**Automated Exam Scheduling System Using Heuristic Algorithm: Case of Strathmore University**

**Kathembo Tsongo Dieudonne**

**112721**

**Submitted in partial fulfillment of the requirements of the Bachelor of Business in Information Technology at Strathmore University**

**School of Computing and Engineering Science**

**Strathmore University**

**Nairobi, Kenya**

**2024**

# Declaration and Approval

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

Student Name: Kathembo Tsongo Dieudonne

Admission Number: 112721

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor Name: Dr. Henri Muchiri

Supervisor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

The existing manual approach to exam scheduling at Strathmore University, primarily reliant on Microsoft Excel, faces significant challenges, leading to disruptions in the scheduling process and compromising administrative efficiency. The limitations include exposure to human errors, suboptimal resource utilization, and an inability to respond to unexpected changes quickly. Recognizing these issues, this solution introduceed an Automated Exam Scheduling System that aimed to reduce human errors, by ensuring a balanced distribution of exams, and incorporating intelligent algorithms for optimized resource allocation. The system promoted fairness, transparency, and collaboration by providing real-time, personalized timetables to students through a user-friendly interface, addressing the shortcomings of the manual scheduling process. This System used Object-Oriented Analysis and Design (OOAD) as a crucial methodology during its design phase. This methodology, OOAD has been used for its effectiveness in modeling complex systems, allowing for a systematic and modular representation of various aspects of the exam scheduling process. Through it, the system's architecture was structured around objects, each encapsulating data and functionalities relevant to specific entities such as courses, rooms, and student preferences.

# List of Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Description |
| BBIT | Bachelor in Business Information Technology |
| CAD | Computer-aided design |
| CASE | Computer-Aided Software Engineering |
| CSS | Cascading Style Sheets |
| DB | Database |
| Dr | Doctor |
| ERD | Entity-Relationship Diagrams |
| GA | Genetic Algorithm |
| GitHub | Git and Hub |
| HTML | Hyper Text Markup Language |
| JS | JavaScript |
| KU | Kenyatta University |
| MariaDB | **Maria-Database** |
| Mr | Mister |
| MySQL | My Structured Query Language |
| NAU | Nnamdi Azikiwe University |
| NPD | New Product Development |
| PHP | Hypertext Preprocessor |
| OOAD | Object-Oriented Analysis and Design |
| PDO | PHP Data Objects |
| RDBMS | Relational Database Management System |
| SQL | Structured Query Language |
| SQLite | **Structured Query Language** **lite** |
| UML | Unified Modeling Language |
| VCS | Version Control System |
| VSC | Visual Studio Code |

# Table of Contents

[Declaration and Approval ii](#_Toc153647732)

[Abstract iii](#_Toc153647733)

[Table of Contents iv](#_Toc153647734)

[List of Figures vii](#_Toc153647735)

[List of Tables viii](#_Toc153647736)

[List of Abbreviations ix](#_Toc153647737)

[Chapter 1: Introduction 1](#_Toc153647738)

[1.1 Background 1](#_Toc153647739)

[1.2 Problem Statement 2](#_Toc153647740)

[1.3 Aim 2](#_Toc153647741)

[1.4 Objectives 3](#_Toc153647742)

[1.4.1 General objective 3](#_Toc153647743)

[1.4.2 Specific Objectives 3](#_Toc153647744)

[1.5 Research Questions 3](#_Toc153647749)

[1.6 Justification 4](#_Toc153647750)

[1.7 Scope and Limitations 4](#_Toc153647751)

[1.7.1 Scope 4](#_Toc153647752)

[1.7.2 Limitations 5](#_Toc153647753)

[Chapter 2: Literature Review 6](#_Toc153647754)

[2.1 Introduction 6](#_Toc153647755)

[2.2 Complexity of Exam Scheduling Timetable in Higher Educational Institutions 6](#_Toc153647757)

[2.3 Challenges faced by Strathmore University Examination Office 8](#_Toc153647758)

[2.3.1 Introduction 8](#_Toc153647759)

[2.3.2 Managing a Large Student Population 8](#_Toc153647760)

[2.3.3 Catering to Special Exam Requirements 8](#_Toc153647761)

[2.3.4 Addressing the Challenge of Cumbersome Timetables 9](#_Toc153647762)

[2.3.5 Handling Exam Timetable Adjustments 9](#_Toc153647763)

[2.3.6 Communication and Complaint Management 10](#_Toc153647764)

[2.3.7 Resource Allocation and Venue Constraints 10](#_Toc153647765)

[2.3.8 Conflict Resolution and Fairness 10](#_Toc153647766)

[2.3.9 Conclusion about challenges faced by the Strathmore University Exam Office 12](#_Toc153647768)

[2.4 Related Systems 13](#_Toc153647769)

[2.4.1 Introduction 13](#_Toc153647770)

[2.4.2 Automated University Timetable System 13](#_Toc153647771)

[2.4.2.1 Contextual understanding of the system 13](#_Toc153647772)

[2.4.2.2 Tools and technologies 13](#_Toc153647773)

[2.4.2.3 Database structure 14](#_Toc153647774)

[2.4.2.4 Modules and Outputs 17](#_Toc153647783)

[2.4.2.5 Limitations of the Automated University Timetable System 19](#_Toc153647787)

[2.4.3 Automated Examination Timetable Generation and Invigilator Scheduling System 19](#_Toc153647788)

[2.4.3.1 Contextual understanding of the system 19](#_Toc153647789)

[2.4.3.2 Tools and Technics 20](#_Toc153647790)

[2.4.3.3 Functional Components 20](#_Toc153647791)

[2.4.3.4 System added values 22](#_Toc153647796)

[2.4.3.5 System limitations 22](#_Toc153647797)

[2.4.4 Re-Engineering of Examination Timetable Generation and Invigilation Scheduling System 24](#_Toc153647798)

[2.4.4.1 Contextual understanding of the system 24](#_Toc153647799)

[2.4.4.2 Tools and techniques 24](#_Toc153647800)

[2.4.4.3 Interaction with the system 25](#_Toc153647802)

[2.4.4.4 System added values 28](#_Toc153647806)

[2.4.4.5 System limitations 28](#_Toc153647807)

[2.5 Heuristic Algorithm for scheduling systems 29](#_Toc153647808)

[2.5.1 Overview about Heuristic Algorithms 29](#_Toc153647809)

[2.5.2 Benefits of Heuristics in Problem Solving 30](#_Toc153647810)

[2.5.3 General Heuristic Model in Problem Solving 31](#_Toc153647811)

[2.5.3.1 Representation Heuristics 31](#_Toc153647812)

[2.5.3.2 Search Heuristics 31](#_Toc153647813)

[2.5.3.3 Termination Heuristics 32](#_Toc153647814)

[2.5.3.4 Implementation Heuristics 32](#_Toc153647815)

[2.6 Conceptual Framework 33](#_Toc153647816)

[2.6.1 General Understanding of the Conceptual Framework 33](#_Toc153647817)

[2.6.2 Explanation of the Conceptual Framework 34](#_Toc153647818)

[2.7 Conclusion 34](#_Toc153647819)

[Chapter 3: Methodology 36](#_Toc153647820)

[3.1 Introduction 36](#_Toc153647821)

[3.2 Software Development Methodology: Prototype approach 36](#_Toc153647822)

[3.2.1 Understanding of the prototype development approach 36](#_Toc153647823)

[3.2.2 Stages of the Prototype Development Process 37](#_Toc153647825)

[3.2.3 Requirements Gathering and Analysis 38](#_Toc153647826)

[3.2.4 Quick Design 38](#_Toc153647827)

[3.2.5 Prototype Construction 38](#_Toc153647828)

[3.2.6 Initial User Evaluation 38](#_Toc153647829)

[3.2.7 Refining the Prototype 38](#_Toc153647830)

[3.2.8 Implementation of the Product and Maintenance 39](#_Toc153647831)

[3.3 Justification of Prototype approach 39](#_Toc153647832)

[3.4 System Analysis: OOAD Analysis and Designing Approach 39](#_Toc153647833)

[3.4.1 Use Case Diagram 40](#_Toc153647834)

[3.4.2 Sequence Diagram 41](#_Toc153647835)

[3.5 System Design 41](#_Toc153647836)

[3.5.1 Class Diagram 41](#_Toc153647837)

[3.5.2 E.R. Diagram 42](#_Toc153647838)

[3.5.3 Database Diagram 42](#_Toc153647839)

[3.6 System Development Tools and Techniques 43](#_Toc153647840)

[3.7 Conclusion 47](#_Toc153647842)

[3.8 Deliverables 47](#_Toc153647843)

[Reference 49](#_Toc153647845)

[Appendix 52](#_Toc153647846)

# List of Figures

[Figure 2.1: Email sharing draft 1 for exam timetable 2023……………………...............………11](file:///F:\IS%20II%20Project\codes%20backup\exam_schedule_8\Proposal_IS_II_ref.docx#_Toc151192584)

Figure 2.2: Email from the exam office to the administrators…………………………………...11 [Figure 2.3: One of the complaints from students…….…………………………………………..12](file:///F:\IS%20II%20Project\codes%20backup\exam_schedule_8\Proposal_IS_II_ref.docx#_Toc151192585)

[Figure 2.4: Response from the exam office to the student complaint 12](#_Toc151499876)

[Figure 2.5. Login. 17](#_Toc151499893)

[Figure 2.6: Main page. 18](#_Toc151499894)

[Figure 2.7: Exam timetable generation. 18](#_Toc151499895)

[Figure 2.8: Functional components 20](#_Toc151499901)

[Figure 2.9: Interaction with the system 21](#_Toc151499902)

[Figure 2.10. Dashboard showing various modules 21](#_Toc151499903)

[Figure 2.11: Module where the timetable generation is being processed 22](#_Toc151499904)

[Figure 2.12: Use Case for the Re-Engineering of Examination Timetable Generation and Invigilation Scheduling System 25](#_Toc151499910)

[Figure 2.13: Home Page 26](#_Toc151499912)

[Figure 2.14: Faculty dashboard 26](#_Toc151499913)

[Figure 2.15: Generated Timetable by the Faculty Administrator 27](#_Toc151499914)

Figure 2:16: General Heuristic Model in Problem Solving……………………………………...31

[Figure 2.17: Prototype representation 37](#_Toc151499925)

[Figure A.1: Gantt Diagram 53](#_Toc151499950)7

# List of Tables

[Table 2.1: Administratorcheck 14](#_Toc151499884)

[Table 2.2: Controlsub 14](#_Toc151499885)

[Table 2.3: Examdate 15](#_Toc151499886)

[Table 2.4: Staff 15](#_Toc151499887)

[Table 2.5: StudentStrength 15](#_Toc151499888)

[Table 2.6: The University 16](#_Toc151499889)

[Table 2.7: The University Department 16](#_Toc151499890)

[Table 2.8: Venue 16](#_Toc151499891)

[Tables 3.1: Tools and techniques………………………………………………………….......43](file:///F:\IS%20II%20Project\codes%20backup\exam_schedule_8\Proposal_IS_II_ref.docx#_Toc149751677)

[Table 3.2: Deliverables………………………………………………………………………..47](file:///F:\IS%20II%20Project\codes%20backup\exam_schedule_8\Proposal_IS_II_ref.docx#_Toc149751678)

[Table A.1: Activity Schedule 52](#_Toc151499948)

Table A.2: Supervision sheet………………………………………………………………….48

# 

# Chapter 1: Introduction

## **Background**

According to the Research Services and Strategy Office at Strathmore University, the institution currently hosts a population of 7300 students, including both full-time and part-time students, offering a diverse range of courses. The university website lists 18 undergraduate courses, including Bachelor of Tourism Management, Bachelor of Science in Hospitality and Hotel Management, and Bachelor of Business Science in Financial Engineering, Financial Economics, and Actuarial Science. Focusing more on the undergraduate full-time program, each semester, students undergo a two-week exam period, excluding weekends, spanning 10 days from Monday to Friday. The university's policy dictates four-time slots per exam day, each lasting 2 hours.

According to the university's regulations, students are typically required to take a maximum of 7 exams. The undergraduate full-time programs organize students into groups such as BBIT 1.1 A, BBIT 1.2 C, BCOM 2.2 B, etc. Each student is expected to belong to one group; however, challenges arise when some students retake units, leading them to enroll in both their regular group and additional groups. This situation sometimes results in students violating university policies, such as having two exams on the same day or consecutively. Such are cases among others that make the University Exam Office face the complex task of crafting exam timetables that accommodate all courses and students. This becomes especially challenging when, out of the 18 courses, approximately 12 with nearly 5000 students schedule their exams simultaneously. Due to venue and time slot constraints, students from different courses are often mixed. From there, results another complexity in the exam timetable creation process, considering various factors like student groups, exam venues, and time slots. However, despite careful planning, complaints from students are common. These complaints range from missing exams in the timetable to collisions in exam time slots and insufficient time between exams for adequate preparation. Approximately 5000 students sitting for exams can lead to an overwhelming number of emails raising concerns.

Upon receiving complaints, the exam office undertakes a thorough reassessment and modification of the timetable to address the raised issues, striving to create a second draft that is as reasonable as possible. An additional challenge arises as changes made to address individual concerns may create wave effects, impacting all students and groups within a given slot. This dynamic can create a loop of problem creation and solution, further complicating the task of achieving a universally satisfactory exam timetable.

## **Problem Statement**

Resource allocation poses a common challenge in real-world management (Zhu et al., 2020), and Strathmore University faces the manual creation and management of exam timetables using conventional tools, notably Microsoft Excel. This longstanding practice is proving increasingly unsustainable in the face of mounting complexities in academic scheduling. Relying on this labor-intensive, error-prone process exposes critical weaknesses. The current Excel-centric solution is marred by limitations. The manual nature of this approach makes it prone to human errors, data input inaccuracies, and scheduling conflicts, risking disruptions in the exam scheduling process and compromising both the academic experience for students and the administrative efficiency of the university.

Efficient allocation of exam slots and venues, especially considering retake students, individual course requirements, and varying student preferences, poses challenges for the Excel-based system. This constraint results in suboptimal resource utilization, potentially leading to student dissatisfaction. Moreover, the system lacks flexibility in responding swiftly to unexpected changes or special requests, making the adjustment of exam timetables—a crucial need—tedious and time-consuming. This inflexibility strains administrative resources and hinders the university's capacity to effectively address evolving requirements. It's noteworthy that schedule simulation, proven effective in learning tools, could enhance the student experience in the exam scheduling timetable by accommodating their needs and concerns (Hyatt, 2011).

## **Aim**

This research work aimed to revisit how Strathmore University has been managing its exam timetables through the development of a web-based Automated Exam Scheduling System. Motivated by the insight that unevenly scheduled exams can lead to conflicts and disruptions, similar to the observations made by a student analyzing KU's final exam schedules (Sienko et al., 2016), this innovative solution harnessed the capabilities of web-based technology and computational optimization techniques to comprehensively automate the scheduling process. The introduction of automation in this project aimed to significantly reduce human errors and ensure a more equitable distribution of exams across the exam week and time slots, minimizing disruptions and conflicts for a smoother examination experience for both students and faculty.

Moreover, the proposed solution integrated intelligent algorithms that consider various parameters, including course requirements, room availability, and faculty preferences. This high level of customization has optimized resource allocation, fostering fairness in scheduling and addressing the current limitations of the manual system. Importantly, the system has provided real-time transparency to students, granting them access to personalized timetables through a user-friendly interface.

## **Objectives**

## **General objective**

To design, develop, and implement an advanced and efficient system that automates the complex process of the Automated Exam Scheduling System.

## **Specific Objectives**

## To identify the Complexity of Exam Scheduling Timetables in higher educational institutions,

## To review and evaluate existing solutions and practices in exam timetable management,

## To develop an Automated Exam Scheduling System that addresses the identified complexities,

## To test and validate the proposed Automated Exam Scheduling System.

## **Research Questions**

1. How are the key complexities painpointing Strathmore University in crafting exam timetables?
2. How are the strengths and weaknesses of current solutions informing the development of an advanced Automated Exam Scheduling System?
3. How can automation and computational optimization techniques be effectively utilized to enhance the efficiency and accuracy of exam scheduling timetables?
4. What impact has had the implementation of the Automated Exam Scheduling System on administrative efficiency, student satisfaction, and resource utilization at Strathmore University?

## **Justification**

First and foremost, the manual creation and management of exam timetables within academic institutions like Strathmore University have been labor-intensive. Along the same line, Vinod and Kadam (2016) stress that creating of best timetable manually requires a large amount of effort because it is subjected to various constraints. By automating the Exam Scheduling System, the university significantly has reduced the occurrence of human errors, ensuring greater accuracy and reliability in exam scheduling timetables. This, not only enhanced the academic experience for students but also reduced administrative burdens. With the implementation of the system, Strathmore University has been able to allocate resources more efficiently, optimize venue utilization, and respond swiftly to unforeseen changes.

Secondly, traditional manual scheduling methods often struggled to accommodate retake students, individual course requirements, and varying student preferences. This system led to suboptimal allocation of exam slots and venues, resulting in student dissatisfaction and resource wastage. By incorporating intelligent algorithms that consider various parameters, such as course requirements, room availability, and student preferences, the proposed system aimed to optimize the allocation of resources. This level of customization promoted fairness in scheduling and ensured that the university could provide a personalized, student-centric approach to exam scheduling, which ultimately enhanced the overall educational experience.

## **Scope and Limitations**

## **Scope**

This project covered the development of an Automated Exam Scheduling System tailored for Strathmore University. The system automated the complex process of creating and managing exam schedules across all courses and programs offered by the university. By incorporating some brilliant algorithms, it optimized resource allocation, taking into account factors such as course requirements, and room availability. Additionally, it offerd real-time, user-friendly access to personalized exam timetables for both students and faculty, promoting transparency and collaboration. The project prioritized rigorous testing, user training, and comprehensive documentation to ensure the system's reliability and effectiveness in revolutionizing the university's exam scheduling processes.

## **Limitations**

While this project aimed to automate and optimize the exam scheduling process, it is important to acknowledge that the so-called full automation could not be feasible. According to Growiec (2022), partial automatability makes people and programmable machines complementary, whereas full automatability makes them substitutable. In this project, human beings were not substituted by the system. The university exam office team remained an essential part of the system's operation. Upon receiving initial data from faculty administrators, they must be responsible for making initial suggestions for exam slots and inserting them into the database. Once this initial input is provided, the system takes over to check for conflicts, goes though some few step solutions as the process of optimizing the scheduling timetable before generating the final copy.

Additionally, resource constraints have been posing significant limitations to the project. Both time and budget limitations have impacted the extent to which advanced features and optimization algorithms could not be implemented in the system. The initial suggestion was to user the heuristic algorithm that is known as an algorithm that find better and efficient problem-solutions. Unfortunatly during the system development the experience came to prove that this algorith cannot help due to the complexity of the system’s constraints. In the process of identifying and solving conflicts, the system was dropping some complicated and unsolvable cases.

Integration with existing university databases and systems may encounter challenges related to data migration and compatibility issues. User adoption and acceptance are critical to the system's success, and any resistance to change or unfamiliarity with the new system among stakeholders may hinder the realization of its benefits.

# Chapter 2: Literature Review

## **Introduction**

##### In higher education environments, the scheduling of examinations has always been a critical and multifaceted endeavor. This chapter delves into four key sections, each unveiling a distinct facet of the mentioned complex process. First, the chapter will explore the complexity of exam scheduling timetables in higher educational institutions, shedding light on the numerous factors that contribute to the complex nature of this task. Next, will be some challenges faced by the Strathmore University Examination Office during exam scheduling, providing real-world insights into specific tears encountered. Subsequently, the chapter will investigate related systems and technologies so far designed to streamline and enhance the exam scheduling process. Finally, this chapter will culminate in providing a conceptual analysis for the proposed system. This later, will grasp the theoretical foundations and fundamental principles shaping how the development of the scheduling system especially the case of Strathmore University will look like.

## **Complexity of Exam Scheduling Timetable in Higher Educational Institutions**

First and foremost, it is important to acknowledge that the complexities of academic exam scheduling transcend the confines of Strathmore University, reflecting global challenges within the realm of exam scheduling systems that extend far beyond its campus. These challenges are not unique but rather universally shared among institutions worldwide. According to Burke (2004), a timetabling scheduling problem always consists of four parameters: 1) a finite set of times T 2) a finite set of resources R 3) a finite set of meetings M 4) a finite set of constraints C. Here, the challenging task is to allocate T and R to M to satisfy C as much as possible. Apart from all these challenges that appear common to institutions, others appear specific to some universities, such as accommodating retake students, navigating individual course requirements, aligning seamlessly with the diverse array of student preferences etc., all while optimizing the allocation of precious resources.

This universality of challenges found a notable illustration in the insightful work of Woumans, De Boeck, Beliën, and Creemers. In their comprehensive study, they shed light on the recurrent and time-consuming nature of examination timetabling that persisted as a prevailing concern in educational institutions on a global scale. Their research served as a vivid testament to the widespread manual exam scheduling challenges encountered in universities worldwide, thereby emphasizing the paramount importance of considering an array of constraints and preferences when endeavoring to construct efficient timetables (Woumans, 2016).

Trying to solve these challenges, relying on conventional tools, notably Microsoft Excel has been manifested as a prevailing phenomenon that were resonating across a multitude of academic institutions. Many universities, much like Strathmore, have heavily depended on Excel-based systems for this purpose. However, these seemingly omnipresent tools have inherently been grappling with limitations in their capacity to adapt to evolving circumstances and efficiently allocate essential resources.

Delving into the profound significance of the examination timetabling problem within the realm of academic institutions, Hussain & Sharma (2017) illuminated how the escalating enrollments of students, the proliferation of diverse courses, and the continuous expansion of degree programs collectively contribute to the ever-growing complexity of the examination timetabling problem. Consequently, this underlines the pressing need to confront the substantial constraints imposed by universities when crafting examination schedules.

Limitations inherent in Excel-based systems, encapsulating the vulnerability to scheduling conflicts and susceptibility to data input inaccuracies, seamlessly aligned with the challenges encountered by Strathmore University, further emphasizing the universal nature of this predicament. It served as a testament to the intricate landscape of the exam scheduling system faced by academic institutions globally, necessitating diligent efforts to surmount these common obstacles.

Although computerized timetabling and administration systems were available to offer break from this perpetual task, each timetabling endeavor remained inherently unique and context-specific, intrinsically shaped by the distinct peculiarities of the institution it dutifully serves. According to Salem ( 2023), the greater the number of constraints in the problem, the more difficult the exam timetabling problem becomes. In addition, some constraints contradict each other, which makes timetabling a challenging task, and most universities have been trying to solve it "manually" every semester. Besides, there are a great number of different constraints that make exam-timetabling problems different from one institution to the other (Salem, 2023). The central challenge revolves around the judicious assignment of examinations to precise time intervals and physical spaces, all in the persistent pursuit of mitigating or altogether eliminating scheduling conflicts. These conflicts have been manifesting themselves in scenarios where students or invigilation members could find themselves simultaneously scheduled to occupy the same geographical locus or when a single room is scheduled to host the concurrent activities of multiple distinct groups.

## **Challenges faced by Strathmore University Examination Office**

## **Introduction**

In section 2.2, some complexities of the exam timetabling faced in higher education institutions were analyzed. Within the broader context of this academic scheduling complexity, it is crucial to emphasize that each educational institution possesses its own unique set of challenges and requirements when it comes to the exam scheduling system. While commonalities exist, the specificity of individual institutions could not be overlooked. Strathmore University, like many other institutions, has been facing the complex task of accommodating diverse subjects, catering to the preferences of both students and faculty and optimizing resource allocation. To develop a healthy solution personalized to Strathmore's distinctive policies, it has been essential to consider the institution's peculiarities comprehensively.

## **Managing a Large Student Population**

Strathmore University currently accommodates a significant enrollment of 7,300 students, distributed across 18 courses and various student groups. This diverse student body poses a notable challenge in creating an efficient and well-structured exam timetable. To tackle this challenge, the university employs a practice of categorizing classes into multiple groups, identified by designations such as BBT 1.1 A, BBT 1.1 B, BBT 2.1 A, and BBT 2.2 C, among others. However, this segmentation introduced an additional layer of complexity to the creation of comprehensive and cohesive exam schedules for each specific group. The objective was to formulate timetables that not only prevent conflicts but also ensure an equitable and accommodating examination experience for all students across the diverse spectrum of class groups.

## **Catering to Special Exam Requirements**

Within these designated student groups, there arise instances where students present unique examination needs, including retake exams or examinations scheduled in groups separate from their regular class assignments. Effectively accommodating these students within the exam timetable poses a considerable and intricate challenge. Take, for example, a scenario where a student is assigned regular exams within one group and retake exams within another. In such cases, the creation of the timetable requires particular attention to detail, ensuring the avoidance of scheduling conflicts and the provision of adequate intervals between successive exams. This level of precision is crucial in crafting an equitable examination schedule that caters to the diverse needs of Strathmore University's student body.

## **Addressing the Challenge of Cumbersome Timetables**

Strathmore University faces a significant challenge with the current practice of distributing exhaustive exam timetables via email to students. The sheer volume of information, often spanning multiple pages, necessitates a more streamlined approach. Instead of overwhelming students with lengthy documents, there is a pressing need to transition to a system where each student, identified by their unique student number, can access a personalized timetable exclusively for the courses they are enrolled in.

This transition aims to offer a more user-friendly and efficient solution. It ensures that students only view the exam schedule relevant to their specific courses, eliminating the need to sift through extensive documents. Moreover, adopting this approach aligns with the university's commitment to personalized education, allowing for tailored schedules that cater to individual needs, such as retaking exams or other exceptional circumstances. This adjustment will absolutely minimize stress and logistical complexities. However, implementing this change poses its own set of challenges. Crafting individualized timetables within the constraints of available time slots and venues requires duteous attention to detail and precision, which cannot be possible with the current system being used by the university exam office.

## **Handling Exam Timetable Adjustments**

When students receive the initial draft timetables and encounter scheduling conflicts or other issues, it is commonplace for them to request adjustments. These adjustment requests, while crucial for ensuring student comfort and fairness, present a substantial and intricate challenge for the examination office to manage. Alterations made to accommodate one student can have cascading effects on the entire timetable, compelling the need for comprehensive coordination and potentially impacting the schedules of other students. Balancing the individual needs and preferences of students with the integrity of the overall exam timetable requires careful attention to detail and adaptability within the examination office's responsibilities.

## **Communication and Complaint Management**

The communication process between the examination office and students, especially when adjustment requests arise, can become a cumbersome endeavor, primarily due to the extensive student population. Effectively managing a substantial flood of complaints, addressing individual concerns, and maintaining a commitment to fairness and equity in scheduling injects an additional layer of complexity into the office's already demanding set of responsibilities. The scale of the student body's inquiries necessitates a systematic approach to ensure timely and considerate responses to each student's unique situation, all while upholding the principles of impartiality and transparency within the examination office's operations.

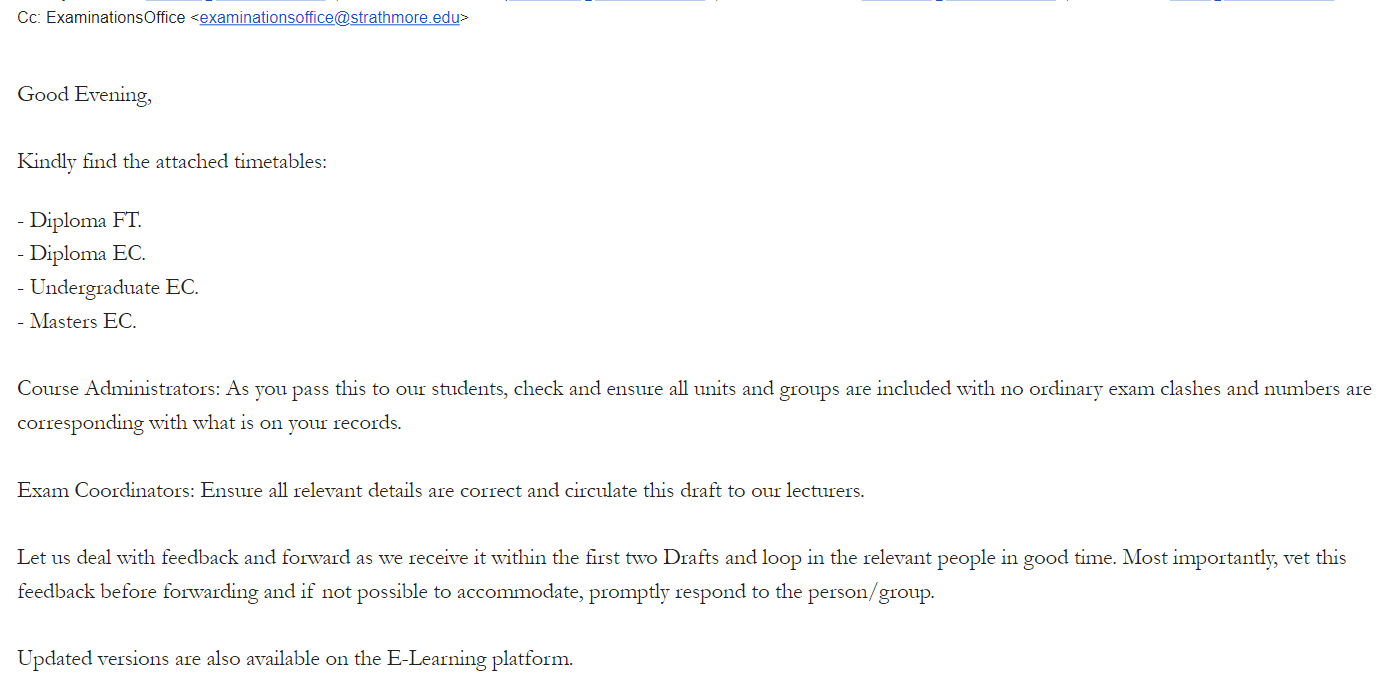
## **Resource Allocation and Venue Constraints**

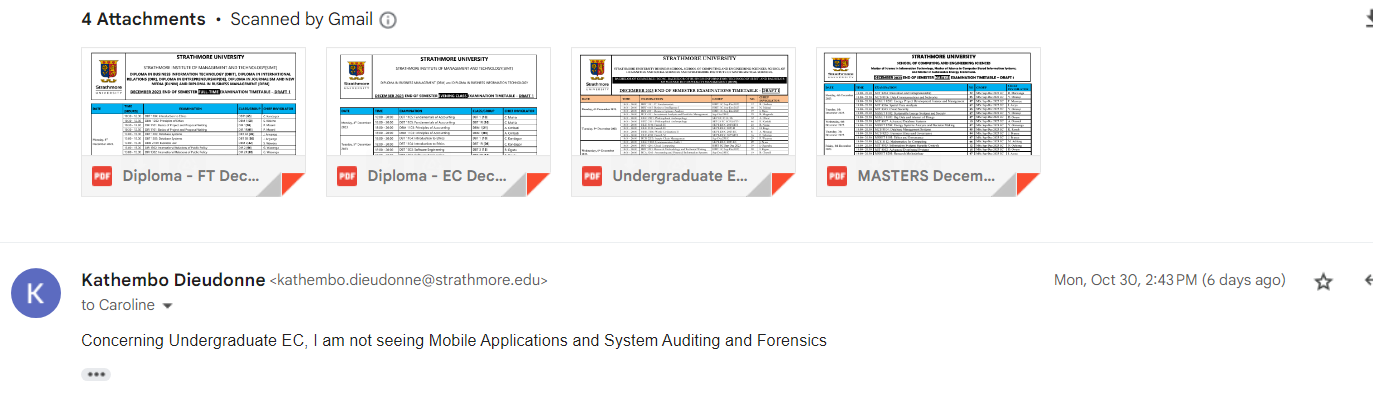
The allocation of appropriate examination venues and resources to accommodate a significant student population poses another formidable logistical challenge. Strathmore University must methodically ensure the availability of a sufficient number of classrooms or venues capable of accommodating multiple exam groups concurrently. This task encompasses a thorough evaluation of venue size and suitability for various examination formats, adding a layer of complexity to the allocation process. Ensuring that venues are not only adequate in number but also personalized to specific requirements of different types of exams demands precise planning and resource management within the university's examination infrastructure.

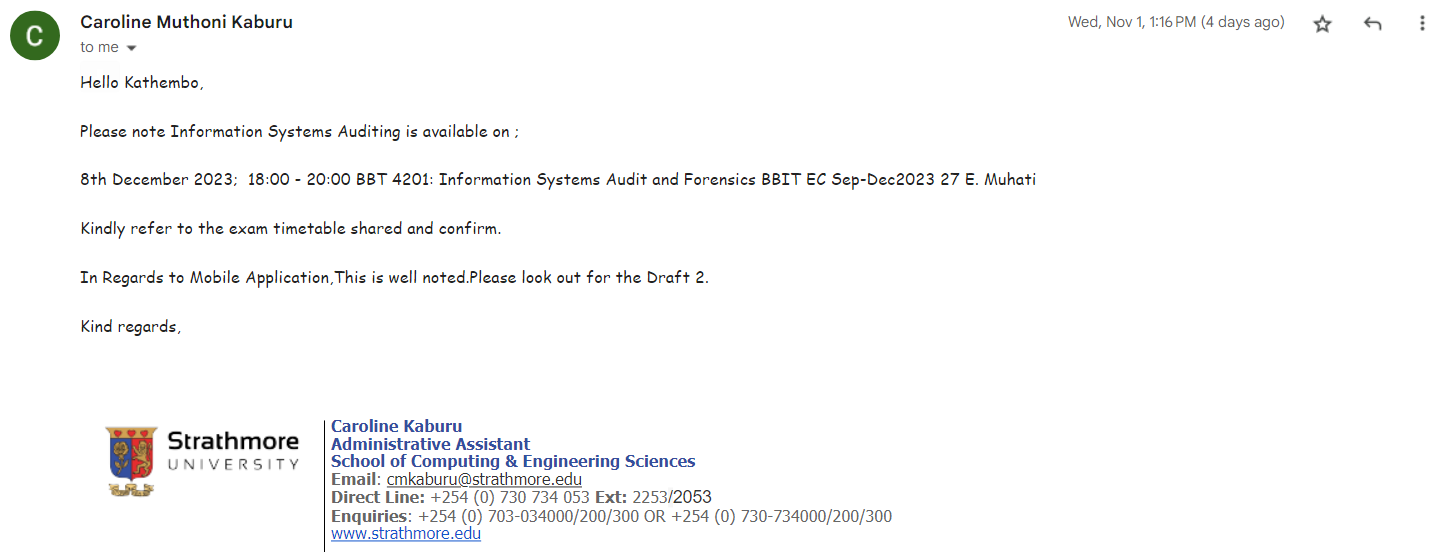
## **Conflict Resolution and Fairness**

Ensuring equitable treatment within the exam timetable for all students holds a paramount importance. Notably, challenges have arisen when conflicts emerge between the scheduling preferences of diverse student groups or when the accommodation of retake students impacts the scheduling of regular exams. Striking a delicate balance between the resolution of conflicts and the steadfast commitment to fairness proves to be a highly intricate endeavor. This process necessitates great consideration of each student's unique needs, all while adhering rigorously to established policies and constraints. The examination office's role in achieving this equilibrium is demanding, as it strives to provide an exam schedule that embodies both equity and compliance with institutional standards.

###### Figure 2.1: Email sharing draft 1 for exam timetable December 2023

Figure 2.2: Email from the exam office to the administrators

Figure 2.3: One of the complaints from students



## Figure 2.4: Response from the exam office to the student complaint

## **Conclusion about challenges faced by the Strathmore University Exam Office**

While several additional challenges could be incorporated into this list, it is imperative to acknowledge that we must pause at this point, reaching a total of seven challenges. Each challenge illuminated above serves to underscore the formidable complexity inherent in the task of exam scheduling system at Strathmore University. Within this context, the examination office is tasked with the intricate role of navigating through a multitude of variables, constraints, and unpredictable circumstances to formulate a schedule that comprehensively caters to the diverse needs of the university's expansive student body.

## **Related Systems**

## **Introduction**

In the territory of academic institutions, the efficient scheduling of examinations remains a pivotal administrative task that significantly impacts the educational experience for both students and faculty members. This section explores three notable works that have tried to address the complex challenges associated with automated university exam timetable systems. The first work, the "Automated University Timetable System" by Joseph Ayo Babalola University, leveraged a diverse array of technologies to streamline the timetabling process. The second work, the "Automated Examination Timetable Generation and Invigilator Scheduling System" from Yusuf Maitama Sule University Kano, employed Genetic Algorithm and Java programming to optimize resource allocation and enhance user interfaces. Lastly, the "Re-Engineering of Examination Timetable Generation and Invigilation Scheduling System" at Nnamdi Azikiwe University introduced user role clarity and refined user experiences. These works collectively demonstrate the ongoing efforts to tackle the complexities of university timetabling, offering insights into technological advancements and the evolving needs of academic institutions.

## **Automated University Timetable System**

#### Contextual understanding of the system

Timetabling is one of the most important administrative activities that requires huge attention from the school authorities in all educational institutions (Oluwaniyi et al., 2016). Finding how it was a serious challenge, Joseph Ayo Babalola University, and Osun State all located in Nigeria, came together and shared skills to solve this problem most educational institutions have been facing. They designed and implemented a web-based automated platform for university examination timetabling that they called the "Automated University Timetable System”.

#### Tools and technologies

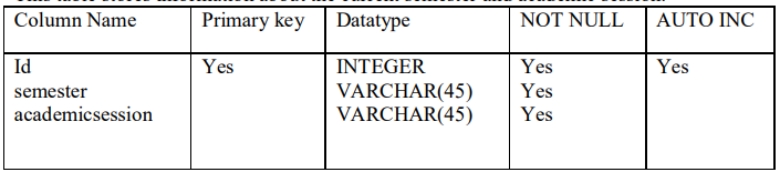
The development of the Automated University Timetable System involved the strategic utilization of a diverse array of tools and technologies. To craft the system's web interface, they made use of Macromedia Dreamweaver, a versatile web development platform known for its user-friendly interface and robust features. In parallel, MySQL played a pivotal role as the system's database, offering healthy data management capabilities and ensuring efficient data storage and retrieval. The programming languages they employed included HTML (Hypertext Markup Language), which allowed to structure and format of content, PHP (Hypertext Preprocessor) for server-side scripting, JavaScript for enhancing user interactions, CSS (Cascading Style Sheets) for sophisticated page design, and SQL (Structured Query Language) to interact with the database. Notably, PHP, being a server-side scripting language, was cleverly integrated into HTML, enabling seamless connections between web pages and the MySQL database. This integration facilitated the dynamic generation of web content, a crucial feature for the Automated University Timetable System's real-time updates and user-friendly interface.

#### Database structure

During the development of the Automated University Timetable System, two distinct databases were crafted: examtimetabledept and examtimetable. The former, examtimetabledept, was architected to serve as the repository for tables about every department within the esteemed university, offering a comprehensive structure for data organization and management (Oluwaniyi et al., 2016). Conversely, the latter, examtimetable, was purposefully designed with a more streamlined focus, exclusively accommodating tables specifically relevant to administrative functions, encompassing critical facets such as administrator checks, staff management, venue allocation, and the intricate timetabling complexities intrinsic to the academic realm (Oluwaniyi et al., 2016).

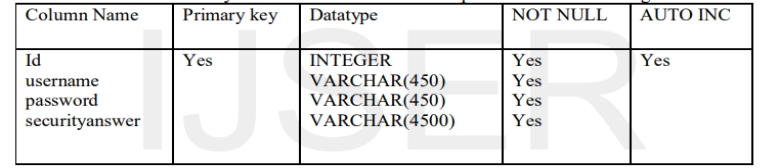
Table 2.1: Administratorcheck

Within the annals of the system's database structure, Table 1, Administratorcheck, stood as a sentinel, diligently safeguarding pivotal information about the current semester and the academic session.



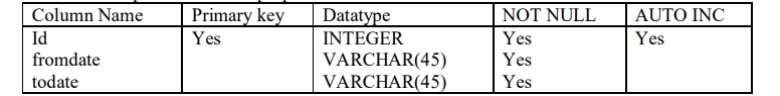
## Table 2.2: Controlsub

Nestled within the database's framework, Table 2, Controlsub, played a multifaceted role, skillfully storing user credentials in the form of usernames and passwords while maintaining the security mantle by incorporating security answers. These answers, in the event of password amnesia, were ingeniously leveraged to facilitate password retrieval (Oluwaniyi et al., 2016).



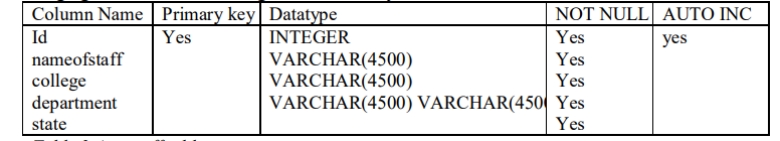
## **Table 2.3: Examdate**

At the heart of the database, Table 3, Examdate, chronicled the crucial details about the scheduled examination dates, ensuring that the institution's academic calendar flowed seamlessly.



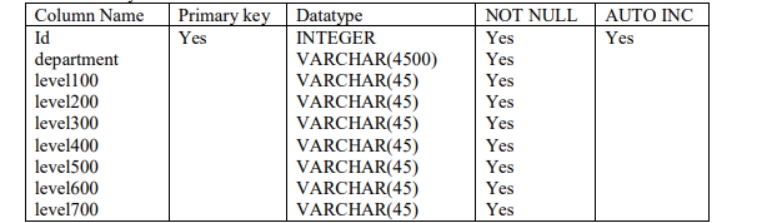
## Table 2.4: Staff

In the complex needlepoint of data management, Table 4, Staff, assumed the role of a sentinel, documenting the diverse states and roles of staff members and invigilators, ensuring efficient personnel management and allocation of responsibilities.



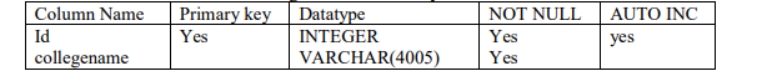
## Table 2.5: StudentStrength

Table 5, StudentStrength, recorded and categorized the total number of students at each academic level sorted by their respective departments within the university.



## Table 2.6: The University

Table 6, The University, served as the guardian of institutional knowledge, housing the prestigious names of all colleges affiliated with the university.



## Table 2.7: The University Department

Extending its role, Table 7, The University Department, was documenting the comprehensive landscape of academia, preserving not only the names of colleges but also the departments within the given institution.

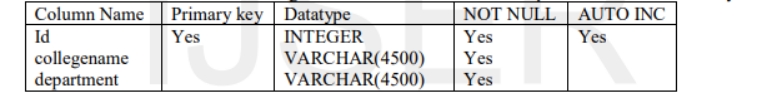
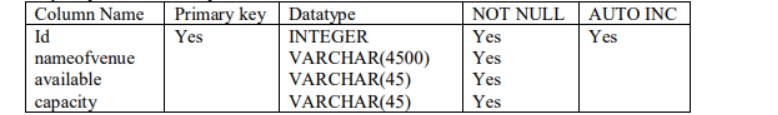
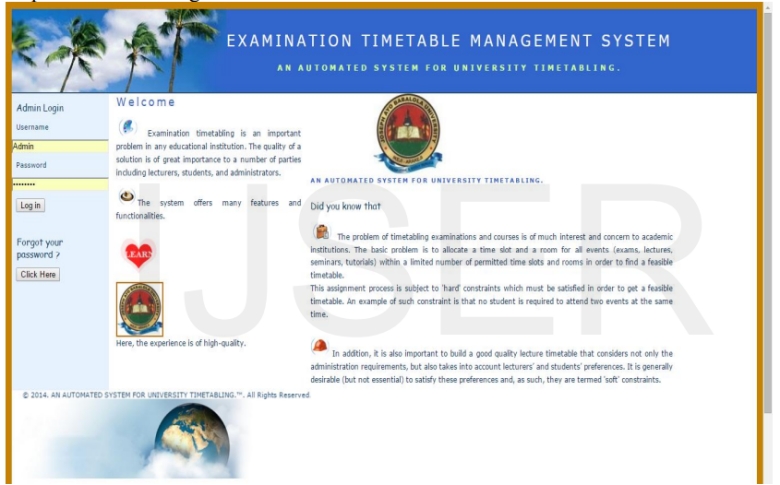


Table 2.8: Venue

Finally, Table 8, Venue, played a pivotal role, maintaining an exhaustive record of all examination venues, methodically cataloging their capacities, and diligently monitoring their availability status, ensuring the unified orchestration of examinations within the university's academic confines.



#### Modules and Outputs

When looking at the login module for the automated Timetable for universities, it can be realized that the login appears the privilege of the admin alone. On the login page, the administrator is the user of the system and can log in to the system by supplying the right username and password. 

## Figure 2.5. Login (Oluwaniyi et al., 2016).

After the admin logs in, the system opens the main page and exposes various menus existing in the system. As can be seen from the main page figure, the system does not seem to have a friendly process. There is a set of processes that are done between the login process and the exam timetable generation process.



## Figure 2.6: Main page (Oluwaniyi et al., 2016).

The final process is the one of generating the examination timetable for the university. Below is the figure of the generated copy showing only three columns and time slots: day, course codes, venue, and time. Once generated, the timetable could be printed, distributed, and uploaded on the university webpage for easy access.

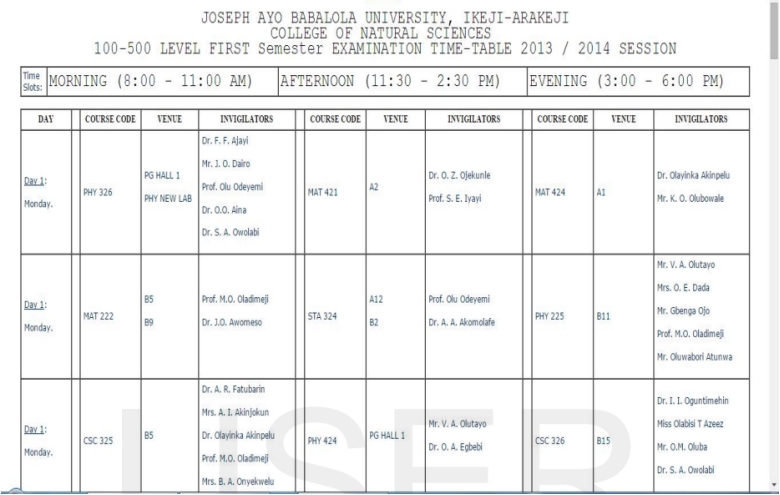


Figure 2.7: Exam timetable generation (Oluwaniyi et al., 2016).

#### Limitations of the Automated University Timetable System

The system may face challenges in efficiently accommodating retake students who have exams in groups different from their regular classes. Since retake students often have unique scheduling requirements, the system might struggle to create timetables that cater to their needs while also ensuring minimal conflicts. Actually, it appears to rely on group-based scheduling, which can be restrictive for retake students. If a retake student has subjects in multiple groups, the system may find it challenging to create a cohesive timetable that minimizes conflicts and ensures reasonable gaps between exams.

Managing data consistency and integration between two distinct databases (examtimetabledept and examtimetable) can be complex. When attempting to relate department-specific data with administrative data, the system might require complex queries and synchronization mechanisms, potentially leading to data consistency challenges.

Handling backups and recovery processes for two separate databases can be more time-consuming and complex. If not managed meticulously, this complexity could lead to operational disruptions and potential data loss.

The system's main page interface might not be user-friendly, potentially making it challenging for administrators and users to navigate and utilize effectively. Improving the user interface could enhance the overall user experience.

## **Automated Examination Timetable Generation and Invigilator Scheduling System**

#### Contextual understanding of the system

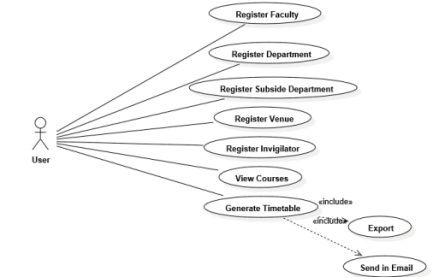
Truly speaking, University timetabling is a highly constrained problem that demands considerable time and effort in the process (Aminu et al., 2019). The Yusuf Maitama Sule University Kano (Nigeria) recognized the severe dilemmas faced by stakeholders when manually creating timetables, resulting in numerous clashes (Aminu et al., 2019).The pressing need for a solution to eliminate energetic and burdensome tasks in examination timetables and invigilator scheduling became evident. This realization led to the development of the Automated Examination Timetable Generation and Invigilator Scheduling System, a pioneering solution. This system tried to streamline, schedule, optimize resource allocation, and minimize errors in the timetable. It eased the burden on university administrators, contributing to a harmonious academic environment. It reflected the university's commitment to technology and innovation in education. The system, intelligently considered room availability, faculty preferences, and student needs, creating a balanced and conflict-free timetable. As a solution to an existing problem, the system managed to enhance the overall academic experience by ensuring smooth examination administration and reducing stress for both students and faculty members.

#### Tools and Technics

The Automated Examination Timetable Generation and Invigilator Scheduling System was built using Genetic Algorithm, a technique that provides an optimal solution by randomly and effectively selecting a time slot for a particular subject, venue, and invigilator. The solution was implemented using Java as a programming language. Graphical User Interfaces were built using Java Swing. And finally, SQLite was used for database-related activities.

#### Functional Components

The Automated Examination Timetable Generation and Invigilator Scheduling System consisted of eight distinct modules, as documented by Aminu et al. in 2019. The below figure (Figure 2.4) illustrates the interaction between users and the system's various components.



## Figure 2.8: Functional components (Aminu et al., 2019)

Within the faculty module, users undertake the task of registering the available faculties within the university. Following the registration of faculties, they proceed with the registration of affiliated departments and the corresponding courses associated with each faculty. In parallel, the system accommodates the registration of examination venues and invigilators, each contained within their respective modules. Notably, invigilators review diligently assigned to each examination while adhering to a prescribed teacher-to-student ratio of 1:40 (Aminu et al., 2019)..

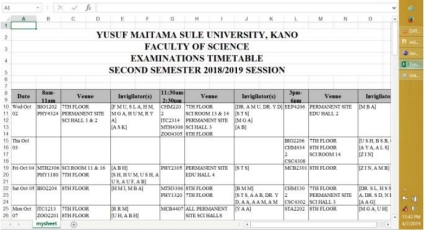
The pivotal "generate timetable" module facilitates users in configuring essential parameters. This includes specifying the examination duration, selecting the desired academic semester, and invoking the timetable generation process by clicking the designated "generate" button. Upon successful generation, the system provides users with the capability to export the resultant timetable in either Excel or PDF formats, allowing flexibility in storage and distribution. Furthermore, users have the option to dispatch this generated timetable via email, appended as an attachment to predefined email addresses.



## Figure 2.9: Interaction with the system (Aminu et al., 2019)



Figure 2.10. Dashboard showing various modules (Aminu et al., 2019)



## Figure 2.11: Module where the timetable generation is being processed (Aminu et al., 2019)

#### System added values

The Automated Examination Timetable Generation and Invigilator Scheduling System, utilizes the Genetic Algorithm, a powerful optimization technique, to provide optimal solutions for subject, venue, and invigilator assignments. According to some researchers, GA is the best choice for multi-objective optimization and is applicable for a large class of problems (Nayyar et al., 2018). Although not explicitly mentioned, this system likely has a more robust approach to data consistency and integration, given its focus on optimization and efficiency. It likely implements better synchronization mechanisms, addressing the data-related challenges of the previous system. Concerning the User Interface Enhancement, we can realize too, that the system has improved the user interface design. Compared to the first system, the process here seems friendlier and easier to navigate.

#### System limitations

In the Automated Examination Timetable Generation and Invigilator Scheduling System, a notable limitation revolves around accommodating retake students who may find themselves scheduled for exams across multiple groups. Retake students often possess distinct scheduling requirements, adding complexity to the task of creating a timetable that effectively ensure each student has a reasonable and conflict-free schedule. To break this down further, students typically retake specific exams from different groups or even different semesters. This complexity results in scheduling conflicts, as the system must manage exam placement to guarantee that retake students have adequate time intervals between their examinations. This is essential not only to prevent exhaustion but also to ensure a level playing field academically.

Moreover, this scheduling challenge intertwines with the broader issue of balancing student preferences. The system encounters difficulties in harmonizing the individual preferences and requirements of retake students. It's like attempting to satisfy everyone's wishes within the confines of limited resources, an inherently challenging task. Finding a solution to this challenge could be vital for the equitable and efficient administration of examinations.

Another notable limitation of the Automated Examination Timetable Generation and Invigilator Scheduling System is its lack of provision for individual students to access their personalized timetables. In this system, the generation process primarily focuses on creating a comprehensive and general timetable that encompasses the entire school's examination schedule. However, it does not offer a user-friendly interface or functionality for students to directly access their specific exam schedules. To put it simply, students do not have the convenience of checking their individual timetables within the system. Instead, the system primarily caters to system administrators who handle the overall scheduling process. This lack of direct interaction between students and the system is another significant shortcoming.

In practical terms, this limitation results in several inconveniences for students. Without easy access to their personalized timetables, they struggle to efficiently plan their study routines and other commitments around their exams. They rely on administrative announcements or manual distribution of printed timetables, which are less efficient and timely. Furthermore, the absence of a student interface within the system means that students cannot quickly check for updates or changes to their timetables. In an academic environment where adjustments to exam schedules can occur, this limitation may lead to confusion and potential scheduling conflicts for students.

Addressing this limitation would involve developing a student portal or interface within the system that allows students to access their individual timetables. Such a feature would greatly enhance the user experience for students, providing them with a convenient and reliable way to stay informed about their examination schedules and any updates that may arise.

## **Re-Engineering of Examination Timetable Generation and Invigilation Scheduling System**

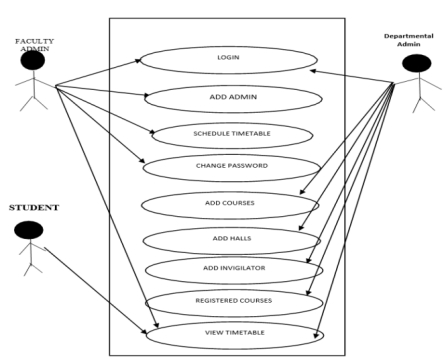
#### Contextual understanding of the system

In 2022, Nnamdi Azikiwe University, situated in Awka, Nigeria, embarked on a technological endeavor aimed at enhancing the automation of its educational management processes. To contextualize this initiative, it’s worthy to refer to a prior study conducted by Mbonu Chinedu (Emmanuel et al. in 2022). This undertaking was rooted in historical circumstances dating back to the conclusion of the 2008-2009 academic year when the Faculty of Physical Sciences at Nnamdi Azikiwe University, encompassing various academic Departments such as Computer Science, Geological Sciences, Mathematics, Physics, Industrial Physics, Pure and Applied Chemistry, and Statistics, sought to develop an advanced examination timetable scheduling system. This system wanted to displace the traditional manual approach to examination scheduling, with the primary objectives being:

1. Identification of deficiencies within the existing examination scheduling system.
2. Development of a scheduling system that prevents the simultaneous scheduling of examinations for two distinct faculty courses on the same day, especially for students within the same academic level and department.
3. Creation of a system capable of assessing examination hall capacities and efficiently allocating students to these halls.
4. Implementation of measures to prevent the double-booking of examination halls during concurrent time slots

#### Tools and techniques

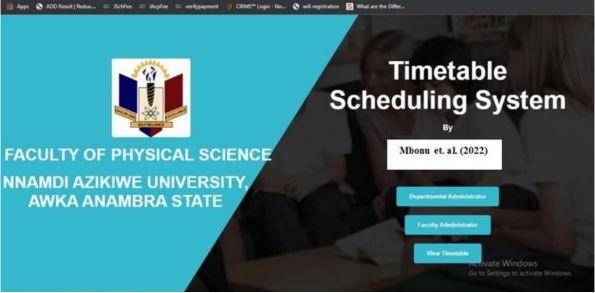
The system was framed through the utilization of the PHP programming language, comprising three integral constituents: the client interface, middleware, and database infrastructure. The client interface was constructed employing HTML version 5, CSS, and JavaScript technologies, while the middleware layer was engineered upon the WGSI web server framework. The database component was underpinned by the MySQL database system. This database, serving as the repository for all pertinent records, was structured employing MySQL (MariaDB) and configured for management via the PHP MyAdmin graphical user interface database management system, seamlessly integrated within the XAMPP Server environment. The system's prerequisites and specifications were defined and documented through the consumption of a Use Case organization, as visually represented in the following figure.



## Figure 2.12: Use Case for the Re-Engineering of Examination Timetable Generation and Invigilation Scheduling System (Emmanuel et al. in 2022).

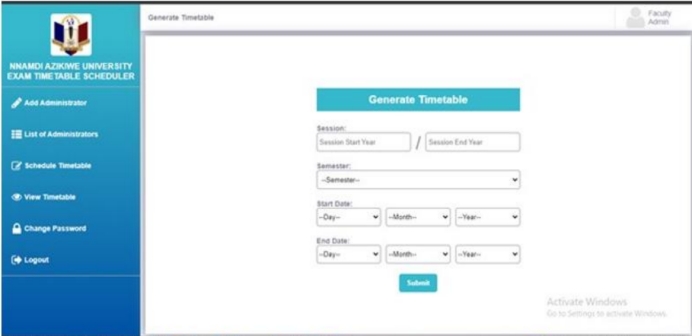
#### Interaction with the system

This independent scheduling system for the Faculty of Physical Sciences at NAU, is a web-based application that does not require installation on physical hardware but rather can be accessed through a local host web server. As visually seen in Figure 2.8, three actors interact with the system: the Faculty admin, Departmental admin, and Students. The Faculty is the super admin of the system who is in charge of adding departmental admin and scheduling examinations. The Departmental admin carries out activities that are important at the departmental level such as; uploading courses (specifying the departments that offer it and the total of number of students), adding invigilators, and providing available halls and their capacity for examination. Though the student does not perform any activity that helps in the scheduling of the examination, at least every student can know the days on which their examination falls. Therefore, a student can access the system to view the generated timetable.



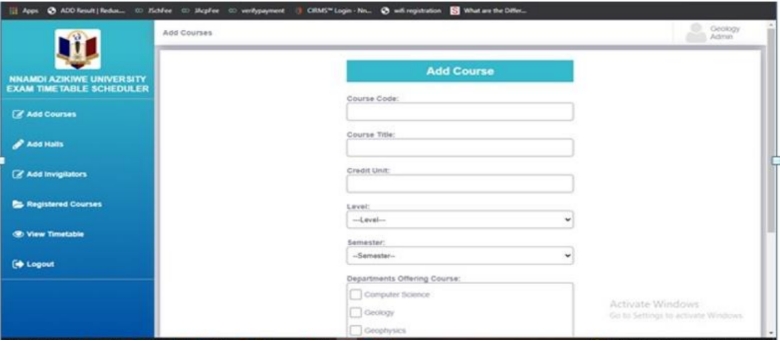
## Figure 2.13: Home Page (Emmanuel et al. in 2022)

As can be seen, the above home page of the new system is accessible to all the users of the system; Faculty administrator, departmental administrator, and the student as shown in figure 2.9. The faculty administrator clicks on the faculty administrator button before supplying the username and password to log in, such is also applicable to the departmental administrator clicking on the departmental administrator button before supplying the username and password to log in. The students only have access to the view timetable button.

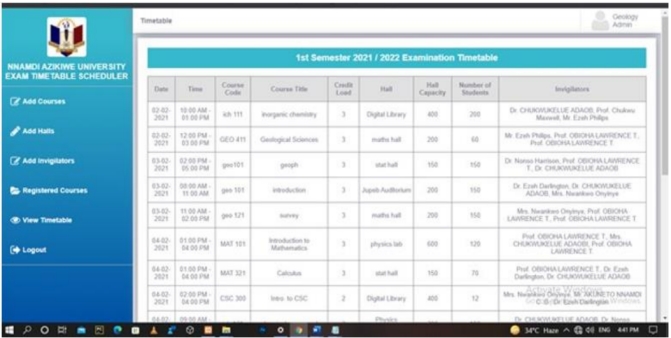


## Figure 2.14: Faculty dashboard (Emmanuel et al. in 2022)

The faculty administrator's dashboard depicted in figure 2.10 shows the dashboard that is visible when the Faculty admin logs in to the system successfully. The faculty administrator has the access to add a departmental administrator, view the list of administrators, schedule a timetable, and as well view the scheduled timetable.

 Figure 2.15: Departmental administrator dashboard (Emmanuel et al. in 2022)

The departmental administrator's dashboard as depicted in figure 2.11 shows the necessary functions a departmental administrator can perform in the new system, they include: add courses, add halls, add invigilators, view registered courses and view timetables.



## Figure 2.15: Generated Timetable by the Faculty Administrator

The generated timetable shown in figure 2.12 is the output interface in the expected examination timetable. This timetable is generated by the Faculty administrator who has the higher privilege of scheduling examinations. The timetable generated by the faculty administrator is a result of the information provided by various departmental administrators in each semester.

#### System added values

In comparison to the Automated Examination Timetable Generation and Invigilator Scheduling System, the Re-Engineering of Examination Timetable Generation and Invigilation Scheduling System has endeavored to enhance the system's functionality and introduce additional features. A primary augmentation pertains to the elucidation of user roles and their respective use cases. In the Automated Examination Timetable Generation and Invigilator Scheduling System, there existed a solitary system user responsible for all interactions and operations. However, the precise identity and role of this user remained ambiguously defined. It remained unclear whether this user occupied the position of a departmental administrator or a faculty administrator, as these specifics were inadequately elucidated. Conversely, within the framework of the Re-Engineered Examination Timetable Generation and Invigilation Scheduling System, notable improvements have been introduced, including a refined user structure with distinct responsibilities. The system now accommodates three distinct user categories, each endowed with unique privileges and varying degrees of system access.

#### System limitations

Upon examination of the system's home page, which serves as the point of entry for all system users, it becomes evident that there are three distinct buttons in existence. For the departmental administrator to access and engage with their dedicated dashboard, tailored to their specific roles and responsibilities, they are required to click the Department button. Similarly, the faculty administrator is mandated to engage with their designated functionalities by clicking the Faculty button. Meanwhile, the student's sole interaction entails the checking of the generated timetable, an action facilitated through the view button. This configuration is subject to an inherent redundancy in user login procedures. It should be imperative to consider a streamlined approach wherein a singular login interface is employed for all system users. Subsequently, predicated upon each user's authentication credentials, the system should intelligently direct them to their required interface, customized to align with their respective roles and positions within the system.

Another notable observation is that the system currently generates a singular timetable for each specific semester. This prompts an important inquiry: how does the system address the scenario wherein repeat students are re-enrolled in courses from previous semesters, distinct from their regular academic schedule? Consider, for instance, a student in their 4th year, semester 1, who is repeating a subject from their 2nd year, semester 2. How does the system effectively accommodate such cases while ensuring the avoidance of scheduling conflicts within the timetables? This is another big challenge coming out as a system limitation.

Furthermore, the process of individually accessing and reviewing personal timetables poses a persistent challenge. By clicking the View button on the homepage, users are presented with the entire timetable for the entire faculty, necessitating meticulous scrutiny to ascertain their specific examination schedules. This issue extends to invigilators who are also burdened with the task of cross-referencing the comprehensive timetable.

To alleviate these concerns, it would be good to create a more user-friendly system wherein users can conveniently access their personalized timetables with ease, and invigilators are presented with a more streamlined and task-specific view of their invigilation schedule.

## **Heuristic Algorithm for scheduling systems**

## **Overview about Heuristic Algorithms**

In contemporary computational paradigms, computers have been playing an integral role in addressing complex problems. The resolution of such complexities has necessitated sophisticated algorithms. This is how the advent of approximation algorithms has become unavoidable in making solution-finding algorithms viable for addressing complex problems. These approximation algorithms leveraging heuristics and meta-heuristics functions, have been designed to achieve pragmatic solutions (Sachin Desale, Akhtar Rasool, Sushil Andhale, Priti Rane, 2015). Through strategic utilization of heuristic functions, their algorithms demonstrate an inherent capability to navigate conveniently complex problem domains, offering practical solutions within acceptable computational bounds. This, actually connect with what Todd and Gigerenzer (2000) had underlined saying the term *heuristic,* from the Greek, means, “serving to find out or discover”. In the context of problem solving as it has been mentioned, heuristics are experientially coming out as derived cognitive “rules of thumb” (Yu-Chi Ho, 1994) that serve as guides in problem-solving processes.

## **Benefits of Heuristics in Problem Solving**

In the context of system scheduling, heuristics play a pivotal role in optimizing the allocation of resources and managing the execution of tasks efficiently. According to Todd and Gigerenzer (2000), heuristics serve as robust guidelines that prove effective across various scheduling scenarios. From well-controlled laboratory settings, as demonstrated by Orbell and Dawes (1991), to the complexities of real-world scheduling challenges, you find the surprising accuracy of heuristics extending.

Pointing to efficiency, it is worth noting that the concept of efficiency in system scheduling is not solely determined by energetic effort but by how efficiently tasks are allocated and managed. Heuristics, with their ability to provide quick, rule-of-thumb solutions, enhance efficiency in the scheduling process. Whether dealing with well-structured scheduling problems or navigating the complexities of ill-structured scenarios (Chi, M.T.H and Glaser, 1985), heuristics demonstrate their adaptability and effectiveness (Charles F. Abel, 2003).

In concrete scheduling situations, heuristics shine. These approximation algorithms can yield results comparable to more complex scheduling strategies that consider all available information and employ complex computational processes. This versatility positions heuristics as valuable tools in the realm of system scheduling, contributing to streamlined and effective task allocation in both theoretical and real-world scheduling environments (Charles F. Abel, 2003).

## **General Heuristic Model in Problem Solving**

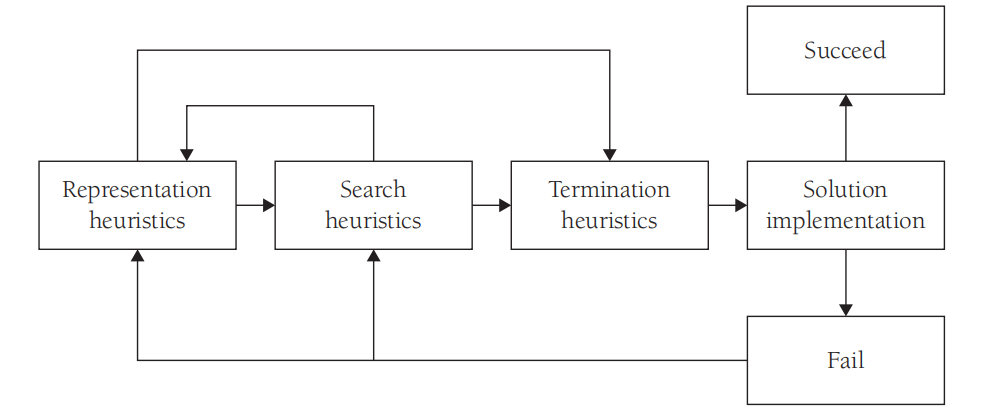


Figure 2:16: General Heuristic Model in Problem Solving (Charles F. Abel, 2003).

The above general heuristic model presents a valuable framework for guiding efficient problem-solving across diverse disciplines. The process starts with representing heuristics, goes through different stages and ends with the solution implementation.

#### Representation Heuristics

In the context of exam scheduling system being proposed, representation heuristics emphasize the need for students, administrators or system users, to create a concrete and tangible representation of the exam scheduling problem. This involves visually organizing information related to exam dates, times, locations, and constraints in a way that is clear and understandable. By creating a visual representation, administrators must more effectively identify and understand constraints associated with exam scheduling. These constraints may include factors such as room availability, invigilator schedules, and avoiding conflicts for students with multiple exams on the same day.

#### Search Heuristics

Once the heuristic administrator has created a concrete representation of the scheduling problem, the search heuristics now will play a pivotal role. The focus here shifts from analyzing these representations to identifying functional and insightful elements within the exam scheduling challenge. Key considerations at this stage, involve determining relationships and patterns among various elements of the problem representation with the goal of guiding the heuristic administrator toward potential solutions. Some questions will guide the Search Heuristics stage such as what relationships and patterns among elements of the exam schedule are evident in this representation? How do specific constraints impact the overall scheduling structure? Do any of the relationships or patterns within the representation seem to point to a potential solution? Can specific configurations within the schedule be recognized as optimal solutions based on identified relationships?

#### Termination Heuristics

After the heuristic administrator has created a concrete representation of the scheduling problem and applied search heuristics in analyzing and identifying potential solutions, the termination heuristics become now relevant as it guide the administrator in concluding the scheduling activity. Some of the key considerations at this stage are first, what we call clear understanding of solutions. By this key consideration, the heuristic administrator will check if there has been a clear-cut solution that satisfactorily addresses the identified challenges in the exam scheduling system. He will also check if the process has reached a point where further exploration may not significantly enhance the clarity of solutions. The second key consideration is what we call sufficiency criteria. Here, the heuristic administrator will check if the process has found a potential solution that meets the identified criterion.

#### Implementation Heuristics

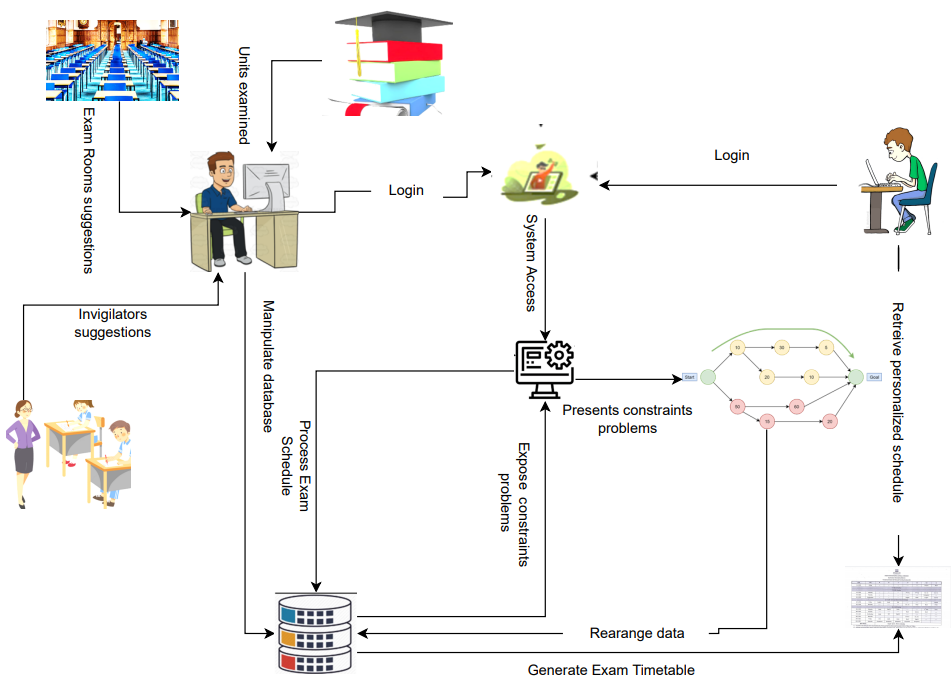
Following the identification of a potential solution through clear representation and insightful analysis using search and termination heuristics, at this point the implementation heuristics phase becomes crucial. This phase guides the heuristic administrator in evaluating the success of the implemented solution, addressing complexities associated with the exam scheduling system. The first key consideration here will be the success assessment. By the success assessment, the heuristic administrator will evaluate whether the implemented solution successfully resolved the identified challenges in the exam scheduling system. Given the complexity of scheduling, he will determine a simplified "yes" or "no" answer to the question of implementation success. The second key consideration will be what is known as variables and evaluation. This include factors such as room availability, adherence to constraints, and overall user satisfaction. The third and final key consideration will be the review and reapplication. In the case of implementation failure, the heuristic administrator guided through a review of the problem-solving model will be encouraged to reapply the various heuristics within this model. This iterative process aims to enhance the overall effectiveness of the exam scheduling solution.

## **Conceptual Framework**

## **General Understanding of the Conceptual Framework**

In a general context, a conceptual framework of a system constitutes a sophisticated, abstract portrayal including the fundamental constituents, interrelationships, and principles defining the architecture and functionality of the system. It operates as a strategic guide or navigational chart for comprehending and configuring the system, offering a systematic and structured approach to grasping its operational dynamics. It is in this line that Kawussi indicates that a conceptual framework materializes in the form of illustrative diagrams or figures, elucidating a myriad of variables that interact harmoniously in the pursuit of research objectives or in the formulation of innovative solutions to address identified research predicaments (Mawussi Kossivi Soviadan, 2019).

Within this foundation, it is imperious to construct a conceptual framework for the Automated Exam Scheduling System, particularly within the context of Strathmore University. Such a framework would provide a graphical representation defining the complex interplay of factors, variables, and constraints inherent to the system, fostering a holistic understanding of its functionality and its alignment with the institution's academic objectives.



## **Explanation of the Conceptual Framework**

In the above conceptual Framework analysis, the system administrator initiates a login process via the user interface. Upon successful login, he gains the capacity to interact with the database, enabling actions such as defining university preferences. Subsequently, he issues commands to the system, instructing it to initiate the examination scheduling process predicated on the database's data. As the system processes the exam schedule, the database uncovers the presence of constraint-related challenges that necessitate resolution before the timetable becomes accessible to students.

Upon encountering constraint-related issues, the system invokes the Heuristic Algorithm, a robust problem-solving tool aiming at finding the best solution in the shortest possible time (Maad M. Mijwel, May 15). This heuristic algorithm conducts a comprehensive evaluation, scrutinizing all conflicts and other constraint-related challenges. It proceeds to restructure the data stored within the database to harmonize with the solution. After the data has been successfully reorganized, the system proceeds to generate the exam timetable. Subsequently, the end-user, typically a student, can access his personalized examination timetable.

## **Conclusion**

In conclusion, Chapter 2, covering the Literature Review, extensively explored the complex landscape of exam timetable systems within various academic institutions. It identified a multitude of challenges, innovative solutions, and the determined efforts undertaken to surmount these obstacles. The systems investigated, including the Automated University Timetable System, Automated Examination Timetable Generation and Invigilator Scheduling System, and Re-Engineering of Examination Timetable Generation and Invigilation Scheduling System, all aimed to address the pivotal issue of optimizing the exam scheduling, refining resource allocation, and enhancing user interfaces by leveraging advanced technologies and methodologies. Nevertheless, each system encountered its specific set of limitations. Challenges such as accommodating retake students, ensuring data consistency and integration, and the lack of personalized timetables for individual students were predominant themes across these systems.

# Chapter 3: Methodology

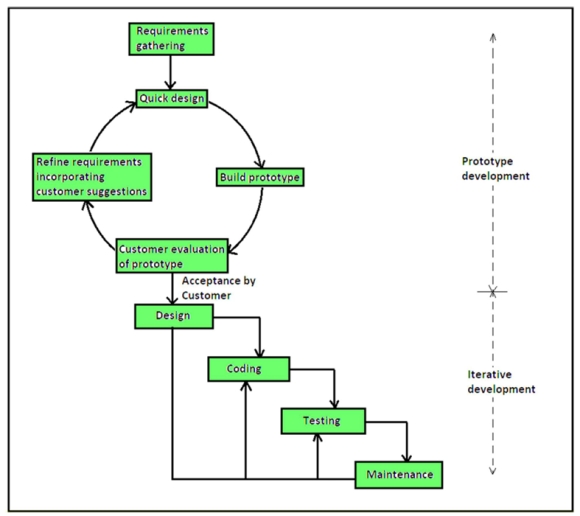
## **Introduction**

This chapter, offers a comprehensive roadmap for developing the Automated Exam Scheduling System, emphasizing adaptability, user-centric design, structured analysis, and efficient tool utilization to meet the diverse needs of the academic institution, especially at Strathmore University. The chapter will be shaped upon various sections. The first section will describe the prototype as the proposed development approach, which allows for flexibility in accommodating changes during system development and emphasizing the iterative nature, with a focus on user involvement for requirements gathering. The second section, will outline some key reasons that drove the developer's choice towards the prototype development approach. The third section will focus on the Object-Oriented Analysis and Design (OOAD) approach as the methodology for the system design by highlighting the identification of objects, their attributes, and methods, promoting a clear understanding of the proposed system's functionality. In the fourth section, the chapter will introduce vital components of the system design, including Class Diagrams, Entity-Relationship Diagrams (ERD), and Database Diagrams. Finally, the last section will delve into the System Development Tools and Techniques. This section will encompass a range of CASE tools and languages.

## **Software Development Methodology: Prototype approach**

In the proposed system, the prototype development approach is taking center stage. This approach, rooted in its flexibility, provides the ideal framework to accommodate changes and adapt to evolving requirements throughout the software development process. At this juncture, it is paramount to have a general view of this prototype approach coming out as a suitable development approach for the proposed system.

## **Understanding of the prototype development approach**

The prototype methodology appears suitable for the development of this system because it helps designers generate and share new ideas, and get feedback from users or customers as the development of the system progresses (Michel Beaudouin-Lafon, Wendy E. Mackay, 2012). Some requirements could be adjusted or changed entirely. As underlined by some writers, the prototype process is a model that is conducted in several stages, which can run to the realization of the final (Nacheva, 2017). This development approach can be created in the early stages of a software project, and immediately before finalizing it. Stages here, can be repeated as long as necessary and as required, which means that the process is iterative (Todd Zaki Warfel, 2009). In using this approach, the users of the proposed system will be given chances to participate in the development process. This participation of the system users in the development process will provide them a chance to meet their requirements and satisfaction. According to Chang and Taylor (2016), Customer participation is always valuable in enhancing NPD performance. 

## Figure 2.16: Prototype representation **(**[prototype approach - Bing images](https://www.bing.com/images/search?view=detailV2&ccid=NSN24t49&id=7B7E0CF23306E34A2EB8512CDB4510820A5C7CF4&thid=OIP.NSN24t49f7AwQkZDlASfPQHaGk&mediaurl=https%3a%2f%2fcdncontribute.geeksforgeeks.org%2fwp-content%2fuploads%2fPrototyping-model.png&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.352376e2de3d7fb03042464394049f3d%3frik%3d9HxcCoIQRdssUQ%26pid%3dImgRaw%26r%3d0&exph=785&expw=885&q=prototype+approach&simid=608015984982910962&FORM=IRPRST&ck=1CB5311ADA54E452AADDF2DB7EAEFDDD&selectedIndex=9&ajaxhist=0&ajaxserp=0))

## **Stages of the Prototype Development Process**

The prototyping process develops in multiple distinct stages, offering a potential pathway to achieving the final product. These stages allow for the creation of prototypes, which can materialize either at the beginning of a software project or during its concluding phases, providing flexibility throughout the development lifecycle. When embracing this approach, it becomes essential to conscientiously follow a well-defined sequence of key stages.

## **Requirements Gathering and Analysis**

In the initial phase of the prototype methodology, significant time is devoted to engaging with system users, primarily comprising the examination office of Strathmore University as well as the students. Through active listening and constructive dialogue, the system designer will diligently query users about their expectations for the system. Understanding all underscored system requirements will enable the system designer to come up with a system that genuinely aligns with intended customer satisfaction.

## **Quick Design**

Following the collection of system requirements, the system designer will proceed to undertake a preliminary system that will not constitute the final design. The primary aim of this rushed design phase will be to furnish users with an initial overview of the system and lay the foundation for the initial prototype.

## **Prototype Construction**

In the third stage, the system designer will attempt to construct a prototype system aimed at embodying the insights derived from the preliminary design phase. It is important to note that the preliminary design will represent an incomplete design, and thus, the resultant prototype will just be a basic one that will carry a simplified representation of the envisioned system.

## **Initial User Evaluation**

Upon the completion of the prototype system, this will be presented to the exam office and the administrators of different departments of Strathmore University for preliminary testing. During this stage, system users will assess the desired system and scrutinize its performance in terms of strengths and weaknesses. As the system assessment is being done, the system designer will be gathering all users' valuable feedback and suggestions to have an important basis for subsequent refinements and integrations.

## **Refining the Prototype**

In case users’ feedback points to a certain dissatisfaction, the system designer will be inclined to refine the prototype model in response to customer feedback and suggestions. This iterative process will persist until users express satisfaction with the presented model.

## **Implementation of the Product and Maintenance**

Once contentment with the model is affirmed, the system designer, in collaboration with the users, will then proceed with the implementation and comprehensive testing of the final system before deploying it for practical use.

## **Justification of Prototype approach**

The decision to opt for the prototype approach in developing the Automated Exam Scheduling System for Strathmore University is grounded in several compelling reasons. Firstly, its flexibility and adaptability is paramount for a dynamic academic institution like Strathmore. Given the ever-evolving nature of academic schedules, the prototype methodology will offer the agility needed to swiftly accommodate changes, ensuring that the timetable remains current and efficient. Furthermore, the prototype approach will place a strong emphasis on user involvement and feedback. In the context of a complex academic environment, where the satisfaction of students and faculty is of utmost importance, this feature will be precious. It will ensure that the system is tailored to the specific requirements and preferences of the end-users, fostering a user-centric design that will enhance user experience.

Effective communication will be facilitated through tangible prototypes, making it easier for non-technical stakeholders to visualize and understand the proposed solution. Ultimately, this approach will center on customer satisfaction, aligning with research that indicates customer participation (Chang & Taylor, 2016). For Strathmore University, this means that the system will be exceptionally tuned to meet the unique scheduling needs of its students and staff. In addition to its adaptability and user-centric focus, the prototype methodology will provide the advantage of risk mitigation. Allowing for the early detection and rectification of potential issues during the development cycle, will help avert more significant problems down the road. This proactive risk management approach will contribute to the overall stability and reliability of the Automated Exam Scheduling System, making it a judicious choice for a critical project of this nature.

## **System Analysis: OOAD Analysis and Designing Approach**

In the context of the automated exam scheduling system: the case of Strathmore University, the Object-Oriented Analysis and Design (OOAD) approach is being proposed as the primary methodology for the system design. According to Parupathy and Bhavani, the logic in the object-oriented programming language is that, based upon the difficulty and functionality of the program, the program is always divided into multiple sub-divisions, and each sub-division, performs the specified functions. All these functions are grouped under a single object named Classes (Pasupathy & Bhavani, 2013). Based on this description, the OOAD appears well-suited for the system being proposed. OOAD presents a capacity to offer a structured and efficient way to address evolving requirements and complexities. In applying this approach, it will be easy to model the system being proposed as a collection of objects including their interactions. The series of stages involved in the OOAD will be essential for understanding and designing the proposed system effectively. The process will include the identification of objects, the definition of their attributes and methods, and the modeling of their interactions. This is how the OOAD will end up promoting a clear and organized way of thinking about how the system will function, which is crucial for the complex task of all exam scheduling systems.

As Pasupathy and Bhavani underline in their article, the OOAD methodology provides an efficient and well-suited mechanism to measure the quality of the software (Pasupathy & Bhavani, 2013). By proposing this methodology, it’s estimable that the proposed Automated Exam Scheduling System will meet the diverse needs of the university while maintaining a structured and efficient development process. This approach will promote adaptability and enhance customer satisfaction, ultimately leading to an effective solution for academic scheduling at Strathmore University.

## **Use Case Diagram**

The Use Case Diagram will be a crucial component of the project being proposed. This diagram, as stipulated in the Internal Journal of Intelligent Engineering and Systems, is a behavior diagram in the Unified Modeling Language (UML) and describes the functional requirements of software (Institut Teknologi Sepuluh Nopember et al., 2021). In this logic, considering the context of the proposed project, the Use Case Diagram will play a pivotal role in defining the system's functionality from the user's perspective. It will provide a visual representation of the various interactions and relationships between the system and its users, which in this context will include administrative staff, students, and other stakeholders. This Use Case Diagram will help identify and document the specific actions that users can perform within the system, outlining the system's expected behavior. This will entail defining use cases such as "create exam schedule", "access personalized timetables" etc.

## **Sequence Diagram**

A sequence diagram, as mentioned by some scientists, shows a set of interacting objects and the sequence of messages exchanged among them. In it, the emphasis is put on the time ordering of the messages (Swain et al., 2010). The importance of this diagram in software engineering makes it another precious component of the Automated Exam Timetable System being proposed for Strathmore University. This diagram will play a vital role in illustrating the interactions and dynamic behavior of the proposed system and all its various components. In the context of this project, the sequence diagram will show how different elements, such as the user interface, scheduling algorithm, and database, collaborate to generate exam timetables. This Sequence Diagram will offer a step-by-step depiction of system interactions, providing a clear visual representation of how the system will process and handle user requests.

Being instrumental in the design and development phases, this diagram will drive the understanding of the flow of major activities within the system. For instance, it will show how a user submits a request to generate an exam timetable, how the system processes this request, communicates with the database to gather relevant data, and ultimately generates a personalized timetable for the student. Sequence diagrams will help ensure that the system functions smoothly and efficiently, mapping out the various components and their interactions.

## **System Design**

Truly speaking, a system design is always a critical phase in the software development life cycle. Here, the architecture and structure of a software system are planned and documented. It must gather all mentioned requirements and specifications in previous sections into a detailed blueprint for the actual implementation of the software. For this to be implemented, the system design will cover components like class diagram, Entity Relationship Diagram, and Database Diagram.

## **Class Diagram**

Class Diagrams are always referred to as UML class diagrams. They are used to describe the static view of an application with the main constituents: classes and their relationships. They are like a description of a concept and may have attributes and operations associated with them (Purchase et al., 2001). In this project, the class diagrams will help define the core entities and their properties, such as students, courses, exam schedules, venues, etc. By visualizing these components and their interconnections, the system's architecture will become more comprehensible. For instance, with this Class Diagram, it will be easy to depict relationships between classes, like how a student class will be associated with specific courses, and how a timetabled class will connect exam slots to venues. Such clarity in system structure will ensure a more organized development process and help in creating a stable and maintainable exam scheduling system.

## **E.R. Diagram**

The realm of software engineering witnesses the wide use of Entity Relation Diagrams in structured analysis and conceptual modeling. This ERD approach is easy to understand, powerful to model real-world problems, and readily translated into a database schema. As underlined in several articles and conferences, the ERD views that the real world consists of a collection of business entities, the relationships between them, and the attributes used to describe them (Song et al., 1995). Considering all this information, it's worth to start noticing that the Entity-Relationship Diagram (ERD) will considerably be needed in the proposed project as it focuses on the relationships between various data entities, providing a clear visualization of how different data elements are interconnected.

In this project, the ERD will help define the database structure, highlighting entities like students, courses, exam schedules, venues, and their relationships. ERD is crucial for data modeling, which is foundational for this system's functioning. It will ensure that data is stored, retrieved, and manipulated accurately, reflecting real-world relationships and constraints. For example, this diagram will represent how a student entity relates to specific courses, how courses will be scheduled in the exam timetable, and how venues will be associated with exam slots. By mapping these relationships and dependencies, ERD will support the effective database design for the system, which is essential for data integrity, consistency, and query performance.

In the context of the Automated Exam Timetable System: the case of Strathmore University, the ERD will be indispensable for managing complex data relationships and ensuring that the information needed for scheduling and managing exams is accurately and efficiently stored and retrieved.

## **Database Diagram**

The Database Diagram, often referred to as the DB Diagram, will also be one of the essential components of this project. This diagram visually represents the structure and organization of the database, showcasing tables, relationships, and key elements involved in the system. In the proposed project, the DB Diagram will serve as a blueprint for the database design. It will help define the database schema, outlining various tables such as student information, course details, exam schedules, venues, etc. This visual representation will ensure that the database is well-structured, normalized, and efficient for data storage and retrieval.

For example, the DB Diagram will illustrate how student information will connect to their enrolled courses and exam schedules, how exam venues will be linked to specific exam slots, and how administrators will manage various data elements. These relationships will be crucial for the functioning of the system, ensuring that data is stored and manipulated accurately.

## **System Development Tools and Techniques**

When discussing tools and techniques for system development, the term CASE (Computer-Aided Software Engineering) emerges, signifying any form of automated support for software engineering. As defined by some writers, Computer-aided design (CAD) is a technology, the task of which is to facilitate the creation, change, analysis, and optimization of projects at the expense of automation of works on design stages and preparations of production (Kolbasin & Husu, 2018).

In the context of the automated exam scheduling system, the application of specific CASE tools will be instrumental in various stages of development. From designing user interfaces and backend logic to conducting thorough testing and documentation, these tools will enhance the development process and help ensure that the system meets the needs of Strathmore University's examination office. By coupling the power of CASE tools tailored for web-based systems, the system designer will stay at the forefront and deliver a user-friendly, and efficient automated exam scheduling solution.

## **Tables 3.1: Tools and techniques**

|  |  |  |
| --- | --- | --- |
| No | Tool | Description |
| 1 | Bootstrap | Bootstrap serves as a vital tool for crafting responsive and user-friendly web interfaces. Within the proposed automated exam timetable system, this tool will streamline the creation of interfaces that adapt seamlessly to various devices, ensuring accessibility for both students and administrators. With Bootstrap's grid system, structuring the proposed web page, layout will become straightforward, facilitating efficient content presentation and interaction. Utilizing Bootstrap components like navigation bars and forms will simplify the implementation of essential features, such as user preferences and navigation within the system.  By integrating Bootstrap, the hope is to accelerate system development while maintaining a consistent and polished appearance, enhancing the user experience. Whether accessed from desktops or mobile devices, Bootstrap will ensure that the system remains accessible and professional. Its pre-designed templates and components will save development time and provide a responsive and appealing design that aligns with user expectations. |
| 2 | CSS | CSS is the tool that adds aesthetics and styling to the system. In the proposed system, CSS will work hand in hand with HTML and its framework Bootstrap to create visually appealing web pages. It will allow the system designer to control the layout, fonts, colors, and overall design of the user interface. By using CSS, the system designer will ensure a consistent and attractive appearance throughout the system, enhancing the user experience. |
| 3 | Draw.io | Draw.io as a diagramming tool, is very indispensable for the creation of visual representations of this proposed automated exam timetable system. With Draw.io, it will be easy to design diagrams that depict the system's architecture, relationships between components, data flows, and user interfaces. These visual models will serve as essential blueprints for the development, helping the developer and his supervisor to understand the system's structure and functionality. |
| 4 | Git | Git, as a distributed version control system, is a powerful tool for managing source code. With Git, the system designer will not be attached to work on one machine as he progresses with the system development. Working from any location, the system designer will be able to synchronize changes through commits. |
| 5 | GitHub | GitHub extends the capabilities of Git by offering a centralized repository to host the code, making it accessible to the public. Moreover, the use of Github will enhance frequent discussions with the supervisor that are related to this automated exam timetable system. GitHub's pull request feature will facilitate code review and integration, ensuring that the code quality remains high. |
| 6 | HTML | HTML forms the backbone of web-based systems, and it will play a pivotal role in this proposed automated exam timetable system. It will be utilized to structure web pages, enabling them to display content, forms, and user interactions. This tool will ensure that information is presented in a consistent and accessible manner, making it the foundation for user interfaces where students can retrieve personalized schedules so administrators can interact with the system. |
| 7 | JavaScript | JavaScript will be essential for interactivity within the proposed automated exam timetable system. With JavaScript, it will be easy to implement dynamic features like input validation, and user interactions. For example, in the case of an interactive calendar for scheduling or handling user preferences aligning with the University policies, JavaScript will be of much use. |
| 8 | MySQL | MySQL is a relational database management system that will serve as the backend storage for the proposed automated exam timetable system. This will be crucial for storing and managing data related to students, courses, exam schedules, venues, and other essential information. MySQL's ability to handle complex queries and large datasets will ensure the efficient retrieval and storage of data for the proposed system. It will provide a secure and organized repository for all the data the system will need to function effectively. Its compatibility with PHP will enable seamless communication between the web-based interface and the database, allowing users to interact with the system retrieving personalized schedules, and maintaining data consistency. |
| **9** | PHP | PHP is a server-side scripting language. This will be crucial for the development of the proposed automated exam timetable system. It will enable dynamic and interactive web pages, making it ideal for creating a user-friendly interface and handling various functionalities. In the context of this project, PHP will play a vital role in processing data, generating exam timetables, and interacting with the database to provide a smooth user experience. PHP will help in creating a web-based platform that allows users to interact with the system, retrieve personalized exam schedules, and communicate with the system. Its ability to integrate with databases and handle complex logic will be pivotal in ensuring the efficient operation of the system. |
| 10 | VCS | In the context of the proposed exam timetable system, VCS will serve as the foundational tool for tracking and managing changes to the source code. This will be crucial for maintaining a historical record of changes, which can be precious when debugging, ensuring code quality, and reviewing past development decisions. Additionally, a VCS will offer a safety net against accidental data loss. |
| 11 | VSC | Visual Studio Code (VSC) offers a comprehensive suite of features for coding, debugging, and testing web applications. This tool will serve as our primary platform for writing and managing HTML code.  Another additional value of VSC is that it provides a powerful capacity to streamline code management by facilitating the integration with GitHub, allowing for efficient code version control and collaboration. This feature will significantly enhance its utility in the proposed web-based system development project. |

## **Conclusion**

In conclusion, the methodology outlined in this chapter, which encompasses a hybrid approach featuring prototype development, Object-Oriented Analysis and Design (OOAD), and an assortment of innovative development tools and techniques, was presented offering a multifaceted and comprehensive strategy for the successful development of the Automated Exam Scheduling System being proposed for Strathmore University. This approach that does not only guarantee adaptability throughout the development process has also been presented placing a paramount importance on user involvement and satisfaction. Furthermore, it has been revealed that the integration of a prototype development approach aligning with the dynamic nature of academic schedules, will enable the system to accommodate changes swiftly and efficiently.

**Chapter 4. System Analysis and design**

**4.1. Introduction**

This chapter delves into the implementation of system requirement analysis and system design, encompassing all necessary diagrams and their relevance to the development of a web-based system for scheduling the exam timetable at Strathmore University.

## 4.2. Software Requirements Analysis

### 4.2.1. Functional Requirements

Functional requirements serve as a detailed roadmap outlining the client's specifications for an application. They encapsulate the project objectives, anticipated benefits, potential constraints, and the project's scope. Moreover, they encompass user needs and portray the activities users can undertake within the system. Given that users often lack a comprehensive understanding of their application requirements, the Functional Requirement Documentation serves as a vital tool in elucidating these complexities. By offering detailed insights into the application's functionality, usage, and interface, it effectively bridges the gap between user expectations and system capabilities. Below, we picture the primary expectations of potential users of the scheduling exam timetable being developed.

1. The system will allow the school admin to create system users such as the faculty admin, exam officers and students, lecturers or invigilators.
2. In the creation process, the school admin will assign to each user basic credentials as well as roles depending on services each user is having access to.
3. Once the user is created, the system will allow him to login and start accessing services
4. Once the user is logged in, the system will provide him some facilities of editing his profile and resetting his password for security purposes
5. In case the user forgets his password, the system will provide him the facility to change his password and log in using the new password
6. The system will allow each faculty admin to set a school tentative exam timetable without caring for collisions.
7. After all tentative exam timetables are set from different faculties, the system will then allow the exam officer to merge all these various timetables into one.
8. Once faculty tentative exam timetables are merged, the system will allow the exam officer to identify collisions in the overall school exam timetable.
9. Once identified, the system will allow the exam officer to make some adjustments to repeat exams schedule that are conflicting with normal exams before executing the conflict solution to the timetable.
10. Once conflicts are solved, the system will allow the user to retrieve only his timetable using his code.

### 4.2.2. Non-functional Requirements

#### Non-functional requirements include restrictions imposed on the product being developed, as well as the development process itself, specifying external constraints that must be adhered to. In software system engineering, these requirements pertain to aspects beyond the functionality of the software, depicting how the software accomplishes its tasks. Examples include software performance, capacity, and constraints.

#### 4.2.2. 1. Capacity

1. The system shall be able to handle a large volume of exam schedules simultaneously.
2. The system shall support concurrent access by multiple users, including students, lecturers, and administrative staff.
3. The system shall be scalable to accommodate potential future growth in the number of enrolled students and exam schedules.

**4.2.2.2. Performance**

1. The system shall promptly notify users of any errors in the input provided, presenting warning messages to guide users in rectifying the issue.
2. A unified login module will serve all system users, including administrators. However, user roles assigned by the system admin will govern access to specific services.
3. The system shall strive to minimize response times to user queries, ensuring swift access to information and functionalities.

**4.2.2.3. Security**

i. User passwords shall be securely encrypted within the database to enhance data protection.

ii. Administrative privileges, granting access to all system services, shall be exclusive to the admin role.

iii. Users are prohibited from self-assigning roles; only the admin possesses the authority to assign roles to users.

## 4.3. System Design

System design is the determination of the overall system architecture—consisting of a set of physical processing components, hardware, software, people, and the communication among them—that will satisfy the system’s essential requirements (Keller, 1992). In this section, encompasses some diagrams related to the methodology used.

**4.3.1 Use Case Diagram**

According to Radosław Klime and Piotr Szwed (2010), "Use cases form an integral part of UML, providing a cohesive narrative about the system's behavior". In this context, the diagram serves as documentation for the system's functional requirements.

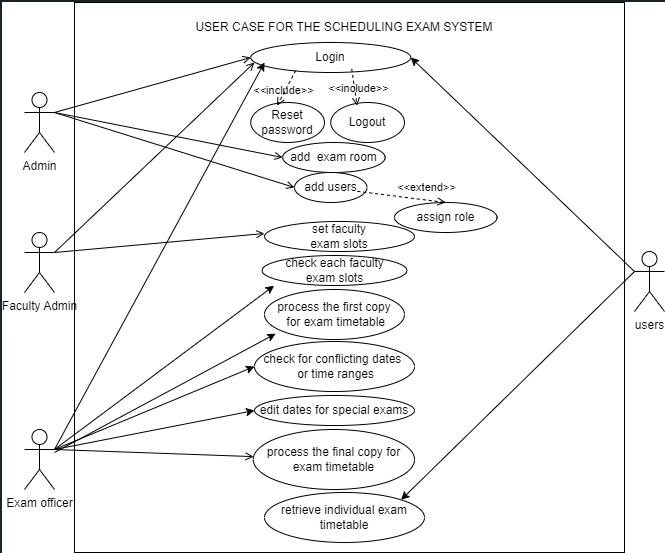


Figure 4.1: Use Case Diagram

In figure 4.1, the use case for the scheduling exam timetable designed for Strathmore University, can been seen three primary actors and one secondary actor that includes 3 images. Primary actors are the school admin, the faculty admin and the exam officer. Secondary actors are users mostly students, lecturers and invigilators. The school admin is the one who manipulates data in the system and assigns credentials including roles to all users. The primary job for the faculty admin is to set the tentative exam timetable for the faculty he is based in. The exam officer merges all tentative exam timetables from different faculties and processes one general timetable for the whole University. The secondary actor as mentioned above is the user: a user can be a student, a lecturer or an invigilator. All these users have the same functionality which is of viewing personal timetable in relation to their different tasks.

## **4.2. Sequence Diagram**

This sequence diagram is describing how and in which order objects are interacting with each other. The sequence starts with the admin who must log in first. In case he logs in using wrong credentials, the system notifies him and requests him to try again. Once the admin is logged in, he can post the data and do more operations to the system. He can create a user and assign him basic credentials to be able to log into the system too. Once the user logs into the system he starts accessing services in relation to his status within the school.

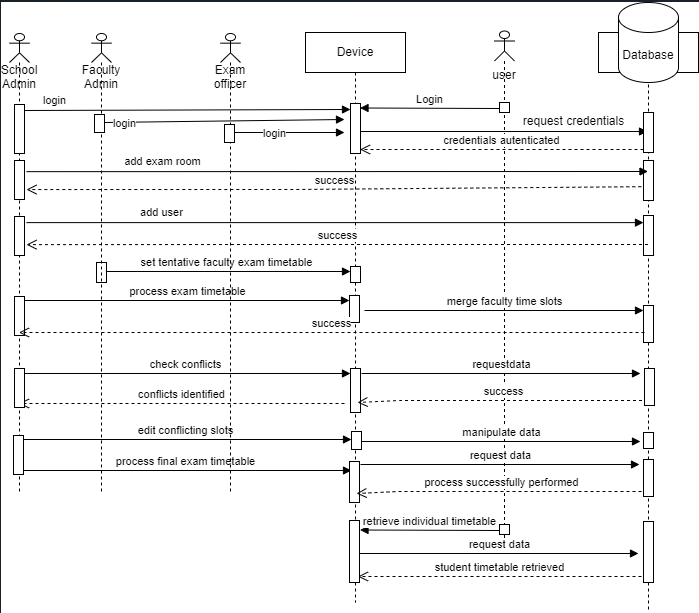


Figure 4.2: Sequence diagram

4.3. Class Diagram

According to Jonas Boustedt (2010) class diagrams are visualization shown as boxes with instance variables and methods connected to each other. The scheduling exam timetable system developed contains 9 classes among which the class Users is a parent class to classes: SchoolAdmin, FacultyAdmin, ExamOfficer, Lecturers/Invigilators and Students. These child classes inherit common attributes from their class Parent Users. Showing how the different classes relate to each other, simple association relationships have been in use.

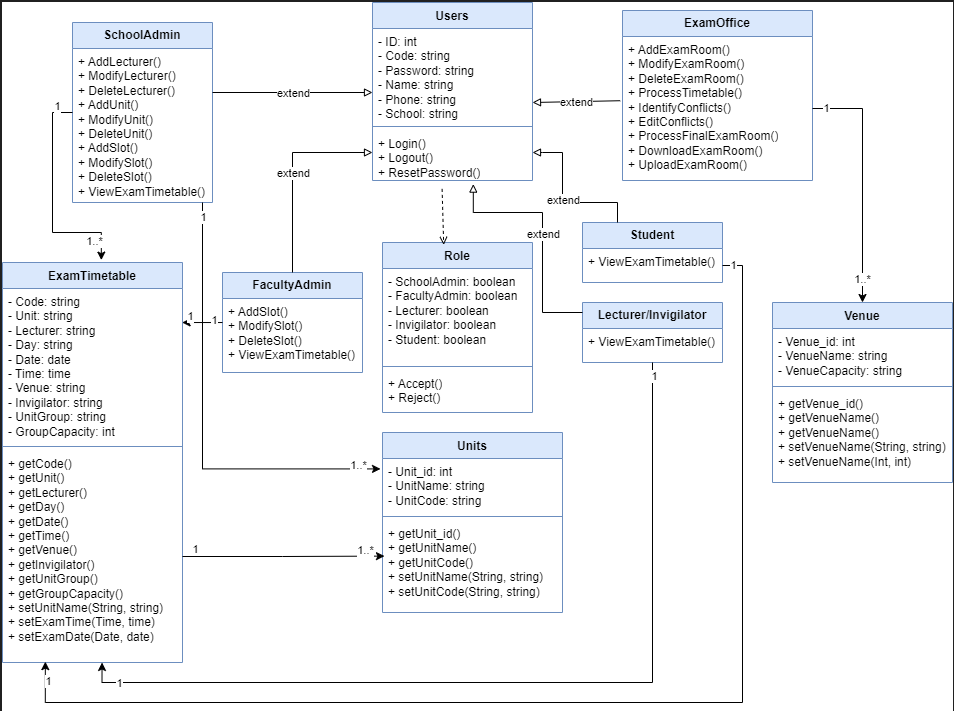


Figure:4.3: Class diagram

### 4.4. Entity Relationship Diagram (ERD)

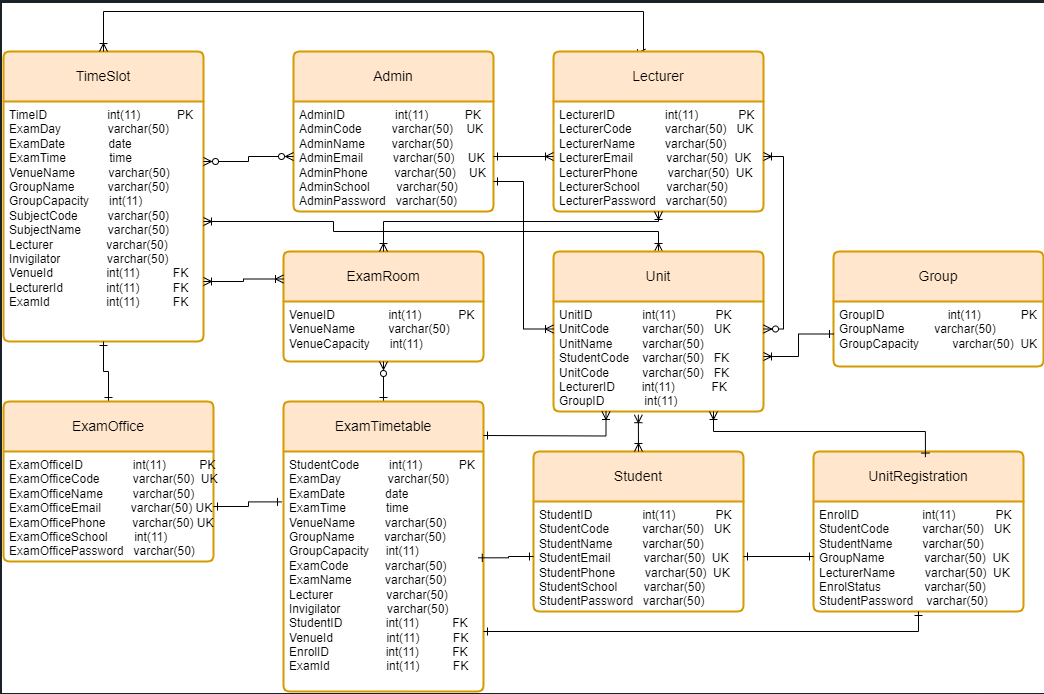
The Entity-Relationship (ER) Diagram serves as a crucial tool for analysts to visualize the data architecture intended for storage in a database system (Sikha Bangui, 2012). By employing an ERD, developers can ensure the creation of a strong database design that facilitates efficient database management and maintenance.

The ERD for this system encompasses 10 key entities and illustrates their interrelationships. Across various entities, some relationships indicate that one school admin can assign credentials to one or more users. Furthermore, one and only one user can access one and only one request based on his code. Additionally, solely the exam officer has the capability to process and identify conflicts in the school exam timetable and process the corrected copy.

## 

4.5. Database Diagram

A database diagram is the skeleton structure that represents the logical view of the entire database. This is an integrated collection of related files, along with details of the interpretation of the data contained therein. Basically, database system is nothing more than a computer-based record keeping system i.e. a system whose overall purpose is to record and maintain information/data (Gunjal, 2003).



# Chapter 5. System Implementation and Testing

## 5.1. Introduction

## This chapter delves deep into the implementation description, functional testing, and non-functional testing, providing comprehensive insights into the outcomes derived from various tests. Through the inclusion of screenshots, the functioning of the system is accurately illustrated, offering a detailed understanding of its operations.

## 5.2. The Implementation Environment

### 5.2.1. Hardware Specification

In the development of this web-based system for scheduling the exam timetable at Strathmore University, a personal computer with the following specifications has been utilized: Memory - 16 GB, Storage - 1 TB, Processor - 1.3 GHz Intel Core i7.

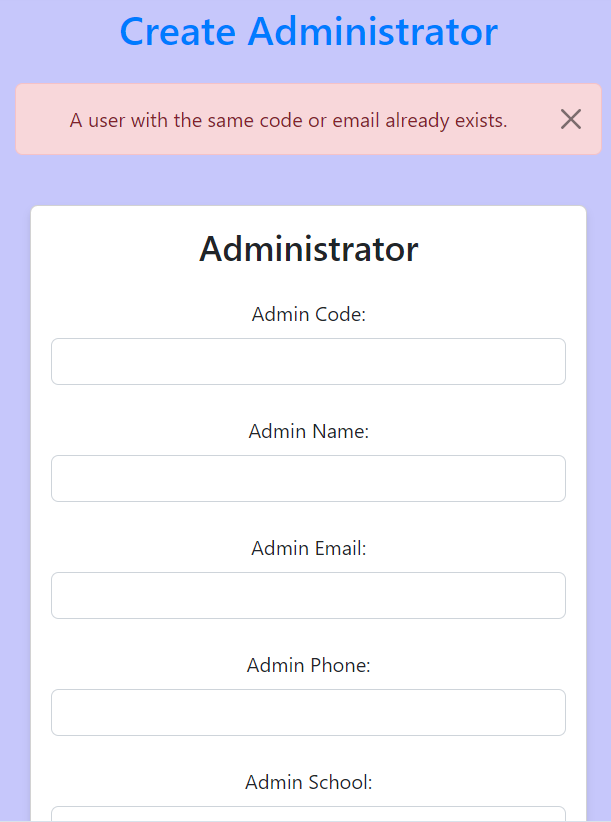
### 5.2.2 Software Specification

## The development of the web-based system for scheduling the exam timetable at Strathmore University was conducted on the Windows 10 operating system. The server setup utilized the XAMPP version, incorporating PHPMyAdmin, Apache, and MySQL as the relational database management system. Development tasks were executed using HTML, CSS, JavaScript, and PHP as the primary markup and scripting languages. Visual Studio Code was chosen as the primary code editor, while GitHub was utilized for version control and development workflows.

## 5.3. System Implementation: User Interface

### 5.3.1. Customer Module

#### 5.3.1.1. Admin Module



## **Deliverables**

In any Information Systems project, deliverables always refer to tangible or intangible outcomes that the project aims to deliver during the different phases of the project lifecycle. These outcomes serve as key milestones and markers of progress. With this understanding, it’s worth mentioning that deliverables for the proposed system, will essentially help validate this project's progress and quality, ensuring alignment with the supervisor's expectations, and ultimately fulfilling the objectives and needs that have been defined.

## **Table 3.2: Deliverables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Activity | Start time | End time | Duration |
| 1 | Concept Notes | 09/15/23 | 09/20/23 | 5 |
| 2 | Chapter 1: Introduction | 09/21/23 | 09/30/23 | 9 |
| 3 | Chapter 2: Literature Review | 10/02/23 | 10/29/23 | 27 |
| 4 | Chapter 3: Methodogoy | 11/02/23 | 11/18/23 | 16 |
| 5 | Proposal Combined | 11/20/23 | 11/28/23 | 8 |
| 7 | Edited Proposal copy | 12/02/23 | 12/03/23 | 1 |
|  | Chapter 4: S. Analysis and Design |  |  |  |
| 8 | Use Case Diagram | 01/8/24 | 01/13/24 | 5 |
| 9 | Sequence Diagram | 01/14/24 | 01/17/24 | 5 |
| 10 | Class Diagrams | 01/18/24 | 01/23/24 | 5 |
| 11 | E.R.Diagrams | 01/23/24 | 01/28/24 | 5 |
| 12 | Database Diagram | 01/28/24 | 01/31/24 | 4 |
| 13 | Chapter 5: System Implementation and Testing | 02/01/24 | 03/30/24 | 58 |
| 14 | Chapter 6: Conclusion and recommendations | 04/01/24 | 04/20/24 | 20 |

# Reference

Aminu, A., Caesarendra, W., Haruna, U. S., Sani, A., Sa’id, M., Pamungkas, D. S., Kurniawan, S. R., & Kurniawan, E. (2019). Design and Implementation of An Automatic Examination Timetable Generation and Invigilation Scheduling System Using Genetic Algorithm. *2019 2nd International Conference on Applied Engineering (ICAE)*, 1–5. https://doi.org/10.1109/ICAE47758.2019.9221700

Burke, E. K. (n.d.). *E. K. Burke, D. de Werra and J. H. Kingston , Applications to Timetabling,, In:J.Gross and J.Yellan(eds). The Handbook of Graph Theory, Chapman Hall/CRC Press, 2004, pp. 445-474*.

Chang, W., & Taylor, S. A. (2016). The Effectiveness of Customer Participation in New Product Development: A Meta-Analysis. *Journal of Marketing*, *80*(1), 47–64. https://doi.org/10.1509/jm.14.0057

Charles F. Abel. (2003). Heuristics and Problem Solving. *New Directions for Teaching and Learning*, *95*, 53–58.

Chi, M.T.H and Glaser. (1985). Problem Solving Ability. In *Human Abilities: An Information Processing Approach* (R. J. Steinberg). W.H. Freeman.

Growiec, J. (2022). AUTOMATION, PARTIAL AND FULL. *Macroeconomic Dynamics*, *26*(7), 1731–1755. https://doi.org/10.1017/S1365100521000031

Hussain, A., & Sharma, M. N. (2017). *An Improved Approach for Examination Timetabling Problem*. *7*(5).

Institut Teknologi Sepuluh Nopember, Fauzan, R., Siahaan, D., Institut Teknologi Sepuluh Nopember, Rochimah, S., Institut Teknologi Sepuluh Nopember, Triandini, E., & Institut Teknologi dan Bisnis STIKOM Bali. (2021). A Different Approach on Automated Use Case Diagram Semantic Assessment. *International Journal of Intelligent Engineering and Systems*, *14*(1), 496–505. https://doi.org/10.22266/ijies2021.0228.46

Kolbasin, A., & Husu, O. (2018). Computer-aided design and Computer-aided engineering. *MATEC Web of Conferences*, *170*, 01115. https://doi.org/10.1051/matecconf/201817001115

Maad M. Mijwel. (May 15). Heuristic Algorithms. *Computer Science, College of Science, University of Baghdad Baghdad, Iraq*.

Mawussi Kossivi Soviadan. (2019). *RESEARCH METHODOLOGY: CONCEPTUAL FRAMEWORK*. https://doi.org/10.13140/RG.2.2.17446.55365

Michel Beaudouin-Lafon, Wendy E. Mackay. (2012). Prototyping Tools and Techniques. In *Human computer Interation Handbook* (3rd Edition, p. 24). CRC Press.

Nacheva, R. (2017). *PROTOTYPING APPROACH IN USER INTERFACE DEVELOPMENT*.

Nayyar, A., Le, D.-N., & Nguyen, N. G. (Eds.). (2018). *Advances in Swarm Intelligence for Optimizing Problems in Computer Science* (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9780429445927

Oluwaniyi, N. O., Afeni, B. O., & Onyekwelu, B. A. (2016). *Development of an Automated University Examination Timetabling System*. *7*(1).

Orbell J. and Dawes R. M. (1991). A Cognitive Miser Theory of Cooperator Advantage. In *American Political Science Review* (pp. 515–528).

Pasupathy, S., & Bhavani, R. (2013). Measuring the Quality of Software through Analytical Design by OOAD Metrics. *International Journal of Computer Applications*, *63*(13), 39–44. https://doi.org/10.5120/10530-5514

Sachin Desale, Akhtar Rasool, Sushil Andhale, Priti Rane. (2015). Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey. *INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS*, *2*(5), 296–304.

Salem, D. (2023). University Examination Timetable Scheduling Using Constructive Heuristic Compared to Genetic Algorithm. *Fayoum University Journal of Engineering*, *6*(1), 39–46. https://doi.org/10.21608/fuje.2022.157195.1018

Song, I.-Y., Evans, M., & Park, E. K. (1995). *A Comparative Analysis of Entity-Relationship Diagrams*. *3*.

Swain, S. K., Mohapatra, D. P., & Mall, R. (2010). *Test Case Generation Based on Use case and Sequence Diagram*.

Todd, P. M. and Gigerenzer G. (2000). Precis of Simple Heuristics That Make Us Smart. In *Behavioral and Brain Sciences* (pp. 727–780).

Todd Zaki Warfel. (2009). *Prototyping: A Practitioner’s Guide*. Rosenfeld Media.

Vinod J. Kadam, Samir S. Yadav. (2016). Academic TImetable Scheduling: Revisited. *International Journal of Research In Science & Engineering*, 417–423.

Woumans. (2016). A column generation approach for solving the examination-timetabling problem. *European Journal of Operational Research*, *253*(1), 178–194.

Yu-Chi Ho. (1994). Heuristics, rules of thumb, and the 80/20 proposition. *IEEE Transactions on Automatic Control*, *39*(5), 1025–1027. https://doi.org/10.1109/9.284886

# Appendix

## Table A.1: Activity Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Activity | Start time | End time | Duration |
|  | **Proposal Crafting** |  |  |  |
| 2 | Concept Notes | 09/15/23 | 09/20/23 | 5 |
| 3 | Chapter 1: Introduction | 09/21/23 | 09/30/23 | 9 |
| 4 | Chapter 2: Literature Review | 10/02/23 | 10/29/23 | 27 |
| 5 | Chapter 3: Methodology | 11/02/23 | 11/18/23 | 16 |
| 6 | Proposal Combined | 11/20/23 | 11/28/23 | 8 |
| 7 | Proposal Defense | 11/28/23 | 11/29/23 | 1 |
| 8 | Proposal Submission | 12/02/23 | 12/03/23 | 1 |
|  | Chapter 4: S. Analysis and Design |  |  |  |
| 9 | Use Case Diagram | 01/8/24 | 01/13/24 | 5 |
| 10 | Sequence Diagram | 01/14/24 | 01/17/24 | 5 |
| 11 | Class Diagrams | 01/18/24 | 01/23/24 | 5 |
| 12 | E.R.Diagrams | 01/23/24 | 01/28/24 | 5 |
| 13 | Database Diagram | 01/28/24 | 01/31/24 | 4 |
|  | Chapter 5: S. Implementation and Testing |  |  |  |
| 14 | Coding | 02/01/24 | 03/30/24 | 58 |
| 15 | Class Presentation | 04/01/2024 | 04/15/2024 | 15 |
| 16 | Document writing | 04/01/2024 | 04/20/2024 | 20 |
| 17 | Final Presentation | 04/20/2024 | 04/22/2024 | 3 |
| 18 | Project Completion | 04/22/2024 | 04/25/2024 | 3 |

## 

## Figure A.1: Gantt Diagram

## A.2: Supervision sheet

|  |  |
| --- | --- |
| Student No | Student Name |
| 112721 | Kathembo Tsongo Dieudonne |

# Supervisor’s Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Comment | Next Meeting | Signed: Supervisor |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |