"AI BASED PERSONALIZED PRODUCT RECOMMENDATION SYSTEM"

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Information Technology

Project Mentor:
Name: Mr. Vipin Jain
Designation : Associate Professor

Submitted By :
Mohammed Arshad (19ESKIT056)
Mohit Singhal (19ESKIT057)
Mukul Palol (19ESKIT059)

Department of Information Technology
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota

Session 2022-2023

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that **Mr. Mohammed Arshad**, a student of B.Tech(Information Technology) semester has submitted his Project Report entitled ”**AI Based Personalized Product Recommendation System**” under my guidance.

Mentor	**Co-ordinator**
Name: Mr. Vipin Jain	Name: Ms. Sanju Choudhary
Designation: Associate Professor	Designation: Associate Professor
Signature:	Signature:

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that **Mr. Mohit Singhal**, a student of B.Tech(Information Technology) semester has submitted his Project Report entitled "**AI Based Personalized Product Recommendation System**" under my guidance.

Mentor	**Co-ordinator**
Name: Mr. Vipin Jain	Name: Ms. Sanju Choudhary
Designation: Associate Professor	Designation: Associate Professor
Signature:	Signature:

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that **Mr. Mukul Palol**, a student of B.Tech(Information Technology) semester has submitted his Project Report entitled "**AI Based Personalized Product Recommendation System**" under my guidance.

Mentor	**Co-ordinator**
Name: Mr. Vipin Jain	Name: Ms. Sanju Choudhary
Designation: Associate Professor	Designation: Associate Professor
Signature:	Signature:

DECLARATION

We hereby declare that the report of the project entitled "**AI Based Personalized Product Recommendation System**" is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of "**Mr. Vipin Jain**" (Dept. of Information Technology) and coordination of "**Ms. Sanju Choudhary**" (Dept. of Information Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Information Technology.It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Team Members	**Signature**
Mohammed Arshad (19ESKIT056)	
Mohit Singhal (19ESKIT057)	
Mukul Palol (19ESKIT059)	

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization.We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Mr. Vipin Jain. He has been a guide, motivator, source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Ms. Sanju Choudhary for her co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Prof. Dr. Anil Chaudhary, HOD, Department of Information Technology, for facilitating, motivating and supporting us during each phase of development of the project. Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

Team Members:
Mohammed Arshad (19ESKIT056)
Mohit Singhal (19ESKIT067)
Mukul Palol (19ESKIT059)

Table of Content

List of Figures

Chapter 1
Introduction

1.1 Problem Statement and Objective

Websites and applications that offer their users or customers an item or a product, have been trying to recommend them relevant products in order to their items/elements which they are interested in. Increasing the time that user spends on the website and increasing the interest of the user to the items in the website are the main reasons why the recommender systems are being used. Accuracy and time-efficiency are the most common problems of recommender systems. We will be trying to design an accurate and fast algorithm which will solve these problems.

People often complain about irrelevant recommendations of the websites they are using. User sometimes even complains about websites like Amazon or Netflix even though they are considered as having the best recommender systems. Since most of the global websites that offer their users an item, the problem can be considered as a worldwide problem.

For people who are totally unaware of how a recommendation system works, we can explain evaluation of the effectiveness of the recommender systems which is one of the most important issues to understand the topic. As it is explained in the Website of Creighton University, recall precision and DCG (Discounted Cumulative Gain) are the common metrics to assess the quality of the recommendation method. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage too. DCG measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks. These are the basic concepts related to calculating recommendations.

1.2 Introduction to Project

Recommender systems have achieved much commercial success and are becoming increasingly popular in a wide variety of practical applications. For example, online stores such as Amazon, iTunes and Walmart.com provide customized recommendations for additional products or services based on a consumer's history. Since it is widely believed that even minor improvements in recommender systems can boost the profitability of e-commerce companies, these systems have been studied intensively by researchers in industry and in academia.

An important limitation of existing studies is that they assume that the properties of items (i.e. products) are static. However, an e-commerce company could tailor some properties of a product for a particular customer, and that could dramatically improve the effectiveness of a recommendation. In particular, we argue that price is a controllable property that the recommender system should incorporate. A consumer might like a recommended product, but may reject it because the price is too high, and the purchasing decision could be changed by a personalized promotion. Of course, outside of the literature on recommender systems, the crucial role of pricing is widely recognized. Researchers in marketing, for example, have shown the importance of personalized promotion for increasing sales volume

1.3 Proposed Solution

It is aimed at replacing the overwhelming quantity and choices of products that user face. The system will collect data and store it for fast and easy reference. The system will provide users with a personalized recommendation based on their search input and previous orders. It will also provide recommendations to new user based on content based filtering. The system is thus helpful to reduce the time and complexity of maintaining the records.

The objective of this project is to implement a well-developed AI recommendation system that helps businesses improve their shopper's experience on website and result in better customer acquisition and retention by building an recommendation engine that recommends the customers about the best personalized Electronic Gadgets based on behavioral data and also analyze large sets of historical and real-time data to understand what each customer wants.

The module proposed for our recommendation system is Model-based collaborative filtering system based on customer's purchase history and ratings provided by other users who bought similar items.m

1.4 Scope of the Project

Neural Networks and Deep Learning have been all the rage the last couple of years in many different fields, and it appears that they are also helpful for solving recommendation system problems.

One of the benefits of Deep Learning is similar to matrix factorization, in that there is an ability to derive latent attributes. Deep Learning, however, can make up for some of the weaknesses of matrix factorization such as the inability to include time in the model — which standard matrix factorization isn't designed for. Deep Learning, however, can utilize Recurrent Neural Networks which are specifically designed for time and sequence data.

Incorporating time into a recommender system is important, because there are often preference seasonal effects. For example, it is likely that in December, more people are going to be watching holiday-themed movies and buying home decorations.

Chapter 2
Software Requirement Specification

2.1 Overall Description

It is aimed at replacing the overwhelming quantity and choices of products that user face. The system will collect data and store it for fast and easy reference. The system will provide users with a personalized recommendation based on their search input and previous orders. It will also provide recommendations to new user based on content based filtering. The system is thus helpful to reduce the time and complexity of maintaining the records.

The objective of this project is to implement a well-developed AI recommendation system that helps businesses improve their shopper's experience on website and result in better customer acquisition and retention by building an recommendation engine that recommends the customers about the best personalized Electronic Gadgets based on behavioral data and also analyze large sets of historical and real-time data to understand what each customer wants.

2.1.1 Product Perspective

2.1.1.1 User Interfaces

The user needs to click the link to the website. Then he/she needs to register to the system by providing a password and an email, otherwise he/she won't be able to use the Recommender System. Then, to benefit from the Recommender System he/she needs to be active on the website by listening and downloading music, adding his/her favorites to the list or sharing them on Facebook/Twitter.

2.1.1.2 Hardware Interfaces

- Minimum Requirements

Client Side			
	Processor	RAM	Disk Space
Google Chrome v84	Intel Pentium III or AMD -800 MHz	128MB	100MB
Server Side			
	Processor	RAM	Disk Space
NPM v6	Intel Pentium III or AMD -800 MHz	1GB	3.5GB
Python v2.7	Intel Pentium III or AMD -800 MHz	1GB	3.5GB

Table 2.1: Minimum Hardware Requirements

- Required Requirements

Client Side			
	Processor	RAM	Disk Space
IE10, Google Chrome 90	All Intel or AMD - 1GHZ	256MB	100MB
Server Side			
	Processor	RAM	Disk Space
NPM v7	All Intel or AMD - 2 GHZ	2GB	3.5GB
Python v3.3	All Intel or AMD - 2 GHZ	2GB	3.5GB

Table 2.2: Recommended Hardware Requirements

2.1.1.3 Software Interfaces

- **Client on Internet**

  Web Browser (Google Chrome v84, Internet Explorer v10)

- **Operating System**

  Microsoft Windows 7+, Linux, Ubuntu v20+, MacOSX

- **Development End**

  Jupyter Notebook, Python, NPM script engine for serving React 16+ application.

2.1.1.4 Communications Interfaces

Our system is a web-based application and hence it does not require much. This system supports Google Chrome & Mozilla Firefox web browsers.

2.1.1.5 Product Functions

With Product Recommender, the user will be able to see electronic gadget recommendations made especially for him/her. The recommendations will be based on the user's previous actions and the actions of the other users who have a similar taste of products as the user, who will get the recommendations. Since the recommendations are made based on the user, they are likely to be unique.

2.1.1.6 User Characteristics

Recommendation System is composed of the following fundamental users:

Users:

- Generate Data
- Get Recommendation

Inter-agent

- Get Recommendation
- Provide Dataset

- Update Dataset

2.1.1.7 Constraints

- The Internet connection is a constraint for the application. Since the application fetches data from the server over the Internet, it is crucial that there is an Internet connection for the application to function.
- The web portal will be constrained by the capacity of the database. Since the database is shared with the larger system, it may be forced to queue incoming requests and as a result, increase the time it takes to fetch data.
- The computers must be equipped with web browsers such as Internet explorer.
- Execution time for the algorithm should take no longer than one second.
- All Python code shall conform to the Python Code Convention standards.
- Product Recommender will be a sub-component of an e-commercial website.
- Users shall be required to log in to the website to get recommendations.
- Recommendation system shall be available to users 99.9% of the time when the e-commercial website is available.

- The system must be operational for each user. It also needs to give unique recommendations for each user.

2.1.1.8 Assumption and Dependencies

Every system requires some certain parameters to work, to work as per the requirement, our system also requires some parameters, and we assume them as fulfilled before using this system, which is as:

- Customers will have a username and password; else, they'll have to register themselves on our website.
- This software needs user to have complete knowledge of recommendation system and its working.
- Software is dependent on access of Internet, as it is a remote application, it is necessary to have internet access.
- Assume that all the information entered by the user will be correct. If any wrong information is found then the system will notify an alert.
- The system is required to save the generated reports.

2.2 Specific Requirements

2.2.1 Functional Requirements

This section outlines the use cases for each user registered to the website (larger system).

2.2.1.1 User Use Case

- **Use Case: Generate Data**

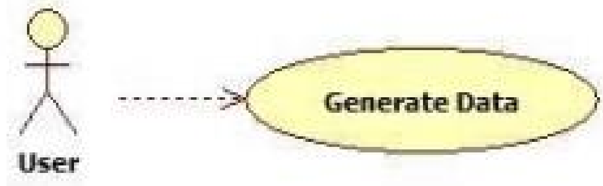

Figure 2.1: Use Case: Generate Data

User can search or buy gadgets from the e-commerce website. Product information is collected according to order information, price range, user information, time of action and rating value. This information will fill the database. Every product has a unique order, price, time of action, user and rating value. So, this process will generate practical data.

- **Use Case: Get Recommendation**

Figure 2.2: Use Case: Get Recommendation

System can suggest product(s) as a recommendation to user based on the dataset which is refined by users' collaborative approach. The main function of our system shows these products based on recommendation algorithms. When a user chooses recommendation part in application, he/she will get the most important point of the project recommendation and project gives user a chance to choose product through recommended tracks according to his/her own previous choices.

2.2.1.2 Inter-Agent Use Case

- **Use Case: Provide Dataset**

Figure 2.3: Use Case: Provide Dataset

Inter-agent provides dataset, in other words the big data, in cooperation with product purchasing and viewing application.

- **Use Case: Update Dataset**

Figure 2.4: Use Case: Update Dataset

An e-commerce web application collects millions of data every day. Inter-agent also delivers this data to our web service. These updates are necessary for making more accurate recommendations.

- **Use Case: Integrate Web Service**

Figure 2.5: Use Case: Integrate Web Service

After the recommendation system project is completed, inter-agent integrates this web service to recommending chatbot application. Then, users will access to our web service and receive recommendations through the application.

2.3 Non-Functional Requirements

2.3.1 Performance Requirements

Performance of making recommendation and updating this recommendation is very important issue because we are aiming to make the system real-timed. In other words, the system should have enough speed that users of the system cannot realize the processing of data. In order to make system real-timed, at the end of viewing a product or rating a product or after purchasing a product, the system shall update recommendations. Besides, our web service should handle multiple users at the same time.

2.3.2 Design Constraints

The main development language of the system is Python Programming Language, which is chosen as a standard by Oracle. IEEE standards and UML standards are used for documentation and diagrams. Portability is an important attribute of the system, as it must be portable to the larger system. Scalability is also important, as the larger system has the potential to increase its number of users, user traffic and products.

2.3.3 Security

Database has to be reached securely and its data should not be broken. It also should not change except inter-agent updates. Moreover, since our dataset contain some personal information of user such as user id, product he/she bought, security design is important in the web service

2.3.4 Usability

The scope of the product is widespread. The only requirement is using e-commerce website and downloading web application. Besides, people from every age shall easily use the system.

Chapter 3
System Design Specification

3.1 Proposed System Architecture

The model will take product name as an argument and give the top five recommendation using content based filtering approach. Content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback. The model will interact with user interface made by html and css and connected with flask.

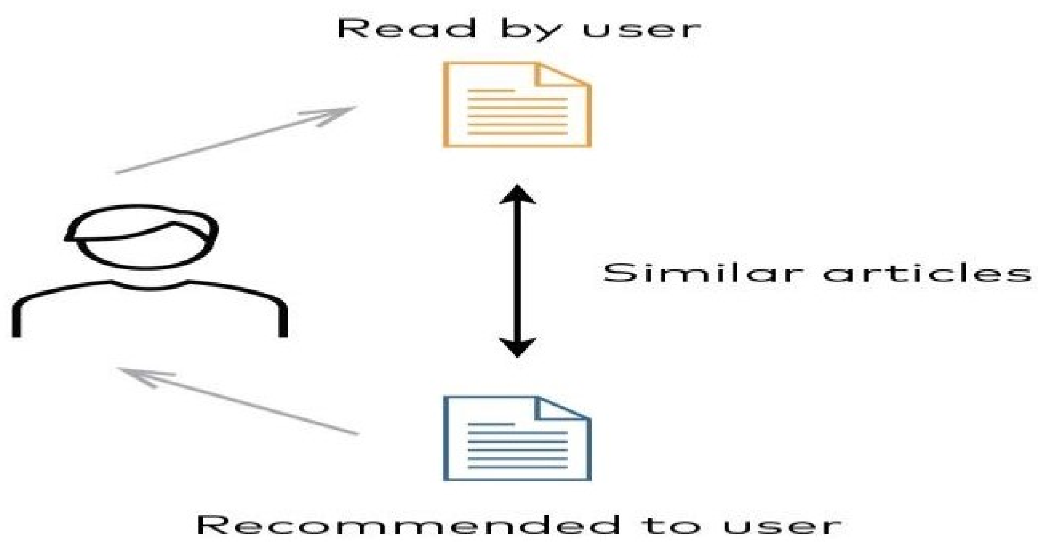

Figure 3.1: Proposed System Architecture

3.2 Module Decomposition Description

3.2.1 Block Diagram

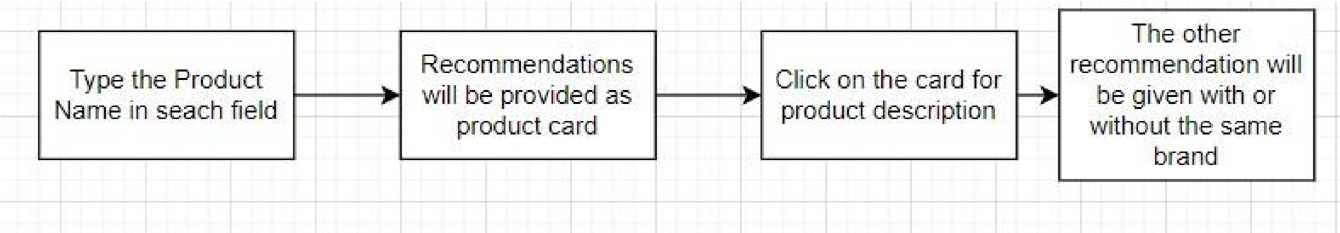

Figure 3.2: Block diagram for Proposed recommendation application

The proposed module of this project are as follows:

3.2.2 Front-End Designing

For front-end we are using react.js framework. React is an open source, JavaScript library for developing user interface (UI) in web application. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

3.2.3 Back-End Designing

For back-end we are using python. Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. It is also the high demand language for machine learning and data science based projects. The libraries like flask can be used to connect web pages with python based machine learning models.Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web application. Flask is based on the WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine.

Also to run model we are using juster notebook.The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience. program used to mix code, comments, and visualizations in an interactive document called notebook that can be shared, reused, and reworked in a web browser. Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating notebook documents. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots.

3.3 High Level Design Diagrams

3.3.1 Use Case Diagram

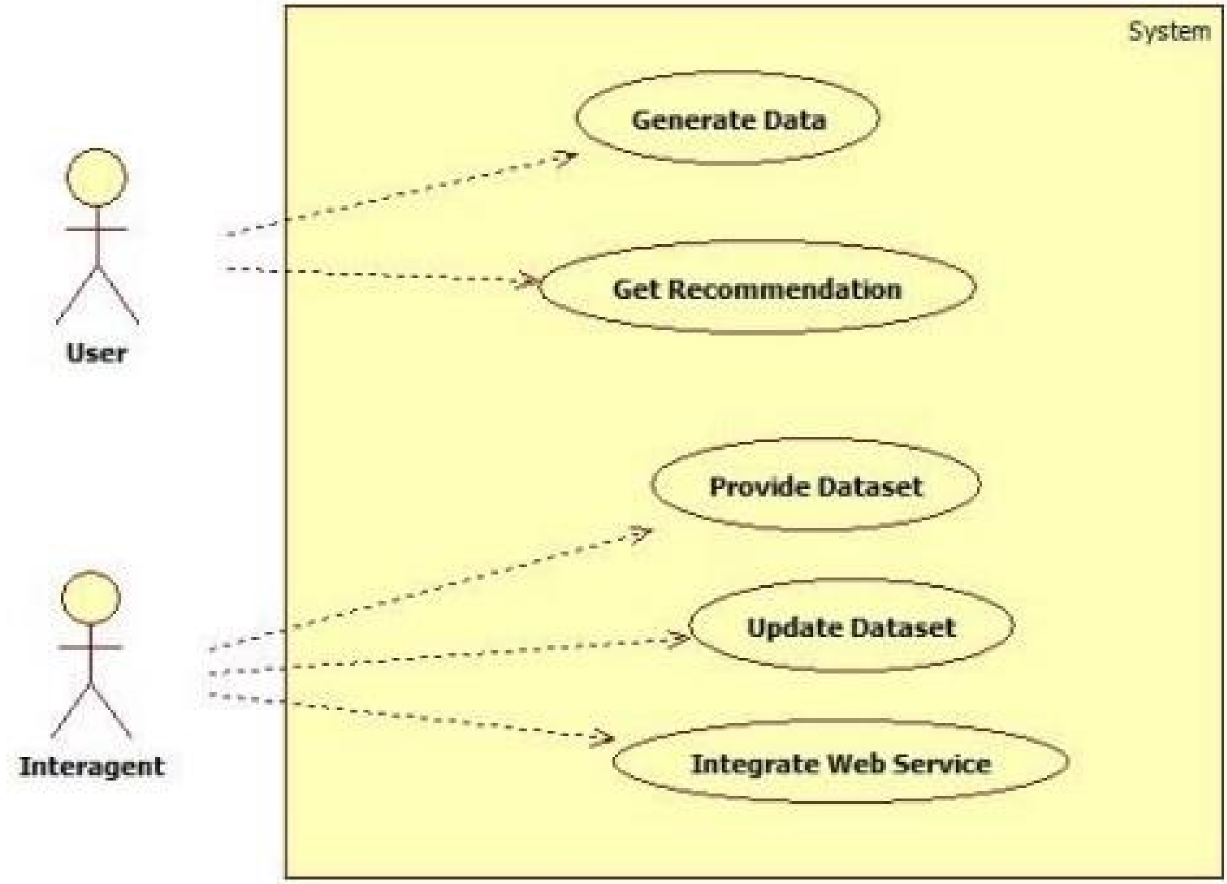

Figure 3.3: Use Case diagram

3.3.2 State Transition Diagram

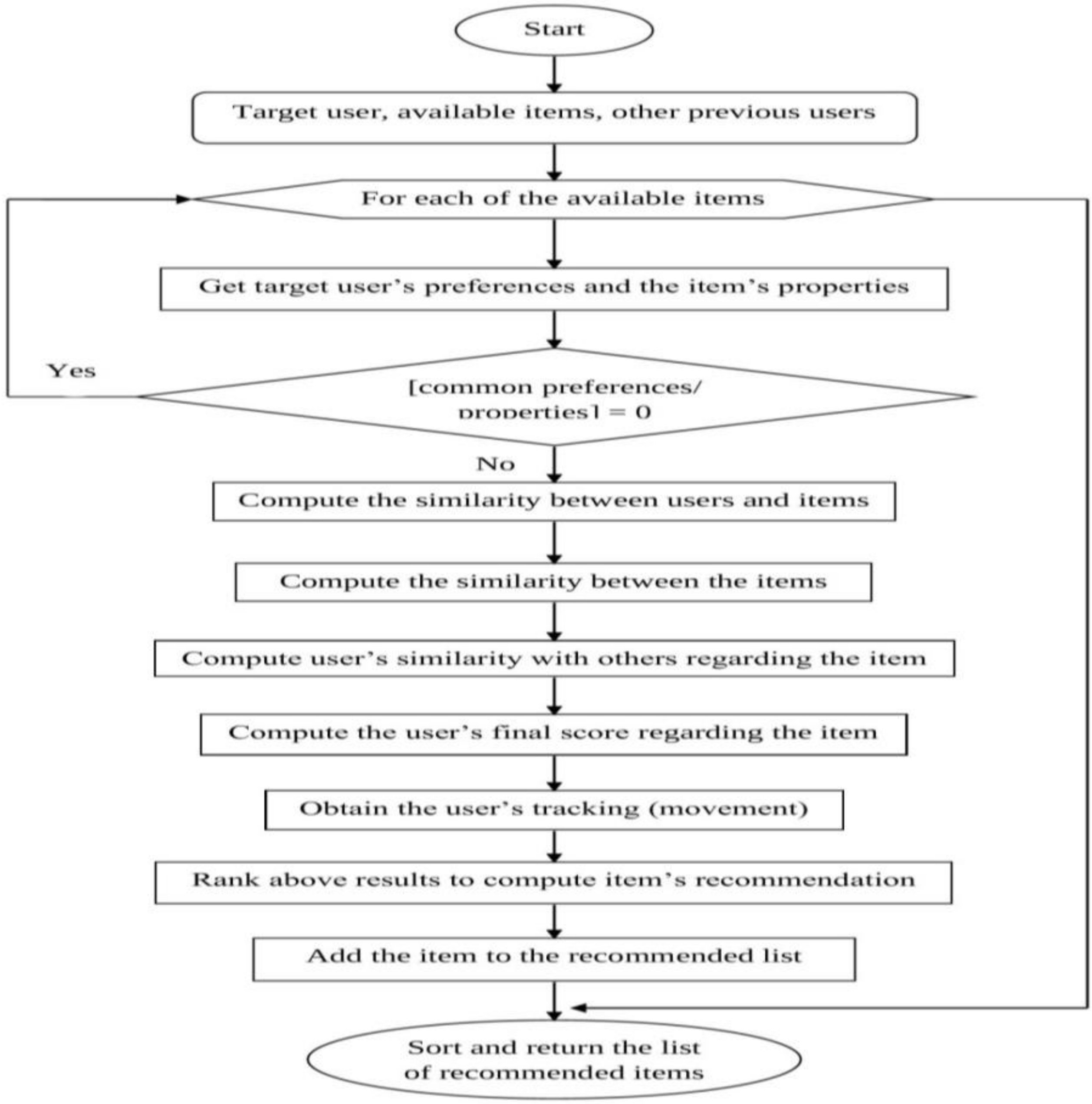

Figure 3.4: State Transition Diagram

3.3.3 Sequence Diagrams

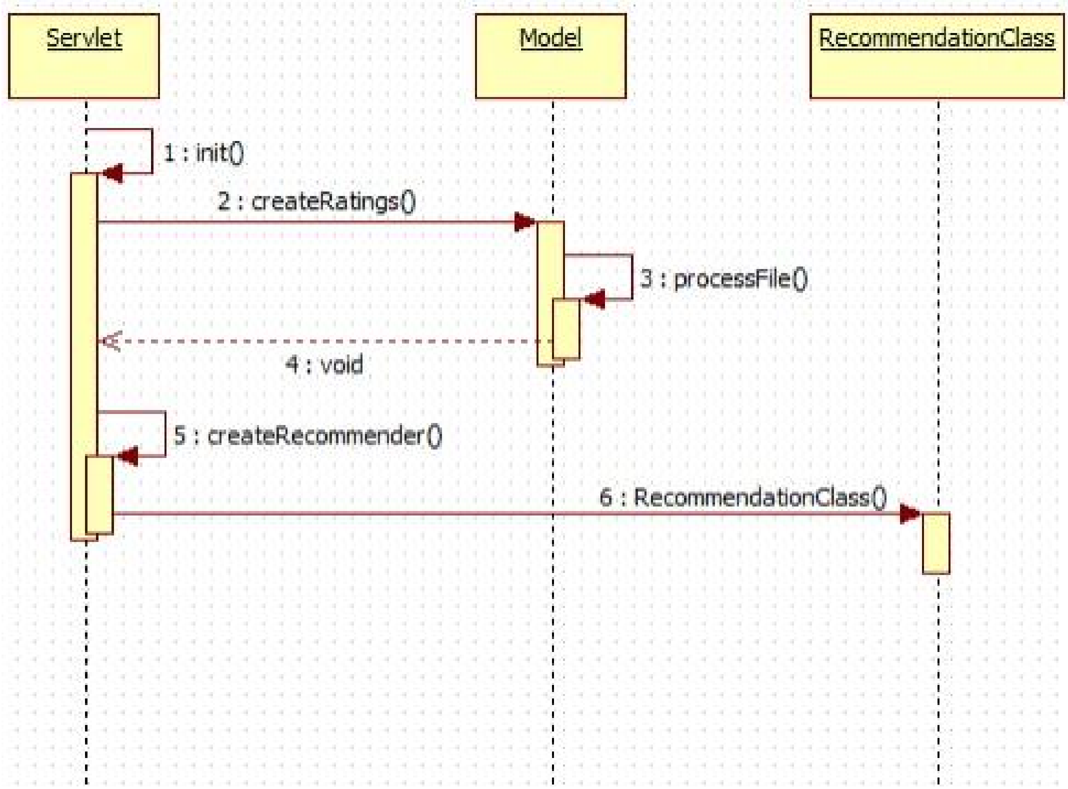

Figure 3.5: Generate Recommendations Sequence Diagram

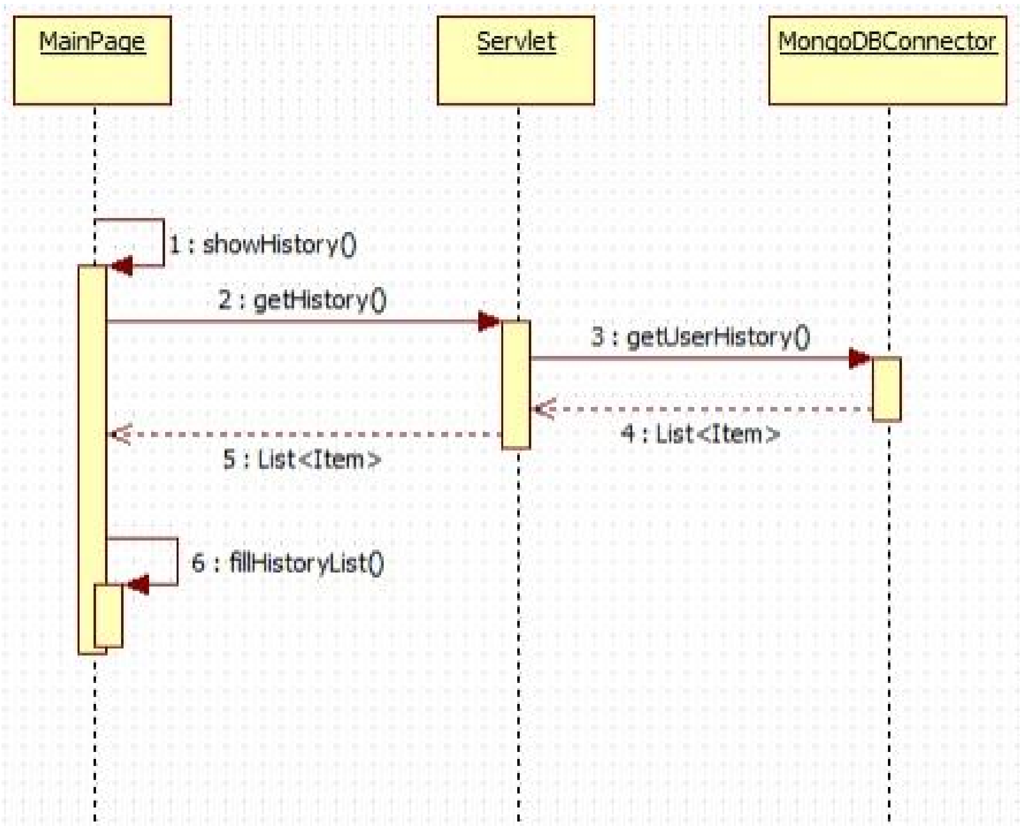

Figure 3.6: View History Sequence Diagram

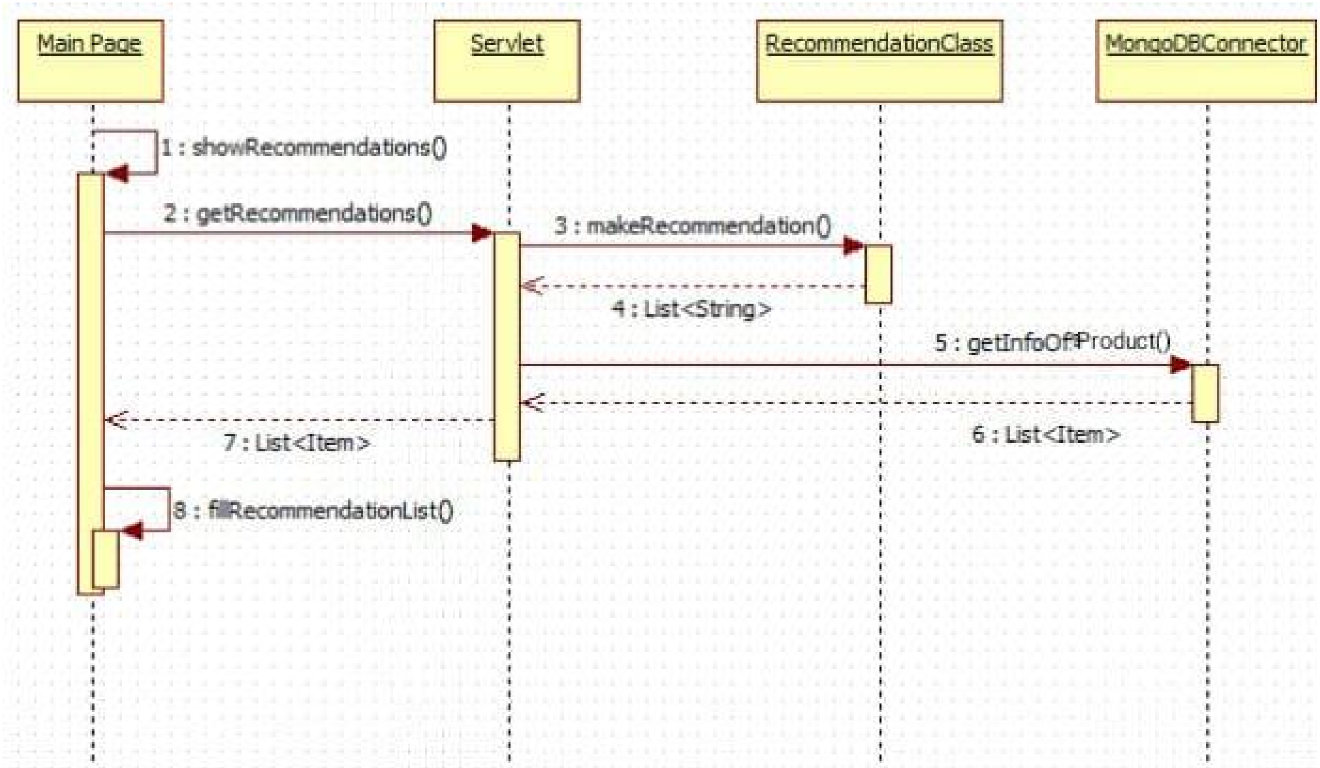

Figure 3.7: View Recommendation Sequence Diagram

3.3.4 Data Flow Diagram

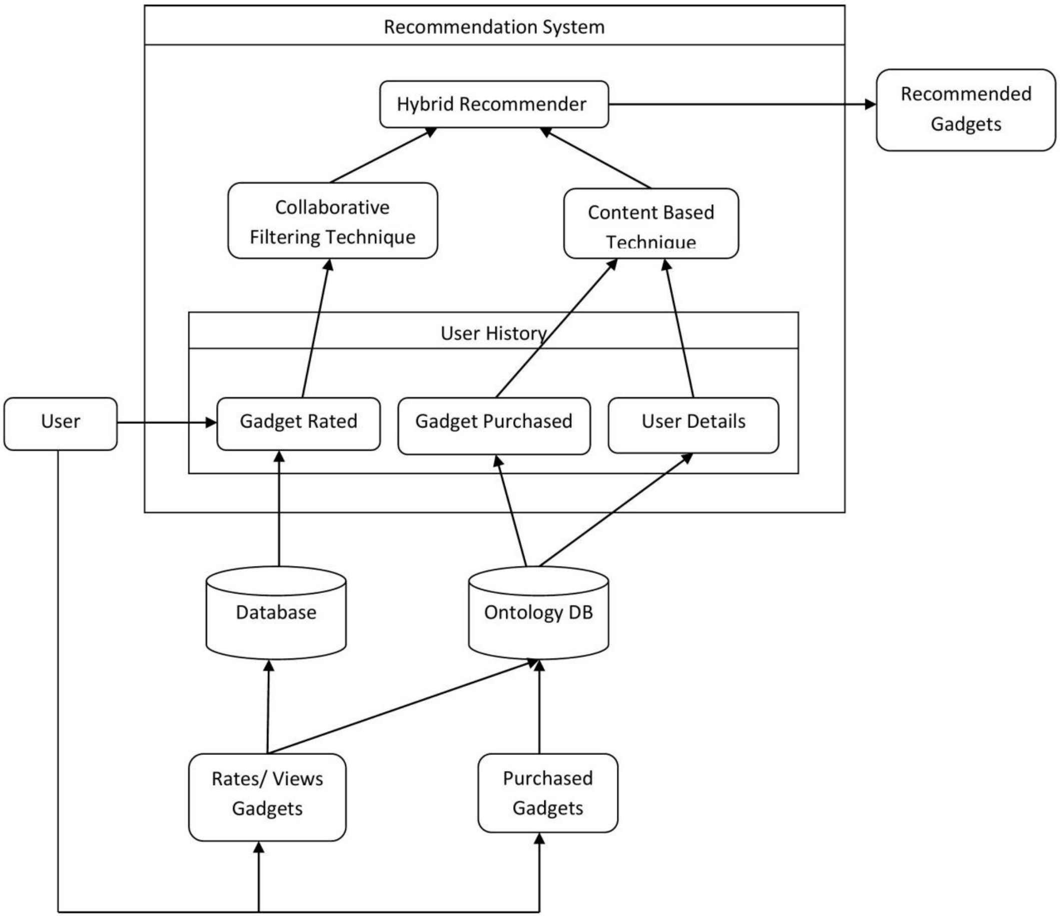

Figure 3.8: Level 2 Data Flow Diagram

3.3.5 Deployment Diagram

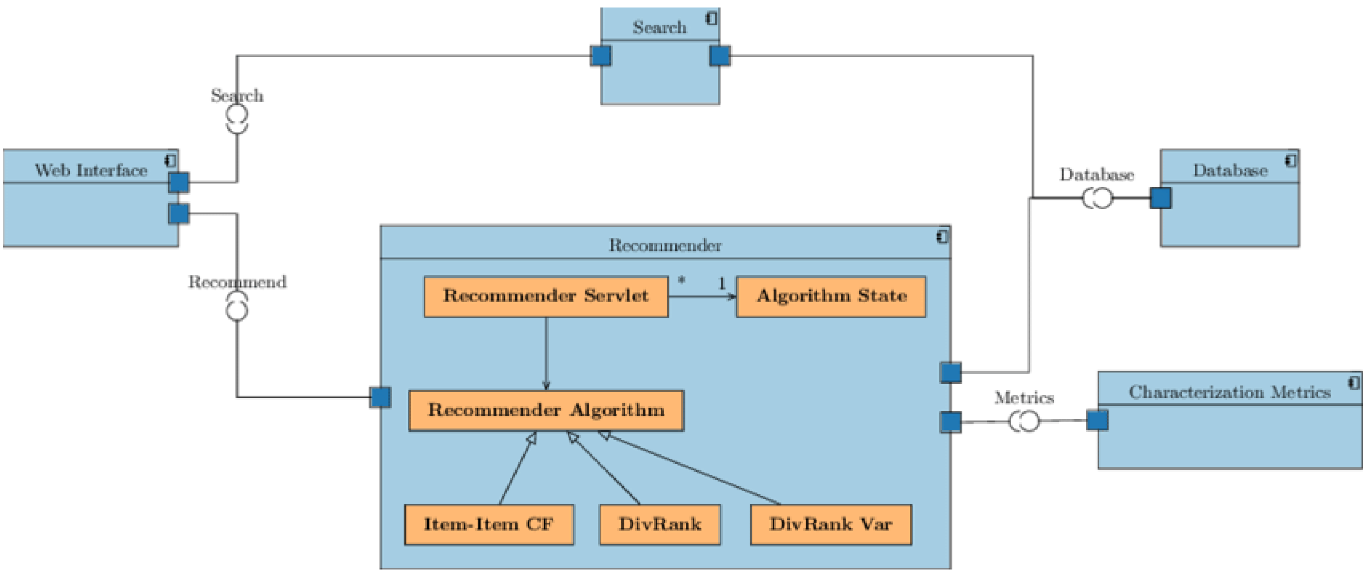

Figure 3.9: Deployment Diagram

3.3.6 Class Diagram

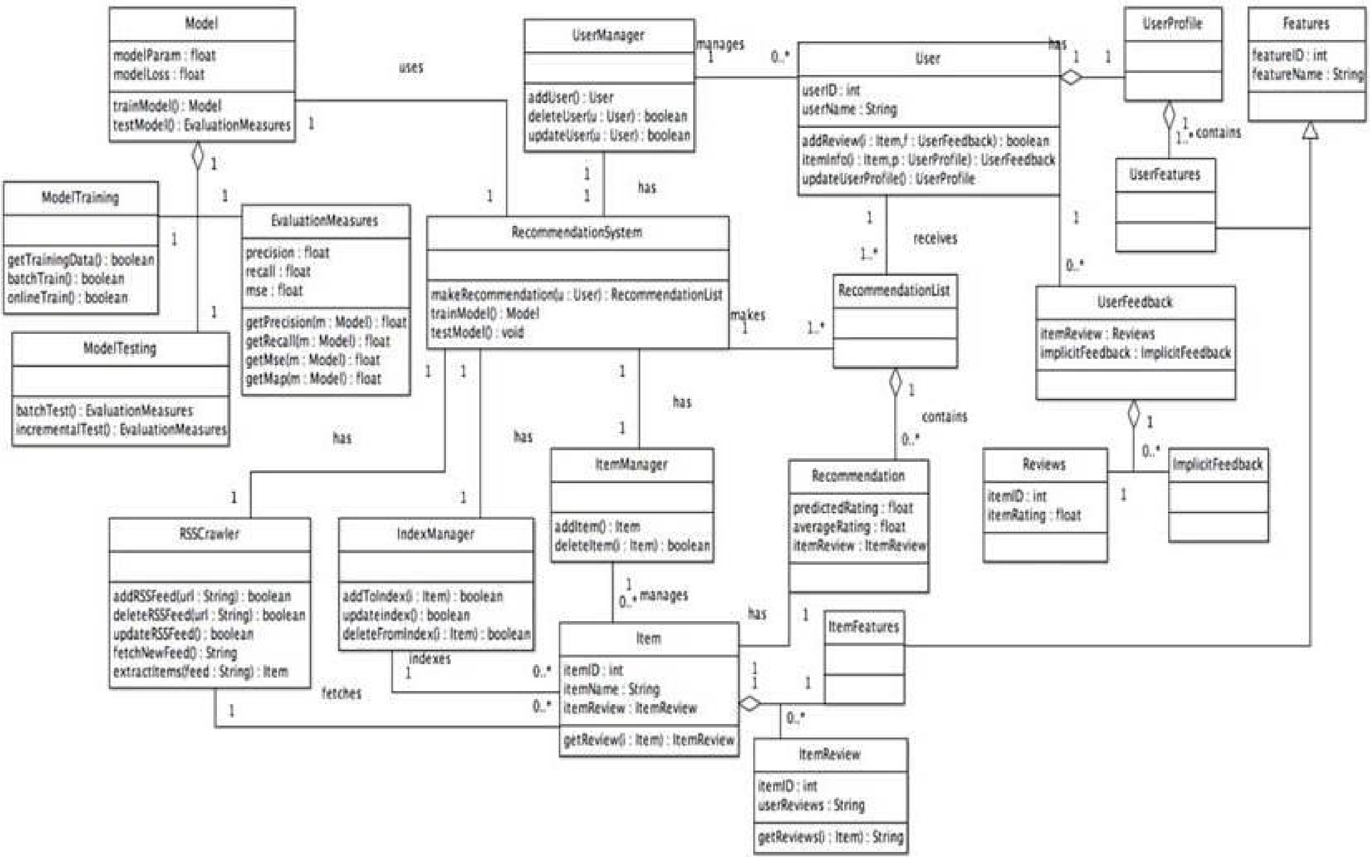

Figure 3.10: Class Diagram

Chapter 4
Background Research

4.1 Recommender Systems

Information workload has become an increasingly important issue these days. No matter what information items a person is interested in knowing more about, there are some recommender systems (RSs) online that recommend relevant items. Recommender systems have shown great potential to help users find interesting and relevant items within a large information space. Recommender systems have developed into an indispensable part of information filtering system these days. Compared to many other fields of information filtering system, RS is a relatively new discipline which emerged in the 1990s. Since that, the interest in the RS area has increased significantly. It can be said that the popularity of RSs is the result of the information needs in this era. There are several points to explain why recommender systems are so popular and powerful.

4.2 Collaborative Filtering Algorithms

According to the taxonomy of recommendation algorithms, there are main two sub-classes of collaborative filtering(CF) algorithms, neighbourhood based and model-based. Two neighbourhood based approaches are user-based CF and item-based CF. Both approaches try to find nearest items or users that are similar to the active user or item. Model-based approaches involves extracting important information from datasets and use the information as a model to make recommendations. One prominent approach in this family is matrix-factorisation. Here the following gives the specific implementations of the three CF algorithms.

4.2.1 Item-Based

There is a bottleneck in user-based CF algorithms. User-based CF algorithms suffer from serious scalability problems. The problem depicts that the computations are expensive in real-world systems where there are potentially millions of users and items. This is how item-based algorithms are introduced to overcome the bottleneck by finding relationships between items rather than users. Item-based CF algorithms are based on the intuition that users are interested in items similar to those previously experienced.

The item-based implementation goes roughly the same process as the user-based. The step of data representation is the same. In similarity computation, the pairwise similarities are calculated based on items instead of users. For example, in the cosine similarity approach, items are represented as vectors in the N-dimentional user space and the similarity between item a and j is calculated as the cosine of the angle between the vector of item a and the vector of item j. As a result, after this step, there will be a data structure produced to store the similarities between items rather than users. In neighbourhood formation, an item neighbourhood is formed by taking a subset of other items with the highest similarity to the active item. In making a prediction, taking the Resnisk's algorithm as an example, the prediction equation is changed to

$$p_{a,j} = \overline{v}_j + \frac{\sum_{i=1}^{k} s_{i,j}(r_{a,i} - \overline{v}_i)}{\sum_{i=1}^{k} |s_{i,j}|}$$

where si,j is the similarity score between item i and j, vj refers to the mean rating of item j across all users, ra,i refers to the rating of user a for item i and k is the size of neighbourhood.

4.2.2 Matrix Factorisation

There are some limitations in user-based and item-based CF algorithms. First, neighbourhood based algorithms heavily depend on the relationships between users and items in order to form predictions or recommendations. This indicates that a limited quantities of data cause it hard to build the relationship model (the similarities between neighbours). This is so called the data sparsity problem . Second, they suffer from the early rater problem. This issues describes that predictions are hard made for items that are newly added to a system as the items are rarely rated in the system (not enough ratings to calculate predictions). It also depicts that predictions or recommendations are hard made for users who newly registered as they do not show enough their taste in the system (not neighbours).

Due to these limitations, the model-based approaches are introduced. There are many well established model-based approaches, among which the matrix factorisation approach is widely-used. It works by making an assumption that there exist some latent features which explain user preferences. Its goal is to uncover these latent features which can afterward be used to predict items for a user more accurately.

It starts with initializing the number of features K and two matrices P ($|U| \times$ K matrix) and Q ($|I| \times$ K matrix), which represents the strength of the association association between a user (P) or an item (Q) and each of the K features. Then the multiplication of P and QT is calculated as the prediction matrix in which the predicted ratings are presented. The following graph gives a nice illustration of how the matrics are constructed.

Chapter 5
Methodology and Team

5.1 Introduction to Big Bang Model

In this model, developers do not follow any specific process. Development begins with the necessary funds and efforts in the form of inputs. And the result may or may not be as per the customer's requirement, because in this model, even the customer requirements are not defined.

This model is ideal for small projects like academic projects or practical projects. One or two developers can work together on this model.

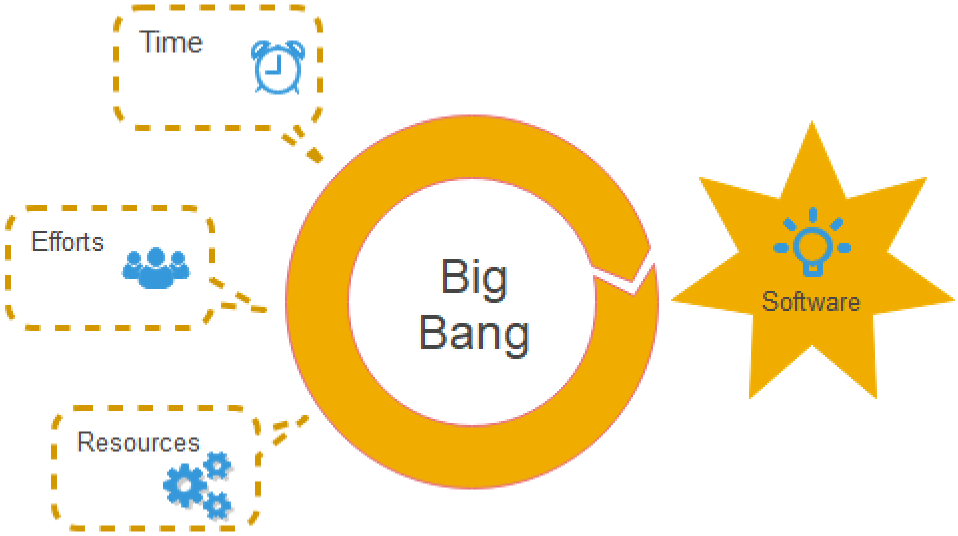

Figure 5.1: Big Bang Model

5.2 Big Bang Model - Design and Application

The Big Bang Model comprises of focusing all the possible resources in the software development and coding, with very little or no planning. The requirements are understood and implemented as they come. Any changes required may or may not need to revamp the complete software.

This model is ideal for small projects with one or two developers working together and is also useful for academic or practice projects. It is an ideal model for the product where requirements are not well understood and the final release date is not given.

Features of Big Bang Model are as follows:

- Not require a well-documented requirement specification
- Provides a quick overview of the prototype
- Needs little effort and the idea of implementation
- Allows merging of newer technologies to see the changes and adaptability

5.3 Big Bang Model - Pros and Cons

The advantage of this Big Bang Model is that it is very simple and requires very little or no planning. Easy to manage and no formal procedure are required.

However, the Big Bang Model is a very high risk model and changes in the requirements or misunderstood requirements may even lead to complete reversal or scraping of the project. It is ideal for repetitive or small projects with minimum risks.

The advantages of the Big Bang Model are as follows:

- This is a very simple model; Little or no planning required
- Easy to manage
- Very few resources required
- Gives flexibility to developers
- It is a good learning aid for new comers or students.

The disadvantages of the Big Bang Model are as follows:

- Very High risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Can turn out to be very expensive if requirements are misunderstood.

5.4 Team Members, Roles & Responsibilities

Mohammed Arshad- Research and Working on UI

Mohit Singhal - Working on Front-End

Mukul Palol - Working on Back-End - Training the model

Chapter 6

Project Screen Shots

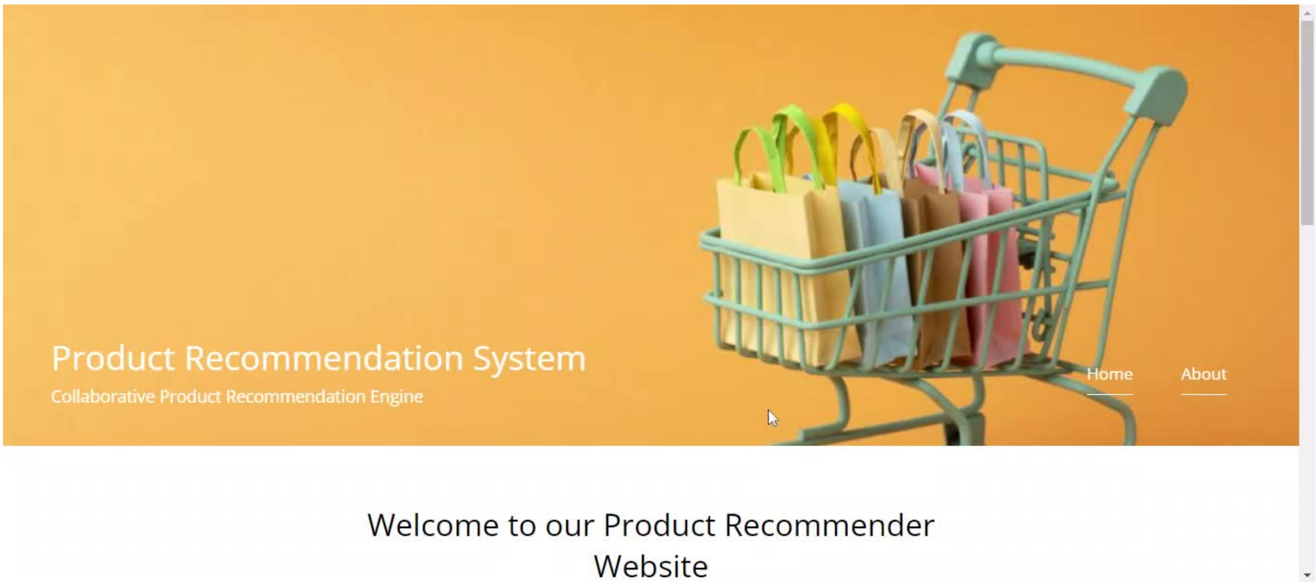

Figure 6.1: Home Page

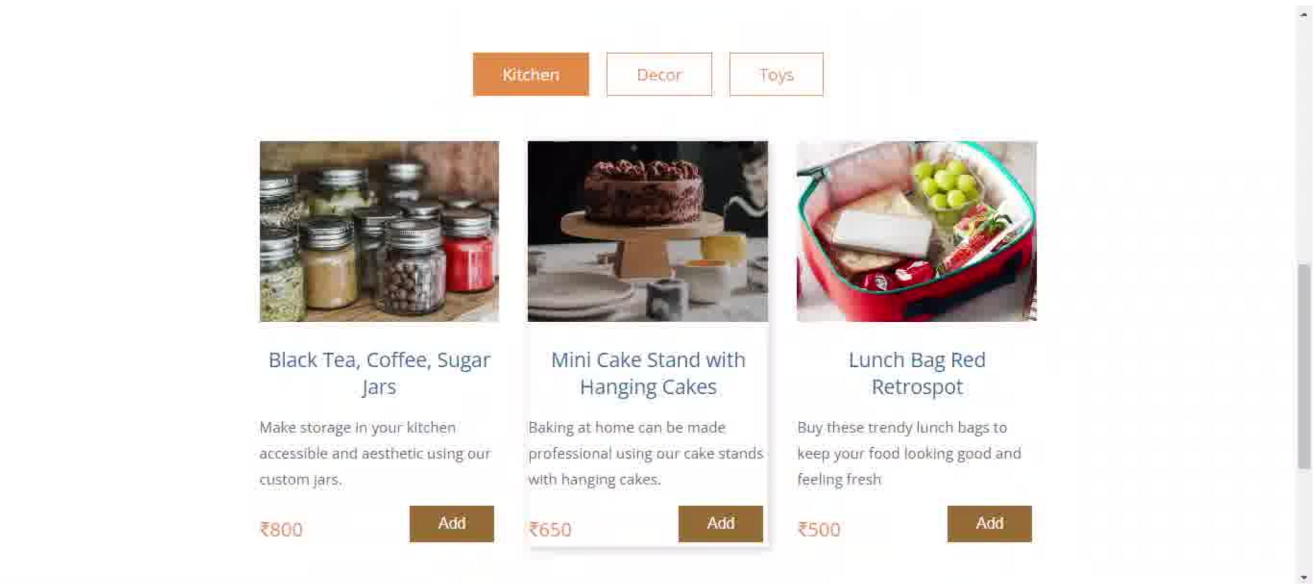

Figure 6.2: Home Page - Kitchen Gallery Page

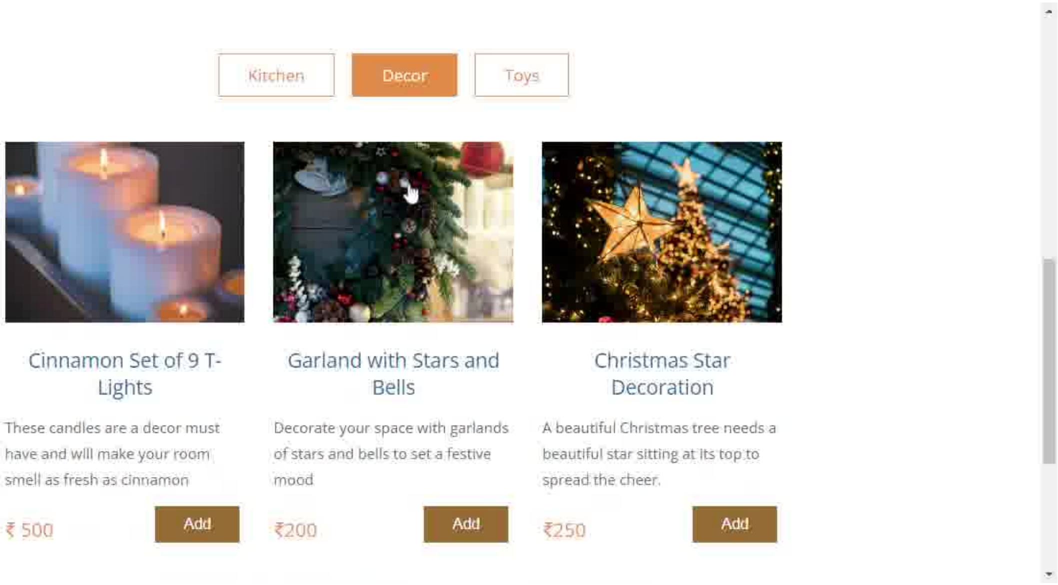

Figure 6.3: Home Page - Decor Gallery Page

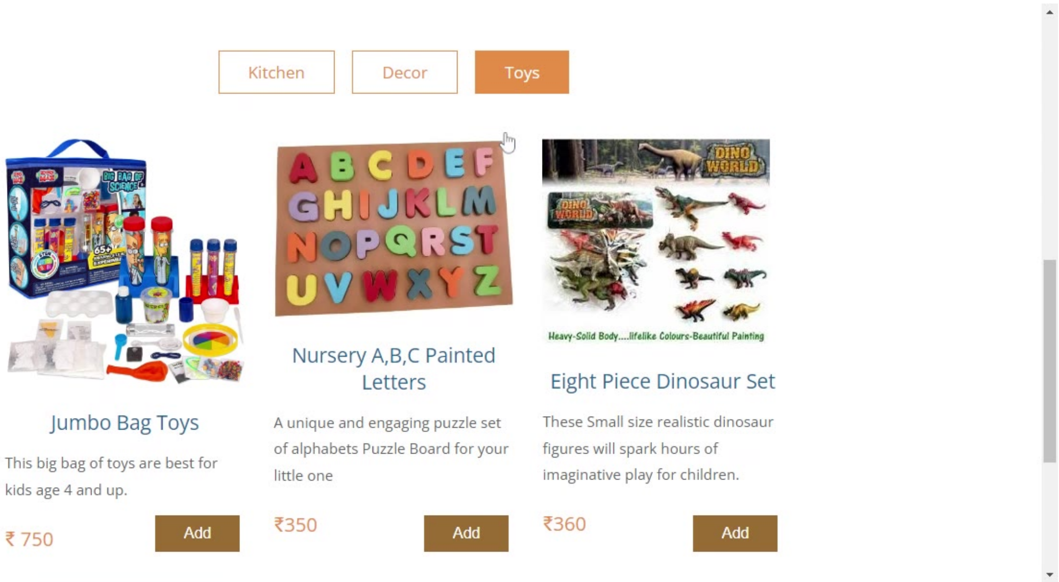

Figure 6.4: Home Page - Toys Gallery Page

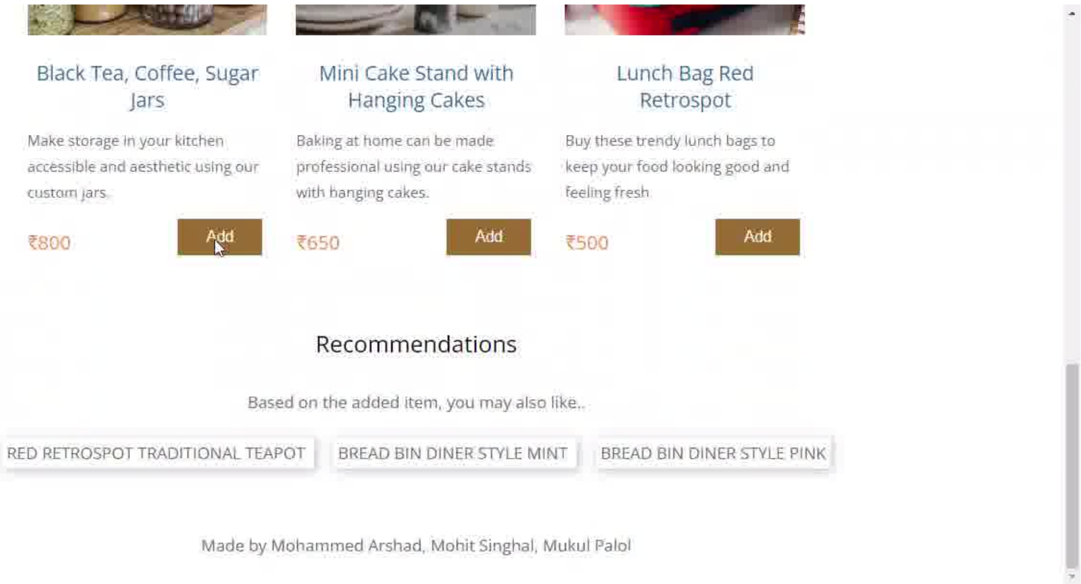

Figure 6.5: Recommendations for Black Tea, Coffee, Sugar Jars

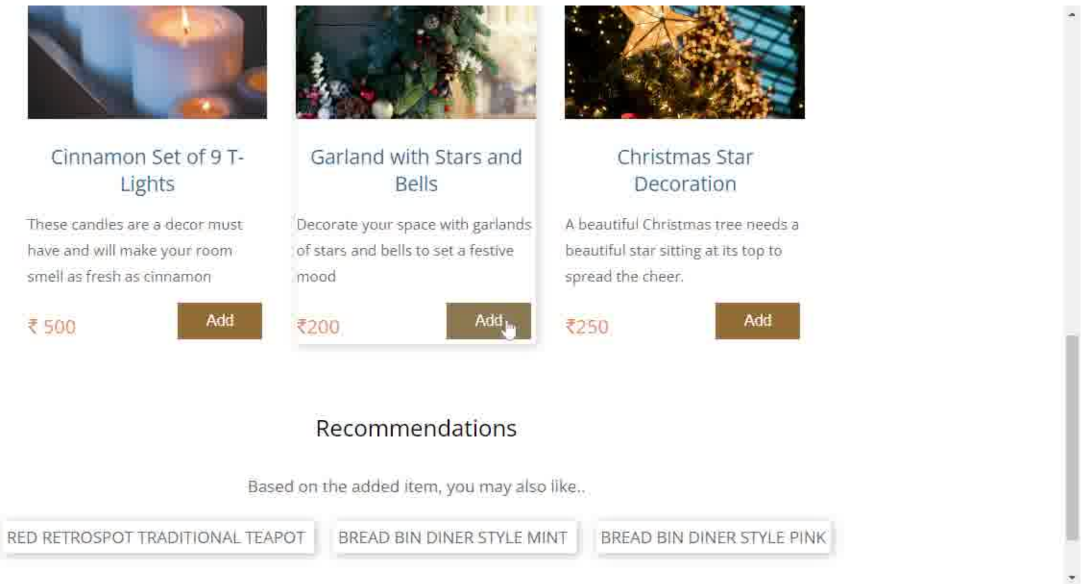

Figure 6.6: Recommendations for Garland with Stars and Bells

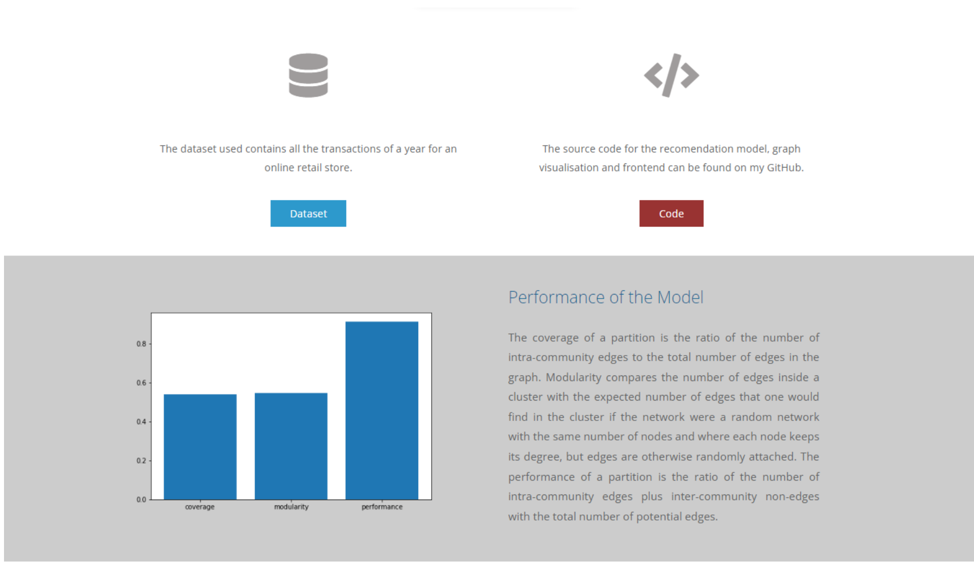

Figure 6.7: About Page

Chapter 7
Conclusion

Machine learning techniques are used to gather and retain data about each device and user in order to establish a link between those users and the gadgets. A user preference technique was used in this work to pick gadgets. Thus, in the realm of personalized recommendation systems, the critical issue of determining what device users require to enhance and satisfy their specific needs from in-depth personalization services has been resolved. The findings indicate that a gadget suggestion system may significantly improve service quality.

Electronic devices have become indispensable in meeting people's fundamental needs. Because of technological advancements, it is now important to meet various functional demands of end-users. As a result, it is critical to recommend devices to clients based on their unique preferences. With the fast advancement of technology, smart gadgets and communication networks have sprung up to cover every part of customer behaviour. These data may also be taught and modelled for future usage in order to deal with potential technological advances.

Chapter 8
Future Scope

- The model can also integrate the content-based filtering approach.
- In the future the model can be updated into a deep learning project using LSTM.
- In the future this model can be migrated to build a webAPI so anyone can deploy the recommendation system on their application.
- Furthermore the website can convert to mobile application and can use camera settings for image based recommendation.
- Expand to support more algorithms.
- Include more evaluation criteria.

References

[1] *Recommender systems handbook (2011). Springer, New York,* Shapira B, Ricci F, Kantor PB, Rokach L

[2] *Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce, 1999, pp 158–166,* Ben Schafer J, Konstan J, Riedl J

[3] *"An Enhanced ContentBased Recommender System for Academic Social Networks", 2014 IEEE Fourth International Conference on Big Data and Cloud Computing.* Vala Ali Rohani, Zarina Mohd Kasirun and Kuru Ratnavelu

[4] *"Introducing Serendipity in a Content-based Recommender System", Eighth International Conference on Hybrid Intelligent Systems.* Leo Iaquinta, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, Michele Filannino and Piero Molino

[5] *"A New Algorithm for Solving Data Sparsity problem Based-On Non Negative Matrix Factorization in Recommender Systems", 2014 4th International Conference on Computer and Knowledge Engineering(ICCKE).* Zeinab Sharifi, Mansoor Rezghi and Mahdi Nasiri

[6] *"WebSell: Intelligent sales assistants for the World Wide Web. In: Proceedings CBR in ECommerce, Vancouver BC; 2001* Cunningham P, Bergmann R, Schmitt S, Traphoner R, Breen S, Smyth B