[Project Title]

Thesis Submitted in Partial Fulfilment of the Requirement

for the Degree of

B.Sc.

In

Computer Science [Student Major]

By

[Student NAME]

To

The Department of Computer Science

Baze University, Abuja

[Month, Year]

**DECLARATION**

This is to certify that this Thesis entitled [Project Title], which is submitted by [Student Name] in partial fulfilment of the requirement for the award of degree for B.Sc. in Information Technology to the Department of Computer Science, Baze University Abuja, Nigeria, comprises of only my original work and due acknowledgement has been made in the text to all other materials used.

Date: [Date Month Year] Name of Student: [Your Name]

**APPROVED BY**  …………………

**HOD**

Dept. of Computer Science

**CERTIFICATION**

This is to certify that this Thesis entitled [Your Project Title], which is submitted by [YOUR

NAME] in partial fulfilment of the requirement for the award of degree for B.Sc. in

Information Technology to the Department of Computer Science, Baze University Abuja, Nigeria is a record of the candidate’s own work carried out by the candidate under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Date: Supervisor:

**APPROVAL**

This is to certify that the research work, Dental Management System and the subsequent preparation by [Your Name] with [Student ID] has been approved by the Department of Computer Science, Faculty of Computing and Applied Science, Baze University, Abuja, Nigeria.

By

|  |  |  |
| --- | --- | --- |
| [Full name]  1st Supervisor |  | Date |

|  |  |
| --- | --- |
| [Full name]  2nd Supervisor | Date |

|  |  |
| --- | --- |
| Dr. C. V Uppin  Head of Department | Date |

|  |  |
| --- | --- |
| Prof. Peter Ogedebe  Dean, Faculty of Computing and Applied Science | Date |

Date

External Examiner

**DEDICATION**

[This is the dedication page.]

**ABSTRACT**

[The abstract provides a clear summary of the project, indicating both content and tone of the project. An abstract includes the method(s) used to analyze the problem, a brief description of the research design, a listing of the key results, a brief description of the significance of the results, selected key conclusions. First-person narrative should not be used in the abstract.]

**TABLE OF CONTENTS**

**ABSTRACT VI**

**LIST OF TABLES….……………………………………….…………………………………………………………….IX**

**LIST OF FIGURES………………………………..….…………………………………...…………………………......X**

**LIST OF ABBREVIATIONS……..……………………….……………...……………………………………….....XI**

**CHAPTER 1: INTRODUCTION 1**

1.1 OVERVIEW 1

1.2 BACKGROUND AND MOTIVATION 1

1.3 STATEMENT OF THE PROBLEM 2

1.4 AIM AND OBJECTIVES 2

1.5 SIGNIFICANCE OF THE PROJECT 2

1.6 PROJECT RISKS ASSESSMENT 3

1.7 SCOPE/PROJECT ORGANIZATION 4

**CHAPTER 2: LITERATURE REVIEW 5**

2.1 INTRODUCTION 5

2.2 HISTORICAL OVERVIEW 5

2.3 RELATED WORK 6

2.4 SUMMARY 6

**CHAPTER 3: REQUIREMENTS, ANALYSIS, AND DESIGN 7**

3.1 OVERVIEW 7

3.2 PROPOSED MODEL 7

3.3 METHODOLOGY 7

3.3.1 METHOD 1 (E.G INTERVIEW) 7

3.3.2 METHOD 2 (E.G OBSERVATION) 7

3.4 TOOLS AND TECHNIQUES 7

3.5 ETHICAL CONSIDERATION 7

3.6 REQUIREMENT ANALYSIS 7

3.7 REQUIREMENTS SPECIFICATIONS 7

*3.7.1 Functional Requirement Specifications 8*

*3.7.2 Non-Functional Requirement Specifications 8*

3.8 SYSTEM DESIGN 9

*3.8.1 Application Architecture 9*

*3.8.2 Use Case 9*

*3.8.3 Data Design 10*

*3.8.4 Activity Diagrams 10*

*3.8.5 Dataflow Diagram 10*

*3.8.6 Control Flow Diagram 10*

*3.8.7 Entity-Relationship Diagram (ERD) 11*

*3.8.8 User Interface Design 11*

*3.9 Summary 11*

**CHAPTER 4: IMPLEMENTATION AND TESTING 12**

4.1 OVERVIEW 12

4.2 MAIN FEATURES 12

4.3 IMPLEMENTATION PROBLEMS 12

4.4 OVERCOMING IMPLEMENTATION PROBLEMS 12

4.5 TESTING 12

*4.5.1 Tests Plans (for Unit Testing, Integration Testing, and System Testing) 12*

*4.5.2 Test Suite (for Unit Testing, Integration Testing, and System Testing) 13*

*4.5.3 Test Traceability Matrix (for Unit Testing, Integration Testing, and System Testing) 13*

*4.5.4 Test Report Summary (for Unit Testing, Integration Testing, and System Testing) 13*

*4.5.5 Error Reports and Corrections 13*

4.6 USE GUIDE 14

4.7 SUMMARY 14

**CHAPTER 5: DISCUSSION, CONCLUSION, AND RECOMMENDATIONS 15**

5.1 OVERVIEW 15

5.2 OBJECTIVE ASSESSMENT 15

5.3 LIMITATIONS AND CHALLENGES 15

5.4 FUTURE ENHANCEMENTS 15

5.5 RECOMMENDATIONS 15

5.6 SUMMARY 15

**REFERENCES 16**

**APPENDICES 17**

**LIST OF TABLES**

TABLE 1 FUNCTIONAL REQUIREMENT SPECIFICATIONS 7

TABLE 2 NON-FUNCTIONAL REQUIREMENT SPECIFICATIONS 8

**LIST OF FIGURES**

FIGURE 1 USE CASE DIAGRAM 10

FIGURE 2 ACTIVITY DIAGRAM 11

FIGURE 1 ENTITY RELATIONSHIP DIAGRAM 12

**LIST OF ABBREVIATIONS**

CPU Central Processing Unit

ERD Entity Relationship Diagram

IT Information Technology

# ABSTRACT

Many students face problems in finding suitable accommodation after getting admission in university. A Hostel Recommender system that allows users to find hostels as per their requirements. It contains lists of hostels which are added by admins. This application will be collecting feedback by students who has already resided or currently residing there. It will calculate that how many students will like their hostels or how many students dislike and show in rating. Our system is recommending hostel to new students and show that this hostel is suitable for you. It helps users to find appropriate hostel rooms at distant places even before visiting. This kind of system reduces student's problem of accommodation. The best thing about this system is that, it allows users to provide a feedback, rating to hostel about their experience which helps other students. This system is beneficial to students and hostel owners. Hostel owners can also benefit from this system as their reach increases in city. In this system we used collaborative filtering algorithm which are make automatic predictions about a user's interests by compiling preferences from several users.

**CHAPTER 1: INTRODUCTION**

**Important**: The main work is presented in one and half (1½) line spacing. It is preferable to use either Times New Roman or Arial size twelve (12 pt). Your documentation must be justified “Full Text”; i.e. both the left and right margins must be aligned.

**1.1 Overview**

This project focuses on developing a modern student accommodation system for a client in Abuja, utilizing an advanced recommender system. The aim is to provide students with personalized accommodation recommendations based on their preferences. By automating the search process, the system will save students significant time and effort. It will use data analytics and machine learning to make informed recommendations and optimize housing resource allocation. It will improve communication between students and housing providers.

**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 dormitories 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.

In the early 2000s, the first digital platforms for student accommodation emerged. These early systems were simple databases that listed available properties, allowing students to search based on basic criteria like location and price (StudentHousing, 2022).

The introduction of recommender systems marked a significant advancement in the student accommodation process. Early recommender systems used basic algorithms to suggest housing options based on limited criteria such as price range and proximity to campus (Smith, 2020). As technology advanced, these systems began incorporating more complex algorithms and data analytics to provide more personalized and accurate recommendations.

As student enrollment and the complexity of housing options continue to grow, existing manual and legacy processes are increasingly inadequate for handling the diverse needs and preferences of students.

By implementing a modern student accommodation system with advanced recommender capabilities, our client can address current inefficiencies and improve the overall housing experience for students, ultimately contributing to their academic success and well-being.

**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

Accommodation

**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 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.

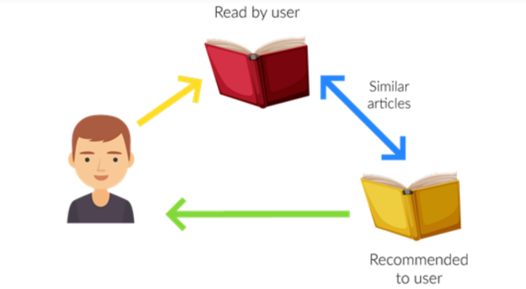


Fig.2.1 Recommendation system

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.

**2.4.1 How Recommendation System Work**

There are four steps of ow recommendation system work:

**1. Collecting user data:**

The first step in building a recommendation system is to collect user data. This can include user ratings, reviews, clickstream data, purchase history, and other behavioral data. The data can be collected either explicitly, through user surveys or feedback forms, or implicitly, through user interactions with the platform (Utsav Desai 2023).

**2. Storing the data:**

Once the user data is collected, it needs to be stored in a database or data warehouse for analysis. The data can be stored in a structured or unstructured format, depending on the type and volume of the data.

**3. Analyzing the data:**

The next step is to analyze the user data to identify patterns and trends. This can be done using various data analysis techniques like clustering, classification, and regression. The goal is to understand the user’s preferences, behaviors, and interests, and to use this information to make personalized recommendations (Utsav Desai 2023).

**4. Filtering and Recommending:**

The final step is to filter the data and make recommendations to the user. This can be done using various recommendation algorithms, such as collaborative, content-based, and hybrid filtering. The algorithm uses the user data and the analysis results to generate a list of recommended items the user will likely be interested in. The recommendations are then presented to the user in a personalized way, such as through a recommendation widget, email, or push notification.

These four steps are the basic components of most recommendation systems, and the specific implementation details may vary depending on the type of system and the application domain.

**2.4.2 Types of Recommendation Systems**

There are three main types of recommendation systems

**2.4.2.1 Content-Based Filtering**

Content-based recommendation systems recommend items to users based on their past preferences and behaviors. This type of system analyzes the user’s historical data, such as their search history, browsing history, or purchase history, and recommends items that are similar to the ones the user has interacted with before.

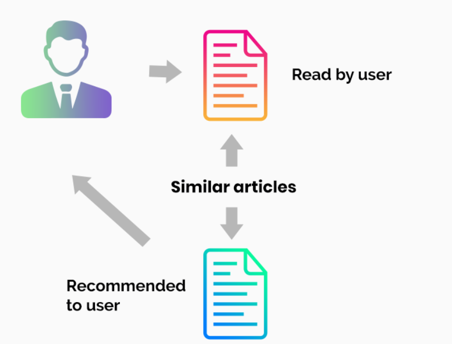


Fig.2.2Content-based filtering

For example, if a user has watched several action movies in the past, a content-based recommendation system might recommend similar action movies to the user. if a user likes to watch movies such as Iron Man, the recommender system recommends movies of the superhero genre or films describing Tony Stark.

**2.4.2.2 Collaborative Filtering**

Collaborative filtering recommendation systems recommend items to users based on the preferences and behaviors of other similar users. This type of system analyzes the user’s historical data, as well as the data of other users with similar preferences, and recommends items that similar users have liked or interacted with before. For example, if two users have similar purchase histories, a collaborative filtering recommendation system might recommend items that one user has purchased to the other user.

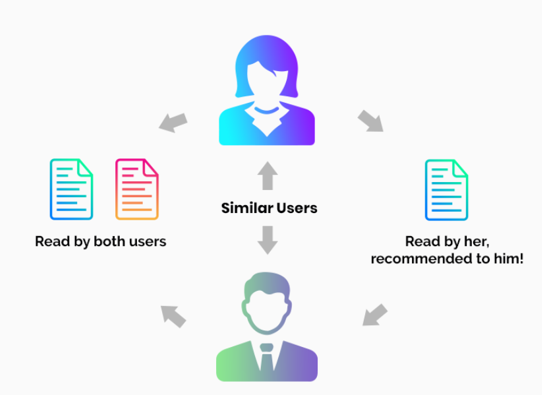


Fig.2.3 Collaborative filtering

For example, if user A likes Apples, Bananas, and Mango while user B likes Apples, Bananas, and Jackfruit, they have similar interests. So, it is highly likely that A would like Jackfruit and B would enjoy Mango. This is how collaborative filtering takes place.

Two kinds of collaborative filtering techniques used are:

User-User collaborative filtering

Item-Item collaborative filtering

**User-User collaborative filtering** is a type of recommendation system that makes predictions for a user based on the preferences of similar users. It works by finding users with similar tastes and recommending items they liked to the target user. **Item-Item collaborative filtering,**on the other hand, recommends items to a user based on the preferences for similar items. It works by identifying items that are similar to the ones a user has liked in the past and recommending them to the user (Utsav Desai 2023).

**2.4.2.3 Hybrid Recommendation Systems**

Hybrid recommendation systems combine both content-based and collaborative filtering techniques to provide more accurate and diverse recommendations. This type of system uses a combination of user data, item data, and other contextual information to generate recommendations. Hybrid recommendation system might use content-based filtering to recommend items that are similar to the ones the user has interacted with before, and collaborative filtering to recommend items that other similar users have liked or interacted with. By combining the strengths of both approaches, hybrid recommendation systems can provide more accurate and diverse recommendations than either content-based or collaborative filtering alone.

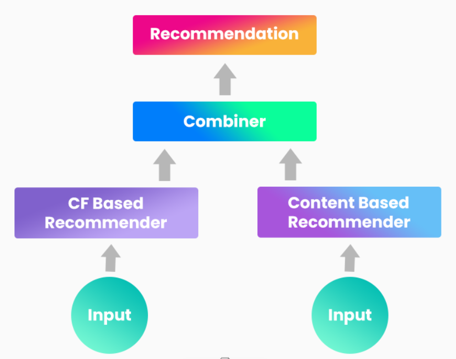


Fig.2.4 hybrid recommendation system

Netflix is an excellent case in point for a hybrid recommendation system. It makes recommendations by juxtaposing users’ watching and searching habits and finding similar users on that platform. This way, Netflix uses collaborative filtering.

By recommending such shows/movies that share similar traits with those rated highly by the user, Netflix uses content-based filtering. They can also veto the common issues in recommendation systems, such as cold start and data insufficiency issues.

**2.4.2.4 How YouTube algorithm work**

The YouTube recommendation algorithm is a complex system that uses a combination of collaborative filtering, deep learning, and other techniques to personalize video recommendations for each user (Utsav Desai 2023).

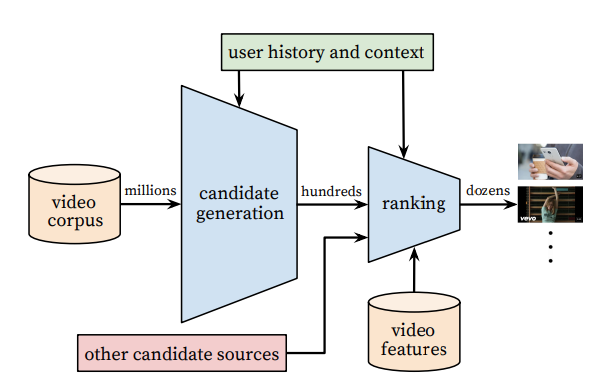


Fig.2.5 How YouTube algorithm work

Here are some key factors that the algorithm takes into account:

**User engagement:** The algorithm considers the videos a user has watched, liked, commented on, or shared to understand their preferences and interests.

**Similarity:** The algorithm identifies videos that are similar to the user’s viewing history, such as videos from the same channel or related topics.

**Popularity:** The algorithm takes into account the overall popularity of a video, such as the number of views, likes, and comments.

**Freshness:** The algorithm also considers the recency of the video to ensure that users are recommended the latest and most relevant content.

**Diversity:** The algorithm tries to recommend a diverse range of content to ensure that users are exposed to new and interesting videos outside of their typical viewing habits.

Overall, the YouTube algorithm is designed to provide personalized and engaging recommendations to each user while keeping them engaged and active on the platform (Utsav Desai 2023).

**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., (2019d). 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. Hence the mobile applications play a significant role in CARS. CARS plays an important role, especially in personalized and direct online marketing. 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 Works

|  |  |  |  |
| --- | --- | --- | --- |
| **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) | A Web-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, machine learning, 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, outlining the four essential steps: collecting user data, storing the data, analyzing the data, and filtering and recommending items. It also categorizes recommendation systems into three main types: content-based filtering, collaborative filtering, and hybrid systems, providing detailed explanations and examples for each category. the chapter explores how advanced recommendation algorithms, such as those used by YouTube, enhance user engagement by considering factors like user engagement, similarity, popularity, freshness, and diversity. It also reviews related work by various researchers, highlighting the strengths and weaknesses of different approaches and methodologies used in recommendation systems.

**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 relationship diagrams.

**3.2 Proposed Model**

**3.3 Methodology**

**3.3.1 Method 1 (e.g 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.

**3.3.2 Method 2 (e.g Observation)**

Observations were carried out to understand the user experience with existing accommodation platforms. This method provided insights into usability issues and features that are most valued by students. The observations informed the non-functional requirements, ensuring the system is user-friendly and efficient.

**3.4 Tools and Techniques.**

Next.js is been used on the front-end for structure, styling, and interactivity. Super base is used on the back-end to generate dynamic content and store/access data from a database. Django is used for the recommendation system. 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

**3.6 Requirement Analysis**

The system is subsequently deployed on the platform, in this case the web system, after passing all functional and non-functional requirements testing and approval by the testing team. After the system has been released, maintenance is essential to ensuring that it delivers consumers a seamless and enjoyable experience and does not irritate them. Regular maintenance is therefore necessary after the system has been implemented. These illustrations demonstrate the system’s overall operation and offer instructions for its application.

**3.7 Requirements Specifications**

The requirement Specification determine how the application should be developed. It describes the functional and non-functional requirement of the application.

**3.7.1 Functional Requirement Specifications**

# Table 1 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 can 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 |

**3.7.2 Non-Functional Requirement Specifications**

# Table 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**

[You are required to give a good introduction and brief explanation outlining the major components of the system.]

**3.8.1 Application Architecture**

Architecture of **student accommodation system (using recommendation system).** It consists of user interface, a medium through which users make interaction with the system, also a medium through which input is given to the system and output is displayed back to the user. The knowledge base includes the database and rule base. The database is a repository for storing information about rooms in student hostels, locations, user preferences, and user comments, user account information, the users account information, hostels information, and entire location.

**3.8.2 Use Case**

# 

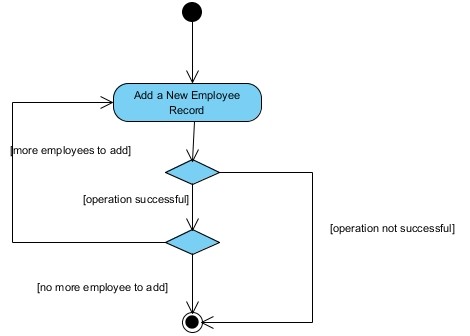
# Figure 1 Use Case diagram

*[A Use Case diagram depicts the interaction between the users and the system. It shows the functions of the system from the user’s point of view and the various actions the user as the actor carries out.]*

**3.8.3 Data Design**

**3.8.4 Activity Diagrams**

An activity diagram is a model that shows the process of a task or action from a use case.



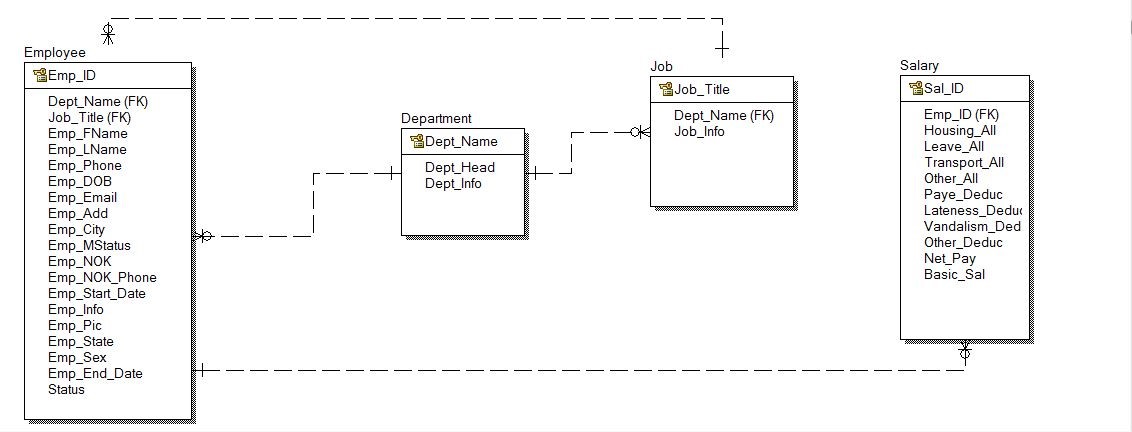
# Figure 2 Activity Diagram

**3.8.5 Dataflow Diagram**

**3.8.6 Control Flow Diagram**

**3.8.7 Entity-Relationship Diagram (ERD)**

[Entity-relationship diagrams show the entities and attributes of tables in a database. Linked ERDs show the relationship between tables or tables. Entities can only have a many-to-one or one-to-many relationship, e.g., in Figure A below.]



**Entity Relationship Diagram**

**3.8.8 User Interface Design**

**3.9 Summary**

**CHAPTER 4: IMPLEMENTATION AND TESTING**

**4.1 Overview**

[You are required to introduce the chapter.

The following materials listed below are the hardware and software components used for the implementation of the database system for which this report has been written.]

**4.2 Main Features**

**4.3 Implementation Problems**

**4.4 Overcoming Implementation Problems**

**4.5 Testing**

**4.5.1 Tests Plans (for Unit Testing, Integration Testing, and System Testing)**

**Figure xx Test Plan Tree**

**4.5.2 Test Suite (for Unit Testing, Integration Testing, and System Testing)**

**Table xx Test Suite Performed**

|  |  |  |
| --- | --- | --- |
| **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 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**4.5.3 Test Traceability Matrix (for Unit Testing, Integration Testing, and**

**System Testing)**

**4.5.4 Test Report Summary (for Unit Testing, Integration Testing, and**

**System Testing)**

**4.5.5 Error Reports and Corrections**

**4.6 Use Guide**

**4.7 Summary**

**CHAPTER 5: DISCUSSION, CONCLUSION, AND RECOMMENDATIONS**

**5.1 Overview**

**5.2 Objective Assessment**

.

**5.3 Limitations and Challenges**

**5.4 Future Enhancements**

**5.5 Recommendations**

**5.6 Summary**

28

**REFERENCES**

*[APA Style or Harvard Referencing.]*

**APPENDICES**

**Appendix A - Project Document**

**Appendix B - Questionnaire**

or Proceedings of Interview or Observation Reports etc

**Appendix C – Source Codes**

**Appendix D – Test Cases**

**Appendix E – User Guide/Manual**

35