MASENO UNIVERSITY

SCHOOL OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION TECHNOLOGY

EDUTIME TIMETABLING SYSTEM

CIT 402: IT PROJECT II

PROJECT SUBMITTED TO THE SCHOOL OF COMPUTING AND INFORMATICS IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

MASENO UNIVERSITY

P.O. BOX PRIVATE BAG

MASENO, KENYA

MAY, 2023

DECLARATION

I do hereby declare that this project is my original work and where there's work or contributions of other individuals, it has been duly acknowledged and relevant citations are given. To my knowledge, no material herein has been previously presented to any other academic institution for examination, degree award, or other awards.

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

KATUMO BENSON MAKAU **Date**

Admission No. CIT/00046/019

Supervisor,

I hereby certify that this project was presented for examination with my approval as the university appointed supervisor.

Supervisor's Name: _____

Signature: _____

Date: _____

DEDICATION

I dedicate this first and foremost to Almighty God, who has been there since the beginning of this project to this far. Special dedication to my supportive parents and friends who have shown total support and compassion towards my achievements. Again, I want to dedicate this proposal to my friends for continually showing support towards this project by sharing ideas on how to make it better.

ACKNOWLEDGMENT

This EduTime timetabling system project would not have been possible without the support and help of many people. I would like to thank the entire Information Technology department for supporting my degree program journey up to this level. Furthermore, I would want to recognize and thank my project proposal supervisors Mr. Chamwama and Madam Settim for all my meetings throughout the semester to keep me on track and provide clarity whenever things seemed to be difficult for me. Finally, I would also like to thank my course mates and friends for their great contributions to this project.

ABSTRACT

Maseno University is one of the best public universities in Kenya. It offers programs from certificate to PhD. level. The process of timetabling various course units in the lecture halls was complicated and time-consuming because it was done manually. The overall objective of this project was to develop a web-based automated timetabling management system for Maseno University. Commercially available timetabling systems are too expensive and also do not meet the requirements at hand for Maseno University. Specifically, the project was required to; identify the required modules of the automated timetabling system, design an automated timetabling system prototype, implement the designed prototype, and test the developed prototype. Information for the modules needed for the development of this system was gathered from stakeholders including lecturers, students, and university management. The design was done through Entity-Relationship Diagrams (ERD) for the database and Use Case Diagrams for identifying the interactions between the system and its actors and an Activity Diagram to model the dynamic aspects of the system i.e. flow from one activity to another. The system was developed using HTML, CSS, JavaScript, Bootstrap, MySQL, and PHP. The system was evaluated using the unit, integration, and system testing. This project shows the potential of using Information Technology (IT) to automate timetabling.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

1.1 Background Information

Due to the increase in course offerings and enrollment surges in higher learning institutions, the demand placed on facilities of academic institutions kept going up making the ability to work within the set constraints of time, facilities, and resources the greatest asset of any learning institution. Problems relating to timetabling vary between different institutions depending on the constraints. In most schools, lecturers manually designed timetables,a task that required them to set aside a week for that task. Again, manual design timetabling was subjected to human errors and could not satisfy all the requirements.

However, due to the inherent challenges, timetabling was still done manually. For example, for each semester, schools were forced to redo the timetables, thus making the task repetitive, tedious, and painful. In the case of Maseno University, departments were forced to communicate before making timetables to ensure lecture halls don't conflict with more than one class assigned to the same hall at the same time. However, this process was tedious and repetitive since the university offers many courses subdivided into several departments. With limited resources, the chances of having a conflict-free timetable were very low. Class representatives were forced to seek lecture halls if a class happened to conflict with another. This caused time wastage for both the students and lecturers. For a timetable to be complete, all the departments had to share information so classes could be arranged. This took up to about a week or two since Maseno University offers so many courses.

Failure to address timetable problems would lead to schedules with a maximum number of disputes that fail to meet a number of side restrictions, allocated time and rooms within the restricted period of time (Henry, 2021, p.g 1). Thus, it was within this context that the proposed automated timetabling system was needed to assess and fill the gap by designing and

implementing the proposed timetabling system. Therefore, within this context, the proposed timetabling system needs to assess and fill the gap by designing and implementing the proposed system to help manage the learning activities at Maseno University.

1.2 Problem Statement

The allocation and management of scarce resources such as lecture venues and laboratories in higher education institutions pose a complex challenge in scheduling classes, resulting in time-consuming manual timetabling processes, potential errors, and dissatisfaction among both students and lecturers.

1.3 Study Objectives

1.3.1 Overall Project Objective

To develop a web-based timetabling management system for Maseno University.

1.3.2 Specific Objectives

i). To identify the required modules of the automated timetabling system.

ii). To design an automated timetabling system prototype.

iii). To code the designed automated timetabling system prototype.

iv). To test the developed prototype.

1.4 Research Questions

i What modules are needed to implement this timetabling system?

ii What is the appropriate and suitable design for this system?

iii What implementation approach will be appropriate for this system?

iv What system testing and validation techniques will be suitable for this system?

1.5 Significance of the Project

The automation of timetabling activities at Maseno University ensured the smooth management of learning activities and saved time for both lecturers and students. It ensured that lectures didn’t collide and lecturers weren't assigned two classes simultaneously. It also ensured that lecture halls were utilized well.

1.6 Limitations

I. The design and methodology selected to implement this system was time-consuming.

II. The evaluation/testing of this system would be better, depending on the environment.

1.7 Assumptions

1. I assumed that the system to be developed would run effectively on the laptops of the users of this system.
2. I assumed that the entire process of developing this system would be cost-effective.
3. I also assumed that this system would be integrated with the existing systems.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter reviewed relevant and related literature. It gave background of the application areas, reviewed and critiqued similar systems, highlighting their functions, strengths, and weaknesses. It included a review of local and foreign-related literature that could help gather ideas that guided the development of the prototype. Several studies had been conducted prior to this project to explore the current state of the field. The proponents reviewed the existing research and gathered ideas from previous surveys to inform the development of a new concept for the proposed prototype. Through this research, the proponents aimed to build upon previous work and contribute new insights to the field.

2.2 Timetabling

Various definitions of the term "timetabling" exists. The Oxford Dictionary, defined timetabling as arranging something to take place at a particular time (Cross, 2005). Meanwhile, Burke (Burke, 2002, p.g 2) defined it as the allocation, subject to constraints, of given resources to objects being placed in space-time in a way that nearly or fully satisfied the set of desirable objectives. The Business Dictionary (Gibson, 2009) defined it as the formal organization of teachers' and learners' time and allocating and coordinating timings and other resources within an educational institution. From these definitions, it was clear that timetabling involved planning and allocating resources in a systematic and equitable manner, taking into account available resources and desired objectives. It was recognized that an automated system would guarantee better resource allocation than manual methods, making the process more efficient and effective.

2.3 Lessons Timetabling in Higher Learning Institutions

Timetabling in Institutions of Higher Learning was considered an optimization problem that took into account many variables and constraints. Optimizing medium and large instances proved to be a difficult task, especially when resources were limited and it was challenging to find a solution that satisfied the defined constraints and requirements. For instance, in Brazilian schools, a schedule for teachers had to meet two requirements: minimization of working days and avoidance of idle time slots (Dorneles, 2014, p.g 32). The problem of timetabling was prevalent in academic institutions such as colleges and universities and had attracted the attention of numerous researchers. However, addressing this issue was challenging due to the complex and soft constraints present and the size of the problem (Chen, 2021, p.g 106522).

2.4 Challenges in Timetabling

Several approaches were put forth to tackle the timetabling problem, including operational research, human-machine interaction, constraint programming, expert systems, and neural networks. Despite these efforts, challenges remained, such as the need for an approach that could easily be reformulated to support changes, a generalized framework to handle various types of timetabling problems, and the ability to incorporate knowledge into the timetabling system (Lee, 2005, p.g 1150). The University Timetabling problem was a type of schedule that was considered a complex problem in academic institutions. This problem involved coordinating lecturers, students, and classrooms to prevent conflicts between lectures.

2.5 Existing Timetabling Systems

2.5.1 Manual Timetabling System

Using the manual system, lecturers had to manually evaluate lessons to ensure they don't clash and resources such as lecture halls were allocated fairly. Different departments had to communicate to ease the process of resource sharing and also minimize the chances of having lessons clash. Despite being complex, the timetable helped reasonably manage almost sixty percent of the lectures. It was also very flexible since lecturers could just shift their lessons to different time slots if they seek the availability of a lecture hall. This method may have seemed simpler; however, it has so many weaknesses: it wasted a lot of time and was very complex. The possibilities of having errors were endless as it was not be easy to avoid lessons clashing. With limited resources, lessons clash, and resources are never fairly allocated (Nguyen-HQ, 1980). All these challenges were because the timetable was made of uninformed guesses, making it unreliable. Every day several lessons clashed on the use of lecture halls, and lecturers had two lessons assigned at the same time slots.

2.5.2 UniTime Scheduling System

UniTime system is a timetabling system built to address allocating lessons to lecturers and lecture halls. This system also addresses the issue of exam timetabling, where exams are planned and scheduled when to happen. The system proved to address almost every problem with timetabling. However, it had its weaknesses. The system did not address the fair allocation of resources (UniTime (2008). The system also did not fully address the issue of conflicts as it allowed for conflict to happen however it produced reports listing the conflicts. The system was also complex to use as its user interface was difficult to understand and access the services it was destined to provide. The system did not provide additional features such as class re-

scheduling with notifications. These weaknesses made the system inefficient as time would still be wasted, and resources would not be fairly shared or allocated.

2.5.3 Utwente Timetabling System

Utwente is a system that generates timetables for institutions. The system provides a calendar that the user relies on to plan and build a timetable. The System, similar to the current manual system at Maseno University, does not factor in the other variables such as equal measure on resource sharing such as halls and labs. The timetable was also built manually, where the user had to manually enter a class and assign it to a specific time slot. The system did not provide a mechanism to avoid or manage conflicts. The system did not provide an easy-to-use interface for class re-scheduling or requesting lab sessions (University of Twente, 2021). Therefore the system is not better placed to fill the gap at Maseno University. The system was not time conscious as the user manually fed in the data instead of automating the process. The system didn't provide a fair resource-sharing algorithm where students get a fair allocation of lecture halls and time slots. Therefore this system proved unreliable in solving the problem at hand as it was not automated and did not factor in fair resource allocation and time-saving.

2.5.4 Sagenda Timetabling System

Like the current manual system, the Sagenda system simply provides a calendar to which you assign a class or a lesson on a particular date and time. The system provides the user with an interface of a calendar where a lesson is set for a particular date. The system was also designed to encompass particular lessons, e.g., violin class. This could work similarly to a task manager or an even reminder; thus, it did not fit the problem. The system did not consider conflict management or even automation of the whole process. The system did not even factor in other resources such as lecture halls, lecturers, capacity, and the number of students (Sagenda,2019). Additionally, the system would likely waste time as the users had to manually enter the lessons to a particular day and time instead of having the process automated. The system also did not

provide a user interface that is appealing to the eyes of the user. It would waste a lot of resources training the users as it's not user-friendly and straightforward. For this reason, the system didn't quantify to be the best-suited system to solve the problem at hand as it also failed the test of fair resource allocation and time-saving.

Systems Reviewed and the common weaknesses across.

S.NO	System	Weaknesses
1	Manual Timetabling System	- Unfair allocation of resources. - Does not address conflicts
2	UniTime Scheduling System	- Unfair allocation of resources. - Does not address conflicts
3	Utwente Timetabling System	- Unfair allocation of resources. - Does not manage conflicts of lessons.
4	Sagenda Timetabling System	- Does not have a mechanism to manage conflicts. - Unfair resource allocation.

Table 1 System Weaknesses

2.6 Conclusion

All the reviewed systems above had common weaknesses across the four. They did not address the issues of fair resource allocation and conflicts management. An automated timetabling system was essential in ensuring that universities don't waste a lot of time planning and scheduling lectures, thus ensuring fast, efficient, and reliable class scheduling and fair resource allocation among students pursuing different courses. This system brings several benefits: time-saving:- since every activity is automated, managed-conflicts:- since no lessons come to clash at any particular point, and fair resource allocation:- since each course gets a fair allocation of the resources within the varsity.

CHAPTER THREE: METHODOLOGY

3.1 Introduction

Research refers to the systematic investigation into and study of materials and sources to establish facts and reach new conclusions. Research methodology incorporates the principles, practices, and procedures required to conduct research. This chapter describes the steps, procedures, techniques, and tools used to realize the research objectives. It is organized as follows Approach to Project Development, Requirements Identification, Design and Development, Testing, Representation of Results, and Ethical Requirements.

3.2 Approach to Project Development

For this project, Prototyping was the project development methodology used. The prototyping model is a systems development method in which a prototype is built, tested, and reworked as necessary until an acceptable outcome is achieved from which the complete system or product can be developed. This model is relied upon in scenarios where not all project requirements are known in detail ahead of time. It is an iterative, trial-and-error process between developers and users.

Prototyping methodology was chosen for this project because, with prototyping, customers could anticipate higher costs, needed changes, potential project hurdles, and, most importantly, potential end-result disasters. Prototyping requires user involvement and enables them to see and interact with a working model of their project. With prototypes, customers can give immediate feedback, request project changes, and alter model specifications. Prototyping, most importantly, helped eliminate misunderstandings and miscommunications during the development process.

3.3 Requirements Identification

Requirements were gathered before and after the system was developed. Gathering the requirements before prototype development enabled the developers to understand the user specifications that needed to be in the system for it to have an impact on the target audience. After the development of the prototype, the data collected enabled the developers to gauge the system's user experience and make changes where possible. The requirements gathering and analysis was accomplished using primary and secondary data.

For the development of this project several procedures were employed to perform requirements identification from different stakeholder viewpoints. The main approach used for the fact finding process included questionnaires and interviews. Both open-ended and closed-ended questionnaires were administered to identify new,existing problems, functional and non-functional requirements. Data Flow Diagrams were used to provide a visual representation of the flow of data through a system or processes and Entity Relationship Diagram was used to model the data stored in a database and show how entities were related to each other.

3.3.1 Functional Requirements

Functional requirements were determined since they were essential components of this project as they provided clear and specific definitions of what the system would do. Below were the main functional requirements of the EduTime timetabling system that were identified:

- User Management: The system should provide the ability to create, manage, and delete users with different roles and privileges, such as administrators, deans, department chairpersons and lecturers. The users of the system are: an Administrator,lecturers, and students. The data to be input will be: pf number, names, email, phone number and user password.
- School Management: The system should allow the creation and management of schools and their details, including school ID and a school name.

- Department Management: The system should allow the creation and management of departments and their details, including department ID, department Name and the school they belong to.
- Course Management: The system should allow the creation and management of courses and their details, including course ID, course name and department that offers the course.
- Units Management: The system should allow the creation and management of units and their details, including unit ID, unit name and the course that offers the defined units.
- Room Management: The system should provide the ability to manage classrooms and laboratories, including the ability to add, edit, and delete rooms/labs, as well as manage room capacities, features, and availability.
- Timetable Generation: The system should be able to generate schedules based on the courses, rooms, and available time slots, ensuring that no two classes take place at the same time in the same room, and no single lecturer is assign two different units at the same day and time slot.
- Conflict Resolution: The system should provide a mechanism for resolving scheduling conflicts that may arise, such as when two classes are scheduled at the same time in the same room.
- Class Scheduling: The system should allow lecturers to request a re-schedule classes and adjust their schedules as needed, taking into consideration the availability of rooms and other scheduling constraints.
- Reporting and Analytics: The system should provide a way to generate reports and analytics, such as timetable generation reports, class-reschedules and usage statistics, to help administrators make informed decisions.
- Notifications: The system should provide notifications to students, professors, and administrators, such as reminders about upcoming classes, changes to schedules, and other important updates.
- Mobile Compatibility: The system should be accessible on mobile devices, providing users with the ability to access their schedules and other information on the go. This

should be achieved with the help of Progressive Web Apps and general web system responsiveness to different screen widths.

3.3.2 Non Functional Requirements

Non-functional requirements define the quality attributes of a software system, such as performance, security, usability, and accessibility, and are critical for ensuring that the system operates as expected and meets the needs of the users. Below are the main non functional requirements of the EduTime Timetabling System:

- Performance: The system should be able to handle a large number of users, courses, and schedules, and should perform well even under heavy loads.
- Scalability: The system should be scalable, allowing the university to easily add new courses, rooms, and users as needed.
- Availability: The system should be available and accessible to users 24/7, with minimal downtime for maintenance and updates.
- Security: The system should be secure, protecting sensitive data such as staff records, schedules, and personal information from unauthorized access.
- Data Integrity: The system should maintain the integrity of data, ensuring that schedules and other information are accurate and up-to-date.
- Usability: The system should be user-friendly, providing an intuitive interface that is easy to use and navigate.
- Interoperability: The system should be able to integrate with other systems used by the university, such as student information systems and learning management systems.
- Reliability: The system should be reliable, ensuring that schedules are generated accurately and are available when needed.
- Maintenance: The system should be easy to maintain, with regular updates and patches to ensure that it remains secure and up-to-date.
- Compliance: The system should comply with relevant laws, regulations, and standards, such as data protection and privacy laws.

3.4 Design and Development

3.4.1 System Design

Systems design defines system elements like modules, architecture, components, and their interfaces and data for a system based on the specified requirements. This project will use Data Use Case diagrams, and sequence diagrams to model the system.

3.4.1.1 Use Case Diagram

A use case diagram is a graphical representation of the interactions between an actor (a person or a system) and a system or software application.

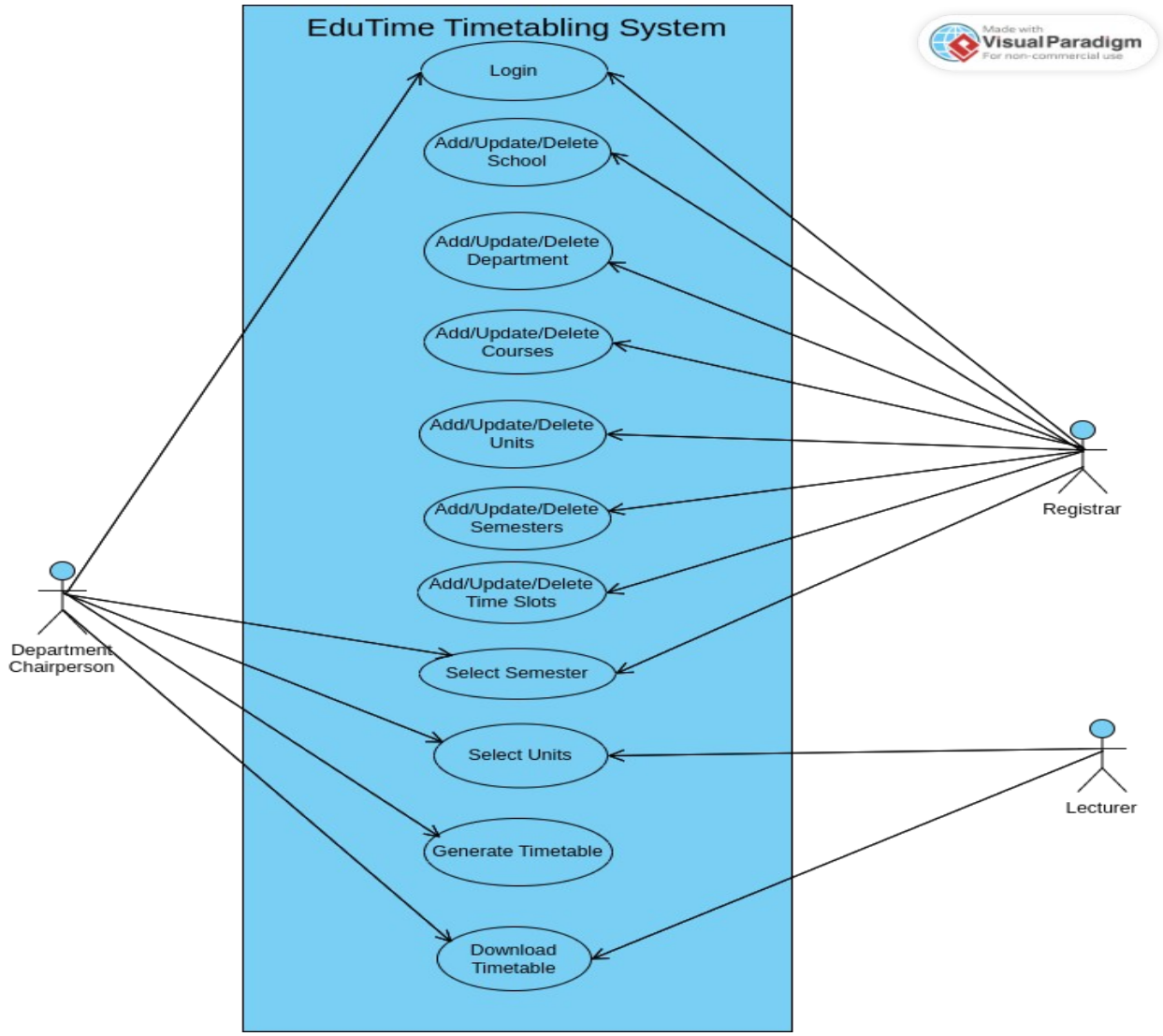

Figure 1: Use Case Diagram

3.4.1.2 Sequence Diagram

A sequence diagram is a type of interaction diagram that shows the interactions between objects or components in a system over time. It depicts the sequence of messages exchanged between the objects in a system, and the order in which these messages are exchanged.

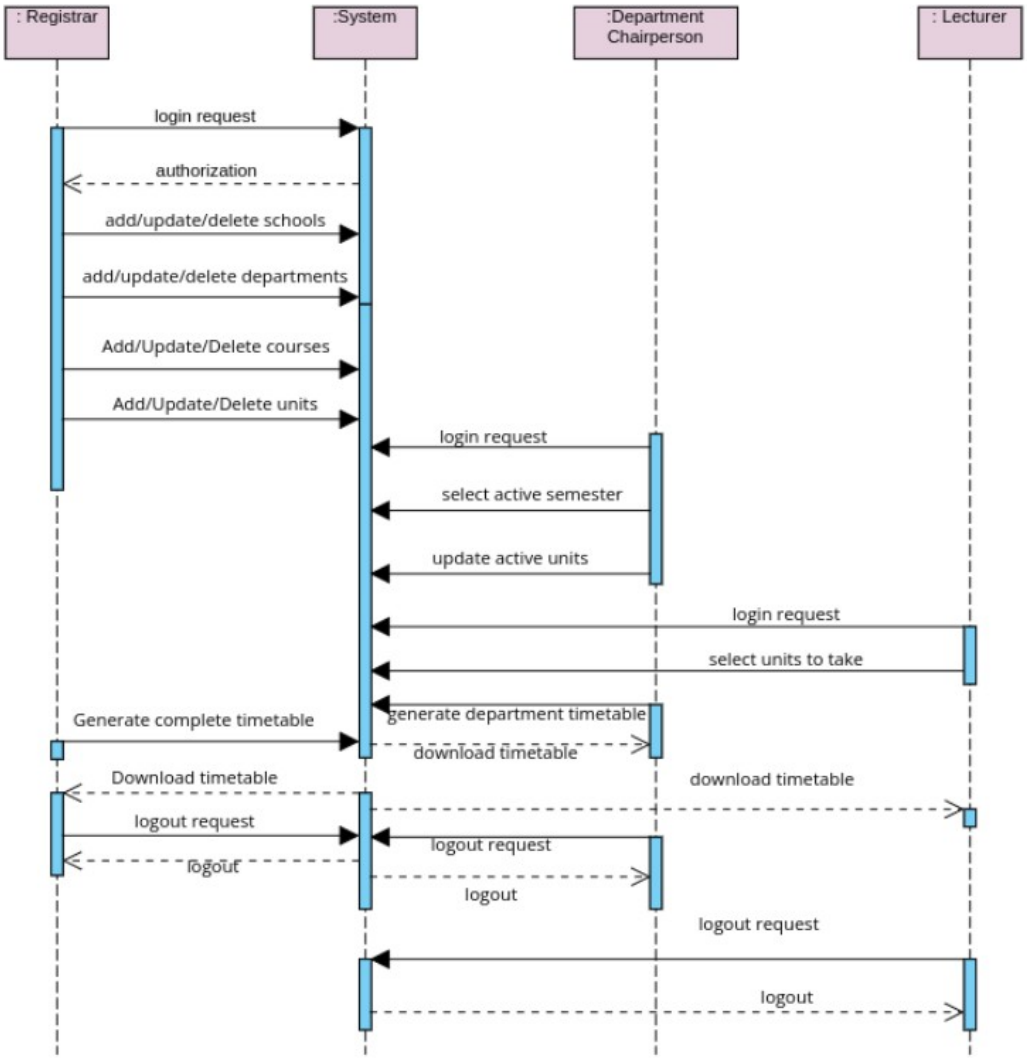

Figure 2: Sequence Diagram

3.4.1.3 Entity Relational Diagram(ERD)

Entity Relationship diagram is the primary design upon which a database is built. Entity Relationship diagrams specify what data will be stored, the entities and their attributes and how entities relate to other entities. This Entity Relationship Diagram below represents the structure of the relational database formation of this timetabling system.

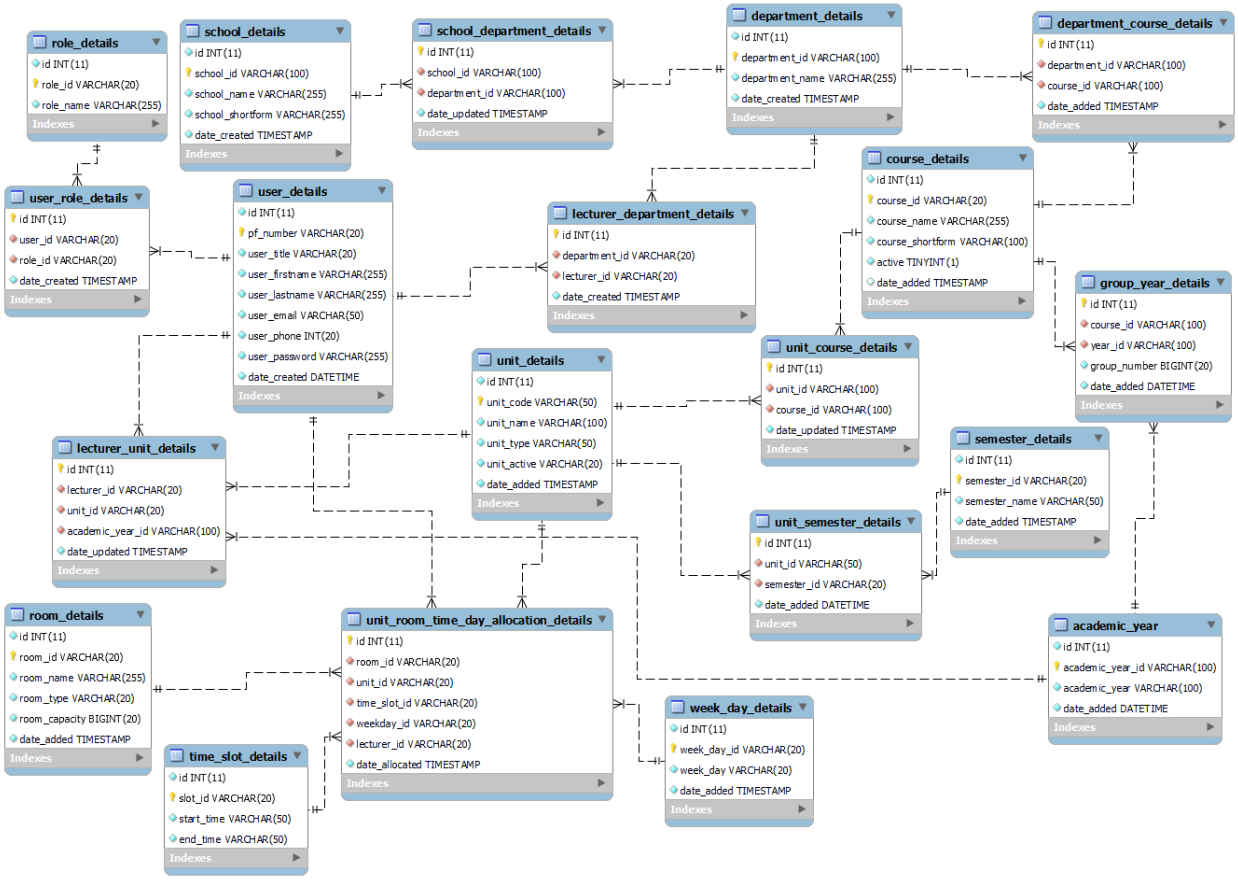

Figure 3: Entity Relationship Diagram (ERD)

3.4.2 Development

In this stage, the programmer began implementing the system by working on required modules and performing unit tests on each module. Tasks were broken down into pieces or modules and based on priorities for easy management and development. The developer adhered to all the best practices at this phase.

3.4.2.1 Hardware Platform

The table below shows the hardware architecture under which this system was developed.

Hardware	**Specifications**
Memory	8 GB
Processor	Intel® Core™ i5-5200U CPU @ 2.20GHz × 4
Graphics	Mesa Intel® HD Graphics 5500 (BDW GT2)
Storage	Hard Disk Drive (1TB)

Table 2 Hardware Specifications

The above hardware requirements were not the only specifications under which the proposed system could be built, but the specifications of the computer that was used. The project was later hosted on a server and thus, it was made accessible to all users via their laptops, tablets and mobile phones.

3.4.2.2 Software Environment

This project was developed under the defined software environment below:

Software	**Version**
Xammp	7.4.30
PHP	7.4.30
Browser	Google Chrome, Mozilla Firefox
Operating System	Ubuntu Linux
Version Control	Git
Hosting Server	Shared Hosting cPanel
Editor/IDE	Visual Studio Code
Database	MySQL

Table 3 Software Environment

The proposed system was developed under the above-defined software environment for effective performance. PHP 7.4.30 is more stable than the latest PHP version 8, and most hosting providers have set their servers to support PHP 7.4. Visual Studio Code editor is a very user-friendly and helpful text editor used by developers to build software. For this project, MySQL was the preferred DBMS. To track the development and code changes, the developer used git version control. After development, the system was deployed to production on a shared hosting so everyone on the internet could access it. For the operating system, the computer used to develop the prototype ran on Ubuntu Linux version 20.04.

3.5 Testing

Software testing is a technique for determining if the actual software product complies with expectations and is error-free. It entails using human or automated software to assess one or more properties of interest. Software testing goal is to find flaws, gaps, or unmet requirements compared to the requirements as written. Software Testing is Important because if there are any

bugs or errors in the software, they can be identified early and solved before the software product delivery. A properly tested software product ensures reliability, security, and high performance, resulting in time-saving, cost-effectiveness, and customer satisfaction.

3.5.1 Unit testing

A unit test are known to test the smallest piece of code that can be logically isolated in a system. In most programming languages, that is a function, a subroutine, a method, or a property. Testing units ensured that each function and method subroutine worked as expected and printed out the output in the formats required to ensure no bugs were within the software. To execute Unit Tests, the developer wrote a sections of code to test a specific functions in the software application. Developers also isolated functions to test more rigorously to reveal unnecessary dependencies between the function being tested and other units so the dependencies could be eliminated.

Significance

Unit testing primary goal was to separate written code for testing to see if it functioned as intended. Unit testing was a crucial stage in the development process because,it aided in finding bugs that could be more challenging to identify in subsequent testing phases.

3.5.2 Integration testing

This type of testing is where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules. This testing level aims to expose defects in the interaction between these software modules when they were integrated.

For this case, incremental testing was used where modules were being related modules were being integrated logically. Other related modules were integrated incrementally, and the process continued until all the logically related modules were integrated and tested successfully.

Significance

Integration testing ensured that the integrated units functioned correctly as one unit and aligned with stated requirements. It ensured no errors or bugs between the different interfaces of different modules.

3.5.3 System Testing

This is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one more extensive computer-based system element. Ultimately, the software is interfaced with other software/hardware systems. System Testing is defined as a series of tests solely aiming to exercise the complete computer-based system. System Testing falls under White box testing, which refers to testing a software application's internal workings or code. System test involves the external workings of the software from the user's perspective.

Significance

System Testing was essential as it was useful in testing fully integrated applications, including external peripherals, to check how components interacted with one another and the system. This type of testing was also important in testing user's experience with the application. This ensured that the deployed software worked as anticipated and met the specified requirements.

3.6 Representation of Results

The final output of the system was represented using screenshots of the software in different stages and screenshots of several modules, if not all, within the system. The system was hosted and assigned a unique domain name so that every person could access it via the internet and test functionality and user experience.

3.7 Ethical Requirements

Since this study used secondary data such as internet, papers and journals, in accordance with research respect and copyright, all the sources of information were acknowledged. The confidentiality and anonymity of the respondents was also maintained. All the test data provided by different respondents was handled discretely. All the participators of this project were supervised to ensure they maintained their professional and ethical behaviours. For the collection of data activity at Maseno University, the following were considered concerning ethics;

- The university policies, procedures, and standards will be adhered to.
- **Respect for persons:-** All participants involved in the research to develop this timetabling system must take part voluntarily, free from any coercion or influence, and their rights, dignity, and autonomy should be respected and appropriately protected.
- **Informed consent:-** This will ensure that the research staff and participants will be updated about all the possible risks and benefits to an individual from participating in a study.
- **Confidentiality and data protection:-**Individual research participant and group preferences regarding anonymity will be respected, and participant requirements concerning the confidential nature of information and personal data will also be respected when collecting data for the development of this system.
- **Integrity:-** Research for this system development will be designed, reviewed, and undertaken to ensure recognized standards of integrity are met, and quality and transparency will also be assured.
- **Conflict of interest:-** The independence of this research is clear, and any conflict of interest will be dealt with accordingly.

REFERENCES

Burke, E.K. and Petrovic, S., 2002. Recent research directions in automated timetabling. European journal of operational research, 140(2), pp.266-280.

Cross, F.L. and Livingstone, E.A. eds., 2005. The Oxford dictionary of the Christian church. Oxford University Press, USA.

Claessens, B.J.C., Eerde. WV Rutte CG & Roe, RA (2007). A Review of the Time Management Literature.

Chen, M.C., Goh, S.L., Sabar, N.R. and Kendall, G., 2021. A survey of university course timetabling problem: perspectives, trends, and opportunities. IEEE Access, 9, pp.106515-106529.

Dorneles, Á.P., de Araújo, O.C. and Buriol, L.S., 2014. A fix-and-optimize heuristic for the high school timetabling problem. Computers & Operations Research, 52, pp.29-38.

Gibson, K., 2009. BusinessDictionary. com. Reference Reviews.

Lee, J., Ma, S.P., Lai, L.F., Hsueh, N.L. and Fanjiang, Y.Y., 2005. University timetabling through conceptual modeling. International Journal of Intelligent Systems, 20(11), pp.1137-1160.

Nguyen-HQ and McDonald-J. , 1980. Class schedule to minimize student conflicts Proceedings of the American Institute for Decision Sciences Ninth Annual Meeting Western Regional Conference. American Inst. Decision Sci, Atlanta, GA, USA; 1980; xii+354 pp.

Sagenda (2019) *Sagenda University Timetabling Demo*, *Sagenda*. Iteration.info Sàrl. Available at: https://www.sagenda.com/demo/ (Accessed: December 11, 2022).

UniTime (2008) *UniTime Timetabling*, *UniTime 4.7*. The Apereo Foundation. Available at: https://demo.unitime.org/UniTime/login.do (Accessed: December 11, 2022).

University of Twente (2021) *Mytimetable - timetable view: University of Twente*, *MyTimetable - Timetable view | University of Twente*. The University of Twente. Available at: https://rooster.utwente.nl/schedule (Accessed: December 11, 2022).

Techie-Menson, H. and Nyagorme, P. 2021, Design and Implementation of a Web-Based Timetable System for Higher Education Institutions.

APPENDICES

GANTT CHART

WEEKS Deliverables	Week 1-2	Week 3-4	Week 5-8	Week 9-10	Week 11	Week 12-14	Week 15-17	Week 18-19	Week 20
Requirements Identification & Analysis									
Database Design									
Prototype Design									
Build authentication Module									
Build routes to add/update and delete schools and departments									
Build routes to add/update and delete courses,units and rooms/labs.									
Implement the timetable generation logic.									
Build reports for timetables generated.									
Ship Final Version of Project Code to GitHub.									

Figure 4 Gantt Chart