A QR-CODE-BASED IDENTITY MANAGEMENT SYSTEM FOR MONITORING HOSTEL RESIDENTS

By

OLAIYA, OREOLUWA ONYEKACHI

17010301046

A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES, IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

2021

DECLARATION

I hereby declare that this project has been written by me and is a record of my own research work. It has not been presented in any previous application for a higher degree of this or any other University. All citations and sources of information are clearly acknowledged by means of reference.

OLAIYA, OREOLUWA ONYEKACHI

Date

CERTIFICATION

This is to certify that the content of this project entitled '**A QR-Code-Based Identity Management System for Monitoring Hostel Residents'** was prepared and submitted by **OLAIYA OREOLUWA ONYEKACHI** in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE.** The original research work was carried out by her under by supervision and is hereby accepted.

______________________________ (Signature and Date)

Mr. J. A. Balogun
Supervisor

______________________________ (Signature and Date)

Dr. M.O. Adewole
Coordinator, Department of Computer Science and Mathematics

DEDICATION

I would like to dedicate this project to God Almighty, for being faithful and merciful, for seeing me through to the end of this project. I also dedicate this work to my father, Mr. Agboola Olaiya and my mother, Mrs. Anna Kelechi, Olaiya for being a major source of support in every way. I finally dedicate this work to Adeoti Faith who has been supportive, encouraging and present during the whole process of delivering this work.

ACKNOWLEDGEMENTS

My sincere gratitude goes to the God who created all things and manifests himself in diverse ways than we can comprehend for his mercy, loving-kindness and presence in the times I was in need. I also appreciate the entire staff and management of Mountain Top University for the immeasurable impact they have had on my life academically and spiritually from the person of the chancellor, Dr. D.K Olukoya down to every member of the senate and the university scholarship board. I specially recognize my supervisor, Mr. J.A Balogun for believing in me and pushing me to do more. I also recognize professor I.O Akinyemi, Dr. M.O Adewole, Dr. Funmilayo Kasali, Mr. I.O Ebo, Late Dr. M.O Oyetunji and a host of other lecturers who have made special contributions to the success of my academic pursuit, as well my dear friend Dev_Wilson. Finally, I duly appreciate my siblings, friends and colleagues without whom this journey could not have been a success. May the God of heaven water your lives.

ABSTRACT

The aim of this study was to develop an information system required for managing the identity of residents of students hostels with the purpose of monitoring movements into and out of their various hostels. The study identified the various user and system requirements, specified the system design and implemented the system.

A review of the literature was being done to identify and understand existing Identity Management Systems for monitoring hostel residences using QR codes. The user and system requirements of the system were identified from system users using informal interviews. The system design was specified using UML diagrams, such are use case, sequence and class diagram. The database was implemented using Firebase. The implementation of the frontend was done using HTML, CSS, and Bootstrap. The backend was implemented using Node JS and Express JS.

The results of the system showed the implementation of the system's database for storing the information alongside the front-end of the web and mobile application. The results revealed that the system was able to uniquely identify each hostel residents using a uniquely generated QR code with which the movement of the students in and out of the hostels was easily monitored.

The study concluded that using the system within the students' hostel will help mitigate most of the challenges associated with monitoring the student's movement in and out of hostels especially with event of accounting for the number of students present in the hostel during head counts.

Keywords: *QR Code, Identity and Access control, Identity management, Information system, Hostel management.*

TABLE OF CONTENTS

CHAPTER FOUR: IMPLEMENTATION AND RESULT

CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATION

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

The topic of identification and access control management has been a topic of interest in organizational and institutional environments. One of such environments is the hostel environment in higher institutions. The need for hostels in the first place stemmed from the fact that education is no longer limited to one's geographical border. From this, the issue of security is birthed and institutions housing students are charged with the responsibility of monitoring their ins and outs from the hostel. For an institution to take the action of monitoring students, the institution should be able to identify each person within the hostel and identify their roles. Identity can be said to be a characteristic that marks off an individual from the rest of the same kind. This kind of characteristic encourages the recognition and identification of an individual or individuals.

Identity Management (IdM) is described as the process of managing individual identities, authentication and authorization, roles, and privileges within or across system boundaries to increase productivity while eliminating redundancy. There are several available Identity and access control management systems that can be used to suit the purpose of this study some of which are: biometrics (face, finger, DNA, etc.), RFID, sensor tags, QR code. QR codes are types of matrix barcodes or two-dimensional barcodes. It is a machine-readable label that contains related information about the items, website, or persons they are attached to (Muthukumar, Albert Mayan, Gokul, & Denil, 2019). This study aims at making use of QR codes for the implementation of its identity and access control system for reasons which include:

they are inexpensive compared to biometric systems, they can be used for identification as user information is encoded into it.

1.2 Statement of the Problem

The daily activity of monitoring students within and out of the hostel is manually managed by the hostel officers (porters). The registration of students as well as allocating them into hostels, all these activities are being carried out manually by the Registration officer. One pertinent issue of the present system is the issue of monitoring if all students are present in the hostel at the end of each day. The hostel officers (porters) are made to move through each room on every floor to count and identify the students in their allocated rooms. Thus, these activities can be seen as redundant and can essentially be avoided. The aforementioned responsibilities of hostel officers can pose a strain on the individuals who are in charge of running the hostels as there is no software (web and mobile) being utilized in this specific circumstance.

1.3 Aim and Objectives of the Study

The aim of this study is to develop an Identity and Access Management System that would allow for ease of monitoring, logging student's whereabouts by house porters, parents, and school management.

The specific objectives are to

i. identify the requirements of the system;

ii. specify the design of the system; and

iii. Implement the system.

1.4 Methodology of the Study

To fully accomplish the aforementioned objectives, the following methods were adopted.

a. A review of the literature was being done to identify and understand existing Identity Management Systems for monitoring hostel residences using QR codes.

b. The user and system requirements of the system were identified from system users using informal interviews.

c. The system design was specified using UML diagrams, such are use case, sequence, class diagram, and so on.

d. The database was implemented using Firebase.

e. The implementation of the frontend was done using HTML, CSS, Javascript while the backend was implemented using Node JS and Express JS.

1.5 Significance of the Study

This study was brought about by the rising need for proper identity and access control management systems in the higher education system. This system would improve to a large extent the productivity of all significant parties highlighted in this study. Such matters include: alleviating all forms of redundancy, curbing the use of paper-based recording system, reducing all forms of human errors and ease the stress associated with individuals running the hostel.

1.6 Scope and Limitation of the Study

The scope of this study was centered on the development of an Identity Management System for Hostel Residents. The study proposes to develop a web-based application that would be used to manage registration, generation of the QR code, and monitoring and logging of day-to-day activities in the hostel. A mobile application that will be used to scan the generated QR code for each student will also be developed during the study.

The limitations of this study include:

a. Developing a system that can handle multiple hostels in an institution and not multiple institutions; i.e., the system would vary based on each institution's need.

b. This system did not go into specific areas like room allocation, access control into each room, and other hostel management features. This system will not allocate rooms to students as that is part of a larger system.

1.7 Definition of Terms

a. **Bootstrap** is a potent front-end framework used to create modern websites and web apps. It is open-source and free to use yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms.

b. **CSS (Cascading Style Sheet)** is the language for describing the presentation of web pages including colors, layouts, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, and printers

c. **Express js** is a backend web application framework for Node.js. it is designed for building web applications and APIs.

d. **Hostel Officer (porter)** is a person whose job is to be in charge of the day-to-day activities involved in the hostel.

e. **HTML (Hypertext Markup Language)** is a language that allows users to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications.

f. **Node. Js** is a platform built on chrome's JavaScript runtime for easily building fast and scalable network applications. Node. js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

g. **Registration officer** is responsible for admitting people into an organization. *Database* A systematically arranged collection of computer data, structured so that it can be automatically retrieved or manipulated. It is also called Databank.

h. **Student** is a person enrolled in a school or other educational institution and who is under learning with goals of acquiring knowledge, developing professions, and achieving employment in their desired field.

i. **System** *is* a combination or arrangement of parts to form an integrated whole. A system includes an orderly arrangement according to some common principles or rules.

CHAPTER TWO

LITERATURE REVIEW

2.1 Information System

Information can be described as processed data. An elementary definition of information would be raw data that has been processed to convey a meaning. Information is data that has been refined to the point where it can be used for analysis. Compared to data, information is structured, explicit, with form and meaning. It can be related to any context based on the way it is being analyzed. Information is a meaning a person can express or extract from a representation of facts or an idea and does this by interpreting known standards of conventions applicable to the area of study in question (Zoikoczy, 1981).

The idea of information in an organizational context is more complicated and challenging than its widespread usage suggests. Without a doubt, every society is an information society, and every organization is an information organization. As a result, information, like other commodities is a basic resource and can be viewed as a concrete or abstract notion. According to Stonecash (1981), information is simply symbolling such as: data, text, images, voices, etc. that convey meaning through their relative ordering, timing, shape, and context. Information is the raw material for making decisions for creating knowledge and fueling the modern organization.

The basic concept of an Information system is a system to convert data from internal and external sources to information and communicate that information in an appropriate form (O'brien, 2004). For any organization to survive, it requires information; this information is a necessary tool or resource which can be derived from various information systems to aid the organization in management and decision making. The information system is made up of nodes that can store data, channels that can distribute data, and actors who act and react in response to the data. As a result, rather than being users of the information system, the actors - the individuals within the company - are members of it. There is no such thing as an information system without people, according to this notion.

Therefore, an appropriate definition of information system would be a system where a collection of people, procedures, a base of data and (sometimes) hardware and software that collects, processes, stores and communicates data for transaction processing at the operational level and information to support Management decision making (Duff & Asad, 1980). Supporting the previous definition, Gary, Shelly, Cashman, & Vermaat (1999), defined information system in their report as, a collection of hardware, software, data, people, and procedures that are designed to generate data that supports users' day-to-day, short-term, and long-term activities in an organization.

Systems can be divided into two categories: functional and structural. From a functional standpoint, an information system is a technological implementation of a medium used for recording and storing data. From a structural standpoint, an information system entails the participation of a group of people, processes, data, and technology, all of which come together to build a unified structure that meets the organization's goals.

According to Adeoti-Adekeye (1997), an information system takes data as raw material and turns it into information through one or more transmutation processes. He also stated that it is made up of the following functional elements that have to do with the organization and its surroundings:

a. perception – the initial entrance of data into the organization, whether recorded or generated;

b. data recording – the act of physically capturing data;

c. processing – alteration following the "specified" organization's requirements;

d. transmission – information travels from one location to another in an information system;

e. storage – the process of storing data for presupposes or a potential future usage;

f. retrieval – search for data that has been recorded and saved information;

g. presentation – the process of communicating data in a way in which it could be understood;

h. decision making – information system makes decisions that concern itself by providing the required information to managers at all levels.

The objective of information systems is to provide information at all levels of management at the most appropriate time, accurately and at an economical cost, using this information to take appropriate actions to modify the state of the system (Yusuf, 2014).

Information management is described as an organization's capacity to handle information. Creating, preserving, retrieving, and making available the appropriate information as quickly as possible, that is information, at the right moment, in the hands of the right people, at the right location for use in decision-making at the lowest cost, in the finest medium (Langemo, 1980). Information management, according to Best (1988), is the cost-effective, efficient, and effective coordination of the creation, control, storage, retrieval, and distribution of data from external and internal sources in order to improve the organization's performance. The definition of management as used in this context, refers to a way of controlling the structure in which information is processed and delivered.

The ability of an organization to gather information systematically and the responsible use of that information in adding value to that organization by giving access to the right individuals, can be termed as information management. Information can be gathered via various means; the internet, records, files and so on. As a result, information management encompasses all systems and processes that support information management programs, such as web content management, document management, records management, digital asset management, learning management systems, and enterprise search (the technical infrastructure) (Reddy, Srinivasu, Rikkula, & Rao, 2009).

Information management has the objective of processing recorded information and distributing them to all users as efficiently as possible to ensure that the value of information is identified and exploited to achieve the fullest benefits. To be able to achieve these objectives, information needs to be managed like all other resources. According to Robek, Brown, and Stephens (1995), the objectives of information management is:

a. To provide accurate, timely, and full information for successful decision-making in an organization's management and operations.
b. To give information and records for the least amount of money possible.
c. To provide the best possible service to the records' users and
d. To get rid of records that are no longer useful to the company.

2.1.1 History of information system

Although the term information system has its root in various areas of disciplines, its advent can be attributed to the development of the first computer, in the 1960s. This led to various organizations adopting the idea of having IS groups and departments because they recognized that there was a need to bring together a range of systems each utilizing hardware and software that are incompatible. The field of information systems has progressed significantly since its inception.

At the time of its inception, information systems were only thought to be useful for electronic data processing (EDP), which powered simple data processing tasks like record-keeping, accounting, and transaction processing. They were known by a variety of names at the time, including automatic data processing systems, transactions processing systems, an information processing system. The addition of the role of data processing into useful reports to computers in the 1960s necessitated the development of business applications that would take advantage of this new role and provide managerial end users with a set of predefined reports based on available data resources, heralding the birth of the Management Information System (MIS). With pre-specified management reports such as sales analysis, cost, and production trend reporting systems, this MIS quickly proved insufficient to serve the decision-making needs of companies, giving rise to the notion of decision support systems.

Executive Information Systems (EIS) was created to address the aforementioned need by combining important data from MIS reports with DSS analytical models that are specially tailored to satisfy the information needs of organizational leaders. The EIS was quickly improved on after advancements in artificial intelligence methods in business allowing most information systems to be able to execute their core duties with little or no human interaction. This new breed of knowledge-based information systems was termed Expert Systems and was accepted by many businesses as advisors in restricted topic areas.

2.1.2 Types of information systems

In reality, there are several types of information systems that apply to specific business needs or business types. Majorly, there are six (6) types of information systems, each supporting a specific organizational level. These systems include:

a. Transaction Processing Systems (TPS)

The core business systems that serve at the operational level of the company are transaction processing systems (TPS). Any event or activity that has an impact on the organization is referred to as a transaction. Transactions may differ from one company to the next depending on the nature of the business. Therefore, it is easy the deduce that the transactions of each organization would differ from the other (Laudon & Laudon, 2006). Transaction processing systems support the day-to-day activities of the business. This type of information system goes through five stages: data entry, data processing, database maintenance, document and report generation, inquiry processing.

b. Office Automation Systems (OAS)

This system can also be described as an Enterprise Collaboration System. An office automation system (OAS) is a grouping of communication technologies, computers, and people who work together to complete official activities. It manages office transactions and assists with official activities at all levels of the organization. It is one of the most extensively utilized forms of information systems for assisting managers in controlling the flow of information in enterprises (Heidarkhani, Khomami, Jahanbazi, & Alipoor, 2013). This type of information system is not specific to one level in an organization, as it provides important support to each user and also enhances communication and productivity. Applications like word processing, e-mails, video conferencing are a part of office automation systems.

c. Knowledge Work Systems (KWS)

A knowledge work system (KWS) is a specialized system designed to promote the innovation process and ensure that knowledge and technical skills are properly integrated into business operations. It provides graphics, analytical, communications, and document management tools to intelligence workers to assist them in creating and disseminating new information and insight. According to O'Brien and Marakas (2011), knowledge management systems are knowledge-based information systems that help employees and managers throughout a company create, organize, and disseminate business knowledge. Some examples of knowledge work systems are computer-aided design systems (CADs), virtual reality systems, and financial workstations.

d. Management Information Systems (MIS)

Management information systems, according to Hasan, Shamsuddin, and Aziati (2013), are information systems that take internal data from a system and summarize it into meaningful and useful forms such as management reports, which are then used to support management activities and decision making. A management information system (MIS) extracts transaction data from underlying TPSs, compiles it, and generates reports, displays, or responses as information products. These information products provide information that is tailored to managers' and supervisors' decision-making needs. Simple routines such as summaries and comparisons are used by management information systems to enable managers to make decisions for which the procedure for arriving at a solution has been specified in advance.

e. Decision Support System (DSS)

Decision Support Systems are a type of computerized organizational information system that assists managers in making decisions that require modeling, formulation, calculation, comparison, and selection of the best option or prediction of scenarios (Heidarkhani, Khomami, Jahanbazi, & Alipoor, 2013). A decision support system is a computer-based interactive information system that, like MIS, is used at the management level of a company. In contrast to MIS, it processes data to assist managers in making decisions. When compared to other information systems, decision support systems have more analytical power. They use a variety of decision models to analyze data or summarize large amounts of data into a format (usually tables or charts) that allows managers to compare and analyze data more easily. They provide an interactive environment in which users can work directly with them, add or change data as needed, and ask new questions.

f. Executive Support Systems (ESS)

An executive support system is a computer-based information system that aids decision-making at the top level of an organization. It is a subset of MIS. ESSs have more general computing capabilities, better telecommunications, and more efficient display options than DSSs. They display critical information in the form of charts or graphs that assist senior executives in solving a variety of problems using advanced graphics software. They use summarized internal data from MIS and DSS, as well as data from external sources about events such as new tax laws, new competitors, and so on, to make effective decisions. They filter, compress and track critical data before making it available to senior executives.

2.2 Identity Management System

The topic of identity is a concept that cannot be consistently defined, as it spans over a wide range of other fields or areas of applications which include: psychology, social psychology, philosophy, social anthropology, and so on. However, concerning the topic of this study, identity can be said to be the characteristic that marks off an individual from the rest of the same kind. Vishwas (2020), described the need for identity to originate from high growth in the number of users, which certainly would lead to service providers requesting for users' identity and users requesting for the authenticity of the service providers. In the area of technology, the need for identification began when it was noticed that there was an increasing number of internet users and service providers cannot allow users to use their systems without being able to identify them and users wanting to know the authenticity of the service providers (Vishwas, 2020).

Identity describes an arrangement of unique qualities or attributes that differentiates one individual from another. Typically, those attributes are derived from demographic characteristics such as place of birth, physical appearance, and an array of social qualifiers, such as place of residence, occupation, and so on (Mburu, Nderu, & Tobias, 2019). According to Rosencrance & Mathias (2020), identity management is the organizational process of ensuring that individuals have the appropriate access to technology resources. The idea of identity management includes the action of identifying an individual, authentication, and authorization of persons, describing individual roles and privileges. The goal of identity management is to guarantee that only authenticated users are granted access to the specific applications, systems, or environments for which they have been authorized.

Identity management (IdM) can be associated with storing user data, maintaining user accounts, and controlling users' access to applications (Kunz, Hummer, Fuchs, Netter, & Pernul, 2014). Technologies used for identity management (IdM) include password management tools, two-factor and multi-factor authentication (Beal, 2021), biometrics (face recognition, fingerprint), barcodes, QR codes, and so on. Beal (2021), highlighted the following as components or requirements of Identity management. They include capture and record of user login information, managing databases of user identities, arrangement of assignment and removal of access privileges, provide a central directory, managing digital identities of devices and applications, restrict access to subsets of data based on specific roles.

Okta (2020), defined Identity Management Systems as information systems or a set of technologies that make it possible to manage individual identities, either within an enterprise or across cross-networks. Identity management is a service that was introduced to facilitate the online management of users and their identities. These services require that user present their identities for authentication to give them access. Different research efforts on identity management, have resulted in the creation of various Identity Management System models and prototypes, each of which met its own set of specifications (Ferdous & Poet, 2012).

These various levels of identity management differ from each other by their architecture, implementation, and functionality. These models and/or prototypes include the silo model which gave users the capability to manage as many credentials as service providers, but this would require a user to memories a large number of logins and passwords (Vishwas, 2020) after the silo model came to the centralized digital identity system, which introduced the identity provider (IDP) which would be the issuer and verifier of user identity. This type of system is further divided into federated digital identity systems, user-centric digital identity systems (Mburu, Nderu, & Tobias, 2019).

The most recent model is the self-sovereign identity (SSI), which aims to return complete ownership of the user's identity to the user while also adding a layer of protection and versatility by allowing the identity holder to disclose only the data required to any service provider. Unlike other systems, it allows users and organizations to communicate securely by removing trust from central authorities such as IDP (Vishwas, 2020). Desmond (2005), highlighted that other than users, there exist other network entities, such as computers, processes, and applications, that may be managed by an identity management system.

2.2.1 Methods of identification

There are various technologies through which the identity of users are managed and there is an existence of various access management technologies that allow or restrict authorized and unauthorized users. These technologies include:

a. Password-based authentication

Passwords are the most used method of authentication. A password can be made up of letters, numbers, or special characters. To protect yourself, you must create strong passwords that include a combination of all possible options. Users cannot remember all of their passwords; therefore, they resort to using easy passwords that they can remember. As a result, passwords have numerous flaws and are insufficient to protect the person or systems from attackers.

b. Multi-factor authentication

This authentication method necessitates the use of two or more distinct methods to identify a person. Mobile phone use, Captcha tests, and so forth are examples of these. This strategy adds numerous layers of security and boosts user confidence.

c. Certificate-based authentication

This technology uses digital certificates to identify the user and the system. Digital certificates are electronic documents that contain a user's digital identity and demonstrate that they own a public key issued by a certification authority.

d. Biometric authentication

A biometric authentication is a form of security that is based on an individual's biological traits. Among the several authentication methods, this technique is unquestionably the most prevalent. It compares user's features to the ones stored in the database to authenticate them. It can be used on gates and doors to manage physical access. Common biometric authentication methods include facial recognition, fingerprint scanners, voice identification, eye scanners.

***e.* Token-based authentication**

This authentication method enables users to input their credentials and receive an encrypted string of random characters. Users can use this token to access systems that are protected. An example of this kind of system is the barcode and QR code. They are matrix codes that can contain encrypted information about a person, an item, and so on.

2.2.2 QR codes

QR code which stands for Quick Response Code was first developed in 1994 by a Japanese company, a subsidiary of Toyota called DensoWave (Rouillard, 2008). QR code is a machine-readable two-dimensional barcode technology that is usually attached to items or persons-based information provided about the item or person (Ramdav & Harinarain, 2018). It was used to assist in tracking automobile parts throughout production (Narayanan Sankara, 2012). According to Wave (2016), QR codes were created to enable high decoding speeds and owe their existence to the development and popularity of barcodes. In the early 60s, the use of barcodes was adopted to assist cashiers, from having to manually key in prices. The limitation of barcodes generated from the fact that barcodes could only hold 20 alphanumeric characters of information (Ramdav & Harinarain, 2018). Simply, the basic difference between the barcode and QR code is that barcodes are one-dimensional codes that can only hold 20 alphanumeric characters, while the QR codes are two-dimensional matrix barcodes that can store information in 7,089 number characters, 4,296 alphanumeric characters, and 1,817 kanji characters (Narayanan Sankara, 2012).

In recent times, the use of mobile devices has presented an efficiency in performing basic daily activities, even with the use of complex technology. From making payments to promoting a business, most of these activities are performed mostly on mobile devices. From there, QR codes have received their popularity. Small organizations use this technology for easy payment, delivery, verification, and so on (Bond, 2021). The general idea of this technology is the transfer of information that has been encoded, decryption of encrypted codes, a symbol that is easily interpreted by scanner equipment (Petrova, Romaniello, Dawn Medlin, & Vannoy, 2016). QR codes have been made available as both software and hardware and they can be scanned with mobile phones or similar devices for that purpose.

2.3 Hostel Management System

The term hostel can be attributed to many definitions depending on the context from which it is being defined. For this study, a school hostel can be defined as a resident environment for students of either secondary or tertiary institutions, which offers shared accommodation to single-sex (Volunteer Forever, 2019). These kinds of accommodations are usually organized and managed by individuals within an institution, hence, rules such as curfews are applied and strictly carried out. Hostels can be said to provide an inexpensive form of accommodation for students, as well as travelers (Appiah, 2016).

From the perspective of Suki and Chowdhury (2015), a hostel is an accommodation that provides the essential needs of students and made is available primarily for those who live far from the educational institution. School hostels are built with institutional or formal characteristics that make provisions for basic amenities for students such as, room allocation, internet connections, and areas for recreational activities (Khozaei, Ayub, Hassan, & Khozaei, 2010). Adebisi, Oletubo, Alade, & Ekpekpe (2017), described hostel as a support facility used to increase instructional effectiveness, improve cleanness, orderliness, and safety, and increase the efficiency and effectiveness of students in the learning environment, they were referring to it as a direct-action element in the learning environment of the education sector.

While the aforementioned concepts offer a variety of perspectives ranging from low-cost housing to education and support services, they all support the idea that a hostel is a location where students are accommodated to pursue an education. The management of the activities, resources, and processes involved within the hostel is done by various individuals, either manually or automatically through various management systems. The process of coordinating all the processes within a hostel environment is called hostel management (Ayanlowo A., Shoewu, Olatinwo, Omitola, & Babalola, 2014). The main focus of any school hostel is the optimal performance of students, given a conducive environment; therefore, hostel management is the university's administrative plan to organize, control the resources, and most significantly to arrange the tools needed to fulfill the student's yearning needs while still maintaining and improving the academic success (Ohaeri & Omorojor, 2020).

In addition, hostel management integrates other related facilities such as water supply, toilet facilities, reading areas, bed and sometimes beddings, along with affordable accommodation (Ezeigweneme & Egolum, 2020). For every hostel management process, there has to be a system, whether manual or automated that must be employed to properly carry out each activity. The resolution on a specific method is dependent on the size or the institution implementing it as well as the constraints of implementation.

Ayanlowo, Shoewu, Olatinwo, Omitola, and Babalola (2014), made mention in a publication that the manual method of managing and administering hostels in institutions would be challenging for the following reasons, which include: difficulties in record management, difficulty in tracking the history of a facility, registration for hostel allocation is done manually, and the whole exercise is time-consuming and a waste of human and material resources. Ayanlowo, Shoewu, Olatinwo, Omitola, and Babalola (2014), defined a hostel management system or software as a user-friendly application suite that automates, coordinates, and oversees all aspects of hostel facility administration.

2.4 Software Development Life Cycle (SDLC)

In the world of systems development, one is not expected to pick up a project without painting a picture of what the expected result should be. The project manager heading a team of developers is tasked with the responsibility of not only highlighting the roles of individual team members but also establish a relationship with the user of the system, to understand and manage their expectations of the system. This meant that they had to apply a reliable software methodology, known as SDLC. The Software Development Life Cycle (SDLC) is a framework that outlines the activities that must be completed at each stage of the software development process. (Khan, Shadab, & Khan, 2020). Software Development Life Cycle is the collection of steps that are to follow for the systematic development, design, and maintenance of the software projects and ensure that all the user requirements are fulfilled with the least amount of resource consumption (Barjtya, Sharma, & Rani, 2017).

SDLC is used majorly in several engineering fields, and industrial fields as a framework to help manage, plan, and control the process of developing a system. It is a continuous process that starts from the moment a decision has been made to launch a project, to when the system has met the user requirements and is ready to be deployed. In that context, project managers or software team leads are frequently seen using this methodology as a guide to ensure the development and design of a modern, high-quality system that meets and exceeds the customers' expectations while remaining cost-conscious and delivering it within the originally stated time frame. As a result, we can conclude that the team's productivity and the quality of the software are both dependent on the efficiency with which software metrics are defined and analyzed throughout the SDLC.

Ergasheva and Kruglov (2007), defined Software Development Life Cycle (SDLC) as a time required for activities that include defining, developing, testing, delivering, operating, and maintaining the software or the system. They described the software development process as a metric of evaluation used to review the accordance of the system being developed to the requirements that were initially stated. Of course, the metrics to be used are based on the company’s preference.

2.4.1 Software development life cycle processes

The Software development life cycle process can be divided into several phases. These phases, however, can be summarised into five (5) essential processes involved in system development. They include Requirement analysis, System Design, Code/Implementation, System Testing, and Deployment and maintenance.

a. Requirement Analysis/Specification

This is the first phase of the Software Development Life Cycle. The objective here is to know and to document accordingly, the user requirement for the system. This phase is one of the most essential phases in SDLC and this is because, it is necessary to know what is expected from the system to be developed before implementation; it helps the project manager have a sense of the expectation of the system, as well as manage those expectations realistically. It ensures that everyone involved in the development of the system understands the scope of the work to be done and how each of the requirements is to be fulfilled. The Requirement Analysis phase in SDLC can be said to be similar to the Requirement Engineering in Software Engineering.

Requirement Engineering is a process of establishing the services that a customer needs from a system, and the limitations to its operations (Adetoba & Ogundele, 2018). The stages involved in both terms are similar and include, defining a problem, carrying out a feasibility study, and carrying of system analysis by collecting data from a variety of sources (questionnaire, interviews, forms, etc). The goal of this phase is to create the test and evaluation requirements that will be used to decide what level of the system's performance is acceptable. The output of the requirement analysis is called the Software Requirement Specification (SRS), a detailed description of how the system to be developed would behave.

b. System Design

This next stage in the Development Life Cycle process is where the developers of the system, or the system architects, develop a high-level design of the system following the stated requirements. The specifics of the design are presented to the system stakeholders, alongside the possible drawbacks that may occur (budget, risks, time constraints) and each of these parameters is reviewed to determine the best design approach to solve the problem. The requirements collected in the SRS document are utilized as input in this phase, and the software architecture that will be used to accomplish system development is generated (Software Testing Help, 2021). This phase of the SDLC describes in detail the specifications, features, and processes that will be required to meet the functional requirements of the proposed system.

In the design phase, there are two stages: the basic design stage and the structured or detailed design stage. The system's specifications and features are part of the basic design stage. It involves a cost analysis to evaluate whether the system is viable before moving on to the next stage of the design process. The structured/detailed design stage is a blueprint for the analysis document's components, compositions, and inter-relationships. The design is then documented, including references to the defined criteria, and presented to the management team and user for evaluation (Sethi & Sharma, 2013). The output of the design phase is a Software Design Document (SDD).

c. System Development (Code) / Implementation

Once the system design has been developed and approved by the system stakeholders, the step in the development life cycle is to develop the actual system. The output of the previous phase is taken as input to implement the system. The programmers start writing a source code or reusing components of the previous system that are still relevant to the system all while keeping in mind what the client asked for. The team is broken up; each member playing their role, the front-end developer implements the user interface where the client interacts with the system, while the back-end developer implements the logic that links that interface to the server, and to the database (depending on the type of system being developed) (Bindia, 2019). In the case where the system to be developed is small, a single developer does all the work. But in the case of a large body of work, tasks are given to different individuals with the appropriate skill levels to perform. It is usually safer to track the progress of each team member in comparison to others to ensure the individual modules would eventually be compatible with one another. The output of this phase is the program code.

d. System Testing

Each component of the system is brought together to form a single system after which it is deployed into the testing environment before it is delivered to the client. Other than testing for the functionality of the system and for bugs, testing is done to ensure that the system meets the requirements stated the first phase of software process, it also checks if the programmers met the template of the design that was approved in the second phase, and overall, it verifies that the system meets the user requirements and validates that it suits the purpose for which it was made. After developing the program code, a test plan is created and executed on a computer system for systems.

The system is considered part of the implementation process if the output of the test run matches the necessary results. Black box and white box testing are the two types of testing methodologies used to evaluate software systems. In the event of a big system that is separated into multiple components and assigned to different people, those people must do unit testing on the components they have been given. The white box test is used in unit testing. It is a method for determining if particular pieces of code, such as a collection of one or more computer program modules, as well as related control data, use processes, and operating procedures, are suitable for use (Kolawa & Huizinga, 2007). When each component of the system has been thoroughly created and tested, it is then integrated with the others, and the system is evaluated to determine its usability and performance.

e. Deployment and Maintenance

This is the final stage of the SDLC process where the system is given to the user or client for them to launch. Before the system is deployed, an acceptance test is conducted against the requirement parameters given to the developers to verify and validate that it satisfies those requirements as stated by the user. During the entire process, it is expected, that there is documentation on how to effectively use the system, to assist the organization using the system. Although a form of training would be done by the developers for the users, the documentation would be provided as a guide for them on how to use the system.

The developers must provide the organization with two papers for each system: an operator/user document that describes how to use the system, what error messages indicate, and so on, and a system document that provides the specifics of the system architecture. Once the system has been deployed, a maintenance team is usually assigned to the client’s organization for procedural system check-ups and to attend to whatever issues may have occurred since the deployment of the system. This group offers user input, as well as consultation and assistance to customers while they are using the product. They also ensure that the system's components that need to be upgraded are updated without putting the system at risk of security breaches.

2.4.2 Software development life cycle models

Software process models or SDLC models can be said to be abstract representations of a software process. Software development life cycle models are an essential element of Software Engineering; a field whose goal is to create high-quality software that is delivered on time, at a low cost, and meets the needs of the company needs. The most challenging task is to find a model that fits the company's structure. To do so, every organization must comprehend the model and its real-time application so that users can relate to, understand, and discover how it works. Every model in the SDLC is made up of several stages, each of which must be understood to determine which step leads to which and what the step's basic requirements are.

The software development model is a representation of the complete software development process; it clearly outlines the major activities that will guide the software development team (Jiujiu, 2018). These methods are structural frameworks for all system development work and duties, from operation and maintenance to the complete software life cycle process, as well as the relationships between the many phases of software development activities (Zhang & Li, 2006). There are many general models for software processes, some of which are: Waterfall model, V-shaped Model, Iterative model, Agile model, Spiral model, and Extreme Programming. As earlier stated, each of these models follows a sequence of steps of the life cycle to fully ensure the success of the software development process.

a. Waterfall Model

The Waterfall is the traditional model of the Software Development Life Cycle. It is one of the oldest models and has been implemented in many major government projects and organizational projects. It was first proposed by Winston W. Royce in the year 1970 and was called the Linear Sequential SDLC model, hence, the Waterfall model can also be called the linear sequential development model. The waterfall model can be described as a sequential software development process, where the processes can be described to be flowing downward through a list of phases that must be executed to successfully build the system (Youssef, 2012). In this approach, each successive phase must be completed one after the other, with the preceding phase being finished before going on to the next. As a result, a recursive waterfall model may be described as one in which a single-phase can be repeated until it is perfected. The waterfall model is an example of a plan-driven process, in which all process tasks must be planned and scheduled before beginning work on them. This model consists of 5 major phases which include: requirement analysis, design, coding, testing, implementation, that are in such a manner that no one phase is ever repeated once it has been passed and does not move to the next phase until the previous phase has been completed (Thakur, Singh, & Chaudhary, 2015). In principle, the result of each phase is a document.

The waterfall model can be applied to fit various applications, but it is best fitted when the software requirements are well understood documented, clear, and fixed. It can also be used when the technology to be used is not dynamic when the software requirements are not ambiguous. Simply, the waterfall model is best fitted for a small project where all the requirements are understood. One of the drawbacks of this model is that it is prone to a lot of risks, it cannot be used for projects with dynamically changing requirements (Munassar & Govardhan, 2010), most times a working system is delivered late during the life cycle.

b. V-Shaped Model

The V-shaped model is an improvement on the waterfall approach, in which testing functionality is introduced at each step of project development rather than at the end, resulting in improved project development (Barjtya, Sharma, & Rani, 2017). Consequently, the project’s goal cannot allude because a test plan has been set before even the program code is written. The focus of the test plan is to meet the functionality requirements that have been specified (Sujit & Pushkar, 2013). Just like the waterfall model, you can only move to the next step when the previous step has been duly completed.

The creation of a V-shaped model, like a waterfall model, starts with the requirements phase. The high-level design phase and the low-level design phase are the following steps. The system architecture and design are emphasized in the high-level phase. An integration test plan is developed at the high-level phase to test the software system's components and their ability to function together. The actual software components are designed at the Low-level design stage, as well as a unit test (Munassar & Govardhan, 2010). Govardhan and Munassar (2010), The implementation phase occurs when all of the program code has been written. Following the completion of the code, the execution path continues down the right side of the V, where the previously produced test plans are now utilized.

The V-shaped model can be used for projects where and accurate product testing is required, it could also be used for small and mid-size projects, where the requirements have been strictly predefined. One of the major disadvantages of the model is that it is not flexible, and it is not appropriate for small projects.

c. Iterative Model

The iterative model also called an incremental model is a design approach that is based on feedback and evaluation of the previous model that was deployed to users and using the information generated to develop a newer version of that same system. The needs are not fully defined in an iterative model; rather, the process starts with a limited set of criteria, with each iteration generating a small version of the product or system, and this is continued until the final version is created (Shylesh). It is important to note that each iteration adds new functionality to the product or system and continues to do so until the entire process is completed, the changes that have been made increase the performance of the system. The main idea behind this technique is to create a system in smaller chunks at a time (increments), allowing software engineers to benefit from what has been learned during the creation of previous parts or versions of the system (Abdullahi & Ogwueleka, 2017).

Unlike the previous model; Waterfall and V-shaped model, in the iterative model, the requirements do not need to be completely stated before development. One major advantage of this model compared to other models is that we get a working version of the application early in the process, therefore, it is less expensive to make changes (Barjtya, Sharma, & Rani, 2017). One of the disadvantages of this model is that it requires many resources and management, correction of a problem in one unit would require correction in all units which would of course consume a lot of time. This incremental model is best suited when the important and major requirements have been completed although some functionalities can evolve with time. Also, this model can be used when there is time to market constraint.

d. Spiral Model

A spiral model is a step-by-step approach that emphasizes risk analysis. Boehm introduced it in 1988 as a risk-driven development process paradigm. Rather than a series of operations, this software process may be depicted as a spiral. The spiral model develops a system using a periodic technique. The following steps are included in each of these cycles: planning, risk analysis, engineering, and assessment. A software project repeatedly passes through each cycle of the spiral, where the baseline of the spiral starts in the planning phase and requirement is gathered and risk is assessed. Subsequent spirals then build on the baseline spiral.

The needs are obtained during the planning phase, according to Sujit and Pushkar (2013). A procedure is conducted in the risk analysis phase to identify risk and alternative solutions. After the risk analysis phase, a prototype is created. The engineering phase produces software, which is then tested after the phase. Before moving on to the next spiral, the client can assess the project's output to data in the evaluation phase (Munassar & Govardhan, 2010). Some benefits of using the spiral model include that software is developed and handled strategically and project monitoring is easy and effective, there is a high amount of risk analysis therefore, it is easy to avoid risks, it is good for large and mission-critical projects. The disadvantages of this model include, that it can be costly, and the success of the project is dependent on the risk analysis phase. The spiral model does not work well with smaller projects. The spiral model can be applied to projects where the constraint on budget and risk evaluation is important, where requirements are complex, and need evaluation to clarify.

e. Agile Model

Agile Software Development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams (Harish & Syan, 2012). It's a conceptual framework that encourages foreseen interactions throughout the development process. It believes that each project should be approached differently and that existing methods should be tailored to the project's needs. Because the agile model assumes that the user's requirements change in the IT world, the customer gets the finished system they want (Khan, Shadab, & Khan, 2020).

The agile model is a hybrid model that combines the iterative and incremental approaches by dividing software into apparatus and delivering a working model of a component at the end of each cycle or iteration. This model generates updated releases, each of which contains some incremental updates. After each iteration, the product is tested to determine whether it is acceptable or not. Extreme programming (XP), crystal methods, scrum, and feature-driven development are some of the most popular agile software development methods.

Sommerville (2011), in his description of The Agile methodology, mentioned a set of statements called manifestos that have been agreed upon by a group of leading developers, which guide the software process and give them the following values: "*Individuals and interactions over processes and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation, Responding to change over following a plan".* The agile method can be used when the users need change dynamically when there is less cost for the changes implemented because of the numerous iterations, and this model can also be used when it only requires initial planning to start the project (Bindia, 2019).

2.4.3 Extreme programming

Extreme Programming (XP) is a methodology for Agile software development. It is a lightweight, efficient, and adaptable software development process that was created to meet the unique needs of small teams developing software in the face of ambiguous and shifting requirements. Extreme Programming (XP) is an iterative software development process that seeks to produce higher-quality software and assist in finding the best solution. The goal of XP is to establish ideals and principles that will influence team behavior. It is a style of agile software development that emphasizes short development cycles and frequent delivery cycles. These releases are intended to boost software productivity and quality through the use of specialized procedures that focus on building checkpoints for the acceptance and satisfaction of client expectations.

According to Wu (2015), the model's fundamental concepts include a set of beliefs, rules, and methods for rapidly building high-quality software that adds the most value to the client in the shortest amount of time possible. According to Beck (2000), and Lippert and Roock (2001), the XP technique is meant to meet the requirements of a competent small team of fewer than ten developers co-located with the client and developing non-critical software using object-oriented technologies. Typically, an XP project results in a software project in which software development begins immediately, virtually no documentation artifacts are created (except for “user stories” written on index cards), and the project proceeds iteratively, with daily prototypes created with the direct input (and occasionally assistance) of stakeholders until the desired effect is achieved.

2.5 Unified Modelling Language (UML)

In the field of object-oriented software engineering, the Unified Modeling Language (UML) is a standardized general-purpose modeling language. The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, building, and documenting software-intensive system artifacts. It provides a standard way to write a system's blueprints, which include both conceptual and concrete elements like business processes and system functions, as well as programming language statements, database schemas, and reusable software components (Padmanabhan, 2012).

Unified Modelling Language was first introduced in the late 1990s, as an evolution of the object-oriented programming language(cite). Its origin is an intersection of the three popular methodologies of the 1980s and 1990s: the Booch method, Rum-baugh’s Object-Modeling Technique (OMT), and Jacobson’s Object-Oriented Software Engineering (OOSE). The first version of UML (UML 1.x), was officially adopted and standardized by the Object Management Group (OMG) in the year 1997 and it integrated James Rumbaugh’s object-modeling technique, Grady Booch’s component notation, Ivar Jacobson’s use case notation, Archie Bowen’s timing analysis, and David Harel’s statecharts (Xiaocong, 2015). According to Keng

(2010), UML 1.x defines nine diagramming techniques: Class diagram, Object diagram, Component diagram, Deployment diagram, Use Case diagram, State-chart diagram, Activity diagram, Sequence diagram, and Collaboration diagram (Booch, Rumbaugh, & Jacobson, 2005). While the UML 2.x defines thirteen diagramming techniques. The most recent version of the unified modeling diagram is the UML 2.5.1 and the diagrams described in this review are a part of that version.

2.5.1 UML diagrams

A Unified Modeling Language (UML) diagram is a graphical representation of a model of a system that partially signifies the design and implementation.

UML diagrams are categorized into three different views. They include; Structural diagrams, Behavioral diagrams, and Interaction diagrams. The elements that must be present in the system are highlighted in structural diagrams. This type of view is commonly used to document the system software's software architecture. Structural diagrams include class diagrams, component diagrams, deployment diagrams, object diagrams, composite structure, and package structure.

Behavioral diagrams are diagrams that depict a system's behavior and are used to describe its functionality. These types of view here include; Activity diagrams, State machines, Use case Diagrams, Interaction models. Interaction diagrams depict the flow of control and data among the components of the system under consideration. Under interaction diagrams, we have communication/collaboration diagrams, sequence diagrams, interaction overview, timing diagrams.

2.5.2 Types of UML diagrams

Altogether, there exist 14 types of UML diagrams. These diagrams span across the structural view, behavioral view, and interaction view. The types of UML diagrams include:

a. Class diagrams

It describes the structure of a system by showing the system's classes, their attributes, and the relationships among the classes. UML class diagrams explore domain concepts in the form of a domain model, analyze requirements in the form of a conceptual/analysis model, and depicts the detailed design of object-oriented or object-based software.

b. Component diagrams

This diagram is used to model the physical aspects of a system. These physical aspects are elements such as executables, libraries, files, documents, etc., which reside in a node. Component diagrams show how a system is split into modules called component and shows the dependencies among them.

c. Deployment diagrams

The hardware used in system implementations, as well as the execution environments and artifacts deployed on the hardware, are described. A UML deployment diagram depicts the configuration of run-time processing nodes as well as the components that reside on them. The physical aspects of an object-oriented system are modeled using deployment diagrams, which are a type of structure diagram. They're frequently used to represent a system's static deployment view (topology of the hardware) (creately, 2021).

d. Object diagrams

This diagram is a type of class diagram that depicts the detailed state of a system at a given point in time. It includes objects and their relationships and can be thought of as a subset of a class diagram or a communication diagram. An object diagram focuses on the characteristics of a group of objects and how they interact.

e. Composite structure

A composite structure diagram is a UML structural diagram that contains classes, interfaces, packages, and their relationships, as well as their relationships, and that provides a logical view of all or part of a software system. It depicts the internal structure of a structured classifier or collaboration (including parts and connectors). This diagram is similar to a class diagram in that it allows you to go into greater detail when describing the internal structure of multiple classes and showing how they interact. Inner classes and parts can be graphically represented, as well as associations between and within classes.

f. Package diagrams

The arrangement and organization of model elements in a middle to large scale project are depicted in a package diagram, which is a type of structural diagram. Package diagrams can depict both the structure and dependencies between sub-systems or modules, as well as different perspectives on a system, such as a multi-layered (aka multi-tiered) application - multi-layered application model. Package diagrams are structural diagrams that demonstrate how various model elements are organized and arranged in the form of packages.

g. Activity diagrams

Activity diagrams are visual representations of activities graphical representations of step-by-step workflows actions and activities. Activity diagrams depict the workflow from start to finish, highlighting the numerous decision paths that exist in the activity's progression of events. They can be used to describe situations in which parallel processing is possible during the execution of some tasks.

h. State machine diagrams

A state diagram, also known as a state machine diagram, is a form of a behavioral diagram in the Unified Modeling Language (UML) that depicts object transitions. The state machine diagram depicts the sequence of states that an object in a system goes through. It records the activity of the software system. State machine diagrams are used to describe an application's behavior.

i. Use case diagrams:

A use case represents a system's functionality in terms of actors, their objectives, which are expressed as use cases, and any relationships between those use cases. A use-case model is a representation of a system's intended functions and its surroundings. It also acts as a contract between the developer and the consumer.

j. Interaction diagrams

A behavior containing a series of messages sent among a set of objects inside a context to fulfill a goal is called an interaction. A message is a description of communication between objects that delivers information in the hopes of the action succeeding. The interaction diagram depicts how different entities in the model interact with one another. The activity and sequence diagrams are combined in this diagram. Interaction is a phrase that refers to a collection of messages that are sent between entities to accomplish specific tasks in the system. It can use any classifier feature that it has access to. The messages and the lifeline are the most important parts of the interaction diagram.

k. Communication/ Collaboration diagrams:

These diagrams, give yet another approach to demonstrate how objects interact to carry out the functionality of a use case or a subset of a use case. Communication diagrams are meant to highlight the links between items rather than the interactions of objects across time, as sequence diagrams do. A collaboration diagram depicts the interactions between objects in terms of sequenced messages, where each message sent between the objects is assigned a number (Er. Singh & Dr. Sidhu, 2018).

l. Sequence diagrams:

This is a type of interaction diagram that depicts how processes communicate with one another and in what sequence. As part of a use-case realization, it's used to demonstrate how objects interact to accomplish the behavior of all or part of a use case. Sequence diagrams contain object/class that shows the object/class involved in the interaction, messages that show an exchange between objects, execution occurrence which shows how the object is executing, and object life line.

m. Interaction overview

It shows the flow of activity across diagrams while also providing an overview of the flow of control of the interactions. It can display a control flow with nodes containing interaction diagrams that demonstrate how a collection of fragments could be launched in different circumstances. The nodes in interaction overview diagrams represent interactions (sd) or interaction usage, and they show the overall flow of control (ref).

n. Timing diagrams

A subset of the interaction diagram in which the emphasis is on temporal limitations. Timing diagrams depict the sequence of events and their corresponding consequences on states and property values. Time is measured horizontally from left to right. They can be used to demonstrate method execution profiling or scenarios involving concurrency (Padmanabhan, 2012).

2.6 System Development Tools

In this study, the following technologies were used to implement the system; HTML5, android studio, Kotlin, CSS, Node JS, Firebase.

a. Android Studio

Android is a popular computer operating system built on the Linux® kernel. Android's first commercial version launched in 2008 as a mobile phone platform, at a time when BlackBerry was the most popular cell phone for business users. The Android platform evolved from the efforts of an organization called the Open Handset Alliance (OHA), which was founded with the stated mission of collaborating to "build a better mobile phone." However, what began as a collaboration between those carriers, hardware makers, and software providers has grown to become the world's most popular platform. Android has been supplied to the market by multiple significant firms across the globe and numerous industries in the decade since its inception. Samsung, another OHA member, is the world's top manufacturer of smartphone devices, thanks to Android. Android Studio is free and runs on Windows, OS X, and most Linux distributions.

b. Hyper-Text Markup Language 5 (HTML5)

Hyper-Text Markup Language (HTML), is a kind of universal publishing language used by the World Wide Web. It was first developed by Tim Berners-Lee, while he was at CERN. HTML 2.0 was developed under the aegis of the Internet Engineering Task Force (IETF) to codify common practice in late 1994. HTML+ (1993) and HTML 3.0 proposed much richer versions of HTML. Despite never receiving consensus in standards discussions, these drafts led to the adoption of a range of new features. The efforts of the World Wide Web Consortium’s HTML Working Group to codify common practice in 1996 resulted in HTML 3.2 (January 1997). This was followed by HTML 4.0, which contributed little to the richness of the World Wide Web. In 2008, W3C proposed HTML5, which focused on semantics, which was in line with the original concept proposed by Tim Berners-Lee on how the WWW should work.

c. Cascading Style sheet (CSS)

CSS is a method designed for the distribution of styling on the World Wide Web (WWW). CSS allowed for the separation of content from style and allowed style to be applied generically over the content. CSS was being used as early as October 1995 and became part of W3C's recommendation in March 1998. Initially, CSS included layout, color, fonts, and simple presentation techniques which allowed for information structuring and highlighting, adding basic richness to the textual level of the website. It equips developers with the means to display information in a way that shows its nature and importance. Some of the rich tools in CSS, include, absolute positioning, with z-index, which allows for overlapping of elements in CSS2. The advent of CSS3 expanded the styling options in CSS2 but a modular style including animation, transition, and transformation.

d. Node JS

Node. Js or Node is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node. js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. It is built based on Google's V8 engine. Both Node and V8 are implemented in C and C++ for less memory consumption and faster performance. Node is based on the Asynchronous I/O event model designed for developing scalable network applications. It fires callbacks on events, and each client event generates its callback (Nirgudkar & Singh, 2017). Node was originally written by Ryan Dahl in 2009, thirteen years after the introduction of the first JavaScript environment.

e. Express JS

Express was created by TJ Holowaychuk, who described it as a web framework inspired by Sinatra, a web framework based on Ruby. Express is a server-side framework built in the NodeJS environment. It handles the client requests to the server and manages the routing and HTTP methods such as GET, POST, PUT, etc. Express.js is lightweight and helps to organize web applications on the server-side into a more organized MVC architecture. Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node-based Web applications. The core features of Express framework are: it allows set up of middleware to respond to HTTP requests, it defines a routing table which is used to perform different actions based on HTTP Method and URL, and lastly, it allows to dynamically render HTML pages based on passing arguments to templates.

f. JavaScript

JavaScript (abbreviated as "JS") is a full-featured dynamic programming language that may be used to make a website more interactive. JavaScript was created by Brendan Eich in the year 1995, at Netscape Communications Corporation. The earlier version of JS was called Mocha and later changed to LiveScript before it then became JavaScript. The JavaScript API makes it possible to run scripts on the client's browser that are uploaded with the web page to provide client-side functionalities. Javascript's ability to manipulate and interact with the Document Object Model. The capabilities of the Document Object Model (DOM) are vast, including the ability to change and create style, tags, and the arrangement of elements. It is a first-class compiled programming language that is lightweight, interpreted, or just-in-time compiled. While JavaScript is best known as a scripting language for Web pages, it is also used in a variety of non-browser contexts, including Node.js, Apache CouchDB, and Adobe Acrobat. JavaScript is a single-threaded, prototype-based, dynamic language that supports object-oriented, imperative, and declarative (e.g. functional programming) programming styles.

g. Firebase

Firebase is a web application platform that enables developers to create high-quality apps. It uses the JavaScript Object Notation (JSON) standard to store data and does not utilize queries to insert, update, delete, or add data to it. It's the part of a system that acts as a database for storing information (Khawas & Shah, 2018). Firebase, according to Singh and Singh (2017), is a "NoSQL" database that is suitable for huge amounts of dispersed data. Firebase is a backend platform for developing Web, Android, and iOS applications, with features such as a real-time database, numerous APIs, multiple authentication methods, and a hosting platform. Because it considers the backend issues of every online application, Firebase is referred to be a Backend as a Service (BaaS) (Vidhate, Shah, & Kasture, 2018). Firebase provides the following services: analytics, cloud messaging, authentication, real-time database, storage, firebase test lab for Android, crash reporting, and notification.

2.7 Related Works

Ritesh et al. (2018), proposed a Hostel Management framework which would be a web application created for managing different activities in the hostel. The research led to the definition of Hostel Management Systems as an online site developed for managing different activities within hostels. This system was proposed to develop the system using HTML and CSS for the frontend development to make it user-friendly and PHP and JavaScript for backend as well as RFID for access control management. Ritesh, et al., (2018) highlighted that the paper-based registration procedures are tedious and costly and stated that the proposed system would reduce the paperwork and redundancy, hence, improving productivity, aid the hostel in data management and integration of student's profiles, give an account of a student with ease. One major limitation identified with this system is, RFID scanners are only for access control but do not possess a feature for individual identification. Hence, anyone could use another person's tag.

Shoewu, Braimah, and Duduyemi (2016) proposed to design a hostel management system (HOMASY), that would provide a computerized and reliable method of registering students and staff into the school. They proposed to use HTML for the front-end development of the system and PHP, and MySQL database application to implement the system. The system proposed to assign rooms to students, eliminating the need for student affairs management. The result of the study was a system that allowed porters and students to register and allow students to make new room appointments.

Shoewu, and Idowu (2012), proposed the development of an attendance management system using biometrics. The proposed Automated Fingerprint Attendance System (AFAS) is a system that records student attendance by comparing a single fingerprint image with the fingerprint image previously stored. This is done to affirm the eligibility of a student. During testing, the system was able to register student's biometric image and compare it with the registered image and was able to determine the students who are not eligible to take an exam.

Stephen, Chukwudebe, and Ezenkwu (2015) proposed an Integrated Identity and Access Management Systems for Tertiary Institutions in Developing Countries. During the research, they proposed a system called Tertiary Identity and Access Management System (T-IAMS). The result of the test carried out on the system demonstrates that student identity can be managed centrally. A significant distinction between the research and this proposal is that the latter aims to provide an effective and efficient way of monitoring student's activity within and out of the hostel, employing an identification system (QR code), while the former is being used to keep track of student's medical record, course registration, and library information system.

Muthukumar, Albert, Gokul, and Denil (2019), described in their paper a system that allows passengers of a train to check themselves in without the need for a conductor, using QR code and a biometric-based authentication system. QR codes can be easily scanned by smartphones or other related devices and the data that has been scanned is received and retrieved from the same device. The proposed system was developed using an Android Developer Kit available on Eclipse IDE. The result is an interface that allows passengers to book seats, perform online transactions as well as scanning the QR code of seats allocated to them to ensure that it is theirs. Although this system implements QR codes for identification, it is limited to train systems and does with propose to monitor passenger activity or regulate access.

Ayanlowo k., Shoewu, Omitola, and Babalola (2014), developed a computer-based Hostel Accommodation System for managing hostel facilities in the institution. It was designed to automate, manage and oversee all processes that involve records of students residing in a large hostel. The system was designed using Visual basic, and the database management tool used for implementation is Microsoft Office Access.

Several systems have been deployed and many more types of research have been carried out on Hostel Management Systems, Identity, and Access control Management, and QR codes separately. Although it can be seen that most of these studies are carried for different purposes, this project aims to integrate each of them which their unique characteristic to form a new system that can be modified and implemented in any organization or institution.

CHAPTER THREE

METHODOLOGY OF THE STUDY

3.1 Method of Identification of User and System Requirements

Using informal interviews and an evaluation of the existing methods for identifying and accounting for students within the hostel, the user and system requirements of the software system were identified during this project. The techniques used in highlighting the system requirements of the software include; Nonfunctional requirements that specify criteria that can be used to judge the operation of the system, functional requirements which describe the services the software must offer, Hardware requirements, and the Software requirements.

During the course of the development of the system, the user requirements were identified and they include; system admin requirements and basic user requirements. In this chapter, the following models; Use case, Sequence, Activity, class, and System architecture, that were used in the system design would be adequately discussed. This chapter would also talk about the method of system implementation, database implementation, and front-end implementation. Finally approaches to testing the system would be highlighted.

3.1.1 Identification of system requirements

This section discusses the functional and non-functional requirements of the system being developed. It highlights the requirements for system implementation. Also described in this section are the hardware and software requirements for the development of the system.

a. Non-functional requirements

The non-functional requirements of the system include:

i. Ease of use: the system was implemented in such a way that the user would be able to operate it with little to no help need. The texts and buttons were properly styled for easy viewing and they perform the function for which they are stated. Users should be to use the system based on their roles.

ii. Response time/speed: a request made to the system should be responded to immediately while displaying the correct output.

iii. The system should identify students who are from other hostels and notify the porters (users) accordingly.

iv. Reliability: when scanning a student, the system is expected to attribute the QR code to a student and display information required from the database which belongs to the student being scanned.

v. Security: users and admin are expected to use the interface assigned to their roles. That is, only the admin (registration officer) is expected to register other users and students.

b. Functional requirements

The functional requirements include;

i. The system should allow only authorized users to log into the system and display an interface for their roles.

ii. The system is expected to record student's status automatically as they are being scanned in or out of the hostel.

iii. Users should be able to view student's state logs.

iv. The system should be able to generate the QR code of each student during registration

v. Registration officer (admin) should add roles and assign them to the appropriate user, and those authenticated users are authorized to view the interface for their roles.

vi. Users shall be able to search for students to know their state (in or out) at any point.

vii. Users (porters) should be able to add reasons on the “View” page in the interface assigned to their roles for the student who is absent when the log is being checked.

c. Hardware requirements:

The desktop for the web application should be at least 64bits, have a minimum of 2GB memory and a minimum of 3GB storage space. For the mobile application, it should have a functioning back camera storage space of about 5GB, 4GB memory, and good battery life.

d. Software requirements:

The software required to implement this system includes; Visual studio code, Windows Operating system from 7 above, android studio, android studio SDK, java JDK, Firebase.

3.1.2 Identification of user requirements

This section highlights the system and user requirements of the system.

a. System admin requirements

For this project, the registration officer is assigned the role of administrator. The admin can register and assign roles to the user (porter and head porter). The admin is also responsible for registering students and generating their QR codes with the given their given information. The admin has the privilege of viewing the logs of all hostels, that is male and female hotels, and can download the log per day.

b. Basic user requirements

The basic users of this system are the porters and head porters. These users view the same interface but differ in what operations they can perform on it. The head porter can download hostel logs and view the hostel log of the hostel they are overseeing (either male or female). The porters can view the logs of the hostels that they are assigned to, scan QR codes of students and generate the logs of that day.

3.2 System Design Methods

The system design was specified using relevant UML diagrams such as use case diagram, sequence diagram, class diagram, and activity diagram. The system architecture was also designed and described in this section. The UML diagrams used in designing the system represent various functions that are available on the system and how they are used by the users of the system as identified in the previous section.

3.2.1 Use-case diagram

The use case diagram figure 3.1 described below, presents users of this system and the various actions that they can perform on the system. It also talks about their different roles, and conditions for each actor to perform various activities and for those activities to be termed as successful.

Table 3.1 describes the assigned role use case. This use case applies to only the admin user; the admin user navigates to the registered porter's page from the menu and selects a registered user from the table then the admin selects an available role and assigns it to the porter. The admin user must be logged in to his account to be able to perform this use case. Table 3.2 describes the register students use case, for which the only actor is the admin user. The admin user must be logged in to perform the task of navigating to the “Register Student” page, fill the form, submit the form, as well as

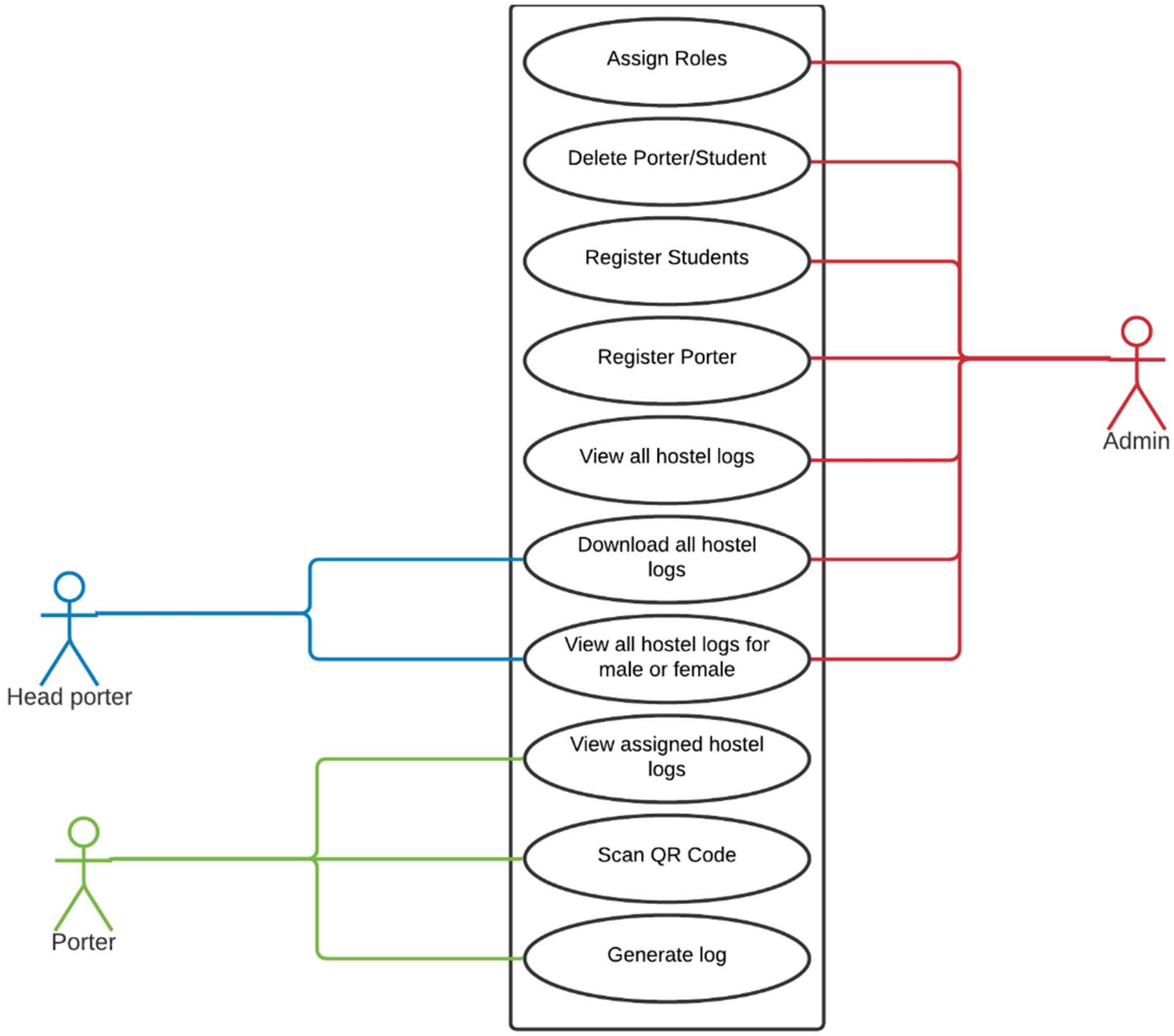


Figure 3.1: Use case diagram of the system

Table 3.1: Assign role use case

Use case Name	Assign roles
Actors	Admin
Flow of Events	• Admin navigates to the registration page • Admin selects a registered user • Admin selects an available role and assigns it to the user
Entry Condition	Admin must be logged in to the account
Exit condition	The admin assigns roles successfully. If the process is not successful, the admin is prompted by a message from the system on what the problem is and is requested to try the process again
Quality Requirements	The admin must successfully assign an available role to a user without errors.

Table 3.2: Register student use case

Use case Name	Register students
Actors	Admin
Flow of Events	• Admin navigates to the "Register Student" page • Admin fills form • Admin submits the form and generates QR code
Entry Condition	The admin must be logged in to the account
Exit condition	Admin successfully registers student and generate QR code. If the process is not successful, the admin is prompted by a message from the system on what the problem is and is requested to try the process again
Quality Requirements	Admin must successfully register the student and generate QR code without errors

generate a QR code for the student. Table 3.3 describes the register porter use case, for which the only actor is the admin user. The admin user must be logged in to perform the task of navigating to the "Register Porter" page, fill the form, submit the form, as well as generate a QR code for the student.

Table 3.4, shows the view of all hostel logs use cases. The actor of this use case is the admin user. The admin user navigates to the "View Hostel" page, clicks on the "View all" button, and views all hostel logs. Table 3.5, shows the download of all hostel logs use cases. The actors of this use case are the admin user and head porter and they need to be logged in to their roles to perform the flow of the event. These roles perform the following events, navigate to the "View log" page and click the "Download log" button. Table 3.6 describes the flow of events in the view of all hostel logs for males and females. This use case applies to the admin and head porter user roles.

The admin role performs the activity of navigating to the "View Log" page and view for both hostel gender, while the head porter can view for a specific hostel gender. Table 3.7 shows the view assigned hostel logs use case. The actor of the use case is the porter. The porter navigates to the "View Log" page, click on view, and then view the logs.

Table 3.8 shows the scan QR code use case. The actor of this use case is the porter. The porter focuses the camera device on the student card, the scanner detects and reads the QR code, and the system displays the student's details and their current status. Table 3.9 describes the generate log use case. The actor here is the porter. The porter moves to the general log button select the log of the day to be generated and click on the generate log button.

Table 3.3: Register porter use case

Use case Name	Register porter
Actors	Admin
Flow of Events	• Admin navigates to the “Register Porter” page • Admin fills form • Admin submits form
Entry Condition	The admin must be logged in to the account
Exit condition	Admin successfully registers porter. If the process is not successful, the admin is prompted by a message from the system on what the problem is and is requested to try the process again
Quality Requirements	Admin must successfully register porter without errors

Table 3.4: View all hostel logs use case

Use case Name	View all hostel logs
Actors	Admin
Flow of Events	• Admin navigates to the “View hostel” page • Admin clicks the “view all” button • Admin views all hostel logs
Entry Condition	Admin must be logged in to the account
Exit condition	Admin successfully views all hostel logs. If the process is not successful, the admin is prompted by a message from the system on what the problem is and is requested to try the process again
Quality Requirements	The admin must successfully view the log of all hostels without issues.

Table 3.5: Download all hostel logs use case

Use case Name	Download all hostel logs
Actors	Admin, Head porter
Flow of Events	• The user navigates to the “View log” page • The user clicks the “Download Log” button
Entry Condition	The user must be logged in to their account according to their roles
Exit condition	The user must successfully download the log. If the process is not successful, the admin is prompted by a message from the system on what the problem is and is requested to try the process again
Quality Requirements	The user must successfully download the log without errors

Table 3.6: View all hostel logs for male and female

Use case Name	View all hostel logs for male and female
Actors	Admin, Head porter
Flow of Events	• The user navigates to the "View Log" page • In the case of head porter, they select from the hostels they oversee. The admin views all logs across all hostels. • The head porter views the log of the hostel
Entry Condition	The user must be logged in to their account according to their roles
Exit condition	Users must successfully view the log of the hostel selected. If the process is not successful, the admin is prompted by a message from the system on what the problem is and is requested to try the process again
Quality Requirements	The user must successfully log without errors. In the case of the admin, he/she views all hostels

Table 3.7: View assigned hostel logs use case

Use case Name	View assigned hostel
Actors	Porter
Flow of Events	• Porter navigates to the "View Log" page • Porter clicks on "View" • Porter views log
Entry Condition	Porter must be logged in to their role account in the hostel they have been assigned to
Exit condition	Porter must successfully view the log of their assigned hostel. If the process is not successful, the admin is prompted by a message from the system on what the problem is and is requested to try the process again
Quality Requirements	Porter must view the log from the hostel

Table 3.8: Scan QR code use case

Use case Name	Scan QR code
Actors	Porter
Flow of Events	• Porter focuses the camera device on the student card • The scanner detects and reads QR code • The system displays student's details and their current status
Entry Condition	Porter must be logged in to the mobile device
Exit condition	Porter must successfully scan students in / out If the process is not successful, the admin is prompted by a message from the system on what the problem is and is requested to try the process again
Quality Requirements	Porter must scan students before they enter or leave the hostel

Table 3.9: Generate log use case

Use case Name	Generate log
Actors	Porter
Flow of Events	• Porter locates the "Generate Log" page • Porter selects a log of the day they want to generate • Porter clicks the generate log button
Entry Condition	Porter must be logged in to their role account
Exit condition	Porter must successfully generate a log of the day they want. If the process is not successful, the admin is prompted by a message from the system on what the problem is and is requested to try the process again
Quality Requirements	Porter must generate the log successfully.

3.2.2 Sequence diagram

Sequence diagrams were used to illustrate the order of action for each process on the system. It shows the internal component and the user involved in the process. Some of the processes designed with this diagram are Login, Register student View logs and so, on.

a. Login sequence

Figure 3.2 below, describes the login activity. When the user sends a request for the login page, a request for the loginForm entities and attributes is sent to the loginModel. The loginModel returns the loginForm entity to the loginController. The loginController displays the login view. The user then fills the form with email and password and logs in. The loginForm sends a post request with submitted details to the loginController and the loginController sends a query to the loginModel to validate the submitted details.

The loginModel responds to the query from the loginController and the loginController sends the user ID and session token to the user if the process was successful, else it displays an error message.

b. View logs sequence

Figure 3.3. below, describes the view Logs sequence. In this sequence, the admin clicks the ViewLogsButton, and the ViewLogsButton requests a log view from the LogController. The LogController requests log entities and attributes from the LogModel. The LogModel returns a log form entity to the LogController. The LogController displays the log view to the admin.

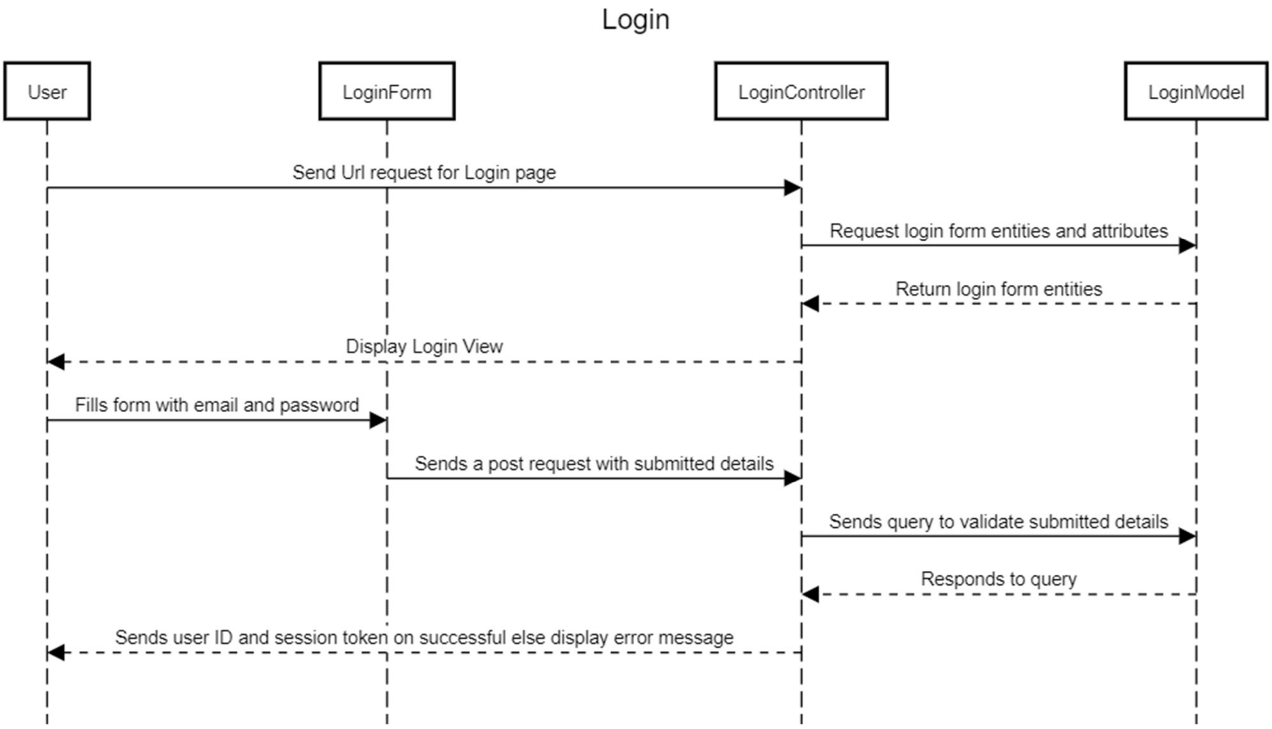

Login
User
LoginForm
LoginController
LoginModel
Send Url request for Login page
Request login form entities and attributes
Return login form entities
Display Login View
Fills form with email and password
Sends a post request with submitted details
Sends query to validate submitted details
Responds to query
Sends user ID and session token on successful else display error message

Figure 3.2: Sequence diagram for Login page.

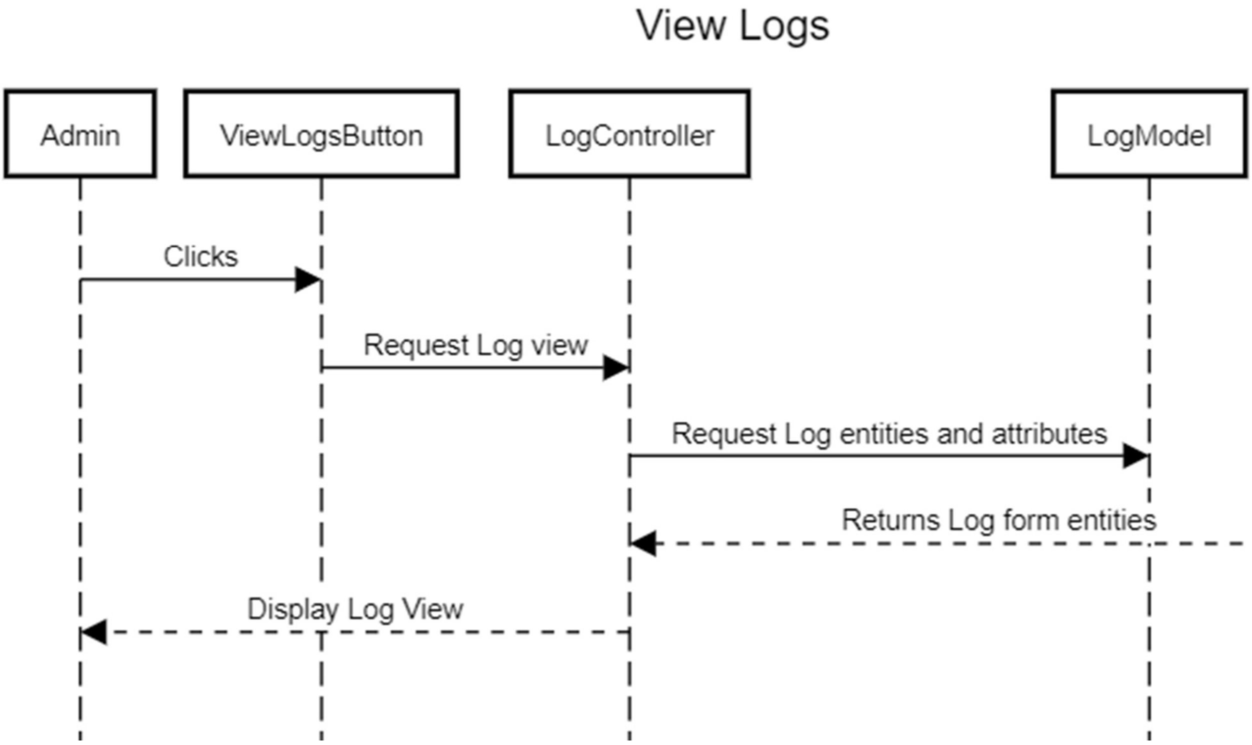


Figure 3. 3: Sequence diagram for View Logs sequence

a. Register porter

Figure 3.4 below, shows that the admin clicks on RegisterPorterButton, which requests the register porter view from the PorterController. The PorterController requests the register porter form entity and attributes from the PorterModel. The PorterModel returns the register form entities to the PorterController, then the PorterController displays the register porter view to the admin. The admin fills the form with the required data in the RegisterPorterForm, then a post request is sent to the PorterController and the PorterController sends an insert query with submitted details. The PorterModel responds to the query. The PorterController then sends a registration successful message to the admin. If registration is not successful, it sends an error message instead.

b. Register student

Figure 3.5 below describes the following actions; Admin clicks on RegisterStudentButton, which requests the register student view from the StudentController. The StudentController requests the registered student form entity and attributes from the StudentModel. The StudentModel returns the register form entities to the StudentController, then the StudentController displays the registered student view to the admin. The admin fills the form with the required data in the RegisterStudentForm, then a post request is sent to the StudentController with the submitted student details. The StudentController automatically generates the QR code and sends an insert query with the submitted details and the QR code image to the StudentModel. The StudentModel responds to the query from the StudentController, then the StudentController sends a registration successful message with the generated QR code image, else if registration is unsuccessful, it returns an error message.

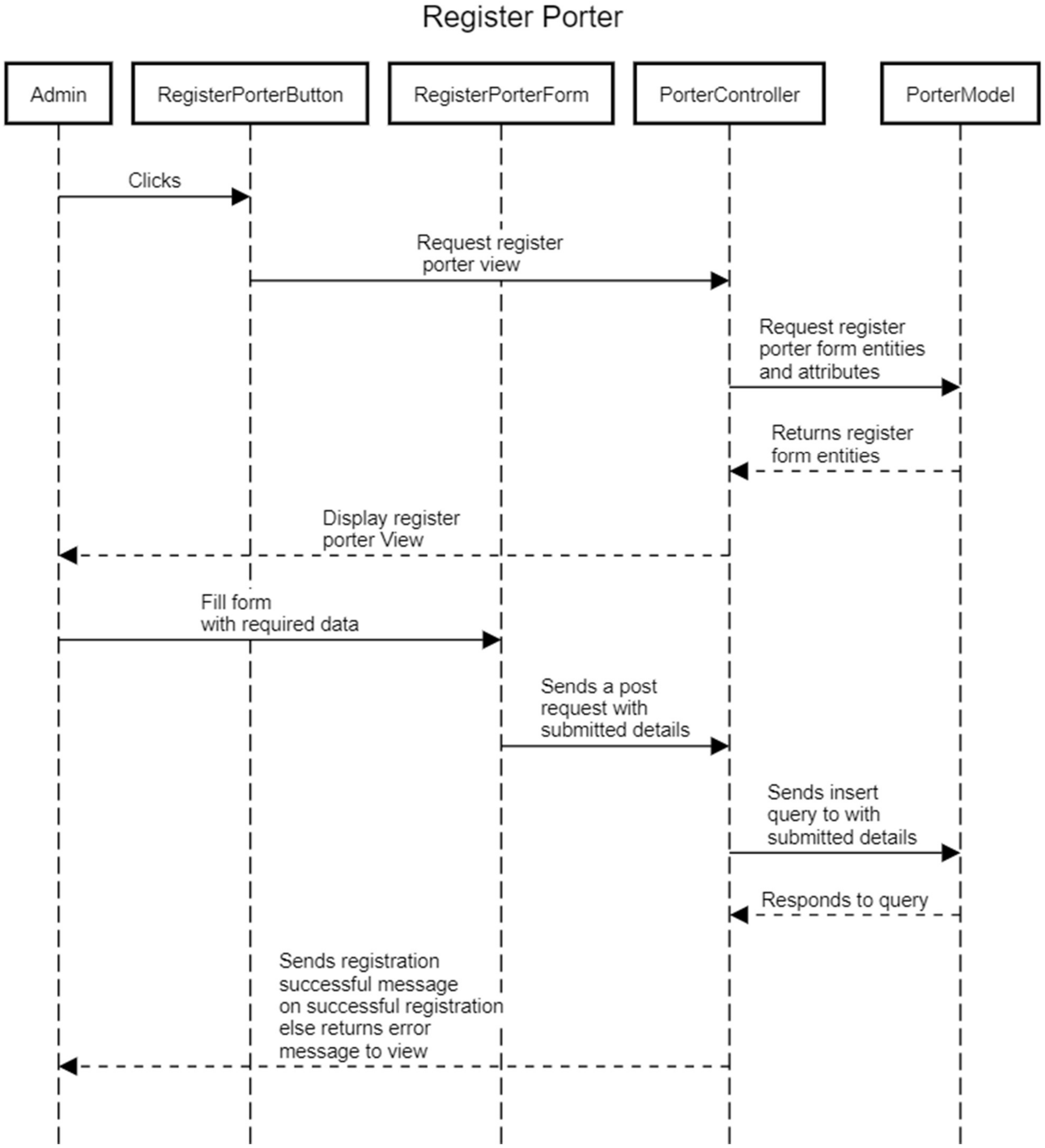

Register Porter
Admin
RegisterPorterButton
RegisterPorterForm
PorterController
PorterModel
Clicks
Request register porter view
Request register porter form entities and attributes
Returns register form entities
Display register porter View
Fill form with required data
Sends a post request with submitted details
Sends insert query to with submitted details
Responds to query
Sends registration successful message on successful registration else returns error message to view

Figure 3. 4: Sequence diagram for Register porter

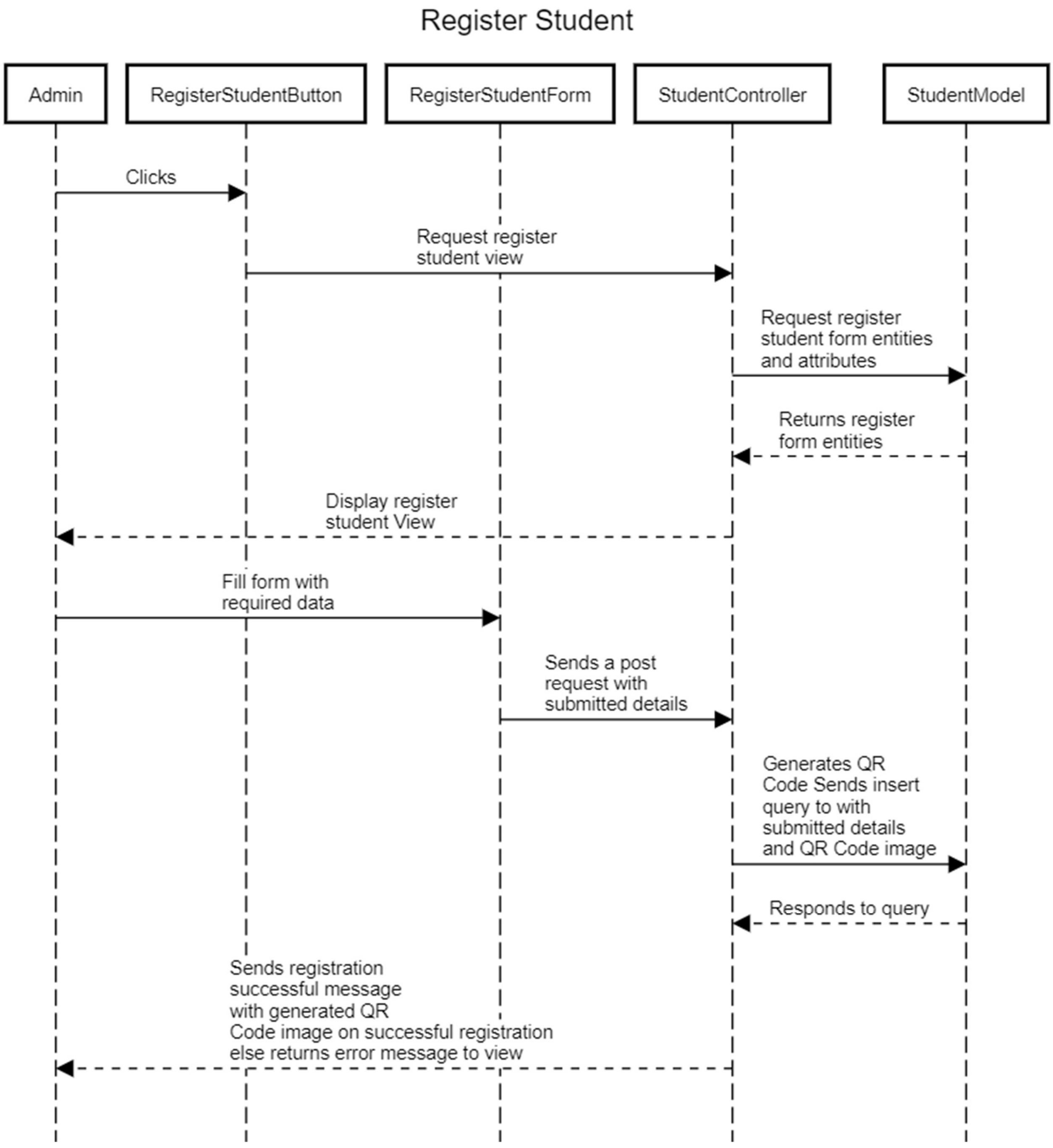

Register Student
Admin
RegisterStudentButton
RegisterStudentForm
StudentController
StudentModel
Clicks
Request register student view
Request register student form entities and attributes
Returns register form entities
Display register student View
Fill form with required data
Sends a post request with submitted details
Generates QR Code Sends insert query to with submitted details and QR Code image
Responds to query
Sends registration successful message with generated QR Code image on successful registration else returns error message to view

Figure 3. 5: Sequence diagram for Register student

a. Scan student QR code

Figure 3.6 above describes the following; porter reads QRCode placed before Scanning Device. The Scanning Device sends the detected QRCode encrypted data to the QRCode Scanner. The QRCode Scanner calls fetchStudentDetail method with decrypted information from QR code to the Data API. The Data API sends a query to fetch student details using matric number and updates the IsInHostel column in the Student Table. The Student Table return query results with student details to the Data API, then the Data API sends the details of the scanned student to the Scanning Device and the Scanning Device displays the scanned information to the porter.

3.2.3 Activity diagram

Figure 3.7 below, describes the flow of activity of the system. Once a user requests to log in, if the request was successful, the user logs in to their account in the specific roles. If the login was unsuccessful, the user is prompted to try again. If a user registers as the admin (registration officer), then the user can perform the following activities, assign roles, register students and porter, view the hostel log of all hostels, download the log of all hostels. If the user logs in as head porter, the user can view the hostel log of all male or female hostels and can download logs of the male or female hostel. If the user is not the head porter or admin, then the user is a porter. The porter can view the hostel log, scan the QR code, and generate a log.

3.2.4 Class diagram

Figure 3.8 below describes the class diagram of the system. The class diagram shows the relationship between the system entities. Attributes of each of these entities represent columns of the tables that they model in the database. The user class which comprises properties that define the user of the system (porter, head porter, and registration officer),

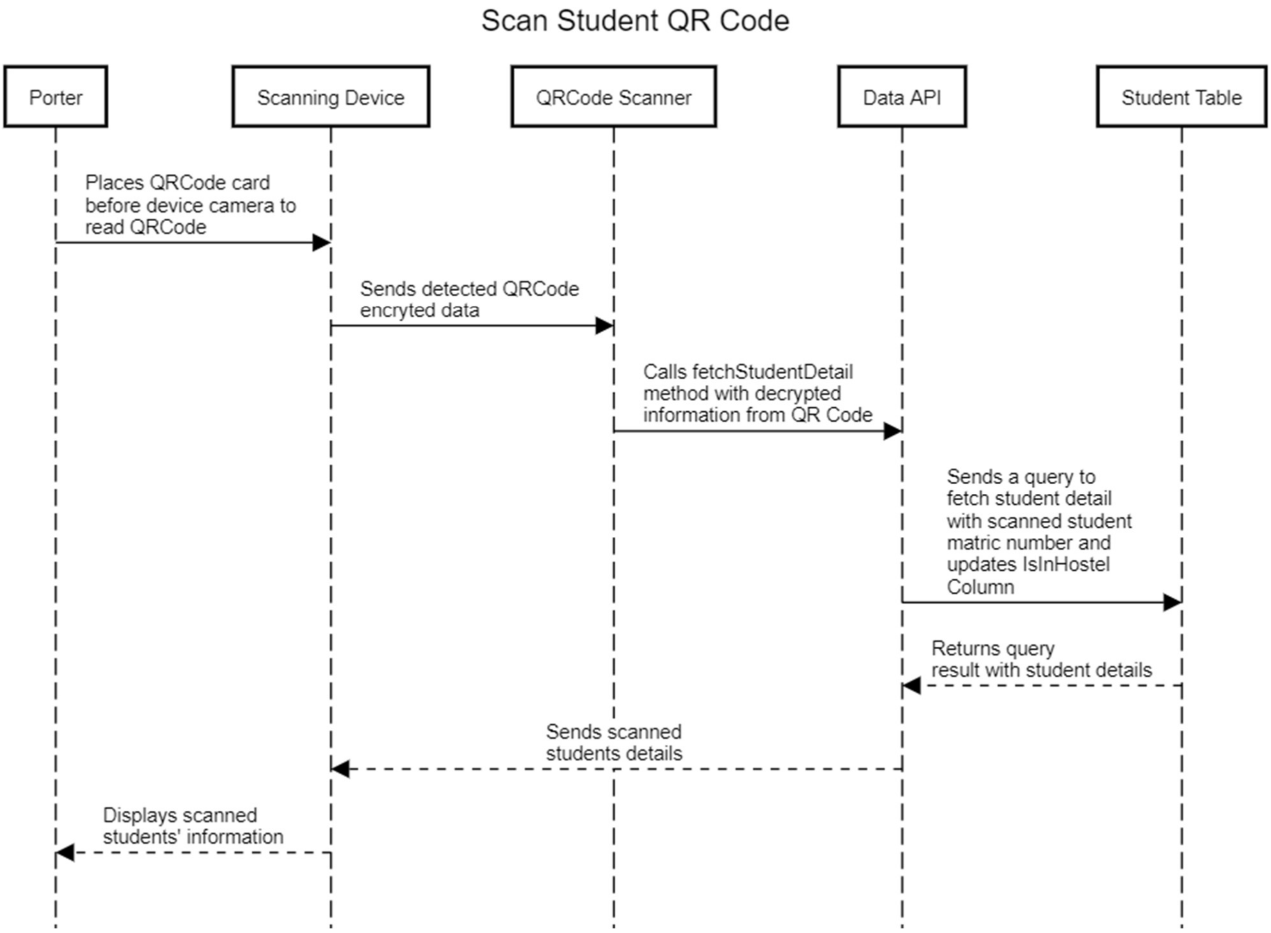

Scan Student QR Code
Porter
Scanning Device
QRCode Scanner
Data API
Student Table
Places QRCode card before device camera to read QRCode
Sends detected QRCode encryted data
Calls fetchStudentDetail method with decrypted information from QR Code
Sends a query to fetch student detail with scanned student matric number and updates IsInHostel Column
Returns query result with student details
Sends scanned students details
Displays scanned students' information

Figure 3. 6: Sequence diagram to scan student QR code

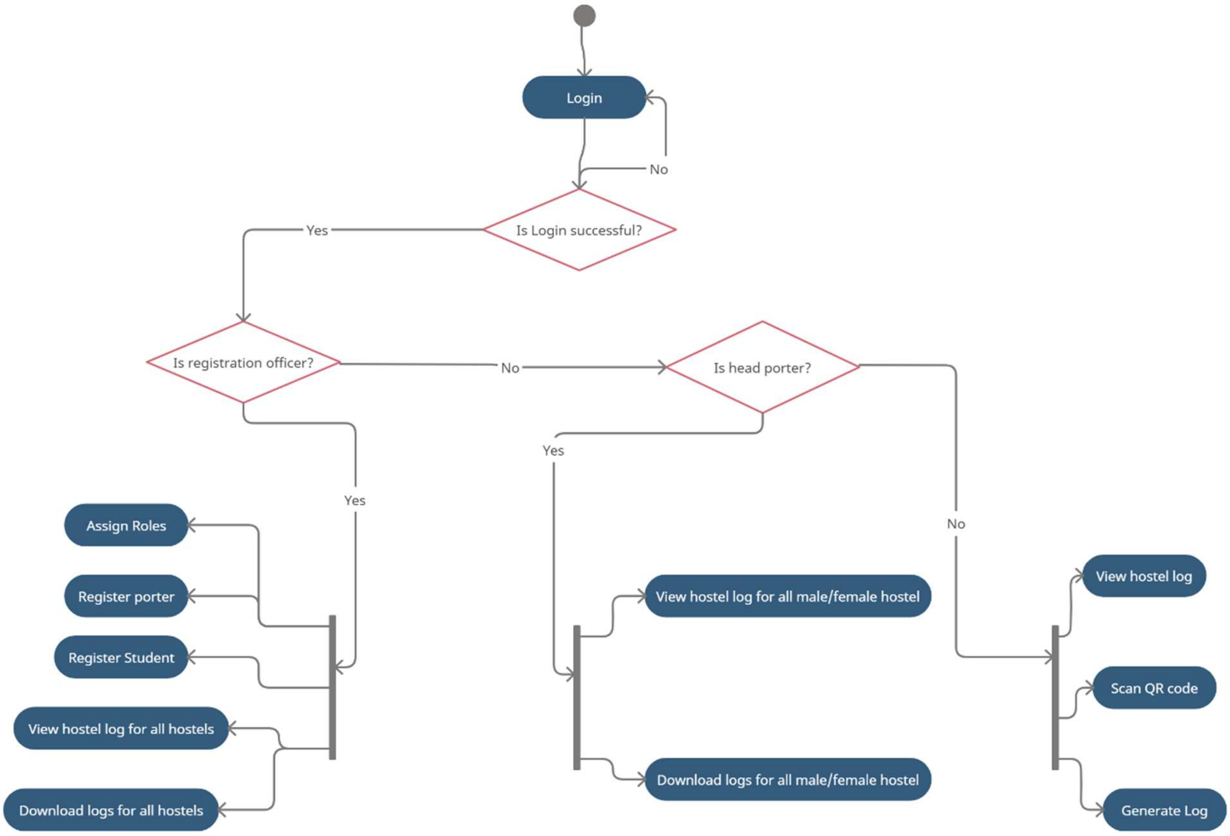


Figure 3. 7: Activity diagram of the system

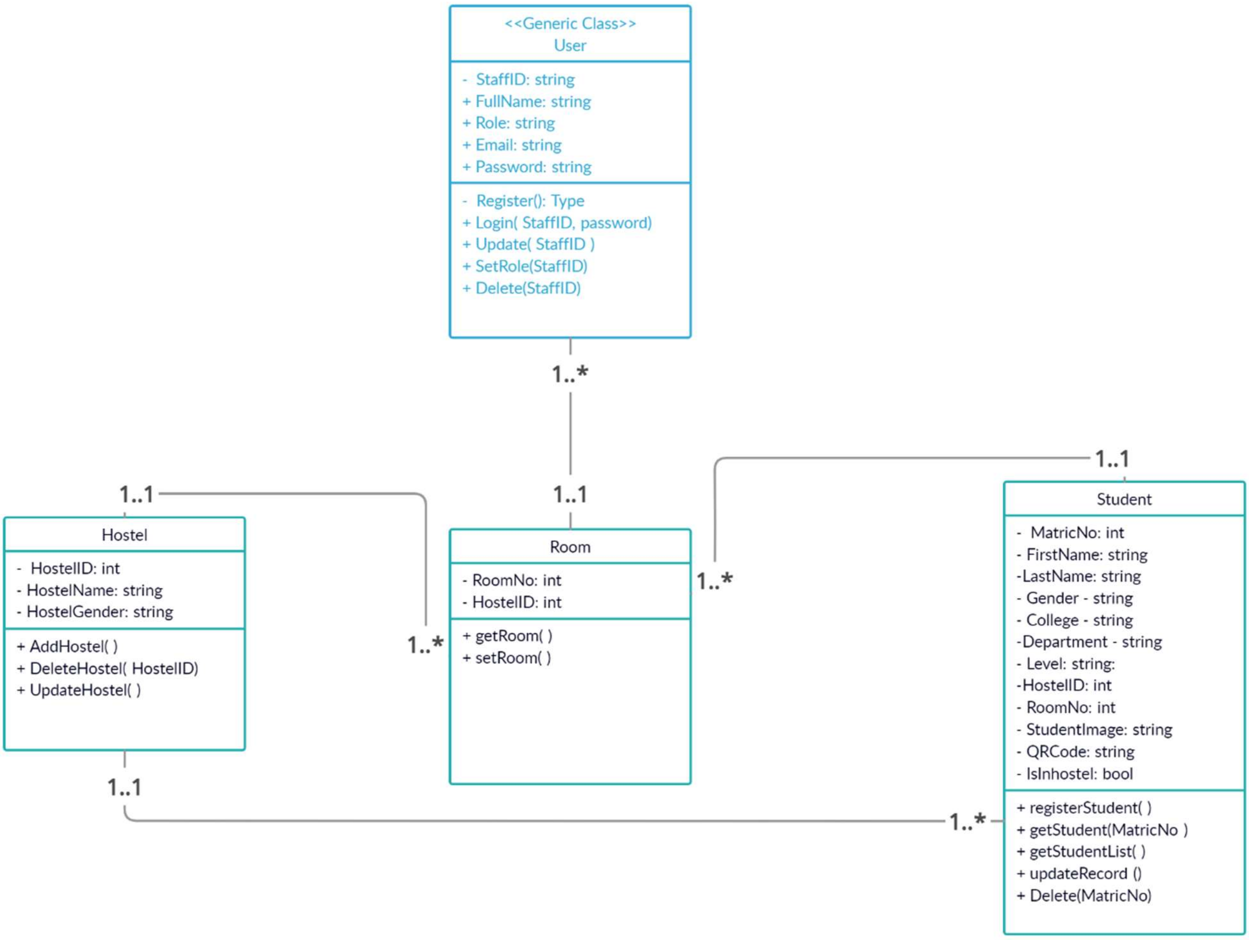


Figure 3. 8: System class diagram

also contains methods that are used for performing different operations and objects of this class.

3.2.5 System architecture

The system architecture in figure 3.9 below, depicts that the system presented via hosting on Node JS and Express server. Once the browser requests for the URL of the system, it is navigated to the express server. The server interfaces with the definition of routes to serve the appropriate page and execute the necessary requests. The system database is managed by Firebase's firestorm database which holds all collections used by the system.

3.3 System Implementation

This system was implemented using several technologies, languages, and dependencies. The database was implemented using firebase, while for the front-end implementation: HTML, Cascading style sheet (CSS), JavaScript, Android studio, CodeScanner dependency. The backend was implemented using NodeJS, and ExpressJS.

3.3.1 Database implementation

a. Firebase

This is a database management service from Firebase that was used in this project to manage storage, retrieval, manipulation of data, across the system. Firestore has been a No-SQL document type database that was used to create collections for porter, hostel, student, and logs with each collection hold documents for each record entity in the system. The firestore database was easily accessible with an auto-generated configuration script from firebase that created an instance of firebase service in the system.

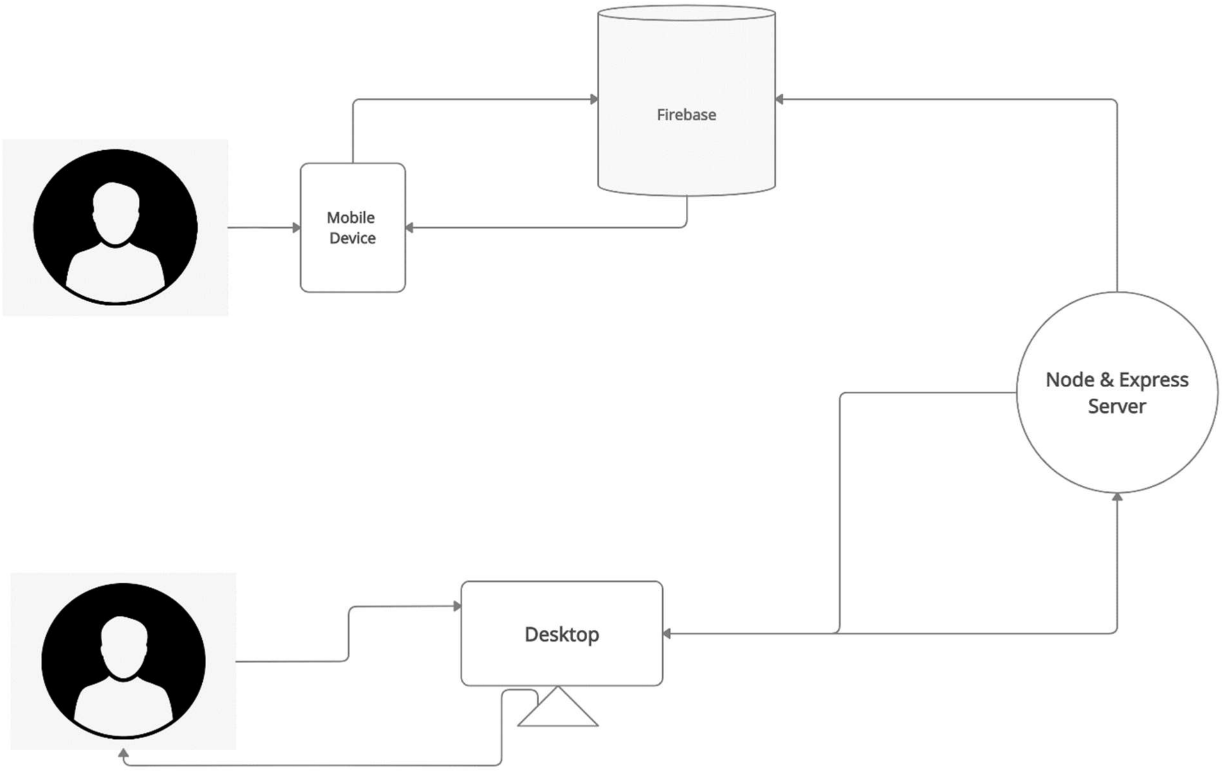


Figure 3. 9: Architecture of the system

3.3.2 Front-end implementation

a. HTML

HTML is the markup project that was used to develop the structure of the web pages and their content in this project. The version of HTML used in the project work is HTML5.

b. Cascading Style Sheet

It was used to style the view pages by accessing the style classes declared in the CSS files. The classes were made available by a declaration of the path of the style sheet in the head tag, using a link tag.

c. Android Studio:

Android Studio is an integrated development environment (IDE) for developing android applications. This editor was used in the development of the mobile application, written in the kotlin language. The android emulator was also used to run the application. Android studio, supports the use of physical device emulators and it was used to scan the QR code on the card.

d. CodeScanner

This is a library for android based on ZXing. It is a dependency that was added to the android studio to develop the QR code scanner. It was used to customize the autofocus and flashlight control, the back-facing camera. It allowed the layout to be customized with a custom CodeScanner layout. The views were implemented with a CodeScannerView.

e. Kotlin

Kotlin is a modern open-source programming language used in Android studio, JavaScript, JVM, and NativeScript. This language was used to code the mobile application that would be used to scan the QR codes on the student's ID cards. The plugin version that was used for system implementation is 1.3.72-release-Studio4.1-5.

f. Visual studio code

Visual studio code is a code editor that supports development operations like debugging, task running, and version control. This editor was used in the development of the web application portion of this project.

g. JavaScript

JavaScript is a text-based programming language used for both client-side and server-side, that allows for the development of an interactive web page. It is used for both frontend and backend implementation across the web development stack. It was used in this project as a scripting language.

h. QR code.js

QRCode.js is a cross-browser JavaScript library that is used for automated QR code generation in HTML canvas. It allows its scripts to be used to generate QR codes either from the client-side or the server and exports the QR code in image format.

i. Firebase

Firebase is a Backend as a Service (BaaS) developed by Google to enable developers to develop applications. It was used in this project to manage the database of the application.

j. Node JS

Node is a type of JavaScript used for backend development. The Node JS framework allows the developer to handle data updates from the frontend and build scalable network applications able to process many simultaneous user requests, amongst other things. In this project, node JS was used to develop the backend.

k. Express JS

Express JS is a free and open-source application framework for Node JS. It is used for designing and building web apps in a considerable amount of time. It is the backend component of the MEAN stack.

CHAPTER FOUR

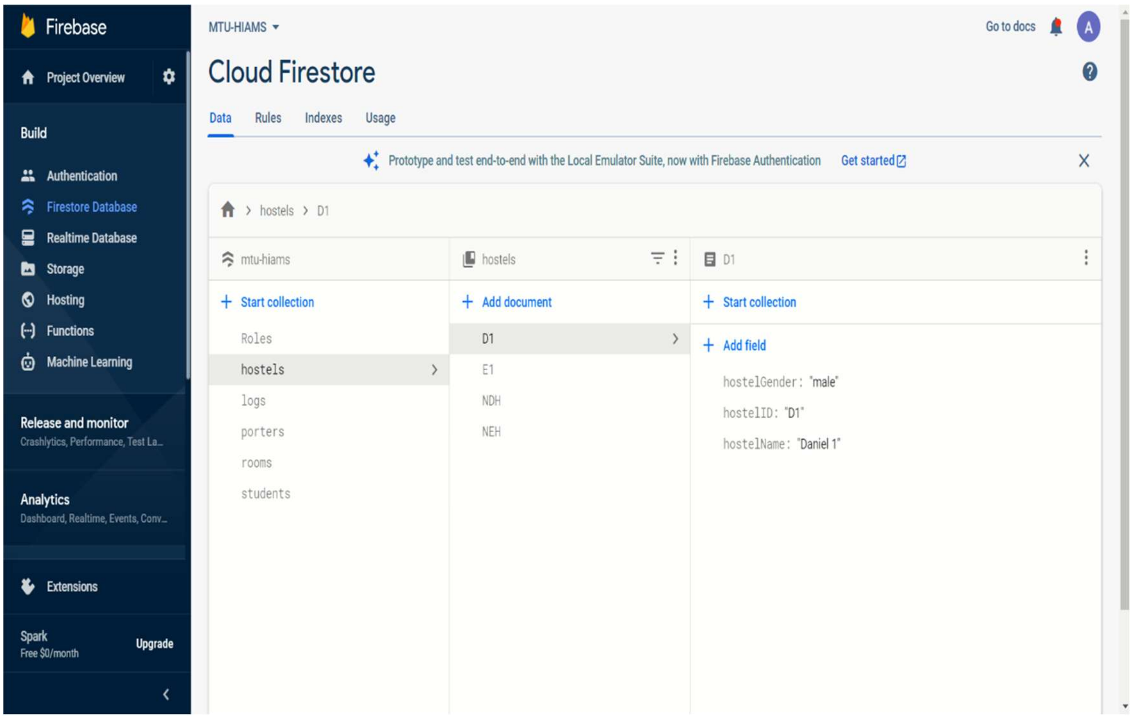
IMPLEMENTATION AND RESULT

This chapter presents the result of the Identity and Access management system and a description of the results obtained. This section covers the result of the database, implemented with Firebase, and the front-end implementation of the identity and access management system, implemented using the following web technologies, HTML, CSS, JavaScript. Also, the result mobile application of this system, developed using android studio, would be duly highlighted.

4.1 Result of Database Implementation of the Identity and Access Management System

Figure 4.1 shows the result of the collection for a hostel. The hostel collection consists of the HostelID field, HostelName field, and HostelGender field, which is used to hold gender managed by the system. Figure 4.2 shows the result of the Logs collection. The logs collection has fields for LogID, LogDate, Hostel. It is used to hold records of hostel activity. Figure 4.3 shows the result of roles collection, which consists of fields for roleName, and roleID.

Figure 4.4 shows the result of the rooms collection. This collection consists of the field name; roomID, room number, and hostel. It holds the record of the number of rooms in the hostel. Figure 4.5, shows the result of the student's collection. This table consists of a list of documents, which include, fullname, gender, hostel, room number, department, level, IsStudentInHostel, studentImage, matric number. It holds the record of the student that would be displayed. Figure 4.6, shows the result of porter's collection in the database. It includes the fields for fullname, staffID, email, role.



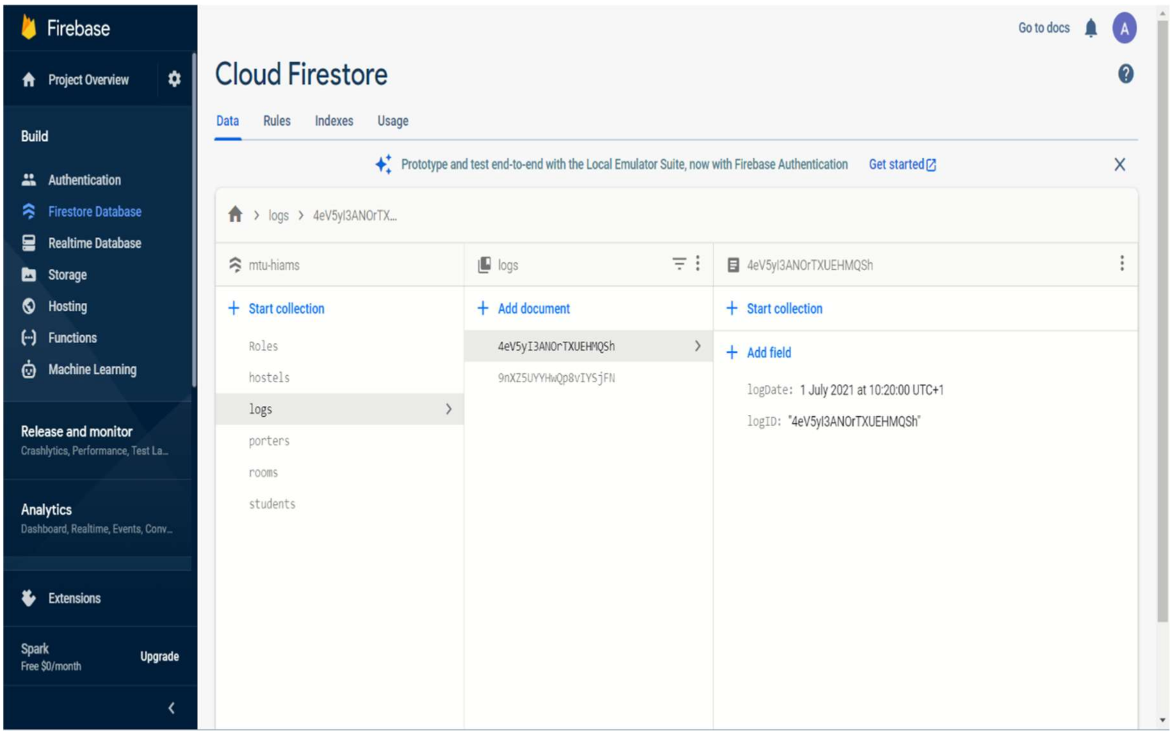
mtu-hiams: Roles, hostels, logs, porters, rooms, students

hostels: D1, E1, NDH, NEH

D1:

hostelGender: "male"
hostelID: "D1"
hostelName: "Daniel 1"

Figure 4.1: Hostel table



mtu-hiams: Roles, hostels, logs, porters, rooms, students

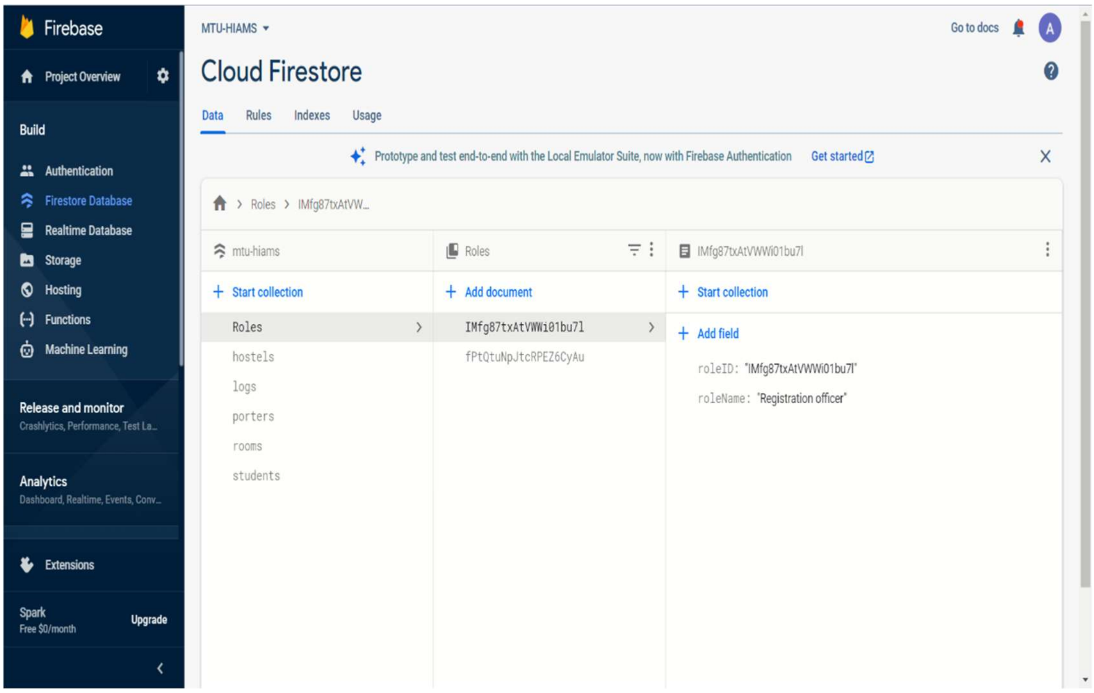
logs: 4eV5yI3ANOrTXUEHMQSh, 9nXZ5UYYHwQp8vIYSjFN

4eV5yI3ANOrTXUEHMQSh

logDate: 1 July 2021 at 10:20:00 UTC+1

logID: "4eV5yI3ANOrTXUEHMQSh"

Figure 4.2: Logs table

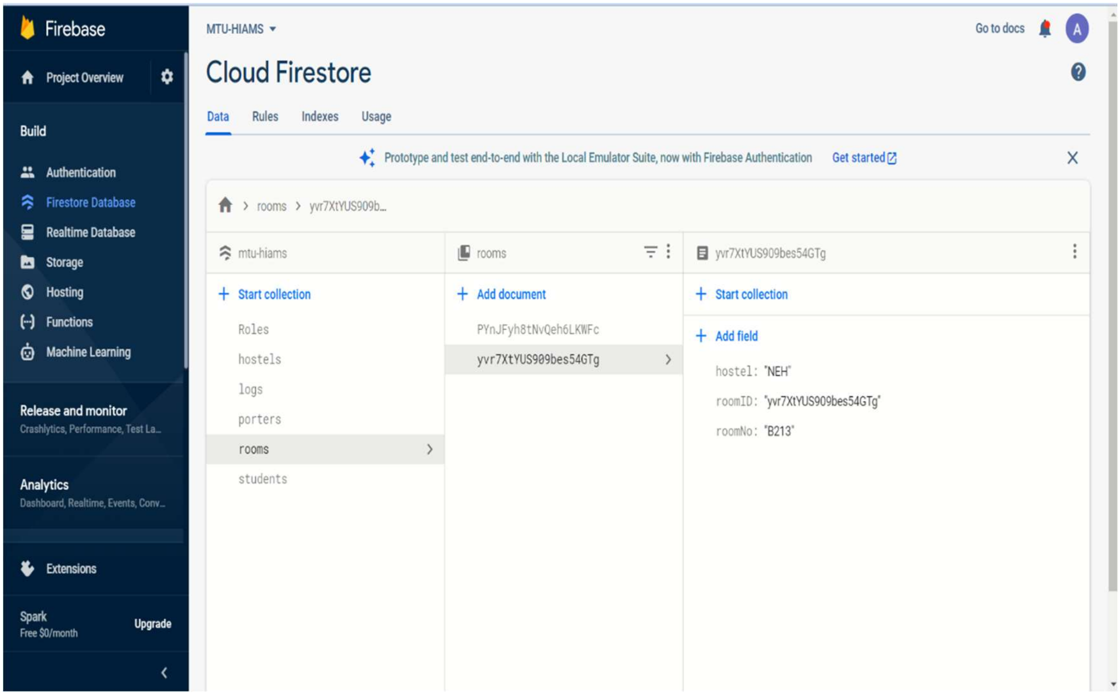


Roles
hostels
logs
porters
rooms
students

IMfg87txAtVWWi01bu7l
fPtQtuNpJtcRPEZ6CyAu

roleID: "IMfg87txAtVWWi01bu7l"
roleName: "Registration officer"

Figure 4.3: Roles table

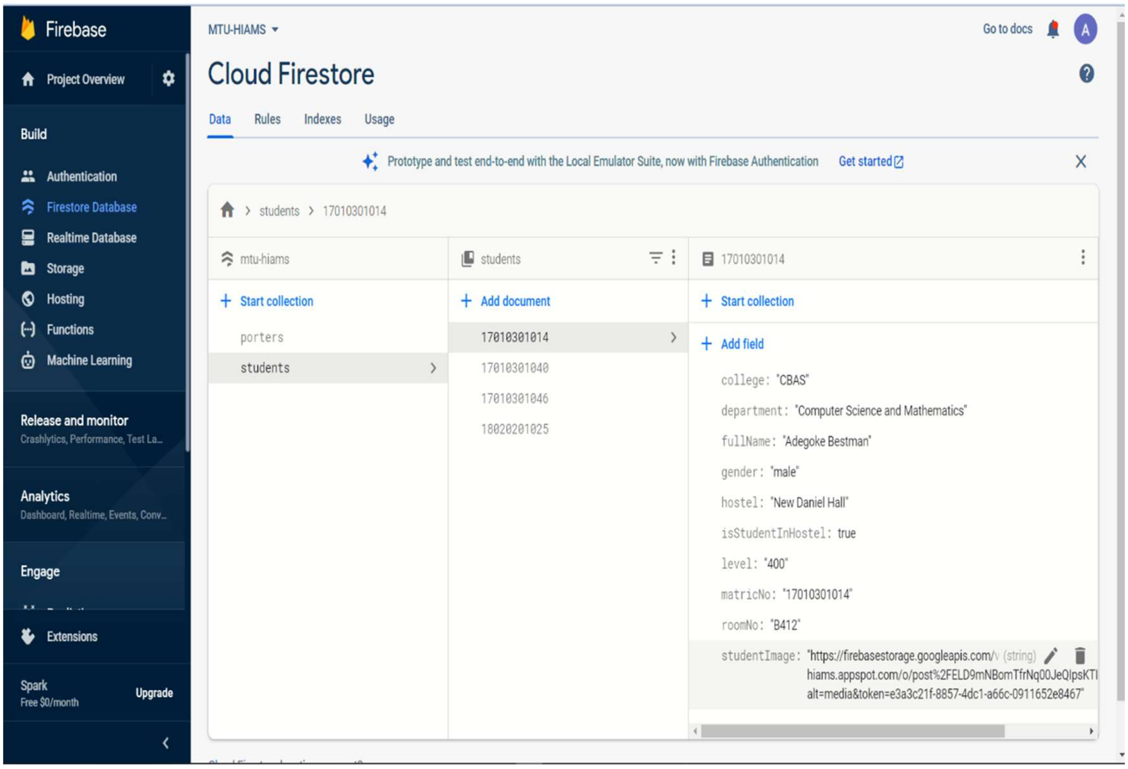


Roles
hostels
logs
porters
rooms
students

PYnJFyh8tNvQeh6LKWFc
yvr7XtYUS909bes54GTg

hostel: "NEH"
roomID: "yvr7XtYUS909bes54GTg"
roomNo: "B213"

Figure 4.4: Rooms table



college: "CBAS"

department: "Computer Science and Mathematics"

fullName: "Adegoke Bestman"

gender: "male"

hostel: "New Daniel Hall"

isStudentInHostel: true

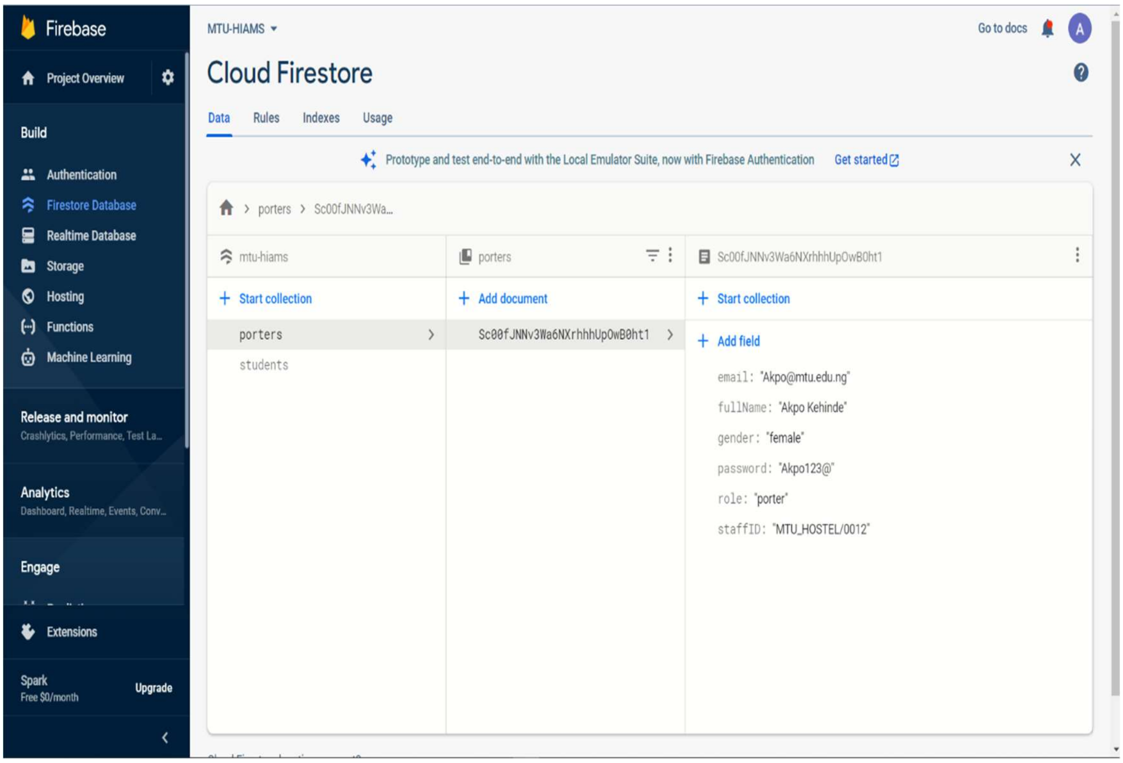
level: "400"

matricNo: "17010301014"

roomNo: "B412"

studentImage: "https://firebasestorage.googleapis.com/v (string) hiams.appspot.com/o/post%2FELD9mNBomTfrNq00JeQlpsKTI alt=media&token=e3a3c21f-8857-4dc1-a66c-0911652e8467"

Figure 4.5: Student collection



```
email: "Akpo@mtu.edu.ng"
fullName: "Akpo Kehinde"
gender: "female"
password: "Akpo123@"
role: "porter"
staffID: "MTU_HOSTEL/0012"
```

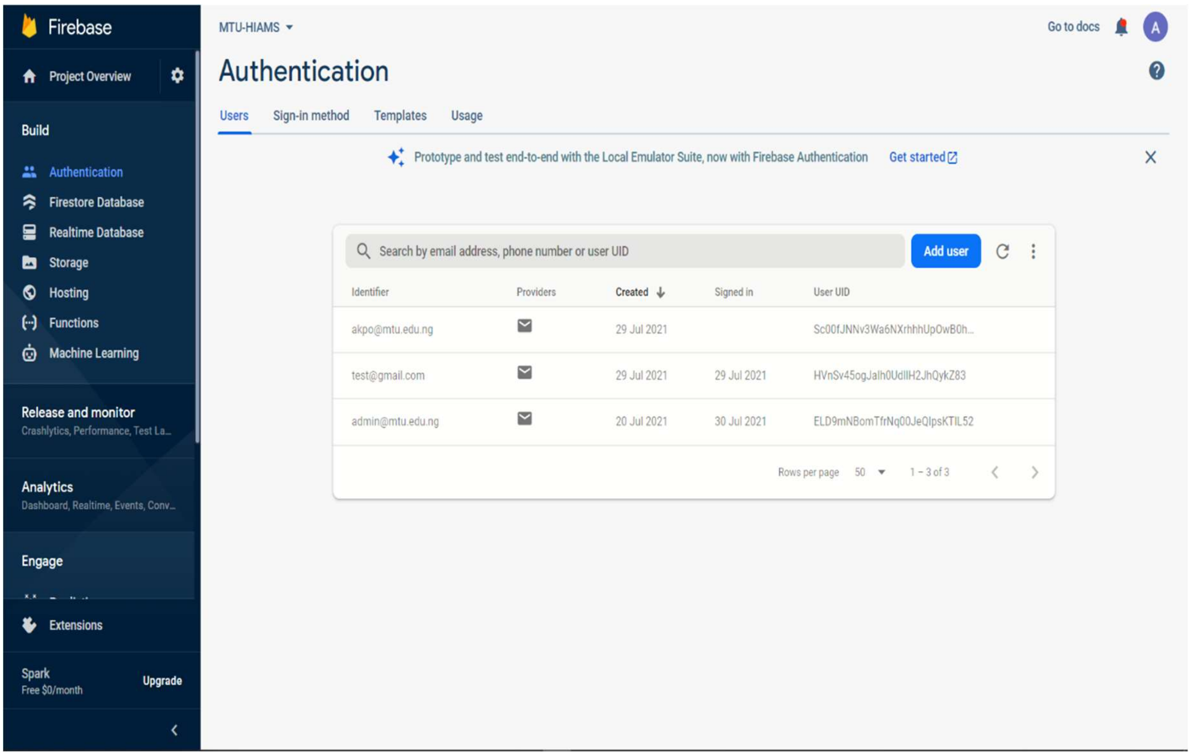
Figure 4.6: Porter's collection

This table holds the records of porters in the database including porters and head porters. Figure 4.7. shows the list of authenticated users in the database.

4.2 Result of Front-end Implementation of the Identity and Access Management System

Figure 4.8 shows the result of the web interface where the users, that is, the admin (registration officer), head porter, porter can log in and view their role interface with their log-in details (email, password). Figure 4.9 shows the result of the web interface used by the porter role. This diagram displays the dashboard used by the porter to navigate the interface. It includes the bar that outputs the number of students in the hostel, those outside of the hostel, and students from other hostels. It includes a button that navigates porter to the view activity log and a view button that allows porter append reasons why some students may be out when the log of the way was taken. Porter can search for students on this interface, to determine if they are in or out.

Figure 4.10 shows the result of the view activity log interface. This interface displays all students of a particular hostel by their name, level, college, department, Room number, last scan, and status. Porter generates the log of the day from this page and can search for students. Figure 4.11 shows the result of the interface that shows porter how many students were not in the hostel as at when the log of the day was taken. It displays the name, level, college, department, Room number, last scan of a student, and an action column, where the porter can add a reason for the absence of a student in the hostel. Figure 4.12 shows the result of the interface that shows the admin dashboard. This page includes bars that show the number of students within the hostel and the number of students that are



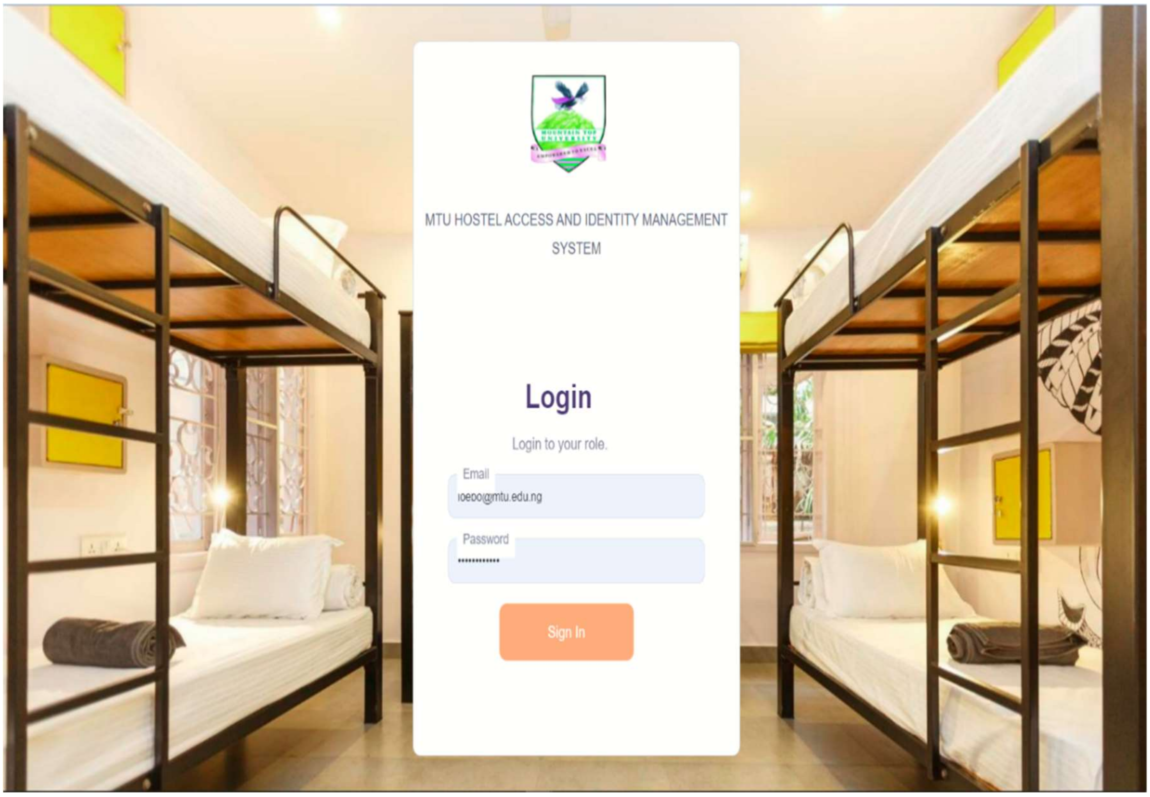
Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

Search by email address, phone number or user UID

Identifier	Providers	Created ↓	Signed in	User UID
akpo@mtu.edu.ng	✉	29 Jul 2021		Sc00fJNNv3Wa6NXrhhhUpOwB0h...
test@gmail.com	✉	29 Jul 2021	29 Jul 2021	HVnSv45ogJalh0UdlIH2JhQykZ83
admin@mtu.edu.ng	✉	20 Jul 2021	30 Jul 2021	ELD9mNBomTfrNq00JeQIpsKTIL52

Rows per page 50 ▾ 1 – 3 of 3

Figure 4.7: List of authenticated users



MTU HOSTEL ACCESS AND IDENTITY MANAGEMENT SYSTEM

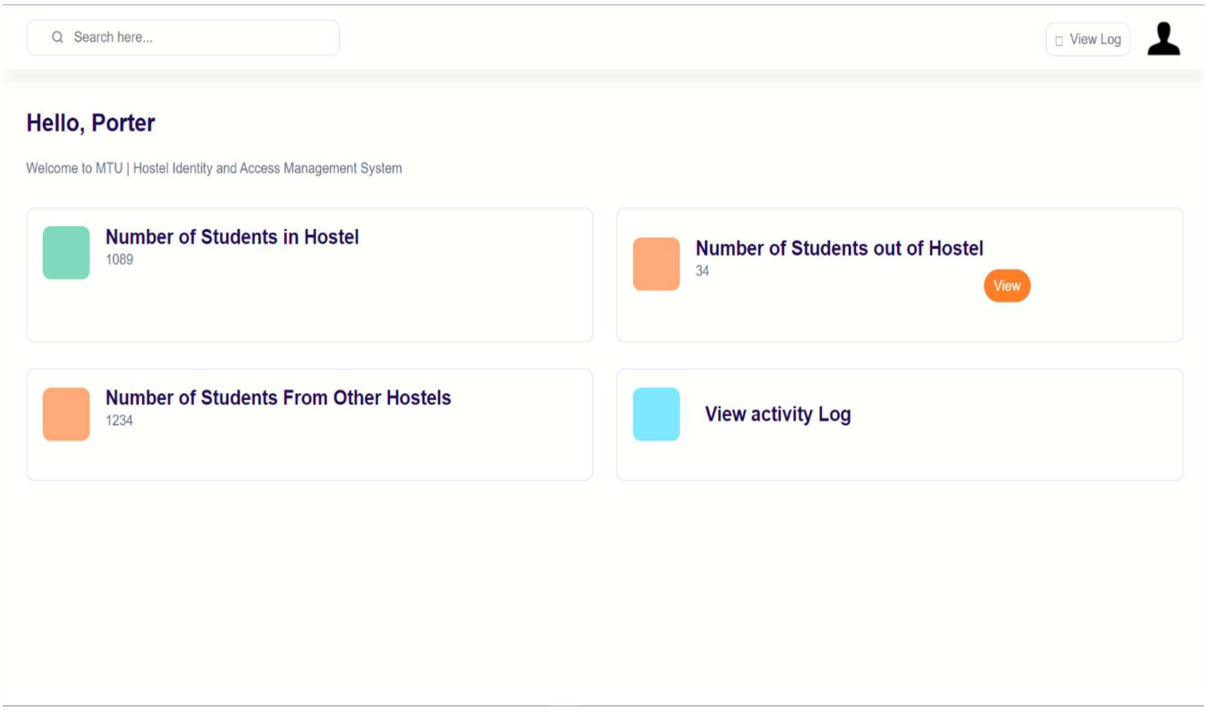
Login

Login to your role.

Email: ioebo@mtu.edu.ng

Password: ••••••••••••

Figure 4.8: Login Page for web application



Hello, Porter

Welcome to MTU | Hostel Identity and Access Management System

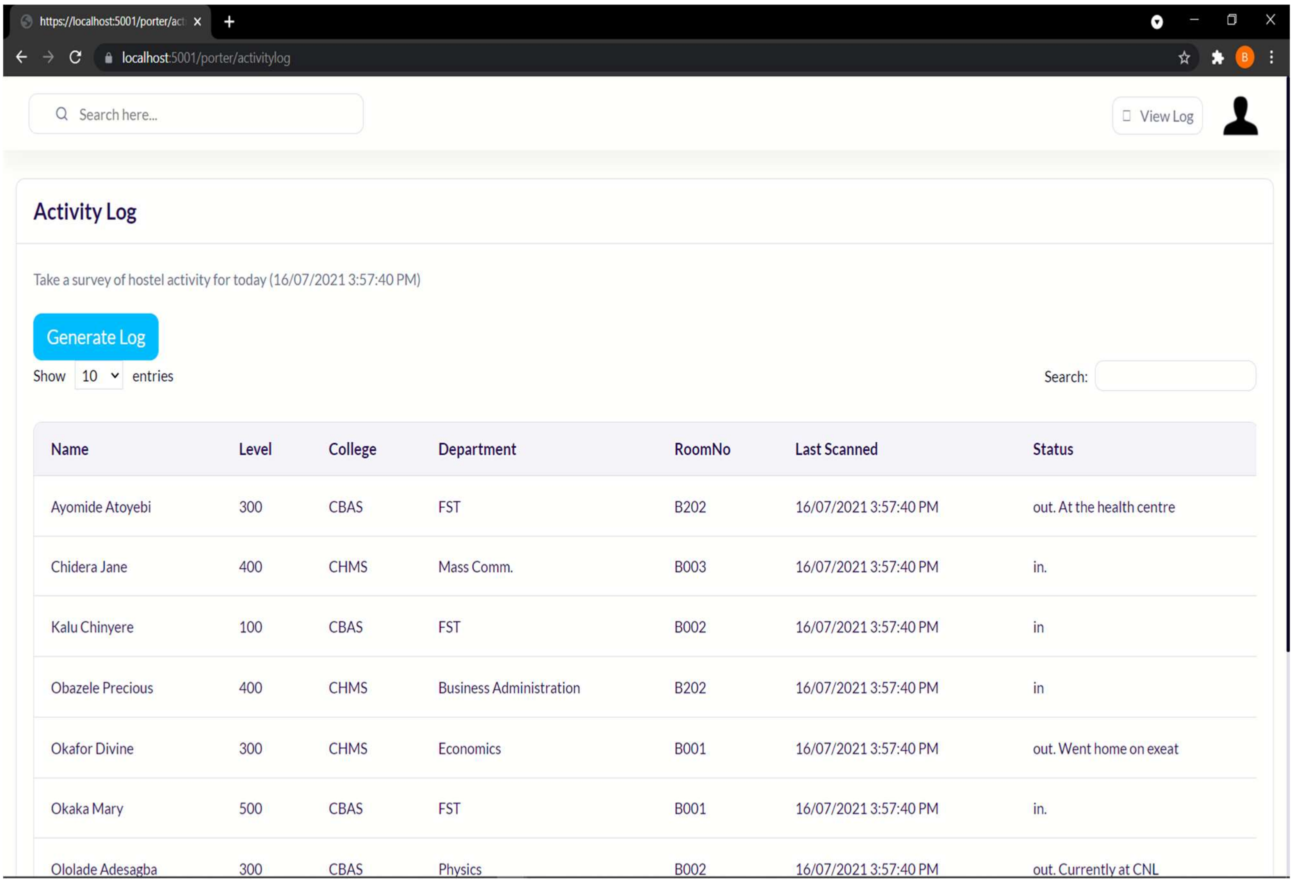
Number of Students in Hostel
1089

Number of Students out of Hostel
34

Number of Students From Other Hostels
1234

View activity Log

Figure 4.9: Dashboard for Porter role

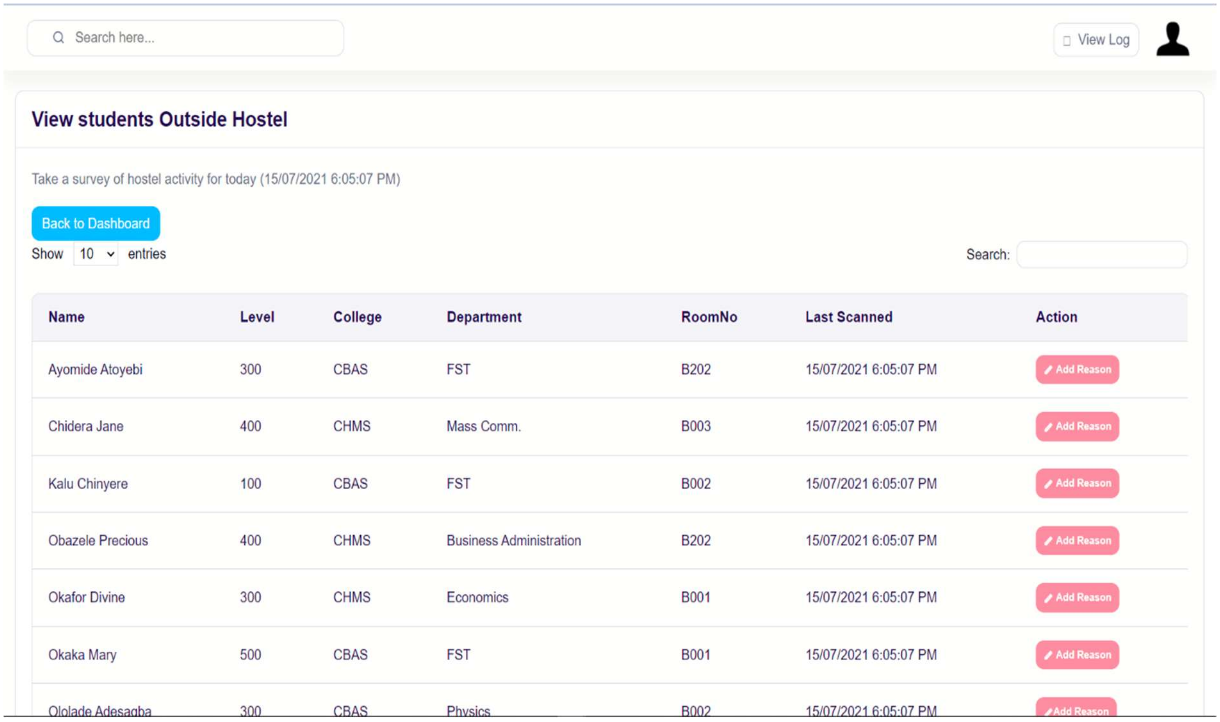

https://localhost:5001/porter/act
localhost:5001/porter/activitylog
Search here...
View Log
Generate Log
Show 10 entries
Search:

Activity Log

Take a survey of hostel activity for today (16/07/2021 3:57:40 PM)

Name	Level	College	Department	RoomNo	Last Scanned	Status
Ayomide Atoyebi	300	CBAS	FST	B202	16/07/2021 3:57:40 PM	out. At the health centre
Chidera Jane	400	CHMS	Mass Comm.	B003	16/07/2021 3:57:40 PM	in.
Kalu Chinyere	100	CBAS	FST	B002	16/07/2021 3:57:40 PM	in
Obazele Precious	400	CHMS	Business Administration	B202	16/07/2021 3:57:40 PM	in
Okafor Divine	300	CHMS	Economics	B001	16/07/2021 3:57:40 PM	out. Went home on exeat
Okaka Mary	500	CBAS	FST	B001	16/07/2021 3:57:40 PM	in.
Ololade Adesagba	300	CBAS	Physics	B002	16/07/2021 3:57:40 PM	out. Currently at CNL

Figure 4.10: View Activity Log on Porter role interface

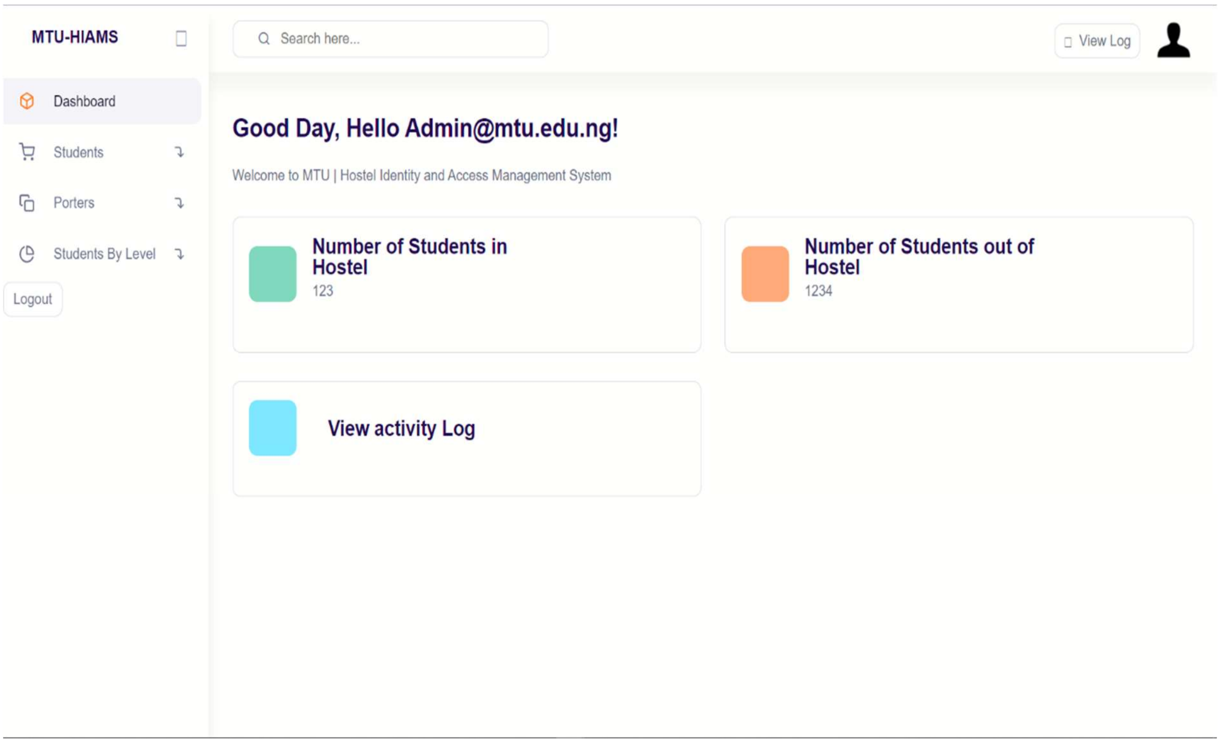


View students Outside Hostel

Take a survey of hostel activity for today (15/07/2021 6:05:07 PM)

Name	Level	College	Department	RoomNo	Last Scanned	Action
Ayomide Atoyebi	300	CBAS	FST	B202	15/07/2021 6:05:07 PM	Add Reason
Chidera Jane	400	CHMS	Mass Comm.	B003	15/07/2021 6:05:07 PM	Add Reason
Kalu Chinyere	100	CBAS	FST	B002	15/07/2021 6:05:07 PM	Add Reason
Obazele Precious	400	CHMS	Business Administration	B202	15/07/2021 6:05:07 PM	Add Reason
Okafor Divine	300	CHMS	Economics	B001	15/07/2021 6:05:07 PM	Add Reason
Okaka Mary	500	CBAS	FST	B001	15/07/2021 6:05:07 PM	Add Reason
Ololade Adesagba	300	CBAS	Physics	B002	15/07/2021 6:05:07 PM	Add Reason

Figure 4.11: View Student Outside Hostel page on Porter Role interface



Good Day, Hello Admin@mtu.edu.ng!

Welcome to MTU | Hostel Identity and Access Management System

Number of Students in Hostel
123

Number of Students out of Hostel
1234

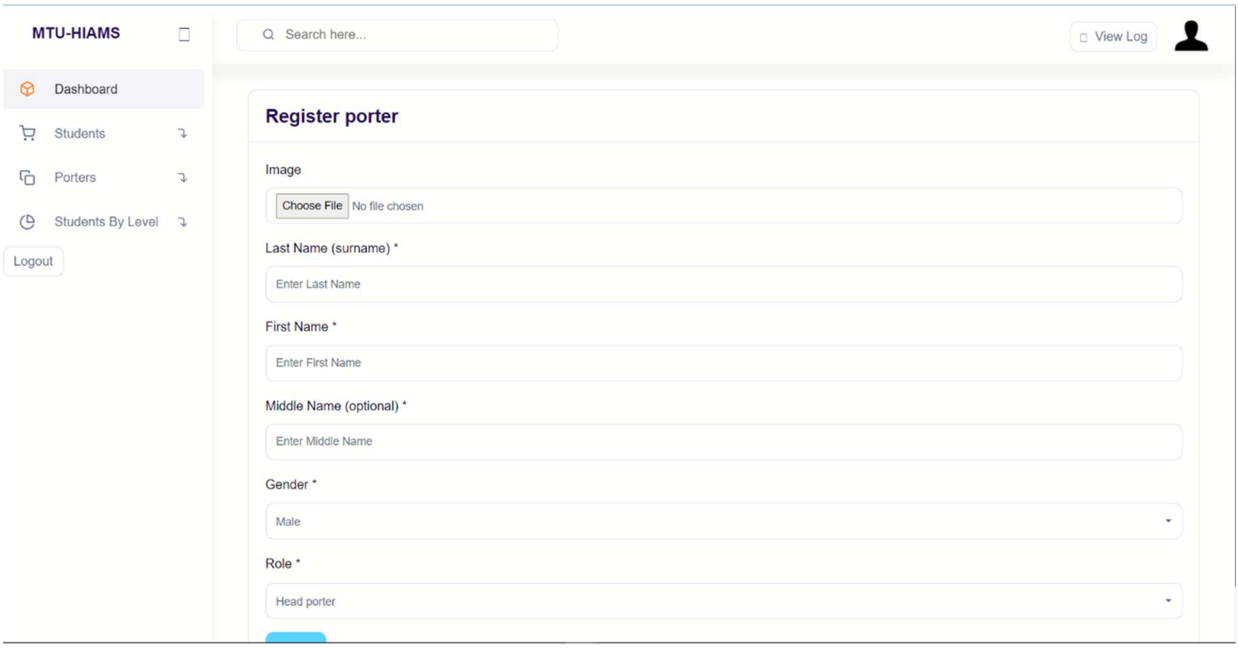
View activity Log

Figure 4.12: Dashboard for admin user role

outside of the hostel. It also includes a button that allows users to view the activity log of hostels. This page includes a navigation pane with the following information, registered students, view registered students, register students, view registered students, and a view of list of students by levels.

Figure 4.13 (a) and Figure 4.13 (b) show the result of the interface that displays the registration form for porter and head porter and the list of registered users. The registration form requests for porter image, full name, gender, role and registers them, while the view registered porters page displays the requested porter details on a table. The view registered porter interface also includes an option to delete a porter's record or assign a new role. Figure 4.14 (a) and shows the result of the interface that displays the registration form for students. The form requests the student's full name, matric number, college, department, college, the room allocated, level, image, gender, hostel name. Once the form is submitted, a QR code is automatically generated. Figure 4.14 (b) shows the result of the interface that displays a list of registered students. This page contains a table that holds the information used to register students and an option to generate a new QR code if need be.

Figure 4.15 shows the result of the interface that outputs the image of a student, their details, and their generated code. Figure 4.16, shows the result of the interface that is viewed by the admin and head porter. This interface displays an activity log that can be viewed by both the admin and the head porter. It contains a table that displays the name of each student, level, college, department, room number, last scan, and status. This interface allows them to download the log of the day. Figure 4.17 shows the result of the login page interface for the mobile application used by porters to scan students. Figure 4.18(a) and figure 4.18(b) shows the result of the interface of the mobile application before and after scanning of student's QR code.



Register porter

Image

Choose File No file chosen

Last Name (surname) *

Enter Last Name

First Name *

Enter First Name

Middle Name (optional) *

Enter Middle Name

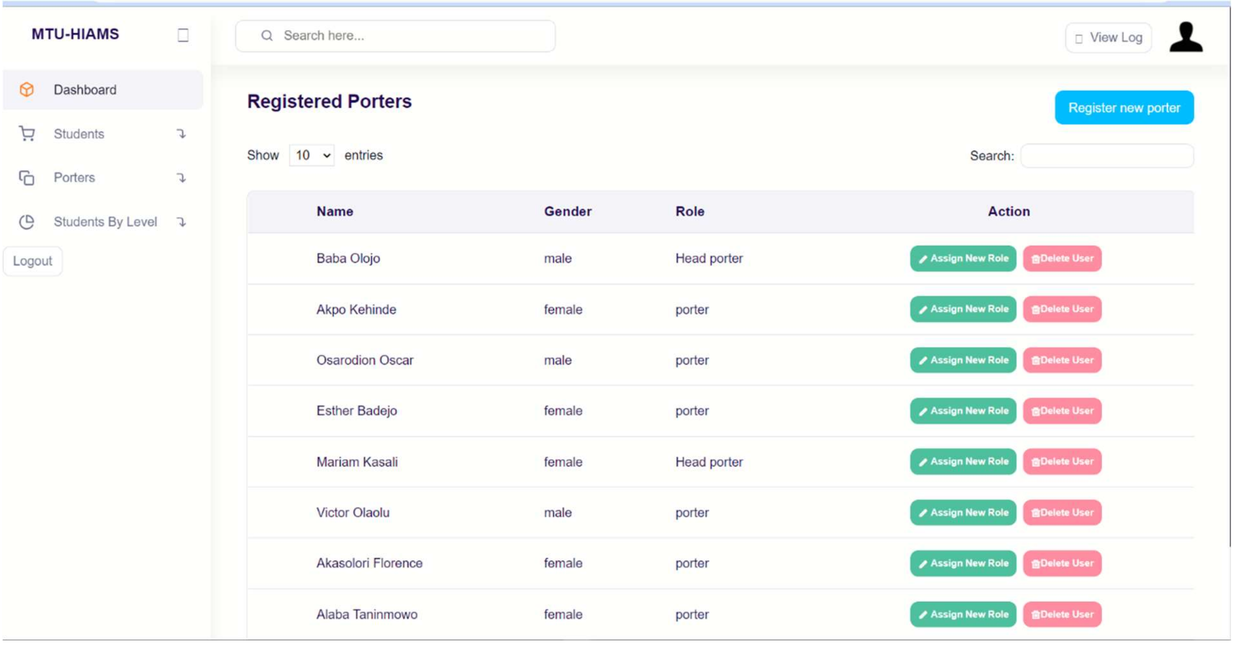
Gender *

Male

Role *

Head porter

Figure 4.13 (a): Registration of Porters page on the admin interface



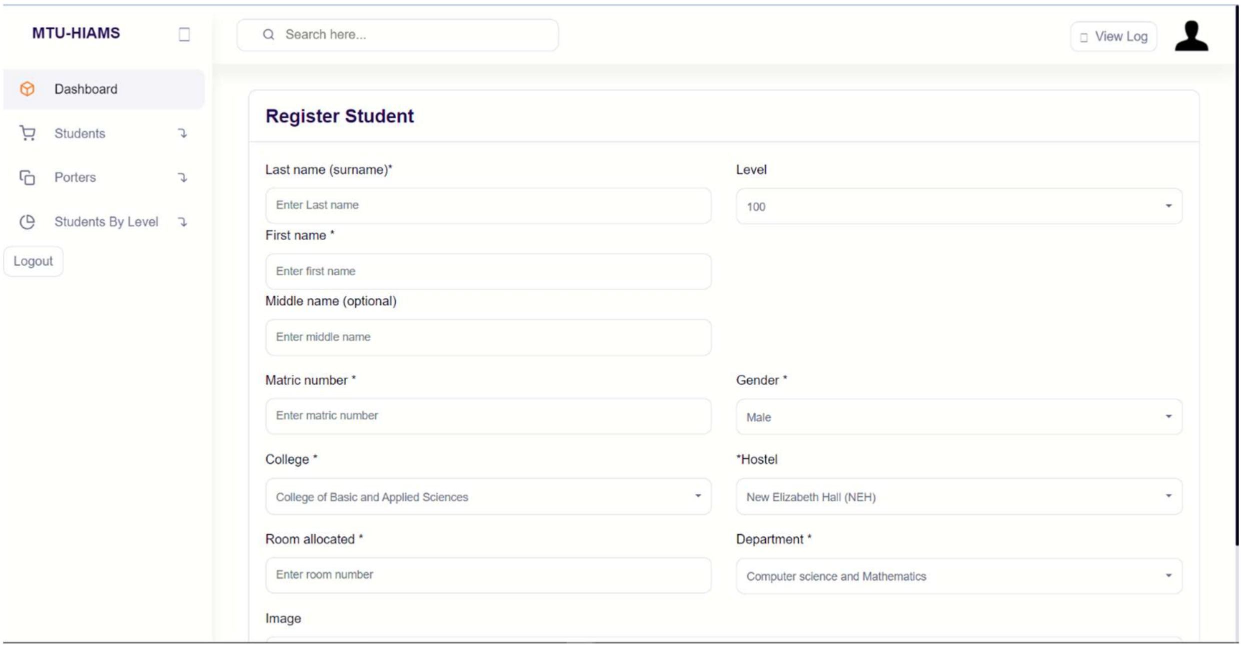
Registered Porters

Show 10 entries

Search:

Name	Gender	Role	Action
Baba Olojo	male	Head porter	Assign New Role Delete User
Akpo Kehinde	female	porter	Assign New Role Delete User
Osarodion Oscar	male	porter	Assign New Role Delete User
Esther Badejo	female	porter	Assign New Role Delete User
Mariam Kasali	female	Head porter	Assign New Role Delete User
Victor Olaolu	male	porter	Assign New Role Delete User
Akasolori Florence	female	porter	Assign New Role Delete User
Alaba Taninmowo	female	porter	Assign New Role Delete User

Figure 4.143 (b): View registered porter page on admin interface



Register Student

Last name (surname)*: Enter Last name

First name *: Enter first name

Middle name (optional): Enter middle name

Matric number *: Enter matric number

College *: College of Basic and Applied Sciences

Room allocated *: Enter room number

Image

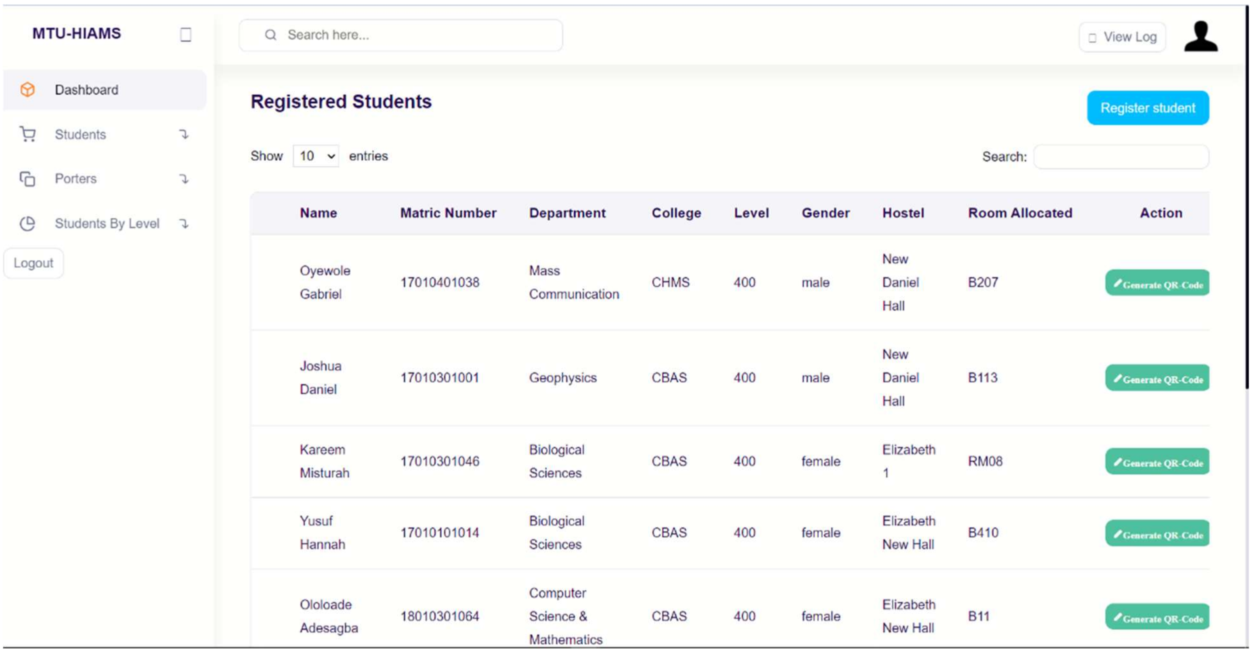
Level: 100

Gender *: Male

*Hostel: New Elizabeth Hall (NEH)

Department *: Computer science and Mathematics

Figure 4.154 (a): Registration of student page on the admin interface



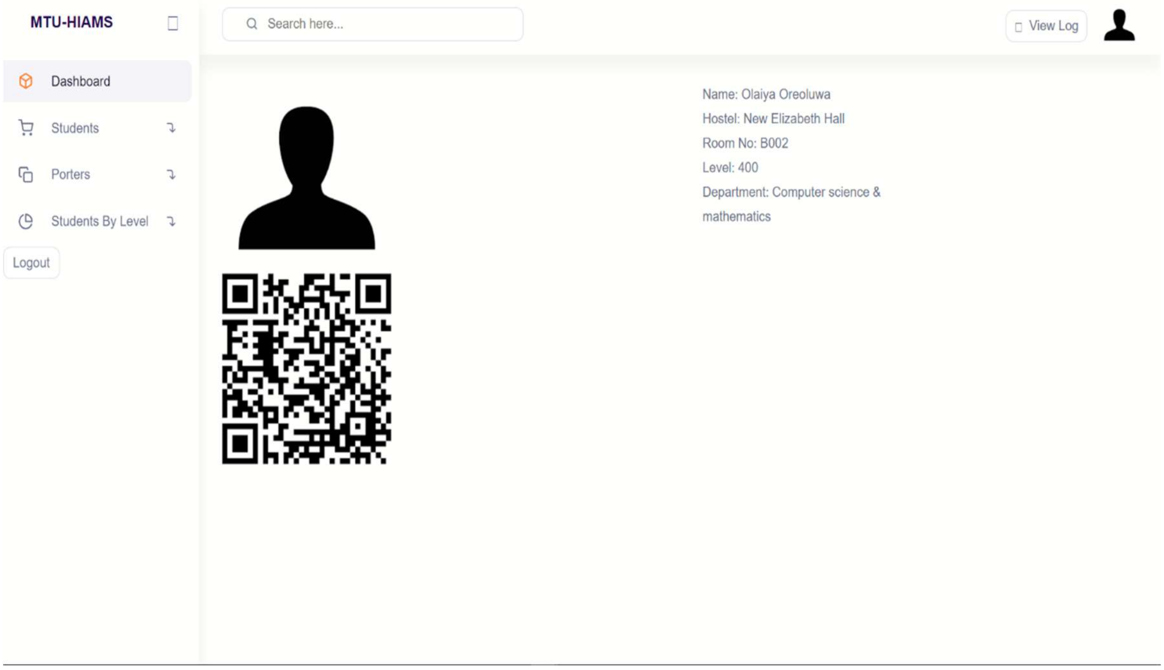
Registered Students

Show 10 entries

Search:

Name	Matric Number	Department	College	Level	Gender	Hostel	Room Allocated	Action
Oyewole Gabriel	17010401038	Mass Communication	CHMS	400	male	New Daniel Hall	B207	Generate QR-Code
Joshua Daniel	17010301001	Geophysics	CBAS	400	male	New Daniel Hall	B113	Generate QR-Code
Kareem Misturah	17010301046	Biological Sciences	CBAS	400	female	Elizabeth 1	RM08	Generate QR-Code
Yusuf Hannah	17010101014	Biological Sciences	CBAS	400	female	Elizabeth New Hall	B410	Generate QR-Code
Ololoade Adesagba	18010301064	Computer Science & Mathematics	CBAS	400	female	Elizabeth New Hall	B11	Generate QR-Code

Figure 4.164 (b): View registered student page on the admin interface



Name: Olaiya Oreoluwa

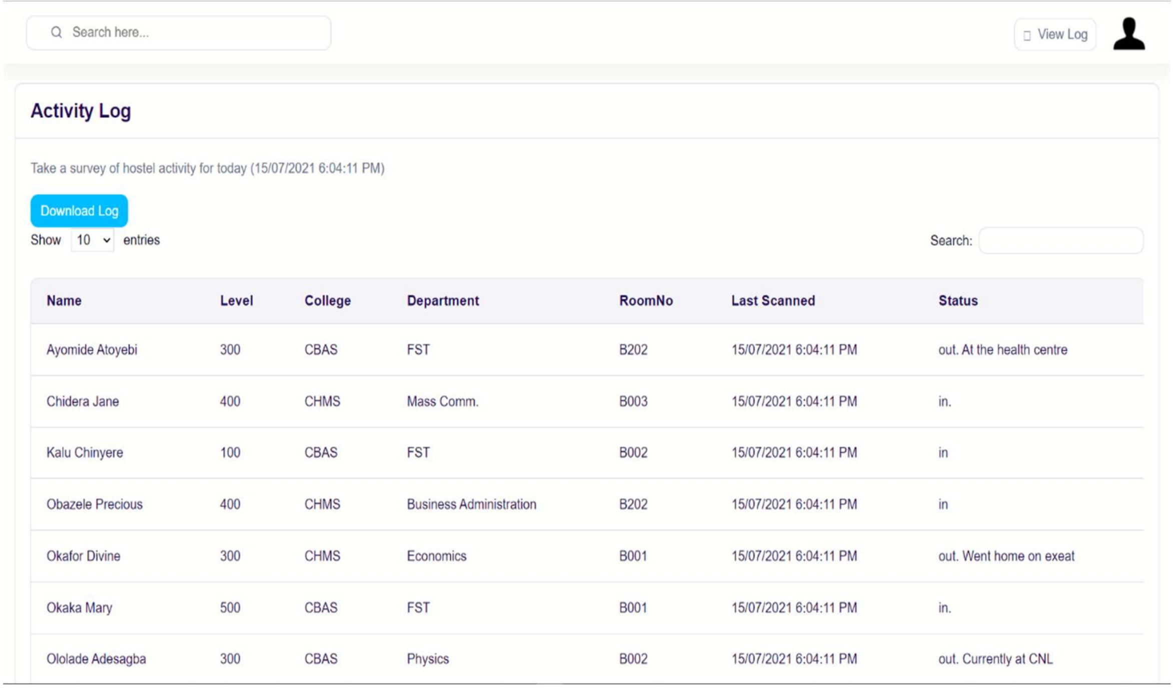
Hostel: New Elizabeth Hall

Room No: B002

Level: 400

Department: Computer science & mathematics

Figure 4.175: Student QR code and details

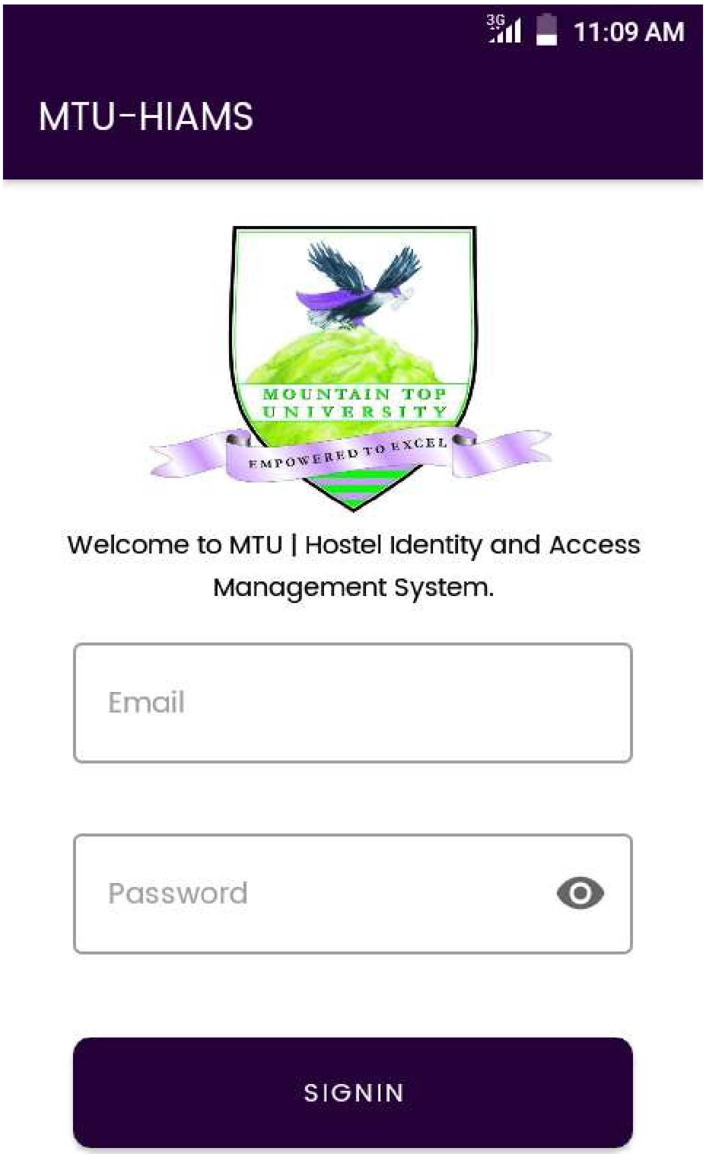


Activity Log

Take a survey of hostel activity for today (15/07/2021 6:04:11 PM)

Name	Level	College	Department	RoomNo	Last Scanned	Status
Ayomide Atoyebi	300	CBAS	FST	B202	15/07/2021 6:04:11 PM	out. At the health centre
Chidera Jane	400	CHMS	Mass Comm.	B003	15/07/2021 6:04:11 PM	in.
Kalu Chinyere	100	CBAS	FST	B002	15/07/2021 6:04:11 PM	in
Obazele Precious	400	CHMS	Business Administration	B202	15/07/2021 6:04:11 PM	in
Okafor Divine	300	CHMS	Economics	B001	15/07/2021 6:04:11 PM	out. Went home on exeat
Okaka Mary	500	CBAS	FST	B001	15/07/2021 6:04:11 PM	in.
Ololade Adesagba	300	CBAS	Physics	B002	15/07/2021 6:04:11 PM	out. Currently at CNL

Figure 4.186: Activity log page for admin and Head porter



Welcome to MTU | Hostel Identity and Access Management System.

Figure 4.197: Login page for mobile application

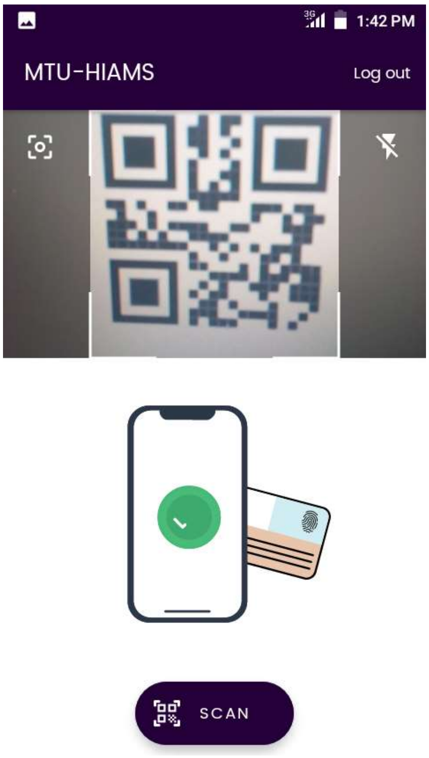


Figure 4.208 (a): Interface of mobile application before scanning

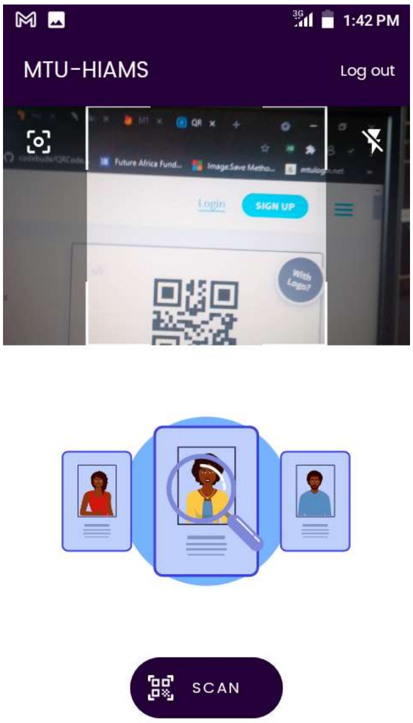


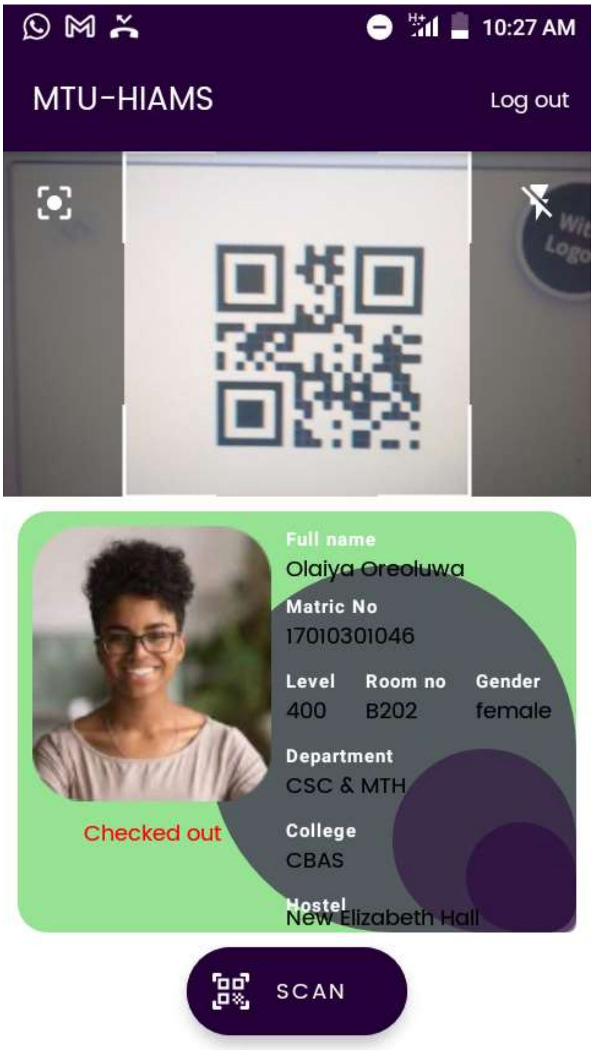
Figure 4.218 (b): Interface of mobile application after scanning

It traverses through the database in search of the details of the student being scanned. Figure 4.19, shows the result of the interface of the mobile application that displays student's details after they are scanned and displays their status (check-in or checked out).

4.3 Discussion of Results

The result of the study on the topic of Identity and Access Management systems presented expected results based on the objectives that were initially stated on the topic. The results of the identification of user and system requirement aided the definition of the system users been the administrative registration officer, head porters, and porters respectively. The result revealed that the administrative registration officer is responsible for the registration of students and consequent generation of QR code for each student as well as registration of porters and head porters and the assignment of roles. The result shows that the porters are responsible for the scanning of QR codes using the mobile interface of the system to validate student access into the hostel and exit out of the hostel. The porters are also responsible for the generation of logs for the hostel to which they are assigned to while the head porter is responsible for the oversight function within the system such as log download and review.

The result also revealed that the methods adopted in this study adequately cater to the user and system requirement initially specified for the system with a unique interface for each user role and a database that handles records for all the involved entities in the system and their various operations. Hence it is safe to say that the result of this system presents a solution to the manual method of access control and identity management within the hostels.



10:27 AM

MTU-HIAMS

Log out

Full name
Olaiya Oreoluwa

Matric No
17010301046

Level	Room no	Gender
400	B202	female

Department
CSC & MTH

Checked out

College
CBAS

Hostel
New Elizabeth Hall

Figure 4.229: Interface displaying student details and status.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

This study developed a hostel identity and management system that allows the administrative registration officer, head porters, and porters to effectively monitor access of students as well as identify each student in the hostel per time without having to manually perform headcounts. The user and system requirements that were necessary to be met by the system were identified alongside the software and hardware requirements of the system during this study. The requirements were also specified using UML diagrams such as use case diagram, activity diagram, sequence diagrams, and a class diagram for user requirement and system modeling respectively. The system was developed using HTML, CSS, and JavaScript for the frontend as well as Firebase for the database management on the system.

5.2 Conclusion

In conclusion, this study has designed and implemented a hostel identity and access management system that solves the challenges of access and student identity management within the hostels. The study was able to identify the various constraints that characterized this process and by this define the system and user requirements. The designs of this system were also specified adequately with relevant UML diagrams so as suit to the expected functions of the proposed system.

5.3 Recommendation

The study recommends that future works in this area be done to integrate various hostel management functions such as room allocation, damages, and repair management, etc. into the system developed in this study to create a robust unified platform that can handle all operations about hostel management.

References

Abdullahi, Y. E., & Ogwueleka, F. N. (2017, August). A Software System Development Life Cycle Model for Improved Students' Communication and Collaboration. *International Journal of Computer Science & Engineering Survey (IJCSES), 8*(4), 1-10.

Adebisi, O., Oletubo, A., Alade, T., & Ekpekpe, A. (2017). Perspectives of Students on Private Hostel Facilities in Proximity to the Federal University of Technolgy, Akure, Nigeria. *Journal of Poverty, Investment and Development, 33*(1), 31-36.

Adeoti-Adekeye, W. B. (1997). The importance of management information systems. *Library Review, 46*(5), 318-327.

Adetoba, B. T., & Ogundele, I. O. (2018, October). Requirements Engineering Techniques in Software Development Life Cycle Methods: A Systematic Literature Review. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 7*(10), 733-743.

Appiah, M. (2016, May). Effects Of Services Quality On Customer Satisfaction: A Case From Private Hostels In Wa-Municipality Of Ghana. *British Journal Of Marketing, 4*(3), 48-54.

Ayaka, A. (2020). *Projects.Ng*. Retrieved From Design And Implementation Of Online Hostel Management System: https://projects.ng/project/design-and-implementation-of-online-hostel-management-system/

Ayanlowo, A., Shoewu, O., Olatinwo, S., Omitola, O., & Babalola, D. (2014, October 1). Development of an Automated Hostel Facility Management System. *Journal of Science and Engineering, 5*(1).

Ayanlowo, k., Shoewu, O., Omitola, O. O., & Babalola, D. D. (2014). Development of an Automated Hostel Facility Management System. *Journal of Science and Engineering, 5*(1), 01-10.

Barjtya, S., Sharma, A., & Rani, U. (2017, July 7). A detailed study of Software Development Life Cycle (SDLC) Models. *International Journal Of Engineering And Computer Science, 6*(7), 22097-22100. doi:10.18535/ijecs/v6i7.32

Beal, V. (2021, April 7). *webopedia*. Retrieved from What is IAM?: https://www.webopedia.com/definitions/iam/

Beck, K. (2000). *Extreme programming explained: Embrace change.* MA: Addison Wesley Longman, Inc.

Best, D. P. (1988, March). The future of information management. *International Journal of Information Management, 8*(1), 13-24.

Bindia, T. (2019, December 15). A Review on Models of Software Development Life Cycle. *CITS, 3*, 60-68. doi:10.31058/j.data.2019.34002

Bond, J. (2021, February 12). *The Benefits of Using QR Codes*. Retrieved from Visual Marketing Australia: https://www.visualmarketing.com.au/the-benefits-of-using-qr-codes/

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide (2nd ed.)*. MA: Addison Wesley.

creately. (2021, July 8). *The Easy Guide to UML Deployment Diagrams*. Retrieved from creately.com: https://creately.com/blog/diagrams/deployment-diagram-tutorial/

Desmond, E. (2005, July). *Introduction to Identity Management.* Retrieved from Oracle.com:

https://docs.oracle.com/cd/B14099_19/idmanage.1012/b14084/intro.htm#:~:text=An%20identity%20management%20system%20can,devices%2C%20processes%2C%20and%20applications.

Duff, W. M., & Asad, M. C. (1980). *Information Management: An Executive Approach.* London: Oxford University Press.

Ellis, D. (1986). Information management and information work. *International Journal of Information Management, 6*(2), 15-26.

Er. Singh, D., & Dr. Sidhu, H. S. (2018, June). A Scrutiny Study Of Various Unified Modeling Language (Uml) Diagrams, Software Metrics Tool And Program Slicing Technique. *Journal Of Emerging Technologies And Innovative Research (Jetir), 5*(6), 254-262.

Ergasheva, S., & Kruglov, A. (2007, January). Software Development Life Cycle Early Phases And Quality Metrics: A Systematic Literature Review. *Information Technologies, Telecommunications And Control Systems, 1694*(2020). Doi:10.1088/1742-6596/1694/1/012007

Ezeigweneme, C., & Egolum, C. (2020, Jan). Analysis Of Challenges In Managing Students' Hostel Facilities In Nnamdi Azikiwe University, Awka Anambra State, Nigeria. *Iconic Research And Engineering Journals, 3*(7), 80-92.

Ferdous, S., & Poet, R. (2012). A Comparative Analysis of Identity Management Systems. *International Conference on High Performance Computing and Simulation (HPCS).* doi:10.1109/HPCSim.2012.6266958

Gary, B., Shelly, T. J., Cashman, M. E., & Vermaat. (1999, March 8). *Discovering Computers 2000.* Course Technology Ptr (Sd). 1st edition.

Harish, R., & Syan, M. (2012). Ananlysis of Agile and Traditional Approach for Software development. *International Journal of Latest Trends in Engineering and Technology, 1*(4), 1-3.

Hasan, Y., Shamsuddin, A., & Aziati, N. (2013). The Impact of Management Information Systems adoption in Managerial Decision Making: A Review. *The International Scientific Journal of Management Information Systems, 8*(4), 010-017.

Heidarkhani, A., Khomami, A. A., Jahanbazi, Q., & Alipoor, H. (2013). The Role of Management Information Systems (MIS) in Decision-Making and Problems of its Implementation. *Universal Journal of Management and Social Sciences, 3*(3), 78-89.

Jiujiu, Y. (2018). Research Process on Software Development Model. *IOP Conference Series: Materials Science and Engineering.* IOP Publishing Ltd. doi:10.1088/1757-899X/394/3/032045

Keng, S. (2010). An Analysis of Unified Modeling Language (UML) Graphical Constructs Based on BWW Ontology. *Journal of Database Management*, 1-8.

Khan, M., Shadab, S., & Khan, F. (2020). Empirical Study of Software Development Life Cycle and its Various Models. *International Journal of Software Engineering (IJSE), 8*(2), 16-26.

Khawas, C., & Shah, P. (2018, July). Application of Firebase in Android App Development-A study. *International Journal of Computer Applications, 179*(46), 49-53.

Khozaei, F., Ayub, N., Hassan, A. S., & Khozaei, Z. (2010). The factors predicting students' satisfaction with university hostels, case study Universiti Sains Malaysia. *Asian Culture and History, 2*(2), 148-158.

Kolawa, A., & Huizinga, D. (2007). Automated Defect Prevention: Best Practices in Software Management. *Wiley-IEEE Computer Society Press*, 75.

Kunz, M., Hummer, M., Fuchs, L., Netter, M., & Pernul, G. (2014). Analyzing Recent Trends in Enterprise Identity Management. *25th International Workshop on Database and Expert Systems Applications*, (pp. 273-277). doi:10.1109/DEXA.2014.62

Langemo, M. (1980). Records management/word processing – a needed team effort. *Records Management Quaterly, 14*(4), 10-14.

Laudon, K., & Laudon, J. (2006). Management Information Systems: Managing the Digital Firm, 9th ed. *International Journal Of Computers Communications & Control, 2*(1), 103-105. Doi:10.15837/Ijccc.2007.1.2342

Lippert, M., & Roock, S. (2001). Adapting Xp To Complex Application Domains. *European Software Engineering Conference, 8th Edition.*

Mburu, Z., Nderu, L., & Tobias, M. (2019). Review Of Digital Identity Management System Models. *International Journal Of Technology And Systems, 4*(1), 21-33.

Muhammed, K., Muhammed, S., Vinod, J., & Prasobh, G. (2009). *Hostel Management System.* Division Of Computer Engineering, Soe. Retrieved March 2021

Munassar, N. M., & Govardhan, A. (2010, September). Comparison Between Five Models Of Software. *International Journal Of Computer Science Issues (Ijcsi), 7*(5), 94-101.

Muthukumar, B., Albert, J., Gokul, N., & Denil, N. (2019). Qr Code And Biometric Based Authentication System For Trains. *International Conference On Frontiers In Materials And Smart System Technologies* (P. 590). Iop Publishing. Doi:10.1088/1757-899X/590/1/012010

Narayanan Sankara, A. (2012, July). QR Codes and Security Solutions. *International Journal of Computer Science and Telecommunications, 3*(7), 69- 72.

Nirgudkar, N., & Singh, P. (2017, May). The MEAN Stack. *International Research Journal Of Engineering And Technology (IRJET), 4*(5), 3237-3239.

O'brien, J. A. (2004). Management Information System. *Mc Graw Hill I 6th Ed.*

O'Brien, J. A., & Marakas, G. M. (2011). *Management Information Systems-10th Ed.* New York: Mcgraw-Hill/Irwin.

Ohaeri, N., & Omorojor, N. (2020, July-Sept). Managing Hostels For Sustainable Students' Academic Performance In Public Universities In Rivers State. *International Journal Of Innovative Social & Science Education Research, 8*(3), 139-147.

Okta. (2020, November 22). *What Is Identity Management And Access Control?* Retrieved May 3, 2021, From Oktane21: Https://Www.Okta.Com/Identity-101/What-Is-Identity-Management-And-Access-Control/

Padmanabhan, B. (2012, February 21). UNIFIED MODELING LANGUAGE (UML) OVERVIEW. *EECS810 – Principles Of Software Engineering*, 1-10.

Petrova, K., Romaniello, A., Dawn Medlin, B., & Vannoy, S. (2016). QR Codes Advantages And Dangers. *13th International Joint Conference on e-Business and Telecommunications (ICETE 2016)* , *2*, pp. 112-115. doi:10.5220/0005993101120115

Petrova, K., Romaniello, A., Medlin, B., & Vannoy, S. (2016). QR Codes Advantages and Dangers. *13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) . 2*, pp. 112-115. SCITEPRESS – Science and Technology Publications. doi:10.5220/0005993101120115

Rajkumar, G., & Sivagama, T. (2018). Hostel Management System Based on Finger Print Aunthentication. *Oriental Journal of Computer Science and Technology, 11*(4), 230-234. doi:10.13005/ojcst11.04.08

Ramdav, T., & Harinarain, N. (2018). The use and benefits os Quick Response Codes for construction materials in South Africa. *Acta Structilia, 25*(2), 94-117. doi:10.18820/24150487/as25i2.4

Reddy, G. S., Srinivasu, R., Rikkula, S. R., & Rao, V. S. (2009). Management Information Systems to help Managers for Providing decision making in an Organisation. *International Journal of Reviews in Computing*, 2076-3336.

Ritesh, B., Aman, K., Beesu, R., Utkarsh, A., Dr. Rajasimha, M., & Shilpa, D. (2018, June). Hostel Management System. *International Journal of Trend in Scientific Research and Development (IJTSRD), 2*(4), 856-862. Retrieved from www.ijtsrd.com

Robek, M. F., Brown, G. F., & Stephens, D. O. (1995). Information and Records Management: Document-Based Information Systems (4th Edition ed.). New York, Columbus: Mc Graw-Hill.

Rosencrance, L., & Mathias, C. (2020, November 25). *identity management(ID management)*. Retrieved from TechTarget: https://searchsecurity.techtarget.com/definition/identity-management-ID-management

Rouillard, J. (2008). Contextual QR Codes. *Third International Multi-Conference on Computing in the Global Information Technology.* Athens: ICCGI. doi:0.1109/ICCGI.2008.25

Sethi, M., & Sharma, A. (2013). Information System and System Development Life Cycle. 118-127. doi:10.4018/978-1-4666-3679-8.ch007

Shoewu, O., & Idowu, O. A. (2012, May). Development of Attendance Management System using Biometrics. *The Pacific Journal of Science And Technology, 13*(1).

Shoewu, O., Braimah, S. A., & Duduyemi, O. (2016, November). Design and Implementation of Hostel Management System (HOMASY): LASU as Case Study. *The Pacific Journal of Science and Technology, 17*(2), 189-196.

Shylesh, S. (n.d.). A Study of Software Development Life Cycle Process Models. Mangalore-575002, India.

Singh, S. H., & Singh, S. U. (2017, March). Study on Google Firebase for Website Development (the real-time database). *International Journal of Engineering Technology Science and Research, 4*(3), 364-367.

Sneha, S., & Rajendra, P. (2016, April). Hostel In Out Management and Monitoring System Using RFID, Face and Thumb Recognition. *International Journal of Innovative Research in Science, Engineering and Technology, 5*(4), 4978-4984. Retrieved 2021

Software Testing Help. (2021, May 30). *SDLC (Software Development Life Cycle) Phases, Process, Models*. Retrieved from Software Testing Help: https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/

Sommerville, I. (2011). *Software engineering / Ian Sommerville. — 9th ed.* Library of Congress Cataloging-in-Publication Data.

Stephen, B. U., Chukwudebe, G. A., & Ezenkwu, C. P. (2015, October). Integrated Identity And Access Management System For Tertiary Institutions In Developing Countries. *Nigerian Journal of Technology (NIJOTECH), 34*(4), 830-837.

Stonecash, J. C. (1981). The IRM showdown. *Infosystem, 28*(10), 42-48.

Sujit, K. D., & Pushkar, D. (2013, August). Software Development Life Cycle (Sdlc) Analytical Comparison And Survey On Traditional And Agile Methodology. *National Monthly Refereed Journal Of Research In Science & Technology, 2*(8), 22-30.

Suki, N., & Chowdhury, I. (2015). Students' Attitude and Satisfaction Living in Sustainable ON-Campus Hostels. *Malaysian Journal of Business and Economics, 2*(1), 35-47.

Thakur, A., Singh, D., & Chaudhary, V. (2015, March). Comparing Various SDLC Models On The Basis Of Available Methodology. *International Journal Of Modern Engineering Research (IJMER), 5*(3), 34-39.

Vidhate, B., Shah, T. A., & Kasture, B. A. (2018). A Study and Implementation of the Firebase Notification Using Chef's rent Method. *IJIRMPS, 6*(5), 49-53.

Vishwas, B. (2020, April 20). *The evolution of Identity Management Models and an introdcution to Self Soverign Identity(SSI)*. Retrieved from Imaginea Labs: https://labs.imaginea.com/the-evolution-of-identity-management-models-and-an-introduction-to-self-sovereign-identity-ssi/

Volunteer Forever. (2019, April 13). *what is an Hostel? An Overview and tips for International Travelers*. Retrieved from volunteerforever.com: https://www.volunteerforever.com/article_post/what-is-a-hostel-an-overview-and-tips-for-international-travelers/

Wave, D. (2016). History of QR Codes. Retrieved from www.qrcode.com/en/about/.

Wikipedia. (2021, January 29). *Identity-management system*. Retrieved from Wikipedia: The Free Encyclopedia: https://en.wikipedia.org/wiki/Identity-management_system#:~:text=Identity%20management%20(IdM)%20describes%20the,%2C%20downtime%2C%20and%20repetitive%20tasks.

Wu, S. J. (2015). *Improvement on Quality Testing Project of SI Software Based on Agile Scrum Mode.* Southwest Petroleum University.

Xiaocong, F. (2015). Chapter 6 - Fundamental UML Structural Modeling. In F. Xiaocong (Ed.), *Real-Time Embedded Systems: Design Principles and Engineering Practices* (pp. 131-184). Newnes. doi:10.1016/B978-0-12-801507-0.00006-7

Youssef, B. (2012, May). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (iJET), 2*(5). Retrieved from http://iet-journals.org/archive/2012/may_vol_2_no_5/255895133318216.pdf

Yusuf, M. (2014). The Impact of Management Information System (MIS) on the Performance of Business Organization in Nigeria. *International Journal of Humanities Social Sciences and Education (IJHSSE), 1*(2), 76-86.

Zhang, Y. S., & Li, X. (2006). Software Development Models: a Survey. *Journal of Computer Engineering and Applications*(3), 109-110.

Zoikoczy, P. (1981). Information Technology: An Introduction., (p. 157). London.