

Online Overseas Shopping Website

Semester 2 Report

Anqi Li

20082481

Panel 10

Panelist: Ian Downey

BSc (Hons) in Software System Practice

Table of Contents

1. Abstract	1
2. Introduction	1
2.1.The Buying Subsystem	1
2.2.The Selling Subsystem.....	1
2.3.The Management Subsystem	1
3. Technologies	2
3.1.MongoDB	2
3.2.Express	2
3.3.Vue.js	2
3.4.Node.js	2
3.5.Amazon Web Services – Simple Storage Service	3
3.6.Firebase – Firebase Hosting	3
3.7.Heroku	3
3.8.PayPal	3
3.9.Google Maps	3
4. Methodology	3
5. Security	4
6. User Manual	5
6.1.System Overview	5
6.2.System Configuration	5
6.3.User Access Levels	5
6.4.Sign Up	5
6.5.Sign In	6
6.6.Sign Out	6
6.7.Edit User Profile	7
6.8.Edit User Logo	8
6.9.Edit Shipping Address	9
6.10. Find Products	9
6.11. Pay for Products	10
6.12. Check Orders	11

6.13. Check Shipping Statue	11
6.14. Manage Catalogues (only for seller).....	12
6.15. Manage Products (only for seller).....	12
6.16. Manage Classifications (only for administrator)	12
7. Modeling	13
7.1.Use Case Diagram	13
7.2.Use Cases	15
7.3.Class Diagram	24
7.4.Sequence Diagram	26
8. Development Plan	30
8.1.Overviews Plan	30
8.2.Plan and Development Each Week	31
9. Testing	42
9.1.Back End	43
9.2.Front End	45
10. Problems and Solutions	46
10.1. Validate Username and Active Account.....	46
10.2. Manage Catalogues and Products Dramatically	46
10.3. Store Images in AWS S3	47
10.4. Connect Multiple Documents in Non-Relational Database	47
10.5. Vue Component Rendering Problem	47
10.6. PayPal Payment Flow	47
10.7. Tracking Information Showing in Maps	48
10.8. Cross Side Problem	48
11. Suggested Additional Features	48
12. Summary	48
13. Appendices	49
References	49

1. Abstract

Nowadays, online shopping is more and more popular. With the development of economic globalization, the proportion of people buying overseas goods is increasing. However, many companies so far only sell products on their official websites. It takes customers much time to search and compare products from different companies. Based on this situation, this project builds an online overseas shopping website that can be used to create a large visual global supermarket. It makes buying overseas products as easy as buying goods from a physical supermarket, while it gives more choices than a physical supermarket. It also provides opportunities for companies to expand abroad, so products can be sold in the seller's country or to other countries.

2. Introduction

The online overseas shopping website is a platform for selling products all over the world, such as electronics, fashion, cosmetics and so on. This website is a responsive website. It separates three subsystems including the buying subsystem, the selling subsystem and the management subsystem. These three subsystems all have login function, and login is a prerequisite for most operations.

2.1. The Buying Subsystem

The buying subsystem allows customers to search and buy products. Products show by classification and there are two different ways to classify. One is based on type of products, another is based on areas. Before buying products, customers need to input shipping information, including full name, phone number, shipping address and postal code. In addition, the system supports the third-party payment, such as PayPal. Customers can cancel the order before payment finished. Once customers finish payment, the system starts to track the logistics information of the products and feeds back to customers by map.

2.2. The Selling Subsystem

The selling subsystem is the main place for seller to manage their products and orders. Product's information is inputted by forms. It may include classifications, catalogues, name, images, price, stock quantity and so on. Most of data is stored in the database. Images are stored in Amazon Simple Storage Service and the objects URLs are saved in the database as same as other information. Sellers can easily find a product to edit and delete because products are classified by sellers before. Besides, sellers can change the order state of waiting for delivery to shipping. It means that products are on the way to the destination and system will track the logistics records of the package and update it in the system by text in time.

2.3. The Management Subsystem

The management subsystem is used by administrators to manage the website. So far, the system only supports administrators to manage classifications by adding, editing and deleting. The system can automatically response the change of classifications to the buying subsystem and the selling subsystem. In the future, the management subsystem will support administrators to audit the business licenses which are uploaded by sellers during the registration process.

Besides, administrators should be allowed to monitor the comments and products, and promptly remove comments and products that have a negative impact on the website environment.

3. Technologies

This website uses the MEVN technology stack (MongoDB, Express, Vue.js, Node.js), which is a modern web development framework for the JavaScript platform. Some third-party platforms and plugins are applied in the system, such as Amazon Web Services, Firebase, Heroku and PayPal.

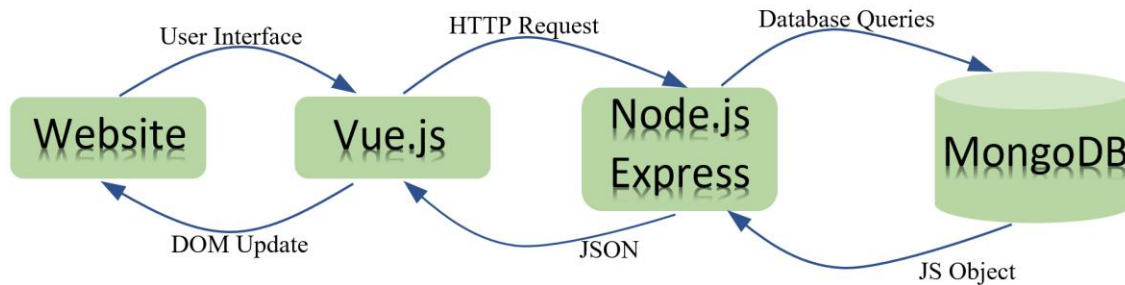


Figure 1. MEVN Stack.

3.1. MongoDB

MongoDB is document store database using JSON structure to store data, so it supports dynamic modes (周, 2018). It is suitable for a shopping website to store product's information because the product is various, and the characteristic and parameter of each kind of products are different.

3.2. Express

Express is currently the most popular web development framework based on Node.js, which can quickly build a complete website. It has some advantages, such as automatic HTTP header handling, elegant error handling, and prevention of XSS. Besides, setting up REST routing is simple. It provides a helper function to parse the POST request and support Connect middleware, which can insert additional requests or response processing (Stanton, 2018).

3.3. Vue.js

Vue.js is a progressive framework for building user interfaces. It uses the MVVM (model, view, view model) design pattern, which is a data-driven mode (Segmentfault.com, 2018). It means that everything is done by manipulating the data and trying to avoid manipulating the Dom tree.

3.4. Node.js

Node.js is a platform for quickly building web services and applications. It is a service platform built in JavaScript language and can be used to build servers on the back end. It is designed for network services using event-driven and asynchronous programming (简书, 2018).

3.5. Amazon Web Services – Simple Storage Service

Amazon Web Services is a secure cloud services platform. This project uses Amazon Simple Storage Service (S3). It is a public service that web application developers can use to store digital recourses, including images, videos, music, and documents (Baik.e.baidu.com, 2019). Besides, S3 provides a RESTful API to programmatically interact with the service.

3.6. Firebase – Firebase Hosting

Firebase is a mobile and web application development platform acquired by Google. One of the services in Firebase is Firebase Hosting which is a static and dynamic web hosting service built for the modern web developer (En.wikipedia.org, 2019). Since the rise of front-end JavaScript frameworks like Angular and Vue, websites and applications are more powerful than before. Hosting gives a subdomain for applications on the firebaseapp.com domain and help developer to deploy their project from local directories to cloud server (Firebase, 2019). Meanwhile, it provides HTTP Secure (HTTPS) and Secure Sockets Layout encryption (SSL) to deliver files.

3.7. Heroku

Heroku is a cloud service platform that supports multiple programming languages, such as Ruby, Java, Node.js, Scala, Clojure, Python, etc. (Baik.e.baidu.com, 2019). It uses platform as a Service which allows customers to develop, run, and manage application without the complexity of building and maintaining the infrastructure (Devcenter.heroku.com, 2019). For example, Heroku integrates with GitHub to make it easy to deploy code living on GitHub to apps running on Heroku.

3.8. PayPal

PayPal is currently the world's largest online payment processor. More than 90% of sellers and more than 85% of buyers in cross-border transactions are using PayPal electronic payment services (Wiki.mbalib.com, 2019). PayPal uses advanced commercial encryption technology to ensure that financial information is stored securely and not disclosed to buyers or sellers. It is a fast, convenient, and secure electronic payment tool.

3.9. Google Maps

Google Maps is a web mapping service which offers satellite imagery, aerial photography, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions, and route planning for traveling by foot, car, bicycle and air (in beta), or public transportation (En.wikipedia.org, 2019). The project uses the Maps JavaScript API supported various services and libraries for developers to customize maps (Google Developers, 2019).

4. Methodology

This project uses Kanban which is an easy way to deliver results in a timely manner and to keep team members from overloading. Its goal is to provide maximum value at the end of the iteration (Ke, 2018). When using Kanban in software development, it usually includes the phases and tasks of the work. There are many cards on the Kanban. These cards are placed in

columns, with a number above each column. The card represents the work item and the card flows from the first step to the last step. The column represents the development process and the number at the top of each column is used to limit the maximum number of cards allowed per column. Since the number of work-in-progress (WIP) is limited, bottlenecks may appear in the early stages of the iteration. When there is a bottleneck or overload, it can be solved by limiting WIP.

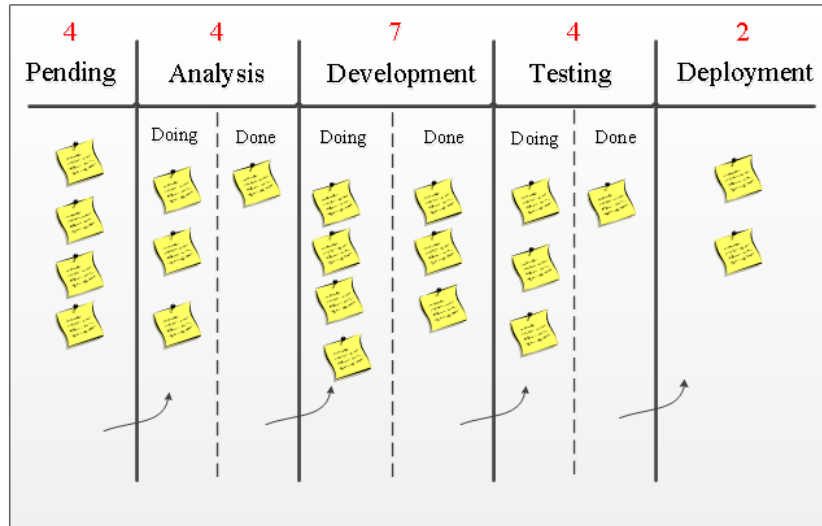


Figure 2. Kanban Example.

The specific use of Kanban will be shown in the development plan, please read the Section 8.

5. Security

In terms of error messages, only the minimum amount of error information is provided to the user to ensure that they do not reveal confidentiality present on the server. Providing incomplete exception details can make complex attacks (such as SQL injection) difficult (Gerber and Perry, 2018).

The system uses verification by both sides, which is verified on the browser and the server side. Not only simple faults are captured on the browser side, such as empty required fields, but system also check for these validations on the server side to prevent malicious code or script code from being inserted into the database.

Authentication is a process of determining that a user is who they claim to be. The system uses two-factor (email and password) authentication. Password in the system must be eight to sixteen long including uppercase letters, lowercase letters, numbers and special characters. A strong password can effectively prevent brute force cracking. Passwords are always stored as encrypted values in the database, such as hashes. When the system authenticates the user, only the encrypted value is compared.

This system uses cookie-based authorization to determines that the current user has access to system resources. Cookies are set security to limit it only for HTTPS so that the system can prevent cookies from being modified. What is more, cookies are signed with special keys, so

the system can detect tampering cookies and stop unauthorized users from continuing to access resources.

The whole system deploys to the cloud service platforms which use HTTP Secure (HTTPS) and Secure Sockets Layout encryption (SSL). The front-end is deployed to Firebase and the back-end is deployed to Heroku, so the website can prevent an attacker from intercepting or changing the content of a transmission.

6. User Manual

6.1. System Overview

Online Overseas Shopping Website is an application, which allows sellers to sell their products and customers to buy products online. The application operates on the browser.

6.2. System Configuration

Online Overseas Shopping Website supports the following browsers: Internet Explorer 11, Firefox (versions 55 and up), Chrome (versions 60 and up) and Safari on Mac (versions 10 and up).

6.3. User Access Levels

Every visitor can register and browse products in the website, but only logged in users are able to do other operations.

6.4. Sign Up

1. Click the 'Sign up' button in the toolbar.
2. Fill your account name, email address and password. (Account name and email address should be unique, Password should be eight to sixteen long including a number, a lowercase letter, a capital letter and a special character)
3. Choose the role of account.
4. Click the 'Submit' button to create an account.

The screenshot shows the 'Sign Up' form in the 'Overseas Shopping' website. The form is titled 'Sign Up' and contains three input fields: 'Name' (with the value 'Shopping'), 'Email Address' (with the value 'shopping@gmail.com'), and 'Password' (with a hint: 'At least a number, lowercase letter, capital letter and special character'). Below the input fields are two radio buttons: 'Customer' (selected) and 'Seller'. At the bottom of the form is a 'SUBMIT' button. Red boxes and numbers 1 through 4 are used to highlight the 'SIGN UP' button in the toolbar, the form fields, the role selection, and the 'SUBMIT' button, respectively.

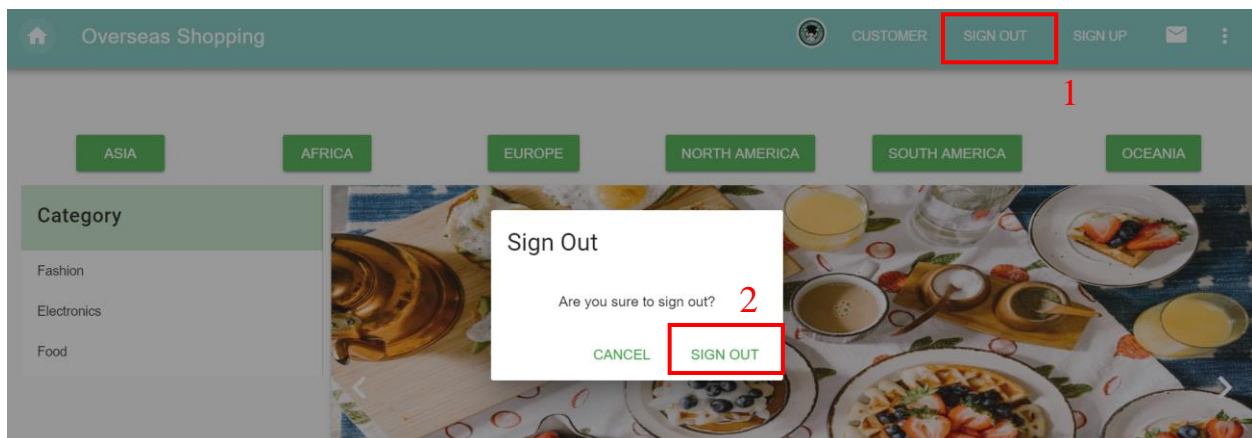
6.5. Sign In

1. Click the 'Sign in' button in the toolbar.
2. Fill your email address and password.
3. Choose the role of account.
4. Click the 'Submit' button to login.

The screenshot shows the 'Sign In' page of the 'Overseas Shopping' application. The top navigation bar is teal and contains a home icon, the text 'Overseas Shopping', a user icon, 'WELCOME', a 'SIGN IN' button (highlighted with a red box and labeled '1'), a 'SIGN UP' button, a shopping cart icon, an envelope icon, and a menu icon. The main content area is white and titled 'Sign In'. It contains two input fields: 'E-mail' and 'Password' (highlighted with a red box and labeled '2'). Below the password field is a character count '0 / 30'. Under the input fields are three radio buttons for account roles: 'Customer' (selected, highlighted with a red box and labeled '3'), 'Seller', and 'Administrator'. At the bottom is a 'SUBMIT' button (highlighted with a red box and labeled '4').

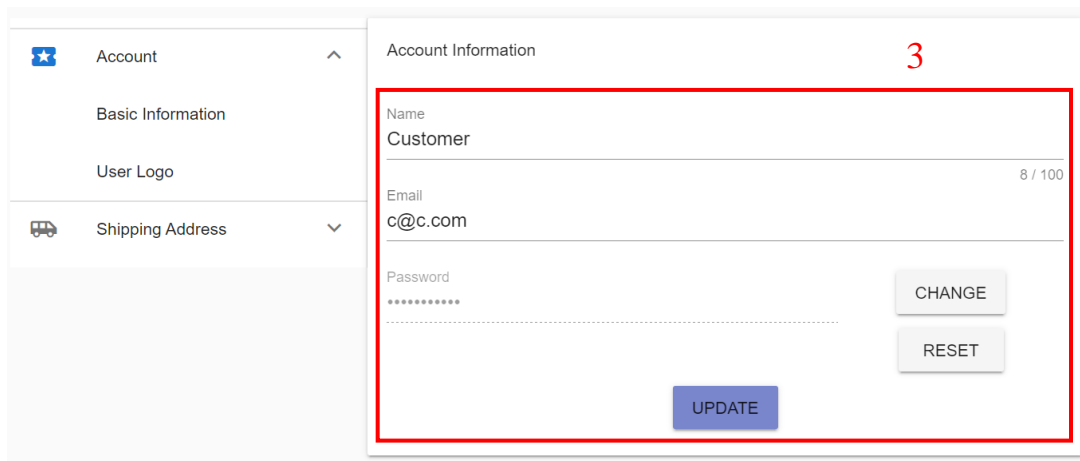
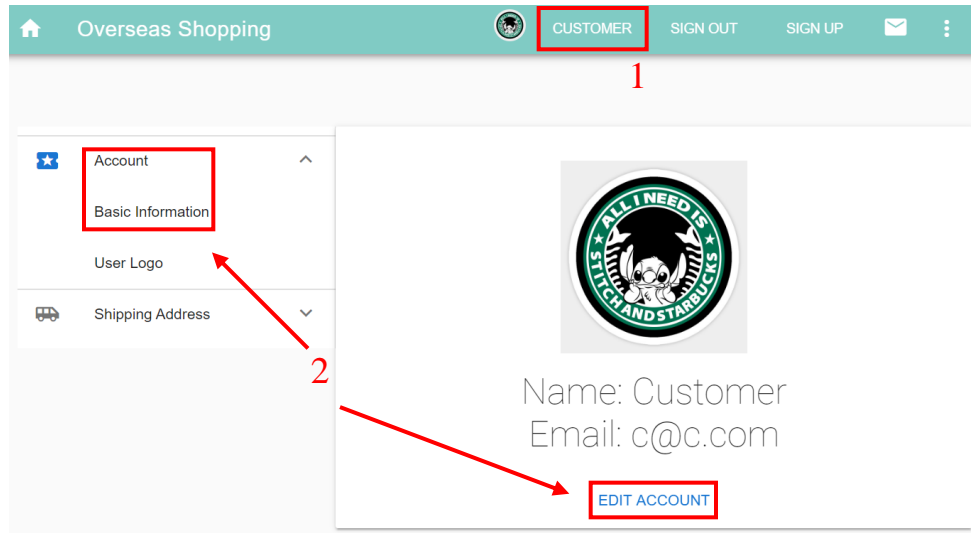
6.6. Sign Out

1. Click the 'Sign out' button in the toolbar.
2. Click the 'Sign out' button in the dialog to logout.



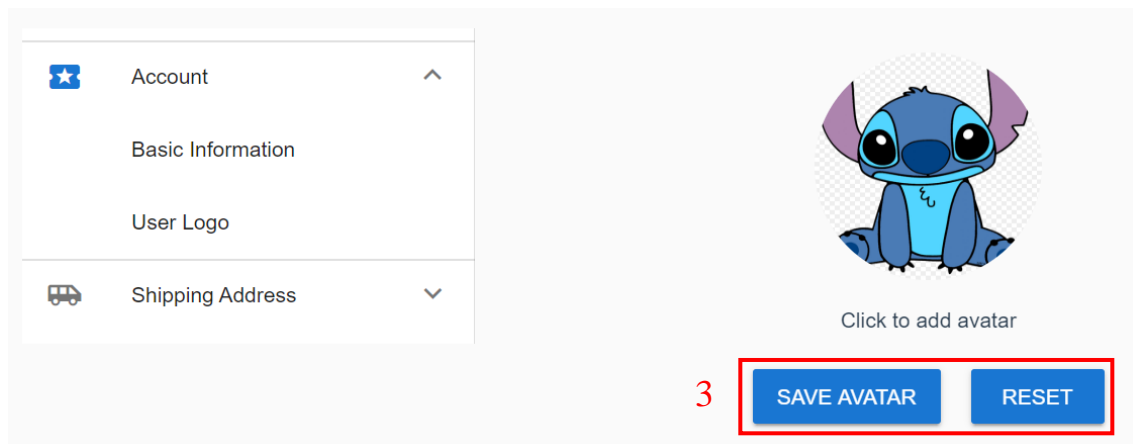
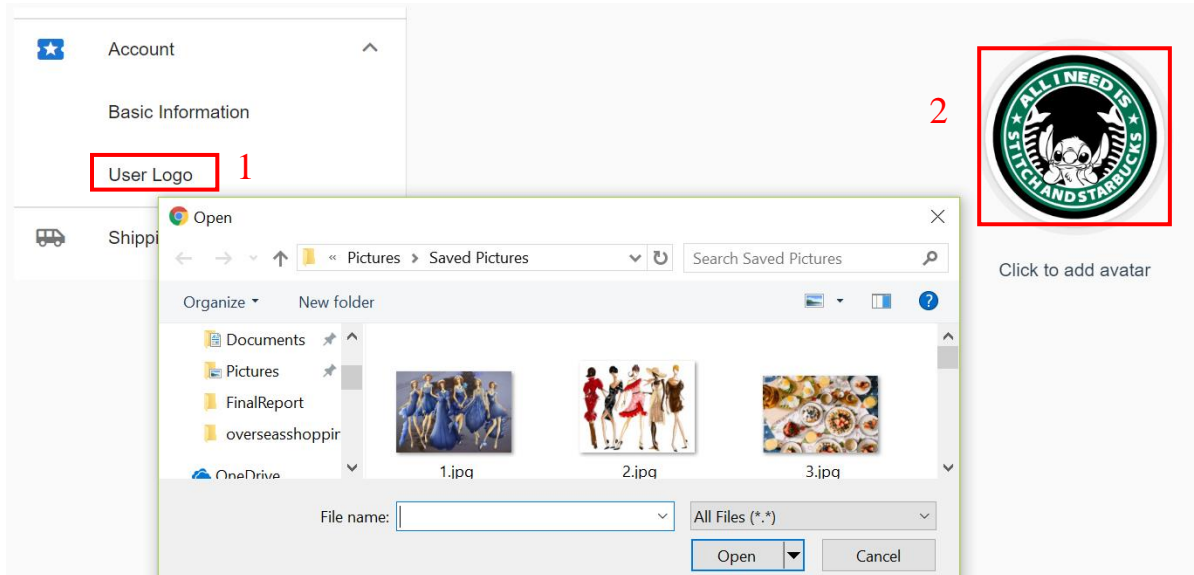
6.7. Edit User Profile

- * You should login your account first. (See in the Section 6.5)
- 1. Click the account name in the toolbar.
- 2. Click the 'Edit Account' button to enter the editing state, or click the 'Account' button, then click the 'Basic Information' to enter the editing state.
- 3. Fill your basic information and click the 'Update' button.



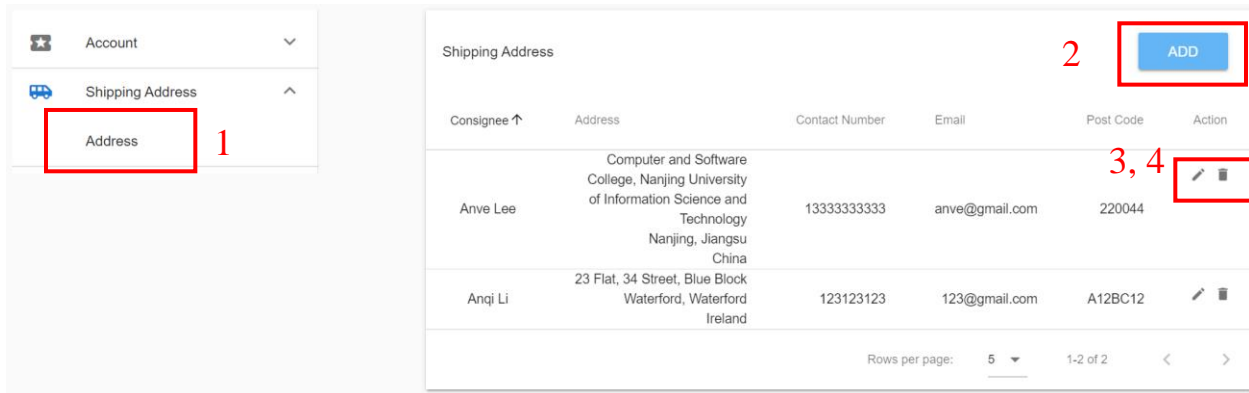
6.8. Edit User Logo

1. Click the 'User Logo' button in the Account Menu.
2. Click your user logo and choose an image.
3. Click the 'Save' button to update logo or the 'Reset' button to cancel editing.



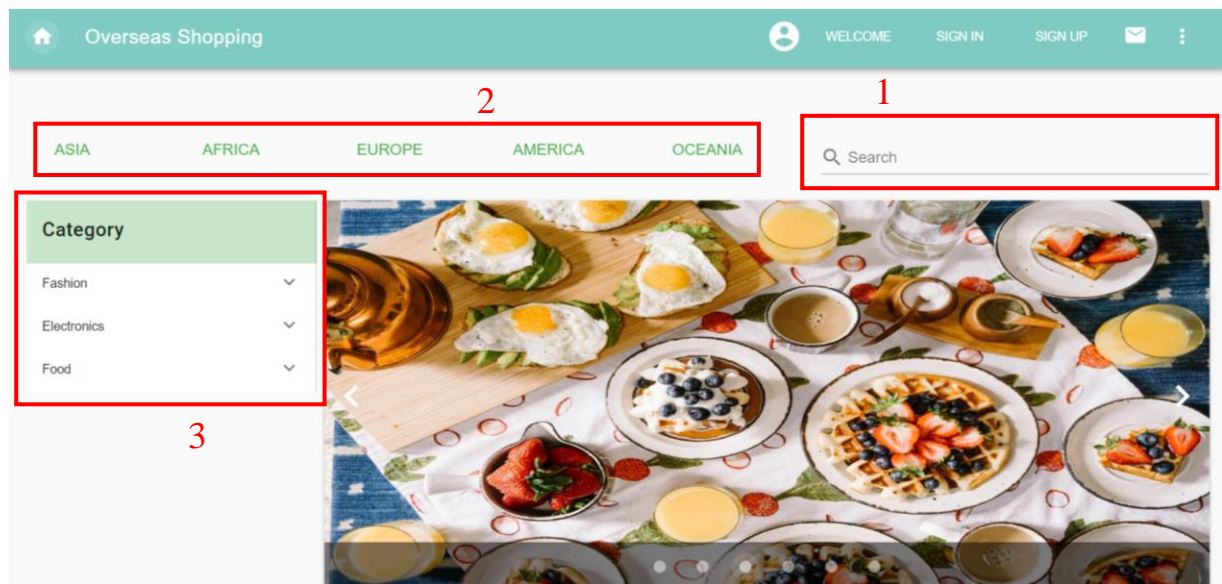
6.9. Edit Shipping Address

1. Click the 'Address' button in the Shipping Address Menu.
2. Click the 'Add' button and fill the details to create a new shipping address.
3. Click the 'Edit' icon to modify a exist shipping address.
4. Click the 'Delete' icon to remove a exist shipping address.



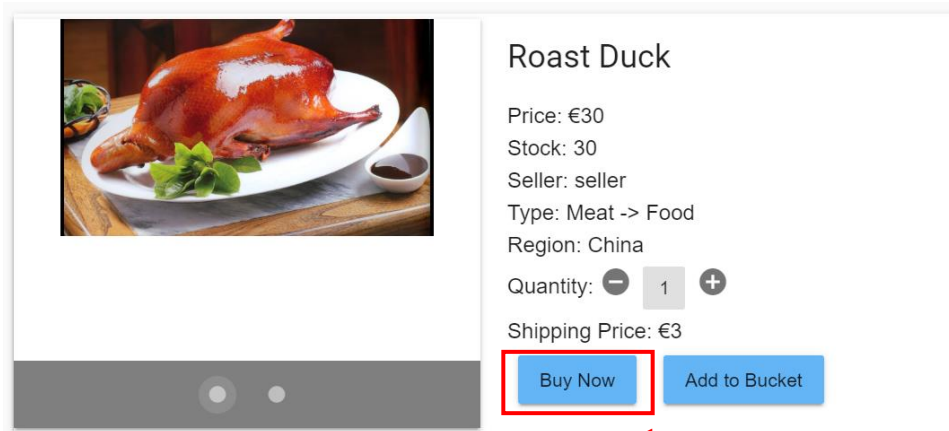
6.10. Find Products


There are three ways to find products, using the search function, using the region classification and using the catalogues.



6.11. Pay for Products

1. Click the 'Buy Now' button on the Product Page.
2. Fill the shipping address.
3. Click the 'Confirm Order' button.
4. Click the 'PayPal' button to pay for the products.

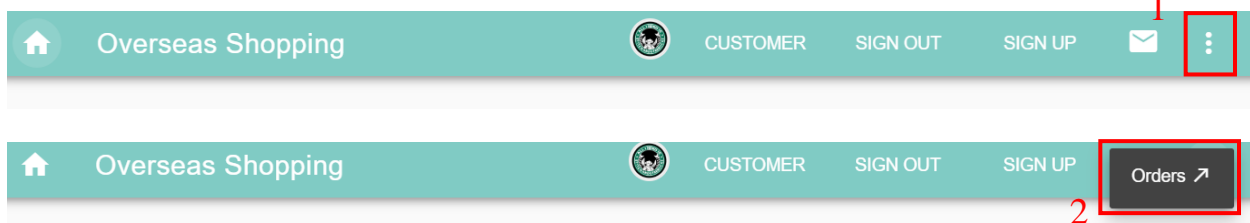


Billing		Payment	
	Roast Duck	€30	
Quantity	1	<div>Shipping address</div> <div>First nameLast name</div> <div>Address</div> <div>CountryProvince/Territory/County/S...City/TownPost Code</div> <div>Contact Number</div>	
Type	Meat		
Region	China		
Subtotal	30		
Shipping	3		
Total	€33	<div>Confirm order</div>	

Total	€19.99
<div>Contact Number</div> <div>35388666666</div> <div>PayPal</div>	

6.12. Check Orders

1. Click the menu icon in the toolbar.
2. Select the 'Orders' or 'Order List' item.



Order List

6.13. Check Shipping Statue

1. Click the order in the order list table to enter the Shipping page (See in the Session 6.12).
2. Choose the order you want to track.

The screenshot shows the 'Order List' page. At the top, the same teal navigation bar is visible, with the menu icon labeled with a red '1'. Below the navigation bar, the 'Order List' title is centered. A table with one row is highlighted with a red box and labeled with a red '2'. The table has columns for Image, Name, Price, Payment, Shipping, and Date. The row contains an image of honey pork, the name 'Honey Pork', a price of '€ 10', a payment status of 'completed', a shipping status of 'delivering', and a date of '2019-04-10'. Below the table, a detailed view of the order is shown, also highlighted with a red box and labeled with a red '3'. This view is divided into three sections: 'Order Information', 'Shipping Information', and 'Tracking Information'. The 'Order Information' section includes details like Customer, Product, Payment, Tracking Number, and Shipping Statue. The 'Shipping Information' section includes First Name, Last Name, Address, City, Destination Country, and Phone Number. The 'Tracking Information' section includes a green checkmark icon, a date and time '2019-02-28 15:24', and the location 'WATERFORD,WD036,IE' with the status 'Statue: Delivered'.

Image	Name	Price ↑	Payment	Shipping	Date
	Honey Pork	€ 10	completed	delivering	2019-04-10

Order Information

Customer: Customer
Product: Honey Pork
Payment: PAYID-LSXEO2Q7SD68391XN604662R
Tracking Number: 1Z30R55F6806450645
Shipping Statue: delivering

Quantity: 1
Payment Statue: completed
Carrier: UPS

Shipping Information

First Name: Anqi
Address: 99 Lake
City: Waterford
Destination Country: Ireland
Phone Number: 35388666666

Last Name: Li
Province/Territory/County/State: Waterford
Postal Code: X9QWP
Email: shopping-seller@gmail.com

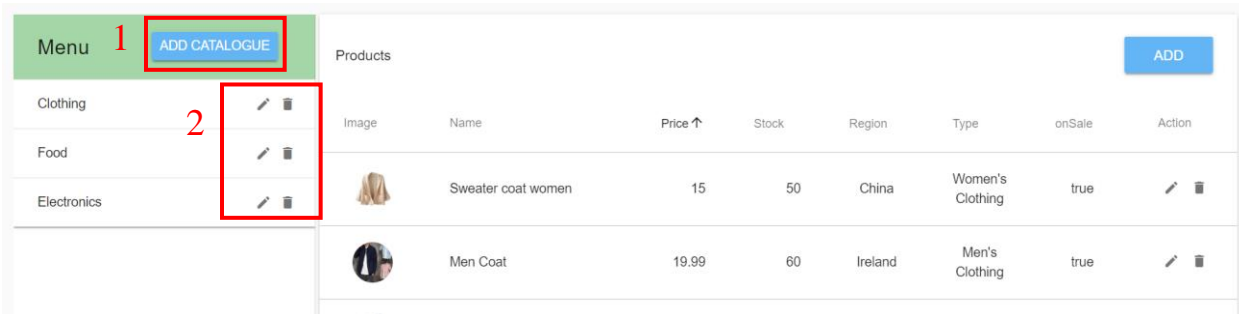
Tracking Information

You can also user the [Trackingmore](#) website for tracking.

2019-02-28 15:24 WATERFORD,WD036,IE
Statue: Delivered

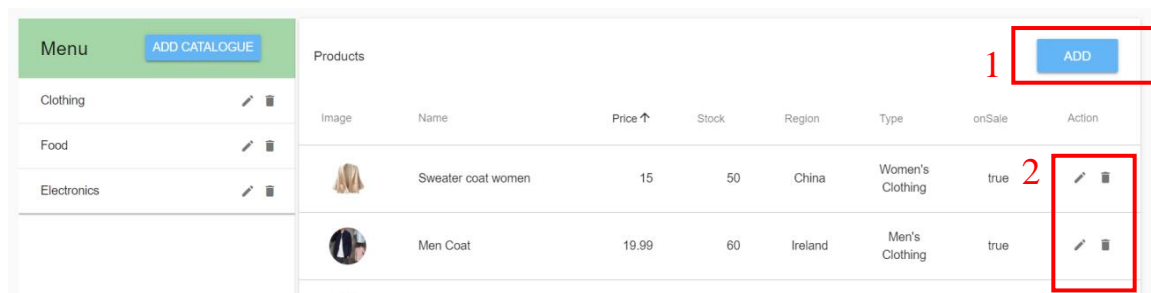
6.14. Manage Catalogues (only for seller)

3. Click the 'Add Catalogue' button in the menu on Seller Home page to create a new catalogue.
4. Click the 'Edit' or 'Delete' icon to modify catalogues.



6.15. Manage Products (only for seller)

1. Click the 'Add' button in the data table to create a new product.
2. Click the 'Edit' or 'Delete' icon to modify products.



6.16. Manage Classifications (only for administrator)

1. Click the 'Add Catalogue' button in the menu on Administrator Home page to create a new catalogue.
2. Click the 'Edit' or 'Delete' icon to modify catalogues.

★

Classification

Region

Category

Type

1

ADD

↑ Title

Subtitle

Active

Action

Electronics

Computer & Tablet

true

2

Electronics

Cell phone & Accessories

true

Fashion

Women's Clothing

true

Fashion

Men's Clothing

true

Food

Meat

true

Rows per page:5

1-5 of 6

7. Modeling

7.1. Use Case Diagram

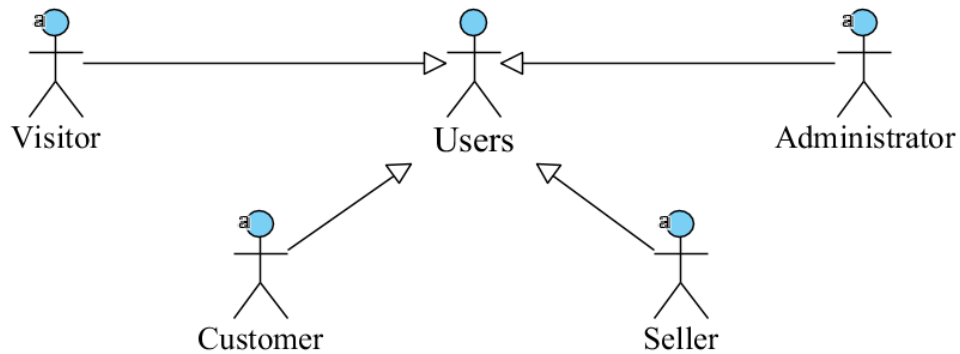


Figure 3. User Case Diagram of Actors.

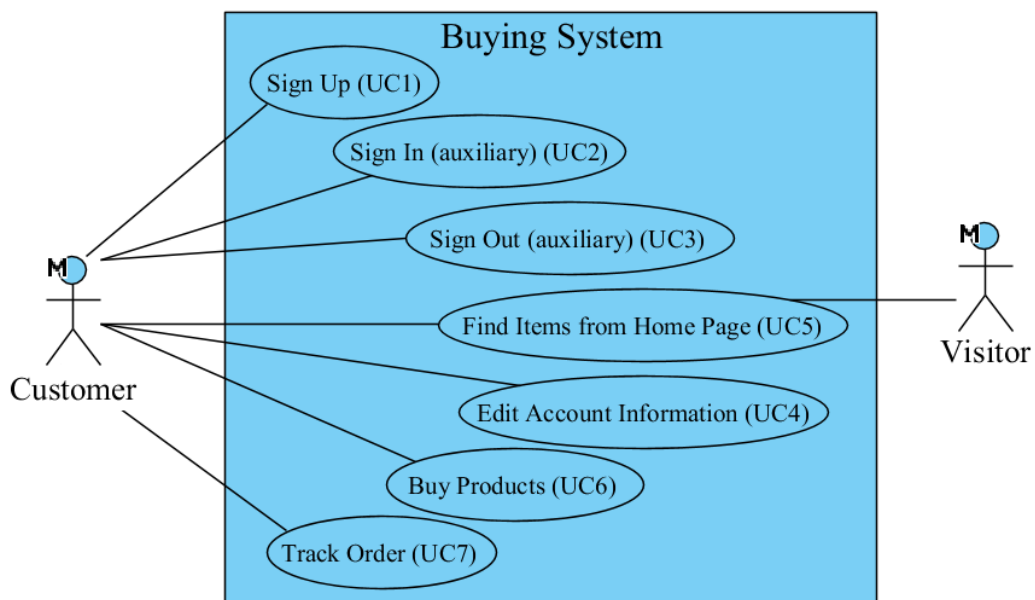


Figure 4. Use Case Diagram of Buying System.

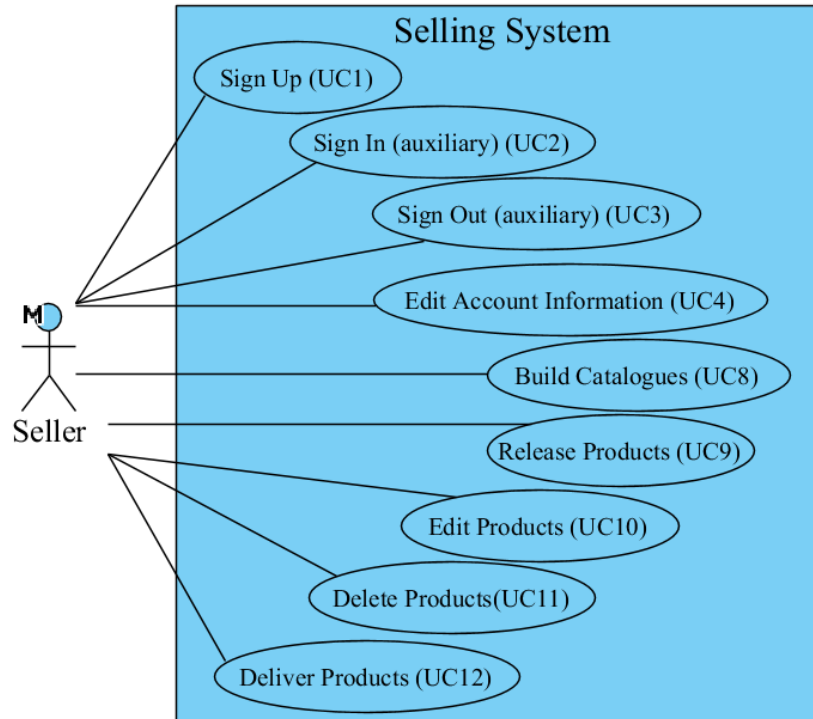


Figure 5. Use Case Diagram of Selling System.

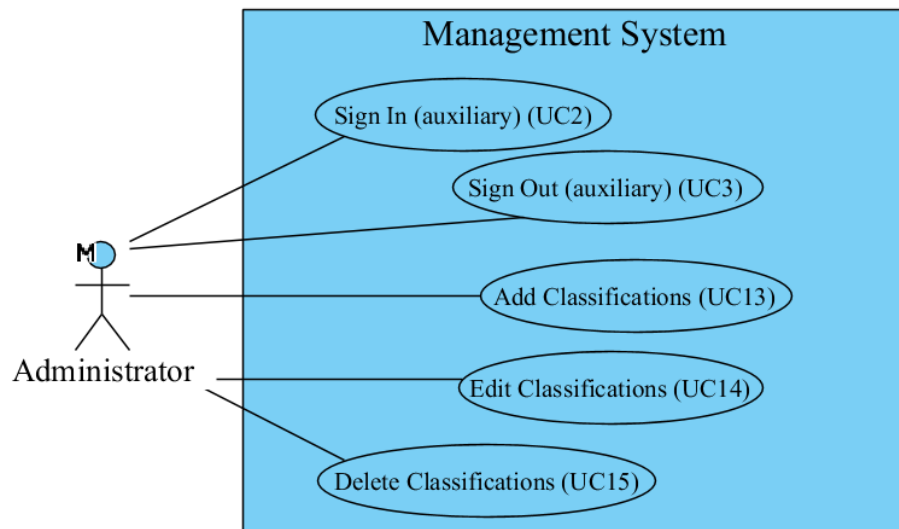


Figure 6. User Case Diagram of Management System.

7.2. Use Cases

Use case 1: Sign Up

Actor: Customer, Seller.

Scenario 1: Sign up.

1. User clicks the signup button.
2. System pop-ups sign up box.
3. User inputs account information, including account name, email address and password.
4. User chooses account type (customer or seller) and clicks the button to create.
5. System sends an activation link to user's email box.
6. User follows the activation link to finish sign up.
7. System creates a new active account successfully.
8. System redirects to sign in.

Scenario 2: As above (1 to 8) but with the following modifications:

Alternative: Replicate name or email address.

At step 3, the name or email address entered by the user already exist in the system.

Allows user to re-input account name or email address.

Scenario 3: As above (1 to 8) but with the following modifications:

Alternative: The format of account information is incorreced.

At step 3, the format of account information entered by the user is incorreced.

Allows user to re-input account information.

Scenario 4: As above (1 to 8) but with the following modifications:

Alternative: Send activation code failure.

At step 5, System fails to send activation code.

Allows user to re-click the create button and get activation code again.

Scenario 5: As above (1 to 8) but with the following modifications:

Alternative: Create a new active account failure.

At step 7, system fails to create a new account.

Allows user to re-create and repeat step 6.

Scenario 6: As above (1 to 8) but with the following modifications:

Alternative: Redirect failure.

At step 8, system fails to redirect to home page.

Allows user to click a button to come back to home page.

Use case 2: Sign In (auxiliary use case)

Actor: Customer, Seller, Administrator.

Scenario 1: Sign in.

1. User clicks the sign in button.
2. System pop-ups sign in box.
3. User inputs username, password and selects account type (customer, seller and administrator).
4. User clicks the enter button to sign in.
5. System matches username and password and sign in successfully.
6. System redirects to the home page.

Scenario 2: As above (1 to 6) but with the following modifications:

Alternative: Match failure.

At step 3, system fails to match username and password in the database.

Allows user to re-input and sign in again.

Scenario 3: As above (1 to 6) but with the following modifications:

Alternative: Redirect failure.

At step 6, system fails to redirect to home page.

Allows user to click a button to come back to home page.

Use case 3: Sign Out (auxiliary use case)

Actor: Customer, Seller, Administrator.

Scenario 1: Sign out account.

1. User is in the status of sign in (can be customer, seller and administration).
2. User clicks the sign out button.
3. System release the resource of user and close the session.
4. System redirects to the sign in page.

Use case 4: Edit Account Information

Actor: Customer, Seller.

Pre-condition: The user is logged in.

Scenario 1: Edit account information.

1. User clicks edit account information button.
2. System jumps to editing page.
3. User changes basic information (name, password, logo etc.).
4. System checks the format of information.
5. User submits the information.
6. System updates information to database.

Scenario 2: As above (1 to 6) but with the following modifications:

Alternative: Check information failure.

At step 4, information does not meet the policy.

Allows user to re-input information again.

Scenario 3: As above (1 to 6) but with the following modifications:

Alternative: Update failure.

At step 6, system fails to update information.

Allows user to re-submit.

Use case 5: Find Items from Home Page

Actor: Customer, Visitor.

Scenario 1: Find several products.

1. User browses products on the home page.
2. User chooses products by catalogues.
3. System displays all products of the catalogue.
4. User inputs fuzzy words in search bar.
5. System displays all products satisfied that words.
6. User finds the product of interest and go to the detail page.
7. System shows the details of products.

Scenario 2: As above (1 to 11) but with the following modifications:

Alternative: Change the order of condition setting.

- A. At step 2.1, Customer chooses product by area catalogue.

At step 3.1, System displays all products in this region.

B. At step 2.1, Customer chooses products by type catalogue.

At step 3.1, System displays all products of this type in this region.

C. Skip step 2 and step 3.

Allows customer directly uses fuzzy search to find products.

Use case 6: Buy Products

Actor: Customer.

Pre-condition: The user is logged in.

Scenario 1: Purchase several products:

1. Customer browses and select products to buy.
2. System jumps to order product page.
3. Customer fills in shipping address form and confirms the order.
4. System presents the PayPal payment button.
5. Customer clicks the payment button.
6. System directs to the third-party payment platform (PayPal).
7. Customer finishes the payment.
8. System sends successful notification to customer.

Scenario 2: As above (1 to 8) but with the following modifications:

Alternative: Direct failure.

At step 6, system fails to direct to the third-party payment platform.

Allows customer to go back and reorder.

Scenario 3: As above (1 to 8) but with the following modifications:

Alternative: Cancel payment.

At step 8, system terminate the payment process.

Return to primary scenario1 at step 2.

Use case 7: Track Order

Actor: Customer.

Pre-condition: The user is logged in.

Scenario 1: Track orders:

1. Customer clicks the orders list.
2. System redirects to the orders data table.

3. Customer selects an order.
4. System opens the tracking information.

Scenario 2: As above (1 to 4) but with the following modifications:

Alternative: Redirect failure.

At step 2, system cannot direct to the orders data table.

Allows customer to refresh the page or go back.

Scenario 3: As above (1 to 4) but with the following modifications:

Alternative: Open tracking information failure.

At step 4, system cannot show the tracking information.

Allows customer to refresh the page or go back.

Use case 8: Build Catalogues

Actor: Seller.

Pre-condition: The user is logged in.

Scenario 1: Build a catalogue.

1. Seller goes to the Seller Home.
2. Seller clicks the add catalogue button.
3. System pop-ups build box.
4. Seller inputs catalogue name.
5. Seller submits the catalogue.
6. System adds the catalogue.

Scenario 2: As above (1 to 6) but with the following modifications:

Alternative: Pop-ups build box failure.

At step 3, system fails to pop-ups build box.

Allows seller to refresh the page.

Scenario 3: As above (1 to 6) but with the following modifications:

Alternative: Add catalogue failure.

At step 6, system fails to add catalogue.

Allows seller to re-submit catalogue.

Use case 9: Release Products

Actor: Seller.

Pre-condition: The user is logged in.

Scenario 1: Release products on the Seller Home.

1. Seller goes to the Seller Home and clicks the add button.
2. Seller chooses a catalogue and clicks the add product button.
3. System redirects to the add products form.
4. Seller fills the product form.
5. Seller submits product's information.
6. System updates product to the database and goes back to the Seller Home.

Scenario 2: As above (1 to 6) but with the following modifications:

Alternative: No conforming to format.

At step 4, information does not meet the conditions.

Allows user to re-input information again.

Scenario 3: As above (1 to 6) but with the following modifications:

Alternative: Update failure.

At step 6, system fails to update products.

Allows seller to re-submit products.

Use case 10: Edit Products

Actor: Seller.

Pre-condition: The user is logged in.

Scenario 1: Edit products.

1. Seller browses products on Seller Home page.
2. Seller selects a product to edit.
3. System goes to the edition page.
4. Seller changes information of products and save it.
5. System updates product to the database and redirects to the Seller Home.

Scenario 2: As above (1 to 5) but with the following modifications:

Alternative: Go to edition page failure.

At step 3, system cannot go to the edition page.

Allows user to refresh the page.

Scenario 3: As above (1 to 5) but with the following modifications:

Alternative: No conforming to format.

At step 4, information does not meet the conditions.

Allows user to re-input information again.

Scenario 4: As above (1 to 5) but with the following modifications:

Alternative: Update failure.

At step 5, system fails to update products.

Allows seller to re-submit products.

Use case 11: Delete Products

Actor: Seller.

Pre-condition: The user is logged in.

Scenario 1: Delete products.

1. Seller finds products by catalogue.
2. Seller selects products to delete.
3. System removes products from the database.

Scenario 2: As above (1 to 3) but with the following modifications:

Alternative: Remove failure.

At step 3, system fails to remove products.

Allows seller to retry.

Use case 12: Deliver Products

Actor: Seller.

Pre-condition: The user is logged in.

Scenario 1: Delivery products.

1. Seller goes to the orders list.
2. System displays all orders.
3. Seller selects an order.
4. System directs to the transaction page.
5. Seller inputs the track number and selects carries.
6. System updates the data.

Scenario 2: As above (1 to 6) but with the following modifications:

Alternative: Update failure.

At step 6, system fails to update data.

Allows seller to retry.

Use case 13: Add Classifications

Actor: Administrator.

Pre-condition: The user is logged in.

Scenario 1: Add classifications.

1. Administrator goes to the Admin Home.
2. Administrator clicks the classification menu and add button.
3. System shows classification form.
4. Administrator chooses the property (type or area) and fills the title and subtitle of classification.
5. System updates classification to the database and closes the classification form.

Scenario 2: As above (1 to 5) but with the following modifications:

Alternative: Update failure.

At step 5, system fails to update data.

Allows administrator to retry.

Use case 14: Edit Classifications

Actor: Administrator.

Pre-condition: The user is logged in.

Scenario 1: Edit classification.

1. Administrator goes to the Admin Home.
2. Administrator clicks the classification menu and chooses an item to edit.
3. System shows that classification information.
4. Administrator edits the classification and saves it.
5. System updates the classification to the database and closes the classification form.

Scenario 2: As above (1 to 5) but with the following modifications:

Alternative: Update failure.

At step 5, system fails to update data.

Allows administrator to retry.

Use case 15: Delete Classifications

Actor: Administrator.

Pre-condition: The user is logged in.

Scenario 1: Delete classification.

1. Administrator goes to the Admin Home.
2. Administrator clicks the classification menu and chooses an item to delete.
3. System pop-ups the delete dialog.
4. Administrator clicks the delete button.
5. System updates the classification to the database and close the delete dialog.

Scenario 2: As above (1 to 5) but with the following modifications:

Alternative: Cancel delete.

At step 4, administrator cancel the delete operation.

Allows administrator to click the cancel button in the delete dialog.

Scenario 3: As above (1 to 5) but with the following modifications:

Alternative: Update failure.

At step 5, system fails to update data.

Allows administrator to retry.

7.3. Class Diagram

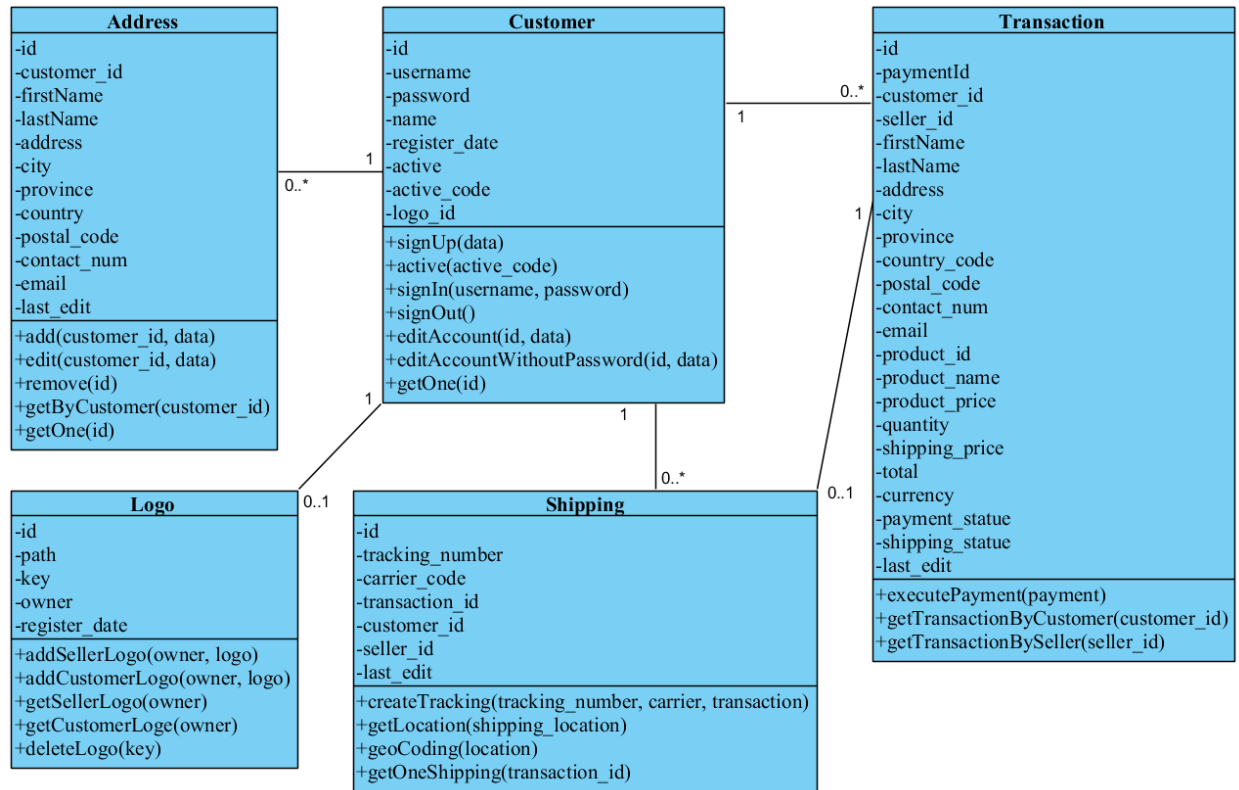


Figure 7. Class Diagram of The Buying System.

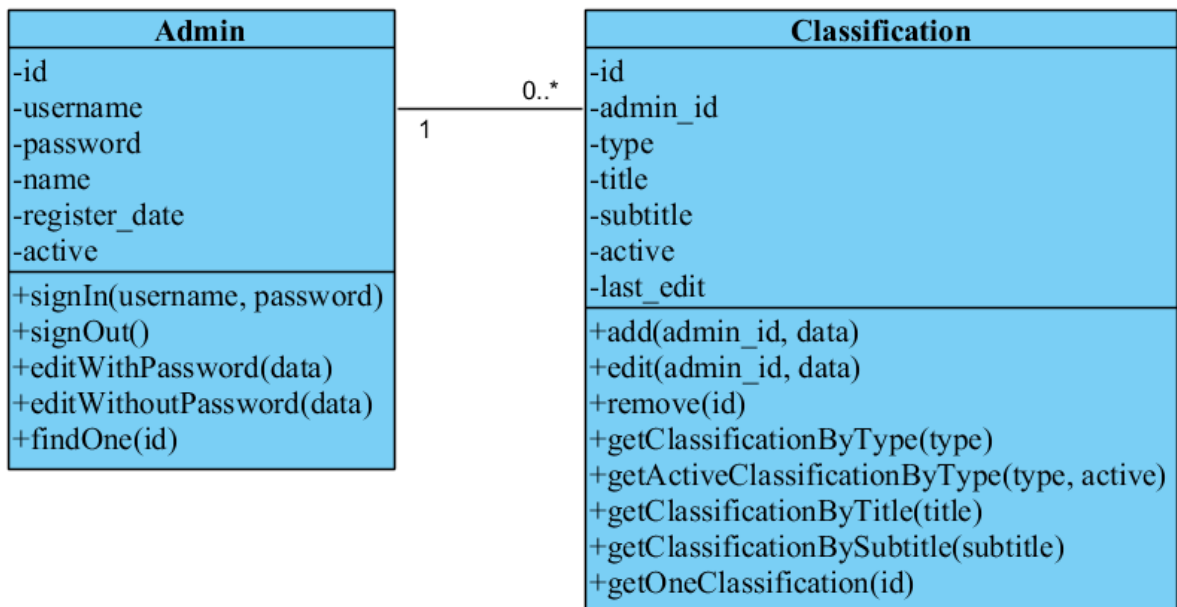


Figure 8. Class Diagram of The Management System.



Figure 9. Class Diagram of The Selling System.

7.4. Sequence Diagram

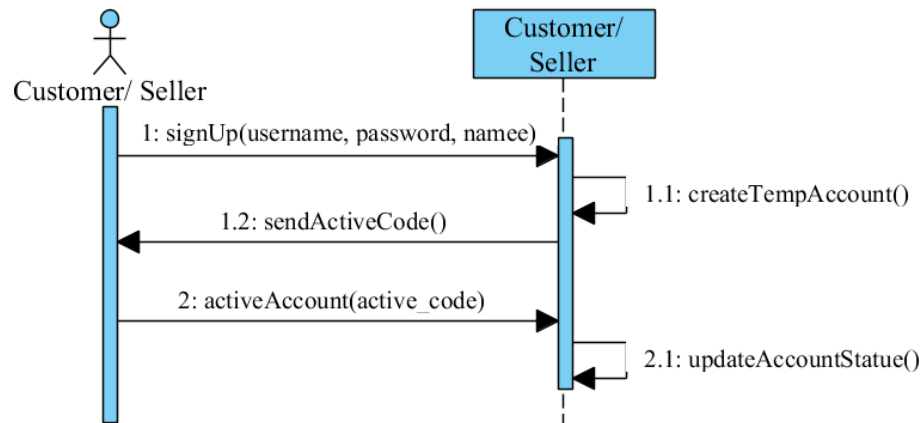


Figure 10. Sequence Diagram of Sign Up.

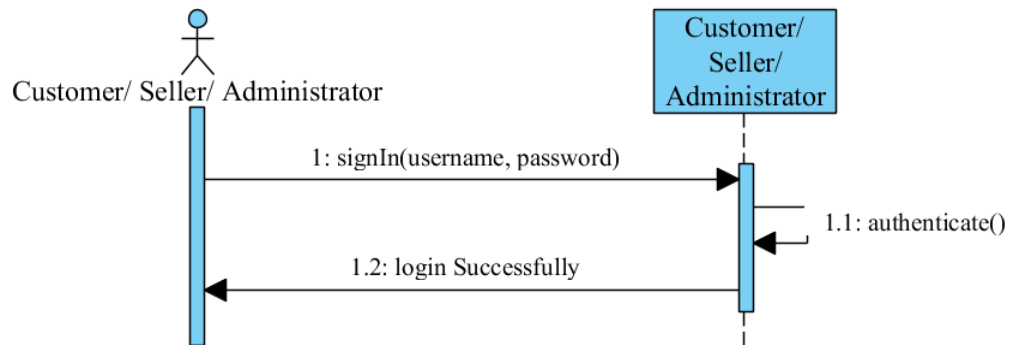


Figure 11. Sequence Diagram of Sign In.

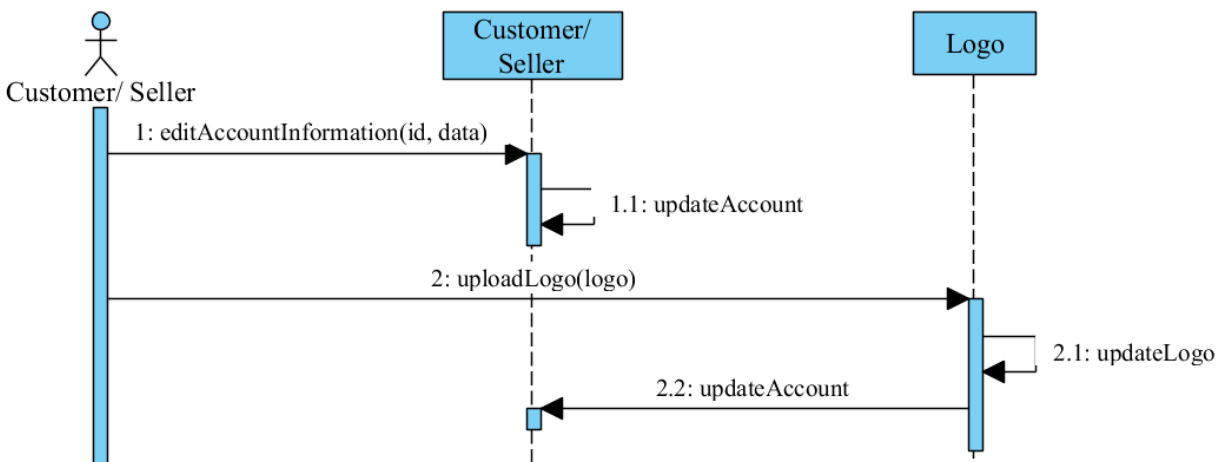


Figure 12. Sequence Diagram of Edit Account Information.

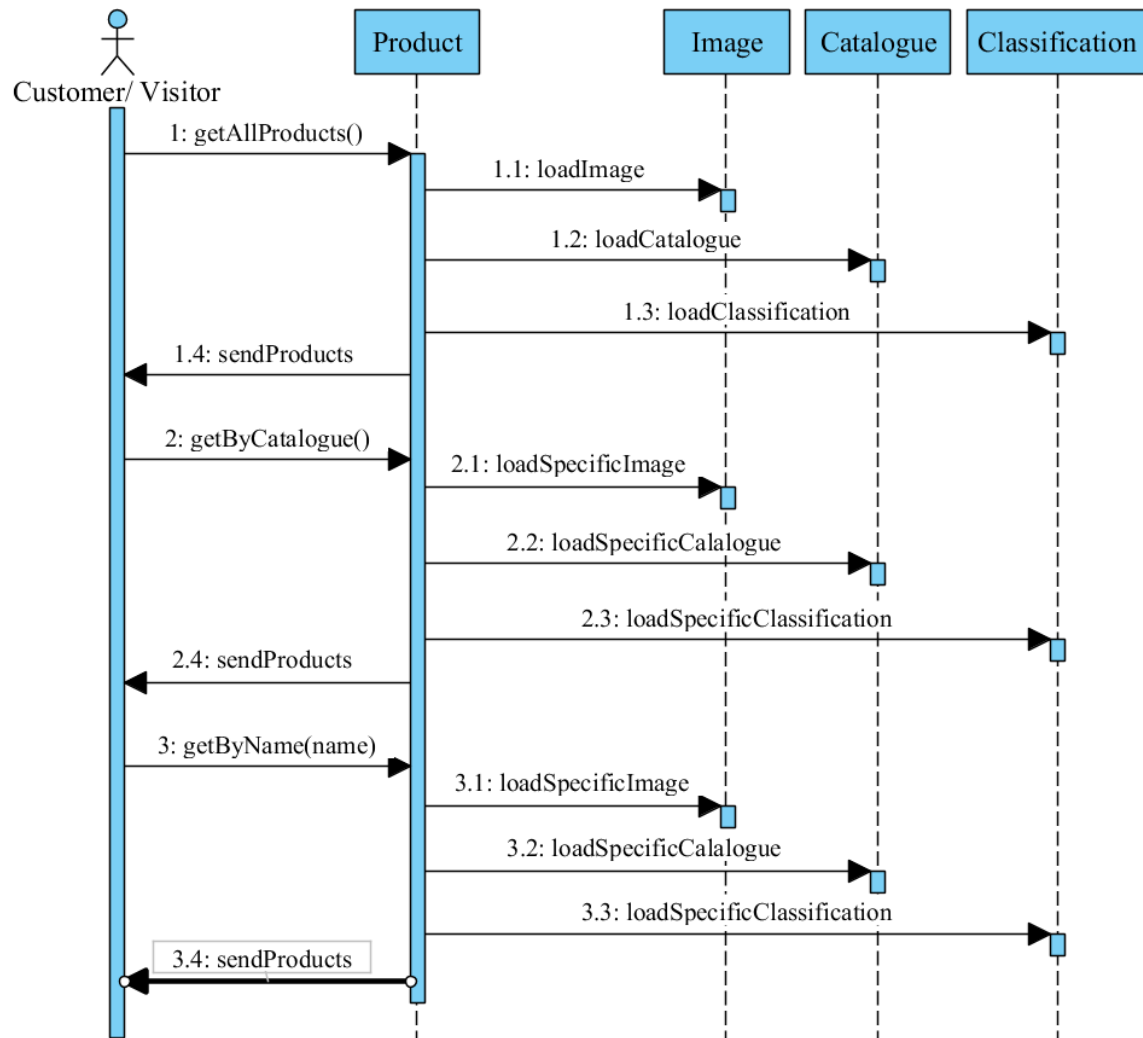


Figure 13. Sequence Diagram of Find Items from Home Page.

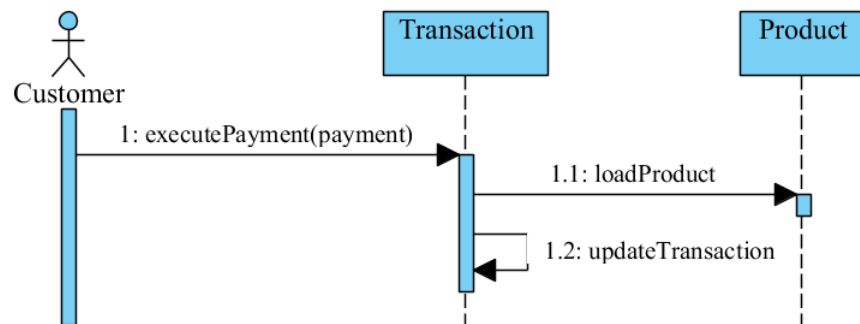


Figure 14. Sequence Diagram of Buy Products.

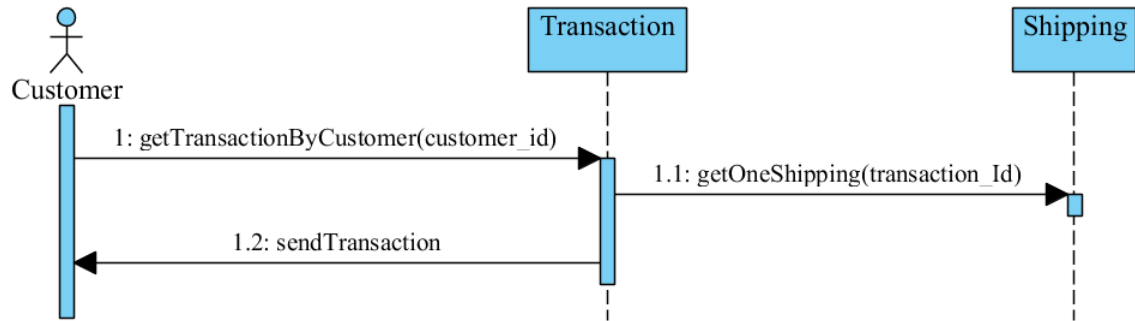


Figure 15. Sequence Diagram of Track Order.

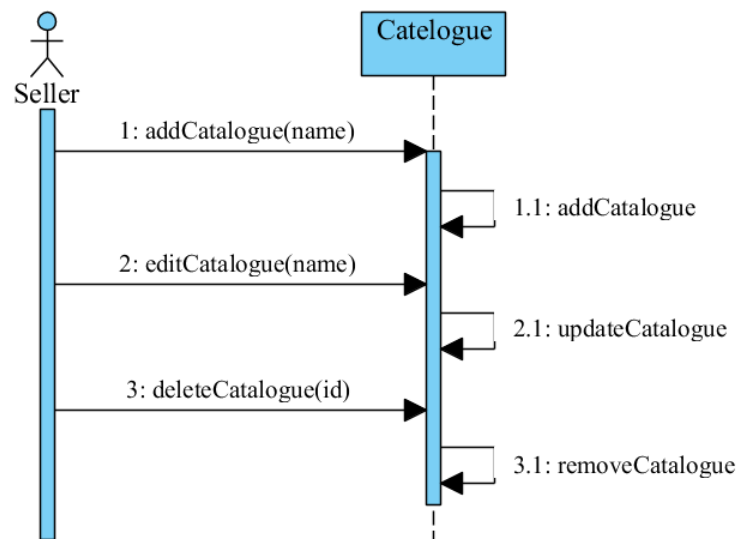


Figure 16. Sequence Diagram of Build Catalogues.

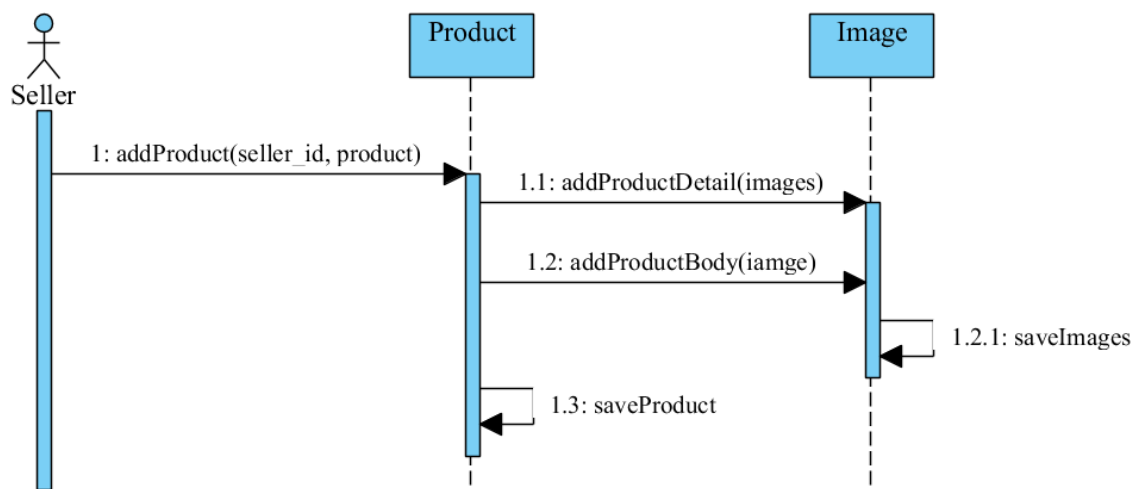


Figure 17. Sequence Diagram of Release Products.

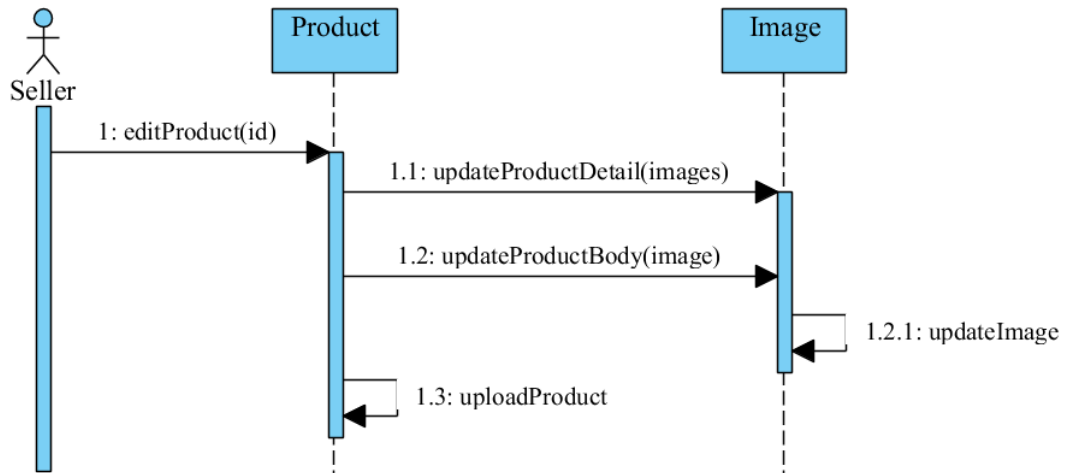


Figure 18. Sequence Diagram of Edit Products.

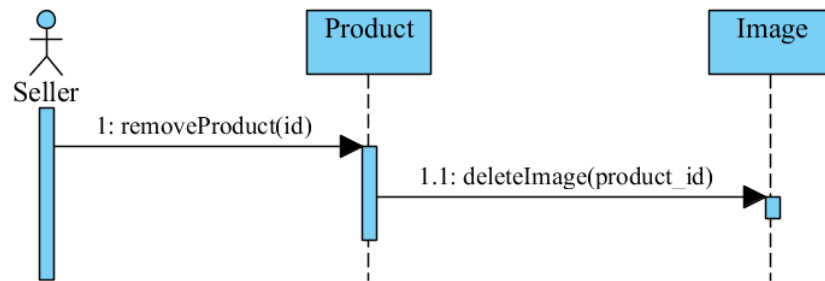


Figure 19. Sequence Diagram of Delete Products.

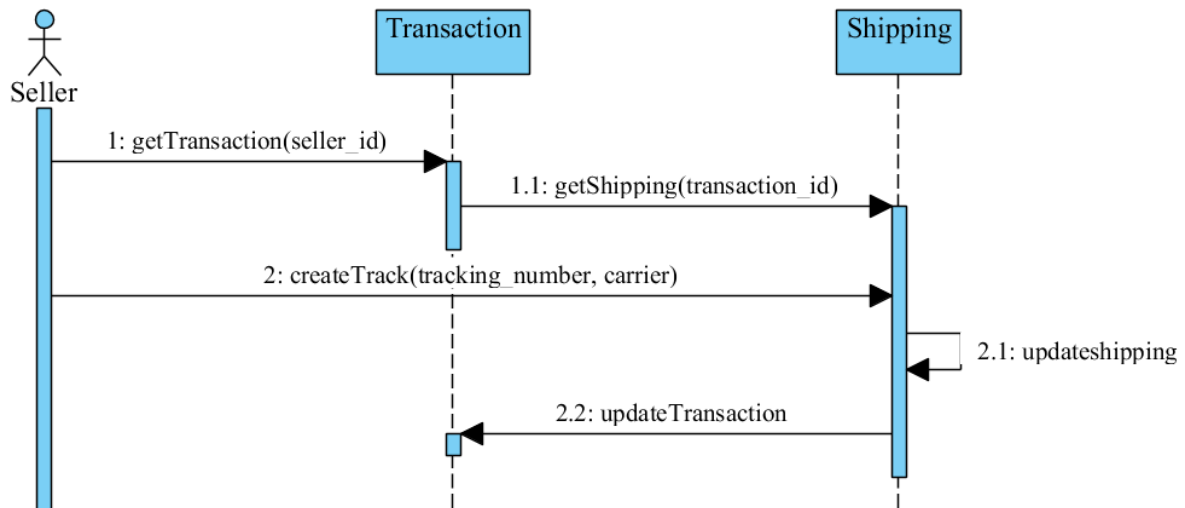


Figure 20. Sequence Diagram of Deliver Products.

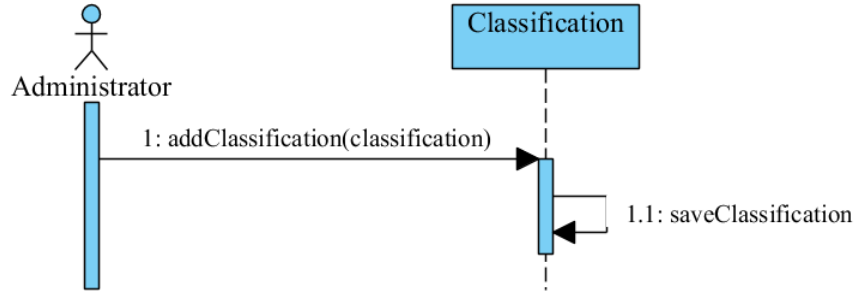


Figure 21. Sequence Diagram of Add Classifications.

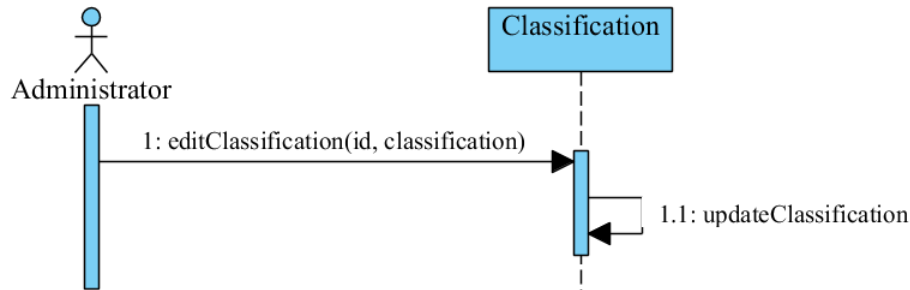


Figure 22. Sequence Diagram of Edit Classifications.

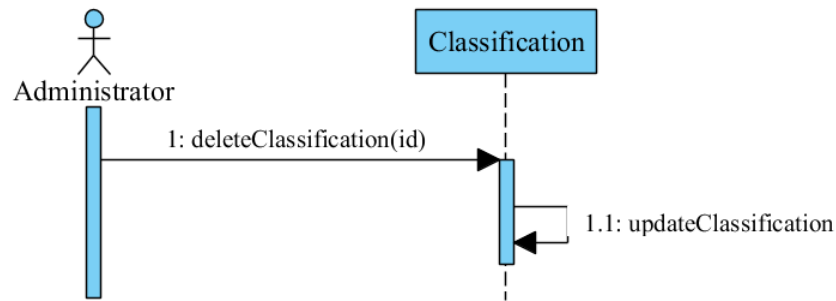


Figure 23. Sequence Diagram of Delete Classifications.

8. Development Plan

Application development is a highly complex mental work. There are six phases in the software development life cycle (SDLC), planning, analysis, design, implementation, testing, deployment and maintenance (Imam, 2019). Based on the SDLC, web application development process has eight phases and planning is an important part of the development (Rongala, 2015).

8.1. Overviews Plan

In this project, development cycle is about twelve weeks. Based on the pervious design report and development cycle, the project finished fifteen use cases (see in Section 7.2) using MEVN technology stack. The back-end development and the front-end development are synchronized.

The plans have been adjusted during the development process. Please refer to Section 8.2 for details.

8.2. Plan and Development Each Week

Code Book of Plan:

* - where the plan changes

/ - no work needs to be done

Green - finished

Yellow - in progress but not finished

Red – has not started yet

Week 1:

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 4: Edit Account Information
Week 3	Use case 8: Build Catalogues
Week 4	Use case 9: Release Products
Week 5	Use case 11: Delete Products
Week 6	Use case 10: Edit Products
Week 7	Use case 5: Find Items from Home Page
Week 8	Use case 6: Buy Products
Week 9	Use case 12: Deliver Products
Week 10	Use case 13: Add Classifications
Week 11	Use case 14: Edit Classifications
Week 12	Use case 15: Delete Classifications

Work Completion:

Plan (Week 1)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 2: Sign In (including: check email & password, create cookies)	Sign In (Customer, Seller, Administrator)	Unit Testing	Sign In Form	E2E Testing
Use case 1: Sign Up (including: email verify, password verify, name verify, send active mail, activate account)	Sign Up (Customer, Seller, Administrator)	Active Link (used to change account status) Unit Testing		Sign Up From E2E Testing
Use case 3: Sign Out (including: delete cookies)		Remove Cookies		Sign Out Dialog E2E Testing
Problems:				
1. How to use active link to update account status.				
2. Form information is submitted in cipher-text.				

Week 2: The computer was damaged, and the project plan was delayed. To ensure that the project can be completed on time, the workload increased in the sixth week.

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up* Use case 2: Sign In* Use case 3: Sign Out*
Week 3	Use case 4: Edit Account Information*
Week 4	Use case 8: Build Catalogues*
Week 5	Use case 9: Release Products*
Week 6	Use case 10: Edit Products* Use case 11: Delete Products*
Week 7	Use case 5: Find Items from Home Page*
Week 8	Use case 6: Buy Products*
Week 9	Use case 12: Deliver Products*
Week 10	Use case 13: Add Classifications*
Week 11	Use case 14: Edit Classifications*
Week 12	Use case 15: Delete Classifications*

Work Completion:

Plan (Week 2)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 2: Sign In (including: check email & password, create cookie)	Sign In (Customer, Seller, Administrator)	Unit Testing	Sign In Form	E2E Testing
Use case 1: Sign Up (including: email verify, password verify, name verify, send active mail, activate account)	Sign Up (Customer, Seller, Administrator)	Unit Testing	Sign Up Form	E2E Testing
Use case 3: Sign Out (including: delete cookie)	Sign Out (remove cookie)	Unit Testing	Sign Out Dialog	E2E Testing
Problems: No problem.				

Week 3: Some use cases have been re-planned. Use case 7 (Track Order) was added in Week 11 and Testing was arranged in Week 12.

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogues* Use case 9: Release Products* Use case 10: Edit Products* Use case 11: Delete Products*
Week 4	Use case 4: Edit Account Information*
Week 5	Use case 5: Find Items from Home Page
Week 6	Use case 6: Buy Products
Week 7	Use case 12: Deliver Products
Week 8	Use case 13: Add Classifications
Week 9	Use case 14: Edit Classifications
Week 10	Use case 15: Delete Classifications
Week 11	Use case 7: Track Order*
Week 12	Testing*

Work Completion:

Plan (Week 3)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 8: Build Catalogues (including create catalogue, edit catalogue and remove catalogue)	Build Catalogues (create catalogue, edit catalogue and remove catalogue)	Unit Testing		Catalogue Menu E2E Testing
Use case 9: Release Products (including submitting products' information and uploading products' images)	Release Products (submit products' information)	Upload Images to MongoDB Unit Testing		Products Table E2E Testing
Use case 10: Edit Products	Edit Products	Unit Testing		Edit Form E2E Testing
Use case 11: Delete Products	Delete Products	Unit Testing		Delete Dialog E2E Testing
Problems:				
1. How to add catalogues on seller home page dramatically.				
2. How to response products belonged to different catalogues in the same table.				

Week 4: Since the plan for the third week has not been fully completed, the remaining work was continued in this week.

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogues Use case 9: Release Products Use case 10: Edit Products Use case 11: Delete Products
Week 4	Continue the unfinished work in Week 3* Use case 4: Edit Account Information
Week 5	Use case 5: Find Items from Home Page
Week 6	Use case 6: Buy Products
Week 7	Use case 12: Deliver Products
Week 8	Use case 13: Add Classifications
Week 9	Use case 14: Edit Classifications
Week 10	Use case 15: Delete Classifications
Week 11	Use case 7: Track Order
Week 12	Testing

Work Completion:

Plan (Week 4)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 8: Build Catalogues (including create catalogue, edit catalogue and remove catalogue)		Unit Testing	Catalogue Menu	E2E Testing
Use case 9: Release Products (including submitting products' information and uploading products' images)	Upload Images to AWS S3	Unit Testing	Products Table	E2E Testing
Use case 10: Edit Products		Unit Testing	Edit Form	E2E Testing
Use case 11: Delete Products		Unit Testing	Delete Dialog	Delete Dialog, E2E Testing
Use case 4: Edit Account Information (Customer and seller)	Edit Seller account, Edit Customer Account	Unit Testing		Edit form, E2E Testing
Problems:				
1. How to upload images in AWS S3.				

Week 5: Due to too much work on the front end in the fourth week, the plan was not completed within the expected time. Therefore, this week continues the remaining work and reschedules the rest.

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogues Use case 9: Release Products Use case 10: Edit Products Use case 11: Delete Products
Week 4	Continue the unfinished work in Week 3 Use case 4: Edit Account Information
Week 5	Use case 4: Edit Account Information*
Week 6	Use case 5: Find Items from Home Page (Back-end) *
Week 7	Use case 5: Find Items from Home Page (Front-end) *
Week 8	Use case 13: Add Classifications Use case 14: Edit Classifications Use case 15: Delete Classifications
Week 9	Use case 6: Buy Products
Week 10	Use case 12: Deliver Products
Week 11	Use case 7: Track Order
Week 12	Testing

Work Completion:

Plan (Week 5)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 4: Edit Account Information (Customer and seller)	Edit Seller account Edit Customer Account Edit address information Upload User's Logo Delete User's Logo	Unit Testing	Show Account Edit Account Form Address Table Edit Address Form Upload Logo Display Logo	E2E Testing
Problems:				
<ol style="list-style-type: none"> How to get images from the AWS S3. Rendering issues in Vue.js. 				

Week 6: The plan in the sixth and seventh week was postponed, and the plan in the eighth week was advanced because Use case 5 involves classification.

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogues Use case 9: Release Products Use case 10: Edit Products Use case 11: Delete Products
Week 4	Continue the unfinished work in Week 3 Use case 4: Edit Account Information
Week 5	Use case 4: Edit Account Information
Week 6	Use case 13: Add Classifications* Use case 14: Edit Classifications* Use case 15: Delete Classifications*
Week 7	Use case 5: Find Items from Home Page (Back-end) *
Week 8	Use case 5: Find Items from Home Page (Front-end) *
Week 9	Use case 6: Buy Products
Week 10	Use case 12: Deliver Products
Week 11	Use case 7: Track Order
Week 12	Testing

Work Completion:

Plan (Week 6)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 13: Add Classifications	Add Classifications	Unit Testing	Classification Page Add Classification Form	E2E Testing
Use case 14: Edit Classifications	Edit Classifications	Unit Testing	Edit Classification Form	E2E Testing
Use case 15: Delete Classifications	Delete Classifications	Unit Testing	Delete Classification Dialog	E2E Testing
Problems: No problem.				

Week 7: To avoid overwork, only the back end of Use Case 5 was completed this week.

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogue Use case 9: Release Products Use case 10: Edit Products Use case 11: Delete Products
Week 4	Continue the unfinished work in Week 3 Use case 4: Edit Account Information
Week 5	Use case 4: Edit Account Information
Week 6	Use case 13: Add Classifications Use case 14: Edit Classifications Use case 15: Delete Classifications
Week 7	Use case 5: Find Item from Home Page (Back-end)
Week 8	Use case 5: Find Items from Home Page (Front-end)
Week 9	Use case 6: Buy Products
Week 10	Use case 12: Deliver Products
Week 11	Use case 7: Track Order
Week 12	Testing

Work Completion:

Plan (Week 7)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 5: Find Items from Home Page (Back-end)	Get all products Get products by classifications Get products by catalogues Get products by name	Unit Testing		
Problems: 1. How to connect multiple documents in MongoDB.				

Week 8: Only the front end of Use Case 5 was completed this week

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogue Use case 9: Release Products Use case 10: Edit Products Use case 11: Delete Products
Week 4	Continue the unfinished work in Week 3 Use case 4: Edit Account Information
Week 5	Use case 4: Edit Account Information
Week 6	Use case 13: Add Classifications Use case 14: Edit Classifications Use case 15: Delete Classifications
Week 7	Use case 5: Find Item from Home Page (Back-end)
Week 8	Use case 5: Find Items from Home Page (Front-end)
Week 9	Use case 6: Buy Products
Week 10	Use case 12: Deliver Products
Week 11	Use case 7: Track Order
Week 12	Testing

Work Completion:

Plan (Week 8)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 5: Find Items from Home Page (Front-end)			Product Cards Search Bar Classification Navigation Catalogue Navigation	E2E Testing
Problems: No problem.				

Week 9:

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogue Use case 9: Release Products Use case 10: Edit Products Use case 11: Delete Products
Week 4	Continue the unfinished work in Week 3 Use case 4: Edit Account Information
Week 5	Use case 4: Edit Account Information
Week 6	Use case 13: Add Classifications Use case 14: Edit Classifications Use case 15: Delete Classifications
Week 7	Use case 5: Find Item from Home Page (Back-end)
Week 8	Use case 5: Find Items from Home Page (Front-end)
Week 9	Use case 6: Buy Products
Week 10	Use case 12: Deliver Products
Week 11	Use case 7: Track Order
Week 12	Testing

Work Completion:

Plan (Week 9)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 6: Buy Products	Get a specific product Execute PayPal Payment	Unit Testing	Product Page PayPal Payment checkout	E2E Testing
Problems: 1. How to decompose PayPal payment processes at the Front and Back Ends				

Week 10:

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogue Use case 9: Release Products Use case 10: Edit Products Use case 11: Delete Products
Week 4	Continue the unfinished work in Week 3 Use case 4: Edit Account Information
Week 5	Use case 4: Edit Account Information
Week 6	Use case 13: Add Classifications Use case 14: Edit Classifications Use case 15: Delete Classifications
Week 7	Use case 5: Find Item from Home Page (Back-end)
Week 8	Use case 5: Find Items from Home Page (Front-end)
Week 9	Use case 6: Buy Products
Week 10	Use case 12: Deliver Products
Week 11	Use case 7: Track Order
Week 12	Testing

Work Completion:

Plan (Week 10)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 12: Deliver Products	Implement TrackingMore API Create Tracking	Unit Testing	Order List Shipping Page	E2E Testing
Problems: No problem.				

Week 11:

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogue Use case 9: Release Products Use case 10: Edit Products Use case 11: Delete Products
Week 4	Continue the unfinished work in Week 3 Use case 4: Edit Account Information
Week 5	Use case 4: Edit Account Information
Week 6	Use case 13: Add Classifications Use case 14: Edit Classifications Use case 15: Delete Classifications
Week 7	Use case 5: Find Item from Home Page (Back-end)
Week 8	Use case 5: Find Items from Home Page (Front-end)
Week 9	Use case 6: Buy Products
Week 10	Use case 12: Deliver Products
Week 11	Use case 7: Track Order
Week 12	Testing

Work Completion:

Plan (Week 11)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Use case 7: Track Order	Convert addresses into geographic coordinates	Unit Testing	Tracking Page Tracking Information Card Tracking Map	E2E Testing
Problems: <ol style="list-style-type: none">How to convert a fuzzy address to geographic coordinate.How to apply the Google Maps JavaScript API to Vue.js.How to automatically adjust the center and zoom size of a map based on multiple markers.				

Week 12: This week, only part of the back-end and front-end tests have been completed and a new plan has been added to implement responsive web pages.

Week 1	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 2	Use case 1: Sign Up Use case 2: Sign In Use case 3: Sign Out
Week 3	Use case 8: Build Catalogue Use case 9: Release Products Use case 10: Edit Products Use case 11: Delete Products
Week 4	Continue the unfinished work in Week 3 Use case 4: Edit Account Information
Week 5	Use case 4: Edit Account Information
Week 6	Use case 13: Add Classifications Use case 14: Edit Classifications Use case 15: Delete Classifications
Week 7	Use case 5: Find Item from Home Page (Back-end)
Week 8	Use case 5: Find Items from Home Page (Front-end)
Week 9	Use case 6: Buy Products
Week 10	Use case 12: Deliver Products
Week 11	Use case 7: Track Order
Week 12	Testing Responsive Web Pages*

Work Completion:

Plan (Week 11)	Back-end (Node.js)		Front-end (Vue.js)	
	Done	Undone	Done	Undone
Testing	Test Sign In (Customer) Test Get Classification by Type Test Classification by Title		Test Sign In (Customer, Seller and Administrator)	
Responsive Web Pages			Convert all pages into responsive	
Problems: No problem.				

9. Testing

The aim of the testing is to test performance, security, usability and functionality of the website. This test plan sets out the scope of the testing efforts and the test results.

9.1. Back-end

Test Cases:

Test Case ID	Test Procedure	Expected Result
TC_B01: Sign In (Customer)	<p>Before testing, prepare two testing data. One is an active account, another is an inactive account.</p> <p>Valid Login: post a valid username and password to /login/customer API.</p> <p>Invalid Login:</p> <ol style="list-style-type: none">1) Post a username that does not exist.2) Post a username that does not active.3) Post a wrong password with a valid username. <p>After testing, remove all testing data.</p>	<p>Valid Login: The response data should be an object with a success message and the customer information.</p> <p>Invalid Login:</p> <ol style="list-style-type: none">1) The response data should be an object with a message of user not exists. No customer information in the response data.2) The response data should be an object with a message of user inactive. No customer information in the response data.3) The response data should be an object with a message of username or password incorrect. No customer information in the response data.
TC_B02: Get Classification by Type	<p>Before testing, prepare two testing data with different type.</p> <p>Request an active classification by type.</p> <p>After testing, remove all testing data.</p>	<p>The response data should be an object with the testing data.</p>
TC_B03: Get Classification by Title	<p>Before testing, prepare two testing data with different title.</p> <p>Request an active classification by title.</p> <p>After testing, remove all testing data.</p>	<p>The response data should be an object with the testing data.</p>

Test Result:

```
Customer Test
Customer insert success.
  POST /login/customer
    ✓ should return a message and a valid customer
    ✓ should return a Not Exist and null data
    ✓ should return an inactive message and null data
    ✓ should return an match failed message and null data
Customers delete success.

4 passing (1s)
```

Figure 24. The Result of TC_B01.

```
Classification Test
Classifications insert success.
  GET /admin/classification/type_active/:type
    ✓ should return a category classification
  GET /admin/classification/title_active/:title
    ✓ should return a category classification
Classifications delete success.

2 passing (1s)
```

Figure 25. The Result of TC_B02 and TC_B03.

9.2. Front-end

Test Cases:

Test Case ID	Test Procedure	Expected Result
TC_F01: Sign In (Customer, Seller and Administrator)	<p>Before testing, open the Home Page.</p> <p>Error Message in Form:</p> <ol style="list-style-type: none">1) Check the required of username.2) Check the format of username.3) Check the required of password.4) Check the length of password. <p>Invalid Sign In:</p> <ol style="list-style-type: none">1) Login with a wrong username.2) Login with a wrong password. <p>Valid Sign In:</p> <ol style="list-style-type: none">1) Login with a customer account.2) Login with a seller account.3) Login with an administrator account.	<p>Error Message in Form:</p> <ol style="list-style-type: none">1) The username field should show an error message with e-mail is required.2) The username field should show an error message with must be valid e-mail.3) The password field should show an error message with password is required.4) The password field should show an error message with password must be at most 30 characters long. <p>Invalid Sign In:</p> <ol style="list-style-type: none">1) The fail message should be show in the Sign In Form with username or password incorrect.2) The fail message should be show in the Sign In Form with username or password incorrect. <p>Valid Sign In:</p> <ol style="list-style-type: none">1) The website should redirect to the Home Page.2) The website should have a URL of Seller Home in the toolbar.3) The website should have a URL of Administrator Home in the toolbar.

Test Result:

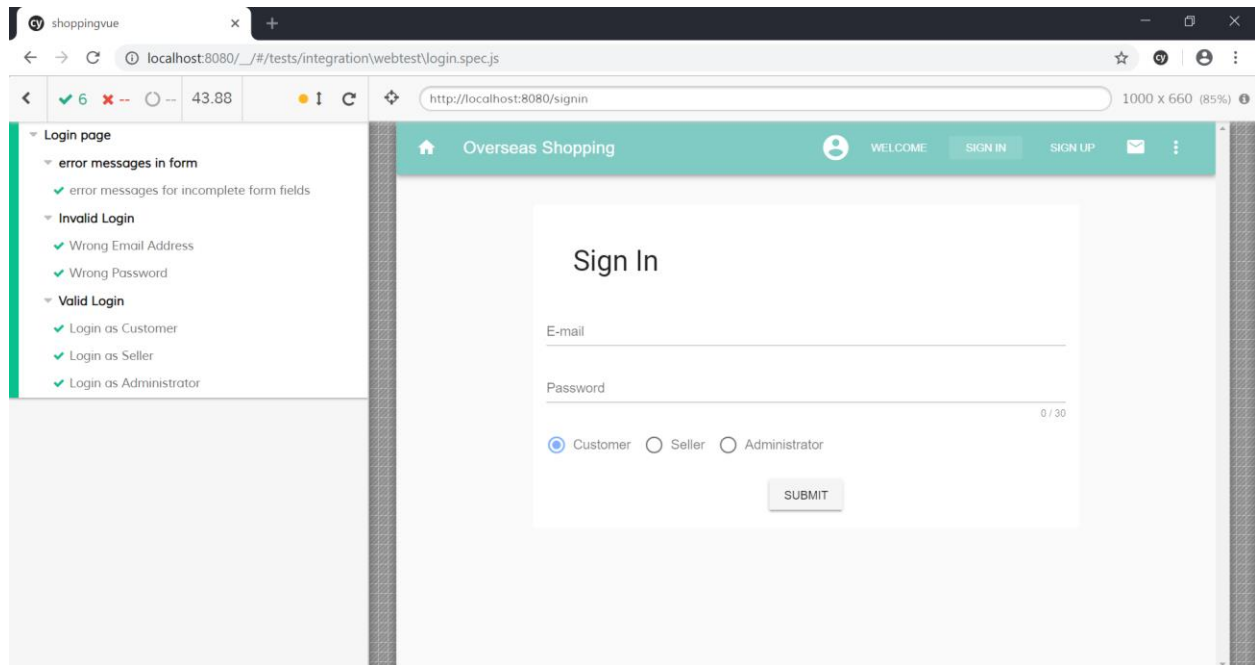


Figure 26. The Result of TC_F01.

10. Problems and Solutions

10.1. Validate Username and Active Account

Problem:

How to validate the username inputted by user is a valid username, rather than a malicious user attempting to occupy multiple usernames.

Solution:

The system creates an active code and saves in database after a user signup, and then the system automatically sends the active code by email to user's email box. Users click the active link to active their account. Once the active link is clicked, the system checks the code to ensure the code is created by the system before and change the active statue of account.

Tools & Reference: Nodemailer (Reinman, 2019).

10.2. Manage Catalogues and Products Dramatically

Problem:

How to manage catalogues and show specific products by different catalogues since seller can create many catalogues.

Solution:

The system uses a menu to manage catalogue and a data table to maintain products. Product' data only shows after sellers choose a catalogue.

10.3. Store Images in AWS S3

Problem:

How to efficiently manage a large amount of image data of online overseas shopping websites.

Solution:

The system uses AWS S3 to store images. At the same time, the paths and keys of the images are stored in the database so that the system can get the images on S3.

Tools & Reference: AWS S3 and its REST API documentation (Docs.aws.amazon.com, 2019).

10.4. Connect Multiple Documents in Non-Relational Database

Problem:

How to combine multiple involved data of products in the website.

Solution:

The system uses populate function provided by Mongoose to connect multiple documents by key before retrieving products.

Tools & Reference: Populate function of Mongoose (Mongoosejs.com, 2019).

10.5. Vue Component Rendering Problem

Problem:

Vue component finishes rendering before getting data. It causes no data to be showed in the website.

Solution:

The system gets data in Created function and uses a variate to control the visibility of components.

Tools & Reference: Vue.js and its guide book (Vuejs.org, 2019).

10.6. PayPal Payment Flow

Problem:

How to decompose PayPal payment processes into the front end and the back end.

Solution:

The system uses two buttons to control the PayPal payment flow. One is to create payment list, another is to send the request with the payment list to the back end to execute the payment.

Tool & Reference: PayPal and its developer documentation (Developer.paypal.com, 2019).

10.7. Tracking Information Showing in Maps

Problem:

How to show fuzzy addresses of the logistics information in Google Maps and automatically adjust the center and zoom size of the maps by multiple markers.

Solution:

The system reformats addresses and then uses Geocoding service in Google Maps JavaScript API to convert addresses to geographic coordinates. Since Map JavaScript API does not support Vue.js, the system uses the algorithm written by the project developer to implement adjustment of the center and zoom size.

Tool & Reference: Google Maps and its Maps JavaScript API documentation (Google Developers, 2019).

10.8. Cross Side Problem

Problem:

The front end cannot connect to the back end because their cross domains are different.

Solution:

The system uses Cross-Origin Resource Sharing (CORS) to allow the front end to access the back-end resources.

Tool & Reference: CORS (MDN Web Docs, 2019)

11. Suggested Additional Features

1. Customers are notified of product updates in their shopping carts or wish list by email.
2. Email calls customers back if they haven't logged in for a long time.
3. System intelligently recommends products to customers when they browse websites.
4. Products are shown by 3-D models generated from 2-D images.

12. Summary

An overseas shopping website should have many functions and I designed twenty-seven user cases in the design report. In the 12-week development cycle, I completed 15 use cases, two back end test cases and one front end test case. The project adds a track order use case and uses responsive web design.

In the project, I mastered the MVEN technology stack and learned to apply many cloud services and third-party APIs, such as AWS, PayPal, Google Maps and so on. I have a deep understanding of responsive website design and implemented it in my project through the Vuetify framework.

Besides, I also learned to make full use of Internet resources and related services and API development documentation to assist project development. In the project, I not only learned a lot of programming techniques, but also learned to plan, test and solve problems. I believe that all these knowledges will have a positive impact on my future development.

13. Appendices

Website link: <https://shoppingwebsite.firebaseio.com>

(test account: c@c.com 567; s@s.com 567; admin@gmail.com 567)

Video link: <https://youtu.be/omQgqlvWaJk>

GitHub link: <https://github.com/SMARTBIGBOSS/shoppingweb> (Back-end)

<https://github.com/SMARTBIGBOSS/shoppingvue> (Front-end)

Reference

Baike.baidu.com. (2019). amazon s3_百度百科. [online] Available at:

<https://baike.baidu.com/item/amazon%20s3/10809744> [Accessed 22 Apr. 2019].

Baike.baidu.com. (2019). Heroku_百度百科. [online] Available at: <https://baike.baidu.com/item/Heroku> [Accessed 22 Apr. 2019].

Devcenter.heroku.com. (2019). GitHub Integration (Heroku GitHub Deploys) | Heroku Dev Center. [online] Available at: <https://devcenter.heroku.com/articles/github-integration> [Accessed 22 Apr. 2019].

Developer.paypal.com. (2019). PayPal Developer Documentation - PayPal Developer. [online] Available at: <https://developer.paypal.com/> [Accessed 25 Apr. 2019].

Docs.aws.amazon.com. (2019). Amazon S3 REST API Introduction - Amazon Simple Storage Service. [online] Available at: <https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html> [Accessed 25 Apr. 2019].

En.wikipedia.org. (2019). Firebase. [online] Available at: <https://en.wikipedia.org/wiki/Firebase> [Accessed 22 Apr. 2019].

En.wikipedia.org. (2019). Google Maps. [online] Available at: https://en.wikipedia.org/wiki/Google_Maps [Accessed 22 Apr. 2019].

Firebase. (2019). Firebase Hosting | Firebase. [online] Available at: https://firebase.google.com/docs/hosting/?gclid=CjwKCAjwzPXIBRAjEiwAj_XTESVo7ji8j9ILk13xJ-qCkoXSrjt-jGPBoKS2-rbxsrbuFOgS7Ut-FxoCQmAQAvD_BwE [Accessed 22 Apr. 2019].

Google Developers. (2019). Overview | Maps JavaScript API | Google Developers. [online] Available at: <https://developers.google.com/maps/documentation/javascript/tutorial> [Accessed 22 Apr. 2019].

Imam, A. (2019). Software Development Life Cycle: The phases of SDLC - TestLodge Blog. [online] TestLodge Blog. Available at: <https://blog.testlodge.com/software-development-life-cycle/> [Accessed 24 Apr. 2019].

Ke, V. (2018). 了解看板(Kanban)方法的核心方法，活化你的管理思維. [online] 進度條 Progress Bar 程式線上教學. Available at: <https://progressbar.tw/posts/66> [Accessed 30 Dec. 2018].

MDN Web Docs. (2019). Cross-Origin Resource Sharing (CORS). [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS> [Accessed 25 Apr. 2019].

Mongoosejs.com. (2019). Mongoose v5.5.3: Query Population. [online] Available at: <https://mongoosejs.com/docs/populate.html> [Accessed 25 Apr. 2019].

Reinman, A. (2019). Nodemailer:: Nodemailer. [online] Nodemailer.com. Available at: <https://nodemailer.com/about/> [Accessed 25 Apr. 2019].

Rongala, A. (2015). 8 Essential Phases in Web Application Development Process - Invensis Technologies. [online] Invensis Technologies. Available at: <https://www.invensis.net/blog/it/8-essential-phases-in-web-application-development-process/> [Accessed 24 Apr. 2019].

Segmentfault.com. (2018). vue 简单入门（一）vue 是什么，为什么我们要学 vue？ - vue 简单入门 - SegmentFault 思否. [online] Available at: <https://segmentfault.com/a/1190000015560315> [Accessed 29 Dec. 2018].

Vuejs.org. (2019). The Vue Instance — Vue.js. [online] Available at: <https://vuejs.org/v2/guide/instance.html> [Accessed 25 Apr. 2019].

Wiki.mbalib.com. (2019). [online] Available at: <https://wiki.mbalib.com/wiki/Paypal> [Accessed 22 Apr. 2019].

简书. (2018). Node.js 是什么？Node.js 与 JavaScript 的区别是什么？. [online] Available at: <https://www.jianshu.com/p/21d3130ca059> [Accessed 29 Dec. 2018].

周, 思. (2018). [online] Zhihu.com. Available at: <https://www.zhihu.com/question/20059632> [Accessed 29 Dec. 2018].