



**MASENO UNIVERSITY**

**SCHOOL OF COMPUTING AND INFORMATICS**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**AUTOMATED TIMETABLING SYSTEM**

**CIT 409: IT PROJECT I**

**PROJECT PROPOSAL 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  
JANUARY, 2023**

## DECLARATION

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

Supervisor's Name: **Dr. MUHAMBE T. MUKISA**

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

## **DEDICATION**

I dedicate this proposal 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/guardians and friends who have shown total support and compassion towards my achievements. Again, I want to dedicate this proposal to my friend Alexander Karanja for continually showing support towards this project by sharing ideas on how to make it better.

## **ACKNOWLEDGMENT**

This timetabling system proposal 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 supervisor Dr. Muhambe Mukisa 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 proposal.

## **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 is complicated and time-consuming because it is done manually. The overall objective of this project is 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 will be required to; identify the required modules of the automated timetabling system, design an automated timetabling system prototype, code the designed prototype, and test the developed prototype. Information for the modules needed for the development of this system will be gathered from stakeholders including lecturers, students, and university management. The design will be 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 will be developed using html, CSS, JavaScript, Bootstrap, MySQL, and PHP. The system will then be evaluated using the unit, integration, and system testing. This project proposal shows the potential of using Information Technology (IT) to automate timetabling.

# **TABLE OF CONTENTS**

<b>AUTOMATED TIMETABLING SYSTEM .....</b>	<b>1</b>
<b>DECLARATION.....</b>	<b>i</b>
<b>DEDICATION.....</b>	<b>ii</b>
<b>ACKNOWLEDGMENT .....</b>	<b>iii</b>
<b>ABSTRACT .....</b>	<b>iv</b>
<b>LIST OF TABLES .....</b>	<b>vii</b>
<b>LIST OF FIGURES .....</b>	<b>viii</b>
<b>CHAPTER ONE: INTRODUCTION .....</b>	<b>10</b>
1.1 Background Information .....	10
1.2 Problem Statement .....	11
1.3 Study Objectives .....	11
1.3.1 Overall Project Objective .....	11
1.3.2 Specific Objectives .....	11
1.4 Research Questions .....	11
1.5 Significance of the Project .....	12
1.6 Limitations .....	12
1.7 Assumptions .....	12
<b>CHAPTER TWO: LITERATURE REVIEW.....</b>	<b>13</b>
2.1 Introduction .....	13
2.2 Timetabling .....	13
2.3 Lessons Timetabling in Higher Learning Institutions .....	14
2.4 Challenges in Timetabling .....	14
2.5 Existing Timetabling Systems.....	14
2.5.1 Manual Timetabling System.....	14
2.5.2 UniTime Scheduling System .....	15
2.5.3 Utwente Timetabling System .....	15
2.5.4 Sagenda Timetabling System .....	16
2.6 Conclusion.....	17
<b>CHAPTER THREE: METHODOLOGY .....</b>	<b>18</b>
3.1 Introduction .....	18

3.2 Approach to Project Development .....	18
3.3 Requirements Identification .....	19
3.4 Design and Development .....	20
3.4.1 System Design .....	20
3.4.1.1 Use Case Diagram.....	21
3.4.1.2 Activity Diagram .....	21
3.4.1.3 Entity Relational Diagram(ERD).....	21
3.4.2 Development.....	22
3.4.2.1 Hardware Platform.....	22
3.4.2.2 Software Environment .....	23
3.5 Testing.....	23
3.5.1 Unit testing .....	24
3.5.2 Integration testing .....	24
3.5.3 System Testing .....	25
3.6 Representation of Results.....	25
3.7 Ethical Requirements .....	26
<b>REFERENCES.....</b>	<b>27</b>
<b>APPENDICES .....</b>	<b>28</b>
GANTT CHART .....	28

**LIST OF TABLES**

Table 1 System Weaknesses ..... 17

Table 2 Hardware Specifications ..... 22

Table 3 Software Environment ..... 23



# LIST OF FIGURES

Figure 1 Gantt Chart .....	28
----------------------------	----

# **CHAPTER ONE: INTRODUCTION**

## **1.1 Background Information**

Over the recent past, there has been an increase in course offerings and enrollment surges in institutions of higher learning. This has raised the demand for more facilities for academic institutions. This has proven that the ability to work within the set constraints of time, facilities, and resources is the greatest asset of any learning institution. Therefore, problems relating to timetabling vary between different institutions depending on the constraints. In most schools, lecturers manually design timetables, requiring them to set aside a week for that task. This manual way of timetabling is prone to human errors and is difficult to meet all the requirements.

Due to the inherent challenges, timetabling is still done manually. For example, for each semester, schools are forced to redo the timetables, thus making the task repetitive, tedious, and painful. In the case of Maseno University, departments are 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 is tedious and repetitive since the university offers many courses subdivided into several departments. With limited resources, the chances of having a conflict-free timetable are very high. Class representatives are forced to seek lecture halls if a class conflicts with another. This way, time is wasted for both the students and lecturers. For a timetable to be complete, all the departments must share information so classes can be arranged. This takes up to about a week or two since Maseno University offers so many courses.

However, this has presented the need for an automated timetabling system. Failure to address the timetabling problems would lead to schedules with a maximum number of disputes that fail to meet several side restrictions, allocated time, and rooms within the restricted period (Henry, 2021, p.g 1). 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**

Management of learning activities is a complex venture in institutions of higher learning. Lecture venues and laboratories are essential but scarce resources. Scheduling a class requires one to consider the nature of the class, the number of students, the time of the day, and whether or not the unit is shared across different programs. Manual designing of timetables thus is a complex and time-consuming affair, which contributes to the loss of valuable time, not to forget the complaints from both students and lecturers over errors in the timetables.

## **1.3 Study Objectives**

### **1.3.1 Overall Project Objective**

To develop a web-based automated 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 will ensure the smooth management of learning activities and save time for both lecturers and students. It will ensure that lectures don't collide and lecturers aren't assigned two classes simultaneously. It will also ensure that lecture halls are utilized well.

## **1.6 Limitations**

- I. The design and methodology selected to implement this system would be time-consuming.
- II. The evaluation/testing of this system would be better, depending on the environment.

## **1.7 Assumptions**

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

## **CHAPTER TWO: LITERATURE REVIEW**

### **2.1 Introduction**

This chapter reviews relevant and related literature. It gives a background of the application areas and reviews and critiques similar systems, highlighting their functions, strengths, and weaknesses. It includes a review of local and foreign-related literature that can help gather ideas that guide the development of the proposed prototype. It gives an overview of the different study that has been done before and the proponents to collect some ideas from another survey to formulate a new concept to apply when developing the proposed prototype.

### **2.2 Timetabling**

There are various definitions of the term timetabling. According to the oxford dictionary, Timetabling is arranging something to take place at a particular time (Cross, 2005). On the other hand, Burke (Burke, 2002, p.g 2) defines timetabling as the allocation, subject to constraints, of given resources to objects being placed in space-time in a way that nearly or fully satisfies the set of desirable objectives. The Business Dictionary (Gibson, 2009) defines timetabling as the formal organization of teachers' and learners' time and allocating and coordinating timings and other resources within an educational institution. From the above definitions, timetabling can be described as planning and allocating resources to objects relative to the available time to satisfy the desired objectives. This process should be carried out systemically with equity in resource allocation in mind. For this process to be efficient, there is a need to have a system that considers the available resources and the objects in need before allocating the resources. An automated system guarantees better resource allocation than having a human do the same.

## **2.3 Lessons Timetabling in Higher Learning Institutions**

Timetabling in Institutions of Higher Learning is an optimization problem that considers many variables and constraints. Optimizing medium and large instances is a very challenging task. When the resources are limited, it is often difficult to find a solution that satisfies the defined constraints and requirements. For example, in Brazilian schools, a schedule for teachers must meet two requirements: minimization of working days and avoidance of idle time slots (Dorneles, 2014, p.g 32). The timetabling problem is common in academic institutions such as colleges and universities. It is a problem that has attracted the interest of many researchers. This issue is challenging to address due to the existing complex and soft constraints and the size of the problem (Chen, 2021, p.g 106522).

## **2.4 Challenges in Timetabling**

several approaches were proposed in tackling the timetabling problem, such as operational research, human-machine interaction, constraint programming, expert systems, and neural networks. However, there are still several challenges to be addressed: easily reformulated to support changes, a generalized framework to handle various timetabling problems, and the ability to incorporate knowledge in the timetabling system (Lee, 2005, p.g 1150). The University Timetabling problem is a type of schedule known as a complex problem arising in academic institutions. The problem involves coordinating lecturers, students, and classrooms to avoid clashes between lectures.

## **2.5 Existing Timetabling Systems**

### **2.5.1 Manual Timetabling System**

Using the manual system, lecturers must manually evaluate lessons to ensure they don't clash and resources such as lecture halls are allocated fairly. Different departments have to communicate to

smoothen the process of resource sharing and also minimize the chances of having lessons clash. Despite being complex, the timetable helps reasonably manage almost sixty percent of the lectures. It is also very flexible since lecturers can just shift their lessons to different time slots if they seek the availability of a lecture hall. This method may seem simpler; however, it has so many weaknesses: it wastes a lot of time and is very complex. The possibilities of having errors are endless as it may not be easy to avoid lessons clashing. With limited resources, lessons clash, and resources are never fairly allocated (Nguyen-HQ, 1980). All these challenges are because the timetable is made of uninformed guesses, making it unreliable. Every day several lessons clash on the use of lecture halls, and lecturers have 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 has proven to address almost every problem with timetabling. However, it has its weaknesses. The system does not address the fair allocation of resources (UniTime (2008)). The system also does not fully address the issue of conflicts as it only allows for conflict to happen by producing a report. The system is also complex to use as its user interface is difficult to understand and access the services it is destined to provide. There is a need to have an easy-to-use system where all services are easily accessible. The system does not provide additional features such as class re-scheduling with notifications. These weaknesses make the system inefficient as time will still be wasted, and resources will 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 is also built manually, where the user has to manually enter a class and assign it to a specific time slot. The system does not provide a mechanism to avoid or manage conflicts. The system does 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 is not time conscious as the user manually feeds in the data instead of automating the process. The system doesn't provide a fair resource-sharing algorithm where students get a fair allocation of lecture halls and time slots. Therefore this system proves unreliable in solving the problem at hand as it's not automated and does 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 is also designed to encompass particular lessons, e.g., violin class. This could work similarly to a task manager or an even reminder; thus, it does not fit the problem. The system does not consider conflict management or even automation of the whole process. The system does not even factor in other resources such as lecture halls, lecturers, capacity, and the number of students (Sagenda,2019). Additionally, the system will likely waste time as the users must manually enter the lessons to a particular day and time instead of having the process automated. The system also does not provide a user interface that is appealing to the eyes of the user. It could waste a lot of resources training the users as it's not user-friendly and straightforward. For this reason, the system doesn't quantify to be the best-



suited system to solve the problem at hand as it also fails the test of fair resource allocation and time-saving.

### **Systems Reviewed and the common weaknesses across.**

<b>S.NO</b>	<b>System</b>	<b>Weaknesses</b>
1	Manual Timetabling System	<ul style="list-style-type: none"><li>- Time wasting</li><li>- Unfair allocation of resources.</li><li>- Does not address conflict of lessons.</li></ul>
2	UniTime Scheduling System	<ul style="list-style-type: none"><li>- Unfair allocation of resources.</li><li>- Does not address conflict of lessons.</li></ul>
3	Utwente Timetabling System	<ul style="list-style-type: none"><li>- Unfair allocation of resources.</li><li>- Does not manage conflicts of lessons.</li></ul>
4	Sagenda Timetabling System	<ul style="list-style-type: none"><li>- Does not have a mechanism to manage conflicts.</li><li>- Unfair resource allocation.</li></ul>

*Table 1 System Weaknesses*

## **2.6 Conclusion**

All the reviewed systems above have common weaknesses across the four. They do not address the issue of allocating resources fairly. Additionally, the systems have to address the issue of conflict management to satisfaction. An automated timetabling system is 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. Such systems are also very crucial in managing lab sessions. The system also continuously reminds its users of a scheduled class or lab session, for example, lecturers. 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 is the proposed project development methodology. 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 can anticipate higher costs, needed changes, potential project hurdles, and, most importantly, potential end-result disasters. Robust prototyping can ensure product quality and savings for years to come. 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, helps eliminate misunderstandings and miscommunications during the development process.

### **3.3 Requirements Identification**

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

#### **Primary Data Sources**

##### **Structured Online Questionnaires**

This will enable the researcher to get answers to specific questions, which will help in prototype design. The researcher will use this method before and after prototype development. This will help understand user requirements (before development) and gauge user experience (after development). This method was chosen because it is quick and easy to use, besides getting the specific results needed. Furthermore, the questionnaires will be online because internet access is widely available and guarantee a quick response.

##### **Open Interviews**

This will enable the researcher to get more information concerning the system since it is possible to rephrase questions to understand the system better. Since the interviews will not be closed, the respondent will be at liberty to give more details concerning the system, which will help them have a deeper understanding of user requirements and user experience.

#### **Secondary Data Sources**

Secondary data sources will be beneficial in collecting information on existing systems. The internet will be a powerful tool here since it will be used to explore tools, frameworks, and

architectures closely related to our prototype, as well as the challenges on existing systems, to prove that the use of technology in the prototype would overcome the existing challenges. Data analysis involves breaking the data into smaller pieces that can be easily understood and aid decision-making. In this case, tables will be used to analyze quantitative data.

## **Justification**

Questionnaires allow the collection of subjective and objective data in a large study population sample to obtain statistically significant results, especially when resources are limited. It is a good tool for the protection of the privacy of the participants. Questionnaires are popular research methods because they offer a fast, efficient, and inexpensive means of gathering information from sizable sample volumes. These tools are particularly effective for measuring subject behavior, preferences, intentions, attitudes, and opinions.

Interviews help one explain, better understand and explore research subjects' opinions, behavior, experiences, phenomenon, etc. Interview questions are usually open-ended questions so that in-depth information will be collected. Lastly, secondary sources help us understand the existing systems and the challenges that the systems have.

## **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 several tools to model the system. UML is a collection of diagrams and models that are used in representing the analysis, design, and implementation of systems in an object-oriented approach. The UML models provide an effective way to represent the design of the proposed system

prototype. The tools selected to model the design of the proposed system include; use case diagrams, activity diagrams, and Entity Relational Diagram.

#### *3.4.1.1 Use Case Diagram*

Use-case diagrams describe a system's high-level functions and scope. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how it operates internally. Use-case diagrams will illustrate and define the context and requirements of an entire system or any essential parts/modules. This way, the system flow will be simplified for easy understanding so that developers can capture all the concepts in the system.

#### *3.4.1.2 Activity Diagram*

An Activity diagram is another UML diagram used to describe dynamic aspects of the system. An activity diagram is an advanced version of a flow chart that models the flow from one activity to another. Developers use an activity diagram to understand the flow of programs on a high level. It also enables them to determine constraints and conditions that cause particular events.

#### *3.4.1.3 Entity Relational Diagram(ERD)*

ER diagram is essential for modeling the data stored in a database. It is the primary design upon which a database is built. ER diagrams specify what data we will store: the entities and their attributes. They also show how entities relate to other entities. This ER Diagram will be used to design the database and model the relationship between entities and their attributes.

### 3.4.2 Development

In this stage, programmers begin creating the complete system by writing code in the programming language of their choice. Tasks are broken down into pieces or modules and given to different developers throughout the coding phase. It is the stage of the software development life cycle that lasts the longest. The developer must adhere to specific code standards at this phase. To create and implement the code, they must also employ programming tools like compilers, interpreters, and debuggers.

#### 3.4.2.1 Hardware Platform

The table below shows the hardware architecture under which the proposed system will be 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 are not the only specifications under which the proposed system could be built, but the specifications of the computer that will be used. The prototype will be hosted after development and thus it will not rely on the specifications of the computer that intends to use the software.

### 3.4.2.2 Software Environment

The proposed system will be developed under the following environment.

Software	Version
Xampp	7.4.30
PHP	7.4.30
Browser	Google Chrome, Mozilla Firefox
Operating System	Linux- Parrot Os
Version Control	Git
Hosting Server	Shared Hosting cPanel
Editor	VsCode
Database	MySQL

*Table 3 Software Environment*

The proposed system will be 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. Vscode editor is a very user-friendly and helpful text editor used by developers to build software. For this project, I will use MySQL as my DBMS. To track the development and code changes, I will use git version control. After development, I will deploy the prototype in shared hosting so everyone on the internet can access the service. For the operating system, the computer used to develop the prototype runs on Linux-Parrot Operating System.

## 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's 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 tests 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 will ensure that each function and method subroutine works as expected and prints the output in the formats required to ensure no bugs are within the software. To execute Unit Tests, developers will write a section of code to test a specific function in the software application. Developers also isolate functions to test more rigorously to reveal unnecessary dependencies between the function being tested and other units so the dependencies can be eliminated. Developers also use the UnitTest framework to develop automated test cases for unit testing.

### **Significance**

Unit testing's primary goal is to separate written code for testing to see if it functions as intended. Unit testing is a crucial stage in the development process because, when done correctly, it may aid in finding early code issues 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 coded by different programmers. This testing level aims to expose defects in the interaction between these software modules when they are integrated.

For this case, Incremental Testing will be used where testing is done by integrating two or more modules that are logically related to each other and then tested for the proper functioning of the



application. Then the other related modules are integrated incrementally, and the process continues until all the logically related modules are integrated and tested successfully.

### **Significance**

Integration testing can ensure that the integrated units function correctly as one unit and align with stated requirements. It can also ensure no errors 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 entails testing fully integrated applications, including external peripherals, to check how components interact with one another and the system. Verify thorough testing of every input in the application to check for desired outputs. Testing of the user's experience with the application. This ensures that the deployed software works as anticipated and meets the specified requirements.

## **3.6 Representation of Results**

The final output of the proposed system will be represented using screenshots of the software in different stages and screenshots of several modules, if not all, within the system. The system will

also be hosted and assigned a unique domain name so that every person can access it via the internet and test functionality and user experience.

### 3.7 Ethical Requirements

Since my study has used secondary data such as papers and journals, in accordance with research respect and copyright, I will acknowledge my sources of information. I will also maintain the confidentiality and anonymity of the respondents. All the test data provided by different respondents will be handled discretely. All the participators of this project will be supervised to ensure they are professional and ethical. For the collection of data activity at Maseno University, the following will be 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	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
Deliverables									
Chapter 1									
Chapter 2									
Chapter 3									
Prototype Design									
Database Design and Implementation									
Authentication Module									
Timetabling Module									
Testing and Bug Fixing									
Ship Final Version of Project Code to Github (V1)									

Figure 1 Gantt Chart