AUTOMATED HOSTEL MANAGEMENT SYSTEM

BY

OLORUNTOBA TOLU OLUWATOMILADE

(17010301022)

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science In Department Of Computer Science, Mountain Top University Ibafo

Ogun state,

Nigeria.

AUGUST, 2021

CERTIFICATION

This is to certify that OLORUNTOBA TOLU OLUWATOMILADE; 17010301022 undertook this research work AUTOMATED HOSTEL MANAGEMENT SYSTEM

and meets the requirements for submission to the Department of Computer science Mountain Top University, in partial fulfillment for the award of Bachelor of Science (B.Sc.) Computer Science Degree 2021.

…… (Signature and Date)

Dr. C. Igiri

Supervisor

…… (Signature and Date)

Matthew O. Adewole PhD

Ag. Head

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. Toluwalope Oloruntoba, and my mother, Mrs. Jumoke Oloruntoba for being a major source of support in every way.

ACKNOWLEDGEMENTS

I want to, first of all, thank the Almighty God for granting me good health and protection throughout my stay in Mountain Top University most especially in the course of this research work. Glory be to God.

This project was accomplished due to the involvement and assistance of a diverse group of individuals for their enormous engagement in my studies; God will undoubtedly recompense you and enable me not to disappoint you..

To my project supervisor, Dr.C. Igiri apart from accepting the noble task of supervising my project work, you have been like a mother to me, thanks for everything.

To all the lecturers of the Computer Science department. Thanks for imparting the best of your knowledge to me.

Though too, I desire to convey my appreciation. to Mr. Adeshina. O for his support, advice, and encouragement throughout my stay at Mountain Top University.

Additionally, I'd really would like to acknowledge my friends and coworkers. Oluwadamilare similoluwa, Ladapo Daniel, and Nnweneka Ebenezer for their friendship and encouragement.

To my siblings, Tosin & Fisayo always being there for me. Finally, to my lovely parents Mr. & Mrs. Oloruntoba O.T

ABSTRACT

HOSTEL MANAGEMENT SYSTEM" is technology created to oversee wide range of activities in the hostel, as the title indicates. The number of academic institutes has been steadily expanding in latest days. As a corollary, the number of dormitories offered to students enrolling at this university is expanding. As a result, the person in charge of the dormitory is put under great deal of strain, and computers are not frequently used in this scenario. This project focuses on the problems connected with facilities provided while avoiding the issues that arise when manual processes are used. These projects were carried out using agile approach.

The evaluation of the issues associated with the current structure leads to the design of a computerized method that is consistent with the present system and is more user-friendly and GUI focused. We can enhance the process performance and hence overcome the flaws in the current system.

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

4.2.1 Rectangle

4.2.2 Arrow

4.2.3 Circle

4.2.4 Open rectangle

CHAPTER 1

INTRODUCTION

1.1 Background to the study

The hostel is a term used to describe an establishment that provides inexpensive meals and housing to a certain group of people. It is also used to describe a place where students can stay when they are away from home. It is located on the school grounds and features large, well-ventilated dorms and single rooms. One of the most important responsibilities of the school administration is to provide safe and hygienic dormitory accommodations.

A lot of data needs to be kept in order to manage the hostel amenities, such as the number of students the hostel can hold, hostel rules and regulations, hostel cost, hostel in and out of students, guest and visitor records, and so on. As a result, a system that can record and analyze all types of data and information is required for the hostel's seamless operation. The data may be simply maintained by the hostel warden.

In schools, hostel management frequently entails overseeing all of the students' activities. All of these are still challenging and require some effort on the part of high management. Because of the manual system of tools, they utilize, handling hostel management roles and responsibilities in modern-day schools has always been a challenge. The Hostel Management System is well-designed.

Student allocation, hostel information, hostel application, and visitor management can all be aided by the Hostel Management System (HMS). In a nutshell, this method will help the hostel staff manage some of the activities.

1.2 Brief historical background of the university

The University

In Makogi Oba, Ogun State, Nigeria, a private university called Mountain Top University was established in 2015. It was developed by a faith believing Organization Mountain of Fire and Miracles Ministries. The university was established by Dr. D.K. Olukoya, the originator and general supervisor of MFM Ministries globally.

Colleges and Departments make up the University. The Colleges make up a crucial element of Mountain Top University even though they are governed by their own laws and regulations.

There are two Colleges, each of which consists of a collection of statutory bodies. Colleges of Basic and Applied Sciences, Humanities, and Management Sciences

Each college has a committee that is made up of representatives from each faculty. Members of the Group Design are from the schools**.**

1.3 Statement of the problem

The campus has 2 dormitories, one for school participants and one for students. The dormitory office now administers all of these hostels administratively. The confirmation of the Application form to the multiple data procedures is done by hand.

As a corollary, there are numerous duplicates that can be easily prevented. As a consequence, the individuals in charge of the hostel are subjected to a lot of pressure, and software is not typically used in this situation. This application addresses the issues related with hostel management while eliminating the problems that occur when legacy systems are being used.

1.4 Aim and objectives

The goal of the research is to create hostel management software that will handle Mountain Top University's hostel operations.

The study's key goals are as follows:

a. To create the system, identify and model the requirements specification.

b. Create a central database system that will function as a hostel database, containing all records relating to the hostel.

c. Allow students to apply for hostels using an online application.

d. To move away from manual student hostel administration methods

1.5 Scope of the Study

The focus of this research study is centered on the construction of a hostel administration system for Mountain Top University. This web-based project automates the student hostel application process, assigns rooms to students, notifies students of their application status whenever they log onto the portal and ensures the integrity of the information being processed by limiting access to only approved individuals through the use of passwords. The technology also develops an automatic database for storing the information of students and staff.

1.6 Limitations

The following are some of the suggested system's drawbacks:

a. The system is unable to accept online payments for student housing and maintenance fees.

b. Because the existing system lacks a mess system, the system is unable to handle other dormitory issues such as mess activities.

The new computer-driven student hostel management and allocation system will, among other things, facilitate timely allocation of hostel rooms to students, check hostel occupancy at any time for information management, add up the total amount realized from hostel fees each session, allow management to plan on improving hostel living conditions and provide the first-hand information.

1.7Terms and Definitions

Hostel: A hostel is a place where students can stay when they are away from home. It is located on the school grounds and features large, well-ventilated dorms and single rooms.

Management Information System (MIS): is a system that offers information that is required to efficiently manage an organization.

Hostel Management System: This is a piece of software that keeps track of the activities of staff and students in a hostel. It allows students to apply for hostels online and assigns rooms to them.

Data: are known facts that could be captured and saved on computer media in the past. Text, statistics, images, and sounds are used to create these facts.

Databases: is a collection of computer data that has been organized in a logical manner and is structured so that it may be automatically accessed or changed. It's also known as Databank.

A computer program is a set of instructions that directs the computer's actions.

System: is a collection or arrangement of pieces that work together to form a unified whole. A system is a logical structure based on a set of shared concepts or laws.

CHAPTER 2

LITERATURE REVIEW

2.1 Hostel Management System

The term hostel can be attributed to many definitions depending on the context from which it is being defined. For the purpose of 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 a 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. They were discussing concerning it as an immediate component in the educational environment of the educational institutions, an endorse resource deployed to boost hygiene, neatness, and protection as well as pedagogical performance and student effectiveness and productivity in the educational environment.

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 for the purpose of pursuing an education.

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 learning 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).

2.2 Information System

Easily, 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 or conventions applicable to the area of study in question (Zoikoczy, 1981). 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 section of individuals, protocols, a database, and (sometimes) computers and software gather, process, store, and send data for functional level workflow and resources to help executives decide (Duff & Asad, 1980). We can divide information systems 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. 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).

2.3 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). The Sdlc Cycle is a series of protocols that need to be observed to enable that all client demands are met without consuming the least amount of energy feasible even during systematic planning, implementation, and administration of enterprise applications. (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.

2.3.1 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 on 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 in the first phase of the 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 in order 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.

2.3.2Software 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 a number of general models for software processes, some of which are: Waterfall model, Iterative model, Agile model, and Spiral model. 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 Liner 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 phrases 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 prior to 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. Some of the drawbacks of this model are 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. Iterative Model

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).

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 over time. Also, this model can be used when there is time to market constraint.

c. 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. At the conclusion of the risk analysis phase, a prototype is created. The engineering phase produces software, which is then tested at the conclusion of 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 in a strategic way 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.

d.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 Unified Modelling Language

The unified modeling language is a standardized modeling language consisting of an integrated set of diagrams that not only allow software developers and system developers to specify, visualize, produce and record their system artifacts. UML represents a collection of best practices in the design of big and complex systems that have proved successful. UML is a highly significant aspect of the object-oriented development and development process of software.

UML mostly employs pictorial notation for software design initiatives. (Paradigm Visual, 2002). The purpose of UML might be stated as a simple modeling tool to model all practical systems in a complex world today. UML offers a technique of visualizing a diagram of system architecture.

2.4.1Conceptual Model

Conceptual models and their relationships are conceptual models. The first step before construction. The conceptual model helps organizations in a system understand how they interact. If you look at a system, a system can be monitored from a variety of points of view including an analyst, designer, coder, tester, quality assurance, customers, and technical authors. All these persons are interested in various aspects of the system and require different levels of detail, for example, the analyst must understand how the system works and functions before it is sent to the designer. A coder must understand the system design so that a writer knows how the system works and how it works.

. To understand the UML conceptual model, it is necessary to learn three main parts: UML Three main components need to be learned to understand the conceptual model of UML;

a.UML

b. Rules building blocks that guide the blocks connection

c. Common Mechanism of UML

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

The web-based hostel management system would be built using PHP and with a MySQL database engine. The language is chosen because it works seamlessly with a Relational Database

Management System like MySQL, and handling dynamic and server-side changes is easier to be done with PHP.

3.2 Input Design

The system design is divided into two portions. The Administrator section and the user (student's) section.

3.2.1 Administrator

● The administration may let distinct students reside in numerous dormitories.

●He can control the status of the fee payment.

A process of converting user originated inputs to a computer-based format. Input design is an important part of the development process since inaccurate input data are the most common cause of errors in the data processing. Erroneous entries can be controlled by input design

There are two major approaches for entering data into the computer. They are:

● Menus

● Dialog Boxes

● Menus.

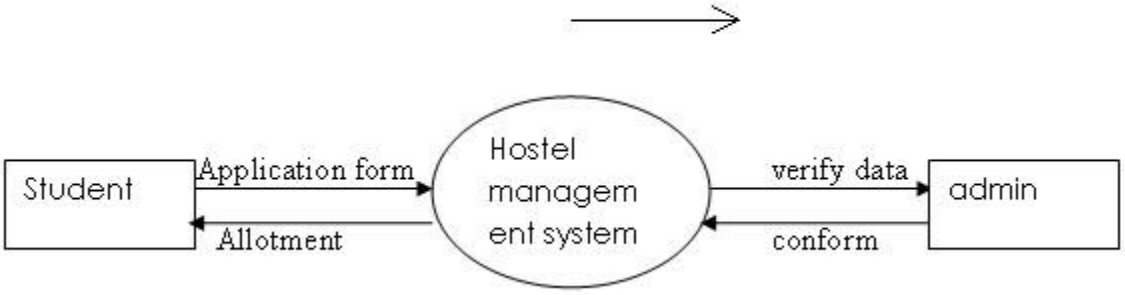
3.3 Process Design

Process design plays an important role in project development. In order to understand the working procedure, process design is necessary. Data Flow diagrams and System Flow charts are the tools used for process design.

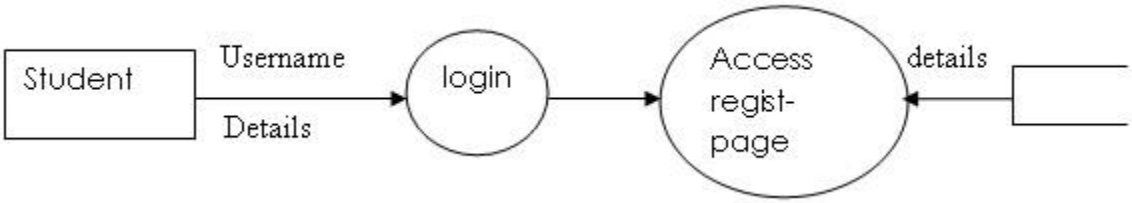
Data Flow Diagram is the logical representation of the data flow of the project. Numerous representations are utilized to symbolize the DFD. It does have a beginning and a conclusion. Squares are employed to symbolize the origin and destination, whilst circles are utilized to symbolize the activity The data flow is represented using arrows. One reader can easily get an idea about the project through Data Flow Diagram.

3.4 Data Flow Diagram

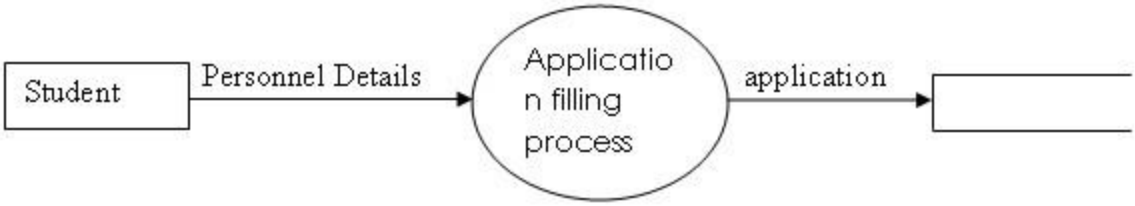
3.4.1 Context level DFD



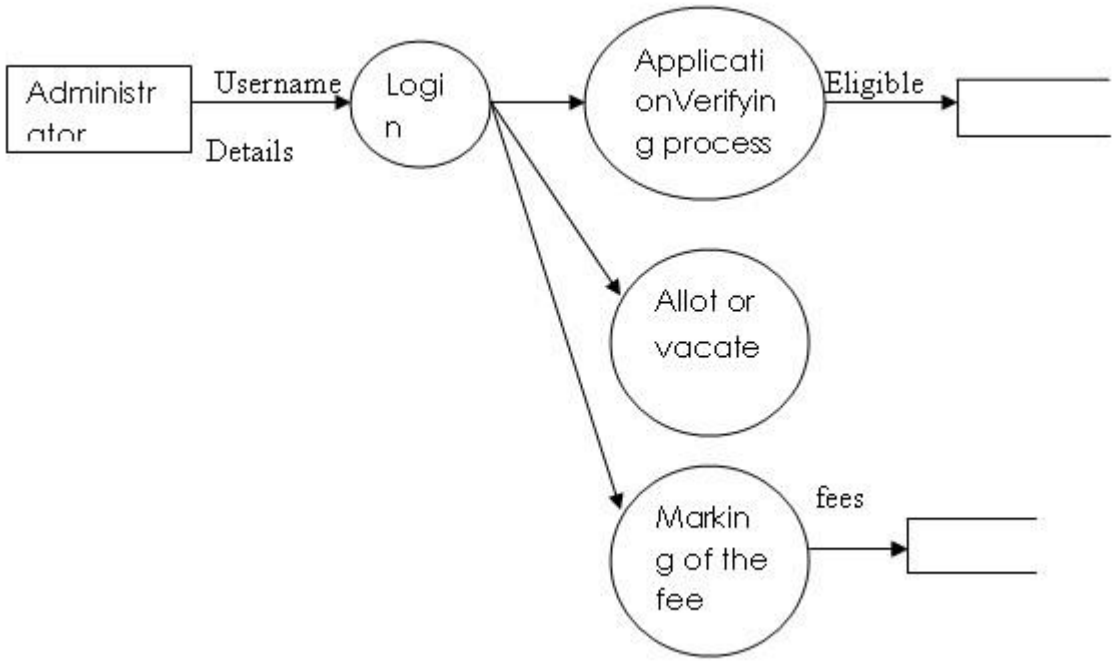
3.4.2 Student Module



3.4.3 Registration Process

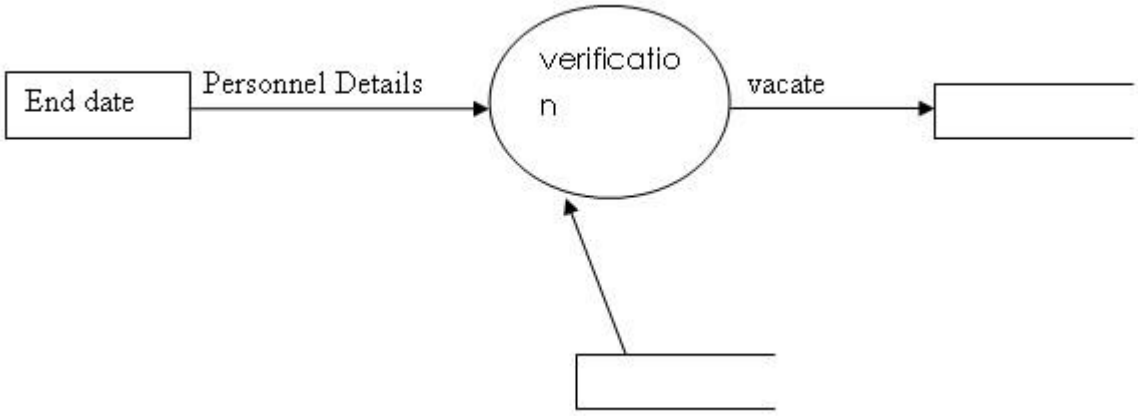


3.4.4 Admin Module



3.4.5 Allotment Process

3.4.6 Vacating Process



3.5 Database Design

The data in the system has to be stored and retrieved from the database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at the analysis stage. They are structured and put together to design the data storage and retrieval system.

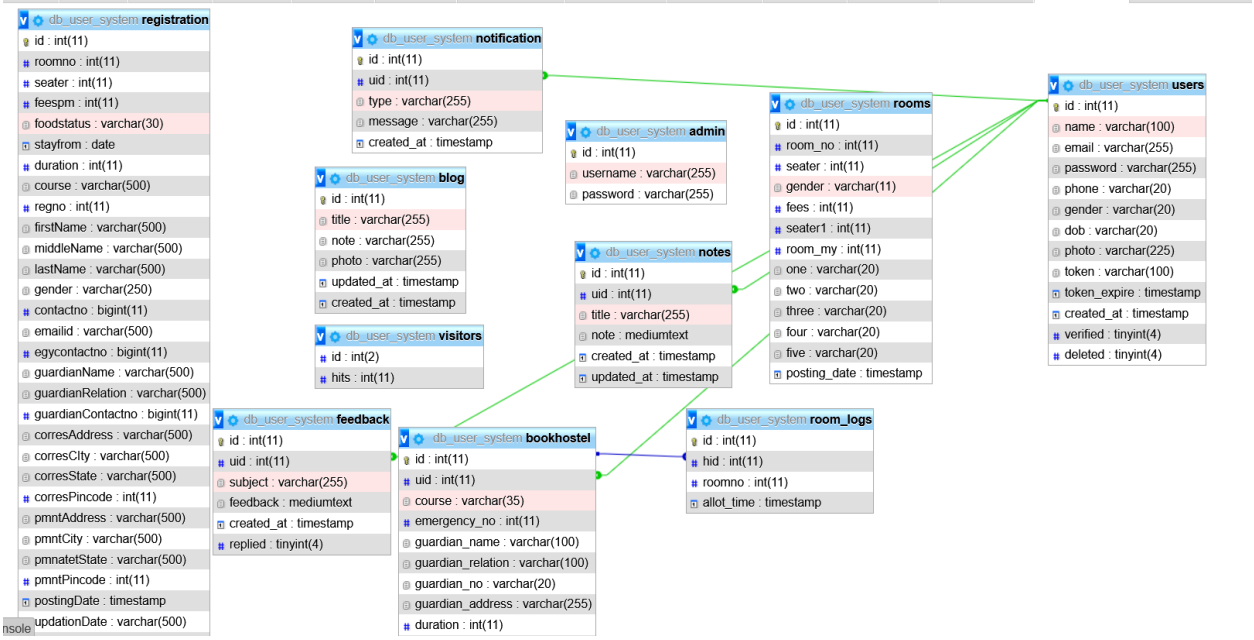


Figure 3.0 Database Schema for the Hostel Management System

1. Student Account Creation

Field Name	Data Type	Description
Name	Varchar	Name of the student
Branch	Varchar	Branch of the student
Userid	Int	Userid of the student
Password	Varchar	Password to use
Retype password	Varchar	Repeating it

2. Administrator Login

Field Name	Data Type	Description
Username	Int	Username of the student
Pass	Varchar	Password of the student

3. Application Form

Field Name	Data Type	Description
Index no	Int	A unique no given to student
Name	Varchar	Name of the student.
Age	Int	Age of the student
dd_birth	Date	Date of birth of the student
Sex	Varchar	Sex of the sutudent
Reservation	Varchar	Reservation if any
Dept	Varchar	department
Course	Varchar	Course of study
Semester	Int	Semester of study
Course_nature	Varchar	Type of course
Date_admit	Date	Date of course admission
Date_end	Date	End of course
Date_host_admit	Date	Date of hostel admission
Local Address	Longtext	Address of student
Permanent Address	Longtext	Address of guardian
Distance	Int	Distance from home

4. Allotment

Field Name	Data Type	Description
Room no.	Int	A unique no given to student
Name	Varchar	Name of the student.
Index no.	Int	Age of the student
Date_admission	Date	Date of birth of the student
Sex	Varchar	Sex of the student
Reservation	Varchar	Caste of the student

5. Vacating and editing

Field Name	Data Type	Description
Room no.	Int	A unique no given to student
Name	Varchar	Name of the student.
Index no.	Int	Age of the student
Date_admission	Date	Date of birth of the student
Sex	Varchar	Sex of the student
Reservation	Varchar	Caste of the student

3.6 System Requirement

System requirement is a description of the needs of a user for an information system. The unique requirements of a user are identified here.

3.6.1User Requirements

User-Interface Requirements

User interfaces are the registration pages developed for the students to register and the porters to manage the students. They consist of the following:

- Home page (students and porters)
- Allocation page (porter)
- Check available and taken hostels (porter)
- Manage hostel (porter)
- Search for student (porter)
- Register a new user or administrator (porter)
- Fill in credentials
- Register for available hostels
- Print allotted hostel as proof of evidence of allocation

To gain access to the e-registration system, the user would need:

- A personal computer
- A username
- A genuine password

3.6.1.2Modeling the System

The flowchart of the system model is shown in figure 4.1 below

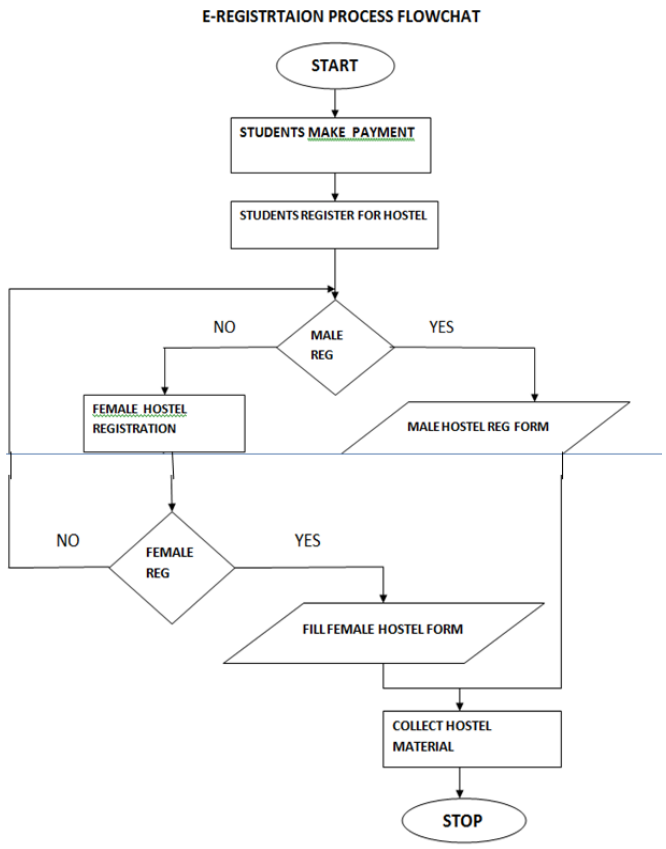


Figure 3.2: Hostel Management System Registration Model Flowcharts

3.7 Software Features

3.7.1 XAMPP

XAMPP installs a complete working PHP/MySQL server environment on Windows platforms.

Installs PHP, MySQL, Apache, and PhpMyAdmin.

3.7.2 SYSTEM TOOLS AND LANGUAGES

3.7.2.1 PHP

3.7.2.2 MYSQL

Chapter 4

SYSTEM IMPLEMENTATION AND TESTING

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence to the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation of change over methods.

Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be tested only after the test phase is done and if it is found to be working according to the requirements. This method also offers the best safety protocols since the old system can take over if errors are found or the inability to handle a certain type of transactions while using the new system.

4.1User Training

After the system is implemented successfully, the training of the user is one of the most important subtasks of the developer. For this purpose, user manuals are prepared and handed over to the user to operate the developed system. Thus, the users are trained to operate the developed system. Both the hardware and software securities are made to run the developed

systems successfully in the future. In order to put the new application system into use, the following activities were taken care of:

●Preparation of user and system documentation

●Conducting user training with demo and hands-on

●Test run for some period to ensure smooth switching over the system.

The users are trained to use the newly developed functions. User manuals describing the procedures for using the functions listed on the menu are circulated to all the users. It is confirmed that the system is implemented up to the user's needs and expectations.

4.2 Security and Maintenance

Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

Field Name	Data Type	Description
Title	varchar	The title of the notice
Contents	varchar	It includes the different notice descriptions.

4.3 Output Design

When we design an output, we must identify the specific output that is needed to meet the system. The usefulness of the new system is evaluated on the basis of its output.

Once the output requirements are determined, the system designer can decide what to include in the system and how to structure it so that the required output can be produced. For the proposed software, it is necessary that the output reports be compatible in format with the existing reports. The output must be concerned with the overall performance and the system's working, as it

should. It consists of developing specifications and procedures for data preparation, those steps necessary to put the inputs and the desired output, i.e. maximum user friendly. Proper messages and appropriate directions can control errors committed by users.

The output design is the key to the success of any system. Output is the key between the user and the sensor. The output must be concerned with the system's working, as it should.

Output design consists of displaying specifications and procedures as data presentation. The user is never left with confusion as to what is happening without appropriate error and acknowledges the message being received. Even an unknown person can operate the system without knowing anything about the system.

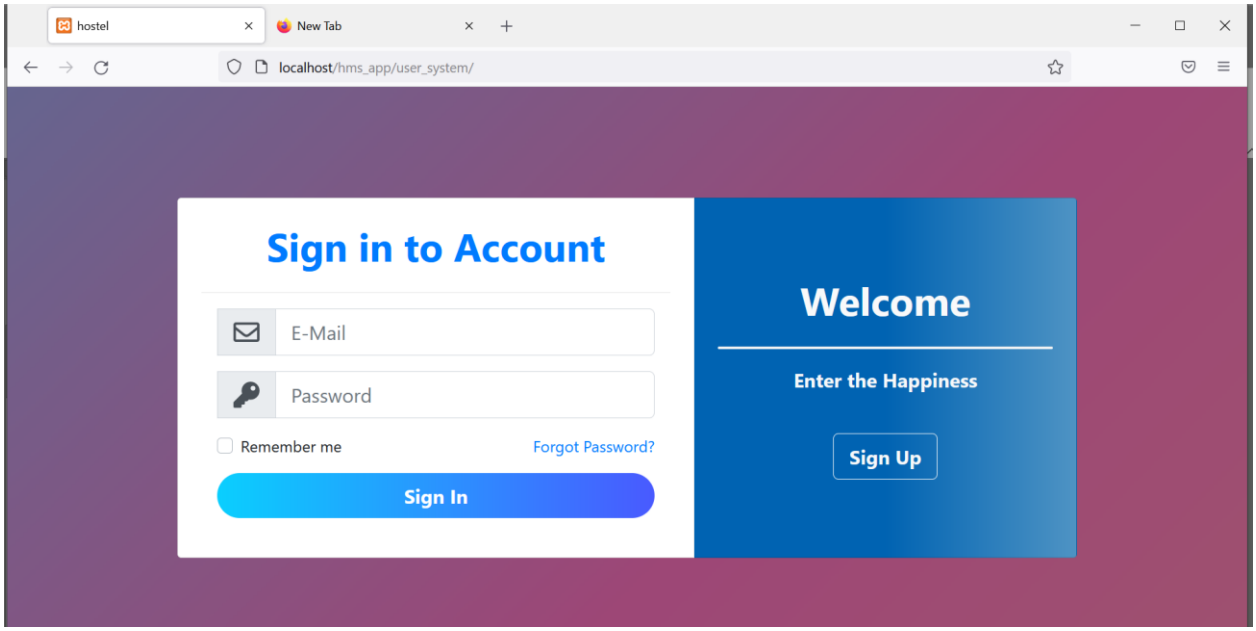


Figure 4.1: Students Login Page accessed via http://localhost/hms_app/user_system/

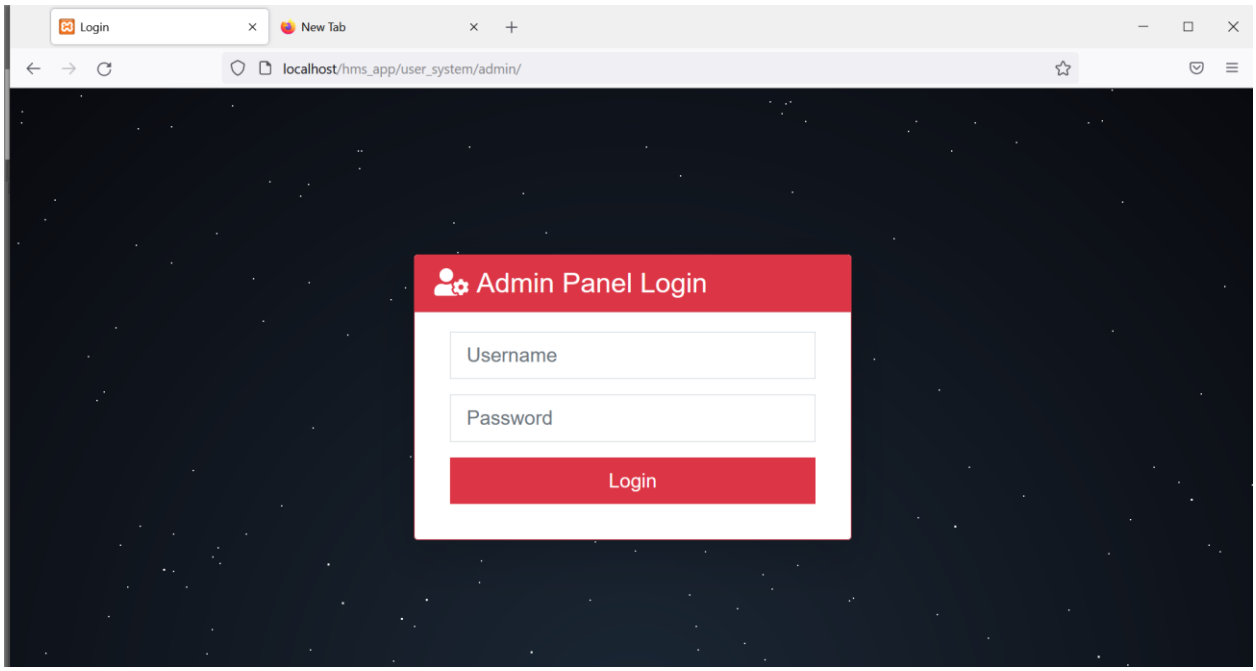


Figure 4.2: Admin Login Page accessed via http://localhost/hms_app/user_system/admin/

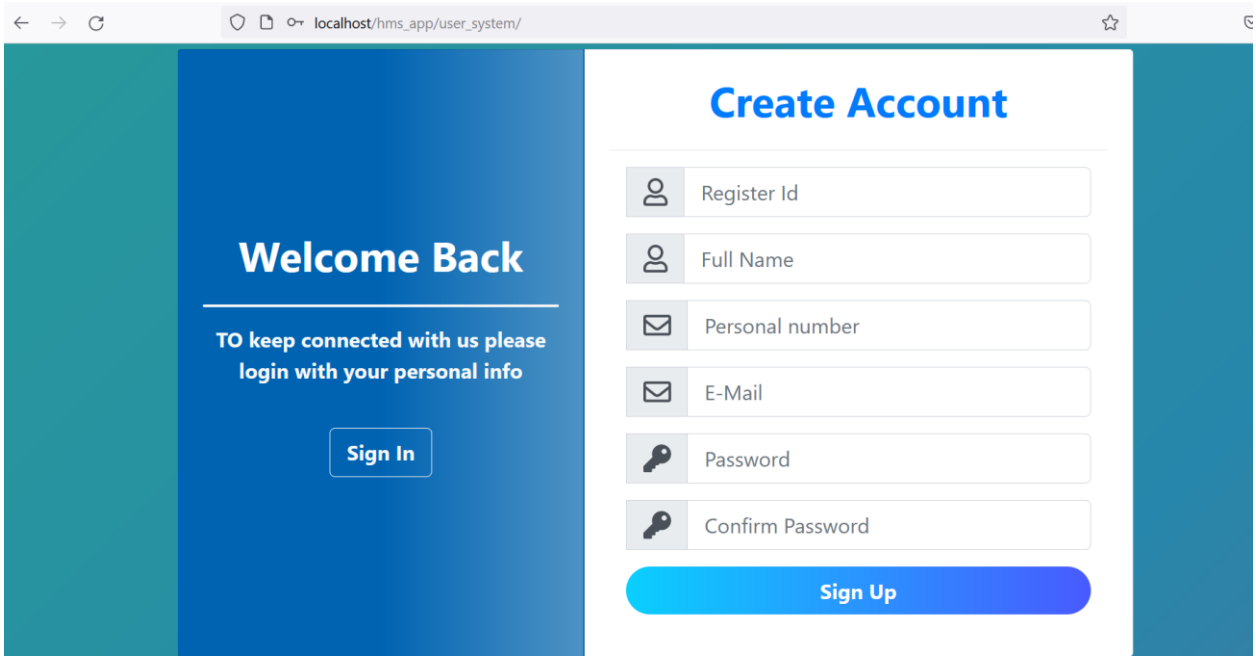


Figure 4.3: Students Registration Page

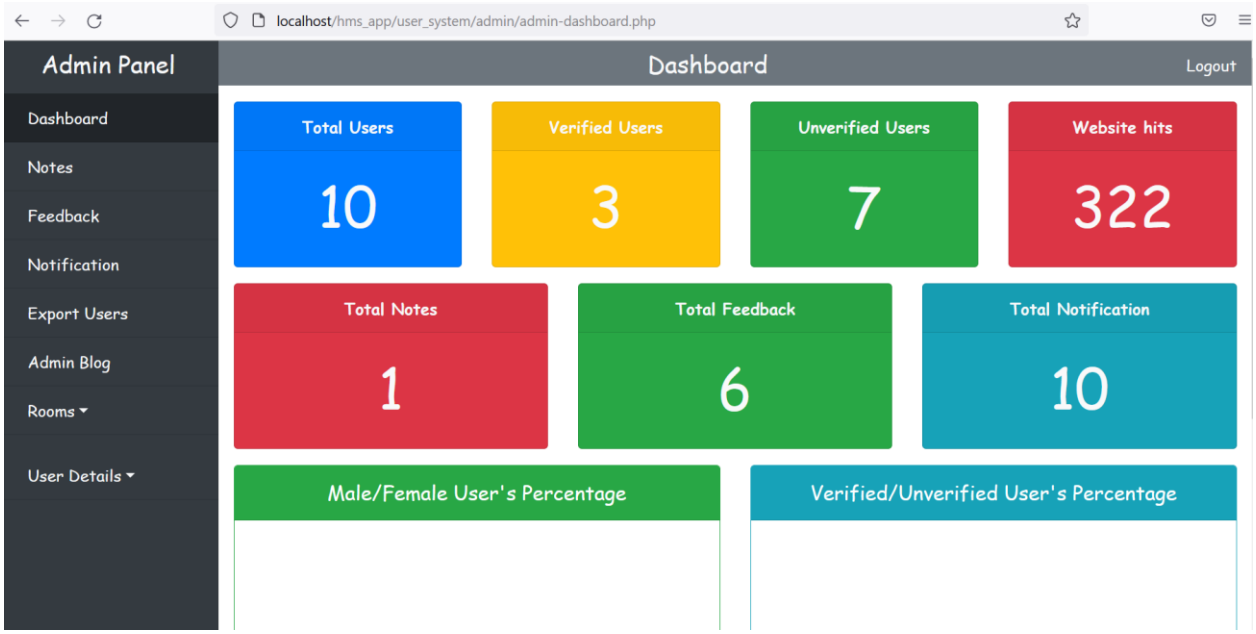


Figure 4.4 Admin Dashboard

4.4 Unit Testing

4.5 Integration Testing

4.6 User Acceptance Testing

The adaptability of a technology by its client is vital to the success of any platform the method under adoption is qualified for user adaptation with regular information during creation of implementation

Chapter 5

RECOMMENDATION AND CONCLUSION

This project was built with the use of PHP and MySQL the implementation was developed based on its fundamental requirements in order to analysis error and faults of existing systems. The requirements are based on the required specification of the user and the optimization of the existing system, with flexibility for future enhancement.

Identification of which the system would be more users flexible and more with accessibility principles built into its User System.

This system has been fully designed and developed, in such a way that it should be recommended for full utilization of its resources across the university hostels and can be extended to accommodate payment integrations.

REFERENCES

Segun O. Olatinwo and et al. 2014. “Development of an Automated Hostel Facility Management System”. Journal of Science and Engineering. 5 (1): 1-10.

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.

Andrews, S. (2017, May 13). *The world’s first youth hostel was established in 1912*. Retrieved from thevintagenews: **https://www.thevintagenews.com/2017/05/13/the-worlds-first-youth-hostel-was-established-in-1912/**

Jevtic, G. (2019, May 15). *What is SDLC? Phases of Software Development, Models, & Best Practices*. Retrieved from PhoenixNAP: https://phoenixnap.com/blog/software-development-life-cycle

kukreja, s. (2021). *Major Types of Information Systems*. Retrieved from managementstudyhq: https://www.managementstudyhq.com/six-major-types-of-information-systems.html

MBA Knowledge Base. (2021). *Computer Based Information Systems (CBIS) – Meaning and Types.* MBA Knowledge Base.

Visual Paradigm. (2002). *What is Unified Modeling Language (UML)*. Retrieved from Visual Paaradigm: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/