

**PROJECT TITLE**: Master Timetable

**DEDAN KIMATHI UNIVERSITY OF TECHNOLOGY**

**SCHOOL OF COMPUTER SCIENCE AND IT**

**DEPARTMENT OF COMPUTER SCIENCE**

**NAME**: Caleb Sawe

**REGISTRATION NUMBER**: C026-01-0729/2020

**SUPERVISOR**: Mr. Naivasha, Patrick.

# Declaration

This proposal is my original work and has not presented for a degree in any other University.

Name: Caleb Sawe

Signature:

Date:

This proposal has been submitted for examination with my approval as University Supervisor

Name:

Signature:

Date:

# Abstract

This project aims to develop a timetabling system that optimizes classroom utilization and eliminates scheduling conflicts at Dedan Kimathi University of Technology (DKUT). The system will address the shortcomings of existing timetabling systems, such as a lack of a centralized platform, inefficient conflict detection and resolution algorithms, and a lack of a room availability tracker.

The new system will be designed to be user-friendly, efficient, and scalable. It will be deployed on a cloud-based platform to ensure scalability and accessibility. The system will be evaluated using a variety of metrics, including the number of scheduling conflicts, classroom utilization, and student satisfaction.

The project is expected to have a number of benefits for DKUT, including improved classroom utilization, reduced scheduling conflicts, increased efficiency of the timetable scheduling process, and improved student satisfaction. The project is also expected to make a contribution to the field of timetable scheduling by developing a new system that addresses the shortcomings of existing systems.

In conclusion, this project will develop a new timetabling system that addresses the shortcomings of existing systems and provides a number of benefits for DKUT.

# Table of Contents

[Declaration 2](#_Toc164245892)

[Abstract 3](#_Toc164245893)

[Table of Contents 4](#_Toc164245894)

[Chapter One: Introduction 7](#_Toc164245895)

[1.1 Background 7](#_Toc164245896)

[1.1.1 Background 7](#_Toc164245897)

[1.1.2 Local Scenario 7](#_Toc164245898)

[1.1.3 Target Group 7](#_Toc164245899)

[1.2 Statement of Problem 7](#_Toc164245900)

[1.2.1 Problem Statement 7](#_Toc164245901)

[1.3 Objectives 7](#_Toc164245902)

[1.3.1 General Objective 7](#_Toc164245903)

[1.3.2 Specific Objectives 8](#_Toc164245904)

[1.3 Research Questions 8](#_Toc164245905)

[1.4 Justification 8](#_Toc164245906)

[1.5 Scope 8](#_Toc164245907)

[1.6.1 Scope 8](#_Toc164245908)

[Chapter Two: Literature Review 9](#_Toc164245909)

[2.1 Introduction 9](#_Toc164245910)

[2.2 Case Studies 9](#_Toc164245911)

[2.2.1 Case Study 1: UniTime 9](#_Toc164245912)

[Shortcomings 9](#_Toc164245913)

[2.2.2 Case Study 2: TimeTabler 9](#_Toc164245914)

[Shortcomings 10](#_Toc164245915)

[2.2.3 Case Study 3: aSc TimeTables 10](#_Toc164245916)

[Shortcomings 10](#_Toc164245917)

[2.2.4 Case Study 4: EduTime 10](#_Toc164245918)

[Shortcomings 10](#_Toc164245919)

[2.3 Summary 11](#_Toc164245920)

[2.4 Research Gap 11](#_Toc164245921)

[Chapter Three: Methodology 12](#_Toc164245922)

[3.1 Introduction 12](#_Toc164245923)

[3.2 Software Development Methodology 12](#_Toc164245924)

[3.3 Fact Finding Techniques 13](#_Toc164245925)

[3.3.1 Target Population 13](#_Toc164245926)

[3.3.2 Sample Size 13](#_Toc164245927)

[3.3.3 Sampling Procedure 13](#_Toc164245928)

[3.3.4 Data collection 13](#_Toc164245929)

[3.3.4.1 Observations 13](#_Toc164245930)

[3.3.4.2 Interviews 13](#_Toc164245931)

[3.3 Software Development Procedure 14](#_Toc164245932)

[Chapter Four: Data Analysis 15](#_Toc164245933)

[4.1 Introduction 15](#_Toc164245934)

[4.2 Observations 15](#_Toc164245935)

[4.2.1 Process 15](#_Toc164245936)

[4.2.2 Issues faced during data collection 15](#_Toc164245937)

[4.2.3 Results 15](#_Toc164245938)

[4.3 Interviews 15](#_Toc164245939)

[4.3.1 Process 15](#_Toc164245940)

[4.3.2 The TimeTabler 15](#_Toc164245941)

[4.3.2.1 Aim 15](#_Toc164246043)

[4.3.2.2 Outcome 16](#_Toc164246044)

[4.3.3 The students 16](#_Toc164246045)

[4.3.3.1 Aim 16](#_Toc164246046)

[4.3.3.2 Outcome 16](#_Toc164246047)

[Chapter Five: System Design 17](#_Toc164246048)

[5.1 Introduction 17](#_Toc164246049)

[5.2 Functional Requirements 17](#_Toc164246050)

[5.3 Non Functional Requirements 17](#_Toc164246051)

[5.4 System Design 18](#_Toc164246052)

[5.4.1 Entity Relationship 18](#_Toc164246053)

[5.4.2 Module Flowchart Overview 18](#_Toc164246054)

[5.4.2.1 Initial Processes 18](#_Toc164246055)

[5.4.2.2 Preliminary and Initialization 18](#_Toc164246056)

[5.4.2.3 Ad-Hoc Processes 19](#_Toc164246057)

[5.4.2.4 Semester Processes 20](#_Toc164246058)

[Chapter Six: System Implementation and Testing 21](#_Toc164246059)

[6.1 Implementation 21](#_Toc164246060)

[6.1.1 User Module 21](#_Toc164246061)

[6.1.2 School Module 21](#_Toc164246062)

[6.1.3 Room Module 21](#_Toc164246063)

[6.1.4 Unit Module 21](#_Toc164246064)

[6.1.5 Schedule Module 21](#_Toc164246065)

[6.2 Testing 21](#_Toc164246066)

[6.2.1 User Module 21](#_Toc164246067)

[6.2.2 School Module 21](#_Toc164246068)

[6.2.3 Room Module 21](#_Toc164246069)

[6.2.4 Unit Module 21](#_Toc164246070)

[6.2.5 Schedule Module 22](#_Toc164246071)

[6.2.6 Test Cases 22](#_Toc164246072)

[6.3 Deployment 22](#_Toc164246073)

[1. Prerequisites: 22](#_Toc164246074)

[2. Clone the Codebase: 22](#_Toc164246075)

[3. Navigate to the App Directory: 22](#_Toc164246076)

[4. Build and Deploy: 22](#_Toc164246077)

[5. Create Admin: 22](#_Toc164246078)

[6. Access the Website: 22](#_Toc164246079)

[7. External Mapping: 22](#_Toc164246080)

[Chapter Seven: Discussion and Conclusion 23](#_Toc164246081)

[7.1 Discussion 23](#_Toc164246082)

[7.1.1 Objective 1 23](#_Toc164246083)

[7.1.2 Objective 2 23](#_Toc164246084)

[7.1.3 Objective 3 23](#_Toc164246085)

[7.2 Limitation 23](#_Toc164246086)

[7.2.1 Absence of Pre-existing Systems for Reference 23](#_Toc164246087)

[7.2.2 Complexity of Conflict Detection and Resolution 23](#_Toc164246088)

[7.3 Recommendation 23](#_Toc164246089)

[7.3.1 Implementation of Automated Lecture Creation 23](#_Toc164246090)

[7.3.2 User Training, Support, and Adoption 23](#_Toc164246091)

[7.4 Conclusion 24](#_Toc164246092)

[References 24](#_Toc164246093)

# Chapter One: Introduction

## 1.1 Background

In this section, I am to provide an understanding of the genesis and identification of the problem that will be highlighted in the problem statement section. We will consider various perspectives to the problem as well as the target group that will be included in our study, research as well as various stages of development of the project.

### 1.1.1 Background

The inefficient utilization of academic resources within an educational institution raises a major concern, particularly as universities in Kenya seek to fully optimize all available resources in provide quality education compellingly given the current economic crisis in the country. At Dedan Kimathi University of Technology, this concern is acutely felt due to increasing clashes of lectures and improper utilizations of classroom facilities. The current traditional timetable scheduling methods has led not only overlapping lectures but also inefficient allocation and utilization of academic spaces. This issue not only affects the students’ learning experience, but also sometimes hinders academic activities such as lectures and CATs. Moreover, it poses a challenge when planning, organizing and conducting short term academic activities such as regular CATs.

### 1.1.2 Local Scenario

With the Dedan Kimathi University of Technology context, the problem of classroom allocation and scheduling is particularly pronounced. The university experience a huge growth rate in student population and the inefficient use of classroom resources impacts both the students, school faculty and learning activities. These activities greatly rely on timely access of these suitable venues for certain periods of time.

### 1.1.3 Target Group

My study primarily targets the academic community at Dedan Kimathi University of Technology, including students, lecturers and administrative staff who are currently affected (both directly and indirectly) by the current scheduling challenges.

## 1.2 Statement of Problem

In this section of the proposal, I will clearly outline the problem statement, explaining why and how it is problematic with supporting evidence.

### 1.2.1 Problem Statement

The current timetable scheduling system ad Dedan Kimathi University is inefficient and leads to conflicts. This is because the system is not fully centralized and does not take into account various factors such as lecture schedules for different departments, availability of classrooms and the capacity of classrooms as well. This results in time overlaps and underutilization of classroom facilities. This scheduling issue is a major concern for the academic community as it disrupts the learning process as a major haul. Statistical evidence derived from background research further emphasizes the growing gravity of this problem showcasing the urgent need for a comprehensive solution.

## 1.3 Objectives

In this section of the proposal, I am to outline my research objectives that should solve the above stated problem.

### 1.3.1 General Objective

The main objective of this project is to design and develop a timetabling system that optimizes and simplifies the timetable creation process, eases room allocation and tracking and eliminate scheduling conflicts at Dedan Kimathi University of Technology.

### 1.3.2 Specific Objectives

1. To develop a centralized timetabling platform that is accessible by all departmental timetable scheduler.
2. To integrate a robust conflict detection and resolution functionality that tackles lecture conflicts.
3. To incorporate a room availability tracker within the timetable scheduling platform.

## 1.3 Research Questions

In this section, I shall outline key inquiries that will guide my investigation into the stated problem and it is imperative that I limit the questions within the project scope.

1. What are the current classroom scheduling practices at Dedan Kimathi University of technology and what challenges do they present?
2. How the scheduling conflicts directly interfere with students’ learning experience and how best should it be solved?
3. What extra features and functionalities should be included in the timetabling system for Dedan Kimathi University of Technology?
4. How effective will the developed system be at eliminating the stated problems?
5. What recommendations can be made for sustainable management of academic resources and scheduling at said university?

## 1.4 Justification

In this section, I shall illustrate the rationale behind conducting the research, developing the system and the potential beneficiaries/benefits.

This research is conducted to address a critical issue affecting the academic community of Dedan Kimathi University of Technology. The findings and results of the timetable scheduling system aim to benefit both students and faculty members by improving overall quality of education. Students will benefit from a more organized and efficient timetable schedule while faculty members will have better tools for planning and delivering their respective course work effectively.

## 1.5 Scope

In this section, I am to outline the focus of the study, including the geographical area within which the project shall affect or be affected by and the target groups/population.

### 1.6.1 Scope

This study will primarily focus on the Dedan Kimathi University of Technology main campus, including all its departments, schools and any other academic groups. The target population comprises of students, graduate assistants, lecturers and any other administrative staff directly involved in the classroom scheduling processes.

# Chapter Two: Literature Review

## 2.1 Introduction

Timetable scheduling is a complex problem that involves a variety of factors, such as the availability of classrooms and instructors (lecturers, etc.), the needs of students and the requirements of academic programs. It is important to create a schedule that is efficient, conflict free and meets the needs of all stakeholders.

There are a number of already existing systems that I studied before proposing this project. They have a number of advantages and disadvantages or areas they fall short of. This chapter will review the literature on a couple of timetable scheduling systems and identify some key challenges and research gaps in this area.

## 2.2 Case Studies

### 2.2.1 Case Study 1: UniTime

UniTime (Tomas Muler, 2019) is a comprehensive educational timetabling system that can be used to create and manage a school’s schedule for classes. It is a distributed system that allows multiple university and departmental schedule managers to coordinate efforts to build and modify a schedule. UniTime is free and open-source software

UniTime has a number of key features that make it a powerful tool for timetable scheduling. These features include:

Centralized scheduling: UniTime provides a centralized platform for managing all aspects of timetable scheduling, including classes, exams, and other events. This makes it easier for schedule managers to coordinate their efforts and ensure that the schedule is efficient and conflict-free.

Reporting: UniTime provides a variety of reports that can be used to analyze the timetable and identify areas for improvement. This information can be used to make informed decisions about future scheduling.

#### Shortcomings

UniTime lacks a robust constraint management module to allows schedule managers to define a variety of constraints on the timetable, such as class times, room availability, and student preferences.

Overall, UniTime is a powerful and versatile timetabling system that can be used to improve the efficiency and effectiveness of timetable scheduling at universities of all sizes. However, it is important to be aware of its limitations before implementing the system.

### 2.2.2 Case Study 2: TimeTabler

TimeTabler (Kieth and Chris Johnson, 2023) is a cloud-based timetable scheduling system that is designed for schools and universities. It is a comprehensive system that allows users to create timetables for classes, exams, and other events. TimeTabler also offers a variety of features that can help to improve the efficiency and effectiveness of the timetable scheduling process.

Some of the key features of TimeTabler include:

* Centralized scheduling: TimeTabler provides a centralized platform for managing all aspects of timetable scheduling. This makes it easy for schedule managers to coordinate their efforts and ensure that the schedule is efficient and conflict-free.
* Constraint management: TimeTabler allows schedule managers to define a variety of constraints on the timetable, such as class times, room availability, and student preferences. This helps to ensure that the schedule meets the needs of all stakeholders.

#### Shortcomings

Despite its many strengths, TimeTabler also has some shortcomings. One shortcoming is that it can be expensive, especially for larger schools and universities. Another shortcoming is that it can be difficult to use for schedule managers who are not familiar with the system.

Overall, TimeTabler is a powerful and versatile timetable scheduling system that can be used to improve the efficiency and effectiveness of timetable scheduling at schools and universities of all sizes. However, it is important to be aware of its limitations before implementing the system.

### 2.2.3 Case Study 3: aSc TimeTables

aSc TimeTables (Applied Software Consultants, n.d.) is another cloud-based timetable scheduling system that is designed for schools and universities. It is a comprehensive system that allows users to create timetables for classes, exams, and other events. aSc TimeTables also offers a variety of features that can help to improve the efficiency and effectiveness of the timetable scheduling process.

Some of the key features of aSc TimeTables include:

* Centralized scheduling: aSc TimeTables provides a centralized platform for managing all aspects of timetable scheduling. This makes it easy for schedule managers to coordinate their efforts and ensure that the schedule is efficient and conflict-free.
* Optimization: aSc TimeTables uses a variety of optimization algorithms to create a schedule that is as efficient as possible. This can help to reduce the number of scheduling conflicts and improve the overall quality of the schedule.
* Reporting: aSc TimeTables provides a variety of reports that can be used to analyse the timetable and identify areas for improvement. This information can be used to make informed decisions about future scheduling.

#### Shortcomings

Despite its many strengths, aSc TimeTables also has some shortcomings. One shortcoming is that it can be difficult to learn and use for schedule managers who are not familiar with the system. Another shortcoming is that it does not offer as many features as some of the other timetable scheduling systems that are available.

Overall, aSc TimeTables is a good choice for schools and universities that are looking for a comprehensive and affordable timetable scheduling system. However, it is important to be aware of its limitations before implementing the system.

### 2.2.4 Case Study 4: EduTime

EduTime (Edu-Orbit, n.d.) is a cloud-based timetable scheduling system that is designed for schools and universities. It is a comprehensive system that allows users to create timetables for classes, exams, and other events. EduTime also offers a variety of features that can help to improve the efficiency and effectiveness of the timetable scheduling process. It has similar features to the previous three systems.

#### Shortcomings

Despite its many strengths, EduTime also has some shortcomings. One shortcoming is that it does not offer as many features as some of the other timetable scheduling systems that are available.

Overall, EduTime is a good choice for schools and universities that are looking for a comprehensive and affordable timetable scheduling system. However, it is important to be aware of its limitations before implementing the system.

## 2.3 Summary

All four of these systems are comprehensive systems that can be used to create timetables for classes, exams, and other events. They also all offer a variety of features that can help to improve the efficiency and effectiveness of the timetable scheduling process, such as centralized scheduling, constraint management, optimization, and reporting.

UniTime is a free and open-source timetable scheduling system that is used by a number of universities around the world. It is a powerful and versatile system, but it can be complex to use.

TimeTabler is a cloud-based timetable scheduling system that is designed specifically for schools and universities. It is a comprehensive system that is easy to use, but it can be expensive.

aSc TimeTables is another cloud-based timetable scheduling system that is designed for schools and universities. It is a good choice for schools and universities that are looking for a more affordable system, even though it is not as feature-rich as some of the other systems that are available.

Overall, all three of these timetable scheduling systems are good options for schools and universities of all sizes. The best system for you will depend on your specific needs and budget.

## 2.4 Research Gap

The literature review has identified a number of research gaps in the area of timetable scheduling. Some of the key research gaps include:

* The development of new methods for modelling and solving timetable scheduling problems.
* The development of new tools and techniques for supporting the timetable scheduling process.
* The evaluation of the effectiveness of different timetable scheduling systems and methods.

# Chapter Three: Methodology

## 3.1 Introduction

In this chapter, I shall review the method of system adaptation I shall employ to develop the Master Timetable. Moreover, I shall identify my target population, sampling them out and obtaining useful data that I may need while developing the system and during testing.

## 3.2 Software Development Methodology

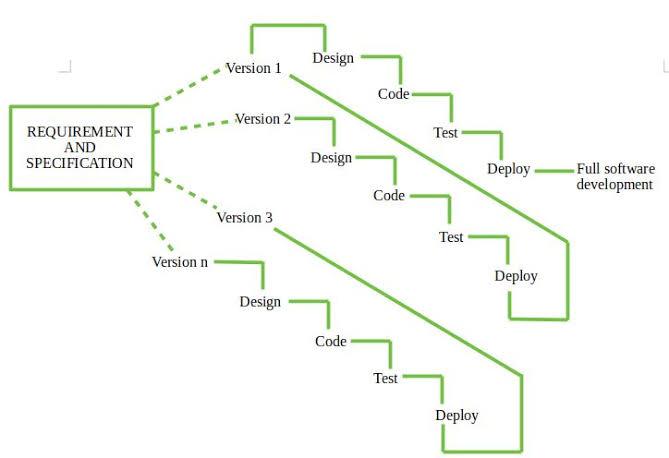
In the development and deployment of this system, I shall use Iterative method of software development.   
In this manner, the first demo will be developed and internally tested. Then deployed a demo to target users for testing purposes. Thereafter, I shall collect feedback from the users on various areas and use the collected feedback to better the application and rollout a new improved version that will follow the same steps. This will be redone until the target users are satisfied with the developed solution.

This methodology has a couple advantages to my use case.

1. I shall need direct interaction with the target users as the solution should be user friendly.
2. It is less costly in terms of time and resources to employ this method as compared to the other methods.
3. In this method, it shall be extremely easy to incorporate changes during the development process.

I shall however have some limitations while using the methodology. For instance:

1. I shall have to operate testing and development with prejudice and ensure they shan’t overlap as this shall cause confusion.

The diagram below illustrates the methodology I shall follow:  


## 3.3 Fact Finding Techniques

### 3.3.1 Target Population

The main target population I have focused in this study is the academic community at Dedan Kimathi University of Technology. This includes all the students, the lecturers and administrative staff that are actively involved in the day-to-day learning activities at the university.

### 3.3.2 Sample Size

Considering the large and growing population of the university, it shall be difficult to collect data from the entire target population. For this reason, I shall calculate an accurate sample size that will whose responses will be useful and effective in my study. I shall use the Taro Yamane (Projectclue12, n.d.) method of determining an adequate sample size.

The formula is:

Whereby;

Note, the method of data collection I shall employ shall determine the margin of error. In this case, I have used a margin of 0.1 against the entire population of Dedan Kimathi University of Technology. This results in a sample size of 99.13… In this case, I shall use 100 members of the university.

### 3.3.3 Sampling Procedure

After finding an adequate number of members to conduct the study on, I shall need to find a method to collect the data in a manner that it won’t be biased. Due to this, I shall use cluster sampling.

This is a form of probabilistic sampling where each population member is assigned to a specific cluster so that their data won’t be biased to one target group. Considering the nature of my study and target population, I shall study a total of one student and lecturer per department for a number of lectures.

### 3.3.4 Data collection

The data I collected during my study was a hybrid of two methods. This ensured the data I collected was without bias and collected in a natural setting while learning activities were taking place.

#### 3.3.4.1 Observations

This involves the researcher going physically to lectures and collecting data on how lectures are conducted and taking notes on them.

#### 3.3.4.2 Interviews

This involves the researcher physically having an interview with different members of the chosen sample size. This involves interviewing students, lecturers and various members of the school faculty.

Here are some of the interview guides I shall follow for different clusters:

1. Students

* Do you have an issue with the current timetable system?
* Have you ever had a different class conflict with yours in terms of venue and time?
* Is the timetable well developed and maintained?
* How can we better the system?
* Are lectures efficiently distributed?

1. Department Timetable Scheduler

* What are the current procedures you follow when creating and drafting timetables?
* What issues do you face when developing the timetable?
* How do you currently handle conflicts?

## 3.3 Software Development Procedure

In this section, I shall outline the comprehensive software development procedures for the design-tailor collaboration app, focusing on key steps such as requirements gathering, system design, implementation, testing, deployment, maintenance and tech support.

1. Requirements Gathering and Analysis

In this stage, the activities involved are collecting and gathering requirement information from the target population. This shall help me as a researcher to better understand the problem, the population’s needs and preferences.

1. System Design

In this stage, I shall start designing the system. This includes the workflow, flowcharts and various other diagrammatic and explanatory representations of the system. I shall begin by designing the overall systems architecture that outlines the system’s major components. Then I go in detail to various components. This shall be handled in later chapters of this document.

1. System Implementation

At this stage, I have designed the system and I have a clear picture of the expected outcome. I shall begin writing the code for the system. Implementing it ensuring that it strictly follows the requirements specified in the first section and it has an outcome as closely depicted in the system design.

1. Testing

In this stage, the application shall undergo thorough testing by a team of different testers under different levels. First the system shall be internally tested by the developer. Once it passes, it shall then be availed to the users at different levels to test their specific modules.

1. Deployment

In this stage, the system has been developed and it has passed testing. It shall then be rolled out and availed to the entire institution so that it can take effect. It shall also allow for verbose post production testing. If there are any issues with the rollouts, they should be addressed in a timely manner and ensuring that it doesn’t conflict the application’s configuration.

1. Maintenance / Tech Support

* At this stage, the application has been in effect. The developer provides tech support due to varying reasons such as:  
  Scaling issues or migration.
* Minor bugs that were overlooked during development and testing but doesn’t affect app’s main functionalities.
* Updates to various modules.

# Chapter Four: Data Analysis

## 4.1 Introduction

In this chapter, the results of the data collection shall be highlighted and discussed. An overview of the data collection shall also be discussed in detail.

## 4.2 Observations

This data collection was meant to investigate a number of issues and processes.

### 4.2.1 Process

This involved physically going to the area of study (Dedan Kimathi University of Technology) and observing these issues:

1. If there are any conflicts happening (if a lecture clashes with a different lecture at a specific time and venue).

2. How often do conflicts occur.

3. How the conflicts are resolved.

4. How classrooms are utilized (whether large classrooms are occupied by a small numbered class or vice versa).

After noting what issues need to be investigated, a frequency of observations needed to be established. After careful deduction, this frequency of observations was established:

- twice a day (morning or mid-morning hours, which were interchangeable) and afternoon hours

- At least three times a week, notably well distributed (e.g Monday, Wednesday and Friday)

### 4.2.2 Issues faced during data collection

An unforeseen issue was some classes failed to take place owing to some weather conditions or personal reasons by lecturers

Also worth noting that some lectures didn’t have full attendance, so it was difficult to gauge if a classroom was being fully utilized.

### 4.2.3 Results

Conflicts occur at random and sometimes frequently when it isn’t resolved well initially.

Resolving the issue was quite cumbersome as it was difficult to locate a free room. It mostly involved the class representative physically going room by room checking. The issue with this was it was difficult to gauge if the room is actually free or it was free due to a lecture being cancelled.

## 4.3 Interviews

### 4.3.1 Process

This involved physically interviewing some stakeholders that might benefit from the system. In this process, there were two categories of interviewees, a timetabler who is also a lecturer and students

### 4.3.2 The TimeTabler

#### 4.3.2.1 Aim

The aim of interviewing the timetabler was to gain insights into the following:

1. How a timetable is created (the process and prerequisites)

2. How rooms are allocated.

3. What are some underlying issues when creating a timetable

#### 4.3.2.2 Outcome

After conducting an interview with the departmental timetabler for the Department of Computer Science, these are the responses:  
- The process is quite simple to understand from the overview but the intrinsic details is what brings a challenge. It is as simple as picking a unit being taught during that semester and adding it on an excel sheet with an allocated room. If a department has a service unit being assigned to it, it needs the department offering the service unit to publish and release their timetable. All lectures occuring need to be at a venue or classroom that is already assigned to the department. It was also worth noting that the preference of a lecturer is taken into account when assigning him/her a lecture. Rooms are allocated per department by the overall university timetabler and what is taken mostly into account is the department.

### 4.3.3 The students

#### 4.3.3.1 Aim

The aim of interviewing the students was to gain insights into the following:

1. If they’ve ever been inconvenienced by a conflicting lecture.

2. How do they resolve the issue.

3. If they’ve ever felt congested in a classroom

4. Any issues they have with the delivery of the timetable

#### 4.3.3.2 Outcome

Out of the 10 students interviewed, they all gave consistent results with the issues they’re facing.

- They sometimes face issues when a lecture gets inconvenienced with a conflicting lecture. They often have to resolve to look for an available room. Most of the time, they find a room that is too small to host the entire number of the students available in that class. Sometimes, they resort to holding the class online. The students all gave a response that they don’t like how the timetable is being availed to them through an excel file only through an email. Also they were dissatisfied with how changes were made with the timetable after it rolled out.

# Chapter Five: System Design

## 5.1 Introduction

In this chapter, the functional and nonfunctional requirements shall be discussed and highlighted. Some sections of the system design shall be highlighted

## 5.2 Functional Requirements

1. Timetable Management:

Create and manage timetables for different academic departments.

Provide a user-friendly interface for easy navigation and manipulation of timetables.

Support addition, modification, and deletion of courses, instructors, and classrooms.

2. Optimization:

Implement an automated optimization algorithm to maximize classroom utilization and minimize scheduling conflicts.

Generate a master timetable reflecting the optimized schedule for all departments.

3. Centralized Access:

Develop a centralized timetabling platform accessible by all departmental timetable schedulers.

Implement user authentication and authorization mechanisms.

Support multiple user roles (e.g., administrators, departmental schedulers) with different access permissions.

Enable real-time synchronization of changes made by one scheduler to all other authorized users.

4. Conflict Detection and Resolution:

Automatically detect and highlight scheduling conflicts, including overlapping lectures or resource double-booking.

Provide options for conflict resolution, including manual adjustments and automated suggestions.

Consider factors like instructor availability, student preferences, and room constraints during conflict resolution.

5. Room Availability Tracking:

Maintain a real-time database of room availability, considering factors like maintenance schedules and special events.

Allow users to view the availability of specific rooms during timetable creation.

Send automatic notifications in case of clashes between room availability and scheduled activities.

## 5.3 Non Functional Requirements

These are requirements that focus on the behaviour of part of the system instead of functional parts

1. Usability and user friendliness:

The system should be simple to use and forms should be easier to fill.

2. Reliability and Availability

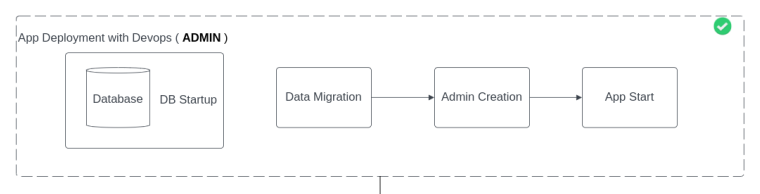
The system should be reliable and available to all users at any time.

## 5.4 System Design

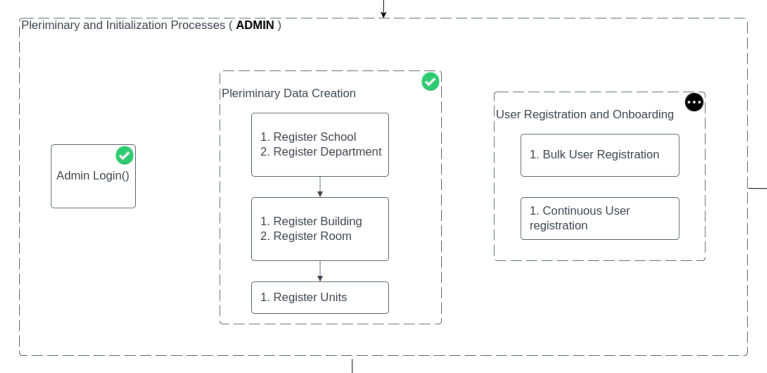
### 5.4.1 Entity Relationship

### 5.4.2 Module Flowchart Overview

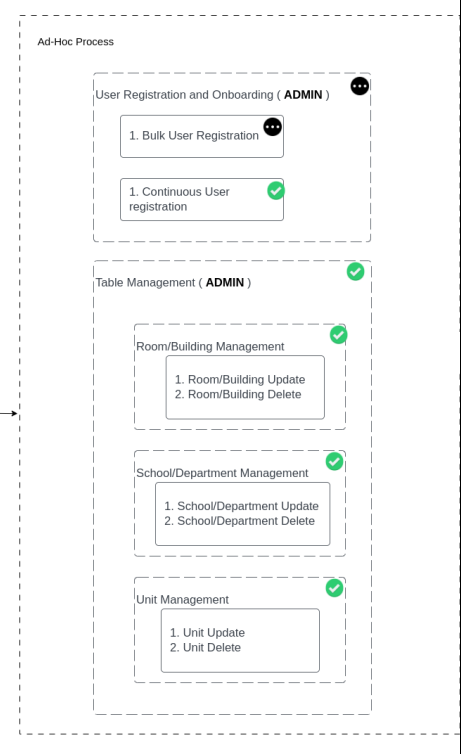
#### 5.4.2.1 Initial Processes



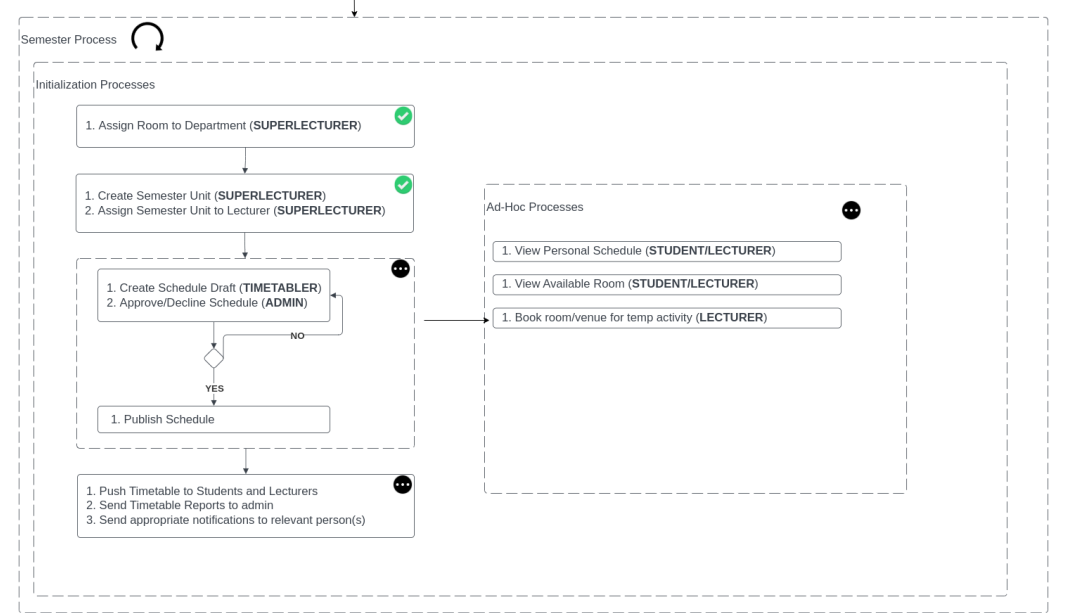
#### 5.4.2.2 Preliminary and Initialization



#### 5.4.2.3 Ad-Hoc Processes



#### 5.4.2.4 Semester Processes



# Chapter Six: System Implementation and Testing

## 6.1 Implementation

Implementation was conducted from the ground up separating modules from their functionality.

### 6.1.1 User Module

The User Module facilitates user management functionalities such as registration, login, authentication, and session handling. It supports both bulk and individual user registrations. Implemented on top of Django's built-in authentication system, it ensures secure user access to the application.

### 6.1.2 School Module

The School Module manages departments and schools within the system. It relies on the User Module for user roles and permissions. Departments and schools can be created, updated, and deleted through this module, providing a hierarchical structure to the application.

### 6.1.3 Room Module

The Room Module handles the management of rooms and their assignments. It is dependent on the School Module to ensure accurate assignment of rooms to departments and schools. This module ensures efficient utilization of available spaces within the institution.

### 6.1.4 Unit Module

The Unit Module is responsible for managing academic units and semester units. It relies on the School Module for departmental information and the User Module for user-related functionalities. Semester units are allocated to departments for teaching in specific semesters.

### 6.1.5 Schedule Module

The Schedule Module integrates lectures into the system's schedule. As the top-level module, it depends on all other modules for data integration. This module generates schedules based on unit allocations and room availability, providing a comprehensive view of academic activities.

## 6.2 Testing

### 6.2.1 User Module

- Unit tests validate user registration, login, authentication, and session management functionalities.

- Integration tests ensure seamless interaction with other modules, preventing conflicts or errors.

### 6.2.2 School Module

- Unit tests verify the creation, updating, and deletion of departments and schools.

- Integration tests validate interactions with the User Module for role and permission management.

### 6.2.3 Room Module

- Unit tests ensure accurate room management and assignment functionalities.

- Integration tests confirm compatibility with the School Module for departmental associations.

### 6.2.4 Unit Module

- Unit tests validate the creation and allocation of academic units and semester units.

- Integration tests ensure proper integration with the School and User Modules for data consistency.

### 6.2.5 Schedule Module

- Unit tests verify the scheduling logic for lectures and academic activities.

- Integration tests test interactions with all other modules to ensure seamless integration.

### 6.2.6 Test Cases

## 6.3 Deployment

The deployment process involves dockerizing the application and deploying it on a cloud instance. Follow these steps:

### 1. Prerequisites:

- Ensure Git and Docker are installed on your system.

- Have access to a cloud instance (e.g., AWS EC2).

### 2. Clone the Codebase:

- Clone the project repository from Git to your local machine.

### 3. Navigate to the App Directory:

- Open a terminal and navigate to the root directory of the project.

### 4. Build and Deploy:

- Execute the following command to build and deploy the application:

“ docker-compose up --build“

### 5. Create Admin:

- Once the deployment is complete, retrieve the container ID for the app using `docker ps`.

- Run the following command to create an admin:

“docker exec -it <container id> python manage.py createsuperuser”

### 6. Access the Website:

- Access the website through the provided IP address (localhost if deployed locally, or the IP of the cloud server if deployed in the cloud).

### 7. External Mapping:

- Optionally, configure basic IP and CNAME mapping externally for accessibility.

# Chapter Seven: Discussion and Conclusion

## 7.1 Discussion

The primary aim of this project was to develop a centralized timetabling platform that streamlines the creation of timetables, enhances efficiency in assigning classrooms, reduces scheduling conflicts, and simplifies the process of locating available rooms at Dedan Kimathi University of Technology.

### 7.1.1 Objective 1

Development of a Centralized Timetabling Platform:

Achieved by implementing a web-based application using Django framework, providing a centralized platform accessible to all departmental timetable schedulers.

### 7.1.2 Objective 2

Integration of Conflict Detection and Resolution Functionality:

Accomplished by incorporating robust conflict detection algorithms into the system, enabling timely identification and resolution of scheduling conflicts.

### 7.1.3 Objective 3

Incorporation of Room Availability Tracker:

Implemented a room availability tracker within the timetable scheduling platform, enabling users to easily identify and allocate free rooms for scheduled activities.

## 7.2 Limitation

### 7.2.1 Absence of Pre-existing Systems for Reference

The lack of existing systems with similar features posed challenges in leveraging prior solutions or frameworks, necessitating the development of novel approaches from scratch.

### 7.2.2 Complexity of Conflict Detection and Resolution

The intricate nature of scheduling conflicts within a university environment introduced complexities in designing and implementing effective conflict detection and resolution mechanisms. This complexity required extensive analysis and iterative refinement to ensure accurate conflict resolution.

## 7.3 Recommendation

To address the identified limitations and further enhance the effectiveness of the timetabling system, the following recommendations are proposed:

### 7.3.1 Implementation of Automated Lecture Creation

Develop an automated system for generating lectures based on available data within the system. Utilizing the wealth of data captured by the timetabling platform, automated lecture creation can significantly reduce manual intervention and streamline the scheduling process.

### 7.3.2 User Training, Support, and Adoption

Prioritize user training programs and provide comprehensive support to facilitate the adoption of the new system. Investing in user education and ongoing assistance will ensure that stakeholders fully understand and effectively utilize the timetabling platform, maximizing its benefits and fostering widespread adoption.

## 7.4 Conclusion

In conclusion, this project aimed to develop a centralized timetabling system to optimize classroom utilization, reduce scheduling conflicts, and simplify room allocation at Dedan Kimathi University of Technology. By leveraging modern web technologies and robust algorithms, the project successfully achieved its objectives, providing a user-friendly platform for efficient timetable creation and resource management. Despite some limitations, the system holds great potential to significantly enhance the university's scheduling processes and contribute to a more streamlined academic environment.

# References

Applied Software Consultants. (n.d.). From https://www.asctimetables.com/

Edu-Orbit. (n.d.). From https://eduorbit.co.in/school-management-software.html

Kieth and Chris Johnson, O. R. (2023, February 8th). From https://www.timetabler.com/

Projectclue12. (n.d.). *Taro Yamane Formular In Calculating Sample Size For Research*. From Medium: https://projectclue1.medium.com/taro-yamane-formular-in-calculating-sample-size-for-research-92b93a39696c

Tomas Muler. (2019). *UniTime*. From https://www.unitime.org/