Design and Implementation of Student Accommodation System

(Using Recommendation System)

**BY**

**Maryam Abba Yusuf**

**BU/22B/IT/6965**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING, FACULTY OF COMPUTING AND APPLIED SCIENCE, BAZE UNIVERSITY, ABUJA.**

**August ,2024**

# DECLARATION

I hereby declared that this research project has been written by me under the supervision of Dr. Usman Bello Abubakar. The work has been presented in any previous research for the award of B.Sc degree to the best of my knowledge. The work is entirely mine and I accept the sole responsibility for any errors that might be found in the work, while the reference to publish material have been duly acknowledge.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Maryam Abba Yusuf Date

BU/22B/IT/6965

**APPROVED BY**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Head of Department,**

Department of Computer Science

# CERTIFICATION

This project entitled “Design and Implementation of Student Accommodation System (using Recommendation system)” meets the requirements governing the award of Bachelor of Science in Software Engineering in Baze University, Abuja.

# APPROVAL

# This is to certify that the research work title Design and Implementation of Student Accommodation System (using Recommendation system) by Maryam Abba Yusuf with BU/22B/IT/6965 has been approved by the Department of Computer Science, Faculty of Computing and Applied Science, Baze University, Abuja, Nigeria.

By

Dr Usman Bello Abubakar

Supervisor Date

Dr Usman Idris Abubakar

Head of Department Date

Assoc Prof Chandrashekar V Uppin

Dean, Faculty of Computing and Applied Science Date

Prof Choji Davou Nyap

External Examiner Date

# DEDICATION

I dedicate this work to Allah (SWA), my source of inspiration, knowledge and understanding, who gave me the grace and strength throughout this program. I also dedicate this work to my parents, for their love and support throughout my life.

# ACKNOWLEDGMENT

All praise, first and foremost, is to God Almighty for His guidance and abundant blessings. My deepest appreciation goes to my parents and siblings for their unwavering love and support. I would also like to acknowledge my project supervisor, Mr. Usman Abubakar, for his invaluable assistance, enduring the stress of reviewing my work and providing necessary feedback. I am deeply grateful to Mr. Alee Dauda for his constant support and guidance throughout this project. His insights and encouragement played a key role in its successful completion.

# ABSTRACT

Many students encounter difficulties finding suitable accommodation after being admitted to university. To address this issue, a Student Recommendation System has been developed to assist students in finding hostels that meet their specific requirements through preference matching.

This system offers a comprehensive list of hostels, which are regularly updated by administrators. Through the platform, students receive personalized recommendations for hostels that align with their preferences. By leveraging preference matching, the system ensures that recommendations are tailored to each student's unique needs, providing a pre-visit insight that helps them secure appropriate accommodation with greater ease.

The Student Recommendation System benefits both students and hostel owners. For students, it simplifies the search process, reduces the stress associated with finding accommodation, and ensures that recommendations reflect their individual preferences. For hostel owners, it enhances visibility and extends their reach within the city, potentially resulting in increased bookings. The system serves as a bridge between students and hostel providers, making the accommodation search more efficient and effective for all parties involved.

Table of Contents

[DECLARATION ii](#_Toc177504694)

[CERTIFICATION iii](#_Toc177504695)

[APPROVAL iv](#_Toc177504696)

[DEDICATION v](#_Toc177504698)

[ACKNOWLEDGMENT vi](#_Toc177504699)

[ABSTRACT vii](#_Toc177504700)

[LIST OF TABLES xi](#_Toc177504701)

[LIST OF FIGURES xii](#_Toc177504702)

[LIST OF ABBREVIATION xiii](#_Toc177504703)

[CHAPTER ONE: INTRODUCTION 1](#_Toc177504704)

[1.2 Overview 1](#_Toc177504705)

[1.2 Background and Motivation 1](#_Toc177504706)

[1.3 Statement of the Problem 2](#_Toc177504707)

[1.4 Aim and Objectives 2](#_Toc177504708)

[1.5 Significance of the Project 2](#_Toc177504709)

[1.6 Project Risks Assessment 2](#_Toc177504710)

[1.7 Scope/Project Organization 3](#_Toc177504711)

[1.8 Definition of Terms 3](#_Toc177504712)

[CHAPTER 2: LITERATURE REVIEW 4](#_Toc177504713)

[2.1 Introduction 4](#_Toc177504714)

[2.2 Historical Overview 4](#_Toc177504715)

[2.3 Recommendation system 5](#_Toc177504716)

[2.3.1 How the Preference System Works 5](#_Toc177504717)

[2.4 Related Work 6](#_Toc177504718)

[2.4 Summary 16](#_Toc177504719)

[CHAPTER 3:REQUIREMENTS, ANALYSIS, AND DESIGN 16](#_Toc177504720)

[3.1 Overview 16](#_Toc177504721)

[3.2 Adopted Methodology 16](#_Toc177504722)

[3.4 Tools and Techniques. 17](#_Toc177504723)

[3.5 Ethical Consideration 18](#_Toc177504724)

[3.6 Requirement Analysis 18](#_Toc177504725)

[3.6.1 Software Requirements 18](#_Toc177504726)

[3.6.2 Hardware Requirements 18](#_Toc177504727)

[3.7 Requirements Specifications 18](#_Toc177504728)

[Table 3.2 Non-Functional Requirement Specifications 19](#_Toc177504729)

[3.8 System Design 21](#_Toc177504730)

[3.8.1 Application Architecture 21](#_Toc177504731)

[3.8.2 Use Case 23](#_Toc177504732)

[3.8.2.1 Use-Case Description 24](#_Toc177504733)

[CHAPTER 4: IMPLEMENTATION AND TESTING 28](#_Toc177504734)

[4.1 Overview 28](#_Toc177504735)

[4.2 Main Features 28](#_Toc177504736)

[4.3 Implementation Problems 28](#_Toc177504737)

[4.4 Overcoming Implementation problem 29](#_Toc177504738)

[4.5 Testing 29](#_Toc177504739)

[4.5 Use Guide 33](#_Toc177504740)

[4.7 User Interface Design 34](#_Toc177504741)

[CHAPTER 5 : DISCUSSION, CONCLUSION AND RECOMMENDATION 41](#_Toc177504742)

[5.1 Overview 41](#_Toc177504743)

[5.2 Objective Assessment 41](#_Toc177504744)

[5.3 Limitations and Challenges 41](#_Toc177504745)

[5.4 Future Enhancements 41](#_Toc177504746)

[5.5 Recommendations 42](#_Toc177504747)

[5.6 Summary 42](#_Toc177504748)

[REFERENCES 43](#_Toc177504749)

# LIST OF TABLES

[**Table 1.1 Project Risks Assessment** 2](#_Toc177509821)

# LIST OF FIGURES

[Figure 2.1 Recommendation System (Analytics vidhya.2021) 5](#_Toc177404363)

# LIST OF ABBREVIATION

RS: Recommendation Systems

CF: Collaborative Filtering

DF: Demographic Filtering

CBF: Content Based Filtering

SVM: Support Vector Machine

CARS: Context Aware Recommender System

# CHAPTER ONE: INTRODUCTION

# 1.2 Overview

This project aims to develop a modern student accommodation system for students in Abuja, leveraging an advanced recommender system. The goal is to deliver personalized housing recommendations tailored to individual preferences. By automating the search process, this system will significantly reduce the time and effort students spend finding suitable accommodation, while also optimizing house allocation.

# 1.2 Background and Motivation

Student accommodation is a critical aspect of university life, ensuring that students have safe, affordable, and convenient places to live while pursuing their studies. In Abuja, the capital city of Nigeria, the demand for student housing has surged with the growing number of higher education institutions and students. Options range from university to private student housing complexes, shared apartments. However, finding the right accommodation that meets the diverse needs and preferences of students remains a challenge in Abuja's dynamic real estate market.

Initially, the process of finding student accommodation was manual, relying on physical visits, word of mouth, and simple listings. This approach was time-consuming, and often resulted in mismatches between students’ needs and available housing options (Johnson, 2019). As student populations grew and the variety of housing options expanded, these manual processes became increasingly inadequate.

The system developed for this project leverages modern algorithms to match students based on their stated preferences. This approach is both personalized and data-driven, ensuring that users are matched with roommates and rooms that align with their living style.

The main goal of this system is to improve the student living experience by minimizing potential conflicts between roommates and promoting positive interactions. Students can specify their detailed preferences, which the system uses to suggest compatible rooms and roommates. This process is efficient, benefiting both students and accommodation managers by saving time and effort.

Additionally, the system gives students a sense of empowerment, knowing their living preferences are considered, which can result in higher satisfaction and increased retention for housing providers.

# 1.3 Statement of the Problem

Student struggles to find suitable accommodation due to limited knowledge of options, information overload from various listings and difficulty matching their needs with the available housing. This project proposes a recommendation system specifically designed for students, utilizing user profiles and data analysis to suggest the best fitting options, simplifying the student accommodation.

# 1.4 Aim and Objectives

The main aim of this research is to design and implement a student accommodation using recommender system to match users with a roommate base on their preferences.

1. To develop a comprehensive and user-friendly software that assists student in finding suitable Accommodation
2. To create a booking and reservation system that allows users to secure their chosen Accommodation, manage bookings, and handle payment transactions securely.

3. To develop a user-friendly platform where students can create profiles specifying their accommodation requirements (location, lifestyle preferences).

# 1.5 Significance of the Project

The Student Accommodation system using a recommender system is significant as it offers many benefits:

1. It ensures that students find suitable accommodation quickly and accurately.
2. It provides users with a recommendation system which allows them to choose an accommodation based on their location.
3. It reduces the time and effort required for manual accommodation.
4. Helps in efficient allocation and management of housing resources.

# 1.6 Project Risks Assessment

**Table 1.1 Project Risks Assessment**

|  |  |
| --- | --- |
| RISK | IMPROVEMENT |
| Inability to carry out research due to loss of hardware/software resources | Be aware of and observe school IT security procedures  Secure Android mobile phone when not in use |
| Loss of work due to equipment failure /loss | Daily Backup of data to multiple sources of storage such as flash drives, hard drives, google drive, etc. for multiplicity |
| Software availability (Unavailability of API’s) | Alternative API’s will be checked for. Software requirements will be identified in good time for possible contentious software. |

# 1.7 Scope/Project Organization

This project was arranged into five chapters: Chapter one as an introduction to the general aim and objective of the project, and the ideas at focus presented. Chapter two deals with relevant literatures of components used in realizing this project while Chapter 3, is design methodology, Chapter 4, is implementation of the methodology and testing. Chapter 5 covered conclusions, limitations, and suggested improvements for the system.

# 1.8 Definition of Terms

Recommendation system: a subclass of information filtering system that provides suggestions for items that are most pertinent to a particular user.

Accommodation: A room, group of rooms, or building in which someone may live or stay.

Preference: A greater liking for one alternative over another or others

# CHAPTER 2: LITERATURE REVIEW

# 2.1 Introduction

The literature review chapter aims to provide a comprehensive overview of the existing knowledge and research related to the development of a student accommodation system using a recommendation system. This chapter will look into the historical evolution of accommodation systems, examine previous research and implementations of recommendation systems in the housing sector, and identify gaps and limitations in current solutions. By critically analyzing the literature, this study seeks to build upon existing knowledge and contribute to the advancement of an efficient and personalized student accommodation system. The exploration of literature will inform the design and implementation of a robust recommendation system tailored to meet the diverse needs of students in Abuja, ensuring optimal matching of preferences with available housing options.

# 2.2 Historical Overview

By the mid-2010s, recommendation systems (RSs) continued to evolve, incorporating advancements from fields such as human-computer interaction, machine learning, and information retrieval. This period saw the development of numerous innovative RS applications. For instance, Spotify and Apple Music utilized sophisticated algorithms to provide personalized music recommendations. Similarly, platforms like Netflix and YouTube enhanced their recommendation engines to deliver more accurate and appealing video suggestions to users.

During this time, RSs also became crucial in marketing, helping businesses enhance sales and customer experiences through personalized content delivery. Amazon's recommendation engine is a prime example, continuously evolving to suggest products based on user behavior and preferences. Social media platforms, such as Facebook and Instagram, have also integrated advanced RSs to tailor content, advertisements, and friend suggestions to individual users.

The focus in RS research shifted towards incorporating deep learning techniques, leading to more refined and precise recommendations. Researchers explored hybrid models combining collaborative filtering, content-based methods, and contextual information to improve recommendation accuracy and relevance. Additionally, the rise of mobile and wearable technologies provided new opportunities for context-aware recommendations, leveraging data such as location, time, and user activity to offer highly personalized suggestions.

Emerging trends included the application of RSs in new domains such as news personalization, healthcare, and smart home devices. For example, news platforms like Flipboard and Google News employed RSs to curate articles tailored to users' interests. In healthcare, RSs were used to recommend personalized treatment plans and wellness activities. Smart home devices, such as Amazon Echo and Google Home, utilized RSs to suggest routines and control connected devices based on user preferences and habits.

# 2.3 Recommendation system

A recommendation system is a type of system learning machine that provides personalized recommendations to users based on their past behaviors, options, and styles. It is a subclass of information filtering systems that use algorithms to advise gadgets to users primarily based on their pastimes or behaviors.

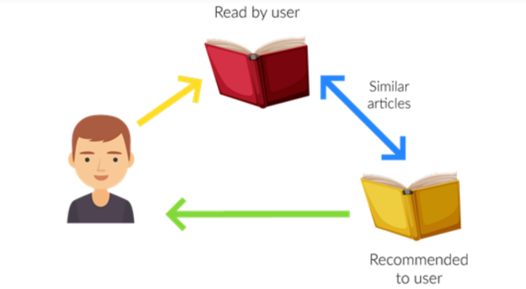


Figure 2. Recommendation System (Analytics vidhya.2021)

Recommendation systems are widely used in e-commerce, social media, entertainment, and other online platforms to increase user engagement and retention, improve customer satisfaction, and drive sales and revenue.

# How the Preference System Works

There are four steps of how recommendation system works:

Collecting User Preferences: After student’s log in, they are required to specify their roommate preferences, which include key factors like cleanliness, sleep habits, quietness, and social behavior. These attributes play a vital role in helping the system identify compatible roommates.

Filtering Available Rooms: When a student begins searching for accommodation, the system first narrows down available rooms by considering the user’s preferences. The get filtered room recommendations function filters out rooms that do not meet the student’s gender preference, only showing rooms with available spots that align with the user's requirements.

Calculating Match Percentages: A significant part of the preference-matching process is carried out by the calculate match percentage function. This function compares the student's preferences with those of the current occupants in a room, calculating a "match percentage. The algorithm evaluates how closely the student's preferences for cleanliness, quietness, and socialness align with those of the existing occupants. It does this by calculating the absolute difference between the user's and occupants' preferences. The smaller the difference in these values, the greater the compatibility, and consequently, the higher the match percentage.

Ranking Rooms: Once match percentages are computed for all available rooms, the system ranks them from the highest to the lowest match percentage. This ensures that the student is presented with rooms that best match their preferences, increasing the likelihood of finding a room that meets their expectations.

Example:

If the user prefers a high level of cleanliness (e.g., a score of 10) and the current occupant prefers a medium level of cleanliness (e.g., a score of 5), the system records a difference of 5. The process is repeated for other preferences, such as quietness and social behavior, and the differences are summed up. If a room has no current occupants, it is treated as a perfect match, with a score of 100%.

# 2.4 Related Work

Lu et al., (2015) A lot of work has been done by the research community to enhance the applicability and performance of Recommendation systems (RSs) over the last few years. New methodologies and algorithms were developed to address many of the technological challenges such as producing more accurate recommendation while reducing online computation time. Several recommendation algorithms have been proposed and successfully implemented in different domains. These algorithms mainly follow demographic filtering (DF), content-based filtering (CBF), collaborative filtering (CF) and hybrid approaches. Recently, RS has expanded its exploration and is using social networks and some contextual information to generate dynamic features in the recommendation.

Dejo et al., (2015), Zhang et al., (2016) and Bernardes et al., (2015). Nowadays social networking sites (such as Facebook, Twitter, etc.) have emerged as a substantial platform for applying Recommendation systems (RSs). These popular sites are considered to be the major source of information about people and hence becoming a great option to leverage novel and innovative approaches for the recommendation, leaving behind the old methods, to increase the accuracy. The contextual information such as time, place, the emotion of people and groups in these social networking sites opens up a new avenue of recommendation known as contextual RS. It also provides a good prospect to bring a dynamic essence in the recommendation. Seasonal marketing and conference recommendation are also emerging as considerable application areas in the context-aware recommendation.

This is the most recognized and widely implemented RS (Singh et al., 2019). CFRS follows the philosophy of “a man is known by his company he keeps.” That means if CFRS believes that if two or more user’s interests matched in the past, then it is likely that in future also their interests should match. For example, if the purchase histories of user1 and user2 strongly overlap then it is high on the cards that if user1 buys a product, then user2 will also buy the same or similar product. CF approaches to keep track of the user’s past reviews and ratings on items to recommend similar items in the future. Even if the user did not deal with a particular item, it would be recommended to him if his peers have used the same. It is obvious that to achieve reasonable recommendation accuracy a large number of user groups are required to be considered, trust is an important factor for reliable recommendation.

Pelánek, (2018), Wang et al. (2015b) and Peis et al., (2018) stated that many problems of common Recommendation systems (RSs) are eliminated by using semantic-based RS. More details of the semantic-based Recommendation system (RS) can be found in the article, as an example, may be referred to, where the authors proposed and evaluated the preference of a semantic-based friend RS for the social network. Though KBRS is capable of providing the required information that cannot be achieved through the conventional approaches, the knowledge modelling and handling techniques in KBRSs are comparatively expensive in nature.

Wang et al. (2015b) and Pelánek, (2018) implies that metadata of a user profile and item description are used to establish a proper matching for the recommendation. Many problems of common RSs are eliminated by using semantic-based RS. More details of the semantic-based Recommendation systems (RS) can be found in the article as an example, may be referred to, where the authors proposed and evaluated the preference of a semantic-based friend RS for the social network. Though KBRS is capable of providing the required information that cannot be achieved through the conventional approaches, the knowledge modelling and handling techniques in KBRSs are comparatively expensive in nature.

Su and Khoshgoftaar, (2019) Lakshmi and Lakshmi, (2014); observed that when a new item or a new user is introduced to an RS, the system will not have any past records (ratings, preferences, search history, etc.) on the basis of which recommendation should be made. This is known as the cold start problem. It is also termed as the new user problem or new item problem. A solution to this problem includes exploiting the demographic information of the user obtained from the user’s profile. This solution is insufficient and not completely correct as users with the same demographic features may show varying interests towards a particular item.

Lakshmi and Lakshmi, (2014) stated that in practice, the RSs work with very large datasets. Hence, the user-item matrix used for CF is extremely sparse, which adversely affects the performances of the predictions or recommendations of the CF systems. It also takes place when a user, having used some particular product, did not bother to rate it. In other cases, users do not rate items that are not known to them. To overcome this problem, Recommendation system (RS) employs an approach called the clustering method. Clustering method refines the data according to the preference of the user, and by doing so, it makes it easy for recommending items. Unfortunately, there are certain issues that are yet to be resolved in the case of multi-level clustering.

Tewari, A.S. and Priyanka, K. (2015) stated that as the Recommendation systems (RSs) work on large datasets, the complexity of the RSs increases in case of a huge number of users and millions of distinct items set. Many systems need to react immediately to online requirements and make recommendations for all users based on their purchases and rating history, which demands high scalability items

Meymandpour and Davis, (2015) and Amazon.in, (2017a, 2017b). Synonymy refers to the problem of multiple words having similar meanings). Most of the Recommendation systems (RSs) are unable to find the same or similar items with different names (synonyms). On account of this incapability, some associated problems emerge. For example, ‘children movie’ and ‘children film’ basically denote the same items, but memory-based CF systems would find no match between them to compute similarity.

Lakshmi and Lakshmi, (2014), Sarwat et al. (2015), Mayeku et al, (2015) and Orellana-Rodriguez et al (2015) stated that if the RS is not familiar with the abbreviations that the users often use during online interactions, it will not be able to recognize the item that the user is looking for. This generates an erroneous recommendation. The solution is to categories the abbreviated words with their full forms and put both the names on the same list. If the target user’s contextual information is available, we can make the RSs ubiquitous Various attributes like time, location, companion, mood, etc., can define a context. The difference between contextual information and demographic information is that demographic properties of a user generally remain the same for a longer period, whereas contextual information changes when the surroundings of the user change. To capture the emotional context, a hefty amount of data managed which leads to various challenges, have proposed CARS that can be built into a database system. A context-aware online learning environment has been presented in. An effective way to extract the worthwhile contexts from user’s comments available on YouTube.

Habibi and Popescu-Belis (2015) and West et al. (2016) In RS, to make a group, among a large set of objects, based on similarity, structures, and patterns, cluster analysis (i.e., unsupervised learning technique) is used. have mentioned the problem of keyword extraction from documents and provided a solution for document recommendation in conversations by applying cluster analysis based on keyword similarity that have been presented on a simple citation-based method for recommending articles by clustering based on the user’s recent history and searching patterns.

Wang et al. (2015a) present a Bayesian network classifier (i.e., a probabilistic model) is applied to solve classification problems in huge networks like social networks. To solve the user’s cold start problem and improve accuracy in the recommendation, proposed a trust-based probabilistic recommendation model for social networks.

Zhang and Zhou (2014) Support vector machine (i.e., supervised learning) is used with an associated learning algorithm for analyzing data using classification (linear and nonlinear) and regression analysis have used this technique along with Hilbert-Huang transform to detect profile injection attacks in CFRS.

Wilson et al. (2014) and Pyo et al. (2015) Extracting a common topic from various documents is called topic modelling. A topic is identified with the help of a different combination of words in a document. LDA (a probabilistic model of a corpus) used for topic modelling in RSs. To overcome the sparsity problem in rating dataset, have proposed an improved CF algorithm for recommending, using the topic modelling on a textual description of items. TV users face difficulties in finding. To help TV viewers in finding the favourite TV program from countless numbers of TV programs (through various channels), have introduced an LDA-based unified topic model for TV program recommendation.

Zheng, (2016), Covington et al. (2016) and Elkahky et al. (2015) believes that deep learning plays a major role in extracting hidden patterns from data and has opened up a new area in data mining research. It can be used in the building of effective and dynamic behavior modelling in RSs. We can gather intrinsic details about the user by understanding the approaches of supervised and unsupervised learning in the deep neural network have proposed ‘deep content-based music recommendation’ to minimize the problems in music RS by predicting the latent factors from music. Using the deep generation model and deep ranking model, have presented a deep neural network for recommendations on YouTube, one of the most popular RS for videos. The deep generation model is used to take input from the user’s side, and the deep learning model is used to rank the recommended videos. have illustrated a content-based RS with a deep learning approach to maximize the similarity between users and their preferred items in latent space. They also extended their models in different domains to extract more features related to users and items.

Table 2.1 Comparative Analysis of the Related work

|  |  |  |  |
| --- | --- | --- | --- |
| **Related Work** | **Method/Approach** | **Strengths** | **Weaknesses** |
| Aggarwal, C. C. (2016) | Knowledge-based recommender systems | Comprehensive suite of tools, user-friendly interface, supports various constraints and preferences | Requires significant technical expertise and resources for implementation and customization |
| Basavesh et al**, (2023).** | Location-Based Recommendation System | Hostel Finder: for Hostels and PGS with Transit | User modelling and user-adapted interaction. |
| Covington, et al, (2016). | Multi-criteria journey aware housing recommender system | Proceedings of the 8th ACM Conference on Recommender systems, | Limited information provided in the document |
| Dejo, el tal, (2015) | Recommendation systems | Principles, methods and evaluation. | Specific limitations not mentioned in the document |
| Ekstrand, **et al, (2015)** | Collaborative filtering recommender systems | Foundations and Trends in Human-Computer Interaction | Theoretical Aspects and Real Applications |
| Elahi, **et al,** (2016) | A survey on collaborative filtering recommender systems | A survey of active learning in collaborative | Specific strengths and weaknesses not mentioned in the document |
| Elkahky, et al, (2015) | Modeling in recommendation systems. | A multi-view deep learning approach for cross domain user | Specific strengths and weaknesses not mentioned in the document |
| FoxTrit (2017) | Personalized Recommender System for Digital Libraries | Reduced scheduling process time, improved allocation fairness | Limited information provided in the document |
| Habibi, M. and Popescu-Belis, A. (2015) | recommendation in conversations’ | Keyword extraction and clustering for document | Specific limitations not mentioned in the document |
| Lakshmi, **et al. (2014)** | Recommendation systems | Issues and challenges of recommendation. | Specific strengths and weaknesses not mentioned in the document |
| Lu, et al, (2015) | Recommender system application developments: a survey | Decision Support Systems on recommendation system applications | Specific strengths and weaknesses not mentioned in the document |
| Mayeku, **et al. (2015)** | Empirical analysis of the impact of recommender systems on sales | Manage Inform System, and development of a flexible system | Specific strengths and weaknesses not mentioned in the document |
| Meymandpour, R. and Davis, J. (2015) | Enhancing recommender systems using linked. | Open data-based semantic analysis of items. | Specific strengths and weaknesses not mentioned in the document |
| Moghaddam, **et al,** (2014) | Item-based Collaborative Filtering Recommendation Algorithms. | Maximizes student accommodation enrollment placements, minimizes | Specific strengths and weaknesses not mentioned in the document |
| Orellana-Rodriguez, **et al** (2015) | Mining affective context in short films for emotion-aware recommendation’ | Enhances efficiency, reduces administrative redundancies | Specific strengths and weaknesses not mentioned in the document |
| Peis, et **al,** (2018) | Semantic recommender systems | Provides insights into the analysis of the state of the software solutions | Specific strengths and weaknesses not mentioned in the document |
| Pelánek, R. (2018) | AWeb-based Recommendation System for Housing Selection | Design, Implementation and Evaluation of Data Mining and Knowledge in Engineering, | Specific strengths and weaknesses not mentioned in the document |
| Pyo, S., Kim, E. and Kim, M. (2015) | LDA-based unified topic modeling for similar TV User grouping and TV program recommendation | study impact on grouping TV users. | Specific strengths and weaknesses not mentioned in the document |
| Sarwat, et al, (2015) | A middleware for context-aware recommendation | study impact on database systems’ | Specific strengths and weaknesses not mentioned in the document |
| Singh, et al  (2019f) | ‘Improving the accuracy of collaborative filtering-based recommendation system | by considering the temporal variance of top-N. | Specific strengths and weaknesses not mentioned in the document |
| Su, X. and Khoshgoftaar, T.M. (2019) | A survey of collaborative filtering techniques | Advances in Artificial Intelligence | Specific strengths and weaknesses not mentioned in the document |
| Tewari, A.S. and Priyanka, K. (2015) | Book recommendation system based on collaborative filtering | Association rule in mining for college students | Specific strengths and weaknesses not mentioned in the document |
| Wang, et al (2015a) | A trust-based probabilistic recommendation | Model for social networks | Specific strengths and weaknesses not mentioned in the document |
| West, J.D., Wesley-Smith, I. and Bergstrom, C.T. (2016) | A recommendation system based on hierarchical clustering | of an article-level citation network | Specific strengths and weaknesses not mentioned in the document |
| Wilson, J., Chaudhury, S. and Lall, B. (2014) | Improving collaborative filtering-based recommenders | using topic modelling | Specific strengths and weaknesses not mentioned in the document |
| Zheng, L. (2016) | A Survey and Critique of Deep Learning on Recommender Systems, | Provides insights on a Survey and Critique of Deep Learning on Recommender Systems | Specific strengths and weaknesses not mentioned in the document |

# 2.4 Summary

This chapter presents an extensive review of the literature on the development of student accommodation systems using recommendation systems, specifically focusing on the needs of students in Abuja. The review aims to provide a broad survey of existing knowledge and research related to recommendation systems, tracing their historical development and examining various implementations in the housing sector. Initially, the chapter discusses the evolution of recommendation systems from their early stages to the mid-2010s, highlighting how advancements in human-computer interaction and information retrieval have significantly improved the accuracy and relevance of recommendations. It includes examples such as Spotify, Apple Music, Netflix, and Amazon, demonstrating the diverse applications of recommendation systems in providing personalized content to users. The chapter then explains the fundamental workings of recommendation systems, It also reviews related work by various researchers, highlighting the strengths and weaknesses of different approaches and methodologies used in recommendation system. while much of the research focuses on advanced algorithms, simpler methods like preference matching can be just as effective, especially in smaller-scale or more focused applications like a student accommodation system

# CHAPTER 3:REQUIREMENTS, ANALYSIS, AND DESIGN

# 3.1 Overview

This chapter focuses on determining the requirements, performing analysis, and developing the system design for Student Accommodation (Recommendation system). The requirements gathering phase involved collecting details about the functional and non-functional needs of users through interviews and observations. Various diagrams have been used to depict the system analysis and design including use cases, activity diagrams, data flow diagrams and entity.

# 3.2 Adopted Methodology

Agile methodology is chosen for its flexibility, continuous feedback, collaboration, and incremental delivery, ensuring the student accommodation system adapts to changing needs, incorporates user input, and delivers a high-quality product efficiently.

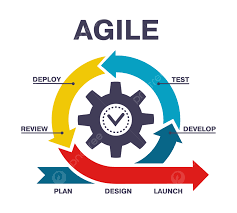


Figure 3.1 Agile Model Source: (<https://pngtree.com/>

Figure 3.1 Agile Model Source: (<https://pngtree.com/>)

**3.2.1 Method 1 (Interview)**

Interviews were conducted with students to understand their accommodation preferences. These interviews helped identify common needs and preferences, which informed the functional requirements of the system.

**3.2.2 Method 2(Observation)**

I will serve as a case study for this project based on my personal experience with university accommodation, where students were unable to choose their roommates. This led to dissatisfaction and conflicts, which could have been mitigated with a better system. This inspired the idea of using a recommendation system to match students based on their preferences, improving their living arrangements and overall experience. Using my own experience will provide insight into the challenges and help validate the necessity and potential effectiveness of the proposed solution.

# 3.4 Tools and Techniques.

Next.js is been used on both front-end and back-end for structure, styling, and recommendation. PostgreSQL is used on the back-end to generate dynamic content and store/access data from a database. Together these tools allow for complete web application development.

# 3.5 Ethical Consideration

The main ethical considerations for this development are:

1. Student data privacy and security
2. Transparency on how student data is used
3. Accessibility and inclusion requirements
4. Fairness, accountability in recommender systems

# 3.6 Requirement Analysis

# 3.6.1 Software Requirements

1. Operating System: Windows
2. Database: PostgreSQL
3. Application program: VS Code
4. Next.js

# 3.6.2 Hardware Requirements

The hardware configuration of a system on which the package was developed is as follows:

1. HP EliteBook
2. 8GB RAM
3. Browser

# 3.7 Requirements Specifications

Table 3. Functional Requirement Specifications

| **Req. No.** | **Description** | **Type** |
| --- | --- | --- |
| R-101 | The server shall Windows 7 or later version. | Configuration |
| R-102 | A user shall be able to sign up. | Functional |
| R-103 | A user shall be able to view available rooms. | Functional |
| R-104 | A user shall be able to make payment | Functional |
| R-105 | Users shall be able to create and update their profile | Functional |
| R-106 | A user shall be able to view matches. | Functional |
| R-107 | The admin shall be able to view pending payments | Functional |
| R-108 | The admin shall be able to add rooms. | Functional |
| R-109 | The admin shall be able to update room. | Functional |
| R-110 | The admin shall be able to delete a room. | Functional |

# 

# Table 3.2 Non-Functional Requirement Specifications

|  |  |  |
| --- | --- | --- |
| **Req.**  **No.** | **Description** | **Type** |
| R-101 | When launched, the application shall stay running unless there is an intentional shutdown of the application or the platform. | Performance |
| R-102 | Availability the system is available to everyone | Performance |
| R-103 | The system should be easy to use and user-friendly | Usability |
| R-104 | The application shall be maintained efficiently | Efficiency |

# 3.8 System Design

# 3.8.1 Application Architecture

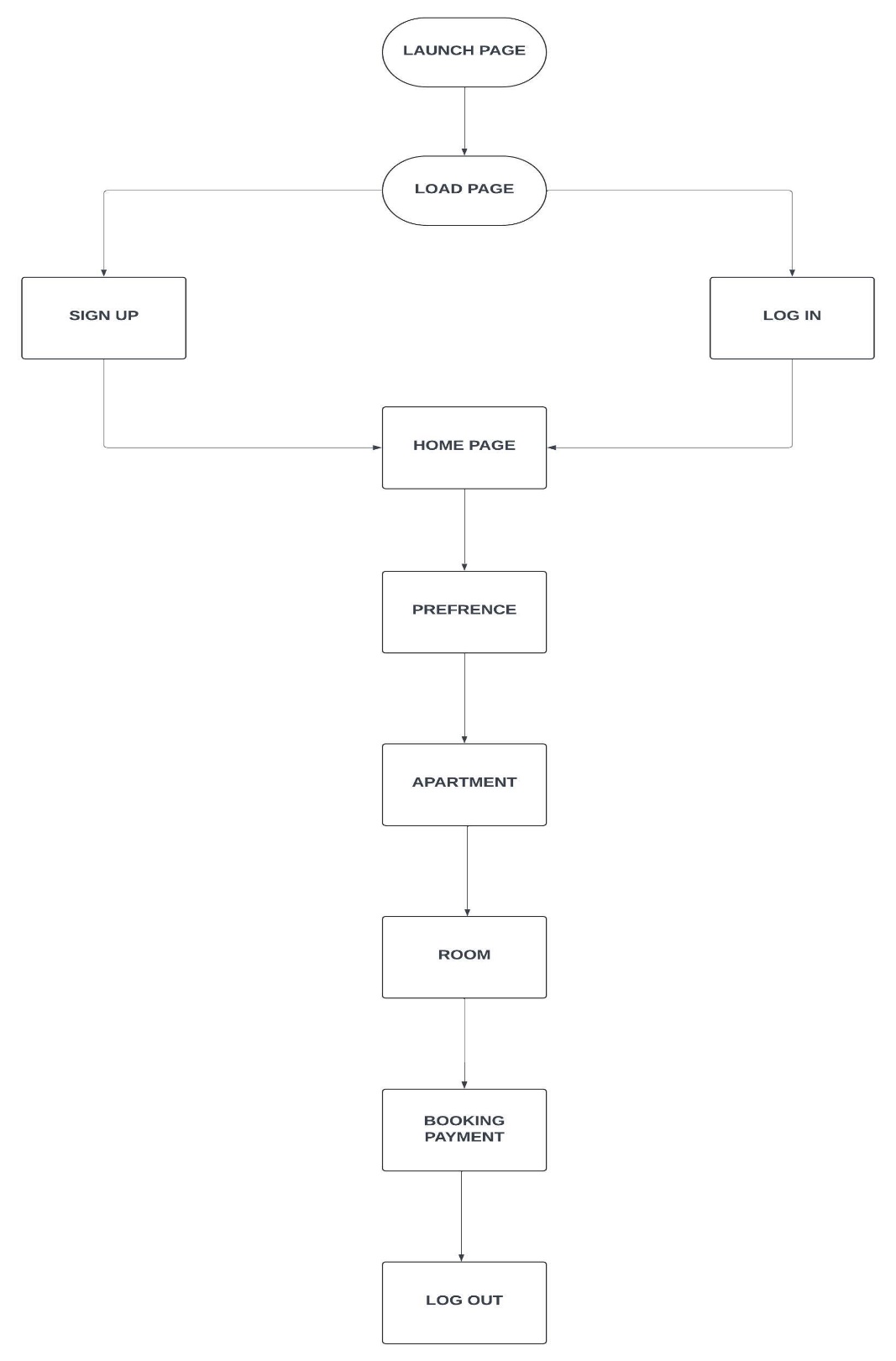


Figure 3.2 System Architecture for Student

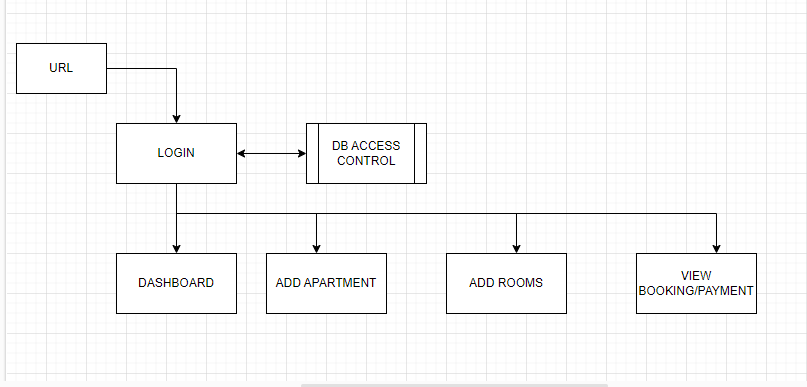


Figure 3.3 System Architecture for Admin

# 3.8.2 Use Case

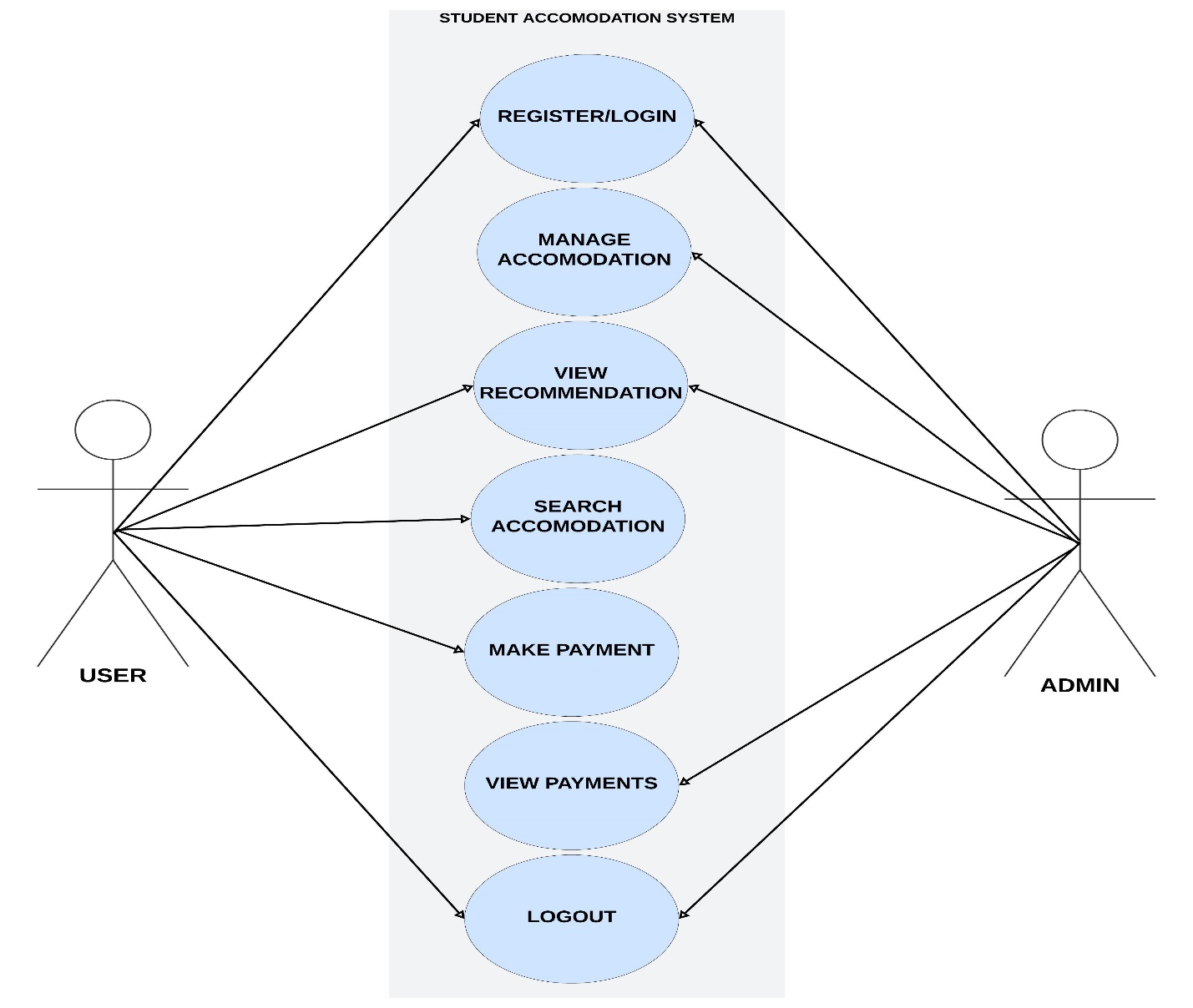


Figure 3.4: Use case Diagram

# 3.8.2.1 Use-Case Description

**Table 3.3 Use-Case Description for Login/Register**

|  |  |  |
| --- | --- | --- |
| USE CASE | LOGIN/REGISTER | |
| Description: | This use case describes the process of logging in to an existing account or registering a new user account within the accommodation management system. | |
| Actors: | User | |
|  |  | |
| Pre-condition: | none | |
| Post condition | Successful login grants access to the system.  Successful registration creates a new user account | |
| Main flow | User   1. User selects the login or register option. 2. If the user selects login, the user provides their login credentials. 3. If the user selects register, the user provides their registration details. | System   1. System validates credentials. 2. The system validates the provided login or registration details. 3. If the validation is successful, the user is logged into the application or registered in the application. 4. Use case ends |
| Exception Conditions: | * Invalid login credentials result in an error message. User can retry or cancel, ending the use case.   Invalid registration result in an error message. User can retry or cancel, ending the use case. | |

**Table 3.4 Use-Case Description for Search Accommodation**

|  |  |  |
| --- | --- | --- |
| Use Case: | Search Accommodation | |
| Description: | This use case describes the process of a user searching for available accommodations based on specific location | |
| Actor: | User | |
| Preconditions: | User is logged in  Accommodation data is available in the system | |
| Post conditions: | A list of matching accommodations is displayed to the user. | |
| Main Flow: | User   1. User Selects “Search Accommodation”. 2. User selects Available location. 3. User selects Available Rooms. | System   1. System searches the database for available Accommodation. 2. System displays list of search result. |
| Exception condition: | No accommodations found matching the searched location | |

**Table 3.5 Use-Case Description for View Payment(Admin)**

|  |  |  |
| --- | --- | --- |
| Use Case: | View Payment | |
| Description: | This use case describes the process of an administrator viewing payment history for accommodations. | |
| Actor: | Admin | |
| Preconditions: | Admin is logged in | |
| Post conditions: | Payment information is displayed. | |
| Main Flow: | Admin   1. Admin selects "View Payments". 2. Admin can filter or search for specific payments. | System   1. System searches the database for available Accommodation. 2. System displays list of search result |
| Exception condition: | No payment data available.  System error during data retrieval. | |

**3.8.3 Activity Diagrams**

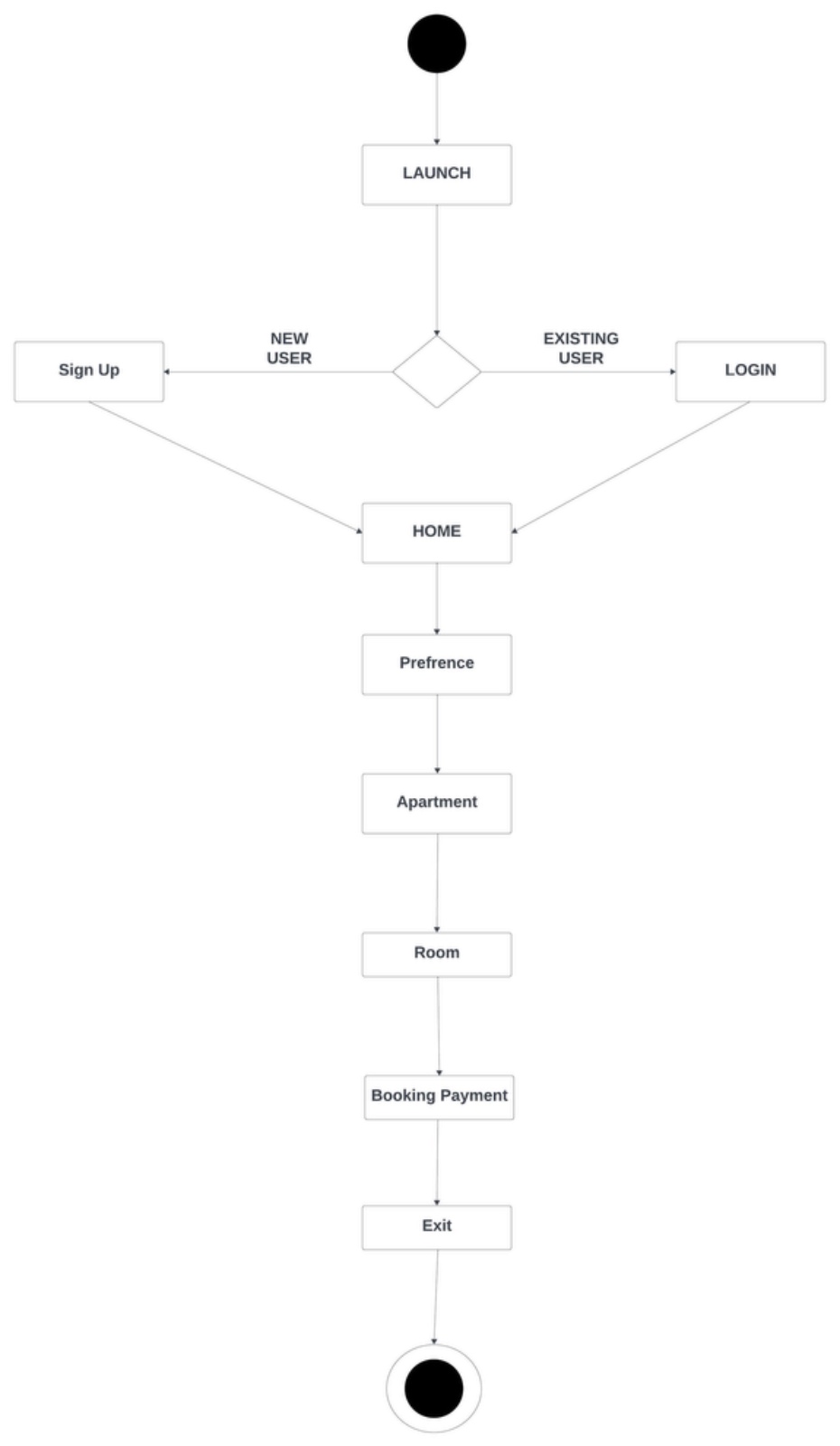


Figure 3.5: Activity Diagram

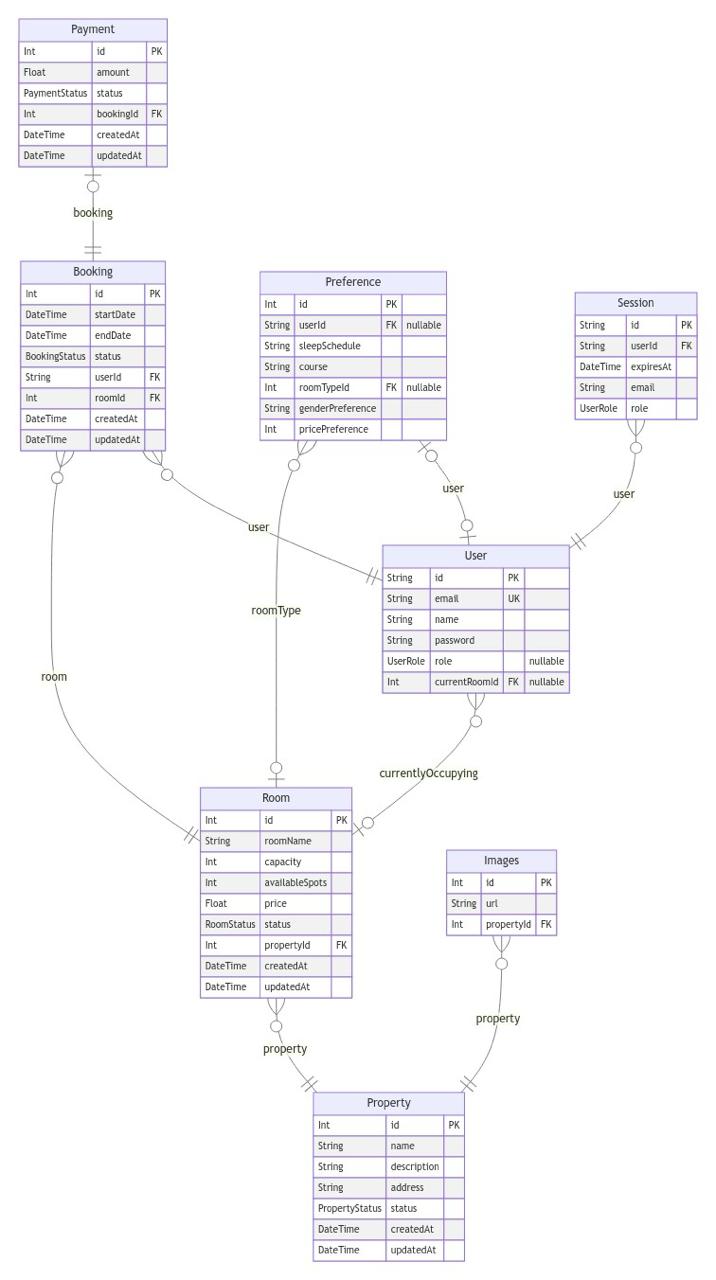


Figure 3.6: Entity Relationship Diagram

# CHAPTER 4: IMPLEMENTATION AND TESTING

# 4.1 Overview

This chapter discusses the practical development of the system and other related factors that assisted in developing the system including the front-end, back-end, database and the challenges encountered and solutions to the challenges encountered.

# 4.2 Main Features

1. The Student Accommodation System Using Recommendation System is designed to make the process of finding accommodation smoother and more personalized for students while streamlining management for administrators.
2. User Registration and Login: The system includes secure login functionality, where both students and administrators can sign in using their credentials. Role-specific access ensures that users can see and interact only with the features relevant to their role.
3. Personalized Roommate Matching: Students input their living preferences, such as cleanliness, social habits, sleep schedules, and noise tolerance. The system uses these preferences to match them with available rooms and suitable roommates, helping students find accommodations that best fit their lifestyle.
4. Accommodation Availability Filtering: Based on the student’s preferences and budget, the system filters through available rooms to recommend the most compatible options. Only rooms that meet the student’s criteria are shown, simplifying the decision-making process.
5. Roommate Compatibility Algorithm: The system calculates a match score between the student and potential roommates by comparing their preferences. Rooms are then ranked based on compatibility, ensuring students are paired with roommates who are most likely to get along.
6. Payment System: Once students select their preferred accommodation, they can proceed to make payments directly through the platform. The system supports secure payment processing, allowing students to pay for their accommodation fees or deposits online.
7. User Management: Administrators have full control over managing user accounts, including adding or removing users, assigning roles, and managing permissions. They can also track student activities and respond to issues related to room assignments or payments

# 4.3 Implementation Problems

Challenges faced during development included:

1. Incomplete information from students.
2. Difficulty in verifying the authenticity of the provided data.
3. Insufficient data to make accurate recommendation.
4. Data inconsistencies and formatting issues.
5. Technical issues and errors.

# 4.4 Overcoming Implementation problem

1. Ensuring user’s personal information id protected and secured.
2. Standardized data formats to ensure consistency.
3. Ensuring secure payment processing.

# 4.5 Testing

Testing was carried out to ensure the application passed its requirements. The application was tested from the perspective of a user to ensure that the efficiency and reliability of the application were put to test. Other forms of testing such as unit testing ensured that the functions of the application from the front-end to the back-end were examined and tested.

**Table 4.1 Testing for User Sign up**

|  |  |
| --- | --- |
| Test Case | User sign up |
| Related Requirement | FR01 |
| prerequisites | User has access to the sign up page |
| Test procedure | 1. Navigate to the sign up page 2. Enter the required information 3. Click on “sign up” button |
| Test Data | User Information |
| Expected Result | Account Created |
| Actual Result | Account Created |
| Status | pass |
| Remark | None |
| Created by | Maryam Abba Yusuf |
| Date of Creation | 10th August,2024 |
| Executed By | Maryam Abba Yusuf |
| Date of Execution | 10th August, 2024 |
| Test Environment | HP Laptop |

**Table 4.2 Testing for User sign in**

|  |  |
| --- | --- |
| Test Case | User sign in |
| Related Requirement | FR02 |
| prerequisites | 1. User has a valid account 2. User has access to sign in page. |
| Test procedure | 1. Navigate to the log in page 2. Enter valid username and password 3. Click on the “Sign in” button |
| Test Data | Valid Username and password |
| Expected Result | User should successfully sign in and be redirected to the home page, or a page to set preferences if not already set. |
| Actual Result |  |
| Status | pass |
| Remark | None |
| Created By | Maryam Abba Yusuf |
| Date of Creation | 10th August,2024 |
| Executed By | Maryam Abba Yusuf |
| Date of Execution | 10th August, 2024 |
| Test Environment | HP Laptop |

**Table 4.3 Testing for Admin**

|  |  |
| --- | --- |
| Test Case | Admin |
| Related Requirement | FR03 |
| prerequisites | 1. Admin has a valid account and is signed in. 2. Admin has access to the admin management dashboard. |
| Test procedure | 1. Login as Admin. 2. Review current room availability and occupancy status. 3. Update room availability (e.g., mark rooms as available or occupied). 4. save changes and verify that updates are reflected in the system. 5. View pending payment |
| Test Data | Room and Apartment status updates |
| Expected Result | Room availability and preferences should be updated successfully. The changes should be reflected in the system for users to see. |
| Actual Result |  |
| Status | pass |
| Remark | None |
| Created By | Maryam Abba Yusuf |
| Date of Creation | 10th August,2024 |
| Executed By | Maryam Abba Yusuf |
| Date of Execution | 10th August, 2024 |
| Test Environment | HP Laptop |

**Table 4.4 Testing for preference**

|  |  |
| --- | --- |
| Test Case | User Preference |
| Related Requirement | FR04 |
| prerequisites | User has a valid account  User has access to the preference page |
| Test procedure | 1. Select gender preference (Male, Female, Other). 2. Adjust sliders for cleanliness, socialness, quietness, and sleep schedule. 3. Click on the “Submit” button |
| Test Data | Selected gender preference and values for cleanliness, socialness, quietness, and sleep schedule. |
| Expected Result | Preferences should be successfully saved, and the user should be redirected to a room recommendations page |
| Actual Result |  |
| Status | pass |
| Remark | None |
| Created By | Maryam Abba Yusuf |
| Date of Creation | 10th August,2024 |
| Executed By | Maryam Abba Yusuf |
| Date of Execution | 10th August, 2024 |
| Test Environment | HP Laptop |

**Table 4.5 Testing for Room Selection**

|  |  |
| --- | --- |
| Test Case | User Room Selection |
| Related Requirement | FR05 |
| prerequisites | 1. User has a valid account and is signed in. 2. User has chosen an apartment and set their preferences. 3. User has navigated to the room selection page. |
| Test procedure | |  | | --- | | 1. Navigate to the room selection page. |  1. Review available rooms based on the chosen apartment and user preferences. 2. Select a room from the available options. 3. Confirm the room selection and proceed to the next step (e.g., payment). |
| Test Data | Selected apartment and user preferences |
| Expected Result | The system should display available rooms that match the selected apartment and preferences. The user should be able to select a room |
| Actual Result |  |
| Status | pass |
| Remark | None |
| Created By | Maryam Abba Yusuf |
| Date of Creation | 10th August,2024 |
| Executed By | Maryam Abba Yusuf |
| Date of Execution | 10th August, 2024 |
| Test Environment | HP Laptop |

**Table 4.6 Testing for payment**

|  |  |
| --- | --- |
| Test Case | User Payment |
| Related Requirement | FR06 |
| prerequisites | 1. User has a valid account and is signed in. 2. User has selected a room and ready to make a payment. 3. Payment gateway is integrated with the system. |
| Test procedure | |  | | --- | | Navigate to the payment page after selecting a room. |  |  |  | | --- | --- | |  | Enter valid payment details(credit card, Bank Transfer etc.). |  |  |  | | --- | --- | |  | Confirm the amount and click the "Pay Now" button. |  |  |  | | --- | --- | |  | Receive confirmation of successful payment. | |
| Test Data | Valid payment details (e.g., credit card number, Bank Transfer) |
| Expected Result | The payment should be successfully processed, and a confirmation message or receipt should be displayed. The user should be redirected to a "Payment Successful" |
| Actual Result |  |
| Status | pass |
| Remark | None |
| Created By | Maryam Abba Yusuf |
| Date of Creation | 10th August,2024 |
| Executed By | Maryam Abba Yusuf |
| Date of Execution | 10th August, 2024 |
| Test Environment | HP Laptop |

# 4.5 Use Guide

1. **User Registration and Login:**

Open the system in a web browser.

If you are a new user, click on the "Register" button and provide the required information to create an account.

If you already have an account, click on the "Login" button and enter your credentials.

1. **Administrators**

After logging in, administrators will have access to administrative functions.

View and manage student and accommodation accounts.

Perform administrative tasks, such as managing room listings, updating accommodation details, assigning students to rooms, and overseeing payment processes.

1. **Student (User)**

After logging in, students can input their preferences for accommodation and roommates (cleanliness, quietness, social habits, etc.).

View recommended rooms and potential roommates based on their preferences.

Make secure payments for their selected accommodation directly through the system.

# 4.7 User Interface Design

**Admin’s Page**

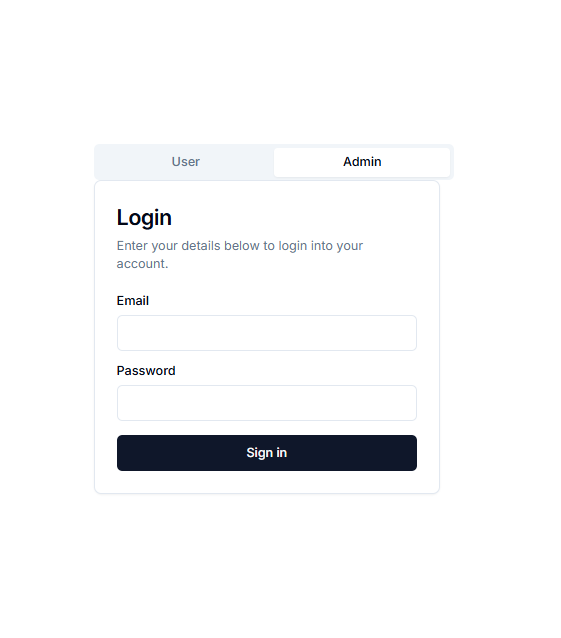
****

Figure 4.1: Login Page

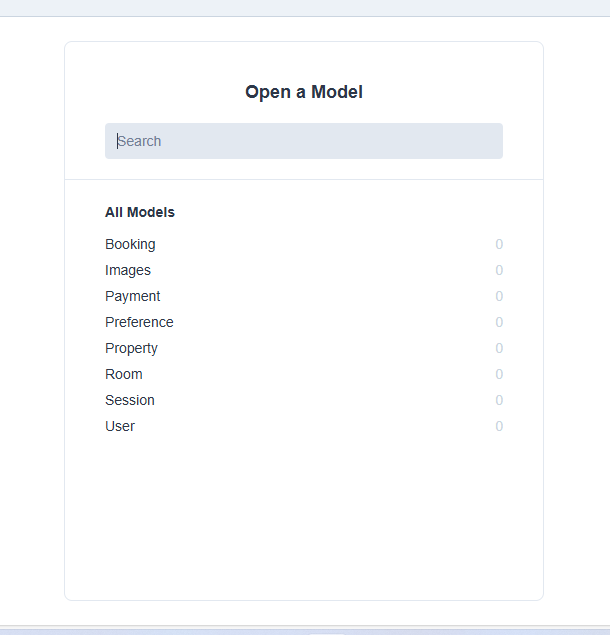
****

Figure 4.2: Model page

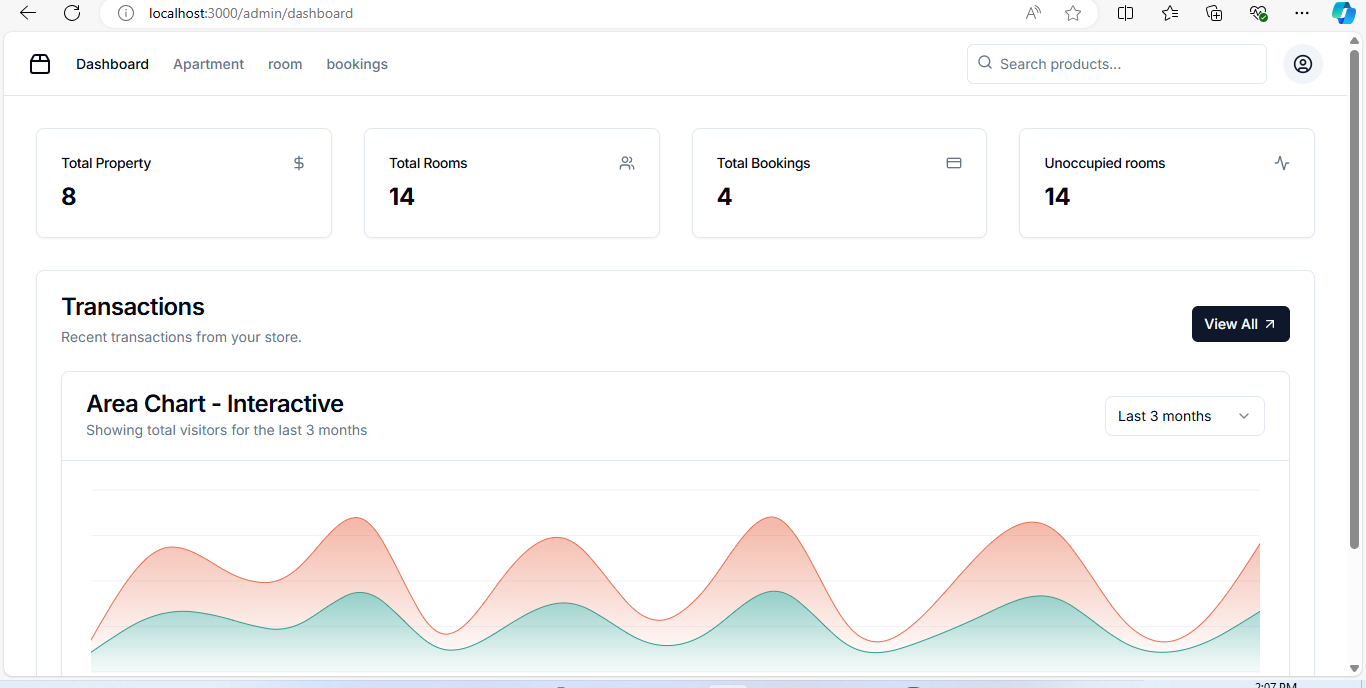
****

Figure 4.3: Dashboard

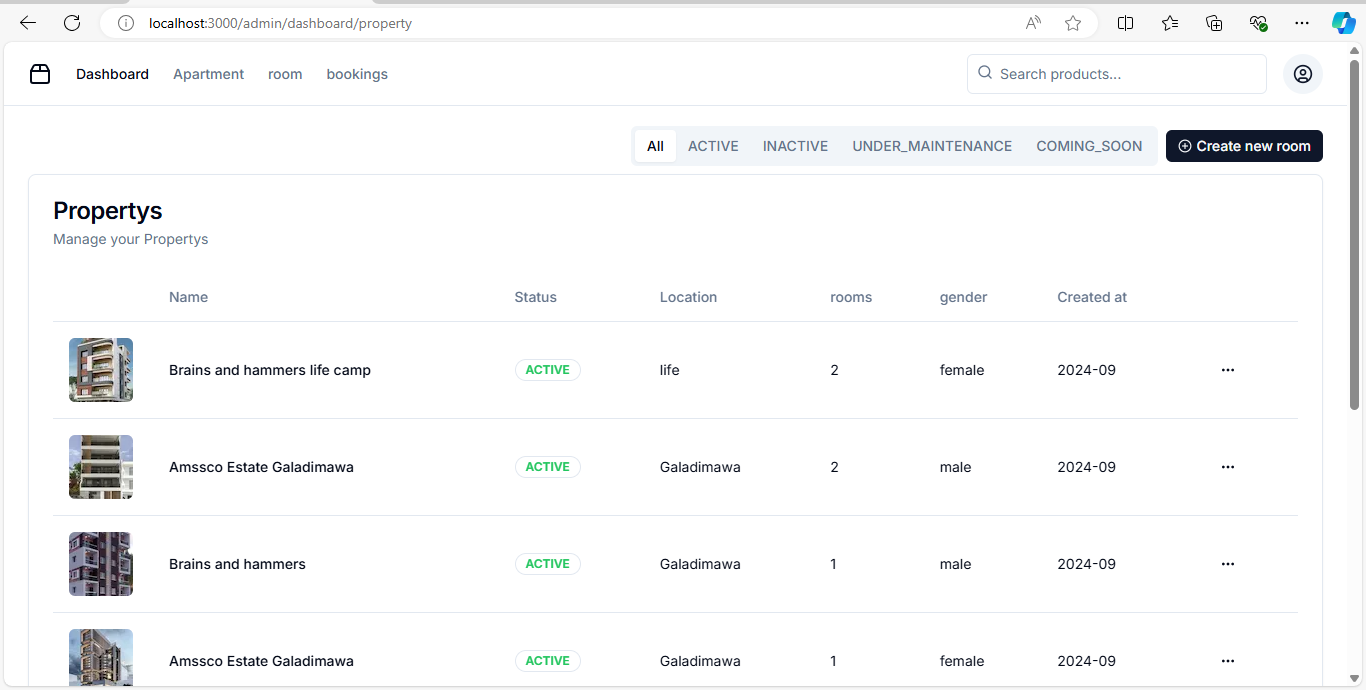
****

Figure 4.4: Apartment page

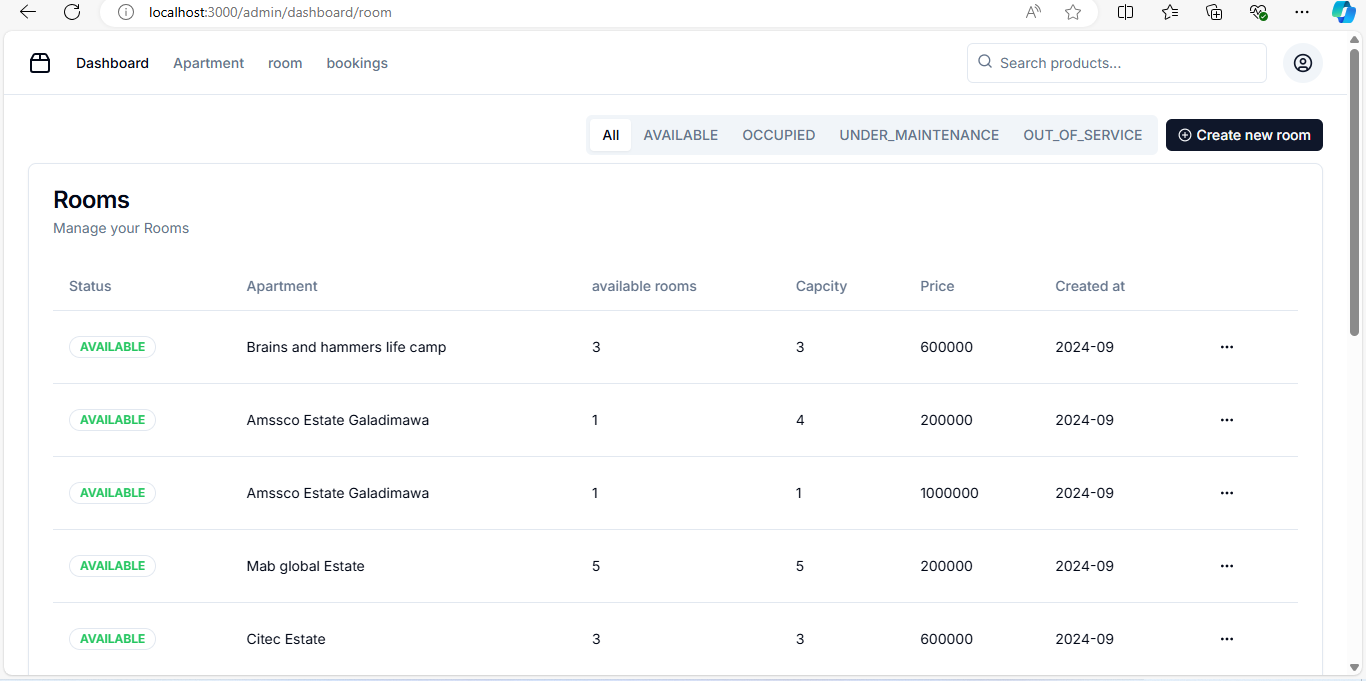
****

Figure 4.5: Room page

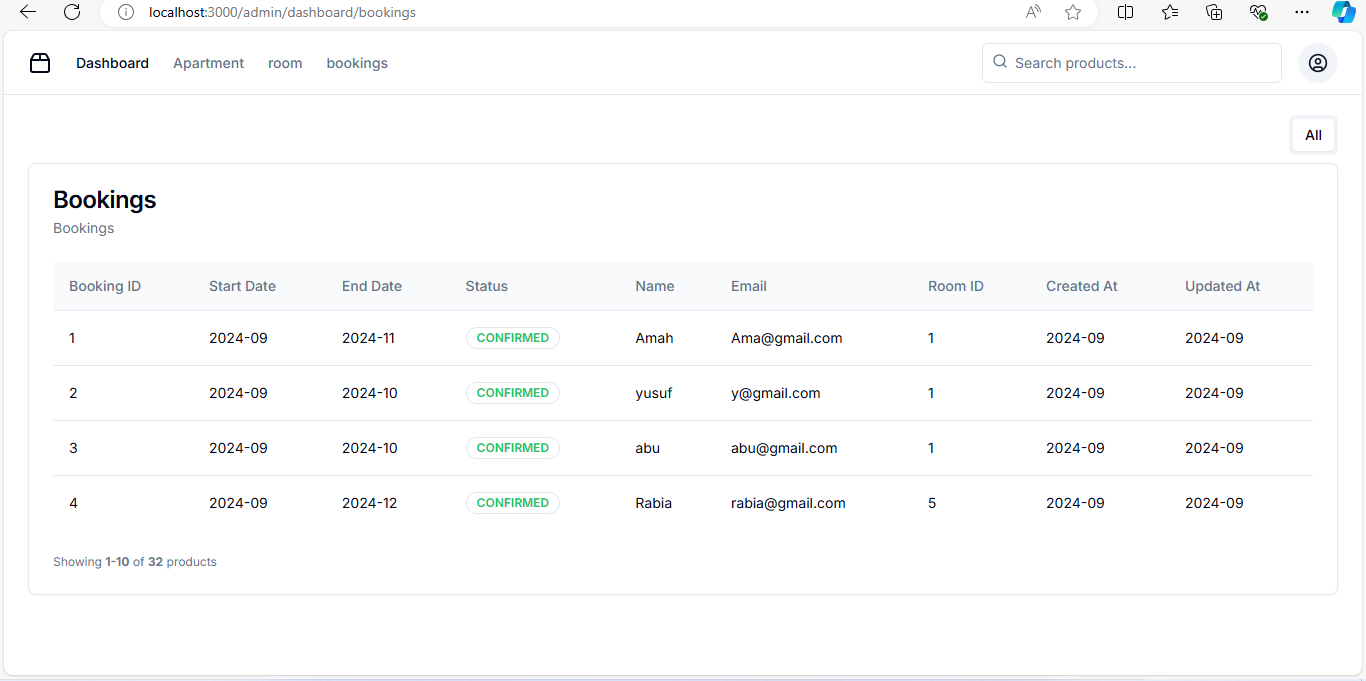
****

Figure 4.6: Booking/payment page

**User’s View**

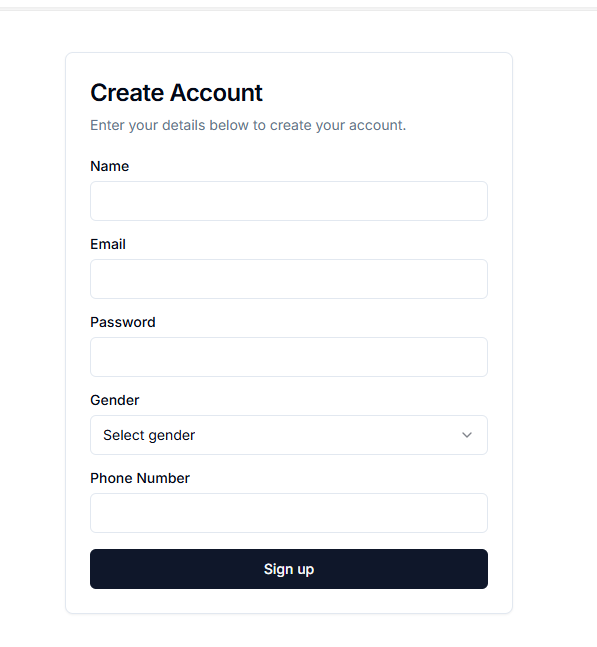
****

Figure 4.7: Sign up page

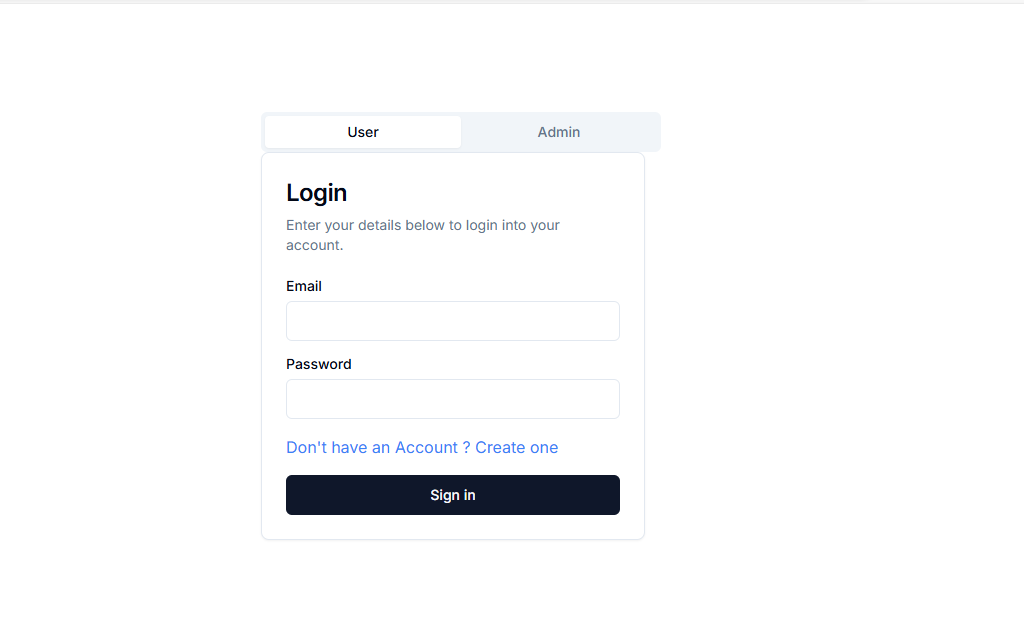
****

Figure 4.8: Login page

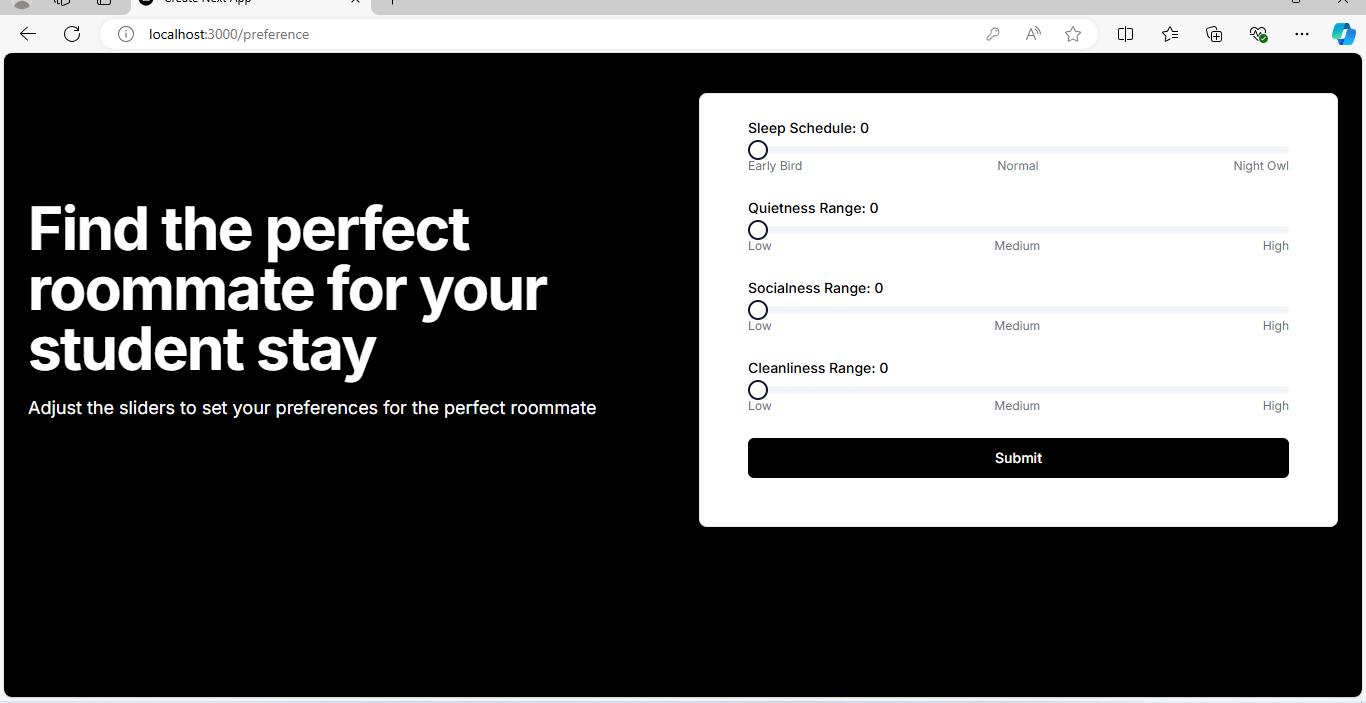
****

Figure 4.9: Preference page

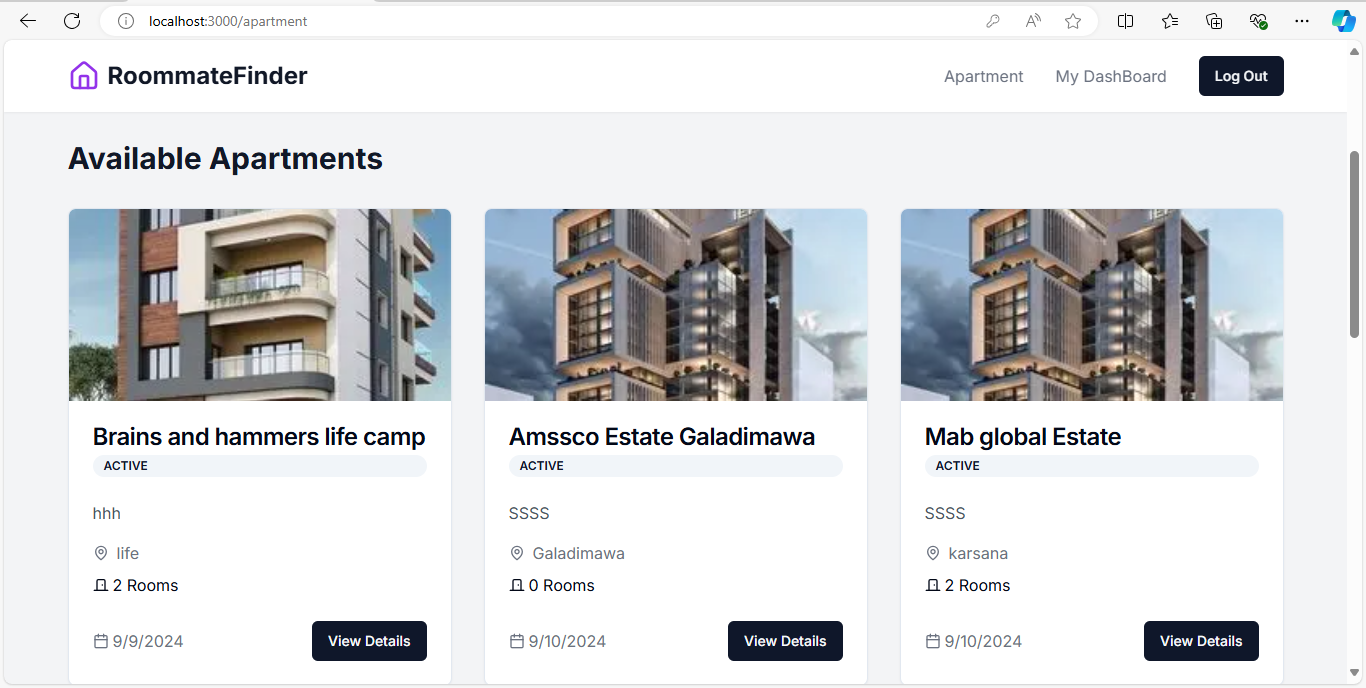
****

Figure 4.10: Apartment page

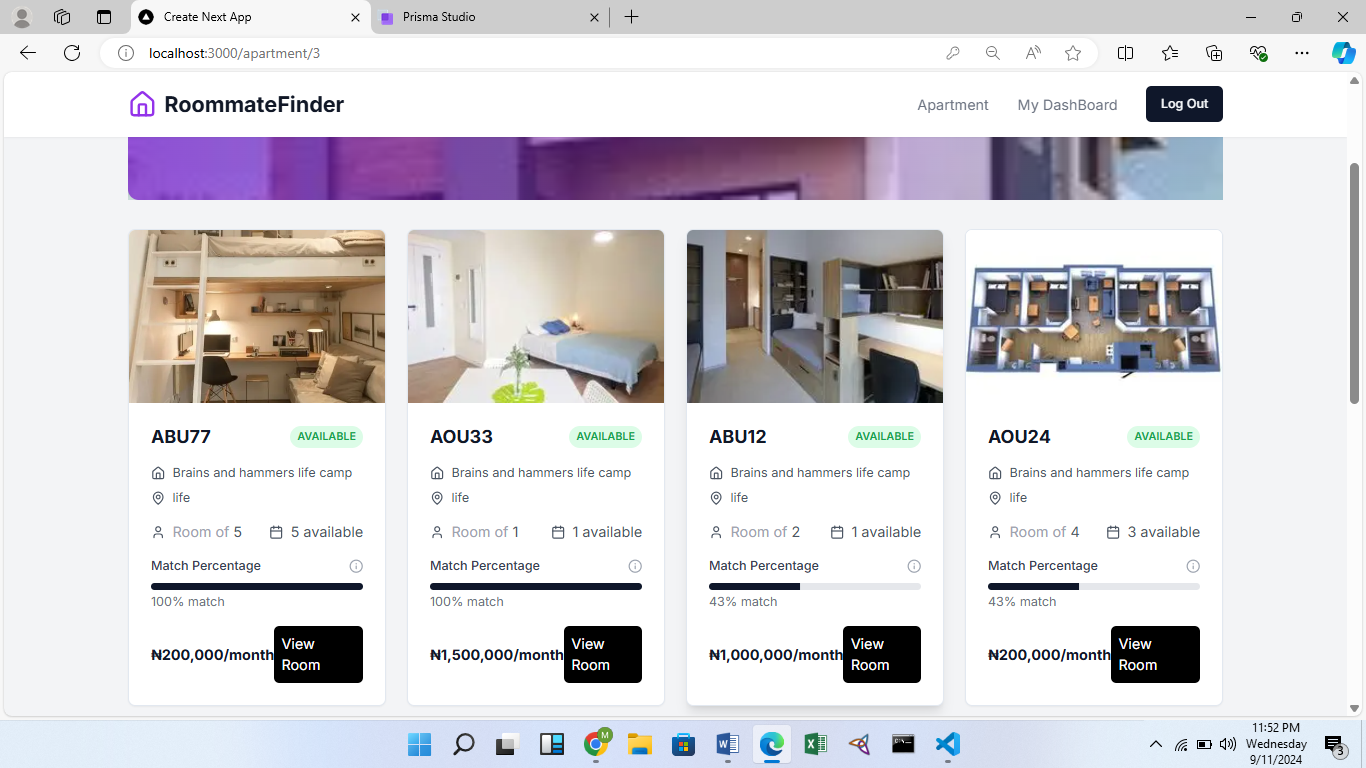


Figure 4.11: Room page

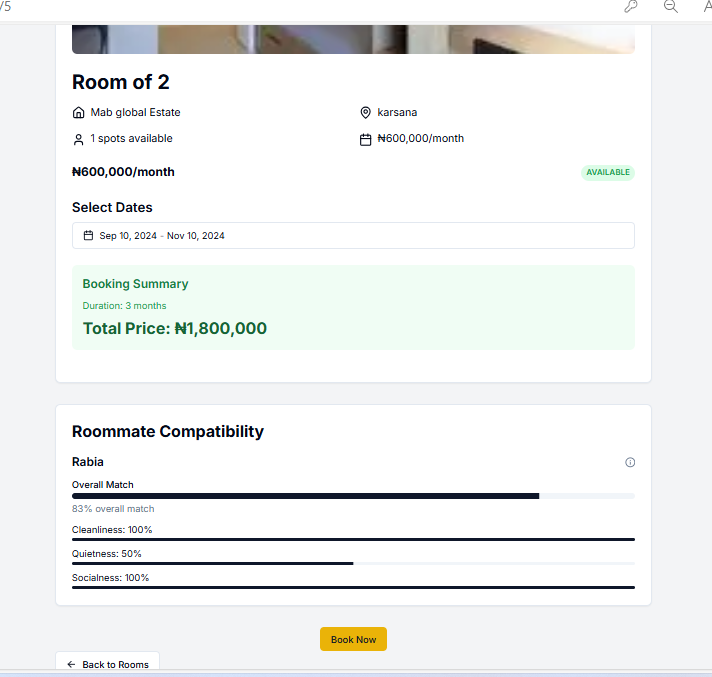
****

Figure 4.12: Booking Page

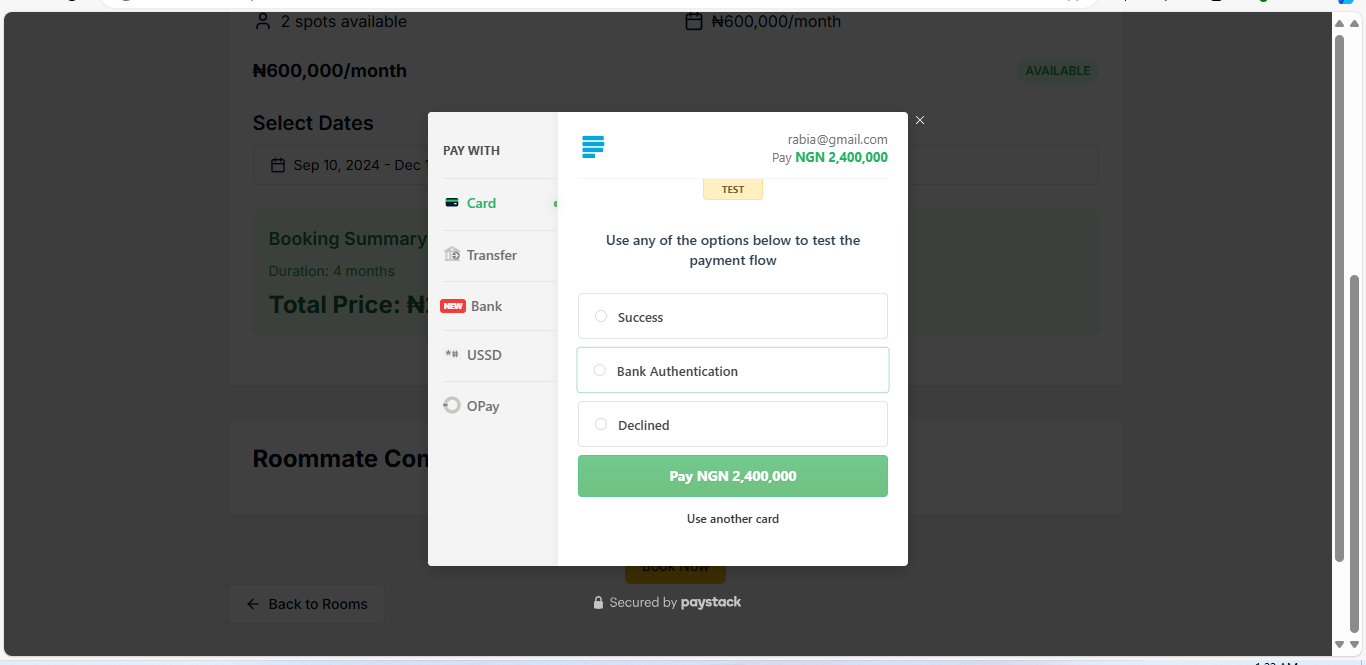
****

Figure 4.13: Payment page

# CHAPTER 5: DISCUSSION, CONCLUSION AND RECOMMENDATION

# 5.1 Overview

This concluding chapter provides a comprehensive reflection on the implementation of the student accommodation recommendation system. It evaluates the achievements relative to the initial objectives of the project. Additionally, it highlights the main challenges encountered and the system's limitations. The chapter concludes with recommendations for future enhancements, along with final observations.

# 5.2 Objective Assessment

* 1. Automating the process of assigning accommodations to students.
  2. Providing personalized housing suggestions based on each student's needs and preferences.
  3. Bringing together all accommodation options in one place for easier browsing.
  4. Using data analysis to enhance the accuracy of the housing recommendations.

The system built during this project effectively matches students' preferences with available housing options. This has significantly reduced the manual work required to manage student accommodation requests, making the process easier for both students and staff.

# 5.3 Limitations and Challenges

1. The mobile interface for the system is underdeveloped, leading to usability issues on smaller devices.
2. The recommendation engine requires more extensive data sources to improve its accuracy.
3. Customizable reports for staff and administrators are limited and require additional technical support.
4. Student self-service functionalities such as payment integration and room change requests remain limited.

The project has accumulated technical debt due to rapid development phases, and this will require dedicated resources for optimization, security, and future-proofing. Additionally, there are concerns regarding long-term maintenance and support, which will require a well-trained team to handle future updates.

# 5.4 Future Enhancements

1. Build mobile apps for iOS and Android to make the system easier to use.
2. Improve the recommendation engine by using more data sources, including outside housing databases.
3. Make reporting easier for staff by creating a simple tool that doesn't require technical help.
4. Add self-service features for students, like payment options and automatic room change requests.

# 5.5 Recommendations

1. Create rules to make sure the system and its processes are always improving.
2. Encourage everyone to use the system by offering training.
3. Set a reminder for renewal of payments
4. Set up a plan for feedback from users to keep improving the system.
5. Keep the system focused on students' needs to make sure it improves their accommodation experience.

# 5.6 Summary

The student accommodation recommendation system is a big step toward making the process more efficient and tailored for students. However, to fully benefit from the system, ongoing support from leadership and stakeholders is needed. With regular updates, proper resources, and smart improvements, the system can boost student satisfaction and improve how things are run. By using modern technology, the institution can become a leader in student housing, setting the stage for future success.

# REFERENCES

Aggarwal, C. C. (2016). Recommender systems. Springer International Publishing.

Burke, R. (2001). Knowledge-based recommender systems. In Encyclopedia of library and information science.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370.

Carpenter, S. R., Pingali, P. L., Bennett, E. M., & Zurek, M. B. (Eds.). (2005). Ecosystems and human well-being: Scenarios, volume 2. Island Press.

Jain, A., & Gupta, C. (2018). Fuzzy logic in recommender systems. In Fuzzy logic augmentation of neural and optimization algorithms: Theoretical aspects and real applications (pp. 255-273). Springer.

Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). The measurement of meaning. University of Illinois Press.

Basavesh, D., Laharishree, S., Sthuthi, S., Tejaswini, N., & Vidya, R. (2023). Hostel finder: Location-based recommendation system for hostels and PGS with transit information. International Journal of Research Publication and Reviews. <https://www.ijrpr.com>

Ikejiaku, B. V. (2009). The relationship between poverty, conflict and development. Journal of Sustainable Development, 2(1), 15. https://doi.org/10.5539/jsd.v2n1p15

Isinkaye, F., Folajimi, Y., & Ojokoh, B. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal, 16, 100-105. https://doi.org/10.1016/j.eij.2015.06.005

Omisore, M. O., & Samuel, O. W. (2014). Personalized recommender system for digital libraries. International Journal of Web-Based Learning and Teaching Technologies, 9(1), 18-32. https://doi.org/10.4018/IJWLTT.2014010102

Ojokoh, B., Omisore, M., Samuel, O., & Ogunniyi, T. (2012). A fuzzy logic based personalized recommender system. International Journal of Computer Science and Information Technology and Security, 2(5), 1008-1015.

Omisore, M. O., Samuel, O. W., & Ogunniyi, T. O. (2013). A multi-technique approach for recommender systems. International Journal of Advanced Trends in Computer Science and Engineering, 2(5), 112-118. https://doi.org/10.30534/ijatcse/2013/02252013

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International World Wide Web Conference (pp. 285-295). ACM.

Daly, E. M., Botea, A., Kishimoto, A., & Marinescu, R. (2014). Multi-criteria journey aware housing recommender system. In Proceedings of the 8th ACM Conference on Recommender Systems (pp. 325-328). ACM.

Lorenzi, F., Fontanella, B., Prestes, E., & Peres, A. (2014). How to improve multi-agent recommendations using data from social networks? In FLAIRS Conference.

Pizzato, L., Rej, T., Chung, T., Koprinska, I., & Kay, J. (2010). RECON: A reciprocal recommender for online dating. In Proceedings of the Fourth ACM Conference on Recommender Systems (pp. 207-214). ACM.

Shanmuganathan, M., & Karthikeyan, R. (2016). A web-based recommendation system for housing selection: Design, implementation & evaluation. Data Mining and Knowledge Engineering, 8(5), 134-137. https://doi.org/10.11648/j.dmke.2016.08.05.14

Schafer, J. B., Konstan, J., & Riedl, J. (1999). Recommender systems in e-commerce. In Proceedings of the 1st ACM Conference on Electronic Commerce (pp. 158-166). ACM.

Schafer, J. B., Konstan, J. A., & Riedl, J. (2002). Meta-recommendation systems: User-controlled integration of diverse recommendations. In Proceedings of the Eleventh International Conference on Information and Knowledge Management (pp. 43-51). ACM.

Terveen, L., Hill, W., Amento, B., McDonald, D., & Creter, J. (1997). PHOAKS: A system for sharing recommendations. Communications of the ACM, 40(3), 59-62. https://doi.org/10.1145/253610.253623

Yuan, X., Lee, J. H., Kim, S. J., & Kim, Y. H. (2013). Toward a user-oriented recommendation system for real estate websites. Information Systems, 38(2), 231-243. https://doi.org/10.1016/j.is.2012.10.003

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering (Technical Report MSR-TR-98-12). Microsoft Research.

Zain, J., & Khurshed, T. (2020). Hostel recommender system [Unpublished final year project]. Department of Computer Science and IT, Bahria University Lahore Campus.

Desai, U. (2023, April 27). Recommendation systems explained: Understanding the basic to advance. Medium. <https://utsavdesai26.medium.com/Apr> 27, 2023