

**COMPUTER SCIENCE AND ENGINEERING**

***TechShop***

A project submitted in partial fulfillment of

the requirements for the Degree of

Bachelor of Computer Science and Engineering

from

VNUK Institute for Research & Executive Education

by

***Duong Duc Hung***

***15020018***

***Bui Lam Quang Ngoc***

***15020019***

**SUPERVISORS**

***Tran The Vu***

***Dang Thi Phuong Thao***

© Copyright 2019 by

[ DUONG DUC HUNG  
BUI LAM QUANG NGOC ]

ALL RIGHTS RESERVED

DECLARATION

We, Hung and Ngoc declare that this project titled, TechShop and the work presented in it are my own. We confirm that:

* This work was done wholly or mainly while in candidature for a BSc degree at VNUK Institute for Research and Executive Education.
* Where any part of this thesis has previously been submitted for a degree or any other qualification at VNUK Institute for Research and Executive Education or any other institution, this has been clearly stated.
* Where I have consulted the published work of others, this is always clearly attributed.
* Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
* I have acknowledged all main sources of help.
* Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

*[Bui Lam Quang Ngoc, 15020019 and Computer Science and Engineering 15]*

*[Duong Duc Hung, 15020018 and Computer Science and Engineering 15]*

CERTIFICATE

We do hereby declare that the research works embodied in this project entitled “**TechShop**” is the outcome of an original work carried out by Duong Duc Hung and Bui Lam Quang Ngoc under our supervision.

We further certify that the dissertation meets the requirements and the standard for the degree of BSc in Computer Science and Engineering.

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

*Name and designation of Supervisor*

***Dedicated to***

***My Parents and my Family***ACKNOWLEDGEMENT

*The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.*

*We respect and thank Dr. Tran The Vu, for providing me an opportunity to do the project work in VN-UK Institute for Research and Executive Education and giving us all support and guidance which made us complete the project duly. We are extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.*

*We owe our deep gratitude to our project guide Ms. Dang Thi Phuong Thao, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.*

Da Nang, 2019 *Bui Lam Quang Ngoc*

*Duong Duc Hung*

ABSTRACT

*This website is designed to attend to all your needs – from buying computing goods properly, selling products in Da Nang, Viet Nam. Here you found the better items of technology in there. Portal help us to maintain the database of various items and clients information. It not only help us to maintain the customer’s information but here we also setup and develop a recommended system to help users having a good experiences when they are interacting with our website.*

*The site may be focused on selling computing goods, uploading new information which relate to stat-of-the-art technologies. In addition to that clients or users could rate or vote the level of their satisfaction with several items on our website. We will be putting an effort to provide the right choice to people when they are searching and looking for new technological products and beware them from the false advertising.*

*We also develop the system that recommend users which items is the most suitable with them. All of this functions based on how many clicks of category-related products they did, and the rate of their voting with any kind of products. This system is a big challenge with us during the process of developing and researching the way to build it.*

**TABLE OF CONTENTS**

[DECLARATION 3](#_Toc15024181)

[CERTIFICATE 4](#_Toc15024182)

[ABSTRACT 7](#_Toc15024183)

[LIST OF FIGURES 12](#_Toc15024184)

[LIST OF TABLES 14](#_Toc15024185)

[SYMBOLS 15](#_Toc15024186)

[ABBREVIATIONS 16](#_Toc15024187)

[CHAPTER 1: INTRODUCTION 17](#_Toc15024188)

[1.1 Purpose of the system 17](#_Toc15024189)

[1.2 Scope of the system 17](#_Toc15024190)

[1.3 Objectives and success criteria of the project 18](#_Toc15024191)

[1.4 Definitions, acronyms, and abbreviations 19](#_Toc15024192)

[1.5 Project plan 20](#_Toc15024193)

[1.5.1 Gantt Chart 20](#_Toc15024194)

[1.5.2 Project Phase Distribution 23](#_Toc15024195)

[1.5.3 Other Tables 23](#_Toc15024196)

[1.6 Overview 24](#_Toc15024197)

[CHAPTER 2: METHODOLOGY AND THEORETICAL BACKGROUND 25](#_Toc15024198)

[2.1 METHODOLOGY 25](#_Toc15024199)

[2.1.1 Software Development Life Cycle 25](#_Toc15024200)

[2.1.2 Agile Development Method 29](#_Toc15024201)

[2.1.3 Wireframe and prototype 30](#_Toc15024202)

[2.1.4 Source code control 32](#_Toc15024203)

[2.2 THEORICAL BACKGROUND 36](#_Toc15024204)

[2.2.1. Overview of web service technology 36](#_Toc15024205)

[2.2.2. Web service RESTful architechture 38](#_Toc15024206)

[2.2.3. Understanding RESTFul Principles and Constraints 39](#_Toc15024207)

[2.2.3.1 RESTFul Client-Server 39](#_Toc15024208)

[2.2.3.2 Stateless 40](#_Toc15024209)

[2.2.3.3 Cache 40](#_Toc15024210)

[2.2.3.4 Layered System. 40](#_Toc15024211)

[2.2.3.5 Interface/Uniform Contract 41](#_Toc15024212)

[2.2.4. Definitions and types of recommendation system 41](#_Toc15024213)

[2.2.4.1 Popularity based 41](#_Toc15024214)

[2.2.4.2 Classification based 41](#_Toc15024215)

[2.2.4.3 Collaborative filtering 42](#_Toc15024216)

[CHAPTER 3: SYSTEM ANALYSIS AND DESIGN 43](#_Toc15024217)

[3.1 CURRENT SYSTEM 43](#_Toc15024218)

[3.2 PROPOSED SYSTEM 43](#_Toc15024219)

[3.2.1 Overview 43](#_Toc15024220)

[3.2.2 Functional Requirements 43](#_Toc15024221)

[3.2.3 Nonfunctional Requirements 44](#_Toc15024222)

[3.2.3.1 Usability 44](#_Toc15024223)

[3.2.3.2 Reliability 44](#_Toc15024224)

[3.2.3.3 Performance 44](#_Toc15024225)

[3.2.3.4 Supportability 44](#_Toc15024226)

[3.2.3.5 Implementation 44](#_Toc15024227)

[3.2.4 System Models 45](#_Toc15024228)

[3.2.4.1 Scenarios 45](#_Toc15024229)

[3.2.4.2 User case model 45](#_Toc15024230)

[3.2.4.2.1 User case diagram 45](#_Toc15024231)

[3.2.4.2.2 User case table 47](#_Toc15024232)

[3.2.4.3 Object Model 53](#_Toc15024233)

[3.2.4.3.1 Activity Diagram 53](#_Toc15024234)

[3.2.4.3.2 Class Diagram 53](#_Toc15024235)

[3.2.4.4 Dynamic Model 53](#_Toc15024236)

[3.2.4.4.1 Searching Technique 53](#_Toc15024237)

[3.2.4.4.2 Recommending technique 53](#_Toc15024238)

[3.2.4.5 Database Design 54](#_Toc15024239)

[3.2.4.6 User Interface – Navigational Paths and Screen Mock-Ups 55](#_Toc15024240)

[3.2.4.6.1 Website sitemap 55](#_Toc15024241)

[3.2.4.6.2 Details 56](#_Toc15024242)

[3.3 GLOSSARY 61](#_Toc15024243)

[CHAPTER 4: IMPLEMENTATION AND RESULT 62](#_Toc15024244)

[4.1 Accessibility 62](#_Toc15024245)

[4.2 User Manual 63](#_Toc15024246)

[4.3 Test Plan 71](#_Toc15024247)

[4.4 Maintenance Plan 71](#_Toc15024248)

[4.5 Setup and Installation 72](#_Toc15024249)

[CONCLUSION AND DISCUSSION 74](#_Toc15024250)

[REFERENCES 75](#_Toc15024251)

[APPENDIXES 76](#_Toc15024252)

[BIOGRAPHY 77](#_Toc15024253)

LIST OF FIGURES

[Figure 1 Giant Chart Session 20](#_Toc15024254)

[Figure 2 Giant Chart Process 1 21](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024255)

[Figure 3 Giant Chart Process 2 22](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024256)

[Figure 4 Giant Chart Process 3 23](#_Toc15024257)

[Figure 5 Water Fall Model 27](#_Toc15024258)

[Figure 6 V-Shaped Model 28](#_Toc15024259)

[Figure 7 Incremental Model 28](#_Toc15024260)

[Figure 8 Wire Frame example 31](#_Toc15024261)

[Figure 9 Prototype example 32](#_Toc15024262)

[Figure 10 Local Version Control 33](#_Toc15024263)

[Figure 11 Centralized Version Control 34](#_Toc15024264)

[Figure 12 Distributed Version Control 35](#_Toc15024265)

[Figure 13 Use case model shopping online 45](#_Toc15024266)

[Figure 14 View item use case 46](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024267)

[Figure 15 Checkout, authentication and payment use cases 47](#_Toc15024268)

[Figure 16 Database Design 54](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024269)

[Figure 17 Website SiteMap 55](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024270)

[Figure 18 Home Site 56](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024271)

[Figure 19 Site Up Site 57](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024272)

[Figure 20 Sign In Site 58](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024273)

[Figure 21 Sign In Site 2 59](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024274)

[Figure 22 View Detail Site 60](file:///C:\Users\Asus\Desktop\Graduation%20project%20document%20template%20-%20v8.docx#_Toc15024275)

[Figure 23 Cart Site 61](#_Toc15024276)

[Figure 24 Home Page - User Manual 63](#_Toc15024277)

[Figure 25 Sign Up Page - User Manual 64](#_Toc15024278)

[Figure 26 Sign In Page - User Manual 64](#_Toc15024279)

[Figure 27 View Detail - User Manual 65](#_Toc15024280)

[Figure 28 Detail and Recommendation - User Manual 66](#_Toc15024281)

[Figure 29 Check Out Page - User Manual 67](#_Toc15024282)

[Figure 30 Check Out 2 - User Manual 68](#_Toc15024283)

[Figure 31 Check Out 3 - User Manual 69](#_Toc15024284)

[Figure 32 Pop Up Purchasing successfully - User Manual 69](#_Toc15024285)

[Figure 33 History Checkout - User Manual 70](#_Toc15024286)

[Figure 34 Detail of Items - User Manual 70](#_Toc15024287)

LIST OF TABLES

[Table 1 Project phase distribution 21](#_Toc14594691)

SYMBOLS

β Sample Test Phase

ABBREVIATIONS

VSC Visual Studio Code

HTML Hypertext Markup Language

CSS Cascading Style Sheets

CHAPTER 1: INTRODUCTION

## Purpose of the system

The purpose of project is to allow users to seek out and purchase technological products. The website is built by two parts: front-end and back-end. HTML & CSS are used to develop front-end and design the interface of our website. Meanwhile, Java and python are programming languages that have been used to build and improve Back-end and recommended system.

Our website also focuses on primary experiences and business model. Furthermore, it help clients become more efficient and reduce operational expense, bring them better serve our customers and address their needs about technological items.

There are many important elements that go into building a successful e-commerce website such as removing friction during the purchasing process, making the checkout smooth and easy, making the website fast and attractive, up selling users on related products, incentivizing buyers, reducing cart abandonment, nurturing past buyers to buy again, remarketing to past visitors who haven’t yet purchased, using the proper payment options.

## Scope of the system

|  |
| --- |
| The website has features that enable any one person to purchase items online, whether they live in anywhere in Viet Nam. It also lets people be more familiar with their experiences similar to other prominent website such as lazada, tiki.  It reduce client’s monthly expenses because of saving travelling fees. In addition to that users can find and buy virtually anything they may need. Web-based stores are open 24/7; this means you can make purchases when you have the desire and spare time. |

## Objectives and success criteria of the project

**Our website objectives are:**

High reachability - The main objective and at the same time need is traction on our web store. Of course if users are selling products online what we require are customers. If users are getting good reachability then our business will definitely grow. Therefore one of the objective is high reachability.

[High Conversions](http://sunarctechnologies.com/boost-up-your-e-commerce-sales-with-some-easy-dos/) - if people are coming on our web store and purchasing something then it will calculate as conversions and from the number of people who are buying stuff from our web store we can calculate the conversion rate.

[Customer satisfaction](http://sunarctechnologies.com/7-easy-ways-to-handle-customer-service-for-e-commerce/) - Customer is the main part of any E-commerce business so it is very important to make our customer happy and satisfied. By providing quality and desirable products, on time delivery, 24\*7 customer support, and timely sale & best deal offers users can make my customer happy. It is one of the main objectives of our website.

**Success criteria:**

A good website should have a simple design and should be easy to understand and use. Regular updates are necessary for e-commerce sites to reflect the changing avail-ability of products, changes to information, addition of new products and deletion of obsolete products.

Technical availability of the website is also very important. Provision of appropriate and correct product information is very important in deciding the quality of an e-commerce website.

The design of website to enable users to experience easily.

Recommended system processing effectively.

Listing, viewing, purchasing and delivering products chosen intuitively.

## Definitions, acronyms, and abbreviations

Home Page: A main page which lists all of top products and information of website.

AboutUs Page: A page which shows the information of group.

Product Page: A page which displays all available products.

SignUp Page: A page which serves registration procedure to users.

SignIn Page: A page which required users to log in registered account.

Recommend system: A system which recommends items based on what items that users have watched.

COD: Cost On Delivery

## Project plan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *A project plan is one of the most important ingredients for a successful project. It is used to document and communicate expectations, to control schedule and delivery, and to calculate and manage risks. In general, it acts as a roadmap for everyone involved in the project.* 1.5.1 Gantt Chart   Figure 1 Giant Chart Session        Figure 2 Giant Chart Process 1    Figure 3 Giant Chart Process 2    Figure 4 Giant Chart Process 3   1.5.2 Project Phase Distribution In this section  Table 1 Project phase distribution   |  |  |  |  | | --- | --- | --- | --- | | **Planning & Analysis** | **Design** | **Implementation** | **Testing** | | %40 | %15 | % 35 | %10 |  1.5.3 Other Tables You can extend this section as 5.3 Log tables, 5.4 Advisor meeting tables, or other tables or diagrams as you wish. |

|  |
| --- |
| Overview |
| The overview of project is to allow users to seek out and purchase technological products. The website is built by two parts: front-end and back-end. HTML & CSS are used to develop front-end and design the interface of our website. Meanwhile, python is a language that has been used to build and improve Back-end and recommended system. |

CHAPTER 2: METHODOLOGY AND THEORETICAL BACKGROUND

## METHODOLOGY

This study implements qualitative methods in the research stage of the recommendation system application development. The applied methods in this case study are interview, analysis. The results from the qualitative methods have great impact on development of well-designed and functioning website. In the following subsections are described the methods used during the research stage of the thesis, and the design and development of the mobile application.

Recommender systems use statistical and knowledge discovery techniques in order to recommend products to users and to mitigate the problem of information overload. The evaluation of the quality of recommender systems has become an important issue for choosing the best learning algorithms.

A recommender system is a subclass of information filtering system that seeks to predict the “rating” or “preference” a user would give to an item. Recommender systems are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general.

Collaborative filtering

Content-based filtering

Hybrid recommender system

### 2.1.1 Software Development Life Cycle

The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process. SDLC is a structure followed by a development team within the software organization. It consists of a detailed plan describing how to develop, maintain and replace specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The software development life cycle is also known as the software development process.

SDLC consists of following activities:

1. **Planning**: The most important parts of software development, requirement gathering or requirement analysis are usually done by the most skilled and experienced software engineers in the organization. After the requirements are gathered from the client, a scope document is created in which the scope of the project is determined and documented.
2. **Implementation**: The software engineers start writing the code according to the client's requirements.
3. **Testing**: This is the process of finding defects or bugs in the created software.
4. **Documentation**: Every step in the project is documented for future reference and for the improvement of the software in the development process. The design documentation may include writing the application programming interface (API).
5. **Deployment and maintenance**: The software is deployed after it has been approved for release.
6. **Maintaining**: Software maintenance is done for future reference. Software improvement and new requirements (change requests) can take longer than the time needed to create the initial development of the software.

There are several software development models followed by various organizations:

* Waterfall Model: This model involves finishing each phase completely before commencing the next one. When each phase is completed successfully, it is reviewed to see if the project is on track and whether it is feasible to continue.

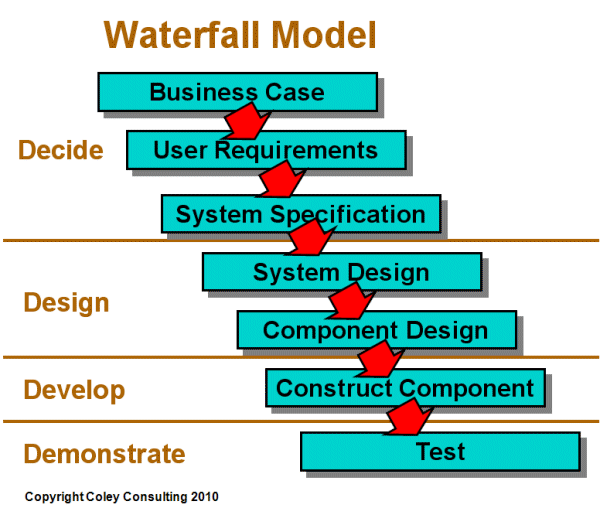


Figure 5 Water Fall Model

* V-Shaped Model: This model focuses on the execution of processes in a sequential manner, similar to the waterfall model but with more importance placed on testing. Testing procedures are written even before the commencement of writing code. A system plan is generated before starting the development phase.

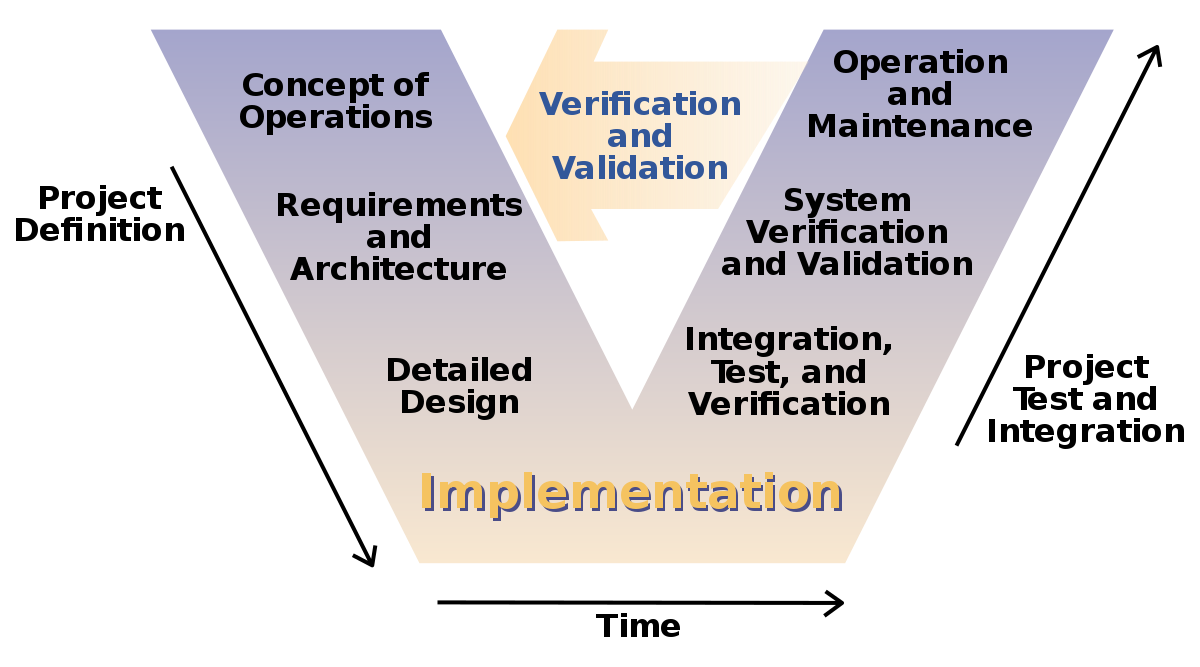


Figure 6 V-Shaped Model

* Incremental Model: This life cycle model involves multiple development cycles. The cycles are divided up into smaller iterations. These iterations can be easily managed and go through a set of phases including requirements, design, implementation and testing. A working version of the software is produced during the first iteration, so working software is created early in the development process.

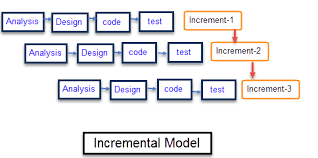


Figure 7 Incremental Model

### 2.1.2 Agile Development Method

The SDLC Model chosen for this case study is the Agile Model. Agile SDLC focuses on quick delivery of functioning software product by adaptable process and constant user feedback agile software development is more than frameworks such as Scrum, Extreme Programming or Feature-Driven Development (FDD).

Agile software development is more than practices such as pair programming, test-driven development, stand-ups, planning sessions and sprints.

Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it. When you approach software development in a particular manner, it’s generally good to live by these values and principles and use them to help figure out the right things to do given your particular context.

One thing that separates Agile from other approaches to software development is the focus on the people doing the work and how they work together. Solutions evolve through collaboration between self-organizing cross-functional teams utilizing the appropriate practices for their context.

There’s a big focus in the Agile software development community on collaboration and the self-organizing team.

That doesn’t mean that there aren’t managers. It means that teams have the ability to figure out how they’re going to approach things on their own.

It means that those teams are cross-functional. Those teams don’t have to have specific roles involved so much as that when you get the team together, you make sure that you have all the right skill sets on the team.

There still is a place for managers. Managers make sure team members have, or obtain, the right skill sets. Managers provide the environment that allows the team to be successful. Managers mostly step back and let their team figure out how they are going to deliver products, but they step in when the teams try but are unable to resolve issues.

When most teams and organizations start doing Agile software development, we focus on the practices that help with collaboration and organizing the work, which is great. However, another key set of practices that are not as frequently followed but should be are specific technical practices that directly deal with developing software in a way that help your team deal with uncertainty. Those technical practices are essential and something you shouldn’t overlook.

Agile web development is actually a broad category of methodologies based on the principles outlined in the [Manifesto for Agile Software Development](http://agilemanifesto.org/principles.html), which was compiled by a team of professional developers in 2001. Specific methods such a scrum and XP are considered agile although they existed before the manifesto was written. Through their combined experiences of working with other developers, the authors recognized the value of adaptive planning and collaboration between self-organizing, cross-functional teams. The goal is to allow for flexibility and provide rapid and continuous improvement of software solutions. Early delivery is also a key goal of agile development, which entails streamlining projects by eliminating time-sucking tasks.

### 2.1.3 Wireframe and prototype

Wireframes, mockups and prototypes actually represent the different stages of design flow.

**Wireframe**, a low-fidelity way to present a product, can efficiently outline structures and layouts. Wireframe is the basic and visual representation of the design. Your wireframe design doesn’t need to focus too much on minutiae, but must express design ideas and should not miss any important parts. A wireframe is like a channel that helps team member understand their projects better.

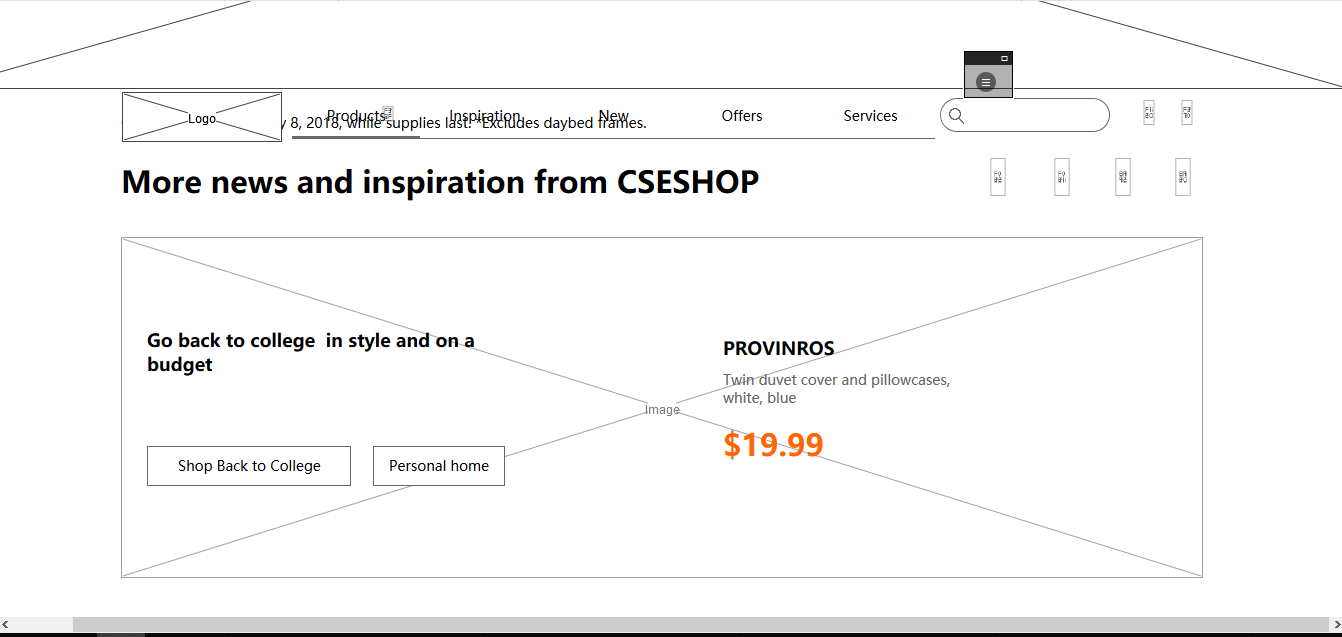


Figure 8 Wire Frame example

**Prototype** is already very close to the finished product. Here, processes can be simulated and user interaction can be tested. A prototype looks very similar to the finished product. Early prototyping can save a lot of development costs and time so that the work of back-end product architecture will not be in vain because of unreasonable [user interface design](https://www.mockplus.com/blog/post/118-7-unbreakable-laws-of-user-interface-design/?r=vincent). A prototype is an excellent tool to obtain user feedback and to test the product.

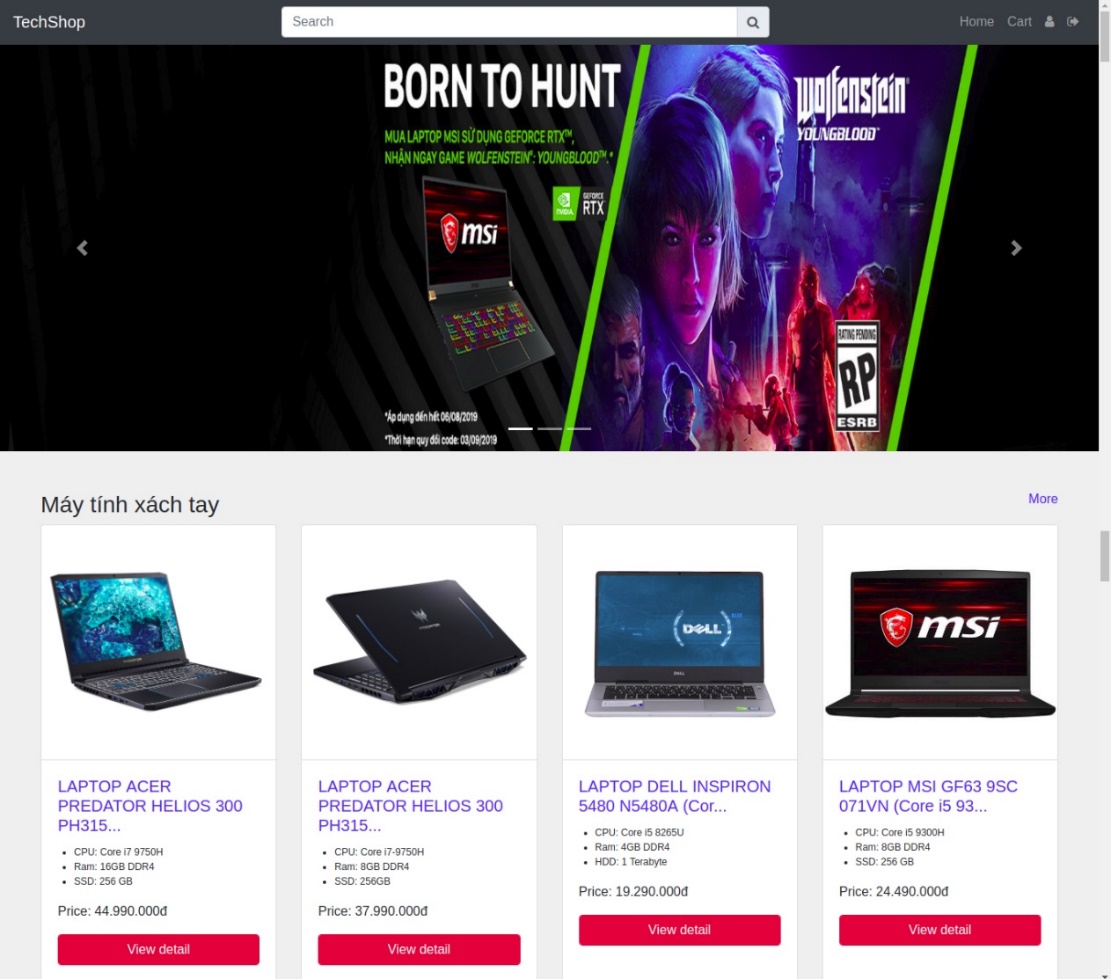


Figure 9 Prototype example

### 2.1.4 Source code control

This chapter will be about getting started with Git. We will begin by explaining some background on version control tools, then move on to how to get Git running on your system and finally how to get it set up to start working with.

**About Version Control**

What is “version control”, and why should you care? Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. For the examples in this book, you will use software source code as the files being version controlled, though in reality you can do this with nearly any type of file on a computer.

**Local Version Control Systems**

Many people’s version-control method of choice is to copy files into another directory (perhaps a time-stamped directory, if they’re clever). This approach is very common because it is so simple, but it is also incredibly error prone. It is easy to forget which directory you’re in and accidentally write to the wrong file or copy over files you don’t mean to.

To deal with this issue, programmers long ago developed local VCSs that had a simple database that kept all the changes to files under revision control.

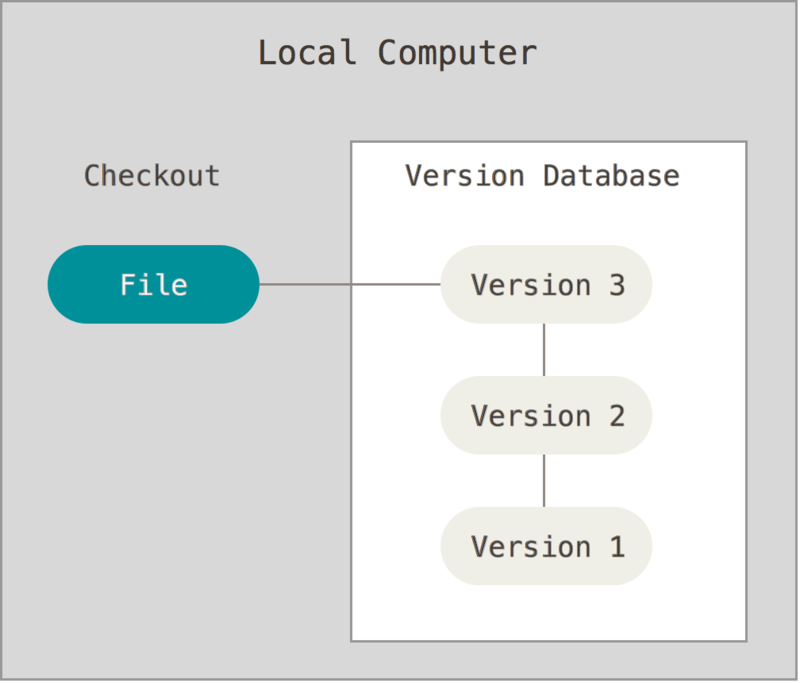


Figure 10 Local Version Control

**Centralized Version Control Systems**

The next major issue that people encounter is that they need to collaborate with developers on other systems. To deal with this problem, Centralized Version Control Systems (CVCSs) were developed. These systems (such as CVS, Subversion, and Perforce) have a single server that contains all the versioned files, and a number of clients that check out files from that central place. For many years, this has been the standard for version control.

This setup offers many advantages, especially over local VCSs. For example, everyone knows to a certain degree what everyone else on the project is doing. Administrators have fine-grained control over who can do what, and it’s far easier to administer a CVCS than it is to deal with local databases on every client.

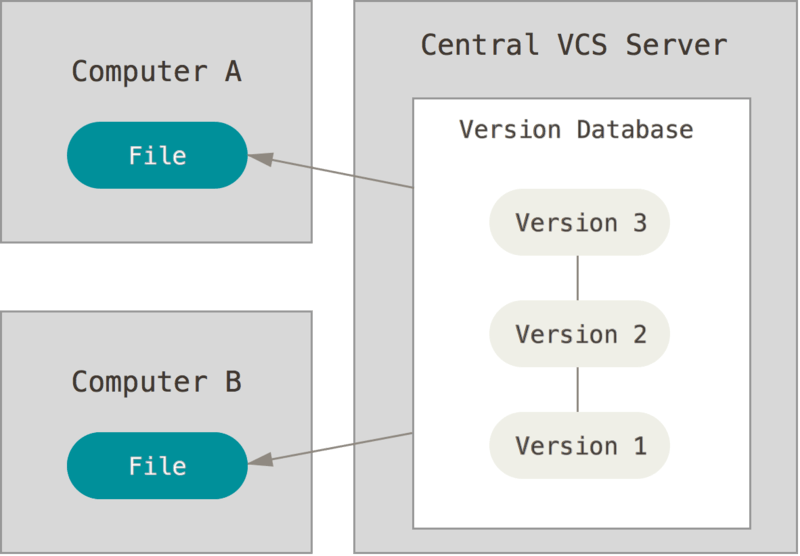


Figure 11 Centralized Version Control

**Distributed Version Control Systems**

This is where Distributed Version Control Systems (DVCSs) step in. In a DVCS (such as Git, Mercurial, Bazaar or Darcs), clients don’t just check out the latest snapshot of the files; rather, they fully mirror the repository, including its full history. Thus, if any server dies, and these systems were collaborating via that server, any of the client repositories can be copied back up to the server to restore it. Every clone is really a full backup of all the data.

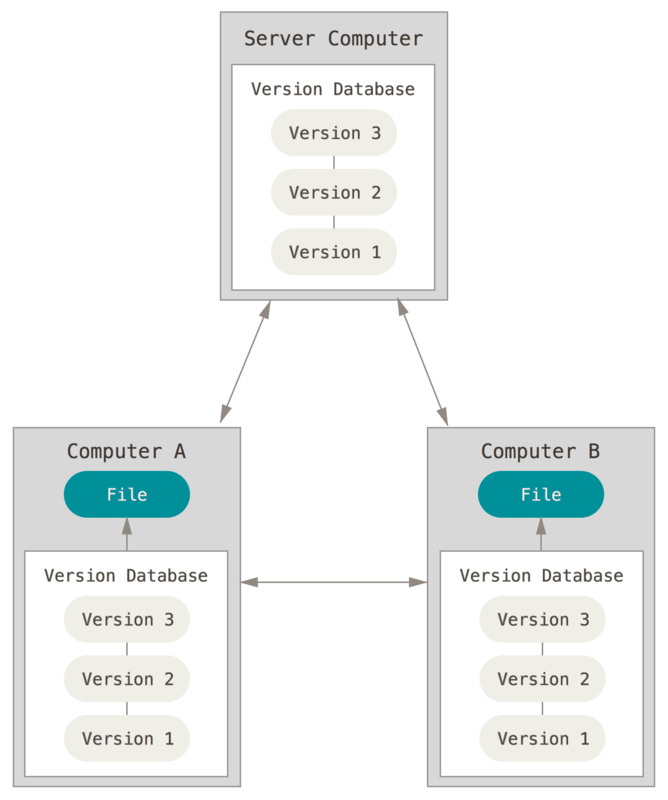
****

Figure 12 Distributed Version Control

## THEORICAL BACKGROUND

This chapter of the thesis will look into the theory of web service and recommendation system. The chapter 2.2 will concentrate on the rest of the theoretical background is exploring web service and recommendation system. The theory is concentrating on how to build web service RESTful apis using the spring-boot framework, web service to save images of the website and a recommendation system using python.

### 2.2.1. Overview of web service technology

**What is Restful Web Service?**

REST is used to build Web services that are lightweight, maintainable, and scalable in nature. A service which is built on the REST architecture is called a RESTful service. The underlying protocol for REST is HTTP, which is the basic web protocol. REST stands for Representational State Transfer.

**RESTful Key Elements**

Web services have really come a long way since its inception. In 2002, the Web consortium had released the definition of WSDL and SOAP web services. This formed the standard of how web services are implemented.

In 2004, the web consortium also released the definition of an additional standard called RESTful. Over the past couple of years, this standard has become quite popular. And is being used by many of the popular websites around the world which include Facebook and Twitter.

REST is a way to access resources which lie in a particular environment. For example, you could have a server that could be hosting important documents or pictures or videos. All of these are an example of resources. If a client, say a web browser needs any of these resources, it has to send a request to the server to access these resources. Now REST defines a way on how these resources can be accessed.

The key elements of a RESTful implementation are as follows:

1. Resources – The first key element is the resource itself. Let assume that a web application on a server has records of several employees. Let's assume the URL of the web application is http://localhost:8080. Now in order to access an employee record resource via REST, one can issue the command http://localhost:8080/item/1 - This command tells the web server to please provide the details of the employee whose item number is
2. Request Verbs - These describe what you want to do with the resource. A browser issues a GET verb to instruct the endpoint it wants to get data. However, there are many other verbs available including things like POST, PUT, and DELETE. So in the case of the example http://localhost:8080/item/1 - the web browser is actually issuing a GET Verb because it wants to get the details of the item record.
3. Request Headers – These are additional instructions sent with the request. These might define the type of response required or the authorization details.
4. Request Body - Data is sent with the request. Data is normally sent in the request when a POST request is made to the REST web service. In a POST call, the client actually tells the web service that it wants to add a resource to the server. Hence, the request body would have the details of the resource which is required to be added to the server.
5. Response Body – This is the main body of the response. So in our example, if we were to query the web server via the request http://localhost:8080/item/1 - the web server might return an XML document with all the details of the item in the Response Body.
6. Response Status codes – These codes are the general codes which are returned along with the response from the web server. An example is the code 200 which is normally returned if there is no error when returning a response to the client.

**Restful Methods**

The below diagram shows mostly all the verbs (POST, GET, PUT, and DELETE) and an example of what they would mean.

Let's assume that we have a RESTful web service is defined at the location. http://localhost:8080/item. When the client makes any request to this web service, it can specify any of the normal HTTP verbs of GET, POST, DELETE and PUT. Below is what would happen. If the respective verbs were sent by the client.

1. POST – This would be used to create a new item using the RESTful web service.
2. GET - This would be used to get a list of all item using the RESTful web service.
3. PUT - This would be used to update all item using the RESTful web service.
4. DELETE - This would be used to delete all employee using the RESTful web service.

Let's take a look from a perspective of just a single record. Let's say there was an item record with the item number of 1.

The following actions would have their respective meanings.

1. POST – This would not be applicable since we are fetching data of item 1 which is already created.
2. GET - This would be used to get the details of the item with item no as 1 using the RESTful web service.
3. PUT - This would be used to update the details of the item with item no as 1 using the RESTful web service.
4. DELETE - This is used to delete the details of the item with item no as 1.

### 2.2.2. Web service RESTful architechture

An application or architecture considered RESTful or REST-style has the following characteristics.

1. State and functionality are divided into distributed resources – This means that every resource should be accessible via the normal HTTP commands of GET, POST, PUT, or DELETE. So if someone wanted to get a file from a server, they should be able to issue the GET request and get the file. If they want to put a file on the server, they should be able to either issue the POST or PUT request. And finally, if they wanted to delete a file from the server, they an issue the DELETE request.
2. The architecture is client/server, stateless, layered, and supports caching:

* Client-server is the typical architecture where the server can be the web server hosting the application, and the client can be as simple as the web browser.
* Stateless means that the state of the application is not maintained in REST.

For example, if you delete a resource from a server using the DELETE command, you cannot expect that delete information to be passed to the next request.

* In order to ensure that the resource is deleted, you would need to issue the GET request. The GET request would be used to first get all the resources on the server. After which one would need to see if the resource was actually deleted.

### 2.2.3. Understanding RESTFul Principles and Constraints

The REST architecture is based on a few characteristics which are elaborated below. Any RESTful web service has to comply with the below characteristics in order for it to be called RESTful. These characteristics are also known as design principles which need to be followed when working with RESTful based services.

#### 2.2.3.1 RESTFul Client-Server

This is the most fundamental requirement of a REST based architecture. It means that the server will have a RESTful web service which would provide the required functionality to the client. The client send's a request to the web service on the server. The server would either reject the request or comply and provide an adequate response to the client.

#### 2.2.3.2 Stateless

The concept of stateless means that it's up to the client to ensure that all the required information is provided to the server. This is required so that server can process the response appropriately. The server should not maintain any sort of information between requests from the client. It's a very simple independent question-answer sequence. The client asks a question, the server answers it appropriately. The client will ask another question. The server will not remember the previous question-answer scenario and will need to answer the new question independently.

#### 2.2.3.3 Cache

The Cache concept is to help with the problem of stateless which was described in the last point. Since each server client request is independent in nature, sometimes the client might ask the server for the same request again. This is even though it had already asked for it in the past.This request will go to the server, and the server will give a response. This increases the traffic across the network. The cache is a concept implemented on the client to store requests which have already been sent to the server. So if the same request is given by the client, instead of going to the server, it would go to the cache and get the required information. This saves the amount of to and fro network traffic from the client to the server.

#### **2.2.3.4 Layered System.**

The concept of a layered system is that any additional layer such as a middleware layer can be inserted between the client and the actual server hosting the RESTFul web service (The middleware layer is where all the business logic is created. This can be an extra service created with which the client could interact with before it makes a call to the web service.). But the introduction of this layer needs to be transparent so that it does not disturb the interaction between the client and the server.

#### **2.2.3.5 Interface/Uniform Contract**

This is the underlying technique of how RESTful web services should work. RESTful basically works on the HTTP web layer and uses the below key verbs to work with resources on the server

* POST - To create a resource on the server.
* GET - To retrieve a resource from the server.
* PUT - To change the state of a resource or to update it.
* DELETE - To remove or delete a resource from the server.

### 2.2.4. Definitions and types of recommendation system

What is Recommendation system?

A recommendation system is an extensive class of web applications that involves predicting the user responses to the options.

What types of recommendation system?

#### **2.2.4.1 Popularity based**

Easiest way to build a recommendation system is popularity based, simply over all the products that are popular, so how to identify popular products, which could be identified by which are all the products that are bought most. Example, in shopping store we can suggest popular dresses by purchase count.

#### **2.2.4.2 Classification based**

Second way to build a recommendation system is classification model, in that use feature of both users as well as products in order to predict whether this product liked or not by the user. When new users come, our classifier will give a binary value of that product liked by this user or not, in such a way that we can recommend a product to the user.

#### **2.2.4.3 Collaborative filtering**

Collaborative filtering models which are based on assumption that people like things similar to other things they like, and things that are liked by other people with similar taste.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

## CURRENT SYSTEM

*The Current system describes the current state of affairs. If the new system will replace an existing system, this section describes the functionality and the problems of the current system. Otherwise, this section describes how the tasks supported by the new system are accomplished now.*

## PROPOSED SYSTEM

*The third section documents the requirements elicitation and the analysis model of the new system.*

### 3.2.1 Overview

This section provides a functional overview of the system. This will again be properly be divided into two parts: functional requirements and nonfunctional requirements.

### 3.2.2 Functional Requirements

Internet browser is required such as: Google Chrome, Mozila Firefox, etc.

Personal details of user needed to be filled in form to help delivery process occurring smoothly.

Keeping records of admission of customers.

Keeping record of the products.

Keeping the daily sell.

Storing feedback of customers.

Keeping details products whether it is delivered or not.

Storing the items selected by users in the temporary storage.

### 3.2.3 Nonfunctional Requirements

Following Non-Functional requirements will be there in the insurance on internet:

**i).** 24 x 7 availability

**ii).** Better component design to get better performance at peak time.

#### 3.2.3.1 Usability

* Website must be used and manipulated by mouse.
* Readability of Text.
* Convenience of a product Catalog.
* Description and Photo of a product.
* Similar Products.
* Contact information of Terms and Service.
* System must notice to users who have purchased or selected temporary items recently.

#### 3.2.3.2 Reliability

* Providing sufficient contact details, not only an e-mail address.
* Components of the project code will be tested alongside the implementation phase to ensure that they are functional.

#### 3.2.3.3 Performance

* Manipulation must be smooth without graphical lagging.
* Validation of the information of users or keeping track of delivery process will be noticeable.

#### 3.2.3.4 Supportability

* The website should be able to run on any browsers.

#### 3.2.3.5 Implementation

* All project graphical user interfaces will be created using a HTML and CSS.
* Recommend System is implemented Python.

*.*

### 3.2.4 System Models

*Describes the scenarios, use cases, object model, and dynamic models for the system. This section contains the complete functional specification, including mock-ups illustrating the user interface of the system and navigational paths representing the sequence of screens. The subsections Object model and Dynamic model are written during the Analysis activity.*

#### 3.2.4.1 Scenarios

A scenario is an outline of events - planned or unplanned, possible or probable - depending on the circumstances.

#### 3.2.4.2 User case model

##### 3.2.4.2.1 User case diagram

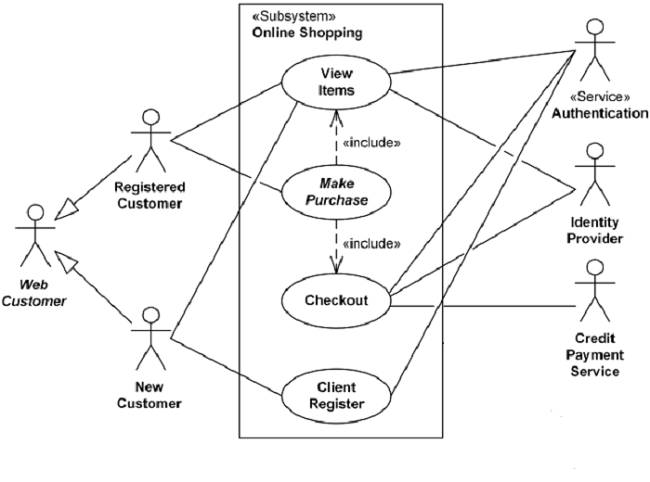


Figure 13 Use case model shopping online

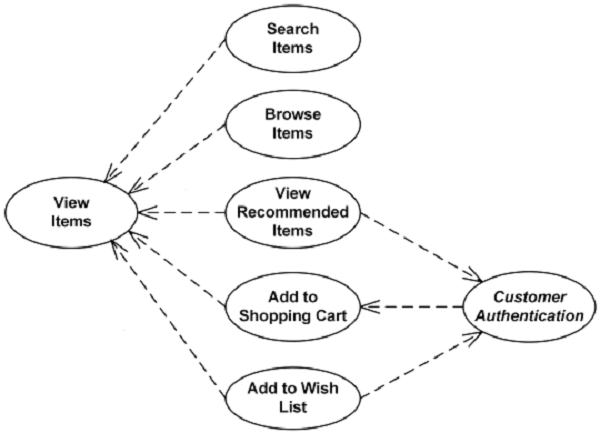


Figure 14 View item use case

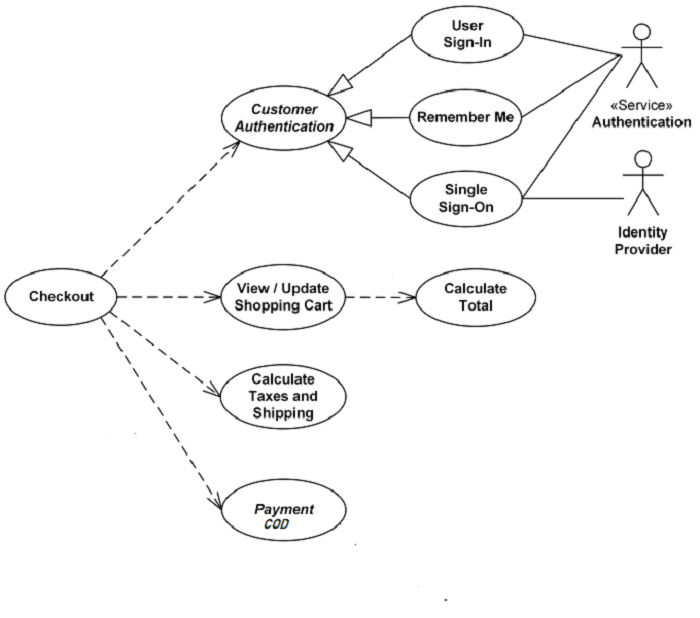


Figure 15 Checkout, authentication and payment use cases

##### 3.2.4.2.2 User case table

|  |  |
| --- | --- |
| **Name:** | Redirecting Home Page |
| **Actor:** | ANONYMOUS USER |
| **Entry**  **Conditions:** | Installed browser : Chorm, FireFox,… |
| **Flow of**  **Events:** | 1. USER turns browser on  2. Navigate to http://localhost:8080 |
| **Exit**  **Conditions:** | Home Page appears on browser. |

|  |  |
| --- | --- |
| **Name:** | SIGN UP |
| **Actor:** | ANONYMOUS USER |
| **Entry**  **Conditions:** | A website is currently in progress |
| **Flow of**  **Events:** | 1. USER clicks on Signup button on navigation bar.  2. Filling fullname (At least 8 characters).  3. Filling appropriately email address (hungduong@gmail.com).  4. Filling password (At least 8 characters).  5. Filling confirmation password.  6. Click Sign Up button. |
| **Exit**  **Conditions:** | Pop up will notices. |

|  |  |
| --- | --- |
| **Name:** | SIGNUP WITH EXIST EMAIL ADDRESS |
| **Actor:** | USER |
| **Entry**  **Conditions:** | A website is currently in progress |
| **Flow of**  **Events:** | 1. USER clicks on Signup button on navigation bar.  2. Filling fullname (At least 8 characters).  3. Filling appropriately email address following (hungduong@gmail.com).  4. Filling password (At least 8 characters).  5. Filling confirmation password.  6. Click Sign Up button. |
| **Exit**  **Conditions:** | Warning pop up will notices . |

|  |  |
| --- | --- |
| **Name:** | SignIn |
| **Actor:** | ANONYMOUSE USER |
| **Entry**  **Conditions:** | A website is currently in progress |
| **Flow of**  **Events:** | 1. USER clicks on Sign In button on navigation bar.  2. Filling registered email address and password.  3. Click on Sign In button. |
| **Exit**  **Conditions:** | Sign In page will refreshes to Home Page. |

|  |  |
| --- | --- |
| **Name:** | Purchasing Item |
| **Actor:** | USER |
| **Entry**  **Conditions:** | A website is currently in progress  Login successfully |
| **Flow of**  **Events:** | 1. Click on View detail below item.  2. Click “ Thêm vào giỏ hàng”. |
| **Exit Conditions:** | Pop up will notices . |

|  |  |
| --- | --- |
| **Name:** | Rating item |
| **Actor:** | USER |
| **Entry**  **Conditions:** | After added to cart succesfully. |
| **Flow of**  **Events:** | 1. User chooses level of rating from 1 to 5 stars.  2. Comment in comment box.  3. Click on Post button. |
| **Exit**  **Conditions:** | Pop up will notices . |

|  |  |
| --- | --- |
| **Name:** | View Cart |
| **Actor:** | USER |
| **Entry**  **Conditions:** | After added to cart succesfully.  A website is currently in progress. |
| **Flow of**  **Events:** | 1. Click on cart button on navigation bar. |
| **Exit**  **Conditions:** |  |

|  |  |
| --- | --- |
| **Name:** | Proceed to checkout |
| **Actor:** | USER |
| **Entry**  **Conditions:** | After added to cart succesfully.  A website is currently in progress. |
| **Flow of**  **Events:** | 1. Click on cart button on navigation bar.  2. Click on “Proceed to checkout” button.  3. Filling address details.  4. Click “Complete your information” button. |
| **Exit**  **Conditions:** | Pop up will notices. |

|  |  |
| --- | --- |
| **Name:** | Order details |
| **Actor:** | USER |
| **Entry**  **Conditions:** | A website is currently in progress.  Proceed to checkout successfully. |
| **Flow of**  **Events:** | 1. Click on next button on Proceed to checkout page.  2. Click on “ Thanh Toán” button. |
| **Exit**  **Conditions:** | Pop up notices . |

|  |  |
| --- | --- |
| **Name:** | Change password of account |
| **Actor:** | USER |
| **Entry**  **Conditions:** | A website is currently in progress.  Existing account. |
| **Flow of**  **Events:** | 1. Click on “user” symbol on navigation bar.  2. Click on “change password” line.  3. Fill old password.  4. Fill new password (at least 8 characters).  5. Fill confirmation password.  6. Click on update button. |
| **Exit**  **Conditions:** | Pop up will notices. |

#### 3.2.4.3 Object Model

##### 3.2.4.3.1 Activity Diagram

**Bỏ cái main activity của web mình zô đây nè**

##### 3.2.4.3.2 Class Diagram

Chụp hình mấy cái class của main activity nè ( không cần giải thích có class chi)

#### 3.2.4.4 Dynamic Model

##### 3.2.4.4.1 Searching Technique

Kĩ thuật dùng để search trong web minhf

##### 3.2.4.4.2 Recommending technique

**Kĩ thuật dùng để recommend – thuật toán sử dụng trong web mình**

#### 3.2.4.5 Database Design

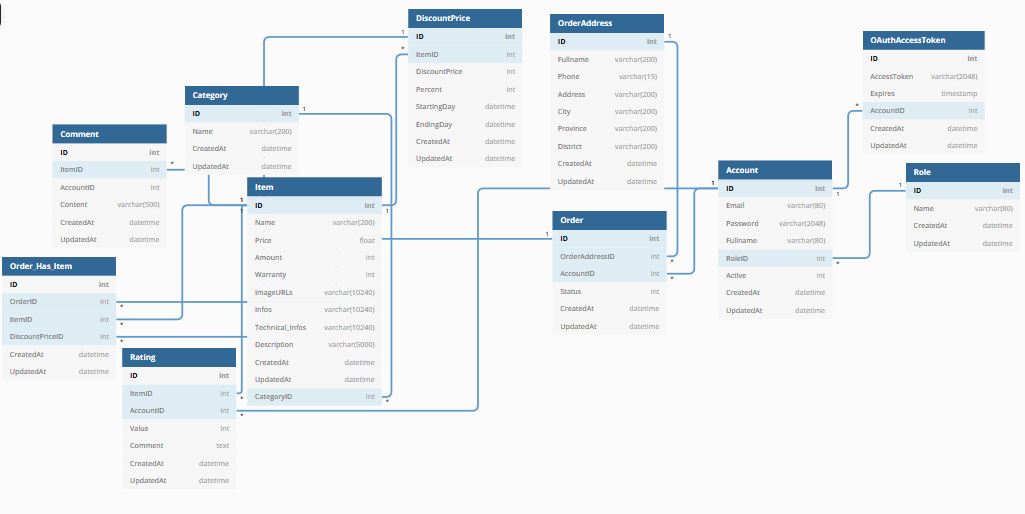


Figure 16 Database Design

#### 3.2.4.6 User Interface – Navigational Paths and Screen Mock-Ups

##### 3.2.4.6.1 Website sitemap

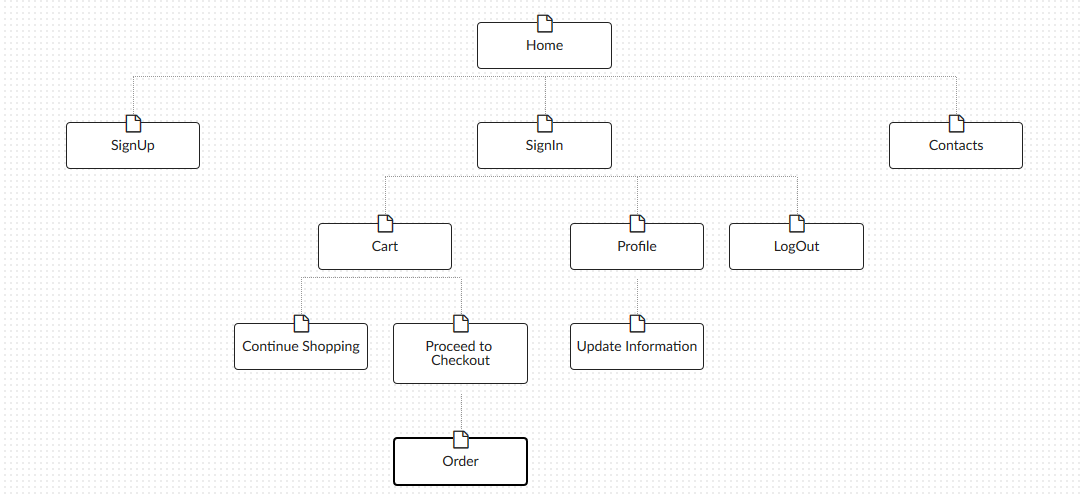
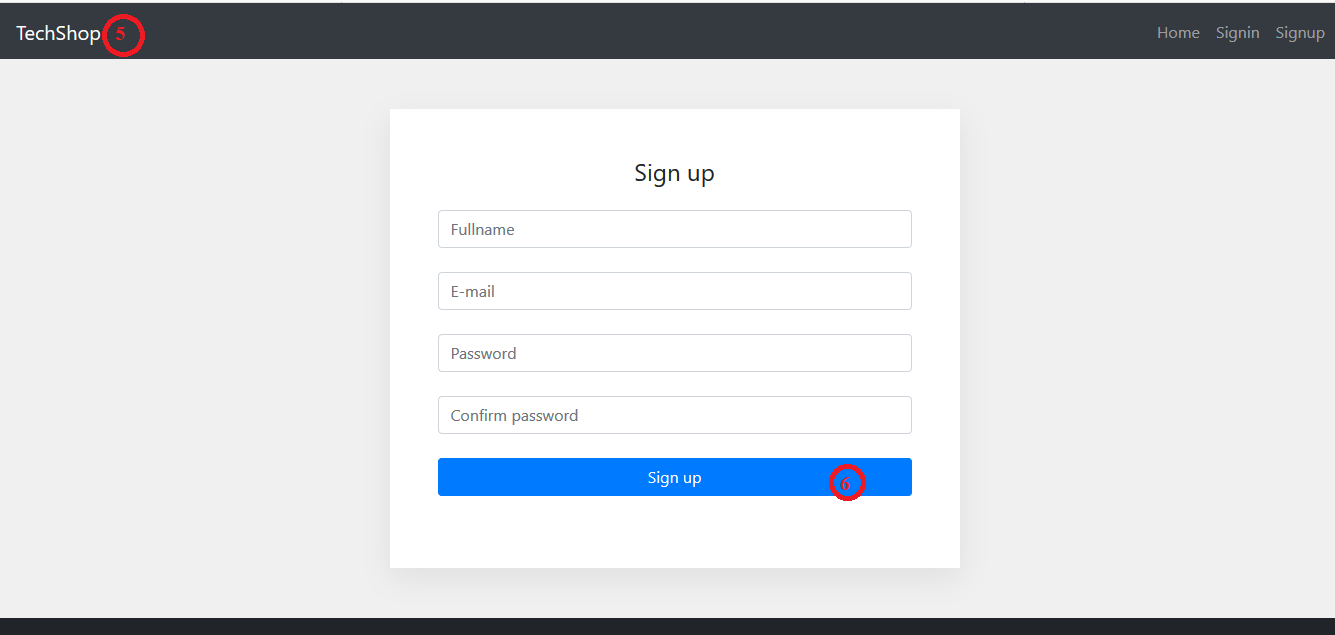
****

Figure 17 Website SiteMap

##### 3.2.4.6.2 Details

Figure 18 Home Site

|  |  |
| --- | --- |
| |  | | --- | |  |   **Interactions:**   1. On Click: goes to Home Page. 2. On Click: goes to Home Page. 3. On Click: goes to Sign Up Page. 4. On Click: goes to Sign In Page. |



|  |
| --- |
| Figure 19 Site Up Site  **Interactions:**  5. On Click: goes to Home Page.  6. On Click: Sign Up.    Figure 20 Sign In Site |

|  |
| --- |
| **Interactions:**  7. On Click: goes to Home Page.  8. On Click: Sign In.  9. On Click: goes to Sign Up Page.  10. On Click: Forgot Password.    Figure 21 Sign In Site 2  **Interactions:**  11. On Clicks: goes to Home Page.  12. On Clicks: goes to Cart Page.  13. On Clicks: goes to Profile Page.  14. On Clicks: Log out and goes to Home Page.    Figure 22 View Detail Site  **Interactions:**  15. On Clicks: goes to Proceed Page.  16. Show recommendation items.    Figure 23 Cart Site |
| **Interactions:**  17. On Clicks: goes to Home Page.  18. On Clicks: goes to Order Page. |

## 3.3 GLOSSARY

*A glossary of important terms, to ensure consistency in the specification and to ensure that we use the client’s terms. A precursor to the****Data Dictionary.***

CHAPTER 4: IMPLEMENTATION AND RESULT

In this chapter, you can write anything about your project. User manual, screenshot, essential part of your project, whatever you want...

You can extend this chapter as 6.6, 6.7 and so on.

|  |
| --- |
| Accessibility |
| **Images**   * Images have descriptive alternative text * Decorative and spacer images have null alternative text.   **Plain English**   * Some pages may still contain complex and technical language, but we will, where possible, simplify complex language unless unable to do so due to legal reasons.   **Consistent navigation**   * The website has been designed to provide a variety of navigation routes to access information. * Key navigation options can be accessed from every page on the site.   **Forms**   * Form fields contain the appropriate text labels, field sets, legends and captions * Forms have accessible error handling which are marked with an asterisk for mandatory fields.   **Link to website help from all pages**   * We provide a link from the footer of the website through to our help pages.   **Correctly structured pages**   * We aim to provide correctly structured and well-titled headings. |

|  |
| --- |
| User Manual |

* Open browser and enter url: [http://localhost:8080](http://localhost:8080/)

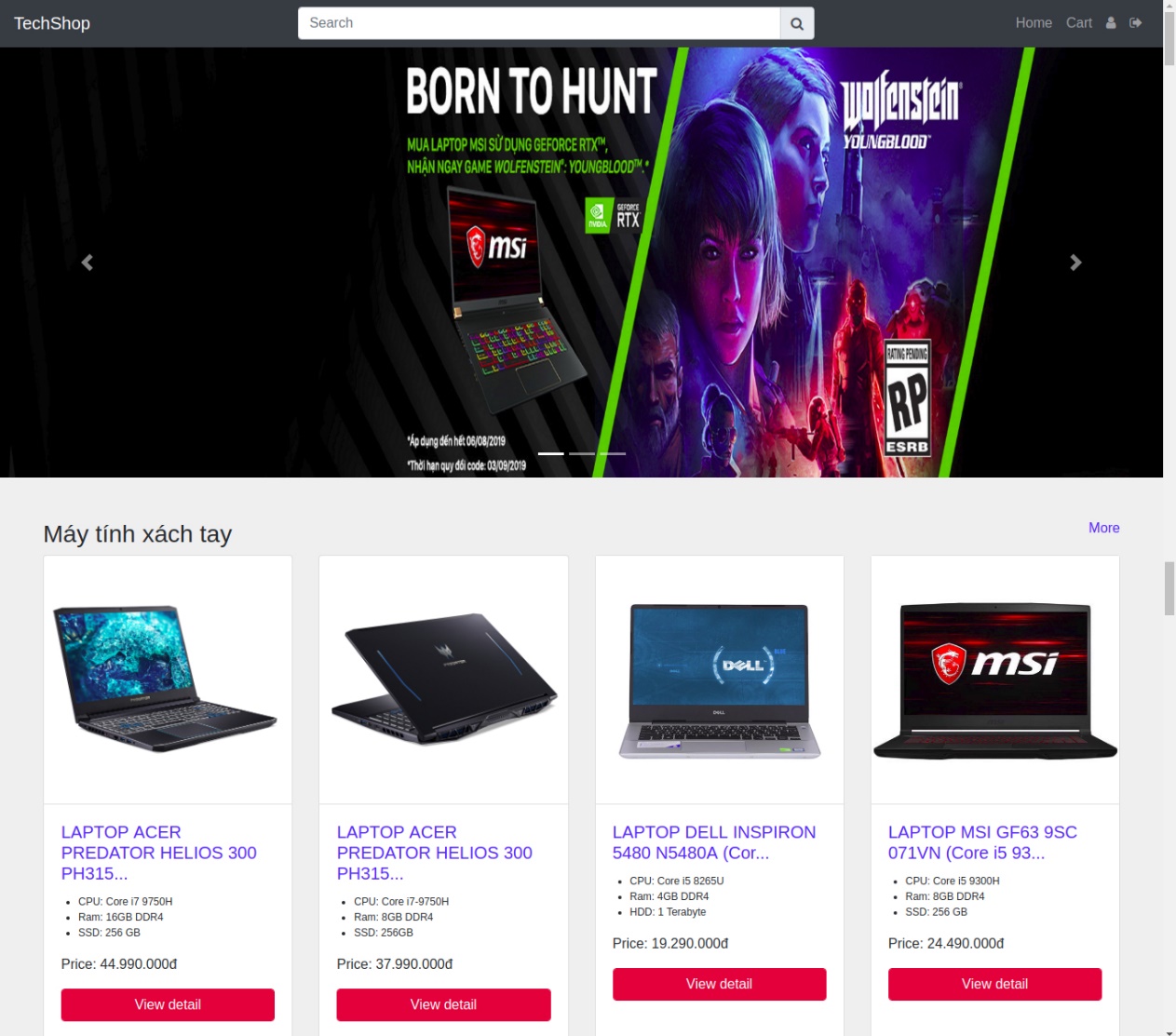


Figure 24 Home Page - User Manual

* Redirect to signup page (<http://localhost:8080/signup>). Create account to buy something. Enter your Fullname, Email, password.

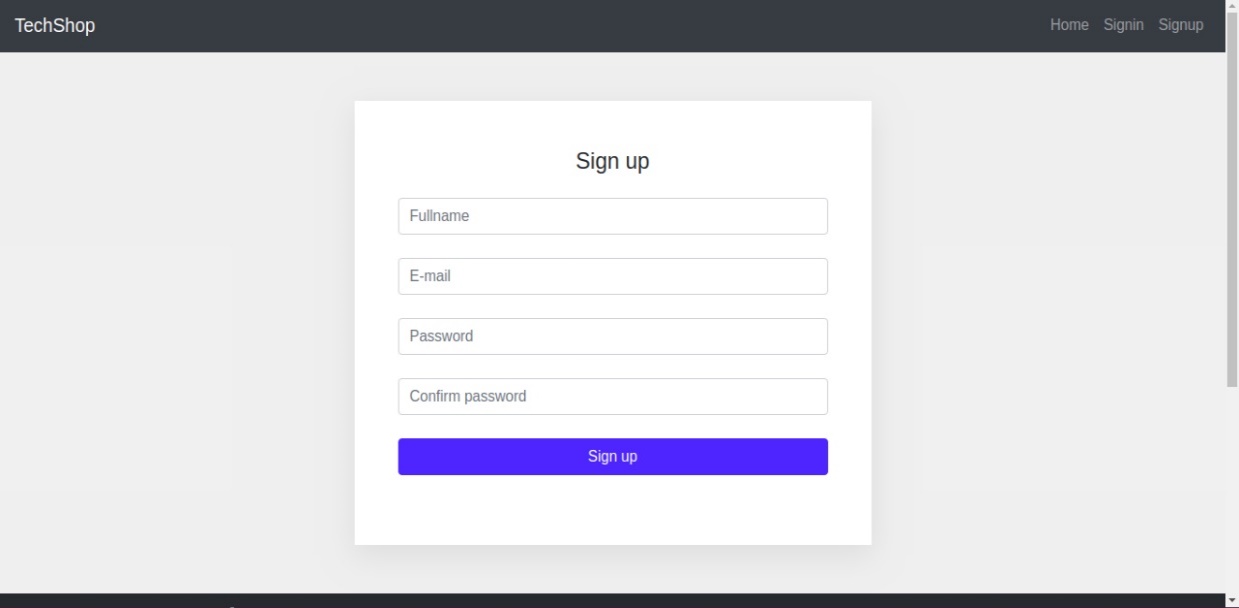


Figure 25 Sign Up Page - User Manual

* After create account successfully. You redirect to login page (<http://localhost:8080/login>). In this page, you must enter email and password to login.

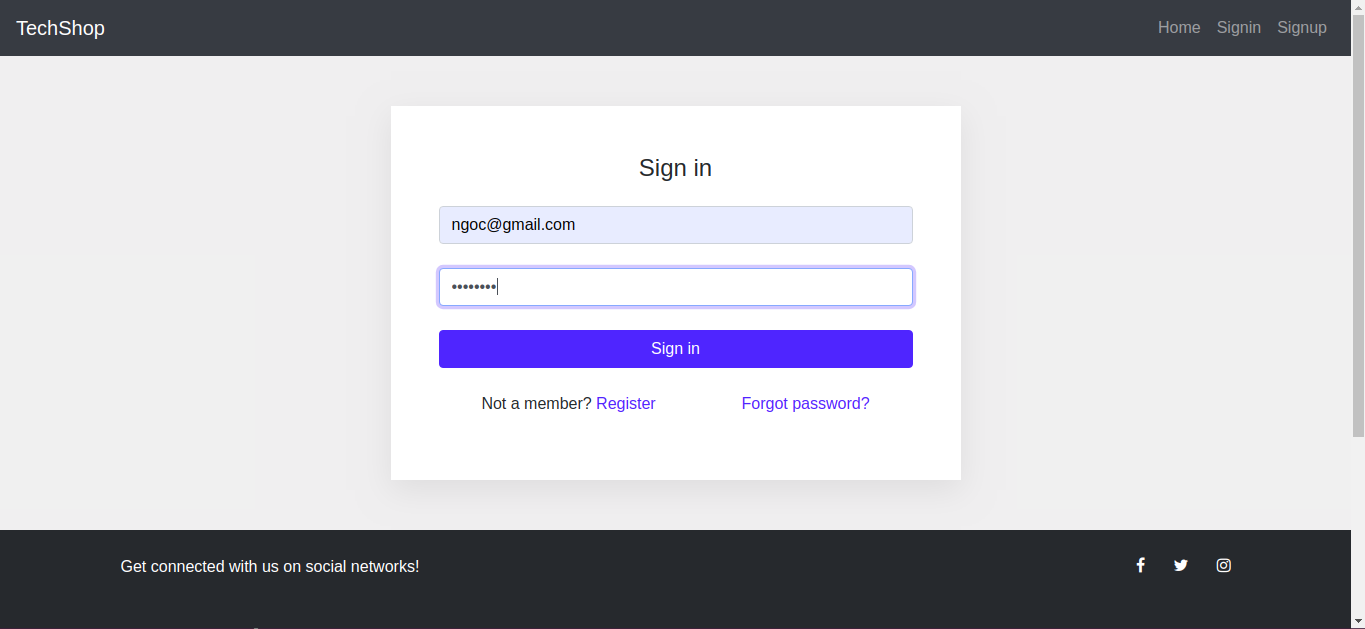


Figure 26 Sign In Page - User Manual

* Redirect to home page or items page, and enter name of item you want to buy.

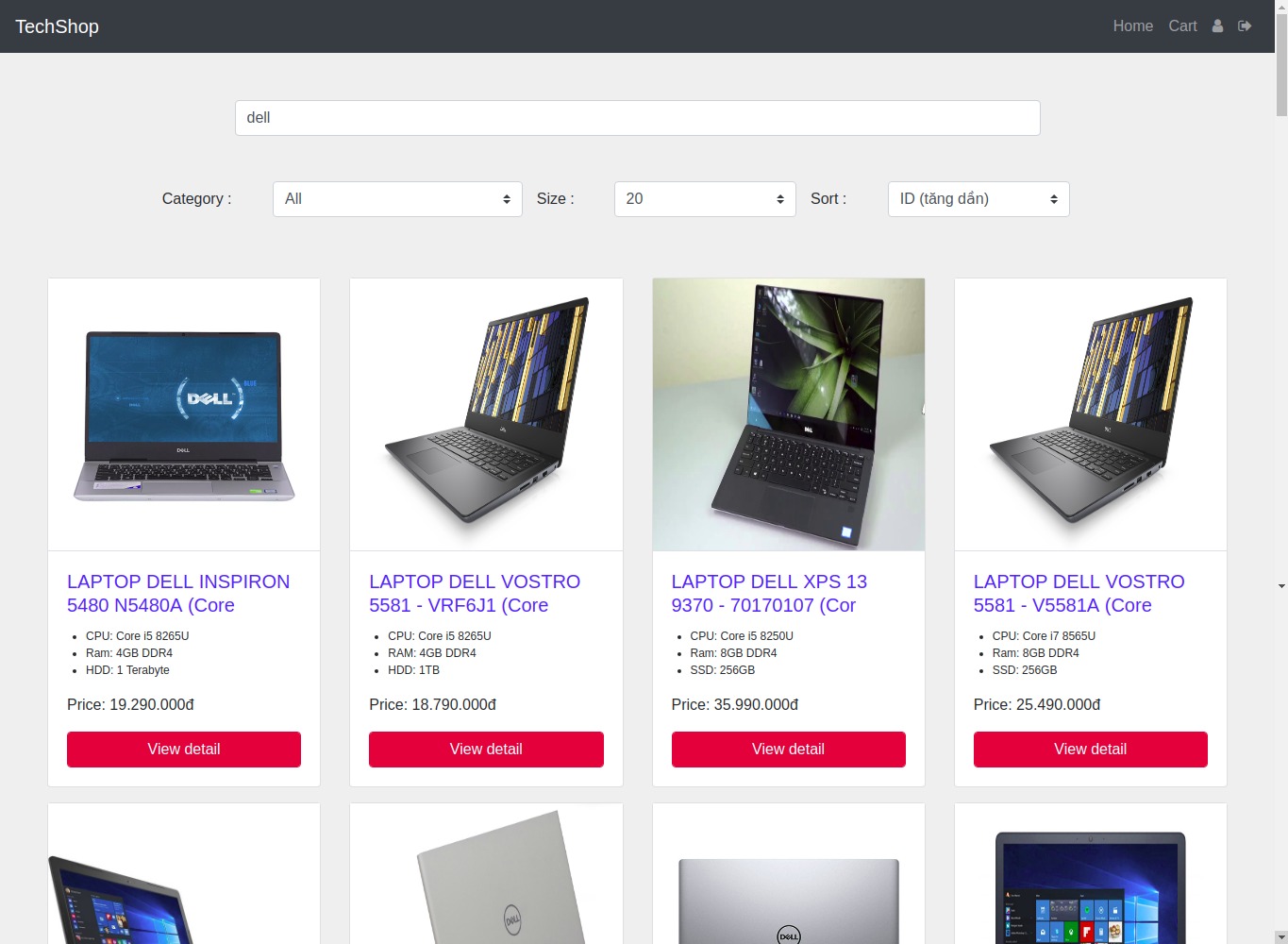


Figure 27 View Detail - User Manual

* When have result of searching, you can click button ‘View detail’ to get more information of item, buy it and rate it . If buy it, you choose the quantity of items and click ‘Thêm vào giỏ hàng’

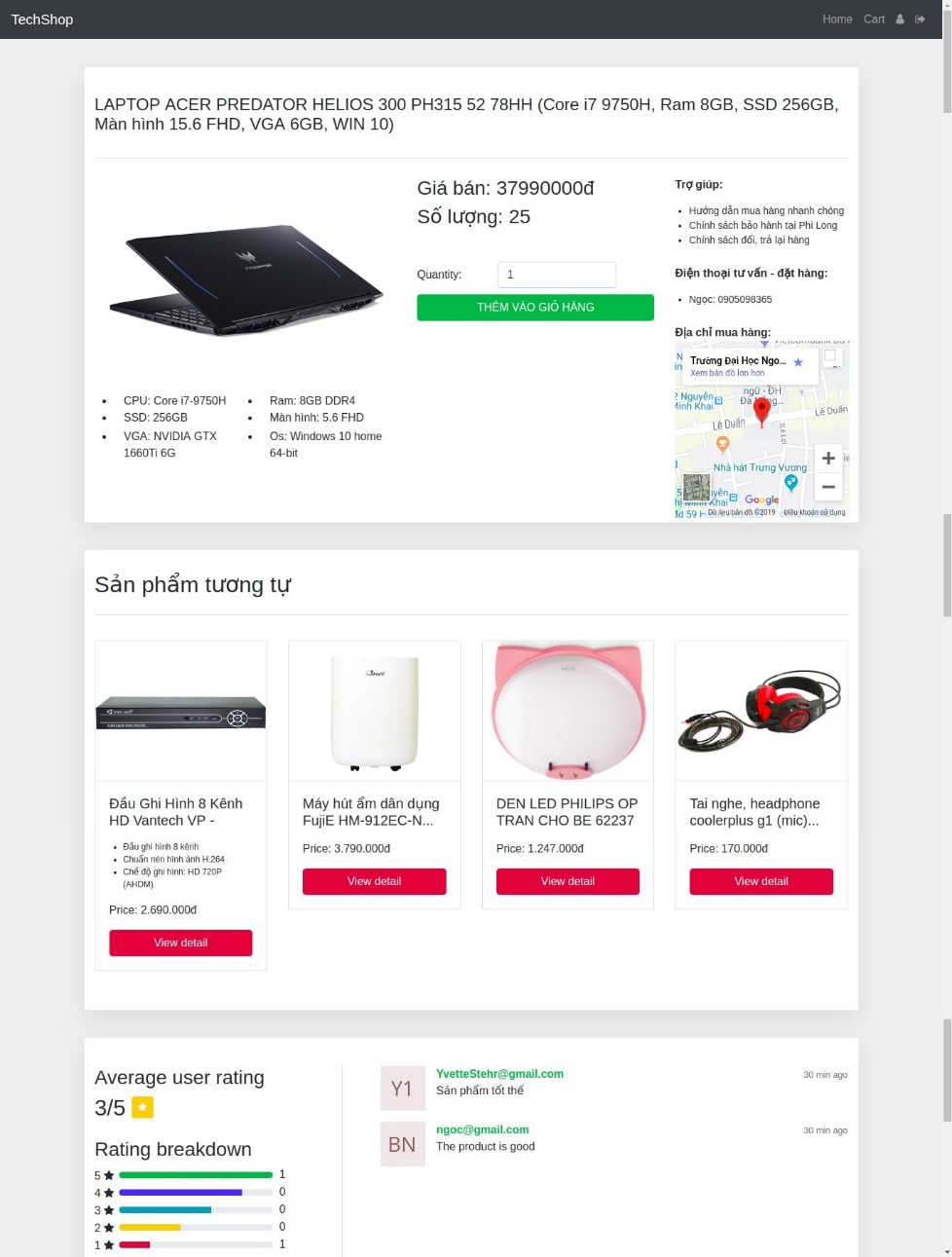


Figure 28 Detail and Recommendation - User Manual

* After click ‘Thêm vào giỏ hàng’, you can go to cart page to check you items that you choose and purchase.

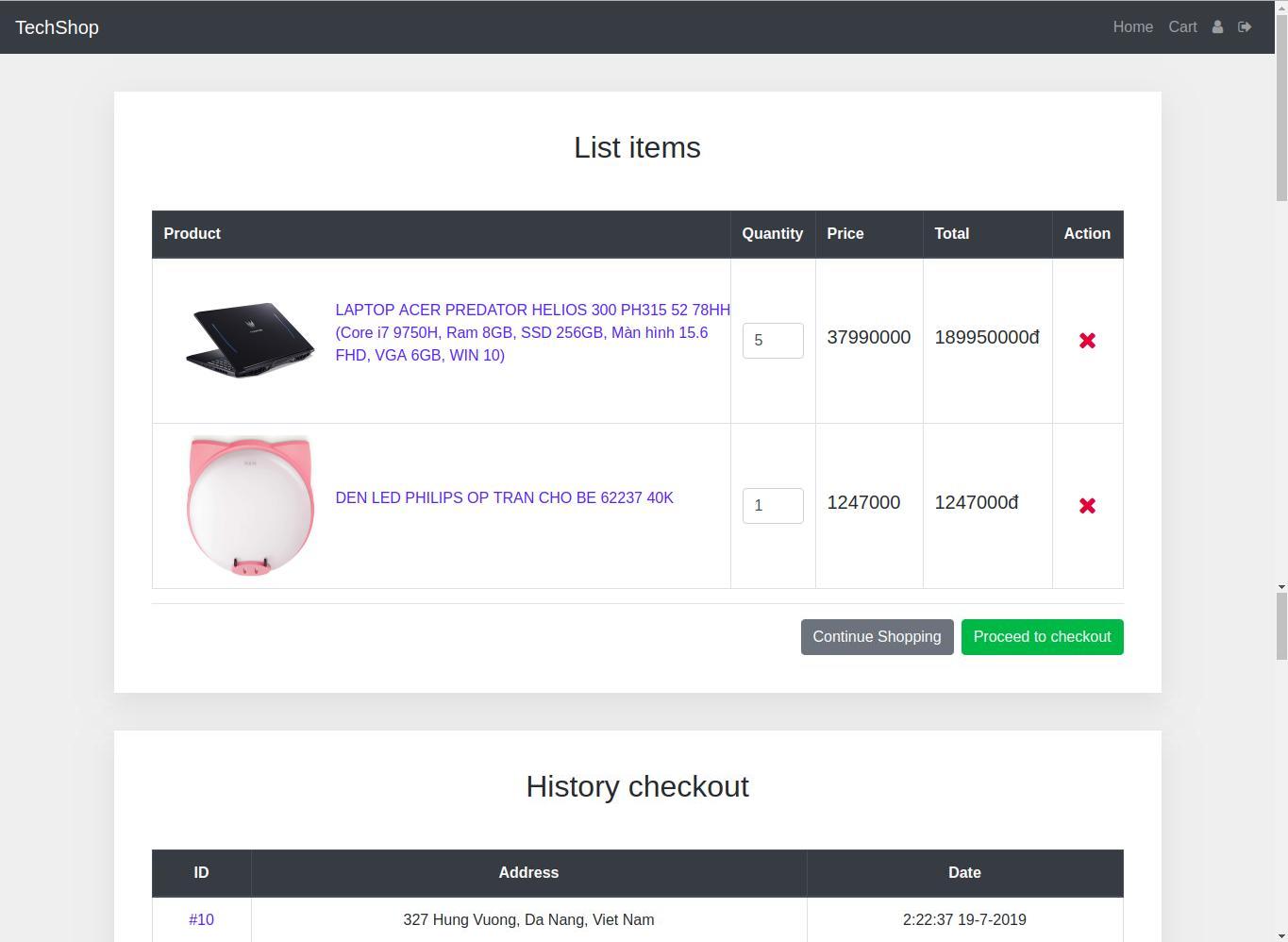


Figure 29 Check Out Page - User Manual

* Click button ‘Procced to checkout’ to redirect to bill user info to give a bill order address. In this page, if you have addresses you can choose the option ‘Have exist addresses’ or you can choose the option ‘Create new address’ and fill user info. After this step you click button ‘Next’ to redirect to Confirmation page to check bill and checkout.

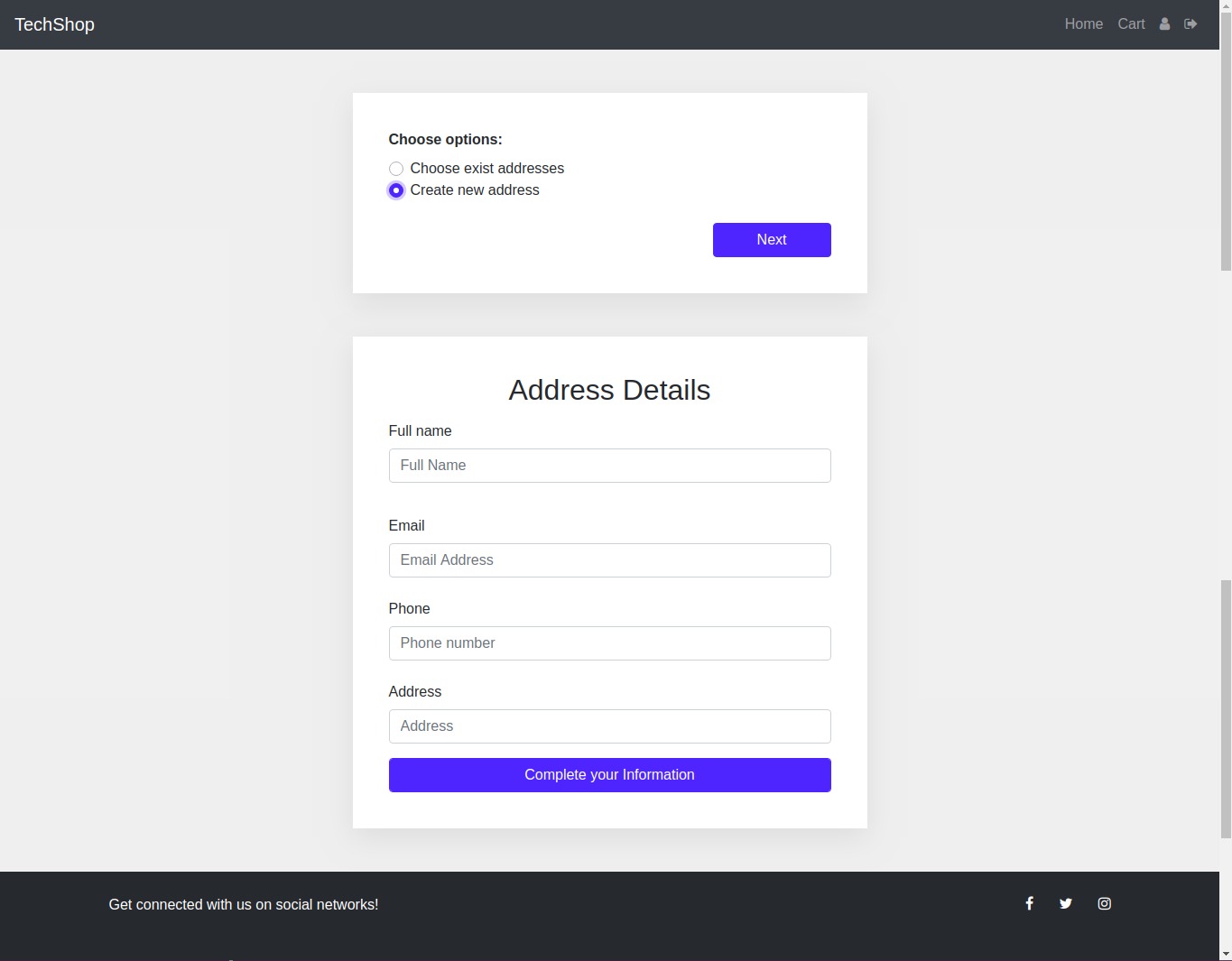


Figure 30 Check Out 2 - User Manual

* In this page, you will check the items you buy and your address. After that, you click button ‘Thanh toán’.

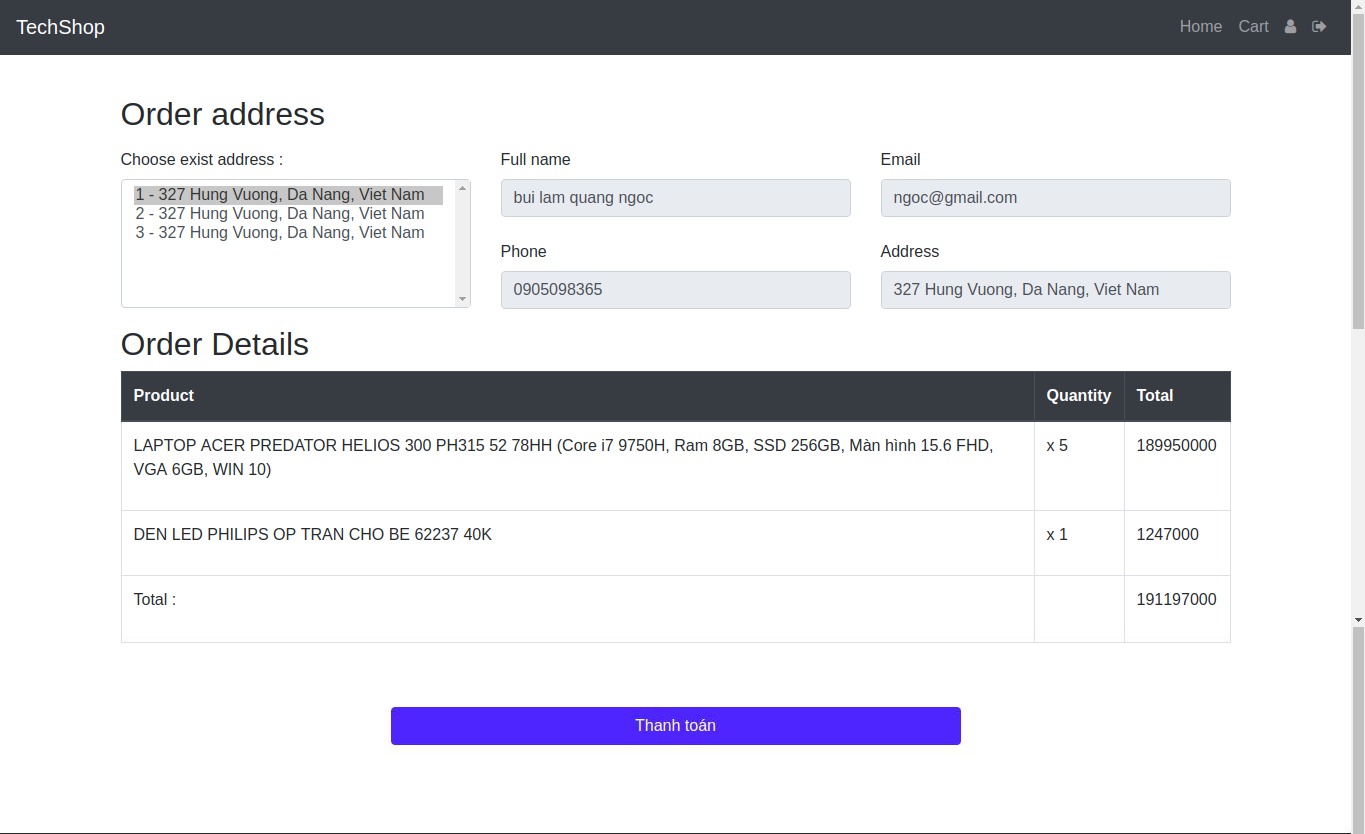
e

Figure 31 Check Out 3 - User Manual

* If you get an alert ‘Thành công’ that mean you success to buy items.

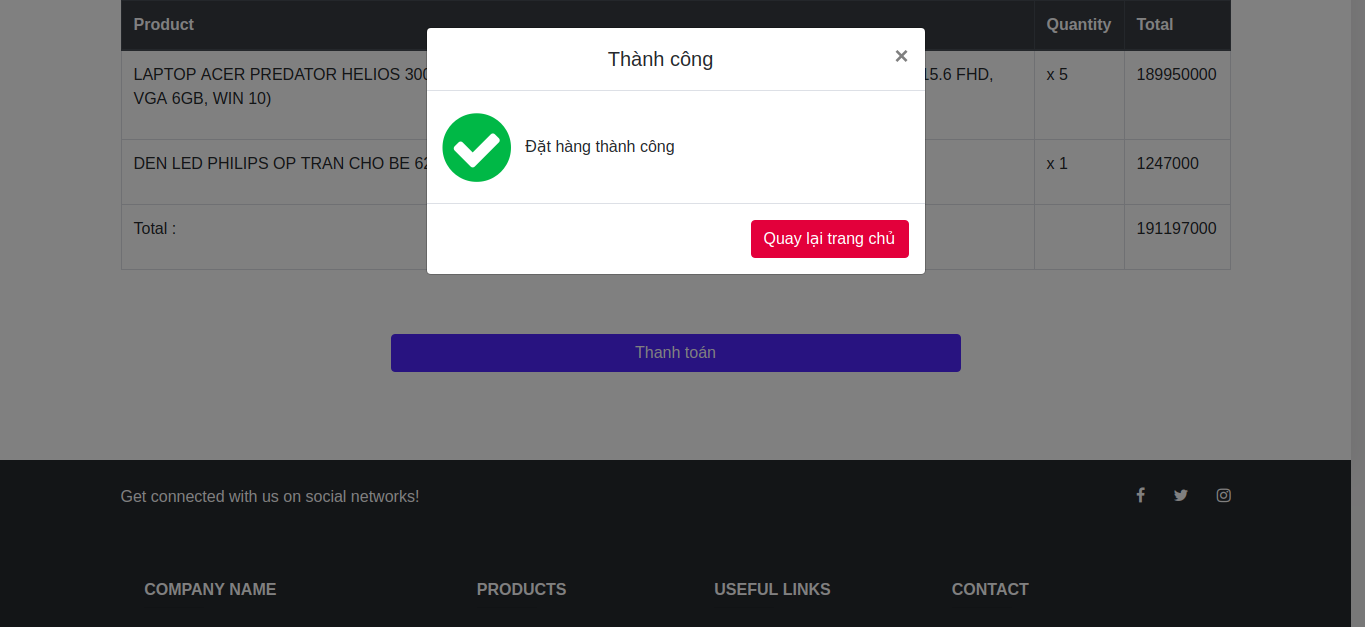


Figure 32 Pop Up Purchasing successfully - User Manual

* At this time, you can go to cart to check you order.



Figure 33 History Checkout - User Manual

* Click to 1 order to view detail order.

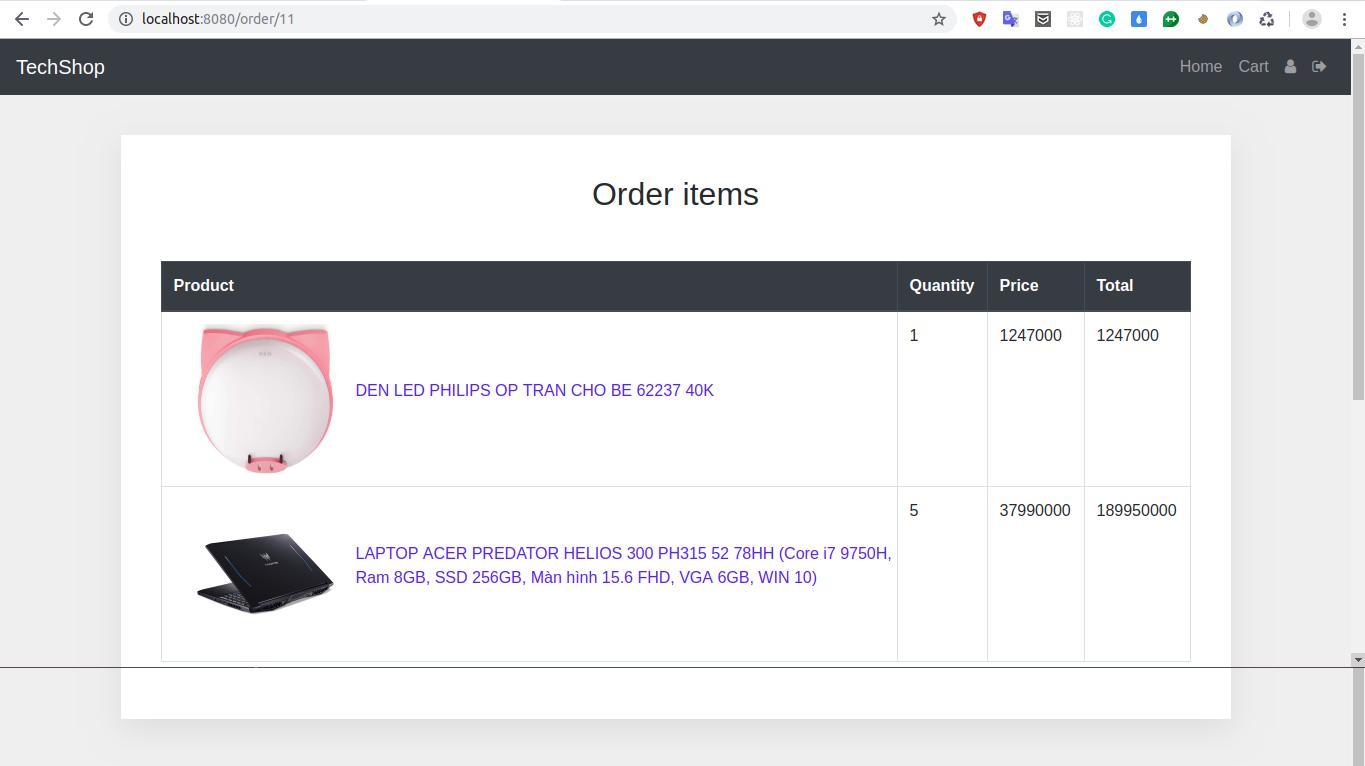


Figure 34 Detail of Items - User Manual

|  |
| --- |
| Test Plan |
| For testing of this website:  1. All the features of thee website are teste by running each function available in the website.  2. The results of the tests conducted on the website are analyzed properly. Only after getting satisfactory results of testing the website can be uploaded on the network i.e. internet. |

|  |
| --- |
| Maintenance Plan |
| For maintenance of the website:  1. The database of the website would be checked regularly.  2. Check all forms to ensure they are working properly.  3. Validate website. (CSS/HTML).  4. Consider updating the website design.  5. Safe payment gateways will be used and all the security checks will be made to ensure a safe deal.  6. Review each page of the site for content accuracy.  7. Renew website domains name.  8. Review our website strategy to align with our business goals.  9. Update core website software and plugins. |

|  |
| --- |
| Setup and Installation |
| Environment required:  Java 8 or higher  - MAVEN  - MySQL version 5.6 or higher (Remember username and password)  - Git (Source version control)  **Steps to run:**  - git clone https://github.com/quangngoc430/Final-Project  - change directory into folder Final-Project  - open cmd (windows) or terminal (ubuntu)  - run `cd backend`  - run `cd shopping`  - change username and password of database into file  '/src/main/resources/application.properties'  - `spring.datasource.username=<username database>`  - `spring.datasource.password=<password database>`  - run `mvn clean install`  - run `mvn spring-boot:run`  - open browser and enter url http://localhost:8080 |

CONCLUSION AND DISCUSSION

*The purpose and objective of online shopping website is achieved*. *Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. An online shop, e-shop, e-store, internet shop, web-shop, web-store, online store, or virtual store evokes the physical analogy of buying products or services at a bricks-and-mortar retailer or in a shopping centre. The process is called Business-to-Consumer (B2C) online shopping. When a business buys from another business it is called Business-to-Business (B2B) online shopping.*

*The design of the project which includes Data Model and Process Model illustrates how the database is built with different tables, how the data is accessed and processed from the tables. The building of the project has given me a precise knowledge about how Python is used to develop a recommend system and how it connects to the database to access the data and how the data and web pages are modified to provide the user with a shopping cart application.*

*Lesson Learnt: we learned that how to manage time effectively, how to control source code, research new engine, how to apply new system and having the experience to deal with the requirements.*

*However, our project still has some pros and cons.*

*Pros: We had recommendation system based on the behavior of user, broaden our horizon to learn deeply and researching methods.*

*Cons: Interface looked like unprofessional, some problems with UI/UX.*

*In the future, We plan that we should improve the UI/UX, improve recommendation system to run it rapidly, maintain this website for further researching and development.*

REFERENCES

*Includes all references: articles, media facts, books, reports, regulations, internet articles, papers that you referenced from the text.*

*Some examples are given below.*

(Kumari, 2016) (Wikipedia, 2016)

*[2] R. Levenshteyn, and I. Fikouras, “Mobile services interworking for IMS and XML Web Services,” IEEE Communications Magazine, pp. 80–87, Sep. 2006.*

*[3] D. Lozano, L.A. Galindo, and L. Garcia, “WIMS 2.0: Converging IMS and Web 2.0. Designing REST APIs for the exposure of session-based IMS capabilities,” in Proc. of The Second International Conference on Next Generation Mobile Applications, Services, and Technologies, 2008, pp. 18–24.*

APPENDIXES

*Type or paste your appendices here. Appendices are a place to organize and include all of the “extra” material that is important to your research work but that is too detailed for the main text.  Examples can include: specific analytical methods, computer code, spreadsheets of data, details of statistical analyses, etc. But, these materials do not speak for themselves. There should be a reference to these materials from the main chapters (complete details included in Appendix A) and there should be some text at the beginning of each appendix to briefly explain what the information is and means that is included in that appendix.*

BIOGRAPHY

NAME: DUONG DUC HUNG

ADDRESS: 177 CU CHINH LAN, DA NANG

REGISTRATION NO: 15020018

NAME: BUI LAM QUANG NGOC

ADDRESS: 28 AN THUONG, DA NANG

REGISTRATION NO: 15020019