# UNI-CONNECT: AN AUTOMATED COMMUNICATION-BASED ACADEMIC SUPPORT SYSTEM FOR McPHERSON UNIVERSITY

**BY**

**OLALERE DAHUNSIMI JOSEPH**

**MATRIC NUMBER: 210202024**

**A PROJECT SUBMITTED TO THE**

**DEPARTMENT OF COMPUTER SCIENCES,**

**COLLEGE OF COMPUTING**

**McPHERSON UNIVERSITY, SERIKI SOTAYO, OGUN STATE, NIGERIA**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCES (B.Sc.) DEGREE IN COMPUTING SCIENCE**

**JULY, 2024**

# CERTIFICATION

This is to certify that **Olalere Dahunsimi** with matriculation number **(210202024)** of the Department of Computer Science, College of Computing, McPherson University, Seriki Sotayo, Ogun State carried out this research project research work **"Uni-Connect: An Automated Communication Based Academic Support System For McPherson University"** in partial fulfilment of the requirements for the award of Bachelor of Science in Computer Science of McPherson University, Ogun State, Nigeria.

Mr. Oluwasanya Oyedele Date
Project Supervisor

Dr. Kayode Oladapo Date
Head of Department

## DEDICATION

I dedicate this work to God, the giver of every good thing, for His guidance and blessings. To my parents Mr. and Mrs. Olalere, for their unending sacrifice and unwavering support, and to myself, for the hard work and dedication I have put into this project.

I also dedicate this project to my late brother, Oluseye Olalere, whose memory continues to inspire me, and to my elder sister Esther Olalere, for her constant encouragement and support.

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to everyone who has supported and guided me throughout the development of this project, Uni-Connect.

First and foremost, I extend my deepest appreciation to my project supervisor, Mr. Oyedele, for his guidance, encouragement, and insightful feedback, which were instrumental in the successful completion of this project and to the Dean and H.O.D respectively Dr. Funsho Alowolodu, Dr. Kayode Oladapo, and to my lecturers Dr. Kayode Olumurewa, Dr Kayode Akinola, Dr. Oladapo Adeduro, Dr. Adeshola Falade, Mr. Joachim Briamah, Mr Daniel Aina, Mrs. Ayodele Awokoya, Mr Daniel Jolayemi, Mr. Olusegun Odugbesan.

I would like to acknowledge my family and friends for their unwavering support and encouragement throughout this journey. Their patience and understanding have been a constant source of motivation. Special thanks go to Mr.Sunday Jayieola, Mr. Ayobami, Phoebe Olalere, Danjuma James, Anetor Oselumese, Olarinmoye Daniel, Agbi Damilola, Oke Timothy, Ajuwon Mariam, Olaniyi Adeyosola, Omilani Israel, Ajani Ifeoluwa, Akinwande Adebola and Olaniyi Enikeoluwa for their direct and indirect contributions to the success of this project.

## TABLE OF CONTENTS

## LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

The Uni-Connect system is an innovative, automated academic support platform developed for McPherson University, designed to enhance communication and efficiently manage academic activities. This web-based solution joins user authentication, appointment scheduling, notifications, and QR code-based attendance tracking to streamline administrative tasks and foster better interactions. Built using Flask and MySQL, Uni-Connect ensures secure data management and a user-friendly interface. Adhering to the System Development Life Cycle (SDLC), this project aims to facilitate seamless communication among students, lecturers, and parents, optimize attendance tracking, and effectively manage academic schedules.

Uni-Connect also integrates analytics and reporting features to support data-driven decision-making and academic planning. The platform is equipped with tools for lecturers to manage courses, assignments, and assessments efficiently. The robust notification system ensures that students and staff are promptly informed of important academic updates and reminders. By integrating with existing university systems and databases, Uni-Connect ensures streamlined operations and minimal disruption. Furthermore, the system incorporates robust security measures to protect user data and ensure privacy. The deployment of Uni-Connect is poised to significantly elevate the academic experience at McPherson University, fostering a more connected, efficient, and supportive educational environment.

# CHAPTER ONE

# INTRODUCTION

## 1.1 Background of Study

This chapter describes Uni-Connect an academic support that would enhance student tracking and parental and lecturer engagement in the academics of students in McPherson University. The automated grade notification system, communicating in real time through the booking appointment system, and an attendance system with active participation of all the users in the learning environment. Parents and lecturers' involvement in the academics of the students is needed to enhance performance among students by Doi *et al*. (2020). The focus of the following management plans of McPherson University is to retain the hospitable environments and integrate the fully automated systems more into the parental involvement and students' progress section since McPherson University acknowledges that academics are appreciated but the institution is finding it difficult to effectively communicate and support parents and students. The involvement of parents in academic activities by students is now becoming a rare thing; this is a problem affecting many universities across different countries by Boonk *et al*. (2018). The education community has a challenge of developing a useful parental involvement system that affects the students' learning achievement at the bachelor's level. The project then prompted the introduction of Uni-Connect, an application used in the tracking of students' performance, as well as the involvement of parents at McPherson University. This means that Uni-Connect shall assist in providing a framework of support to parents, students and learners geared toward the achievement of academics by filling a communication void between the abovesaid parties by utilizing the instruments available in the modern world. Therefore, the objective of this research is to add to the developing literature as to how Higher Education Institutions can enhance parental engagement and support. Consequently, I was able to establish the key factors which limit the involvement of parents in matters of their wards performance in school. In a bid to uncover the challenges associated with the level of parental involvement at the university this study aimed at identifying them as having to do with conflicts in terms of schedules and available time and the general lack of understanding among parents on the connection between their engagement and the student's grades (Ogg & Anthony, 2020). With the help of Uni-Connect, McPherson University can create new appropriate learning environments, which will increase its cooperation with parents and schools. The following can be attained through the use of Python for automation; Automated systems. This is among the specializations of Python where it excels in creating automated systems.

## 1.2 Statement of Problem

One of the problems is the level of parental engagement in academic affairs in McPherson University; the aspect of students' performance has been negatively affected by the reduced parental engagement in their academic activities. The obstacles include conflict of schedule, time limitation, and even the parents' unawareness on the relationship of their involvement to their children's academic performance (Alinsunurin, 2020). These barriers affect the communication between parents, lecturers, and students, thus the formation of a favorable academic climate (Austin-Ohanenye, 2021). Besides, the communication and academic support mechanisms that exist in McPherson University are on the decline due to the following gaps. To this end, this study aims at developing Uni-Connect, an automated platform that is intended to fuel communication, parental participation, and students' support. The study shall seek to determine the current extent of these challenges, the effects, if any, on the students' performance, and probable measures to enhance the cooperation between the parents and lecturers in the academic fraternity.

## 1.3 Aim and Objectives

### 1.3.1 Aim

The aim of this project is the integration of an automated support system, known as Uni-Connect, to increase parents and lecturers' engagement in the students' academic performance at McPherson University. The goal of the project is to decrease communication disruptions, enhance parents' knowledge, and analyze difficulties in current academic assistance, which would lead to a comprehensively united academic climate.

### 1.3.2 Specific objectives

The specific objectives of this project is to extend an automated support system, thus termed Uni-Connect, to foster parents' and lecturers' involvement in students' academic accomplishment at McPherson University. The specific objectives are:

1. **Decrease communication disruptions:** Minimize the breaks and the distances between students, parents, and the lecturer.

2. **Enhance parental knowledge:** In turn, parents need to be given all the information and educational material that will help their child in the learning process.

3. **Flexible system design:** Make certain that Uni-Connect is flexible enough to accommodate future requirements that would incorporate new technology or education trends at McPherson University.

## 1.4 Significance of Study

Thus, the significance of this study cannot be overemphasized regarding the different stakeholders in the academic world and other related areas. The following aspects highlight the importance of conducting this research. The following aspects highlight the importance of conducting this research:

1. Advancing parental involvement.
2. Enhancing academic supportive services.
3. Bridging communication gaps.
4. Enhancing Academic success.

## 1.5 Scope of Project

The understanding of the range of this project indicates that the integration of Uni-Connect at McPherson University will be consistent with the academic areas by ensuring that the key users, which are the parents, the lecturers, and the students benefit from aspects such as the application. Uni-Connect is the general approach to the automated addressing of different segments of the communication and support of learners to make the learning process closer and more caring. This project to reduce the intensity of communication barriers will involve the use of automated communication tools which include an attendance system through special coded numbers and also grading system which will inform the students on the grades they have scored.

## 1.6 Methodology Overview

The methodology for developing the Uni-Connect project follows a structured approach to ensure the delivery of a high-quality, reliable, and user-friendly academic support system. It is also important to note that the project uses the System Development Life Cycle (SDLC) method which is a logical framework of phased approach of software development.

## 1.7 Organization of Work

The project is divided into five chapters, each centered on another aspect of McPherson University's Uni-Connect development and implementation.

The first chapter provides an overview of the project, including its background, objectives, significance, and scope. It describes the difficulties faced by McPherson University in terms of academic communication and support, and introduces Uni-Connect as a solution to these issues.

The literature review examines existing research and developments in the field of automated academic support systems and parental involvements in higher education. It discusses various technological solutions and their impact on academic performance and communication. This chapter also explains the gap in the current research that Uni-connect aims to address.

Chapter 3 describes the research design and the methodologies used in the development of Uni-Connect. It covers the system requirements design specifications and implementation techniques. The literature review also includes details about the technologies used, such as python for automation, QR code scanning and the database management system using SQL queries.

The fourth chapter details the actual development process of Uni-Connect, including the coding, testing, and deployment stages. It provides a step-by-step account of how the system was built, the challenges encountered, and how they are addressed.

The final chapter contains the summary of the findings of the research, evaluates the success.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Introduction

This chapter presents an extensive analysis of the existing literature related to the development and implementation of automated communication-based academic support systems. The focus is on understanding the actual level of research on technology used for academic support, parental involvement in higher education, and the efficacy of automated systems in educational settings are the main points of emphasis.

## 2.2 Parental Involvement in Higher Education

According to the extent of parental involvement, the social and academic performance of students is affected by Berkowitz *et al*. (2017). It has been a well understood fact that parent's engagement plays an important part to raise the student's academic performance and accomplishment. Analysis from prior research shows that children who have engaged parents perform better in their academies (Kim, 2019). However, issues like tight working schedules, lack of communication, physical working base which is far from the school pose some of the challenges that hitherto do not allow parents germane influence on their wards higher education.

### 2.2.1 Benefit of parental involvement

Studies have conducted a number of times in identifying the impact of parental involvement on students' achievement, truancy and social behavior. These benefits also extend to the tertiary institution since involved parents can provide their children with the much encouragement by Tan *et al*. (2019). Further, Parents involvement in education makes the parent's relations of parent and child much stronger, create common grounds for smooth communication and healthy trusting relations. Thus, parental involvement helps educators and schools to establish the synergistic partnership to facilitate parents in offering their recommendations, contributing to the should activities, and solving problems, which contributes to the formation of the coherent school environment. The above benefits reveal a rationale for building a strong network among parents, communities, and schools in order to raise achievements in schooling, thus improving the wellbeing of students.

### 2.2.2 Barriers to parental involvement

Despite the important benefits of parental involvement in higher education, several barriers can impede its effectiveness. Understanding these problems is very important for creating strategies to foster engagement and support for students.

**1. Communication gaps:** Effective communication between parents and educational is critical for meaningful parental involvement. However, several academic environments are characterized by inefficiency and restricted access to communication with other people, resulting in information deficiencies by Boonk *et al*. (2018).

**2. Lack of awareness:** Some parents may not be very much informed on how they can canget involved or exactly what it means for them to be involved in their child ‘s education by Tan *et al.* (2019).

**3. Socioeconomic factors:** It is also influenced by socioeconomic status because parents of high status are likely to be more involved in their children’s education than parents of low status. Poor families also have some disadvantage because these families may not afford to buy resources, they may be financially challenged, and they may have no time for big changes because one has to go to work. A number of these may hamper their capacity to engage sufficiently in their child's education, even though they are willing and interested by Berkowitz *et al*. (2017).

### 2.2.3 Strategies to increase parental involvement

Various techniques have been proposed to increase parental involvement in higher education, such as:

1. **Flexible scheduling:** Offering events and meetings at different times to accommodate parents' schedules.

2. **Virtual communication Tools:** Utilizing technology to facilitate communication between parents and the university.

3. **Parental education programs:** Providing resources and workshops to educate parents on the importance of their involvement (Hornby & Lafaele, 2019).

## 2.3 Academic Support Systems

Tutors and educational support structures have the central function to help students to reach their academic goals. It includes tutoring programs, advising services and learning resources that are available to aid the learners in their learning process. Other conventional models of academic assistance though useful, do not have the degree of integration and immediacy that is necessary not only to meet modern student's needs for help but to address all the learners' needs in academic setting.

### 2.3.1 Traditional academic support systems

Conventional learning assistance arrangements require students and instructors to be in the same room at the same time, perhaps in the form of a tutoring session, a professor's office hours, or academic counseling appointments. Although all these methods work efficiently, they are bound by issues to do with availability of resources and timetables.

### 2.3.2 Technology-enhanced academic support

The conventional form of academic assistance is generally derived from appointments consisting of tutor-student, faculty-student, and academic counselor-student conferences. Though, such methods do work, they can hardly be employed due to the lack of resources, and sometimes organizational schedules by Means *et al*. (2018).

### 2.3.3 Challenges in implementing technology-enhanced support

However, there are some difficulties related to such usage of technology in enhancing the processes of continuing education. These challenges include:

**1. Digital divide:** Technology, in particular, the internet is not available to all students and those who have access do not use it to the same expected extent.

**2. Resistance to change:** Another challenge is that there can be resistance from some of the students or Faculty in embracing technology.

**3. Data privacy concerns:** This is because, the protection of student's data is an important factor that cannot be taken lightly.

## 2.4 Automation in Education

Education process automation entails use of different technologies to deliver educational services that would otherwise be accompanied by some level of human intervention. This comprises of auto grading tools, class attendance and participation tools and communication tools. Application of AI, particularly ML has already taken the wheel of advancement of increased automation in education. Research has found that the utilization of the automated systems can enhance the efficiency, decrease the amount of administrative burden, and offer the feedback to the students as well as the educators, instantly by Luckin *et al*. (2018).

### 2.4.1 Automated grading systems

Automated grading systems use algorithms to evaluate student work, providing immediate feedback and reducing the grading burden on instructors. These systems can be particularly useful for large classes where manual grading is time-consuming (Balfour, 2019).

## 2.5 Technology-Enhanced Learning

Technology enhanced learning (TEL) is the intent use of tools and media to support learning. This is encompassed by Learning Management Systems (LMS), virtual classroom and other collaborative tools. TEL also positively impacts students' engagement, and multiple ways of learning and teaching. Combining TEL with automation create a learner supporting environment in which students can study with help which comes from the technology at their disposal (Kirkwood & Price, 2020).

### 2.5.1 Learning Management Systems (LMS)

Software like Moodle and Blackboard supports broadly the LMS and other tools that involve content delivery, submissions of assignments, and grading. These systems support blended and online learning environments by Coates *et al*. (2021).

### 2.5.2 Virtual classrooms

Virtual classrooms enable live, interactive classes over the internet. Tools such as Zoom and Microsoft Teams facilitate real-time teaching and collaboration, making education more accessible to remote students Martin *et al*. (2020).

### 2.5.3 Digital collaboration tools

Application which includes Google Workspace and Microsoft Office 365may be used in order students and educators to carry out their projects and assignments. These tools enable other features of collaborative work and document exchange (Almala, 2020).

## 2.6 Existing Solutions and Gaps

Several existing solutions exist to improve the communication and academic assistance in higher education. Other tools and systems in use include learning management systems such as the moodle and blackboard which ensures that management of course and interaction between the teacher and students is well done by Coates *et al.* (2021). But, such systems create less opportunities for the specific features of Parental Engagement and Management and the Maintenance of Automatic Support. Remind and ClassDojo are similar to OneNote in that are communication tools, but are more commonly used in K-12 education than in higher education.

## 2.7 Related Works

Johnson *et al.* (2020) proposed a study on parental involvement and perceptions of school climate in California, specifically targeting middle and high schools. The objectives were to investigate the impact of parental involvement on teacher-student relationships and evaluate the intersections of race and class on policy development and implementation. The methodology involved surveys from parents to gather their perspectives on school climate, parental involvement, and student outcomes. Results highlighted the importance of ongoing data collection from parents to monitor trends and changes in parental attitudes over time. However, there was a gap in the limited information available on parents and families that could influence their school experiences and involvement in their children's education.

Kim (2019) conducted a meta-analysis to investigate the relationship between parental involvement and academic achievement in East Asian countries. The focus was on a context different from previous studies, which mostly included research conducted in the United States. The methodology concentrated on studies from East Asian countries characterized by high achievement levels, a standardized education system, and no policies encouraging family-school relations. The meta-analysis revealed a positive association between parental involvement and academic achievement. A notable gap identified was the lack of meta-analytic studies in East Asian contexts, which differ significantly from Western educational systems.

Tan, Lyu, and Peng (2019) evaluated the assumption that parental involvement benefits students' achievement regardless of their socioeconomic status (SES). This meta-analysis of 98 studies published between 2000 and 2017 examined if patterns of associations between specific parental involvement variables and the academic achievement of K-12 students varied with parental SES as measured by educational level. Results showed six specific aspects of parental involvement positively associated with student achievement. However, the study identified a gap in the assumption that these benefits were uniform across different SES levels.

Fantozzi *et al*. (2019) addressed challenges posed by climate change in building design, focusing on bioclimatic design strategies and the integration of Building Automation Control Systems (BACS). The study reported positive results in terms of indoor adaptive thermal comfort, with simulated operative temperatures complying with the adaptive comfort model for more than 98% of the reference year. However, a gap was identified in understanding the long-term performance and maintenance considerations of buildings with integrated bioclimatic design and BACS.

Qureshi *et al*. (2021) analyzed Internet of Things (IoT) solutions for schools, focusing on enhancing security and privacy within educational settings. The study conducted a review of existing literature on security and privacy concerns in educational organizations, smart learning technologies, and current IoT solutions. Results indicated that the S-IoST system enhanced security within educational institutions through advanced communication systems and IoT technologies. The identified gaps suggest potential avenues for further research to refine and validate the proposed S-IoST system.

Wook et al. (2019) developed the OMFeedback system to automatically analyze student feedback and assess lecturers' teaching performance. The system utilized the Vader Sentiment Intensity Analyzer to analyze feedback and calculate scores based on values in the Vader Lexicon. Results showed that 49.3% of the students had positive opinions, 49.2% had neutral opinions, and only 1.5% had negative opinions about their lecturer's teaching performance. The gap identified was the reliance on machine learning techniques, whereas a lexicon-based approach incorporating capitalization of words and emoji characters could improve textual feedback analysis.

Hornby and Blackwell (2018) investigated the impact of online learning tools on the academic performance of undergraduate students during the COVID-19 pandemic. Using a quantitative research design, the study surveyed 500 undergraduate students across five universities and analyzed the data using statistical software. Results indicated that students who consistently used online learning tools scored, on average, 10% higher on their exams compared to those who did not use these tools. The study identified a gap in the availability of digital resources for students in rural areas.

Hynninen, Knutas, and Hujala (2020) performed sentiment analysis on a large set of open-ended course feedback from university courses collected between 2016 and 2019. The study indicated that over 80% of the feedback was positive, and less than 20% was negative. The identified gap was that while sentiment analysis can be an indicator of student satisfaction, it does not provide actionable themes like other computational approaches, such as topic modeling.

Onan (2019) utilized a deep learning approach to mine opinions from instructor evaluation reviews. The study involved a text corpus containing 154,000 reviews analyzed using conventional classification algorithms, ensemble learning methods, and deep learning architectures. Results indicated statistically significant differences in predictive performance among the methods used. The gap identified was the need for efficient sentiment classification of instructor evaluation reviews using advanced machine learning techniques.

Gottipati, Shankararaman, and Lin (2018) extracted course improvement suggestions from students' feedback using a text analytics approach. The study combined opinion mining and natural language processing to develop a solution model for suggestion extraction. The results included the development of a tool for extracting and visualizing suggestions from student comments. However, the identified gap was the need for a more comprehensive and systematic approach to analyzing qualitative feedback from students.

Motz, Mallon, and Quick (2021) aimed to reduce missed assignments and improve student outcomes in online learning environments through a mobile app intervention. The study conducted two experimental pilots to test the effectiveness of the intervention. Results showed that the mobile app significantly decreased missed assignments and improved course grades compared to the control group. The gap identified was whether the positive effects of the intervention would persist over the long term.

Ibrahim, Bader-El-Den, and Cocea (2018) developed a data mining framework to analyze end-of-unit general textual feedback and classify instances into assessment-related and non-assessment-related subsets. The study used four machine learning algorithms to analyze the feedback data. Results showed higher accuracy for general feedback data set models compared to assessment-related models. The gap identified was the lack of significant difference between using unigram and bigram features in building the models.

Doi, Isumi, and Fujiwara (2020) investigated the relationship between parental involvement behaviors and the self-esteem of adolescents living in poverty. The study involved students from the Kochi Child Health Impact of Living Difficulty (K-CHILD) study in Japan. Results indicated a positive association between the number of parental involvement behaviors and self-esteem scores of adolescents, both in poverty and not in poverty. The gap identified was the need for further research on how parental involvement can be tailored to support adolescents in different socioeconomic contexts.

# CHAPTER THREE

# METHODOLOGY

This chapter provides a detailed explanation of the methodologies and technologies used in the creation of an academic support system is provided in this chapter. It includes the overview of the system, its architecture, and the way the chosen web application is built with help of Flask, MySQL, and any other related instruments. The methodology describes the procedure that has been followed so that the system will deliver the desired attributes and fulfil the intended purpose of the identified target group that include the students, tutors, and parents.

## 3.1 Overview

The academic support system is a computerized program, which is aimed at ensuring effective students' and parents' communication and their interaction with the teachers. The LMS includes several features like user identity confirmation, booking of appointments, message issuing, use of QR codes for attendance, and students' feedback. All these features in umbrella strategy seek to improve one's academic journey by offering well-organized and time-saving ways of handling and communicating academic matters.

## 3.2 Technologies Used

The development of the academic support involved the use of several technologies to make the application efficient and easily scalable. The primary technologies used are:The primary technologies used are:

**1. Flask:** A light-weight yet, feature-rich web framework that helps in developing the core web application, Flask is used. It is chosen because; it is easy to use, flexible and does not have many complex features.

**2. MySQL:** One of the most popular RDFMs, MySQL is used for the data storage of users, appointments, notifications, and attendance. Owing to reliability and performance, this type of data is suitable for managing the system's tasks.

**3. Werkzeug:** An extensive utility package designed to be integrated with the WSGI micro web application framework improving on the utility set of Flask.

**4. QR code generation and scanning:** The mobile application utilizes libraries such as qrcode for creating QR codes, and cv2 which is a part of the OpenCV for scanning and decoding the QR codes that will allow tracking of the attendance feature.

**5. Session management:** Achieved with the help of the Flask framework used by the application to handle the current user sessions and roles.

## 3.3 Three Tier Architecture

The workflow of Uni-Connect academic support system in Figure 3.1 is divided into three-tier based on scalability, maintainability and data management. This architecture divides the system into three distinct layers: and these include the Presentation Layer, the Application Layer and Data Layer.

### 3.3.1 Presentation layer

The Presentation Layer commonly referred to as User Interface Layer is largely involved in the presentation of data to the users, in addition to capturing data entered by the users. This layer is with the help of HTML, CSS, and JavaScript that students, lecturers, and parents can have easy and convenient interfaces. Hence, in the framework of the Uni-Connect system, the function of the Presentation Layer can be outlined as follows.

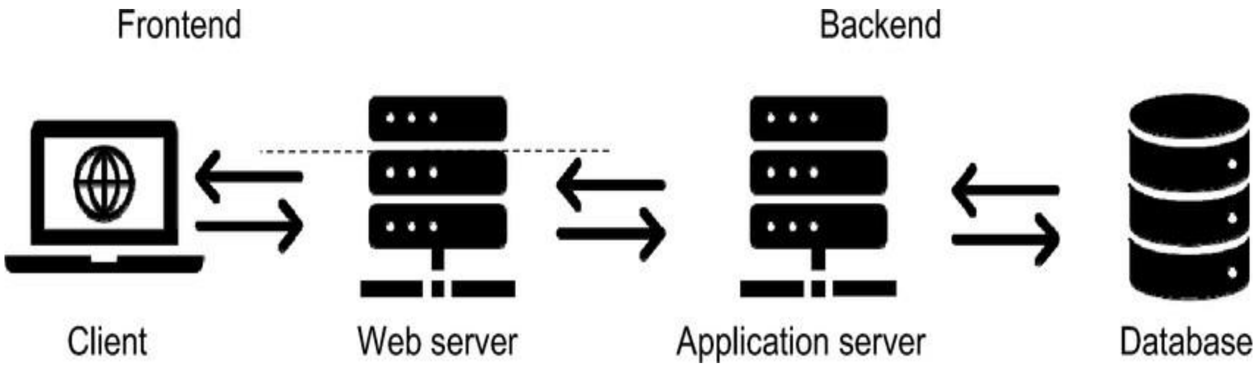


**Figure 3.1: Three tier architecture (Almala, 2020).**

It offers distinct views for different users, which means children, teachers, and parents will be able to work with the system conveniently. This layer is also in charge of inputs received from the users for controlling core features like identification, meeting setup, attendance, and reminders. The Presentation Layer aims at the clear presentation of the data and makes the users interface with the system quite elegant and easy.

### 3.3.2 Application layer

In the Application Layer, the Presentation Layer is interfaced with the Data Layer; data processing and business logic are performed, and data handling takes place. This layer is mainly built with the help of Flask framework that receptive to receiving requests, steps through data, and provides fitting responses. Also, another library called Werkzeug is incorporated into the flask to houses

utilities.

It can be said that the Application Layer is one of the principal elements of the Uni-Connect system as well as it contains several crucial components and functions. It handles the user requests and performs the required business operations so that the implemented system works as per the requirement. This layer is used to handle the authentication of users and also sessions which can allow a user into the system. Moreover, the Application Layer includes appointment making between students and lecturers, the creation as well as the scanning of QR codes for attendance tracking, and sends notifications to users regarding appointments, attendance, and other important updates. By efficiently coordinating these tasks, the Application Layer ensures the smooth functioning of the system.

### 3.3.3 Data layer

A Data Layer is a subcontractor in terms of information storage and retrieval, data retention and confidentiality. The major part of this layer is MySQL as it contains all the system data except for files and documents; users' information, appointments, attendance records, and notifications. Each of these functionalities is carried out by the Data Layer in the Uni-Connect system: record creation, reading, updating, and deletion. And it is also the guarantee of data's unification, verifying that only accurate and truthful information is stored in the database. This layer also handles interaction between various data objects like students, lecturers, courses and the courses' attendance records. With fast and safe data access with the use of SQL queries and ORM (Object Relational Mapping), the Data Layer supplements the functionality of the whole system.

### 3.3.4 Integration of layers

Thus, the three layers, namely Presentation, Application, and Data layers, maintain the functionality of the Uni-Connect system. The Presentation Layer simply receives data from the user and forwards the same to the application layer for further processing. The Application Layer carries the business logic of the system and can call to the Data Layer to process the necessary data or to store some data in there. The Data Layer deals with data storage and retrieval or, in other words, the 'storage space' for information data should be preserved in. Results obtained from the Application Layer are then passed back to the Presentation Layer in order to be presented back to users.

This threelayered architecture proves to be a strong and flexible architecture used by the Uni-Connect system that integrates proper data management, proper business flow and the proper flow to the users. This way of dividing the responsibilities of each layer allows for the easy management and evolution of the software system and therefore its sustainability.

### 3.4 Implementation Details

In this unit, all the essential information about the procedure of applying the Uni-Connect academic support system is disclosed. This describes the technologies used to develop the system, the development cycles to be used, together with the developmental process to be followed in developing the system. The chapter is divided into aspects of the development that are not related to the user interface, such as backend and development, database construction and integration, testing strategies, deployment, and the maintenance of the program.

Backend Development was characterized by several technologies used with a view of having an efficient program. Flask web framework was adopted because of its light weight nature and sleek design of this particular framework for the python language. Flask was used alongside Werkzeug, a WSGI utility library that augments Flask especially in assessment of requests and sending responses. MySQL was the choice of relational DBMS to persist data that needed to be reliable, and retrieved quickly at that point. Some of the major functions incorporated in the process were the security of the users and their permissions, the appointment making process, a notification mechanism, attendance calculation, and proficient database management. To manage the authentication of users Flask sessions were employed and the newly developed bcrypt for the passwords. Lecture booking enabled students to book appointments with the lecturers and the lecturers could also see the appointments made. Using Flask-Mail the notification system was capable to mailing the users in relation to date, attendance, and other relevant information regarding appointments.

Recording/papering of attendance was done using QR codes where the ‘qrcode’ library was used to generate the codes and ‘cv2’ for scanning the codes.

Frontend Development hence concentrated on creating an easy to use and adaptable web design using HTML, CSS, JP, JavaScript and Bootstrap. To enhance the usability of the application, the strategy of creating an intuitive user interface was adopted with an emphasis on students, lecturers, and parents’ device usage. User registration, login form, appointment booking form, and attendance scanning form were created, and the form validation was accomplished at the clientside by JavaScript. For rendering of dynamic content, templates created by the help of Flask’s templating engine Jinja2 allowed for proper integration of the data with the actual interface.

Database Design was relevant to the functionality of this particular system. One of the principles of the creation of the schema was to provide easier access to the data and their protection against integrity loss. These were Users with data like ID, name, email, password, and role, Courses to store details of the courses and lecturers assigned, Appointments for the booking and viewing of the appointments, attendance to record the attendance of students along with other details; and Notifications to handle the messages sent to the users. The ER diagram depicted some relation like one-to-many relation between lecturers and courses offered, many to many relation between the student and appointment made and one to many relation between the student and the attendance sheet.

The merging of the frontend and backend was done using the Flask routing techniques as well as, the template rendering. Information from the user via frontend was entered through HTTP requests to the backend, which interacted with the database to obtain the necessary data for introduction at the frontend. This guaranteed that all elements on the site were in some way integrated and that the content was dynamic with real time data. Deployment meant putting the system in a productive use and the procedures included the following. To overcome this, the application is hosted on a cloud-based, hence has the capability to self-scale and self-heal in case of failure. Some other requirements include installation of web server, database server and other general security measures that enable proper deployment of the application.

The implementation of the Uni-Connect academic support system followed a structured and methodical approach, leveraging various technologies and best practices to create a robust, scalable, and user-friendly application. The integration of backend and frontend components, efficient database management, comprehensive testing, and careful deployment and maintenance planning ensured the system met its objectives and provided a valuable tool for students, lecturers, and parents.

## 3.5 Design Approach

The development of the different components of the Uni-Connect academic support system is based on a design process to warrant for the creation of an effective, efficient, and easily expansible application. This approach includes: the system structure, or how it will be built; look and feel or the structure of the application's interfaces; the structure of the data to be stored; and protection of the data and system. The first and foremost intention is to design the system in such a way that is convenient for students, lecturers, and parents when using it and at the same time meet the functionality and usage requirements.

### 3.5.1 User interface design

The user interface design focuses on creating an intuitive and engaging experience for all users.

Key principles followed in the design include:

1. **Simplicity:** The interface is kept simple, allowing users to navigate the system easily.
2. **Consistency:** A layout, color scheme, and typography are maintained across all pages to provide a unified user experience.
3. **Responsiveness:** The design is responsive, making sure that the application is accessible and usable on d different devices, including desktops, tablets, and phones.

4. **Accessibility:** Considerations for accessibility are integrated into the design, ensuring that the system is usable by individuals with different abilities.

### 3.5.2 Database design

The database design for the Uni-Connect system is structured to ensure efficient data storage, retrieval, and security. By leveraging MySQL, a robust relational database management system, the design adheres to principles of normalization, relationships, indexing, and security. Normalization involves organizing the database to reduce redundancy and improve data integrity. The Uni-Connect database schema is normalized to at least the third normal form (3NF). Relationships between tables are established using primary and foreign keys, which help in efficient data retrieval and maintain data coherence. Key relationships in the Uni-Connect database include:

**a. Student table:**

1. Primary key: student_id

2. Contains student details such as name, email, and contact information.

**b. Lecturer table:**

1. Primary key: lecturer_id

2. Contains lecturer details such as name, email, and department.

**c. Course table:**

1. Primary key: course_id

2. Foreign key: lecturer_id (references Lecturer Table)

3. Contains course details such as course name and description.

**d. Attendance table:**

**1.** Primary key: attendance_id

**2.** Foreign keys: student_id (references Student Table), course_id (references Course Table)

3.Contains attendance records with fields for attendance date and attendance status.

**e. Notification table:**

1. Primary key: notification_id

2. Foreign key: student_id (references Student Table)

3. Contains notifications sent to students regarding appointments, attendance, etc.

### 3.5.3 Security consideration

Security is a paramount concern in the design and implementation of the Uni-Connect system. Several measures are taken to ensure the security of user data and the application:

**1. Authentication and authorization:** Secure authentication mechanisms are implemented using Flask's session management and bcrypt for password hashing. Role-based access control ensures that users have appropriate permissions based on their roles (students, lecturers, or parents).

**2. Data encryption:** Sensitive data, such as passwords, are encrypted before storage. Secure communication protocols (e.g., HTTPS) are used to protect data in transit.

**3. Input validation:** User inputs are validated to prevent common security vulnerabilities such as SQL injection and cross-site scripting (XSS).

**4. Regular updates:** The system is regularly updated with security patches and updates to mitigate potential vulnerabilities.

## 3.6 System Development Life Cycle

The System Development Life Cycle (SDLC) provides a structured methodology for the development of the Uni-Connect academic support system. This lifecycle ensures that the project is systematically managed and executed from inception to deployment and beyond. Each phase of the SDLC is crucial to ensuring the successful delivery and functionality of the system.

### 1. Planning phase

The planning phase is the foundational step where the project scope, goals, and objectives are defined. During this phase, the primary focus is on establishing what the Uni-Connect system will achieve and how it will meet the needs of its stakeholders, including students, lecturers, and parents. The scope of the project is determined by identifying the core functionalities such as user authentication, appointment scheduling, notifications, and attendance tracking. Detailed goals and objectives are outlined to guide the development process, ensuring that the system will enhance communication, academic tracking, and support. This phase also involves stakeholder

identification to ensure that the requirements of all involved parties are considered. A timeline and budget are estimated to allocate resources efficiently and set realistic expectations for project delivery.

### 2. Analysis phase

The analysis phase involves a comprehensive gathering and analysis of requirements from stakeholders. This process begins with collecting detailed requirements through interviews, surveys, and workshops. Functional requirements, such as the need for appointment scheduling and notification systems, are documented alongside non-functional requirements like performance and security. The requirements analysis ensures that all collected data is clear, complete, and actionable. This phase includes the creation of use case diagrams and user stories to visualize the interactions between users and the system. Additionally, a feasibility study is conducted to assess the technical, operational, and financial viability of the project, ensuring that the system can be developed within the given constraints.

**3. Design phase**

In the design phase, the system’s architecture and detailed design are developed based on the requirements gathered. The system architecture outlines the overall structure of Uni-Connect, including the presentation layer, application layer, and data layer. The design focuses on creating a robust and scalable architecture that supports the system's functionalities. The database design is a critical component, involving the creation of a schema that includes tables, relationships, indexes, and security measures to ensure data integrity and efficient retrieval. User interface design is also a key aspect of this phase, with mockups and prototypes created to ensure an intuitive and user-friendly experience. Technical specifications are documented to guide the development process, detailing the technology stack, design patterns, and coding guidelines.

**4. Implementation phase**

The implementation phase involves the actual coding and construction of the system based on the design specifications. During this phase, the development team writes code for both the frontend and backend of the application. Flask is used for the web application framework, while MySQL handles database management. QR code generation and scanning functionalities are implemented using appropriate libraries. Integration of various components ensures that the frontend interfaces seamlessly with the backend and that data flows correctly between the user interface and the database. Configuration tasks are carried out to set up server environments and deploy the application infrastructure.

**5. Testing phase**

The testing phase is dedicated to ensuring the quality and functionality of the system. It begins with unit testing, where individual components or modules are tested to verify their correctness. Integration testing follows to ensure that combined modules work together as intended. System testing is conducted to evaluate the entire application against the specified requirements, identifying any discrepancies or issues. User Acceptance Testing (UAT) involves real-world testing by end-users to confirm that the system meets their needs and expectations. Any bugs or issues identified during testing are documented and resolved to improve the system's stability and performance.

**6. Deployment phase**

The deployment phase involves the release of the Uni-Connect system to its production environment. This phase includes preparing for deployment by configuring servers and environments necessary for running the application. The system is launched for use by the end-users, and training is provided to ensure that they can effectively navigate and utilize the system's features. Documentation is created, including user manuals and technical guides, to support users and administrators. This phase ensures a smooth transition from development to live operation.

**7. Maintenance phase**

The maintenance phase is an ongoing process that ensures the system remains functional and relevant over time. This phase includes providing technical support to address user issues and answer queries. Regular updates and enhancements are implemented to fix bugs, improve performance, and add new features based on user feedback. Continuous monitoring of system performance and security is conducted to maintain operational efficiency and protect against vulnerabilities.

### 3.6.1 System requirement analysis

The System Requirement Analysis phase in Figure 3.2 is a critical step in the development of the Uni-Connect academic support system. This phase focuses on understanding and documenting the needs and expectations of all stakeholders to ensure that the system meets its intended goals and functions effectively in Figure 3.3.

#### 3.6.1.1 Use case diagram

A Use Case Diagram in Figure 3.4 is a visual representation used in software engineering to capture and illustrate the functional requirements of a system. It serves as a tool to describe how different types of users, known as actors, interact with the system to achieve specific goals or perform tasks. The primary components of a use case diagram include actors, use cases, and relationships.

**1. Actors:** These represent the different types of users or external systems that interact with the application. Actors can be human users, such as students, lecturers, and parents, or other systems that the application communicates with.

**2. Use cases:** These are the specific functionalities or services provided by the system. Each use case represents a particular task or goal that an actor wants to achieve. For example, use cases might include "Schedule Appointment," "Track Attendance," and "Send Notification."

**3. Relationships:** These illustrate how actors and use cases are related. Common types of relationships include:

**4. Association:** A direct link between an actor and a use case indicating that the actor interacts with the system to perform that use case.

**5. Include:** A relationship showing that one use case includes the functionality of another use case, representing a dependency.

**6. Extend:** A relationship indicating that a use case can extend or enhance the behaviour of another use case, typically adding optional features.

Use case diagrams provide a high-level overview of the system's functionality from an end-user perspective, helping stakeholders understand how the system will be used and what features are required. They are useful for capturing user requirements, designing system interactions, and communicating functionality during the development process.

#### 3.6.1.2 Sequence diagram

A sequence diagram in Figure 3.5 is a type of interaction diagram that illustrates how objects or components within a system interact with each other over time to accomplish a specific task or scenario. In the context of the Uni-Connect project, which involves an automated academic support system, a sequence diagram helps visualize the flow of interactions in Figure 3.6 between different system components, such as students, lecturers, and the backend services. A sequence diagram in Figure 3.5 is a type of interaction diagram that illustrates how objects or components within a system interact with each other over time to accomplish a specific task or scenario. In the context of the Uni-Connect project, which involves an automated academic support system, a sequence diagram helps visualize the flow of interactions in Figure 3.6 between different system components, such as students, lecturers, and the backend services.

#### 3.6.1.3 Activity diagram

An activity diagram in Figure 3.7 is a type of UML (Unified Modeling Language) diagram that visually represents the workflow of activities or processes within a system. It is particularly useful for understanding the sequence of actions and decision points that occur in a particular use case. For the Uni-Connect project, the activity diagram can illustrate various processes such as attendance marking, appointment scheduling, or notification handling.

#### 3.6.1.4 Entity relational diagram

An Entity-Relationship Diagram (ERD) in Figure 3.8 is a visual representation of the database structure, illustrating the relationships between entities within a system. For the Uni-Connect project, the ERD will capture the core entities, their attributes, and the relationships between them**.** The deployment phase involves the release of the Uni-Connect system to its production environment. This phase includes preparing for deployment by configuring servers and environments necessary for running the application. The system is launched for use by the end-users, and training is provided to ensure that they can effectively navigate and utilize the system's features. Documentation is created, including user manuals and technical guides, to support users and administrators. This phase ensures a smooth transition from development to live operation.

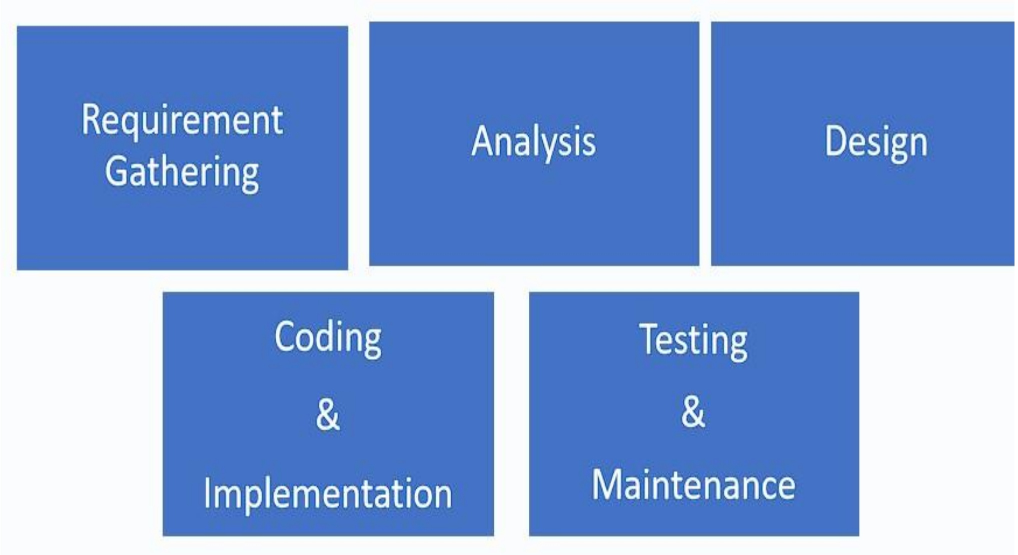


**Figure 3.2: SDLC phases of Uni-Connect**

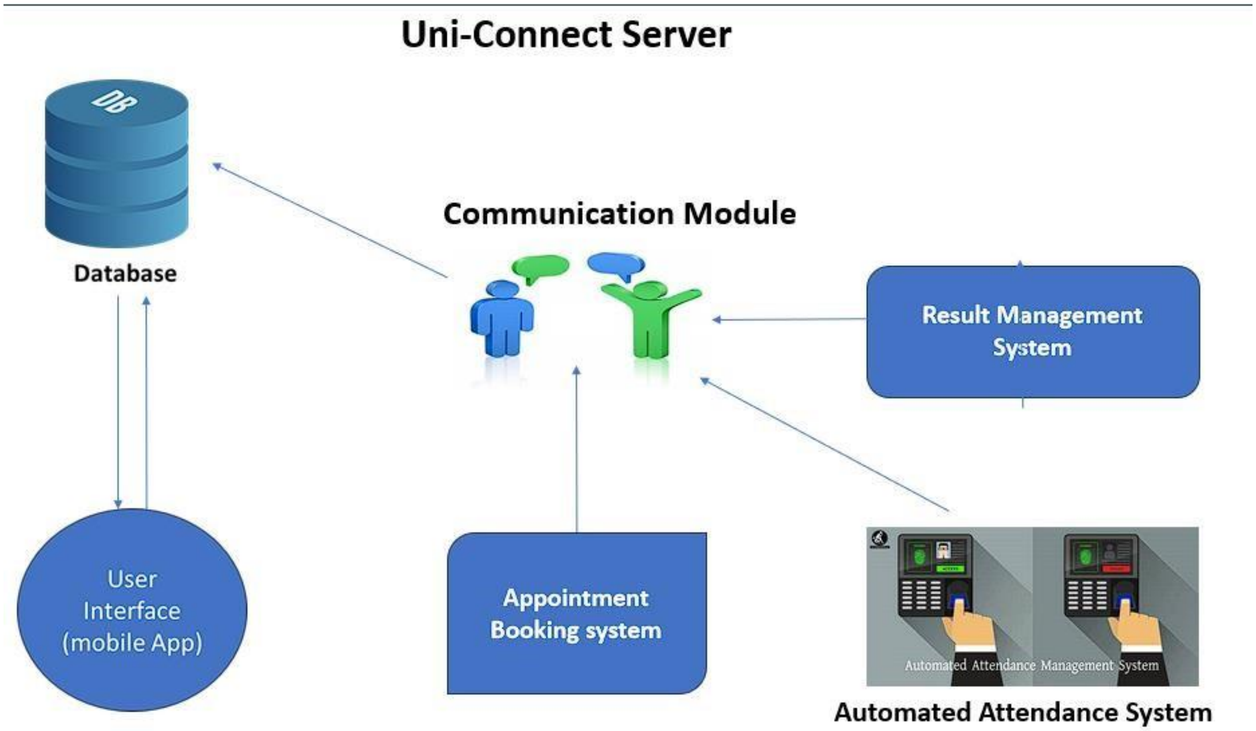

Uni-Connect Server
DB
Database
Communication Module
Result Management System
User Interface (mobile App)
Appointment Booking system
Automated Attendance Management System
Automated Attendance System

**Figure 3.3: System analysis of uni-connect**

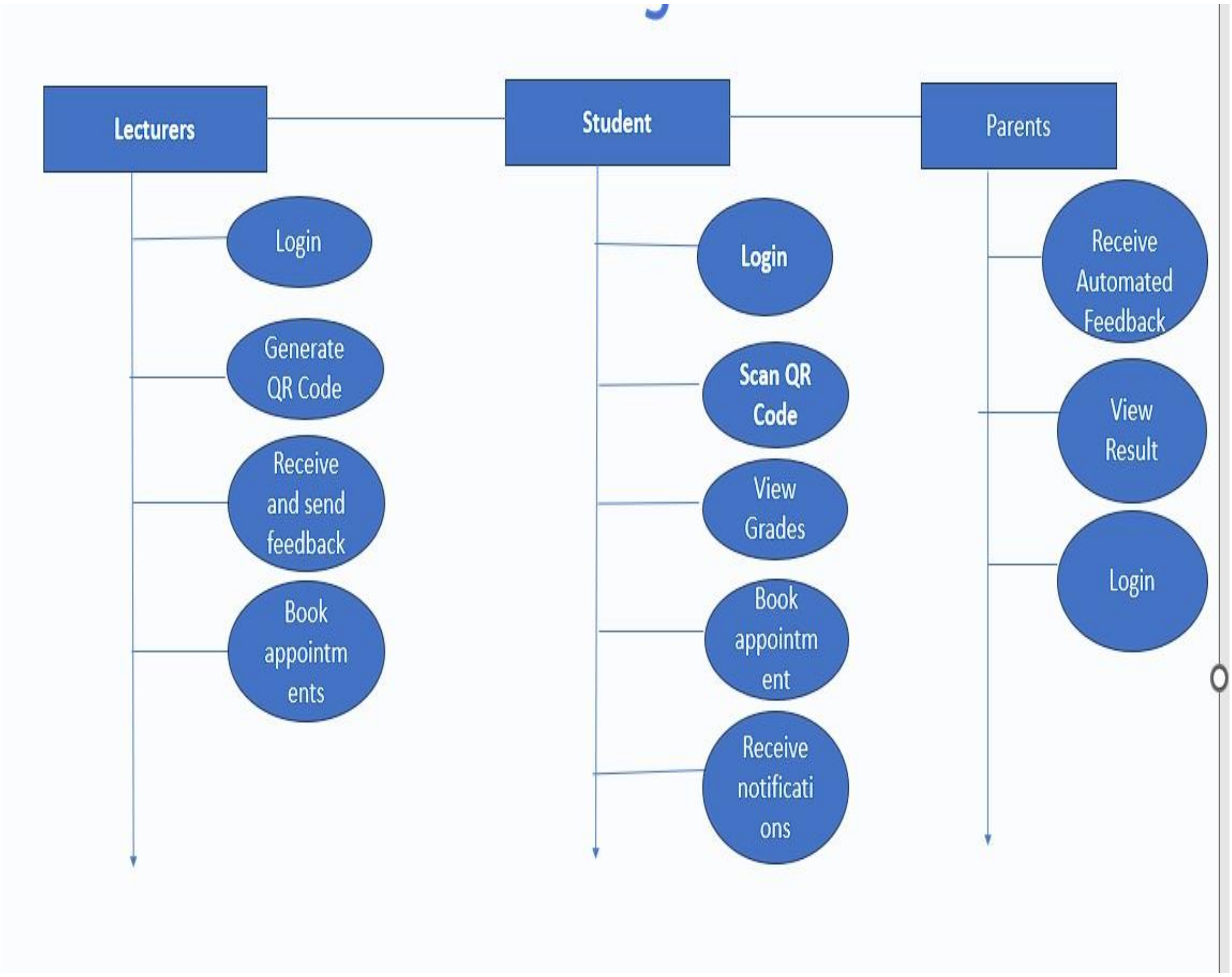


**Figure 3.4: Use case diagram**

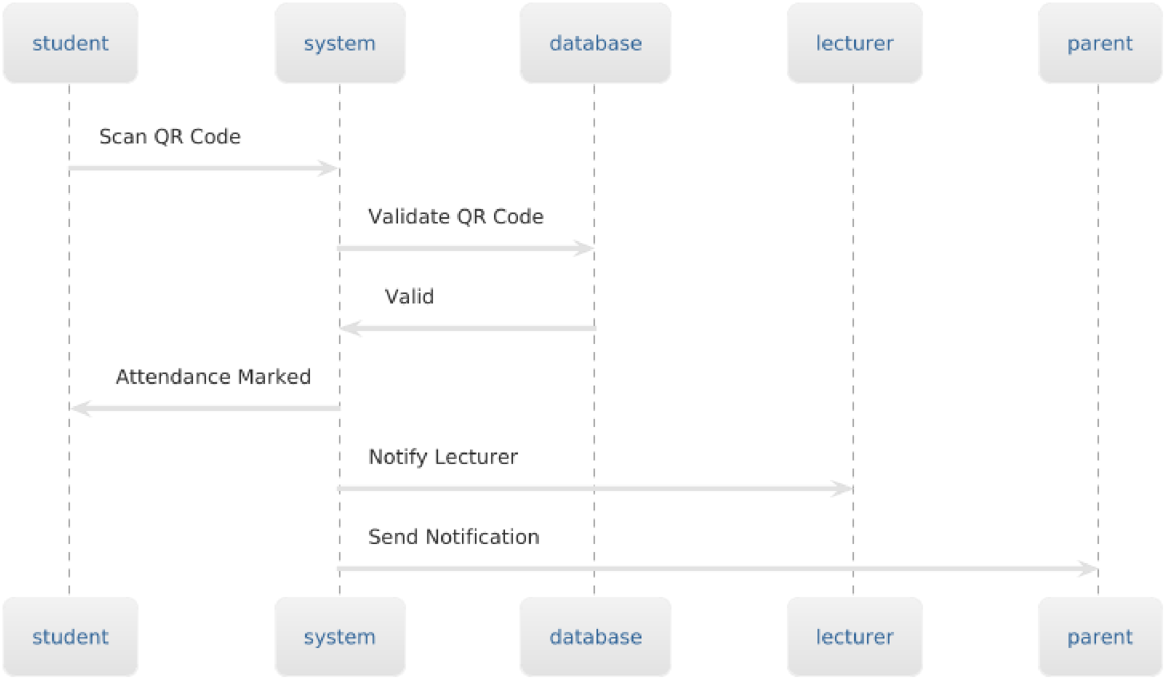


**Figure 3.5: Sequence diagram of uni-connect**

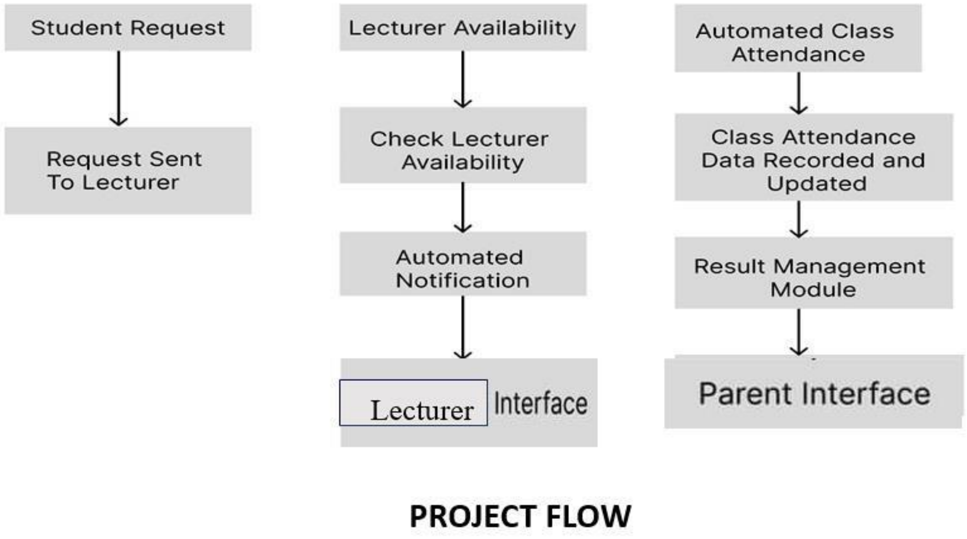


**Figure 3.6: Project flow of uni-connect**

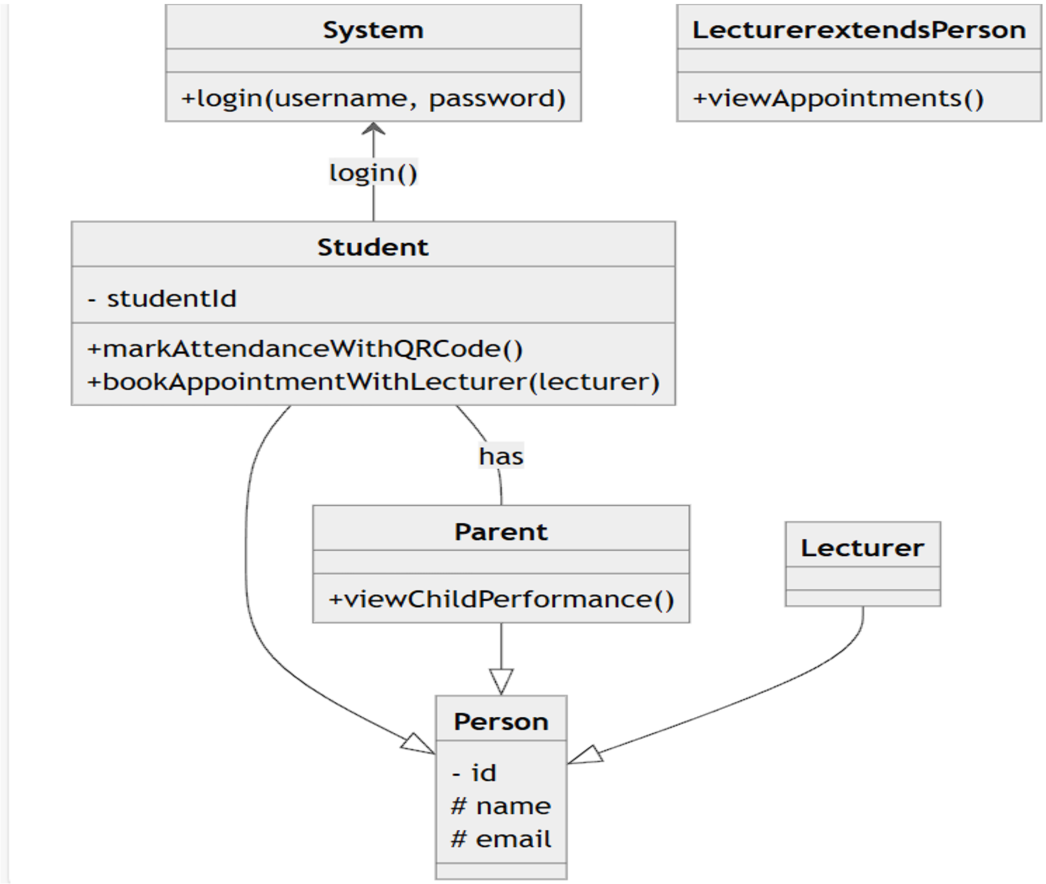


**Figure 3.7: Activity diagram of uni-connect**

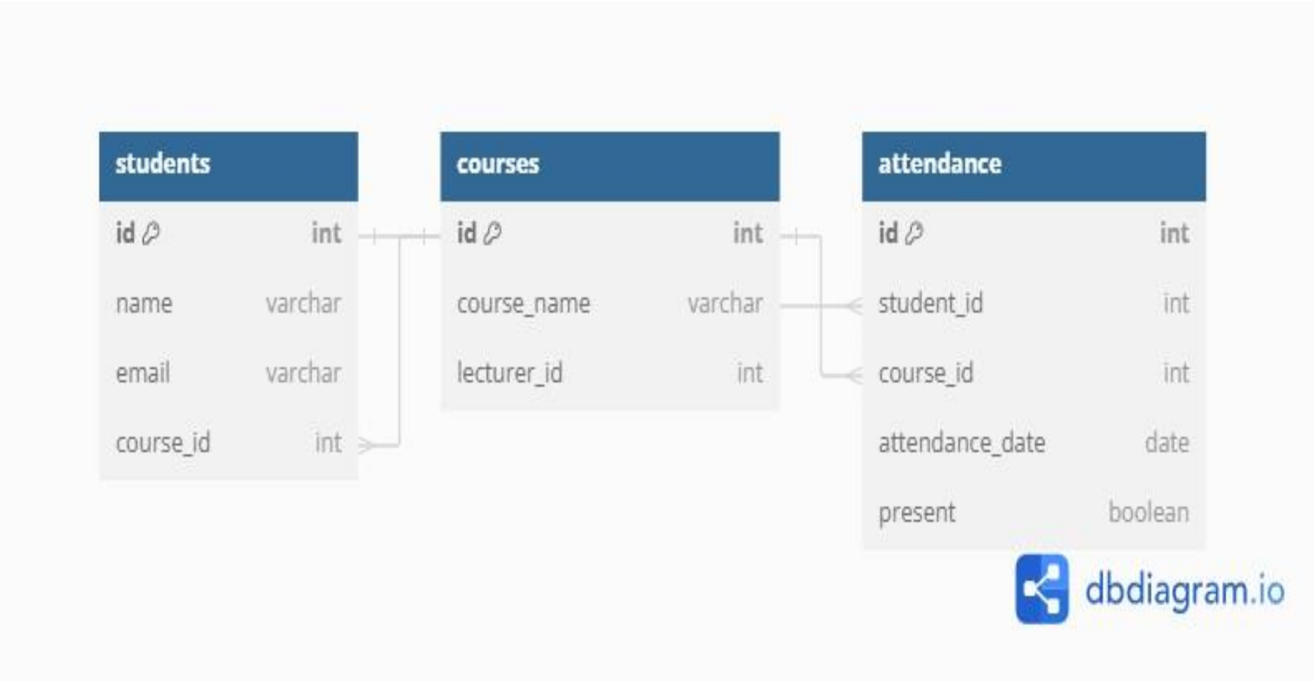


**Figure 3.8: Entity relational diagram**

## 3.7 Algorithm

The algorithm outlines the step-by-step process to perform key functionalities of the Uni-Connect system. Below are the main algorithms for core functionalities including user authentication, appointment scheduling, attendance tracking, and notification sending.

### a. User authentication

Algorithm:

1. Start
2. Input: Username, Password
3. Fetch user details from the database where Username = input_username.
4. Check if user exists:
5. Verify Password: Compare input password with stored hashed password.
6. Generate Session Token: Create a session token for authenticated users.
7. Store Session Token: Save the token in the session and database.
8. Redirect to User Dashboard: Based on the role, redirect the user to the appropriate dashboard.
9. End

### b. Appointment scheduling

Algorithm:

1. Start
2. Input: StudentID, LecturerID, Date, Time
3. Validate Input: Ensure all fields are filled and valid.
4. Check Availability:
5. Create Appointment Record:

6. Send Confirmation Notification:

7. End

**3. Attendance tracking**

Algorithm:

1. Start

2. Input: QR Code Image

3. Decode QR Code: Extract data from the QR code using an image processing library.

4. Parse Data:

5. Validate Data: Ensure extracted data is accurate and not empty.

6. Get Current Student ID: Retrieve the current student's ID from the session.

7.Check Existing Record:

8. Insert Attendance Record:

9. Send Attendance Notification:

10. End

**4. Notification sending**

Algorithm:

1. Start

2. Input: UserID, NotificationMessage

3. Fetch User Details: Retrieve user contact details (email/phone) from the database using UserID.

4. Format Notification Message: Create a structured notification message based on the input.

5. Send Notification

6. Record Notification

# CHAPTER FOUR

# IMPLEMENTATION AND RESULT

## 4.1 Introduction

Chapter Four of this project, titled "Implementation and Results," delves into the practical aspects of bringing the Uni-Connect system to life. This chapter outlines the systematic approach taken to develop the system, from translating the design specifications into a working application to ensuring that the system meets all functional requirements. It includes detailed descriptions of the system's software and hardware requirements, the technologies and tools used in development, and the methodologies adopted to achieve an efficient and effective implementation.

The implementation phase is crucial as it transforms theoretical designs into a functional system that users can interact with. This chapter will document the step-by-step process followed during development, highlighting key milestones and challenges encountered. Additionally, it will showcase the results of the implementation, demonstrating how the system functions in real-world scenarios through various tests and user interactions.

The primary focus of this chapter is to provide a comprehensive overview of the technical aspects of the Uni-Connect system, ensuring that every component works seamlessly together to create a reliable and user-friendly academic support platform. Through detailed explanations and illustrations, this chapter aims to give a clear understanding of how the system was built and the outcomes achieved from its implementation.

## 4.2 System Specifications

The system specification for Uni-Connect outlines the detailed hardware and software requirements necessary to develop, deploy, and maintain an efficient academic support platform. These specifications ensure that the system can handle its functionalities effectively, providing a seamless experience for all users, including students, lecturers, and administrators.

### 4.2.1 Hardware requirements

#### a. Server requirements

1. Processor: Quad-core Intel or AMD processor, 2.5 GHz or faster

2. RAM: Minimum of 8 GB

3. Storage: Minimum of 500 GB SSD 4. Network: Gigabit Ethernet

#### b. Client requirements

1. Processor: Dual-core Intel or AMD processor, 2.0 GHz or faster

2. RAM: Minimum of 4 GB

3. Storage: Minimum of 20 GB available space

4. Network: Stable internet connection (minimum 5 Mbps)

### 4.2.2 Software requirements

#### a. Server-side requirements

1. Operating System: Linux (Ubuntu 18.04 or later) or Windows Server 2016 or later

2. Web Server: Apache 2.4 or Nginx 1.14 or later

3. Database: MySQL 5.7 or later

4. Programming Language: Python 3.6 or later

5. Framework: Flask 1.1.2 or later

6. Libraries: Werkzeug, qrcode, OpenCV (cv2), SQLAlchemy

**b. Client-side requirements**

1. **Web browser:** Latest versions of Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge
2. **Operating system:** Windows 10, macOS Mojave or later, Linux distributions, Android 9.0 or later, iOS 13.0 or later
3. **Frameworks/libraries**: HTML5, CSS3, JavaScript (ES6), Bootstrap 4 or later.

## 4.3 Result and Discussions

### 4.3.1 Landing page

The landing page of Uni-Connect in Figure 4.1 serves as the primary entry point for users, providing an overview and access to the platform's key features. Designed with a user-centric approach, the landing page aims to facilitate easy navigation and provide a welcoming interface for students, lecturers, and parents.

### 4.3.2 Login and signup page

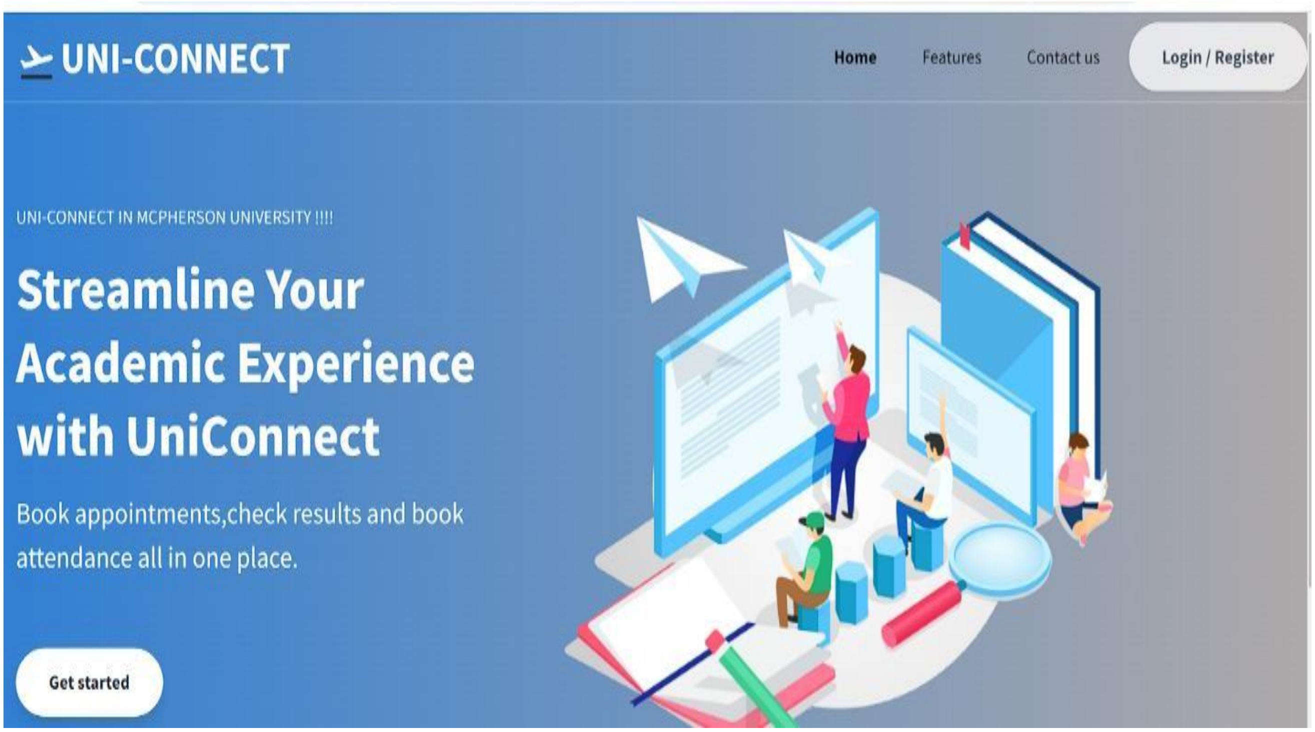
The login and signup pages of Uni-Connect in Figure 4.2 are critical components of the platform, providing secure and efficient access for users

### 4.3.3 Lecturer’s page

The Lecturers Features Page on the Uni-Connect in Figure 4.3 platform is specifically designed to cater to the needs of lecturers, providing them with a range of tools and functionalities to manage their academic responsibilities effectively.

### 4.3.4 Parent’s page

The Parent Page on the Uni-Connect platform in Figure 4.4 is designed to facilitate communication and involvement of parents in their children's academic life.

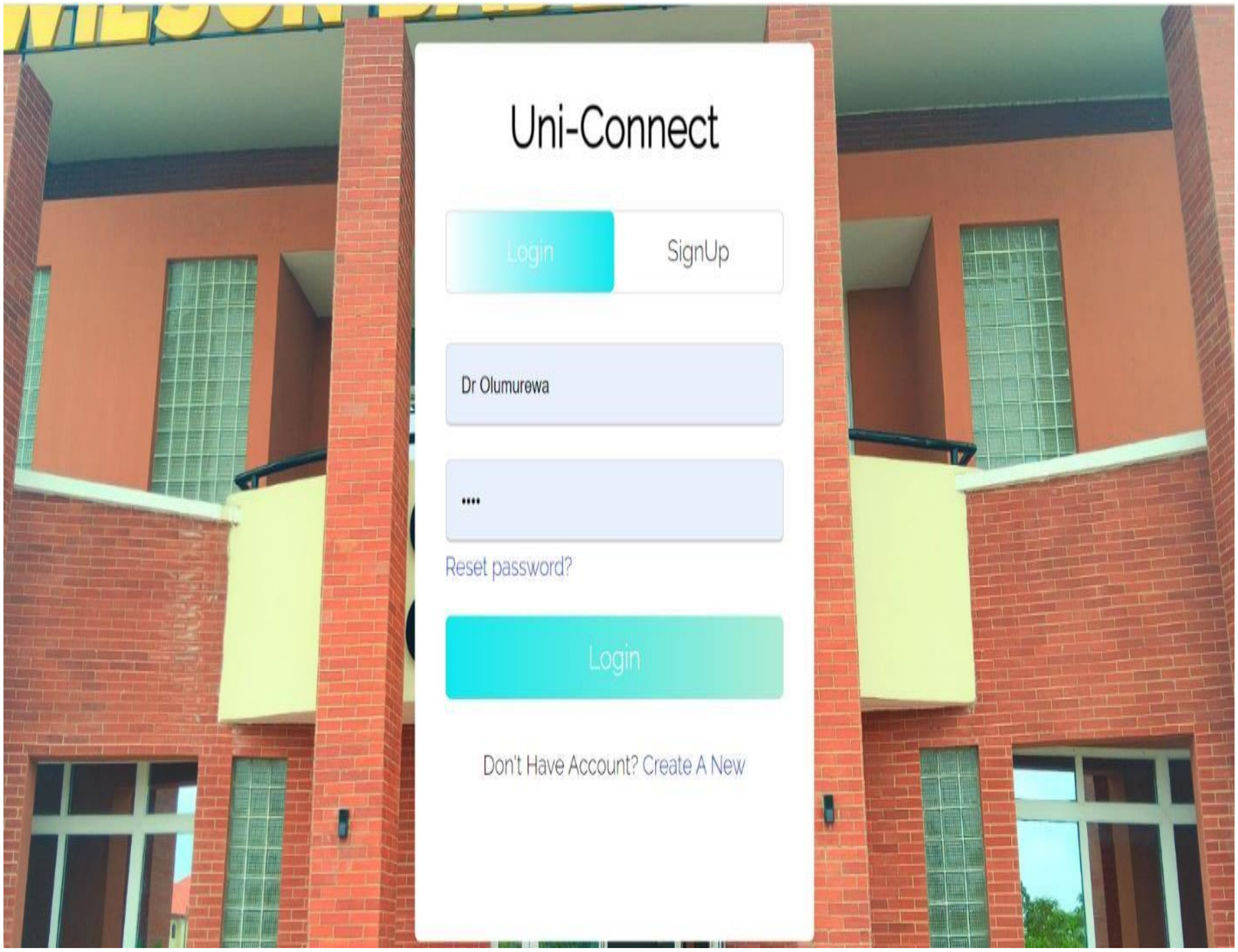


UNI-CONNECT IN MCPHERSON UNIVERSITY !!!!

Streamline Your Academic Experience with UniConnect

Book appointments,check results and book attendance all in one place.

**Figure 4.1: Landing page**



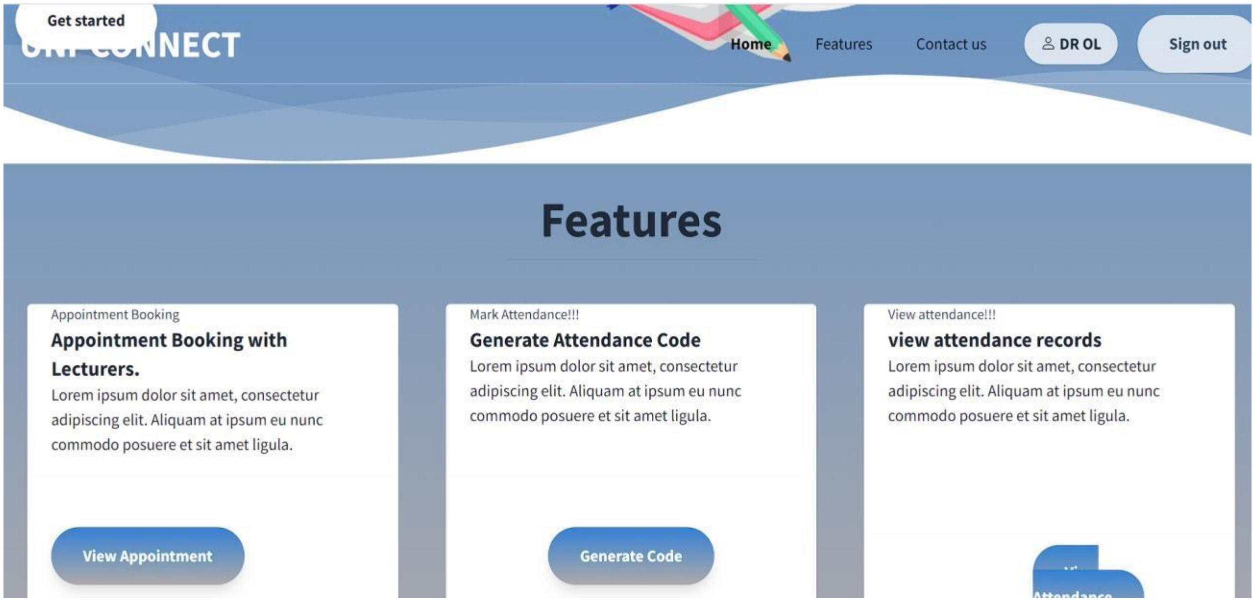
Uni-Connect

Dr Olumurewa

••••

Don't Have Account? Create A New

**Figure 4.2: Login and signup page for users**



Features

Appointment Booking

Appointment Booking with Lecturers.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam at ipsum eu nunc commodo posuere et sit amet ligula.

Mark Attendance!!!

Generate Attendance Code

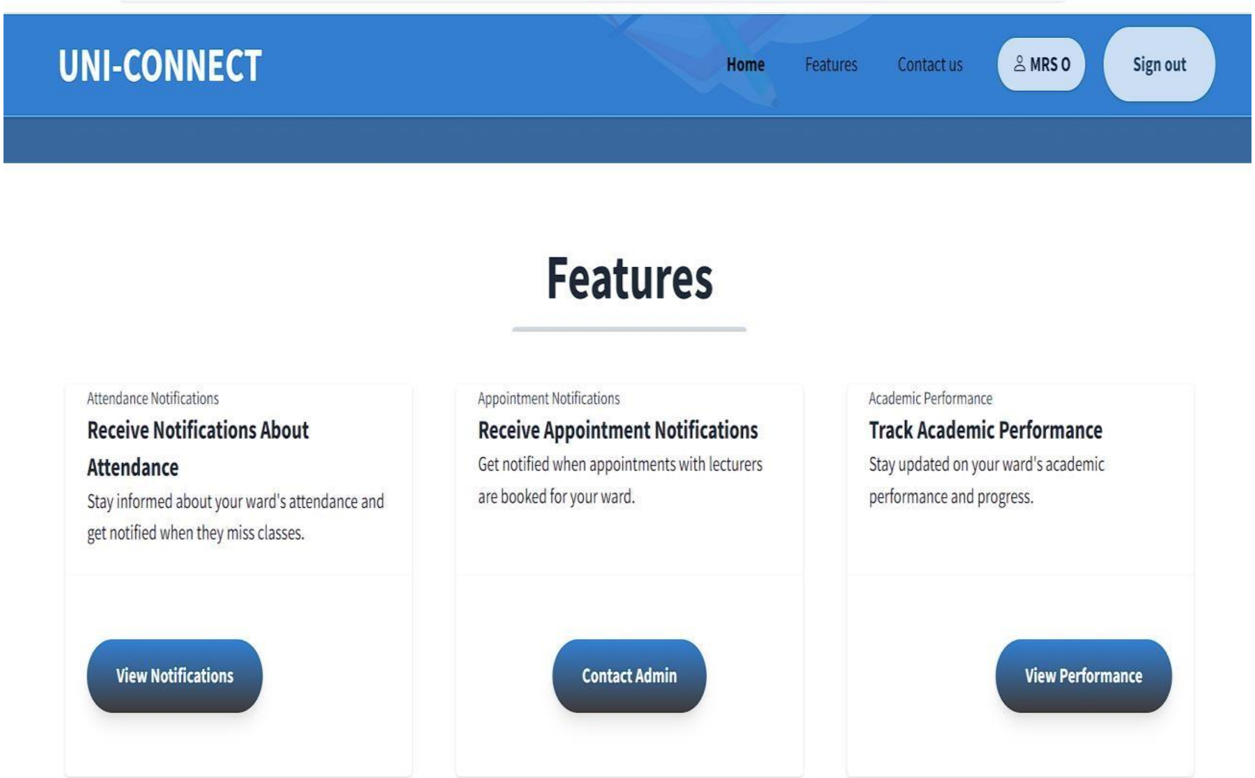
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam at ipsum eu nunc commodo posuere et sit amet ligula.

View attendance!!!

view attendance records

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam at ipsum eu nunc commodo posuere et sit amet ligula.

**Figure 4.3: Lecturer page**



Features

Attendance Notifications

Receive Notifications About Attendance

Stay informed about your ward's attendance and get notified when they miss classes.

Appointment Notifications

Receive Appointment Notifications

Get notified when appointments with lecturers are booked for your ward.

Academic Performance

Track Academic Performance

Stay updated on your ward's academic performance and progress.

**Figure 4.4: Parent page**

### 4.3.5 Student’s page

The Student Page on the Uni-Connect platform in Figure 4.5 serves as the primary interface for students, offering a comprehensive suite of tools and features designed to enhance their academic experience. This page is tailored to meet the needs of students, providing easy access to essential resources, schedules, communication tools, and academic performance tracking. It aims to facilitate a structured and efficient approach to managing their educational journey.

### 4.3.6 QR generator page

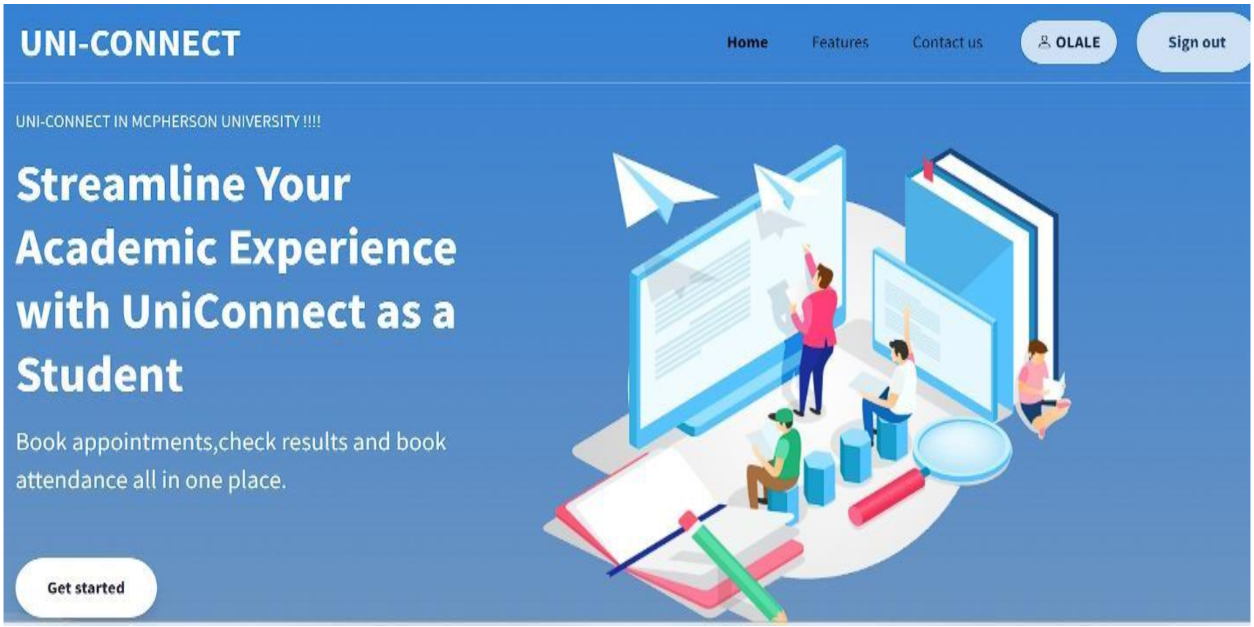
The QR Generator Page in Figure 4.6 is a pivotal feature in the Uni-Connect platform designed to streamline the attendance tracking process. This page is primarily used by lecturers to generate QR codes that students can scan to mark their attendance. This method ensures a quick, efficient, and secure way of recording attendance, minimizing manual errors and administrative workload.

### 4.3.7 QR scanner page

The QR Scanner Page in Figure 4.7 is a crucial component of the Uni-Connect platform, designed to facilitate seamless and efficient attendance tracking for students. This page allows students to scan QR codes generated by their lecturers to mark their attendance for each class session.

### 4.3.8 Appointment booking page

The Appointment Booking Page in Figure 4.8 is a vital feature of the Uni-Connect platform, designed to streamline and facilitate the scheduling of appointments between students and lecturers.

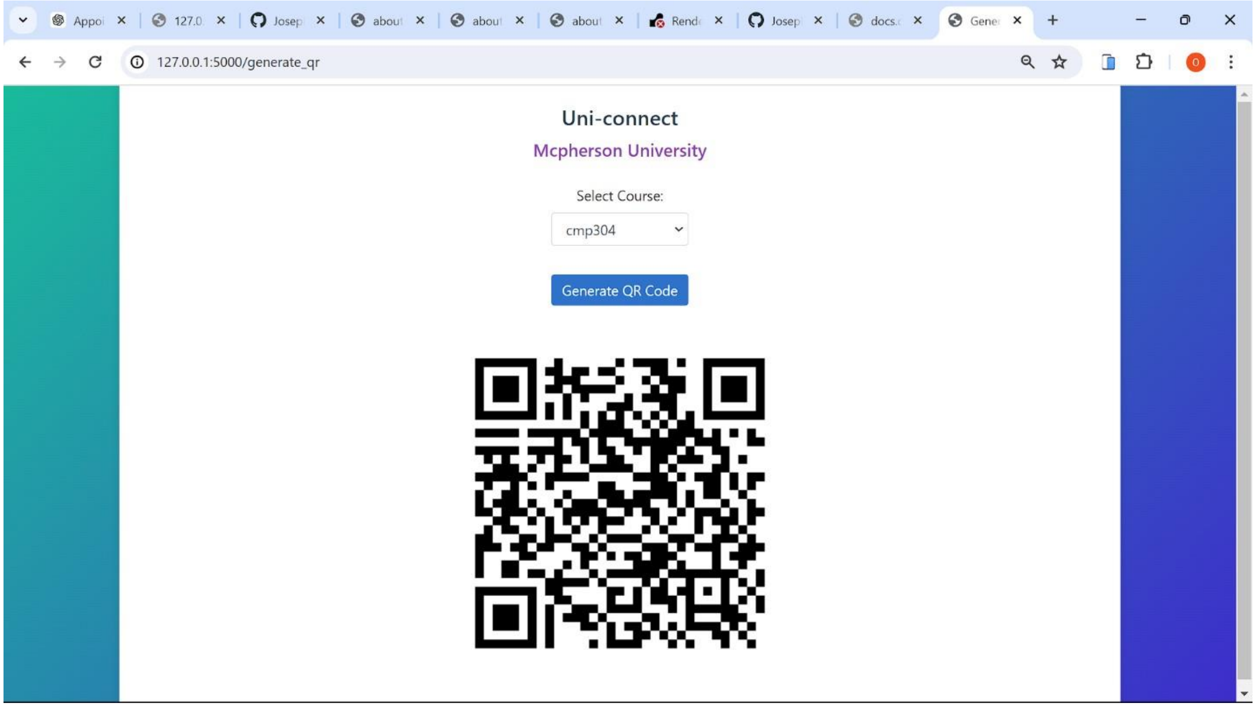


UNI-CONNECT IN MCPHERSON UNIVERSITY !!!!

Streamline Your Academic Experience with UniConnect as a Student

Book appointments,check results and book attendance all in one place.

**Figure 4.5: Student page**



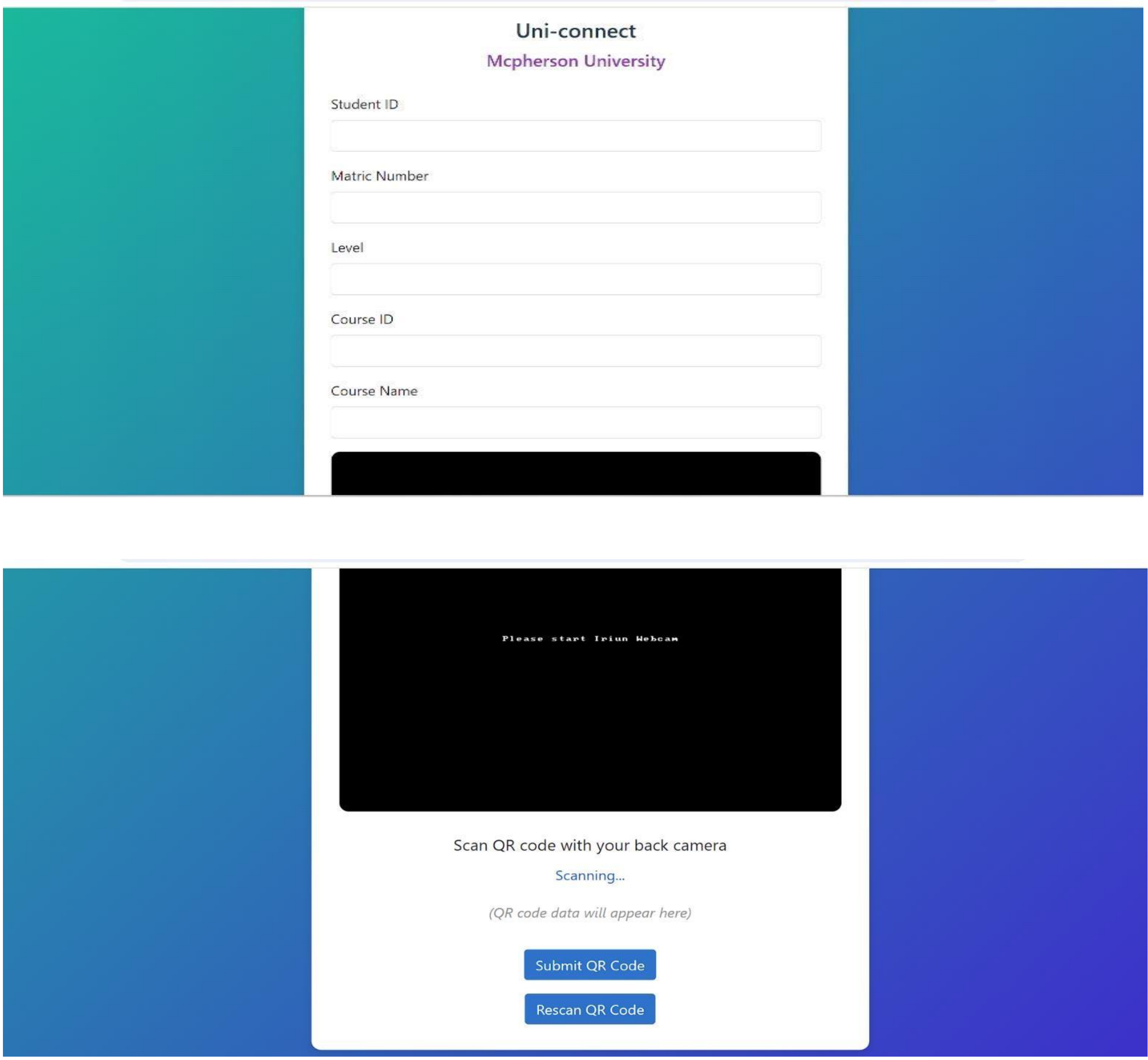
Uni-connect

Mcpherson University

Select Course:

cmp304

**Figure 4.6: QR generator page**



Uni-connect

Mcpherson University

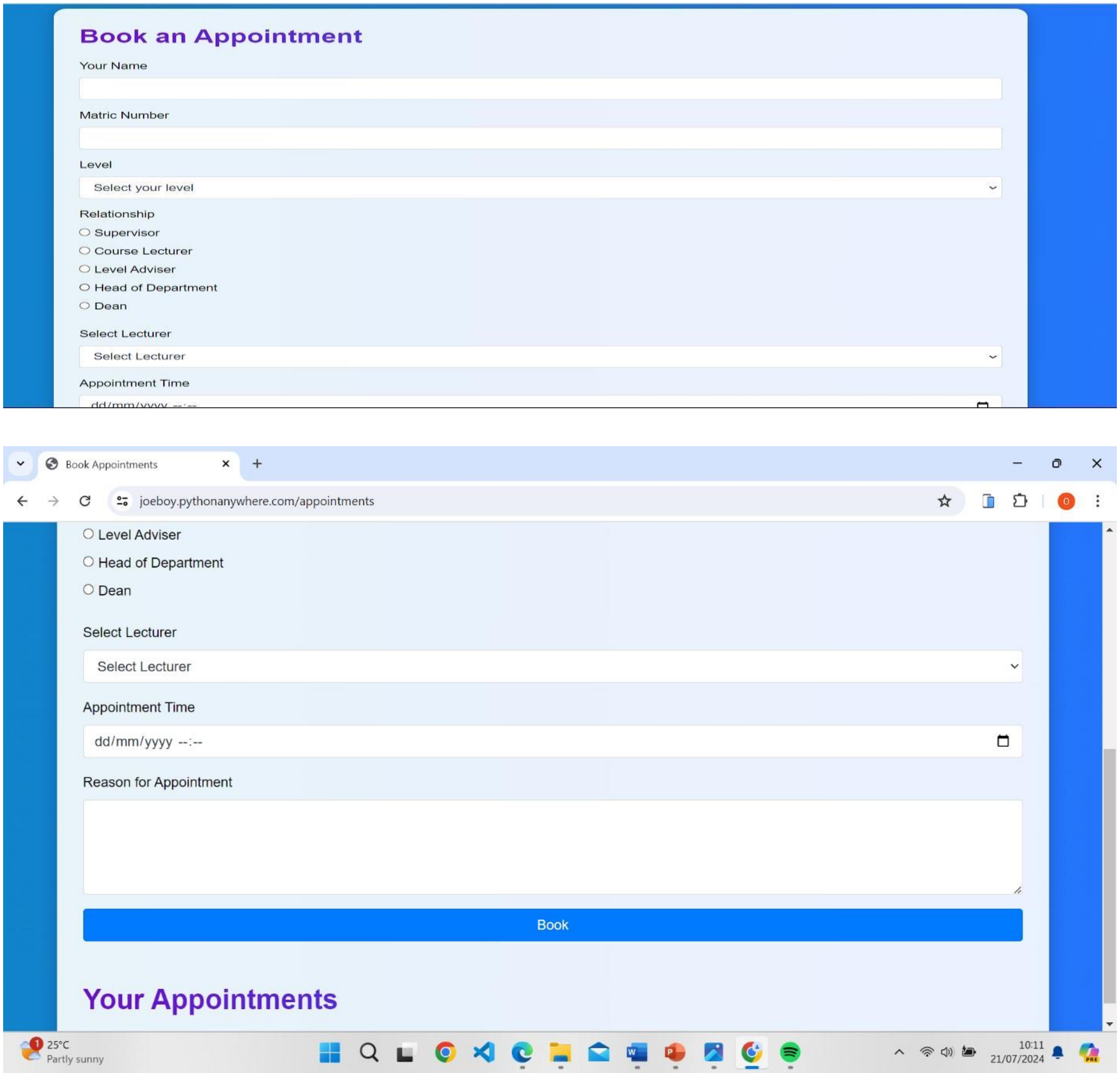
Please start Iriun Webcam

Scan QR code with your back camera

Scanning...

*(QR code data will appear here)*

**Figure 4.7: QR scanner page**



Book an Appointment

Your Name

Matric Number

Level

Select your level

Relationship

○ Supervisor

○ Course Lecturer

○ Level Adviser

○ Head of Department

○ Dean

Select Lecturer

Select Lecturer

Appointment Time

dd/mm/yyyy --:--

○ Level Adviser

○ Head of Department

○ Dean

Select Lecturer

Select Lecturer

Appointment Time

dd/mm/yyyy --:--

Reason for Appointment

Book

Your Appointments

**Figure 4.8: Appointment booking page**

### 4.3.9 Notification page

The Notification Page in Figure 4.9 is an integral component of the Uni-Connect platform, designed to keep students, lecturers, and parents informed about important updates, events, and reminders. This page ensures that all users stay connected and up-to-date with relevant information, enhancing the overall communication and efficiency of the academic support system.

## 4.4 Automated notification page

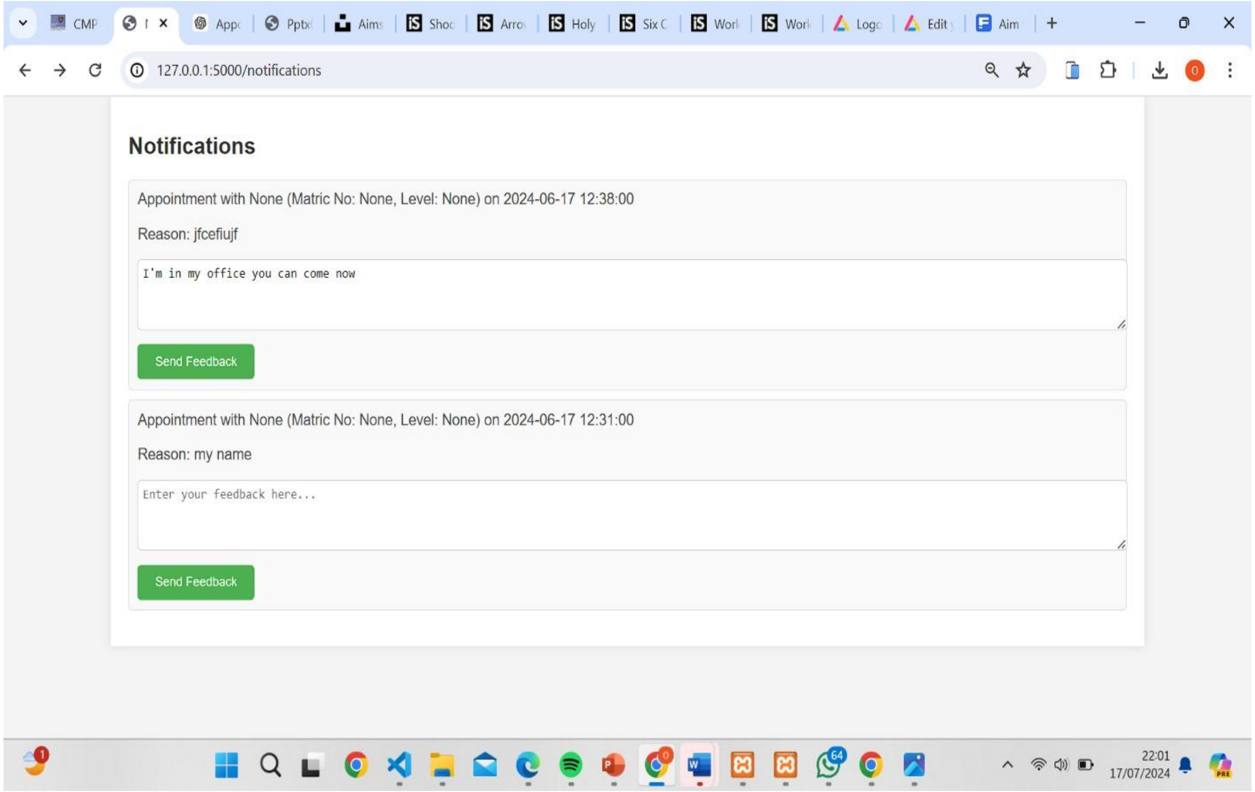
The Notification Page in Figure 4.10 is an integral component of the Uni-Connect platform, designed to keep students, lecturers, and parents informed about important updates, events, and reminders. This page ensures that all users stay connected and up-to-date with relevant information, enhancing the overall communication and efficiency of the academic support system.

### 4.4.1 List of attendance page

The Attendance Page on the Uni-Connect platform in Figure 4.11 is designed to provide users with detailed information about attendance records, enabling students, lecturers, and administrators to track and manage attendance efficiently.

### 4.4.2 Result page

The Result Page on the Uni-Connect platform in in Figure 4.12 is designed to provide users with detailed information about result records, enabling students, lecturers, and administrators to track and manage result efficiently using Microsoft excel as showm in Figure 4.13.



## Notifications

Appointment with None (Matric No: None, Level: None) on 2024-06-17 12:38:00

Reason: jfcefiujf

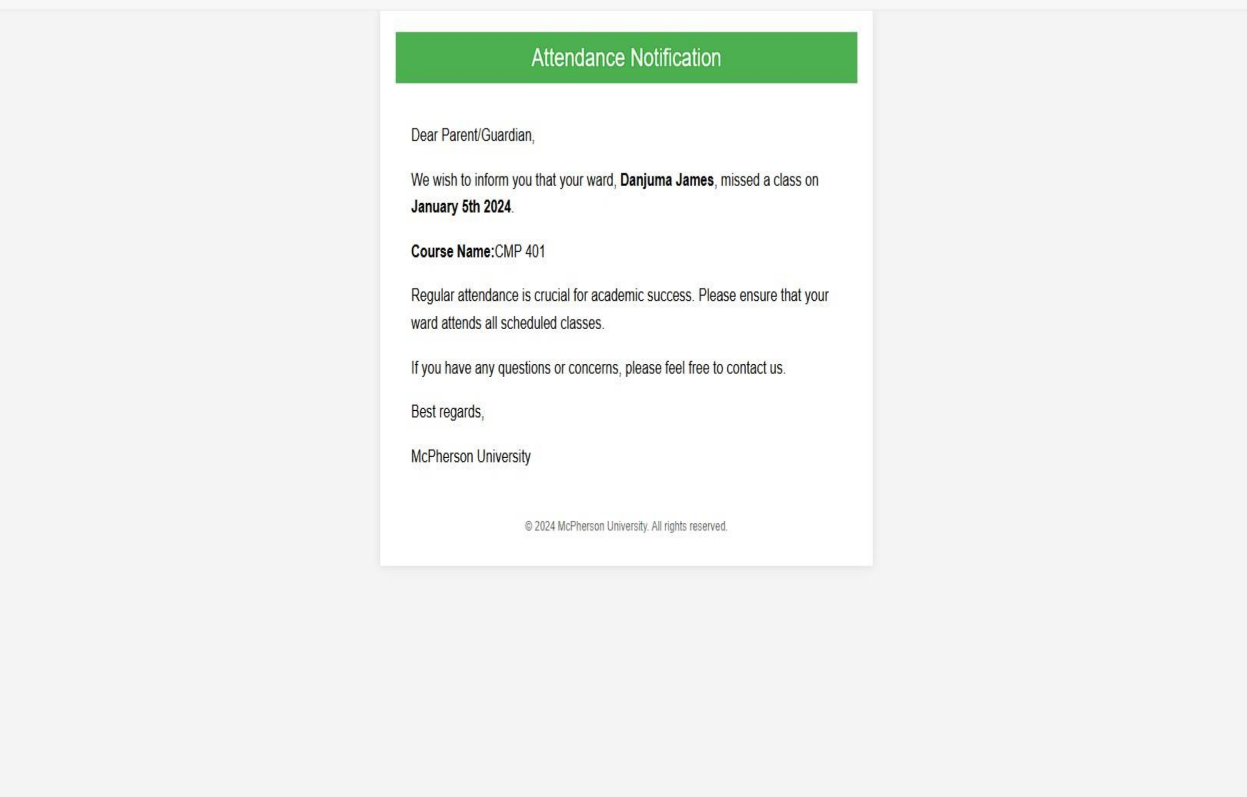
I'm in my office you can come now

Appointment with None (Matric No: None, Level: None) on 2024-06-17 12:31:00

Reason: my name

Enter your feedback here...

**Figure 4.9: Notification page**



Attendance Notification

Dear Parent/Guardian,

We wish to inform you that your ward, **Danjuma James**, missed a class on **January 5th 2024**.

**Course Name:**CMP 401

Regular attendance is crucial for academic success. Please ensure that your ward attends all scheduled classes.

If you have any questions or concerns, please feel free to contact us.

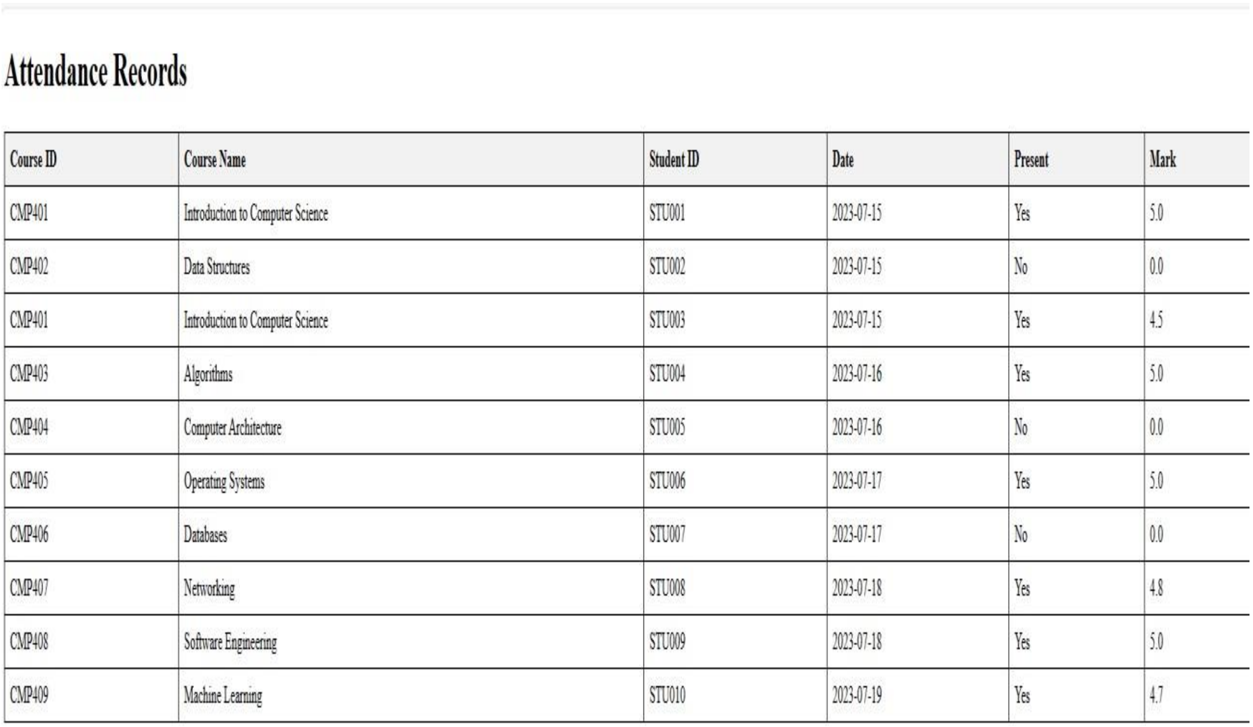
Best regards,

McPherson University

© 2024 McPherson University. All rights reserved.

\

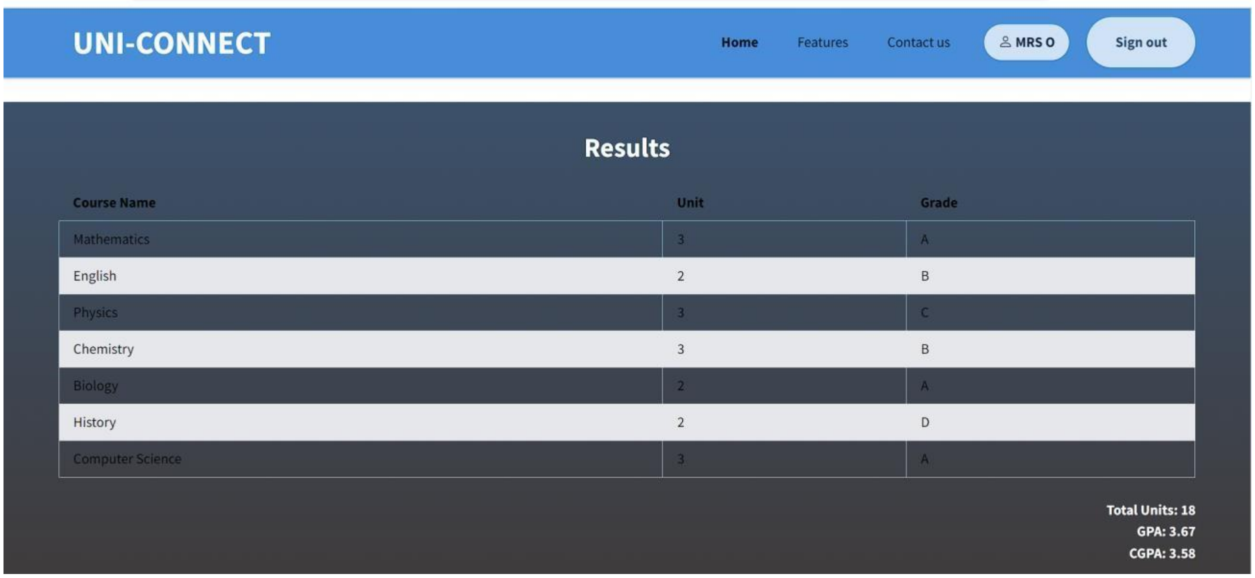
**Figure 4.10: Notification page**



## Attendance Records

| Course ID | Course Name | Student ID | Date | Present | Mark |
|---|---|---|---|---|---|
| CMP401 | Introduction to Computer Science | STU001 | 2023-07-15 | Yes | 5.0 |
| CMP402 | Data Structures | STU002 | 2023-07-15 | No | 0.0 |
| CMP401 | Introduction to Computer Science | STU003 | 2023-07-15 | Yes | 4.5 |
| CMP403 | Algorithms | STU004 | 2023-07-16 | Yes | 5.0 |
| CMP404 | Computer Architecture | STU005 | 2023-07-16 | No | 0.0 |
| CMP405 | Operating Systems | STU006 | 2023-07-17 | Yes | 5.0 |
| CMP406 | Databases | STU007 | 2023-07-17 | No | 0.0 |
| CMP407 | Networking | STU008 | 2023-07-18 | Yes | 4.8 |
| CMP408 | Software Engineering | STU009 | 2023-07-18 | Yes | 5.0 |
| CMP409 | Machine Learning | STU010 | 2023-07-19 | Yes | 4.7 |

**Figure 4.11: Attendance page**

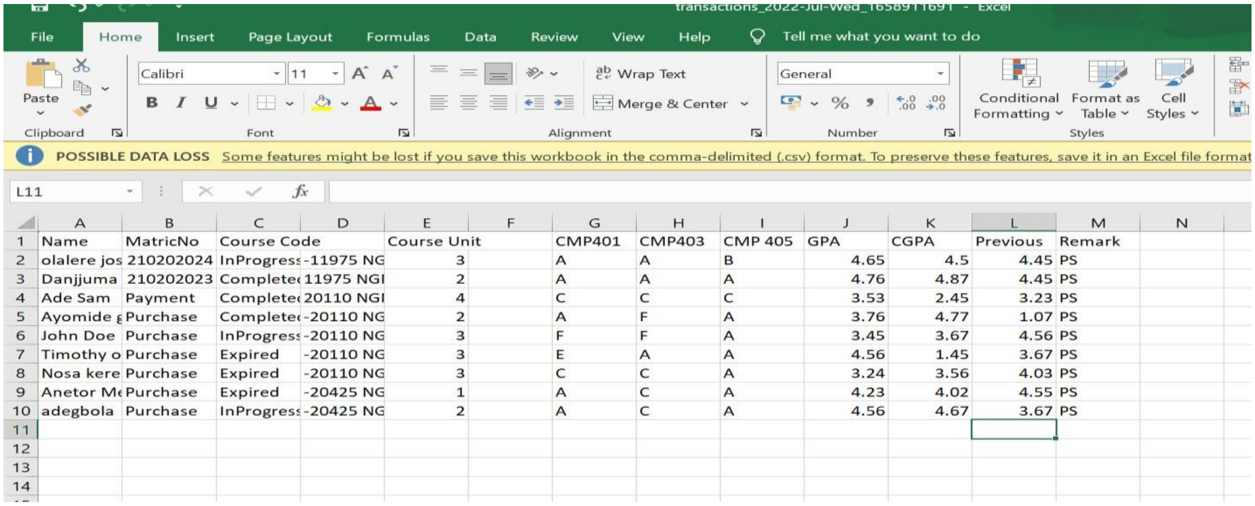


Results

| Course Name | Unit | Grade |
|---|---|---|
| Mathematics | 3 | A |
| English | 2 | B |
| Physics | 3 | C |
| Chemistry | 3 | B |
| Biology | 2 | A |
| History | 2 | D |
| Computer Science | 3 | A |

Total Units: 18
GPA: 3.67
CGPA: 3.58

**Figure 4.12: Result page**



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Name | MatricNo | Course Code | | Course Unit | | CMP401 | CMP403 | CMP 405 | GPA | CGPA | Previous | Remark | |
| 2 | olalere jos | 210202024 | InProgress | -11975 NG | 3 | | A | A | B | 4.65 | 4.5 | 4.45 | PS | |
| 3 | Danjjuma | 210202023 | Complete | 11975 NGI | 2 | | A | A | A | 4.76 | 4.87 | 4.45 | PS | |
| 4 | Ade Sam | Payment | Complete | 20110 NGI | 4 | | C | C | C | 3.53 | 2.45 | 3.23 | PS | |
| 5 | Ayomide g | Purchase | Complete | -20110 NG | 2 | | A | F | A | 3.76 | 4.77 | 1.07 | PS | |
| 6 | John Doe | Purchase | InProgress | -20110 NG | 3 | | F | F | A | 3.45 | 3.67 | 4.56 | PS | |
| 7 | Timothy o | Purchase | Expired | -20110 NG | 3 | | E | A | A | 4.56 | 1.45 | 3.67 | PS | |
| 8 | Nosa kere | Purchase | Expired | -20110 NG | 3 | | C | C | A | 3.24 | 3.56 | 4.03 | PS | |
| 9 | Anetor Me | Purchase | Expired | -20425 NG | 1 | | A | C | A | 4.23 | 4.02 | 4.55 | PS | |
| 10 | adegbola | Purchase | InProgress | -20425 NG | 2 | | A | C | A | 4.56 | 4.67 | 3.67 | PS | |
| 11 | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | |

**Figure 4.13: Excel page**

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Summary

The system offers a robust platform for managing various academic activities, such as attendance tracking, appointment scheduling, and notifications. The use of QR codes for attendance has streamlined the process, reducing the possibility of errors and ensuring accurate records. The automated communication feature enhances the relationship between students, lecturers, and parents, fostering a supportive academic environment.

Overall, the Uni-Connect system has proven to be a valuable tool in enhancing the academic experience at McPherson University. The project successfully demonstrates how technology can be harnessed to improve educational processes and outcomes.

## 5.2 Conclusion

The development and implementation of the Uni-Connect system represent a significant advancement in the way McPherson University handles academic support and communication. By leveraging modern web technologies and integrating features that address the specific needs of students, lecturers, and parents, the Uni-Connect system has achieved its primary objectives.

## 5.3 Future Work

The Uni-Connect project significantly contributes to the body of knowledge in the field of educational technology and academic support systems. This system has been designed to address specific needs within an academic environment, demonstrating how technology can enhance educational processes.

## 5.4 Recommendations

The recommendations are as follows:

1. Expanding the system’s integration capabilities with other university management systems (such as Learning Management Systems, Student Information Systems) would streamline operations and data flow.
2. : Implementing a feedback mechanism where users can submit suggestions and report issues will help in continuous improvement of the system based on real user experiences.
3. While the current design is functional, further refinement of the user interface to make it more intuitive and visually appealing could improve user engagement and satisfaction.
4. Providing comprehensive training for users and establishing a support system to assist with any issues or questions related to the system.

# REFERENCES

Doi, S., Isumi, A., & Fujiwara, T. (2020). The association between parental involvement 142 behavior and self-esteem among adolescents living in poverty: Results from the K-CHILD Study. International Journal of Environmental Research and Public Health, 17(17). https://doiorg./10.3390/ijerph17176277

Boonk, L., Gijselaers, H. J., Ritzen, H., & Brand-Gruwel, S. (2018). A review of the 138 relationship between parental involvement indicators and academic achievement. Educational Research Review, 24, 10-30. https://doi.org/10.1016/j.edurev.2018.02.001

Ogg, J., & Anthony, C. J. (2020). Process and context: Longitudinal effects of the interactions between parental involvement, parental warmth, and SES on academic achievement. Journal of School Psychology, 78, 96-114. https://doi-org.ezp.waldenulibrary.org/10.1016/j.jsp.2019.11.004 Alinsunurin, J. (2020). School learning climate in the lens of parental involvement and school leadership: Lessons for inclusiveness among public schools. Smart

Learning Environments, 7(1), 1-23. https://doi.org/10.1186/s40561-020-00139-2 Austin-Ohanenye, G. (2021). Teachers' and parents' perceptions of parental involvement in elementary students' education: A phenomenological study [Doctoral dissertation, Northcentral University]. ProQuest Dissertations Publishing, 2021.

28415329.

https://www.proquest.com/openview/aa4b855c741b0dd17c5959405e628653/1?cbl=

18750&diss=y&pq-origsite=gscholar

Bailey, T. (2017). The impact of parental involvement on student success: School and family partnership from the perspective of students.

https://digitalcommons.kennesaw.edu/teachleaddoc_etd/22/

Brotherson, S. E., Saxena, D., Lammers, S., & Bushaw, K. (2020). Opportunities, barriers, and parent preferences. https://doi.org/10.1111/1467-9604.12009

Berkowitz, R., Astor, R. A., Pineda, D., DePedro, K. T., Weiss, E. L., & Benbenishty, R. (2017). Parental Involvement and Perceptions of School Climate in California. Urban Education, 56(3), 393–423. https://doi.org/10.1177/0042085916685764

Kim, S. W. (2019). Meta-Analysis of Parental Involvement and Achievement in East Asian Countries. Education and Urban Society, 2(2), 312–337. https://doi.org/10.1177/0013124519842654

Tan, C. Y., Lyu, M., & Peng, B. (2019). Academic Benefits from Parental Involvement are Stratified by Parental Socioeconomic Status: A Meta-analysis. Parenting, 1–47. doi:10.1080/15295192.2019.169483

Motz, B. A., Mallon, M. G., & Quick, J. D. (2021). Automated Educative Nudges to Reduce Missed Assignments in College. IEEE Transactions on Learning Technologies, 14(2), 189–200.
doi:10.1109/tlt.2021.3064613

Hornby, G., & Blackwell, I. (2018). Barriers to parental involvement in education: an update. Educational Review, 70(1), 109–119. doi:10.1080/00131911.2018.1388612

Qureshi, K. N., Naveed, A., Kashif, Y., & Jeon, G. (2021). Internet of Things for education: A smart and secure system for schools monitoring and alerting. Computers & Electrical Engineering, 93, 107275. doi:10.1016/j.compeleceng.2021.107275

Fantozzi, F., Hamdi, H., Rocca, M., & Vegnuti, S. (2019). Use of Automated Control Systems and
Advanced Energy Simulations in the Design of Climate Responsive Educational Building for Mediterranean Area. Sustainability, 11(6), 1660. doi:10.3390/su11061660

## APPENDIX/APPENDICES

```
from flask import Flask, render_template, request, redirect, url_for, flash,
session,jsonify,request from flask_mysqldb import MySQL
from werkzeug.security import generate_password_hash, check_password_hash import
qrcode import base64 import io
from datetime import datetime import cv2 import numpy as np import MySQLdb.cursors
app = Flask(__name__) app.secret_key = 'your_secret_key' # Database configuration
app.config['MYSQL_HOST'] = '127.0.0.1' app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '' app.config['MYSQL_DB'] =
'academic_support_system' app.config['MYSQL_CURSORCLASS'] = 'DictCursor'


mysql = MySQL(app)


def create_tables():
    with app.app_context():
        try:
            cur = mysql.connection.cursor() cur.execute("""
                    PRIMARY KEY (id),
                    FOREIGN KEY (user_id) REFERENCES usersss(id),
                    FOREIGN KEY (appointment_id) REFERENCES appointments(id),
                    FOREIGN KEY (lecturer_id) REFERENCES usersss(id),
                    FOREIGN KEY (student_id) REFERENCES usersss(id)
                )
            """)
        cur.execute("""

          CREATE TABLE IF NOT EXISTS attendance (        id INT
        AUTO_INCREMENT PRIMARY KEY,     student_id INT NOT NULL,
        course_id INT NOT NULL,     attendance_date DATE NOT NULL,        present
        TINYINT(1) NOT NULL DEFAULT 0,         mark FLOAT NOT NULL DEFAULT
        0,
              FOREIGN KEY (student_id) REFERENCES usersss(id),

              FOREIGN KEY (course_id) REFERENCES courses(id)

          )           """)
        mysql.connection.commit()cur.close()except AttributeErroras e:print(f"Error: {e}.
Check your MySQL configuration and ensure the MySQL server is running.")        except
Exception as e: print(f"An unexpected error occurred: {e}")
@app.route('/') def landing_page():     if 'user_id' in session:
      return redirect(url_for('landing_after_login'))     return render_template('landing.html')
@app.route('/home') def home():     return render_template('login.html')


@app.route('/signup', methods=['POST']) def signup():

   email = request.form['email'] password = request.form['password']
   confirm_password = request.form['confirm_password'] role = request.form['role'] if
password == confirm_password:        hashed_password =
generate_password_hash(password)        cur = mysql.connection.cursor()
      cur.execute("INSERT INTO usersss (email, password, role) VALUES (%s, %s, %s)",
(email, hashed_password, role))        mysql.connection.commit()        cur.close()
      flash("Signup successful. You can now login.")        return redirect(url_for('home'))
else:
      flash("Passwords do not match. Try again.")        return redirect(url_for('home'))
@app.route('/login', methods=['POST']) def login():
   email = request.form['email']     password = request.form['password']     try:
        cur = mysql.connection.cursor()

        cur.execute("SELECT * FROM usersss WHERE email = %s", (email,))

        user = cur.fetchone()         cur.close()         if user and
check_password_hash(user['password'], password):
            session['user_id'] = user['id']

            session['role'] = user['role']

            session['user_abbr'] = user['email'][:5].upper()

            flash(f"You were successfully logged in as {session['user_abbr']}") if user['role'] ==
            'Student':
                return redirect(url_for('student_dashboard'))

            elif user['role'] == 'Lecturer':

                return redirect(url_for('lecturer_dashboard'))             elif user['role'] == 'Parent':
                return redirect(url_for('parent_dashboard'))         else:
            flash("Invalid login credentials. Please try again.") except MySQLdb.Error as e:
flash(f"Database connection error: {e}")     return redirect(url_for('home'))
@app.route('/logout') def logout():
    session.clear()

    flash("You were successfully logged out")     return render_template('landing.html')


@app.route('/landing2') def landing_after_login():     if 'user_id' in session:
        cur = mysql.connection.cursor()

        cur.execute("SELECT email FROM usersss WHERE id = %s", [session['user_id']])
user = cur.fetchone()         cur.close()
        user_abbr = user['email'][:5].upper() if user else '' return
        render_template('landing.html', user_abbr=user_abbr)
    else:
     return redirect(url_for('home')) @app.route('/notifications') def notifications():     if
'user_id' not in session:
     return redirect(url_for('home'))     cur = mysql.connection.cursor()     if session['role']
== 'Lecturer':        lecturer_id = session['user_id']        cur.execute("""
          SELECT notifications.*, appointments.appointment_time,
appointments.reason,                usersss.email AS student_email,                usersss.name
AS student_name,                usersss.matric_no AS student_matric_no,
usersss.level AS student_level
          FROM notifications

          JOIN appointments ON notifications.appointment_id = appointments.id

          JOIN usersss ON appointments.student_id = usersss.id

          WHERE notifications.lecturer_id = %s AND appointments.feedback IS NULL

     """, [lecturer_id])     elif session['role'] == 'Student':        student_id =
session['user_id']        cur.execute("""
          SELECT notifications.*, appointments.appointment_time,
appointments.feedback,        usersss.email AS lecturer_email
          FROM notifications

          JOIN appointments ON notifications.appointment_id = appointments.id

          JOIN usersss ON appointments.lecturer_id = usersss.id

          WHERE notifications.student_id = %s

     """, [student_id])     notifications = cur.fetchall()     cur.close()
  return render_template('notifications.html', notifications=notifications)

@app.route('/send_feedback', methods=['POST']) def send_feedback():     if 'user_id' not in
session or session['role'] != 'Lecturer':
     return redirect(url_for('home'))     feedback = request.form['feedback']
appointment_id = request.form['appointment_id']     cur = mysql.connection.cursor()
```

```
   cur.execute("UPDATE appointments SET feedback = %s WHERE id = %s AND
lecturer_id = %s",

            (feedback, appointment_id, session['user_id']))     mysql.connection.commit()
cur.close()
   flash("Feedback sent successfully.")     return redirect(url_for('notifications'))
@app.route('/appointments', methods=['GET', 'POST']) def appointments():     if 'user_id'
not in session:
      return redirect(url_for('home'))

   cur = mysql.connection.cursor() cur.execute("SELECT id, email, role FROM usersss
  WHERE role = 'Lecturer'")  lecturers = cur.fetchall()
   if request.method == 'POST':        student_id = session['user_id']        lecturer_id =
request.form['lecturer_id']        appointment_time = request.form['appointment_time']
reason = request.form['reason']
      cur.execute("SELECT id FROM usersss WHERE id = %s AND role = 'Lecturer'",
(lecturer_id,))

      lecturer = cur.fetchone()        if not lecturer:            flash("Selected lecturer does not
exist.")           return redirect(url_for('appointments'))
      cur.execute("INSERT INTO appointments (student_id, lecturer_id, appointment_time,
reason) VALUES (%s, %s, %s, %s)",

               (student_id, lecturer_id, appointment_time, reason))
mysql.connection.commit()        appointment_id = cur.lastrowid
      cur.execute("SELECT name, matric_no, level FROM usersss WHERE id = %s",
(student_id,))

      student_info = cur.fetchone()        student_name = student_info['name']
student_matric_no = student_info['matric_no']        student_level = student_info['level']
      cur.execute("INSERT INTO notifications (user_id, message, is_read, appointment_id,
lecturer_id, student_id, student_name, student_matric_no, student_level, reason) VALUES
(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)",
                (lecturer_id, reason, False, appointment_id, lecturer_id, student_id,
student_name, student_matric_no, student_level, reason))        mysql.connection.commit()
      cur.close()

      flash("Appointment requested successfully.")        return
redirect(url_for('appointments'))   cur.close()
   return render_template('appoint.html', lecturers=lecturers)

@app.route('/scan') def scan():return render_template('scanner.html')
@app.route('/scan_qr', methods=['GET', 'POST']) def scan_qr():     if 'user_id' not in
session or session['role'] != 'Student':
      return redirect(url_for('home')) if request.method == 'POST':        data =
request.form.get('data')  # Extract form data        decoded_text = data  # Get the decoded
text from form data        if decoded_text:
         # Process the decoded text here            session['qr_data'] = decoded_text
return redirect(url_for('mark_attendance_direct'))  else:
         flash('No QR code data found', 'danger')           return redirect(url_for('scan'))
return render_template('scanner.html')
@app.route('/mark_attendance_direct', methods=['POST'])

def mark_attendance_direct():     if 'user_id' not in session:
      return redirect(url_for('home') decoded_text = session.get('qr_data')     if
decoded_text:
      # Process the decoded text here  session.pop('qr_data', None)  # Clear the QR data
from session
      # Mark attendance logic here

      flash('Attendance marked successfully.', 'success' return redirect(url_for('home'))
else:
      flash('No QR code data found', 'danger')        return redirect(url_for('scan'))
@app.route('/generate_qr', methods=['GET', 'POST']) def generate_qr():     if 'user_id' not
in session or session['role'] != 'Lecturer':
        return redirect(url_for('home')) if request.method == 'POST' course_id =
request.form.get('course_id')   date = request.form.get('date'0   if not course_id or not date:
            return "Bad Request: Missing course_id or date", 400         cur =
mysql.connection.cursor()         cur.execute("SELECT course_name FROM courses
WHERE id = %s", (course_id,))
        course = cur.fetchone()         if not course:
            return "Bad Request: Invalid course_id", 400

            course_name = course['course_name']         lecturer_id = session['user_id']
qr_data = f"{course_id},{course_name},{lecturer_id},{date}"         qr = qrcode.QRCode(
version=1,
            error_correction=qrcode.constants.ERROR_CORRECT_L,            box_size=10,
border=4,
        )

        qr.add_data(qr_data)         qr.make(fit=True)
        qr_img = qr.make_image(fill='black', back_color='white')         img_buffer =
io.BytesIO()         qr_img.save(img_buffer, format='PNG')
        img_str = base64.b64encode(img_buffer.getvalue()).decode()         cur.close()
return render_template('generate_qr_code.html', qr_code=img_str)     cur =
mysql.connection.cursor()
    cur.execute("SELECT * FROM courses WHERE lecturer_id = %s", [session['user_id']])
courses = cur.fetchall()     cur.close()     return render_template('generate_qr_code.html',
courses=courses)
@app.route('/get_attendance', methods=['GET']) def get_attendance():     if 'user_id' not in
session:
        return redirect(url_for('home'))     cur = mysql.connection.cursor()     if session['role']
== 'Student':         student_id = session['user_id']
        cur.execute("SELECT * FROM attendance WHERE student_id = %s", [student_id])
elif session['role'] == 'Lecturer':         lecturer_id = session['user_id']         cur.execute("""
            SELECT attendance.*, courses.course_name
```

```
          FROM attendance

          JOIN courses ON attendance.course_id = courses.id

          WHERE courses.lecturer_id = %s

      """, [lecturer_id])

   attendance_records = cur.fetchall()     cur.close()     return
render_template('view_attendance.html', attendance_records=attendance_records)
@app.route('/add_courses') def add_courses():     if 'user_id' not in session or session['role']
!= 'Lecturer':
      return redirect(url_for('home'))     lecturer_id = session['user_id']
   courses = ['CMP 401', 'CMP 407', 'CMP 314']     cur = mysql.connection.cursor()     for
course_name in courses:
      cur.execute("INSERT INTO courses (course_name, lecturer_id) VALUES (%s, %s)",
(course_name, lecturer_id))     mysql.connection.commit()     cur.close()
   return "Courses added successfully!"

def notify_absentees():     with app.app_context():
      cur = mysql.connection.cursor()         attendance_date = datetime.now().date()
cur.execute("""
          SELECT usersss.id, usersss.email, usersss.role, courses.course_name

          FROM usersss

          JOIN attendance ON usersss.id = attendance.student_id

          JOIN courses ON attendance.course_id = courses.id

          WHERE attendance.present = 0 AND attendance.attendance_date = %s

      """, [attendance_date])         absentees = cur.fetchall()         for absentee in absentees:
          lecturer_email = get_lecturer_email(absentee['course_id'])            parent_email =
get_parent_email(absentee['id'])
```

```
          send_email(lecturer_email, "Student Absence Notification", f"Student
{absentee['email']} missed the class for {absentee['course_name']} on {attendance_date}")

          send_email(parent_email, "Student Absence Notification", f"Your child
{absentee['email']} missed the class for {absentee['course_name']} on {attendance_date}")
cur.close() def get_lecturer_email(course_id):     cur = mysql.connection.cursor()
   cur.execute("SELECT usersss.email FROM usersss JOIN courses ON usersss.id =
courses.lecturer_id WHERE courses.id = %s", [course_id])     lecturer = cur.fetchone()
cur.close()
   return lecturer['email'] if lecturer else None def get_parent_email(student_id):

   cur = mysql.connection.cursor()

   cur.execute("SELECT email FROM usersss WHERE role = 'Parent' AND id = %s",
[student_id])     parent = cur.fetchone()     cur.close()
   return parent['email'] if parent else None def send_email(to, subject, body):
print(f"Sending email to {to} with subject {subject} and body {body}")
@app.route('/student_dashboard') def student_dashboard():
   return render_template('landing2.html', user_abbr=session.get('user_abbr'))

@app.route('/lecturer_dashboard') def lecturer_dashboard():render_template('parent.html',
user_abbr=session.get('user_abbr'))
@app.route('/parent_dashboard') def parent_dashboard():
   return render_template('parent_dashboard.html', user_abbr=session.get('user_abbr')) if
__name__ == '__main__':
   create_tables()     app.run(debug=True)
```

```
<!DOCTYPE html>

<html lang="en">

<head>
```

```
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>UNI-CONNECT - Landing Page</title>
   <meta name="description" content="Simple landing page for Uni-Connect">
   <meta name="keywords" content="Uni-Connect, landing page">
   <meta name="author" content="Uni-Connect">
   <link rel="stylesheet"
href="https://unpkg.com/tailwindcss@2.2.19/dist/tailwind.min.css"/>
   <script src="https://cdnjs.cloudflare.com/ajax/libs/jsqr/1.0.0/jsQR.js"></script>
                <link
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700"
rel="stylesheet"/>


   <!-- Other head elements -->
                    <link   rel="stylesheet"
         href="https://cdnjs.cloudflare.com/ajax/libs/bootstrapicons/1.10.0/font/bootstrap-
icons.min.css">


   <style>
      .gradient {     background: linear-gradient( #3381d5 0%, #afaaaa 100%);
      }
      .deep-black {          color: #000000;
```

```
    }

    .user-abbr-container {           display: flex;           align-items: center;
       padding: 10px; /* Add padding to the container */           margin-right: 10px; /* Add
some margin to the right */
    }

  </style>

</head>

<body class="leading-normal tracking-normal text-white gradient" style="font-family:
'Source
Sans Pro', sans-serif;">

  <!--Nav-->

  <nav id="header" class="fixed w-full z-30 top-0 text-white">

    <div class="w-full container mx-auto flex flex-wrap items-center justify-between mt-0
py-2">          <div class="pl-4 flex items-center">
         <a class="toggleColour text-white no-underline hover:no-underline font-bold text-
2xl lg:text-4xl" href="/">

            UNI-CONNECT

         </a>

       </div>

       <div class="block lg:hidden pr-4">

         <button id="nav-toggle" class="flex items-center p-1 text-pink-800 hover:text-
gray-900">
```

```
            <svg class="fill-current h-6 w-6" viewBox="0 0 20
20"><title>Menu</title><path d="M0 3h20v2H0V3zm0 6h20v2H0V9zm0 6h20v2H0v-
2z"/></svg>

         </button>

</body>

</html>
```