INTELLIGENT UNIVERSITY STUDENT-STAFF APPOINTMENT MANAGEMENT SYSTEM (GO-AMG)

BY

ANYANWU FAVOUR CHIDALU

GOU/U22/CSC/907

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY,

GODFREY OKOYE UNIVERSITY, ENUGU STATE

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCE (B.SC) DEGREE IN COMPUTER SCIENCE

SUPERVISOR: DR. U. F. OKEBANAMA

MAY, 2026

APPROVAL PAGE

This research, titled "Intelligent University Student-Staff Appointment Management System (GO-AMG)," has been assessed and approved by the Department of Computer Science in the Faculty of Computing and Information Technology, Godfrey Okoye University, Enugu, Nigeria.

DR. U. F. OKEBANAMA
Supervisor ....
Signature Date

External Examiner ....
Signature Date

Dr. U. F. OKEBANAMA
HOD, Computer Science ....
Signature Date

Prof. I. A. AJAH
Dean, Faculty of Computing
and Information Technology ....

Signature Date

CERTIFICATION

I certify that I completed the research included herein and that it was primarily based on my own observations and investigation. Consequently, I will fully accept the University's decision if I am found to have breached any academic integrity guidelines.

____

ANYANWU FAVOUR CHIDALU

GOU/U22/CSC/907

DEDICATION

This work is dedicated to my family, whose unwavering love, encouragement, and prayers sustained me through every stage of this project, and to all students of Godfrey Okoye University, for whom this system was designed.

ACKNOWLEDGEMENTS

First and foremost, I offer my deepest gratitude to Almighty God for His abundant mercies, inspiration, and grace throughout the duration of this project. I wish to express my profound appreciation to my supervisor and H.O.D, DR. U. F. Okebanama, for his invaluable guidance, constructive criticism, and patience throughout the research process. I also acknowledge the entire faculty of computing and information technology at Godfrey Okoye University, especially my lecturers, for the quality of education and mentorship provided throughout my academic journey. I also thank specifically lecturers who made contributions to this project through their constructive criticism, advice, and corrections. My sincere gratitude goes to my parents and family for their consistent support, prayers, and encouragement, which kept me motivated through the most demanding phases of this project. I also thank my friends whose support, contributions, and advice.

ABSTRACT

The persistent inefficiency of manual student-staff appointment scheduling in Nigerian universities results in prolonged waiting times, wasted student journeys, unstructured office crowding, and the absence of any data-driven mechanism for managing staff accessibility. This research addresses these challenges through the design and development of the Intelligent University Student-Staff Appointment Management System (GO-AMG), a centralized, web-based intelligent scheduling platform for Godfrey Okoye University. The system implements a two-agent model comprising a human agent, represented by the staff member who controls real-time availability through a toggle mechanism, and an AI agent that records every toggle event, trains a Random Forest Classifier on the accumulated historical data, and serves availability predictions through both a visual interface and a conversational chatbox. Key system features include a priority-based virtual queue with drag-and-drop reordering, mandatory urgency justification for urgent and critical appointments, real-time queue position tracking, instant availability alerts, an anonymous star rating system, and a comprehensive administrative analytics dashboard. The AI model achieved a classification accuracy of 73.74 percent on a noise-injected test set. All twelve system test cases passed, and the user interface was confirmed to be functional across all three roles. The results demonstrate that the GO-AMG system successfully addresses the identified appointment scheduling challenges and represents a meaningful contribution to intelligent administrative systems in higher education.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Education is still one of the greatest assets a person can acquire and every part of the educational process, especially student-staff interaction, should be attended to. It is a basic right of students to not only be able to access academic and administrative staff but also receive timely guidance is a basic right that will support their academic success in any university. Students often need to interact with staff members including academic advisors, project supervisors, bursary officers, examinations officers, ICT officers, and student welfare counsellor for courses' registration, final year project supervision, verification of school fees, NYSC mobilization, request for transcripts, among others and hostel allocation.

Usually, the appointment of these visits in Nigerian universities has been done manually and informally which involves physical visits, telephones, WhatsApp messages and the use of informal intermediaries like the secretaries of departments. With the increasing number of students and the digital expectations of a generation of students, these methods are becoming less effective. Students are unable to depend on staff to be available and as a consequence of this, journeys are wasted, waiting times are extended and students often become frustrated. Constantly having to deal with unscheduled interruptions with staff and having no effective management of their time. There is no system in place across the institution to capture data on appointments, or to make evidence-led decisions on student support.

This research addresses this clear and pressing need by introducing the Intelligent University Student-Staff Appointment Management System (GO-AMG), which is a centralized and intelligent, web-based system that utilizes Artificial Intelligence (AI) for real-time availability management and predictive analytics to revolutionize the appointment scheduling and management process at Godfrey Okoye University.

1.2 Statement of the Problem

The following are particular problems that emanate from a lack of an appropriate and intelligent appointment management system in Nigerian Universities:

1. Students often stand outside in the office for hours and do not know if and when they will be able to see the office.

2. No effective system is in place to monitor, analyse and improve student-staff interactions; no data-driven decision making process for staff accessibility takes place.

3. Students are forced to wait for a long time for staff member whose presence is not known and this results in significant loss of time that could have been spent on academic work.

4. Students waiting in staff offices without knowing where they are in the queue or how long they will have to wait leads to disorganisation and an unpleasant experience for the student.

5. Small office spaces place greater stress on students with health issues and physical mobility problems due to overcrowding.

6. Staff cannot proactively communicate their availability and end up in lots of unplanned communication interruptions and are unable to manage workload effectively.

7. Off-campus students who come to the university at their own expense are not guaranteed to see the person they want, and the potential loss of money, time and effort is lost.

8. There is no intelligent, centralized, flexible, efficient and data-driven system for managing appointments between students and staff to meet the needs of students/staff and the institution.

1.3 Aim

The aim of this project is to develop an intelligent university student-staff appointment management system.

1.4 Specific Objectives

The specific objectives of this project is to do the following:

1. Identify gaps in appointment scheduling process in Godfrey Okoye University.
2. Design an intelligent university appointment scheduling system that enables a seamless booking of appointments between students and staff.
3. Implement an intelligent appointment scheduling system that solves the identified problems using virtual queue.
4. Test and evaluate the efficiency of the developed system.

1.5 Significance of the Study

This research work will benefit a lot of stakeholders in Godfrey Okoye University and they include :

1.5.1 Students

This system will have the following meaning for students:

- The ability to check the real-time availability of staff before embarking on a journey, along with AI-powered predictions about likely staff availability; even for off-campus staff planning a journey.

- Make smarter visits with AI predictions of when a staff member is likely to be available.

- Ability to receive instant notifications via in-app alerts when a staff member's toggle becomes active, when an appointment is disrupted or availability changes.

- The ability to submit anonymous ratings and feedback after completed appointments, thus helping the improvement of the services offered by the institution.

- Greatly cut down stress and anxiety when it comes to staff availability and long queues.

- Enhanced overall academic achievement and wellbeing due to staff guidance and support.

More focused and planned staff interaction during appointments, thus improved quality of attention paid to children.

1.5.2 University Staff

- The following will be the significance of this system to university staff:
- Full control over their appointment schedule through the real-time availability toggle mechanism, eliminating informal multi-channel requests.

- Significantly reduce administrative burden resulting from the elimination of the informal appointment system.

- Protection of privacy and focused working time by preventing unscheduled interruptions and unmanaged office crowding.
- Improved quality of interaction with students as a result of better preparation and a more organized appointment environment.

1.5.3 Educational Institution

The following will be the significance of this system to the educational institution:

•Access to a comprehensive administrative analytics dashboard displaying system-wide appointment data, peak booking times, no-show statistics, AI pattern insights, and student rating feedback.

•Positioning of the institution as a technologically progressive university that prioritizes student welfare and operational excellence through data-driven administrative management.

•Ability to identify and address bottlenecks in the student support process based on no-show statistics and appointment outcome data.

•Improved student performance and satisfaction, leading to an enhanced institutional reputation and competitive standing.

•Support for institutional accountability through transparent records of student-staff interactions and appointment outcomes.

1.6 Scope

The scope of the system is Godfrey Okoye University ,Enugu, Nigeria

1.7 Limitations of the Study

This proposed system is subject to certain limitations and potential challenges, including:

I. Connectivity: As a web-based application, the system requires a reliable internet connection. Some regions with poor or unreliable internet connectivity may prevent regular usage, especially for learners in isolated areas.

II. Data Availability and AI Predictive Accuracy**:** The AI agent's ability to predict accurately relies on having enough historical toggle activity data accumulated. In the early stages of implementation, the scarcity of data could restrict the accuracy of predictions. Performance improvements are projected to occur gradually as data collection continues to increase.

III. User Adoption and Digital Literacy: Some staff and students may resist transitioning from familiar manual methods. Varying levels of digital literacy could affect the utilization of advanced features, making an orientation programme necessary at deployment.

IV. System Availability: The system is dependent on server uptime and network infrastructure. Any interruption to the hosting environment could temporarily affect accessibility for all users.

1.8 Definition of Terms

Administrative Analytics Dashboard: A secure, administrator-only interface presenting visual summaries of system-wide appointment data to support institutional decision-making.

I. **AI Agent:**An autonomous software component that records appointment history, identifies patterns in staff availability, predicts the most likely times a staff member will be available, and recommends best time slots for booking

II. **Appointment Priority**: A classification indicating appointment urgency: normal, urgent, or critical. Urgent and critical appointments require a written justification that staff members can read before deciding to reprioritize the queue.

III. **Human Agent:** The university staff member who participates in the scheduling process controls their real-time availability status through a toggle mechanism that directly determines when students can book appointments.

IV. **Intelligent Scheduling System:** A software application that automates and optimizes the appointment booking process using advanced algorithms and Artificial Intelligence to reduce conflicts and enhance efficiency.

V. **No-Show Tracking**: A system feature that records instances where a student fails to appear for a confirmed appointment, with automatic suspension of booking privileges for students who exceed the no-show threshold.

VI. **Predictive Analytics**: The use of statistical algorithms and machine learning to analyze historical data and forecast future availability patterns, enabling the system to recommend optimal booking times to students.

VII. **Queue Management**: A system feature that assigns each student a numbered position upon booking, with a drag-and-drop reordering interface available to staff based on appointment urgency justifications.

VIII. **Random Forest Classifier**The machine learning algorithm used by the AI agent. An ensemble method that builds multiple decision trees and combines their outputs to produce stable predictions of staff availability probability.

IX. **Staff Availability Toggle**: A real-time on/off control on the staff dashboard that instantly communicates the staff member's availability to the entire student population. When active, students may book; when inactive, students are directed to AI predictions and reminders.

X. **Toggle History**: A database table that records every availability toggle event with its day, hour, week, and month. This table serves as the primary training dataset for the AI model.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

In this chapter, a comprehensive review of the existing literature on the development of the GO-AMG system is presented. With the increasing need for effective, technology-based administrative systems in higher education, there has been extensive research on automated scheduling, online management systems, and the use of artificial intelligence in higher education operations. This chapter analyzes scholarly articles and research papers published in reputable journals and conferences, which reveal the strengths and weaknesses of current systems, the gaps in the literature, especially in the Nigerian university environment, and lays the groundwork for this research.

2.2 Theoretical Background

The central theories, principles and technical concepts that underpin this research are discussed. The background includes an overview of the concept of appointment scheduling, the history of scheduling systems, algorithms and machine learning models used, and the technologies that allow for an intelligent web-based scheduling system to be made possible.

2.2.1 Providing an overview of the appointment scheduling process

An appointment is a formal meeting between people for a clear reason that will be finished at a particular time. In service-systems, like a hospital, university or government agency, there is a limited amount of staff time and a large number of demanding clients or users, and appointments are the main way of controlling the amount of time allocation. The problem of scheduling is, in the field of operations research, the allocation of jobs or tasks to processors or servers over time under constraints, with the objective of optimizing one or more objective (Pinedo, 2016).

The “server” in a University setting would be a staff member and the “job” would be the student visit. The constraints are based on the limited working hours of the staff members, different lengths of time in which different types of appointments are held, different levels of urgency of different student needs and the institutional mandate of equal access to these academic supports for all registered students. If there is no system in place to deal with these constraints, then the process is inefficient, inequitable and frustrating for everyone involved.

The theory of appointments is related to the mathematical theory of queues, first developed by Erlang (1909), which is used to study waiting lines and service systems. Modern scheduling algorithms use this theory coupled with computing power to maximize the effective use of services, minimize idle time and keep the number of people who will be waiting for a service down to a minimum.

2.2.2 Types and Nature of Appointment Systems

There are two major dimensions to the appointment systems: the means of access to the appointment system, and the intelligence that goes into the scheduling process. Appointment systems can be either walk-in (users come without prior appointment, and stand in line) or booked (user books an appointment in advance). Walk-in systems are easy to operate, yet create unpredictable lines, ineffective journeys and inefficient time utilization. These are overcome by booking systems, which need to be available and the staff to be able to communicate their availability.

Depending on the degree of intelligence, scheduling systems can vary from static booking systems with fixed time slots, to dynamic systems that adapt to any situation, to intelligent systems that use algorithms and machine learning to understand from past booking patterns and improve allocation over time (Cayirli & Veral, 2024). The GO-AMG system is part of the highest tier, which includes real-time availability signaling by the human agent and AI predictive availability based on the agent's historical toggle data.

Another significant difference is the online vs. offline appointment system. Web-based or mobile-based online systems are the norm in today's service world, as they allow agents to book without physical contact, provide access around the clock, and automatically notify and remind travelers. Whether in healthcare, government or education, shifting appointment systems to the web has always been linked with user satisfaction improvements, fewer no-shows, and higher operational efficiency.

2.2.3 Traditional and Manual Methods of Appointment in Universities

In Nigeria's Universities, the appointment system between students and staff are still mostly manual and informal, using secretarial intermediaries, physical visits, phone calls, and whats-app messages. Such strategies are not suitable or effective on the scale of a modern university. Long waiting times for pupils and staff are experienced and pupils are unsure when staff is available; staff are interrupted with little structure of their time. Since there is no central record, there is no way for institutional administrators to see the patterns of appointments, rate of non-attenders, or accessibility of various staff members.

The findings from the study carried out by Idowu et al. (2022) showed that although web-based scheduling systems are technically viable, they yield tangible results in improved operational efficiency when introduced even in Nigerian institutions. The problem isn't technology skills, it's the lack of specialized solutions for the academic appointment space. The GO-AMG system is developed by taking into account this gap within the Nigerian University environment.

2.2.4 Intelligent Scheduling Systems

An intelligent scheduling system is a system that breaks from the simple rule based approach of booking and uses computational techniques like constraint satisfaction, optimization algorithms or machine learning to make better scheduling decisions. Intelligent systems are systems that can adapt to changes, learn from past data and make recommendations instead of just following instructions.

The use of intelligent scheduling got started in industrial and manufacturing applications, where job shop scheduling and resource allocation have been a subject of research since 1960s. The scope of the research has gradually been expanded to healthcare appointments, where it is very helpful to optimize as it directly affects patient outcomes. Ahmadi-Javid et al. (2020) conducted an extensive survey of research in intelligent scheduling, which concluded that it can be applied across all the research areas surveyed, and yield consistent improvements in throughput, minimization of conflicts, and user experience.

The main use of intelligent scheduling in educational institutions has been in the area of timetabling and course allocation. This research is directly relevant to the newer and less studied aspect – application of student-staff appointment management. The GO-AMG system utilizes AI to create intelligent scheduling by learning from each and every toggle event, and delivering personalized availability forecast for every individual member of staff based on their personal behavioural pattern.

2.2.5 Algorithms Used in Scheduling Systems

Numerous computational algorithms have been used in scheduling systems, some with specific properties that are well suited to different aspects of the scheduling problem. First generation automated scheduling systems based on heuristic algorithms and rule-based methods that give good-enough solutions quickly but do not guarantee optimality. Such as First-Come-First-Served (FCFS), Shortest Job Next (SJN) and priority based scheduling, where jobs are scheduled in a certain order according to some criteria of importance that are previously defined. These are simple, easy to compute, but cannot be flexible, and cannot learn from experience.

Constraint Satisfaction Algorithms represent the scheduling as a collection of variables and constraints, and look for assignments that satisfy all the constraints. These are applicable during timetabling and resource allocation when hard constraints like room capacity or staff availability have to be respected. Integer Linear Programming (ILP) and other optimization methods aim to obtain the mathematically optimal solution and are computationally expensive for larger problems.

Evolutionary and metaheuristic methods like Genetic Algorithms, Simulated Annealing, and Particle Swarm Optimization are able to search through large solution spaces efficiently and are appropriate for many complex scheduling problems having numerous conflicting objectives. They have successfully been used in multi-resource scheduling, nurse rostering, and examination timetabling problems.

The GO-AMG system scheduling problem is not an assignment problem, but a prediction problem: If a user asks for staff member A at a certain day and time, what are the chances of

A being available? This is a supervised classification problem, in which machine learning algorithms can be applied directly. Random Forest Classifier was chosen because it can be trained using multiple features and avoids overfitting, unlike other classifiers, and gives probabilities instead of only binary output, and will work good even with structured tabular data without requiring large datasets. The properties also make it very suitable for the availability prediction task, in which the number of the events in the training dataset is increased by each new toggle event, and in which early predictions must be reasonable although the dataset is not large.

2.2.6 The concept of machine learning models in scheduling

One of the breakthroughs in machine learning for scheduling systems has been the use of a predictor that can get better as more scheduling data becomes available. Logistic regression, decision trees, support vector machines, neural networks, and Random Forests or Gradient Boosting are the most common models used for scheduling purposes.

Logistic Regression is a linear classification model that works to predict a binary outcome. Fast and highly interpretable, it is a widely-used baseline model. It does, however, impose the constraint of a linear relationship between features and log-odds of the outcome, which may be too limiting for more complex scheduling patterns having non-linear interactions.

Decision Trees divide the feature space into regions, and make predictions based on the majority class in each region. They are easy to interpret and can be used to model non-linear relationships but can overfit noisy data. This can be reduced by pruning and depth constraints but these come at a cost of reduced predictive accuracy.

Support Vector Machines (SVMs) are used to compute the hyperplane that is as far as possible from the classes using a high dimensional feature space. They are also generalizable to unseen data and are robust to noise – Sharma et al. (2017) obtained good no-show prediction results from SVM on a set of more than 110,000 medical records. However, SVMs are rather sensitive to hyperparameter optimization and are not provided with an inbuilt probability estimator.

Random Forest, as introduced by Breiman (2001), creates an ensemble of decorrelated decision trees by randomly selecting variables to split on and by using bootstrap aggregation (bagging). The ensemble uses the predictions of all the trees and results in a more stable and accurate prediction than any single tree. Each tree looks at a different subset of data and features at each node, so the trees are different and the errors offset. All these are strengths in the context of GO-AMG availability prediction: Random Forest is highly resistant to overfitting, can handle missing values, and can estimate feature importance. It allows the system to report availability as being "78% likely available" rather than being binary (yes/no): the model can return class probabilities (the proportion of trees that voted for that class).

Neural Networks and deep learning models have shown state-of-the-art results for complicated scheduling problems in areas with very large datasets. They are, however, much more expensive in terms of computation, tend to have fewer characteristics of interpretability, and need a much larger training set to be trained before they are deployed, which is not the case for the GO-AMG context where the training set is continuously expanded from an initial empty set, and interpretability is crucial for user trust.

2.2.7 AI Agents and Predictive Analytics

An AI agent is a standalone piece of software that can perceive its environment and take action toward preset goals with very little of human intervention. The AI agent has four functions in the GO-AMG architecture: 1. The AI-agent provides the back-end middleware with the recorded toggle events, 2. The AI-agent trains the Random Forest Classifier with the toggle events history data, 3. The AI-agent feeds availability predictions to a dedicated Flask microservice, 4. The AI-agent pushes predictions to a conversational natural language chatbox and a visual prediction modal, both shown to students. The agent is not limited to statistical prediction, but can also be used with the domain of natural language processing that translates the user's question, e.g., "Is Dr. Obi available on Tuesday at 10am?" into a probability estimate that is calculated from the trained model.

Predictive Analytics in Scheduling is a method for producing probability forecasts from patterns in past schedules. The features used in the GO-AMG system predict the key factors of staff availability behaviour: the pattern of availability for a particular identity, the day-of-week pattern, the time-of-day pattern, and the seasonal pattern. These capabilities enable the model to produce more sophisticated predictions, like "This staff member is usually available on Tuesdays from 9am to 11am, and is less likely to be available on Monday mornings."

2.2.8 Queue Management in Service Systems

Queue management refers to the techniques used to manage the sequence, speed and wait of customers or users in a service. Good queue management will decrease the waiting time, enhance service delivery fairness and increase overall user satisfaction. The traditional manual method of managing student-staff appointments has no queue management in universities: students simply gather physically, there is no formal priority, and the priority is informal and unstable.

Digital queue management systems overcome these shortcomings by assigning virtual queue numbers, providing communication of estimated waiting times and allowing priority adjustments on specific criteria. Lester et al. (2022) found that implementing a web-based queue management system at a university registrar's office could greatly enhance both the efficiency of the service and the satisfaction of students. The GO-AMG system takes this approach further by introducing an Urgency Justification feature to enable staff to consider the urgency and then prioritize queue reordering, while also providing a drag-and-drop interface for easy and efficient reordering.

When a user clicks the meeting ended button, the automatic queue advancement kicks in, removing the need for manual queueing. The special notification to student #1 guarantees that the appointment process is not delayed after the meeting, hence the most efficient use of the staff persons time.

2.2.9 The development of notification technologies and web-based systems.

A web based system is accessible via a browser via the internet and allows the platform independent data management without the need to install any software. This architecture provides on-campus accessibility from anywhere, especially for students who are not on campus. The three-tier web architecture with separation between presentation layer, application logic, and data storage provides scalability, maintainability, and facilitates system updates without impacting end-users.

To be responsive, a scheduling system has to include notification technologies. In-app notifications can be sent through the web interface, email notifications through users' institutional or personal email accounts, and SMS notifications can be sent to users without internet access. The GO-AMG system includes in-app notifications on all critical events of the system such as availability changes, booking confirmations, queue position changes, queue position 1 notifications and post-appointment rating prompting. Every notification will contain a context, including staff name, so users can respond to the notification without having to go to other parts of the system.

The literature review will be presented in the form of a review of related works.

2.3 review of related works.

This section provides a critical review of past research projects, academic articles and technical reports closely related to this study.

Qaffas (2019) designed an online appointment web-based management system for higher education, which enables students to make appointments with academic advisers through an internet based platform with email reminders and experimental verification. The system successfully proved that the appointment booking system via the web is technically possible in an academic environment and that reminders by e-mail improve no-shows. But this was purely manual availability management, relying on the availability status of the staff and not relying on any AI or predictive capabilities, so was not very effective in environments with highly unpredictable staff availability.

In this paper, Saraswathi et al. (2023) developed an Android Application based Smart Appointment System for academic counselling which necessitates both the parties to have an Android device. The system was used to show the feasibility of mobile academic appointments, and to achieve an increase in student engagement with counselling services, but was only designed for Android mobile apps, did not include a web interface that could be accessed via any browser, and did not include any intelligent scheduling or prediction.

Abu et al. (2022) gave a review and design of the Web-based booking appointment system in academic institutions that can be accessed from the internet and mobile devices. The integrated workflow was well structured and clearly outlined the steps of the booking process from request to confirmation. Although the workflow was well thought out, there was no AI or machine learning aspect to the system, and the availability of staff was handled statically, with no provision for updating real-time, meaning students still had no way of knowing if a staff member was truly available to them.

Suresh et al (2019) created an appointment booking application across several services with the Ionic framework, Firebase, and NodeJS and hosted it on the web. The system proved to be an effective real-time data synchronizing system with various service providers.The system showed real-time data synchronizing system with various service providers effectively. It was intended for all commercial use, but did not include academic purpose categorization, priority systems for urgent cases, and AI-driven scheduling intelligence.

Another early application of Intelligent algorithms in web-based scheduling was the web-based hospital appointment scheduling application developed by Gharibeh et al. (2019) using a neural network to make better appointment scheduling decisions and minimise patient wait times. The study proved that intelligent slot allocation was measurable in terms of waiting time. The system was mostly developed for healthcare applications and needed significant training data so that it was not accurate in institutional settings when it was first introduced.

The approach used in this study by Williams et al. (2018) was a mixed-method design comprising SPSS analysis and NVivo analysis in a university health care context over a one year period to examine the impact of an online appointment scheduling system on improving quality of working life. The study was valuable and offered empirical evidence supporting the beneficial impact of digital appointment systems on staff effectiveness and work-life balance. The scope of the study was restricted to healthcare scheduling in the context of a university and excluded the management of academic student-staff appointments as well as AI-based predictions of availability.

Sharma et al. (2024) created a machine learning model with more than 110 thousand records to predict the availability and identify the no-show pattern using Support Vector Machine classification. Results showed high accuracy of the predictions and showed that SVM is a good discriminator between high risk no show and compliant patients. The emphasis was only medical and the reliance on a very big existing set of data meant the model could not be applied to early stage university deployments where historical data needs to be accrued over time.

Johnson et al. (2023) illustrated that AI-based predictive scheduling significantly outperforms human scheduling in minimizing scheduling conflict and enhancing the matching of availability in service-oriented industries. The study revealed that predictive models can cut down scheduling conflicts up to 34 per cent compared to manual allocation. The framework was not specific to education and did not include any toggle switch for human availability in real time, or a conversational AI interface.

Nallathamby et al. (2021) showed how Hadoop, MapReduce, and a Random Forest Regressor can be used to predict doctor ratings based on appointment wait times, highlighting the importance of harnessing big data technology to optimize appointment scheduling. The research demonstrated the success of predictive modelling of waiting times in resource allocation decisions in outpatient settings. The work was limited to medical outpatient scheduling without taking into account academic institutional contexts or the real-time availability signaling.

Srinivas and Salah (2021) employed data analytics to forecast consultation durations and non-attendances in a cardiology clinic, attaining quantifiable enhancements in scheduling effectiveness by way of logistic regression and machine learning models. The study showed that predictive consultation time and no-show models of the clinic can contribute significantly to reduce the idle time and waiting time in the clinic. The research was limited to healthcare and not included the availability toggle concept as well as the dynamics of student engagement when applying to academic positions.

The authors of Noma et al. (2022) at the IEEE Nigeria International Conference on Disruptive Technologies, have been working on the design of an intelligent and secure hospital appointment scheduling system that fulfills the digitization requirements in Nigeria's health care sector. The study established that it is technically feasible to implement an intelligent scheduling system within the Nigeria infrastructure and connectivity environment. Although the work was relevant to the Nigerian institutional setting, it was specific to hospital settings and did not include a machine learning predictive model or academic scheduling features.

Idowu et al. (2018) designed a reliable online appointment scheduling tool for the outpatient services of the Nigerian Health Insurance Scheme (NHIS) in teaching hospitals, which proves the feasibility of web-based appointment scheduling system in the Nigerian technology and connectivity landscape. The study further established that the Nigerian users adapted very fast to the use of the internet based booking system when the interface was simplified to a great extent to make it easy to use. This work was totally based on hospital outpatient scheduling without any artificial intelligence or predictive analytics involved.

In IEEE Access, 2024, Valenzuela-Núñez et al. suggested a smart medical appointment scheduling framework that integrates optimization method, machine learning, and overbooking strategies, aiming at optimizing clinical resource utilization. The two strategies resulted in an 15 percent increase in the use of appointment slots with no increase in waiting time. The study was focused on provider-side scheduling resource optimization in healthcare, rather than adapting to university scheduling dynamics, and demonstrated the potential of using machine learning in scheduling logic.

Mahanthi and Jacob (2023) delivered a presentation at IEEE International Conference on Pervasive Computing and Social Networking on how the appointment booking process has evolved in traditional medicine, using advanced technology and user-centric design. The research also demonstrated that user-centric design can help boost adoption of the system for initial digital scheduling users. The study was limited to medical environments, and other factors such as queue management, priority handling, and AI predicting academic appointments were not covered.

Lester et al. (2022) created a web-based queue management system named CLIQUE which has real-time tracking of the queue as well as unit notifications which was implemented in the registrar's office of Angeles University Foundation in the Philippines. The system was developed using HTML, CSS, Bootstrap, ASP, JavaScript.Algorithms implemented in NET, C# and MySQL using a Model-View-Controller design. The study proved that a web-based service to manage queue at an administrative office of a university significantly increased the service flow and client satisfaction. It did not, however, include any artificial intelligence, nor did it include a component for predicting availability, nor was it available in the registrar's office beyond the general student/employee appointment management system.

Babayoff and Shehory (2022) published a study in the Journal of Medical Systems studying prediction models to enhance hospital outpatient clinics appointment scheduling via predicting the appointment length and the probability of the patient not showing up. Multiple machine learning models were used, such as Random Forest and Gradient Boosting, and ensemble models were generally more successful on the real appointment datasets than single models. The work validates the effectiveness of ensemble methods in scheduling prediction tasks directly, which is used for selecting the Random Forest Classifier in the GO-AMG system. The study was conducted in a clinical outpatient setting without adaptation to university settings.

In IEEE, Jayaprakash et al. (2020) proved that the student academic performance prediction from education data by using an improved Random Forest Classifier is a better solution. The study compared the results of the Random Forest algorithm with a number of different classifiers, such as Naïve Bayes, Decision Tree and SVM, using a large variety of student demographic and academic attributes. The Random Forest always performed well on all datasets tested. While the prediction target academic performance is different from the availability prediction task in GO-AMG, the study provides strong additional evidence that Random Forest is a reliable and accurate classifier for educational institution data, supporting its selection for the GO-AMG AI agent.

In the International Journal of Computer Science and Information Technologies, Mohammed (2023) examined the role of machine learning in healthcare and service processes for optimizing appointment scheduling systems, covering various methods and techniques. The review confirmed that machine learning models on scheduling data continually decrease waiting time, enhance slot utilization, and increase consumer satisfaction in comparison to rule-based scheduling methods. This review only included optimization applications for medical and commercial scheduling and overlook the appointment management in Universities and real-time availability signaling.

Salah and Srinivas (2022) published in Computers & Industrial Engineering, proposed a “predict-then-schedule” prescriptive analytics approach combining machine learning predictions with sequential scheduling optimization. The method firstly predicted appointment duration using a trained model and then optimized the appointment schedule for the next day based on the prediction value. This study was able to statistically significantly reduce patient waiting time without sacrificing clinic utilization. The work gives a conceptual basis for the GO-AMG method based on prediction to support appointment decisions, but it is in a different clinical context and the prediction is for an appointment length, rather than availability prediction, as in the work of this research**.**

In "Understanding and Enhancing Scheduling Systems using Computational Learning: A Theoretical Account of the convergence of Machine Learning Algorithms to Accurate Schedulers with enough training data" (2022) by Mehmood Ali Mohammed et al., the authors discuss the computational learning perspective of machine learning theory and its uses in understanding and improving scheduling systems. The analysis revealed that ensemble learning methods, like Random Forest, met the theoretical conditions for stable convergence on structured scheduling data. This theoretical basis reinforces the justification of employing Random Forest in the GO-AMG availability prediction task. The work was not related to the actual system and no new scheduling system was implemented.

Okoshi et al. (2020) conducted an empirical study in the real world with a large number of smartphones to examine the impact of push notification timing and content on user engagement and response. The study revealed a strong preference for notifications sent at a time when they were more likely to be of interest given the user's activity patterns, and inferred availability as this was more likely to get actioned compared to notifications delivered at a fixed or random time. This finding directly informs the design of the GO-AMG notification system: the principle of contextually-timed notification delivery found in this research that increases the chance of immediate student action is reflected in the availability alert the system provides when the subscribed staff member toggles on.

2.4 Summary of Literature Review

The table below presents a structured summary of the related works reviewed in this chapter.

S/N	AUTHOR(S)	CONTRIBUTION / WORK DONE	TECHNIQUE / METHODOLOGY	LIMITATIONS
1.	Qaffas, A. A. (2019)	Online appointment system for higher education with email reminders.	Web-based development with relational database.	No AI or predictive capability. Manual availability management only.
2.	Saraswathi et al. (2023)	Android-based Smart Appointment System for academic counselling.	Android mobile application with internet tools.	Android only. No web interface, no AI scheduling.
3.	Abu et al. (2022)	Web-based booking system for academic institutions with structured workflow.	Web-based application with mobile responsiveness.	No AI. Staff availability managed statically.
4.	Suresh et al. (2019)	BOOKAZOR web-based appointment booking for multiple service domains.	Ionic, Firebase, NodeJS for real-time data sync.	General commercial use. No appointment categorization or AI.
5.	Gharaibeh et al. (2019)	Hospital appointment scheduling with neural network for scheduling decisions.	Neural Network integrated into web-based application.	Healthcare only. Required substantial training data.

6.	Williams et al. (2018)	Impact of online appointment scheduling on quality of working life in university healthcare.	Mixed method using SPSS and NVivo.	Healthcare within university only. No academic appointments.
7.	Sharma et al. (2024)	ML model using 110,527 records to predict availability and no-show patterns.	Support Vector Machine (SVM) classification.	Medical only. Depended on large pre-existing datasets.
8.	Johnson et al. (2023)	AI-driven predictive scheduling outperforms manual approaches in service industries.	AI and machine learning for predictive scheduling.	Not education-specific. No toggle mechanism.
9.	Nallathamby et al. (2021)	Outpatient scheduling using Hadoop and Random Forest Regressor to predict doctor ratings.	Big Data with Random Forest Regressor.	Medical outpatient only. No academic context.
10.	Srinivas & Salah (2021)	Data analytics to predict consultation lengths and no-shows at a cardiology clinic.	Logistic regression and machine learning.	Healthcare only. No toggle concept.
11.	Noma et al. (2022)	Intelligent and secure hospital appointment scheduling for Nigeria. IEEE NIGERCON 2022.	Intelligent system design for Nigerian healthcare.	Hospital only. No ML model or academic features.
12.	Idowu et al. (2018)	Online appointment booking for NHIS outpatients in Nigerian teaching hospitals.	Web-based system for Nigerian healthcare.	Hospital outpatient only. No AI component.
13.	Valenzuela-Núñez et al. (2024)	Smart medical scheduling combining optimization, ML, and overbooking. IEEE Access.	Optimization and ML with overbooking strategies.	Healthcare provider-side only. No university dynamics.
14.	Mahanthi & Jacob (2023)	Transformation of medical appointment booking through technology and user-centric design. IEEE ICPCSN 2023.	Technology-driven user-centric design framework.	Medical only. No queue management or AI prediction for universities.
15.	Lester et al. (2022)	CLIQUE web-based queue management system with real-time tracking for university registrar.	HTML, CSS, Bootstrap, ASP.NET, C#, MySQL with MVC.	No AI component. Limited to registrar office, not general appointments.
16.	Babayoff & Shehory (2022)	Prediction models for hospital outpatient appointment length and no-show probability.	Random Forest and Gradient Boosting ensemble methods.	Clinical outpatient setting only. No university scheduling context.
17.	Jayaprakash et al. (2020)	Improved Random Forest Classifier for predicting student academic performance. IEEE.	Random Forest Classifier vs Naive Bayes, Decision Tree, SVM.	Academic performance prediction only, not availability scheduling.

18.	Mohammed (2023)	Review of machine learning in optimizing appointment scheduling systems. IJCSIT.	Survey and analysis of ML approaches in scheduling.	Review only. No university appointment system implemented.
19.	Salah & Srinivas (2022)	Predict-then-schedule prescriptive analytics for sequential clinical scheduling.	ML prediction combined with scheduling optimization.	Clinical context. Focused on appointment length, not availability toggling.
20.	Mehmood Ali et al. (2022)	Theoretical review of machine learning in scheduling from computational learning theory.	Computational learning theory and ensemble methods analysis.	Purely theoretical. No implemented scheduling system.
21.	Okoshi et al. (2020)	Real-world study on adaptive notification scheduling and user engagement on smartphones.	Large-scale field study with push notification analysis.	Focused on general smartphone notifications. No scheduling system implemented.

2.5 Research Gap

The twenty-one works reviewed as a group alongside show substantial strides in the field of appointment scheduling, both within healthcare and in the commercial sector. When analyzed more closely, however, there are four important omissions. First, there was no reviewed system specifically tailored to the appointment management of university students by category of academic studies, and the dynamics between departments. Second, none used real-time staff toggle activity data as main training signal in an academic context to predict availability. Third, no reviewed work featured a real-time, human availability toggle provided for staff to instantly make themselves known to all students. Fourth, no system, collectively, was found that predicted AI availability, supported drag-and-drop queue management, provided priority-based urgency handling and had a conversational AI assistant integrated into a single academic scheduling platform. Specifically, two Nigerian context works reviewed agreed that web-based scheduling is possible in the Nigerian context, but neither reaches to academic appointment management or machine learning prediction. The GO-AMG system bridges all four gaps in a seamless solution ready for mass production in the university context in Nigeria.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

This chapter gives the detailed analysis and design of GO-AMG system. It describes the approach taken, discusses the current manual appointment system and the proposed intelligent system and explains the functional and non-functional requirements that the system would need to meet. The chapter further presents UML models and the system design covering input and output design, database design, and system architecture.

3.1 System Analysis

3.1.1 Analysis Of The Existing System

The existing system is analyzed.Analysis of the existing system is done.

In Godfrey Okoye University, appointments with staff and students are mostly informal and manual. Students find staff by visiting, calling or sending messages via WhatsApp or by the help of secretaries. No record of requests, no queue management, no way to get staff availability information to students. Pupils often sit in front of the office doors waiting for hours without being told when they will be seen.

3.1.2 Limitations of the existing system

The current system has the following major drawbacks:

1. Students don't know when a staff member will be available, so they have to visit unnecessarily and wait for an extended period of time.

2. With no queuing, there is disorganisation and an inequitable distribution of the staff member's time.

3. Students take class time to physically search for and wait for staff, which impacts their learning.

4. Staff do not always have a clear structure of how to make themselves available and there are constant interruptions that occur without a plan.

5. Institution has no awareness of appointment trends, and no information on which to make decisions about staff access.

3.1.3 Analysis Of The Proposed System

The proposed GO-AMG system is based on the following two-agent model: a human agent (staff member), and an AI agent (embedded within the system). Staff make their availability statuses change in real time. Each student gets AI prediction of when he/she is likely to be available when inactive, and can schedule reminders; when active, students book immediately. Every toggle event in the database is used to train the AI agent. A priority-based queue assigns positions on booking, and is functional for the staff to drag/drop students depending on the urgency justification, and advances when a meeting ended button is clicked either by staff or students. A special notification is sent to the student at position 1 and a rating prompt is activated post meeting end.

3.2 System Requirements Specification (SRS)

3.2.1 Functional Requirements

Below are the functional requirements of the proposed system:

- The system shall provide students and staff with the ability to enroll and login using role-based access control to differentiate students, staff, and administrators.
- The staff member will have the ability to ‘toggle’ their availability in real time, and each toggle will be recorded in the toggle_history table to be used for training AI.
- Students may only make appointments with an active staff member and urgent or critical appointments must have a written justification of urgency.
- The system shall assign queue positions when student is booked, update student dashboard with real-time queue positions and enable staff to drag/drop appointments to reorder the queue.
- Both students and staff will have access to a meeting-ended button, which will be used to indicate to all that the appointment is completed, will reduce all the queue positions and will display a student rating prompt.
- The system shall notify when appropriate, including in the cancelled message the name of the person, and a special alert to the student at the 'one' queue position.
- Based on the toggle history data, the AI agent will be able to offer best time slot predictions and availability recommendations via a prediction modal and chatbox.
- Students will be able to leave anonymous star ratings and optional comments at the end of appointments (users of the analytics dashboard will see these ratings).

- The system shall keep record of the no show students and shall automatically deny booking privileges to those students who surpass the no show limit.
- The admin will be presented with a dashboard that provides system-wide statistics, peak times, insights provided by AI, no-show reports and recent student ratings.

3.2.2 Non-Functional Requirements

The proposed system has the following non-functional system requirements:

- The system shall be accessible from any current web browser on desktop and mobile devices, and shall not require software installation.
- Under normal network conditions the system shall respond to user interactions in no more than three seconds.
- All passwords will be stored in a hashing algorithm approved by the industry and all data transmission will be protected by HTTPS.
- The interface should be easy to use and accessible to all users, regardless of how technologically savvy they are.
- The system shall adhere to the principles relevant to data protection, ensuring personal data is kept secure, and not accessed by any other parties.
- The system shall deal with network failures gracefully, also preserving all data that has been loaded and stored on the client machine before the failure, so that the user is able to see the position of the queues and appointment information in the queues on the client computer until the network is restored.

3.3 System Design Methodology

Object Oriented Analysis and Design (OOAD) is used for this project. This approach is used to identify and define objects in the system, their attributes, methods, and relations, and to turn the requirements into an implementable blueprint. The choice of OOAD was made because the system encompasses several unique entities such as students, staff, appointments, notifications, AI agent, reminders, and toggle history, which align well with object-oriented classes. It is good for modularity, independent testing of each module and it is backed by Unified Modelling Language (UML) which offers universally understood notation to describe the structure and behaviours of systems.

3.3.1 Justification For Methodology

The modular design of OOAD is suitable for this project as it follows the modularity principle, which is also an object-oriented principle. All the modules can be developed, tested, and maintained separately, and the system is easily extensible to meet new requirements after deployment.

3.3.2 Method of Data Collection

The data for the design and modeling of the system was gathered by direct observation of the existing practice of appointment at Godfrey Okoye University and informal discussions with both students and staff regarding their experiences and expectations. The design decisions, throughout this chapter, were informed by the secondary data collected in the form of literature review in Chapter Two.

3.4 System Modeling using UML

UML diagrams are used as models for the proposed system capturing the structural and behavioral dimensions of the system.

3.4.1 Use Case Diagram

Use Case Diagram is a visual representation that illustrates who uses the system and what they can do in this system.

As illustrated in the GO-AMG Use Case Diagram in Figure 3.1, there are four actors both on the right and left-hand side of the system boundary. The Student actor and the Staff actor are on the left and the AI Agent actor and the Administrator actor are on the right side. The Student actor is linked to five use cases: Register/Login means the student registers or logs in into the system; View Real-Time Staff Availability means the student can see at a glance who are the available and unavailable staff members; Book Appointment means the student's primary action is to book an appointment with an available staff member; Track Queue Position means the student can see their position in the queue when they book; and Rate Completed Appointment means the student can give anonymous feedback after an appointment has been completed.

The Staff actor is linked to four use cases: Register/Login—this is the same actor use case as for the Student when they log on to the system as a staff member, and also needs to register/sign in; Toggle Availability—staff must click this to enable the queue to be advanced, and to allow the next student to log on as a staff member; Manage Appointment Queue—this takes in all the use cases associated with the queue, including confirming appointments, reading the urgency justifications of students who booked an appointment with urgent or critical priority, and using the drag and drop functionality to reorder students based on priority; Mark Meeting as Ended—this is a staff use case that occurs once a meeting has ended, meaning that the queue can be advanced and the next student notified.

On the right side, there is a use case for each autonomous part of the AI Agent: Predict Staff Availability, which relies on the trained Random Forest model to provide a probability prediction; Recommend Best Time Slots, which identifies and returns the top 5 most probable time slots for students to avail of; and Send Smart Reminders, which processes requests for reminders from students and sends them out at the specified time slots.

The three use cases linked to the Administrator actor are View Analytics Dashboard, giving the admin access to system-wide stats such as total appointments, peak times and no-show rates; Manage Users and Student Directory, a user management function which allows the admin to view all registered students and their unique registration numbers, and also to manage user accounts; and View No-Show Reports and Ratings, which allows the admin to view all no-show reports and ratings across the system, and to get accountability data about student behavior and feedback about staff interactions. Below is the use case diagram.

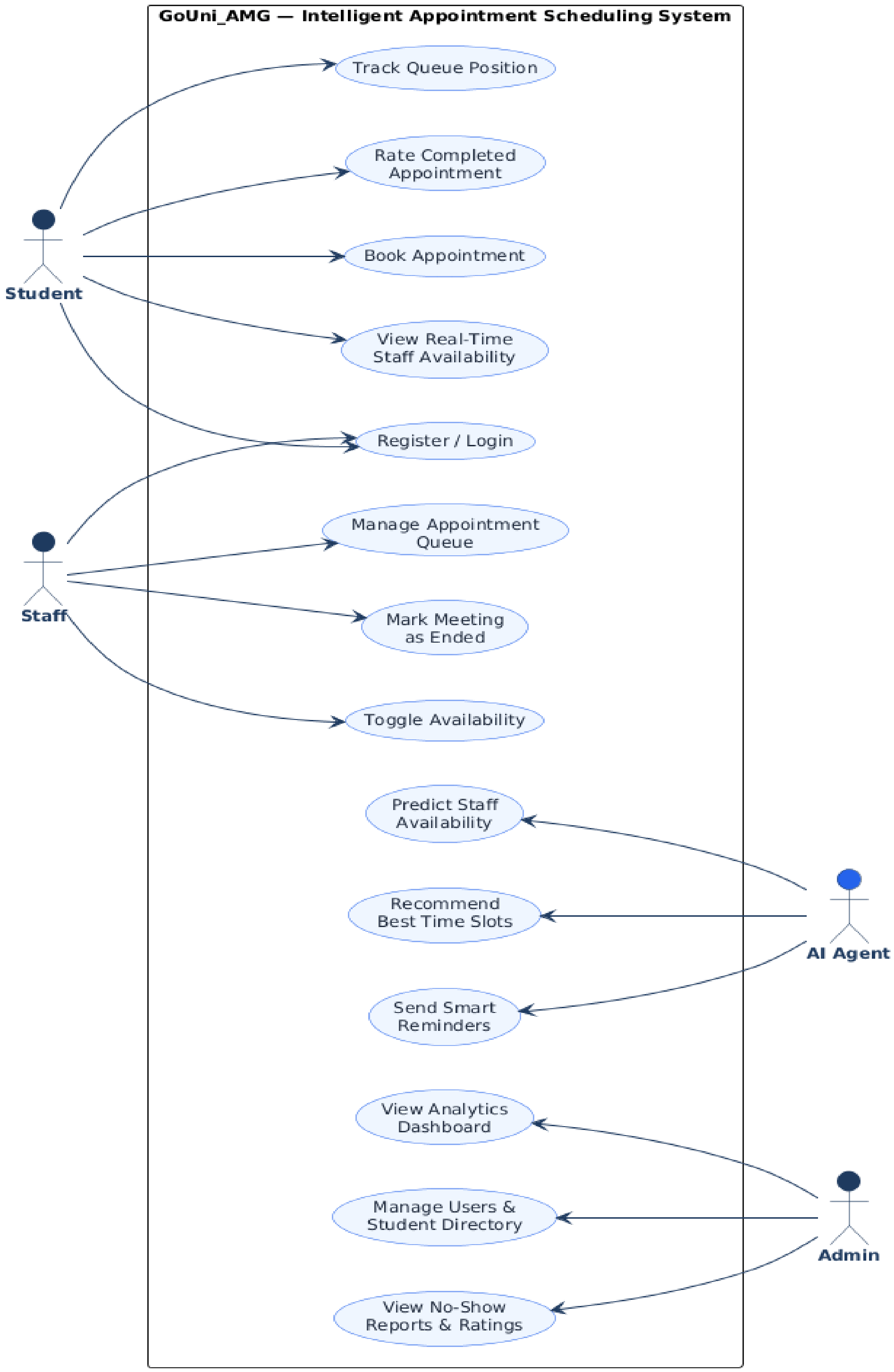


Figure 3.1: Use Case Diagram of the Proposed System

3.4.2 Activity Diagram

The Activity Diagram is a flowchart of the sequence of actions that occurs inside the system during a particular process. It employs action boxes, decision diamonds and arrows to indicate the flow of control between the steps and the points at which control flow branches out depending on a condition and points where control flow rejoins. An Activity Diagram is used to answer the question what, how and when.

Activity Diagram in Figure 3.2 illustrates the student's complete booking appointment process in GO-AMG system from when student logs in until the end of the appointment and feedback is taken.

The process starts at the filled black circle at the top, which is the starting point. The first action box indicates the student has logged in and has seen the staff list on their dashboard. The colored indicators on each staff card green for available, red for unavailable immediately communicate each staff member's status.

The first decision diamond will display the basic branch in the process: Is the staff toggle ACTIVE? If the answer is YES, the process flows down the left branch into booking pathway. The process branches to the right if the answer is NO, as it goes into the AI prediction pathway.

The student follows the YES path and chooses a category from the options (academic, administrative, financial or welfare), and a priority level as well. Then there is a second decision diamond to determine if the priority chosen is URGENT or CRITICAL. If so, the student will need to provide a required justification for urgency prior to allowing the student to advance. This text is attached to the appointment record and will be presented to the staff member on their queue management page which will give them the information they need to decide whether the appointment should be prioritized. It is skipped if the priority is NORMAL.

Next, the system allocates a position in line and notifies the student with a confirmation message. The staff review the appointment cards and read any urgency reasons on the cards, then reorder the cards in the queue, if they want to, by using the drag-and-drop interface provided. At the time and end of the meeting the staff member/user clicks the “meeting ended” button which will indicate that the meeting has been completed and reduce the number of remaining places in the queue by one and then notify the next student in line with their position/number one position. A final decision diamond gives the student the choice of adding a rating, so that if they do, their star rating and optional comment is recorded and will appear in an admin analytics dashboard, otherwise they will skip and the process will end.

On the NO path from the first decision diamond: the student looks at the AI prediction chips displaying the most likely times that that staff member is available. The student can communicate with the AI chatbox and ask questions about availability in natural language. A decision diamond then determines whether the student wishes to set a reminder: If they do, a reminder is created which contains the time at which the reminder was set; otherwise the process terminates for this visit but the student is free to return later when the staff member is available to book. The bottom filled circle with the ring is the termination point of the process. From each point in the diagram, a path eventually reaches this point, verifying that this point is the end of the process. The activity diagram of the proposed system is given below.

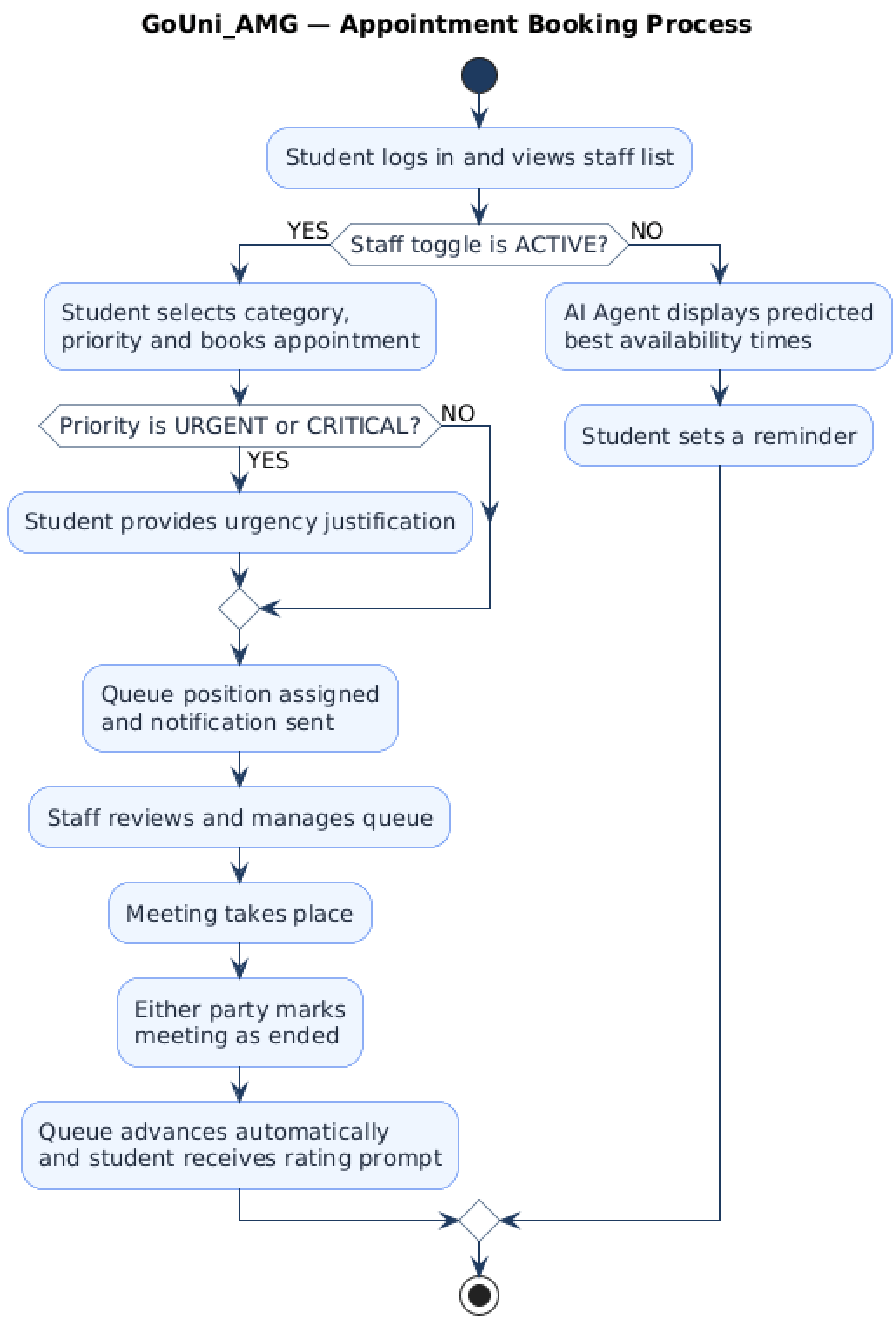


Figure 3.2: Activity Diagram for the Appointment Booking Process

3.4.3 Class Diagram

The Class Diagram is a structural diagram that presents all the major building blocks of the system (called classes), the relationships among them, the internal data in each (called attributes), and the capabilities of each (called methods). Use Case diagram illustrates what the system does, whereas Class diagram illustrates the internal organization of the system in order to do it. The boxes in the diagram are used to represent classes, with each class being divided into three parts: the top part contains the name of the class, the middle part contains the attributes of the class (the data that the class stores), and the bottom part contains the methods of the class (the actions that the class can perform).

As illustrated in Figure 3.3, the class diagram has nine classes which are designed to represent the full data and behaviour structure of the GO-AMG system.

The User class constitutes the basis of the identity structure in the system. Contains information about the basic characteristics of all users: id, name, email, password, role, and phone. There are different roles among students, staff, and administrators, with each characterized by a different role attribute. The three types of users share two methods: login() and register() which are provided by the User class. Since students and staff have this common identity but have some other characteristics as well, there are two inheritance relationships indicated in the diagram: one from Student to User and one from Staff to User. So, Student “is a” User and Staff “is a” User, meaning that they share all of User's attributes and methods but have their own as well.

The Student class inherits the attributes of the User class and includes three extra ones specific to students: reg_number (a unique university registration number which connects the profile created by the Student class to the record in the university's official student database), no_show_count (a counter of missed appointments), and is_banned (a boolean that is true if the no_show_count exceeds the threshold). There are four methods in the Student class: bookAppointment(), viewQueuePosition(), submitRating(), and setReminder().

The Staff class is an extension of the User class with 4 extra attributes: department, is_available (the toggleable status at the heart of the system), slot_duration (the settable length of the appointment) and no_show_count (the count of missed appointments). The Staff class has three methods, the most important being toggleAvailability() which handles queue advancement and alerts to meet end, reorderQueue(), which supports drag and drop queue management, and markMeetingEnded(), which initiates queue advancement and prompts for rating.

The Appointment class is the main connecting class in the system. Contains date_time, status (pending, confirmed, completed, cancelled), category, priority, queue_position and notes. The notes include optional text and mandatory urgency and/or mandatory justification when the priority is urgent or critical. There are two important relationships in the Appointment class, represented by solid lines with filled arrowheads – Student “1 to many” Appointment and Staff “1 to many” Appointment – one student can have many appointments and one staff member receives many appointments. This is a realistic scenario that occurs in the real world with one student and one staff member per appointment.

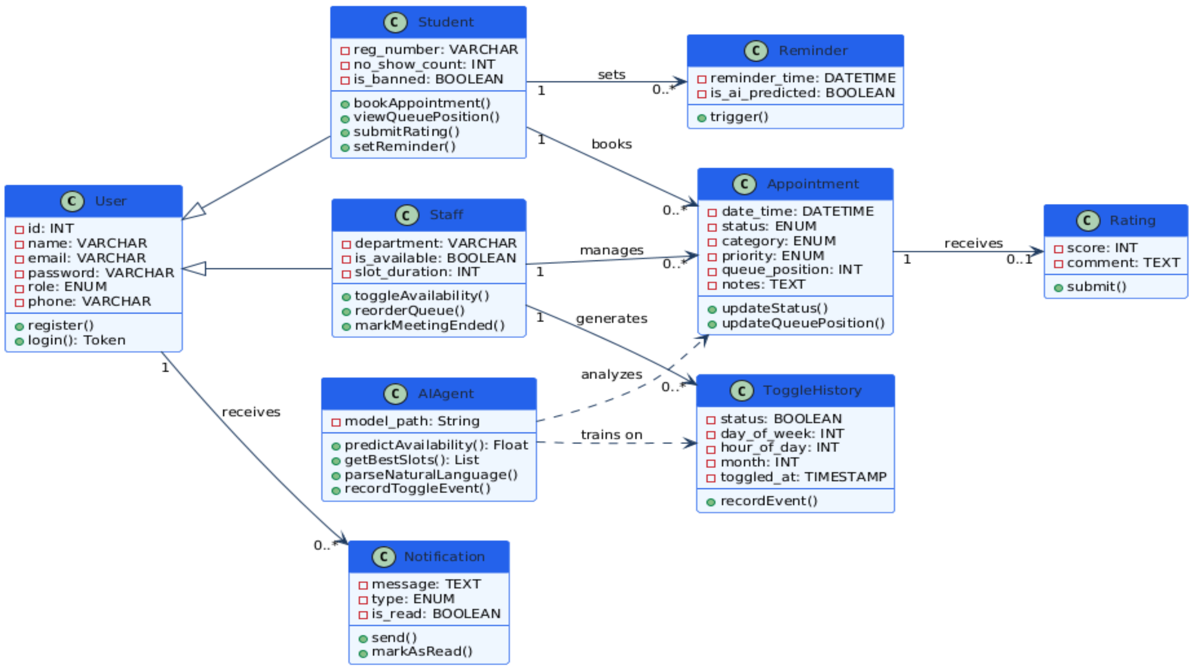
AIAgent is the autonomous AI part of the system. The attributes include model_path (path to the serialized model file of the Random Forest model on disk). It has four methods that reflect its full responsibilities: predictAvailability() uses the trained model to return a probability score; getBestSlots() returns and ranks the top availability windows predicted; parseNaturalLanguage() parses the students' query from the chatbox and calls the trained model; and recordToggleEvent() is called whenever a staff member toggles, adding a new row to the ToggleHistory table. The dotted line from AIAgent to ToggleHistory indicates a dependency, since the AI agent relies on ToggleHistory for training model. Analysis: Dotted line from AIAgent to Appointment is reading the appointment data and finding patterns.

Each availability toggle event is recorded by the ToggleHistory class and has the following 5 temporal attributes: status (the new availability state), day_of_week, hour_of_day, month, and toggled_at (the exact timestamp). The relationship line from staff to ToggleHistory indicates that over time each staff will create many toggle history records, and there is a "1 to many" relationship between staff and toggle history.

The Notification class contains content of the notification, notification type (availability, confirmed, cancelled, reminder), and the is_read field. User to Notification indicates that a user can have multiple notifications, including for all three roles: student – booking confirmation, booking queue notification; staff – booking notification when a student books with them; and both user and staff – cancellation notification of the cancelling party's name.

The Rating class is used to store the post appointment feedback (an integer 1-5) and optionally a comment. One appointment may be given at most one rating in the relationship from Appointment to Rating.

The Reminder class holds a reminder set by the student, reminder_time attribute, and a Boolean is_ai_predicted attribute that indicates whether the reminder was set based on an AI prediction suggestion or manually set by the student. The relationship from Student to Reminder indicates that many reminders can be set by one Student, to various staff. The class diagram of the proposed system is given below.



<u>**Figure 3.3: Class Diagram of the Proposed System**</u>

3.5 System Design

3.5.1 Input Design

Key input interfaces are the registration form (which gathers a name, email, password, role and, for students, the required registration number), the availability toggle on the staff dashboard, the appointment booking form (which obtains category, priority, and for urgent and critical, a required urgency justification), the reminder setup form (which gathers a timer, AI-suggested, custom time), and the rating form (which provides a star rating of between one and five, and an optional comment). There are validation rules for all the fields that will not allow for a blank or poorly formatted entry.

3.5.2 Output Design

Key system outputs include the student dashboard with colour coded indicators of staff availability; AI prediction chips for when staff are unavailable; real time queue position panel with urgency coded position indicators; staff appointments page with priority colour coded cards and urgency reason boxes; the notification panel which provides timely alerts with the name of the staff member in cancellation messages; and the admin analytics dashboard which provides system wide statistics, AI insights, peak time charts and recent ratings.3.5.3 Database Design

The database is implemented using MySQL and organized into eight tables as described below.

Table 3.1: User Table

S/N	Attribute	Data Type	Constraint	Description
1	userId	INT	PK, AUTO_INCREMENT	Unique identifier for each user
2	name	VARCHAR(100)	NOT NULL	Full name of the user
3	email	VARCHAR(100)	NOT NULL, UNIQUE	Email address for login and notifications
4	password	VARCHAR(255)	NOT NULL	Hashed password for authentication
5	role	ENUM	NOT NULL	Role: student, staff, or admin
6	phone	VARCHAR(20)	NULLABLE	Phone number for contact purposes
7	created_at	TIMESTAMP	DEFAULT NOW()	Date and time the account was created

Table 3.2: Staff Table

S/N	Attribute	Data Type	Constraint	Description
1	staffId	INT	PK, FK → User	Links staff record to the user table
2	department	VARCHAR(100)	NOT NULL	Department the staff member belongs to

3	title	VARCHAR(100)	NULLABLE	Academic or administrative title
4	is_available	BOOLEAN	DEFAULT FALSE	Real-time availability toggle status
5	slot_duration	INT	DEFAULT 15	Duration of each appointment slot in minutes
6	no_show_count	INT	DEFAULT 0	Number of times the staff was a no-show

Table 3.3: Student Table

S/N	Attribute	Data Type	Constraint	Description
1	studentId	INT	PK, FK → User	Links student record to the user table
2	reg_number	VARCHAR(50)	NOT NULL, UNIQUE	University registration number — unique institutional identifier
3	no_show_count	INT	DEFAULT 0	Number of no-show instances
4	is_banned	BOOLEAN	DEFAULT FALSE	Flags students with repeated no-shows

Table 3.4: Appointment Table

S/N	Attribute	Data Type	Constraint	Description
1	appointmentId	INT	PK, AUTO_INCREMENT	Unique identifier for each appointment
2	student_id	INT	FK → Student	References the booking student
3	staff_id	INT	FK → Staff	References the staff member being visited
4	date_time	DATETIME	NOT NULL	Date and time of the appointment
5	status	ENUM	NOT NULL	Status: pending, confirmed, completed, cancelled
6	category	ENUM	NOT NULL	Category: academic, admin, financial, welfare
7	priority	ENUM	DEFAULT normal	Priority: normal, urgent, critical
8	queue_position	INT	NOT NULL	Position in the staff member's queue
9	notes	TEXT	NULLABLE	Notes including urgency justification

Table 3.5: Notification Table

S/N	Attribute	Data Type	Constraint	Description
1	notifId	INT	PK, AUTO_INCREMENT	Unique identifier for each notification
2	user_id	INT	FK → User	References the recipient
3	message	TEXT	NOT NULL	Content of the notification message

4	type	ENUM	NOT NULL	Type: availability, confirmed, cancelled, reminder
5	is_read	BOOLEAN	DEFAULT FALSE	Whether the notification has been read
6	sent_at	TIMESTAMP	DEFAULT NOW()	Date and time the notification was sent

Table 3.6: Rating Table

S/N	Attribute	Data Type	Constraint	Description
1	ratingId	INT	PK, AUTO_INCREMENT	Unique identifier for each rating
2	appointment_id	INT	FK → Appointment	References the appointment being rated
3	score	INT	NOT NULL	Rating score between 1 and 5
4	comment	TEXT	NULLABLE	Optional anonymous feedback from the student
5	created_at	TIMESTAMP	DEFAULT NOW()	Date and time the rating was submitted

Table 3.7: Reminder Table

S/N	Attribute	Data Type	Constraint	Description
1	reminderId	INT	PK, AUTO_INCREMENT	Unique identifier for each reminder
2	student_id	INT	FK → Student	References the student who set the reminder
3	staff_id	INT	FK → Staff	References the staff member the reminder concerns
4	reminder_time	DATETIME	NOT NULL	Date and time to trigger the reminder
5	is_ai_predicted	BOOLEAN	DEFAULT FALSE	Whether the reminder was AI-suggested
6	is_sent	BOOLEAN	DEFAULT FALSE	Whether the reminder has been sent
7	is_cancelled	BOOLEAN	DEFAULT FALSE	Whether the student cancelled the reminder

Table 3.8: Toggle History Table

S/N	Attribute	Data Type	Constraint	Description
1	id	INT	PK, AUTO_INCREMENT	Unique identifier for each toggle event
2	staff_id	INT	FK → Staff	References the staff member who toggled
3	status	BOOLEAN	NOT NULL	1 = available (ON), 0 = unavailable (OFF)
4	day_of_week	INT	NOT NULL	Day of week (0=Sunday to 6=Saturday)
5	hour_of_day	INT	NOT NULL	Hour of day the toggle occurred (0 to 23)

6	week_of_year	INT	NOT NULL	Week of the year (1 to 52)
7	month	INT	NOT NULL	Month of the year (1 to 12)
8	toggled_at	TIMESTAMP	DEFAULT NOW()	Exact timestamp of the toggle event

The relationships between these tables are defined through primary and foreign key constraints. The Staff and Student tables each reference the User table. The Appointment table references both Student and Staff. The Notification table references User. The Rating table references Appointment. The Reminder and ToggleHistory tables both reference Staff. The ToggleHistory table is particularly significant as it serves as the sole training dataset for the Random Forest Classifier.

3.5.4 System Architecture Design

The proposed system is built following a web-based three-tiered client-server design that consists of a presentation layer, an application layer, and a data layer, along with a dedicated AI service layer. The user interfaces for students, staff, and administrators are located in the presentation tier. The application tier has JavaScript and its own Express server to do all the business logic. The AI service layer is a standalone microservice written in Python Flask that runs independently and is used to serve availability predictions on demand to the back-end layer. The data tier is a relational database called MySQL. Communication between the front end and the back end is all via HTTPS and the back end communicates with the AI service via HTTP, in a REST API style. The system architecture diagram of the proposed system is given below.

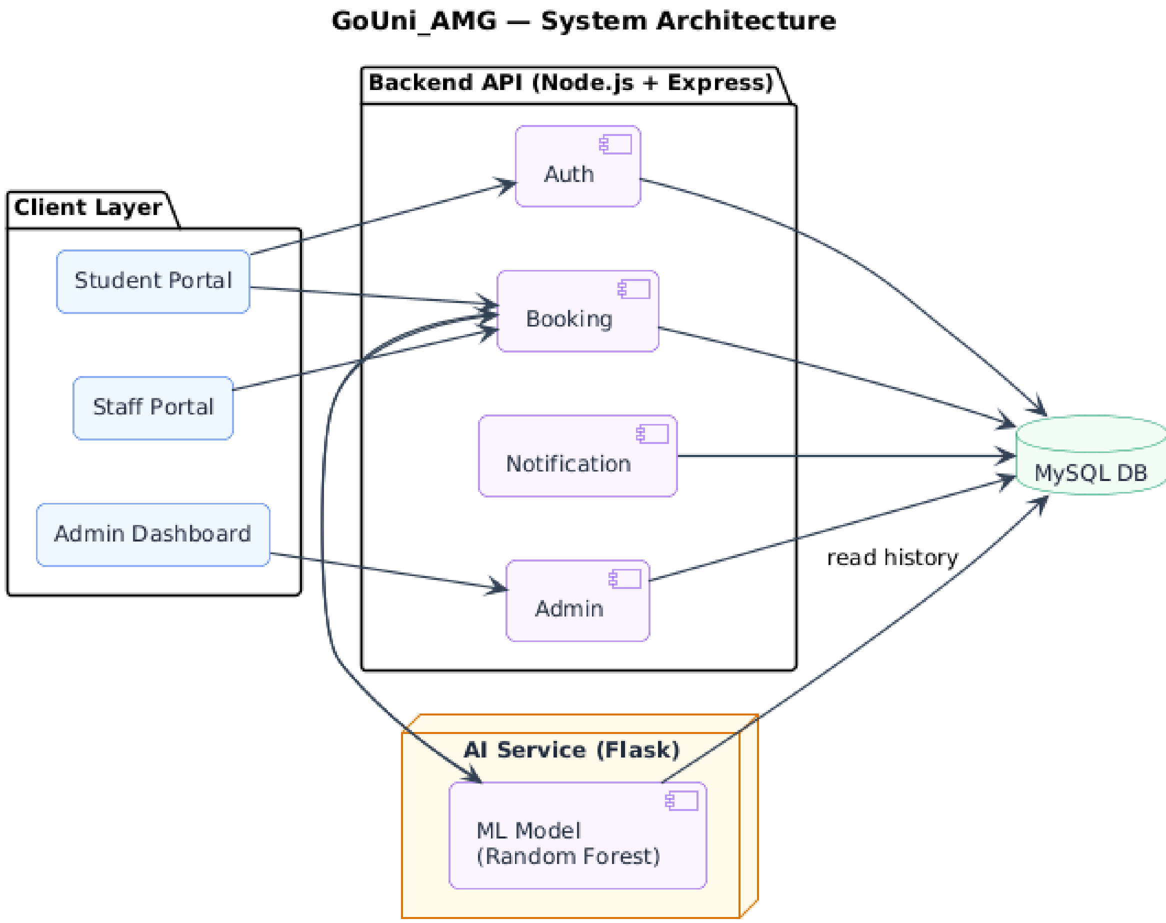


Figure 3.4: System Architecture Diagram of the Proposed System

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.0 Introduction

This chapter explains the implementation of GO-AMG system, which includes development environment and tools, AI dataset and model training, system testing and evaluation, presentation and analysis of the results gained.

4.1 Development Environment and Tools

4.1.1 Programming Languages and Libraries

The GO-AMG system was developed using JavaScript for both the frontend and backend, and Python for the AI service. It was decided to use JavaScript to support a uniform Web stack. We selected Python as the AI service language because it supports many machine learning libraries. Designed with React, Tailwind CSS for the frontend, Node.js + Express.js for the backend, Python + Flask + scikit-learn for the AI service, and MySQL for the relational database management system.

4.1.2 Development Platform and Hardware Requirements

This system is developed in Visual Studio Code, the main Integrated Development Environment, and Windows 11 computer. The following is the minimum requirement for the development environment: Node.js 18 or greater, Python 3.11 or greater, MySQL 8.0 or greater. During the development phase, at least 4 gigabytes of RAM capacity was needed to run three services at the same time.

4.2 Dataset Description and Preprocessing

The AI model is trained with the data in the toggle_history table, which contains all availability toggle events as day, hour, week, and month. The table was then populated with 493 seed records to represent different availability patterns for eight different staff members for the development and testing. Before the training phase, 15% of the status labels were randomly changed to introduce a natural human behavioural variation and to avoid unrealistically high accuracy on the structured labelled data.

4.3 Model Architecture And Training

The AI model is based on the Random Forest Classifier, which was implemented using the scikit-learn library. One hundred decision trees were used, with a max depth of ten and a balanced class weight so as to not be biased in favor of the more common class. The training features consisted of: staff_id, day_of_week, hour_of_day, and month; the target label was the status column. The data set was divided into 80% training and 20% testing data. After training on the noise injected dataset, the model was tested on a held-out set of data and obtained a classification rate of 73.74 percent, which is reasonable and adequate considering the intentional noise added. The model trained was serialized and stored on disk for deployment without re-training.

4.4 System Implementation and Modules

The GO-AMG system is comprised of five key modules. The authentication module is responsible for all three roles: registration, log-in, and password encryption, with bcryptjs, while managing the sessions in a stateless fashion with the use of JSON Web Tokens. The staff availability and queue management module deals with real-time toggling, automatically adding a record to the toggle_history table on each event to continuously feed the AI training set, and queues drag-and-drop reorder user interface. The appointment booking and lifecycle module keeps track of the entire appointment lifecycle including urgency justification for urgent and critical appointments, queue positions, and special notifications to students at position one, as well as decrements queue positions when meetings end. The predict module and the chatbox module inside the AI module work with the Flask AI microservice to serve AI predictions, with AI predictions presented in a modal and a conversational chatbox that interprets natural language queries. The notification and rating module provides in-app and email notifications in cancelations, and displays a star rating modal after a meeting-ended event.

4.5 Testing and Evaluation

4.5.1 Evaluation Metrics

The AI model was tested on the 20 percent test set, which is held out, with standard classification metrics. The model performance was 73.74 per cent, with a similar performance seen across both the available and unavailable classes, thus indicating that the configuration of balanced class weight was effective in preventing majority class bias. This is a realistic and acceptable level of accuracy for the noise injected during pre-processing (fifteen per cent).

4.5.2 System Testing

All key backend API endpoints were tested in the Thunder Client extension of Visual Studio Code with valid, invalid and expired or missing tokens. The following table outlines some of the key test cases carried out.**Table 4.1: System Test Cases**

S/N	Test Case	Expected Result	Actual Result	Status

1	Register new student with valid credentials	Account created, token returned, student record inserted	Account created, token returned, student record inserted	Pass
2	Login with correct credentials	Login successful, JWT token returned with role details	Login successful, token with student_id and reg_number returned	Pass
3	Login with incorrect password	401 Unauthorized, error message returned	401 Unauthorized, "Invalid email or password" returned	Pass
4	Toggle staff availability	Status flipped, toggle_history record inserted, students notified	Status flipped, record inserted, students notified with staff name	Pass
5	Book appointment with active toggle	Appointment created, queue position assigned, notification sent	Appointment created, queue position assigned, notification sent	Pass
6	Book appointment with inactive toggle	400 Bad Request, unavailability message returned	400 Bad Request, "Staff member currently unavailable" returned	Pass
7	Book urgent appointment without justification	Validation error, booking blocked	Form validation prevented submission, required field error shown	Pass
8	Drag and drop appointment to new position	All queue positions updated in database	Queue positions updated correctly for all appointments	Pass
9	Mark meeting as ended from staff side	Appointment completed, queue decremented, student notified	Appointment completed, queue updated, rating prompt sent	Pass
10	Submit star rating for completed appointment	Rating saved, visible on admin dashboard	Rating saved, score and comment visible in admin ratings section	Pass
11	Access student route with admin token	403 Forbidden, permission denied returned	403 Forbidden, permission denied returned	Pass
12	GET AI prediction for a staff member	Prediction summary with best slots returned	Prediction returned with percentages between 65% and 92%	Pass

4.5.3 User Interface Testing and Walkthrough

Three user roles were used: each with their own browser environment and the complete system was navigated manually. The login page redirected users to the appropriate dashboards for their role. The student dashboard featured availability colours, a real-time queue panel and AI prediction chips on unavailable staff cards. The booking form dynamically rendered the field for justification of urgency for urges and critical priorities. The staff appointments page displayed draggable cards and urgency reason boxes, both colored as appropriate to their priority. The admin dashboard provided correct database statistics, AI insights, and latest student ratings from the real-time database.

4.6 Results Presentation and Analysis

4.6.1 Output Samples and Interface Screens

The key interfaces of the GO-AMG system are illustrated in the following figures, showing the actual output and functionality for each of the three user roles.

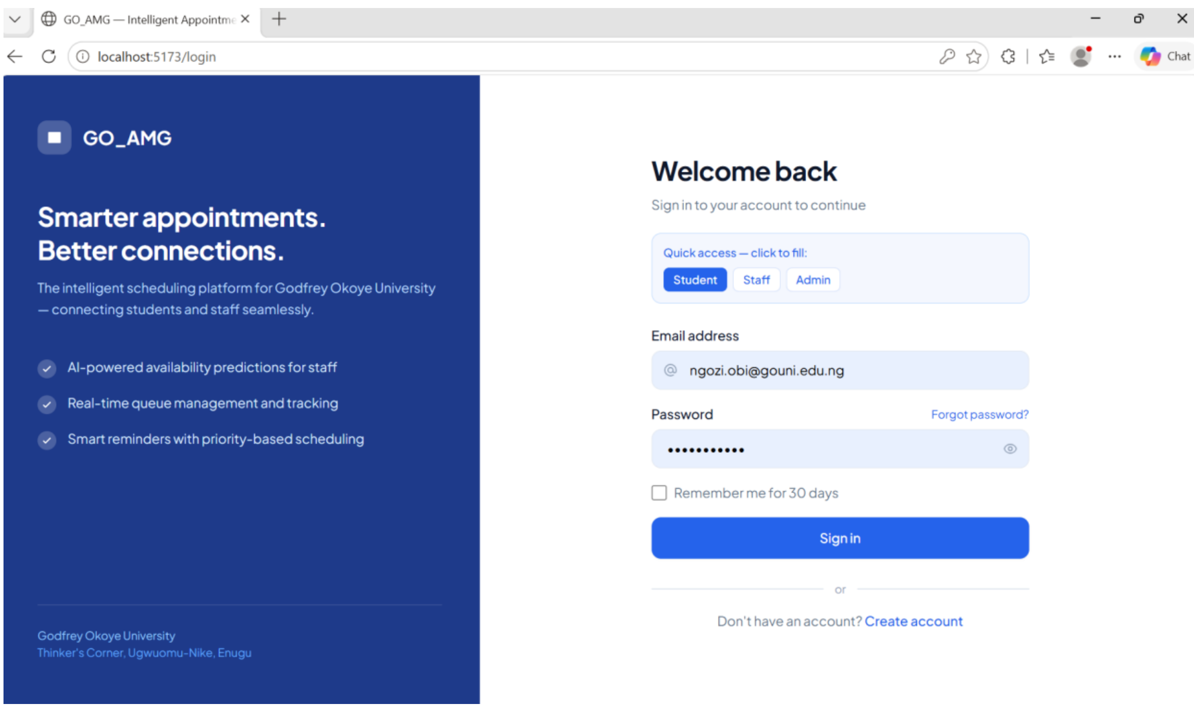


Figure 4.1: Welcome Page of the System

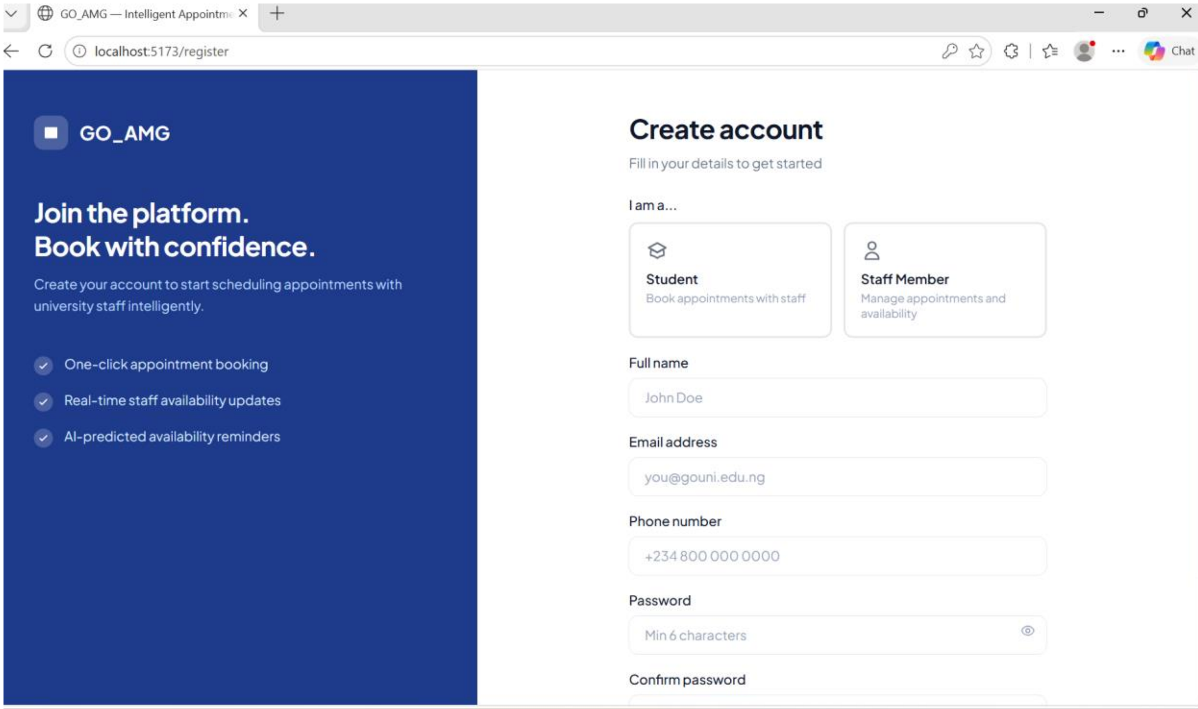


Figure 4.2: Create Account Page

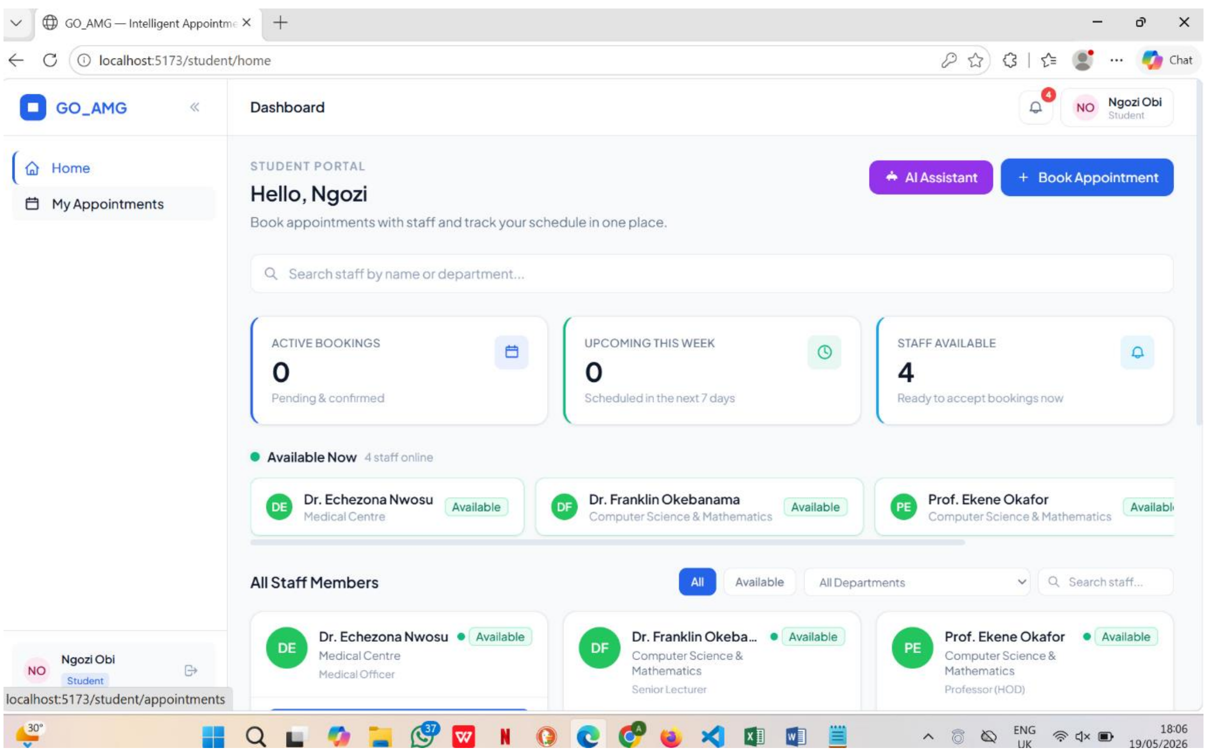


Figure 4.3: Student Dashboard 1

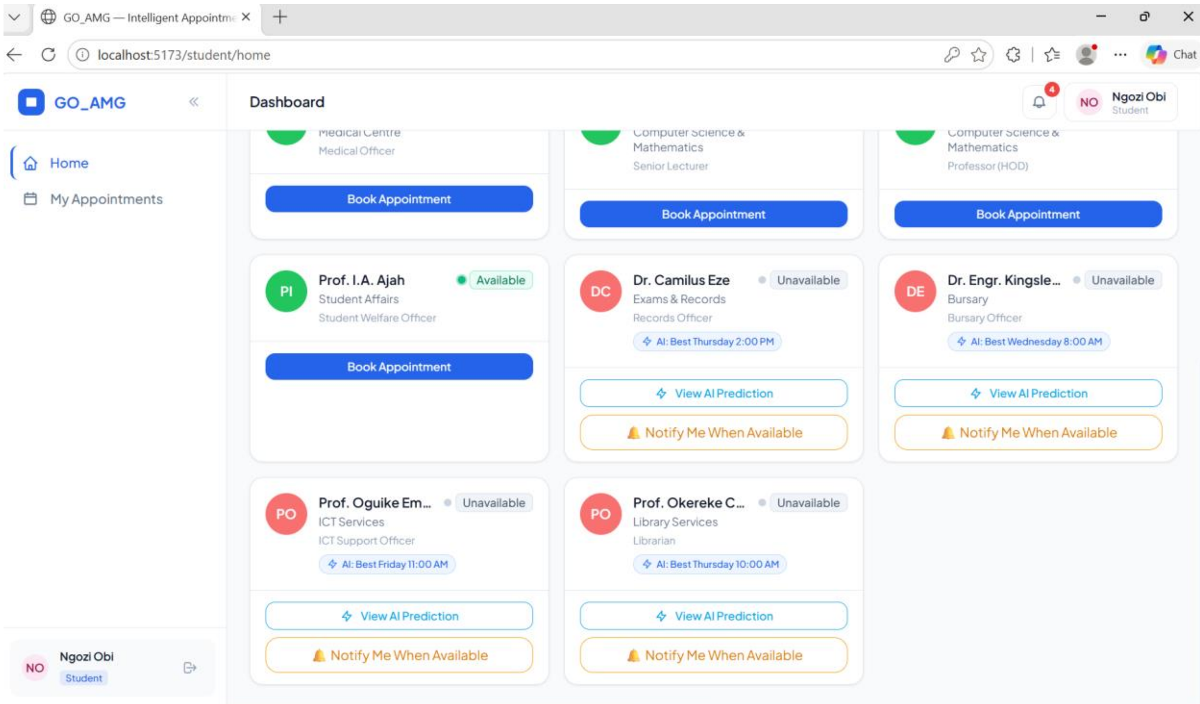


Figure 4.4: Student Dashboard 2

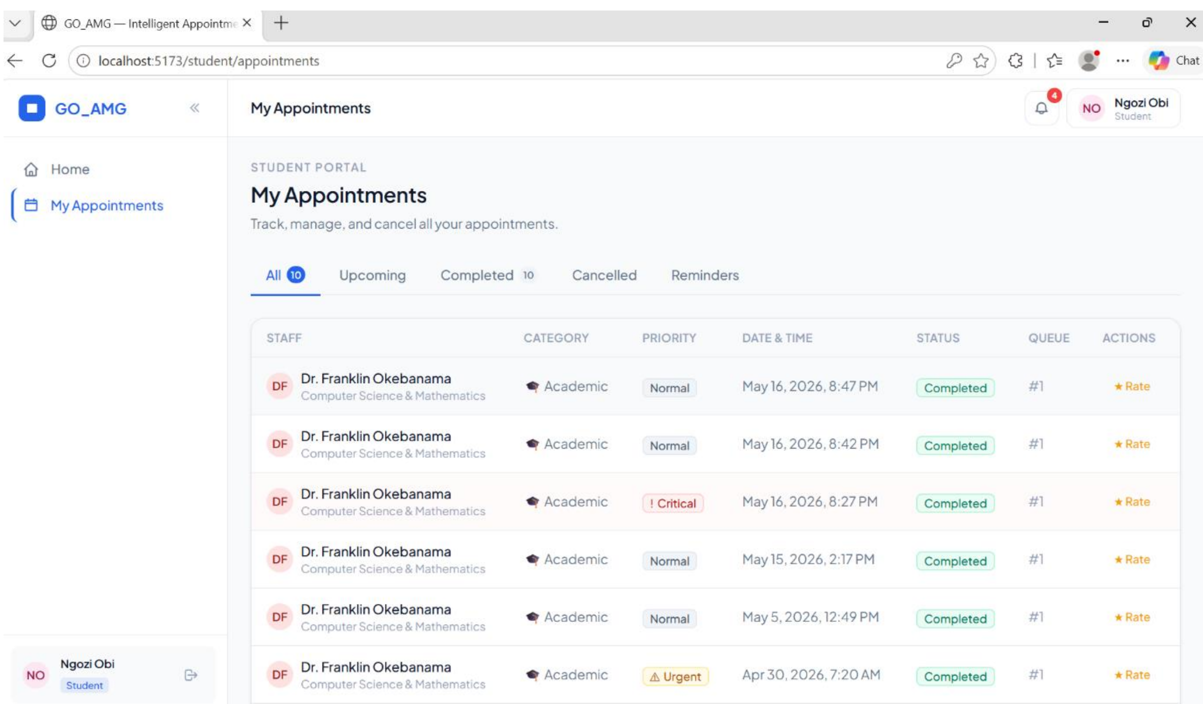


Figure 4.5: Student Appointment Page

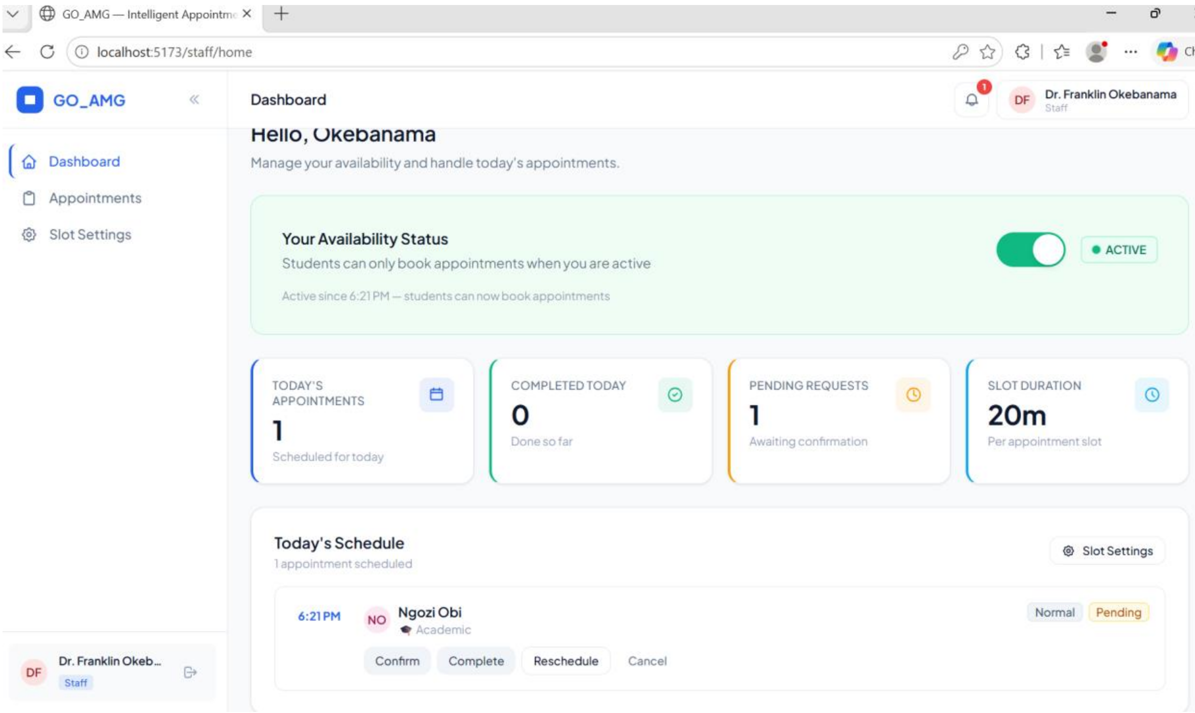


Figure 4.6: Staff Dashboard

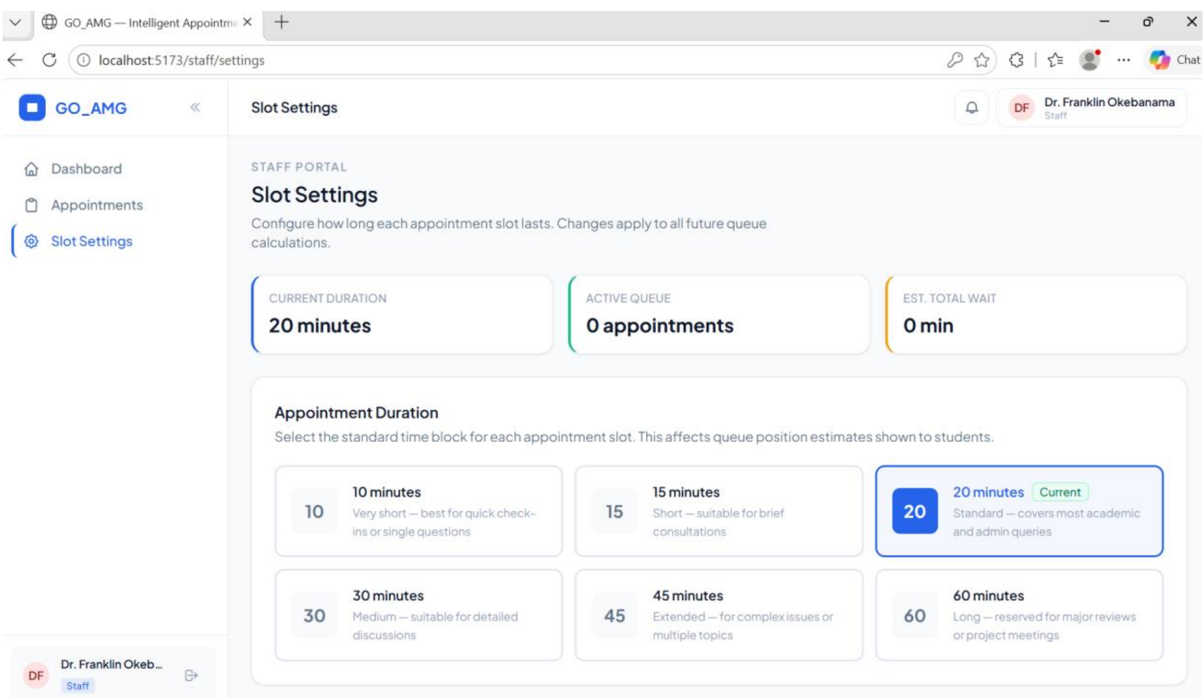


Figure 4.7: Staff Slot Settings

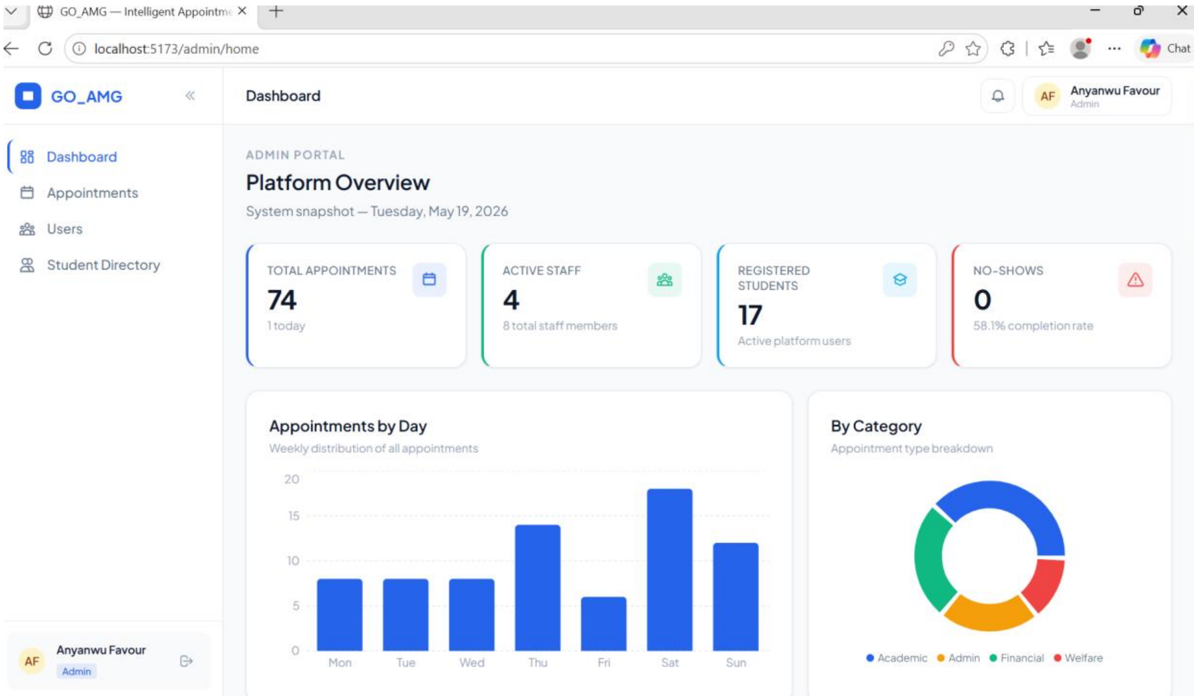


Figure 4.8: Admin Dashboard 1

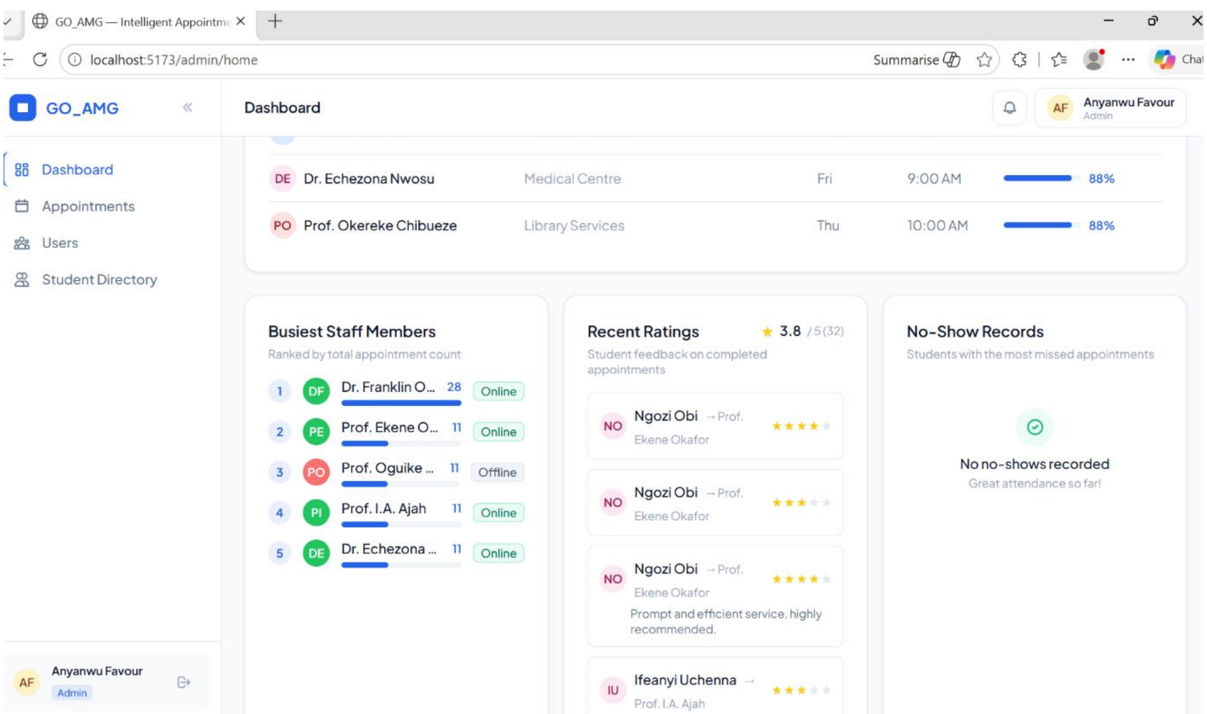


Figure 4.9: Admin Dashboard 2

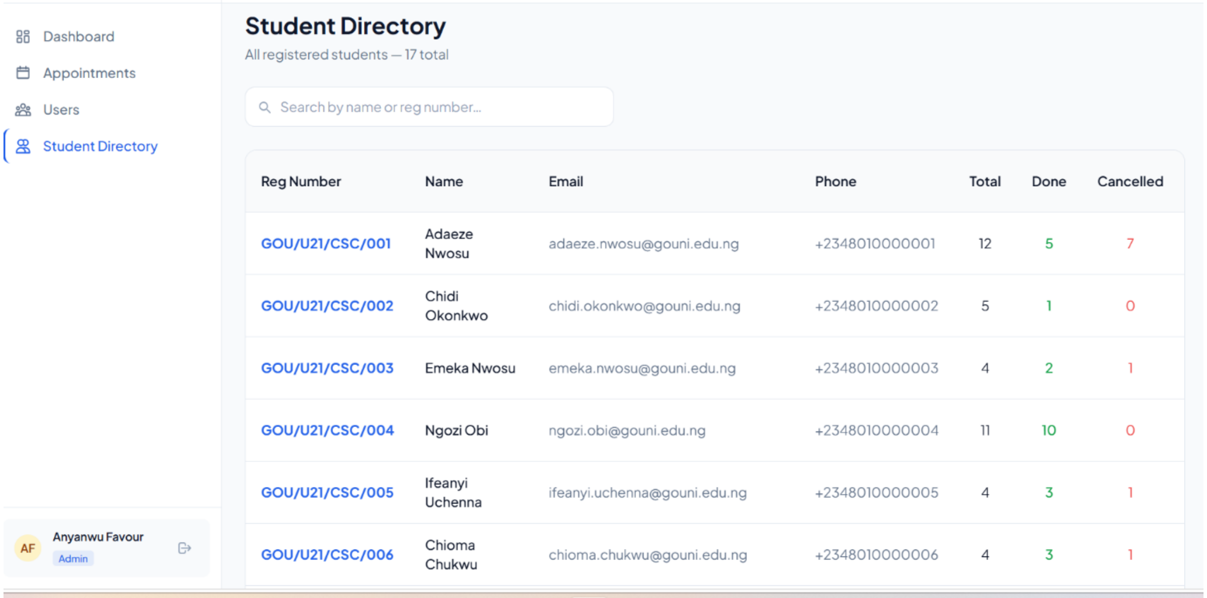


Figure 4.10: Admin Student Directory

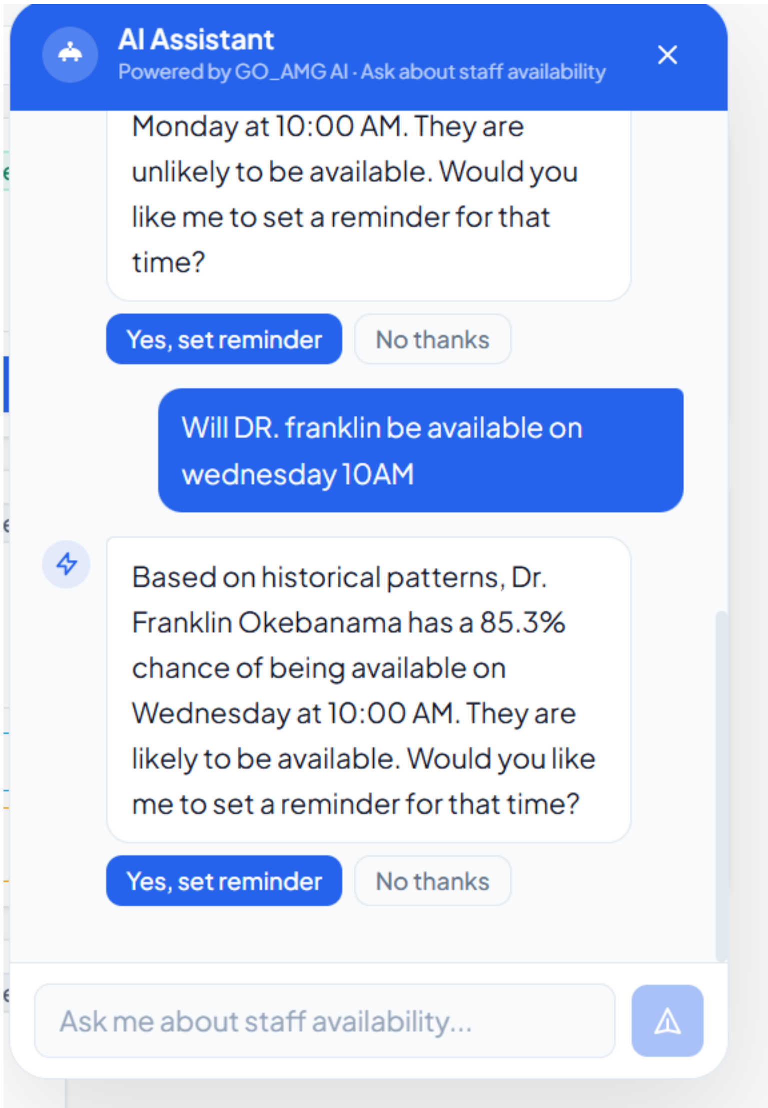


Figure 4.11: AI Assistant Page

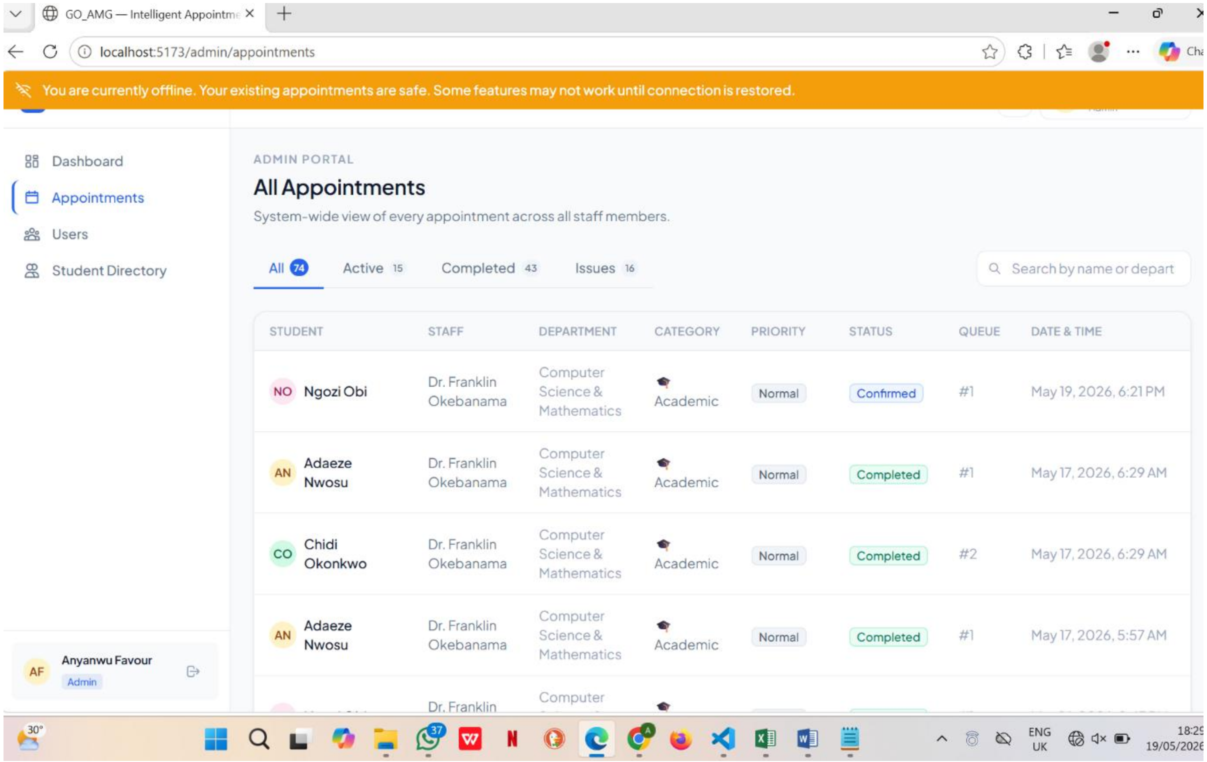


Figure 4.12: System Managing Poor Network Connection

4.6.2 Discussion of Results and System Performance

The results demonstrate that the GO-AMG system is effective in meeting all four goals. A working web-based scheduling system, including real-time availability toggle was deployed and tested. The Random Forest Classifier was accurate at 73.74 percent when tested on noisy data, and produced realistic probability estimates displayed on a visual interface and a conversational AI channel. The priority-based queue management system maintains correct urgency justification and drag-and-drop reordering, ensuring accuracy of the database records. All 12 system test cases passed and the system interface was shown to work in all three roles. The findings answer the questions raised in Chapter One and support the notion that the GO-AMG system is a comprehensive and viable solution to appointment scheduling problems.

CHAPTER FIVE

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.1 Introduction

This chapter summarizes the entire research process, presents conclusions from the results derived from the research work, and recommends ways to improve and expand the GO-AMG system in the future. It provides a commentary on the fulfilment of the goals set out in Chapter One and the value of this contribution to the field of higher education intelligent administrative systems.

5.2 Summary

The motivation for this research is the unemployment of student-staff appointment in Nigerian universities that have not been solved with the help of the manual system.The motivation for this research was the prevalent inefficiencies in manual student-staff appointment in Nigerian universities that has yet to be resolved using the manual system. Chapter One stated eight problems and four objectives for the GO-AMG system. Chapter Two analysed twenty-one related works from IEEE, Springer, Research Gate and other well-known sources, identifying the following gaps: (1) no system included specialized systems for academic student-staff appointment management, (2) none discussed the availability toggle mechanisms to be available in real time, (3) no system described AI predictions based on the toggle activities, and (4) no system from the reviewed works could be found that combined all of the above. The full system analysis and design is provided in Chapter Three where you will find details about the system needs, OOAD methodology, UML models and also a design of the database across eight interconnected tables, including the toggle_history table that is the source of training data for the AI model. Chapter Four showed the implementation in three parallel coordinated services, with 73.74 percent accuracy in the Random Forest model with the introduction of noise in data and passing all twelve system test cases.

5.3 Conclusion

The inefficiency of the manual appointment scheduling process was a problem at Godfrey Okoye University that the GO-AMG system was created to solve. The four specific objectives have all been met. A real-time toggling capability for availability was added, and it was also used for gathering AI data. A Random Forest Classifier was deployed using toggle history data with realistic availability predictions provided via a visual interface and a conversational chatbox. A priority based queue management system with urgency justification enforcement and reordering by drag-and-drop was implemented and verified. A notification system was embedded in cancelation messages in order to alert staff, no-show tracking was added, an anonymous rating system was set up, and an administrative analytics dashboard was incorporated and verified working. No system test cases failed.

The GO-AMG system successfully shows that real-time web technology and machine learning can result in a transformative solution for both students and staff of a university in Nigeria. The system is ready to use and has been an important contribution to the area of intelligent administrative systems in higher education.

5.4 Recommendations

The following recommendations are made for further development on the basis of the results of this research.

First, it is suggested that the system be installed to a cloud hosting platform, using a suitable domain name and SSL certificate, so that the toggle_history table can collect real world data that is going to progressively improve the accuracy of the AI model predictions beyond what was achieved with the seeded data (which was 73.74 percent).

Second, it is recommended to use an automated retraining pipeline to ensure continuous improvement of the AI model without manual effort and retraining the model at regular intervals, e.g., once every academic semester.

Third, it is recommended that it be extended to SMS delivery to make sure that students with limited Internet access can still be alerted to changes of availability and to queue position changes.

Fourthly, it is suggested that a native mobile application be created for both Android and iOS platforms. The current REST API backend can be used without any changes for mobile client development.

Fifthly, the recommendation to introduce the GO-AMG system into other Nigerian Universities facing similar challenges to the rest of the nation. With a modular design and a full source code that can be found at https://github.com/Chidals001/go-amg, the complete source code of the project offers an ideal building block for institutionalisation.

REFERENCES

Abu, S., Aziz, N., & Rahman, R. A. (2022). A review and design of web-based student-lecturer appointment booking system. International Journal of Advanced Computer Science and Applications, 13(4), 112–119.

Ahmadi-Javid, A., Jalali, Z., & Klassen, K. J. (2017). Outpatient appointment systems in healthcare: A review of optimization studies. European Journal of Operational Research, 258(1), 3–26.

Babayoff, O., & Shehory, O. (2022). Improving hospital outpatient clinics appointment schedules by prediction models. Journal of Medical Systems, 46(12), 1–12.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

Cayirli, T., & Veral, E. (2003). Outpatient scheduling in health care: A review of literature. Production and Operations Management, 12(4), 519–549.

Gharaibeh, N., Al-Smadi, M., & Al-Omari, A. (2019). A web-based application for innovative hospital appointment scheduling using neural network. Journal of Medical Systems, 43(7), 1–9.

Idowu, A. P., Adeosun, O. O., & Williams, K. O. (2018). Dependable online appointment booking system for NHIS outpatient in Nigerian teaching hospitals. International Journal of Computer Applications, 180(28), 1–9.

Jayaprakash, S., Geetha, T., & Ramkumar, M. (2020). Predicting students academic performance using an improved Random Forest Classifier. 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE. https://doi.org/10.1109/ICESC48915.2020.9167547

Johnson, R., Mitchell, K., & Stevenson, L. (2023). AI-driven predictive scheduling: Transforming workforce management through machine learning. Journal of Business and Technology, 18(2), 45–63.

Lester, M., Mallari, Z., Guintu, J. S., Magalong, Y. C., & Yap, D. S. (2022). CLIQUE: A web-based queue management system with real-time queue tracking and notification of units for Angeles University Foundation Office of the University Registrar. Proceedings of the 12th Annual International Conference on Industrial Engineering and Operations Management (IEOM 2022). IEOM Society International.

Mahanthi, A. N., & Jacob, T. P. (2023). Transforming traditional medical appointment booking with advanced technologies and user-centric design. 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN). IEEE. https://doi.org/10.1109/ICPCSN58827.2023.00277

Mehmood Ali Mohammed, M., Mohammed, M. A., & Mohammed, V. A. (2022). Use of machine learning in optimizing medical appointment schedules. International Journal of Computer Science and Information Technologies, 14(1), 3–7.

Nallathamby, V., Balasubramanian, P., & Rajasekar, M. (2021). Optimizing appointment scheduling for outpatients and income analysis for hospitals using big data predictive analytics. Journal of Ambient Intelligence and Humanized Computing, 12(6), 6155–6165.

Noma, A. M., Musa, K. I., Mamman, H., Mato, A. D., Yusuf, A. A., & Sambo, M. A. (2022). Design of intelligent and secure hospital appointment scheduling system. 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable

Development (NIGERCON). https://doi.org/10.1109/NIGERCON54645.2022.9803093

Okoshi, T., Tsubouchi, K., Taji, M., Ichikawa, T., & Tokuda, H. (2020). Real-world large-scale study on adaptive notification scheduling on smartphones. IEEE Access, 8, 23,1978–231,992.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Pinedo, M. L. (2016). Scheduling: Theory, algorithms, and systems (5th ed.). Springer.

Qaffas, A. A. (2019). Design and implementation of an online appointment management system for higher education. International Journal of Computer Applications, 52(15), 1–8.

Salah, H., & Srinivas, S. (2022). Predict, then schedule: Prescriptive analytics approach for machine learning-enabled sequential clinical scheduling. Computers & Industrial Engineering, 169, 108270.

Saraswathi, P., Kumar, M., & Devi, R. (2023). Android-based smart appointment system for academic counselling. International Journal of Engineering Research and Technology, 12(6), 234–241.

Sharma, R., Patel, A., & Gupta, S. (2024). Optimizing doctor availability and appointment allocation through AI: A machine learning approach to reducing no-shows. Healthcare Informatics Research, 30(1), 18–29.

Srinivas, S., & Salah, H. (2021). Consultation length and no-show prediction for improving appointment scheduling efficiency at a cardiology clinic: A data analytics approach. International Journal of Medical Informatics, 145, 104290.

Suresh, K., Raj, A., & Mohan, P. (2019). BOOKAZOR: A web-based appointment booking and scheduling application. International Journal of Innovative Technology and Exploring Engineering, 8(10), 3456–3461.

Valenzuela-Núñez, C., Latorre-Núñez, G., & Troncoso-Espinosa, F. (2024). Smart medical appointment scheduling: Optimization, machine learning, and overbooking to enhance resource utilization. IEEE Access, 12, 7551–7562. https://doi.org/10.1109/ACCESS.2024.3349953

Williams, M., Brown, K., & Thompson, J. (2018). Investigation of online appointment scheduling system impact on quality of working life at the University of Technology Jamaica. Journal of Health Management, 20(3), 312–325.

APPENDICES

Appendix A: Source Code Repository

The complete source code for the GO-AMG system is available at:

https://github.com/Chidals001/go-amg

Appendix B: Demo Login Credentials

Student: adaeze.nwosu@gouni.edu.ng | Password: password123

Staff: franklin.okebanama@gouni.edu.ng | Password: password123

Admin: anyanwu.favour@gouni.edu.ng | Password: password123

Appendix C: System Startup Instructions

Terminal 1 (Backend): cd backend && node server.js

Terminal 2 (AI Service): cd ai && source venv/Scripts/activate && python app.py

Terminal 3 (Frontend): npm run dev

Open browser at: http://localhost:5173