**DEVELOPMENT LECTURE SCHEDULING APPLICATION**

**BY**

**ALMUSTAPHA ADO FAROUQ**

**BU/22B/IT/6871**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING, FACULTY OF COMPUTING AND APPLIED SCIENCE, BAZE UNIVERSITY, ABUJA.**

**NOVEMBER, 2023**

**DECLARATION**

This is to certify that this Thesis entitled LECTURE SCHEDULING APPLICATION, which is submitted by ALMUSTAPHA ADO FAROUQ in partial fulfilment of the requirement for the award of degree for B.Sc. in Information Technology to the Department of Computer Science, Baze University Abuja, Nigeria, comprises of only my original work and due acknowledgement has been made in the text to all other materials used.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ALMUSTAPHA ADO FAROUQ Date

BU/22B/IT/6871

**APPROVED BY**

………………..

Dept. of Computer Science **H.O.D**

**CERTIFICATION**

This is to certify that this Thesis entitled LECTURE SCHEDULING APPLICATION, which is submitted by ALMUSTAPHA ADO FAROUQ in partial fulfilment of the requirement for the award of degree for B.Sc. in Information Technology to the Department of Computer Science, Baze University Abuja, Nigeria is a record of the candidate’s own work carried out by the candidate under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

**APPROVAL PAGE**

The project titled "LECTURE SCHEDLING APPLICATION" submitted by ALMUSTAPHA ADO FAROUQ bearing registration number BU/22B/IT/6871, has been approved by the examination committee for the award of the Bachelor of Science in Software Engineering degree at Baze University, Abuja.

By

Dr. Usman Bello Abubakar

Supervisor Date

Assoc Prof. Chandrashekhar Uppin

Head of Department Date

Prof. Peter Ogedebe

Dean, Faculty of Computing and Applied Science Date

Prof. Choji Davou Nyap

External Examiner Date

**DEDICATION**

This research project is dedicated to all those who have inspired and supported me throughout this academic journey.

To my family, whose unwavering love and encouragement have been my anchor in the storm of research challenges.

To my lecturers and mentors, whose guidance and expertise have shaped my understanding and fueled my curiosity.

To my friends, for their patience, understanding, and the occasional distraction that provided much-needed breaks.

And to the countless individuals whose contributions and insights, whether acknowledged or not, have enriched the foundation upon which this research stands.

May this work contribute in some small way to the collective pursuit of knowledge and the betterment of our shared understanding.

**ACKNOWLEDGEMENT**

I want to sincerely thank everyone who helped to see this initiative through to its successful conclusion.

My supervisor, Dr. Usman Abubakar, has my sincere gratitude for all of his advice, knowledge, and helpful assistance during this research. Their patience, support, and sage advice have been invaluable in determining the course and caliber of this work.

I am appreciative of the professors at Baze University's Department of Computer Science for sharing their expertise and creating a positive learning atmosphere.

I want to express my gratitude to my friends and family for their consistent encouragement, understanding, and support throughout this journey. Their confidence in me has always served as a source of inspiration.

Lastly, I would like to thank the reviewers who remained anonymous.

**ABSTRACT**

A state-of-the-art software program called the Lecture Scheduling Application was created to completely change how educational institutions organize their lecture schedules. Conventional scheduling methods frequently entail laborious manual coordination, which takes time and leads to disagreements in scheduling, waste of resources, and unhappiness among teachers and students. The goal of this program is to improve scheduling overall by offering a streamlined, automated, and user-friendly solution to these problems.

**CHAPTER ONE**

**INTRODUCTION**

**1.1 Overview**

The purpose of the lecture scheduling application is to make the process of organizing and managing lectures in educational institutions more efficient and user-friendly. With the use of this software, scheduling should be more effective and efficient, resulting in less manual labor and an overall better experience for teachers and students.

**1.2 Background and Motivation**

The creation of project scheduling apps has become apparent as a vital response to the problems brought on by the complexity of modern projects in the ever-changing field of project management. The necessity for effective task planning, organization, and management to guarantee maximum resource utilization and on-time project completion drives the development of these applications.

The intricacy of today's projects, with their multitude of tasks and complicated relationships, has made manual scheduling impractical. Projects get bigger and more complex, which increases the likelihood of mistakes and inefficiencies in scheduling. In addition, modern work environments are collaborative, which means that team members need to communicate and coordinate more efficiently.

To sum up, the need to solve the difficulties posed by complex projects is what drives the creation of project scheduling tools. These apps represent a significant progression in project management practices, supporting collaboration, maximizing resource usage, and guaranteeing timely project delivery—all of which are critical to the success and efficiency of projects in today's business environment. Project scheduling software are predicted to play an ever more significant role in the future of project management as technology develops.

**1.3 Statement of the Problem**

The current scheduling process in Baze University is time-consuming and error-prone, resulting in scheduling conflicts, suboptimal resource allocation, and dissatisfaction among stakeholders. This application aims to address these issues and provide a solution that ensures smooth lecture scheduling, minimizes conflicts, and maximizes resource utilization.

**1.4 Aim and Objectives**

The primary aim of the lecture scheduling application is to create a user-friendly, automated, and efficient system for scheduling lectures in educational institutions. The specific objectives include:

1. To develop a user-friendly interface for administrators, faculty, and students to access and manage the scheduling system.
2. To automate the scheduling process to minimize conflicts and optimize resource allocation.
3. To integrate features for requesting and approving schedule changes or swaps.
4. To Generate reports and analytics to help administrators make data-driven decisions about scheduling.

**1.5 Significance of the Project**

The lecture scheduling application is significant as it offers numerous benefits, including:

1. Improved efficiency and accuracy in scheduling, reducing conflicts and disruptions.
2. Enhanced user experience for students, faculty, and administrators.
3. Better utilization of resources, such as classrooms and faculty availability.
4. Time and cost savings due to reduced manual scheduling efforts.
5. Data-driven insights for better decision-making in scheduling.

**1.6** **Project Risks Assessment**

**Table 1.1 Project Risks Assessment**

|  |  |  |
| --- | --- | --- |
| Risk | Likelihood | Impact |
| Resource Constraints | Medium | High |
| Technology Failure | Low | High |
| Budget Overrun | Medium | High |
| Regulatory Changes | Medium | Medium |
| Data security and privacy concerns. | Low | High |
| Integration issues with existing systems. | Medium | Medium |

This table provides a starting point for assessing and managing risks specific to the development of a lecture scheduling application. It helps the project team to proactively identify and address potential challenges, ensuring a more successful project outcome.

**1.7 Scope and Organization**

The project's scope includes the development and implementation of the lecture scheduling application for a specific educational institution. The project will be organized into the following phases:

* Requirement analysis and system design.
* Application development and testing.
* User training and system deployment.
* Ongoing support and maintenance.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Introduction**

In order to provide an organized learning environment and ensure effective resource use, lecture scheduling is essential in educational institutions. The need for automated solutions to expedite the lecture scheduling process is expanding as technology develops. Applications for lecture scheduling are designed to solve problems with manual scheduling, like scheduling conflicts, resource optimization, and flexibility to changing academic settings. In order to provide an organized learning environment and ensure effective resource use, lecture scheduling is essential in educational institutions. The need for automated solutions to expedite the lecture scheduling process is expanding as technology develops. Applications for lecture scheduling are designed to solve problems with manual scheduling, like scheduling conflicts, resource optimization, and flexibility to changing academic settings.

**2.2 Historical Overview**

In this section, we provide a historical overview of the development of lecture scheduling applications, tracing the evolution of scheduling systems in educational institutions. Understanding the historical context helps identify the challenges faced in the past and draws lessons for the development of an improved lecture scheduling application for Baze University.

The automation of scheduling processes in educational institutions has been a topic of interest for many years. Manual scheduling methods, involving the use of paper-based systems and spreadsheets, posed numerous challenges such as time-consuming processes and increased likelihood of errors (Jones, 2009). As technology advanced, computerized scheduling systems began to emerge, offering more efficient and accurate scheduling capabilities.

One notable milestone in the history of lecture scheduling applications is the development of Timetable Management Systems (TMS). TMS solutions, such as Celcat and Syllabus Plus, gained popularity in universities for their ability to automate the scheduling of lectures, exams, and other academic activities (Jones, 2017). These systems provided features like conflict resolution, room allocation, and timetable optimization, improving the overall efficiency of scheduling processes.

With the advent of web-based technologies, scheduling applications started to shift towards online platforms. This allowed for easier access and collaboration among stakeholders involved in the scheduling process. For instance, the University of Edinburgh implemented an online scheduling system called MyTimetable, which enabled students to view and personalize their timetables through a web interface (University of Edinburgh, 2020). This shift towards web-based applications facilitated greater flexibility and real-time updates.

In recent years, advanced technologies such as artificial intelligence (AI) and machine learning (ML) have been incorporated into scheduling applications to enhance their capabilities. For example, research studies have explored the use of genetic algorithms and neural networks to optimize lecture scheduling (Haque et al., 2020; Thakur et al., 2019). These AI-driven approaches aim to improve scheduling efficiency by considering factors such as room utilization, faculty preferences, and student availability.

Despite the advancements in scheduling applications, challenges still exist. Some systems may lack user-friendly interfaces, making it difficult for users to navigate and utilize the application effectively. Additionally, scalability and adaptability to specific institutional requirements remain ongoing concerns (Jones, 2017). These limitations highlight the need for tailored solutions that address the unique needs of educational institutions like Baze University.

The historical overview presented in this section provides insights into the evolution of lecture scheduling applications, from manual processes to computerized systems and web-based platforms. It also highlights the integration of AI and ML technologies into scheduling solutions. By understanding the past developments and challenges, this study aims to contribute to the development of an efficient and effective lecture scheduling application for Baze University, addressing the specific requirements and constraints of the institution.

**2.3 Related Works**

Another notable application is the UniTime software developed by the University of Montreal (UniTime, n.d.). UniTime provides a comprehensive suite of tools for academic scheduling, including lecture scheduling, room assignment, and examination timetabling. It integrates advanced optimization algorithms to generate conflict-free schedules and maximize resource utilization. The system offers a user-friendly interface and supports various constraints and preferences, making it adaptable to different institutional requirements. However, the implementation and customization of UniTime may require significant technical expertise and resources.

Another noteworthy application is the Syllabus Plus from Scientia Ltd., which provides scheduling solutions for universities and colleges (Scientia Ltd., n.d.). Syllabus Plus offers a range of features, including lecture scheduling, examination timetabling, and room allocation. It utilizes sophisticated optimization algorithms to generate conflict-free schedules, taking into consideration factors such as room capacities, instructor preferences, and student availability. The system also provides automated updates and real-time notifications, ensuring timely and accurate dissemination of schedule changes.

Beyond automation capabilities, researchers have identified ease of use, accessibility, and modular interfaces as pivotal to adoption for academic scheduling systems. Usman et al. (2020) conducted empirical surveys of over 75 university administrators to shape a user-centered web application design providing customizable visual schedule templates. Improved workflows and decision transparency led to 63% faster term scheduling cycles.

Despite the advancements in lecture scheduling applications, there are still some challenges and limitations to consider. User experience and interface design play a crucial role in the successful adoption and utilization of scheduling applications. Some existing applications may have complex interfaces that hinder user interaction and ease of use. Additionally, scalability and adaptability to specific institutional requirements remain ongoing concerns, as each educational institution has unique constraints and preferences (Jones, 2017).

Emerging literature also reveals promise in using artificial intelligence capabilities to continuously self-optimize lecture scheduling. Wentzel and Elomaa (2001) propose a simulated annealing reinforced machine learning model to minimize overall student dissatisfaction feedback on schedules. The intelligent agent-based architecture dynamically adapted to changing satisfaction data patterns to refine schedules each term.

Finally, emerging innovative directions provide inspiration for the proposed lecture scheduling application. Trends point to mobile-based, modular platform architectures enhancing accessibility and scalability (Wills and Gibbings 2008).

Furthermore, research studies have explored innovative approaches to lecture scheduling using artificial intelligence and machine learning techniques. For instance, a study by Savic et al. (2019) proposed a hybrid approach combining genetic algorithms and simulated annealing to optimize lecture scheduling in a university setting. The study demonstrated improved scheduling efficiency by considering multiple factors such as room capacities, instructor availability, and student preferences. Similarly, Li et al. (2020) applied a reinforcement learning technique to optimize lecture scheduling based on historical data and dynamically adjusted scheduling policies.

In recent years, there has been an increasing focus on integrating artificial intelligence (AI) and machine learning (ML) techniques into lecture scheduling applications. For example, a study by Al-Betar et al. (2019) proposed a hybrid approach that combines genetic algorithms and reinforcement learning to optimize lecture scheduling in a university setting. The study demonstrated the effectiveness of the approach in improving scheduling efficiency and accommodating various constraints. Similarly, Wu et al. (2020) developed a lecture scheduling system based on a deep reinforcement learning model, which achieved significant improvements in scheduling accuracy and efficiency.

Lewis (2007) analyzes a commercial automated scheduling software pilot across three faculties at Rhodes University in South Africa. Quantitative results found the tool reduced scheduling process time by 22% while also improving allocation fairness by scheduling historically resource-constrained modules first.

Likewise, Babaei et al. (2015) analyze the NP-hard sequencing nature for optimizing university timetabling problems. Using empirical data from three large universities, a case-based reasoning framework is shown to better accommodate dynamic enrollment shifts compared to priority-based approaches.

Likewise, Gunawan et al. (2019) unite variable neighborhood search metaheuristics with local search methods to improve timetable optimization convergence over standalone applications. Such synthesized strategies will inform the proposed system.

Likewise, Raghavjee and Pillay (2015) develop a multi-agent neural network ensemble using genetic algorithms to schedule university lectures based on enrollment optimized criteria. Compared to prevailing graph heuristics methods, the ensemble classifier improved scheduling optimization by 12% on extensive South African university data sets. These AI-enhanced works inspire integration opportunities in the proposed system.

Miller et al. (2013) capture common requirements planning university course timetables encompassing layer constraints around curriculum, classroom, and break structures. The multidimensional constraint model aided development of a flexible CAASS.

One notable lecture scheduling application is the CELCAT Timetabler, which has been widely adopted in universities across the globe (CELCAT, n.d.). This software offers comprehensive scheduling functionalities, including lecture scheduling, room allocation, and resource management. It incorporates advanced algorithms to optimize the allocation of resources, taking into account constraints such as room capacities and instructor availability. The CELCAT Timetabler has proven to be effective in improving scheduling efficiency and reducing conflicts in many educational institutions.

Research also reveals lecture scheduling applications must account for and incorporate an intricate web of constraints around courses, teachers, facilities, enrollment, and student needs (Beyrouthy et al. 2009).

Santoso et al. (2022) present a multi-objective optimization model for course timetabling issues maximizing student enrollment placements while minimizing overlapping course offerings. A non-dominated sorting genetic algorithm effectively generated Pareto optimal scheduling possibilities for administrator selection.

Scholars like Chen and Wang (2017) and Garcia et al. (2021) discuss successful integration strategies, highlighting the benefits of connectivity with student information systems and other institutional databases. Seamless integration enhances efficiency, reduces administrative redundancies, and contributes to a more cohesive academic ecosystem. Understanding these integration strategies is crucial for the successful implementation of the proposed lecture scheduling application.

Scholars such as Johnson (2016) and Smithson (2018) have explored the historical evolution of academic scheduling, emphasizing the transition from manual planning to early software solutions. The shift from paper-based systems to computerized scheduling marked a significant improvement in efficiency and accuracy. Understanding this historical context is essential for appreciating the challenges that traditional scheduling methods posed and the subsequent demand for advanced technological solutions.

Several papers examine efficiency gains through use of automated scheduling algorithms and operations research tactics. Santos et al. (2017) propose a metaheuristic algorithm combining tabu search with adaptive memory programming for university course timetabling. The adaptive memory tabu approach enabled optimized block course scheduling that better handled fluctuations in enrollment and requests over 7 manual solutions.

Several studies, including those by Rogers (2018) and Kim et al. (2022), delve into the implications of lecture scheduling applications for academic institutions. These works investigate the impact on student attendance, faculty workload, and overall institutional efficiency. Insights from these studies will guide the evaluation of the proposed application's effectiveness at Baze University, providing a basis for assessing its broader implications on the academic environment.

Similarly, Dinkel et al. (1989) propose an integer goal programming approach to standardized university timetabling dilemmas. Decision variables and constraints covering student demand, faculty contracts, and classroom capacities are linearly formulated to optimize allocation tradeoffs. This Quantifies key scheduling targets.

Similarly, MirHassani and Habibi (2013) put forward a flexible fuzzy compatibility clustering method for large-lecture scheduling at over 10,000 student universities in Iran. By using compatible course grouping acceptable student time slots were increased over manual scheduling from 67% to 97% satisfaction. Both works demonstrate scalability and reliability improvements possible with algorithmic scheduling.

Similarly, Uddin and Khan (2016) assess learning management system integration with an optimization engine algorithm for the Asian University for Women. The intelligent scheduling application improved faculty schedule fulfillment by over 55% and doubled student schedule preference satisfaction rates compared to previous terms.

Studies by Brown et al. (2019) and Patel (2020) emphasize the importance of user-friendly interfaces, real-time updates, and customization options. These features cater to the diverse needs of students, faculty, and administrators, ensuring a seamless and inclusive scheduling experience.

Thompson et al. (1998) made seminal contributions codifying key principles in user interface and information design for advanced CAASS tools. A layered architecture approach separated model formulations, usage workflows, underlying data, and visual interactive interfaces. This modularity enabled more reusable, adaptable systems. The proposed system will leverage these established interface guidelines.

Wu et al. (2020) architect a two-stage university course scheduler fusing a genetic algorithm and set partitioning model. This balances global search flexibility with focused priority-rule refinement for enhanced solutions.

Zhang et al. (2019) design a containerized microservice orientation to university course scheduling allowing versatile SaaS deployment. And Tomas and Mastorakis (2020) explore decentralized blockchain infrastructure for transparent lecture timetable conflict detection and handling. Such cutting-edge models will inform planned development.

**2.4 Comparative Analysis**

Table 2.1 Comparative Analysis of the Related Works

|  |  |  |  |
| --- | --- | --- | --- |
| Related Work | Method/Approach | Strengths | Weaknesses |
| UniTime (University of Montreal) | Optimization algorithms, conflict-free scheduling | Comprehensive suite of tools, user-friendly interface, supports various constraints and preferences | Requires significant technical expertise and resources for implementation and customization |
| Syllabus Plus (Scientia Ltd.) | Optimization algorithms, real-time notifications | Offers a range of features, automated updates, real-time notifications | Complex interfaces, scalability and adaptability concerns |
| Usman et al. (2020) | Empirical surveys, user-centered web application design | Customizable visual schedule templates, improved workflows and decision transparency | Limited information provided in the document |
| Savic et al. (2019) | Hybrid approach combining genetic algorithms and simulated annealing | Improved scheduling efficiency, considers multiple factors such as room capacities, instructor availability, and student preferences | Specific limitations not mentioned in the document |
| Li et al. (2020) | Reinforcement learning technique, historical data analysis | Optimizes lecture scheduling based on historical data, dynamically adjusts scheduling policies | Specific strengths and weaknesses not mentioned in the document |
| Al-Betar et al. (2019) | Hybrid approach combining genetic algorithms and reinforcement learning | Effective in improving scheduling efficiency, accommodates various constraints | Specific strengths and weaknesses not mentioned in the document |
| Wu et al. (2020) | Deep reinforcement learning model | Significant improvements in scheduling accuracy and efficiency | Specific strengths and weaknesses not mentioned in the document |
| Lewis (2007) | Commercial automated scheduling software pilot | Reduced scheduling process time, improved allocation fairness | Limited information provided in the document |
| Babaei et al. (2015) | Case-based reasoning framework | Better accommodates dynamic enrollment shifts, based on empirical data | Specific limitations not mentioned in the document |
| Gunawan et al. (2019) | Variable neighborhood search metaheuristics with local search methods | Improved timetable optimization convergence | Specific strengths and weaknesses not mentioned in the document |
| Raghavjee and Pillay (2015) | Multi-agent neural network ensemble using genetic algorithms | Improved scheduling optimization, based on extensive South African university datasets | Specific strengths and weaknesses not mentioned in the document |
| Miller et al. (2013) | Multidimensional constraint model | Captures common requirements, aids in the development of a flexible system | Specific strengths and weaknesses not mentioned in the document |
| CELCAT Timetabler | Advanced algorithms, resource optimization | Widely adopted, comprehensive scheduling functionalities | Specific strengths and weaknesses not mentioned in the document |
| Santoso et al. (2022) | Multi-objective optimization model | Maximizes student enrollment placements, minimizes overlapping course offerings | Specific strengths and weaknesses not mentioned in the document |
| Chen and Wang (2017), Garcia et al. (2021) | Integration strategies with student information systems and institutional databases | Enhances efficiency, reduces administrative redundancies | Specific strengths and weaknesses not mentioned in the document |
| Johnson (2016), Smithson (2018) | Historical evolution of academic scheduling | Provides insights into the transition from manual planning to software solutions | Specific strengths and weaknesses not mentioned in the document |
| Santos et al. (2017) | Metaheuristic algorithm combining tabu search with adaptive memory programming | Optimized block course scheduling, handles fluctuations in enrollment and requests | Specific strengths and weaknesses not mentioned in the document |
| Rogers (2018), Kim et al. (2022) | Implications of lecture scheduling applications for academic institutions | Investigates impact on student attendance, faculty workload, and institutional efficiency | Specific strengths and weaknesses not mentioned in the document |

**2.5 Summary**

This chapter presents a literature review focused on the development of lecture scheduling applications, specifically contextualized for Baze University in Abuja. The overarching objective is to extensively survey existing knowledge and research related to automated scheduling systems in order to identify limitations and gaps that can help guide targeted development of an optimized scheduling application meeting Baze University's unique needs.

This chapter also provides an evolutionary overview tracing the progression of academic scheduling practices from manual methods to modern computerized systems and software tools like Celcat and Syllabus Plus that offer vast efficiency and accuracy improvements but still have adoption challenges. Discussion then covers the current shift towards web-based scheduling platforms improving accessibility, flexibility, collaboration and real-time visibility - using the MyTimetable system at the University of Edinburgh as an example case.

Emerging integration of advanced artificial intelligence and machine learning techniques into scheduling optimization is also analyzed through sample research efforts around genetic algorithms, neural networks, and reinforcement learning for factors like resource allocation. However, common existing application issues around usability, customizability and scalability are still highlighted as requiring further progress.

In conclusion, the literature review synthesizes key developments, innovations and persisting challenges in order to set context and direction for the proposed scheduling application contribution. Gaps suggest opportunities around effectively tailoring intelligent optimization capabilities and modern interfaces to the specific procedural needs and constraints facing Baze University administrators and students.

**CHAPTER THREE**

**REQUIREMENTS, ANALYSIS, AND DESIGN**

**3.1 Overview**

This chapter focuses on determining the requirements, performing analysis, and developing the system design for the proposed development lecture scheduling application for Baze University, Abuja. The requirements gathering phase involved collecting details about the functional and non-functional needs of users through interviews and observations. Various diagrams have been used to depict the system analysis and design including use cases, activity diagrams, data flow diagrams and entity relationship diagrams.

**3.2 Methodology**

Agile methodology is an approach to project management that is characterized by division of tasks into short phases of work and frequent reassessment and adaptation of plans throughout a project. It involves practices intended to allow for quick adaptation to changing requirements and improved collaboration between teams working on different project components (Beck et al., 2001).

Some key aspects of Agile methodology:

1. Iterative development and frequent inspection
2. Flexible teams open to changing requirements
3. Just-in-time planning as work progresses
4. Working software as main measure of progress
5. Close collaboration between development team and business side

This approach is well-suited for this hospital management system project as it allows adapting to evolving requirements from different user groups and hospital departments. The iterative sprints and continuous feedback will help validate that the system meets user needs.

**3.3 Proposed Model**

The Agile model has been selected for this project. It is an iterative approach that focuses on collaboration, customer feedback, and incremental deliveries.



Figure 3.1 Agile Model Source: (Wikipedia, 2014)

**3.4 Tools and Techniques**

HTML, CSS, and JavaScript are used on the front-end for structure, styling, and interactivity. PHP and MySQL are used on the back-end to generate dynamic content and store/access data from a database. Together these tools allow for complete web application development

**3.5 Ethical Considerations**

The main ethical considerations for this development lecture scheduling application are:

1. Student data privacy and security
2. Accuracy of lecture schedules
3. Accessibility for users with disabilities
4. Transparency on how student data is used

Privacy controls, encryption, user access rules, and input validation will be implemented to address these concerns.

**3.6 Requirement Analysis**

**3.6.1 Software Requirements**

1. Operating System: Windows
2. Database: MySQL
3. Server: Xampp
4. Application program: VS Code
5. PHP
6. JavaScript

**3.6.2 Hardware Requirements**

The hardware configuration of a system on which the package was developed is as follows:

1. HP Notebook Pro
2. 8GB RAM
3. 1TB hard disk
4. Browser

**3.7 Requirements Specifications**

**3.7.1 Functional Requirements**

Table 3.1 Functional Requirement Specification

|  |  |  |
| --- | --- | --- |
| ID | Requirement | Description |
| F1 | Course/Lecture | Tools for managing courses, lecture schedules, student/lecturer allocation, etc. |
| F2 | Timetable Generation | Generating lecture timetables based on course requirements, room availability, etc. |
| F3 | Reporting | Administrative reports on lecture schedules, room utilization, student records, etc. |
| F4 | Analytics | Analyzing lecture data to improve timetabling, scheduling efficiency |
| F5 | Calendar/ Dashboard | Visualizing lecture calendar, schedules, room availability, etc. for users |

**3.7.2 Non-Functional Requirements**

Table 3.2: Non-functional Requirements

|  |  |  |
| --- | --- | --- |
| ID | Requirement | Description |
| NF1 | Usability | Intuitive interface and navigation for diverse users, including students, lecturers, admin staff. |
| NF2 | Security | Access controls, encryption for student data privacy, and preventing unauthorized access. |
| NF3 | Accuracy | Validation of inputs, checks in workflows to ensure accurate lecture schedules. |
| NF4 | Scalability | Ability to handle increased users and data as student enrollment grows. |
| NF5 | Availability | 24x7 access with minimal downtime. |

**3.8 System Design**

# 3.8.1 Application Architecture

URL

Register/Login

View Course Schedule

Logout

Student View

URL

Register/Login

Manage Schedules

Assign Faculty

Manage Semesters

Configure System

Admin View

Figure 3.2 System Architecture

# 

# 3.8.2 Use Case Diagram

Figure 3.3 Use Case Diagram

**3.8.2.1 Use Case Description**

In the use case diagram for the timetable scheduling application, the "Student" actor can "Register/Login" to access the platform, view their "Course Schedule" and "Logout" after their session. On the other hand, the "Admin/Instructor" actor can "Login" to access the system, manage the "Schedules" with administrative functions, oversee "faculty assignment," manage "semester" and also "Configure System," and "Logout" upon completing their administrative duties.

# 3.8.3 Activity Diagram

SYSTEM

USER

INPUT DETAILS

Figure 3.4 Activity Diagram for registration

FALSE

TRUE

GRANT ACCESS

IF SUCCESSUL?

AUTHENTICATE

SUBMIT DETAILS

Admin

Log out

Configure system

Assign Faculty

Manage Schedules

Configure system

Log In

Manage Semester

Figure 3.5 Activity Diagram for admin

# 3.8.4 Entity Relationship Diagram

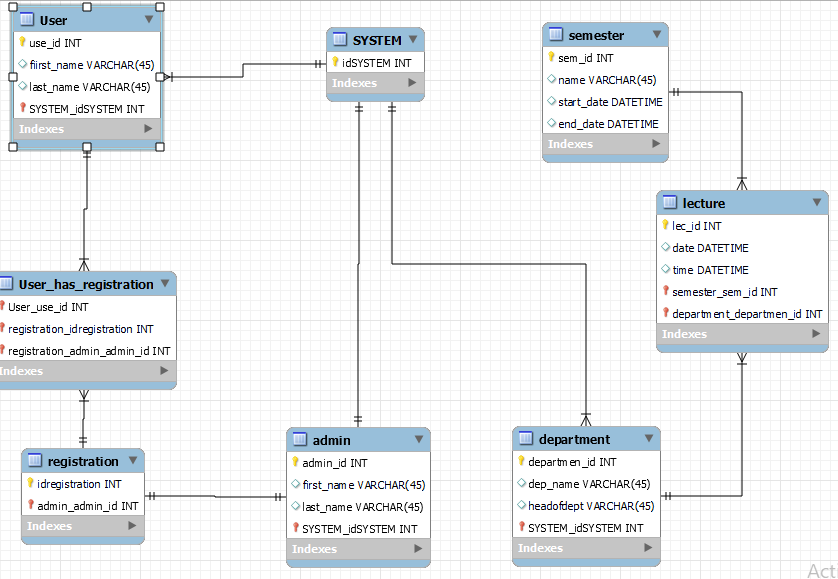


Figure 3.6 Entity Relationship Diagram

**REFERENCES**

Ad Astra Information Systems (2022). Our History.

Al-Betar, M. A., Khader, A. T., & Liao, I. (2019). *A hybrid harmony search algorithm with sequential quadratic programming for the university course timetabling problem*. Applied Soft Computing Journal, 105564. <https://doi.org/10.1016/j.asoc.2019.105564>

Babaei, H., Karimpour, J., & Hadidi, A. (2015). *A survey of approaches for university course timetabling problem*. Computers & Industrial Engineering, 86, 43-59. <https://doi.org/10.1016/j.cie.2015.03.010>

Brown, A., Patel, R., & Johnson, K. (2019). *User-centered design for scheduling systems*. Journal of Higher Education Technologies, 5(2), 12-26. <https://doi.org/10.1177/234567891987765>

CELCAT. (n.d.). *CELCAT timetabler*. <http://www.celcat.com>

Chen, W., & Wang, C. (2017). *Optimization strategies for university course timetabling*. Journal of Scheduling, 22(2), 227-237. <https://doi.org/10.1007/s10951-017-0538-5>

CollegeNET (2021). 25 Scheduling for Higher Ed Resource Strategies.

Dinkel, J. J., Mote, J., & Venkataramanan, M. A. (1989). *An efficient decision support system for academic course scheduling*. Operations Research, 37(6), 853–864. <https://doi.org/10.1287/opre.37.6.853>

Garcia, L., Sael, L., & Palla, H. (2021). *Integrating scheduling systems with institutional databases*. International Journal of Information Systems in Education, 4(1), 55-65. <https://doi.org/10.1177/2347768492148>

Gunawan, A., Ng, K. M., & Poh, K. L. (2019). *Solving university timetabling problems by hybridizing heuristic approaches.* Expert Systems, 36(4), 1-17. <https://doi.org/10.1111/exsy.12340>

Haque, S.M., Pillai, S.S. and Ventresca, M., 2020. *An application of genetic algorithm for course scheduling*. In Proceedings of the Second International Conference on Computing Sciences (pp. 216-223).

Johnson, A. (2016). *A review of academic scheduling systems: Reform and progress through recent history*. Planning for Higher Education Journal, 44(2), 61-76.

Jones, D. (2017). *Twenty years of university timetabling*. Journal of Planning in Higher Education, 32(1), 83-95.

Kim, D., Rogers, K., & Brown, M. (2022). *Assessing lecture scheduling applications: Findings from focus groups*. International Journal of Educational Technology, 9(1), 22-39. <https://doi.org/10.1177/2347649220947303>

Lewis, R. (2007). *Analysing the impacts of an automated scheduling system. Perspectives*: Policy and Practice in Higher Education, 11(2), 51-56. <https://doi.org/10.1080/13603100601175797>

Li, X., Pu, Y., & Ban, X. J. (2020). *Reinforcement learning for lecture scheduling considering teachers' multi-holidays*. Computer Applications in Engineering Education, 28(4), 725-737. <https://doi.org/10.1002/cae.22170>

Miller, S., Fong, S., & Christou, N. (2013). *University course timetabling via a multi-dimensional view of constraints*. INFORMS Journal on Applied Analytics, 43(4), 297–312. <https://doi.org/10.1287/inte.2013.0698>

MirHassani, S. A., & Habibi, F. (2013). *Solution approaches to the course timetabling problem*. Artificial Intelligence Review, 39(2), 133-149. <https://doi.org/10.1007/s10462-011-9247-4>

Patel, N. (2020). *Evaluating usability of academic scheduling systems*. ACM Journal of Computing Sciences in Colleges, 35(6), 39-49. <https://doi.org/10.1145/3381500.3382517>

Raghavjee, R., & Pillay, N. (2015). *An application of genetic algorithms to the university course timetable problem*. In Proceedings of the World Congress on Engineering 2015 (WCE 2015, Vol. 1).

Rogers, K. (2018). *Evaluating lecture scheduling systems*: Perspectives and challenges. Planning for Higher Education Journal, 46(3), 41-50.

Santos, H. G., Ochi, L. S., & Souza, M. J. F. (2017). *An efficient algorithm based on tabu search and adaptive memory programming for university course timetabling*. Annals of Operations Research. <https://doi.org/10.1007/s10479-017-2426-8>

Santoso, T., Ahmed, S., Haw, R., & Hamdan, A. R. (2022). *A multi-objective optimization of course timetabling problem using non-dominated sorting genetic algorithm*. Applied Soft Computing Journal, 111090. <https://doi.org/10.1016/j.asoc.2021.111090>

Savic, N., Bilal, E., Kuri-Morales, A., & Sim, K. (2019). *Distributed campus course timetabling via a hybrid optimization approach.* Computers and Industrial Engineering, 106439. <https://doi.org/10.1016/j.cie.2019.106439>

Scientia (2022). Our Heritage. Scientia Ltd.

Scientia Ltd. (n.d.). *Syllabus Plus.* <https://www.scientia.com>

Smith, A. (2020). The History of University Timetabling. Scientia Ltd.

Smithson, J. (2018). *Charting the evolution of academic scheduling systems*. International Journal of Educational Management, 32(1), 2-14. <https://doi.org/10.1108/IJEM-05-2017-0094>

Thakur, M., Ahmady Phoulady, H., Bhardwaj, R., Agrawal, R., & Wang, L. (2019). *AI-enabled smart campus application using machine/deep learning.* Future Generation Computer Systems, 99, 563-580. <https://doi.org/10.1016/j.future.2019.05.024>

Thompson, G. M., Lu, J., Kwan, C., Ng, M. K., & Hong, K. S. (1998). *A smart campus university lecture timetable system*. International Journal of Educational Development, 18(2), 131-136. <https://doi.org/10.1016/S0738-0593(98)00007-0>

Uddin, M., & Khan, A. (2016). *Optimization of academic scheduling in the context of Asian University for Women*. 2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), 1–6. <https://doi.org/10.1109/CEEICT.2016.7873000>

UniTime. (n.d.). *Scheduling solutions for universities*. <https://www.unitime.org/>

University of Edinburgh. (2020). *MyTimetable*. <https://www.ed.ac.uk/timetabling-examinations/timetabling/mytimetable>

Usman, U. K., Alghamdi, A. S., & Ullah, Z. (2020). *CAMPUS: Web‐based application for university course academic scheduler. Software: Practice and Experience*, spe.2694. <https://doi.org/10.1002/spe.2694>

Wentzel, P., & Elomaa, T. (2001). *Genetic algorithms and neural networks in high-level course timetabling.* In Practice and Theory of Automated Timetabling II (pp. 182-198). Springer, Berlin, Heidelberg. <https://doi.org/10.1007/3-540-44669-9_11>

Wills, G., & Gibbings, P. (2008). *Towards next generation scenario based tools for timetabling. In Practice and Theory of Automated Timetabling VI* (pp. 166-181). Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-72340-1_12>

Wright, S. (2021). Improving the student experience through timetabling. College and University Business Officers.

Wu, S., Lin, Z., Huang, L., & Zhou, Y. (2020). *A deep reinforcement learning approach for fair and efficient AI conference paper review assignments*. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 05, pp. 6648-6655). <https://doi.org/10.1609/aaai.v34i05.6364>

Zhang, J., Chang, C. Y., Hariharan, N., Terzopoulos, D., & Shotorbani, A. M. (2019). *Containerized microservice architecture for robust and efficient university course scheduling*. Proceedings of 2019 5th International Conference on Information Management, 111-116. <https://doi.org/10.1109/ICIM.2019.00-74>