the duration of examination.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Type | Length | Null | Comment |
| DurationId | Int | 255 | No | Auto increment. Uniquely identifies every duration. |
| Duration | Varchar | 255 | Null | The duration of each exam |

*Table7 (Duration table)*

Tbl\_login

The login table contains information about every login detail, loginId, username, and password.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Type | Length | Null | Comment |
| loginId | Int | 10 | No | Auto increment. Uniquely identifies each login |
| Username | Varchar | 50 | No | Signifies the name of the user who logs in. |
| Password | Varchar | 45 | No | Password of every user |
|  |  |  |  |  |

*Table8 (Login table)*

Tbl\_users

The users table contains information about every user, userId, Username, and password.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Type | Length | Null | Comment |
| userId | Int | 50 | No | Uniquely identifies every user |
| Usernaeme | Varchar | 100 | No | Signifies the name the user uses to login. |
| Password | Varchar | 100 | No | Every user needs password as a form of authentication |

*Table9 (Users table)*

**Tbl\_finaltimetable**

The finaltimetable table contains information on level, programme, FinalId, courseId, Date, Duration, VenueCode and department.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Type | Length | Null | Comment |
| Level | Int | 30 | No | Levels in every department. |
| Programme | Varchar | 255 | No | Programmes in every department |
| FinalId | Int | 11 | No | Auto increment every timetable generated |
| CourseId | Varchar | 255 | No | Identifies every course |
| Date | Varchar | 50 | No | Date of each examination |
| Duration | Varchar | 255 | No | Identifies each duration of an exam |
| venueCode | Varchar | 255 | No | Identifies each venue |
| Department | Varchar | 100 | No | Identifies the department of the exam for the date and duration. |

*Table10 (Finaltimetable table)*

# CHAPTER 4

# TESTING, EVALUATION AND MAINTENANCE

## 4.1 INTRODUCTION

In this chapter, we will focus on the testing and maintenance procedure for the Electronic examination timetable generator. This chapter also contains the chosen testing methodologies and testing approach we used to ensure that the application was functioning well before the final release.

## 4.2 ORDER OF TESTING

We conducted series of testing to ensure that the application (Electronic examination timetable generator) accomplishes all its tasks. The first test we conducted was unit testing. Under this type of testing, we tested the various pages independently to ensure that they were achieving their individual objectives. The main motive behind this test is to check if a page can function without depending on the other pages. This helped in identifying the bugs or errors in each of the pages. The other test conducted was the integration testing where all the modules were combined and tested as a group. This was done to verify functional, performance and reliability requirements placed on the individual modules brought together. This was done to ensure that the various links lead users to the right modules (web pages).

We proceeded with validation testing where we checked to ensure that the application met the specifications and that it fulfills its intended purpose.

## 4.3 RESOURCE REQUIREMENTS FOR TESTING

### 4.3.1. Hardware Requirement

Electronic examination timetable generator is a web-based application and will run on any computer having JavaScript enabled web-browser that supports HTML5.

A server will be required for testing purposes and it is on the server the various web pages will be hosted on. A live internet connection will be needed to access the server to request for pages.

### 4.3.2 Software requirement

The basic software requirement for the electronic examination timetable generator is the use of an internet browser to make server requests, display results from server requests and also interact with all components of the system. The browser that supports the electronic examination timetable generator based on testing is Google chrome due to some features of HTML5 which other browsers don’t support.

## 4.4 TESTING STRATEGY

The testing of the electronic examination timetable generator was conducted by the project group members. We individually tested the various parts of the system by applying the test cases to the areas they were applicable.

## 4.5. TESTING METHODOLOGY

This testing process is designed to verify that the electronic examination timetable generator works as expected and that it meets the software requirement specifications.

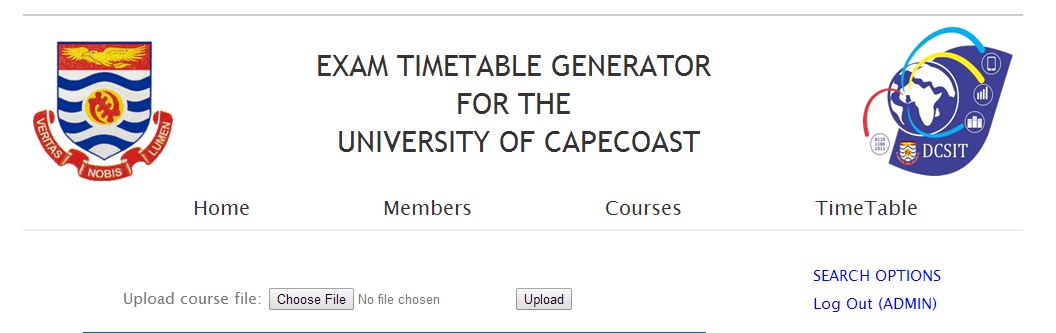
This testing also seeks to validate that the electronic examination timetable generator satisfies the needs of the examination board. Hence, this testing is an integral part of the project quality assurance.

## 4.6 TESTING PROCESS

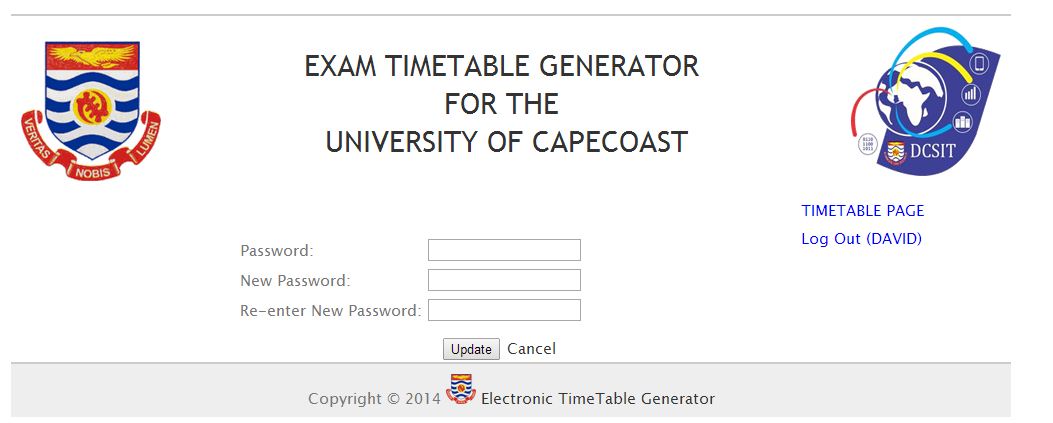
### 4.6.1. Functionality Testing

This type of test bases its test cases on specifications of the application under test. We tested the functions by feeding the system with input to examine the output. Testing activities carried out on the electronic examination timetable included:

* Ensured that all links on the various webpages were working correctly and that the various links lead to the appropriate webpages. The tested links includes:
  + **Internal links** - these are the links that links users to other resources on the electronic examination timetable webpage based on their privileges. We as a group tested the application by logging in as H.O.D and also as an admin to check if all the respective links assigned to the above users were working correctly.



*Figure8* (*Admin’s internal links (excluding the two buttons))*



*Figure9 (HODs internal links)*

* + **Anchor links**: these are the links on the electronic examination timetable generator that lead users to resources on the same page. We logged onto the system and tested all the anchor links and had no errors to report.
* Tested all forms to ensure that they were working as expected. Some testing here included:
  + **Scripting checks**: An example is when a user refuses to fill a required field on form and gets an error message. This testifies that all message prompts related to this are working correctly.
  + **Check default fields**: We tested to ensure that all default fields were working correctly upon form load. E.g. date and time.
  + **Message prompts**: We checked to see if success or failure messages are shown at appropriate time based on successful submission, loading or deletion.
* HTML and CSS testing
  + **Syntax errors checking**: We conducted this check to detect any bad rendering of the webpages in relation to the HTML and CSS syntaxes. Aside Google chrome, the other browsers failed to run the application as desired due to some HTML5 features.
  + **Readable Color Schemas**: This test was conducted to ensure that the color schema used can be easily read on various screens. We used w3school’s colors which enabled us to conform to the standard of color to use on webpages. This made our color schema readable.
  + **Compliance to Standard**: We did this test to check if the site is in compliance with any of the web standard bodies. We conducted a test during which W3C HTML and CSS validator was run and the result that was shown was very satisfactory.

### 4.6.2. Usability Testing

Though the electronic examination timetable generator was developed to solve a problem, it is not supposed or intended to introduce the user to new problems or extensive tasks. This calls for the system to be user friendly or usable hence we conducted usability testing. This was one of the most important activities conducted by the group. The various areas we concentrated on are as follows:

* System navigation testing
  + We tested all the buttons and links on the various pages to check if they were visible and consistent. After conducted this test, the result proved that the entire system has familiar look and feel.
* Site content testing
  + We checked for content legibility, spelling and grammatical errors. After this check there were some few errors we had and the appropriate corrections were made.
  + We also made provision for cases where images are unable to load due to bad internet connection or when image display is disabled. In such cases, the image is replaced by the text “alt”. We tested this by disabling the loading of images and the result proved that all images were replaced with the text “alt”.

### 4.6.3. Interface Testing

At this stage of the testing process, we tested the Application and Database Server.

* Application:

We conducted a test at this stage to check if all requests or inputs from users are sent correctly to the database and the output are displayed in a format that is legible and readable by user. Appropriate prompt messages were displayed to inform users about the success or failure of their activities.

* Database Server:

The database connections were tested to ensure that its connection with the application is always consistent and responsive once the application is running. We tried to populate the database using the interface and it was successful which proved that the interface connection with the database works correctly. We also performed some queries to test if the database provided the expected and exact results. This was also to ensure that the integrity of the data is maintained.

### 4.6.4. Database Testing

This type of testing was a critical aspect of the testing process and so it was carried out carefully. Testing activities we conducted here includes:

* We tested to check if the integrity of the data in the database is maintained while creating, updating or deleting any data. This test was conducted to point out any integrity violation such as primary key violation. We tested to ensure that such violation was not committed when creating, updating and deleting data in the database. Some of these violations were dealt with during our initial form checks.
* A test to check if errors are shown during queries and population. The test sought to check if during the query and population procedures there were errors encountered and also the way the errors are rendered.
* We again tested to ensure that data retrieved from the database is accurately displayed on the interface. This test checked for the correct rendering of the results from the database queries. After the test, the results proved satisfactory due to the HTML5 and CSS syntax strictly for correct rendering of the application’s web interface.

### 4.6.5. Compatibility Testing

This type of testing helps in ensuring that the application runs on the right platform, condition and environment. This test is intended to help the contents of the application to be displayed correctly and also ensure that some functionalities work perfectly or as required. The series of tests conducted here includes:

* **Browser compatibility test**:

Same webpage will be displayed differently on different browsers so this drew our attention to test the browsers that are compatible with the electronic examination timetable generator. For that reason, we conducted our test using Internet Explorer 8, Google chrome, Mozilla Firefox and Opera browsers. The results we gathered from the test proved that it was only Google chrome that could actually display contents and also supported some new HTML5 features e.g. date and time features. On other browsers, contents were scattered and disorganized. Also, some new HTML5 features were not able to be displayed.

* **Operating System compatibility:**

After testing the electronic examination timetable generator on browsers and with the results we gathered, it was right for us to test it on different operating systems. In doing so we tested the application on operating systems such as Windows XP, Windows 7 and Windows 8. The application was able to run successfully on all the operating systems hence made us draw the conclusion that the basic requirement is the browser which is Google chrome.

### 4.6.6. Security Testing

Considering a system like the electronic examination timetable generator which is used to schedule for examination, data that goes and comes out of the system are very sensitive and can change the whole university’s activities and calendar. This calls for ensuring that only authorized users get access to, make changes or delete certain data. The testing activities under this stage included:

* Testing of unauthorized access to administrative pages:
  + We tried to by-pass the log in page by typing in the URL of the admin page but we were denied access because we could only gain access by passing through the log in page.
  + We also ensured that the other users (Head of departments (H.O.Ds)) are only permitted or given the privilege to view contents they are allowed to view. We tested to ensure that none of the admin links or pages appeared on H.O.Ds end. Initially we encountered a similar problem but due to the testing process, we were able to detect and correct such issues.
* Administrative rights solely to the administrator to upload new courses, edit and delete them, specify examination dates or duration and also set times for writing of examination. These activities are very crucial and are only to be carried out by authorized users i.e. Admin. We tested the system and the results proved that it was only when one logs onto the system as an admin before he/she is able to conduct the above activities.
* We also tested all the reset password links and the results proved that they were all working as required. This helps the user to reset his/her password when necessary. We also tested the delete member buttons to ensure that a user from whom his/her privilege is taken from can no longer log onto the system. Our results showed that all these links and buttons are working perfectly.

## 4.7. DEPENDENCIES

### 4.7.1. Hardware Dependencies

Hosting the application will require the server to be running and accessible during testing or accessing stage.

### 4.7.2. Software Dependencies

The database server which is wamp server should also be running during testing. The exported database needs to be imported before testing the system.

## 4.8. MAINTENANCE

The electronic examination timetable generator will be maintained on demand in order to satisfy the examination board. Changes like uploading new courses, changing examination time, changing, deleting or updating dates will be done by the administrator. Addition of new features can be requested by the examination board and the group will be ready to add such features.

# CHAPTER 5

# CONCLUSION AND FUTURE RECOMMENDATION

## 5.1. INTRODUCTION

In this chapter, we will try to summarize all the chapters we elaborated on. We will also write about our proposed recommendations to help in improving the web application electronic examination timetable.

## 5.2. CONCLUSION

This project team chose the electronic examination timetable generator after analyzing and witnessing the problems both the examination board and students go through during examination periods. We also highlighted on our goal and objectives.

The light and background study of our system was also discussed in the second chapter. It helped the project team in understanding and knowing more about the system by studying similar works. This motivated the project team to try and also help this university in solving similar problems solved in other universities and countries.

The design and implementation (methodology) or the development approach used to solve the problem by the project team was discussed in the third chapter. The development of the various modules was also highlighted in that chapter.

In order to ensure that the final application which is the electronic examination timetable generator would work always, we conducted series of testing and evaluation. This was effectively discussed in the chapter four and all the test order, process, etc. were discussed. The maintenance plan was also discussed in that same chapter.

## 5.3. RECOMMENDATION

Due to several constraints among which includes short duration or time frame that was given to the project team to complete the or get a working system, we were not able to add certain key features which needed time and attention in order to completely finish them before adding them to the system. Below are the proposed future recommendations from the project team:

* Borrowed courses

Students are allowed to borrow course (s) from other department and during examination period, these students are not supposed to write the borrowed courses on the same day and time they write their departmental courses. This becomes extremely difficult when students are permitted to choose their own courses.

* Clash of Resit and current semester courses
  + Some students fail in certain courses and are made to resit for them and this application during scheduling needs to avoid the clashing of resit course (s) and any of their current semester courses.
  + The application during scheduling can assign both the resit and current semester papers on the same day but should be at different times.
* Assuming a class is larger than all venues available, the programme should be able to allocate smaller classes to that particular class without any problems.

# REFERENCES

[1] L.V. Kantorovich: A new method of solving some classes of external problems, Doklady Akad Sci USSR, 28, 1940, 211-214.

[2] Alberto Colorni, Marco Dorigo “A Genetic Algorithm to solve the timetable problem” Available: http://citeseerx.ist.psu.edu

[3] Evaggelos Nonas, Alexandra Poulovassilis, “optimization of active rule agents using a genetic algorithm approach pdf”

Available: <http://ebookbrowse.com/optimisation-of-active-rule-agents-using-a-genetic-algorithm-approach-pdf-d381872402>

[4] J. J. Grefenstette, editor. Proceedings of the First International Conference on Genetic Algorithm and their Applications. Practice and Theory of Automated Timetabling VI Proceedings of the 6th International Conference on the Practice and Theory of Automa.

[5] J. J. Grefenstette, editor. Proceedings of the Second International Conference on Genetic Algorithm and their Applications. Practice and Theory of Automated Timetabling VI Proceedings of the 6th International Conference on the Practice and Theory of Auto.

[6] N. R. Jennings. Coordination Techniques for Distributed Artificial Intelligence. University of London Mile End Rd. London EI 4NS UK, 1995.

[7] Leon Bambrick Supervisor Dr. B Lovell

“Lecture Timetabling Using Genetic Algorithms”

Available: <http://secretgeek.net/content/bambrilg.pdf>

[8] Sanjay R. Sutar, Rajan S. Bichkar

“University Timetabling based on Hard Constraints using Genetic Algorithm”

Available: <http://research.ijcaonline.org/volume42/number15/pxc3877964.pdf>

[9] CARTER M. W. 1986 A Survey of practical Applications of Examination Timetabling Algorithms OR Practice 34, 193-202.

[10] CORNED D., FANG H. S. and MELLISH C. Solving the Modular Exam Scheduling Problem with Genetic Algorithms Proceedings of the Sixth International Conference of Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh, 1993

[11] PAECHTER B., CUMMING A., LUCHIAN H. and PETRIUC M. Two Solutions to the General Timetable Problem Using Evolutionary Methods IEEE WCCI 1994

[12] David A. Gustafson (2002). Schaum’s Outline of Theory and Problems of Software Engineering. The McGraw-Hill Companies Inc., 0-07-140620-4.

# APPENDIX

# USER MANUAL

## 1.1 GENERAL INFORMATION

The electronic examination timetable software is an application that is used to generate a timetable for the University of Cape Coast. The application accepts some inputs from the administrator and then it generates the timetable for a certain period (dates and time) for exams. This timetable generator has been designed mainly in order to eliminate the manual generation of timetable and also to eliminate provisional time table.

### 1.1.1. System Overview

Electronic examination timetable application has been designed in such a way that it allows information about venues, class sizes and courses to be added to the system. The system provides an electronic version of the time table in the sense that there is no manual generation done. An administrator of the system can import courses that have been saved in a comma separated values (.csv) file, for every course in this file there is a course code, course name, level, number of students, programme and department. This application saves the .csv file uploaded into the database. The administrator can also select a start date and end date period for examination and also select time to start and end each exams. The administrator will then proceed to generate the timetable. Another added feature is adding of members to the system. Certain access level is given to members by the administrator. These members have the ability to view the generated timetable, search and lookup certain departments or courses.

### 1.1.2. Organization of the manual

The user’s manual consists of five sections: General Information, System Summary, Getting started, using The System, and Reporting.

General Information section explains in general terms the system and the purpose for which it is intended.

System Summary section provides a general overview of the system. The summary outlines the uses of the system’s hardware and software requirement, system’s configuration, user access levels and system’s behavior in case of any contingencies.

Getting started section explains how to get The Electronic Examination Timetable and install it on a device. This section presents briefly system menu.

Reporting section describes in what way information collected by the application are presented and how to access the information.

## 2.1. SYSTEM SUMMARY

System summary section provides a general overview of the system. The summary outlines the uses of the system’s hardware and software requirements, system’s configuration, user access levels and system’s behavior in case of any contingencies.

### 2.1.1. System Configuration

Electronic Examination timetable is a web based application that runs on a windows XP, windows7 and windows8 operating system. The application requires a connection to a wamp server in order to save data to the database. Also for an administrator to upload courses into the system a comma separated value (.csv) file should be used to save the courses before it can be uploaded. Data saved on the database can be seen using any major internet browser (chrome, Mozilla Firefox). Import the “timetable” file into the database to access the database.

### 2.1.2. User Access Levels

There are two main users of the Electronic Examination Timetable software i.e. the administrator and the members (H.O.Ds). The administrator has a higher access level as compared to that of the members. The administrator can add users (members), upload courses, select start date and end date for exams, and also generate the timetable. While the members can only view the generated time table, the members can also change their passwords by resetting passwords. It is only the administrator that can save data to the database. Also, only the administrator and registered or added members can login into the system.

### 2.1.3. Contingencies

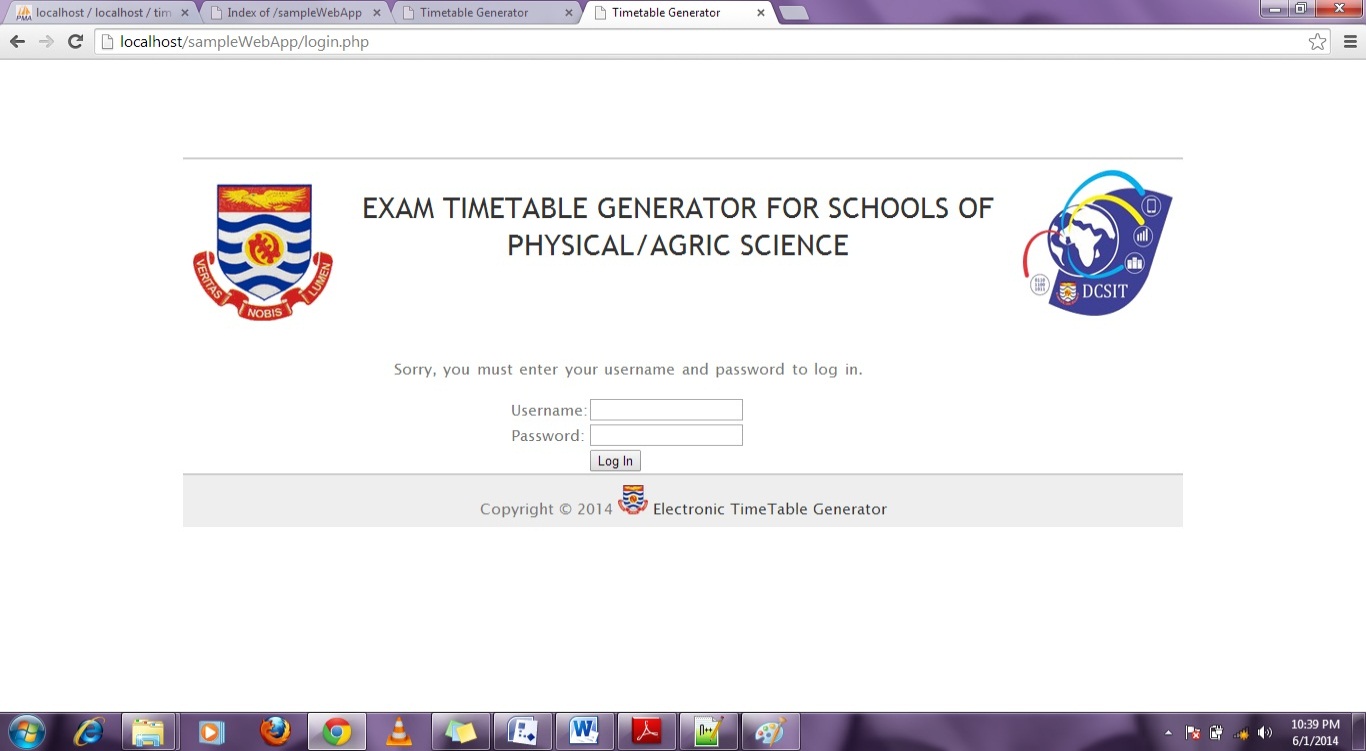
Only data stored in a comma separated value (.csv) file can be uploaded into the system. Also, the system will not save data into the database if the system is not connected to the WAMP server.

## 3.1. GETTING STARTED

This section explains how to get the Electronic Examination Timetable running on the system, how to login and also how to navigate the system. The section presents briefly system menu. The folder “sampleWebApp” needs to be copied into the www folder in the wamp folder.

Installation and logging in

Logging in can be done by the administrators and also members. To login the system requires login in of the correct username and password. If the username or password is not correct the user cannot login into the system.



*Figure10 (Login Page)*

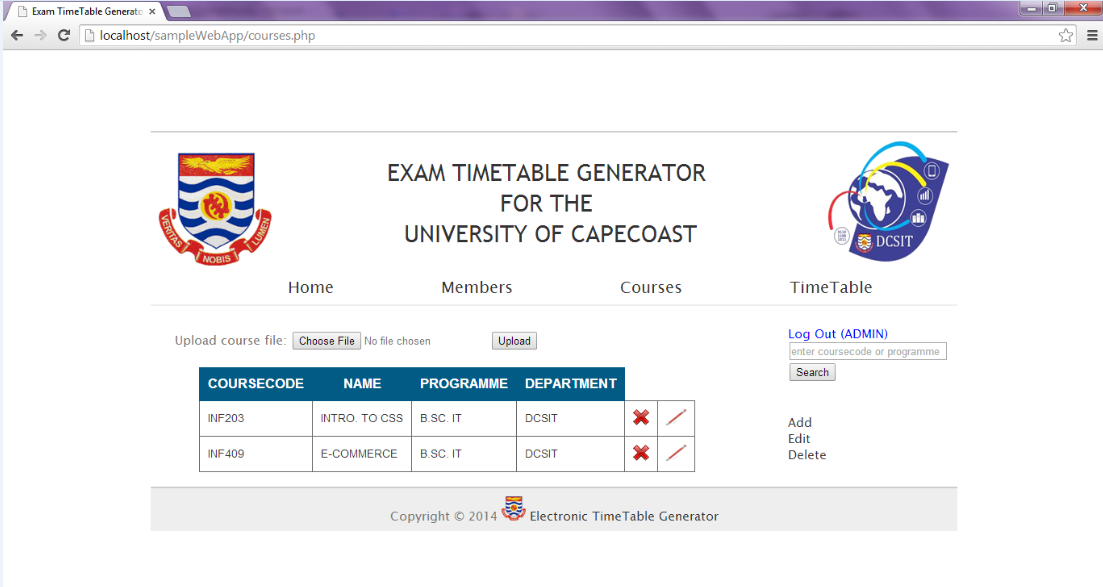
The administrator and the members can login from this page using correct username and password.

### 3.1.1. Administrator

As an administrator you can view the following pages;

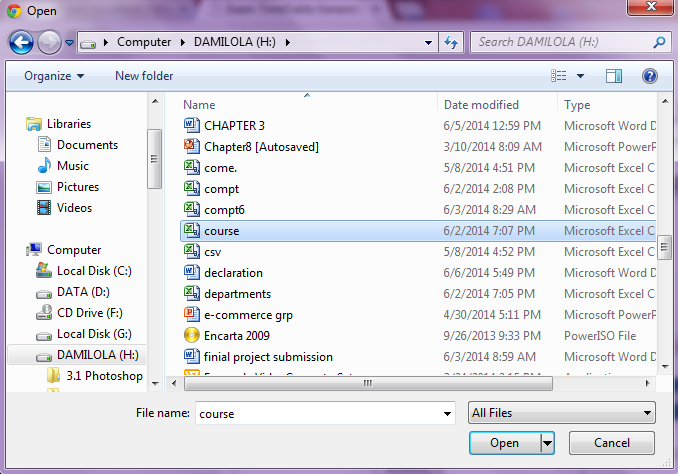
**Course page**

After login into the system the administrator can view the course page where he uploads new course information that has been stored in a .csv file.



*Figure11 (Course Page)*

After clicking the choose file button you are directed to choose a file which contains the course information that has been stored in a .csv format, you then click on open and again click on upload. The uploaded file overwrites the previous file.



*Figure12 (Course upload dialog box)*



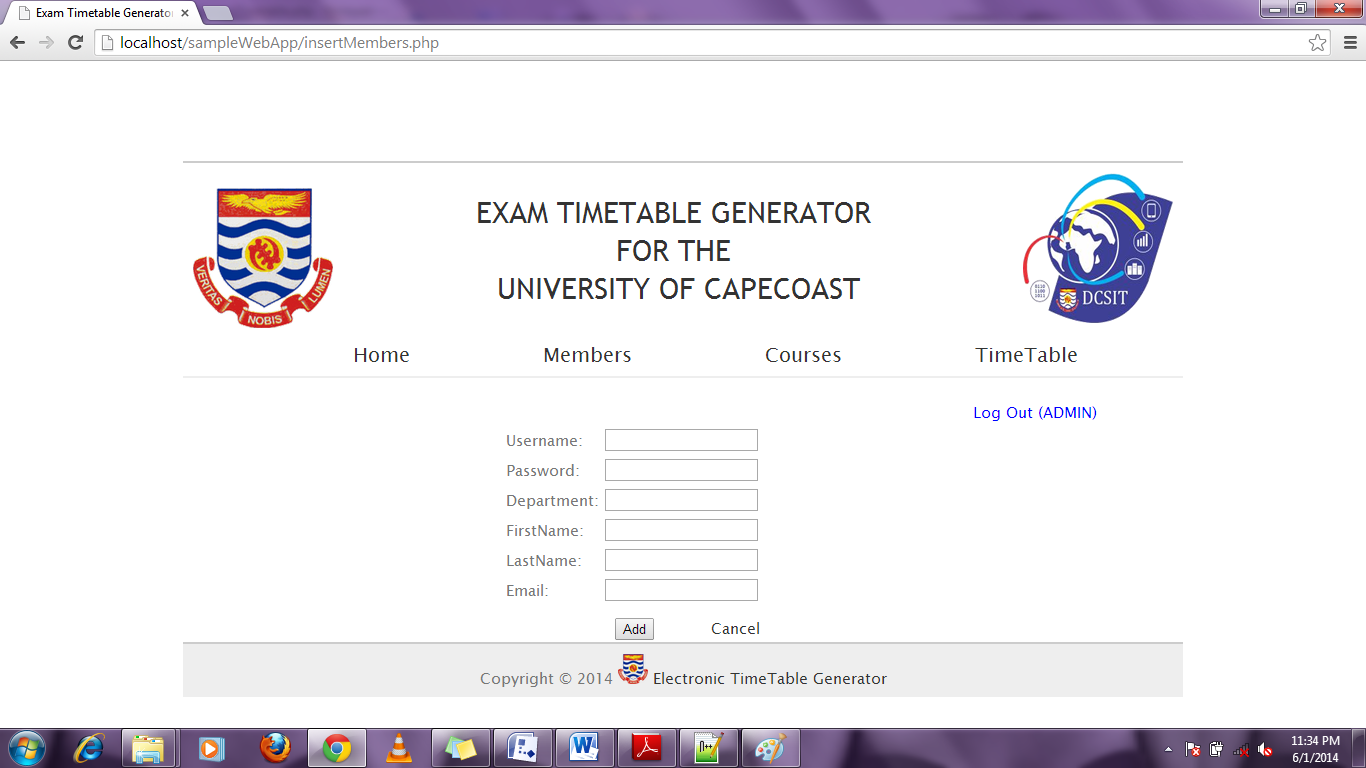
*Figure13 (Course Page)*

****

*Figure14 (Course Page)*

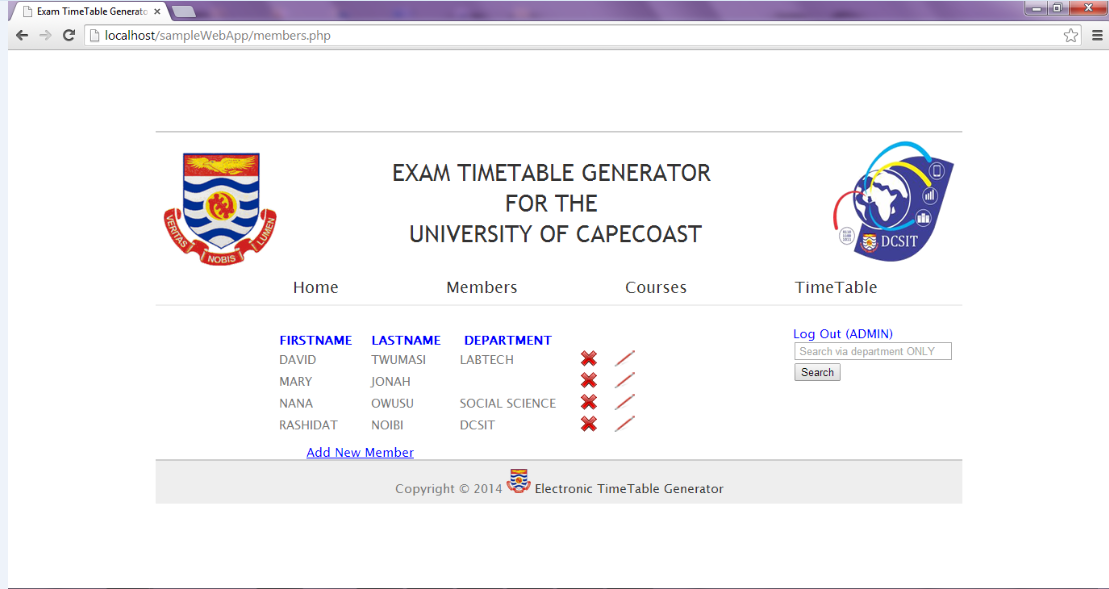
* Members Page

The administrator adds members to the system by clicking on the “add new member” link. The link directs the administrator to this page:



*Figure15 (Add Member Page)*

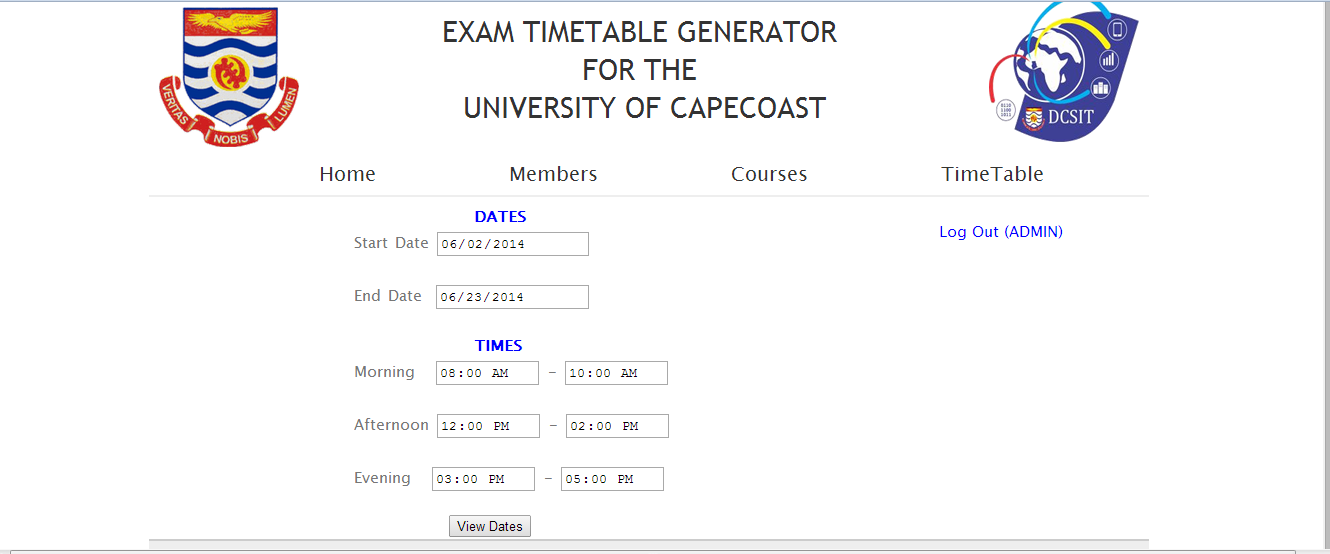
The administrator can add and delete members by clicking on the cancel and edit images respectively.



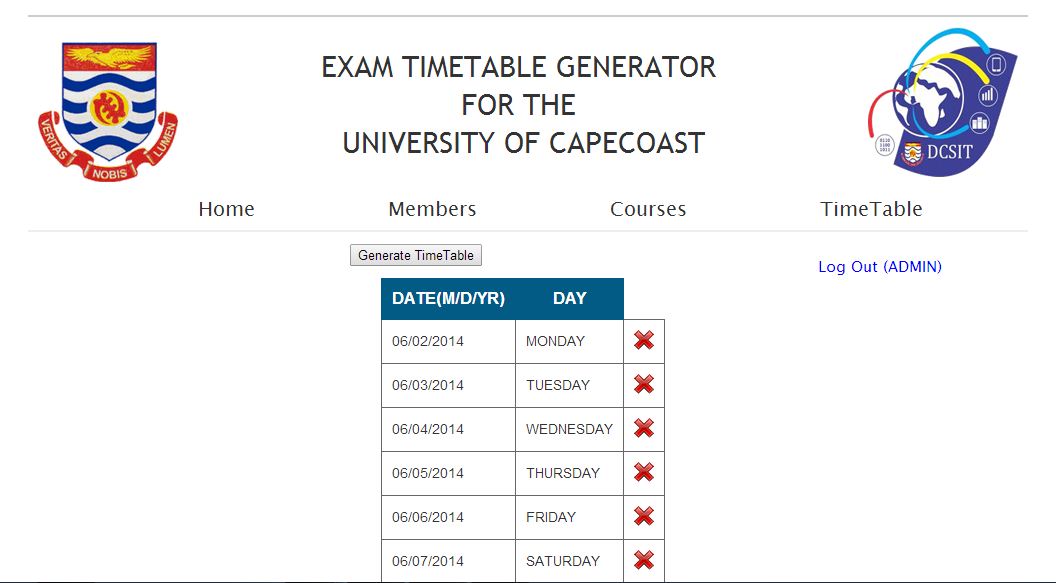
*Figure16 (Delete and Edit Member details Page)*

* **Home page**

Here, the administrator selects start date and end date for the examination he also selects the time periods for the exams.

*Figure17 (Admin Homepage (Date and Time))*

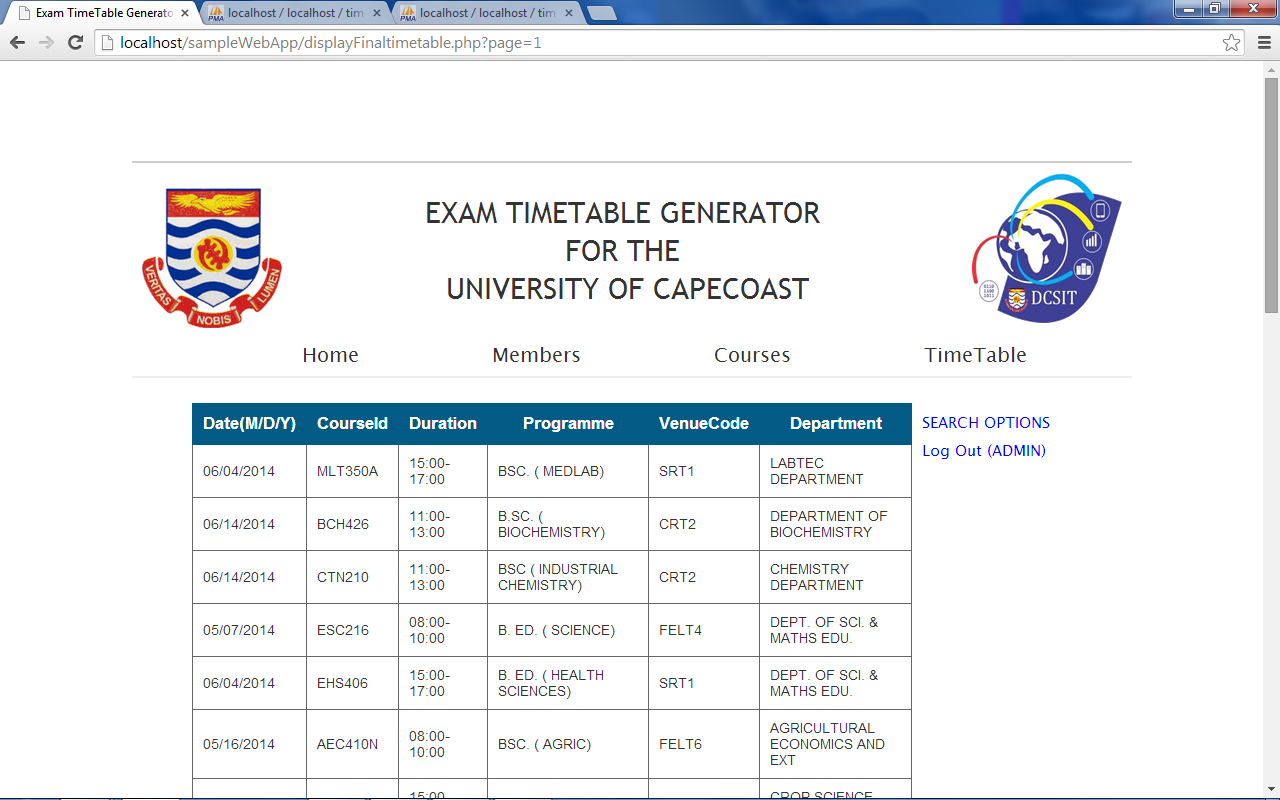
After clicking on view dates, the administrator can view the dates with days of exams and he can also edit and delete days not wanted, for example weekends and or public holidays. Then you click on generate to generate the timetable.

********

******

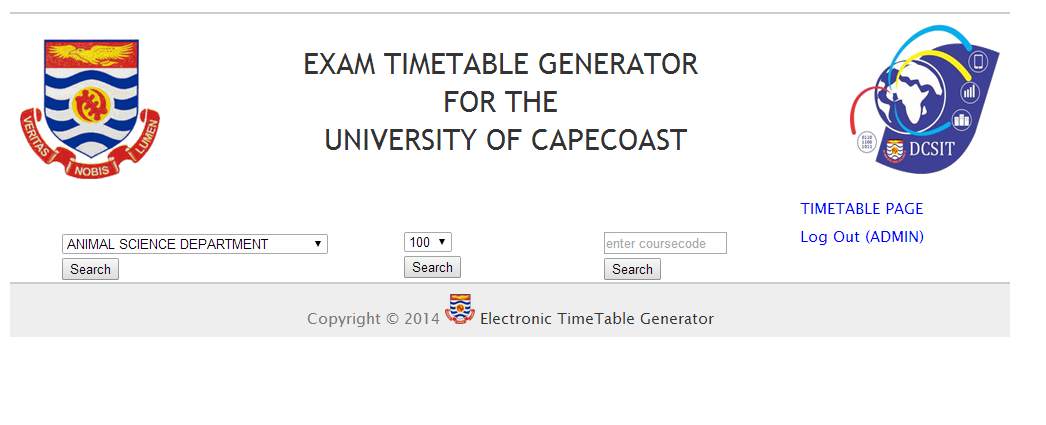
*Figure18 (Delete Date Page)*

After clicking on “Generate TimeTable” button, the administrator can then view the timetable page. The administrator can also click on search options to search by either department, level or course code.





*Figure 19 (Timetable page)*

*Figure20 (Search Page)*

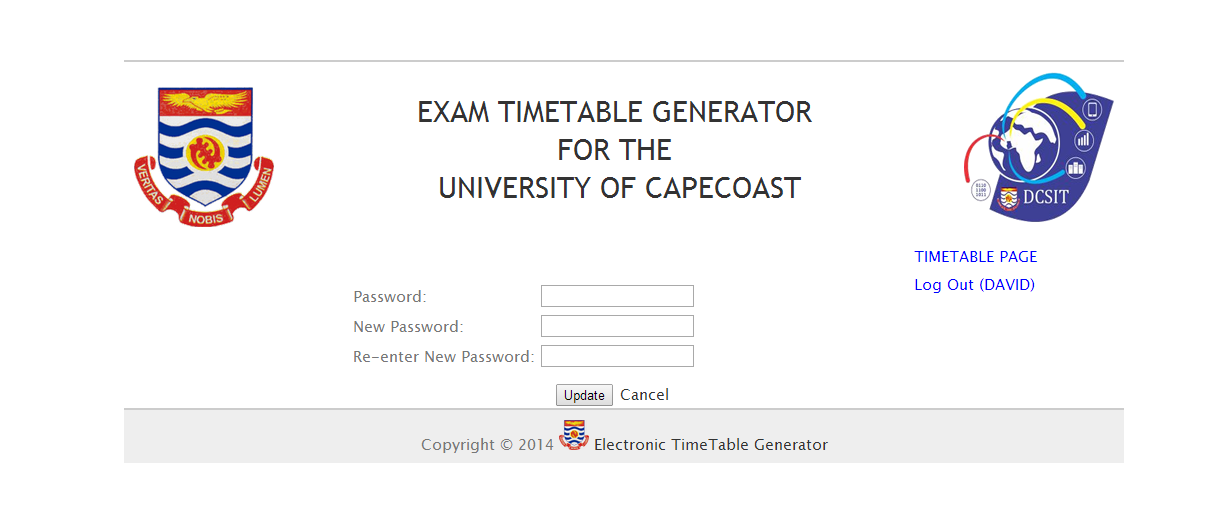
The administrator can then logout by clicking on the “Log Out (ADMIN)”.

### 3.1.2. Members

As a member you can view the following pages;

After login into the system with correct username and password, the member is sent to a page that requires him to reset his or her password. If the member is logging in for the first time he or she is advised to reset password and then proceed to viewing timetable page. The members can also search and lookup.

* **Reset Password**



*Figure21 (Members Reset Password Page)*

* Timetable page

The member can only reset password and view generated timetable. The access level is different as compared to that of the administrator. The member can then log out by clicking on the logout link.

## 4.1. REPORTING

Reporting section describes in what way information collected by the application are presented and how to access the information. Information collected by the application is being stored in the database and then presented to the user in a tabular form. The time table generated can be printed by printing the timetable page. As long as you provide correct password and username, you will get access to the system and be able to view what your access level can view.