



ST. MARY'S UNIVERSITY
FACULTY OF INFORMATICS
Department of Computer Science

Senior Project I

Product recommendation for ecommerce

Group members

Bamlak Molla	RCD/3077/2011
Biruk Nigussie	RCD/1912/2011
Sahib Salih	RCD/1937/2011
Siham Ramadan	RCD/1938/2011

Advisor: Shimelis Tamiru

Addis Ababa, Ethiopia
Feb 2022

Authentication of our work

Advisee's Declaration

We the undersigned. Declare that this research paper is our original work. Prepared under the guidance of Mr. Shimelis Tamiru . All sources of materials used for the manuscript have been duly acknowledged.

Name: Bamlak Molla

Name: Biruk Nigussie

Name: Sahib Salih

Name: Siham Ramadan

Date of Submission February 5, 2022

Name and signature of Members of the Examining Board

Name	Title	Signature	Date
_____	Chairperson	_____	_____
<u>Shimelis Tamiru</u>	Advisor	_____	_____
_____	Examiner	_____	_____

Abstract

The vast amount of data available on the Internet has led to the development of recommendation systems. This project proposes the use of soft computing techniques to develop product recommendation systems. It addresses the limitations of current algorithms used to implement recommendation systems, evaluation of experimental results, and conclusion. This report provides a detailed summary of the project “Product Recommendation system for E-commerce” as part of fulfillment of the senior Writing Project, Computer Science Department, St.Mary University’s. The report includes a description of the topic, system architecture, and provides a detailed description of the work done till point. Included in the report are the detailed descriptions of the work done: snapshots of the implementations, various approaches, and tools used so far. The report also includes the project schedule and deliverable.

Acknowledgement

First, we do not own the sufficient gratitude to our God. We offer our sincere appreciation for the learning opportunities provided by St. Mary university. We also like to thank our advisor Mr. Shimelis for continued support and encouragement and also guide us throughout the semester in our senior project.

Table of Contents	Pages
Chapter 1 Project Overview.....	1
1. Introduction.....	1
1.1 Background of the study.....	2
1.2 Statement of the problem.....	4
1.3 Objectives of the Project.....	5
1.3.1 General objective.....	5
1.3.2 Specific objectives.....	5
1.4 Significance of the project.....	5
1.5 Feasibility study.....	7
1.5.1 Economic Feasibility.....	7
1.5.2 Technical feasibility.....	7
1.5.3 Operational feasibility.....	7
1.6 Beneficiaries of the Project.....	8
1.7 Methodology.....	9
1.7.1 Data Collection.....	9
1.7.2 System Development Methodologies.....	9
1.8 Development environment/ programming tools and other tools.....	10
1.9 Scope of the project.....	12
1.10 Limitations of the project.....	13
1.11 Risks, assumptions and constraints.....	13
1.11.1 Risks.....	13
1.11.2 Assumptions.....	14
1.11.3 Constraints.....	14
1.12 Work Breakdown Structure.....	15
Chapter 2 Requirement Analysis and Specification.....	17
2.1 Current system.....	18
2.2 Business rules.....	19
2.3 Proposed system.....	20
2.3.2 Functional requirements.....	21
2.3.3 Nonfunctional requirements.....	21
2.3.4 System models.....	22
Chapter 3 System design.....	41
3.1 Introduction.....	41
3.2 Purpose of the system.....	41
3.3 Design goals.....	42
3.4 Current software architecture.....	42
3.5 Proposed software architecture.....	43
3.5.1 Sub-system decomposition.....	44
3.5.2 Component diagram.....	45
3.5.3 Deployment diagram.....	46
3.5.4 Persistent data management.....	46
3.5.5 Access control and security.....	48
3.5.6 Global software control.....	48
3.5.7 Boundary conditions.....	49
Reference.....	50

List of Figures

Figure 1 . Collaborative filtering and Content-Based filtering	3
Figure 2 . Example of order.....	6
Figure 3 . Example of Co-occurrence matrix of products.....	6
Figure 4 . Agile for a small update and get there fast also if small bugs can lead to series problems.	10
Figure 5 . Work Break Down structure.....	15
Figure 6 . A data-driven approach to building a recommender system.....	20
Figure 7 . Use-case diagram.....	23
Figure 8 . Sequence diagram for Login.....	28
Figure 9 . Sequence diagram for manage products.....	28
Figure 10 . Sequence diagram for search products.....	29
Figure 11 . Sequence diagram for buy products.....	29
Figure 12 . Sequence diagram for add to cart.....	30
Figure 13 . Sequence diagram for review.....	30
Figure 14 . Sequence diagram for feedback form.....	31
Figure 15 . State chart diagram.....	32
Figure 16 . Activity diagram.....	33
Figure 17 . Customer buys Product.....	35
Figure 18 . Customer gives Feedback.....	35
Figure 19 . Customer gives Review.....	35
Figure 20 . Customer add to Cart.....	35
Figure 21 . User receives Recommendation list.....	35
Figure 22 . Data model contains Recommendation list.....	35
Figure 23 . Recommendation system makes Recommendation list.....	35
Figure 24 . Review stored in Data model.....	36
Figure 25 . Feedback stored in Datamodel.....	36
Figure 26 . Cart stored in Datamodel.....	36
Figure 27 . Class diagram.....	37
Figure 28 . Landing Page.....	38
Figure 29 . Login Page.....	38
Figure 30 . Sign up Page.....	39
Figure 31 . Product Page(home page).....	39
Figure 32 . Contact Us Page.....	40
Figure 33 . About Us Page.....	40
Figure 34 . Illustrates the architecture of the Netflix recommendation system.....	43
Figure 35 . System architecture on recommender system.....	44
Figure 36 . Component diagram.....	45
Figure 37 . deployment diagram.....	46
Figure 38 . ER diagram.....	47

List of table

Table 1 . Gantt chart.....	16
Table 2 . Detail description of use case for Login.....	24
Table 3 . Detail description of use case for Manage product.....	25
Table 4 . Detail description of use case for Search product.....	25
Table 5 . Detail description of use case for Add to cart.....	26
Table 6 . Detail description of use case for Buy product.....	26
Table 7 . Detail description of use case for Review.....	27
Table 8 . Detail description of use case for Fill feed back.....	27
Table 9 . Access modifiers.....	36
Table 10 . User-system interaction matrix.....	48

Chapter 1 Project Overview

1. Introduction

A lot of things happened in this notable era that shaped the world where we are today. Look around you and see how everything is within your reach. Everything is just a click away! Everything is online and the possibilities are endless. One of the many exciting things we can do now is ecommerce.

So what exactly is ecommerce?

Ecommerce (electronic commerce) is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network, primarily the internet.

Any site where you can obtain items for sale over the internet is considered an ecommerce website.

Ebay and Amazon, the grandfathers of ecommerce, paved the way to a newer and more convenient way of shopping. They are actually not that old to be called grandfathers, but the higher level of respect is implied there. Of course online buying and selling were already a thing before, but because of these companies, e-commerce website became an easier and more preferred platform to a lot of consumers.

People were very skeptical to use these sites to purchase their items there because they are prone to fraudulent activities and scamming schemes. It is still evident up to this day, but developers come up with solutions to better people's shopping experience. Examples of these solutions are reviews, return policy, and cash on delivery. These features help consumers decide if they should purchase items from an e-commerce website or not.

The main difference between ecommerce and traditional commerce:-

Ecommerce is a form on the online shopping market where customers can buy goods and services easily from their **home** including a few discounts and the goods will be delivered at the doorstep.

Traditional Commerce is the process of exchanging goods and services in the form of money directly. Traditional Commerce involves **face to face** and in person dealing with all the parties to perform the exchange of goods and services with predefined prices. One of the obvious differences between an ecommerce website and an ordinary business or company website is in the features that it supports. A company website may just house information on the brand's product and services so users will need to

contact the company directly if they want to do business with them. Meanwhile, an ecommerce website works pretty much like how a physical store works. Users can purchase items, arrange for delivery and payments on the same site without the need to call a human person to help them with their order.

There are a lot of ecommerce websites selling a variety of items now. From clothing pieces to food items to gardening materials, one must be creative to make his or her brand more unique than the other. Even social media accounts are becoming ecommerce platforms and are developed to be ecommerce websites nowadays.

The only difference is that on an e-commerce website, you don't have to deal with a human to be able to buy and get your items delivered to your doorstep. An ecommerce website must be as functional as it is with or without the owners or managers monitoring the site and dealing with their customers every second of the day.

Through out this process the user needs help to get personalized recommendations, to take correct decisions in their online transactions, increase sales and redefine the web browsing experience, retain the customers, enhance their shopping experience. So here we need the recommendation system.

1.1 Background of the study

This project will works on product recommendation, which is basically a filtering system that seeks to predict and show the items that a user would like to purchase. Personalized product recommendations are an integral part of any successful retailer's e-commerce strategy, helping customers discover products that are most relevant to their interests and reduce barriers to purchase.

It is important to segment new and returning visitors. Since you don't have information on new users, you should use generic product recommendations.

Once the customer has started browsing the site, you can start gathering more information about them. Then you can start to match the type of generic recommendation to each customer. For example, if the customer has sorted the items on the category page from low to high, the customer is likely to be price-sensitive and so they should be presented with recommendations of sales and offers.

Once the customer has started viewing some items, you now have information about their preferences and needs. For such shoppers, as well as returning visitors (which

you have past information of), you should present with personalized recommendations.

There are 2 types of filtering or the combination of both:

1. Collaborative filtering

Creates recommendations by comparing the current user to other users that have similar preferences. Collaborative filters are expected to increase diversity because they help us discover new products. Some algorithms, however, may unintentionally do the opposite. Because collaborative filters recommend products based on past sales or ratings, they cannot usually recommend products with limited historical data.

2. Content-based filtering

Makes recommendations based only on the specific user's data, such as their past purchases or searches. Content-based filtering has many benefits compared to collaborative filtering, including: No data from other users is required to start making recommendations. Unlike collaborative filtering, content-based filtering doesn't need data from other users to create recommendations.

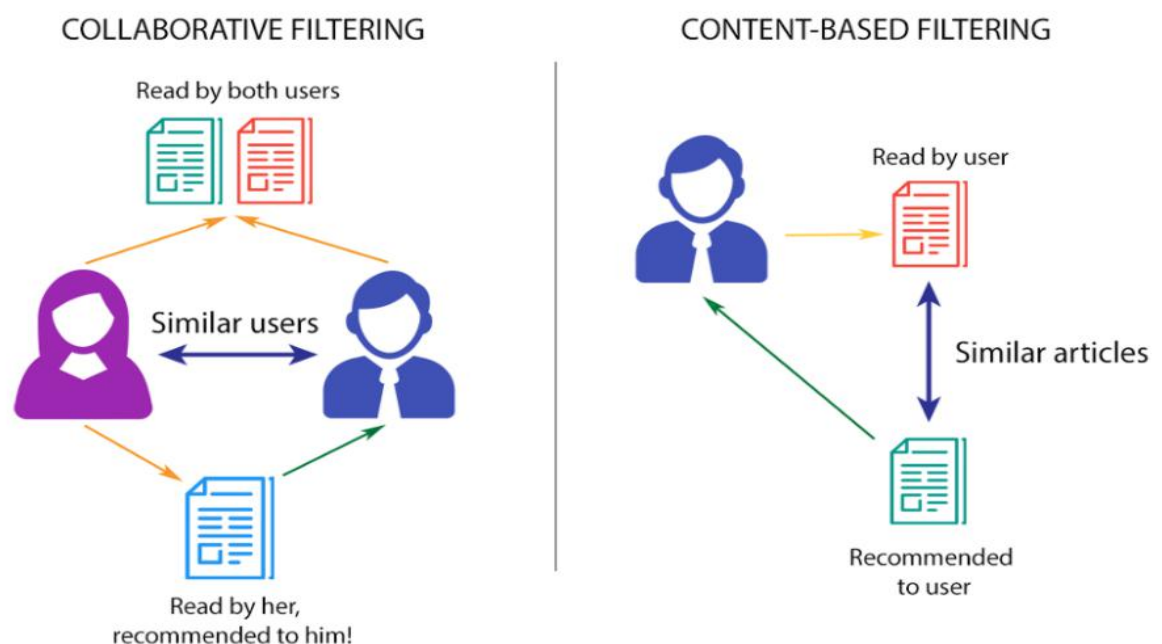


Figure 1. Collaborative filtering and Content-Based filtering

The third one is called hybrid. A hybrid recommendation engine looks at both the meta (collaborative) data and the transactional (content-based) data. Because of this, it outperforms both.

1.2 Statement of the problem

E-commerce websites nowadays offer a multitude of items. This gives customers a lot of choice but can also hinder their search. Due to time constraints and the accepted level of effort exerted on a search, customers **may not be able to go through all of the items** on the site. As a result, some customers cannot find the exact item they were looking for or an item they truly desire. Consequently, some customers may purchase an item with a high level of cognitive dissonance or not make a purchase at all.

This is where presenting customers with personalized product recommendations can help. These recommendations allow customers to find the products they need and are relevant to them quickly and easily. The items generated are personalized according to the individual's browse and purchase behavior and product information (such as product inventory). They are created in real-time in order to be the most accurate and relevant as possible.

This optimizes the products discovery process and eliminates the issue of long and effortful searches. The ease of use and being given more (and relevant) choice enhance the user experience and as a result enhance the customer engagement and brand loyalty. On top of that, personalized product recommendations tend to trigger impulse purchases as well as induce a higher average order value.

Accordingly, personalized product recommendations induce more conversions, as customers who click on the product recommendations have a 5.5 times higher conversion rate than those who do not. Overall, the personalized product recommendations account for 12% of total revenues.

The products can be recommended based on the top overall sellers on a site, mainly based on Content based and collaborative based machine learning algorithms. This works by collecting data from e-commerce site.

Customers will prefer finding products easily and quickly, and the easier it is for them to find things, the faster their purchase will be. Our recommender system allow them to shortlist products based on their parameters.

The Recommend-er systems have become needed in consumers' daily lives on the online platform, ranging from e-commerce, and social media to news outlets. Nowadays, Recommend-er system is one of the major factors for the economic growth. So this system plays an important role for the development of one country. Our preferences and biases are super-charged by machine learning algorithms that learn from our taste, and recommend more of what we desire to see.

Product Recommendations are essential to meeting the aggressive performance goals of online retailers and are a proven method for driving profitability. The product recommendation system can drive conversions by suggesting other products customers have bought with the item the customer is browsing, products that are in line with the customer's search queries, or products that pair well with an item already in a customer's cart. Product recommendations are especially useful for businesses that have large product catalogs, since this provides them with more options to connect with and guide customers.

1.3 Objectives of the Project

1.3.1 General objective

The main objective of this project is to build a product recommendation system for e-commerce.

1.3.2 Specific objectives

To achieve the general objectives, the specific objective of our work are:

- ✓ To understand the concept of recommender system and the designing process.
- ✓ To create customized products and services.
- ✓ To customize services around standardized products and services.
- ✓ To provide point of deliver customization.
- ✓ To provide quick response throughout the value chain.
- ✓ To attract new customers.
- ✓ To categorize the products.

1.4 Significance of the project

Recommender system will become an increasingly important topic as we demand more personalized content be placed in our daily feeds. Nowadays, we are all familiar with the recommended videos on YouTube.

Developing a product recommendation system transforms the ecommerce with AI powered recommendation system in to self-learning technology with world-category artificial intelligence that spans one another edge.

The recommender system plays vital role in many industries ranging from retail, E-commerce, and entertainment to food delivery, etc. It heavily uplifts the user experience on any platform. Imagine that scrolling the marketplace feed repeatedly, and satisfied with all recommended stuff in our hands even though we may not want it.

The main goal of this project is to develop a dynamic pricing system to increase ecommerce profits by adapting to supply and demand levels. This recommender system is that if a person likes a particular item and will also like an item that is like it and to recommend that it will make use of the user's past item metadata.

Co-occurrence Matrix

Concepts of why we can use it to create a recommendation system for similar products. The information we tend to be utilizing is just client order data and specifically we have an interest in orders wherever products are purchased with alternative products.



Figure 2. Example of order

We have 3 different items, two different brands that are both independently purchased with socks. Using this data, we are going to build a bridge from the NIKE to the FILA via the socks.

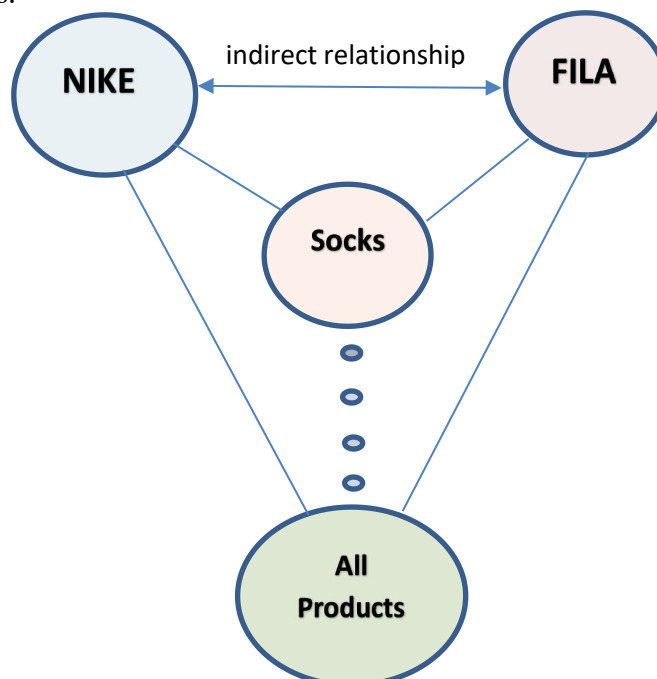


Figure 3. Example of Co-occurrence matrix of products.

So, we are going to have thousands of orders. The customer expected that both brands of shoes occur with things like colors, sportswear, etc. We can utilize these co-occurrences of products to build a multi-dimensional vector representation of all the products in the catalogue.

Data

As mentioned, our data is customer order data. For more datasets the customer probably wants to join an order level table so the customer can see all items that occurred within a specific order id.

1.5 Feasibility study

A feasibility study is part of the initial design stage of any project/plan. It is conducted in order to objectively uncover the strengths and weaknesses of a proposed project or an existing business.

1.5.1 Economic Feasibility

The economic feasibility is to identify the functional benefits and costs associated with the development project.

Beneficiaries

- ✓ This project will benefit the country by improve Cash flow according to online payments.
- ✓ It introduce global market.
- ✓ Also helps customers to reduce cost.

1.5.2 Technical feasibility

The technical feasibility is technical resources of the project and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

- ✓ A brief description of the business to assess more possible factors which could affect the study.
- ✓ The human and economic factor.
- ✓ The possible solutions to the problem.

1.5.3 Operational feasibility

The operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified .

- ✓ The system is easy to use.

- ✓ It solves a problem of show unwanted items by customers.
- ✓ The system will succeeded.

1.6 Beneficiaries of the Project

There are 4 main beneficiary of this project: these are:

1. Users

Customers and potential buyers can receive a lot of information that makes their shopping experience faster and easier, as well as better customer service. They will also purchase products and services 24/7 in and convenient and convenient way.

Recommendation systems are a very popular and effective paradigm in retail business. With recommendation system shoppers can find items they like with less effort.

Customer behavior analysis focuses on understanding the type of customers, what they like, what they do not like, what their pattern of interaction with the items is, customer value, etc. If we manage to model these aspects of a customer, we can anticipate their future needs.

The main benefits coming from customer behavior analysis system are:

- 1. Boost in sales,**
- 2. Better understanding of customers and**
- 3. Long tail strategies.**

2. Companies of organizations

Companies will love this recommender systems to help users discover new and relevant items (products, videos, jobs, Music), creating a delightful user experience while driving incremental revenue.

3. Governments-Trade facilities

Recommender system refers to the use of the internet and other information and communication technologies to support governments in providing improved information and services to citizens unable to make effective choices from the range of information to which they are exposed.

Researchers and managers recognize that recommender systems offer great opportunities and challenges for business, government, education, and other domains, with more recent successful developments of recommender systems for real-world applications becoming apparent.

4. The project team

- ✓ This will commonly help us to know techniques include collaboration.
- ✓ To develop the know-how of certain products in computer science.

- ✓ To understand the real world practical challenge.
- ✓ To fulfill requirement for BSC's.
- ✓ To make an applicable systems development into practice.

1.7 Methodology

1.7.1 Data Collection

We use the following methods to collect relevant data required for our project is basically focus on data-set.

Data-set is a collection of data pieces that can be treated by a computer as a single unit for analytic and prediction purposes. This means that the data collected should be made uniform and understandable for a machine that doesn't see data the same way as humans do. We need Data-set because our project is machine learning also there are other data collection methods needed:

- ✓ **Observation:** we use observation and gather data about our project inputs, outputs, and outcomes. With regard to monitoring and evaluation, observation generally refers to when we observe how our project works and activities done early. Observation allows us to see what is happening in the project.
- ✓ **Users:** enable users to create data reports using SQL queries. Simply using collected data on the database.

1.7.2 System Development Methodologies

Most modern development processes can be vaguely described as agile. Other methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, and extreme programming.

We are going to use Agile development process.

What is Agile ?

Its practices include requirements discovery and solutions improvement through the collaborative effort of self-organizing and cross-functional teams with their end user, adaptive planning, evolutionary development, early delivery, continual improvement, and flexible responses to changes in requirements, capacity, and understanding of the problems to be solved.

It has values like

- ✓ Individuals and interactions over processes and tools,
- ✓ Working software over comprehensive documentation,
- ✓ Customer collaboration over contract negotiation,

- ✓ Responding to change for the next plan.

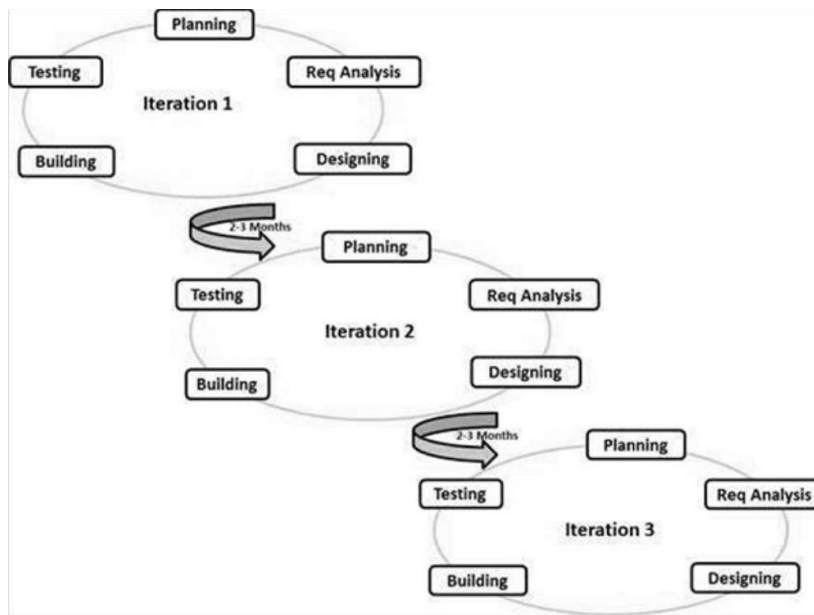


Figure 4. Agile for a small update and get there fast also if small bugs can lead to series problems.

1.8 Development environment/ programming tools and other tools

● For recommendation system: -

We prefer **python**. There are many reasons, python provided the following advantages:

- ✓ Encourages better understanding of user requirements.
- ✓ Permits to accessibility and portability of application, clearly defined and more manageable parts.
- ✓ It has a great library ecosystem like Pandas, Numpy and so on .
- ✓ It's easy to understand and allows for quick data validation.
- ✓ It's flexibility.

Python (Machine Learning):-is an interpreter, object-oriented, high-level programming language with dynamic semantics. Python supports modules and packages, which encourages program modularity and code reuse. We chose python because of its benefits that make it particularly suitable for machine learning and deep learning projects.

- **For E-commerce:** we will use these helpful languages

React(Js): is a JavaScript framework that helps to build a front-end fast and multipurpose web applications and it's suitable to work with Django.

Under React js we will be using these languages:

- **Html** is the code that is used to structure a web page and its content.
- **CSS** is the language we use to style an HTML document to describe how HTML elements should be displayed helps us to keep the informational content of a document separate from the details of how to display it. The details of how to display the document are known as its style.
- **JavaScript** is a text-based client-side programming language which helps web developer to do Web Application Development and make dynamic and interactive web pages by implementing custom client-side scripts. Allow to build large-scale web application easily.

For Back-end we will be using:

Django(python) is a Python-based free and open-source web framework that is suitable to work with machine learning projects.

MYSQL is an easy choice of Python developers. It's Relational Database Management System (RDBMS).

uses SQL query: database helps to automate data retrieving.

Rest API: is helps to communicate the front end(react) and back end(Django).

Development Tools

- **Adobe XD** is a vector-based user experience design tool for web apps and mobile apps, developed and published by Adobe Inc. It is available for mac and Windows, although there are versions for iOS and Android to help preview the result of work directly on mobile devices.
- **Visual Studio Code** - is a cross platform editor made for Windows, Linux and mac. It's easy to use, lightweight and includes git for version control.
- **Jupyter Notebook** is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and

transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

- **Git** is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- **EdrawMax** is 2D business technical diagramming software which help create flowchart, organizational chart, mind map, network diagrams, floor plans, workflow diagrams, business charts and engineering diagrams.

GitHub is a cloud-based hosting service that lets you manage Git repositories. If you have open-source projects that use Git, then GitHub is designed to help you better manage them.

1.9 Scope of the project

In this section we discuss the scope of our project. Product recommendation system based on content-based and collaborative-based recommendation model is for e-commerce website that is able to perform the product recommendation for customers. The scope is all listed as follow

- Develop an e-commerce website.
- Find a data-set to train a model.
- Collect and organize information on users and products.
- Compare one User to all other users.
- Create a function that finds products that one User has not used, but which similar users have.
- Rank and recommend.
- Evaluate and test the model.

1.10 Limitations of the project

The limitation of the project includes

- Main limitation is we expected to do the e-commerce website to take a data for recommendation system.
- The data set we will give to the system may be very small so we could need additional data for the system.
- Available time to develop the system and the time required to develop a full broad-spectrum application is not equal. So, we are trying to develop the application with the time constraint given by our advisor and the department.
- Training the model is hard because it need large amount of data and it may not be accurate if the data is less.
- The system will be limited based on the given data-set.

1.11 Risks, assumptions and constraints

1.11.1 Risks

The main risks that we are going to face are: -

- ✓ Difficulties for us to build an application for website because of our lack of knowledge and coding experience for a web. We will certainly make mistakes and follow the philosophy of learning by doing.
- ✓ The lack of information about data of the recommended products.
- ✓ The team member may not present because of different problems
- ✓ Scarcity of resources or required materials
- ✓ Damage on the computers that we work on.
- ✓ The University might not allow us to use other programming languages that will make the development phase easier.
- ✓ Attacked by Malwares or Viruses

We also have alternative solutions for the above risks. These are: -

- ✓ We will address the issue by devoting more time on study of website programming.
- ✓ Try to give understanding of the use of the new system.
- ✓ Other team members will cover the tasks.
- ✓ Use the available resources in an efficient manner.
- ✓ We will have a backup on different computers.
- ✓ We will use the provided language for developing mobile apps.

- ✓ We will use Antivirus for Window PC and will put back up on Linux PC.

1.11.2 Assumptions

In general, we have the following assumptions:

- ✓ We assume that we will perform our task effectively and present it.
- ✓ We assume that our project will be the best project among other projects.
- ✓ We assume that the user must have an internet connection with speed of at least 100Kb/s

1.11.3 Constraints

When measurements have been defined, the constraint that limits the system to reach its goal can be found. Theory of Constraints is a methodology that is used to identify the most important constraints in a system and then improving the constraint until it is no longer the limiting one. While working on this project, we might face the following constraints:

- ✓ The time we have to finish this project might be not enough by some reasons like political issues or with unexpected disease(covid-19).
- ✓ The team members might be absent or being late.
- ✓ We might not get all the resources we need to accomplish the project on time.
- ✓ The software tools we are going to use might be expired before we complete the project.

1.12 Work Breakdown Structure

Work Breakdown structure (WBS) is a hierarchical and incremental decomposition of the project into phases, deliverable and work packages. It is a tree structure, which shows a subdivision of effort required to achieve an objective. Here is the WBS we create for this project.

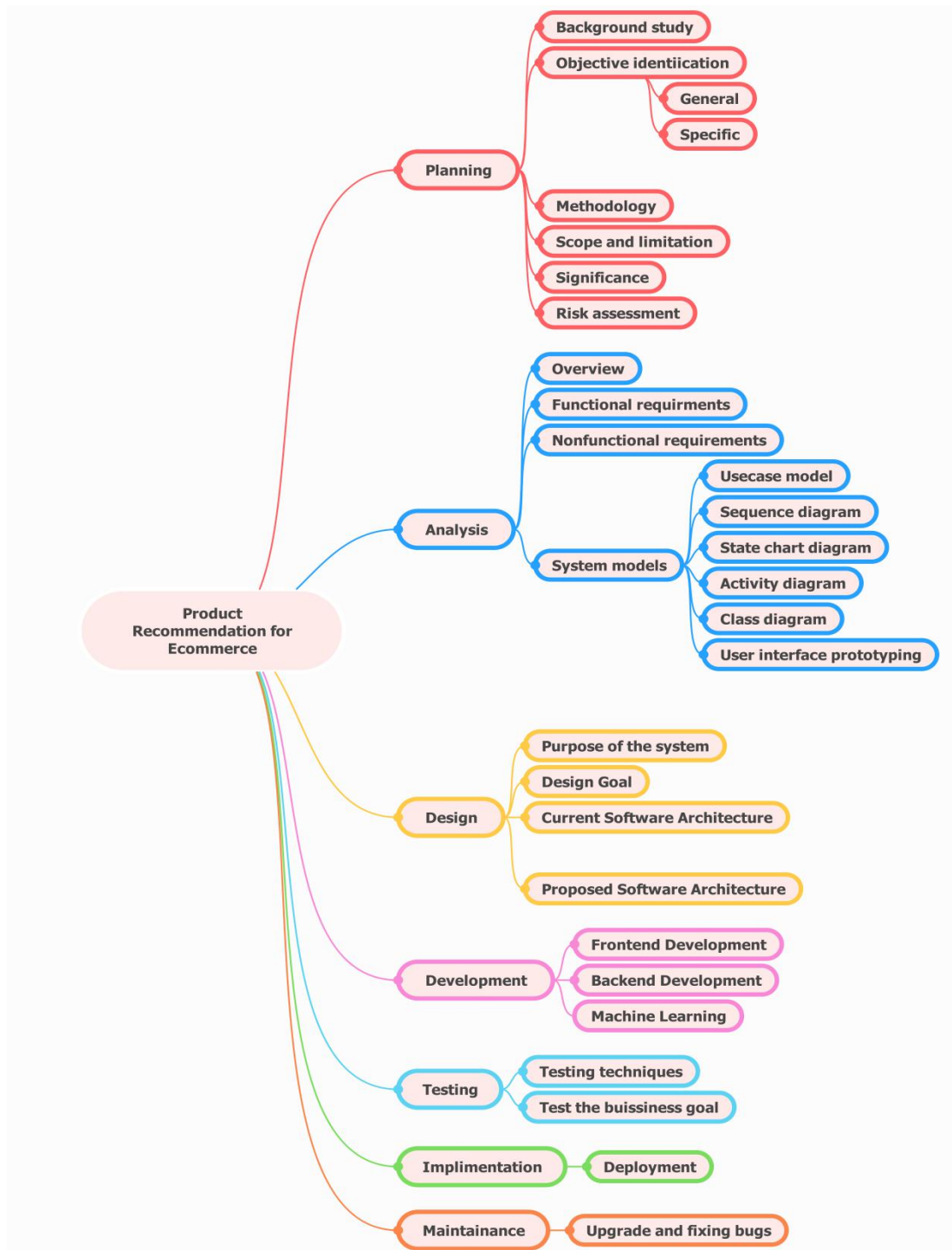


Figure 5. Work Break Down structure

Project Schedule (Gantt chart)

Gantt chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis. The width of the horizontal bars in the graph shows the duration of each activity. We illustrate our project schedule using gantt chart.

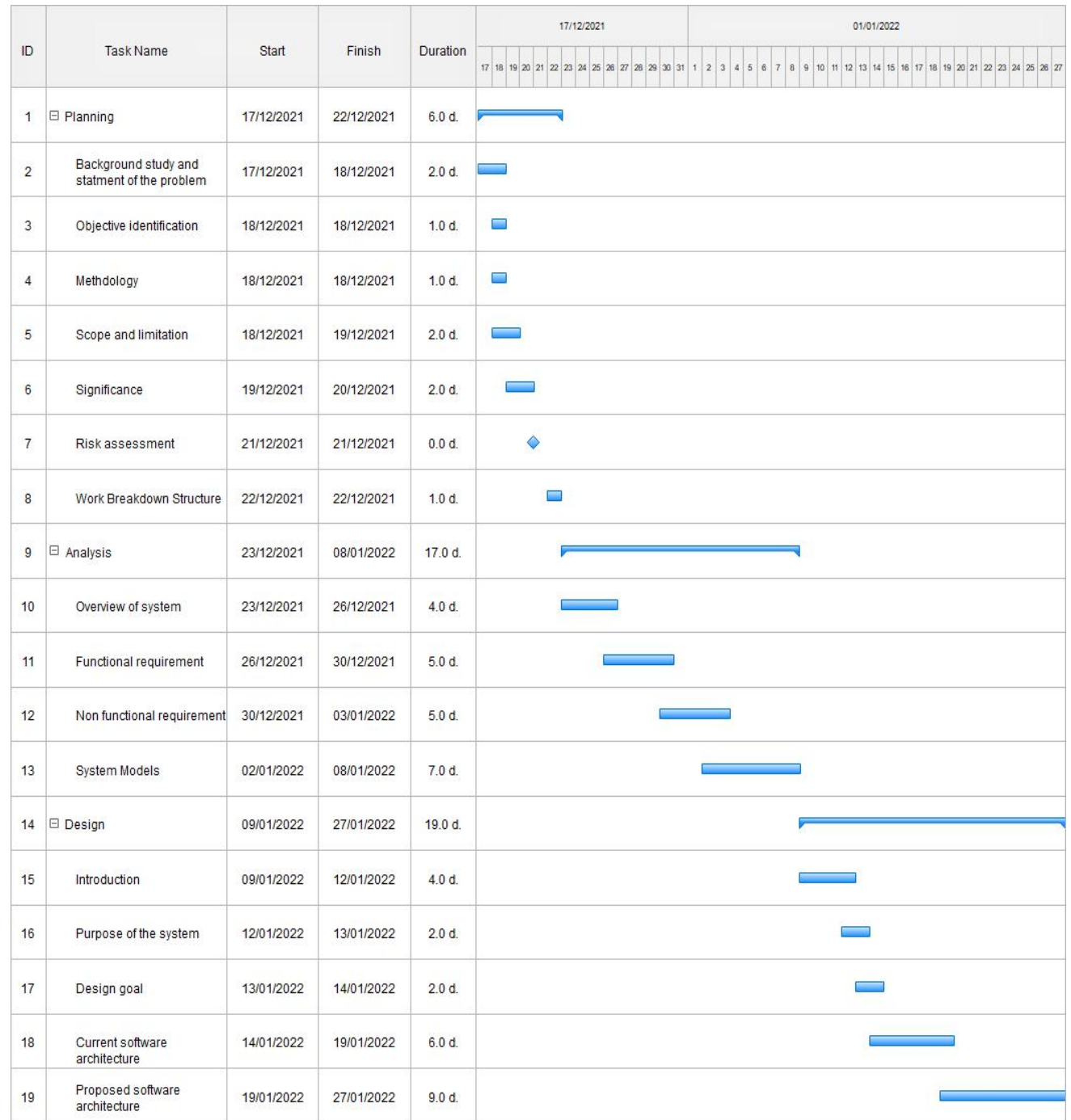


Table 1. Gantt chart

Chapter 2 Requirement Analysis and Specification

Requirement focuses on the collection of requirements from different the requirement elicitation, activities are performed to formulate the objectives for the system. Typical resulting artifacts are, for example, textual requirement descriptions, scenario descriptions, use cases, and sketches of prototypical user interfaces. The following recommendation approaches support activities related to requirements elicitation & definition.

There are a few methodologies of programming advancement, for example, water-fall, prototyping, winding, deft, etc. Every one of these will fundamentally incorporate phases of prerequisite, framework examination, design.... Alongside the development of the Internet, it is noticed that since online business was born in 1995 both the business to consumer and business to business e-commerce have been developing dramatically. As a kind of PC based data framework, internet business frameworks incorporate individuals, strategies, PC organizations, programming and data set.

Recommender frameworks are seriously applied to suggest items and administrations like films, books, computerized cameras, and monetary administrations.

A determination is a bunch of prerequisites that characterize site conduct and client collaboration. As such, prerequisites are capacities and functionalities that are needed in the finished result.

A determination archive incorporates general areas and some online business and stage explicit segments additionally fundamental, on the grounds that without a reasonable comprehension of all components, highlights and prerequisites, nobody can give you any ensures that turn of events and execution will remain affordable enough for you and timetable.

The field of RS had its beginnings in mental science, estimation hypothesis, data recovery, guaging hypothesis, the board science, and promoting. RS can be characterized as a framework that can assemble information from clients, total and interaction the information, and suggest a thing.

This paper endeavors to characterize a conventional client and thing model to be utilized in a system for proposal. An efficient audit of the writing was executed to concentrate on what client and thing data RSs use in the proposal cycle. This paper too presents and portrays a client and general thing model, and finishes up with a conversation of the outcomes. [Online Bank Account Creation and Monitoring For Commercial Bank of Ethiopia and Awash Bank,2020]

2.1 Current system

Knowing the current system very well, helps to identify the problems. Lets see ecommerce and recommendations separatly.

Recommendations:-Based on the analysis through observation and document analysis along with commercial data websites and public data sets for recommendations we identified that the current product recommendation system has these major services.

1. PERSONALIZED RECOMMENDATION:

In RS, the personalized recommendation system allows the application to perform users profile analysis and result customization. This also performs the recommendation in any format that is relevant to each and every user based on the user's implicit and explicit behaviors. Personalized recommendation engines are classified into five types. This depends on their approach to the recommendation system such as

- ✓ Content-Based Filtering
- ✓ Collaborative filtering
- ✓ Item based
- ✓ Demographic recommendations.

Content-based recommendation method: The Content-based recommendation method is based on the information about item content and ratings a user has given to items. This technique combines these ratings to profile of the user's interests based on the features of the rated items. The recommendations of a content-based system are based on individual information and ignore contributions from other users.

Collaborative Filtering: Collaborative filtering technique based on user's history in the form of rating given by the user to an item as their information source. It can be accomplished by making relation between the users or between items. Collaborative filtering is categorized into three types:user-based, item-based, and model-based.

For example lets take user-based approach.

User-based Approach: user based approach makes recommendation based on the interest of the user having the similar taste. It correlates user as per the rating given to the items.The collaborative filtering based recommendation system doesn't require the prior knowledge about the product descriptions and features; rather it completed independent in nature. So, different types of ecommerce sites can use this approach for new recommendations. Unlike the prior content based approach it evaluates the quality of the products and it expressed through user ratings and reviews.

Collaborative filtering techniques are able to perform recommendations to the individual based on their others preferences. Although the technique is good, the quality of the recommendations depends on the size of the previous rating dataset. It also generates gray sheep problem and privacy issues.

Item-based: Item-based Approach is based on the items as the user rated items similarly are probably similar.

Demographic recommendation: this technique uses information about user only. The demographic types of users include gender, age, and knowledge of languages, disabilities, ethnicity, mobility, employment status, home ownership and even location. The system recommends items according to the demographic similarities of the users.

2. NON-PERSONALIZED RECOMMENDATION:

Non-personalized recommendation system recommend items to consumers based on what other consumers have said about the product in an average. That is, the recommendations are independent of the customer, so all customers get the same recommendation. The Recommender systems are based on both personalized and non-personalized; however, analyzing and suggesting products need some historical information's and features of the item. It shortens customers' product searching time in online shopping, which improves the business online. In order to serve customers instantly and efficiently, it is essential to recognize each customer's unique and particular needs and recommend a personalized shopping list based on their interest and preferences. Finding effective recommendation system is also a major challenge.

Ecommerce (Without recommender system):- It is difficult for the customers to find the products they want in the existing ecommerce websites. Limited resource situation, data valid time and cold start problems are some of the issues that have not been very well considered in existing ecommerce. Various limitations are

- ✓ Lack of consumer satisfaction
- ✓ No personalized recommendation
- ✓ Unable to solve cold start problem
- ✓ Limited resource situation not properly handled
- ✓ Data valid time not handled properly and less efficient.

[International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN:2349-2163
Volume 2 Issue 1 (January 2015)]

2.2 Business rules

Business rules, in their simplest definition, are directives that define (or constrain) business activities. Designed to help an organization achieve its goals, those without clear business rules and procedural definitions will often find a chaotic workplace with inconsistent outcomes, poor morale amongst employees, and dissatisfied customers.

Before moving on, let's look at how you'd build a recommender system. Assuming you already have a platform in the shape of a website or an app where you want to add a recommender system, it'd go something like the cycle shown in below

(figure-data driven approach to building recommender system). [Business and Rule Examples,2022]

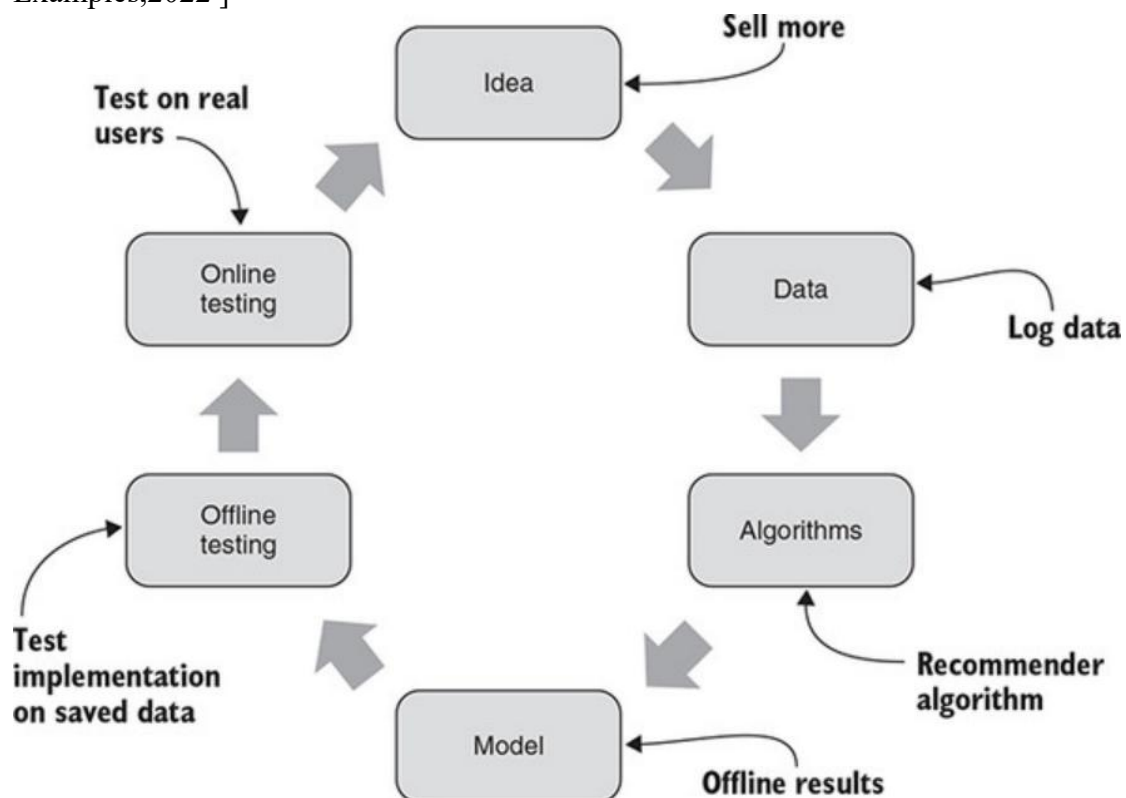


Figure 6. A data-driven approach to building a recommender system

2.3 Proposed system

Recommender system provides users with their individual tastes and services of recommendation which overcomes the problem of retrieving users' needs due to information overload. There are different ways of building recommendation systems techniques such as collaborative algorithm and content-based algorithm.

There are various reasons as to why we want to use recommender systems.

- Increase the number of items consumed by the user: In an online shopping business this may translate into “increasing the number of items or products sold”, whereas in a news portal this may translate into “increasing the number of news items read by the user”.
- Enable the user to select items that may be hard to find without a precise recommendation. This functionality enables non-popular items and items that the users do not usually consume to stand out in the way of the user, so she can notice them.
- Increase the user satisfaction. A well designed recommender system enhances the experience of the user with the site. The user will find the recommendations interesting, relevant and, with a properly designed human-computer interaction, she will also enjoy using the system. The combination of effective, i.e., accurate, recommendations and a usable interface will increase the user's subjective

evaluation of the system. This in return will increase system's usage and the likelihood that the recommendations will be accepted, will secure the users' loyalty.

- Better understand the user's tastes: From the users' preferences, either collected explicitly or predicted, service providers and business managers may build new marketing policies that suits users' tastes on an individual level.

[Uni Assignment Center,2022]

2.3.2 Functional requirements

These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. (Ian Sommerville , 2011)

These are the main functional requirements in our system:

Manage product is used to allow admin to manage product on the database.

Search product used to allow customer and admin to search for product.

Add to cart is used to allow customer to add products to the cart.

Buy Product is used to allow the customer to buy the product.

Review is used to allow customer to give review on the products.

Fill feed back form is used to allow the customer to fill feed back form.

2.3.3 Nonfunctional requirements

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services. (Ian Sommerville , 2011)

These are the non functional requirement of our system:

✓ **Usability**

Our system is easy to learn and operate. Our User interface for this system will be simple and clear. Our service doesn't require special training.

✓ **Availability**

Our system is available in everywhere through the internet. Only restricted by the down time of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will be shown.

✓ **Performance:**

Our system will have good performance i.e., fast response time and optimal workload.

✓ **Security:**

Our system have a security privilege that secures the system. And should not leave any cookies on the user's computer containing the user's password; our system's back-end servers shall only be accessible to authenticated administrators.

✓ **Portability:**

Our system is fully portable because its development bases on html,css and javascript which are highly supported by every web browser and OS. So That end user part is fully portable and any system using any web browser should be able to use the features of the system.

✓ **Error handling:**

When the users of the system interact with the system errors may appear. To control these inaccuracies the system will generate different messages. Data errors which are entered in to the system may not be completely avoided but they can be minimized. To do this, most of the system execution buttons will be controlled according to the sequence which the user is expected to follow. Or this can be done by generating different system responses to the input of the users.

✓ **Reliability:**

Our system is effective and consistent in that integrity of information is maintained and supplied to the system.

2.3.4 System models

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system.

It is about representing a system using some kind of graphical notation, which is now almost always based on notations in the **Unified Modeling Language (UML)**.

Models help the analyst to understand the functionality of the system; they are used to communicate with customers.

Models can explain the system from **different perspectives**:

- An **external** perspective, where you model the context or environment of the system.
- An **interaction** perspective, where you model the interactions between a system and its environment, or between the components of a system.
- A **structural** perspective, where you model the organization of a system or the structure of the data that is processed by the system.
- A **behavioral** perspective, where you model the dynamic behavior of the system and how it responds to events.

2.3.4.1 Use case model

A use case can be taken as a simple scenario that describes what a user expects from a system. Each use case represents a discrete task that involves external interaction with a system. In its simplest form, a use case is shown as an ellipse with the actors involved in the use case represented as stick figures.(Ian Sommerville , 2011)

List of Use cases and actors

List of use cases

1. Login
2. Search product
3. Manage product
4. Buy Product
5. Add to cart
6. Review
7. Fill feed back form

List of actors

1. Admin
2. Customer
3. Database

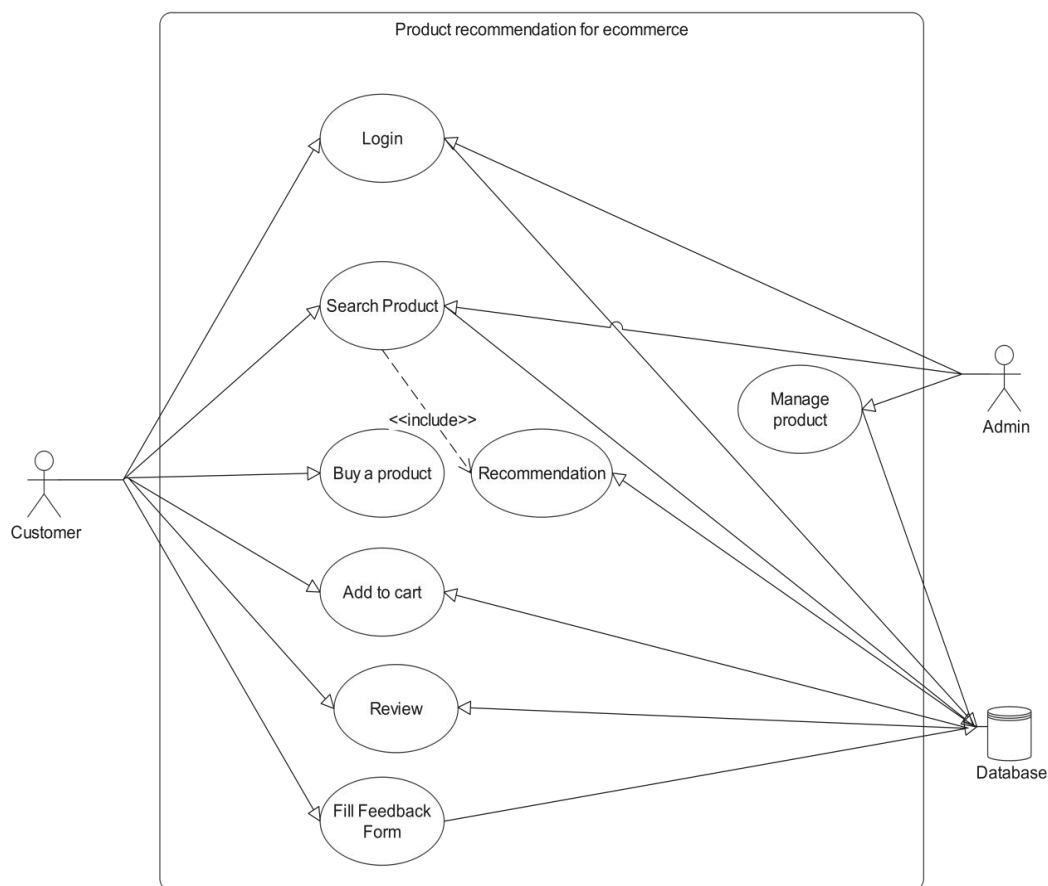


Figure 7. Use-case diagram

General description of Use case and actors

Use cases description

1. **Login** is used to allow customer and admin to get in to the system.
2. **Manage product** is used to allow admin to manage product in the database.
3. **Search product** used to allow customer and admin for product.
4. **Add to cart** is used to allow customer to add products to the cart.
5. **Buy Product** is used to allow the customer to buy the product.
6. **Review** is used to allow customer to give review on the products.
7. **Fill feed back form** is used to allow the customer to fill feedback details.

Actors description

1. **Admin** is a person who manages the system.
2. **Customer** is a person who uses the system.
3. **Database** is used for storing and retrieval of information.

Use-case detailed description

1. Use case name	- Login.
2. Use case ID	- PRS01.
3. Participate actors	-Admin. - Customer. - Database.
4. Precondition	- Get registered and have an account
5. Main-flow events	5.1 The user click on login. 5.2 The system display login interface. 5.3 The user enter login details. 5.4 The system verify login detail from server side. 5.5 The system open the dashboard. 5.6 Use case ends.
6. Alternative event	6.1 If the user enter wrong login detail. 6.1.1 The system advise the user to retry again. 6.2 If login detail is incorrect. 6.2.1 The system advise the user to retry again.
7. Post condition	- Other functional process.
8. Include	- None.
9. Extend	- None.

Table 2. Detail description of use case for Login

1. Use case name	- Manage product.
2. Use case ID	- PRS02.
3. Participate actors	-Admin. - Database.
4. Precondition	- Admin must logged in.
5. Main-flow events	5.1 The admin click on manage product. 5.2 The system display manage product interface. 5.3 The admin manage product.(Add, Delete, Edit details) 5.4 The system manage product details to database in server side. 5.5 The system show success message to admin. 5.6 Use case ends.
6. Alternative event	6.1 If the admin enter incomplete product detail. 6.1.1 The system advise the fill all details.
7. Post condition	- Other functional process.
8. Include	- None.
9. Extend	- None.

Table 3. Detail description of use case for Manage product

1. Use case name	- Search product.
2. Use case ID	- PRS03.
3. Participate actors	-Customer. - Database.
4. Precondition	- Customer must logged in.
5. Main-flow events	5.1 The customer click on search. 5.2 The system display search interface. 5.3 The customer enter search details. 5.4 The system match search detail from server side. 5.5 The system open the result and recommend other products. 5.6 Use case ends.
6. Alternative event	6.1 If the customer enter wrong search detail. 6.1.1 The system advise the user to retry.
7. Post condition	- Other functional process.
8. Include	- Recommendation.
9. Extend	- None.

Table 4. Detail description of use case for Search product

1. Use case name	- Add to cart.
2. Use case ID	- PRS04.
3. Participate actors	- Customer. - Database.
4. Precondition	- Customer must see products.
5. Main-flow events	5.1 The customer click on add to cart. 5.2 The system display stored message to customer. 5.3 The system store products to buy in database server side. 5.4 The system shows other products. 5.5 Use case ends.
6. Alternative event	- none
7. Post condition	- other functional process.
8. Include	- None.
9. Extend	- None.

Table 5. Detail description of use case for Add to cart

1. Use case name	- Buy product.
2. Use case ID	- PRS05.
3. Participate actors	- Customer. - Database.
4. Precondition	- Customer need to add to cart.
5. Main-flow events	5.1 The customer click on product. 5.2 The system display product description. 5.3 The customer click buy. 5.4 The system match store buy detail on database. 5.5 The system show message to customer process is completed. 5.6 Use case ends.
6. Alternative event	- none.
7. Post condition	- other functional process.
8. Include	- None.
9. Extend	- None.

Table 6. Detail description of use case for Buy product

1. Use case name	- Review.
2. Use case ID	- PRS06.
3. Participate actors	- Customer. - Database.
4. Precondition	- Customer must logged in.
5. Main-flow events	5.1 The system display review(stars). 5.2 The customer click on starts(review). 5.4 The system stores in database on server side. 5.6 Use case ends.
6. Alternative event	- none
7. Post condition	- other functional process.
8. Include	- None.
9. Extend	- None.

Table 7. Detail description of use case for Review

1. Use case name	- Fill feed back.
2. Use case ID	- PRS07.
3. Participate actors	- Customer . - Database.
4. Precondition	- Customer must logged in.
5. Main-flow events	5.1 The customer click on Contact us. 5.2 The system display Contact us interface. 5.3 The customer enter feedback details and send. 5.5 The system store in database. 5.6 Use case ends.
6. Alternative event	6.1 If the user enter wrong feedback detail. 6.1.1 The system advise the user to retry.
7. Post condition	- other functional process.
8. Include	- None.
9. Extend	- None.

Table 8. Detail description of use case for Fill feed back

2.3.4.2 Sequence diagram

Sequence diagrams in the UML are primarily used to model the interactions between the actors and the objects in a system and the interactions between the objects themselves. The UML has a rich syntax for sequence diagrams, which allows many different kinds of interaction to be modeled. As the name implies, a sequence diagram shows the sequence of interactions that take place during a particular use case or use case instance. (Ian Sommerville , 2011)

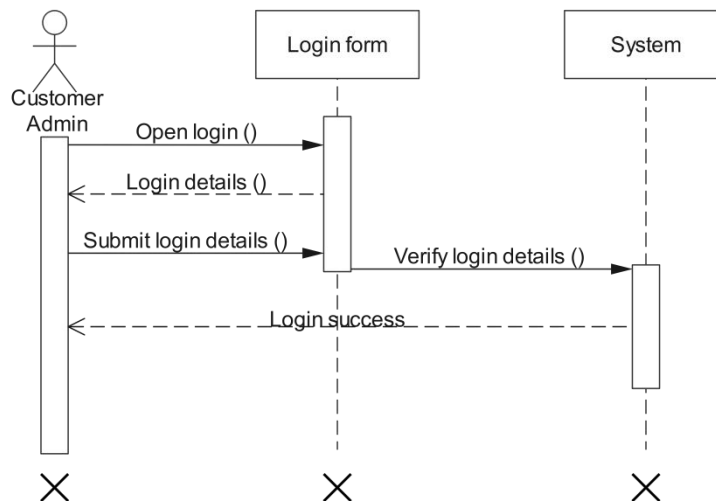


Figure 8. Sequence diagram for Login

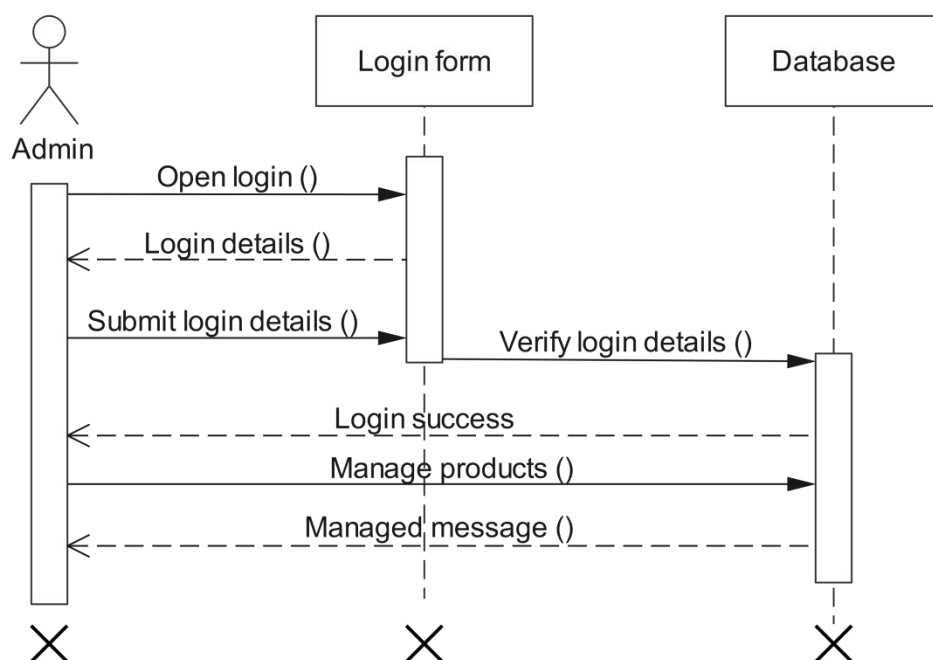


Figure 9. Sequence diagram for manage products

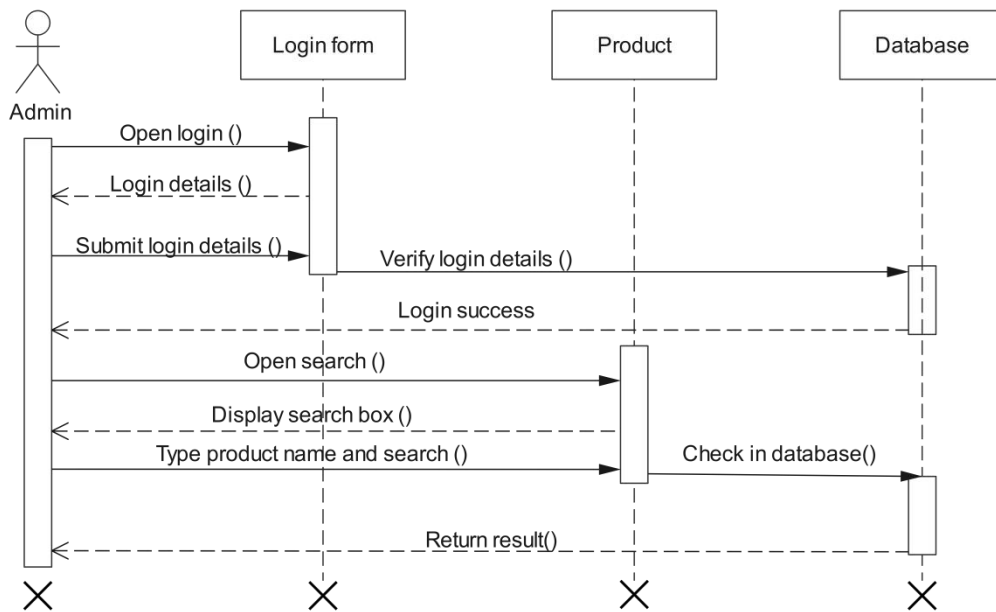


Figure 10. Sequence diagram for search products

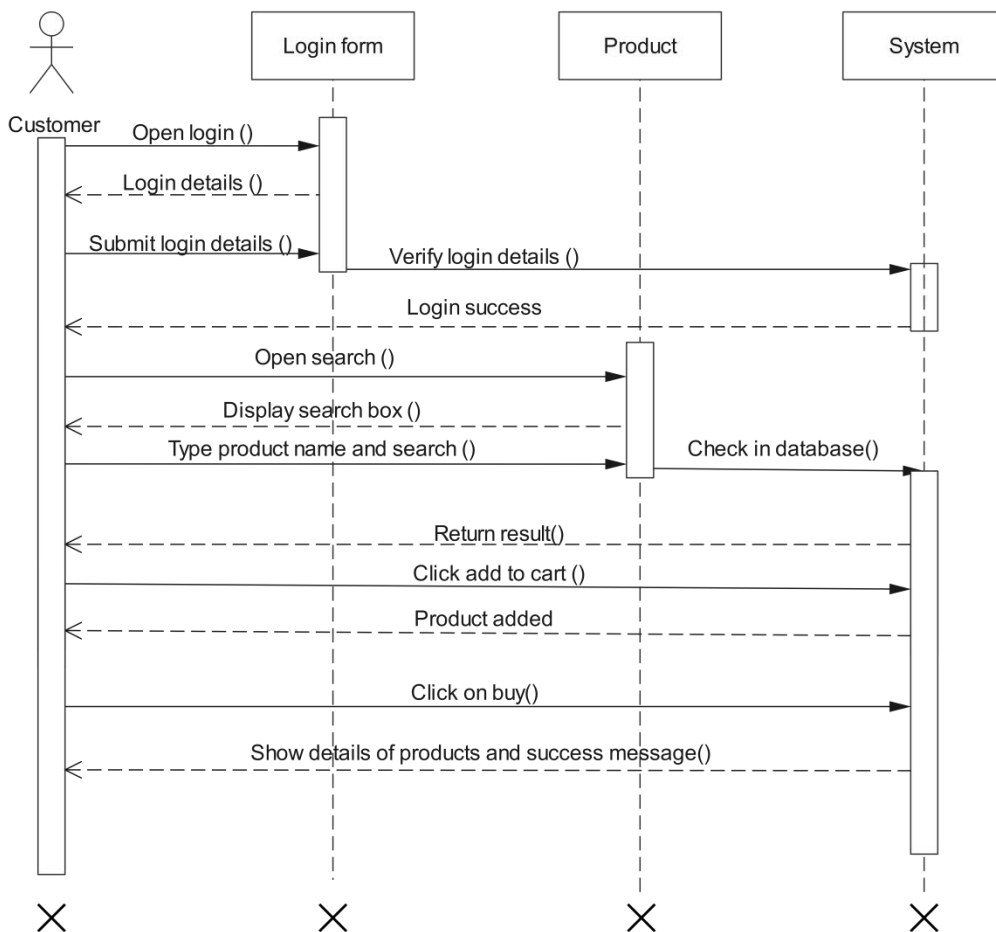


Figure 11. Sequence diagram for buy products

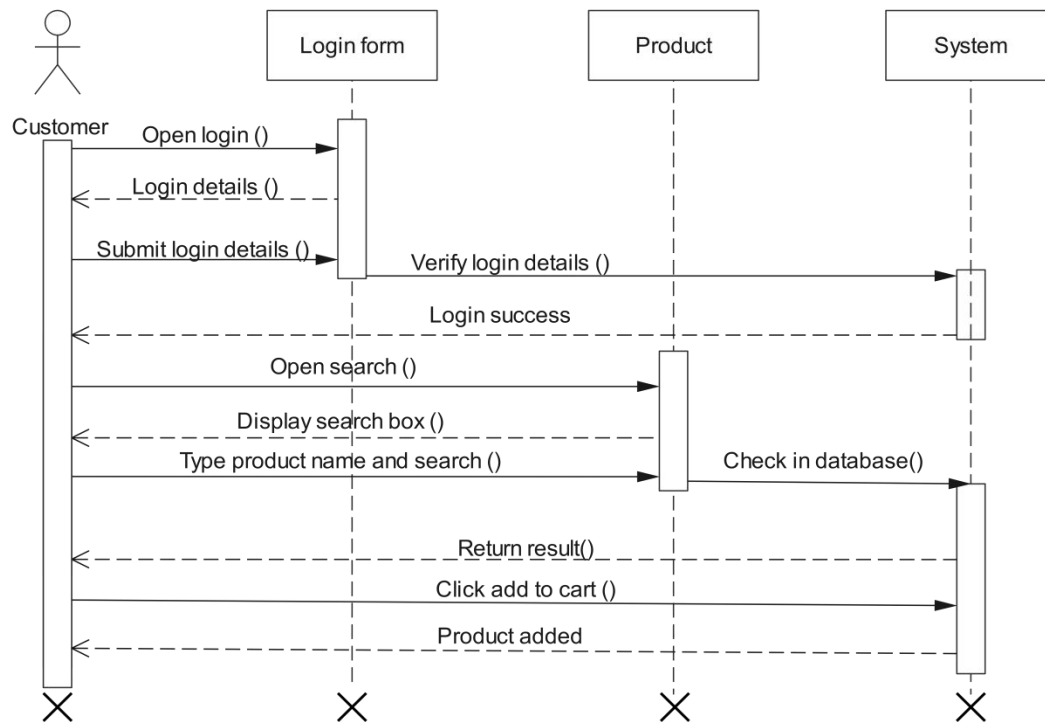


Figure 12. Sequence diagram for add to cart

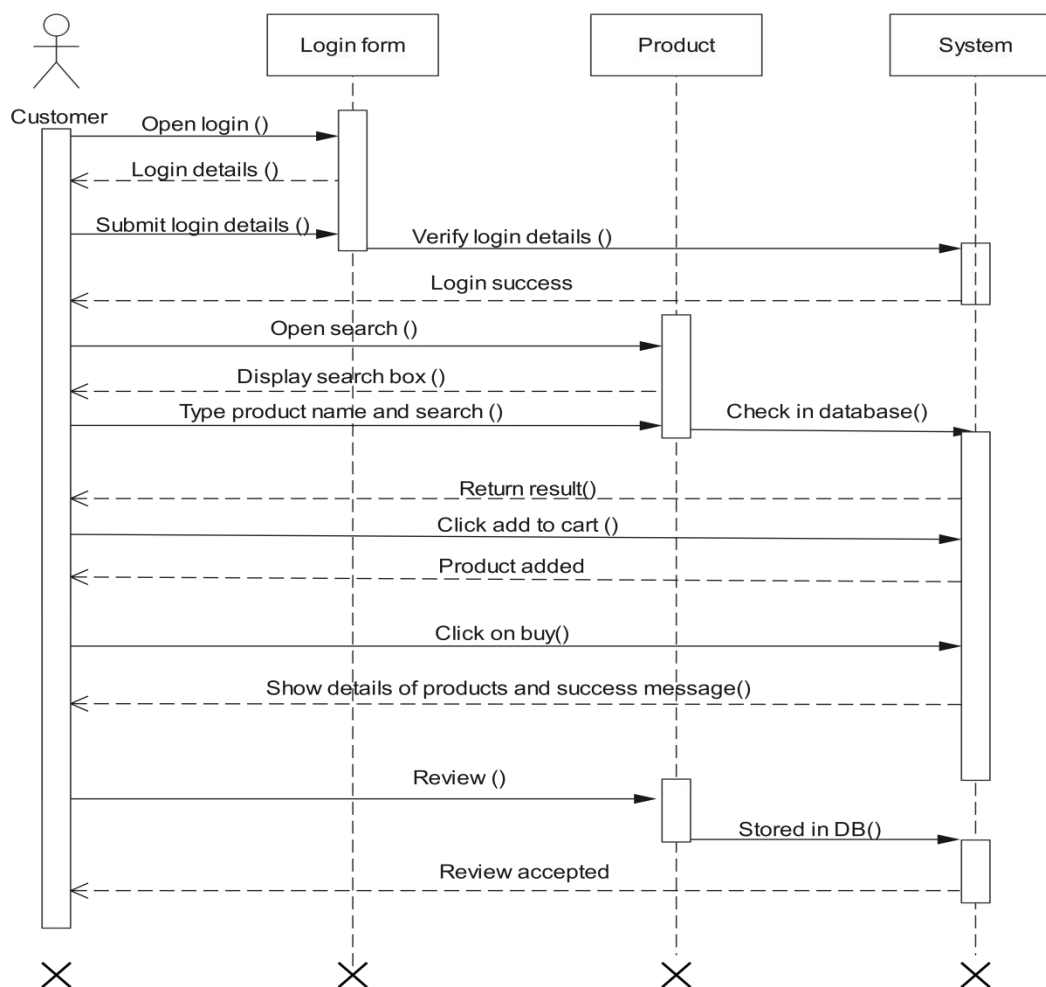


Figure 13. Sequence diagram for review

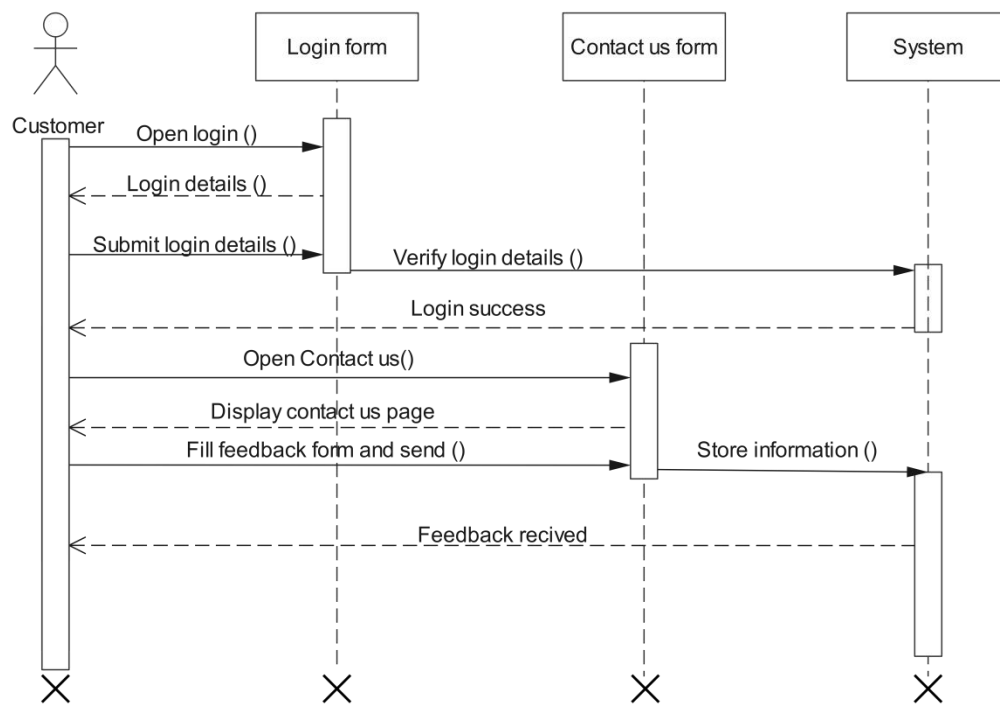


Figure 14. Sequence diagram for feedback form

2.3.4.3 State chart diagram

State diagrams are used to model a system's behavior in response to internal or external events. State diagrams show system states and events that cause transitions from one state to another. They do not show the flow of data within the system but may include additional information on the computations carried out in each state.

(Ian Sommerville , 2011)

State chart Diagram for product recommendation for ecommerce

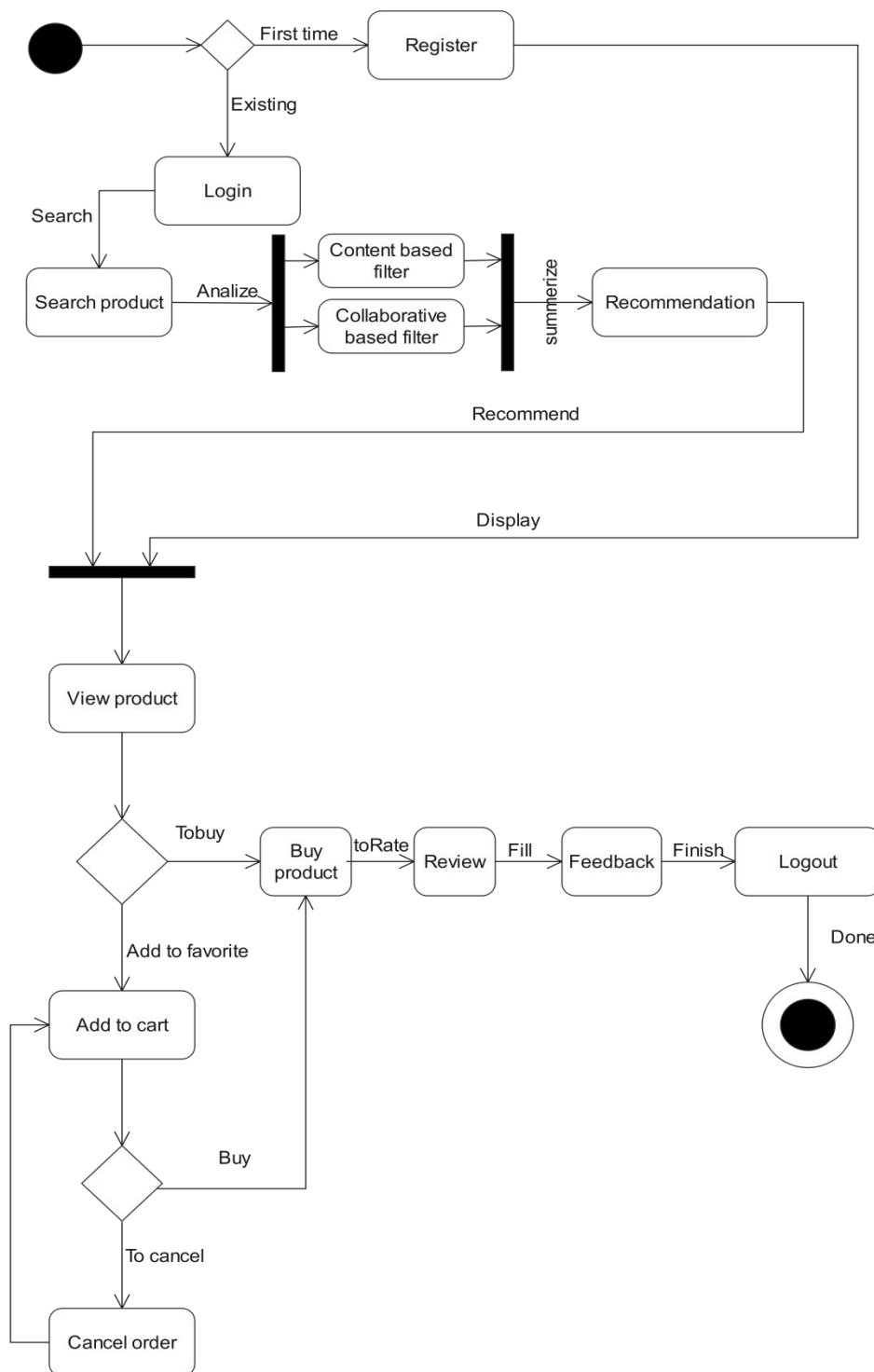


Figure 15. State chart diagram

2.3.4.4 Activity diagram

Activity diagrams may be used to model the processing of data, where each activity represents one process step. Activity diagrams are intended to show the activities that make up a system process and the flow of control from one activity to another. The start of a process is indicated by a filled circle; the end by a filled circle inside another circle. Rectangles with round corners represent activities, that is, the specific sub-processes that must be carried out. (Ian Sommerville , 2011)

Activity diagram for product recommendation for ecommerce

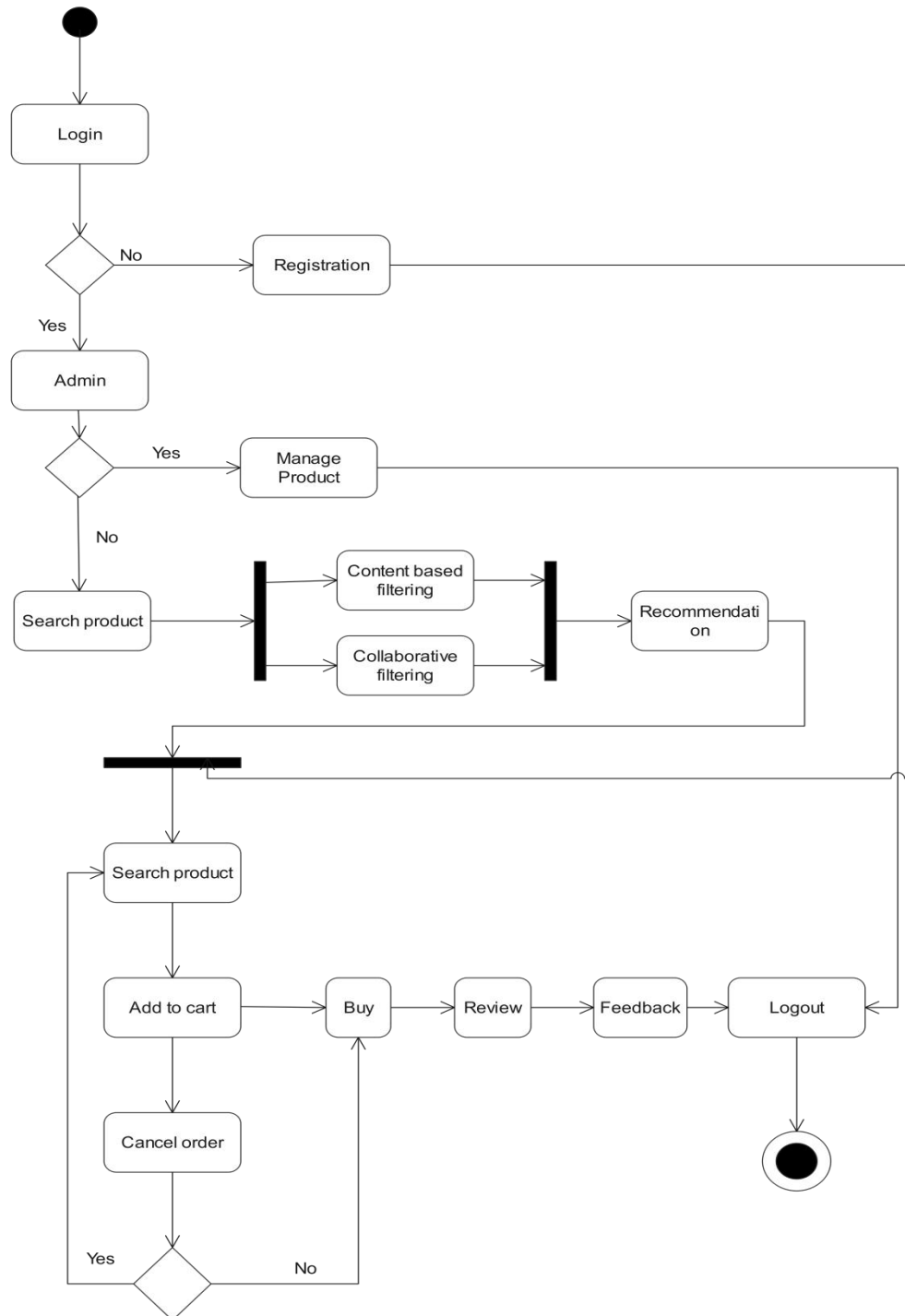


Figure 16. Activity diagram

2.3.4.5 Class diagram

Class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes. Loosely, an object class can be thought of as a general definition of one kind of system object. An association is a link between classes that indicates that there is a relationship between these classes. Consequently, each class may have to have some knowledge of its associated class. (Ian Sommerville , 2011)

1. List of Class:

- | | |
|------------|-------------------------|
| ✓ Product | ✓ Recommendation system |
| ✓ User | ✓ Recommendation List |
| ✓ Admin | ✓ Feedback |
| ✓ Customer | ✓ Review |
| ✓ Cart | ✓ Datamodel |

2. List of Attribute:

- ✓ **Product:** Product_id, Product_type, Product_catagoty, Product_Description, Product_Price
- ✓ **User:** user_id, password, email
- ✓ **Admin:** adminName
- ✓ **Customer:** customerName
- ✓ **Cart:** cartid, quantity, dateAdded
- ✓ **Recommendation system:** PredictedRating, avarageRating
- ✓ **Feedback:** message
- ✓ **Review:** noOfStar
- ✓ **Datamodel:** modelParam,modelLoss

3. List of Operations:

- ✓ **Product:** Display(), getProductDetail()
- ✓ **User:** Login(), Logout(), verifyLogin(), SearchProduct()
- ✓ **Admin:** manageProduct()
- ✓ **Customer:** Buy()
- ✓ **Cart:** additem(), removeitem(), updateQuantity(), viewCartDetail()
- ✓ **Recommendation system:** Recommend()
- ✓ **Feedback:** send()
- ✓ **Review:** rate()
- ✓ **Datamodel:** trainModel(), getTrainingData(), test(), acuuracy()

4. Relationship of Classes:

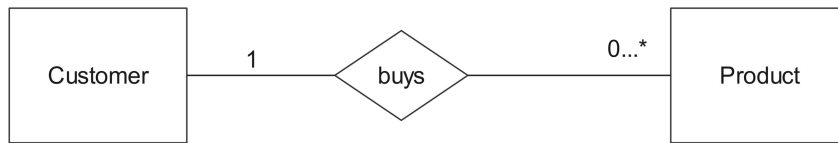


Figure 17. Customer buys Product

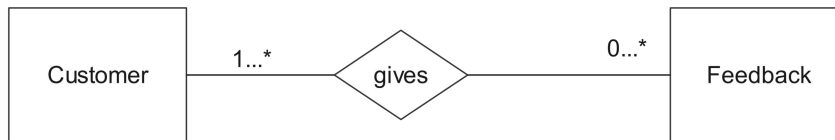


Figure 18. Customer gives Feedback

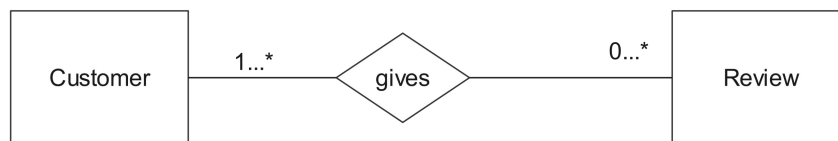


Figure 19. Customer gives Review

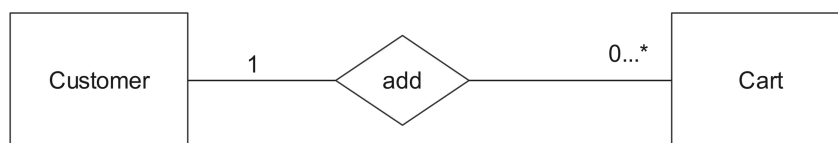


Figure 20. Customer add to Cart

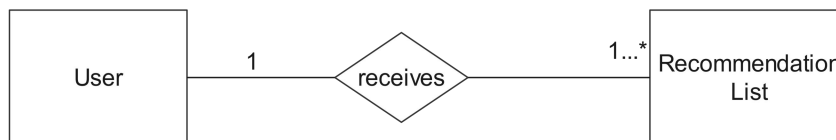


Figure 21. User receives Recommendation list

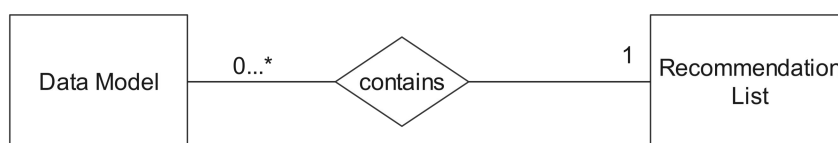


Figure 22. Data model contains Recommendation list

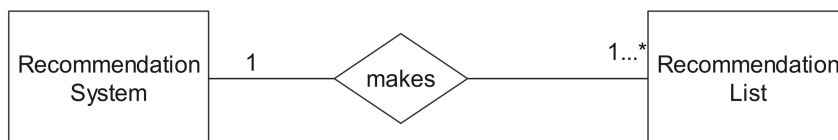


Figure 23. Recommendation system makes Recommendation list

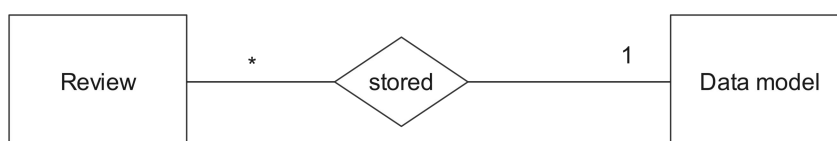


Figure 24. Review stored in Data model

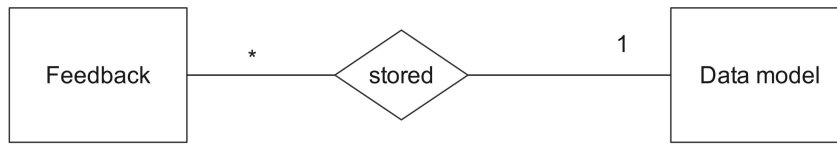


Figure 25. Feedback stored in Datamodel

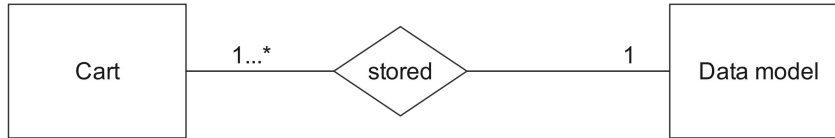


Figure 26. Cart stored in Datamodel

5. Access modifier

	Public	Private	Protected
Product	Product_id, Product_type, Product_catagoty, Product_Description, Product_Price		
User		user_id, email	password
Admin		adminName	
Customer		customerName	
Cart		cartid, quantity, dateAdded	
Recommendation system			
Feedback		message	
Review		noOfStars	
Datamodel	modelParam, modelLoss		

Table 9. Access modifiers

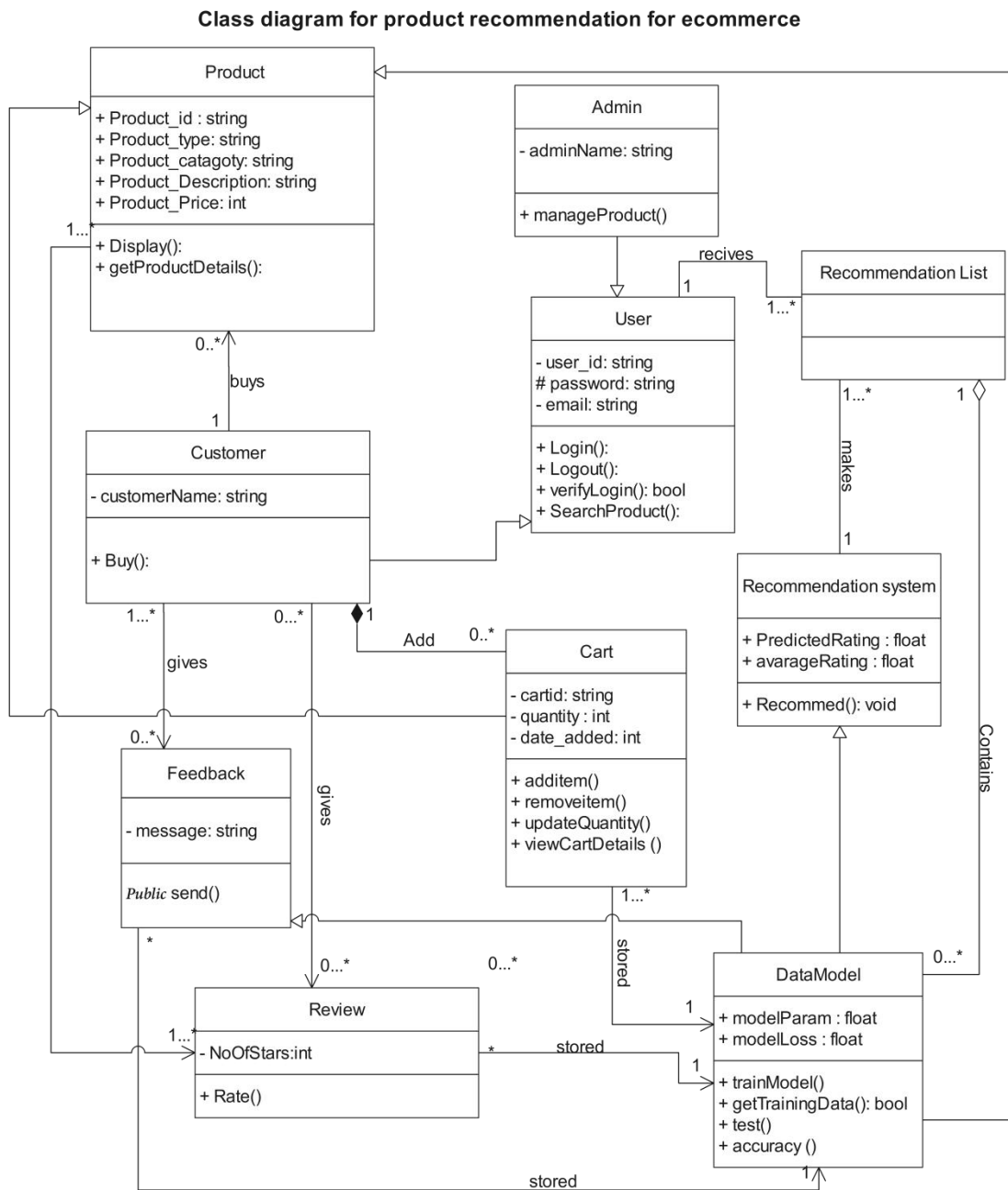


Figure 27. Class diagram

2.3.4.6 User interface prototyping

UI Prototypes are an essential part of designing user flows and interfaces. They allow designers to show their design through an interactive and engaging product, resulting in a better understanding of the design for everyone involved. The goal of effective UI is to make the user's experience easy and intuitive, requiring minimum effort on the user's part to receive maximum desired outcome.

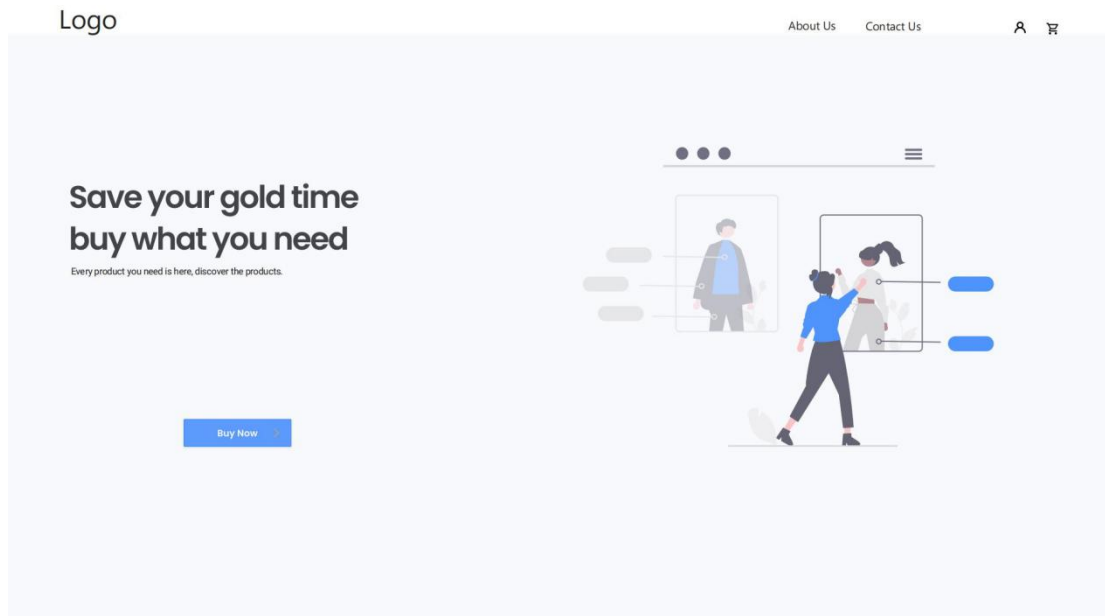


Figure 28. Landing Page

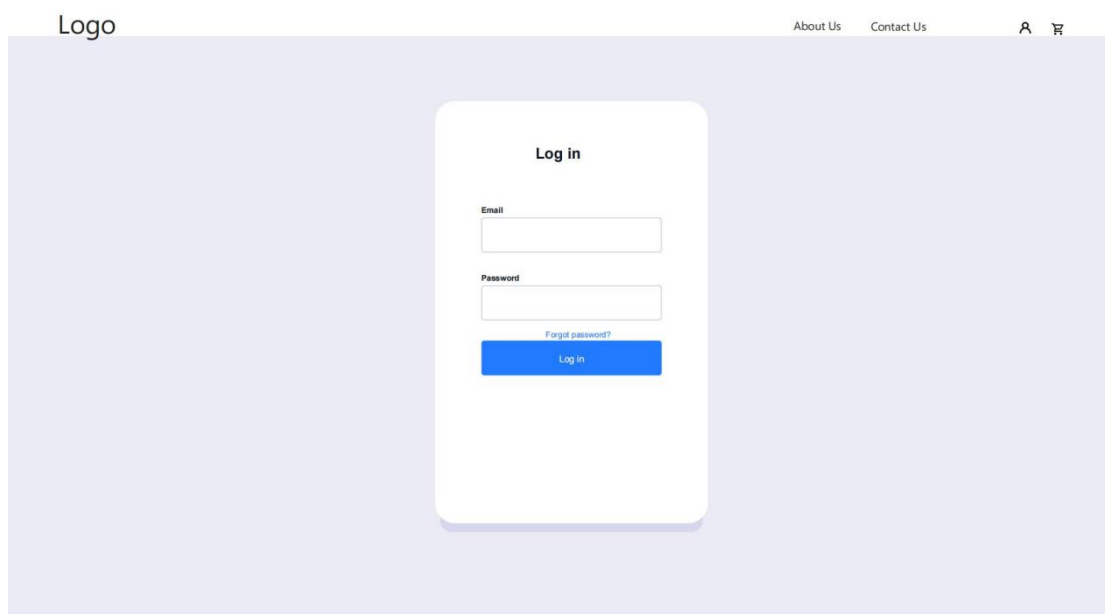


Figure 29. Login Page

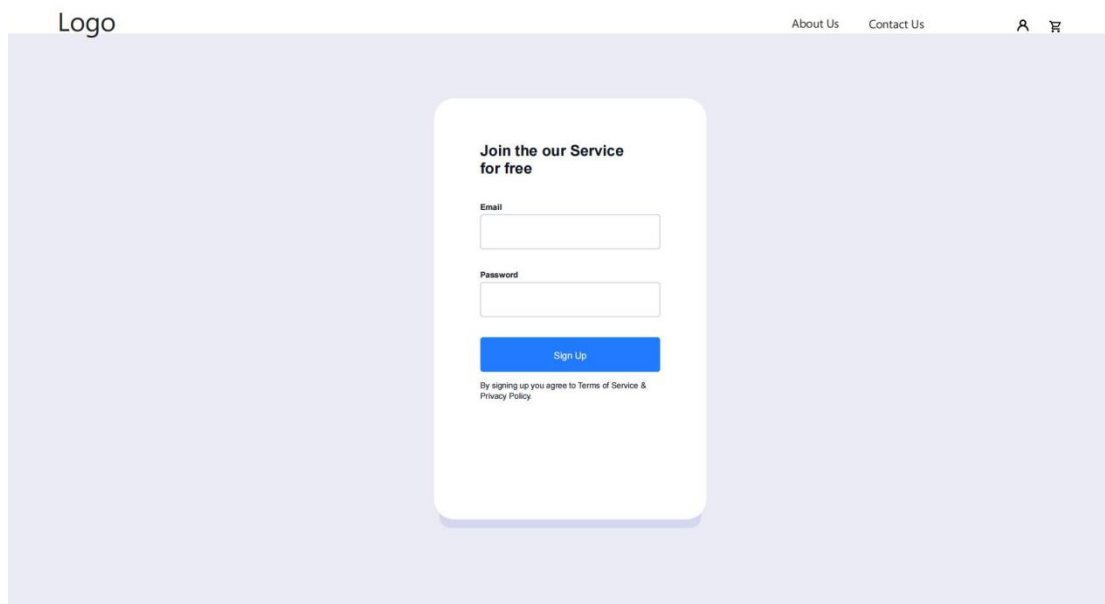


Figure 30. Sign up Page

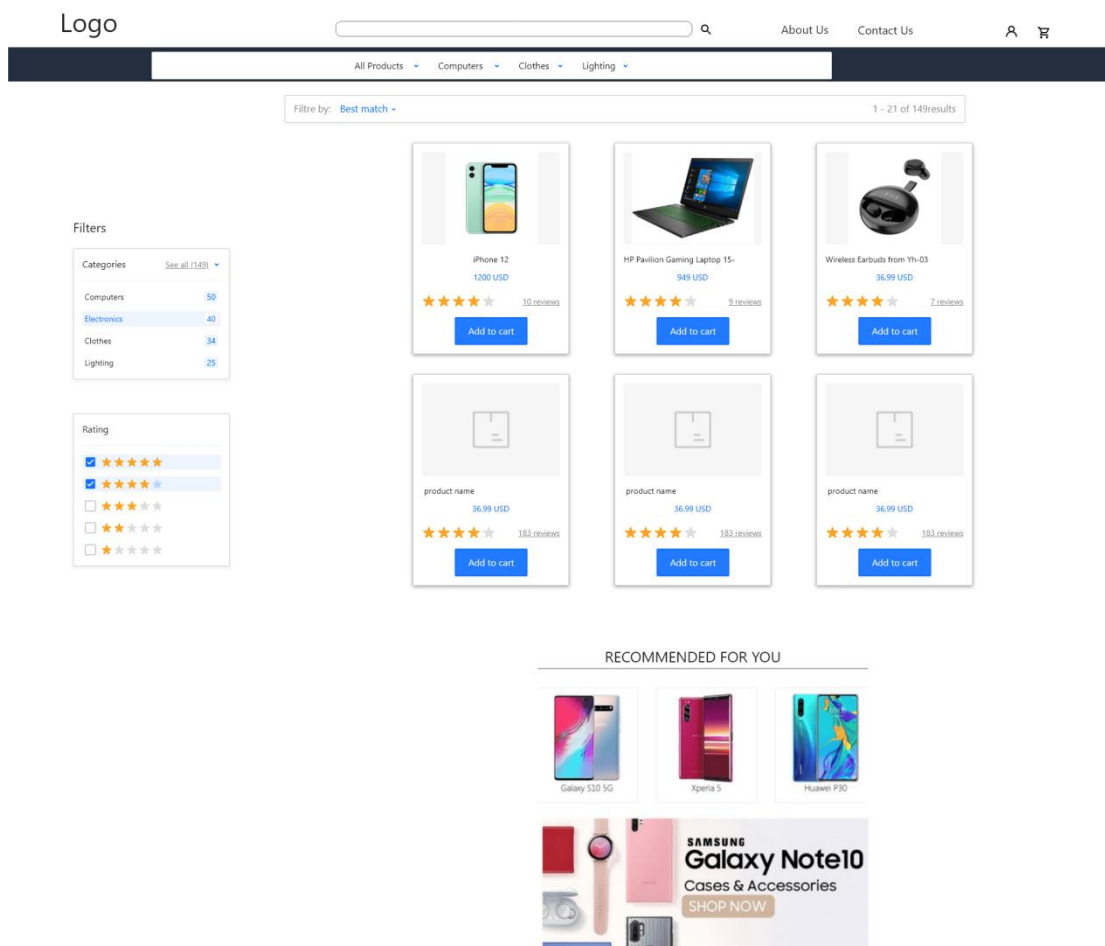


Figure 31. Product Page(home page)

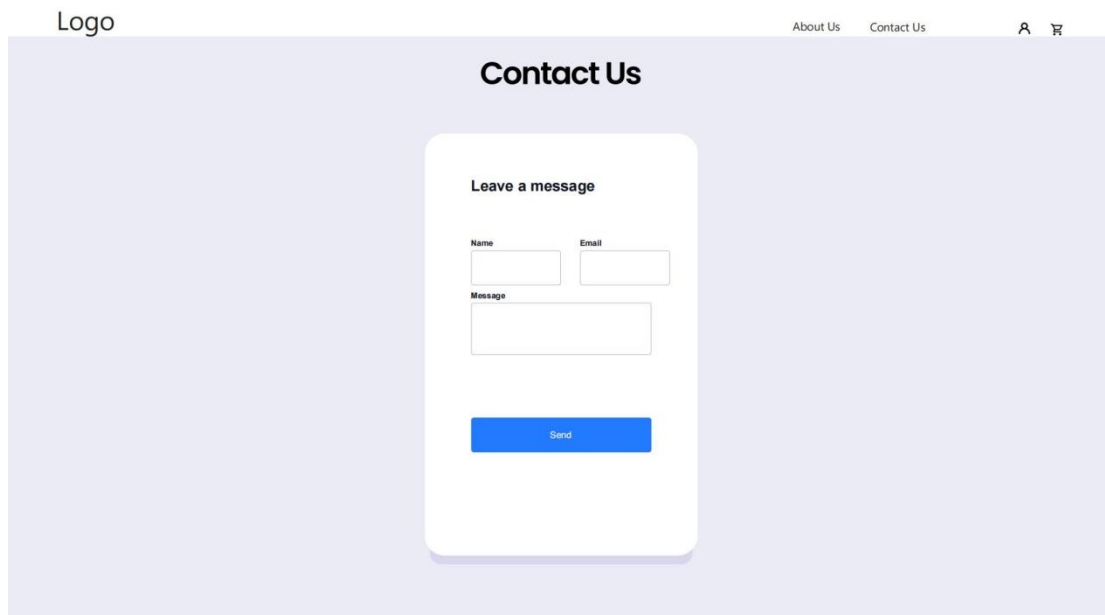


Figure 32. Contact Us Page

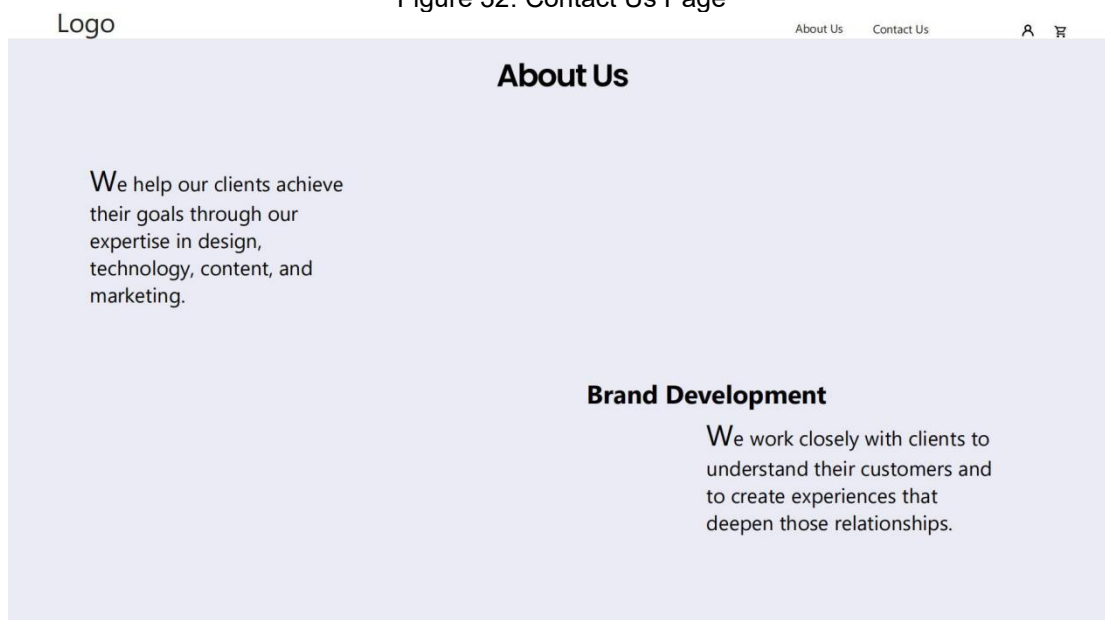


Figure 33. About Us Page

Chapter 3 System design

3.1 Introduction

System design system design is a clear description of each of these strategies, a subsystem decomposition, and a deployment diagram representing the hardware/software mapping of the system.[Bruegge,_Bernd_Dutoit,Allen H. Dutoit, 2014]

Systems design is major phase of information system. It is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System developers additionally select techniques for building the system, like the hardware/software methodology, the persistent data management strategy, the worldwide control stream, the access control strategy, and the treatment of boundary conditions. The consequence of system design is a model that incorporates subsystem decomposition and a clear description of each of these procedures.

We use object-orient software engineering approach, so in this chapter we will cover about purpose of the system, design goals, software architecture, sub-system decomposition, component diagram, deployment diagram, Persistent data management, access control and security, global software control, boundary conditions.

In this chapter, we will see about:

- ❖ Identifying design Goals: in the process of going from the analysis model to building the system design, design goals must be identified.
- ❖ Identifying Subsystems:Assign objects relating to one use case to the same subsystem
- ❖ Identifying persistent data:Examine all classes that must survive system shutdown (controlled or unexpected). Good indicator of persistent data.
- ❖ Identifying boundary condition: must address initialization, termination and failure of the system.

3.2 Purpose of the system

The significance of the project is to raise user experience by ensuring the user receives regular recommendations that suit their taste through efficiency and the system will offer customers recommendations by sorting through their searched result and what they have purchased. This is because a recommendation system is a perfect way to offer a captivating user experience means productive. And in market sector also this recommendation system massively will increase sales and profitability.

3.3 Design goals

Our system has the following Design goals to achieve its functionality. We discussed some the design goals as Non-functional requirement dealing with:

- ❖ The system will have good performance i.e., fast response time and optimal workload as performance.
- ❖ The system will provide reliable and accurate recommendation.
- ❖ The system is easy to learn and operate. The User interface for this system will be simple and clear. i.e., the service doesn't require special training as usability.
- ❖ This system is available in everywhere through the internet as availability.
- ❖ The system should have a security privilege that secures the system as security.
- ❖ The system is available anyone can share and use anywhere as portability.
- ❖ The system will support a number of user at a time.
- ❖ When the users of the system interact with the system errors may appear. To control these inaccuracies the system will generate different messages. Data errors which are entered in to the system may not be completely avoided but they can be minimized. To do this, most of the system execution buttons will be controlled according to the sequence which the user is expected to follow. Or this can be done by generating different system responses to the input of the users as error handler.
- ❖ The system is effective and consistent in that integrity of information is maintained and supplied to the system as reliability.
- ❖ The system will minimize the cost.

3.4 Current software architecture

let see architecture like **Netflix**

The architecture is divided into three layers: Online, Nearline and Offline.

- The **online** layer addresses recent user events, delivering real-time response, restricting the computational complexity of the algorithms and the volume of data that can be processed.
- The Near-line layer is the middle layer, providing a computation similar to online, but with a longer response time.
- The layer offline has no limitation on the amount of data and computational complexity, but has a flexible response time and the results can become outdated constantly due to the data frequent renewal.

The following Figure illustrates the architecture of the Netflix recommendation system:

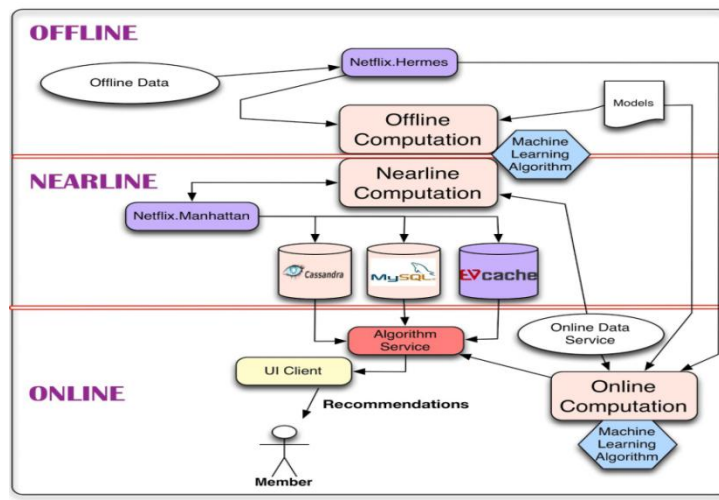


Figure 34. Illustrates the architecture of the Netflix recommendation system.

(Chong david, 2021)

3.5 Proposed software architecture

We propose this architecture as we are working on the ecommerce, and for the recommender system.

3 - tier architecture separates it tier from each other on basis of users.

1. **Client- side:** At this layer multiple views of database can be provided by the application. All views are generated by applications, which reside in Django server.
2. **Application tier:** Where system's machine learning model is inside the Django server. Django is a web server and a python framework for delivering web pages to users, a web server is that accepts requests with HTTP. Django can also handle different request types like GET, POST.
3. **Database tier:** At this tier, only databases reside and Database server is the server runs the database applications. In the system we use MySQL server database application because it is most compatible to Microsoft technologies we use for data retrieval purpose.

As we mentioned before our recommendation system have three tier and it looks like diagrammatically represented:

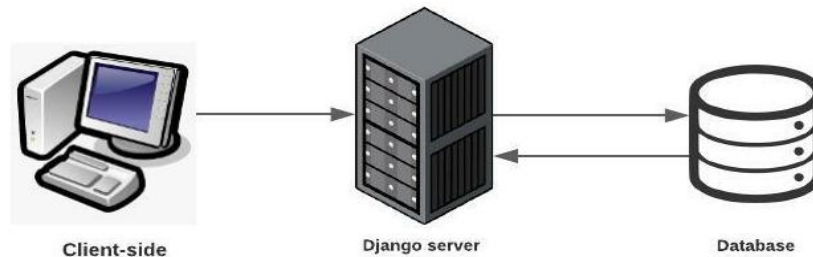


Figure 35. System architecture on recommender system.

3.5.1 Sub-system decomposition

Subsystem decomposition reduces the complexity of the system. It describes the decomposition into subsystems and the responsibilities of each. This is the main product of system design.[Allen H. Dutoit, 2014]

- **Login sub-system:** is utilized to permit customer and admin to get in to the system and login is a time saver for people who don't want to waste time filling out personal information in order to register. Provides near-instant access to a specific store.
- **Product sub-system:**
 - **Manage product sub-system:** is used to allow admin to manage product in the database and can be described as an organizational function that involves the process of conceiving, planning, developing, testing, launching, and delivering products in the market. It also is in charge of retiring a product due to low customer satisfaction or increasing costs.
 - **Search product sub-system:** used to allow customer and admin for product and it is the online search facility that helps shopper find products on our website.
 - **Buy Product sub-system:** is used to allow the customer to buy the product and allow purchase products to acquire by exchange or concession.
- **Add to cart sub-system:** is used to allow customer to add products to the cart and allows customers to choose items to purchase without actually completing the payment and again allows customers to save items to their cart so they can continue shopping, then complete the checkout process later on in the process.
- **Review sub-system:** is used to allow customer to give review on the products help and convert new customers.
- **Fill feedback form sub-system:** a way to collect opinions about our system. The goal is to gain a better understanding of the overall customer experience so we can identify areas for improvement o our system.

3.5.2 Component diagram

A component is an independent, executable unit that provides other components with services or uses the services of other components.[Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel, 2015]

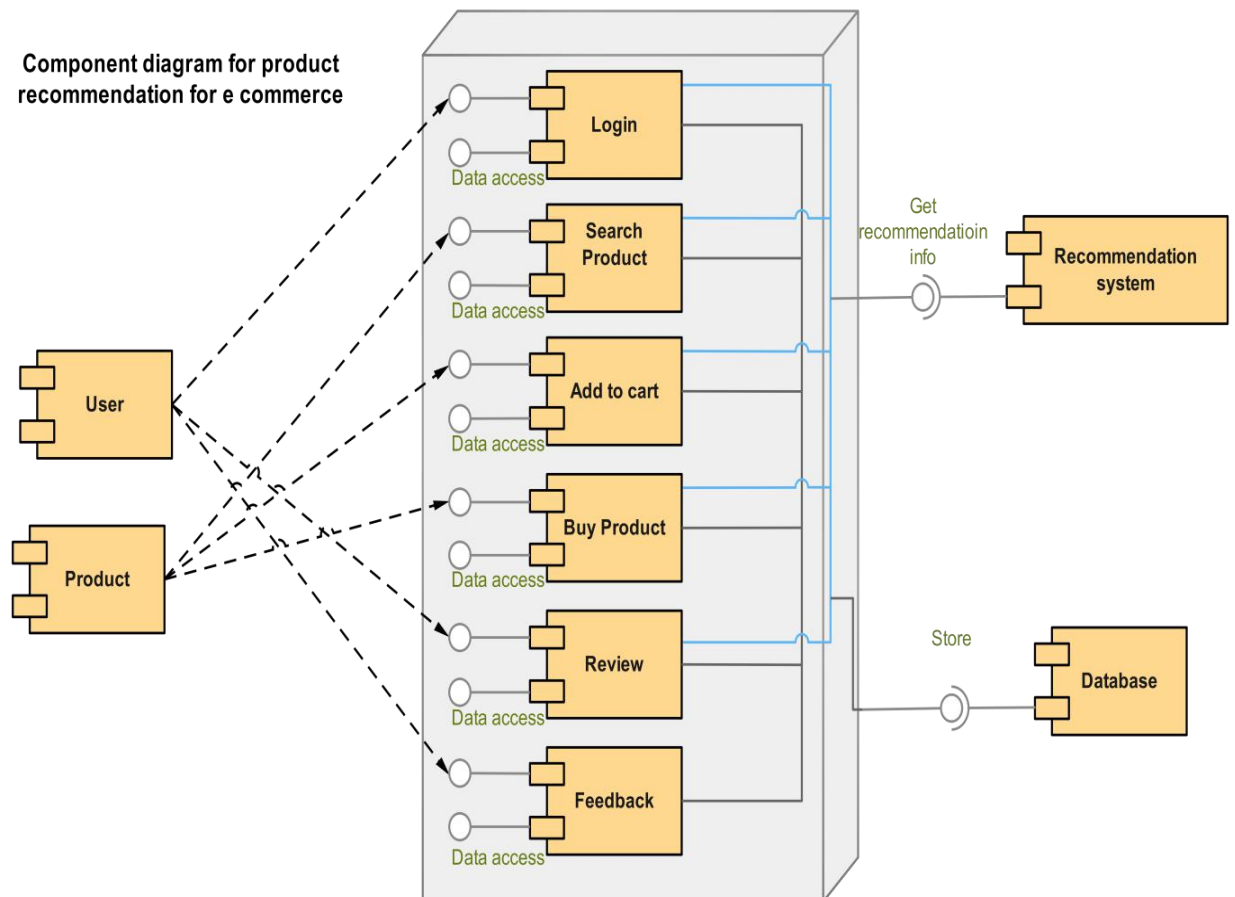


Figure 36. Component diagram

3.5.3 Deployment diagram

A UML deployment diagram can be a useful way to show how your containers or components are deployed. Often such a diagram is better presented as an informal boxes and lines diagram, but the option is there.[Simon Brown, 2012]

The hardware topology used and the runtime system assigned can be represented by the deployment diagram. The hardware encompasses processing units in the form of nodes as well as communication relationships between the nodes.[Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel, 2015]

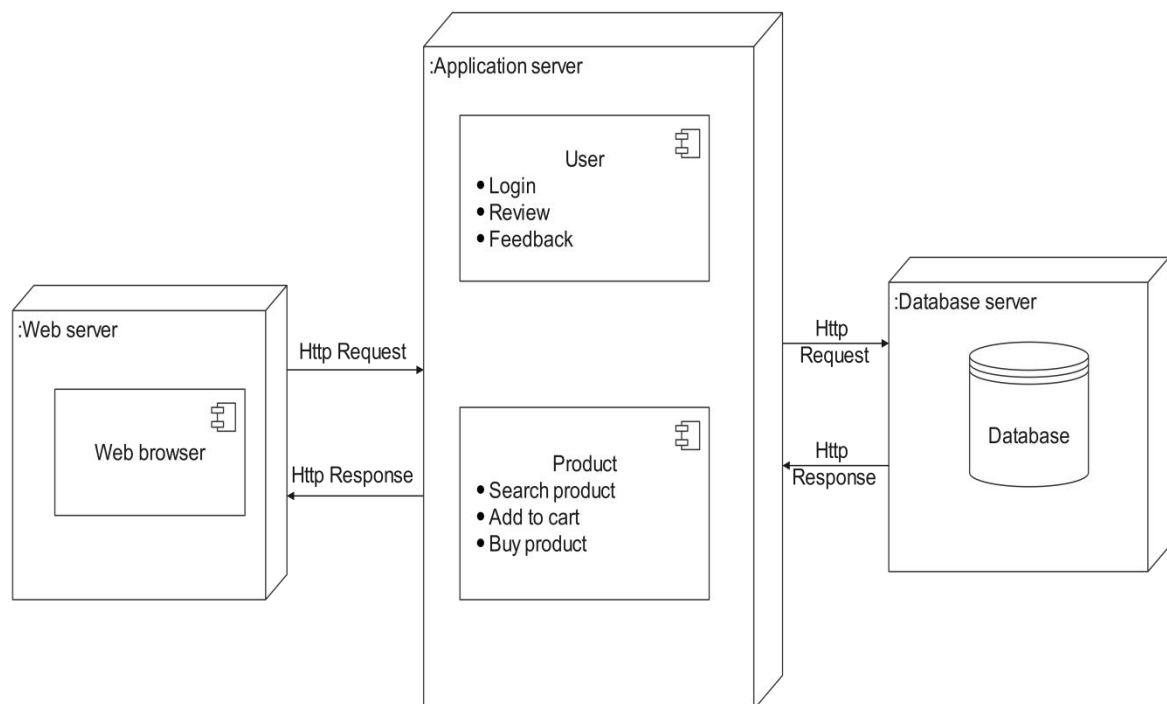


Figure 37. deployment diagram

3.5.4 Persistent data management

Persistent data management: describes the persistent data stored by the system and the data management infrastructure required for it. This section typically includes the description of data schemes, the selection of a database, and the description of the encapsulation of the database.[Bruegge, Bernd Dutoit, Allen H. Dutoit, 2014]

3.5.5 Access control and security

Access control is an authority which is an essential component of data security that commends who's allowed to access a resource. Access control retain confidential information such as customer data, personally identifiable information, and intellectual property from falling into the wrong hands.

The system's authorization described below:

Actor Operations	Admin	Customer	Database
Login	✓	✓	✓
Search Product	✓	✓	✓
Manage Product	✓	X	✓
Buy Product	X	✓	✓
Add to cart	X	✓	✓
Review	X	✓	✓
Feedback	X	✓	✓

Table 10. User-system interaction matrix.

3.5.6 Global software control

Global software control describes how the global software control is implemented. In particular, this section should describe how requests are initiated and how subsystems synchronize. This section should list and address synchronization and concurrency issues.[Bruegge,_Bernd_Dutoit, Allen H. Dutoit, 2014]

When multiple users are directed to the same item, the item will reach its capacity. In this case capacity is a key for recommendation systems. This will in turn lead to deteriorated user experience, such as long waiting times or out of stock items. This motivates the need for a recommendation system that respects the items' capacities.

3.5.7 Boundary conditions

Boundary conditions: describes the start-up, shutdown, and error behavior of the system. (If new use cases are discovered for system administration, these should be included in the requirements analysis document, not in this section.). [Bruegge,_Bernd_Dutoit, Allen H. Dutoit, 2014]

Startup

The database server shall be the first to initialize the data registered this should communicate with the application server once both servers communicate with each other to authorize accessing of the application server then, the system start services, all services required to provide the functionality to the customer after logged in, the system runs the necessary services.

Shutdown

After authorization of accessing application server have the ability to shut down when it's done. The process is similar with the previous one (startup). when the application server complete it's task then the database server will shut down after that the system will shut the functionality that intend to provide.

Error

The application server and the database server should have a connection, means that the system must communicate with the database in order to avoid mismatch information.

Both servers have a service which the application server provides functionalities for the customer and the database server have a data needed for the functionalities. In this process, error may be founded when the servers fail or disconnection between the servers, to avoid this kind of error using exception handling is one solution the other one is when there is no access of one server, These errors generally will be fixed by the application server of the system.

Reference

1. Bruegge, Bernd Dutoit, Allen H. Dutoit,"Object oriented software engineering", 2014, Pearson.
2. "Business and Rule Examples",2022.
3. David Chong, "Deep Dive into Netflix's Recommender System", 2021.
4. Ian Sommerville , "Software engineering", 2011, Pearson.
5. Ijirae, "International Journal of Innovative Research in Advanced Engineering", 2015, AM Publication india.
6. Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel,"UML @ Classroom", 2015, Springer International Publishing Switzerland.
7. Simon Brown, "Software Architecture for Developers", 2012, Leanpub.