

**DESIGN AND IMPLEMENTATION OF A HOSTEL  
MANAGEMENT SYSTEM**

**BY**

**EBITIGHA OLUWADAMILOLA FRANCIS                      20/2750**

**NWABUEZE UCHECHUKWU FERDINAND                      20/2901**

**IMADE UYIOSA DANIEL    21/2763**

**A PROJECT WORK SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF  
SCIENCE B.Sc. (HONS)  
IN SOFTWARE ENGINEERING**

**SUBMITTED TO:**

**THE DEPARTMENT OF SOFTWARE ENGINEERING  
SCHOOL OF COMPUTING AND ENGINEERING SCIENCES  
BABCOCK UNIVERSITY, ILISHAN REMO**

**OGUN STATE, NIGERIA**

**MARCH, 2024**

## DECLARATION

We declare that the project work, Design and Implementation of a Hostel Management System was carried out by the following People:

EBITIGHA OLUWADAMILOLA FRANCIS

20/2750

\_\_\_\_\_

Date

NWABUEZE UCHECHUKWU FERDINAND

20/2901

\_\_\_\_\_

Date

IMADE UYIOSA DANIEL

21/2763

\_\_\_\_\_

Date

21/2763

## CERTIFICATION

We hereby certify that the project was carried out by the under listed student under the supervision of the department of Software Engineering, School of Computing and Engineering Sciences, Babcock University, Ilishan Remo, Ogun State.

Ebitigha Oluwadamilola Francis	20/2750
--------------------------------	---------

Nwabueze Uchechukwu Ferdinand	20/2901
-------------------------------	---------

Imade Uyiosa Daniel	21/2763
---------------------	---------

Dr. Adekola OluBukola	_____
-----------------------	-------

(Supervisor)

Date

_____	_____
Name of External Supervisor	Date

Dr. Adekola OluBukola	_____
-----------------------	-------

(Head, Department of Software Engineering)

Date

## **DEDICATION**

First and foremost, we dedicate this project to the Almighty, the source of all wisdom and guidance, God. Gratitude fills our hearts for the blessings and strength bestowed upon us throughout this academic journey. With deep appreciation, we thank God for His unwavering support, providence, and the opportunities that paved the way for our collaborative effort. May this project stand as a testament to the divine grace that guided us through challenges and illuminated our path to success. Without him none of this could have been possible and we are grateful for guiding all of us through.

We also dedicate this project to all those who have been our guiding lights throughout this academic journey, especially to our family members whose unwavering support and encouragement have been our greatest source of strength. To our professors and mentors, thank you for imparting your knowledge and inspiring us to push the boundaries of my understanding. This achievement is a testament to the collective efforts of everyone who believed in us. May this project stand as a tribute to the collaborative spirit that fuels growth and learning.

## **ACKNOWLEDGEMENTS**

We extend our heartfelt gratitude to all those who played a pivotal role in the completion of this project, contributing their time, expertise, and support.

First and foremost, we express our deepest appreciation to our families for their unwavering encouragement and understanding during the demanding phases of this academic pursuit. Their constant support provided the foundation upon which we built our achievements.

Our sincere thanks go to our academic advisors and mentors for their invaluable guidance and wisdom. Their expertise and constructive feedback were instrumental in shaping the direction and quality of our work. We are truly grateful for the time and effort they invested in nurturing our intellectual growth.

To our fellow classmates and friends who shared their insights and experiences, thank you for fostering a collaborative and enriching learning environment. Your contributions have been invaluable in shaping the diverse perspectives reflected in this project.

Lastly, we acknowledge each other, Ebitigha Oluwadamilola Francis, Nwabueze Uchechukwu Ferdinand and Imade Uyiosa Daniel, for our collective dedication, teamwork, and perseverance. This project bears the imprint of our collaborative spirit, and we are proud of the achievements we have accomplished together.

May this project and its acknowledgments stand as a tribute to the collaborative efforts of a supportive community that made this academic journey both fulfilling and rewarding.

## **ABSTRACT**

The increasing number of educational institutions worldwide has necessitated efficient management of assets, particularly hostel facilities. The prevailing manual Hostel Management System often involves extensive paperwork and calculations, leading to inefficiencies and potential data loss. This research addresses the limitations of conventional techniques and emphasizes the importance of adaptability, maintainability, and reliability in developing an effective Hostel Management System (HMS).

The challenges posed by manual hostel management techniques have spurred research efforts to develop automated systems. This study aims to design and implement an automated HMS to streamline processes, enhance efficiency, and overcome the drawbacks of traditional methods. The proposed system offers a user-friendly, graphical user interface, improved reliability, efficiency, and security with access control mechanisms. A hostel Management system should be implemented as it will reduce the time and resources spent on managing the Hostel and allow administrators to keep track of everything surrounding the hostel.

**Keywords:** Hostel, Management, Web System, Room allocation

## **TABLE OF CONTENTS**

### Contents

<b>DESIGN AND IMPLEMENTATION OF A HOSTEL MANAGEMENT SYSTEM</b>	<b>i</b>
<b>DECLARATION.....</b>	<b>i</b>
<b>CERTIFICATION.....</b>	<b>ii</b>
<b>DEDICATION.....</b>	<b>iii</b>
<b>ACKNOWLEDGEMENTS .....</b>	<b>iv</b>
<b>ABSTRACT.....</b>	<b>v</b>
<b>TABLE OF CONTENTS .....</b>	<b>vi</b>
<b>LIST OF FIGURES .....</b>	<b>ix</b>
<b>CHAPTER ONE .....</b>	<b>1</b>
1.0. INTRODUCTION.....	1
1.1 BACKGROUND OF STUDY .....	1
1.2. STATEMENT OF THE PROBLEM .....	2
1.3 AIM AND OBJECTIVES OF THE STUDY .....	3
1.4 RESEARCH METHODOLOGY .....	4
1.5 SCOPE OF THE STUDY .....	5
1.6 SIGNIFICANCE OF THE STUDY .....	6
1.7 ETHICAL CONSIDERATIONS .....	6
1.8. OPERATIONAL DEFINITION OF TERMS .....	7
1.9. ARRANGEMENT OF SUBSEQUENT CHAPTERS .....	8

<b>CHAPTER TWO .....</b>	<b>10</b>
2.0. INTRODUCTION.....	10
2.1.1 HISTORY OF A HOSTEL.....	11
2.1.2. HOSTEL MANAGEMENT SYSTEM .....	12
2.2. HISTORICAL BACKGROUND OF THE RESEARCH .....	12
2.3. OVERVIEW OF EXISTING SYSTEM .....	13
2.4. REVIEW OF RELATED WORK.....	15
2.4.1. Orah Student Information System .....	15
2.4.2 Zuan Hostel Management System .....	16
2.4.3 Summary of Reviewed Literatures .....	17
2.5. STRENGTH AND WEAKNESS OF EXISTING SYSTEM .....	18
2.6. GENERAL COMMENTS.....	21
<b>CHAPTER THREE .....</b>	<b>22</b>
<b>SYSTEM ANALYSIS AND DESIGN .....</b>	<b>22</b>
3.0. INTRODUCTION.....	22
3.1. RESEARCH DESIGN .....	22
3.1.1. ITERATIVE PROCESS MODEL.....	22
3.2. SYSTEM REQUIREMENTS .....	24
3.2.1. USER REQUIREMENTS .....	24
3.2.2. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS .....	25
3.2.2.1. FUNCTIONAL REQUIREMENTS .....	25



3.2.2.2. NON – FUNCTIONAL REQUIREMENTS .....	26
<b>3.3. SYSTEM ARCHITECTURE .....</b>	<b>27</b>
3.3.1. USE CASE DIAGRAM.....	27
3.3.2. SEQUENCE DIAGRAM.....	29
3.3.3. ACTIVITY DIAGRAM.....	32
3.3.4. ENTITY RELATIONSHIP DIAGRAM.....	34
<b>3.4. DEVELOPMENT TOOLS .....</b>	<b>36</b>
<b>CHAPTER FOUR.....</b>	<b>38</b>
4.0. INTRODUCTION.....	38
4.1. SYSTEM IMPLEMENTATION .....	38
4.1.1. MODULES/ SERVICES .....	39
4.1.2. API INTEGRATION OF THE SYSTEM .....	41
4.2. SYSTEM TESTING .....	47
4.2.1 UNIT TESTING .....	47
4.2.2. INTEGRATION TESTING .....	48
<b>CHAPTER 5.....</b>	<b>49</b>
5.1. SUMMARY .....	49
5.2. CONCLUSION .....	50
5.3. RECOMMENDATION .....	50
5.4. LIMITATIONS TO STUDY .....	51
5.5. REFERENCES.....	52

## LIST OF FIGURES

Fig 3. 1. Use case Diagram for Hostel Management System .....	28
Fig 3. 2 Sequence Diagram for Hostel Management System .....	30
Fig 3. 3Sequence Diagram showing Hall admin deduct points.....	31
Fig 3. 4 Sequence Diagram showing Student making Complaint .....	32
Fig 3. 5Activity Diagram for a student Logging in in Hostel Management System .....	33
Fig 3. 6Entity Relationship Diagram for Hostel Management System .....	35
Fig 4. 1Swagger documentation showing the registration API for a student	41
Fig 4. 2Swagger documentation showing sample of successful response to registration	41
Fig 4. 3Swagger documentation for Logging in the system .....	42
Fig 4. 4Swagger documentation for successful response to a Log in Attempt.....	42
Fig 4. 5Swagger documentation showing endpoint to change password .....	43
Fig 4. 6Swagger documentation for Confirming Email Addresses .....	43
Fig 4. 7Swagger Documentation showing HallService API endpoints .....	44
Fig 4. 8Swagger documentation showing Creation of Halls .....	44
Fig 4. 9Swagger documentation that shows endpoint for Assigning rooms to Students..	45
Fig 4. 10Swagger documentation on the endpoint to retrieve all Endpoints in a Hall .....	45
Fig 4. 11. Swagger Documentation showing Endpoint to create a Complaint .....	46
Fig 4. 12. Swagger Documentation Showing Sample Successful Response for a Complaint.....	46

# **CHAPTER ONE**

## **1.0. INTRODUCTION**

### **1.1 BACKGROUND OF STUDY**

There has been an astronomical increase in the number of educational institutions established, especially in the last four decades, all over the world. This development has brought education to the doorstep of people. Consequently, it has increased knowledge and helped produce a population of enlightened citizens who can easily abide by the rules of civilized society and contribute meaningfully to the process of democratic governance.

Most of the newly established educational institutions, however, are using the old conventional techniques for managing their assets, especially hostel facilities. The ongoing manual Hostel Management System demands a lot of paper-work and calculation. The information stored on paper can be lost, stolen or destroyed, under any circumstances. These old techniques take 80% of the time and effort used in managing Hostel Records. (Jha et al., 2022) These old techniques, with their inherent limitations, have impacted negatively on the overall organizational efficiency of these educational systems.

The challenges posed by conventional techniques in hostel management have inspired research efforts to develop automated systems that address these issues. This research emphasizes the significance of adaptability, maintainability, portability, reliability, testability, reusability, and understandability in software. These characteristics are

crucial in the context of developing a Hostel Management System (HMS) to ensure its effectiveness and sustainability. (Jha et al., 2022) .

Management of Student records is an important and draining tasks for Hostel Officers and Administrators. In this research work, the aim is to provide and develop an automated hostel management system that will help simplify the uphill process, overcome the drawbacks of traditional methods of hostel management, and overall improve efficiency.

The developed system overcomes the drawbacks of traditional methods of hostel management; it is more user-friendly, graphical-user-interface-oriented, reliable, efficient, and secured with access control mechanisms.

The research also touches upon the importance of reusability in software development. In the context of an HMS, reusability implies the ability to use existing design and knowledge to build a system that is reliable, quick to market, and maintainable. This aligns with the need for an HMS to be adaptable and flexible to different educational institutions' requirements.

## **1.2. STATEMENT OF THE PROBLEM**

The fundamental theories of economics assert that the success of an organization is a function of how well it harnesses and optimizes the available resources towards achieving its organizational objectives. Therefore, if the manual process of managing hostel activities isn't properly managed, unnecessary time and effort will be put into management, affecting productivity and efficiency.

A major concern especially in today's world is how to efficiently use and manage resources. This is not absent in the educational sector as there is need to manage the time and resources used in the management of student and room records in halls. No matter

the volume of funds that might have gone into an investment, if there is no effective coordination of efforts / resources, the whole investment will end up as a colossal waste and the vision behind the efforts will not be achieved. Thus management is a major factor that determines the index of success achievable by an organization in its quest towards realizing its corporate objectives.(Choudhury et al., 2017)

The challenges of poor management of hostel activities, starting from registration to assigning rooms to students, as well as monitoring the activities of students, have been a concern for many years now. In addition the use of old traditional methods like files and papers makes it really difficult to find and efficiently store records as the size of the school increases. In this research work, we have identified problems in hostel management which include inefficient storage and reservation process, tedious administrative tasks and limited communication. As a result, a hostel management system is proposed to be built to overcome these manual and difficult processes. This study, therefore, is to develop a fully functional system to efficiently manage all records pertaining to Student and rooms in Halls.

### **1.3 AIM AND OBJECTIVES OF THE STUDY**

The aim of this project is to develop and design a Hostel Management System for the management of hostel activities, thereby increasing the rate at which students' needs are met and the structure across the school is well coordinated.

The specific objectives are to:

1. Designing of a user friendly interface for hostel administrators to efficiently manage room reservations and resident information.
2. Implementing a scalable and adaptable system architecture to accommodate future growth and changes in hostel management requirements.

3. Evaluate the performance of the system's performance and effectiveness through testing.

## **1.4 RESEARCH METHODOLOGY**

The research methodology, system, or research work refers to the methods to be used in successfully carrying out the project. In this study, relevant documents obtained from existing literature were qualitatively analyzed for convergence, and relevant details were extracted using inductive approach. There are different methodologies that illustrate and describe the lifecycle of the project, from start to successful completion. The selection of an appropriate Software Development Lifecycle Model (SDLC) is pivotal in ensuring effective project management and successful system implementation. For the Hostel Management System development, the chosen SDLC model is Feature-Driven Development (FDD).

Feature Driven Development (FDD) provides a structured approach to managing complex projects by emphasizing the identification, design, and implementation of features as the fundamental building blocks of the development process. FDD is characterized by its feature-centric approach, iterative and incremental development, domain object modeling, and strong client involvement.

- I. Study of the previous literature on the project: Previous literature on the development of a Management system has been studied to fully understand the concept and functionality of the project; and understanding the system as a whole.
- II. Designing the system: To be able to design the system properly and effectively a thorough requirement analysis gotten from stakeholders including students and hostel administrators.

- III. Implementing the design of the system: The implementation of the system will be done with the use of a strongly typed language C# with the use of ASP.NET core which also helps to ensure a robust and secure system.
- IV. Testing the system: To achieve testing the system effectively different manual tests will be employed such as unit tests, integration testing and system tests to test the system as a whole.
- V. Documenting the project: To effectively document the project the functional and non-functional requirements will be declared explicitly and a swagger documentation will be made for the API documentation.

## **1.5 SCOPE OF THE STUDY**

The scope of this study revolves around enhancing Hostel Management Systems by implementing advanced functionalities, fostering adaptability, and improving user experience.

It aims to provide a comprehensive and efficient solution that caters to the diverse needs of hostel administrators, staff, and residents while optimizing operational processes within residential facilities.

This involves the development of a system that caters features for:

- i. Room allocation
- ii. Financial Management
- iii. Approve and View Student's Registrations
- iv. Room maintenance
- v. Complaints

## **1.6 SIGNIFICANCE OF THE STUDY**

In the world of educational institutions and lodging administration, the creation of a Hostel administration System (HMS) is crucial. Numerous advantages and solutions to problems with conventional hostel administration techniques can be obtained by putting in place an effective HMS.

Numerous hostel administration responsibilities, including room assignment, student registration, fee administration, and record-keeping, are streamlined and automated by an HMS. This minimizes the possibility of errors and reduces manual labour, increasing operational efficiency.

Accurate and secure data management is guaranteed by the system. Through the centralization of student data, housing assignments, and financial transaction information, an HMS lowers the possibility of inconsistent data. With the old traditional methods there is always tendencies of inconsistent data especially as the size of students grows. The success of this study would produce a functioning software that can be used to maintain and manage room and student records.

## **1.7 ETHICAL CONSIDERATIONS**

This study took into consideration ethical issues regarding the impact of the result on the environment and benefactors. Being a software-oriented research, this study has little or no impact on damaging the environment physically. The main data used were open source software, which is available for use without any violation of copyright laws. Other data such as students' projects were used within the confines of the laws of the Federal Republic of Nigeria.



## **1.8. OPERATIONAL DEFINITION OF TERMS**

- I. **Hostel:** A hostel is a budget-oriented, shared-room ("dormitory") accommodation that accepts individual travelers (typically backpackers) or groups for short-term stays and that provides common areas and communal facilities. To be considered a hostel, the property must provide short-term, shared (dormitory-style) accommodation for individual travelers, though many hostels also provide private rooms. The word "dormitory" refers to a room where travelers independently book individual beds in a shared room as opposed to booking entire rooms like in a hotel or guesthouse.
- II. **The Hostel Management System (HMS)** is a system for managing the various activities in the hostel. It is used for managing the hostel information. It manages the student information, room information, room allocation details, fee details, mess bill details, and employee details of the hostel. It is also used to generate reports of student details and fee details for the student. It keeps track of the number of students in the room and the availability of the room. It helps organize the manual work from which it is very difficult to find the records of students.
- III. **Resident:** A resident refers to an individual who resides in a particular place such as a hostel or dormitory. In this current context the residents are the students who occupy a room in the hostel either on a short term or long term basis.
- IV. **Database Schema:** A database schema is a logical blueprint that defines the structure and organization and relationships of data within a database. This helps to fully organize the hierarchical structure and organization of the Hostel.

- V. Software Development Life Cycle: The Software Development Life Cycle (SDLC) is a fundamental framework that governs the process of software development, encompassing planning, design, implementation, testing, deployment, and maintenance stages. [3]
- VI. Software Engineering: Software Engineering is the application of systematic, disciplined, quantifiable approaches to the development, operation and maintenance of software. It is a systematic and disciplined activity that addresses software quality issues. The difference between professional and end user is the attention given to quality concerns.

## **1.9. ARRANGEMENT OF SUBSEQUENT CHAPTERS**

The project is divided into five chapters. The subsequent chapters of this project are as follows:

Chapter One (Introduction): This covers the general introduction of the study.

Chapter Two (Literature Review): In this chapter, event management systems designed by various software engineers and programmers in the past will be reviewed, and their opinions on this project will be properly documented, along with their various design and implementation methodologies and techniques. In a bid to strike a comparison between the past and the current project timeline,

Chapter Three (System Analysis and Design): In this chapter, the various existing timelines of the project will be analyzed with the aim of clearly defining the project objectives and creating a niche for the project in the programming world.

Chapter Four (Implementation and Documentation): This chapter would include the complete implementation of the project work and detailed documentation to guide the

prospective clients of the project, along with a user guide to guide various users on how to work with the system.

Chapter Five (Summary, Conclusion, and Recommendation): In this chapter, being the final chapter, the project would be reviewed in a bid to represent it in a more concise manner. Future milestones for the project would also be outlined, as would recommendations on all aspects of the project.

## **CHAPTER TWO**

### **LITERATURE REVIEW**

#### **2.0. INTRODUCTION**

This chapter briefly discusses the review of the technique used for the existing system, and also techniques used in the development of the new system, here the projects that has been carried out concerning the hostel management system would be discussed, and also researches that had been done on the project work.

Hostel Management Systems play a crucial role in efficiently managing the operations of hostels and student accommodations. The management of hostels has witnessed a significant transformation in recent years, owing to the advent of advanced technology and the growing demand for streamlined and efficient operations [4]. Hostel management systems have emerged as indispensable tools that not only facilitate the day-to-day administrative tasks but also enhance the overall guest experience [5]. This literature review aims to provide a comprehensive overview of the key research and developments in the field of Hostel Management Systems, highlighting the historical context, current trends, and future prospects.

The need for effective hostel management systems arises from the diverse and dynamic nature of the accommodation industry. In this project we will define our scope to focus on educational institutions to provide solutions to meet their specific requirements. These systems have evolved from basic record-keeping mechanisms to sophisticated software applications and, more recently, to cloud-based platforms and mobile applications that offer seamless accessibility and automation.

The evolution of hostel management systems has been marked by a series of technological advancements, driven by the quest for efficiency, cost-effectiveness, and

improved guest services (Bhardwaj et al., 2022). From early computer-based systems in the 1990s to the integration of Internet of Things (IoT) technology and data analytics in the 21st century, these systems have continually adapted to the ever-changing demands of the industry. [7]. Moreover, research in this domain has extended beyond the core functionalities of reservations, room allocation, and billing, delving into areas such as access control, mobile applications, and smart hostel solutions.

This literature review will draw upon a wide range of sources, including academic studies, industry reports, and case studies, to explore the historical background of hostel management systems, the various features and components that constitute these systems, and the challenges and opportunities they present. By examining the literature, this study aims to provide a comprehensive understanding of how hostel management systems have evolved, how they are currently being used, and what potential developments lie ahead in this dynamic field.

### **2.1.1 HISTORY OF A HOSTEL**

The inaugural hostel was documented in 1909 in Germany. It was established by Richard Schirmann, a teacher who recognized the necessity for an economical accommodation option for his students during brief holiday periods. [8], [9].

A hostel is an inexpensive lodging facility for usually young travelers that typically has dormitory-style sleeping arrangements and sometimes offers meals and planned activities. Hostels are popular among backpackers, students, and budget-conscious travelers who are looking for affordable places to stay while traveling.

In terms of educational context, it refers to a housing facility provided by educational institutions, such as schools, colleges, and universities, where students can live while pursuing their studies. Hostels or dormitories are designed to accommodate students who

come from outside the local area, including those from different cities or countries.

These facilities serve as a home away from home for students during their academic pursuits.[10]

The hostel serves as more than just a residence for students seeking education away from home; it's a dynamic hub of learning and growth. Beyond its role as a living space, the hostel functions as a practical laboratory for human development. Here, education transcends the classroom, as students glean knowledge not only from instructors but also from their peers. Engaging in analytical discussions with fellow residents enriches their understanding of the curriculum and often contributes significantly to character development. Hostel life isn't solely about mastering theoretical concepts; it's a platform for honing personal skills and fostering independence.[11]

### **2.1.2. HOSTEL MANAGEMENT SYSTEM**

A hostel management system is a software package that provides assistance in automating and managing administrative tasks involved in managing a hostel. (ALi, 2023) In a school setting, the system keeps record of the room information, student information, room allocation details, fee details and employee details of the hostel. It is also used to generate various reports ranging from student details to room status report.

## **2.2. HISTORICAL BACKGROUND OF THE RESEARCH**

For a long time the administrative processes in hostel management has been a manual process that is time-consuming and error-prone. It involves maintaining paper records of student information, room allocation, hostel dues, and other administrative tasks. This system is often inefficient and can lead to delays in processing requests and resolving issues. Taking Babcock University as a case study, which today still makes use of this traditional style of Hostel Management.

Due to a number of factors, including the increasing number of students living in hostels, the growing complexity of hostel management, and the need to improve the efficiency and effectiveness of hostel operations there has been a growing interest in developing automated hostel management systems to address the limitations of the traditional system [13]. Automated hostel management systems use computer software to streamline and automate various administrative tasks, such as student registration, room allocation, hostel dues management, and visitor management to help streamline and make the process easier. It is important to develop a hostel management system to automate a lot of these tasks especially due to the exponential increase in population in school. N [14].

Over time, technological advancements and the growing demand for better hostel management solutions spurred further innovation. The historical evolution of Hostel Management Systems is marked by key milestones, such as the transition from on-premises software to cloud-based solutions, the integration of access control systems for enhanced security, and the development of user-friendly interfaces for both administrators and residents. (Agrawal et al., 2022.; Vincent P.)

### **2.3. OVERVIEW OF EXISTING SYSTEM**

A lot of samples of different HMS were found and related work to them on google when searched for “Hostel Management System”. These software were examined to extract the common functionalities found in them. These include:

1. **Student Registration and Profile Management:** The systems allow for the registration and maintenance of student or resident profiles, including personal details, contact information, and room preferences.

2. **Room Allocation and Reservation:** Efficiently allocate rooms to students based on various criteria, such as room type, availability, and student preferences. They also commonly support online room reservation and selection.
3. **Billing and Payment Processing:** The reviewed systems provided automated billing and invoicing processes, including rent, additional charges, and deposits and online payment options for students, providing convenience and reducing the need for manual transactions.
4. **Attendance Regulation:** Systems also provided room for monitoring and recording student attendance.
5. **Use access control systems for secure entry to accommodation areas:** Systems also provided secure access control to prevent unauthorized users from accessing the system.
6. **Room Maintenance and Service Requests:** They allowed students to request maintenance or services through the system, such as repairs, cleaning, or supplies. This allowed for streamlining the tracking and management of service requests to ensure timely resolution.
7. **Communication and Notifications:** The systems also provided support to facilitate communication between administrators and residents through integrated messaging and notification features.
8. **Reporting and Analytics:** The systems allowed administrators to generate reports on occupancy rates, revenue, maintenance requests, and other relevant data. This allowed the use of data analytics to make informed decisions about accommodation management.



9. User-Friendly Interface: More of the modern systems provided an intuitive and user-friendly interface for both administrators and residents to access and interact with the system.

## **2.4. REVIEW OF RELATED WORK**

In this literature survey carried out, the research study produced multiple sites and software and came across multiple samples relating to Hostel Management System. Based on public knowledge available on the Internet, reviews on some of the websites found during research.

### **2.4.1. Orah Student Information System**

A Student Information System (SIS), also known as a Student Management System (SMS), is specialized software crafted to assist educational institutions in overseeing and organizing comprehensive student information. It encompasses a wide array of data, ranging from grades, attendance records, behavioral insights, and beyond.

Orah is a comprehensive application system specifically designed to address the day-to-day challenges faced by schools, particularly those with boarding facilities. Recognizing the unique demands of managing boarding schools, Orah offers a user-friendly solution. [16].

Among its various functionalities, one crucial aspect is the meticulous tracking of students. Beyond academic pursuits, ensuring the safety and well-being of students is paramount for boarding schools [17]. Orah addresses this need by providing a robust system that enables schools to know the whereabouts of each student promptly. In emergency situations, quick access to information from any location within the school is crucial. The cloud-based system includes dedicated apps for both staff and students, allowing the latter to log their movements easily. Staff, regardless of their location on the

premises, can access real-time information about students' locations. [16]. The live roll feature facilitates multiple staff members accounting for pupils simultaneously, swiftly identifying any missing students. Managing student movement is a significant challenge for boarding schools, and Orah effectively streamlines this process. [18]

Another key functionality is the efficient management of student leave. Given the dynamic nature of boarding school life, students frequently come and go for various activities. Regardless of the reason for student leave, it is imperative that approvals are obtained and responsibilities for the child's safety are appropriately delegated. Orah simplifies this process by allowing students or parents to submit leave requests. The school can establish its own approval pathways, and automated notifications streamline communication for boarding house staff, eliminating unnecessary administrative burdens. The approval process includes setting terms and conditions, ensuring that individuals hosting students during breaks understand their responsibilities. The cloud-based system facilitates real-time tracking of students' locations, allowing staff to check on them remotely without the need for shift change handovers. In certain situations, secondary school pupils can even sign themselves back in, further enhancing the efficiency of the system. [16]

#### **2.4.2 Zuan Hostel Management System**

Zuan Hostel Management System (HMS) is an advanced software application designed to assist hostel managers in automating and streamlining their day-to-day operations [19]. It encompasses a broad array of features, including:

- i. **Student Admissions and Room Allocation:** Zuan HMS efficiently handles the student admissions process and allocates rooms based on individual preferences and requirements.

- ii. **Fee Collection:** The system automates the fee collection process, offering residents various payment options for added convenience.
- iii. **Maintenance Management:** Zuan HMS effectively tracks and manages maintenance requests from residents, ensuring prompt resolution of issues.
- iv. **Visitor Management:** The system facilitates the tracking and management of visitors to the hostel, enhancing security and control.
- v. **Reporting:** Zuan HMS generates diverse reports on hostel occupancy rates, room availability, and other key metrics, aiding informed decision-making.
- vi. **Online Booking:** Residents can conveniently book rooms online, providing them with an accessible way to plan their stay.
- vii. **Mobile App:** Zuan HMS extends its functionality with a mobile app, enabling residents to access account information, check room availability, and submit maintenance requests on the go.
- viii. **Customization:** The system can be tailored to meet the specific needs of each hostel, ensuring adaptability.
- ix. **Training:** The system offers comprehensive training to hostel staff, ensuring proficiency in utilizing its features effectively.

Zuan HMS stands out as a comprehensive and flexible hostel management system, empowering managers to enhance the efficiency and effectiveness of their daily operations. [19]

#### **2.4.3 Summary of Reviewed Literatures**

Orah is highlighted as a solution designed to address the day-to-day challenges faced by schools with boarding facilities. It emphasizes meticulous student tracking as a crucial aspect, ensuring the safety and well-being of students. The system employs a cloud-based approach with dedicated apps for both staff and students, allowing real-time

monitoring of student movements. The live roll feature aids in quickly identifying missing students. Additionally, Orah efficiently manages student leave, providing a streamlined process for request submission, approvals, and real-time tracking of student locations during leave periods. The system aims to simplify administrative tasks and enhance overall efficiency in managing boarding school operations.

Zuan HMS is presented as an advanced software application catering to hostel managers for automating and streamlining day-to-day operations. Its features cover various aspects, including student admissions and room allocation, fee collection, maintenance management, visitor management, reporting, online booking, and a mobile app for residents. The system's flexibility is emphasized through customization options, enabling adaptation to the specific needs of each hostel. Zuan HMS also offers implementation support, training for staff, and ongoing technical support. Overall, Zuan HMS is portrayed as a comprehensive and flexible solution empowering hostel managers to enhance the efficiency and effectiveness of their operations.

In summary, both Orah and Zuan HMS address specific challenges faced by educational institutions with boarding facilities, offering tailored features to streamline processes and improve overall management efficiency. Orah primarily focuses on student tracking and leave management, while Zuan HMS provides a broader set of features for hostel management, emphasizing adaptability and support services.

## **2.5. STRENGTH AND WEAKNESS OF EXISTING SYSTEM**

The strengths of a Hostel Management System (HMS) can significantly enhance the management of hostel facilities. With HMS, residents don't need to physically visit administrative offices, saving time and effort. This is particularly beneficial for busy

students who can handle tasks like room allocation, fee payments, and incident reporting from the convenience of their rooms.

HMS could streamline communication for hostel residents of various ages. From simple concerns like maintenance requests to more complex matters, such as security issues, the system allows for efficient communication and issue resolution.

However, as with other Online Software, concerns exist. Some residents may be hesitant to fully embrace virtual interactions for sensitive matters, such as roommate conflicts. Additionally, the system should address potential privacy concerns to ensure data security.

In the context of African countries, where telecommunication infrastructure can be a challenge, an HMS should be designed with consideration for varying levels of internet access and affordability. This includes developing mobile-friendly interfaces and possibly incorporating alternative communication methods.

Furthermore, policy frameworks and regulations need to be established to guide the implementation of Hostel Management Systems across different regions. This ensures ethical practices, protects user data, and sets standards for the responsibilities of both administrators and residents.

Strengths of existing Hostel Management Systems (HMS) include:

- i. Improved efficiency and productivity: HMS can automate many of the time-consuming tasks involved in hostel management, freeing up staff to focus on more important things.
- ii. Improved accuracy: HMS can help to reduce the risk of errors in data entry and other manual tasks.

- iii. Increased transparency and accountability: HMS can provide hostel managers with real-time data on occupancy rates, room availability, and other key metrics. This data can be used to make better decisions about resource allocation and to improve the overall management of the hostel.
- iv. Enhanced customer service: HMS can help hostel managers to provide better customer service to their residents by making it easier for them to book rooms, pay fees, and submit maintenance requests.
- v. Comprehensive features: Most HMS offer a wide range of features, including student admissions and room allocation, fee collection, maintenance management, visitor management, reporting, online booking, and mobile apps. This makes them a one-stop shop for hostel managers who are looking for a complete solution for managing their operations.

**Weaknesses of existing HMS include:**

- i. Complexity: Some HMS can be complex and difficult to use, especially for hostel managers and staff who are not tech-savvy.
- ii. Cost: HMS can be expensive, especially for hostels on a tight budget.
- iii. Limited customization options: Some HMS do not offer many customization options, which can be a limitation for hostel managers who have specific needs.
- iv. Poor integration with other systems: Some HMS do not integrate well with other software systems that hostel managers may be using, such as student information systems or accounting systems.
- v. Lack of user-friendliness: Some HMS are difficult to use, both for hostel managers and residents. This can lead to frustration and errors.

## **2.6. GENERAL COMMENTS**

In conclusion, Hostel Management Systems have evolved significantly from manual processes to sophisticated, automated systems. They offer numerous advantages, but it's essential to address their weaknesses for optimal performance. The body of literature reviewed here provides valuable insights into the development, implementation, and impact of these systems, offering a foundation for further research and improvement in the field of Hostel Management Systems.

## **CHAPTER THREE**

### **SYSTEM ANALYSIS AND DESIGN**

#### **3.0. INTRODUCTION**

This chapter explicitly elaborates the methodology used in the achievement of this project work, requirement gathering, implemented system, database structure and development, database design, input and process design and so on, we also considered the methods used for the creation of the application that is meant to automate the allocation of the hostels for the students of the university.

#### **3.1. RESEARCH DESIGN**

The chosen software process model for this project is the Iterative hybrid model. This model combines elements of incremental and iterative approaches, facilitating continuous software delivery to meet customer needs.

##### **3.1.1. ITERATIVE PROCESS MODEL**

The Iterative Hybrid Agile process model combines the iterative approach, where development occurs in cycles, and elements of agile methodologies. This model is a flexible and adaptive approach that incorporates the best practices of both iterative and Agile methods to manage complex projects.

The key characteristics and advantages of the Iterative Hybrid Agile process model which led to its use for the research project include:

1. Incremental Development: Similar to iterative models, the development process occurs in iterations or cycles. However, in this hybrid model, each



iteration may incorporate Agile practices of delivering small, incremental features.

2. **Flexibility:** It allows for adjustments and refinements throughout the project lifecycle. This adaptability is similar to Agile methodologies, enabling changes in requirements and priorities even during the development process.
3. **Iterative Feedback:** Like iterative models, there's a focus on obtaining feedback at various stages of development, allowing for continuous improvement and adjustment based on stakeholder input.
4. **Agile Principles Integration:** The model may adopt Agile principles such as collaboration, customer involvement, and adaptive planning. It embraces Agile practices while still following an iterative development structure.
5. **Risk Management:** By breaking down the project into iterations and integrating Agile practices like continuous feedback, the model aims to mitigate risks by addressing issues incrementally.
6. **Deliverables and Customer Satisfaction:** It emphasizes the continuous delivery of functional components or features, ensuring that customers receive value-added functionalities early in the development process.

The Iterative Hybrid Agile model is well-suited for the Hostel Management System as the requirements are subject to change allowing for flexibility in adapting to evolving customer needs.

It combines the structured approach of iterative models with the adaptability and customer-centric focus of agile methodologies, aiming to strike a balance between predictability and flexibility in managing complex projects.

## **3.2. SYSTEM REQUIREMENTS**

This part is the area that gives the descriptions to what services are to be rendered by the system. It describes the functionalities of the system as a whole. The system requirement is divided into two:

- i. Functional Requirements
- ii. Non – Functional Requirements

The functional requirements describe what a system should do while the non – functional requirements place constraints on how the system will do.

### **3.2.1. USER REQUIREMENTS**

1. The user shall be able to log in securely to the system with unique credentials (username/password or other authentication methods).
2. The user shall have different roles (admin, staff, and student) with specific permissions and access levels.
3. The system shall allow students to view and update their personal information (contact details, room allocation).
4. The user shall be able to manage room allocations, including assigning rooms to students, checking vacancies, and updating room statuses.
5. The system shall facilitate staff members in managing hostel facilities (maintenance requests, inventory, cleaning schedules)
6. Users shall have access to a calendar/scheduling feature to book common facilities
7. The system shall support financial functionalities, enabling users to manage payments (fees, fines, refunds) and generate financial reports.

8. There shall be a communication module allowing announcements, notifications, and messages to be sent between staff and students or among users.
9. The system shall maintain a comprehensive database for hostel records, including student profiles, staff details, room inventories, and historical data.
10. The user shall have access to reporting tools to generate various reports (occupancy, maintenance logs, and financial summaries) as per specific requirements.
11. The system shall provide an easy-to-use interface that is intuitive, responsive, and accessible across different devices (desktop, mobile).
12. Users shall have the ability to track and manage security aspects, such as visitor logs, access control, and emergency protocols.

### **3.2.2. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

#### **3.2.2.1. FUNCTIONAL REQUIREMENTS**

1. The system shall provide a secure login mechanism allowing users to access the system with unique credentials.
2. The system shall differentiate between user roles (admin, staff, and student)
3. The system shall assign different access levels (admin, staff, student) with specific permissions.
4. The system shall enable students to view and update their personal information, including contact details and room allocations.
5. The system shall allow staff to assign check room availability
6. The system shall automatically update room availability.
7. The system shall allow students to make maintenance requests
8. The system shall allow students to make complaints.

9. The system shall manage financial transactions, enabling users to handle payments (fees, fines, refunds)
10. The system shall generate financial reports.
11. The system shall incorporate a communication module to facilitate announcements, notifications
12. The system shall maintain a comprehensive database to store hostel records, including student profiles, staff details, room inventories, and historical data.
13. The system shall offer reporting tools for users to generate various reports (occupancy, maintenance logs, and financial summaries) as required.
14. The system shall provide an intuitive, responsive, and accessible to be used on the web.
15. The system shall include log in every action carried out on the system.

#### **3.2.2.2. NON – FUNCTIONAL REQUIREMENTS**

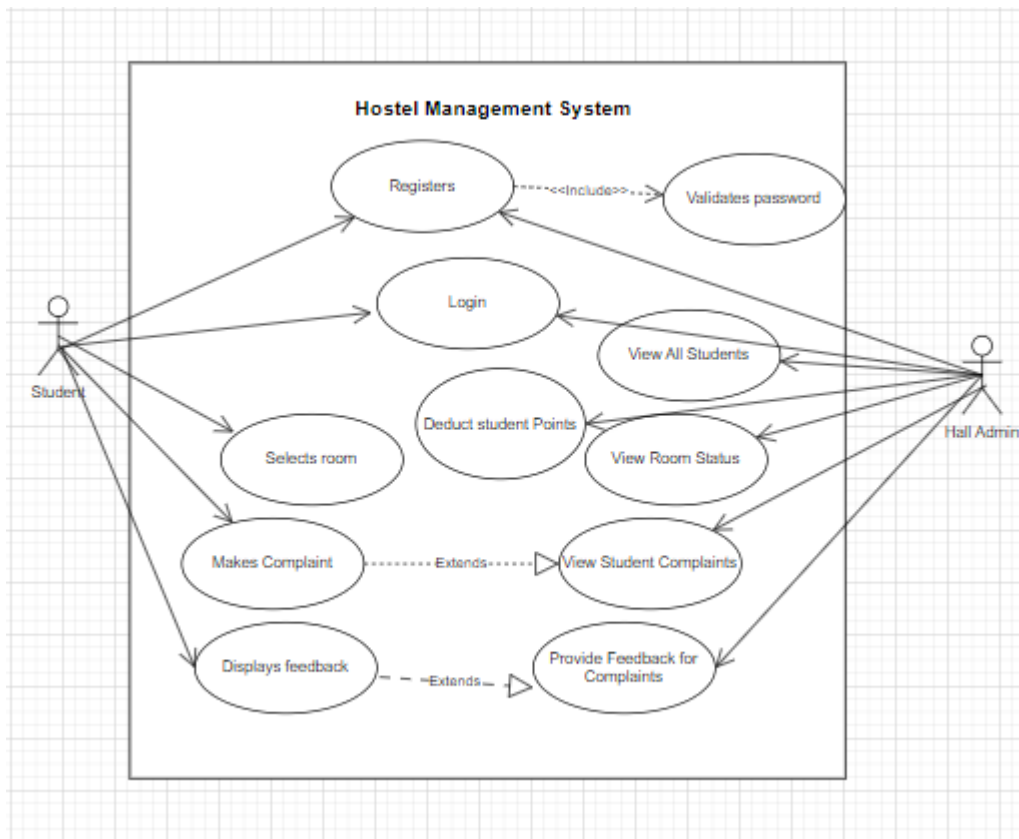
1. The system shall respond to user interactions within 2 seconds for optimal user experience.
2. The system shall handle a minimum of 500 simultaneous users without performance degradation.
3. The system shall maintain 99.9% uptime, ensuring availability for users at all times.
4. The system shall employ encryption methods to protect sensitive data like personal information and financial records.
5. The system shall be compatible with major web browsers (Chrome, Firefox, Safari) and mobile devices (iOS, Android).
6. The system shall provide clear and concise instructions to users for seamless navigation and operation.

7. The system shall comply with accessibility standards (WCAG) to ensure usability for users with disabilities.
8. The system shall perform regular data backups and have a recovery plan in place in case of system failures.
9. The system shall maintain audit trails to track user activities and changes made within the system.
10. The system shall have comprehensive user and administrative documentation for reference and training purposes.
11. The system shall adhere to relevant data protection laws (GDPR, HIPAA) and industry standards.
12. The system shall have a dedicated support team and regular maintenance schedules for updates and bug fixes.

### **3.3. SYSTEM ARCHITECTURE**

#### **3.3.1. USE CASE DIAGRAM**

A use case is a narrative description of a specific interaction between a user and a system. It outlines the steps the user takes to achieve a specific goal, and the system's responses to those steps. Use cases are used in software development, systems engineering, and other fields to capture and communicate the requirements of a system. An actor may be a class of users, roles users can play, or other systems. Cockburn (1997) distinguishes between primary and secondary actors. A primary actor is one having a goal requiring the assistance of the system. A secondary actor is one from which the system needs assistance.



*Fig 3. 1. Use case Diagram for Hostel Management System*

The use case diagram above (Fig 3.1.) shows how the system works for two actors: student and hall admin.

The Student can:

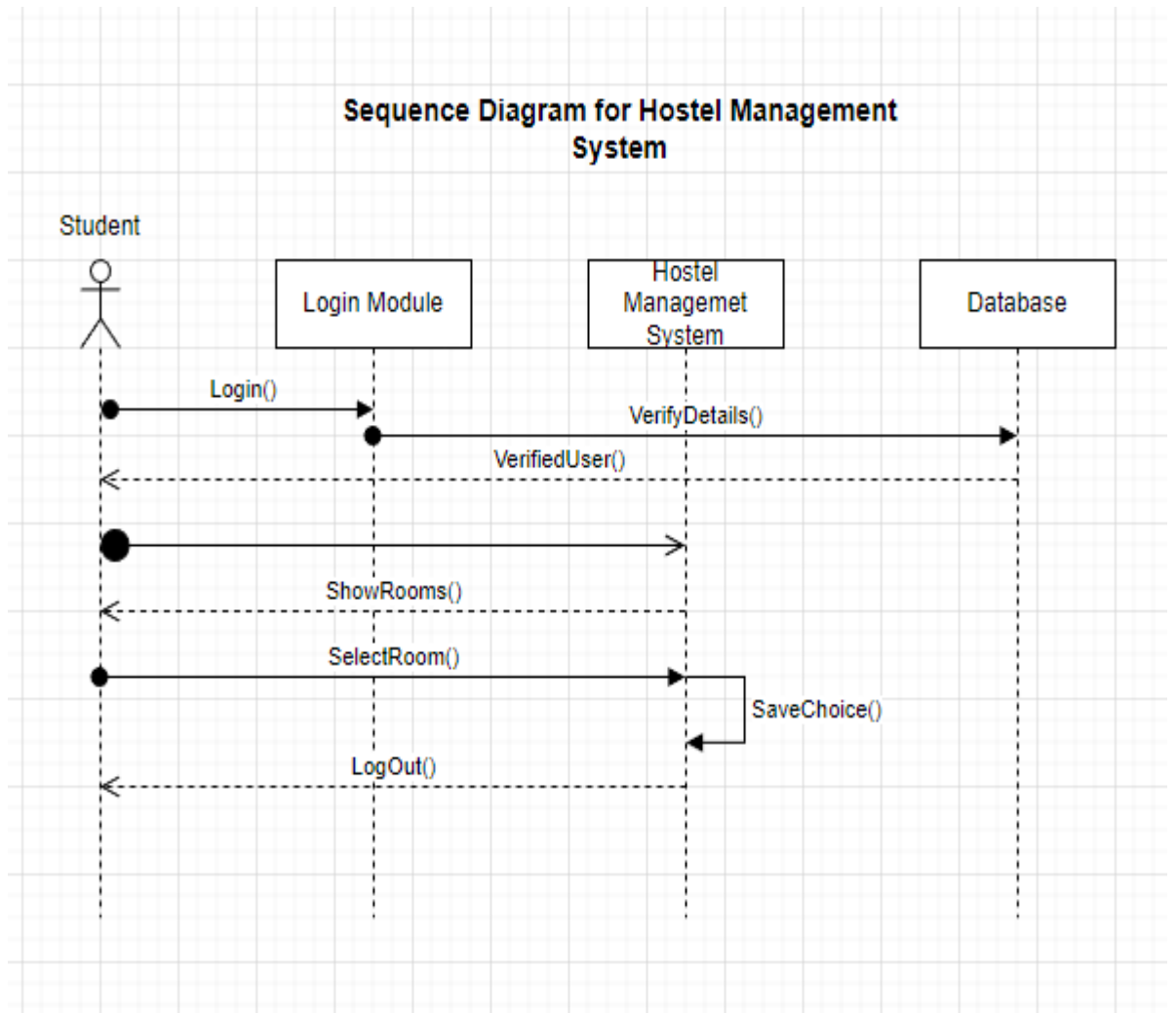
- i. Login by registering, including validating their password
- ii. View all students
- iii. Select a room
- iv. View room status
- v. Makes complaints
- vi. View student complaints (which extends to providing feedback for complaints)
- vii. Display feedback

The Hall Admin can:

- i. Login by registering, including validating their password
- ii. View all students registered in the hall
- iii. Assign deduction points to Students

### **3.3.2. SEQUENCE DIAGRAM**

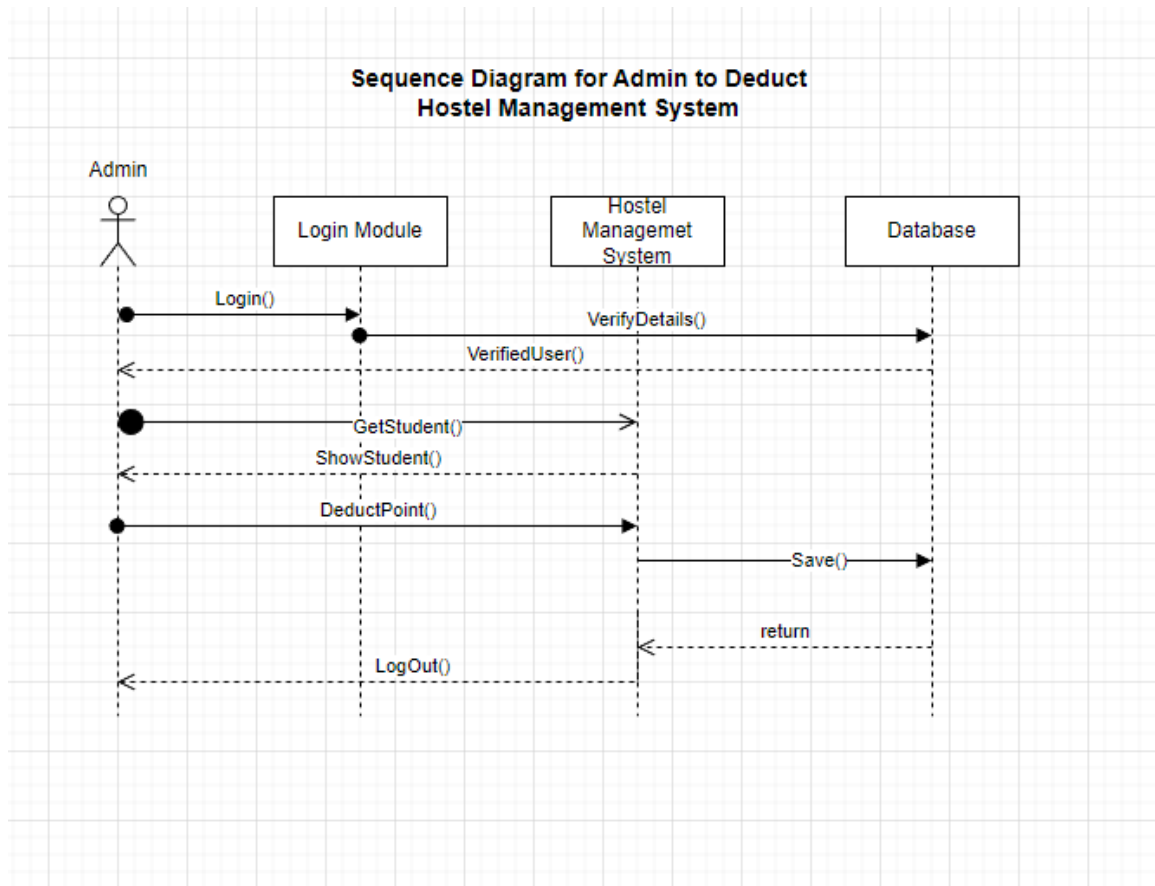
A sequence diagram is a type of interaction diagram in Unified Modeling Language (UML) that illustrates how objects in a system interact and collaborate in a specific scenario or sequence of events. It represents the flow of messages, actions, and interactions between various objects or components within the system over a period of time. It contains objects and shows how the objects relate to each other and their timeline.



*Fig 3. 2 Sequence Diagram for Hostel Management System*

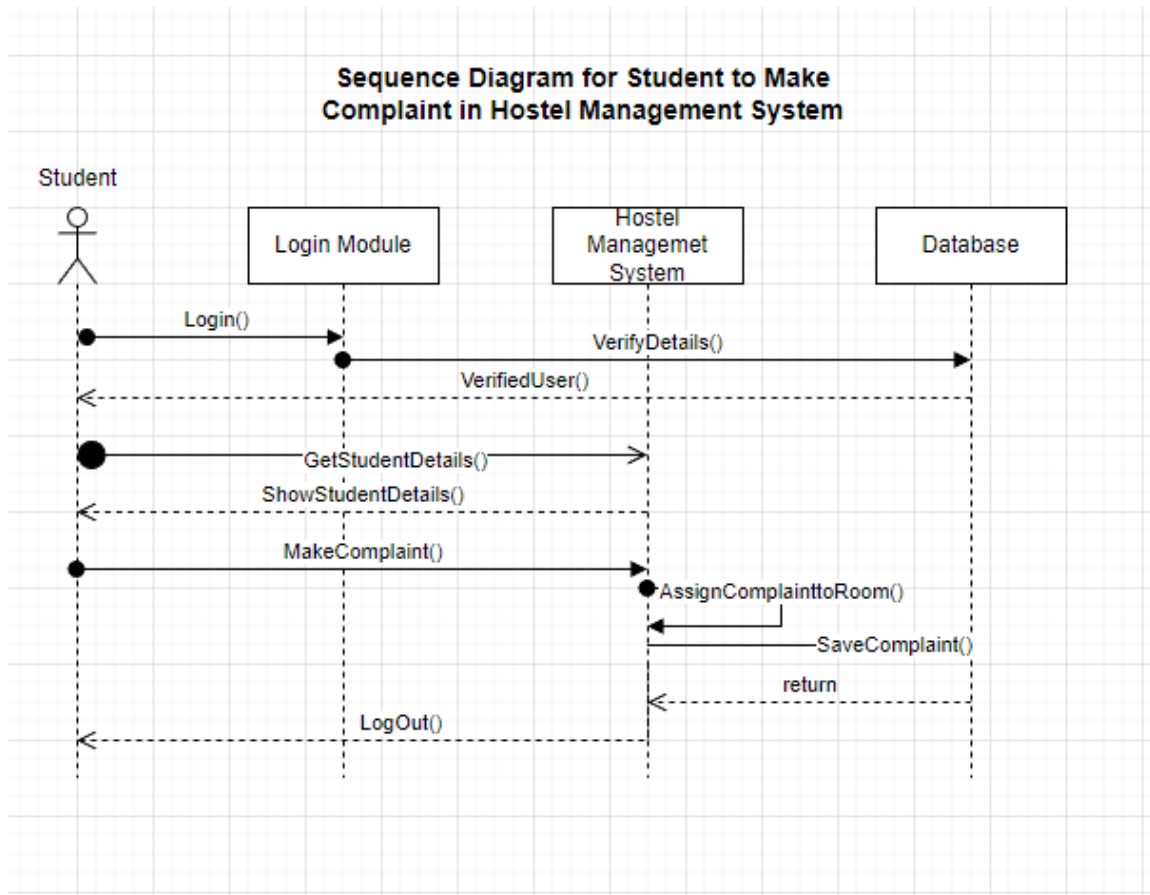
The image above (Fig 3.2) shows the sequence diagram for a student logging in the hostel management system. The sequence diagram depicts a student logging in, having their credentials verified, viewing available rooms, selecting a room, saving their choice and finally logging out, with the system interacting with a database to verify details and save the selection.





*Fig 3. 3Sequence Diagram showing Hall admin deduct points*

The above image (Fig 3.3) shows the interaction between an admin and the hostel management system for deducting points from a student. The hostel management system's sequence diagram shows an admin logging in, having their credentials verified, retrieving student information, deducting points from a student's account (which likely involves saving the update to a database), receiving a response, and finally logging out.



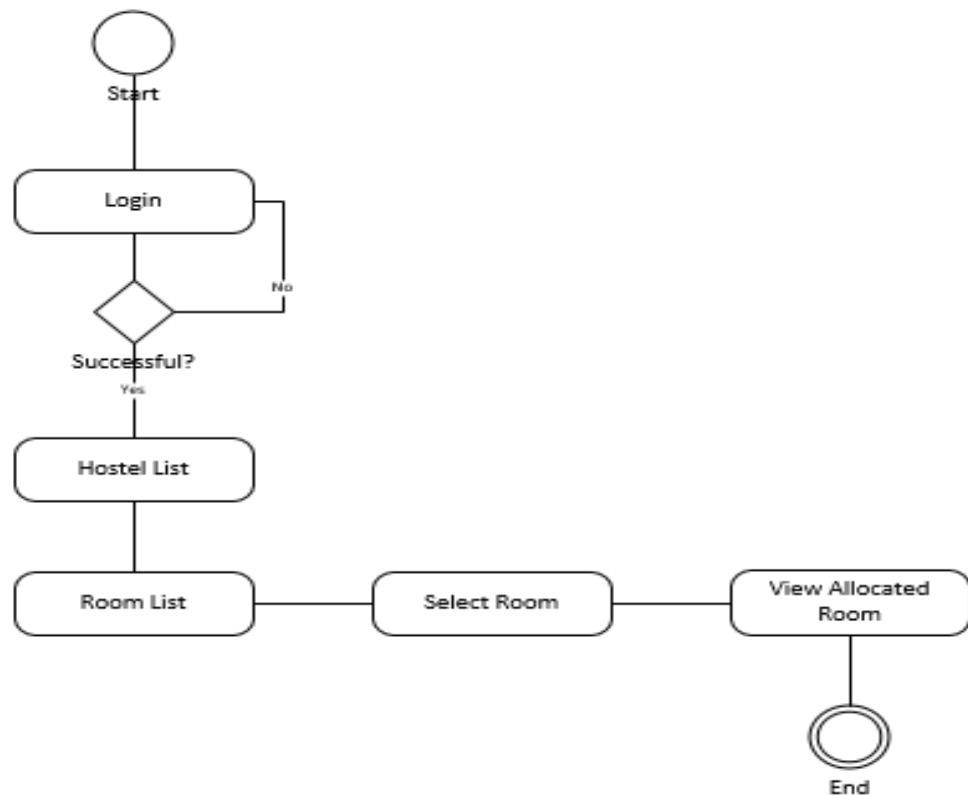
*Fig 3. 4 Sequence Diagram showing Student making Complaint*

The sequence diagram above (Fig 3.4) depicts the interaction between a student and the hostel management system for making a complaint. The hostel management system's sequence diagram showcases a student logging in, having their credentials verified, viewing their information, initiating a complaint process which involves submitting details and likely assigning it to their room, saving the complaint to a database, and finally logging out.

### 3.3.3. ACTIVITY DIAGRAM

An activity diagram is a visual representation within Unified Modeling Language (UML) used to illustrate the flow of activities or workflows within a system or process. It's particularly useful for modeling the dynamic aspects of a system, focusing on the sequence of actions, decision points, parallel activities, and the overall flow of control.

The activity diagram contains circular nodes (start and end point) which indicate the beginning and end of the flow of activities performed in the system.



**Activity Diagram for Hostel Management System**

*Fig 3. 5Activity Diagram for a student Logging in in Hostel Management System*

The hostel management system's activity diagram (Fig 3.5) illustrates a student booking a room. They first login, with the system verifying their credentials. Upon success, the system shows available hostels. The student chooses a hostel, then a room from the available options within that hostel. The system checks room availability and allocates it if possible. If unavailable, the student selects another room. Finally, the system allocates and displays the booked room information.

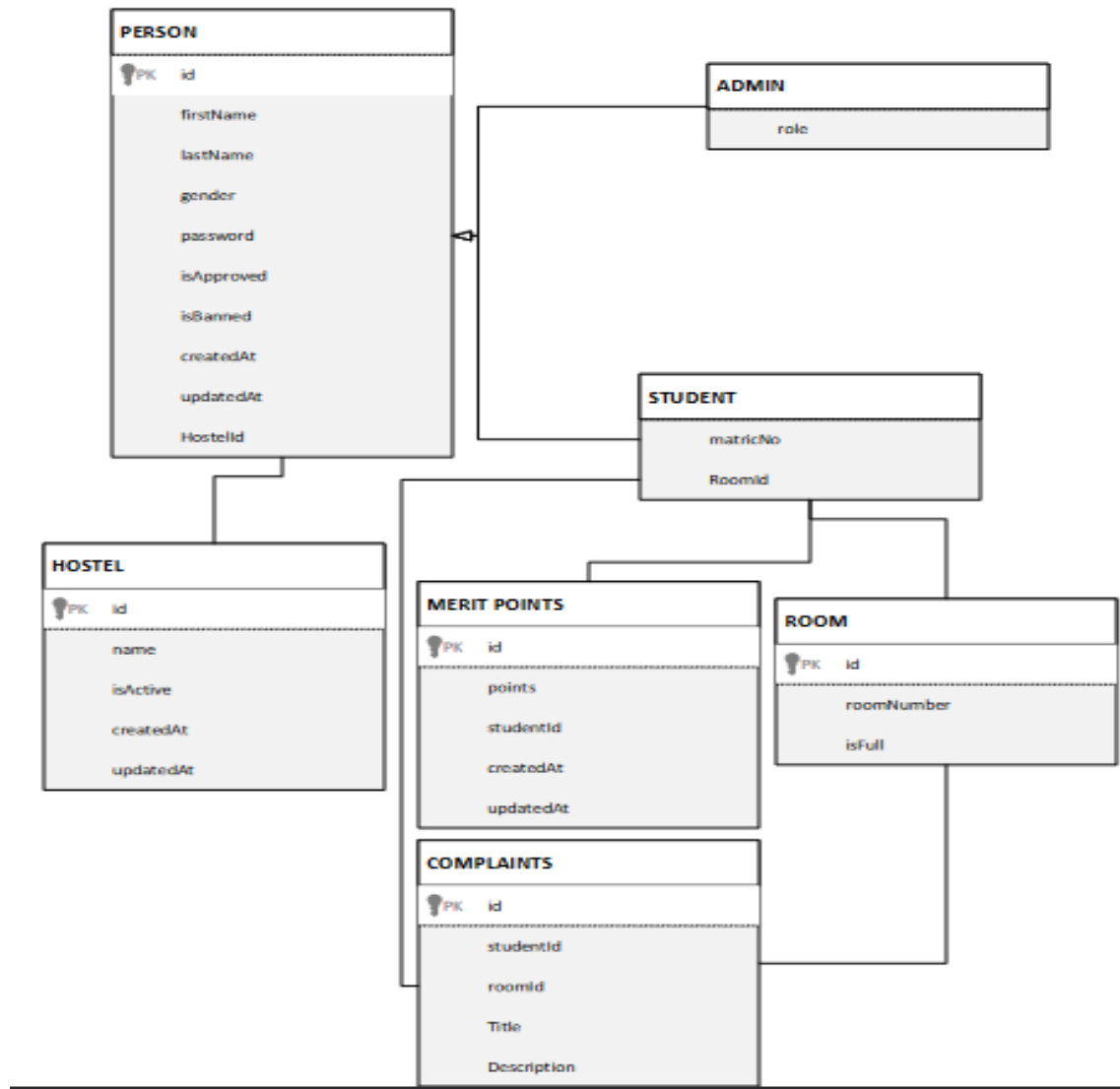
### **3.3.4. ENTITY RELATIONSHIP DIAGRAM**

An Entity-Relationship Diagram (ERD) is a visual representation used to model the structure of a database. It illustrates the entities within a system or a database, their attributes, and the relationships between them.

Entities are objects or concepts in the real world that have data stored about them. They are represented as rectangles in an ERD. Attributes describe properties of entities and are shown within the rectangles.

Relationships depict how entities are connected or associated with each other. They're typically shown as lines connecting entities, describing the nature and cardinality (such as one-to-one, one-to-many, or many-to-many) of the associations.

ERDs are essential tools in database design as they provide a clear visualization of the database structure, aiding in understanding the relationships between different elements and ensuring the proper organization and functionality of the database system.



*Fig 3. 6Entity Relationship Diagram for Hostel Management System*

The hostel management system's ERD (fig 3.6) showcases entities like People (with attributes including name, ID, and approval status), Students (inheriting from People and having details like hostel and room assignments), and Admins (also inheriting from People). Hostels have names and activity status, while Rooms have IDs and occupancy information. Merit points are tracked with dedicated entities linked to Students. Complaints connect to Students and their rooms, containing details like title and description. Relationships include inheritance for People subtypes (Student/Admin), Students belonging to a Hostel and Room, one Hostel having many Students and Rooms,

Rooms belonging to one Hostel and occupied by at most one Student, Students having multiple MeritPoint records, and a Complaint being submitted by a Student for a specific Room. The ERD is the basis for the database schema used in this project.

### **3.4. DEVELOPMENT TOOLS**

For the development and Implementation of the Hostel Management System, the following tools will be employed.

1. **C#:** C# is a robust, object-oriented programming language commonly used in developing Windows applications and web services. In the context of a Hostel Management System, C# can be employed for backend development. It's known for its strong typing, scalability, and integration capabilities with Microsoft technologies. It can handle complex logic, data processing, and database interactions efficiently.
2. **React.js:** React.js is a popular JavaScript library for building user interfaces. It's known for its component-based architecture and reusability, making it suitable for creating responsive, interactive, and dynamic front-end interfaces. In the context of a Hostel Management System, React.js can be utilized to create a user-friendly and intuitive interface for hostel staff, students, and administrators to interact with the system.
3. **Figma:** Figma is a collaborative interface design tool used for creating wireframes, prototypes, and designs. It allows teams to work together in real-time, making it easier to design and iterate on the user interface of the Hostel

Management System. Designers and developers can create mockups, visualize the interface, and gather feedback from stakeholders before actual implementation.

4. MS SQL (Microsoft SQL Server): MS SQL is a relational database management system (RDBMS) developed by Microsoft. It's widely used for storing and managing structured data. In a Hostel Management System, MS SQL can be employed as the database backend to store information such as student details, room allocations, payment records, staff information, and more. Its reliability, scalability, and security features make it suitable for handling the data requirements of a hostel management application.

## **CHAPTER FOUR**

### **4.0. INTRODUCTION**

The detailed analysis and design done in the previous chapter culminated into discussion of results in this chapter. Here, concluding implementation and results were comprehensively captured. This chapter, which focuses on the execution of the design and development of a hostel management system (HMS), represents a significant turning point in our research.

Now that we have moved from the theoretical to the practical domain, we will focus on the complex procedure of putting the Hostel Management System into practice. During this phase, the conceptualized design is translated into a workable and intuitive system that satisfies the particular requirements and difficulties faced by hostel management. By bridging the theoretical and practical divide, the intended features and functions will be smoothly incorporated into the day-to-day operations of hostel management.

### **4.1. SYSTEM IMPLEMENTATION**

System implementation is a crucial phase in any project lifecycle, where theoretical designs and plans are put into action. It marks the transition from planning to execution, requiring meticulous attention to detail, coordination, and flexibility. It involves translating the conceptual framework into tangible solutions that meet the project's objectives and stakeholders' requirements. Effective implementation ensures seamless integration with existing processes, minimal disruption to operations, and maximum user acceptance. Implementing a hostel management system involves the deployment of software solutions designed to automate and optimize various aspects of hostel administration. It encompasses a range of functionalities, including student registration,



room allocation, inventory tracking and maintenance requests. The system was implemented in the form of services that help offer different advantages such as code organization, scalability, maintainability, and reusability.

#### **4.1.1. MODULES/ SERVICES**

The approach taken for this project was to breakdown the modules into different services into smaller, more manageable components is a common practice. This approach, known as micro services architecture, allows for greater flexibility, scalability, and maintainability. When combined, these services help a hostel management system run smoothly by giving administrators the resources they need to effectively oversee hostel amenities and services and improve the overall experience for both staff and students. By encapsulating specific functionalities within modular services, we can create a scalable, maintainable, and adaptable system that meets the evolving needs of educational institutions.

1. **UserService:** The UserService is in charge of overseeing system activities pertaining to users. It manages tasks including profile administration, user roles and permissions, authentication, and user registration. It manages user registration procedures, verifies user input, and saves user data in the database. It also implements user authentication techniques such token-based authentication and username/password authentication. Users can also upload images, modify preferences, update their profile information, and change their passwords. The UserService makes sure that tasks pertaining to users are completed quickly, safely, and in accordance with any applicable laws or corporate specifications.

2. **AuthService:** The system's authorization and authentication procedures are managed by the AuthService. It guarantees that privileged actions and access to protected resources are limited to authenticated and authorized users only. It can be used to generate session cookies or authentication tokens, authenticate users, and intercept and validate incoming requests. To ascertain if a user is authorized to carry out a certain operation or access a given resource, it enforces access control policies. Additionally, it creates, verifies, and revokes authentication tokens—also known as JWT tokens—that are used for user authorization and authentication. In order to safeguard the system from potential security risks and unauthorized access, the AuthService is essential to the security and integrity of the authorization and authentication processes.
  
3. **HallService:** HallService is in charge of overseeing all hostel-related operations, such as reservations for rooms, inventory control, and maintenance requests. It takes care of activities pertaining to the upkeep and administration of hostel amenities, guaranteeing a seamless and effective stay for both administrators and residents. It oversees room availability, allots spaces to students, and keeps track of room occupancy. Additionally, it makes it easier for students to submit, track, and resolve grievances and maintenance requests for hostel amenities or services, guaranteeing prompt upkeep and repairs. In order to give students a comfortable living environment, the HallService makes sure that hostel operations are effectively managed, resident demands are met quickly, and hostel facilities are kept in good condition.

#### 4.1.2. API INTEGRATION OF THE SYSTEM

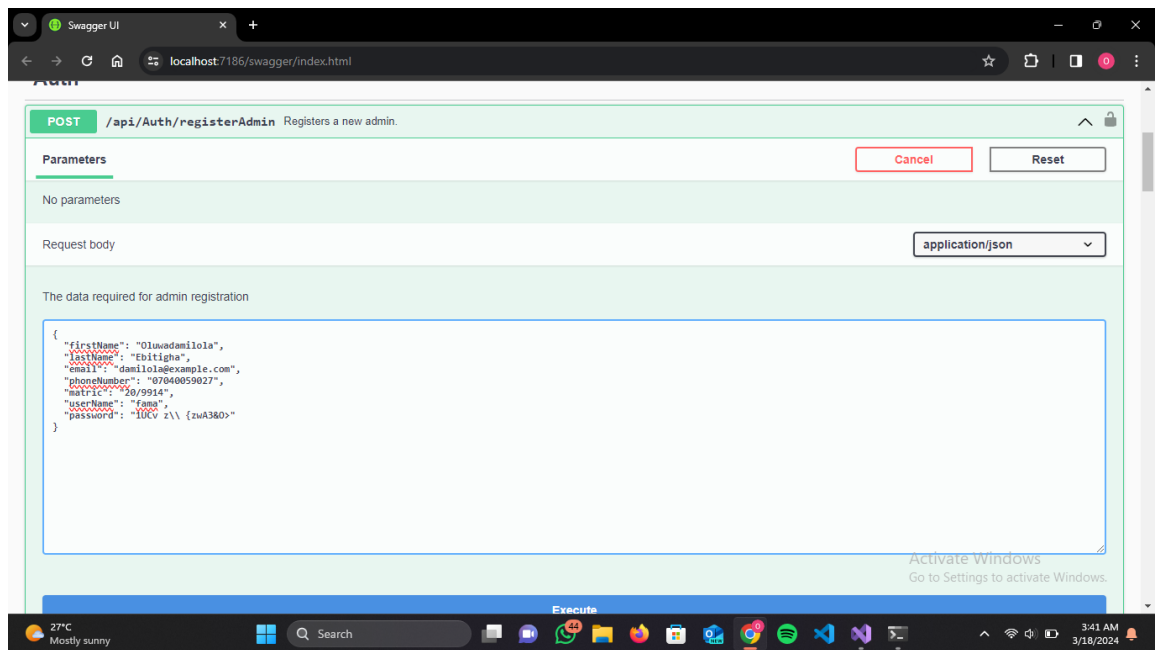


Fig 4. ISwagger documentation showing the registration API for a student

The diagram above depicts the API endpoint for creating a new user (student) in the Hostel Management system

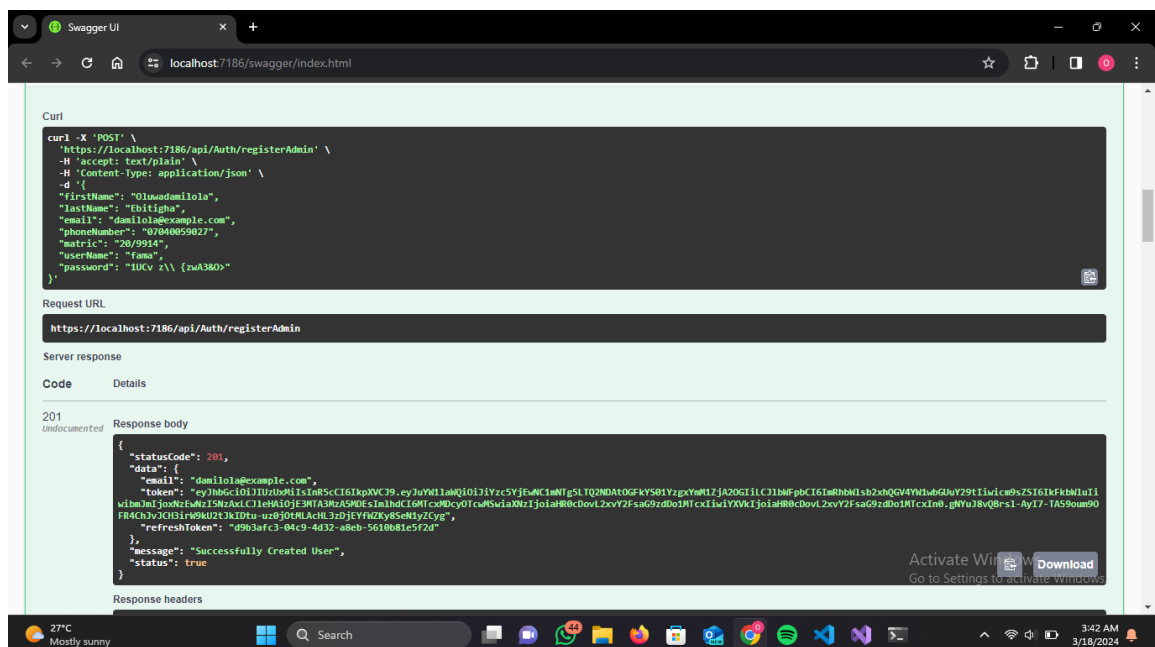


Fig 4. 2Swagger documentation showing sample of successful response to registration

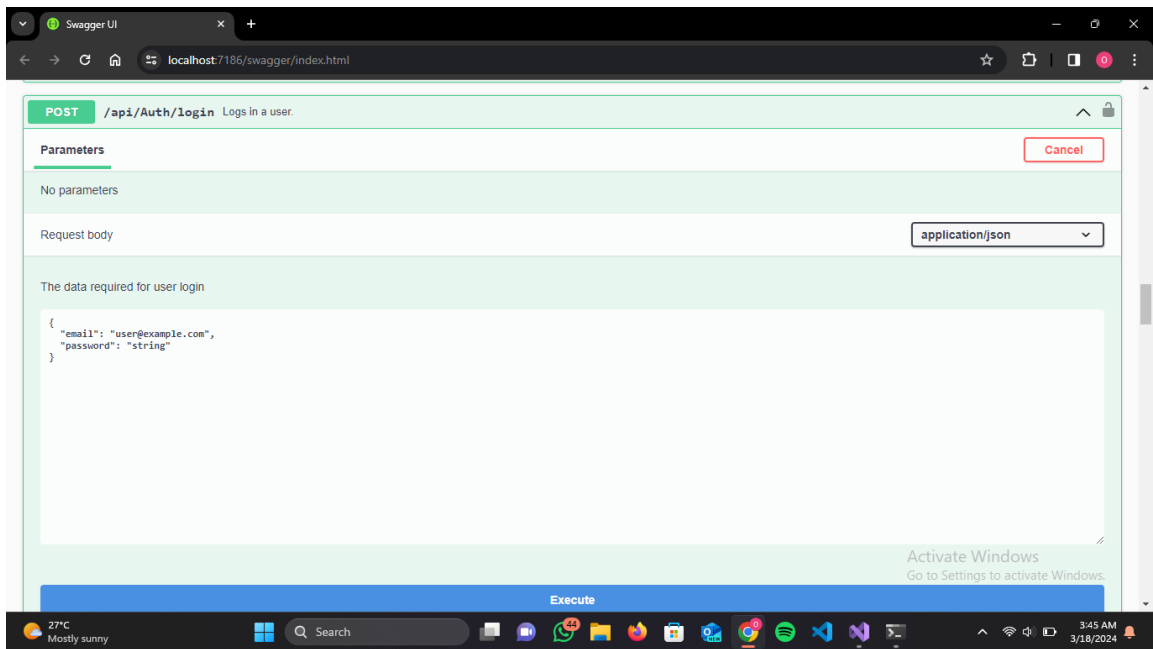


Fig 4. 3Swagger documentation for Logging in the system

The above image shows the endpoint to allow users to be able to log in and authenticate themselves into the system

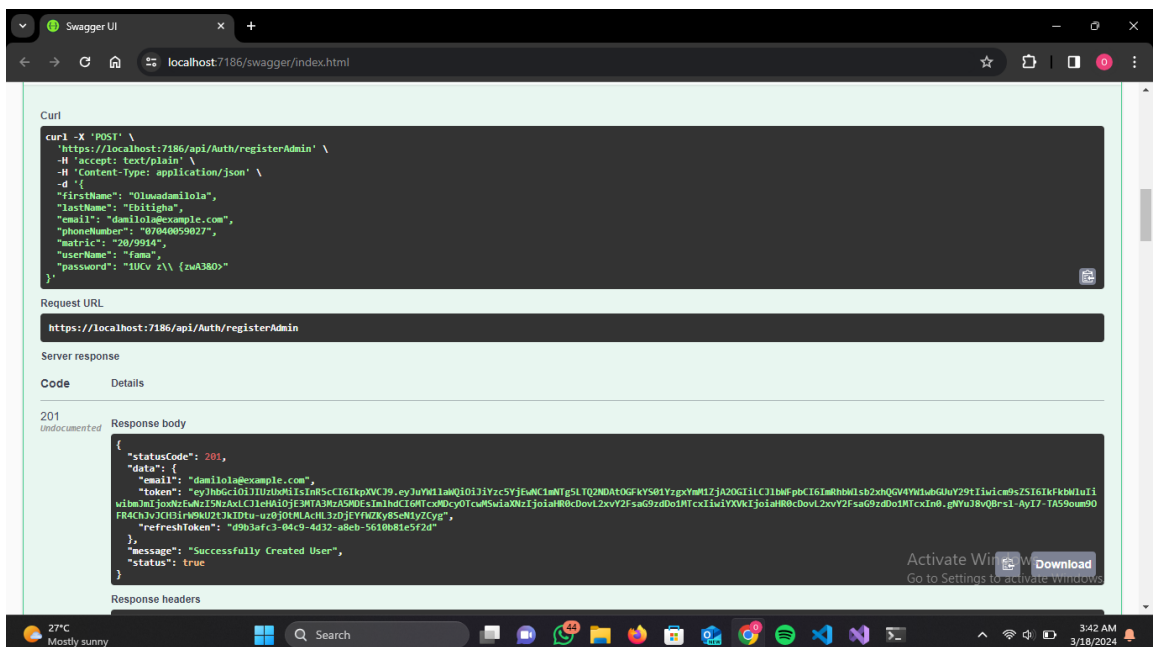


Fig 4. 4Swagger documentation for successful response to a Log in Attempt

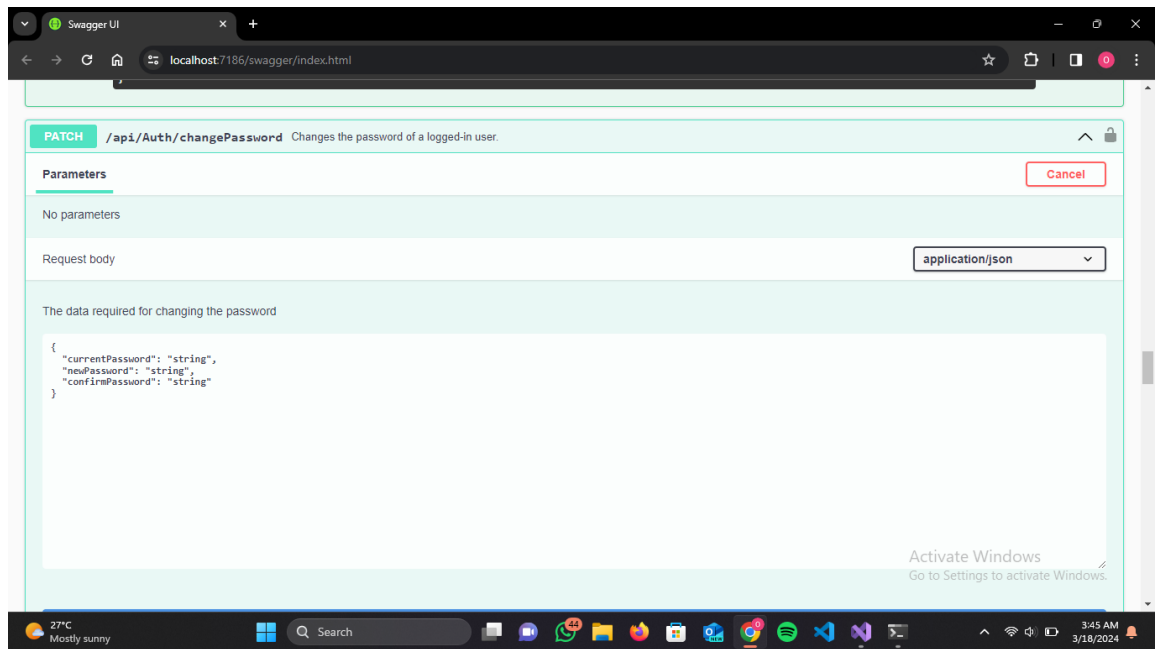


Fig 4. 5Swagger documentation showing endpoint to change password

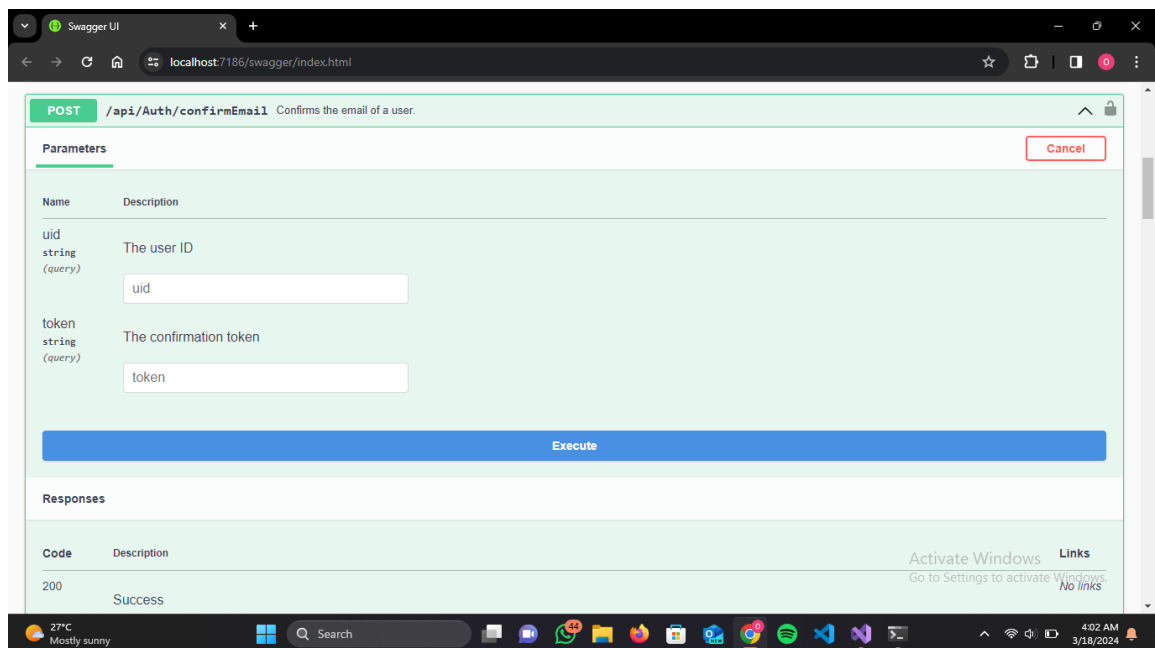
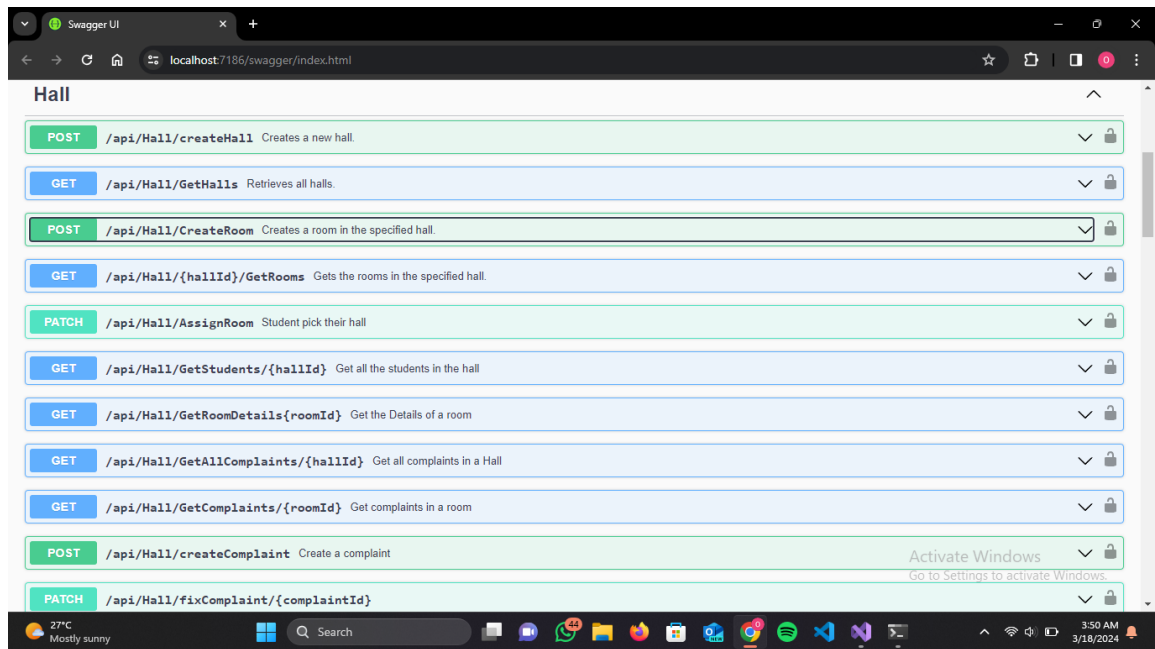
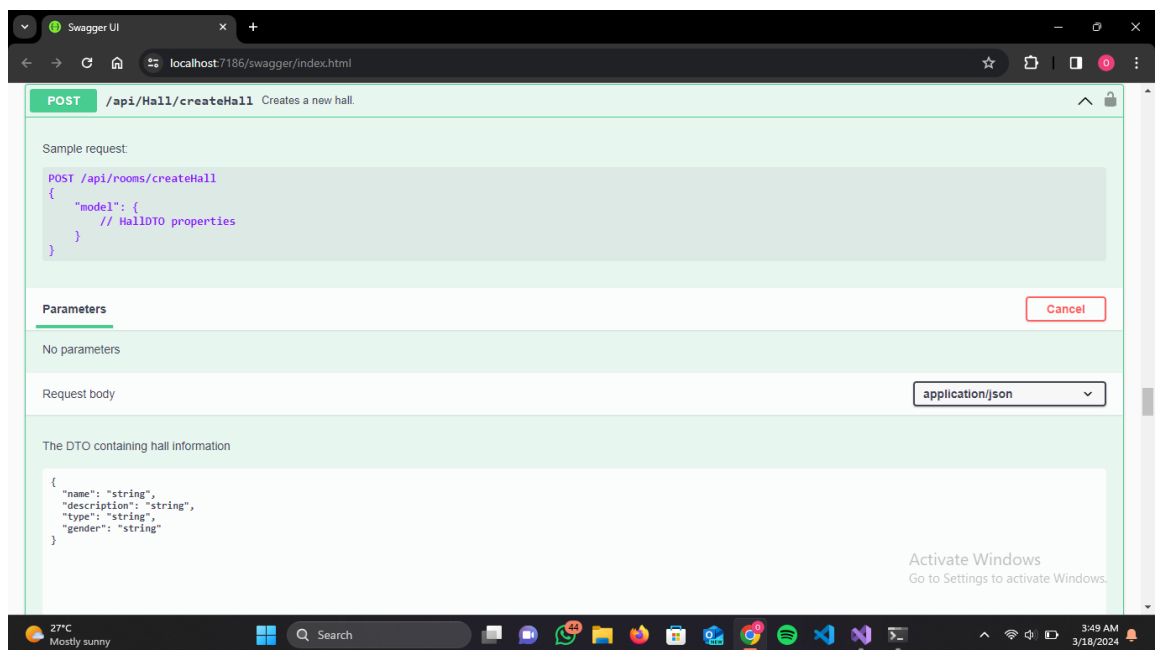


Fig 4. 6Swagger documentation for Confirming Email Addresses

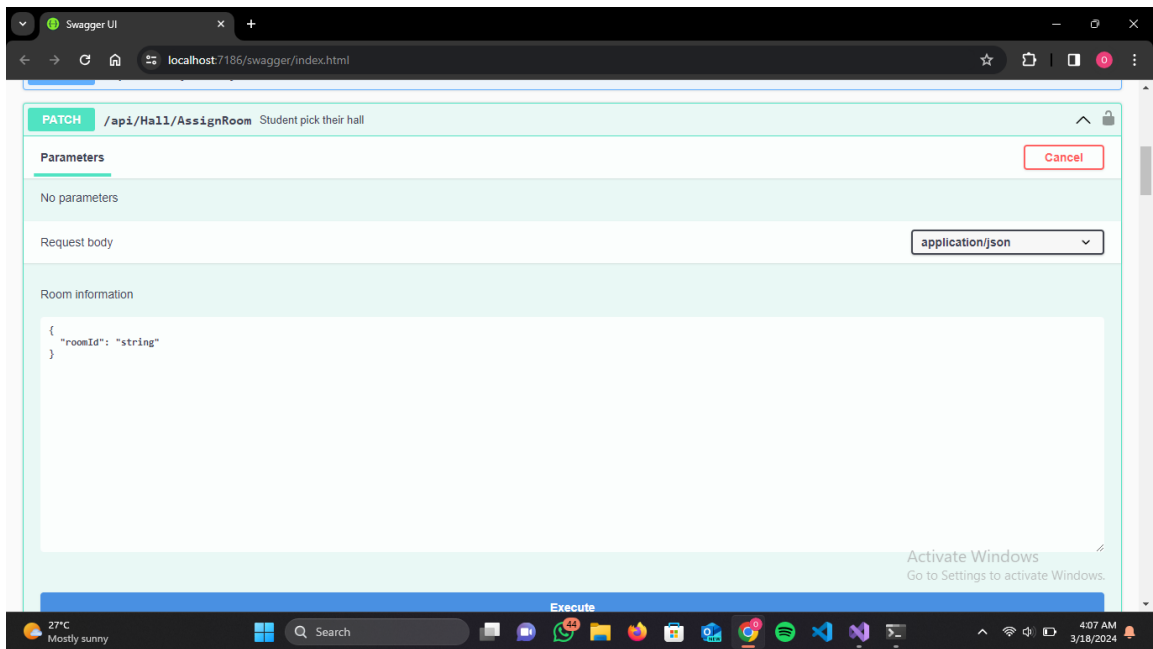


*Fig 4. 7Swagger Documentation showing HallService API endpoints*

The above image shows the endpoints available for the Hostel Management System which are essential services which range from creation of halls and rooms and changes to the status of the rooms and being able to retrieve and manage information about a student and a room.

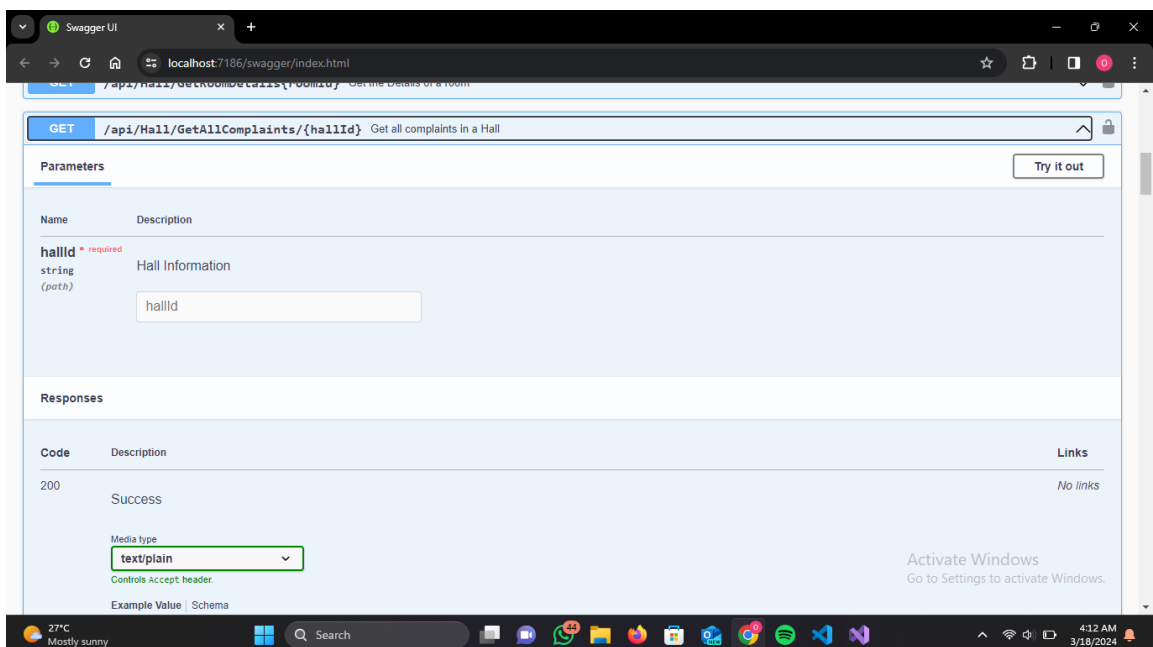


*Fig 4. 8Swagger documentation showing Creation of Halls*



*Fig 4. 9Swagger documentation that shows endpoint for Assigning rooms to Students*

This endpoint allows student to pick the halls by putting in the roomId or Name of the room they want to lodge in. The API gives a response based on if the transaction is successful or if the room is full and gives appropriate information to work with.



*Fig 4. 10Swagger documentation on the endpoint to retrieve all Endpoints in a Hall*

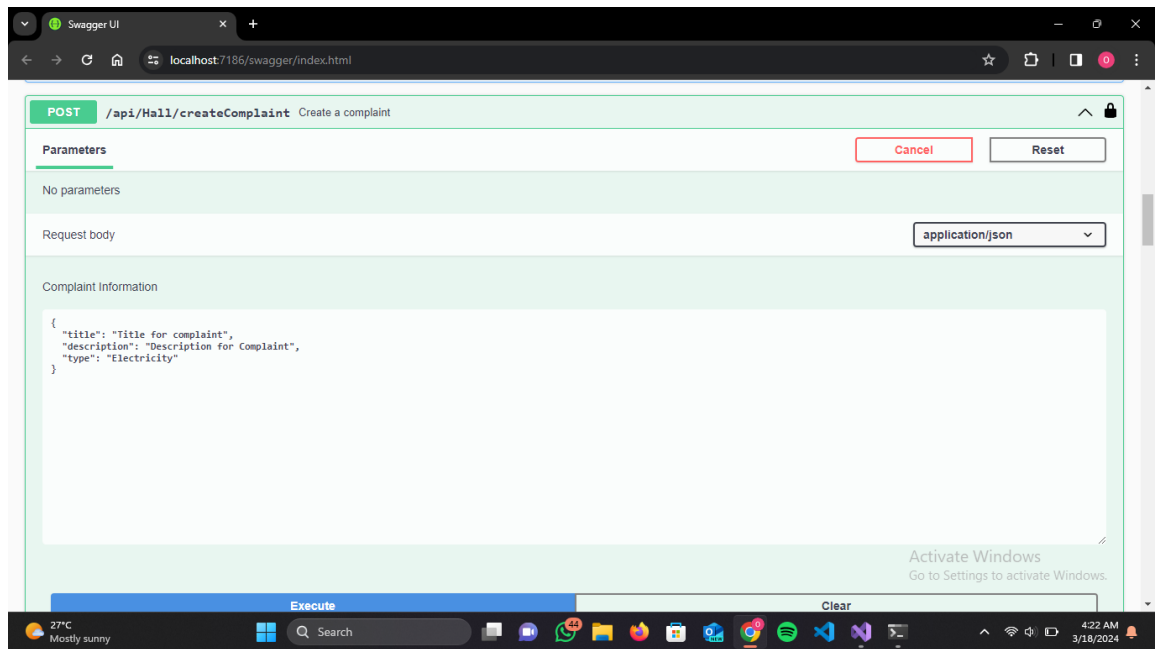


Fig 4. 11. Swagger Documentation showing Endpoint to create a Complaint

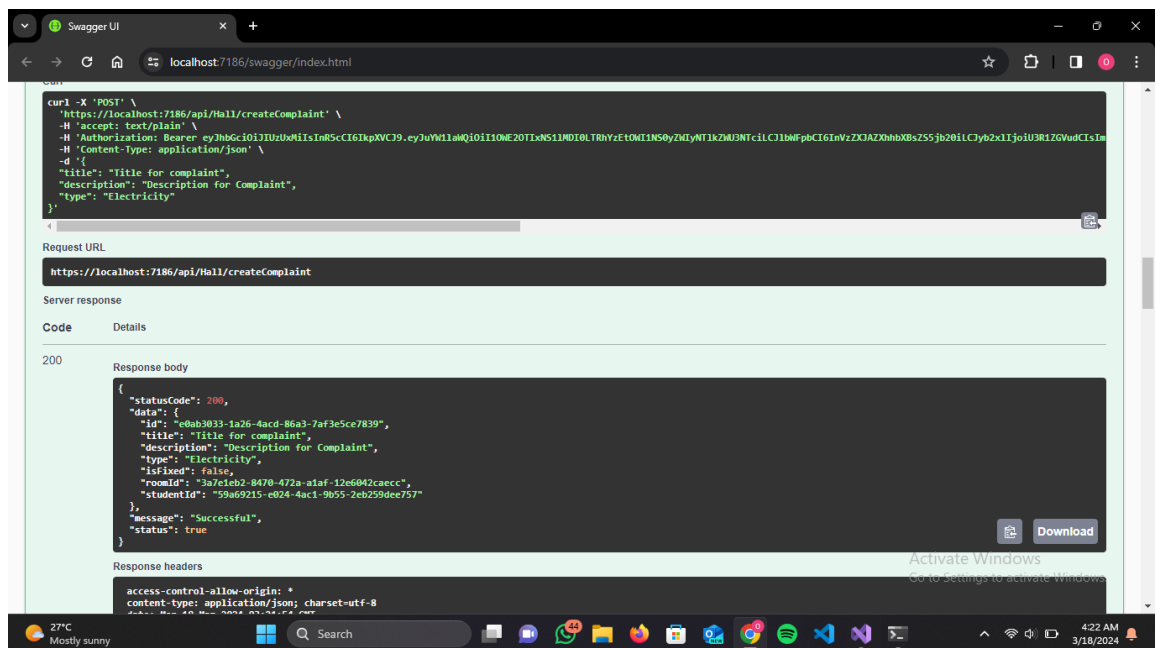


Fig 4. 12. Swagger Documentation Showing Sample Successful Response for a Complaint

In the above diagrams (fig 4.11 and fig. 4.12) shows how to create a complaint by a student who is logged in and shows a sample response expected from the server if the request is successful.



## **4.2. SYSTEM TESTING**

This is an important aspect of software engineering which involves the verification of the integrated system to evaluate how well the designed and implemented system complies with the specified requirements. System Testing is an essential phase in the development of any system. Its primary objective is to verify that the system is functioning as expected and meeting its specification and needs of the users. It is imperative to identify and reduce defects or errors before the system is released and launched to general public. To commence the system testing phase, the first step is to establish the expectations and requirements of the Hostel management System. The Black box testing technique was used. The black box testing was carried out by users who were not aware of the internal structure of the system.

### **4.2.1 UNIT TESTING**

Unit testing is an integral part of a software development process. Unit testing plays a pivotal role in the development of the Hostel Management System (HMS), ensuring that individual components of the software function correctly and reliably. This meticulous process aims to detect any errors during the early stage of development, preventing them from impacting the overall performance of the system. Unit testing is carried out by partitioning the code into smaller units such as individual functions, classes or modules. In the case of our system we can divide them into the User modules, Authentication modules and the Hall Modules. These units are then tested in isolation with test and control data to ensure they are working properly. During testing a series of automated test cases are formulated to validate the correctness of the code across different scenarios and inputs.

#### **4.2.2. INTEGRATION TESTING**

Integration testing involves verifying the interaction between the Hostel Management Systems components to ensure they function seamlessly. The objective is to detect any issue that may arise during the integration of the different parts of the system. The Hostel Management System (HMS) integration testing procedure starts with comprehensive test planning, where the goals and scope are established. Next, the test environment is set up, and test cases are created to cover a range of integration situations. Data consistency checks, error handling mechanism testing, and integration point validation between HMS modules and external interfaces are all done throughout test execution to guarantee system resilience. This is done to make sure the Hostel Management System satisfies quality standards.

## **CHAPTER 5**

### **5.1. SUMMARY**

The hospitality industry, particularly the budget-friendly segment, has witnessed a surge in demand for efficient management solutions. Hostel Management Systems (HMS) have emerged as a pivotal technology addressing this need by automating and streamlining various operational aspects of hostels. This study explores the functionalities, benefits, and impact of HMS on various stakeholders within the hostel ecosystem.

Our Hostel Management Software has been designed to streamline and automate various essential tasks in managing hostels. It offers various functions that help to make the process of Hostel Management easier on School Hostel Admins. This research focuses on enhancing the internal software qualities within the Hostel Management System, with a specific emphasis on software quality.

Thus the developed system built on the robust foundation of C# and MSSQL, offers a seamless user experience and enhanced scalability. This use of Microsoft technologies ensures optimal performance, security, and compatibility with future updates, providing a solid framework for efficient hostel management.

The developed system's accessibility is also an important point to note, as it operates as an online software, allowing users to connect from anywhere with a working internet connection. This flexibility empowers hostel managers and owners to oversee and manage hostel activities remotely, contributing to a more agile and responsive administration.

Our Online Hostel Management System, developed with C# and MSSQL, represents a leap forward in hostel administration technology. By embracing modern programming languages and database management systems, our solution sets a new standard for

efficiency, accessibility, and security in the realm of hostel management, promising a more seamless and responsive experience for administrators and users alike.

## **5.2. CONCLUSION**

The research work investigated the design and implementation of a Hostel Management System (HMS) with the objective of addressing the challenges faced in manual hostel management. The research findings demonstrated that our developed system offers a comprehensive solution for enhancing various hostel operations, enhancing efficiency, and improving the overall experience for both student and School Hall Admins.

The successful implementation of our Hostel Management System (HMS) demonstrates the potential for technology to significantly improve the efficiency and effectiveness of hostel operations. By providing a more streamlined and user-friendly experience for both guests and staff, this research contributes to the advancement of the hostel industry and the broader hospitality sector.

## **5.3. RECOMMENDATION**

Based on the findings of this research we've been able to build a commendable understanding of the challenges faced by schools in hostel management. The use of a strongly typed language such as C# also enhances the scalability and maintenance of the software. Future research efforts could be directed towards confirming and refining user experience and the use of the data gotten to provide analytical techniques to improve on the system

## **5.4. LIMITATIONS TO STUDY**

A significant limitation concerns the difficulty of extrapolating the research results. The study was limited to a particular setting and kind of Hostel, such as a certain kind or size of hostel. Consequently, considering the possible differences in variables like hostel size, location, and organizational structure, extending the results to different hostel settings could prove to be a challenging undertaking.

The industry's adoption of technology in the hostel sector is another important factor. It could be challenging for hostels with little funding or technical infrastructure to accept and use the suggested solution. The degree to which the targeted dormitory industry adopts technology may have an impact on how effective the solution is.

Variations in culture and law could also be obstacles. Regional variations in legal and cultural standards may have an impact on the system's application. To guarantee the system's global relevance and acceptance, future research projects should investigate how well it adapts to other cultural and legal contexts.

In summary, acknowledging and resolving these constraints offers a nuanced viewpoint on the Online Hostel Management System research. Subsequent research endeavors have to concentrate on enhancing the system's functionality, guaranteeing its efficacious integration throughout a wider range of hostel settings, all the while managing the complexities posed by these intrinsic limitations.

## 5.5. REFERENCES

- [1] H. P. C. E. in R.-T. Jha, L and U. 2022, “A comprehensive study of and possible solutions for a hostel management system,” *books.google.com*, 2023,
- [2] B. Choudhury, D. Kumar, D. K.-A. I. J., and U. 2017, “Online Hostel Management System,”
- [3] S. P.-I. J. of S. and Research and undefined 2023, “Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software ,”
- [4] E. KENNETH, E. C.-I. R. And, and U. 2020, “Analysis of Challenges in Managing Students’ Hostel Facilities in Nnamdi Azikiwe University, Awka Anambra State, Nigeria,”
- [5] P. Harina, K. Kavya, S. M.-2022 1st International, and U. 2022, “Hostel Management,” *ieeexplore.ieee.org*, 2022 Available:  
<https://ieeexplore.ieee.org/abstract/document/10040481/>
- [6] S. Bhardwaj, K. V. and Tornalds., and U. 2022, “Hybrid Technology Based Smart Hostel Management System Using Artificial Intelligence and Internet of Things,” *ieeexplore.ieee.org*, 2022, Available:  
<https://ieeexplore.ieee.org/abstract/document/10059715/>
- [7] M. Lakshaga Jyothi and R. S. Shanmugasundaram, “Applications of IoT Technology and Computer Vision in Higher Education–A Literature Review,” *scholar.archive.org*, vol. 05, pp. 2454–9150, 2019, doi: 10.35291/2454-9150.2019.0287.

- [8] M. ARANSIOLA, "AN AUTOMATED HOSTEL MANAGEMENT SYSTEM," 2019, Available: <http://ir.mtu.edu.ng/xmlui/handle/123456789/138>
- [9] V. OLAWUYI, "AN ONLINE HOSTEL MANAGEMENT SYSTEM," 2020. [Online]. Available: <http://ir.mtu.edu.ng/xmlui/handle/123456789/165>
- [10] S. O'Hanlon, "Home together, home apart: boarding house, hostel and flat life in Melbourne, c1900-1940," 1999, Available: [https://scholar.archive.org/work/prcgqgtrabfvxh4gbgiaceakl4/access/wayback/http://aueast.erc.monash.edu.au/fpfiles/16726283/5446090\\_monash\\_6164\\_SOURCE2\\_4.pdf](https://scholar.archive.org/work/prcgqgtrabfvxh4gbgiaceakl4/access/wayback/http://aueast.erc.monash.edu.au/fpfiles/16726283/5446090_monash_6164_SOURCE2_4.pdf)
- [11] A. Mishra, *Students and the hostel life: A study of university students*. 1994. Available: [https://books.google.com/books?hl=en&lr=&id=PzWMCFz1TZYC&oi=fnd&pg=PR6&dq=Students+in+hostel+not+only+learn+the+theoretical+material+they+also+learn+how+to+enhance+their+personal+abilities+and+learn+to+live+independently+\(Mishra,+1994\)&ots=bfTvvhxhDKf&sig=vMaZpzoRB2RmdIBLLqsvdMxdHg](https://books.google.com/books?hl=en&lr=&id=PzWMCFz1TZYC&oi=fnd&pg=PR6&dq=Students+in+hostel+not+only+learn+the+theoretical+material+they+also+learn+how+to+enhance+their+personal+abilities+and+learn+to+live+independently+(Mishra,+1994)&ots=bfTvvhxhDKf&sig=vMaZpzoRB2RmdIBLLqsvdMxdHg)
- [12] I. ALi, "Hostel Management System." 2021. Available: [https://www.academia.edu/35675809/Hostel\\_Management\\_System](https://www.academia.edu/35675809/Hostel_Management_System)
- [13] K. Ayanlowo, O. Shoewu, S. O.-J. of S. and, and U. 2019, "Development of an automated hostel facility management system," *Acad. Ayanlowo, O Shoewu, SO Olatinwo, OO Omi. DD BabalolaJournal Sci. Eng. 2014*•*academia.edu*, 2014, Available: [https://www.academia.edu/download/53324189/Development\\_of\\_an\\_Automated\\_Hostel\\_Facil.pdf](https://www.academia.edu/download/53324189/Development_of_an_Automated_Hostel_Facil.pdf)

- [14] Qorez, "Online Hostel Management System," 2015, [Online]. Available: <http://ir-library.mmarau.ac.ke:8080/handle/123456789/2059>
- [15] S. Agrawal, S. Rastogi, S. T.-I. C. on, and U. 2023, "Cloud Based Hostel Facility Automation System," *ieeexplore.ieee.org*, 2019, [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10140316/>
- [16] T. B. Emmanuel and M. N. Lameck, "Design And Development Of An Automated Student Information System For Queens' Academy Private School In Kabwe Zambia," *multiresearch.net*, 2019, [Online]. Available: <https://www.multiresearch.net/cms/publications/CFP14192019.pdf>
- [17] Y. Pandey, N. S.-I. Online, and U. 2021, "Student satisfaction towards the virtual learning in Higher education Sector.," *search.ebscohost.com*, 2021. Available: <https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=13053515&AN=151374899&h=9RxyswkL3eOuZeHiVNhdFYjkSbFZRkYHMXLaJlkpgi6HWkO8hWPSnlZeMasD1uL%2BsaRpHHh0hqfycJ1aW9JOUg%3D%3D&crl=c>
- [18] R. Kocielnik, S. Amershi, P. B. the 2019 C. C. on, and undefined 2019, "Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems," *dl.acm.org*, Available: <https://dl.acm.org/doi/abs/10.1145/3290605.3300641>
- [19] P. Wong, H. M.-A. I. T. And, and U. 2022, "Design and Development of Online Hostel Management System to Improve Application and Management Process," *publisher.uthm.edu.my*, 2022, [Online]. Available: <https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/7496>