

**THE UNIVERSITY OF DA NANG
DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY**



GRADUATION THESIS

**MAJOR: INFORMATION TECHNOLOGY FACULTY
SPECIALTY: SOFTWARE ENGINEERING**

PROJECT TITLE:

THE WEB SYSTEM SUPPORTING FOR PET PRODUCTS

**Instructor : HUYNH HUU HUNG, Ph.D
Student : NGUYEN HUU QUOC CUONG
Student ID : 102150216
Class : 15TCLC1**

Da Nang, December 2020

**THE UNIVERSITY OF DA NANG
DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY**



GRADUATION PROJECT THESIS

MAJOR: INFORMATION TECHNOLOGY FACULTY

SPECIALTY: SOFTWARE ENGINEERING

PROJECT NAME:

THE WEB SYSTEM SUPPORTING FOR PET PRODUCTS

Supervisor : HUYNH HUU HUNG, Ph.D
Student : NGUYEN HUU QUOC CUONG
Student ID : 102150216
Class : 15TCLC1

Da Nang, December 2020

This image shows a full page of a document template designed for handwritten notes or essays. It features approximately 28 evenly spaced, horizontal black dotted lines across the entire width of the page. The background is plain white, providing a clear space for writing. There are no margins, headers, footers, or other markings present on the page.

This image shows a full page of a document template designed for writing. It features a series of evenly spaced, horizontal black dotted lines across the entire width of the page. The background is plain white, providing a clear space for text entry. There are no margins, headers, or footers visible on this page.

SUMMARY

Topic title: The web system supporting for pet products

Student name: Nguyen Huu Quoc Cuong

Student ID: 102150216

Class: 15TCLC1

Summary:

Nowadays, along with the growth of the economy and the development of society, dogs and cats have become human friends. Besides, a place to share valuable knowledge about the well-being of the pets, and a service that provides quality food at reasonable prices is extremely needed right now. Providing food and utensils for animals has become essential.

But you have no experience buying these items. We build this app so that you can purchase relevant, trusted items through a rated product or a recommended product from the system.

PHIẾU DUYỆT ĐỒ ÁN TỐT NGHIỆP

I. Phần dành cho Sinh viên

1. Họ và tên: Nguyễn Hữu Quốc Cường 2. Mã Sinh viên: 102150216
3. Lớp: 15TCLC1
4. Tên đề tài: The web system supporting for pet products
5. Số điện thoại: 0792260250 6. E-mail: quoccuong.ab62@gmail.com
7. Họ và tên GVHD: Ts. Huỳnh Hữu Hưng

II. Phần dành cho Hội đồng

STT	Nội dung đánh giá	Kết luận
1.	<i>Trình bày báo cáo theo đúng mẫu qui định của Khoa</i>	
2.	<i>Không có sự sao chép nội dung báo cáo và chương trình đã có</i>	
3.	<i>Biên dịch mã nguồn và chạy được chương trình</i>	
4.	<i>Có kịch bản thực hiện với dữ liệu thử nghiệm</i>	
5.	<i>Kết quả thực hiện chương trình đúng theo báo cáo</i>	
6.	<i>Có sự đóng góp, phát triển của tác giả trong đồ án</i>	

Kết luận: ☐ Đạt yêu cầu ☐ Phải sửa chữa lại. ☐ Không đạt yêu cầu

Đà Nẵng, ngày 11 tháng 12 năm 2020

Chủ tịch Hội đồng

(Ký và ghi họ tên)

Cán bộ duyệt kiểm tra

(Ký và ghi họ tên)

SUPERVISOR'S COMMENTS

GRADUATION PROJECT REQUIREMENTS

Student Name: Nguyen Huu Quoc Cuong

Student ID: 102150216

Class: 15TCLC1 Faculty: Information Technology

Major: Software Engineer

1. *Topic title:* The web system supporting for pet products
2. *Project topic:* ☐ has signed intellectual property agreement for final result
3. *Initial data:*
Data is generated by myself and searched on the internet.
4. *Content of the explanations and calculations:*
 - Pet Supplies, Pet Food, and Pet Products
 - Manage products
 - Manage users
 - Manage orders
 - Review products
 - Recommend products related
5. *Drawings, charts (specify the types and sizes of drawings):*
 - Use Case diagram
 - Activity diagram
 - Sequence diagram
 - Class diagram
6. *Supervisor:* Huynh Huu Hung ,Ph.D
7. *Date of assignment:*
8. *Date of completion:*

Da Nang, / 12 / 2020

Head of Division

Supervisor

PREFACE

I would first like to thank my supervisor Huynh Huu Hung, Ph.D for your continuous support, supervision, motivation, and guidance throughout the tenure of my project in spite of your hectic schedule. You remained a driving spirit in my project and your experience gave me the understanding in handling research projects as well as helping me to clarify the abstruse concepts, requiring knowledge and perception, handling critical situations and in understanding the objective of my work.

I also want to thank our families and friends, who gave me the strength and confidence during my time of learning and during the implementation phase of this project. They have given a lot of love and encouragement for me which helped pass over the difficulties and fatigues.

Without your generous help, my senior year would not have been successful.

Sincerely,

Nguyen Huu Quoc Cuong

GUARANTEE

We guarantee:

- 1. The contents of this senior project are performed by myself following the guidance of supervisor Huynh Huu Hung Ph.D.*
- 2. All references used in this senior project thesis are quoted with the author's name, project name, time and location to publish clearly and faithfully.*
- 3. All invalid copies, educated statute violation or cheating will be borne the full responsibility by myself.*

Students,

Nguyen Huu Quoc Cuong

TABLE OF CONTENT

Instructor's comments	
Reviewer's comments	
Summary	
Graduation project requirements	
Preface	i
Guarantee	ii
Table of content	iii
List of tables and figures	v
List of symbols and abbreviations	vi
	Page
TECHNOLOGY	8
1.1. NodeJS	8
1.1.1. Introduction	8
1.1.2. Node JS architecture	9
1.1.3. Features	10
1.1.4. Express Framework	11
1.2. React	11
1.2.1. Introduction	11
1.2.2. Features of React	11
1.3. Redux	13
1.3.1. Introduction	13
1.3.2. Redux Data Flow	13
1.4. MongoDB	14
1.4.1 Introduction	14
1.4.2. Mongoose	15
1.5 Recommendation System	15
1.5.1 Introduction	15
1.5.2 Recommend System	17
ANALYSIS AND DESIGN	18
2.1. Main features	18
2.2. Recommendation system	18
2.3. Use-case diagram	19

2.3.1 Overall diagram	19
2.3.2 Overview of guest functions	20
2.3.3 Overview of members functions	21
2.4. Activity diagrams	22
2.4.1. Login	22
2.4.2 Register	23
2.2.4 Add product	26
2.5 Sequence diagram	27
2.5.1 Purchase function	27
2.5.2 Login	28
2.6. Class diagram	29
IMPLEMENTATION RESULT	30
3.1. Login	30
3.2 Sign up form	30
3.3 Product Page	31
3.4 Recommend product	34
3.5 Profile Page	34
3.6 Admin Page	35
CONCLUSION AND FUTURE WORK	38
1.Achieve Results	38
2.Futureworks	38
REFERENCES	39

LIST OF FIGURES, TABLE

1. List of figures

Figure 1.1 Express framework	11
Figure 1.2 Redux data flow	12
Figure 1.3 Object Mapping between Node and MongoDB	12
Figure 2.1 . The overall use case of the guest	18
Figure 2.2 Cart management use case	20
Figure 2.3 The overall use case of the registered users	19
Figure 2.4 The overall use case of the registered users	22
Figure 2.5 Admin management use case	23
Figure 2.6 Login activity diagram	24
Figure 2.7 Register activity diagram	24
Figure 2.8 Purchase activity diagram	25
Figure 2.9 Add product activity diagram	26
Figure 2.10 Sequence diagram for purchase products	27
Figure 2.11 Sequence diagram for login	28
Figure 3.1 Login form	32
Figure 3.2 Sign up form	32
Figure 3.3 Product screen	34
Figure 3.4 Product detail	36
Figure 3.5 Cart	36
Figure 3.6 Shipping form	37
Figure 3.7 Order form	37
Figure 3.8 Similar Product	38
Figure 3.9 Manage shippers	38
Figure 3.10 Manage customers	38

2. List of tables

Table 3.1 Login test case	11
Table 3.2 Create delivery request test case	12
Table 3.3 Delivery test case	18
Table 3.4 Manage type of delivery test case	20

LIST OF ABBREVIATIONS

No.	Items	Description
1	Html	Hyper Text Markup Language
2	JS	JavaScript
3	API	Application Programming Interface
4	OOP	Object Oriented Programming
5	MVC	Model View Controller
6	IDE	Integrated Development Environment
7	REST	Representational State Transfer
8	HTTP	HyperText Transfer Protocol
9	JSON	JavaScript Object Notation
10	ORM	Object Relational Mapping
11	V8	Chrome V8 engine

INTRODUCTION

1. Project overview

1.1. Context and purpose

Previously, dogs and cats were only raised for the purpose of guarding houses and catching rats, but today they are considered to be human friends and cared for as a family member.

Providing food and utensils for animals has become essential. But you have no experience buying these items. We build this app so that you can purchase relevant, trusted items through a rated product or a recommended product from the system.

Therefore, the Pet Support Website is designed to :

1. Provide necessary products for pets
2. Share the needed knowledge to take care of pets
3. Support a platform for people to share knowledge and experiences.

The goal of topic:

- The system helps users to find information about products for pets and necessary pet care knowledge quickly and accurately.
- Website is designed to be flexible and easy for users.

1.2. Project scope

- Sales system simulation and product suggestion system

2. Structure of the thesis

Chapter 1: TECHNOLOGIES – This chapter introduces all knowledge theories and technologies used in this project.

Chapter 2: ANALYSIS AND DESIGN – This chapter covers the main features, software requirement specifications and database design of the project.

Chapter 3: IMPLEMENT AND TEST – This chapter shows an implementation of this project, including pictures and a brief explanation for each main function.

CONCLUSION – The concluding section of the project simultaneously emphasizes the problem solved, as well as presenting issues still unresolved and provides recommendations and suggestions.

REFERENCES – Presentation of information related to the project: source code, etc.

Chapter 1: TECHNOLOGY

Below are theories of technologies I did use in developing my project.

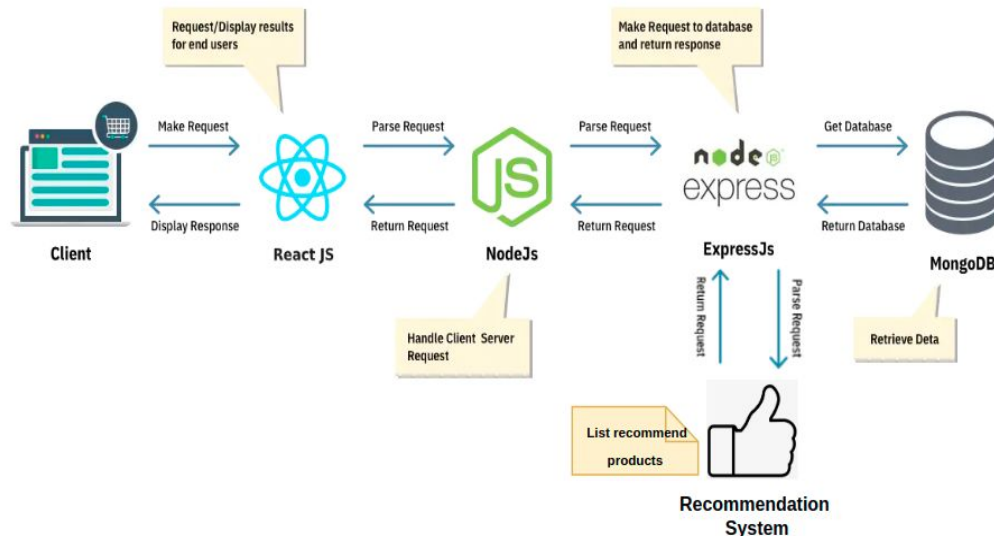


Figure 1. 1: Overview technology diagram on project

1.1. NodeJS

1.1.1. Introduction

Node.js can be defined as a dynamic, cross-platform and open-source JavaScript framework or runtime environment that is built on the Google Chrome JavaScript V8 engine [1]. Node.js, developed by Ryan Dahl in 2009, was initially implemented as a client-side scripting language. Nowadays, it is used to execute JavaScript code and scripts that run server-side to create dynamic web pages. The latest version of Node.js is 10.10.0.

Node.js provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

A Node.js app is run in a single process, without creating a new thread for every request [1]. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

When Node.js needs to perform an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency.

1.1.2. Node JS architecture

a) The Node runtime

Node's runtime is another term for the set executable programs that actually run Node applications, and is a combination of (middle of the stack, see Figure 1):

- *Node API*: The Node API is a set of built-in modules provided by Node.js out of the box for you to build applications [1]. Many of these modules, like the File System (fs) API, sit atop lower-level programs (the Node Core) that communicate with the underlying OS.
- *The Node core*: a set of JavaScript modules that implement the Node API [1]. (Apparently some of the modules depend on libuv and other C++ code but that's an implementation detail).
 - JavaScript engine: Chrome's V8 Engine: A fast JavaScript-to-machine code compiler to load, optimize, and run your JavaScript code
 - The event loop: implemented using an event-driven, non-blocking I/O library called libuv to make it lightweight and efficient (and scalable)

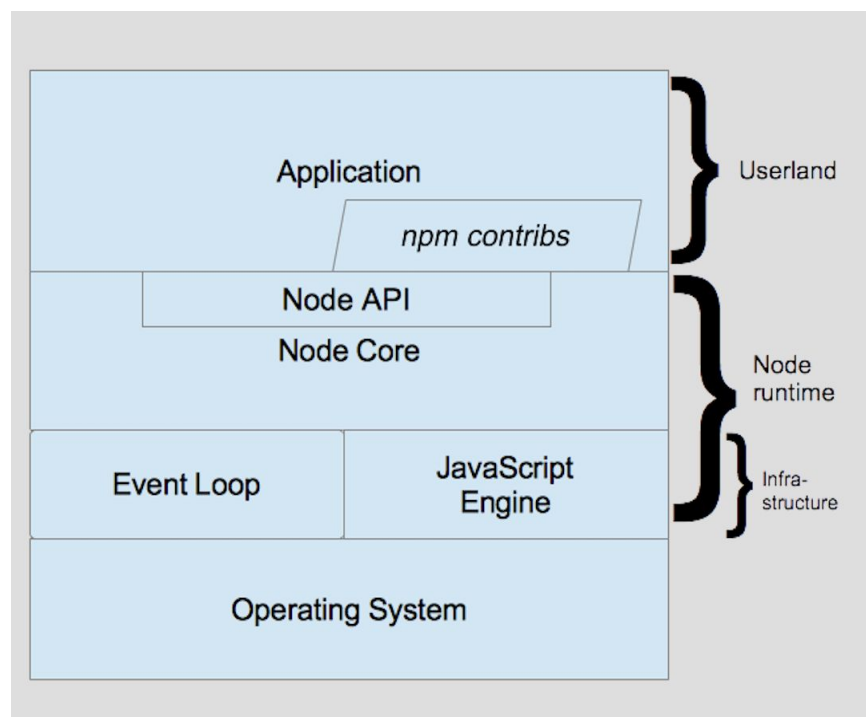


Figure 1. 2 : The Node.js architecture stack

b) Infrastructure

The Node runtime's infrastructure is comprised of two major components:

- JavaScript engine: The JavaScript engine used by Node is Chrome's V8 engine, which runs all of the JavaScript code.
- Event loop: The event loop consists of various phases where callbacks are invoked:
 - Timers phase: `setInterval()` and `setTimeout()` expired timer callbacks are run
 - Poll phase: The OS is polled to see if any I/O operations are complete, and, if so, those callbacks are run
 - Check phase: `setImmediate()` callbacks are run

1.1.3. Features

a) Modularity

Major advantage of Node JS Platform is that it's modularity [1]. Each and every functionality is divided and implemented as a separate module or package.

b) Non-blocking or Asynchronous IO

Asynchronous event-driven IO helps concurrent request handling – This is probably the biggest selling point of Node.js [1]. This feature basically means that if a request is received by Node for some Input/Output operation, it will execute the operation in the background and continue with processing other requests.

c) Single Threaded but Highly Scalable

Node.js uses a single threaded model with event looping [1]. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

d) Better Socket API

Node JS Platform provides a very good Socket Module API to develop Real-time, Multi-User Chat and Multi-Player Gaming Applications very easily. It supports Unix Socket programming like pipe.

1.1.4. Express Framework

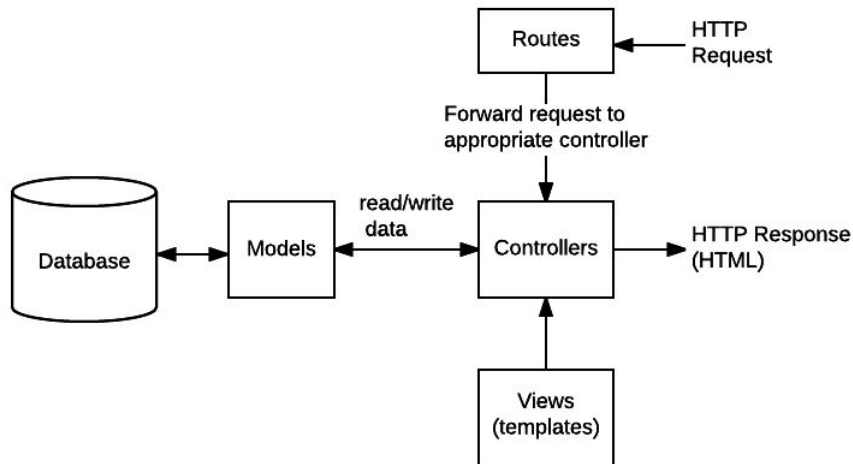


Figure 1.3 Express framework structure

- Express framework [5] is a small, but flexible framework built based on NodeJs. It provides powerful features for web or mobile development.
- There are many packages supported by many developers all over the world.
- Provide more features to increase performance.
- Used as a core function in almost every popular framework.

1.2. React

1.2.1. Introduction

React is a JavaScript library that aims to simplify development of visual interfaces.

Developed at Facebook and released to the world in 2013, it drives some of the most widely used code in the world, powering Facebook and Instagram among many, many other software companies.

Its primary goal is to make it easy to reason about an interface and its state at any point in time, by dividing the UI into a collection of components.

React is used to build single-page web applications, among with many other libraries and frameworks that were available before React came into life.

1.2.2. Features of React

a) Virtual DOM

Virtual DOM (Document Object Model) is the key feature that enables React JS to build scalable and fast applications. It contains a memory reconciliation algorithm

that helps React to create a representation of a web page in virtual memory. So Virtual DOM is the representation of the original DOM.

Whenever changes occur in the web application, the entire user interface is re-rendered in the Virtual DOM representation. Developers can check the difference between new DOM and earlier DOM representation. Later on, the original DOM will update only those components that have actually changed and not all components of virtual DOM. Thus, it helps in developing the app faster with the wastage of memory.

b) JSX

JSX stands for JavaScript XML. It is a syntax extension that describes how the web or mobile app UI should look like. The markup syntax closely resembles HTML. It is a combination of JavaScript and XML.

With JSX, developers can build blocks of React UI, and react components easily by making the syntax of HTML. It is identical to the HTML the programmers will inject in the web page. Thus, JSX is the best feature offered by React JS as it allows web developers to go for an easy way out.

c) One Way Data Binding

One way data binding does not permit developers to edit any properties of the React JS component directly. With the help of a callback function, they can modify the property. This process is known as one way data binding.

React JS follows one-way data binding or unidirectional data flow that gives better control throughout the application. Additional features are required to manage the data flow that comes from another direction.

As React components are immutable, the data within them cannot be changed. Using Flux, developers can keep data unidirectional and makes the application more flexible that improves its efficiency.

d) React Native

It is a custom render for React that uses native components rather the web components just like react as building blocks. To get a profound understanding of React Native, one should need to know about the basic React components. Such as stats, conditions, components, props, and JSX.

Once you have clarity about React, you need to explore the information specific to React Native to learn about the native elements. Apart from making the React code compatible with iOS and Android platforms, React Native provides access to the features these platforms offer.

e) Declarative

As we discussed before, React JS creates an intuitive and interactive user interface for mobile and web applications. React effectively renders and updates the right elements when the data changes. The library creates simple views for every state in web applications. Declarative sight creates the code that is simpler to debug and more readable.

f) Component

React JS framework is based on components. It contains numerous components that create a user interface of web applications and each component is designed with a specific logic. The logic is written in JavaScript instead of templates, so developers can easily pass the data throughout the application and keep the state out of the DOM.

React JS components play a key role in defining interactions and visuals in the applications.

1.3. Redux

1.3.1. Introduction

Redux is a predictable state container for JavaScript apps. It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as [live code editing combined with a time traveling debugger](#).

You can use Redux together with [React](#), or with any other view library. It is tiny (2kB, including dependencies), but has a large ecosystem of addons available.

1.3.2. Redux Data Flow

Redux follows the unidirectional data flow. It means that your application data will follow in one-way binding data flow. As the application grows & becomes complex, it is hard to reproduce issues and add new features if you have no control over the state of your application.

Redux reduces the complexity of the code, by enforcing the restriction on how and when state update can happen. This way, managing updated states is easy. We already know about the restrictions as the three principles of Redux(as shown in figure 1.1). Following diagram will help you understand Redux data flow better

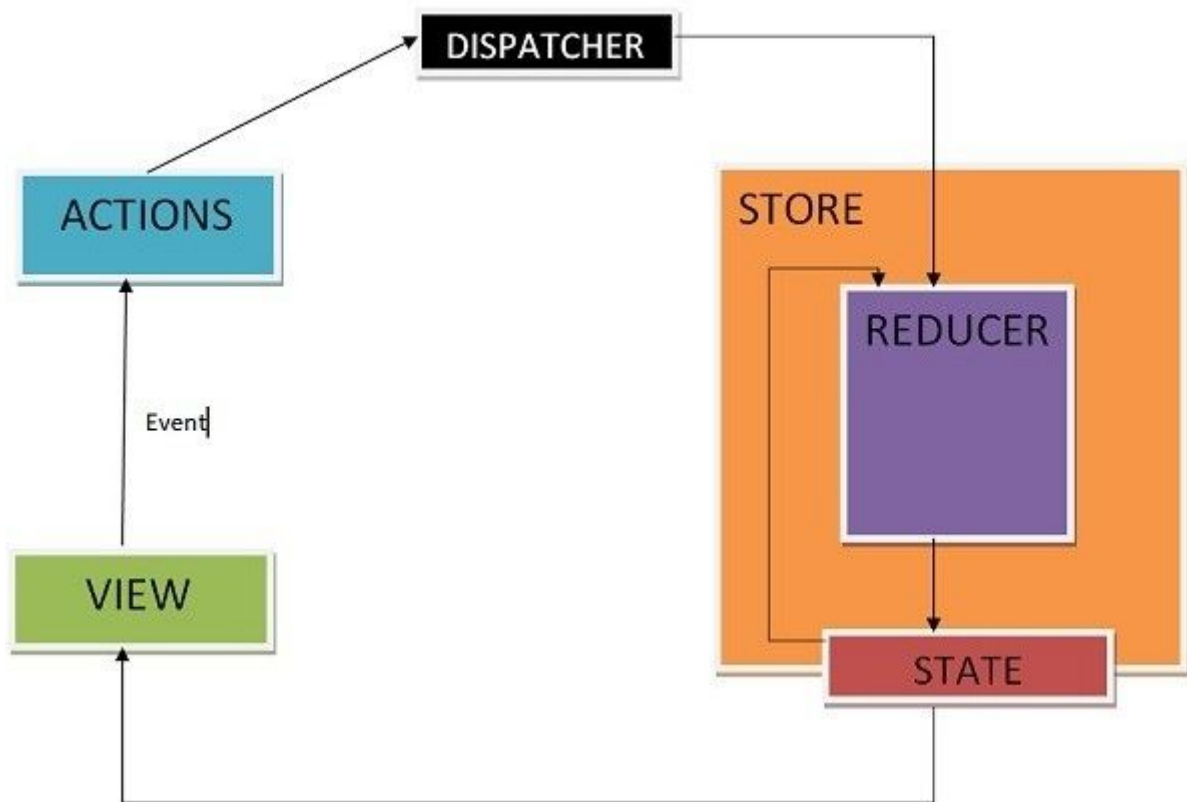


Figure 1.4 Redux flow

- An action is dispatched when a user interacts with the application.
- The root reducer function is called with the current state and the dispatched action. The root reducer may divide the task among smaller reducer functions, which ultimately returns a newstate.
- The store notifies the view by executing their callback functions.
- The view can retrieve updated state and re-render again.

1.4. MongoDB

1.4.1 Introduction

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability [4]. MongoDB works on concept of collection and document.

Database: Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collection: Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not

enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

Document: A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

1.4.2. Mongoose

a) Define

Mongoose[5] is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

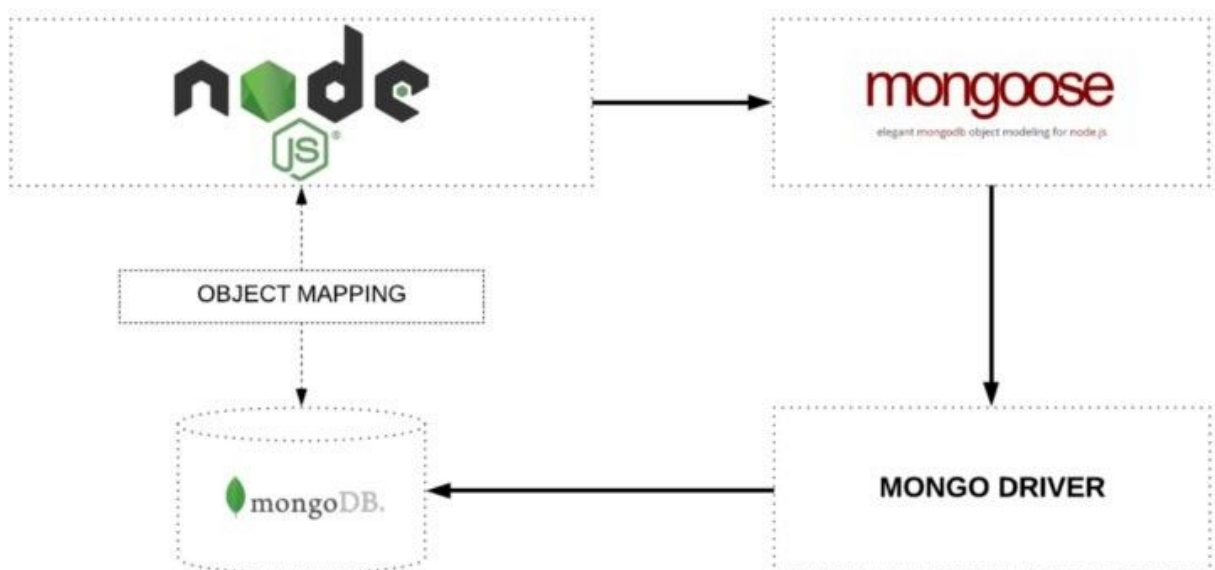


Figure 1.5 Object Mapping between Node and MongoDB

MongoDB is a schema-less NoSQL document database. It means you can store JSON documents in it, and the structure of these documents can vary as it is not enforced like SQL databases. This is one of the advantages of using NoSQL as it speeds up application development and reduces the complexity of deployments.[5].

1.5 Recommendation System

1.5.1 Introduction

- A **recommender system**, or a **recommendation system** (sometimes replacing 'system' with a synonym such as platform or engine), is a subclass of

information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are primarily used in commercial applications.

- Recommender systems are utilized in a variety of areas and are most commonly recognized as playlist generators for video and music services, product recommenders for online stores, or content recommenders for social media platforms and open web content recommenders. These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. There are also popular recommender systems for specific topics like restaurants and **online dating**. Recommender systems have also been developed to explore research articles and experts, collaborators, and financial services.

Recommendation paradigms

The distinction between approaches is more academic than practical, but it's important to understand their differences.

Broadly speaking, recommender systems are of 4 types:

1. **Collaborative filtering** is perhaps the most well-known approach to recommendation, to the point that it's sometimes seen as synonymous with the field. The main idea is that you're given a matrix of preferences by users for items, and these are used to predict missing preferences and recommend items with high predictions. All you need to get started is user and item IDs and a notion of preference by users for items (ratings, views, etc.).

2. **Content-based filtering** algorithms are given user preferences for items and recommend similar items based on a domain-specific notion of item content. This approach also extends naturally to cases where item metadata is available (e.g., movie stars, book authors, and music genres).

3. **Social and demographic** recommenders suggest items that are liked by friends, friends of friends, and demographically-similar people. Such recommenders don't need any preferences by the user to whom recommendations are made, making them very powerful.

4. **Contextual** recommendation algorithms recommend items that match the user's current context. This allows them to be more flexible and adaptive to current user needs than methods that ignore context (essentially giving the same weight to all of the user's history). Hence, contextual algorithms are more likely to elicit a response than approaches that are based only on historical data.

1.5.2 Content-based filtering

Recommender systems are active information filtering systems that personalize the information coming to a user based on his interests, relevance of the information, etc. Recommender systems are used widely for recommending movies, articles, restaurants, places to visit, items to buy, and more.

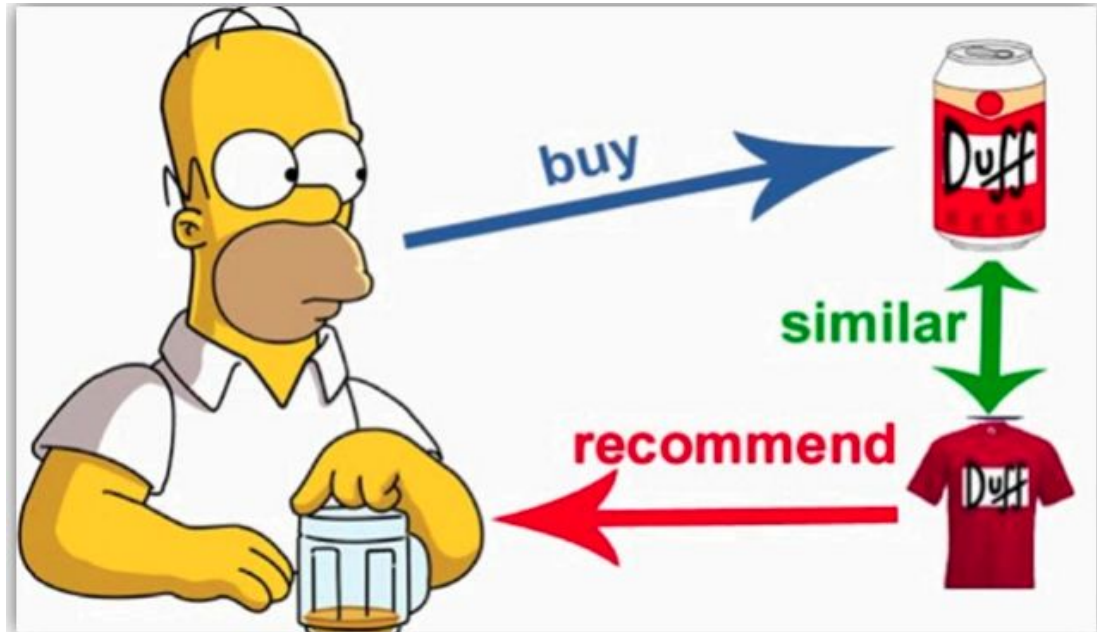


Figure 1.6 Recommend System

How do content-based recommender systems work?

A content-based recommender works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link). Based on that data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on those recommendations, the engine becomes more and more accurate.

A recommender system has to decide between two methods for information delivery when providing the user with recommendations:

- **Exploitation.** The system chooses documents similar to those for which the user has already expressed a preference.
- **Exploration.** The system chooses documents where the user profile does not provide evidence to predict the user's reaction.

Now that we've taken a broad look at what recommender systems are and the different variations, let's work through an implementation of a content-based filtering system

Chapter 2: ANALYSIS AND DESIGN

This chapter will detail the requirements: function, constraints and other necessary factors to provide a complete and comprehensive description of the requirements for the application. There are Use-case diagrams, Activity diagrams, Class diagrams, Sequence diagrams, and Databases. These diagrams will show an overview of the functions that the application can provide. In addition, it defines the architecture, modules, and data for a system to satisfy specified requirements.

2.1. Main features

The website is divided into 2 main parts, the part for users (guests and members) and administrators (admin).

- Main functions for guest user include :
 - Watch product
 - Search product
 - Watch information of product
- Main function for registered user include :
 - Have all function of guest account
 - Personal account management
 - Cart management
 - Product rating & review system
- Main function for admin include :
 - User management
 - Product management
 - Order management

2.2. Recommendation system

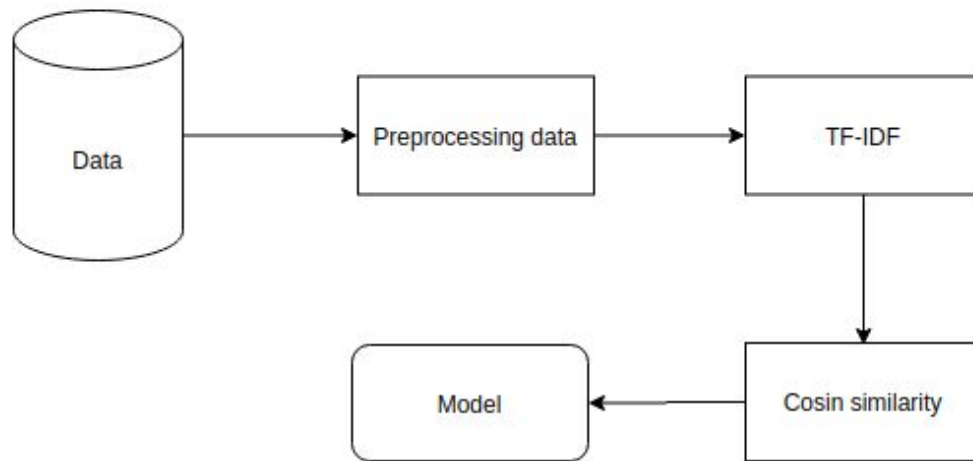


Figure 2.1. The model of recommendation system

Build model recommend system by content-based:

- Step 1: Collect data from a database with 1000 samples of items like products for pets , ... , along with an item-id and a textual description of the item.

- Sample data :

{

"description": "DescriptionThe Bond & Co. apparel and accessories allow you to create the pawfect wardrobe for your furry fashionista. Crafted of the finest materials and details, this collection makes sure they're always dressed to impress with their adorable personality right on their sleeve. From pops of color - and charm - these styles are tailored to keep their tail wagging all year long.- White & Plaid Dog Dress from Bond & Co.- Tailored to keep their tail wagging in the finest materials and details- Breathable stretch jersey t-shirt top and pleated plaid flannel skirt- Foil-printed with \"PRIN-SASS\" lettering- Elasticized waist for a contoured fit- Stitched leash portal for quick access to their collar D-ring- Pulls over for easy dressing",

"id": "5fce25da6b38863f52ced730"

}

- Step2: Preprocessing data:
 - Remove items that don't have a description.

- Remove stop words.
- Step 3: Vectorize description by using TF-IDF
 - The TF*IDF algorithm is used to weigh a keyword in any document and assign the importance to that keyword based on the number of times it appears in the document. Put simply, the higher the TF*IDF score (weight), the rarer and more important the term, and vice versa.
 - Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF*IDF weight of that term.
 - The TF (term frequency) of a word is the number of times it appears in a document. When you know it, you're able to see if you're using a term too often or too infrequently. $TF(t) = (\text{Number of times term } t \text{ appears in a document}) / (\text{Total number of terms in the document})$.
 - The IDF (inverse document frequency) of a word is the measure of how significant that term is in the whole corpus.

$IDF(t) = \log_e(\text{Total number of documents} / \text{Number of documents with term } t \text{ in it})$.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF
Term x within document y

$tf_{x,y}$ = frequency of x in y
 df_x = number of documents containing x
 N = total number of documents

TF-IDF calculation

Figure 2.2. TF-IDF calculation

- Step 4: Calculating Cosine Similarity:
 - Here we've calculated the cosine similarity of each item with every other item in the dataset, and then arranged them according to their similarity with item i , and stored the values in results

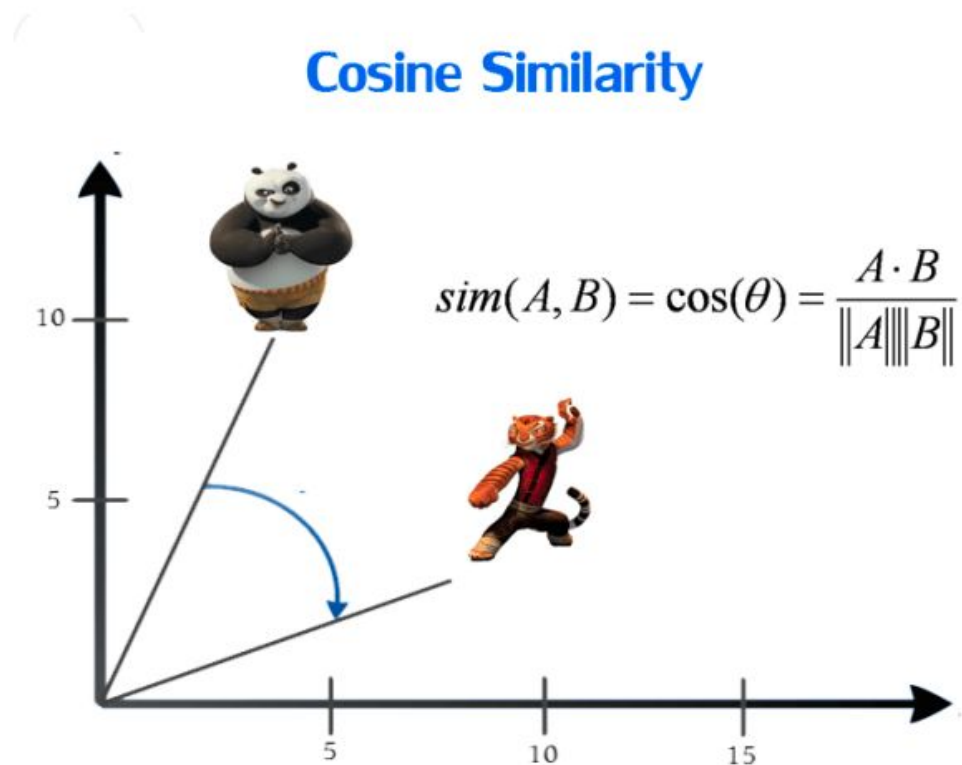


Figure 2.2. Cosine similarity

- Step 5: Save model:
 - Each item we just save 100 other items with highest cosine similarity

```
Recommended: Relax fit organic ctn jeans-reg (score:0.8908101955877065)
Recommended: Relax fit organic ctn jeans-long (score:0.8866113828050025)
Recommended: Reg fit organic ctn jeans-short (score:0.507668259865595)
Recommended: Reg fit organic ctn jeans-long (score:0.48801052800273903)
Recommended: Reg fit organic ctn jeans-reg (score:0.48488884889129785)
Recommended: Custodian pants (score:0.1925730494862419)
Recommended: Shop pants (score:0.18030173682681883)
Recommended: Shop pants (score:0.1733375276479681)
Recommended: Custodian pants (score:0.1710311820622527)
Recommended: Inga shorts (score:0.17023045978100093)
```

Figure 2.3. Recommendations similar to products

2.3. Use-case diagram

2.3.1 Overall diagram

The web system includes these features:

- Register account
- Login system
- View products
- Search products
- Payment
- Manage products
- Manage user
- Manage order

With these features above, the web system has three actors (guest, member and the admin) interacting with the system. The overall use-case is shown in the figure 2.1.

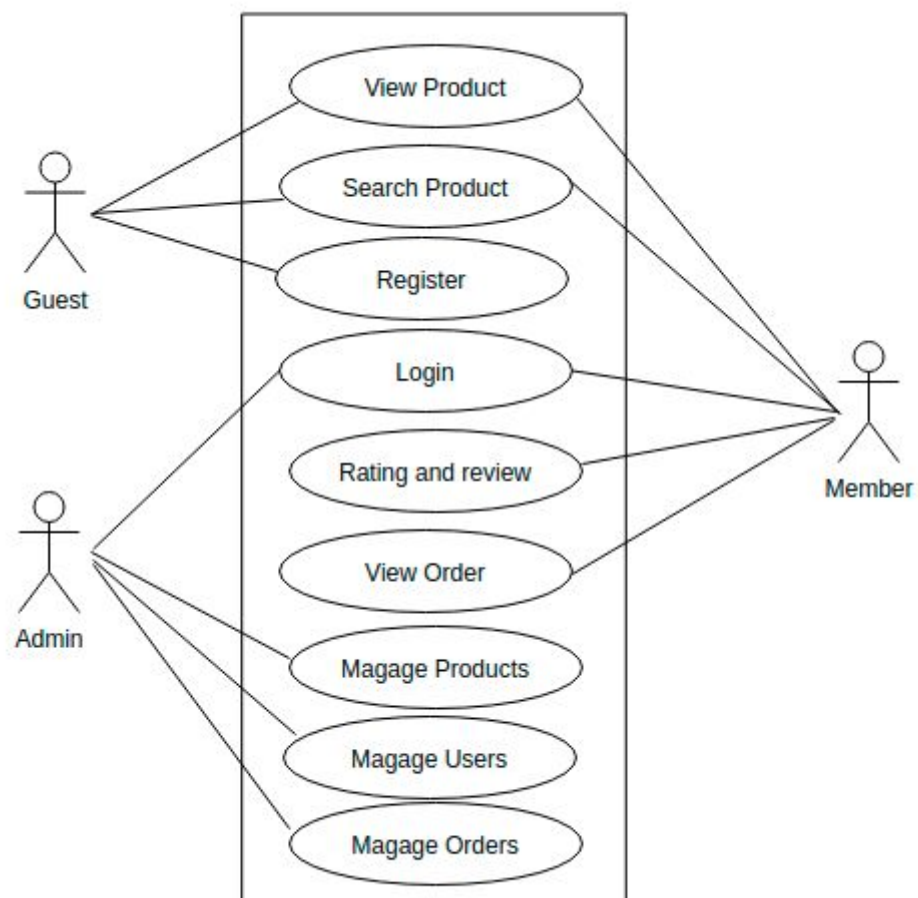


Figure 2.4. The overall use case

2.3.2 Overview of guest functions

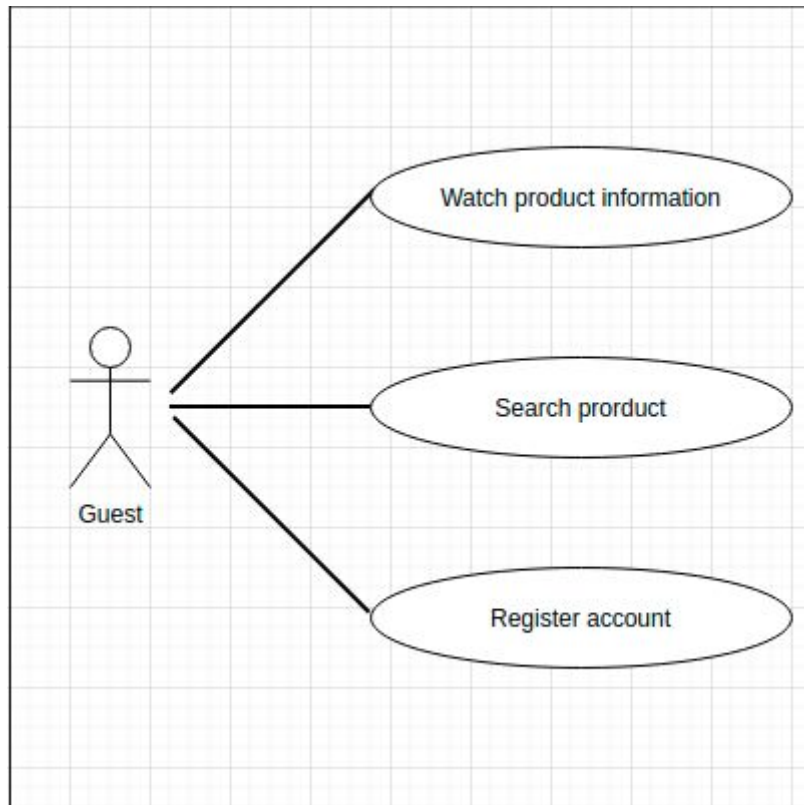


Figure 2.5. The overall use case of the guest

“Actor Guest” has the following features: Watch products, search products, register accounts.

Actor students are shown in Figure 2.2 and Figure 2.3 .

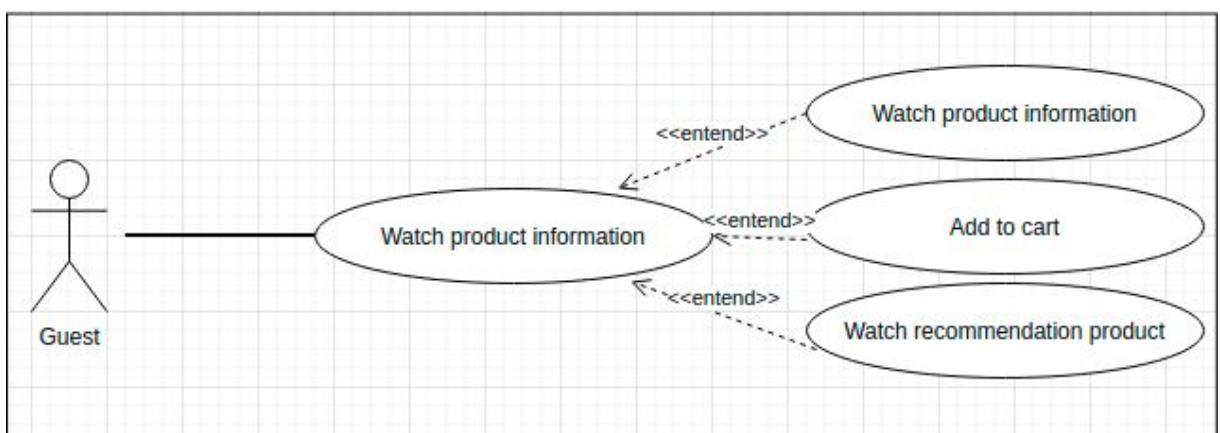


Figure 2.6. Cart management use case

2.3.3 Overview of members functions

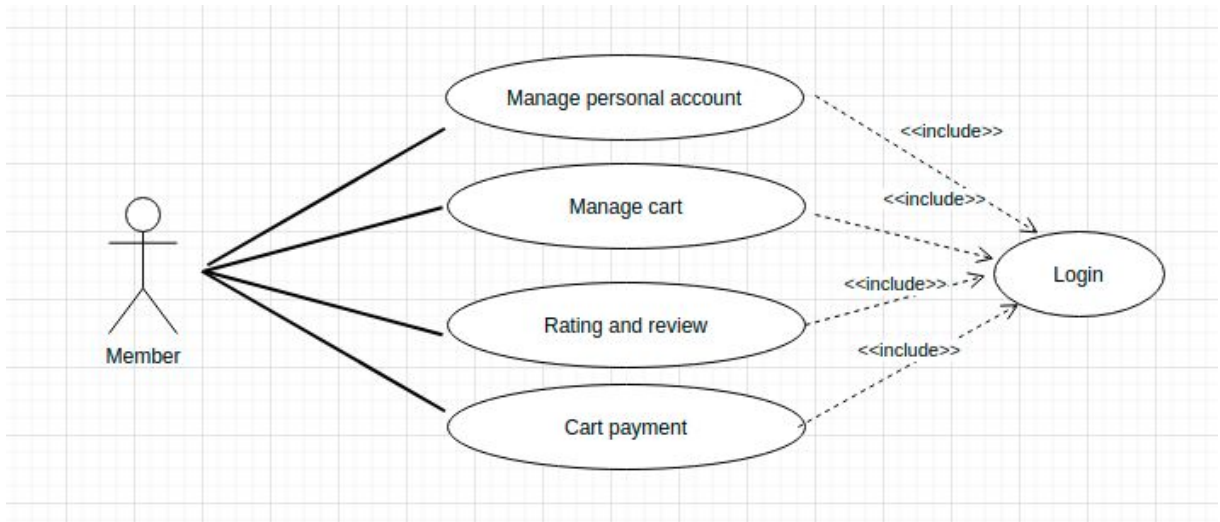


Figure 2.7. The overall use case of the registered users

“Actor Member” has the following features: Login, update profile , manage cart, rating and review product, cart payment .

Actor members are shown in Figure 2.4 and Figure 2.5 .

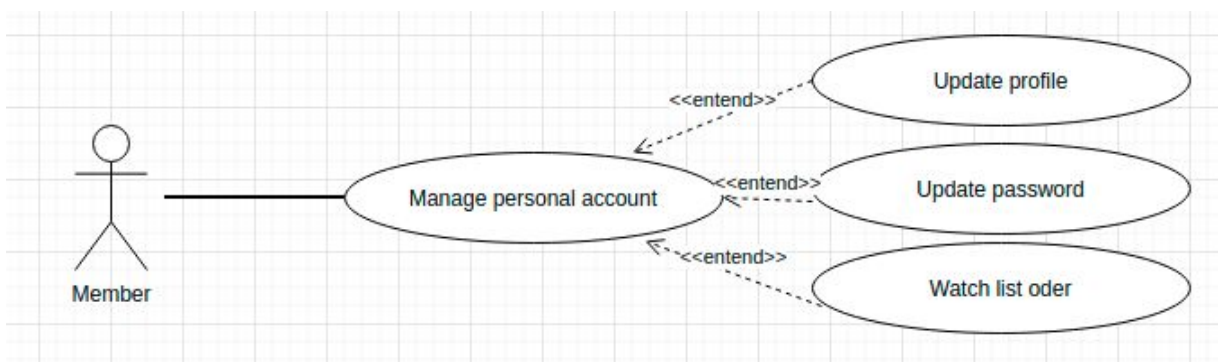


Figure 2.8. The overall use case of the registered users

2.3.4 Overview of admin functions

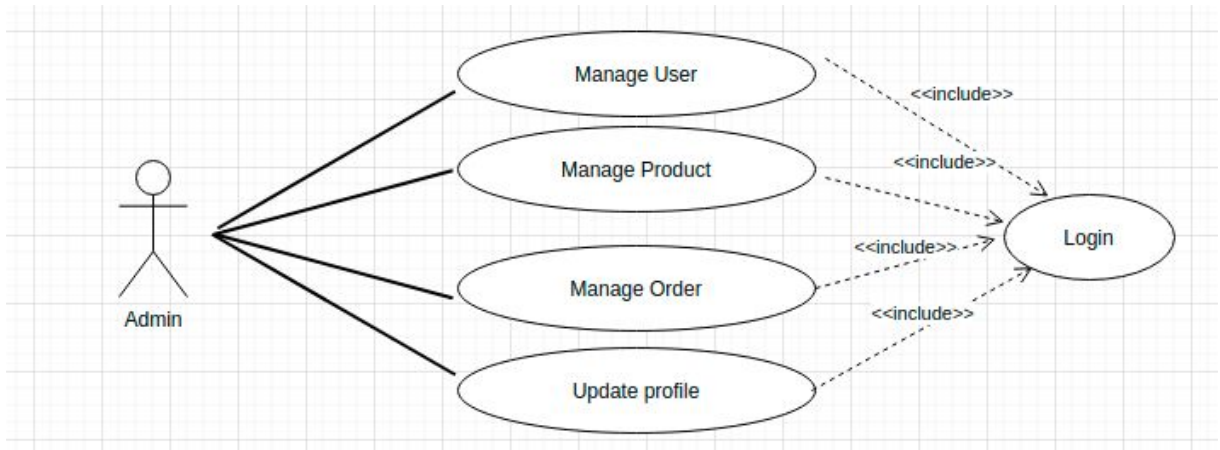


Figure 2.9. Admin management use case

“Actor Admin” has the following features: Login, manage users, manage products, manager orders, update profile .

Actor admin is shown in Figure 2.6.

2.4. Activity diagrams

2.4.1. Login

The diagram shown in figure 2.7 describes the process with user sign in to the system. User has to fill the user's information into the login form. After that, the system calls API to check automatically the user's information. If the user's information is valid then the system saves that information to the Local Storage and navigates to the home page.

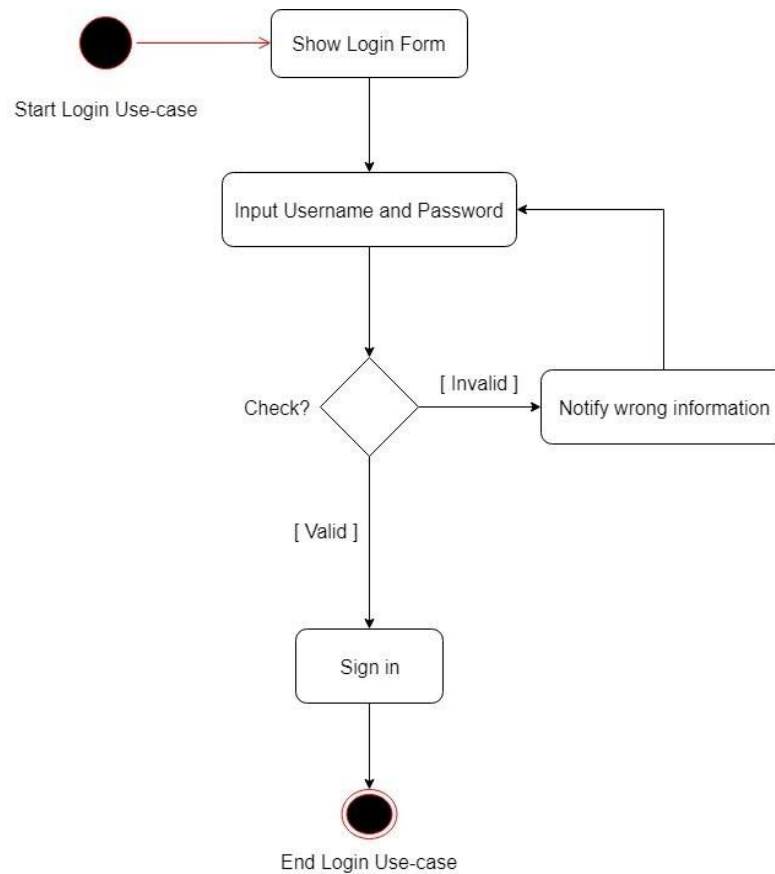


Figure 2.10 Login activity diagram

2.4.2 Register

The diagram shown in figure 2.6 describes the process in which users register a new account. User has to fill user' information into the registration form. After that, the system automatically checks the user's information. If the user's information is valid then the system saves that information to the database and notifies successful message to the user.

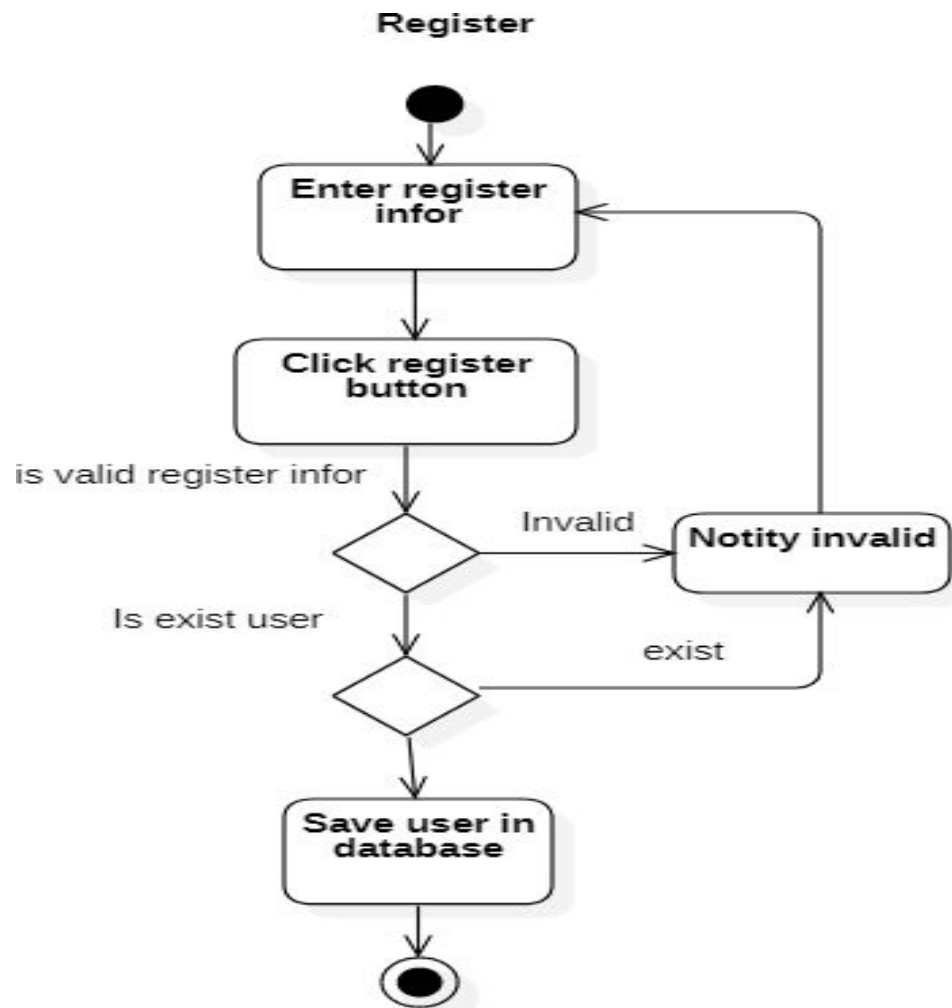


Figure 2.11 Register activity diagram

2.4.3. Purchase merchandises

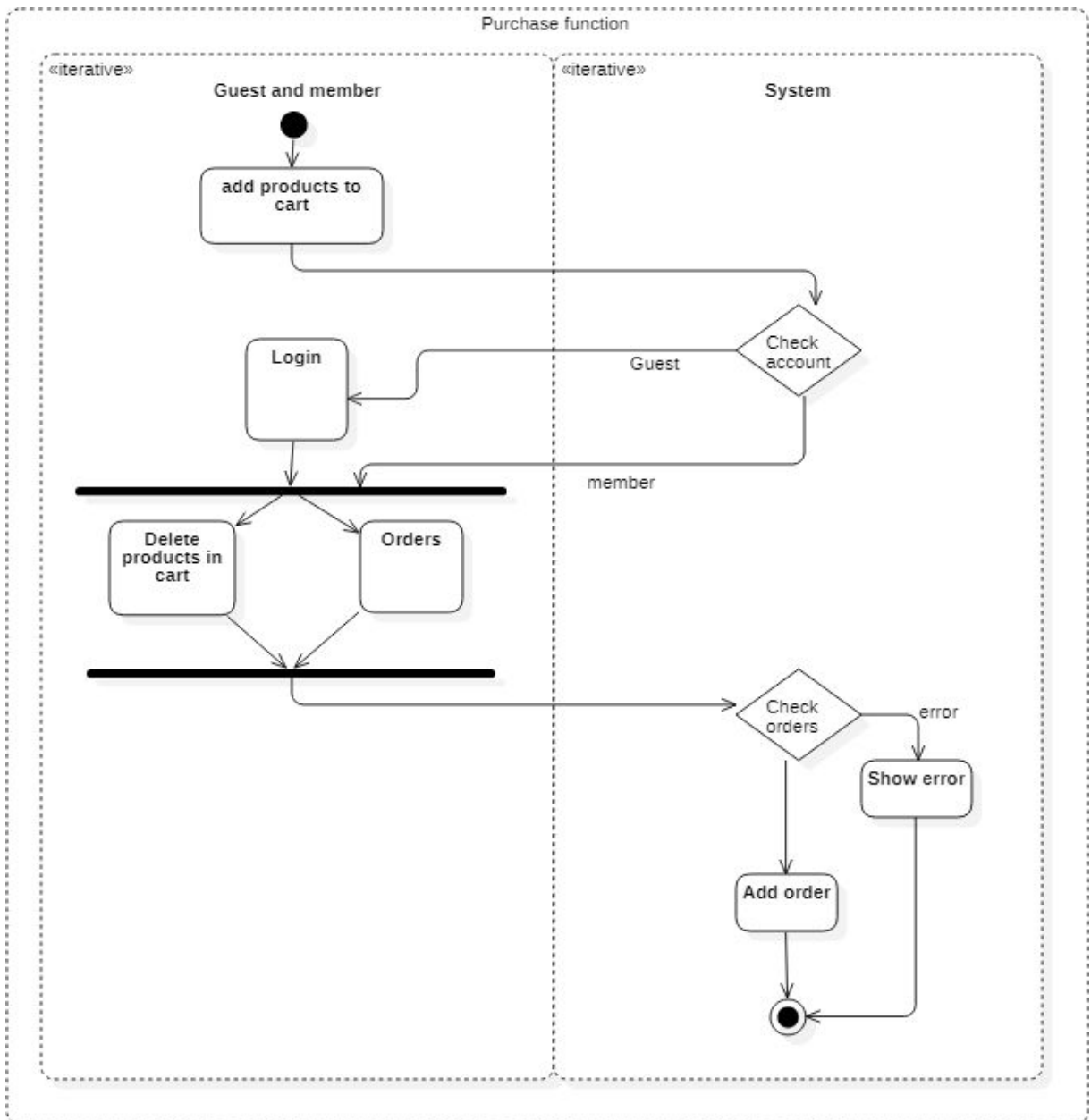


Figure 2.12 Purchase activity diagram

Activity diagram shown in figure 2.12 provides description of purchase functions.

2.4.4 Add product

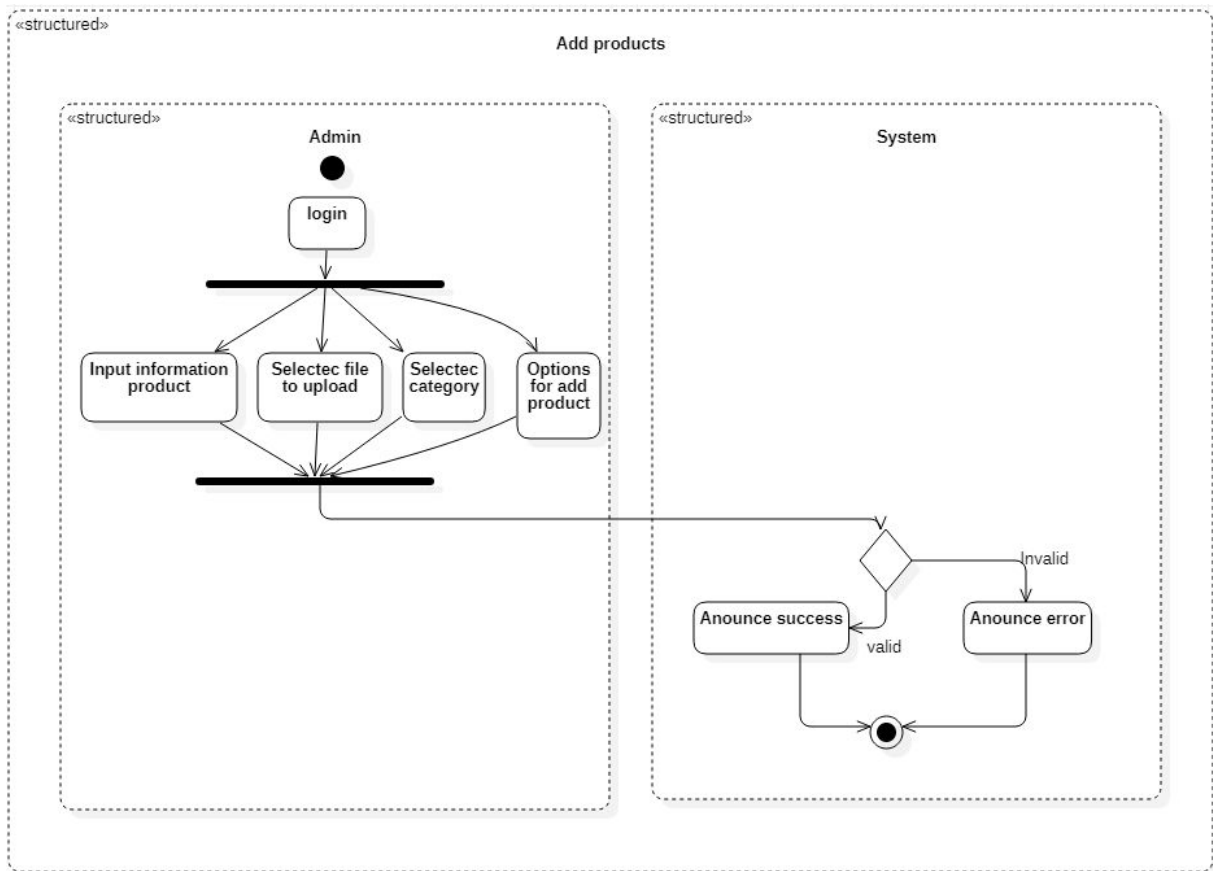


Figure 2.13 Add product activity diagram

Activity diagram shown in figure 2.13 provides description of add product functions.

2.5 Sequence diagram

2.5.1 Purchase function

Sequence diagram shown in figure 2.15 provides description of purchasing product functions.

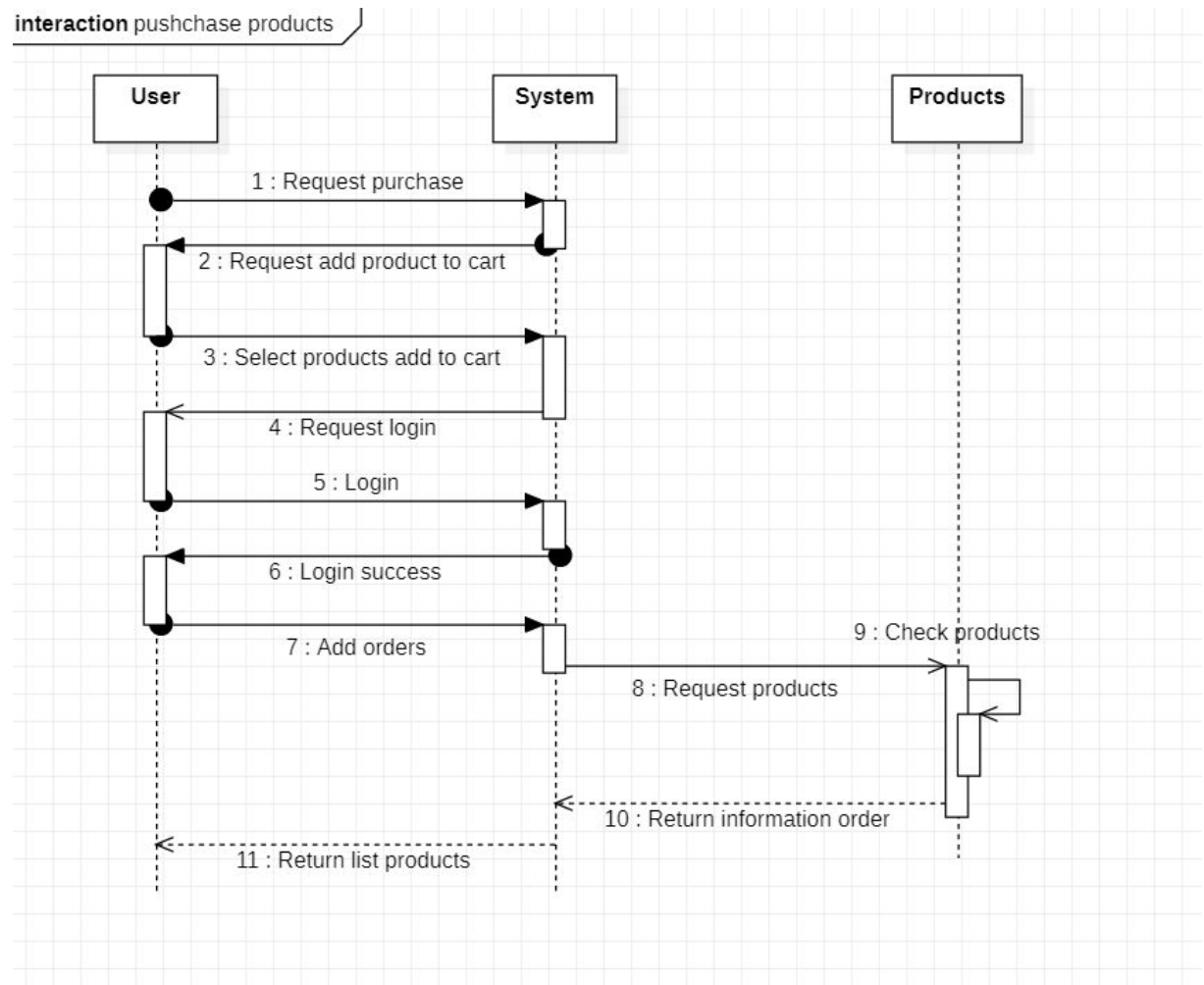


Figure 2.14 Sequence diagram for purchase products

2.5.2 Login

Sequence diagram shown in figure 2.15 provides a description of login function.

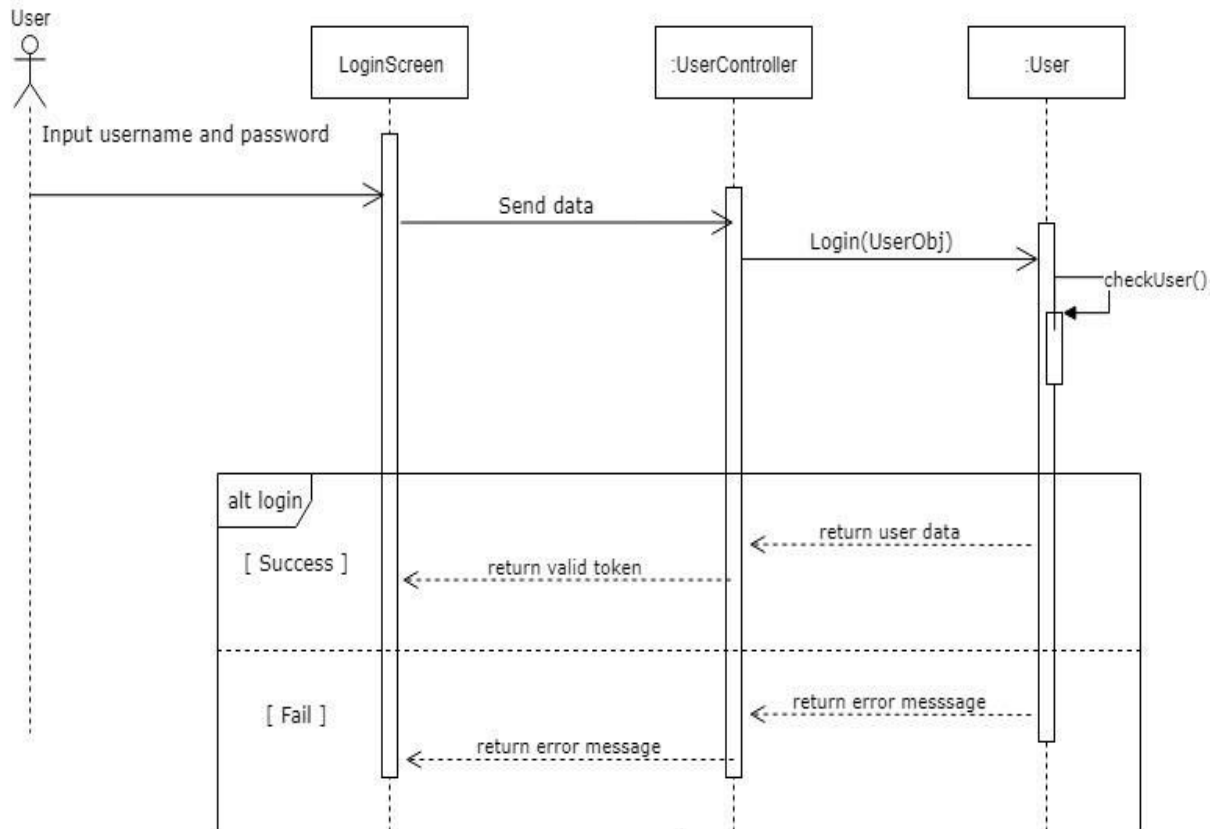


Figure 2.15 Sequence diagram for login

2.6. Class diagram

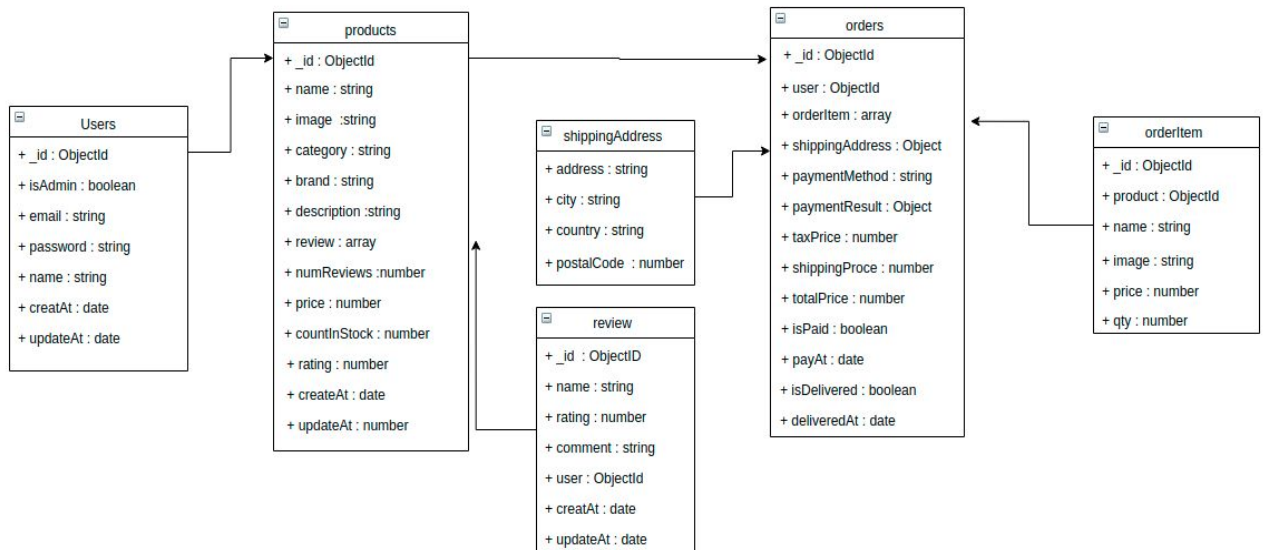


Figure 2.16 Class diagram

The system including classes and their relationships is shown in Figure 2.18

Chapter 3: IMPLEMENTATION RESULT

3.1. Login

Normally, users who do not need to log in to the system will still be able to search for product information.

However, to use functions such as managing carts, payment courses, users need to log into the system. The interface of the login function is shown in the figure 3.1.

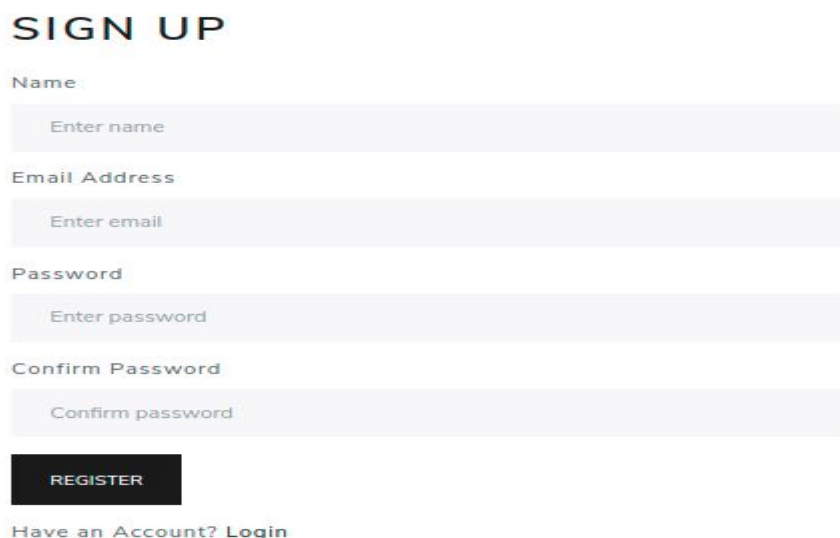


The image shows a 'SIGN IN' form. At the top, the text 'SIGN IN' is displayed in a large, bold, black font. Below it, there are two input fields. The first is labeled 'Email Address' and contains the text 'quoccuong.ab62@gmail.com'. The second is labeled 'Password' and contains a series of dots, indicating a masked password. Below the password field is a black button with the text 'SIGN IN' in white. At the bottom of the form, there is a link that says 'New Customer? Register'.

Figure 3.1 Login form

3.2. Sign up form

Account registration page (as shown in Figure 3.2).



The image shows a 'SIGN UP' form. At the top, the text 'SIGN UP' is displayed in a large, bold, black font. Below it, there are four input fields. The first is labeled 'Name' and contains the text 'Enter name'. The second is labeled 'Email Address' and contains the text 'Enter email'. The third is labeled 'Password' and contains the text 'Enter password'. The fourth is labeled 'Confirm Password' and contains the text 'Confirm password'. Below the password fields is a black button with the text 'REGISTER' in white. At the bottom of the form, there is a link that says 'Have an Account? Login'.

Figure 3.2 Sign up form

3.3 Product Page

At the initial product screen will display all products by category as slides (as shown in Figure 3.6).

PRODUCTS









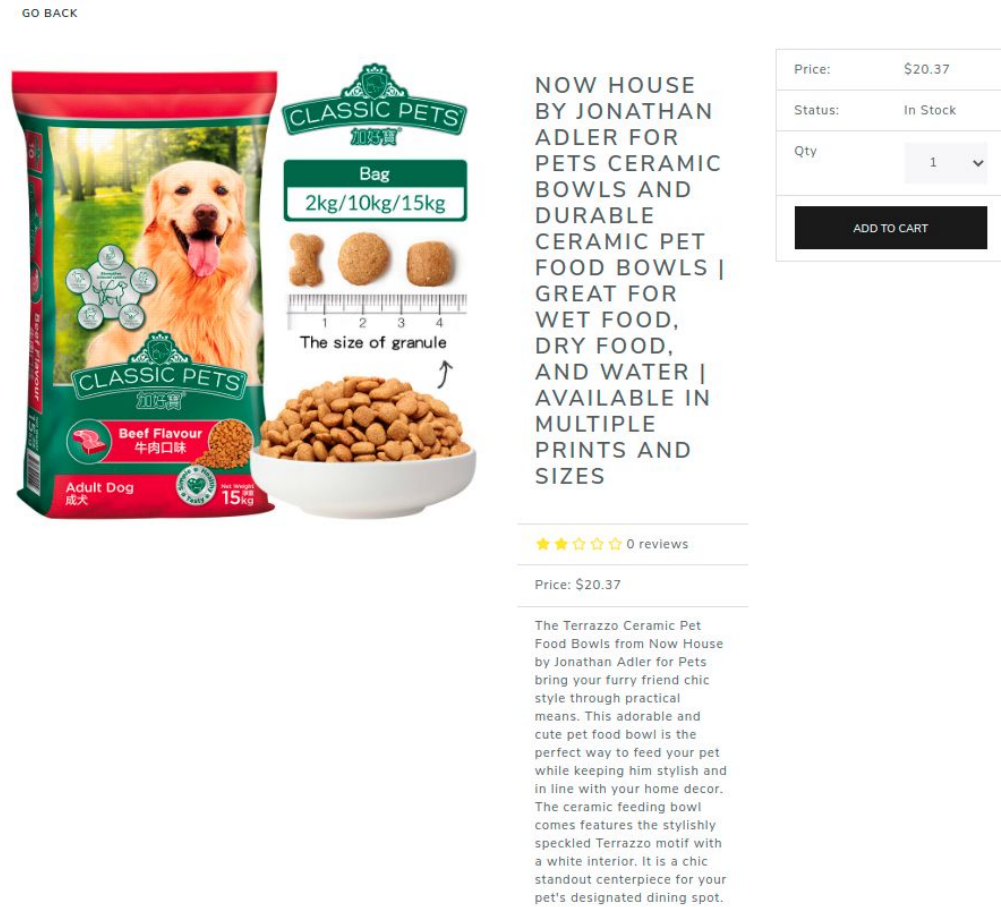
 <p>Blue Buffalo Life Protection Formula Natural Adult Dry Dog Food</p> <p>★★★★★ 1 reviews</p> <p>\$27.48</p>	 <p>Cesar Gourmet Wet Dog Food Variety Packs – 24 Trays</p> <p>★★★★★ 1 reviews</p> <p>\$10</p>	 <p>Cesar Gourmet Wet Dog Food Variety Packs – 36 Trays</p> <p>★★★★☆ 0 reviews</p> <p>\$11</p>	 <p>Now House by Jonathan Adler for Pets Ceramic Bowls and Durable Ceramic Pet Food Bowls Great for Wet Food, Dry Food, and Water Available in Multiple Prints and Sizes</p> <p>★★★★☆ 0 reviews</p> <p>\$20.37</p>
 <p>Original Adult Dry Dog Food Chicken Brown Rice</p> <p>★★★★☆ 0 reviews</p> <p>\$48.99</p>	 <p>Purina Pro Plan Adult Sensitive Skin & Stomach Salmon & Rice Formula Dry Dog Food</p> <p>★★★★☆ 0 reviews</p> <p>\$10</p>	 <p>American Journey Salmon & Sweet Potato Recipe Grain-Free Dry Dog Food</p> <p>★★★★★ 0 reviews</p> <p>\$48.99</p>	 <p>Roll over image to zoom in PetHonesty Allergy Relief Immunity Strength & Digestive Health Soft Chews Dog Supplement</p> <p>★★★★★ 1 reviews</p> <p>\$27.48</p>

Figure 3.3 Product screen

Pet Care

Displays details of product information such as price, origin, quantity, product details (as shown in Figure 3.4).

GO BACK



CLASSIC PETS
2kg/10kg/15kg
Beef Flavour
Adult Dog
15kg

NOW HOUSE
BY JONATHAN
ADLER FOR
PETS CERAMIC
BOWLS AND
DURABLE
CERAMIC PET
FOOD BOWLS |
GREAT FOR
WET FOOD,
DRY FOOD,
AND WATER |
AVAILABLE IN
MULTIPLE
PRINTS AND
SIZES

Price: \$20.37
Status: In Stock
Qty: 1
ADD TO CART

★★★★★ 0 reviews

Price: \$20.37

The Terrazzo Ceramic Pet Food Bowls from Now House by Jonathan Adler for Pets bring your furry friend chic style through practical means. This adorable and cute pet food bowl is the perfect way to feed your pet while keeping him stylish and in line with your home decor. The ceramic feeding bowl comes features the stylishly speckled Terrazzo motif with a white interior. It is a chic standout centerpiece for your pet's designated dining spot.

Figure 3.4 Product detail

At the shopping cart, users can delete or increase the number of products that they have put in the cart (as shown in Figure 3.5).

SHOPPING CART



Now House by
Jonathan Adler for
Pets Ceramic Bowls
and Durable Ceramic
Pet Food Bowls |
Great for Wet Food,
Dry Food, and Water
| Available in
Multiple Prints and
Sizes

\$20.37

1

SUBTOTAL (1) ITEMS
\$20.37
PROCEED TO CHECKOUT

Figure 3.5 Cart manager

Pet Care

At the shipping page, fill the information for shipping address (as shown in Figure 3.6).

[Sign In](#) [Shipping](#) [Payment](#) [Place Order](#)

SHIPPING

Address

City

Postal Code

Country

CONTINUE

Figure 3.6 Shipping form

Show total price and information ordered


SHIPPING

Address: Da Nang, Viet Nam, Da Nang 15000, Viet Nam

PAYMENT METHOD

Method: PayPal

ORDER ITEMS

	Now House by Jonathan Adler for Pets Ceramic Bowls and Durable Ceramic Pet Food Bowls Great for Wet Food, Dry Food, and Water Available in Multiple Prints and Sizes	1 x \$20.37 = \$20.37
---	--	-----------------------

ORDER SUMMARY

Items	\$20.37
Shipping	\$100.00
Tax	\$3.06
Total	\$123.43

PLACE ORDER

Figure 3.7 Order form

3.4 Recommend product

Show list recommend products

SIMILAR PRODUCT

Instinct Raw Boost Small Breed Grain-Free Recipe...
★★★★★ (114)
\$39.99

Instinct Raw Boost Gut Health Grain-Free Recipe...
★★★★★ (2)
\$69.99

Instinct Raw Boost Grain-Free Recipe with Real...
★★★★☆ (8)
\$69.99

Instinct Raw Boost Whole Grain Real Chicken &...
★★★★★ (1)
\$54.99

Instinct Raw Boost Small Breed Grain-Free Recipe...
★★★★★ (31)
\$25.99

Figure 3.8 Similar Product

3.5 Profile Page

In profile users, we can view and edit personal information, ordered list

USER PROFILE

Name

Ngọc Thảo

Email Address

admin@example.com

Password

Enter password

Confirm Password

Confirm password

UPDATE

MY ORDERS

ID	DATE	TOTAL	PAID	DELIVERED	
5fac006ca7d58e2dba94e19b	2020-11-11	105.75	×	×	DETAILS
5fac13a2a7d58e2dba94e19d	2020-11-11	203.49	×	×	DETAILS
5fac1416a7d58e2dba94e19f	2020-11-11	203.49	×	×	DETAILS
5fad66f2a7d58e2dba94e1a7	2020-11-12	310.47	×	×	DETAILS
5fb4c2be5c5d910004d50cec	2020-11-18	156.34	×	×	DETAILS
5fb4d3ee2d1e204e57c2ba63	2020-11-18	156.34	×	×	DETAILS
5fb4e288dd208872e3bef585	2020-11-18	156.34	×	×	DETAILS
5fbfbfdba1159ef1b243fb20d	2020-11-23	156.34	×	×	DETAILS
5fbfbfc8ce925231f8d936e04	2020-11-23	156.34	×	×	DETAILS
5fbfbfcb7d4cc920906ef9a4	2020-11-23	156.34	×	×	DETAILS
5fbbfef1343eb7124570c06e9	2020-11-23	156.34	×	×	DETAILS
5fbbfef5843eb7124570c06eb	2020-11-23	179.76	×	×	DETAILS
5fc0f0a54ffd3e4df2f815db	2020-11-27	563.38	×	×	DETAILS
5fc0f1074ffd3e4df2f815dd	2020-11-27	123.43	×	×	DETAILS
5fc0f1f5b2f3b652e6f61078	2020-11-27	156.34	×	×	DETAILS

Figure 3.9 List Order and Profile

3.6 Admin Page

In the admin page, we can watch information of users, products, list orders .

USERS













ID	NAME	EMAIL	ADMIN	
5fabfc3bc7615b2c01b2c46a	Ngọc Thảo	admin@example.com	✓	 
5fabfc3bc7615b2c01b2c46b	John Doe	john@example.com	✗	 
5fabfc3bc7615b2c01b2c46c	Jane Doe	jane@example.com	✗	 
5fabfcaca7d58e2dba94e195	a	quoccuong.ab62@gmail.com	✗	 
5fb514e63135b70004c0af8c	Khánh	khanhmai0097@gmail.com	✗	 
5fc495aab2f3b652e6f6107a	Thảo Ngọc	a@a.com	✗	 

Figure 3.10 List Users

Products list, admin can create and update products.

PRODUCTS

[+ CREATE PRODUCT](#)

ID	NAME	PRICE	CATEGORY	BRAND	
5fad67dca7d58e2dba94e1aa	Blue Buffalo Life Protection Formula Natural Adult Dry Dog Food	\$27.48	Chicken & Brown Rice Recipe	Blue Buffalo	 
5fad683ba7d58e2dba94e1ab	Cesar Gourmet Wet Dog Food Variety Packs – 24 Trays	\$10	Amazon	Cesar	 
5fad6840a7d58e2dba94e1ac	Cesar Gourmet Wet Dog Food Variety Packs – 36 Trays	\$11	Poultry Variety	Cesar	 
5fb553254aa0a440fb205e3d	Now House by Jonathan Adler for Pets Ceramic Bowls and Durable Ceramic Pet Food Bowls Great for Wet Food, Dry Food, and Water Available in Multiple Prints and Sizes	\$20.37	Dog Food	Dry Food	 
5fb5533c4aa0a440fb205e3e	Original Adult Dry Dog Food Chicken Brown Rice	\$48.99	Blue Buffalo Life Protection Formula was created for the holistic health and well-being of adult dogs. All formulas start with real meat, whole grains, garden veggies and fruit, plus added LifeSource Bits, a precise blend of nutrients that have been enhanced with a Super 7 package of antioxidant-rich ingredients. This Adult Chicken & Brown Rice Recipe features delicious, protein-rich deboned chicken and other natural ingredients for a healthy meal your dog will love.	Dry Food	 

Figure 3.11 List Products

EDIT PRODUCT

Name

Sample name

Price

0

Image

/images/sample.jpg

Choose File

Browse

Brand

Sample brand

Count In Stock

0

Category

Sample category

Description

Sample description

UPDATE

Figure 3.12 Add Products

Pet Care

In the ordered list, the admin watches the details of the order .

ORDERS

ID	USER	DATE	TOTAL	PAID	DELIVERED	
5fabfccfa7d58e2dba94e196	a	2020-11-11	\$203.49	✗	✗	DETAILS
5fac006ca7d58e2dba94e19b	Ngọc Thảo	2020-11-11	\$105.75	✗	✗	DETAILS
5fac13a2a7d58e2dba94e19d	Ngọc Thảo	2020-11-11	\$203.49	✗	✗	DETAILS
5fac1416a7d58e2dba94e19f	Ngọc Thảo	2020-11-11	\$203.49	✗	✗	DETAILS
5fad66f2a7d58e2dba94e1a7	Ngọc Thảo	2020-11-12	\$310.47	✗	✗	DETAILS
5fb4c2be5c5d910004d50cec	Ngọc Thảo	2020-11-18	\$156.34	✗	✗	DETAILS
5fb4d3ee2d1e204e57c2ba63	Ngọc Thảo	2020-11-18	\$156.34	✗	✗	DETAILS
5fb4e288dd208872e3bef585	Ngọc Thảo	2020-11-18	\$156.34	✗	✗	DETAILS
5fbfbfda1159ef1b243fb20d	Ngọc Thảo	2020-11-23	\$156.34	✗	✗	DETAILS
5fbafc8ce925231f8d936e04	Ngọc Thảo	2020-11-23	\$156.34	✗	✗	DETAILS
5fbfbfcb7d4cc920906ef9a4	Ngọc Thảo	2020-11-23	\$156.34	✗	✗	DETAILS
5fbffe1343eb7124570c06e9	Ngọc Thảo	2020-11-23	\$156.34	✗	✗	DETAILS
5fbffe5843eb7124570c06eb	Ngọc Thảo	2020-11-23	\$179.76	✗	✗	DETAILS
5fc0f0a54ffd3e4df2f815db	Ngọc Thảo	2020-11-27	\$563.38	✗	✗	DETAILS
5fc0f1074ffd3e4df2f815dd	Ngọc Thảo	2020-11-27	\$123.43	✗	✗	DETAILS
5fc0f1f5b2f3b652e6f61078	Ngọc Thảo	2020-11-27	\$156.34	✗	✗	DETAILS

Figure 3.13 List ordered

CONCLUSION AND FUTURE WORK

1.Achieve Results

In this project, I have learned about JavaScript language and understood how to work with ReactJS, Nodejs and MongoDB.

I used:

- Mongoose syntax to program asynchronously with NodeJS,
- ReactJS to build the application,
- MongoDB to store databases.

I have also improved all of my skills including: technical, researching and presentation since I started building this project. With them, I have built the graduation project that meets the initial requirements and have these following features:

- The system supports the user to find suitable products for pets,
- The system supports the user to find the needed information to take good care of pets,
- The system also provides users with a place to exchange information and chat with each other.

However, this web system still has following issues:

- The system does not support users who want to exchange pets.
- The number of products and information on the web is still limited.

2.Futureworks

In the future, I will keep on developing more features such as:

- Recommend product smarter
- Update more information and new products to meet the needs of everyone.

REFERENCES

- [1] Website: <https://en.wikipedia.org/wiki/JavaScript>
- [2] Website: <https://en.wikipedia.org/wiki/ECMAScript>
- [3] Website:
<https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/>
- [4] Website: [https://en.wikipedia.org/wiki/V8_\(JavaScript_engine\)](https://en.wikipedia.org/wiki/V8_(JavaScript_engine))
- [5] Website: <https://expressjs.com/>
- [6].<https://reactjs.org/>(Data access: 23/10/2019)
- [7].<https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>

Pet Care
