

Design, Implementation and Automation of University Timetable Using a Multi-Server Queuing Model

Sanvee Gael Kuakuvi, Akwuma Nathaniel Eru, Solomon Ahiaba, Temitope Olufunmi Atoyebi, Prema Kirubakaran

Department of Information Technology, Nile University of Nigeria, Abuja, Nigeria

Published September 2026

Abstract

Manual entry of courses in order to form a timetable can be stressful, time-consuming, inefficient and also leads to scheduling conflicts. So, this study focuses on how to automate and generate a university timetable using a Multi-server Queuing Model so as to make doing the task efficient. The methodology will be centered on applying a multi-server queuing model for visualizing timetable creation as a virtual queue. This theoretical model will be implemented via a dedicated web application built on a Python-based technology stack: Streamlit for the creation of an interactive user interface, Pandas for data management and constraint handling, SimPy to power the core discrete-event simulation responsible for processing the queue and allocating the resources and the database will be handled using supabase. Output from the system, including the final timetable and performance analytics, is visualized dynamically with Plotly. Expected findings during the testing phase should confirm the system's efficacy, showing it was able to generate a fully optimized, conflict-free timetable in less than one minute, thus increasing lecture room occupancy significantly and eliminating any administrative errors. This study will provide a powerful and scalable solution for complex academic scheduling. Key recommendations are for academic institutions to adopt such automated systems to enhance operational efficiency and resource utilization. The recommended work based on this is to expand the model to involve more types of constraints and adapt the framework for other scheduling challenges, such as university examinations.

Keywords: university timetable automation; multi-server queuing model (M/M/c); discrete event simulation; timetable scheduling; resource optimization; SimPy; Streamlit; Pandas; Supabase; Plotly

1 Introduction

Academic timetables are specific guidelines provided in a structured form by which academics or lecturers are assigned their courses, venues, and specific time slots to aid in the efficient delivery of academic courses. Therefore, the processes of generating academic timetables are crucial in ensuring efficient academic activities, optimal resource utilization, and efficient student learning outcomes. The timetabling approaches in academic institutions, particularly in universities, have generally been described as a complex combinatorial optimization problem due to various factors. Timetabling in academic institutions involves the allocation of limited resources to each institution, and at the same time, emphasis is placed on ensuring various constraints are equally satisfied.

For instance, some of the constraints in academic timetabling may depend on lecturer availability, credit units of each academic course, student enrollment rates, capacity limits in various classes, and other limitations within the institution. The challenges associated with timetable generation are

particularly evident in developing countries where institutional growth often exceeds infrastructure expansion. In Nigeria, there has been an increase in student enrollment, growth in academic programs, and a lack of teaching facilities, and these factors have put more pressure on available scheduling systems.

Previous studies have identified that several universities, especially in Nigeria, experience a lack of lecture halls and other teaching facilities, creating problems in university timetabling, including scheduling conflicts and inefficiencies in utilizing available academic facilities [1]. In addition, through increased enrollment rates, universities are competing to get various available lecture halls and periods, and this situation makes it complicated to prepare a university timetable. Despite this, several higher learning research studies have revealed that several higher learning centers still apply manual methods to assist them in preparing timetables. Although these methods are applicable due to their flexible nature, several timetabling and scheduling procedures applied by these higher learning centers have often depended on manual methods, which are highly prone to several inadequacies and inefficiencies. Research by [2] identified some of these inadequacies to include manual timetabling methods, where a higher number of clashes between lectures have been experienced.

Poor timetable generation affects the quality of education and institutional efficiency. Scheduling conflicts can make students miss lectures, create noisy learning environments, and affect academic coordination negatively. Moreover, inefficient scheduling increases the stress of lecturers' workloads and reduces instructional effectiveness. Recent research has shown that optimized academic timetabling systems improve the utilisation of resources within an institution, reduce administrative workload, and enhance student learning experiences [3]. In response, automated techniques for timetable generation have been developed to include genetic algorithms, constraint satisfaction programming, hybrid optimization techniques, and artificial intelligence-based scheduling models.

Genetic algorithms and heuristic optimization methods have become popular because they result in near-optimal scheduling solutions for highly constrained environments. However, studies have indicated that most of them require considerable computational resources and technical expertise, which may limit the adoption of the techniques in resource-constrained university environments [4]. Another way through which academic scheduling challenges can be addressed is through the use of Queuing Theory. It refers to a mathematical study of systems wherein limited resources are allocated to competing demands over time. Queuing Theory has been widely applied in service-oriented systems such as healthcare, telecommunications, transportation, and cloud computing to improve service efficiency and reduce congestion. Recent studies have demonstrated that queuing-based models provide effective analytical tools for optimizing resource allocation in complex service systems [1].

The Multi Server Queuing (M/M/c) model, in particular, allows multiple service channels to operate simultaneously while serving stochastic arrival requests. In order to do that, the problem of lecture scheduling in university timetable generation may be seen as a queuing model with arrival of the scheduling request for lectures and the lecture rooms being the parallel servers. Therefore, the aim of this study is to investigate the application of the multi-server queuing model - M/M/c - for the development of an automated timetable generation system for universities. This would also provide a platform for an efficient solution for timetable generation problems which are common in Nigerian universities.

Academic timetable generation is a basic administrative task in any university calendar as it determines how lecturers, courses, lecture halls, and time slots are allocated in ensuring effective teaching and learning. However, some universities in developing countries are still struggling to come up with efficient and conflict-free academic timetables. The simultaneously rising student enrollment rates and increased number of academic programs have complicated academic scheduling, turning this traditionally administrative task into a daunting task. Even as technology advances in ensuring effective management in various sectors of the academia, some universities are still using traditional approaches in academic timetable scheduling. These approaches are dependent on either manual or semi-manual scheduling processes. Literature indicates that most conventional approaches are prone to lecturers' time clashing, having double allotments of halls, and improper distribution of courses to specific teaching periods. These errors have been cited to necessitate routine adjustments to the timetables, thus interfering with academic activities as highlighted in [5].

Furthermore, another effect of inefficient allocation and planning time slots is wasting available academic resources, where some halls are not used to capacity while others are over-used. Despite the development of various automated timetabling approaches, such as genetic algorithms, constraint programming, and hybrid metaheuristics, that can be applied to solve these problems, they still rely heavily on computationally intensive resources and advanced technical knowledge and thus cannot be practically implemented in universities due to a lack of sufficient computational power [5]. Additionally, recent studies show that the current state of affairs in university academic scheduling indicates that most of the existing models assume timetabling to be a static problem and fail to account for its dynamic nature, meaning that there is a need to find an alternative scheduling approach that would ensure efficient use of academic resources and allow addressing scheduling needs dynamically.

This study attempts to examine the potential of using the Multi Server Queuing (M/M/c) model in designing an automated university lecture timetable generation system that will help minimize scheduling conflicts and optimize resource allocation in universities.

The aim of this study is to design and implement an automated university lecture timetable generation system based on the Multi Server Queuing (M/M/c) model to enhance academic resource allocation and mitigate scheduling conflicts.

The specific objectives of the study are to:

1. design a web application that provides role-aware interfaces for administrators, lecturers and students to manage and interact with the automated timetable scheduling system;
2. implement a conflict detection and logging mechanism that identifies, categorizes and records scheduling conflicts; and
3. test and evaluate a relational database system capable of storing and managing all scheduling data, including course records, lecturer profiles, room configurations, section assignments, simulation run results and timetable outputs.

This research is significant for the following reasons:

It offers an automated method in the generation of timetables in the context of universities by using the M/M/c model, thereby improving efficiency in the planning and management of timetables in academic institutions. The findings are significant in improving the management of lecture timetables in the context of universities, especially in institutions where there is an increasing number of students and limited infrastructural facilities, thereby restricting lecture planning timetables.

From the perspective of managing timetables in the context of universities, the system will improve a lecturer's task efficiency in the following ways:

The system will assist in eliminating timetable planning inefficiencies, thereby minimizing timetables conflicts such as the simultaneous allocation of lecture halls and lecturers' time.

Lecturers are expected to benefit by having an automated timetable generation system that will offer a clearer and more balanced timetable, which should result in less stress and workload with regards to timetable clashes and last-minute timetable adjustments. Students are equally expected to benefit by having enhanced learning continuity with regards to minimized lecture clashes and an enhanced arrangement for lecture rooms, which should further improve their general academic performance and commitment.

In academic and research circles, this study has made a significant contribution to the application of Queuing Theory in academic timetable generation. In adapting the result of the Multi Server Queuing (M/M/c) model in academic timetable generation, this study has extended the application of operations research in academic management and has set the tone for further research in academic timetable generation.

2 Related literature

The university academic timetable is a core administrative activity that offers ordered coordination in teaching and learning processes through course allocation to lecturers, classrooms, and time slots. The institutional resources utilized, the quality of instruction, and the continuity of students' learning are influenced by the effectiveness of this process. Universities now have larger student populations, more academic programmes, and complex interdisciplinary course structures; these factors contribute significantly to making timetable generation a large-scale combinatorial optimization problem, best addressed by systematic and intelligent scheduling techniques [6].

Conventionally, most higher learning institutions, especially those in developing countries, have traditionally adopted the manual or semi-automated approach, which involves spreadsheet-based coordination and human-based decision-making to create academic timetabling. Although these methods are characterized by operational flexibility, research indicates that they are usually prone to scheduling conflicts, inefficient classroom utilization, and increased workloads for administrators and limited adaptability to changing academic demands [7]. It is based on these challenges that researchers have attempted to develop automated scheduling approaches by including mathematical optimization, artificial intelligence approaches, and simulation-based modeling in generating timetables efficiently and reliably.

2.1 Conceptual framework

Essentially, the conceptual framework of this research is based on bringing together approaches to automated schedule systems, optimization techniques, queuing theory, and discrete event simulation. These approaches, in combination, form a framework through which it would be possible to design an automated university timetabling system. Academic timetabling is essentially a resource allocation problem. It is characterized by scarce resources such as halls, lecturers, and time slots competing to be allocated to other competing interests. In this context, [6] argues that university timetabling is characterized by the allocation of various activities to shared resources. However, these activities are complicated by enhanced student enrollments, diversified programs, and scarcity of infrastructural

support. Results have indicated that complex schedules are difficult to manage with manual methods of scheduling, leading to potential timetable conflicts and administrative inefficiencies [8]. Therefore, there is a need for automated timetabling systems that are able to model and express mathematical models of the scheduling requirement for the allocation of academic resources systematically.

One essential element of automated timetabling systems is optimization techniques. These strategies seek to balance lecturer responsibilities, reduce schedule conflicts, and make the most of the lecture facilities that are available. Many researchers agree that algorithmic optimization approaches, heuristics, and metaheuristics can provide optimal solutions to complicated scheduling problems [9]. However, most scheduling systems that utilize optimization algorithms consider lecture timetables as static processes and do not take into account the dynamics of scheduling requests caused by different circumstances such as differences in courses offered, different lecturer schedules, and resource constraints. Therefore, queuing theory is introduced as a modeling technique in order to describe scheduling demands as dynamic service demands competing for finite academic resources. As a matter of fact, queuing theory is a set of mathematical methods used to analyze systems where some entities require services provided by several servers. This theory enables assessing the performance parameters of a system such as waiting time, efficiency of the provided services, and resource usage [5]. In case of academic timetabling systems, lectures' scheduling requests could be considered as arriving entities, while lecture halls and time slots could be described as service facilities providing requested services to the entities.

As a complementary approach to queuing theory, discrete event simulation is introduced as another key aspect helping to model scheduling operations as they occur in real life situations and then evaluate the performance of the system under changing academic conditions. Discrete event simulation is helpful since it enables studying the operation of the system through a sequence of events, such as scheduling of lectures, identification of schedule clashes and resolution of conflicts, and resource release upon completing lectures. Scheduling using simulation helps reduce potential scheduling problems and ensures the best use of available academic resources since multiple scenarios could be simulated before implementing the solution [10]. Therefore, the theoretical framework used in this study includes principles of constraint satisfaction, queuing-based resource allocation, and simulation-based performance evaluation as ways to create an automated lecture scheduling system addressing both static and dynamic scheduling issues.

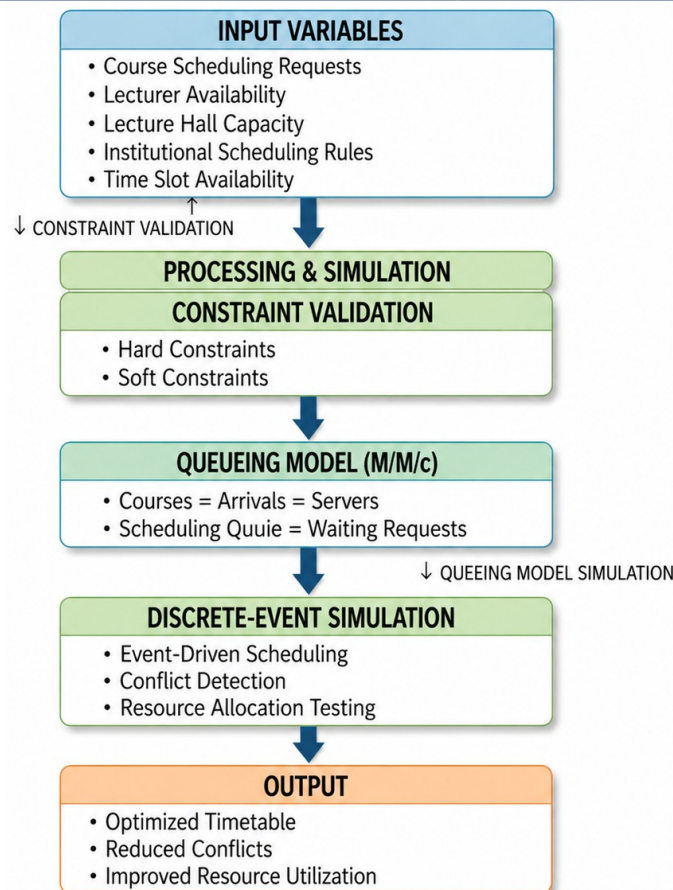


Figure 1. Conceptual framework for an optimized timetable system.

Based on the assumptions of this research project (Figure 1), scheduling requests are treated as inputs to the system; constraint satisfaction and queuing-based allocation are used to process these inputs. Finally, discrete event simulation is performed in order to generate an output in the form of optimally scheduled academic timetables. This integrated methodology will help meet the objectives set out in the study by adopting an organized methodology that will facilitate effective scheduling conflicts and optimized resource usage, thereby improving reliability in the timetable at the university academic setting.

2.2 Theoretical framework

There are three interrelated theoretical underpinnings forming the theoretical framework of this study that help in the creation of an automated system for academic timetable generation. Discrete event simulation theory, constraint satisfaction theory, and queuing theory belong to those. Combining the three theories helps understand how scheduling demands could be modeled as conflicts in terms of allocating scarce academic resources and then resolved through applying computational approaches to the problem.

2.3 Queuing theory

Queuing Theory can be described as a mathematical concept used for analyzing and examining competition for scarce resources. This concept is mainly focused on waiting lines or queues and examines such things as arrival rates, service rates, the number of service facilities, waiting times, and utilization rates regarding the analyzed systems. Queuing models could be referred to as one of the

major uses of operations research. They are extensively used by organizations for assessing the efficiency of resources utilization in service systems characterized by long waiting times due to insufficient service capacity [11]. Competition for resources has been analyzed with regard to some of the service-oriented systems, including healthcare, telecommunication, transportation, and computing systems, through the employment of queuing theory. It has been established that queuing models are important with regards to the optimization of service systems through prediction of the level of congestion, evaluation of waiting times, and identifying the bottlenecks of performance efficiency [12]. Considering the potential for such analysis, queuing theory has significant applicability in modeling complex scheduling systems in which there are many requests competing for common resources. The theory also has potential for decision-making that allows system designers to evaluate different service opportunities and determine the optimum allocation strategy.

It has also emerged that there is an organized way of approaching the allocation of the teaching facilities such as lecture rooms, labs, and lecturers within the academic timetable environment. In this case, the lectures or the lecture requests are treated as the entities that need service, whereas the facilities and the lecturers play the role of the service providers. Should the resources be unavailable at the requested time slot, the scheduling of the requests may be delayed or rescheduled, just like the case of clients waiting in the system of the resources' service providers. The use of the multiple-server queue system, like the case of the M/M/c model, ensures the smooth running of many channels of service at the same time. It can be noted that the use of the multiple-server system is effective in situations of high demand with scarce resources since it enhances the efficiency of service by reducing the waiting time compared to the use of the single-server system [11]. In addition, queuing theory offers the opportunity to perform evaluation of the performance of scheduling systems through quantifiable parameters such as the average time spent waiting, queue size, system utilization rate, and rate of output. This offers important knowledge on the efficiency with which academic resources are employed, helping to identify instances of inefficiencies in developing timetables.

This also means that the Queuing Theory is an essential part of an efficient scheduling system for processing scheduling requests. In this context, therefore, Queuing Theory is seen as an essential academic theory of reference for the intended research, with the goal of developing an automated system of academic timetabling that is geared towards the optimization of resource utilization and the elimination of conflicts associated with academic timetabling.

2.4 Discrete event simulation model

Discrete Event Simulation (DES) is among the most significant and widely used analytical modeling tools, which is often encountered in finding the needed insight into specific systems characterized by variations in value at particular points in time. It is particularly concerned with the operational characteristics of the system or its specific behavior, especially while in operational use, in relation to specific instances or particular events, referred to as discrete events, such as the arrival of entities and the allocation of resources. The application of DES in modeling the specific behaviors of such complex systems is often very effective in finding the complexities and variations in the particular system, which may not be easily and conveniently modeled by analytical models concerning the specific behaviors of the particular system, such as queuing equations or specific optimization models. "The strength of Discrete Event Simulation is in modeling real system operations by defining individual events and their relationships," especially in effectively modeling the specific system in finding the allocation of particular resources and in effectively resolving conflicts [13].

2.5 Empirical review

The existing literature on automated timetabling also involves investigations on the development and implementation of automated timetable generators using heuristic and meta-heuristic methodologies. In this context, [8] presented a model on the application of the Genetic Algorithm in automated timetable generation, which resulted in the achievement of near conflict-free schedules with high accuracy and also proved that academic timetabling is solvable using evolutionary methodologies while tackling hard and soft constraints. In addition, [14] presented a model on generating timetables by incorporating student preferences, which proved that incorporating evolutionary methodologies in timetabling generates efficient timetables by addressing hard and soft constraints.

Beyond pure genetic algorithms, hybrid and advanced metaheuristic approaches have also been applied to timetabling problems. Recent research emphasizes that hybrid metaheuristic structures, which incorporate greedy strategies together with genetic or other search mechanisms, are most effective in improving the quality of solutions and computational efficiency in large-scale scheduling instances. For instance, [15] suggested a hybrid metaheuristic which combines several search techniques to create feasible timetables faster compared to the use of commercial solvers, thus highlighting the possibility of employing hybrid algorithms in solving timetabling problems. There are also other examples of empirical research dedicated to the topic and demonstrating ways in which constraint and optimization models handle timetabling issues. Studies conducted on constraint-based scheduling models have also illustrated that embedding rule-driven logic into automated systems can improve the feasibility and adaptability of a schedule considerably, as compared to heuristic approaches. For example, [16] developed a constraint based automated timetable generator that outperformed standard genetic and simulated annealing models in terms of conflict elimination and execution time.

In contrast to algorithm only approaches, some research has focused on mathematical optimization models for academic scheduling [17]. A mixed integer linear programming (MILP) model of university coursework timetabling was reported to satisfy its objectives better than heuristic schedules while balancing lecturer workload. In general, timetabling models employ rigorous techniques for expressing and optimizing timetabling objectives; there has been an expansion of trends in timetabling research with the integration of hybrid and multi-objective issues in timetabling models. In a research by [16], it was emphasized that it was important to address issues concerning hybrid modes of learning within timetabling systems by creating a model that could accommodate both in-person and online modes of learning. These empirical studies, therefore, help to attest the premise for the use of automated timetabling systems, considering the benefits of algorithmic, constraint-based, and optimization techniques for the efficiency of timetabling systems. Nevertheless, the limitations experienced with the complex calculations, changing resources, and changing requirements of the timetabling systems present a need for continued research on the integration of queuing and simulation-technology-based systems.

2.6 Research gaps

Despite this significant research work on automated academic timetabling and the numerous optimization techniques for solving timetabling problems, there exist some major limitations to the implemented techniques, which highlight the scope for further research to overcome these limitations. First, while there exists a significant body of research on various algorithms on timetable generators, most of the algorithms implemented are heuristic and metaheuristic algorithms like genetic algorithms and hybrid metaheuristic algorithms. Although such algorithms perform optimally

in generating feasible timetables, they are often computationally intensive and require immense computational time and effort to optimize. For instance, both [8] and [14] proved that solving academic timetabling problems using genetic algorithm-based techniques can result in highly conflict-free timetables; however, such techniques are computationally intensive and are not suitable for dynamic environments where changes in academic timetabling conditions are frequent (i.e., the unavailability of lecture rooms due to unforeseen situations). Second, while the introduction of constraint algorithms results in improving the feasibility of timetabling problems by incorporating rule-driven techniques, it has limited ability to address issues of time-varying changes.

Last but not least, while pioneering works on queuing models have demonstrated practical value in service system performance analysis, relatively few empirical studies have so far applied the theory of queuing directly to university academic scheduling. Most automated timetabling research is still firmly based on heuristics, rule-based systems, or centric optimization instead of analytical resource allocation models such as queueing. Consequently, possible gains from queuing models in terms of dynamic resource allocation and performance monitoring in academic environments are relatively unexplored. Together, these gaps suggest three critical areas that have not been adequately addressed in the literature:

- **Dynamic representation of scheduling:** The time-dependent and unpredictable aspects of actual academic scheduling are not adequately captured by current timetabling techniques.
- **Frameworks for integrated simulation and optimization:** Few studies successfully assess and improve schedule performance prior to deployment by combining optimization techniques with discrete event simulation.
- **Models for analytical resource allocation:** In order to represent lecture scheduling as a service system where waiting behavior and resource usage may be anticipated and optimized, queueing theory has not been effectively employed.

By filling in these gaps, academic timetabling systems will be better equipped to manage complexity in the real world, adapt to changing scheduling requirements, and deliver quantifiable performance enhancements that are closely related to the study's goal.

3 Methods

3.1 Analysis of the existing timetabling system

Currently, the majority of universities use manual/semi-automated methods for generating academic timetables that require substantial administrative intervention and utilize rudimentary scheduling software tools. Such an approach leads to inefficiencies in terms of excessive lecture schedule clashes, inappropriate allocation of lecture halls and imbalance of lecturers' workloads in respective departments. More often than not, some halls might be used too frequently whereas others would be unused. This shows that an efficient optimization tool is missing to regulate allocation. In addition, adjustments to lecture schedules due to changes in lecturer availability and course details tend to cause difficulties and prolong the whole process of schedule generation since the process relies on manual adjustments. Finally, such an approach does not provide a reliable mechanism for detecting and resolving schedule conflicts. Thus, there is a clear need to implement an efficient timetable generation tool based on the principles of queueing theory.

3.2 System requirements

The design of the automated timetable generation system has been developed based on several functional and non-functional requirements that determine the behavior of the system and its quality attributes. These requirements would ensure that the system adequately tackles the weaknesses of existing systems of timetabling while remaining reliable, usable, and efficient in execution.

3.2.1 Functional requirements

The developed system is required to possess several basic functionalities associated with automatic lecture scheduling. First of all, the system should be able to receive, store and manage the necessary input data related to courses, lecturers and lecture halls. Secondly, it should be able to analyze and use this data in creating lecture schedules according to certain criteria. Thirdly, the system shall identify and correct possible conflicting situations arising from allocation of lecturing periods in time slots. Also, it should contain features enabling management of the database by adding, editing and removing course, lecturer and hall information. Finally, the system needs to generate efficient lecture schedules according to the M/M/c Multi-server Queuing Model, ensuring efficient allocation of available resources.

3.2.2 Non-functional requirements

Non-functional requirements define the quality and operational constraints the system must satisfy beyond its specific functions. Firstly, the system shall generate a complete timetable for all schedulable courses under five seconds on normal operation conditions. Secondly, the system shall provide a role aware interface that allows easy navigation for the Timetable scheduler, lecturers, and students. Furthermore, the system shall be accessible from any device with a web browser and stable internet connection. Also, the system shall be structured in cleanly separated modules, known as authentication, room management, course registry, lecturer allocation, simulation engine, and analytics, so that each module can be updated without affecting the rest of the system. These non-functional specifications ensure that the system is reliable, scalable, and maintainable when used in an academic environment.

The functional and non-functional requirements and their assessment are summarized in Table 1.

Table 1. Functional and non-functional requirements assessment.

Requirement type	Requirement	Status
Functional	Authentication and role management	Achieved
Functional	Timetable simulation	Achieved
Functional	Conflict detection and logging	Achieved
Functional	Analytics and visualization	Achieved
Functional	Excel export format	Achieved
Functional	Email notifications to lecturers	Achieved
Non-Functional	System availability	Achieved
Non-Functional	Row-level security structure	Achieved
Non-Functional	Modular file structure	Achieved

3.3 Performance requirements

The developed system is expected to comply with a number of performance requirements ensuring efficiency and speed in lecture timetable generation and management. Specifically, it needs to be able to respond quickly to scheduling requests regardless of the number of lecturers, courses and lecture halls included in the process. The system should guarantee efficiency in using the Queuing Theory formula for generating schedules. It should also ensure optimal usage of lecture halls and periods available, avoiding any cases of their underutilization or overloading. Also, there is need for the system to perform efficiently despite the increase in the amount of data involved as more and more users register for different courses and constraints increase in number. In this regard, the system must also minimize the generation of conflicts when generating the final timetable, ensuring accuracy and stability in the final output.

3.4 Queuing model formulation (M/M/c model)

For generating efficient timetables in accordance with Queuing Theory, the multi-server queuing scheduling mechanism will be adopted, and it is used to represent and enhance the allocation of the requests of lecture to available lecture halls. In the proposed model, scheduling requests of courses are considered entering entities that enter and are processed by a queuing system, where they are assigned to lecture-room servers. The first “M” in the model stands for Poisson (Markovian) nature of arrival of scheduling requests, that is they arrive randomly and independently over time. The second “M” means that service times for these entities have exponential distribution, implying the allocation of lecture sessions into available periods and lecture halls. Finally, “c” indicates the number of servers, in our case it is the number of lecture halls that can process simultaneous course requests. The system can be described mathematically by a number of basic queuing parameters. The arrival rate λ is the average number of lecture scheduling requests arriving per unit of time, and the service rate μ is the average number of requests that one lecture hall can serve per unit of time. Since several lecture halls are used, the utilization (traffic intensity) of the system is

$$\rho = \frac{\lambda}{c\mu} \quad (1)$$

where c is the number of available lecture halls (servers). For the system to remain stable and avoid an infinite build-up of the queue, the following criterion should be satisfied:

$$\rho < 1 \quad (2)$$

This means that the total service capability provided by multiple lecture halls should exceed the rate of incoming lecture schedule requests. The main initiative of the algorithm used by the system is to enable incoming scheduling requests to queue whenever all lecture halls become busy, thus preventing the loss of any request. As soon as any of the lecture halls becomes free, the next available request is allocated to it.

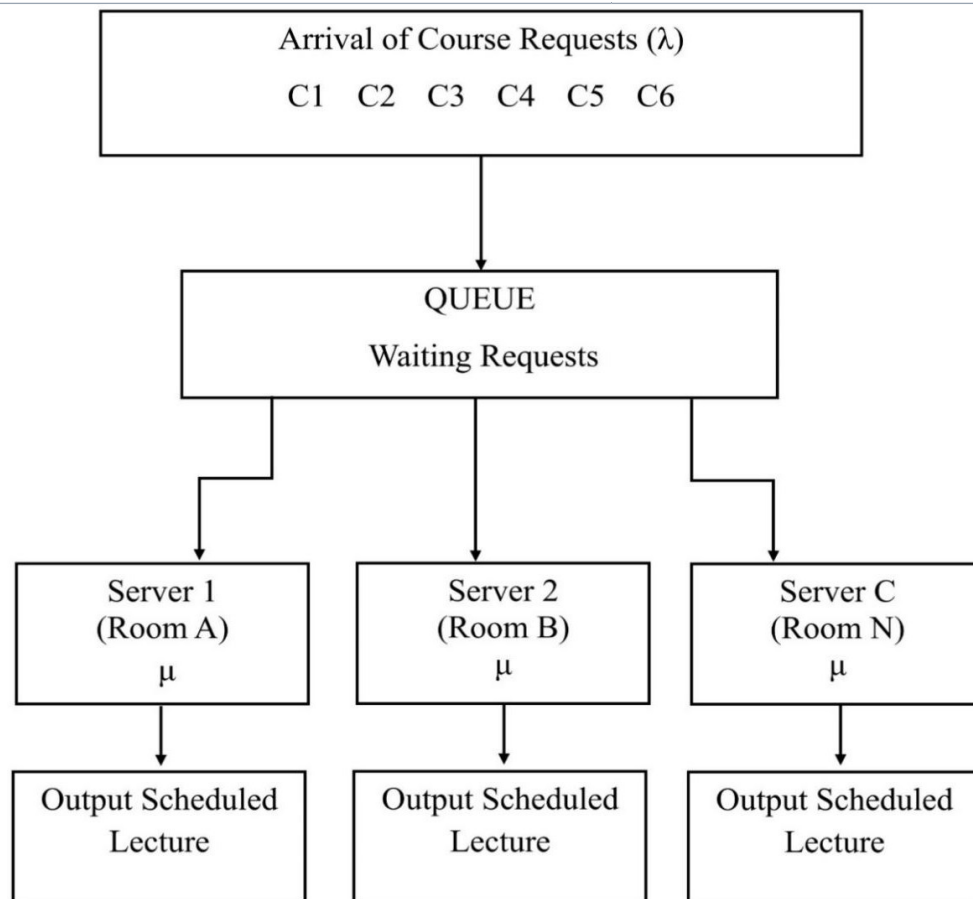


Figure 2. M/M/c lecture scheduling model.

This ensures minimal wait times and reduces conflict rates during the generation process. Conceptually, the process can be presented as shown in Figure 2, where courses come, are queued up and passed through several lecture halls (servers), before getting the finalized scheduled output.

3.5 System workflow and data flow model

These models illustrate the way in which information flows through the system from the input phase until the final output phase. For instance, in our system, the process begins by allowing authorized administrators or lecturers to input data such as course details, lecturer profiles, and lecture hall information. This data is then validated and passed into the scheduling engine, where the Multi-Server queuing model (M/M/c) is applied to make efficient allocations of lecture sessions considering the available resources. At the processing stage, the system continually checks for any possible conflicts and allocates lecture halls in an efficient manner until a final schedule is developed. The resulting schedule is stored in the database layer and eventually becomes available to system users through the interface. Figure 3 depicts a flowchart representing the complete process of operations performed by the automated timetable generation system. This flowchart provides a graphical representation of how the process starts at the system login, followed by data input, validation, implementation of the M/M/c scheduling model and ends with the final timetable schedule generation. The interaction also shows the sequential working of the system as each step leads to another step sequentially. In other words, it shows the working procedure of the system.

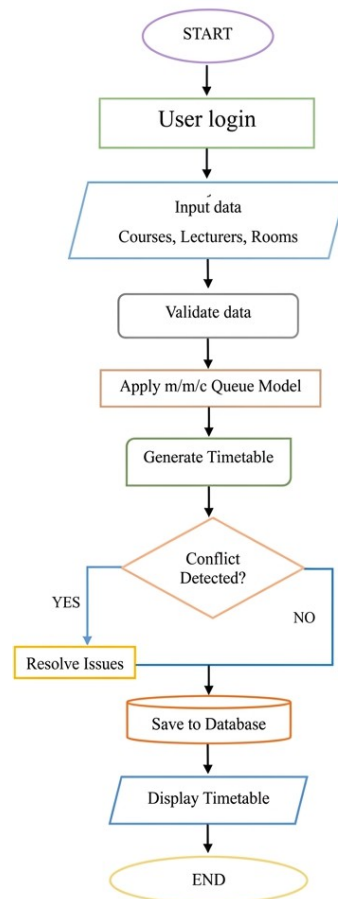


Figure 3. System workflow flowchart.

3.6 Implementation tools and technologies

The frontend of the system has been developed through the latest web-based technologies that provide easy-to-use graphical interface where administrators and lecturers can easily enter data to be processed. The backend of the system has been created on the other hand through server-side scripting languages designed to perform specific functions such as scheduling, implementing Multi-Server Queuing (M/M/c), storing data and managing databases, among others. Relational database management technologies are employed in storing and managing structured data, including creating profiles of lecturers, courses, lecture halls and schedules and visualizing analytical output like heat maps and scheduling summaries. The implementation of such technologies in the system ensures reliable, scalable and efficient operation of the system in automating timetable generation, within the university area.

3.7 System evaluation method

The proposed automatic timetable generator system will be evaluated based on its efficiency in generating schedules, reducing timetable conflicts, and maximizing resource utilization compared to existing conventional manual timetabling methods. To assess the system's performance, some key metrics will be considered, including the scheduling time, frequency of timetable conflicts, utilization rate of lecture halls, and response time during the scheduling operations. Simulation tests will also be performed by entering different amounts of data concerning lecturers, courses, and lecture halls in order to evaluate the system's performance under varying workload conditions. The outputs developed in the scheduling engine will be analyzed in order to gauge the effectiveness with which

the Multi-Server Queuing model (M/M/c) manages queues, service rates, and allocation efficiencies. Comparative analysis will be performed to evaluate the improvements from manual scheduling to the proposed automated system. Performance results obtained through experiments have been presented in terms of statistics and visualization in order to assess the reliability, scalability, and practicality of the system in real-world academic scheduling environments.

3.8 System access and user authentication

The automated timetable generation system designed and implemented in this study has been referred to as LecTabler. LecTabler system starts running from the user authentication interface, which serves as the entry point into the system for all users, including administrators and lecturers. This module ensures only valid users can access the system by authenticating users before allowing them access to the system feature, and requiring valid login credentials. The module is mainly responsible for securing access to the scheduling system by ensuring that unauthorized users cannot interfere with its operations by manipulating timetable information or other confidential data stored within. After successful verification, users gain access to the system after logging in and they are directed to their respective dashboards, depending on their roles, where they perform various activities on the system, such as management of data, viewing timetables or submitting scheduling requests.

3.9 System architecture

The system was developed with a layered approach (Figure 4), in which the layers represent the relationship between the UI, logic, and database sections. User interface represents the first stage and consists of an access channel through which administrators and lecturers enter necessary data, including course details, personal information of lecturers, and their scheduling needs.

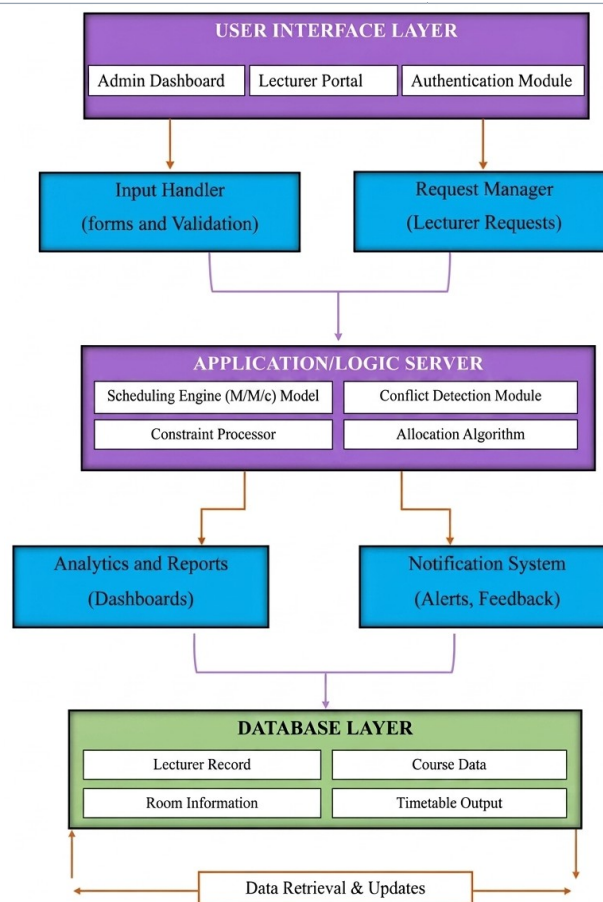


Figure 4. System architecture diagram.

3.10 Queuing model implementation (M/M/c)

Queuing modeling is a technique applied in the LecTabler scheduling engine to handle allocation requests efficiently. Specifically, the Multi-Server Queuing (M/M/c) model is employed to allocate lecture requests to multiple lecture halls in an optimized way. In this context, lecture allocation requests are random requests added to the queue, whereas several lecture halls are seen as parallel servers.

Lecture requests are processed by the system using constraints related to room availability, lecture timings, among other factors.

3.11 System modules

The automated timetable generation system consists of various interrelated modules (Figure 5) that operate in concert to generate timetables automatically. The first component in the system is the authentication module that allows for user login and access controls. The second module is the data management layer which is concerned with the creation and persistence of data such as lecturer profiles, courses, and lecture hall information. The scheduling engine module utilizes the Multi-Server Queuing model (M/M/c) to allocate lecture sessions effectively.

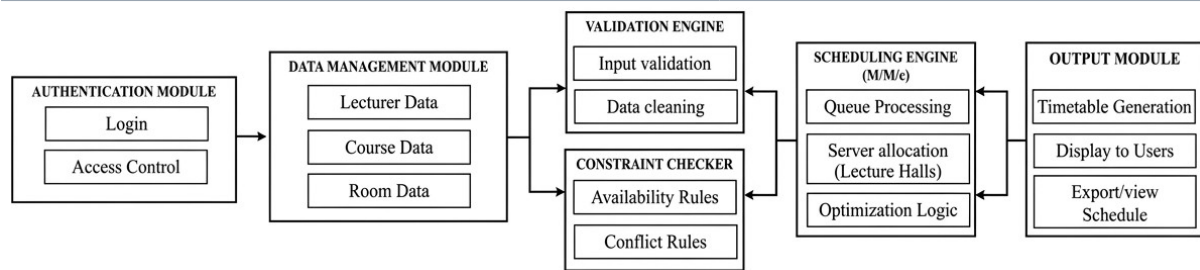


Figure 5. System module interaction diagram.

3.12 Conflict detection and constraint management

The conflict detection feature is built into the LecTabler scheduling system. This feature is designed to detect conflicts in scheduling and solve the identified clashes, if any, during the timetabling process. Conflicts might occur due to overlapping of lectures in a particular time slot, lecturer double booking, or overbooking of the lecture hall beyond its capacity. Throughout the entire scheduling process, all inputs are analyzed for conflicts against predefined constraints such as lecturer availability, hall capacity, and others pertaining to lecture hall conditions. In case conflicts are detected, the system then identifies and solves them by applying predefined rules to keep the timetable valid.

All the operations in timetable creation are guided by operational constraints and scheduling rules in LecTabler, which determine lecturer allocation, lecture hall allocation, and time-slot distribution for each course. Operational constraints include such rules as preventing overlapping of lectures for one lecturer, room capacity, and lecturer workloads according to the institution's academic regulations. Such rules should be constantly taken into consideration during scheduling to maintain consistency and prevent invalid allocations.

4 Results and discussion

It is critical to evaluate the performance of the LecTabler system in order to see whether it provides any improvements compared to traditional manual timetable creation processes. Such evaluation should include the assessment of key performance indicators that show the level of efficiency, effectiveness, and resource utilization within the system. It will be necessary to compare the system with the conventional method and automated system in terms of the identified indicators, emphasizing the enhancements achieved through the integration of the Multi-server Queuing Model and automated scheduling engine.

The time spent on scheduling is rather high in the manual scheduling system. Its main problem is related to human dependency and repeated corrections. The automated scheduling system, on the other hand, eliminates the problem, which decreases scheduling time. Also, there are no scheduling conflicts in the LecTabler system due to efficient validation and conflict avoidance implemented in this system. When it comes to resource utilization, the efficiency in the manual system is average due to improper allocation strategies, whereas the LecTabler system uses a more systematic approach and, therefore, optimizes lecture hall utilization. LecTabler is also improved in terms of flexibility, as it enables dynamic adjustments and request handling, unlike the rigid structure of manual scheduling.

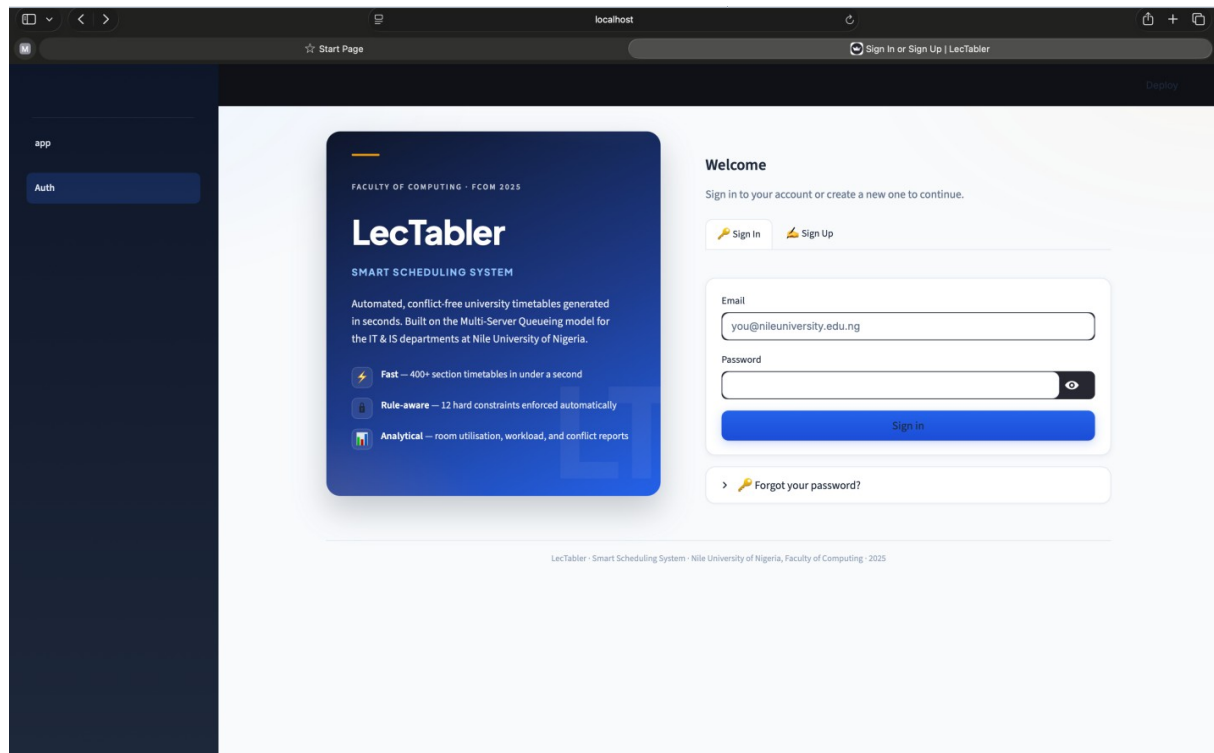


Figure 6. LecTabler authentication interface.

The login page of LecTabler (Figure 6) implements a controlled and secure environment for the usage of the system, ensuring proper role-based access throughout the application.

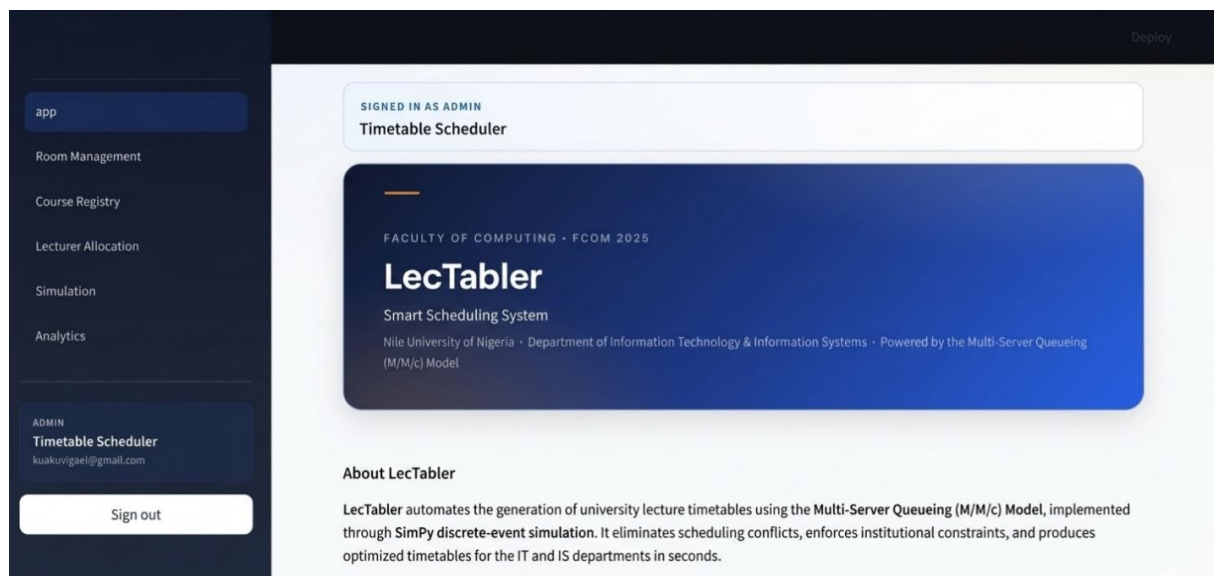


Figure 7. LecTabler main dashboard.

The main dashboard of the LecTabler system (Figure 7) acts as the entry point for administrators after successfully logging into the system.

Figure 8. Room management interface.

Room allocation and resource utilization are the functions performed by the room management component (Figure 8) in LecTabler in order to achieve efficient scheduling.

Figure 9. Lecturer management and data entry interface.

The lecturer management interface (Figure 9) is mainly involved in creating and managing lecturers and courses in the LecTabler system.

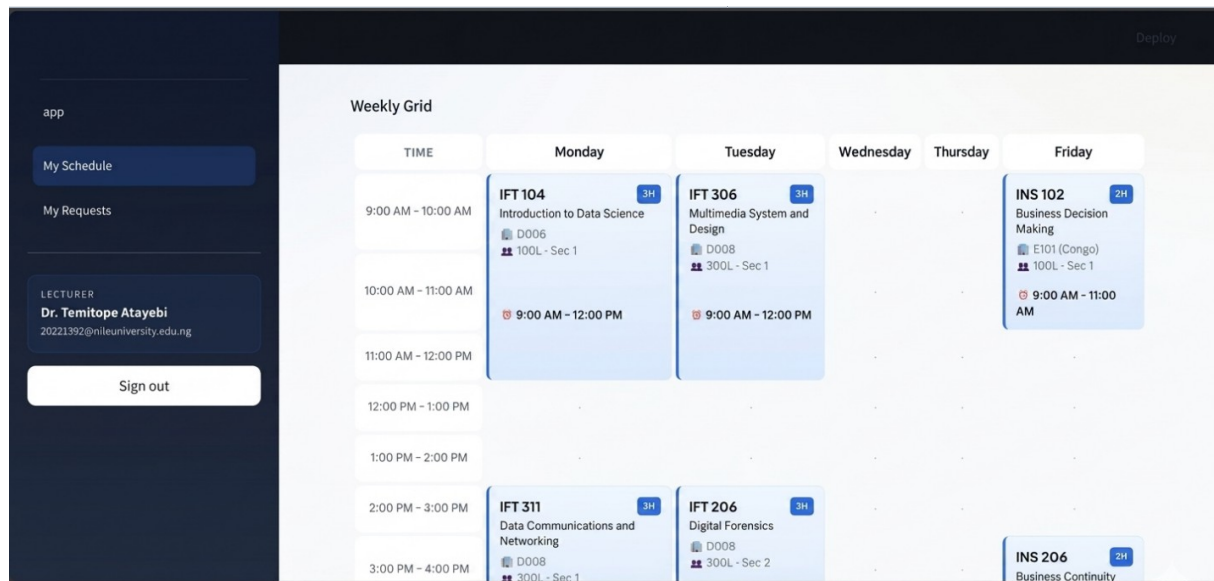


Figure 10. Weekly timetable output.

The automated generated weekly timetable for academic staff members is illustrated in Figure 10, which depicts a timetabling output showing allocation of courses, time and venues respectively.

Figure 11. Simulation queue processing and conflict status.

In the simulation process (Figure 11), conflicts are continually monitored, and their resolution is implemented while ensuring scheduling results conform to available resources and logic.

5 Conclusion

This study focused on developing a solution for designing an automated University Timetable Generation System using the Multi-server Queuing Model (M/M/c). The existing timetabling approaches employed in most universities have proven inefficient due to their tendency to produce scheduling conflicts, inappropriate allocation of resources, and time-consuming adjustment periods. To address these problems, a systematic process was designed where the lecture requests were viewed as a queuing system with lectures being scheduled according to the arrival times of the

requests at multiple service centers represented by lecture halls. Through the application of data management, scheduling, conflict detection, and optimization techniques, the proposed system ensured efficient allocation of academic resources. This study shows that adopting the systematic queuing model greatly improves the timetabling process through increased efficiency, reduced human intervention, and improvement of the overall efficiency of timetable generation within a university environment.

Based on the findings, the study concludes that the application of Queuing Theory, especially the Multi-Server Queuing (M/M/c) model, in University Timetable Generation is a systematic and efficient means of addressing scheduling problems experienced in academic institutions. The conventional timetabling approaches used by many institutions were determined to be inefficient because of their dependence on manual scheduling, which has been known to generate scheduling conflicts, unequal workload among lecturers, and improper use of lecture halls. Unlike most existing scheduling approaches that consider lecture requests deterministically, the proposed approach considers lecture requests to be stochastic and schedules them differently according to the service center that represents different lecture centers. Therefore, the new model enables the handling of scheduling in an enhanced manner. In general, the application of the queuing model has been determined to be very effective in minimizing problems associated with inefficient scheduling, including manual timetable making, conflict possibilities, and other challenges which are likely to impact the effectiveness of academic operations. This is to mean that the proposed model can serve as an effective tool for improving the efficiency of scheduling operations in educational institutions. First, the automated scheduling approach has been established to minimize the time needed for timetabling operations, besides ensuring conflict-free timetables owing to the validation of the rules applied. Second, the employment of the rule-based system together with the queuing model allows for sufficient consideration of such factors as lecturer availability, course needs, and lecture hall capacity. Third, the scalability of the system ensures its efficiency even when the number of lecturers, courses, and lecture halls is on the rise. Hence, based on the findings, the queueing-based automated scheduling model applied during the study is effective in timetabling operations in universities. Furthermore, the system increases efficiency by enhancing precision, lowering administrative costs, and facilitating quicker decision-making.

Based on the findings, the following recommendations are made in order to improve the effectiveness of the created timetables and promote their usage by educational institutions. They include suggestions made from the experience gained in creating a queue-based timetable, the limitations of traditional time tabling procedures, and possible improvements to the implemented model.

- 1. Adoption of the automated timetable system in various institutions:** The use of an automated timetable generating system is highly recommended in place of manual or semi-automatic time tabling methods because it will help avoid scheduling conflicts, improve resource utilization, and reduce administrative burden connected with timetable creation and adjustment. Large adoption will ensure consistency and standardize the scheduling procedure throughout departments.
- 2. Integration with existing institutional management systems:** It would also be helpful to connect the developed software with the existing management platforms in universities, which are used for course registration and student databases.
- 3. System maintenance and training for administrative staff and lecturers:** It is recommended that the system undergo constant maintenance to ensure smooth performance and efficiency regarding its security and performance.

Acknowledgment

I extend my heartfelt gratitude to God Almighty for His guidance and blessings throughout the duration of this project. I am deeply grateful to my HOD, Prof. A. Prema Kirubakaran, for her unwavering support, insightful feedback, and expertise, which significantly enhanced the quality of my research and writing. I also wish to express my sincere appreciation to my supervisor, Prince Eru Akwuma Nathaniel (mNCS, FSG, MAAU, UAN, PUAN), for his consistent support, mentorship, and valuable insights throughout this journey.

References

- [1] Shu J, Zhang P, Sheng J, Liu Z. Node selection model for block transmission in a blockchain network. In: *2021 International Conference on Cyber-Physical Social Intelligence (ICCSI)*; 2021.
- [2] Nontananandh S, Kuwahara S, Shishido KI, Inazumi S. Influence of perforated soils on installation of new piles. *Applied Sciences*. 2022;12(15):7712.
- [3] Azeez M, Mariam AS. *Development and implementation of an automated timetabling system for tertiary institution*. International Research Journal of Modernization in Engineering Technology and Science. 2025.
- [4] Alshayeji MH, Sindhu SCB, Abed S. Viral genome prediction from raw human DNA sequence samples by combining natural language processing and machine learning techniques. *Expert Systems with Applications*. 2023;218(19):119641.
- [5] Alonge O, Sakpere W, Adediran E. *An automated timetable scheduler using NSGA II for optimized scheduling in educational institutions*. 2024.
- [6] Marwah MA. *Optimized automatic course timetabling service architecture for integration with vendor management systems*. International Journal of Advanced Computer Science and Applications. 2022.
- [7] Ali A, et al. *Automated timetable scheduler*. 2025.
- [8] Adedayo S, Adebimpe E, Samuel A. *Development of an automated timetable generator using genetic algorithm*. Lautech Journal of Computing and Informatics (LAUJCI). 2023;2(1).
- [9] Alghamdi H, Alsubait T, Alhakami H, Baz A. A review of optimization algorithms for university timetable scheduling. *Engineering, Technology & Applied Science Research*. 2020;10(6):6410–6417. <https://doi.org/10.48084/etasr.3832>
- [10] Efrosinin D, Vishnevsky V, Stepanova N. Optimal scheduling in general multi-queue system by combining simulation and neural network techniques. *Sensors*. 2023;23(12):5479.
- [11] Dudin A, Dudin S, Dudina O. Analysis of a queuing system with mixed service discipline. *Methodology and Computing in Applied Probability*. 2023;25(2):57.
- [12] Kang K, Doroudi S, Delasay M, Wickeham A. A queueing-theoretic framework for evaluating transmission risks in service facilities during a pandemic. *Production and Operations Management*. 2023;32(5):1453–1470.
- [13] Pang HJ, Cheng JK, Liu Y, Pahang M, Abdullah AS, Tun P, et al. A discrete event simulation approach for efficient student enrolment processes at Universiti Malaysia Pahang Al-Sultan Abdullah. *International Journal of Industrial Management*. 2025;19(3):116–128.
- [14] Mahlous AR, Mahlous H. Student timetabling genetic algorithm accounting for student preferences. *PeerJ Computer Science*. 2023;9:e1200.
- [15] Almeida J, Figueira JR, Francisco AP, Santos D. *A hybrid meta-heuristic for the generation of feasible large-scale course timetables using instance decomposition*. 2023.
- [16] Bryan J. A constraint-based automated timetable generator for educational institutions. *i-manager's Journal on Software Engineering*. 2025;20(1):5.
- [17] Mallari CB, San Juan JL, Li R. The university coursework timetabling problem: An optimization approach to synchronizing course calendars. *Computers & Industrial Engineering*. 2023;184:109561.
- [18] Dudin A, Nazarov A, Moiseev A, editors. *Information technologies and mathematical modelling*. Queueing theory and applications. Communications in Computer and Information Science, vol. 1803. Cham: Springer; 2023.

-
- [19] Emergent Mind. *Discrete event simulation*. December 2025. <https://www.emergentmind.com/topics/discrete-event-simulation-des>
- [20] Lv W, Liu R, Yan F, Wang Y. Discrete event simulation-based analysis and optimization of emergency patient scheduling strategies. *Healthcare (Basel)*. 2025;14(1):99.
- [21] Rahimifard A, Nakhai-Kamalabadi I, Khalili-Damghani K, Raissi S. Simulation-based framework for stochastic multi-mode resource-constrained project scheduling. *MethodsX*. 2025;15(1):103496.