**DESIGN AND IMPLEMENTATION OF A TIMETABLE SCHEDULING SYSTEM FOR NILE UNIVERSITY OF NIGERIA**

**BY**

**TAMUNOKORITE VICTOR BRIGGS**

**19212016**

**A PROJECT SUBMITTED TO THE DEPARTMENT OF SOFTWARE ENGINEERING AND INFORMATION TECHNOLOGY, FACULTY OF NATURAL AND APPLIED SCIENCES, NILE UNIVERSITY OF NIGERIA, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING, NILE UNIVERSITY OF NIGERIA**

**JUNE 2023**

NILE UNIVERSITY OF NIGERIA

**STATUS CONFIRMATION FOR BACHELOR’S PROJECT**

**LECTURE SCHEDULING SYSTEM FOR NILE UNIVERSITY OF NIGERIA**

**ACADEMIC SESSION : 2022/2023**

I, **TAMUNOKORITE VICTOR BRIGGS** agree to allow this Degree’s Project to be kept at the Library under the following terms:

1. This Degree Project is the property of Nile University of Nigeria.
2. The library has the right to make copies for educational purposes only.
3. The library is allowed to make copies of this report for educational exchange between higher educational institutions.
4. \*\* Please Mark (√)



|  |  |  |
| --- | --- | --- |
| **k** | CONFIDENTIAL | (Contains information of high security or of great importance to Nigeria as STIPULATED under the OFFICIAL SECRET ACT ) |
| **k** | RESTRICTED | (Contains restricted information as determined by the Organization/institution where research was conducted) |
| **l** | FREE ACCESS |  |

|  |  |  |
| --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Approved by,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (WRITER’S SIGNATURE) |  | (SUPERVISOR’S SIGNATURE) |
| Permanent Address :  1, Paul Unongo Street, GRA, Gboko, Benue State, Nigeria  Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Supervisor’s name:  Dr. Bilkisu Muhammad-Bello  Date : 21st June, 2023 |

# **CERTIFICATION**

This is to certify that, the project titled “DESIGN AND IMPLEMENTATION OF A LECTURE SCHEDULING SYSTEM FOR NILE UNIVERSITY OF NIGERIA” by TAMUNOKORITE VICTOR BRIGGS 191212016 has been approved by the undersigned for meeting the requirements for the award of Bachelor of Science in Software Engineering (BSc. Hons in Software Engineering by the department of Software Engineering and Information Technology, Nile University of Nigeria, Abuja.

……………………………….

Dr. Bilkisu Muhammad-Bello

PROJECT SUPERVISOR

……………………………………

Dr. Ibrahim Anka Salihu

HOD SOFTWARE ENGINEERING AND INFORMATION TECHNOLOGY

…………………………………….

Prof. Musa Muhammad

DEAN (FNAS)

..........................................................

EXTERNAL EXAMINER

# **DECLARATION**

I hereby declare that the work in this project is my own except for quotations and summaries which have been duly acknowledged.

Student : …………………………………………………….

Date : …………………………………………………….

Supervisor : …………………………………………………….

DR. BILKISU MUHAMMAD-BELLO

# **DEDICATION**

I dedicate this study to God, my ever-supportive family, and all the lecturers who have taught me throughout my time at Nile university of Nigeria, most especially my supervisor, Dr. Bilkisu Muhammad-Bello, who supported me through this project, and Dr. Umar Adam Ibrahim, who has helped me over the years to become a better software engineer.

# **ACKNOWLEDGEMENT**

I would like to express gratitude to God for guiding me through this journey and providing for me. I would also like to thank my family for supporting me and being with me throughout. Finally, I would like to thank my project supervisor, Dr. Bilkisu Muhammad-Bello, whose guidance helped me complete this project.

# **ABSTRACT**

Timetabling is an important component of the educational system at all levels. With the introduction of technology, a lot of the processes in the educational system have been automated and made more efficient. However, timetabling still remains an issue. The timetabling process is a tedious one and the current system is inefficient and does not produce optimal results. The application of technology to the process of timetabling will go a long way in making the process more efficient and more effective. This project proposes a web-based timetable generation system which will implement the Genetics Algorithm to optimize the timetable generation process. The system developed implements a customized version of the Genetics Algorithm, to fit the school’s constraints, and is able to generate a timetable with no clashes.

**TABLE OF CONTENTS**

[**CERTIFICATION** I](#_Toc138113226)

[**DECLARATION** II](#_Toc138113227)

[**DEDICATION** III](#_Toc138113228)

[**ACKNOWLEDGEMENT** IV](#_Toc138113229)

[**ABSTRACT** V](#_Toc138113230)

[**CHAPTER ONE** 1](#_Toc138113231)

[1.1 Background of the Study 1](#_Toc138113232)

[1.2 Statement of the Problem 3](#_Toc138113233)

[1.3 Significance of the Study 3](#_Toc138113234)

[1.4 Justification of the Study 4](#_Toc138113235)

[1.5 Aim and Objectives 4](#_Toc138113236)

[1.6 Scope and Limitation of the Study 4](#_Toc138113237)

[1.7 Definition of Operational Terms 5](#_Toc138113238)

[**CHAPTER TWO** 6](#_Toc138113239)

[2.1 General Information 6](#_Toc138113240)

[2.2 Related Works. 7](#_Toc138113241)

[2.3 Summary of the Review 12](#_Toc138113242)

[**CHAPTER THREE** 18](#_Toc138113243)

[3.1 System Analysis 18](#_Toc138113244)

[3.1.1 Analysis of the Existing System 18](#_Toc138113245)

[3.1.2 Limitation of the Existing System 18](#_Toc138113246)

[3.1.3 Justification for the New System 18](#_Toc138113247)

[3.1.4 Description of the New System 19](#_Toc138113248)

[3.2 Design of the Proposed System 19](#_Toc138113249)

[3.2.1 Data Model 19](#_Toc138113250)

[3.2.2 Functional Requirement (*Use Case Diagram*) 21](#_Toc138113251)

[3.2.3 System Architecture *(Deployment Diagram)* 23](#_Toc138113252)

[3.2.4 Software Structure *(Class Diagram)* 24](#_Toc138113253)

[3.2.5 Workflow of Use Cases (*Activity Diagram*) 25](#_Toc138113254)

[**CHAPTER FOUR** 27](#_Toc138113255)

[4.1 System Requirement for Development 27](#_Toc138113256)

[4.2 System Menus Implementation 27](#_Toc138113257)

[4.3 Database Implementation 34](#_Toc138113258)

[4.4 System Testing 37](#_Toc138113259)

[**CHAPTER FIVE** 40](#_Toc138113260)

[5.1 Summary of the study 40](#_Toc138113261)

[5.2 Conclusion and recommendations 40](#_Toc138113262)

[**REFERENCES** 41](#_Toc138113263)

[**APPENDIX** 43](#_Toc138113264)

**LIST OF TABLES**

[**Table 2. 1: Summary of Literature Review 12**](#_Toc136342232)

[**Table 4. 1: Test Cases 37**](#_Toc136342242)

[**Table 4. 2: Test Results 39**](#_Toc136342242)

**LIST OF FIGURES**

[**Figure 3.1: Entity Relationship Diagram of the Proposed System 21**](#_Toc136342232)

[**Figure 3.2: Use Case Diagram for the Lecture Scheduling System 23**](#_Toc136342242)

[**Figure 3.3: Deployment Diagram for the Lecture Scheduling System 24**](#_Toc136342242)

[**Figure 3.4: Class Diagram for the Lecture Scheduling System 25**](#_Toc136342242)

[**Figure 3.5: Activity Diagram for the Lecture Scheduling System 26**](#_Toc136342242)

**Figure 4.1: LSS Registration Page**  [**28**](#_Toc136342242)

**Figure 4.2: LSS Login Page**  [**28**](#_Toc136342242)

**Figure 4.3: Input validation on LSS Login Page**  [**29**](#_Toc136342242)

**Figure 4.4: List of Departments in the system**  [**29**](#_Toc136342242)

**Figure 4.5: List of Lecturers in the Software Engineering Department**  [**30**](#_Toc136342242)

**Figure 4.6: List of Buildings in the System**  [**30**](#_Toc136342242)

**Figure 4.7: List of Courses in the system**  [**31**](#_Toc136342242)

**Figure 4.8: List of Physics 101 (PHY101) Sections**  [**31**](#_Toc136342242)

**Figure 4.9: List of the Types of Labs in the System**  [**32**](#_Toc136342242)

**Figure 4.10: Form for Adding Departments**  [**32**](#_Toc136342242)

**Figure 4.11: Form for Adding Courses**  [**33**](#_Toc136342242)

**Figure 4.12: Form for Editing Courses**  [**33**](#_Toc136342242)

**Figure 4.13: The system informs the Super Administrator that the Timetable Generation Process has begun**  [**34**](#_Toc136342242)

**Figure 4.14: LSS Directory Structure showing MVC architecture implementation**  [**35**](#_Toc136342242)

**Figure 4.15: The Timetable Generation Algorithm is implemented as a job on the backend**  [**36**](#_Toc136342242)

**Figure 4.14: The generated timetable and other metadata is printed in the server’s log file upon completion**  [**36**](#_Toc136342242)

# **CHAPTER ONE**

INTRODUCTION

*Timetabling is important to every institution. The issue of creating an optimal timetable is a very critical one, and there are specific constraints and rules applied to it in every institution. This chapter discusses the problems with the current timetabling system at Nile University of Nigeria, and introduces a solution that aims to solve this problem.*

1. **Background of the Study**

Effective lecture scheduling is a vital component that supports the smooth operation and success of any academic term in educational institutions. A properly organized timetable created at the start of the term is crucial as it heavily influences and molds both students' and lecturers' activities throughout the term.

An extensive analysis of a wide range of factors and restrictions is required to develop a complete and useful timetable. These include essential components including the availability of suitable lecturers, the assignment of classrooms or venues, and the avoidance of any conflicts or overlaps in the scheduling of lessons for students. The process of developing a timetable seeks to strike a delicate balance by taking these numerous factors into account, resulting in an arrangement that is ideal and facilitates efficient teaching and learning.

Today's market is filled with software tools for creating schedules. The products in question, however, are not designed to meet the institutional requirements that apply in educational institutions (for instance, if a school mandates that all GST courses be taken online). As a result, there are more people involved in the process of creating schedules, which poses new problems. Human involvement in the process opens the door to potential oversight and mistakes, which could result in unintentional violations of restrictions and disparities in the timelines that are produced. As a result, it generally takes several iterations and subsequent adjustments to arrive at a finished and satisfactory timetable. When more people are involved in the process, it takes longer and costs more money. It can also cause disruptions to the activities of students and lecturers if the timetable is not established in a timely manner.

Students in several departments at the school still don't know whether version of their timetable is official or final three weeks into the majority of academic terms, demonstrating the inefficiencies of the institution's existing approach.

The implementation of a digital Lecture Scheduling System (LSS) appears as a promising approach to address these challenges and improve the accuracy and efficiency of the timetable development process at Nile University of Nigeria. The Lecture Scheduling System transforms how timetables are created at the school, by utilizing cutting-edge algorithms and processing power. By automating the process, the LSS makes sure that timetables are generated meticulously while sticking to the established constraints and considerations at the school. The scheduling process is greatly simplified by this automation, which also lessens the need for extensive manual modifications and decreases the possibility of errors.

This digital Lecture Scheduling System (LSS) at Nile University in Nigeria has the potential to completely alter the way that schedules are made, resolving many of the issues with the current system. The LSS equips the school to develop precise, conflict-free, and constraint-compliant timetables by utilizing the power of algorithms and computational intelligence. The benefits that follow include increased operational effectiveness, decreased administrative burden, improved resource utilization, and a more supportive learning environment for both students and lecturers. In the end, the incorporation of a digital lecture scheduling system has the promise of maximizing the efficiency and productivity of academic terms, creating a richer educational experience for all stakeholders.

* 1. **Statement of the Problem**

The school's current system for creating timetables has serious inefficiencies. The procedure takes a long time, and as a result, timetables are usually not complete, even at the beginning of the term. Additionally, the system is vulnerable to unnoticed constraint violations because of its reliance on manual labor. Sadly, it is the lecturers and students who suffer the most from these flaws because they are forced to adjust their schedules in order to account for the faults of the current timetable generation system.

The current system is inefficient partly because of the dependence on manual labour, which necessitates complex coordination and several iterations. Due to the significant risk of human error and oversight in these processes, timetables may be incorrect or incomplete. As a result, it falls on students and lecturers to handle last-minute changes and resolve schedule issues that result from these restrictions.

An inadequate or defective schedule has consequences that go beyond discomfort. Organization of study schedules, attendance in class, and involvement in extracurricular activities are all difficult for students. Creating course materials, working with colleagues, and ensuring that students have the best possible learning experience are all challenges faced by lecturers. The ensuing interruptions and annoyances can have a negative effect on the educational environment as a whole and prevent students and lecturers from achieving their academic objectives.

* 1. **Significance of the Study**

The proposed lecture schedule generation system, which is specifically designed for Nile University of Nigeria and takes into account the institution's unique constraints, will significantly increase the efficiency of the timetable development process by addressing the flaws of the current system. The early release of final timetables will make it possible for students and lecturers to effectively organize their activities for the term, which will increase their faith in the system and increase student and lecturer confidence.

* 1. **Justification of the Study**

The current system of timetable creation at the school is not efficient, and the consequences of its inefficiency have been felt by the students and lecturers; thus, there is a need for a new system that automatically generates correct timetables for every semester using data like room availability, lecturer availability, courses, course sections, and so on. This will enable the system to create a timetable that takes into account all the important constraints that are considered when creating timetables at the university.

* 1. **Aim and Objectives**

The aim of this study is to improve the efficiency of the timetable creation process at the school by developing a web-based Lecture Scheduling System for the school.

The objectives of this research work are:

I. To design a web-based Lecture Scheduling System that will generate timetables for academic terms.

II. To implement the designed Lecture Scheduling System for the school.

III. To test the system and evaluate its efficiency in generating timetables for the school.

* 1. **Scope and Limitation of the Study**

The scope of this study is to develop a Lecture Sduling System (LSS) for Nile University of Nigeria. It will be a web-based system that automates the process of timetable creation and allows students and lecturers to view and download the generated timetable. The system will also make sure the timetable does not have any clashes or violations of the specified constraints.

The limitations of the proposed system are:

1. The system only solves the problem of lecture scheduling and will not have capability for exam scheduling.

2. The system is web based, meaning it can only be accessed by users with an internet connection.

* 1. **Definition of Operational Terms**

1 **Timetable**: It is a calendar that organizes the classes and time periods of students and teachers during the day.

2 **Algorithm**: This is a step-by-step procedure for solving a particular problem in a finite amount of time.

3 **Database**: A centralized repository of structured data, designed to support efficient data storage, retrieval and maintenance.

4 **Lecture Scheduling System (LSS)**: A web-based system that will automate the process of timetable creation by using an algorithm to generate a timetable without clashes and constraint violations.

# **CHAPTER TWO**

LITERATURE REVIEW

*Literature review is important in any academic research, as it provides examination of existing works and knowledge on the topic at hand. This chapter discusses some of the works related to automated timetable generation.*

1. **General Information**

The difficult process of allocating lectures, meetings, and classes to certain classrooms, timeslots, and days is known in the world of education as timetabling. It is a rigorous and time-consuming operation, despite the fact that it may appear simple at first. The timetabling method is particularly specialized to each use case and institution because each educational establishment has its own distinct set of guidelines and constraints that must be taken into account.

Regrettably, manual timetable generating techniques are still widely used in many universities. The academic term's schedule must be developed following extensive deliberation between faculty administration and other concerned parties. The time and effort required for this manual technique appears to be unproductive because they may be better spent on more important duties.

Some universities use Commercial Off-The-Shelf (COTS) timetabling software in an effort to lessen the human workload. These systems, however, frequently lack the flexibility needed to accommodate the unique constraints of various institutions. As a result, institutions, including the subject of this study, are forced to use manual techniques to hone and enhance the timetables produced by COTS software. This not only defeats the automation's stated goal but also drags out the timetabling procedure and makes the problems worse.

Researchers have proposed the idea of an automated lecture scheduling system to address these problems by streamlining and automating the process of creating timetables by utilizing the power of contemporary computing. While factors like permitting flexibility for manual revisions, creating user-friendly interfaces, and addressing other important difficulties have been highlighted in past research, a variety of algorithms have been investigated to optimize the development of timetables.

Educational institutions can modernize timetabling by using an automated lecture scheduling system. The physical labor involved in developing and perfecting timetables may be reduced with the use of sophisticated algorithms and computing approaches. With the help of this automation, institutions are able to efficiently create timetables while taking into account their own constraints and preferences.

* 1. **Related Works**

A previous study (Sutoyo & Mangkona, 2022) proposed a university course planning system based on a modified version of the Euclidian Distance Algorithm in a bid to develop a lecture scheduling system that has no collisions. Euclidian Distance is an algorithm used to find the distance between two different points. The study uses the Euclidian Distance formula with a distribution of course tutors and the schedule of available rooms. The results of the study show that this modified Euclidian Distance Algorithm works well in scheduling lectures at the university.

Another study (Dahlan, et al., 2019), explores the option of using the Welsh Powell Graph Coloring Algorithm. The results of the research revealed that this algorithm can solve the lecture scheduling problem well without clashes. It also reveals that the amount of data fed into the algorithm greatly influences its processing time.

A study from Southeast University (Islam, Shahriar, Perves, & Hassan, 2016) aimed to find a suitable system for generation of timetables for courses and exams and minimize the constraints in the problem space and find a good solution. The researchers gathered and analyzed the requirements for the system and decided to apply the Tabu Search algorithm to solve the problem. Tabu Search is a metaheuristic search method for solving mathematical optimization problems. The research concludes that the Tabu Search algorithm can give the best optimal solution to the problem or at least, a solution which is optimally close to the best. It also concludes that the algorithm can be implemented in the solution of the problem easily.

In another study (Ebieto, 2018) the researcher aims to develop software that will optimally create timetables for educational institutions. The system uses Dynamic Slot Algorithm (DSA) combined with the application of Constraint Satisfaction Programming. The system also provides a flexible interface for users. The study concludes that the system performs satisfactorily in solving the timetabling problem, and it has been used by the Faculty of Engineering at the University of Port Harcourt.

A study at the Saboo Siddik College of Engineering, Mumbai (Sayed, Ahmed, Aamir, & Zaeem, 2015) , reiterates the issues with manual timetable scheduling, and proposes an automated system to solve these issues. The researchers classify the problem as a constraint satisfaction problem. The researchers identified the constraints by asking stakeholders questions. The paper explores the usage of a Genetic Algorithm, which is a type of Evolutionary Algorithm. The Genetic Algorithm was combined with a heuristic specific greedy algorithm which resulted in a decrease in execution times.

Another study (Patil, Khichi, Jadhav, & Basa, 2021) uses a customized algorithm to solve the timetabling problem. This algorithm, developed by the researchers, takes input, classifies them as hard or soft constraints, and applies some rules to them to generate a timetable. The system generates an optimal timetable.

The use of a customized algorithm to solve the problem of timetable generation is also proposed by another study (Amarnadh, Hemanthi, Sravanthi, & Sanketh, 2020) .This study highlights the importance of timetabling, the shortcomings of the manual system of timetabling, and suggests an automated timetable generator. The proposed web-based system makes use of a custom algorithm that takes in input and applies some rules within specified constraints on the input data and generates a timetable. The study concludes that the proposed system reduces the time taken during timetable generation and generates a more optimal timetable.

In a paper titled “Timetable Generator” (Markal, Ghorpade, & Chalke, 2020), researchers state the problems associated with manual timetable creation, and propose a system that will generate a timetable using the Genetic algorithm. The aim of the study is to develop an efficient application that will make timetable generation and distribution much easier. The Genetic Algorithm used by the system ensures that clashes are eliminated. The study concludes that the proposed system will make the process of timetable generation much more efficient and easier.

A study (Kumar, Singh, & Sharma, 2013) discusses the problem of timetable generation at universities. The paper explores the option of using Particle Swarm Optimization (PSO) to provide an optimal solution to the problem of timetabling at Punjabi University, Patiala. The study concludes that PSO is a promising approach for solving the timetabling problem, and the results of the experiments satisfied both teachers and students.

The use of Graph Coloring technology is explored in a study (Gajbhiye, Shende, Sahu, & Behar, 2018) at College of Engineering, Nagpur, Maharashtra, India. The study takes a user-friendly approach to implement the graph coloring algorithm in the solution of the timetabling problem. The results of testing the proposed system were successful, and the study concludes that the graph coloring algorithm is capable of generating a timetable without conflicts.

The Genetics Algorithm (GA) is also explored as a possible solution to the problem in another paper (Brahmbatt, Patel, Prajapati, Gevariya, & George, 2022). The study examines the current manual system of timetabling and states its flaws. An automatic timetable generator using Genetics Algorithm (GA) is proposed to solve this problem. The study concludes that the system will function faster and better in generating timetables for educational institutions.

A study on timetable generation (Puttaswamy, Khan, V, & A., 2018), emphasizes how tedious the manual timetable generation process is, and explores a few algorithms and approaches that could be applied to solve the problem. The study highlights the following techniques to solve the problem:

• Simulated Annealing (SA).

• Tabu Search.

• Genetic Algorithm (GA).

• Crossover.

• Mutation.

• Fitness Function.

• Heuristics.

All of these approaches are examined towards the development of an automatic timetable generator.

A timetabling mechanism using python is proposed in this study (Kaur, Bhatti, Kakkar, & Goyal, 2019). The study introduces the problem to the reader and defines the scope. The study proposes a system built in python, implementing the Genetics Algorithm, to solve the timetabling problem. The study concludes that the proposed system, compared to manual timetable generation, satisfies all the challenges in the process of timetabling.

Another study (Techie-Menson & Nyagorme, 2021) focuses on the design and implementation of a web-based timetable system for higher institutions. In the study, Rapid Application Development (RAD) was used to develop the system. The study also explores various techniques for timetable development which include:

• Graph Heuristics.

• Hill Climbing (HC).

• Tabu Search.

• Simulated Annealing (SA).

• Genetic Algorithms (GA).

The study concludes that the web-based system has the ability to produce near-optimal timetables.

The last reviewed study (Abduljabbar & Abdullah, 2021) explores the use of an Evolutionary Algorithm to solve the timetable scheduling problem. The study aims to optimize the Genetics Algorithm (GA) to solve the timetabling problem and to generate timetables with no clashes. The study concludes that the approach is indeed effective in solving the timetabling problem.

* 1. **Summary of the Review**

Table 2.1: *Summary of Literature Review*

|  |  |  |
| --- | --- | --- |
| TITLE | **SUMMARY** | **LITERATURE GAP** |
| Sutoyo, M. N., & Mangkona, A. T. (2022). Implementing the Modified Euclidean Distance Method in the Course Planning of the USN Kolaka Information. *IT Journal Research and Development (ITJRD).* | This study aims to prove that a  modified version of the Euclidean  Distance algorithm can solve the  problem of lecture scheduling. The  results of the study show that the  Euclidean Distance Algorithm can in  fact, be used to solve the problem  of lecture scheduling without  collisions (clashes). | * No consideration of course sections. |
| Dahlan, A., Nurdini, Marsali, Y.,  Heri, S., Sepyan, P. K., Hendra, N.,  . . . Ketut, S. I. (2019).    Lecture Scheduling System Using  Welch Powell Graph Coloring  Algorithm in Informatics  Engineering Departement of  Universitas Malikussaleh. *Journal of*  *Physics: Conference Series, 7.* | This study explores the possibility  of using a graph coloring algorithm  called “Welsh Powell Graph  Coloring Algorithm” to solve the  problem of lecture scheduling. The  study concludes that the algorithm  can solve the lecture scheduling  problem adequately, without  clashes. | * No consideration of course sections. * Algorithm processing time is greatly influenced by size of input data. |
| Islam, T., Shahriar, Z., Perves, M., & Hassan, M. (2016). University Timetable Generator Using Tabu Search. *Journal of Computer and Communications.* | Finding an appropriate system to create course and exam schedules while avoiding restrictions was the main goal of the research. The Tabu Search algorithm was selected by the researchers after they considered the system's needs. According to the study's findings, the algorithm can deliver the best ideal solution or one that comes very near to it. The algorithm was also thought to be simple to apply to the issue at hand. | * No consideration of course sections. * The study takes lecturer’s preference into account, but the implementation does not guarantee consideration of this constraint. |
| Ebieto, C. (2018). Lecture Timetable Scheduling Software. *International Journal of Scientific and Research Publications.* | This study described the design of a system for optimal timetable creation for educational institutions using the Dynamic Slot Algorithm. The system was implemented at the University of Port Harcourt and the results obtained by the system were much better than those obtained by the manual process. | * No consideration of course sections. |
| Sayed, P. S., Ahmed, A., Aamir, A., & Zaeem, A. (2015). Automated Timetable Generator. *International Journal for Innovative Research in Science & Technology.* | This paper proposed the development of a practical timetable generation system using an Evolutionary Algorithm (EA), to mitigate the problems associated with the manual process. The system performed efficiently in solving the timetable generation problem. | * No consideration of course sections. |
| Patil, P. P., Khichi, J., Jadhav, M., & Basa, V. (2021). Automatic Timetable Generator. *International Research Journal of Engineering and Technology.* | In this study, researchers developed a custom algorithm to solve the timetabling problem. This algorithm, given data, was able to generate an optimal timetable based on the provided constraints. | * No consideration of course sections. |
| Amarnadh, M. V., Hemanthi, B., Sravanthi, G., & Sanketh, S. S. (2020). Automatic Timetable Generator. *International Journal of Research.* | This study analyzes the timetabling problem and various algorithms that have been implemented to solve it. The researchers built upon this past work by developing a hybrid algorithm to solve the timetabling problem. This hybrid algorithm worked efficiently to solve the timetabling problem. | * No consideration of course sections. |
| Markal, S., Ghorpade, S., & Chalke, D. (2020). Timetable Generator. *IOSR Journal of Computer Engineering.* | The difficulties of manually creating schedules are highlighted in this article, and researchers suggest a solution that uses the genetic algorithm to build schedules automatically. The goal is to create a useful application that makes the creation and dissemination of timetables simpler. Conflicts in the generated timetables are successfully eliminated by the system's Genetic Algorithm. The study comes to the conclusion that the proposed system greatly improves the effectiveness and simplicity of timetable generation. | * No consideration of course sections. |
| Kumar, A., Singh, K., & Sharma, N. (2013). Automated Timetable Generator Using Particle Swarm Optimization. *International Journal on Recent and Innovation Trends in Computing and Communication.* | This paper looks at the problem of creating class schedules in colleges and how Particle Swarm Optimization (PSO) could potentially be used as a solution. The study focuses particularly on the scheduling system at Punjabi University, Patiala. According to the study's findings, PSO has the potential to effectively address the timetabling issue, and the experimental outcomes were pleasing to the teachers and students who participated in the process. | * No consideration of course sections. |
| Gajbhiye, S., Shende, N., Sahu, U., & Behar, Y. (2018). Automated TImetable Generator for Educational Institutions using Graph Colouring Technology**.** *International Journal of Applied Research.* | This study explores the creation of an automated timetable generation system for educational institutions. To efficiently assign resources, including classrooms and timeslots for lectures and activities, the system makes use of graph coloring technology. The study's main goal is to increase the precision and efficacy of timetable generation procedures. The study emphasizes the advantages of the suggested system and how it may simplify the timetabling procedure in educational institutions. | * No consideration of course sections. |
| Brahmbatt, D., Patel, H., Prajapati, K., Gevariya, J., & George, D. (2022). Automatic Timetable Generation Using Genetics Algorithm. *Journal of Emerging Technologies and Innovative Research.* | This article describes a study on the creation of a system for automatically creating schedules using genetic algorithms. By using the Genetic Algorithm to optimize the distribution of resources including classrooms, timeslots, and faculty assignments, the research intends to solve the difficulties associated with manual timetable construction. The goal of the project is to increase the precision and effectiveness of timetable generation in educational institutions. The study demonstrates how well the suggested system generates timetables that adhere to various constraints and yield positive outcomes. | * No consideration of course sections. |
| Kaur, M., Bhatti, J., Kakkar, M. K., & Goyal, D. (2019). Timetable Handling Mechanism Using Python. *International Journal of Scientific & Technology Research.* | The study presents a Python-based solution in order to address the problems with manual timetable management. The objective of the project is to simplify the process of planning, putting together, and overseeing timetables at educational institutions. The study emphasizes how well the suggested approach manages timetables, makes the best use of resource allocation, and reduces conflicts. The paper advances the area by offering a useful and automated Python timetable management method. | * No consideration of course sections. |
| Techie-Menson, H., & Nyagorme, P. (2021). Design and Implementation of a Web-Based Timetable System for Higher Education Institutions. *International Journal of Educational Research and Information Science.* | The design and implementation of a web-based scheduling system especially suited for higher education institutions is the subject of the study presented in this article. By creating a web-based solution, the research focuses on addressing the drawbacks of manual timetable administration. The goal of the project is to develop a user-friendly interface that makes it simple for instructors, students, and administrators to access and manage timetables. In highlighting the system's characteristics and capabilities, the research emphasizes how well it can automate timetable development, resource allocation, and schedule administration. By providing a workable and user-friendly web-based solution for effective timetable administration in higher education institutions, the study makes a contribution to the field. | * No consideration of course sections. |
| Abduljabbar, I. A., & Abdullah, S. M. (2021). An Evolutionary Algorithm for Solving Academic Courses Timetable Scheduling Problem. *Baghdad Science Journal.* | This article presents a study on utilizing an evolutionary algorithm to solve the challenge of scheduling academic courses. The research focuses on solving the difficulties involved in manually creating timetables and suggests an evolutionary algorithm as a remedy. The goal of the study is to design a timetable that is effective and free of conflicts by allocating resources like classrooms, timeslots, and faculty assignments as efficiently as possible. The study demonstrates how well the suggested algorithm generates timetables that adhere to different limitations and deliver positive outcomes. The study makes a contribution to the subject by proposing an evolutionary algorithm as a solution to the problem of scheduling academic courses. | * No consideration of course sections. |

# **CHAPTER THREE**

SYSTEM ANALYSIS AND DESIGN

*Before the development of software, it is important to do system analysis, and design the system to be developed. This chapter discusses the analysis and design of the timetable scheduling system.*

1. **System Analysis**
   * 1. **Analysis of the Existing System**

The existing system of lecture scheduling at Nile University of Nigeria is inefficient and the school ends up having to fall back to a manual process, where various stakeholders (amongst the school staff), meet to deliberate and draft the timetables for various departments for the semester.

This is usually a tedious process, as some constraints may be violated during the process, and at the end of the day, the timetable may not be completely finalized when school resumes for the new semester. This affects the plans of students and lecturers, as they have to make adjustments to their own schedules to accommodate for the shortcomings of the existing system.

### **Limitation of the Existing System**

The limitations of the existing system include the following:

i. The process takes time and sometimes, delivery is late.

ii. It requires manpower which could be used for other tasks.

iii. Possibility of errors from human mistakes.

### **Justification for the New System**

The current system has shown how limited it is in producing results on time, and the consequences have been felt by lecturers and students for some time.

The new system will address the issues that plague the current system by generating timetables quickly, with limited human involvement (limited to data entry), thus greatly reducing the possibility of human errors.

### **Description of the New System**

The new system is an automatic timetable generator, which will collect the necessary data concerning lectures, rooms, courses and their sections, and lecturers, analyze them and create a timetable without clashes.

The system will allow students, lecturers, and all stakeholders view the timetable, once it is generated.

## **Design of the Proposed System**

### **Data Model**

The system’s database captures information that is vital for the successful execution of the system. The models for the database are:

1. User model: This model captures the information of the system’s registered users. Information stored in this table includes name, email address, and password.

2. Department Model: This model captures all the departments in the school. Each department has multiple lecturers and courses. Information stored in this model includes name, faculty, code, and the department’s priority in each building.

3. Course Model: This model captures the various courses offered in each department. Information stored in this model includes course title, number of credits, level, number of hours, and course code. The model also stores information that determines whether the course is a practical course or not, and if the course is a GST or not.

4. Lecturer Model: This model captures information of the lecturers in various departments. It captures the name of the lecturer.

5. Section Model: This model captures information of all the sections for each course. The information captured in this model includes the semester, academic year, size of the section, and section id.

6. LabType Model: This model captures information of all the types of labs available in the school.

7. Building Model: This model captures information of all the buildings in the school.

8. Room Model: This model captures the information of all the classrooms in the school. It includes information like room number, building, capacity, and an indicator of the room being a lab or not.

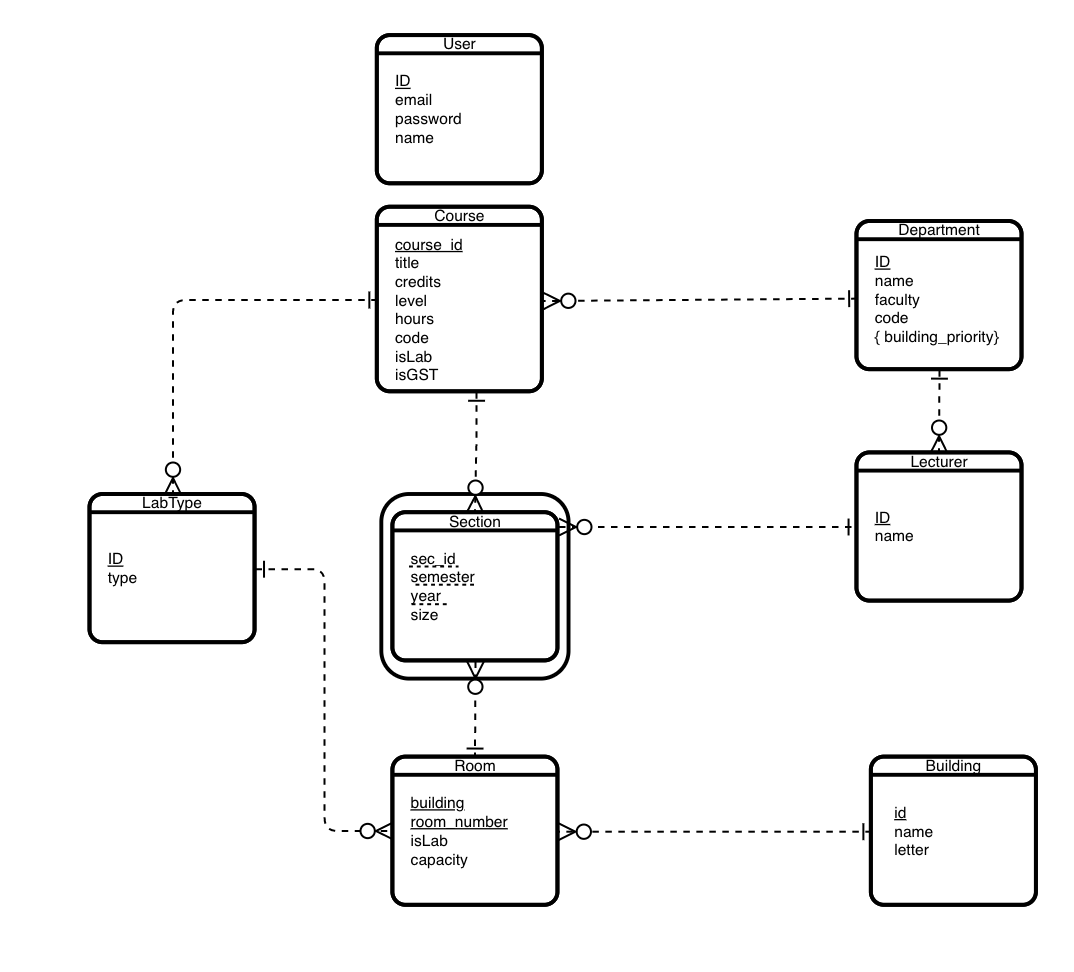


Figure 3.1: *Entity Relationship Diagram of the Proposed System*

### **Functional Requirement (*Use Case Diagram*)**

The functions available to the users of the system are:

1. Login/Register/Logout: Faculty administrators and the system’s super administrator can login and register on the system. They can also logout at any time they wish.

2. Manage Faculty Administrators: The super administrator can manage the faculty administrators in the system. This management includes functions like adding faculty administrators, editing faculty administrators’ information, and deleting faculty administrators.

3. Add Inputs: The faculty administrators can add the inputs to the system. These inputs include departments, buildings, courses, lecturers, and all the information required to generate timetables for the school.

4. Generate Timetable: The super administrator can generate the timetable once all of the inputs have been entered.

5. Approve Timetable: The super administrator can approve the timetable.

6. View Timetable: All stakeholders can view the generated timetable once it has been approved.

7. Edit Timetable: Faculty administrators can edit the timetable before it is approved.

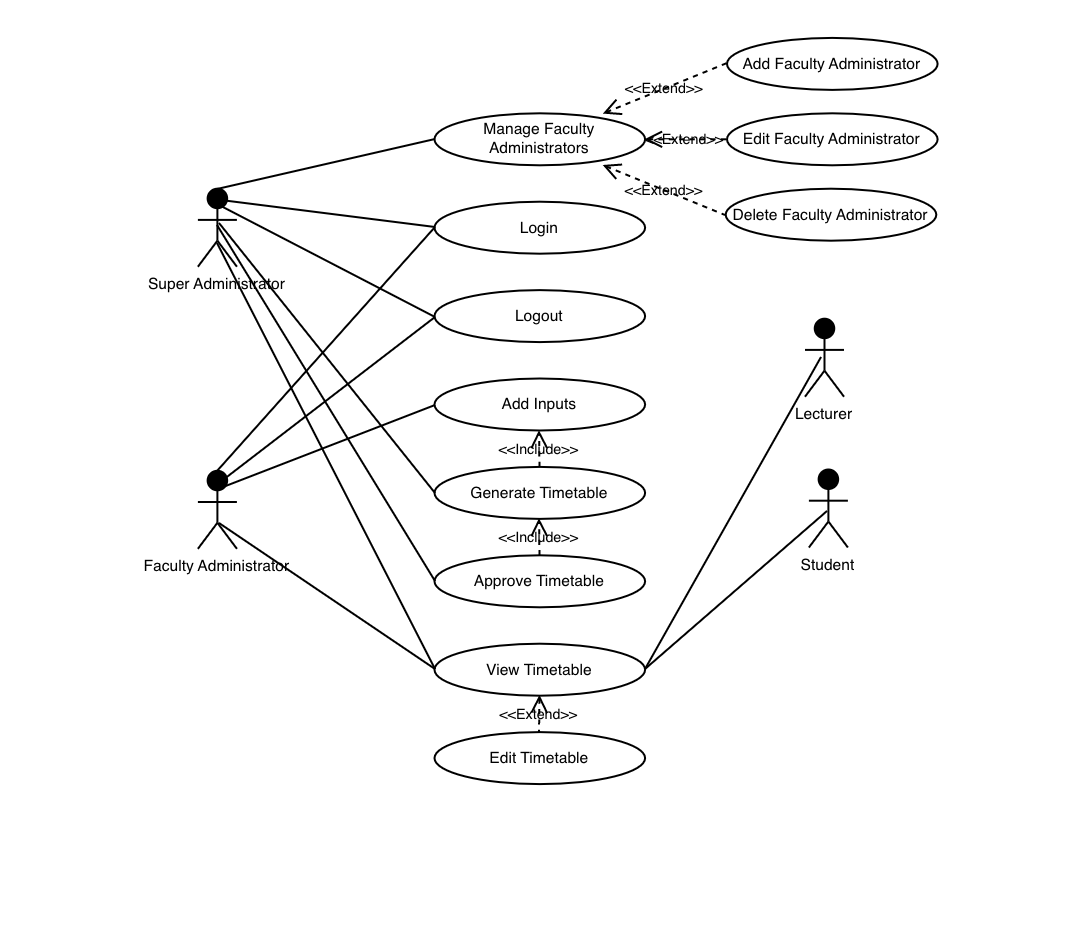


Figure 3.2: *Use Case Diagram for the Lecture Scheduling System*

### **System Architecture *(Deployment Diagram)***

The system will be deployed to an Apache web server, while the database will be hosted on a MySQL database server.

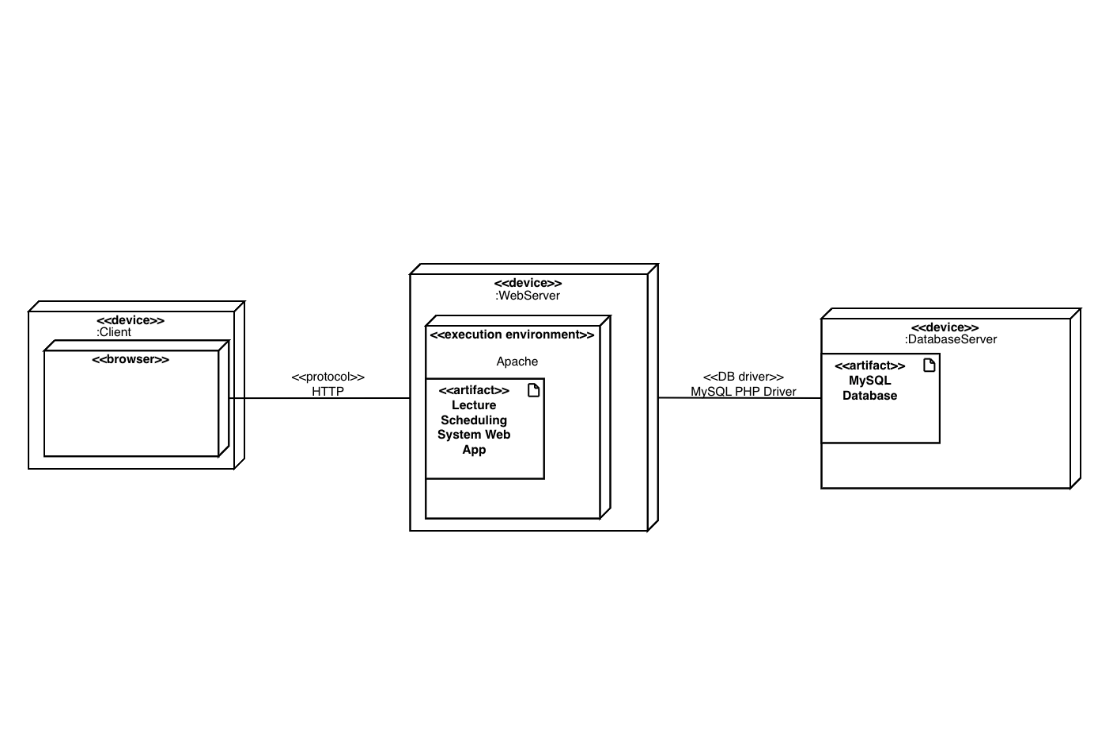


Figure 3.3: *Deployment Diagram for the Lecture Scheduling System*

### **Software Structure *(Class Diagram)***

The classes in the system include:

1. User: This class represents the registered users of the system.

2. SuperAdmin: This is a class that extends the user class and represents the system’s super administrator.

3. FacultyAdmin: This class extends the user class and represents the faculty administrators.

4. Lecturer: This class represents the lecturers.

5. Department: This class represents the departments in the school.

6. Course: This class represents the sections of all courses.

7. Room: This class represents all the classrooms in the school.

8. TimeSlot: This class represents the various time slots on the timetable.

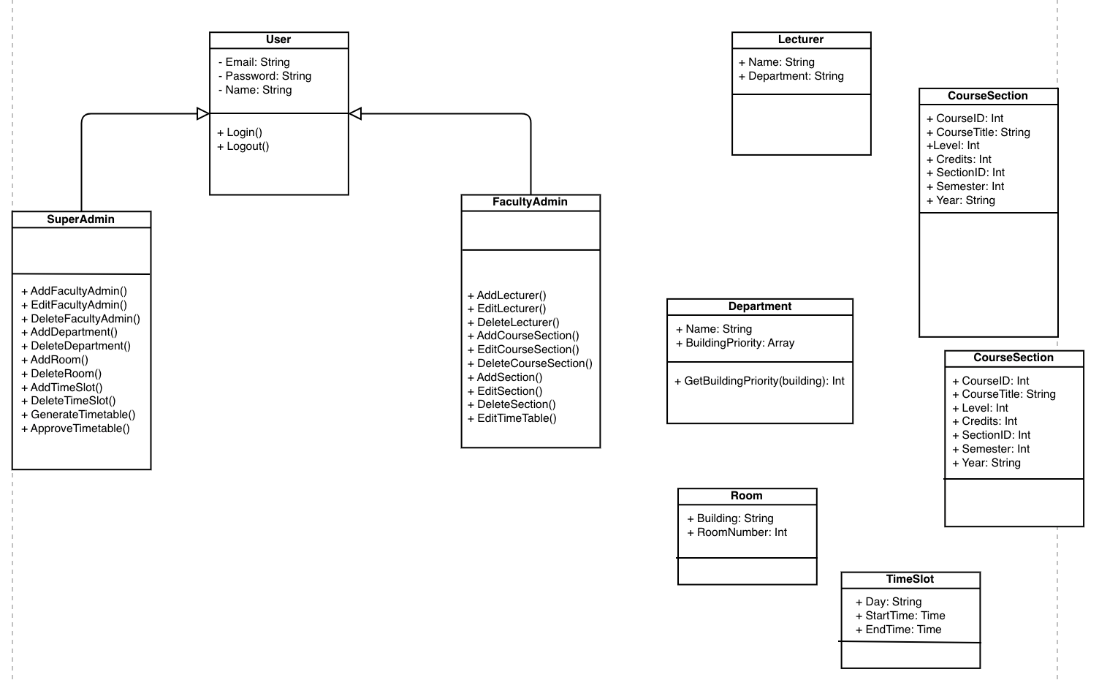


Figure 3.4: *Class Diagram for the Lecture Scheduling System*

### **Workflow of Use Cases (*Activity Diagram*)**

All users can view the timetable when it has been approved.

Eligible users (super admin and faculty administrators) can login to the system. If the wrong credentials are entered, the system returns an appropriate error message. After successful login, the system determines if a user is a super administrator or faculty administrator.

Super administrator can manage faculty administrators, generate the timetable if inputs have been uploaded, approve the timetable, view the timetable, and logout.

Faculty administrators can add inputs, view the timetable, edit the timetable, and logout.

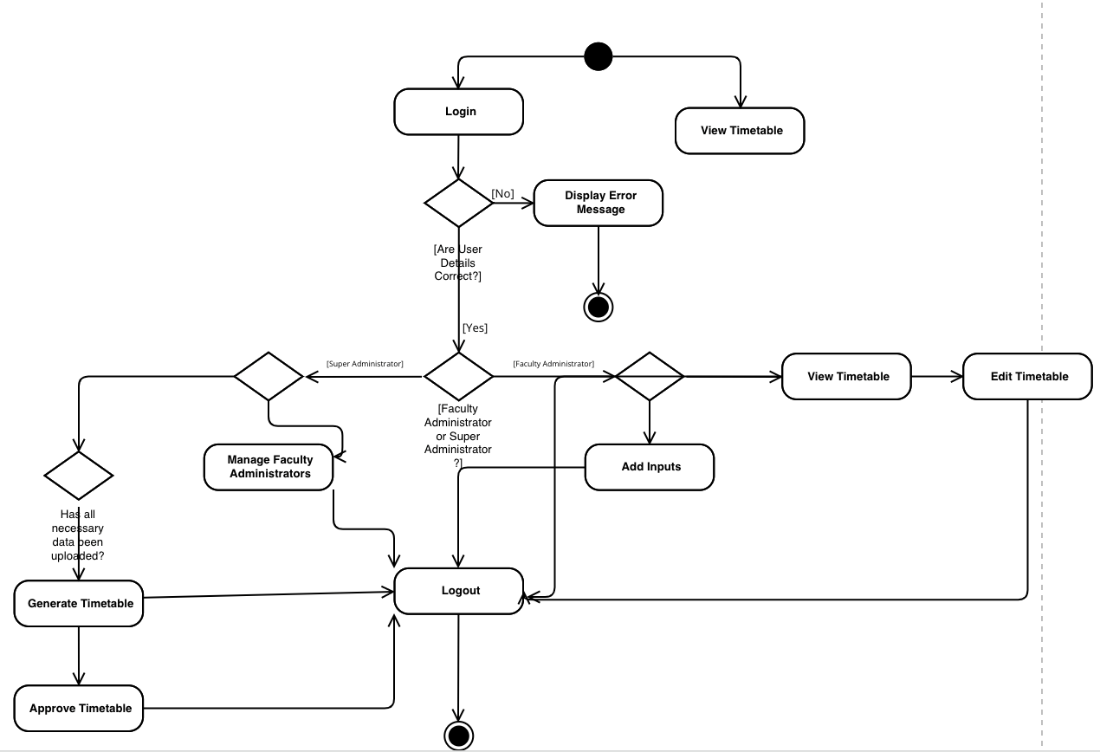


Figure 3.5: *Activity Diagram for the Lecture Scheduling System*

# **CHAPTER FOUR**

IMPLEMENTATION AND DISCUSSION

*In this chapter, the implementation details of the system are discussed. The tools used, menus implemented, and database implementation are discussed here.*

* 1. **System Requirement for Development**

The tools/languages used to develop the system include:

1. PHP: PHP is a popular scripting language used to develop web applications.

2. Laravel: Laravel is a full stack PHP web development framework used to develop the frontend and backend of web applications.

3. Blade: Blade is Laravel’s templating engine used for developing dynamic frontend interfaces.

4. MySQL: MySQL is a relational database system.

5. Visual Studio Code: Visual Studio Code is a text editor that is used for all forms of programming and development.

* 1. **System Menus Implementation**

The system provides interfaces for users to access and manipulate its information. Below are the interfaces provided by the system:

1. Register/Login: The system provides an interface for the super administrator and faculty administrators to login or register if they have not created accounts.

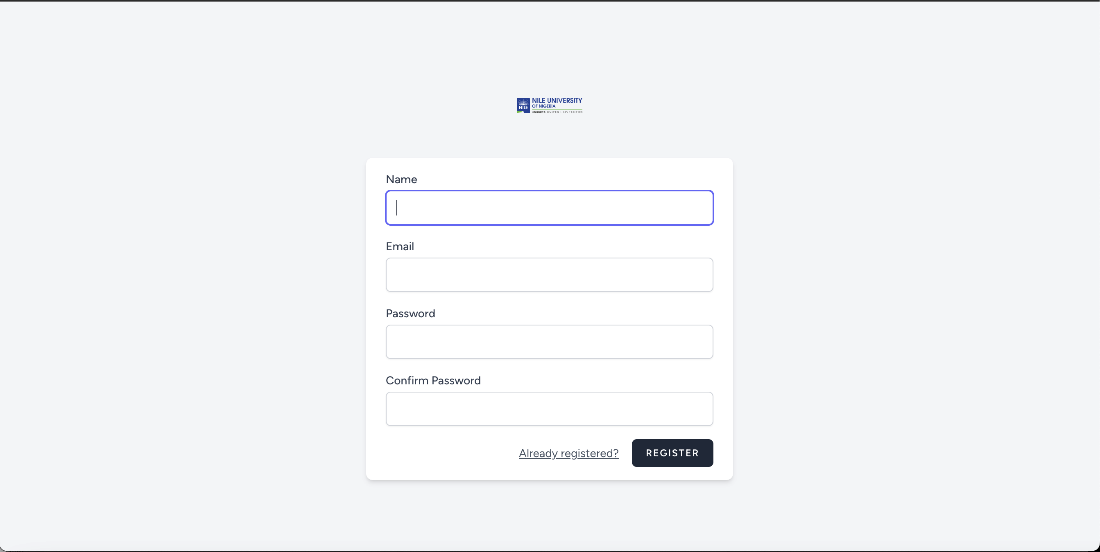


Figure 4.1: *LSS Registration Page*

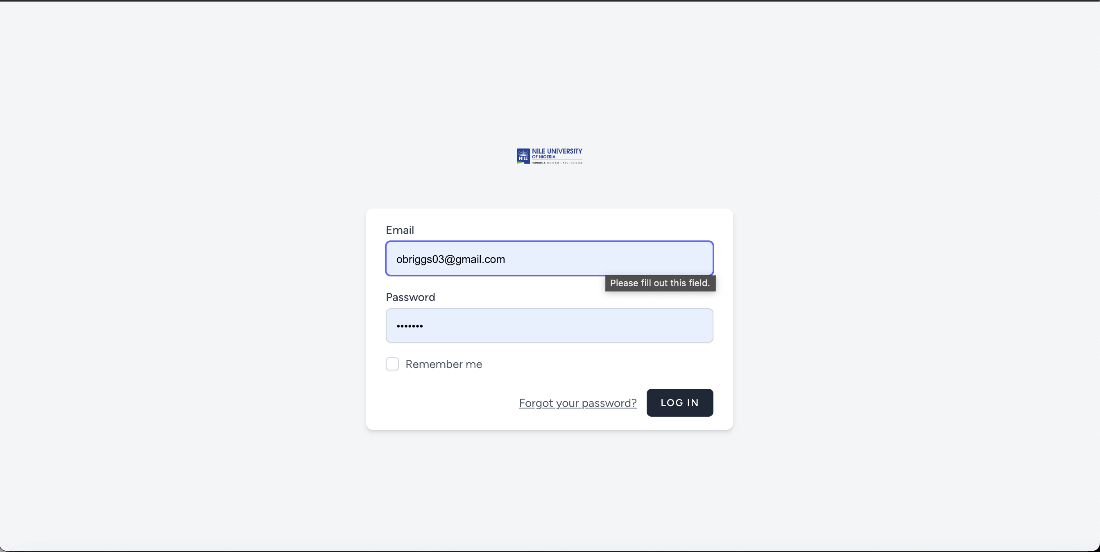


Figure 4.2: *LSS Login Page*

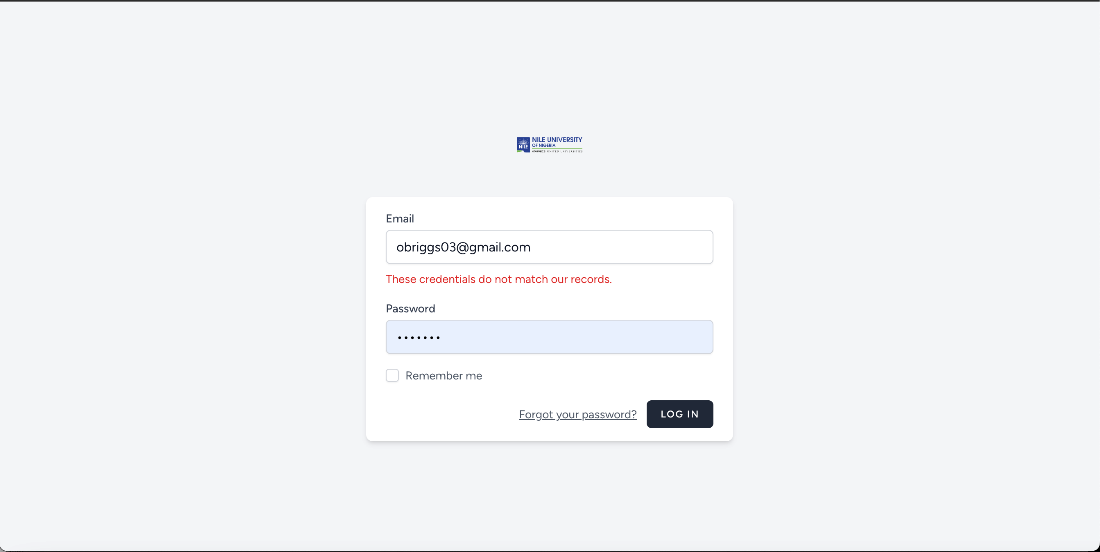


Figure 4.3: *Input validation on LSS Login Page*

1. View Inputs: The system provides an interface for administrators to view the inputs that are already in the system.

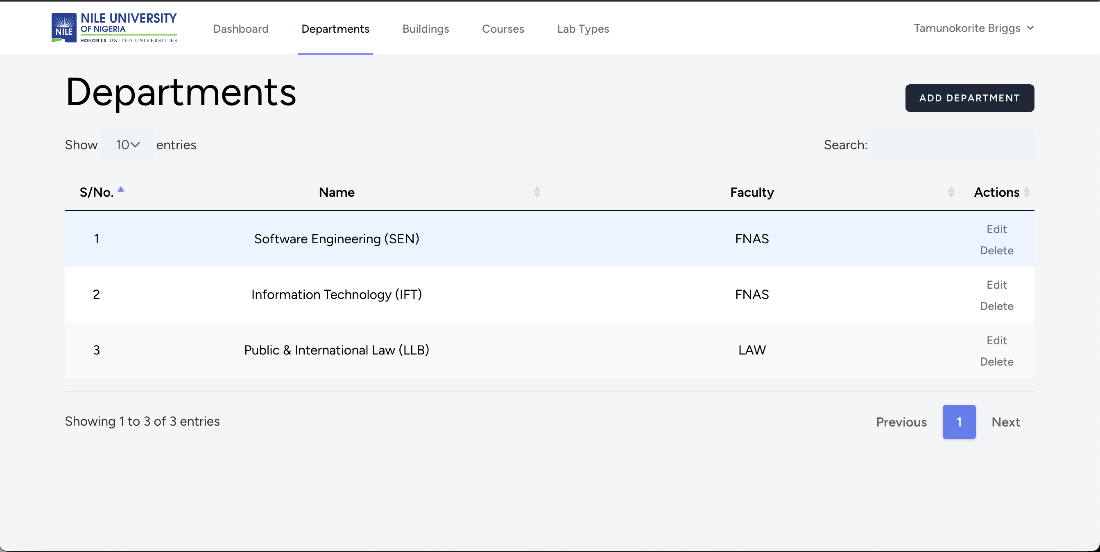


Figure 4.4: *List of Departments in the system*

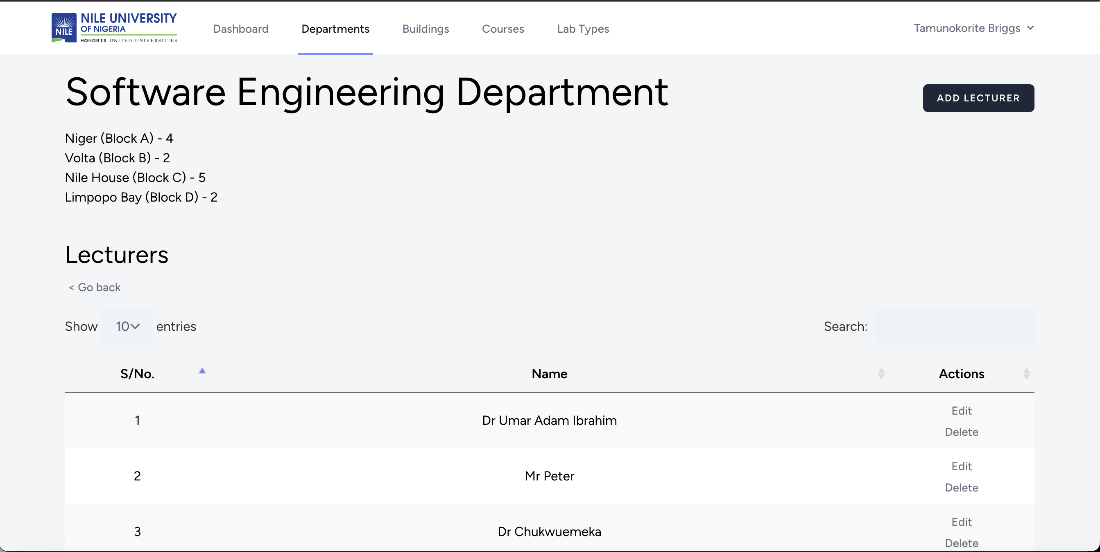


Figure 4.5: *List of Lecturers in the Software Engineering Department*

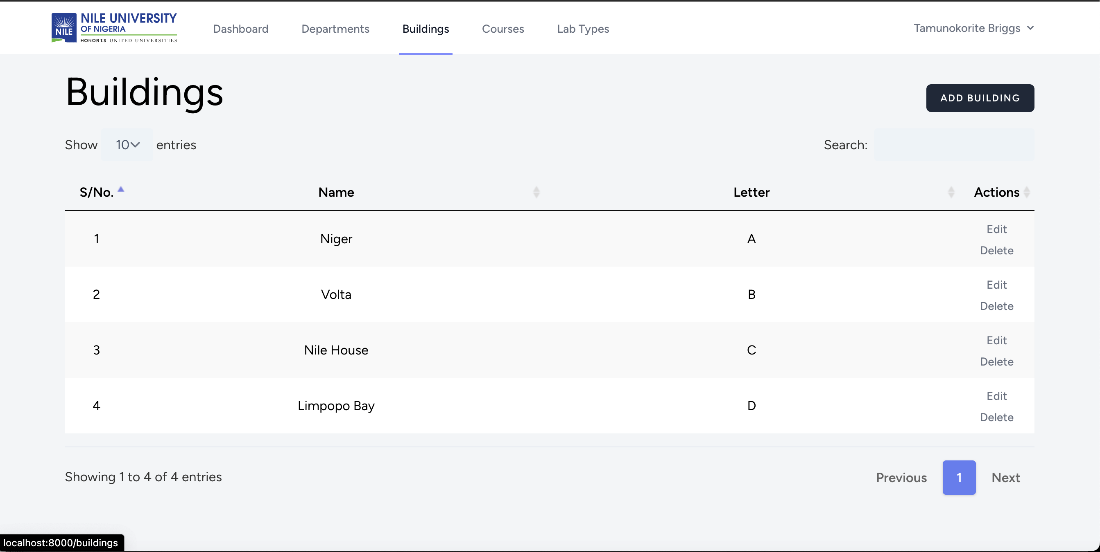


Figure 4.6: *List of Buildings in the System*

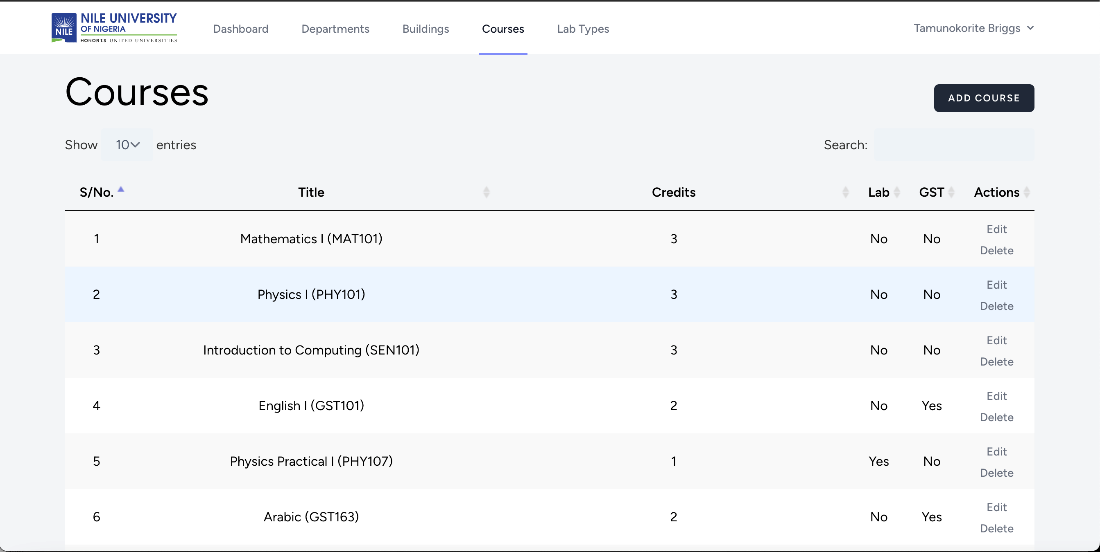


Figure 4.7: *List of Courses in the system*

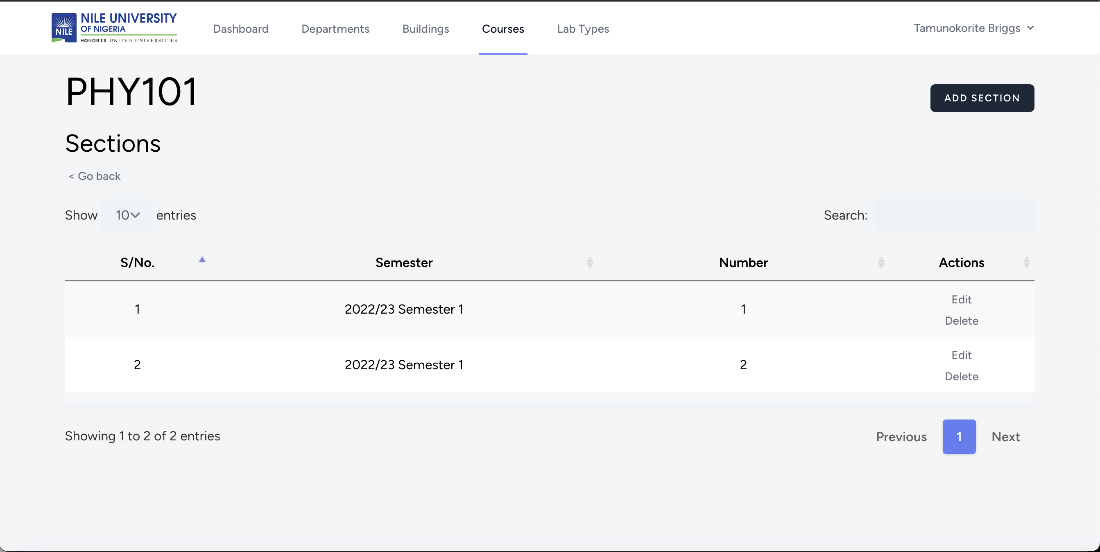


Figure 4.8: *List of Physics 101 (PHY101) Sections*

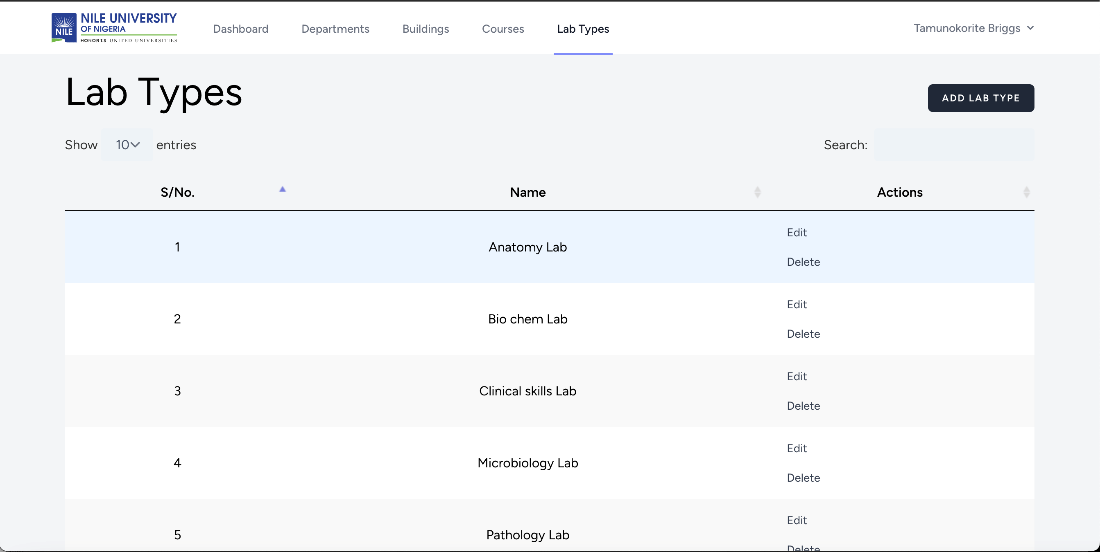


Figure 4.9: *List of the Types of Labs in the System*

1. Add Inputs: The system also provides an interface for administrators to add inputs into the system.

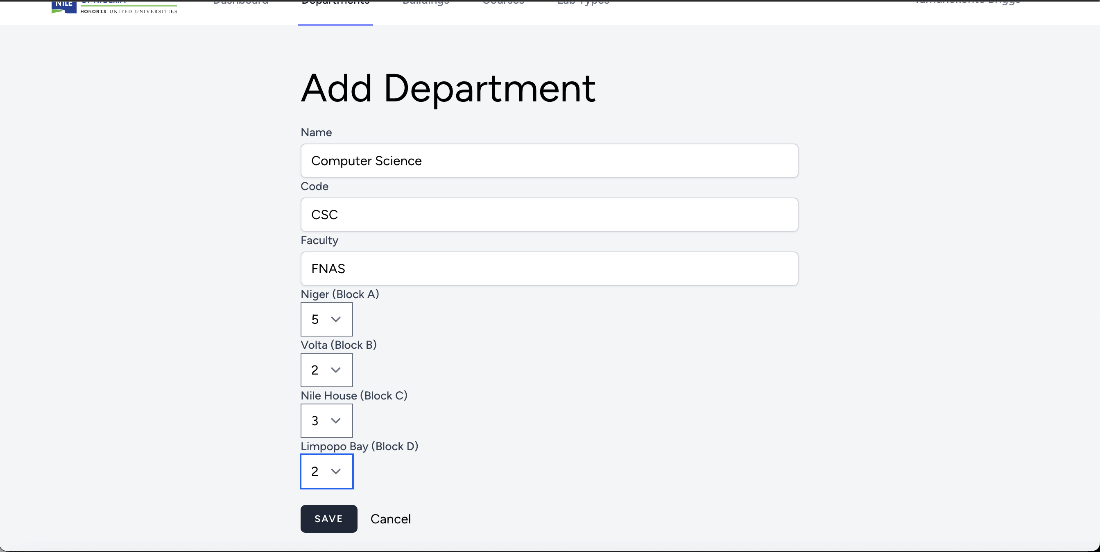


Figure 4.10: *Form for Adding Departments*

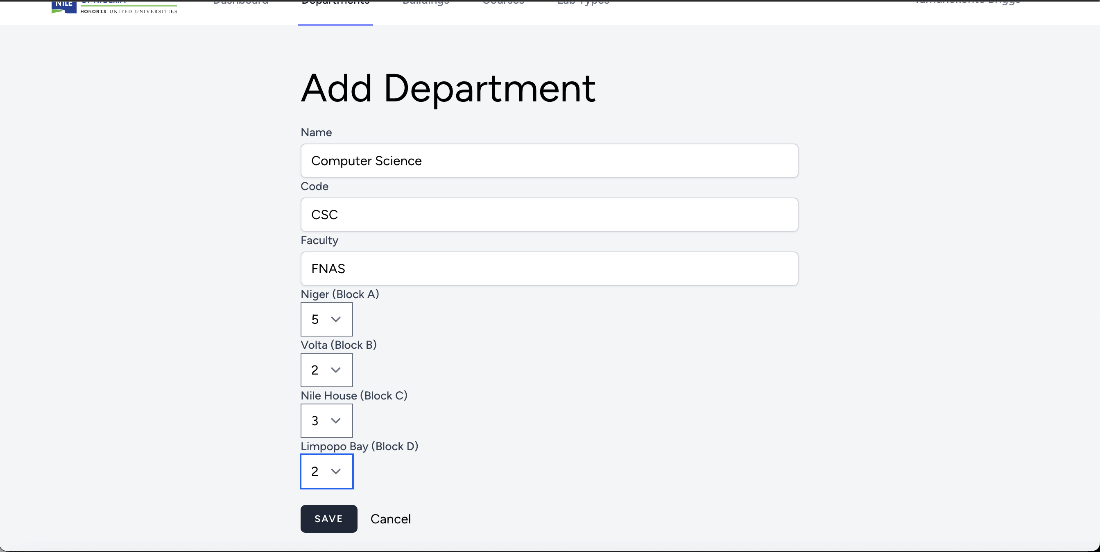


Figure 4.11: *Form for Adding Courses*

1. Edit Inputs: The system also provides an interface for administrators to edit the inputs.

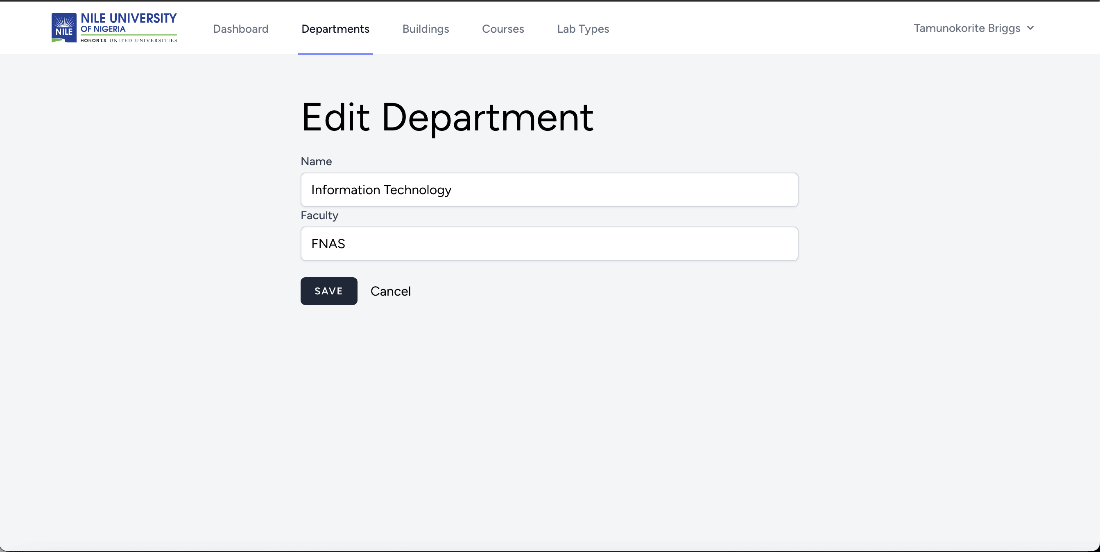


Figure 4.12: *Form for Editing Courses*

1. Delete Inputs: The system also allows administrators to delete inputs.
2. Generate Timetable: The system provides an interface for the super administrator to generate the timetable. This interface triggers an event in the system that starts the process of generating the timetable based on all the provided inputs.

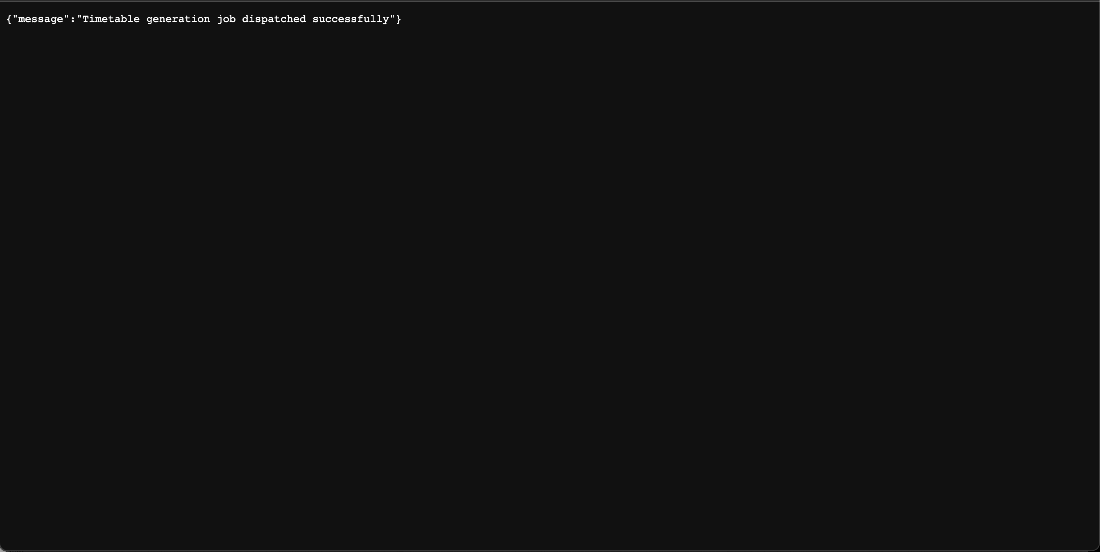


Figure 4.13: *The system informs the Super Administrator that the Timetable Generation Process has begun*

1. View Timetable: The system provides an interface for all stakeholders to view the timetable that has been generated.
   1. **Database Implementation**

The backend of the system is implemented using Model-View-Controller (MVC) Architecture. In this architecture, the Views represent the interface the user sees and uses to interact with the system. These views interface with the Controllers, which are used to access and manipulate the Models (the data models of the system). The controllers serve as the middleman between the Views and the Models.

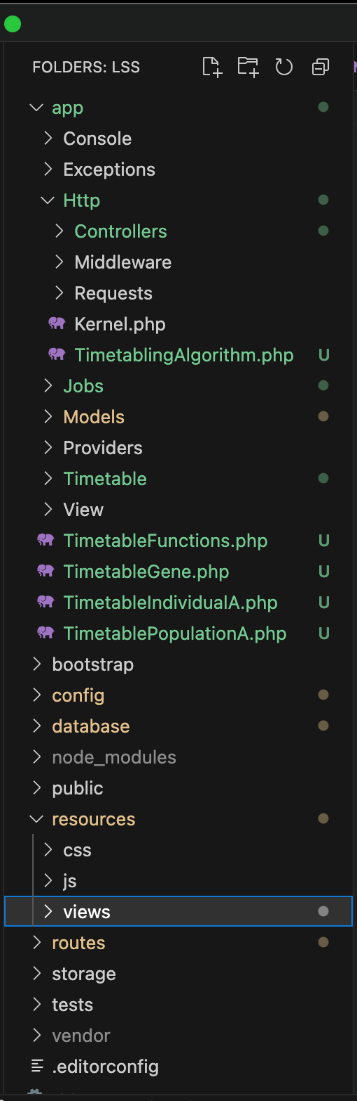


Figure 4.14: *LSS Directory Structure showing MVC architecture implementation*

The system uses an implementation of the Genetics Algorithm to generate the timetable upon user’s request. This algorithm is implemented on the backend as an asynchronous job, which is added to the queue of jobs the server is performing. The asynchronous nature of this job allows the system to continue executing other jobs while the timetabling algorithm is still executing. This allows the system to serve the users, even when the timetable is still generating. This implementation contributes to faster execution of the system, as the users are not interrupted, and are only alerted once the timetable is ready.

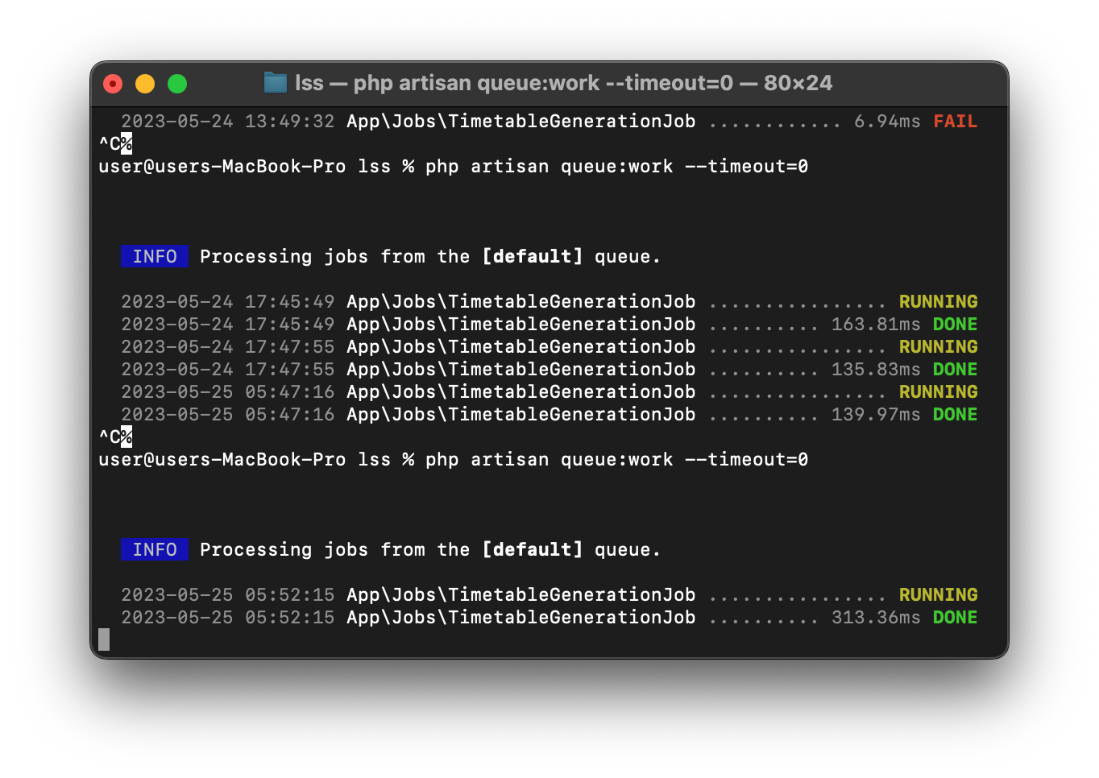


Figure 4.15: *The Timetable Generation Algorithm is implemented as a job on the backend*

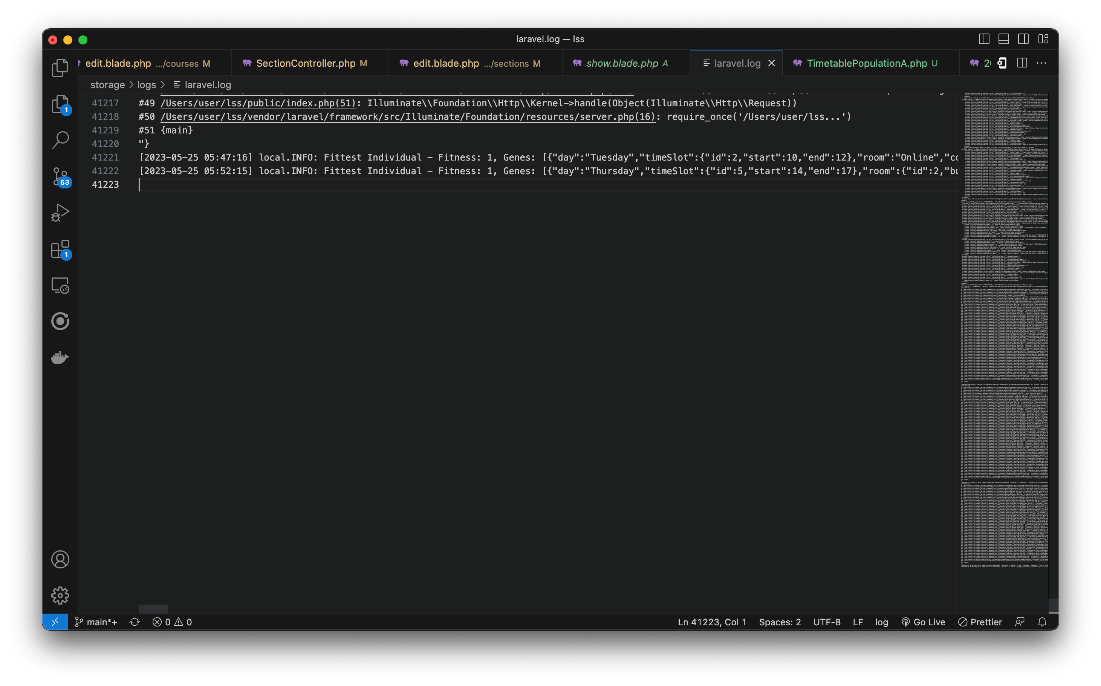


Figure 4.16: *The generated timetable and other metadata is printed in the server’s log file upon completion.*

* 1. **System Testing**

Table 4.1: *System Test Cases*

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Test Case ID** |
| **Login** | 1st Test - Login with wrong email address and wrong password. System should reject login and output an appropriate error message. | T001 |
|  | 2nd Test - Login with correct email address and wrong password. System should reject login and output an appropriate error message. | T002 |
|  | 3rd Test - Login with correct email address and correct password. System should log in user and redirect them to their dashboard. | T003 |
| **Registration** | 1st Test - Register with required information left blank. System should reject registration and output an appropriate error message. | T004 |
|  | 2nd Test – Register with all information entered. System should register user, log them in, and redirect them to their dashboard. | T005 |
| **Authorization** | 1st Test – View departments without authentication. User should be redirected to home page. | T006 |
|  | 2nd Test – View departments as a logged in user. System should respond with the page for viewing departments. | T007 |
|  | 3rd Test – Add department without authentication. System should not allow user access this view but redirect them to the home page. | T008 |
|  | 4th Test – Add department as a logged in user. System should allow request. | T009 |
|  | 5th Test – Edit department without authentication. System should not allow user access this view but redirect them to the home page. | T010 |
|  | 6th Test – Edit department as a logged in user. System should allow request. | T011 |
|  | 7th Test – Delete department without authentication. System should not allow user access this view but redirect them to the home page. | T012 |
|  | 8th Test – Delete department as a logged in user. System should allow request. | T013 |
| **Add Inputs** | 1st Test – Add department with required information left blank. System should reject creation and output an appropriate error message. | T014 |
|  | 2nd Test – Add department with all information provided. System should create department and output redirect user to the department list view. | T015 |
| **Generate Timetable** | 1st Test – Go to generate timetable link. System should respond with a message indicating that the timetable generation process has started. | T016 |

Table 4.2: Test Results

|  |  |
| --- | --- |
| **Test ID** | **Result** |
| T001 | PASSED |
| T002 | PASSED |
| T003 | PASSED |
| T004 | PASSED |
| T005 | PASSED |
| T006 | PASSED |
| T007 | PASSED |
| T008 | PASSED |
| T009 | PASSED |
| T010 | PASSED |
| T011 | PASSED |
| T012 | PASSED |
| T013 | PASSED |
| T014 | PASSED |
| T015 | PASSED |
| T016 | PASSED |

# **CHAPTER FIVE**

CONCLUSION AND RECOMMENDATION

*This chapter presents a summary of the content of this document. It also concludes the study, and offers recommendations based on the study.*

## 5.1 **Summary of the study**

This project proposes a web-based Lecture Scheduling System made specifically for Nile University of Nigeria, to make the process of timetable generation at the school more efficient by incorporating the school/s specific constraints for timetable generation, and generating a timetable, within minutes, without clashes and constraint violations.

## 5.2 **Conclusion and recommendations**

The developed system works efficiently to generate timetables for Nile University of Nigeria.

I recommend the adoption of the system by the school, as it will go a long way in improving the entire academic information system at the school.

# **REFERENCES**

Abduljabbar, I. A., & Abdullah, S. M. (2021). An Evolutionary Algorithm for Solving Academic Courses Timetable Scheduling Problem. *Baghdad Science Journal*.

Amarnadh, M. V., Hemanthi, B., Sravanthi, G., & Sanketh, S. S. (2020). Automatic Timetable Generator. *International Journal of Research*.

Brahmbatt, D., Patel, H., Prajapati, K., Gevariya, J., & George, D. (2022). Automatic Timetable Generation Using Genetics Algorithm. *Journal of Emerging Technologies and Innovative Research*.

Dahlan, A., Nurdini, Marsali, Y., Heri, S., Sepyan, P. K., Hendra, N., . . . Ketut, S. I. (2019). Lecture Scheduling System Using Welch Powell Graph Coloring Algorithm in Informatics Engineering Departement of Universitas Malikussaleh. *Journal of Physics: Conference Series*.

Ebieto, C. (2018). Lecture Timetable Scheduling Software. *International Journal of Scientific and Research Publications*.

Gajbhiye, S., Shende, N., Sahu, U., & Behar, Y. (2018). Automated TImetable Generator for Educational Institutions using Graph Colouring Technology. *International Journal of Applied Research*.

Islam, T., Shahriar, Z., Perves, M., & Hassan, M. (2016). University Timetable Generator Using Tabu Search. *Journal of Computer and Communications*.

Kaur, M., Bhatti, J., Kakkar, M. K., & Goyal, D. (2019). Timetable Handling Mechanism Using Python. *International Journal of Scientific & Technology Research*.

Kumar, A., Singh, K., & Sharma, N. (2013). Automated Timetable Generator Using Particle Swarm Optimization. *International Journal on Recent and Innovation Trends in Computing and Communication*.

Markal, S., Ghorpade, S., & Chalke, D. (2020). Timetable Generator. *IOSR Journal of Computer Engineering*.

Patil, P. P., Khichi, J., Jadhav, M., & Basa, V. (2021). Automatic Timetable Generator. *International Research Journal of Engineering and Technology*.

Puttaswamy, A., Khan, H. M., V, C. S., & A., P. (2018). A Study on Automatic TImetable Generator. *International Journal of Science and Innovative Engineering & Tehcnology*.

Sayed, P. S., Ahmed, A., Aamir, A., & Zaeem, A. (2015). Automated Timetable Generator. *International Journal for Innovative Research in Science & Technology*.

Sutoyo, M. N., & Mangkona, A. T. (2022). Implementing the Modified Euclidean Distance Method in the Course Planning of the USN Kolaka Information. *IT Journal Research and Development (ITJRD)*.

Techie-Menson, H., & Nyagorme, P. (2021). Design and Implementation of a Web-Based Timetable System for Higher Education Institutions. *International Journal of Educational Research and Information Science*.

# **APPENDIX**

**Project Timeline**

