**MACHAKOS UNIVERSITY**

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF INFORMATION COMMUNICATION TECHNOLOGY**

**DESIGN AND IMPLEMENTATION OF AN EXAMINATION SCHEDULING SYSTEM**

**GICICI PETER CIIRA  
J17-5552-2015**

**Submitted in partial fulfilment of requirement of the Degree of Bachelor of Science Computer Science**

**September 2019**

# DECLARATION

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

**Student**

Name: Gicici Peter Ciira

Adm. No: J17-5552-2015

Sign. …………………………………

Date. …………………………………

**Supervisor**

The project of Gicici Peter was reviewed and approved for examination by the following:

Name………………………………

Sign…………………………………

Date…………………………………

# DEDICATION

I dedicate this project to my family, lecturers, friends and all those took park in the project for the support they gave me during the period of doing this project proposal.   
God bless you all.

Great appreciation to all those in one way or another contributed to the success of this project writing.

Special thanks go to my supervisor for his guidelines, advice and motivation.

# ACKNOWLEDGEMENT

I thank God because of seeing me through and His strength, this far is all about Him. Great thanks to my family too for their prayers.

Special thanks to my supervisor and my family for their advice and motivation that enabled this project be successful.

Finally, I express my sincere appreciation to the friends and fellow students of class J17-2015 as well as my seniors for their assistance.

# ABSTRACT

There existed a lot of work like consolidating the exam timetable, setting sitting arrangement and invigilating allotment which was done manually or with some systems which took a lot of time and required man power which also had no perfection. The pre-existing system lacked the notification prompt meant for notifying the student of any changes on the timetable. Also, the aspect of dealing with the arising problems such as increase or decrease in number of students, absenteeism the moderators or sitting arrangement. Though not common, there were issues of room collision where two or more groups could be allocated one room at the same time. The purpose of developing this advanced scheduling system was to computerize the traditional way of conducting exams and avoid the common errors like collusion in the exam rooms and make it accessible to both the exam department as well as the students who are usually ignored during the whole procedure of scheduling examinations. It is a web-based system that can be used by both the exam coordinator and students on their PCs and mobile phones. The system was developed using alliterative methodology where later unit testing, acceptance testing, database testing and web-browser compatibility testing methods were carried out.

Table of Contents

[DECLARATION ii](#_Toc18401841)

[DEDICATION iii](#_Toc18401842)

[ACKNOWLEDGEMENT iv](#_Toc18401843)

[ABSTRACT v](#_Toc18401844)

[LIST OF FIGURES xi](#_Toc18401845)

[LIST OF TABLES xii](#_Toc18401846)

[CHAPTER ONE: INTRODUCTION 1](#_Toc18401847)

[1.1 Introduction 1](#_Toc18401848)

[1.2 Background of Study 1](#_Toc18401849)

[1.3 Statement of the Problem 1](#_Toc18401850)

[1.4 Purpose of the Study 2](#_Toc18401851)

[1.5 Research Objectives 2](#_Toc18401852)

[1.5.1 General Objective 2](#_Toc18401853)

[1.5.2 Specific Objectives 2](#_Toc18401854)

[1.6 Justification of the Study 3](#_Toc18401855)

[1.7 Scope 3](#_Toc18401856)

[1.8 Limitations of the Study 3](#_Toc18401857)

[1.9 Conceptual Model 4](#_Toc18401858)

[CHAPTER TWO: LITERATURE REVIEW 5](#_Toc18401859)

[2.1 Introduction 5](#_Toc18401860)

[2.2 Information Technology and Timetabling 5](#_Toc18401861)

[2.3 Review of Existing Systems 6](#_Toc18401862)

[2.3.1 ASC Timetables 6](#_Toc18401863)

[2.3.2 The TimeTabler 7](#_Toc18401864)

[2.4 Summary of Gaps Filled 8](#_Toc18401865)

[CHAPTER THREE: METHODOLOGY 9](#_Toc18401866)

[3.1 Introduction 9](#_Toc18401867)

[3.2 Research Design 9](#_Toc18401868)

[3.3 Sources of Data 9](#_Toc18401869)

[3.3.1 Previous Work Records 9](#_Toc18401870)

[3.3.2 Personal Experience 9](#_Toc18401871)

[3.4 Methods of Collecting Data 10](#_Toc18401872)

[3.4.1 Observation 10](#_Toc18401873)

[3.4.2 Interviews 10](#_Toc18401874)

[3.4.3 Questionnaires 11](#_Toc18401875)

[3.5 Methodology 11](#_Toc18401876)

[3.5.1 Planning and Requirements 12](#_Toc18401877)

[3.5.2 Analysis 12](#_Toc18401878)

[3.5.3 Design 12](#_Toc18401879)

[3.5.4 System Implementation 12](#_Toc18401880)

[3.5.5 Testing 12](#_Toc18401881)

[3.5.6 Evaluation 12](#_Toc18401882)

[3.6 Tools for Implementation 12](#_Toc18401883)

[3.6.1 Hardware Tools 12](#_Toc18401884)

[3.6.2 Operating System 13](#_Toc18401885)

[3.6.3 Visual Studio 13](#_Toc18401886)

[3.6.4 MYSQL and phpmyadmin 13](#_Toc18401887)

[3.6.5 Apache HTTP Server 13](#_Toc18401888)

[3.6.6 PHP and JavaScript 13](#_Toc18401889)

[3.6.7 HTML, CSS and Bootstrap 13](#_Toc18401890)

[3.7 Tools for Data Analysis 13](#_Toc18401891)

[3.7.1 Excel 13](#_Toc18401892)

[3.7.2 SQL Server Reporting Services 13](#_Toc18401893)

[CHAPTER FOUR: SYSTEM ANALYSIS AND DESIGN 14](#_Toc18401894)

[4.1 Introduction 14](#_Toc18401895)

[4.2 Data Analysis Results 14](#_Toc18401896)

[4.2.1 Interviews Results 14](#_Toc18401897)

[4.2.2 Questionnaires Results 15](#_Toc18401898)

[4.2.3 Observations Results 15](#_Toc18401899)

[4.3 Requirement Analysis 16](#_Toc18401900)

[4.3.1 Functional Requirements 16](#_Toc18401901)

[4.3.2 Non-Functional Requirements 16](#_Toc18401902)

[4.3.3 User Requirements 17](#_Toc18401903)

[4.4 Analysis of the Current System 17](#_Toc18401904)

[4.5 System Architecture 19](#_Toc18401905)

[4.6 System Use Case Diagram 20](#_Toc18401906)

[4.7 System Sequence Diagram 21](#_Toc18401907)

[4.8 Data Flow Diagrams 22](#_Toc18401908)

[4.8.1 General DFD 22](#_Toc18401909)

[4.8.2 Admin DFD 22](#_Toc18401910)

[8.2.3 User DFD 23](#_Toc18401911)

[4.9 Flowchart Diagram 24](#_Toc18401912)

[4.10 Physical Design 25](#_Toc18401913)

[4.10.1 All Database Tables 25](#_Toc18401914)

[4.10.2 Table ‘Allowed’ 25](#_Toc18401915)

[4.10.3 Table ‘Batches’ 25](#_Toc18401916)

[4.10.4 Table ‘Courses’ 25](#_Toc18401917)

[4.10.5 Table Slots 25](#_Toc18401918)

[4.10.6 Create Table SQL Query 26](#_Toc18401919)

[4.11 Entity-Relation Diagram 26](#_Toc18401920)

[CHAPTER FIVE: IMPLEMENTATION AND TESTING 27](#_Toc18401921)

[5.1. Introduction 27](#_Toc18401922)

[5.2. Development Tools 27](#_Toc18401923)

[5.2.1 HTML, CSS Bootstrap 27](#_Toc18401924)

[5.2.2 Hypertext Pre-processor (PHP) 27](#_Toc18401925)

[5.2.3 MySQL 27](#_Toc18401926)

[5.2.4 JavaScript 27](#_Toc18401927)

[5.2.5 Visual Studio Code IDE 28](#_Toc18401928)

[5.3 Testing 28](#_Toc18401929)

[5.3.1 Unit Testing 28](#_Toc18401930)

[5.3.2 Acceptance Testing 28](#_Toc18401931)

[5.3.3 Database Testing 28](#_Toc18401932)

[5.3.4 Web-Browser Compatibility 28](#_Toc18401933)

[5.4 Proposed Changeover Strategy 29](#_Toc18401934)

[5.4.1 Phased Changeover 29](#_Toc18401935)

[5.4.2 Parallel Changeover 29](#_Toc18401936)

[5.5 Main Components of the System 29](#_Toc18401937)

[5.5.1 Dashboard 29](#_Toc18401938)

[5.5.2 Allocate Timetable 30](#_Toc18401939)

[5.5.3 View Timetable 30](#_Toc18401940)

[CHAPTER SIX: DISCUSSION, CONCLUSION AND RECOMMENDATIONS 31](#_Toc18401941)

[6.1 Introduction 31](#_Toc18401942)

[6.2 Discussion 31](#_Toc18401943)

[6.3 Recommendation 31](#_Toc18401944)

[6.4 Future Work 31](#_Toc18401945)

[6.5 Conclusion 32](#_Toc18401946)

[REFERENCES 33](#_Toc18401947)

[APPENDICES 35](#_Toc18401948)

[APPENDIX 1: SQL Query 35](#_Toc18401949)

[APPENDIX 2: Sample Code for Timetable Management 39](#_Toc18401950)

[APPENDIX 3: Questionnaire 43](#_Toc18401951)

[APPENDIX 4: Project Budget 44](#_Toc18401952)

[APPENDIX 5: Project Activities 45](#_Toc18401953)

# LIST OF FIGURES

[Figure 1: Conceptual Model 4](#_Toc8734682)

[Figure 2: Iterative SDLC Model 11](#_Toc8734683)

[Figure 3: Pie Chart Results 14](#_Toc8734684)

[Figure 4: Bar Chart Results 15](#_Toc8734685)

[Figure 5: Current System Flowchart 18](#_Toc8734686)

[Figure 6: Proposed System Architecture 19](#_Toc8734687)

[Figure 7: Use Case Diagram 20](#_Toc8734688)

[Figure 8: Sequence Diagram 21](#_Toc8734689)

[Figure 9: System Flowchart 24](#_Toc8734690)

[Figure 10:E-R Diagram 26](#_Toc8734691)

# LIST OF TABLES

[Table 1: Project Budget 55](#_Toc9346376)

[Table 2: Project Activities 55](#_Toc9346377)

[Table 3: Gant Chart 56](#_Toc9346378)

# CHAPTER ONE: INTRODUCTION

## 1.1 Introduction

This phase looks out at the level of performance of the existing exam scheduling systems, their usability and how limited they are. Also, there has been a comparison between the algorithms. The definitions explain the current systems and the problems associated with them, and looks at the merits of using a more advanced system overall the others.

## 1.2 **Background of Study**

Handling examinations happens to be one of the most crucial activity all over the learning institutions. With the complexity of coming up with a more advanced system, technology has been seen as an alternative to take up the tiresome tasks involved. Technology is growing at unmeasurable rate across all the fields, education sector being one of the many positively impacted. Driven by the widespread understanding of the technology’s capabilities, the power of automating systems in a more dynamic way is approaching rapidly and taking over the current static-like systems. This growth will make it have a domination force in education field both locally and internationally.

To fit in such an era of evolving technology, learning institutions must devise on effective and advanced scheduling systems.

## 1.3 **Statement of the Problem**

The tedious tasks of data introduction and revision of usually incomplete solutions are the bottlenecks in this case (Seco, D., 2011). When the exam moderators land on a timetabling system, they expect to carry out the whole process of exam timetabling with much ease, faster and in a more dynamic way and expect to reach out the students. This happens not to be the case as the current system is much more way fixed without a way to modify it, hence keeps giving the same results without considering the current situation. For example, allocating the same room to a certain group of students since 1st year not considering if they increased or decreased in number.

There happens to be no way of the student knowing when there are any adjustments on the timetable. Currently, the students have to visit the school’s noticeboard to have access of the timetable. This system with a notification module is all that can help bridge the communication gap.

Though the existing systems are said to be functioning with much perfection, errors arise where there are collusions in the exam room allocation. This happens when a room can be allocated to more than one group of students at the very same time. It is not a common thing but once it comes to be, there’s a lot of misunderstanding and time-wastage in finding a new room to sit for the exams.

The aspect of flexibility has been omitted making the systems lack the dynamic scope. Affected fields may change and there’s a need to deal with the so rising problem. A lecturer may be in need to have a certain sitting arrangement. Current systems are hard to predict or change the aspect of siting arrangement and spacing.

Some of the systems are application-based running remotely on a specific computing device. The rapid evolving technology and arise of numerous operating systems makes it hard to install, configure and run the software independently based on the type of operating system.

The problem is being studied for last more than four decades, but a general solution technique for it is yet to be formulated (Datta D. et.al, 2006).

## 1.4 **Purpose of the Study**

The purpose of developing a more advanced exam scheduler system is to automate the regular way of organizing the examinations in an educational institute and making it more interactive and adjustable according to the current arising issues. This makes the students to be involved in the scheduling process in terms of communication via the notification.

## 1.5 **Research Objectives**

### 1.5.1 General Objective

To solve the need for a more interactive, communicative and flexible exam scheduler by implementing an advanced and universal across the learning institutions worldwide irrespective of the location or the type of operating system being used.

### 1.5.2 Specific Objectives

1. To notify the students when change occurs on the timetable
2. To implement and deploy the proposed system on the web through a web domain
3. To create a single scheduling platform for students, lecturers and moderators
4. To solve the problem emerging from current arising issues

## **1.6** **Justification of the Study**

Automatic timetable generation is a new technological trend in terms of making the whole process simple and fast. However, it’s not guaranteed that the parties involved are at the best of acquiring the services or the returns expected. The issue of not being satisfied by the current systems being used is at an awake. Learning institutions are the main relevance of the study targeting the examination moderators, lecturers and students as well. This study may be used as a pillar and reference of similar researches that may arise in the near future.

## 1.7 Scope

The main scope of this is analysing the modification of current exam schedulers as used by exam moderators. Problems in flexibility of the system and communication between exam department and the students are the main issues taken into place. This study deals with a fast and effective process to generate exam timetable with much more effectiveness. This includes allocation of rooms, lecturers, time and enhance interaction between students and moderators. At large, all the activities carried out by the examination department.

## **1.8 Limitations of the Study**

The development of this system faced a number of challenges which include:

1. Some students were not friendly and therefore this led to insufficient source of information when carrying out the study
2. A problem in trying to access the students and moderators involved as they attended classes would also lead to insufficient source of information and wrong information being given.
3. Limited knowledge by the students about the system being used led to lack of enough and wrong information
4. Expensive and time consuming much time was allocated for gathering information locally and over the internet.

## 1.9 Conceptual Model

Figure 1: Conceptual Model

Timetable

Screen View

Rules / Constraints

Edit

Timetabling

Engine

Delete

Review

Download

Web Site

Timetable

Print

Lecturer

Hall

Degree

Labs

Departments

Semester

Admin

Batch

Student

Subject

Verification

1

st

Level

2

nd

Level

3

rd

Level

The first level indicates all those which were involved as basic requirements into making the timetable. Lack of any of the objects would have led to the system being not able to meet its objectives. Second level are all those processes which are mostly logical and saw it having the timetable generated. For the third level, the user has the options on how to view the timetable by either on the screen of a computer, printing it or downloading it from [http://examsch.000webhostapp.com](http://examsch.000webhostapp.com/).

# CHAPTER TWO: LITERATURE REVIEW

## 2.1 Introduction

A timetable is an organised list, usually set out in tabular form, providing information about a series of arranged events in particular, the time at which it is planned these events will take place. They are applicable to any institution where activities carried out by different individuals at a specific given time. This makes timetables a framework of all activities in a learning institution including scheduling the exams. As a result, schools have devoted time, energy and human capital to the implementation of nearly optional timetables which must be to satisfy all required constraints as specified by participating entities (Robertus, 2002).

## 2.2 Information Technology and Timetabling

Solutions to timetabling problems have been proposed since the 1980s. Research in this area is still active as there are several recent related papers in operational research and artificial intelligence journals. This indicates that there are many problems in timetabling that need to be solved in view of the availability of more powerful computing facilities and advancement of information technology (S.B. Deris et.al, 1997).

The problem was first studied by Gotlieb (1962), who formulated a class-teacher timetabling problem by considering that each lecture contained one group of students, one teacher, and any number of times which could be chosen freely. Since then the problem is being continuously studied using different methods under different conditions. Initially it was mostly applied to schools (de Gangs, 1981; Tripathy, 1984). Since the problem in schools is relatively simple because of their simple class structures, classical methods, such as linear or integer programming approaches (Lawrie, 1969; Tripathy, 1984), could be used easily.

However, the gradual consideration of the cases of higher secondary schools and universities, which contain different types of complicated class-structures, is increasing the complexity of the problem. As a result, classical methods have been found inadequate to handle the problem, particularly the huge number of integer and/or real variables, discrete search space and multiple objective functions.

There are two main problems in timetabling. The first one is related to the combinatorial nature of the problems, where it is difficult to find an optimal solution because it is impossible to enumerate all nodes in such a large search space. The second one is related to the dynamic nature of the problems where variables and constraints are changing in accordance with the development of an organization (S.B. Deris et al., 1997). Therefore, a timetabling system must be flexible, adaptable and portable, otherwise the users will not use the system optimally or even as decision aids such as for storing, retrieving, and printing timetables, when the timetable planning decisions are made manually.

A school timetable consists of four main elements which include students, teachers, rooms and time allocation or slots. To be able to solve the entities being fed with, algorithms are used to solve the various complex computations so as to give expected results. One of the most used is the NP-Complete where the time required to solve the problem using any currently known algorithm increases very quickly as the size of the problem grows (Ossam Chohan; 2009).

Genetic algorithms are also widely used, though they have difficulty dealing with problems with "deceptive" fitness functions (Mitchell, 1996), those where the locations of improved points give misleading information about where the global optimum is likely to be found.

The reason for the difficulty is because of the great complexity of the construction of size of lectures and examinations, due to the scheduling size of the lectures and examinations periods and high number of constraints and criteria of allocation which is usually circumvented with the use of little strict heuristics, based on solutions from previous year (Moreira, J.J., 2008).

A few worth mentioning EAs, used for the school timetabling problem, are those of Abramson et al. (1992), Piola R.(1994), and Bufe et al. (2001). Similarly, EAs, used for the university class timetabling problem, are those of Carrasco et al. (2001), Srinivasan et al. (2002) and Datta et al...

## 2.3 Review of Existing Systems

There are a number of Enterprise Resource Planning (ERPs) systems that are currently being used around the world. The most used are the Timetabler and the ASC.

### 2.3.1 ASC Timetables

ASC Timetables is a tool that is used to generate timetables in learning institution developed by a group known as Applied Software Consultants, hence ASC. It does the common tasks such as organizing time, teachers and students and mostly used for scheduling class times. The software has been developed in Slovakia back in 1993. It comes in an .exe format where one installs the software and runs it on a Windows operating system at a price of 99 USA dollars after the trial period has expired. Once downloaded, the user is taken though the set-up process. The earliest most successful version was of 1998, though it did not support online running as the current distribution does.

The software allows the user to enter new subjects, classes, teachers and lessons. Once fed with data, the ASC allows the user to generate the teaching timetable with a click of a button. However, the software can only run on Windows 2000 and higher only.

**Advantages of ASC**

The ASC is simple to use as it comes with readily available tutorials and has fewer buttons to deal with and the interface is beautiful and user friendly. ASC can be run on a PC and desktop as an application and also online after installation.

**Disadvantages of ASC**

ASC is not a free software and one has to pay for it, the least amount being 99 USD and comes with a limitation on the type of operating system to run on, that is, only on Windows. When using this software, it has been proven hard to export data to other applications such as spreadsheets. Another major disadvantage of the ASC is that it has no aspect of room size and one cannot set specified spacing. Like many applications, ASC is only available to the admin.

### 2.3.2 The TimeTabler

The TimeTabler is a software that is used to schedule learning activities on larger institutions such as colleges and universities. It has been around for over 40 years and available to over 80 countries worldwide. The TimeTabler is downloaded and is available for Windows, Linux and MAC operating systems. It helps in allocating teachers to various classes at a given time. There is no involvement of the students as the admin is the only one who can access the software. All that happens is that the admin feeds data about the rooms, teachers and subjects and then sets time allocation for lessons and the system generates the timetable on a click of a button. It comes with a feature called FIT where if a problem lesson will not fit in the timetable, the TimeTabler quickly searches for various solutions one can opt to go for. Another feature is that it has an aspect of managing staff and can record on their absenteeism.

**Advantages of the TimeTabler**

TheTimetabler is available for the most common operating systems which include Windows, Linux and MAC OS. Just like the ASC, it is easy to use the TimeTabler and data can be exported to Web Pages and XML.

**Disadvantages of TimeTabler**

The TimeTabler is limited to only the admin and comes at a cost where one has to pay to use it. This software is only available as an installation package and cannot be accessed over the internet.

## 2.4 Summary of Gaps Filled

The introduction of the scheduling system increased the interaction between the students and the exam department during the generation of the timetable and sends emails once changes emerge on the timetable.

Also, the system helps in ‘manual’ allocation of rooms in a different way from the GA algorithms. This helps the moderator deal with any issues that arise and assist in modifying the entire timetable instead of giving same results.

# CHAPTER THREE: METHODOLOGY

## 3.1 Introduction

This chapter describes the research methodology that was used to carry out the study. Research methodology provides details regarding the procedures used in conducting a study (Mugenda & Mugenda, 2003).

## 3.2 Research Design

This study has adopted correlational study method. According to Upagade and Shende (2012), research design is the arrangement of condition from collection and analysis of data in a manner that aims to combine relevance to the research purpose with economy in procedure. The reason for choosing this method is that the whole development involved comparing the current system with this system and improved on its negativity.

## 3.3 Sources of Data

It involves the places where one gets the knowledge and information all that is needed to solve or come up with a solution to something. The sources of data used include:

### 3.3.1 Previous Work Records

It included the areas similar to the study that has already been done or worked upon. For this case, a previous work done by Isaac Nnachi, a student of Akanu Polytechnic back in 2015 was read to help in designing a better system from the vulnerabilities identified.

The main limitation of this source was that the article was documented at earlier days when technology had not emerged that much and some of the technology such as using of Visual Basic has been outdated.

### 3.3.2 Personal Experience

This source involves something that one has gone through and had a positive or negative view about it. One going through a negative experience may opt to go for a change for a better system. In this study, there have been experiences about the current exam timetabling system.  
The advantage of this source of data is that one gets the best ideas for improvement since one is the victim.

## 3.4 Methods of Collecting Data

Preliminary investigations of the existing system were done through the following ways-

### 3.4.1 **Observation**

Observation is a method that involves an individual getting first-hand information. It involved visiting the school, looked at the current system, its operation and the effects of its output on both the students and lecturers during the exam periods.

**Advantages of Observation**

1. Got first-hand information right by visiting the lecture rooms
2. It was much cheap as compared to questionnaires as it involved making rounds within the university compound and no printed documents were required

**Disadvantages of Observation**

1. Time allocation and scheduling inconveniences for the researcher since there were classes to attend to
2. Behaviors changed at sometimes and compromising the data accuracy such as being tired due to the course work

3.4.2 Interviews   
For this technique, the analyst will got involved in questioning the students and lecturers and getting information from them, for example students during their free times.

**Advantages of Interviews**

1. They were timesaving as the target is already identified and instant feedback which included the students, lecturers and the exam moderator
2. Helped to come up with worries that were felt by those interviewed and got even to know their emotional feelings

**Disadvantages of Interviews**

1. Some of the students and lecturers were resistance to being interviewed as they had course works to attend to
2. It was hard to record and tabulate all the information at once

3.4.3 Questionnaires

Questionnaires are special documents whose purpose is to allow gathered raw facts to be documented and involves a large number of people. The students were the main target and were given out the forms and filled the in the answers for the asked questions about the existing systems.

**Advantages of Questionnaires**

1. Personal and controversial questions were taken into place as for many activities were anonymous. No one knew which person filled a certain part in the form
2. It was relatively cheap as the groups of users and operators were scattered and no costs for the object as the subjects (mostly students) already had the forms

**Disadvantages of Questionnaires**

1. There were no direct contact between the students or lecturers involved and did not get the feeling and direct opinions of the group involved
2. Misinterpretation of feedback gotten from the students, lecturers and the exam department

## 3.5 Methodology

The developed system used iterative methodology during its development as shown and explained below:

IMPLEMENTATION

EVALUATION

TESTING

DESIGN

ANALYSIS

PLANNING

Figure 2: Iterative SDLC Model

### 3.5.1 Planning and Requirements

It gives out the information on how the current system works, that is its operations and all the tasks involved. It brought to light what is happening. The main aim was to give the differences between the current system and the developed system. For this case, a notification to students and URL is an advantage over the current timetabling system.

### 3.5.2 Analysis

Requirement analysis gave out explanation on how the current system operates, that is, how it works and the tasks involved.

### 3.5.3 Design

The attributes of the present system were studied to determine what changes will be needed to add on that are not met by the system presently. The result of this part consisted of the specifications, which described both what the system altered and how it performs its tasks. The main addition is a notification module.

### **3.5.4 System Implementation**

During this phase the new enhanced system was installed in the production environment and exam moderators, students and lecturers trained on how to use the system.

### 3.5.5 Testing

The designed system was validated through a sequence of unit, integration, performance, system and acceptance testing. The objective was to ensure that the system functions as expected and that problems are solved. This was to make sure the notification works perfectly, whether the system can run on URL and satisfied the exam department in terms of ease of usage.

### 3.5.6 Evaluation

The developed system used adaptive maintenance method so as to be able to curb arising issues such as changes in school management like increase of faculties and departments.

## 3.6 Tools for Implementation

This involved all those equipment and software used to come up with the proposed system.

### 3.6.1 Hardware Tools

The main devices used include a laptop and a smartphone. The laptop was used as a tool that had all the software needed to come up with the system. A smartphone was used as a source of internet connection through hotspot.

### 3.6.2 Operating System

The operating system used in this case is Ubuntu Bionic Beaver version 18.04 which is a type of Linux operating system. Its simplicity in usage made it a better option to use.

### 3.6.3 Visual Studio

It was the development environment that was used for coding. The VS is one of the best in terms developing web-based systems. The Advanced Exam Scheduler runs on URL.

### 3.6.4 MYSQL and phpmyadmin

MySQL was used as the database engine while phpmyadmin is the database administration tool because of its effectiveness in portability and web hosting services.

### 3.6.5 Apache HTTP Server

The reason for going for this cross-platform web-server is because of its capability to support PHP programming language which is also used in development.

### 3.6.6 PHP and JavaScript

PHP language can easily be embedded into HTML easily and also can combined with many web templates. HTML was used in the development of the AES system. For JavaScript, it enabled creation of interactive web systems. The system has a number of buttons to interact with.

### 3.6.7 HTML, CSS and Bootstrap

All these was used to come up with the interface of the system. CSS and Bootstrap made the interface to be more appealing.

## 3.7 Tools for Data Analysis

The following tools were used to analyze the data and explains the processes that were followed.

### **3.7.1 Excel**

The timetable was viewed on the spreadsheet and analysis done to determine its vulnerabilities.

### 3.7.2 SQL Server Reporting Services

The SSRS was taken as a visual self-service analytics and output spreadsheet versions of reports on how the data is being handled in the database. An example is exam batch reports.

# CHAPTER FOUR: SYSTEM ANALYSIS AND DESIGN

## 4.1 Introduction

System analysis and design involves the process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components, (Yen, D.C. and Davis, W.S., (1998).  In this case, it has been conducted for the purpose of studying the system and its parts in order to understand its objectives much better.

## 4.2 Data Analysis Results

To archive the objectives of developing this system, various methods of data collection were involved. This include observation and personal experience, conducting interviews and use of questionnaires. The following are the results gotten from these methods:

### 4.2.1 Interviews Results

The data collected from this method was analyzed and sampled using Microsoft Excel. 50 volunteers agreed to be interviewed main queries being if they were satisfied with the current exam scheduling process and their view on this system. The results are as follows:

* + Comfortable with existing scheduling process- 7 (14%)
  + Need a better scheduling system with ease of use- 35 (70%)
  + Those who had no idea on the scheduling process- 8 (16%)

This data is presented on the pie chart below:

Figure 3: Pie Chart Results

### 4.2.2 Questionnaires Results

Having presented the idea of coming up with this scheduling system, the main questions in the questionnaire forms were: feelings towards the existing system and whether there is need of having a better way of scheduling exams. 35 typed sheets were issued. Results are as follows:

* + Okay with the current scheduling process- 5
  + Need for a better scheduling system- 25
  + Insufficient data due to errors in the forms-5

This information is represented in the bar chart below:

Figure 4: Bar Chart Results

### 4.2.3 Observations Results

Observations were made taking notes of the process of exam scheduling and the tasks involved. The results were analyzed made it clear that:

* There was need for changing the process of scheduling
* A new and better system was required to schedule exams.

## 4.3 Requirement Analysis

It involves the description of the features and behavior of the system, including elements that defines how the users interact with the software and other non-functional elements. The following are the two types of requirements taken into place in this system:

### 4.3.1 Functional Requirements

Functional requirements define a function of a system or its components, where a function is described as a specification of behavior between inputs and outputs. Generally, this means what the system does. The following are some of the main activities that the system is capable of based on the objectives of developing the system:

1. The system sends an email to students and lectures when there is a change on timetable
2. The system allows addition, verification and authentication of users, i.e. students, lecturers and exam moderators
3. The system gives an error prompt on mismatching or feeding wrong data
4. The system has limitations of tasks performed based on the level of user accessing the system
5. The system allows the exam moderator have full access to the system
6. The system allows file and data sharing during its operation
7. The system runs a specific URL (Universal Resource Locator) and accessed remotely, hosted by 000webhosting company; [http://examsch.000webhostapp.com](http://examsch.000webhostapp.com/)

### 4.3.2 Non-Functional Requirements

These include the general aspect of the system, placing reality on how the system will do its tasks. Some of the areas in the system under the non-functional requirements are;

1. The system is user friendly and easy to use
2. The system has the aspect of consistency and reliability
3. The system provides various WIMP (Windows, Icons, Menus, Pointers) interactions to users
4. The system is very responsive

### 4.3.3 User Requirements

This system is meant for examination department, students and lectures of a specified institution. The following are the requirements that are needed by the user to be able to interact or use the system.

1. The user needs a computing device, either a smartphone, desktop or a laptop
2. The user needs internet connection and to access the system’s URL
3. The user needs a stable browser to use the system

## 4.4 Analysis of the Current System

Generation of timetables differs from the aspect of the size of the institution and the number of students being involved. For smaller institutions, the process is done manually. This part of the article looks into more complex scheduling systems since the system targets colleges and universities.

Currently, the exam department gathers information about room details, the number of students, lecturers, the faculties and departments involved as the bases of scheduling exams. The time range of exam duration is set, courses added to the time set, rooms allocated and lecturers to supervise the exams are selected. If ready, the timetable is printed, sent to departments and posted on the noticeboards for students to view. The timetable is posted a few weeks before exam period to check on errors. Room collision happens to be the main problem and the exam department is notified and review the timetable.

The procedural process is as follows:

1. Room, Students, Lecturers, Faculties details are gathered
2. Time range is set, including time per paper and the exam duration
3. The courses are added to the time set, commonly known as course input
4. Rooms are allocated depending on their sizes
5. Lecturers are set to invigilate exams based on department one is
6. Timetable is printed and sent to departments
7. The timetable is posted on the noticeboards for error checking by students
8. If there are collisions or any other error, the issue is reported to the exam department and reviewed.

These processes are represented the data-flow diagram as shown below:

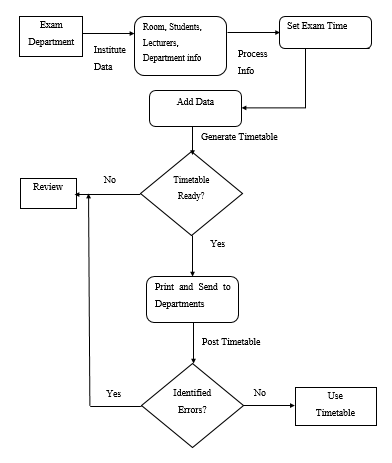


Figure 5: Current System Flowchart

## 4.5 System Architecture

In developing this system, the three-tier architecture has been used as the general structure in coming up with the system. Here is how the three-tire model has been incorporated into the system:

* **Tier 1:** This is the client or user side of the system. The user interacts with formatted HTML and PHP pages, feeding their queries there, which is submitted to the system middleware for processing.
* **Tier 2**: It is the application or the middleware part of the architecture. Both the users and the admin part access this tier respective of the rules one has over the system.
* **Tier 3:** Happens to be the back end of the architecture, storing the records and data of parties involved such as timetable records and number of students in a specific course.

USER

Computing  
 Device

Connects with Browser

Internet

[http://examsch.000webhostapp.com](http://examsch.000webhostapp.com/)

WEB/  
Application

Server

SOFTWARE DATABASE

DATABASE

PDO (PHP Data Objects)

Figure 6: Proposed System Architecture

## 4.6 System Use Case Diagram

Use case diagrams describe what a system does from an external observer's standpoint. The emphasis of use case diagrams is on what a system does rather than how. They are used to show the interactions between users of the system and the system. The use case below represents the several users called actors and the different ways in which they interact with the system.



Figure 7: Use Case Diagram

## 4.7 System Sequence Diagram

This describes how objects interact with each other through messages during the execution of a use case or any operation. They illustrate how messages are sent and received between objects and the sequence of message transfer. It also describes how operations are carried out according to the time of operation.

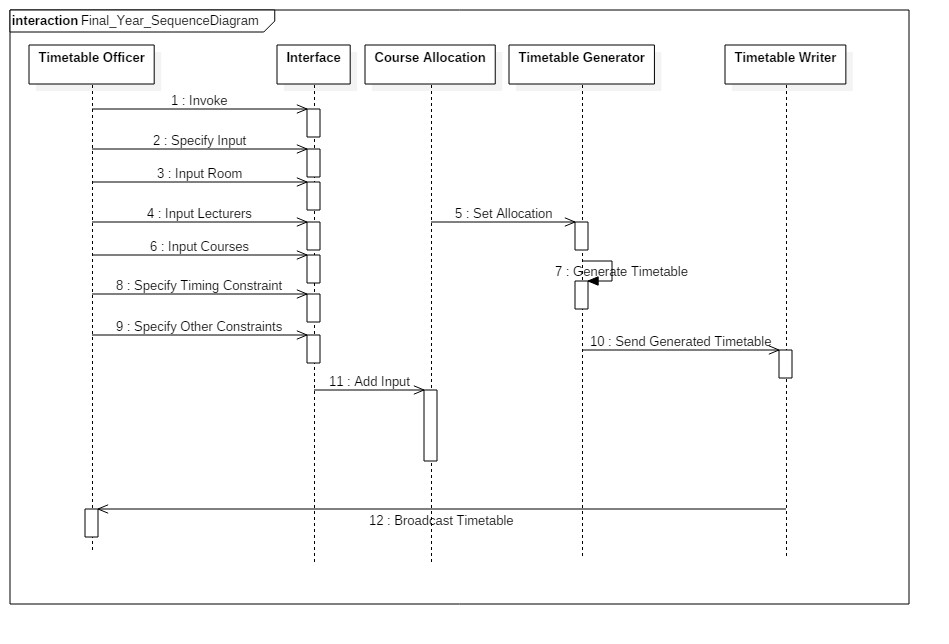


Figure 8: Sequence Diagram

## 4.8 Data Flow Diagrams

For better understanding of how the system works and its components, the different DFDs where been used each at its own level:

### 4.8.1 General DFD

Timetable Viewing and Printing

User

Admin

Timetable Generation

All Timetable Details

Generate reports

Figure 9: Overall DFD, Level 1

It indicates the users, generation and how they are connected.

### 4.8.2 Admin DFD

Validate login details

1.

Login

Admin

2.

Maintain Master Tables

Manage rooms/ students/courses/ department details

Generate reports

3. Database

Figure 10: Admin DFD, Level 2

These are the processes that the admin goes through when preparing the timetable.

### 8.2.3 User DFD

View Timetable

Check for Errors

Response to errors

Exam Dept. Notified

4. Database

Report to Department

User

Confirm Errors

Get Timetable

Update Timetable

Figure 11: User DFD, Level 3

This level indicates how the timetable is released to the students and lecturers via their respective departments and the how the whole activities flow.

## 4.9 Flowchart Diagram

This describes the diagrammatic representation of the logical paths followed during the development of this system:

**Start**

Student queries for

the timetable

Check

available

timetable

Sets institution, course

Details and year

Process Query

View Timetable

**Stop**

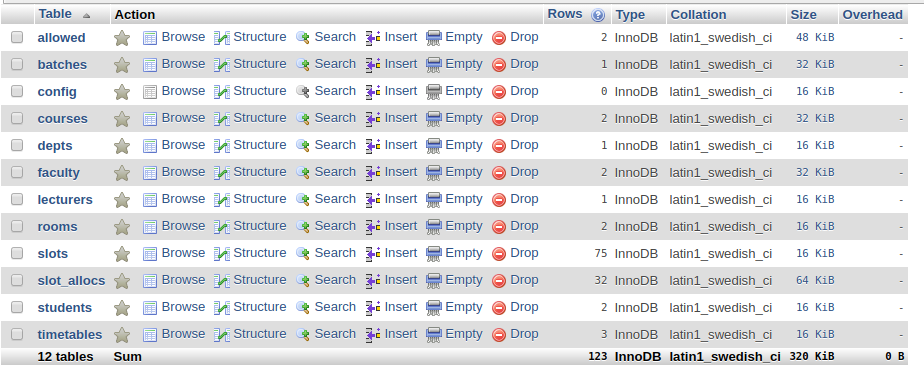
None

Available

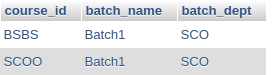
Figure 12: System Flowchart

## 4.10 Physical Design

### 4.10.1 All Database Tables



### 4.10.2 Table ‘Allowed’



### 4.10.3 Table ‘Batches’



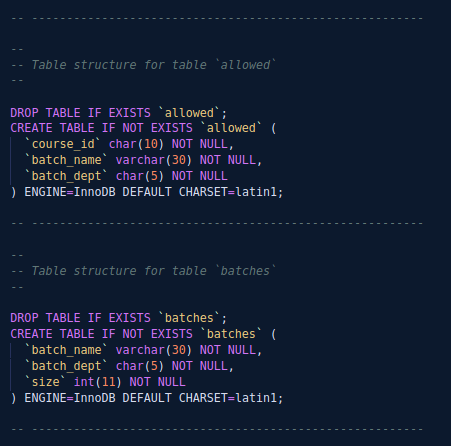
### 4.10.4 Table ‘Courses’



### 4.10.5 Table Slots



### 4.10.6 Create Table SQL Query



## 4.11 Entity-Relation Diagram

This is an organization of related objects. It gives an overview of a system by showing its classes and their relationships. Class diagrams only displays what interacts and not what happens during the interaction.

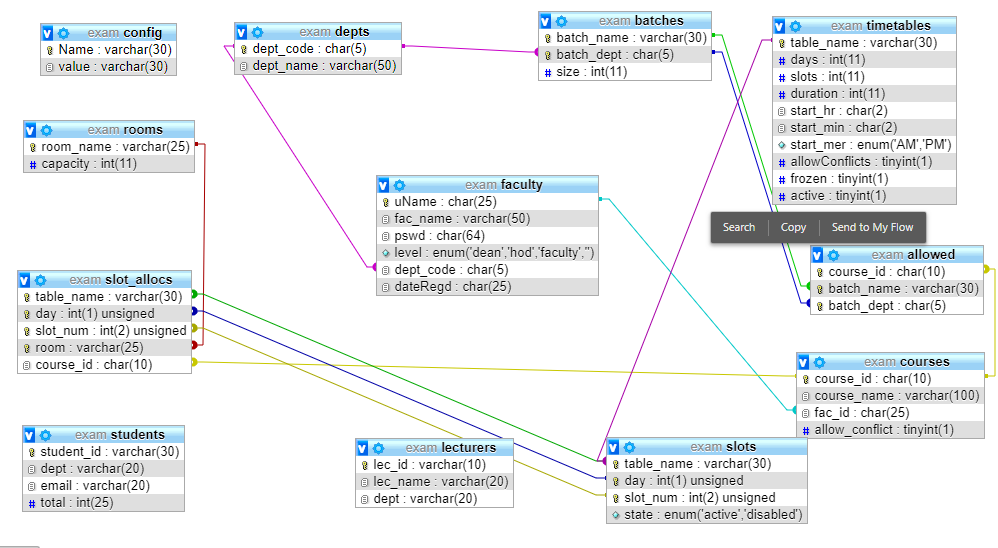


Figure 13:E-R Diagram

# CHAPTER FIVE: IMPLEMENTATION AND TESTING

## 5.1. Introduction

The aim of this section is to discuss tools used in the development of the system and testing of the examination scheduler system.

## 5.2. Development Tools

The development of this system was guided by an iterative SDLC. It is a web-based timetabling system that requires a computer device with a browser and internet. Throughout the coding process, the following languages and tools were used:

### 5.2.1 HTML, CSS Bootstrap

All of these were used to come up with the interface of the scheduling system. CSS and Bootstrap made the interface to be more appealing. The mark-up language was embedded into the two to achieve a good interface.

### 5.2.2 Hypertext Pre-processor (PHP)

PHP language can easily be embedded into HTML and also can combined with many web templates. For this project, the language was used to come up with both the client-side and the server-side. PHP helped in the project connecting to the database, helping either of the users to feed their queries in terms of addition, deletion and viewing of records. In this development phase, PHP 7.3 version was employed and, Object Oriented Programming (OOP) and, PHP Data Objects (PDO) paradigms largely used.

### 5.2.3 MySQL

It’s a popular choice of database for use in web applications, and it is a central component of the widely-used LAMP (Linux, Apache, MySQL, and PHP) web application software stack. This formed an important aspect of this project since the data fetched or viewed on the client side originates from the MYSQL database. The tables, stored procedures and processes are saved and run from this component.

### 5.2.4 JavaScript

It’s a programming language that can be included on web pages to make them more interactive. JQuery version 3.10 was used. JQuery is applied in this project for the purpose of input validations, form filters, table sorting and displays, run and produce JSON functions, and display of time.

### 5.2.5 Visual Studio Code IDE

It was the development environment that was used for coding. The VS is one of the best in terms developing web-based systems.

## 5.3 Testing

Testing involves evaluation of the whole system and its components to investigate if it meets the required specifications. Different tests were done on the scheduling system to identify any gaps, bugs, errors or missing requirements in contrary to the actual desire or requirements. The types of tests done on the system include:

### 5.3.1 Unit Testing

The purpose of unit testing was to ensure that each program is fully tested since it addresses the testing of functional units within a system as the main building blocks. This was done using written test plan and prepared test data.

### 5.3.2 Acceptance Testing

This type of testing was performed by the client and verified whether the end to end flow of the system is as per the business requirements or not and if it is as per the needs of the end user. The client accepts the software only when all the features and functionalities work as expected. It was the last phase of testing, after which the software goes into production. This system allow the users to check whether all the functionalities of the scheduler are working as per the requirements.

### 5.3.3 Database Testing

Whenever an input or data is entered on the frontend application of this system, it’s stored in the database and the testing of such database is known as Database Testing or Back-end testing. This involved testing of the table structure, schema, stored procedures and the data structures. In this type of testing, GUI was not involved. As the project developer, I am directly connected to the database with proper access and can verify data by running queries on the database. Issues identified such as data loss, deadlocks, data corruption can fixed before the system came to life.

### 5.3.4 Web-Browser Compatibility

This type of testing was performed because the system is a web-based application and it ensures the software run with combination of different browsers and operating systems. This also validates that this system runs on all versions of web browsers. The system was tested using Opera, Google Chrome and Mozilla Firefox browsers.

## 5.4 Proposed Changeover Strategy

### 5.4.1 Phased Changeover

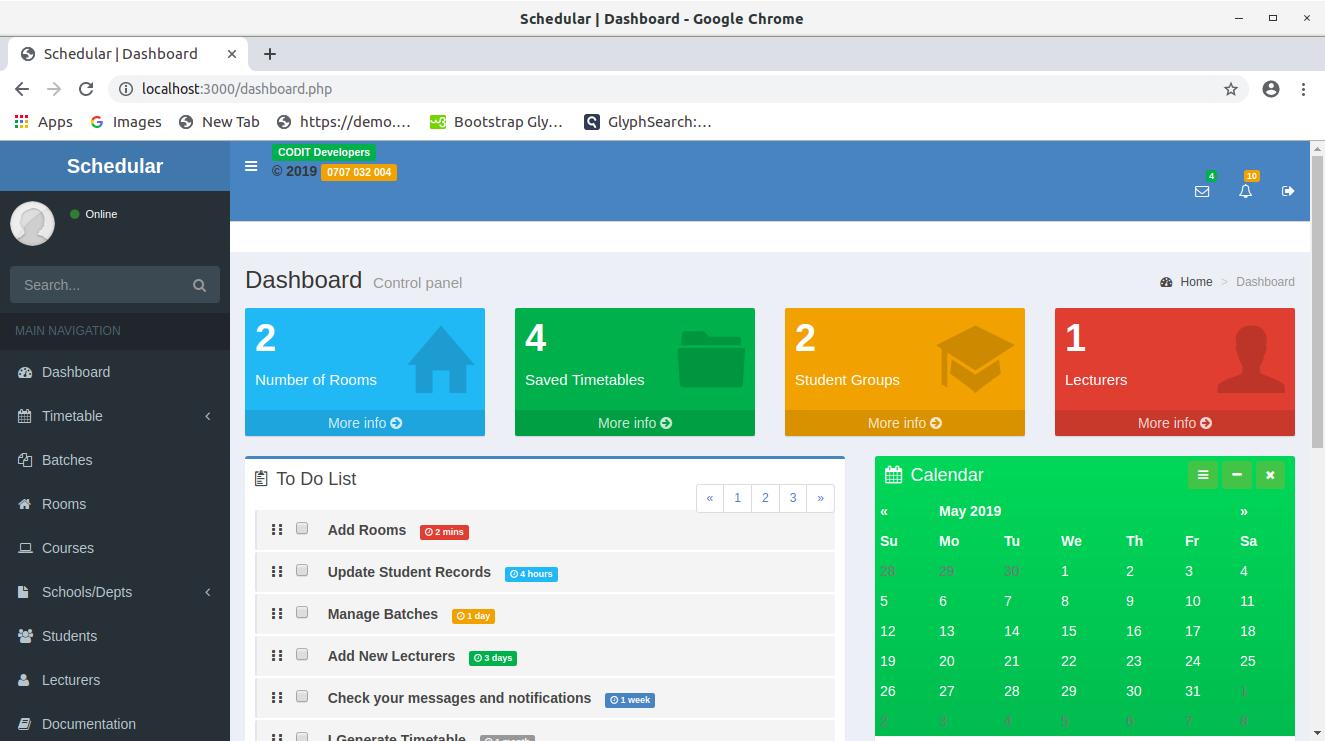
The new system is implemented one stage at a time. Advantages of using this system directly is that it is first, much accurate and less tiresome. The main disadvantage one may take some time before understanding the whole process of timetable generation.

### 5.4.2 Parallel Changeover

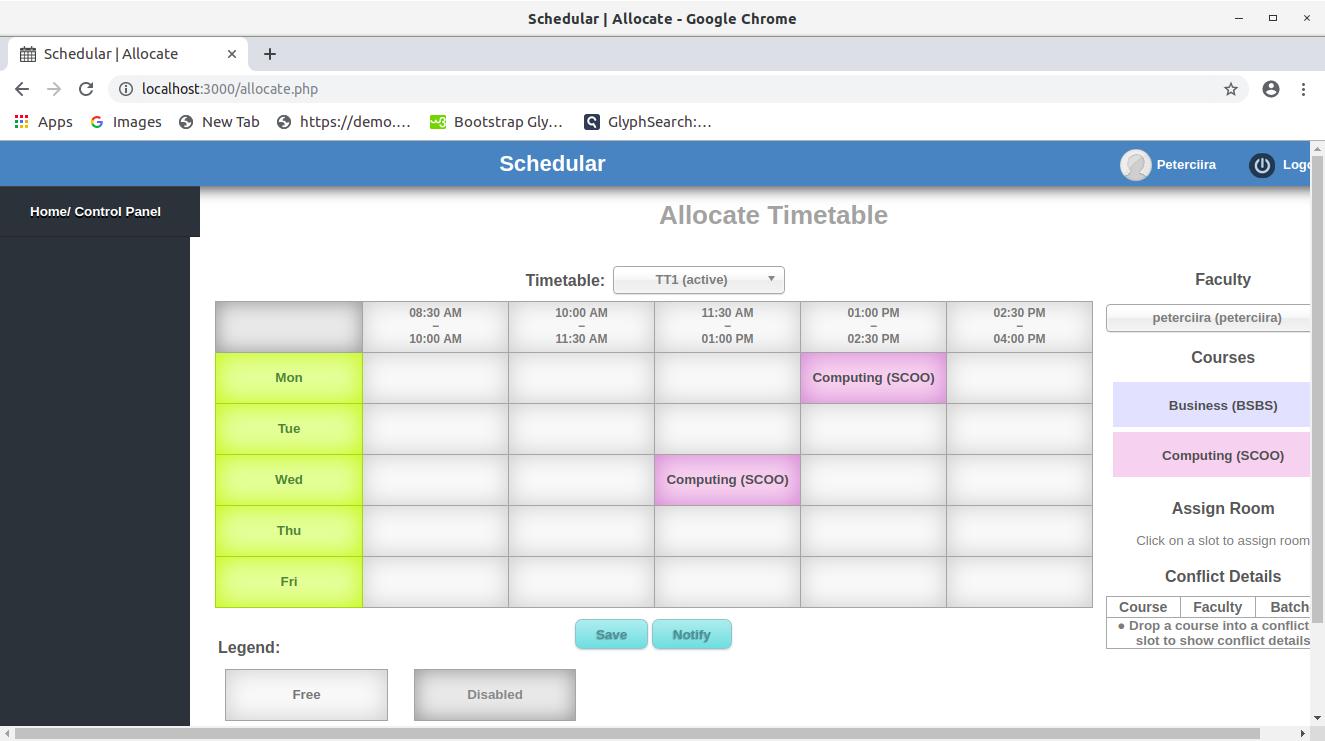
For this case, the new timetabling system runs hand in hand with the other existing system. It is not that productive though this scheduler is much useful when dealing with a large institution.

## 5.5 Main Components of the System

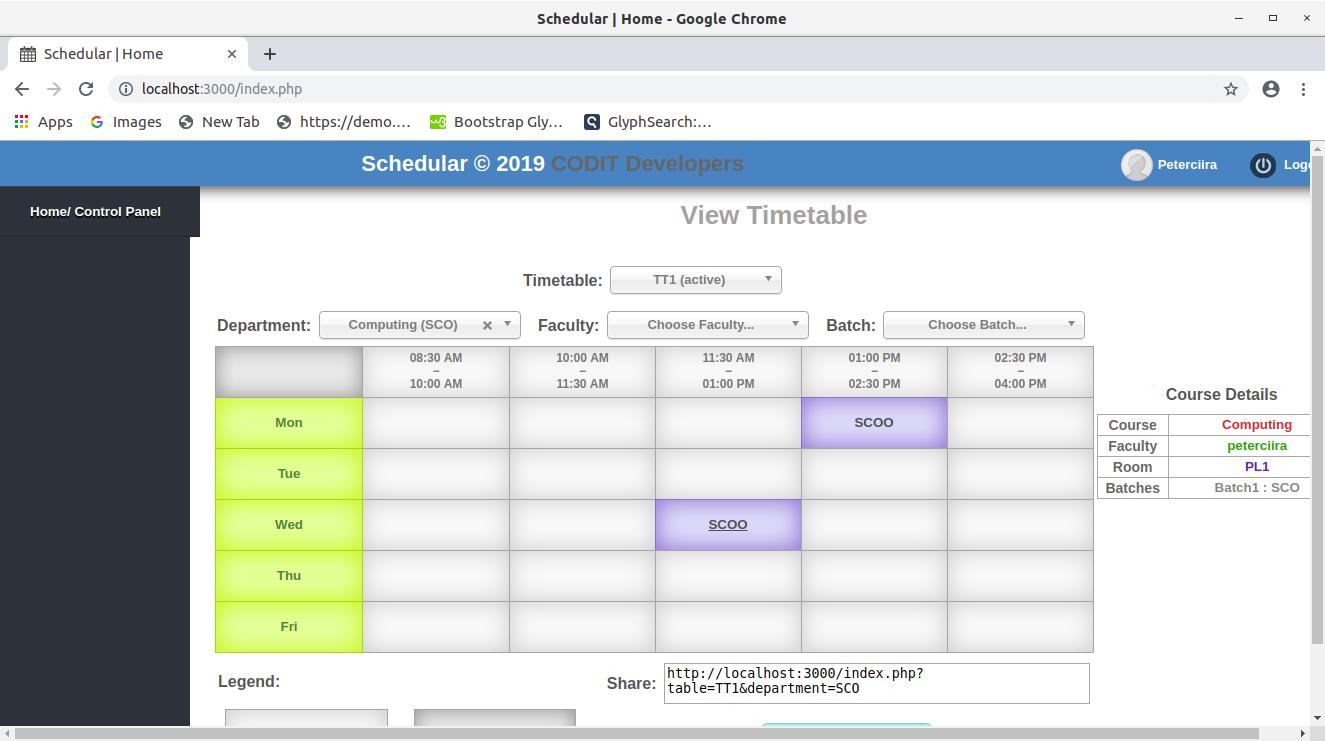
### 5.5.1 Dashboard



### 5.5.2 Allocate Timetable



### 5.5.3 View Timetable



# CHAPTER SIX: DISCUSSION, CONCLUSION AND RECOMMENDATIONS

## 6.1 Introduction

The need of developing the EXAM SCHEDULER SYSTEM was to bridge the communication gap between the students, lecturers and exam moderators. The system also helps in notifying the students on a change and when the timetable is generated, and tries to solve the problem of room collision.

## 6.2 Discussion

The project went for identifying the main problems being experienced during the exam periods in terms of scheduling. Various systems were analysed and discussed as per Chapter 2 of this documentation. All the objectives have been met by the system.

The timetable system runs on a web browser hence everyone involved in the exam activities has access to the timetable. If a change happens, the students are notified via their class email. For the rooms’ collision problem, the system has a prompt to notify you that a certain room has already been booked. In general, the objectives were fully met and each addressed extensively.

## 6.3 Recommendation

Schools are everywhere and doing exams remains a key thing in any learning institution. The problem arises when one is dealing with a large institution such colleges and universities. Most of the scheduling process is not fully satisfying but with this kind of system, any learning institution can be able to generate their own examination timetable much easy.

## 6.4 Future Work

This system and documentation can be improved and innovated to higher levels than what it aims to undertake currently. Future work is permitted on this project in areas that it did not cover and deemed as beneficial and applicable for this system. It can go for any kind of timetabling process in a learning institution.

## 6.5 Conclusion

The source of idea of coming up with this project is out of personal experience during one of the exam periods at the university. There was a need to address the issue and include the students who are always ignored during the scheduling process.

The system allows the students to get a notification on their emails once there is a change or a timetable has been edited or generated. Also, the system allows students and lecturers to view the timetable remotely as long as one has a browser and internet. This is via [http://examsch.000webhostapp.com](http://examsch.000webhostapp.com/). Room collusion issue has been well handled and one is notified if the rooms happened to overlap each other.

The system is highly recommended for large institutions as it is fast, user-friendly and simple to use.

# REFERENCES

Bufé, M., Fischer, T., Gubbels, H., Häcker, C., Hasprich, O., Scheibel, C., Weicker, K., Weicker, N., Wenig, M. and Wolfangel, C., (2001, April). Automated solution of a highly constrained school timetabling problem-preliminary results. In *Workshops on Applications of Evolutionary Computation* (pp. 431-440). Springer, Berlin, Heidelberg.

Carrasco, M.P. and Pato, M.V., (2000). A multiobjective genetic algorithm for the class/teacher timetabling problem. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 3-17). Springer, Berlin, Heidelberg.

Carpente, L., Cerdeira-Pena, A., De Bernardo, G. and Seco, D., (2011). An Integrated System for School Timetabling. In *ICAART (1)* (pp. 599-603).

Datta, D., Deb, K. and Fonseca, C.M., (2007). Multi-objective evolutionary algorithm for university class timetabling problem. In *Evolutionary scheduling* (pp. 197-236). Springer, Berlin, Heidelberg.

Forrest, S., (1993). Genetic algorithms: principles of natural selection applied to computation. *Science*, *261*(5123), pp.872-878.

Gotlieb, C .C. (1962). "*The construction of class-teacher timetables*." Proceedings of IFIP Congress, North-Holland Pub. Co., Amsterdam, 73-77.

Lawrie, N.L., (1969). An integer linear programming model of a school timetabling problem. *The Computer Journal*, *12*(4), pp.307-316.

Moreira, J.J., (2008). A system for automatic construction of exam timetable using genetic algorithms. *Tékhne-Revista de Estudos Politécnicos*, (9), pp.319-336.

Mitchell, M., (1998). *An introduction to genetic algorithms*. MIT press.

Mugenda, O.M. and Mugenda, A., G. (2003). *Research methods quantitative and qualitative approaches*.

Ossam, C., (2009). *University Scheduling using Genetic Algorithm* (Doctoral dissertation, Master Thesis-Department of Computer Engineering, Dalarna Univeristy, 3-4).

Piola, R., (1994, June). Evolutionary solutions to a highly constrained combinatorial problem. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence* (pp. 445-449). IEEE.

Robertus J. W. (2002). *School timetable construction-Algorithm and Complexity*, *1st ed. Netherlands*: University of Eindhoven Press

Tripathy, A., (1984). School timetabling—a case in large binary integer linear programming. *Management science*, *30*(12), pp.1473-1489.

Upagade, V. and Shende, A., (2009). Research Methodology, S. *Chand & Company Ltd., New Delhi*.

Yen, D.C. and Davis, W.S., (1998). *The Information System Consultant's Handbook*: Systems Analysis and Design. CRC press.

# APPENDICES

## APPENDIX 1: SQL Query

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

-- --------------------------------------------------------

--

-- Table structure for table `allowed`

--

DROP TABLE IF EXISTS `allowed`;

CREATE TABLE IF NOT EXISTS `allowed` (

`course\_id` char(10) NOT NULL,

`batch\_name` varchar(30) NOT NULL,

`batch\_dept` char(5) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `batches`

--

DROP TABLE IF EXISTS `batches`;

CREATE TABLE IF NOT EXISTS `batches` (

`batch\_name` varchar(30) NOT NULL,

`batch\_dept` char(5) NOT NULL,

`size` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `config`

--

DROP TABLE IF EXISTS `config`;

CREATE TABLE IF NOT EXISTS `config` (

`Name` varchar(30) NOT NULL,

`value` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `courses`

--

DROP TABLE IF EXISTS `courses`;

CREATE TABLE IF NOT EXISTS `courses` (

`course\_id` char(10) NOT NULL,

`course\_name` varchar(100) NOT NULL,

`fac\_id` char(25) NOT NULL,

`allow\_conflict` tinyint(1) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `depts`

--

DROP TABLE IF EXISTS `depts`;

CREATE TABLE IF NOT EXISTS `depts` (

`dept\_code` char(5) NOT NULL,

`dept\_name` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `faculty`

--

DROP TABLE IF EXISTS `faculty`;

CREATE TABLE IF NOT EXISTS `faculty` (

`uName` char(25) NOT NULL,

`fac\_name` varchar(50) NOT NULL,

`pswd` char(64) NOT NULL,

`level` enum('dean','hod','faculty','') NOT NULL DEFAULT 'faculty',

`dept\_code` char(5) NOT NULL,

`dateRegd` char(25) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `lecturers`

--

DROP TABLE IF EXISTS `lecturers`;

CREATE TABLE IF NOT EXISTS `lecturers` (

`lec\_id` varchar(10) NOT NULL,

`lec\_name` varchar(20) NOT NULL,

`dept` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `rooms`

--

DROP TABLE IF EXISTS `rooms`;

CREATE TABLE IF NOT EXISTS `rooms` (

`room\_name` varchar(25) NOT NULL,

`capacity` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `slots`

--

DROP TABLE IF EXISTS `slots`;

CREATE TABLE IF NOT EXISTS `slots` (

`table\_name` varchar(30) NOT NULL,

`day` int(1) unsigned NOT NULL,

`slot\_num` int(2) unsigned NOT NULL,

`state` enum('active','disabled') NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `slot\_allocs`

--

DROP TABLE IF EXISTS `slot\_allocs`;

CREATE TABLE IF NOT EXISTS `slot\_allocs` (

`table\_name` varchar(30) NOT NULL,

`day` int(1) unsigned NOT NULL,

`slot\_num` int(2) unsigned NOT NULL,

`room` varchar(25) NOT NULL,

`course\_id` char(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `students`

--

DROP TABLE IF EXISTS `students`;

CREATE TABLE IF NOT EXISTS `students` (

`student\_id` varchar(30) NOT NULL,

`dept` varchar(20) NOT NULL,

`email` varchar(20) NOT NULL,

`total` varchar(25) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `timetables`

--

DROP TABLE IF EXISTS `timetables`;

CREATE TABLE IF NOT EXISTS `timetables` (

`table\_name` varchar(30) NOT NULL,

`days` int(11) NOT NULL DEFAULT '5',

`slots` int(11) NOT NULL DEFAULT '0',

`duration` int(11) NOT NULL DEFAULT '90',

`start\_hr` char(2) NOT NULL DEFAULT '08',

`start\_min` char(2) NOT NULL DEFAULT '30',

`start\_mer` enum('AM','PM') NOT NULL DEFAULT 'AM',

`allowConflicts` tinyint(1) NOT NULL DEFAULT '0',

`frozen` tinyint(1) NOT NULL DEFAULT '0',

`active` tinyint(1) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Indexes for dumped tables

--

--

-- Indexes for table `allowed`

--

ALTER TABLE `allowed`

ADD PRIMARY KEY (`course\_id`,`batch\_name`,`batch\_dept`),

ADD KEY `course\_id` (`course\_id`),

ADD KEY `batch\_name` (`batch\_name`,`batch\_dept`);

--

-- Indexes for table `batches`

--

ALTER TABLE `batches`

ADD PRIMARY KEY (`batch\_name`,`batch\_dept`),

ADD KEY `batches\_department` (`batch\_dept`);

--

-- Indexes for table `config`

--

ALTER TABLE `config`

ADD PRIMARY KEY (`Name`);

--

-- Indexes for table `courses`

--

ALTER TABLE `courses`

ADD PRIMARY KEY (`course\_id`),

ADD KEY `fac\_id` (`fac\_id`);

--

-- Indexes for table `depts`

--

ALTER TABLE `depts`

ADD PRIMARY KEY (`dept\_code`);

--

-- Indexes for table `faculty`

--

ALTER TABLE `faculty`

ADD PRIMARY KEY (`uName`),

ADD KEY `dept\_code` (`dept\_code`);

--

-- Indexes for table `lecturers`

--

ALTER TABLE `lecturers`

ADD PRIMARY KEY (`lec\_id`);

--

-- Indexes for table `rooms`

--

ALTER TABLE `rooms`

ADD PRIMARY KEY (`room\_name`);

--

-- Indexes for table `slots`

--

ALTER TABLE `slots`

ADD PRIMARY KEY (`table\_name`,`day`,`slot\_num`);

--

-- Indexes for table `slot\_allocs`

--

ALTER TABLE `slot\_allocs`

ADD PRIMARY KEY (`table\_name`,`day`,`slot\_num`,`room`),

ADD KEY `fk\_course\_id` (`course\_id`),

ADD KEY `fk\_room` (`room`),

ADD KEY `fk\_slot` (`day`,`slot\_num`);

--

-- Indexes for table `students`

--

ALTER TABLE `students`

ADD PRIMARY KEY (`student\_id`);

ALTER TABLE `timetables`

ADD PRIMARY KEY (`table\_name`);

--

ALTER TABLE `allowed`

--

ALTER TABLE `slot\_allocs`

ADD CONSTRAINT `fk\_course` FOREIGN KEY (`course\_id`) REFERENCES `courses` (`course\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `fk\_room` FOREIGN KEY (`room`) REFERENCES `rooms` (`room\_name`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `fk\_slot` FOREIGN KEY (`table\_name`, `day`, `slot\_num`) REFERENCES `slots` (`table\_name`, `day`, `slot\_num`) ON DELETE CASCADE ON UPDATE CASCADE;

## APPENDIX 2: Sample Code for Timetable Management

<?php

require\_once('functions.php');

if(!sessionCheck('level','dean'))

{

header("Location: ./login.php");

die();

}

require\_once ('connect\_db.php');

?>

<!DOCTYPE HTML>

<html>

<head>

<title>Schedular</title>

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

<link rel="shortcut icon" type="image/png" href="images/favicon.png"/>

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

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

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

<script type="text/javascript" src="js/jquery.min.js" ></script>

<script type="text/javascript" src="js/form.js"></script>

<script type="text/javascript" src="js/chosen.js"></script>

<script>

$(function()

{

$("#main\_menu a").each(function() {

if($(this).prop('href') == window.location.href || window.location.href.search($(this).prop('href'))>-1)

{

$(this).parent().addClass('current');

document.title+= " | " + this.innerHTML;

$("#shadowhead").html(this.innerHTML);

return false;

}

})

$("select").chosen();

$("#fac\_level").change(function(){

$("input[value="+ $("option:selected",this).attr('class') +"]",this.parentNode).attr('checked','checked');

})

})

</script>

</head>

<body style="white-space:nowrap">

<div id="header">

<div id="account\_info">

<link rel="stylesheet" href="../assets/bower\_components/bootstrap/dist/css/bootstrap.min.css">

</div>

<div id="header\_text">Schedular</div>

</div>

<div id="shadowhead"></div>

<div id="nav\_bar">

<ul class="main\_menu" id="main\_menu">

<li class="limenu"><a href="dashboard.php">Home/ Control Panel</a></li>

<div class="inline">

<input type="radio" class="styled" name="level" id="addDean" value="dean"><label for="addDean">Dean</label>

</div>

<span class="inline stretch"></span

<td>'.$row['size'].'</td>

</tr>';

}

?>

</tbody>

</table>

</div>

</div>

<script src="../assets/js/dataTable/jquery.dataTables.min.js"></script>

<?php elseif (valueCheck('action','students')) : ?>

<div class="box">

<div class="boxbg"></div>

<div class="information"><div class="icon add"></div></div>

<div class="title">Add Students</div>

<div class="elements">

<form method="post" action="students.php?action=add">

<select name="student\_id" class="stretch" data-placeholder="Choose Course ID..." required>

<option label="Choose Course..."></option>

<?php

foreach($db->query('SELECT \* FROM courses') as $cId)

echo "<option value=\"{$cId['course\_id']}\">{$cId['course\_id']}</option>";

?>

</select>

<!-- <input type="text" name="department" class="styled uInfo" required pattern="[^:]{2,30}" title="2 to 30 alphanumeric characters" placeholder="Batch Name" /> -->

<select name="dept" class="stretch" data-placeholder="Choose Department..." required>

<option label="Choose Department..."></option>

<?php

foreach($db->query('SELECT \* FROM depts') as $dept)

echo "<option value=\"{$dept['dept\_code']}\">{$dept['dept\_name']}</option>";

?>

</select>

<input type="email" name="email" class="styled details" placeholder="Email" />

<input type="text" name="total" class="styled details" required pattern="[0-9]{1,3}" title="Number less than 1000, this will be used to suggest rooms" placeholder=" Total Number" />

<div class="blocktext info"></div>

<div class="center button">

<button>Add</button>

</div>

</form>

</div>

</div>

<div class="box">

<div class="boxbg"></div>

<div class="information"><div class="icon remove"></div></div>

<div class="title">Delete Students</div>

<div class="elements">

<form method="post" action="students.php?action=delete" class="confirm">

<select name="course\_id" class="updateSelect stretch" data-placeholder="Choose course code..." required>

<option label="Choose course code..."></option>

<?php

foreach($db->query('SELECT \* FROM students') as $student)

echo "<option value=\"{$student['student\_id']} : {$student['dept']}\">{$student['student\_id']}</option>";

?>

</select>

<input type="hidden" id="confirm\_msg" value="Are you sure you want to delete the selected Student Group?">

<div class="blocktext info"></div>

<div class="center button">

<button>Delete</button>

</div>

</form>

</div>

</div>

<div class="col-sm-10 col-sm-offset-1">

<link rel="stylesheet" href="../assets/bower\_components/bootstrap/dist/css/bootstrap.min.css">

</form>

</div>

</div>

<div class="col-sm-10 col-sm-offset-1">

<link rel="stylesheet" href="../assets/bower\_components/bootstrap/dist/css/bootstrap.min.css">

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700,300italic,400italic,600italic">

<div class="panel panel-primary" class="col-sm-10 col-sm-offset-1">

<div class="panel-body">

<div class="table-responsive table-sorting">

<table class=" table table-bordered table-striped" id="tSortable22">

<thead>

<tr style="width:auto">

<th style="width:1%">Lecturer Number.</th>

<th style="width:1%">Name</th>

<th style="width:1%">Department</th>

</tr>

</thead>

<tbody>

<?php

$query = $db->prepare("SELECT \* FROM lecturers");

$query->execute([$\_GET['lec\_id'],$\_GET['lec\_name'],$\_GET['dept']]);

while ($row = $query->fetch(PDO::FETCH\_ASSOC))

{

echo '<tr>

<td>'.$row['lec\_id'].'</td>

<td>'.$row['lec\_name'].'</td>

<td>'.$row['dept'].'</td>

</tr>';

}

?>

</tbody>

</table>

</div>

</div>

<script src="../assets/js/dataTable/jquery.dataTables.min.js"></script>

<?php else: ?>

<div class="box">

<div class="boxbg"></div>

<div class="information"><div class="icon add"></div></div>

<div class="title">Add Room</div>

<div class="elements">

<form method="post" action="rooms.php?action=add">

<input type="text" name="room\_name" class="styled details" required pattern="[^:]{2,25}" title="2 to 25 alphanumeric characters" placeholder="Room Name" />

<input type="text" name="capacity" class="styled details" required pattern="[0-9]{1,3}" title="Number

<thead>

<tr style="width:auto">

<th style="width:1%">Room No.</th>

<th style="width:1%">Capacity</th>

</tr>

</thead>

<tbody>

<?php

$query = $db->prepare("SELECT \* FROM rooms");

$query->execute([$\_GET['room\_name'],$\_GET['capacity']]);

while ($row = $query->fetch(PDO::FETCH\_ASSOC))

{

echo '<tr>

<td>'.$row['room\_name'].'</td>

<td>'.$row['capacity'].'</td>

</tr>';

}

?>

</tbody>

</table>

</div>

</div>

</div>

<?php endif; ?>

<script src="../assets/js/dataTable/jquery.dataTables.min.js"></script>

</body>

</html>

## APPENDIX 3: Questionnaire

* Tick appropriately

Gender: Male Female

Age: …………

Do you like the current scheduling system? Yes No

If yes, what are some of the reasons?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Would you propose to the school to change the scheduling procedures?

Yes No

Have you interacted with such a system before? Yes No

## APPENDIX 4: Project Budget

The following table below represents the costs that was incurred during and throughout the implementation of this system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ITEM** | **BRAND NAME** | **QUANTITY** | **DESCRIPTION** | **COST (KSHS.)** |
| Smartphone | Huawei Y5 Prime 2018 | 1 Piece | Communication, Hotspot | 11,000 |
| Laptop | Acer Aspire | 1 Piece | 4GB, 500GB | 25,000 |
| Internet | Telkom Kenya |  | Data Connection | 500 |
| Miscellaneous |  |  |  | 5,000 |
| **TOTAL** |  |  |  | **41,500** |

Table 1: Project Budget

## APPENDIX 5: Project Activities

Project activities involve all the actions and processes that took place during the development of system. These activities include: Feasibility study, Study investigation, System analysis, System design, Coding, Testing, Implementation of the project and then System documentation. The time is the duration that it will take for each task to be completed or done.

Table 2: Project Activities

|  |  |
| --- | --- |
| **ACTIVITY** | **TIME** |
| Feasibility Study | 2 Week |
| Study | 4 Weeks |
| System Analysis | 2 Weeks |
| System Design | 4 Weeks |
| Coding | 6 Weeks |
| Testing | 2 Weeks |
| Implementation | 2 Weeks |
| Documentation | - |
| **Total** | **22 Weeks** |

The Gant Chart below highlights the processes tabled above.

Table 3: Gant Chart

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activities** | **2** | **4** | **6** | **8** | **10** | **12** | **14** | **16** | **18** | **20** |
| Feasibility study |  |  |  |  |  |  |  |  |  |  |
| Study |  |  |  |  |  |  |  |  |  |  |
| System Analysis |  |  |  |  |  |  |  |  |  |  |
| System Design |  |  |  |  |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |  |  |  |  |
| Documentation |  |  |  |  |  |  |  |  |  |  |