

**UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HA NOI**  
**ICT DEPARTMENT**



**GROUP PROJECT REPORT**

***Online store management system***

**SUPERVISOR:** *Dr. Nghiêm Thị Phương*

**GROUP MEMBERS:**

- \_ Đặng Hoàng Phúc BA9-050*
- \_ Đoàn Đình Nam BA9-045*
- \_ Nguyễn Thị Nhàn BA9-049*

Hanoi, March 2022

# Table of contents

A/ INTRODUCTION .....	5
I/ Context and Motivation .....	5
II/ Objectives .....	5
III/ Project timeline .....	5
IV/ Thesis structure .....	6
B/ METHODOLOGY .....	8
I/ Functional requirements.....	8
II/ Diagrams.....	9
1. Use-case diagram .....	9
2. Sequence diagrams.....	11
a) Landing page .....	11
a1) Login.....	11
a2) Register .....	11
a3) View Product .....	12
a4) View a product.....	13
a5) Add product to wish list.....	13
a6) Add product to cart .....	14
a7) Payment .....	14
a8) Edit account .....	15
a9) Delete Account .....	16
a10) Logout.....	17
b) Admin Page .....	17
b1) Login.....	17
b2) View dashboard .....	17
b3) Manage product .....	18
b3.1) Insert Product .....	18
b3.2) View and Update Product.....	18
b4) Manage Bundle.....	19
b4.1) Insert Bundle .....	19

b4.2) View and Update Bundle .....	20
b5) Manage Category .....	21
b5.1) Insert Sub Category .....	21
b5.2) View and Update Sub Category .....	22
b5.3) Insert Category .....	23
b5.4) View and Update Category .....	24
b6) Assign Product to Bundle Relation .....	24
b6.1) Insert Relation .....	24
b6.2) View and Update Relation .....	25
b7) Manage Manufacturer .....	26
b7.1) Insert Manufacturer .....	26
b7.2) View and Update Manufacturer .....	26
b8) Update contact us .....	27
b9) Update About Us .....	28
b10) Manage Coupon.....	28
b10.1) Insert Coupon .....	28
b10.2) View and Update Coupon .....	29
b11) Update Term .....	29
b12) View Customer .....	30
b13) View Order .....	30
b14) View Payment .....	31
b15) Manage User.....	31
b15.1) Insert User .....	31
b15.2) View User .....	32
b15.3) Edit Profile .....	32
b16) Logout.....	33
3. Class diagrams .....	34
a) Landing page .....	34
a1) View Product .....	34
a2) View a product.....	34

a3) Add product to wish list.....	35
a4) Add product to cart .....	36
a5) Payment .....	37
b) Admin Page .....	37
b1) Manage Product and Bundle .....	37
b2) Manage Category.....	38
b3) Assign Product to Bundle Relation .....	39
b4) Manage Manufacturer .....	39
b5) Manage Coupon.....	40
b6) Manage User.....	41
4. Database diagram .....	42
a) Roles of tables .....	43
b) Relationships .....	44
III/ System architecture .....	45
C/ TOOLS & TECHNICAL CHOICES .....	47
I/ PHP .....	47
II/ MySQL .....	47
III/ API .....	47
IV/ HTML .....	47
V/ CSS.....	48
VI/ JS.....	48
VII/ AJAX .....	48
D/ RESULT AND DISCUSSION.....	49
I/ Results.....	49
II/ Discussion.....	56
E/ CONCLUSION & FUTURE WORK.....	56
I/ Conclusion .....	56
II/ Future work .....	56

# A/ INTRODUCTION

## I/ Context and Motivation

Business stores have played an important role in our lives for thousands of years. For many people, going to the stores to shop and manually select the items they like and need for themselves as well as for their family and friends has become an integral part of their lives.

However, in today's world, everything has changed. The trend of online shopping is exploding because of its convenience. Now, customers don't need to go to the store to buy products in person. They simply stay at home and visit the websites of different stores to buy things.

With the changing world today, many stores have decided to have their own website to do business online and manage their business. Therefore, it is necessary to have a system of websites.

The main goal of this project is to build a system for a store to digitize its business. The system consists of two main web pages: the landing page and the admin page.

The store showcases its products and publishes its contact information and other information on the landing page. Customers can visit the landing page to view, search, select and buy the shop's products.

The admin page is where the store manages information about products, categories, orders, and users. They can also view cash flow statistics and products through this page.

## II/ Objectives

In this project, we aim to research and develop an online store software system. Our system goal is to create an environment where managers can easily manage store products and orders, customers can select and order the products they want. effectively. This system can be useful for small and medium business stores. There are two main goals:

- Develop auxiliary tools to manage products, orders, user accounts...
- Develop an interface that provides a smooth and enjoyable experience for store and shopper interaction.

## III/ Project timeline

## GANTT CHART

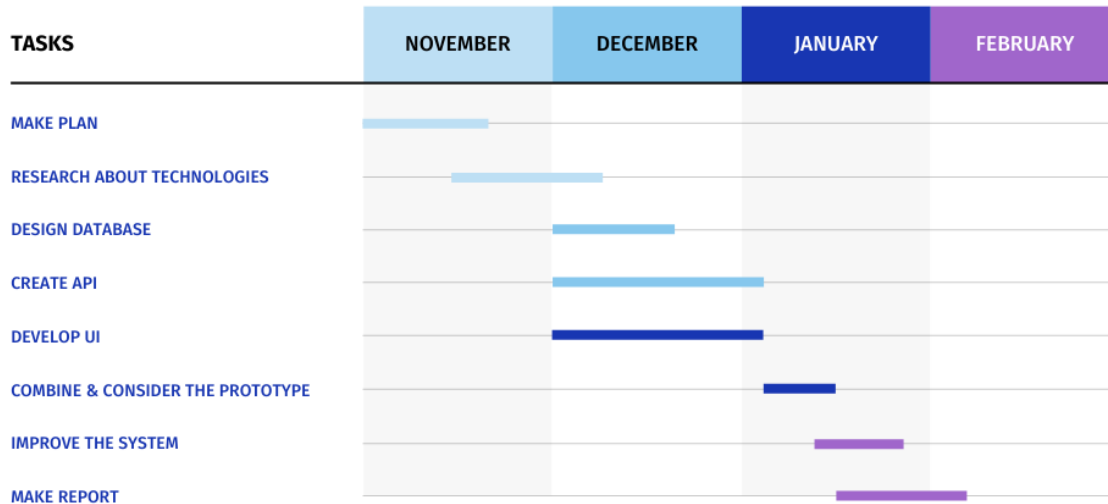


Figure 1. GANTT chart

This Team project started in November. From November to early December, we plan to design the tools used to create the system, and research those technologies at the same time. Once we had enough knowledge of the system, we started to design the database. In the month and a half from December, we set the following goals to accomplish:

- **The first** is to implement the database, create an API for user requests, and improve the login part.
- **The second** is to design UI/UX and improve it in the meantime.

In mid-January, we combined backend and frontend together. Then we review the program and improve as much as we can.

From mid-January, we started writing reports for the project.

## IV/ Thesis structure

The rest of the thesis will be structured as follows:

- Section 2: Methodology

- Section 3: Tools & Technical choices
- Section 4: Implementation
- Section 5: Result & Discussion
- Section 6: Conclusion & Future work
- Section 7: Appendix

## B/ METHODOLOGY

### I/ Functional requirements

There are 2 main types of actors: users (including customers), and the administrator (the website manager).

- Manage Profile
- View product details and bundle details.
- Add products to cart and manage orders.
- Add products to the wish list and manage the wish list.
- Apply coupon codes.
- Select payment method.
- Products and bundles management
- Product – bundles relations
- Categories and manufacturers management
- Update contact, about us details
- Coupon management
- Update term details
- List Customers
- Manage Customer's Order
- Payment Management
- User Management System
- View Total Earnings



## II/ Diagrams

To draw use-case diagrams and sequence diagrams, we use the Visual Paradigm for UML v.10 tool.

To draw class diagrams, we use the “Draw io” tool. We use it because it’s easy to use, it has been introduced and used in some courses before.

To draw database diagrams, we use the “dbdiagram.io” tool.

### 1. Use-case diagram

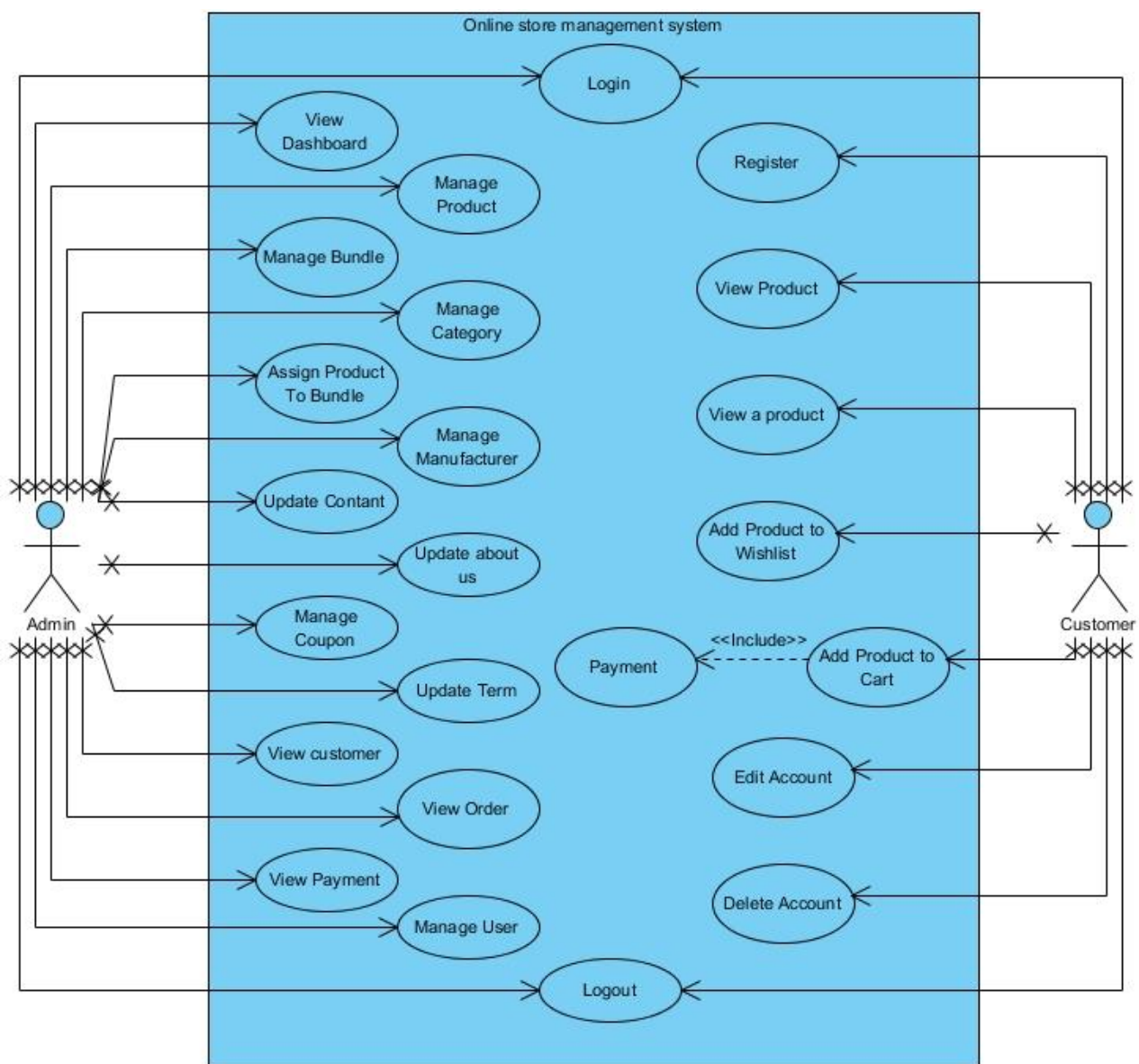


Figure 2. Use-case diagram

This is the initial use case diagram that we designed for the system. We tried our best to develop our system based on this design and below are the functions that we succeeded in implementing.

## 2. Sequence diagrams

### a) Landing page

#### a1) Login

This use case demonstrates the access of an account to the system. The account belongs to a customer. After filling in all information in the login form and pressing the “Login” button, the system validates the email and password entered from the account list. The customer will be logged in if the information is valid, else, the system will display an error message and request the customer to fill in the form again.

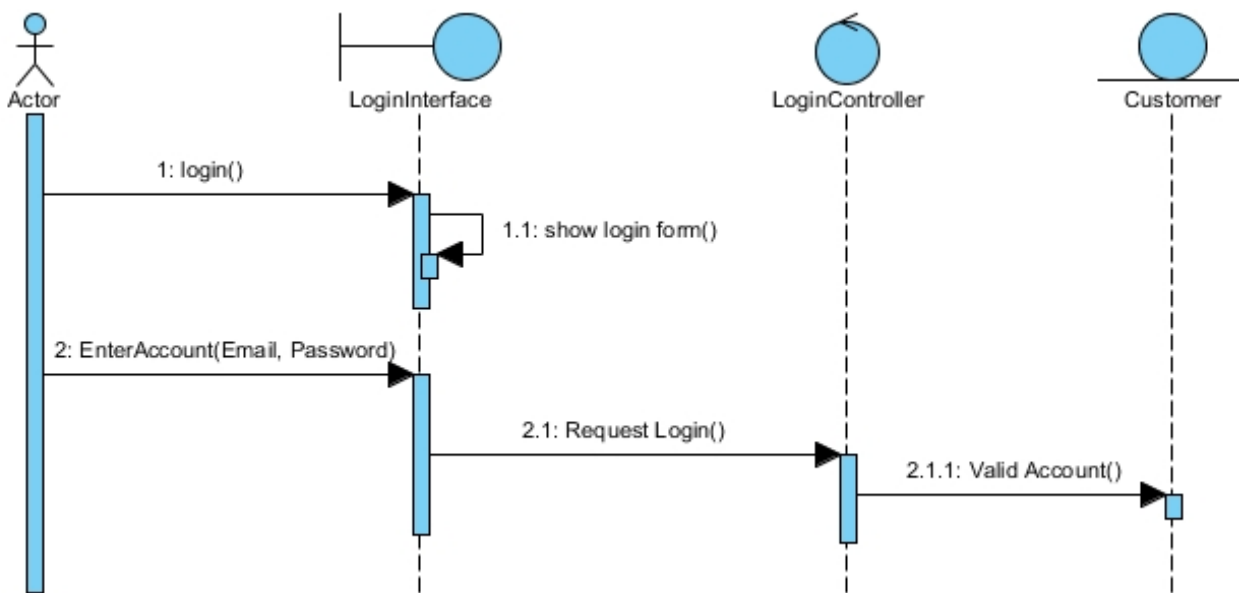


Figure 3. Login sequence diagram (Landing page)

#### a2) Register

This use case demonstrates the access of an account to the system. The account belongs to a customer. After filling in all information in the register form and pressing the “Register” button. The customer's account will be created if the information is valid, else, the system will display an error message and request the customer to fill in the form again.

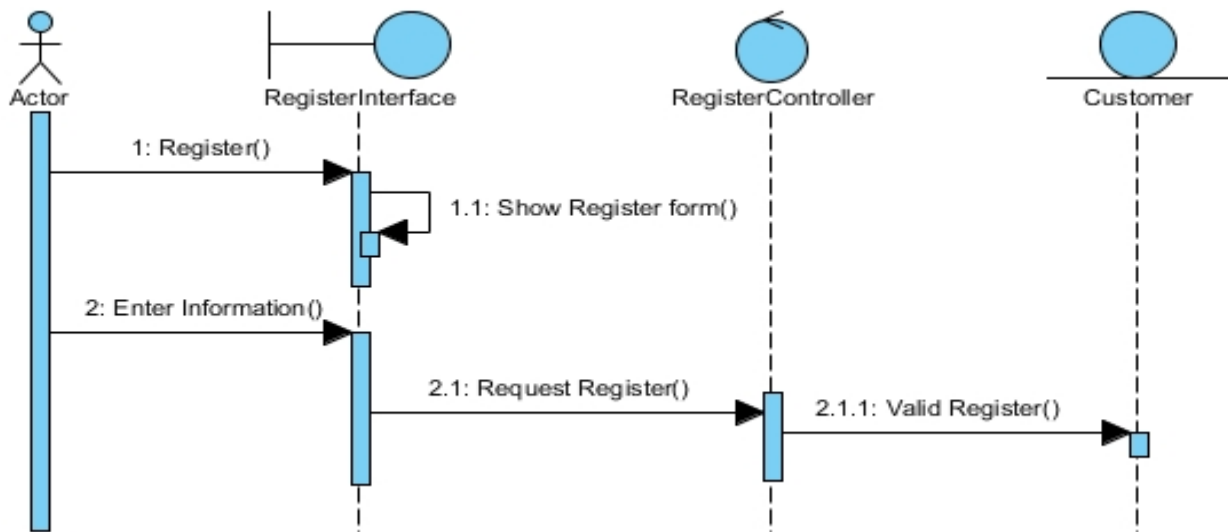


Figure 4. Register sequence diagram (Landing page)

### a3) View Product

This use case allows customers to view all the products of the online store. First, the actor needs to visit the Landing page. Therefore, the actor can click on the button ‘Shop’ in the Header to go to the Products page. The system then retrieves the data from the database and displays the products in a list of boxes. Each box contains the overall information about the product.

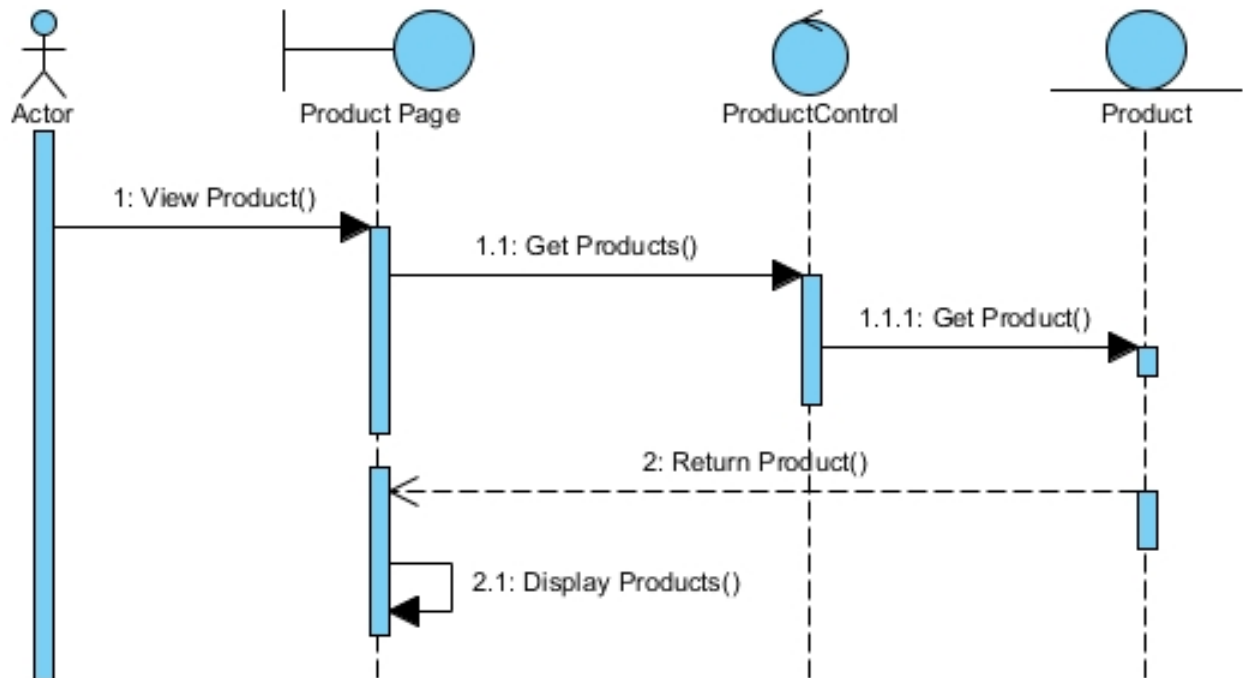


Figure 5. View Product sequence diagram (Landing page)

#### a4) View a product

From the shop page, the customer can click on the button ‘View details’ or click on the image of the product to go to that specific product page. The system then retrieves the data of the chosen product and the customer can now view the product’s information.

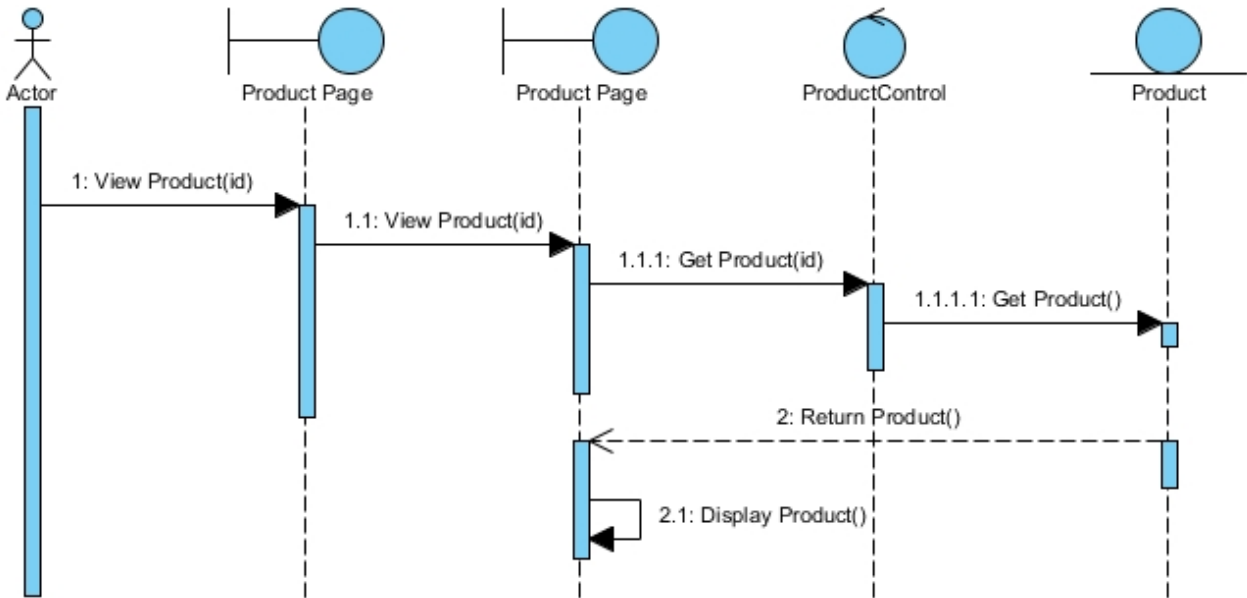


Figure 6. View a Product sequence diagram (Landing page)

#### a5) Add product to wish list

This use case allows customers to add products to their wish list. From the Product View Page of a specific product, if the customer wants to add that product to the wishlist, he/she can click on the “Add to Wishlist” button. The system then updates the wishlist of that customer.

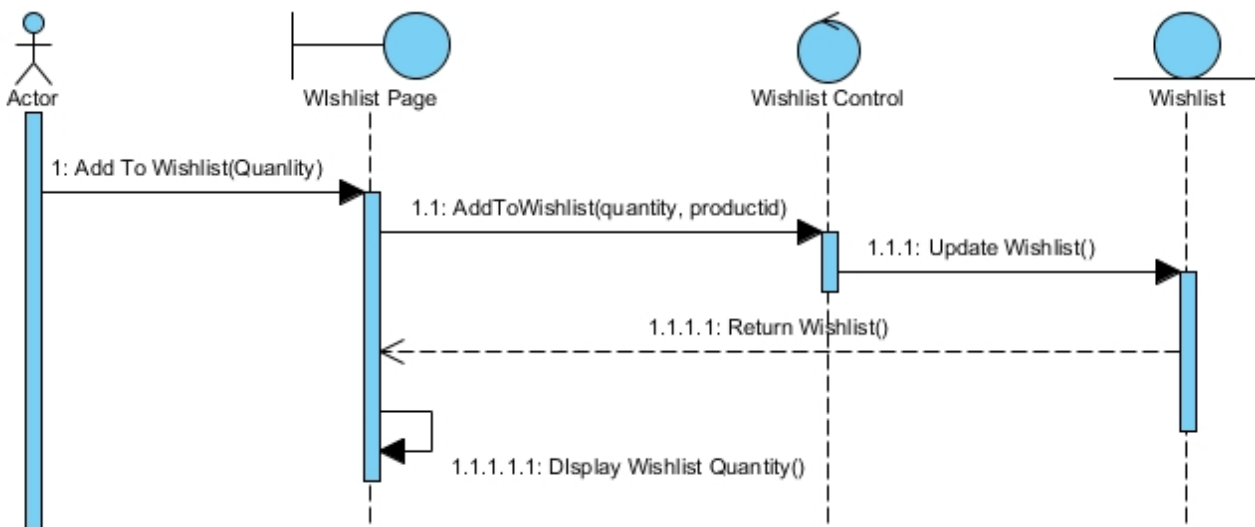


Figure 7. Add product to wish list sequence diagram (Landing page)

#### a6) Add product to cart

This use case allows customers to add products to their cart. From the Product View Page of a specific product, if the customer wants to add that product to the cart, he/she can modify the quantity, size button and then click on 'Add to Cart' button. The system then updates the cart of that customer. The total quantity of products in the cart is also updated at the Header of the page.

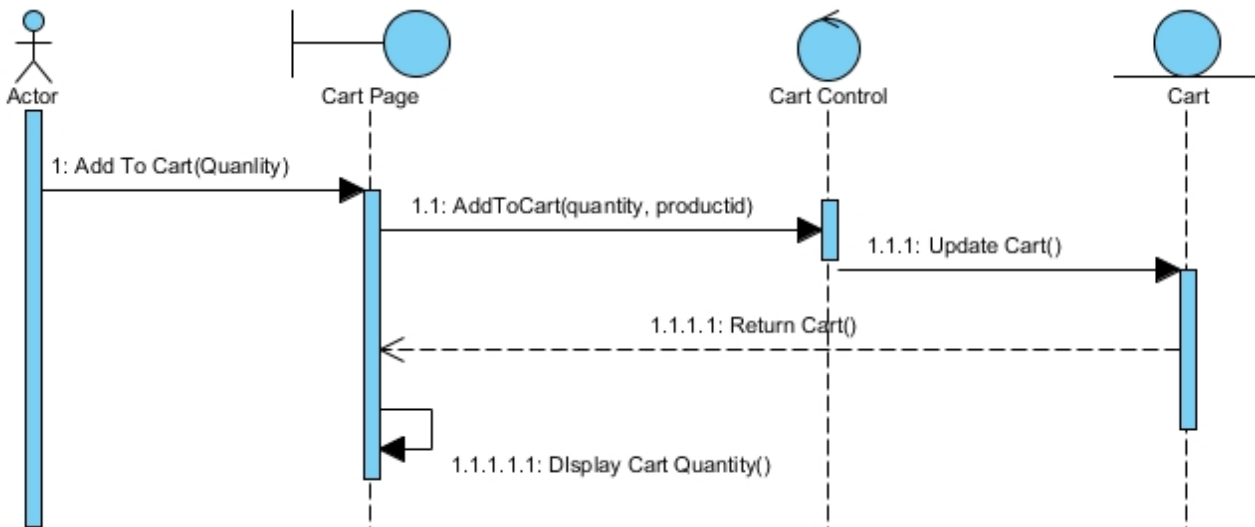


Figure 8. Add product to cart sequence diagram (Landing page)

#### a7) Payment

This use case allows the customer to confirm the order of the products in the shopping cart and choose a payment method. If the customer wants to buy the products that he/she added to the cart, he/she needs to visit the Cart page. There will be a list of products that are in the cart, the total amount of money. After checking all the information, the Customer presses the "Proceed to Payment" button and chooses the "pay offline" or "buy now" payment type to place an order. The order then is saved to the database, and the cart will be updated to be empty.

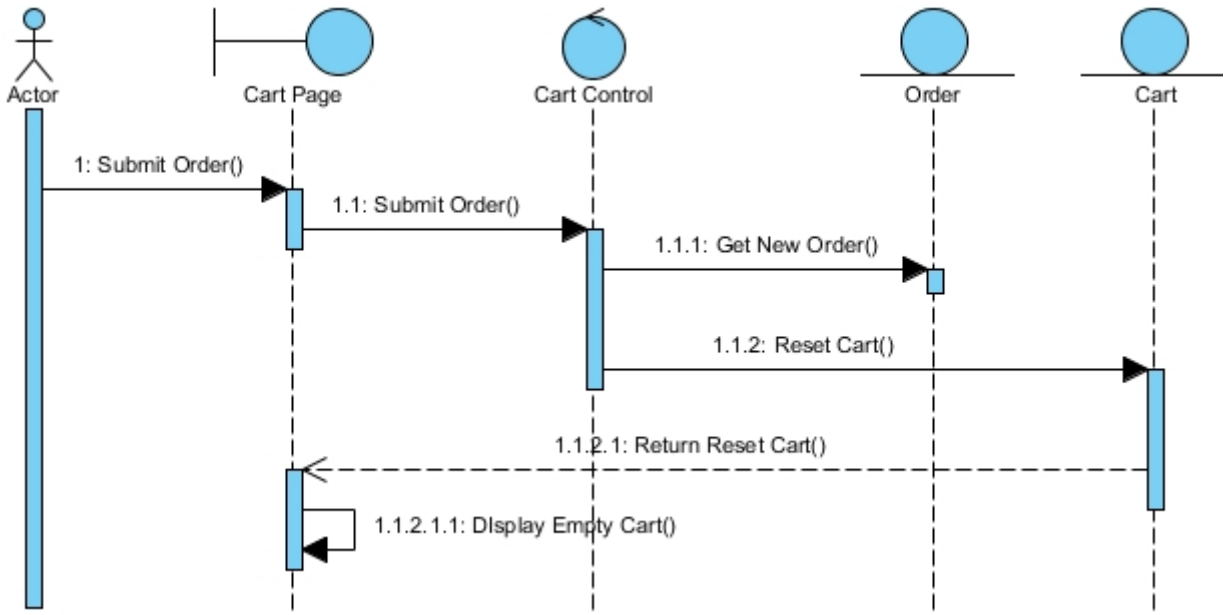


Figure 9. Payment sequence diagram (Landing page)

#### a8) Edit account

This use case allows customers to edit their account information. Customer visits the My Account Page. Then select "edit account" will have a form for the customer to fill out, after filling in the necessary changes, the customer presses the 'Update now' button to update the data. After that, the user's information will be updated by the system. The customer can then view the updated information.

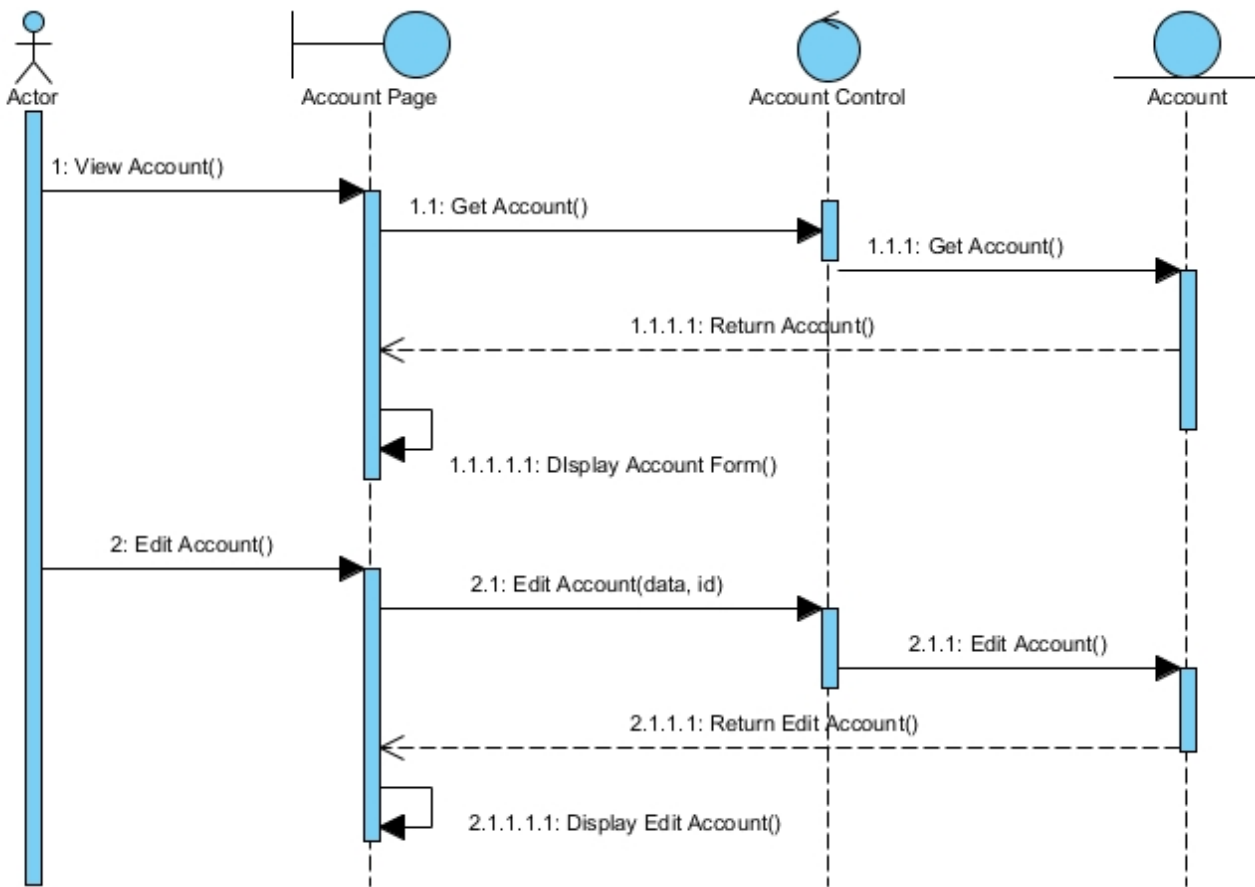


Figure 10. Edit account sequence diagram (Landing page)

#### a9) Delete Account

This use case allows customers to delete their own accounts when no longer needed. First, the customer visits the My Account Page. Then select “delete account”. The website will give the message "Do You Really Want To Delete Your Account!", the customer chooses "Yes, I want to delete" to delete the account, then the customer's account will be deleted from the system. If the customer does not want to delete, they can choose "No, I Don't want to delete".

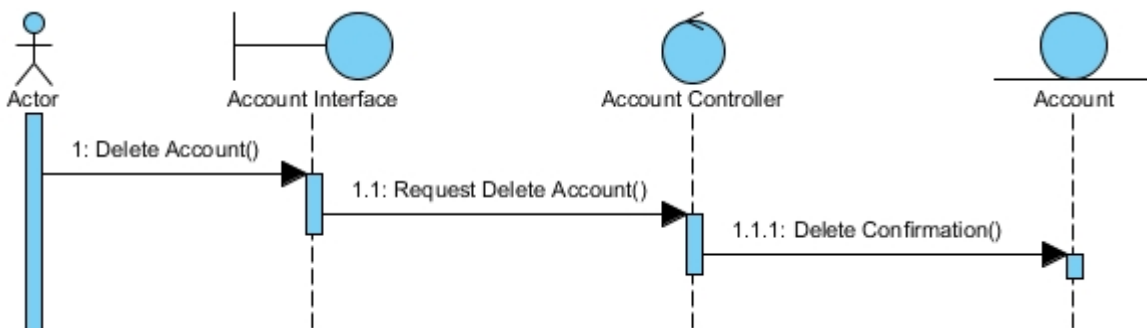


Figure 11. Delete account sequence diagram (Landing page)



### a10) Logout

This use case allows the customer to log out of the account in use. The customer first visits the My Account Page. Then choose “Log Out”. The customer's account will be logged out immediately and returned to the homepage.

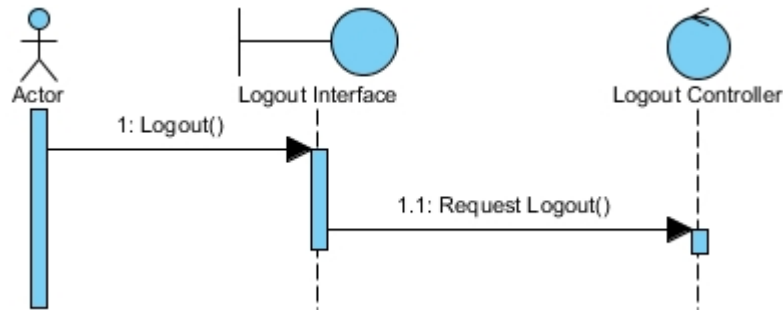


Figure 12. Logout sequence diagram (Landing page)

## b) Admin Page

### b1) Login

This use case demonstrates the access of an account to the system. The account belongs to an admin. After filling in all information in the login form and pressing the “Login” button, the system validates the email and password entered from the account list. The customer will be logged in if the information is valid, else, the system will display an error message and request the admin to fill in the form again.

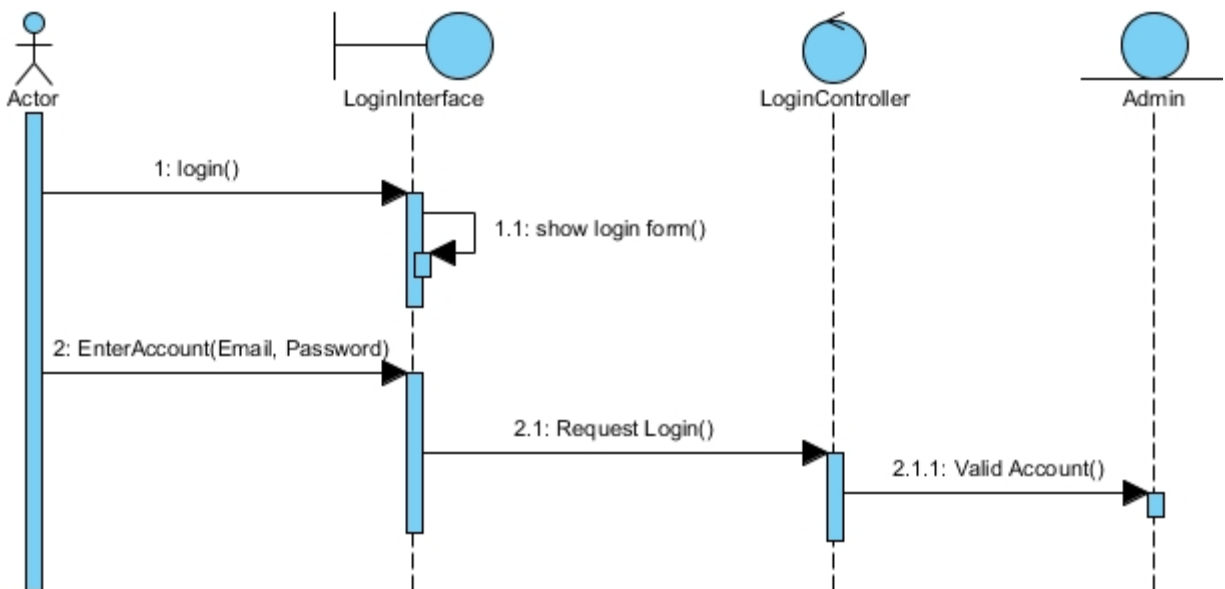


Figure 13. Login sequence diagram (Admin page)

### b2) View dashboard

This use case allows the admin to view the dashboard page. After logging into the admin account, the Dashboard page will appear with some key stats and the latest orders.

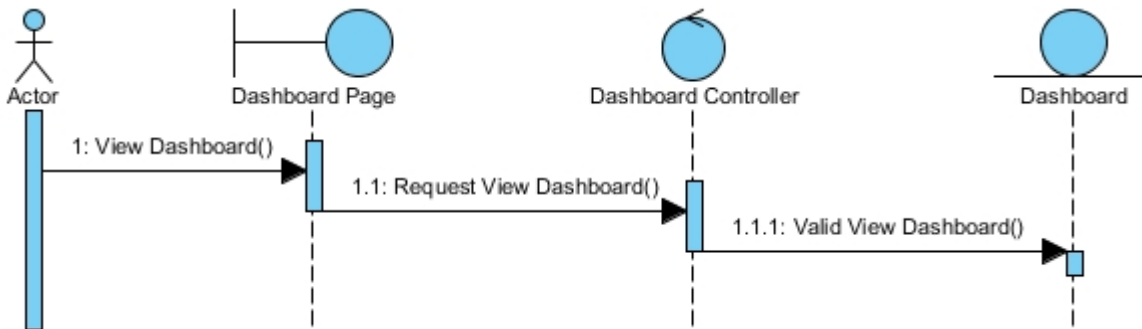


Figure 14. View dashboard sequence diagram (Admin page)

### b3) Manage product

#### b3.1) Insert Product

This use case allows the admin to insert new products into the product list. On the Admin Panel column click "Products" and select "Insert Product". After entering all the information about the new product in the product insert form, select the "Insert product" button at the bottom of the page. The system will update the information of the new product into the database and display the product on the store's website.

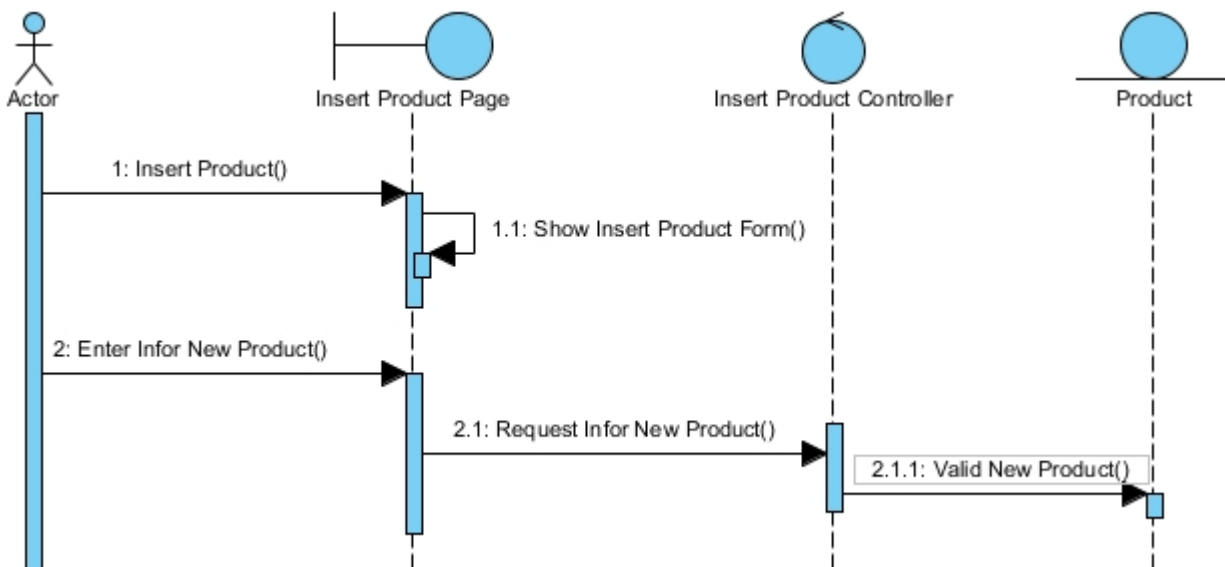


Figure 15. Insert product sequence diagram (Admin page)

#### b3.2) View and Update Product

This use case allows the admin to view and update products in the product list. On the Admin

Panel column click "Products" and select "View Products". All product information will appear on the board. When an admin wants to update a product, he needs to select the "Edit" or "Delete" button of that product. After clicking the "Edit" button, the product editing form will appear, the administrator enters the information to be edited and selects "Update product". The system will update the product's change data into the database.

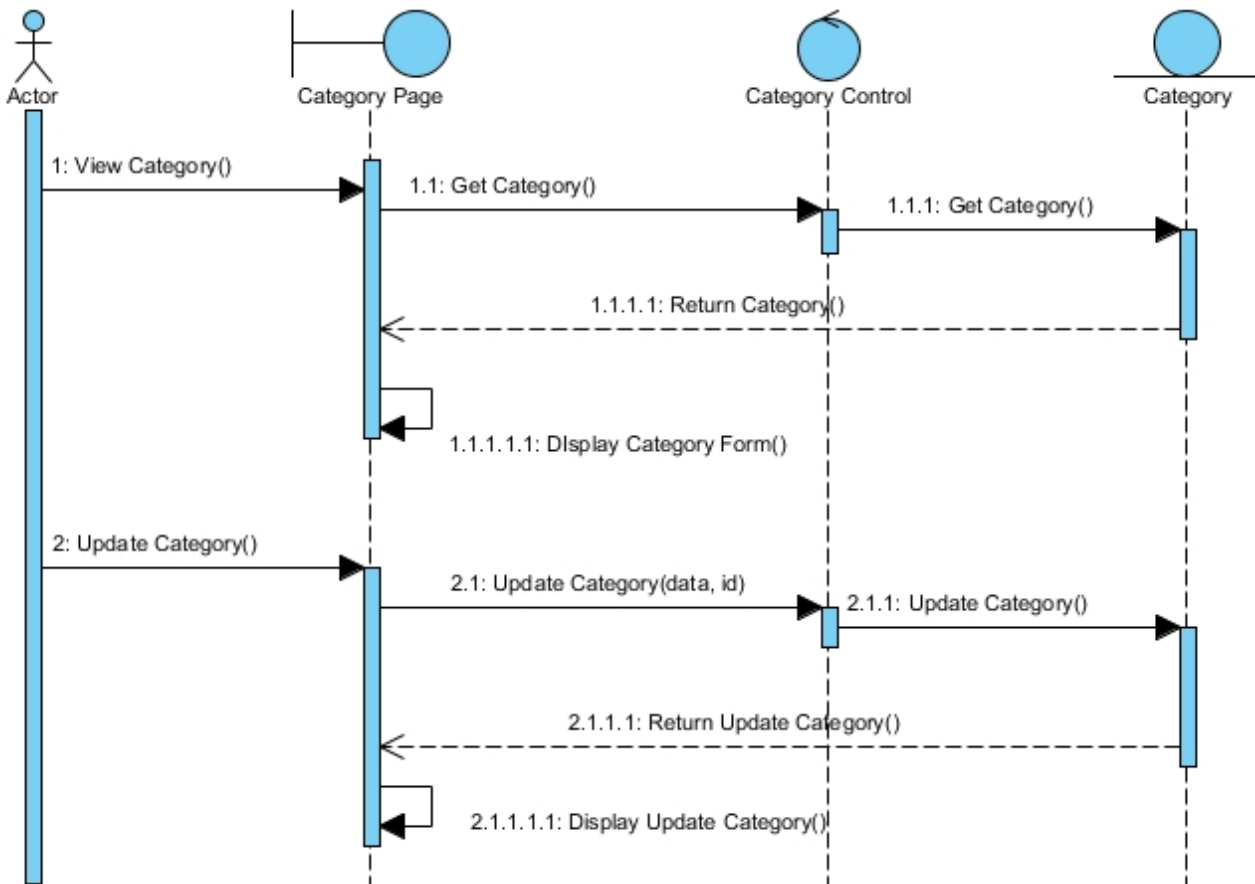


Figure 16. View and update product sequence diagram (Admin page)

#### b4) Manage Bundle

##### b4.1) Insert Bundle

This use case allows the admin to insert new bundles into the bundle list. On the Admin Panel column click "Bundles" and select "Insert Bundle". After entering all the information about the new bundle in the bundle insert form, select the "Insert Bundle" button at the bottom of the page. The system will update the information of the new bundle into the bundle list.

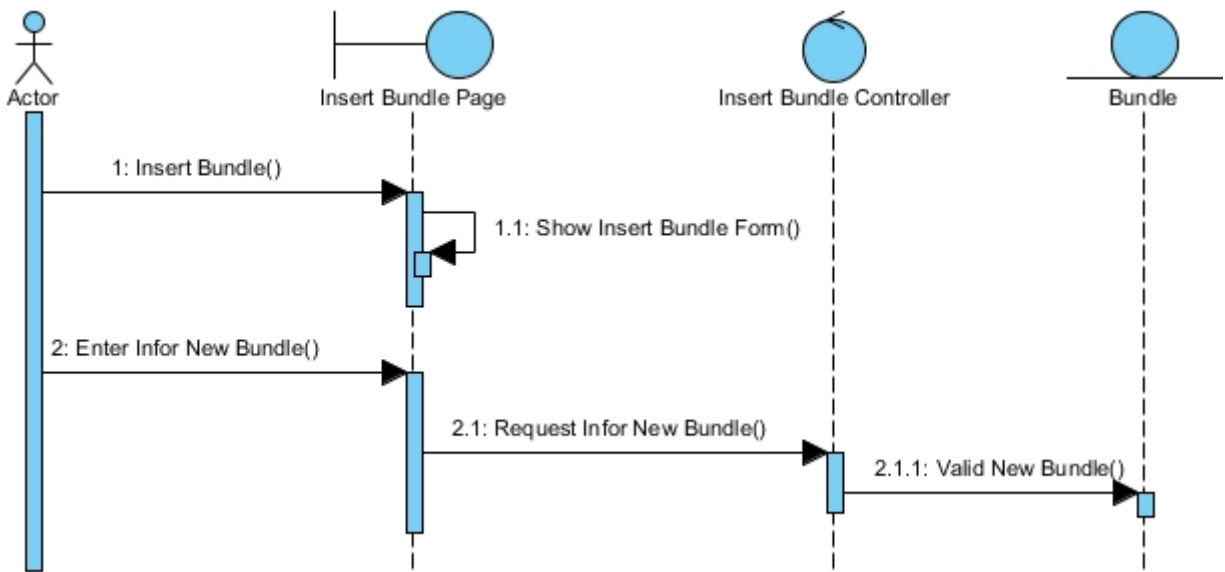


Figure 17. Insert bundle sequence diagram (Admin page)

#### b4.2) View and Update Bundle

This use case allows the admin to view and update bundles in the bundle list. On the Admin Panel column click "Bundles" and select "View Bundles". All bundle information will appear on the board. When an admin wants to update a bundle, he needs to select the "Edit" or "Delete" button of that bundle. After clicking the "Edit" button, the bundle editing form will appear, admin enters the information to be edited and selects "Update Bundle". The system will update the bundle's change data into the database.

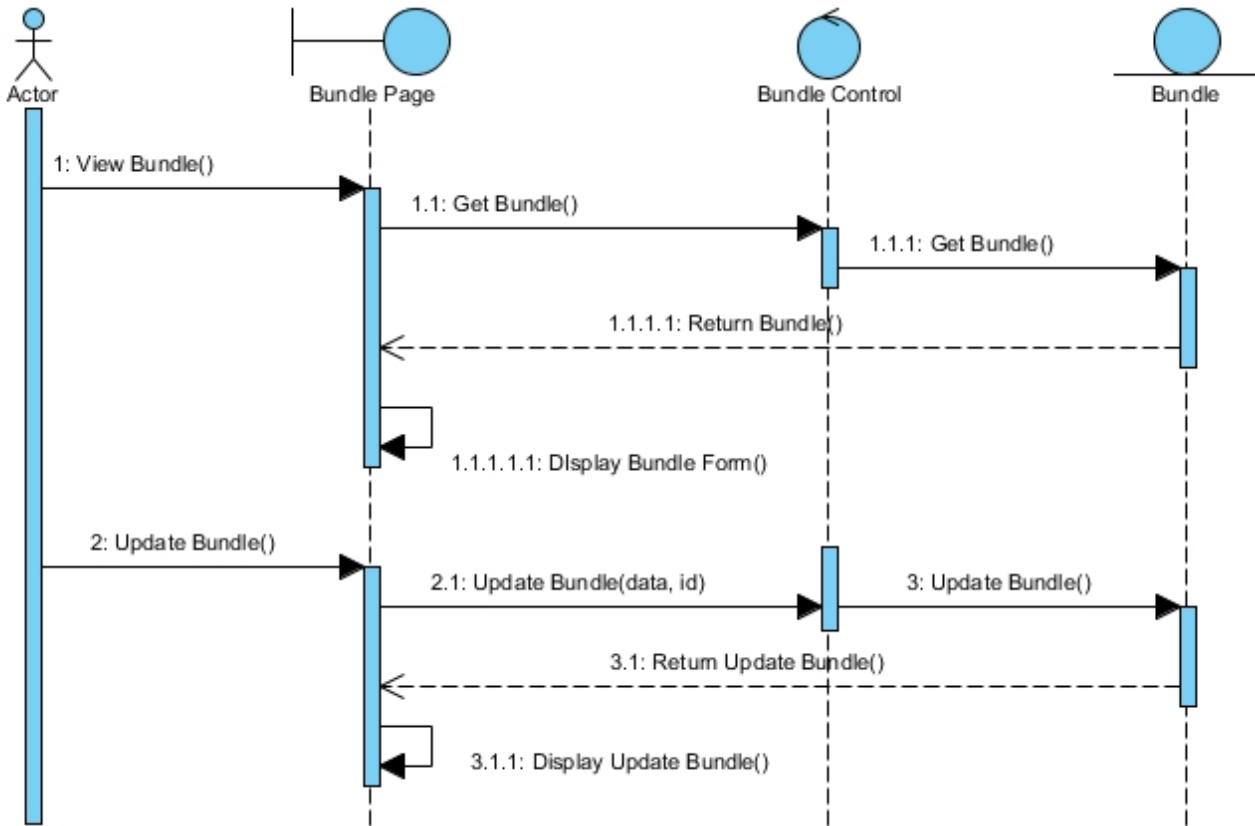


Figure 18. View and update bundle sequence diagram (Admin page)

## b5) Manage Category

### b5.1) Insert Sub Category

This use case allows the admin to insert new sub categories into the sub category list. On the Admin Panel column click " Sub Categories" and select "Insert Sub Category". After entering all the information about the new sub category in the sub category insert form, select the "Submit now" button at the bottom of the page. The system will update the information of the new sub category into the sub category list.

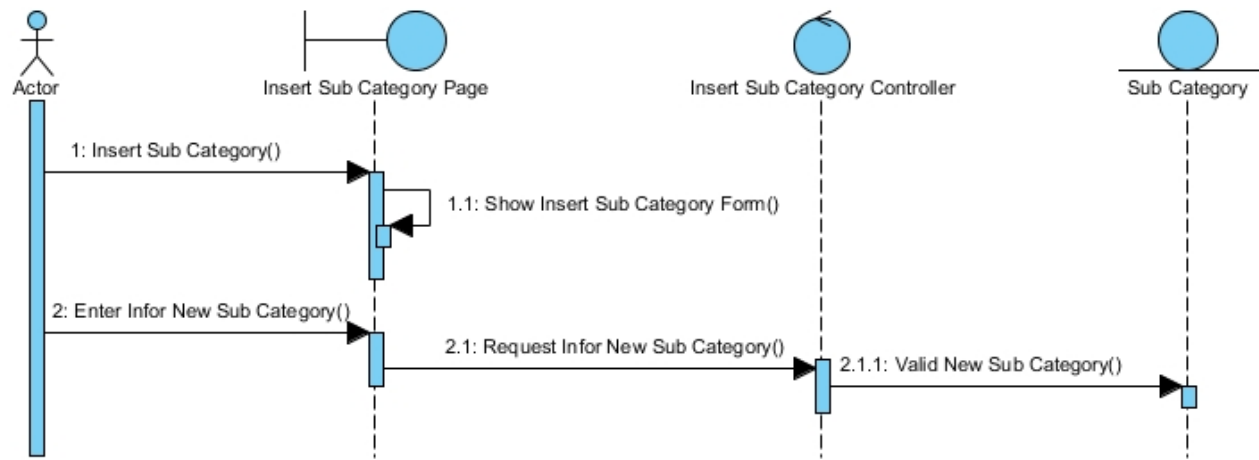


Figure 19. Insert sub category sequence diagram (Admin page)

#### b5.2) View and Update Sub Category

This use case allows the admin to view and update sub categories in the sub category list. On the Admin Panel column click "Sub Category" and select "View Sub Category". All sub category information will appear on the board. When an admin wants to update a sub category, he needs to select the "Edit" or "Delete" button of that sub category. After clicking the "Edit" button, the sub category editing form will appear, admin enters the information to be edited and selects "Update Now". The system will update the sub category's change data into the database.

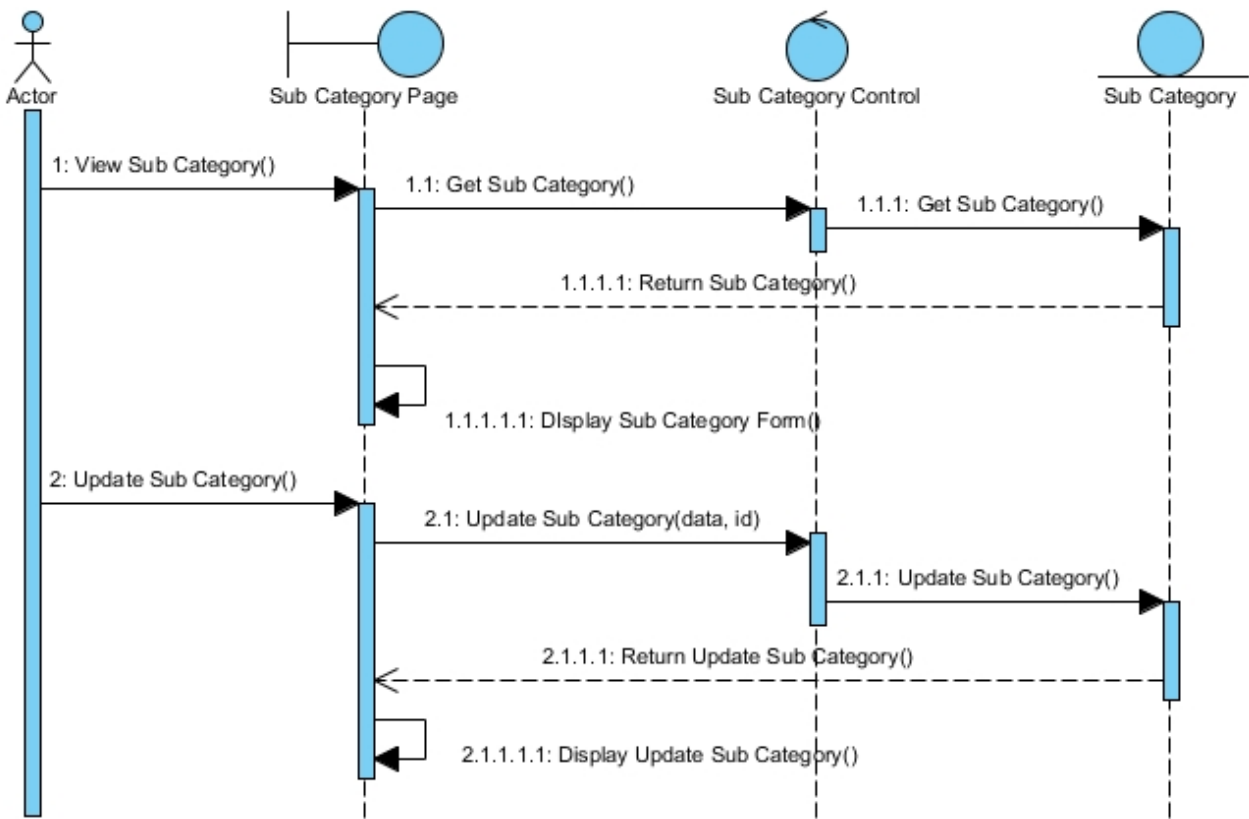


Figure 20. View and update sub category sequence diagram (Admin page)

### b5.3) Insert Category

This use case allows the admin to insert a new category into the category list. On the Admin Panel column click "Categories" and select "Insert Category". After entering all the information about the new category in the category insert form, select the " Insert Category" button at the bottom of the page. The system will update the information of the new category into the category list.

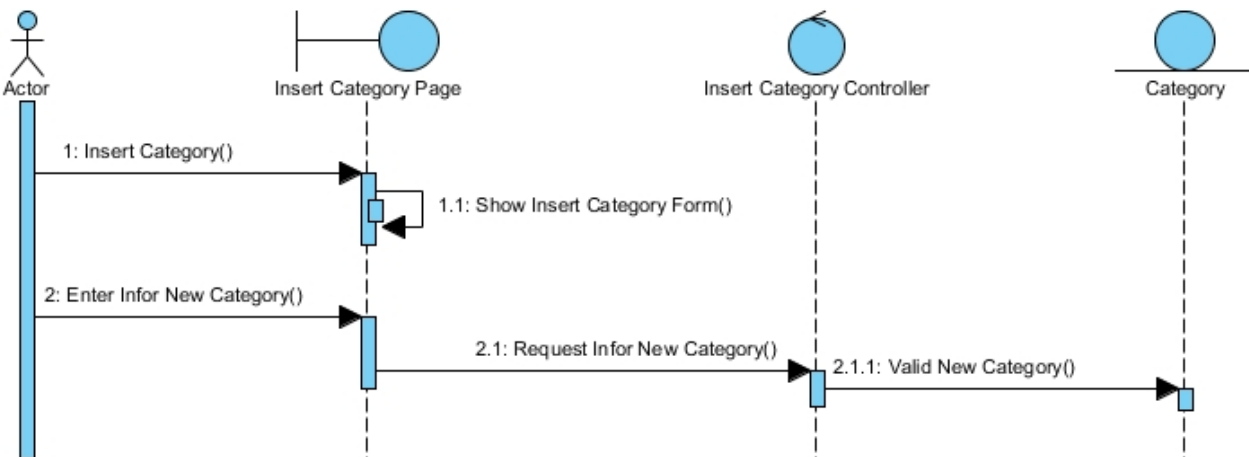


Figure 21. Insert category sequence diagram (Admin page)

#### b5.4) View and Update Category

This use case allows the admin to view and update categories in the category list. On the Admin Panel column click "Category" and select "View Category". All category information will appear on the board. When an admin wants to update a category, he needs to select the "Edit" or "Delete" button of that category. After clicking the "Edit" button, the category editing form will appear, admin enters the information to be edited and selects "Update Category". The system will update the category's change data into the database.

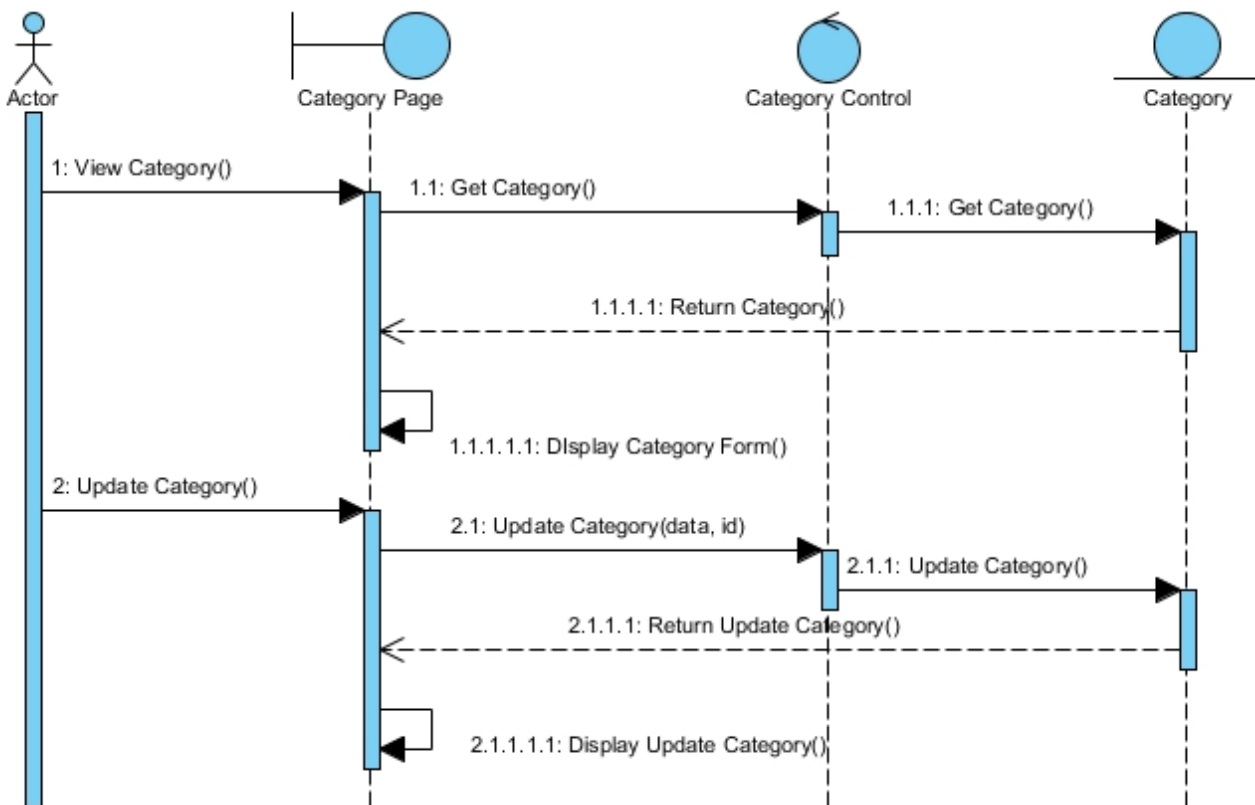


Figure 22. View and update category sequence diagram (Admin page)

### b6) Assign Product to Bundle Relation

#### b6.1) Insert Relation

This use case allows the admin to assign products to bundles relation. On the Admin Panel column, click "Assign Product To Bundle Relation" and select "Insert relation". After selecting the product and bundle for the relation, select the "Insert relation" button at the bottom of the page. The system will update the information of the new relation to the relation list.



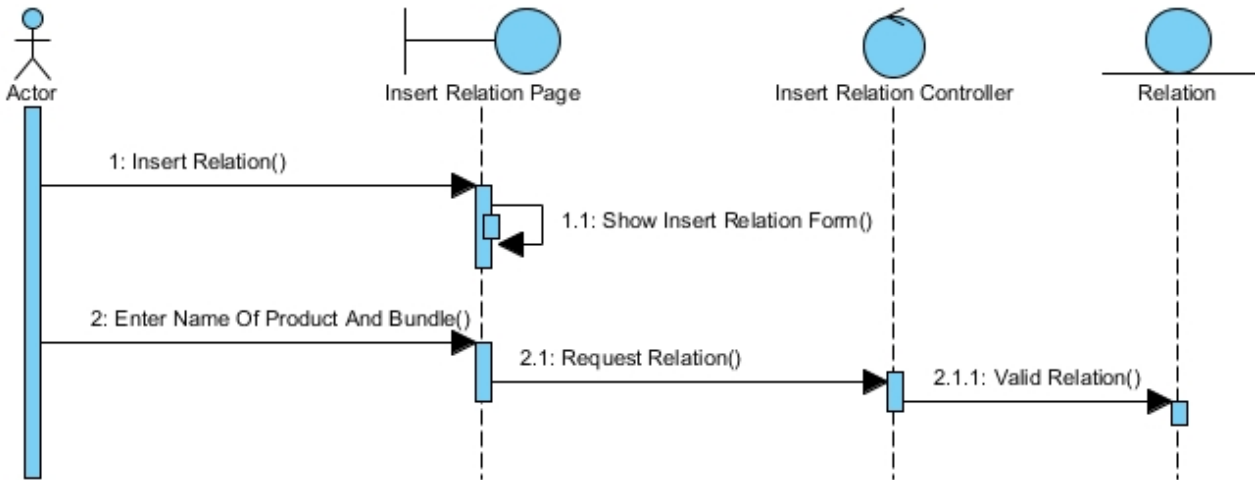


Figure 23. Insert relation sequence diagram (Admin page)

#### b6.2) View and Update Relation

This use case allows the admin to view and update the relation in the relation list. On the Admin Panel column click "Assign Product To Bundle Relation" and select "View Relation". All relation information will appear on the board. When an admin wants to update a relation, he needs to select the "Edit" or "Delete" button of that relation. After clicking the "Edit" button, the relation editing form will appear, admin enters the information to be edited and selects "Update Relation". The system will update the relation's change data into the database.

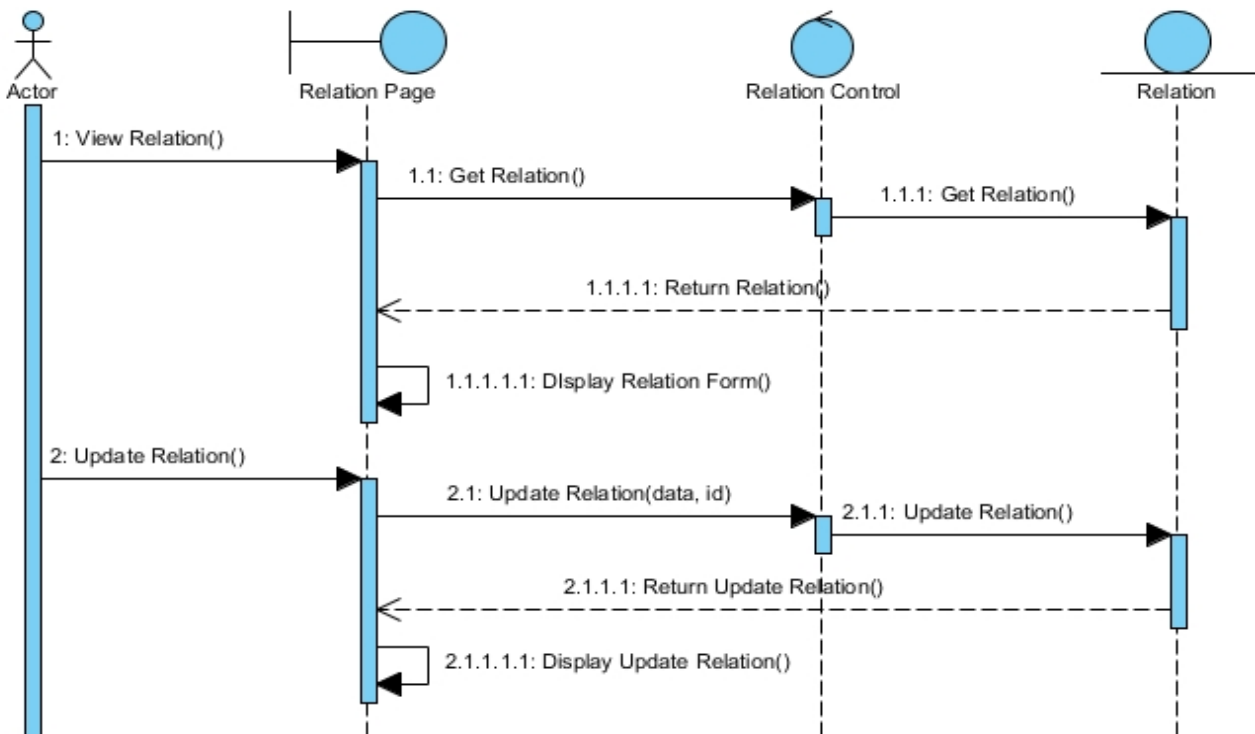


Figure 24. View and update relation sequence diagram (Admin page)

## b7) Manage Manufacturer

### b7.1) Insert Manufacturer

This use case allows the admin to insert a new manufacturer into the manufacturer list. On the Admin Panel column click "Manufacturers" and select "Insert Manufacturer ". After entering all the information about the new manufacturer in the manufacturer insert form, select the "Insert Manufacturer " button at the bottom of the page. The system will update the information of the new manufacturer into the manufacturer list.

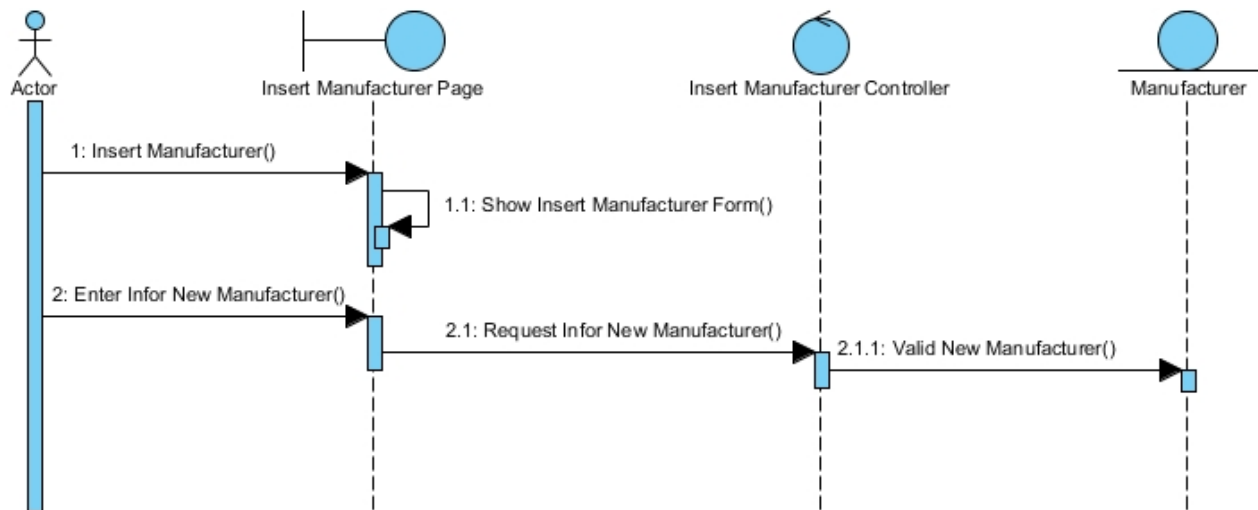


Figure 25. Insert manufacturer sequence diagram (Admin page)

### b7.2) View and Update Manufacturer

This use case allows the admin to view and update the manufacturer in the manufacturer list. On the Admin Panel column click "Manufacturer" and select "View Manufacturer". All manufacturer information will appear on the board. When an admin wants to update a manufacturer, he needs to select the "Edit" or "Delete" button of that manufacturer. After clicking the "Edit" button, the manufacturer editing form will appear, admin enters the information to be edited and selects "Update Manufacturer". The system will update the manufacturer's change data into the database.

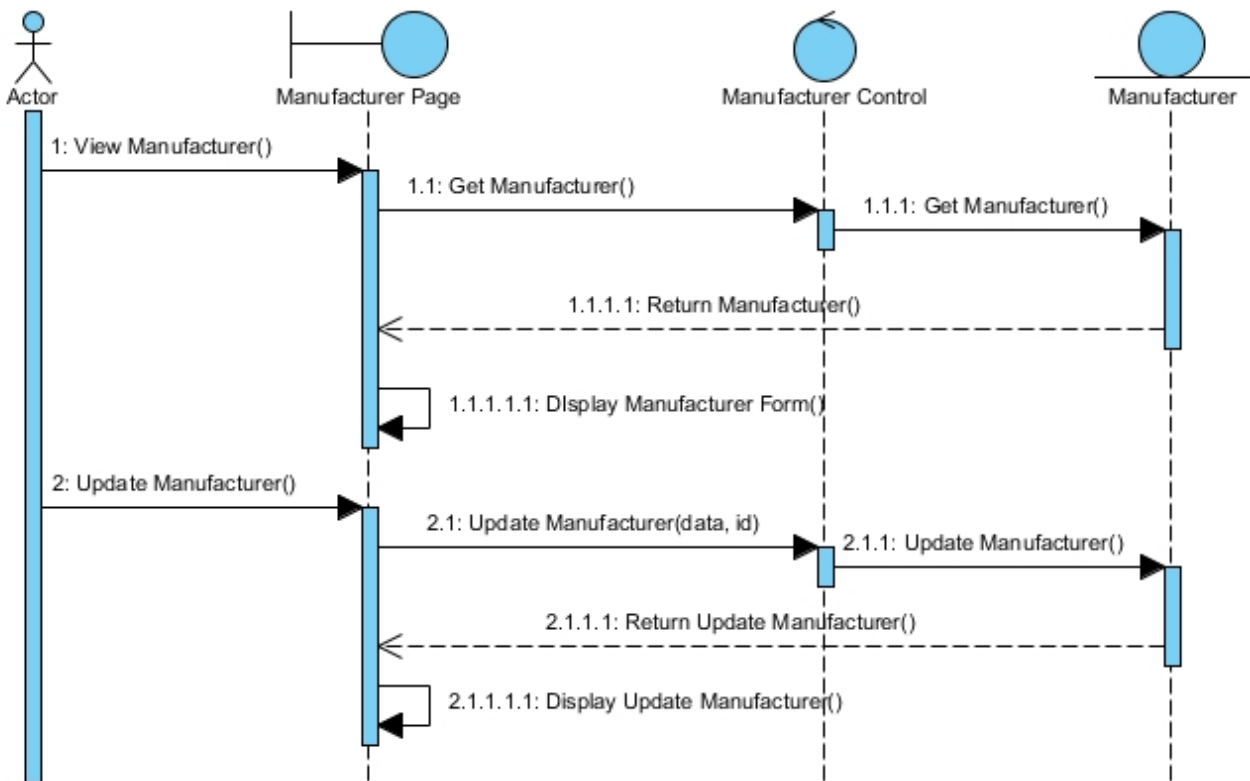


Figure 26. View and update manufacturer sequence diagram (Admin page)

### b8) Update contact us

This use case allows the admin to update information on the contact us page. On the Admin Panel column click "Update Contact us". After entering contact edit information in the edit contact us form, select the "Update Contact Us" button at the bottom of the page. The system will update contact information on the contact us page.

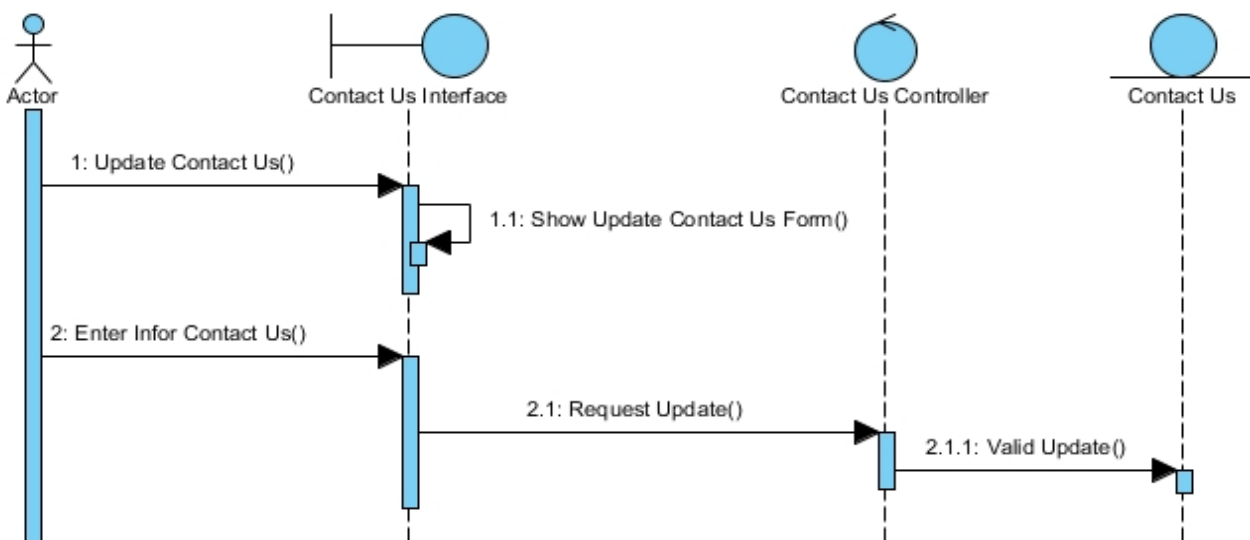


Figure 27. Update contact us sequence diagram (Admin page)

### b9) Update About Us

This use case allows the admin to update information on the about us page. On the Admin Panel column click "Update About Us". After entering the edit information in the edit about us form, select the "Update About Us" button at the bottom of the page. The system will update the information on the about us page.

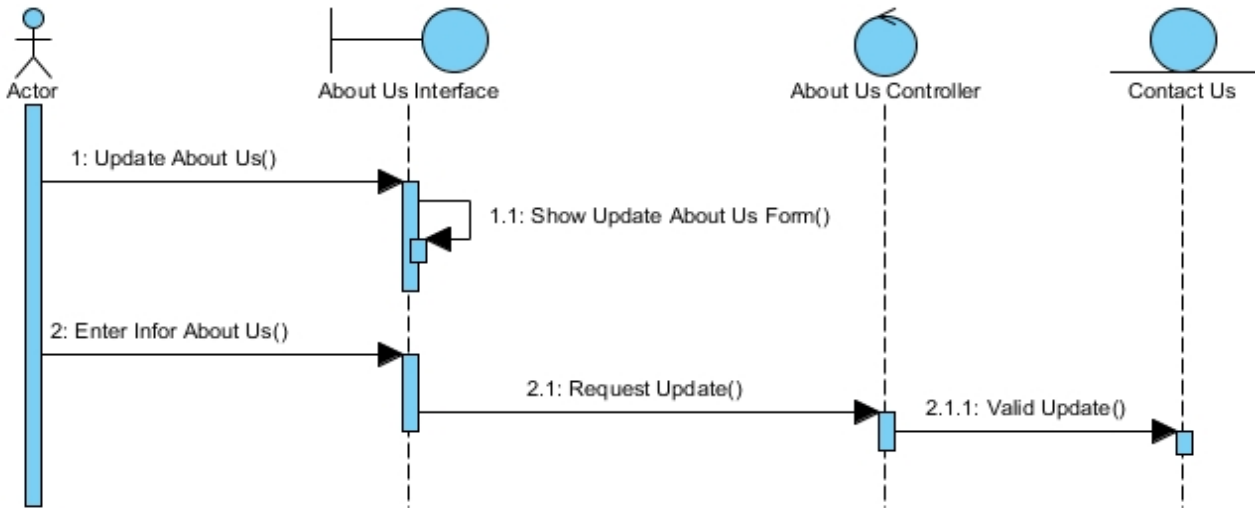


Figure 28. Update about us sequence diagram (Admin page)

### b10) Manage Coupon

#### b10.1) Insert Coupon

This use case allows the admin to insert a new coupon into the coupon list. On the Admin Panel column click "Coupon" and select "Insert Coupon". After entering all the information about the new coupon in the coupon insert form, select the "Insert Coupon" button at the bottom of the page. The system will update the information of the new coupon into the coupon list.

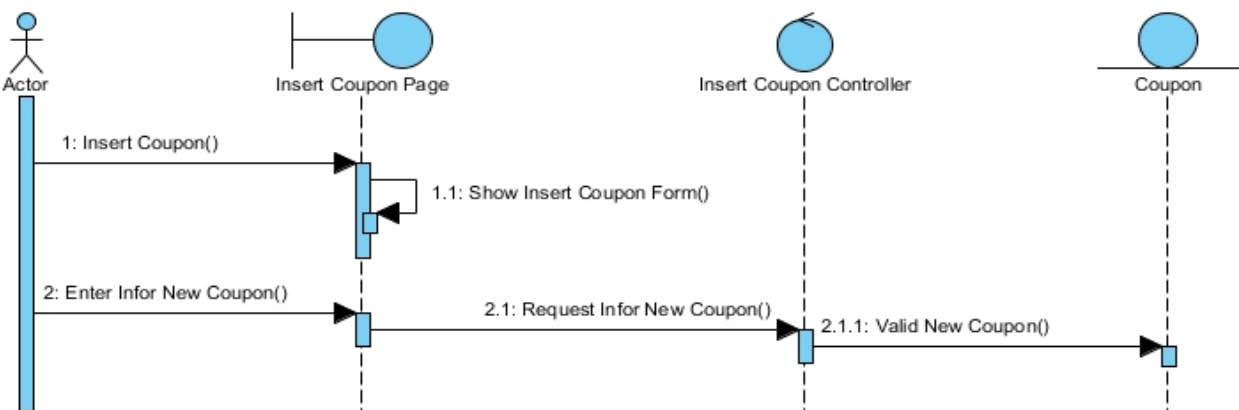


Figure 29. Insert coupon sequence diagram (Admin page)

### b10.2) View and Update Coupon

This use case allows the admin to view and update coupons in the coupon list. On the Admin Panel column click "Coupon" and select "View Coupon". All coupon information will appear on the board. When an admin wants to update a coupon, he needs to select the "Edit" or "Delete" button of that coupon. After clicking the "Edit" button, the coupon editing form will appear, admin enters the information to be edited and selects "Update Coupon". The system will update the coupon's change data into the database.

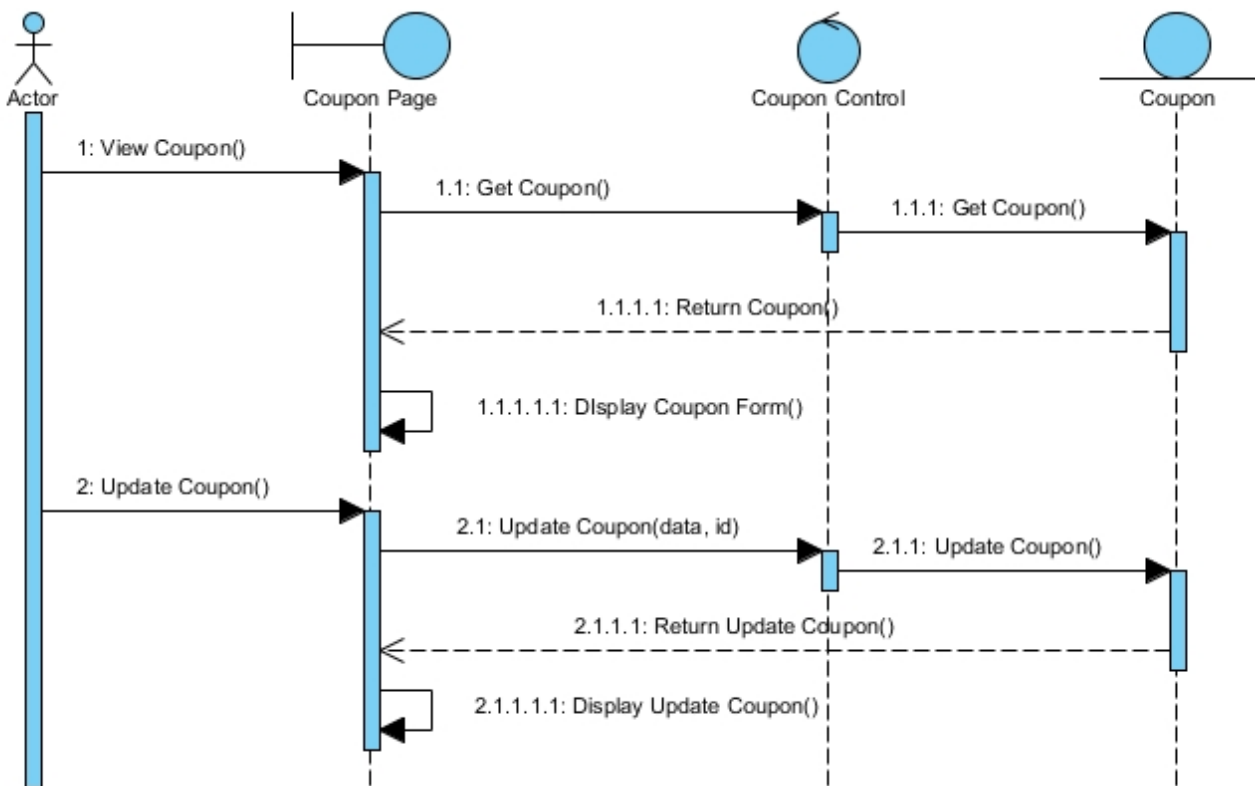


Figure 30. View and update coupon sequence diagram (Admin page)

### b11) Update Term

This use case allows the admin to update information on the term page. On the Admin Panel column click "Update Term". After entering term edit information in the edit term form, select the "Update Term" button at the bottom of the page. The system will update term information on the term page.

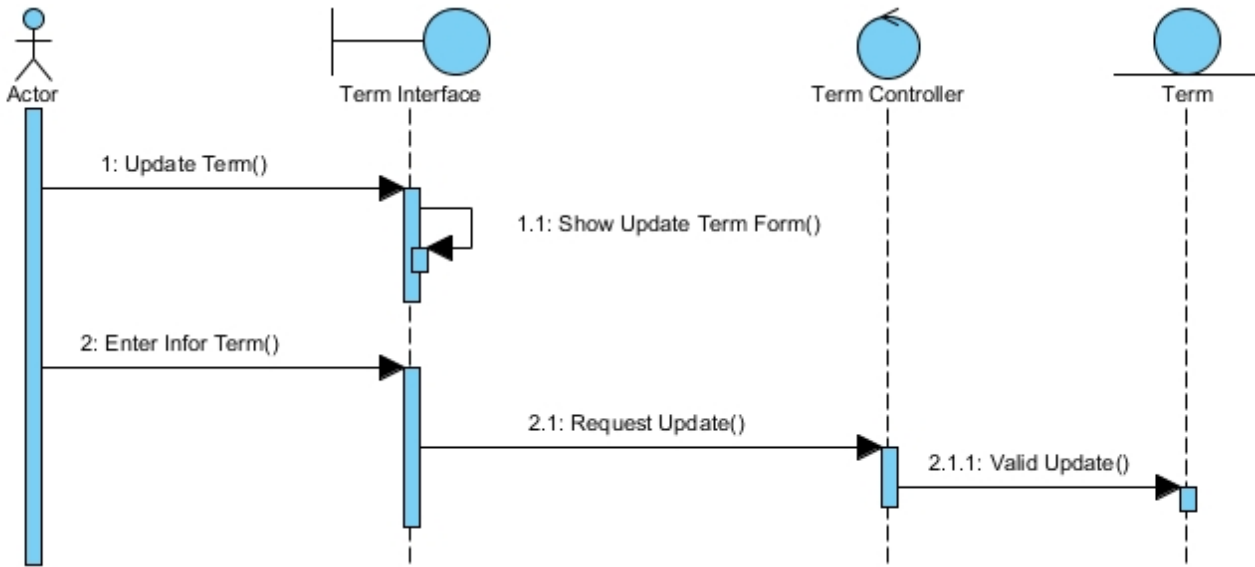


Figure 31. Update term sequence diagram (Admin page)

### b12) View Customer

This use case allows the admin to view customers in the customer list. On the Admin Panel column click "View Customer". All relation information will appear on the board. Admin can also delete a customer account by clicking the "Delete" button.

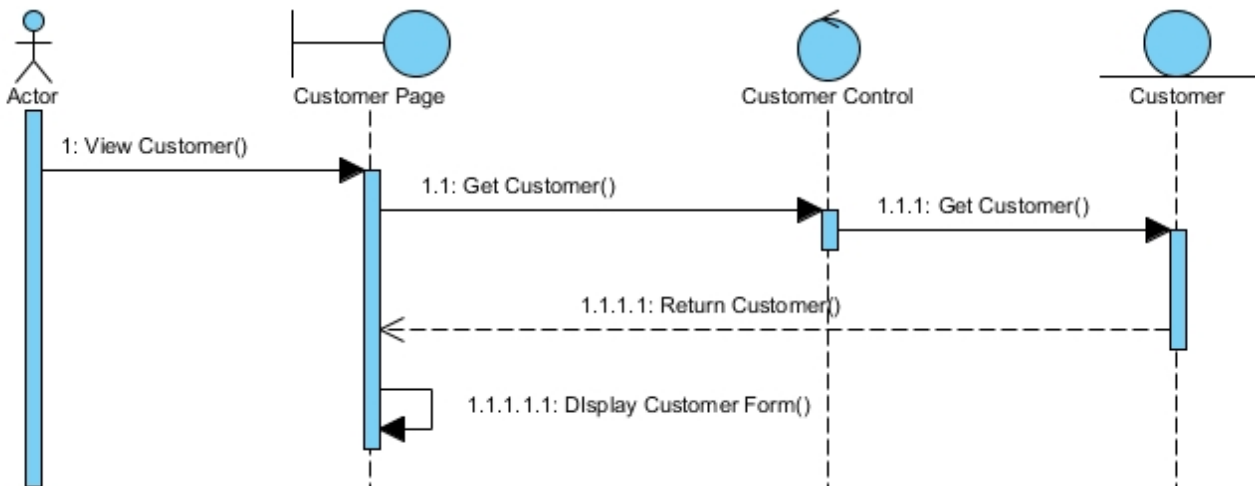


Figure 32. View customer sequence diagram (Admin page)

### b13) View Order

This use case allows the admin to view orders in the order list. On the Admin Panel column click "View Order". All relation information will appear on the board. Admin can also delete an order by clicking the "Delete" button.

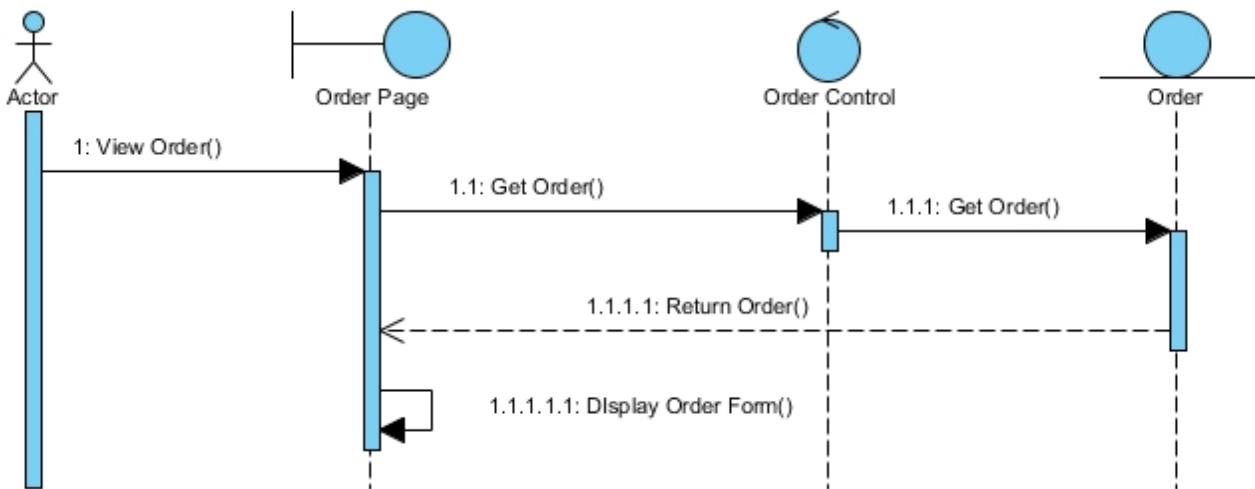


Figure 33. View order sequence diagram (Admin page)

#### b14) View Payment

This use case allows the admin to view the payment in the payment list. On the Admin Panel column click "View payment". All relation information will appear on the board. Admin can also delete a payment by clicking the "Delete" button.

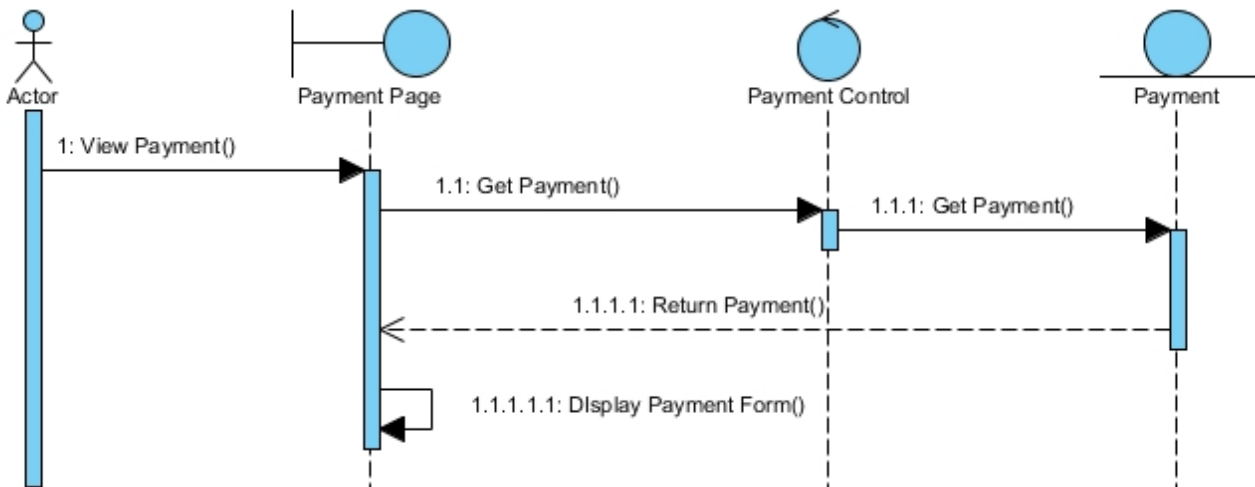


Figure 34. View payment sequence diagram (Admin page)

#### b15) Manage User

##### b15.1) Insert User

This use case allows the admin to insert a new user into the user list. On the Admin Panel column click "User" and select "Insert User". After entering all the information about the new user in the user insert form, select the "Insert User" button at the bottom of the page. The system will update the information of the new user into the user list.

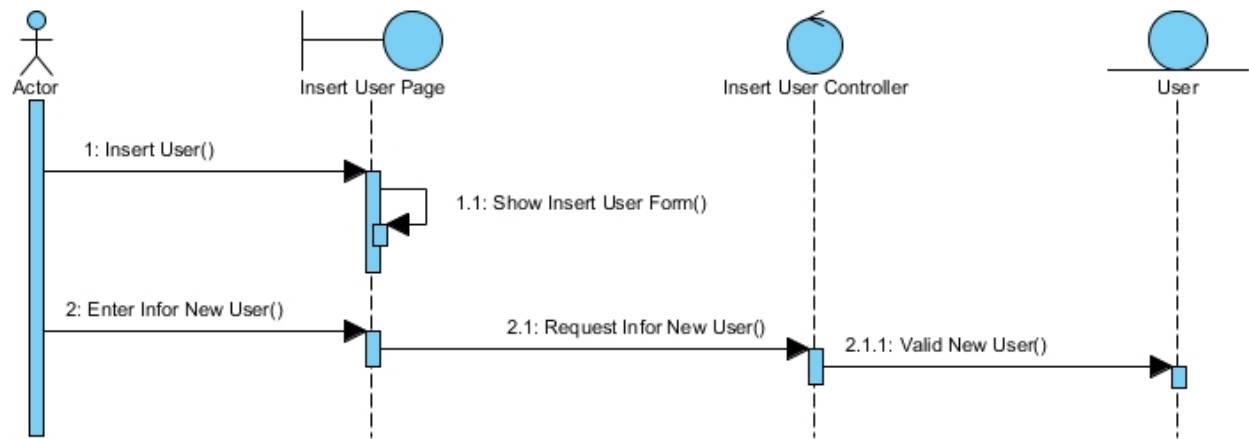


Figure 35. Insert user sequence diagram (Admin page)

#### b15.2) View User

This use case allows the admin to view users in the user list. On the Admin Panel column click "View User". All relation information will appear on the board (User Name, Email, Image, Country, Job). Admin can also delete a user by clicking the "Delete" button.

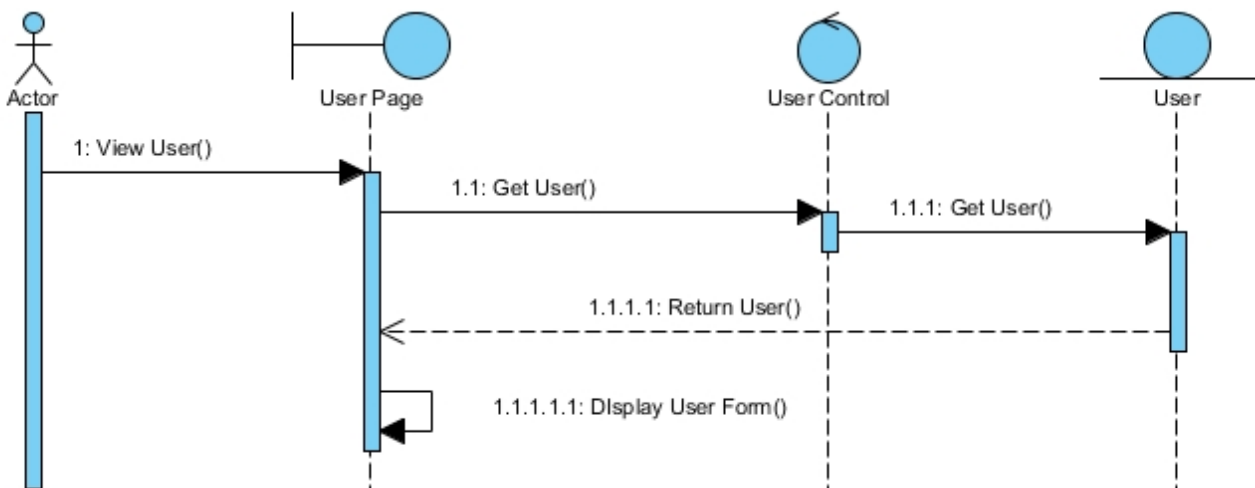


Figure 36. View user sequence diagram (Admin page)

#### b15.3) Edit Profile

This use case allows the admin to edit his profile. On the Admin Panel column click "Users" and select "Edit Profile". After entering all information (Name, Email, City...), select the "Update User" button at the bottom of the page.



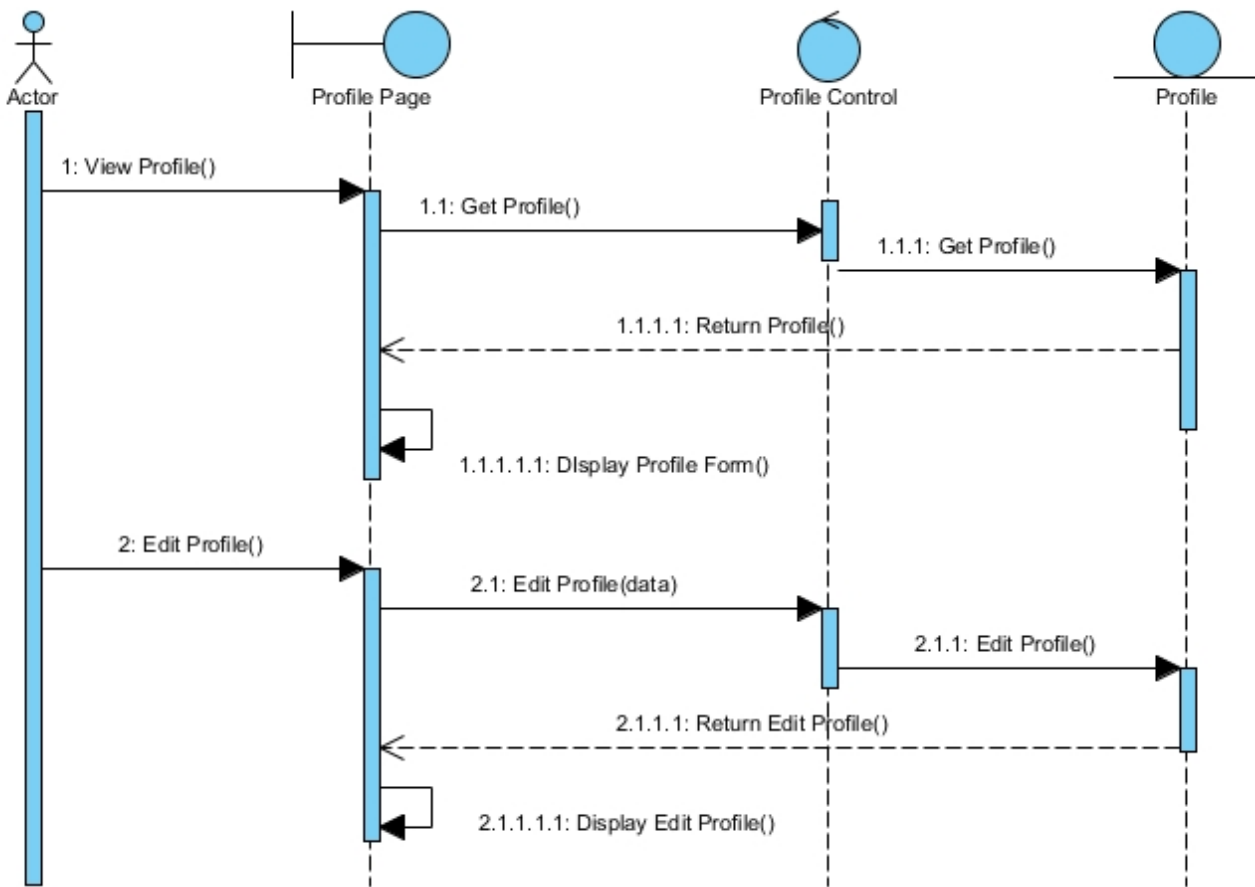


Figure 37. Edit profile sequence diagram (Admin page)

## b16) Logout

This use case allows the admin to log out of the account in use. Click on "Admin" and select "Log Out". The administrator account will be logged out of the system.

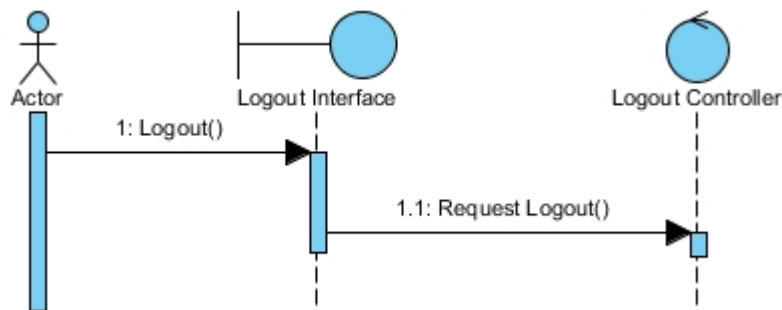


Figure 38. Logout sequence diagram (Admin page)

### 3. Class diagrams

#### a) Landing page

##### a1) View Product

This diagram allows the “Customer” to view “Product”. Customers can find information about all products of the shop like: name, price, some images, description.

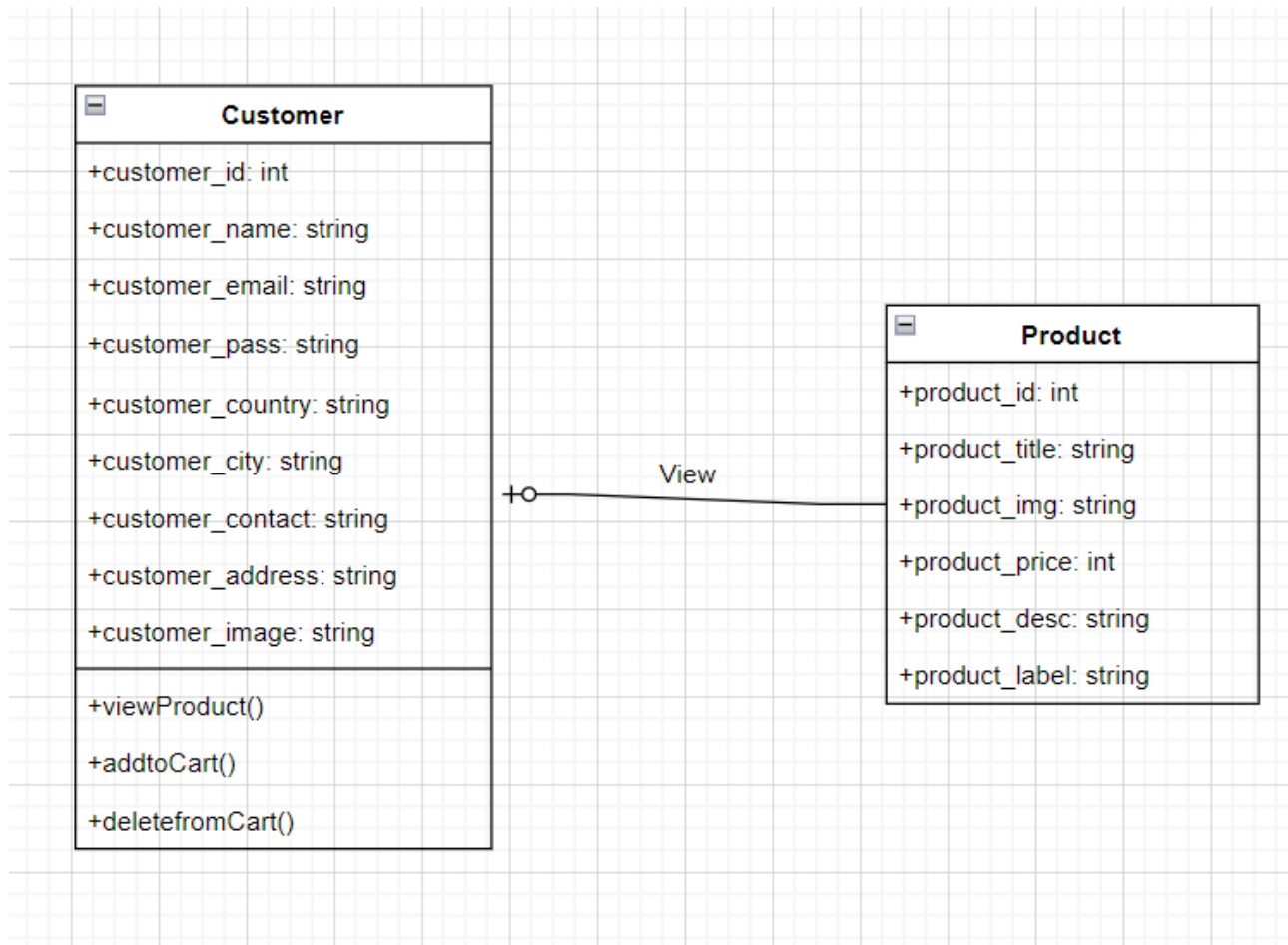


Figure 39. View product class diagram (landing page)

##### a2) View a product

This diagram allows the “Customer” to view a “Product”. Customers can find information about products and add products to the Wishlist.

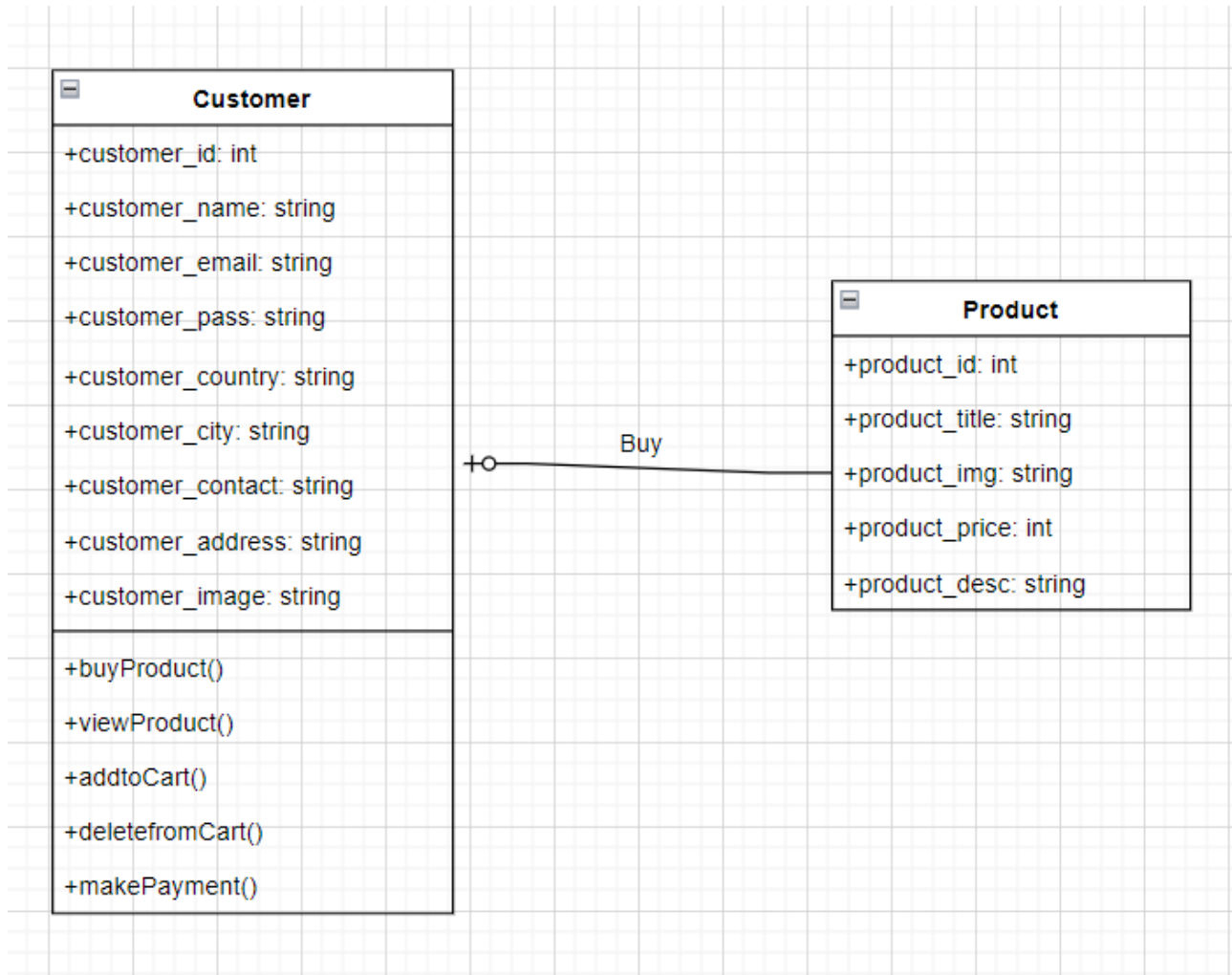


Figure 40. View a product class diagram (landing page)

### a3) Add product to wish list

This diagram allows the customer to view the wishlist: view the products they added and remove or buy items.

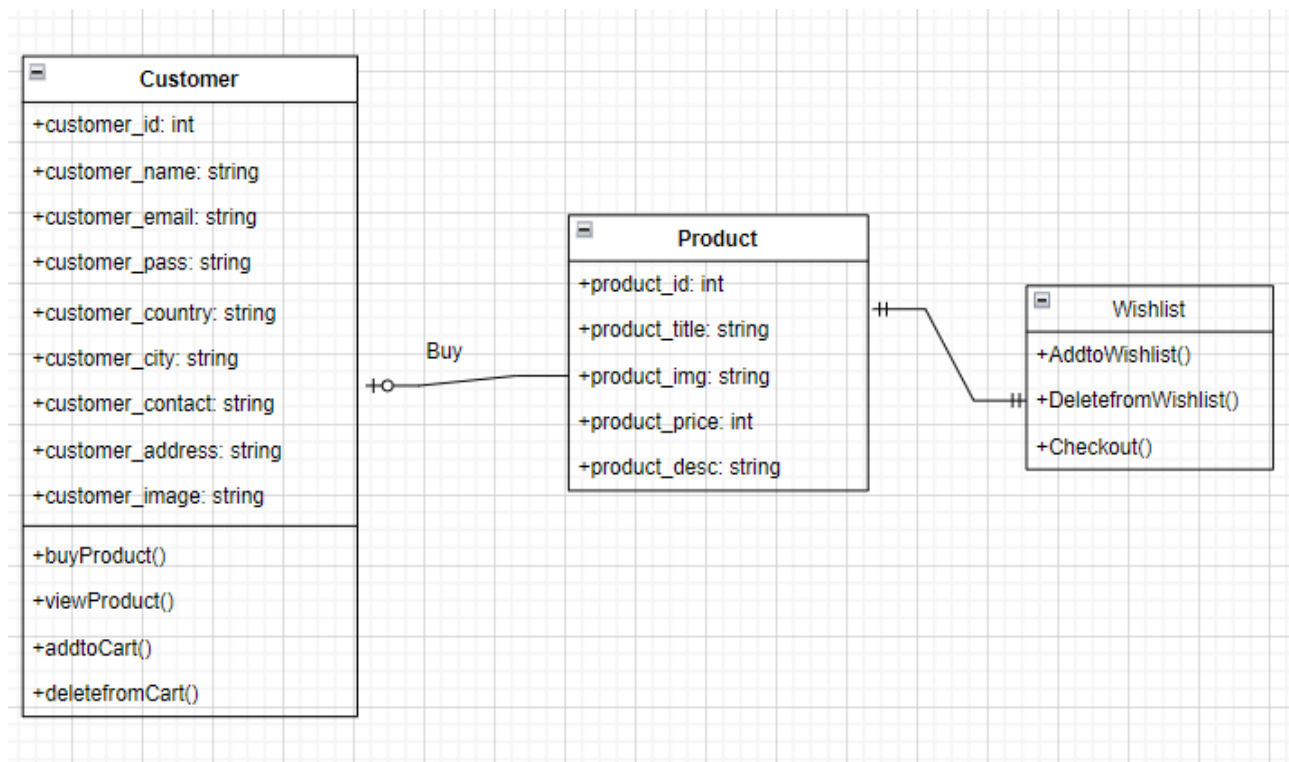


Figure 41. Add product to wishlist class diagram (landing page)

#### a4) Add product to cart

Customers view the products that are added to cart. They can change the quantity, remove, add product before check it out.

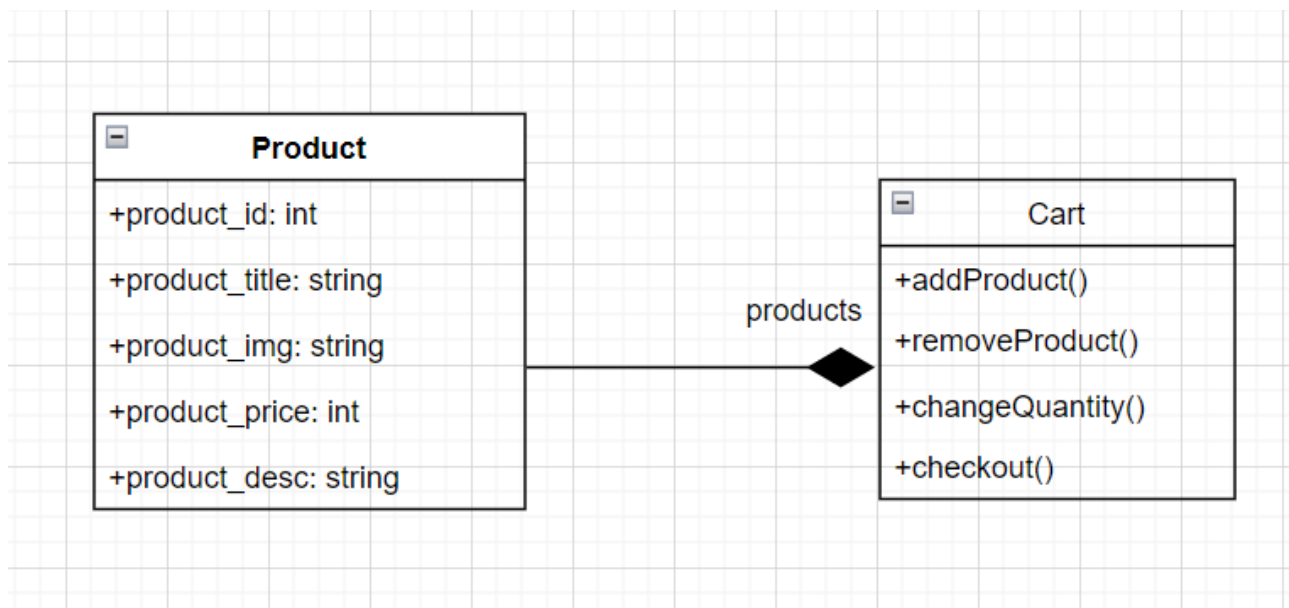


Figure 42. Add product to cart class diagram (landing page)

### a5) Payment

This diagram allows customers to view the payment like: date, amount, time.

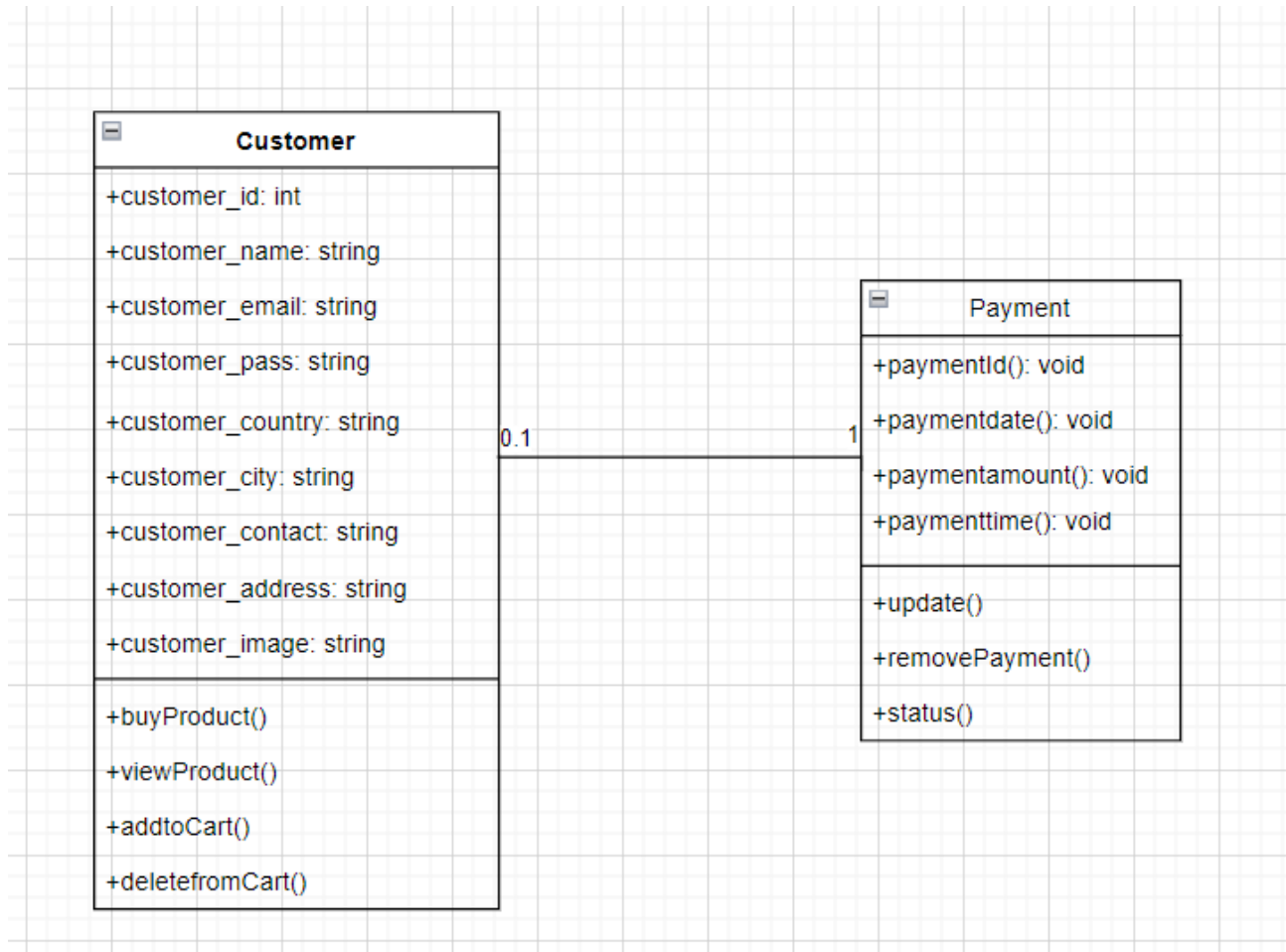


Figure 43. Payment class diagram (landing page)

### b) Admin Page

#### b1) Manage Product and Bundle

Admin manages all activities of the Product and Bundle like: add, update, delete, search, ..

Product and bundle use same attributes: product\_id, product\_name, product\_category, product\_keyword, product\_bundle, product\_price..

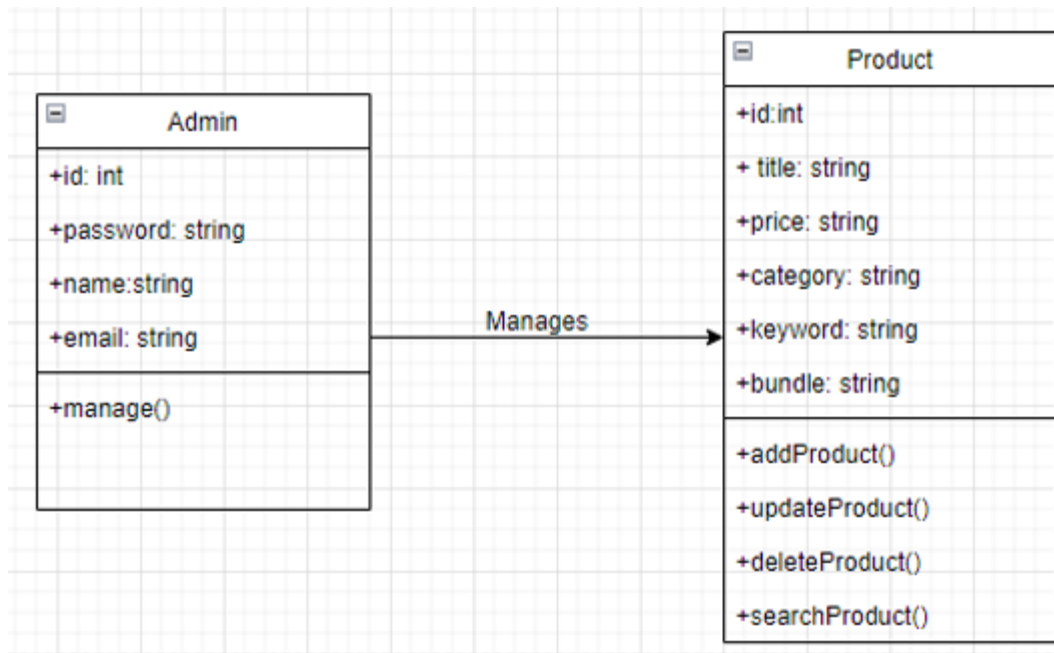


Figure 44. Manage product class diagram (Admin page)

## b2) Manage Category

Admin can categorize products through Category. He has the right to view, add, delete, update..

Category attributes: category\_id, category\_name....

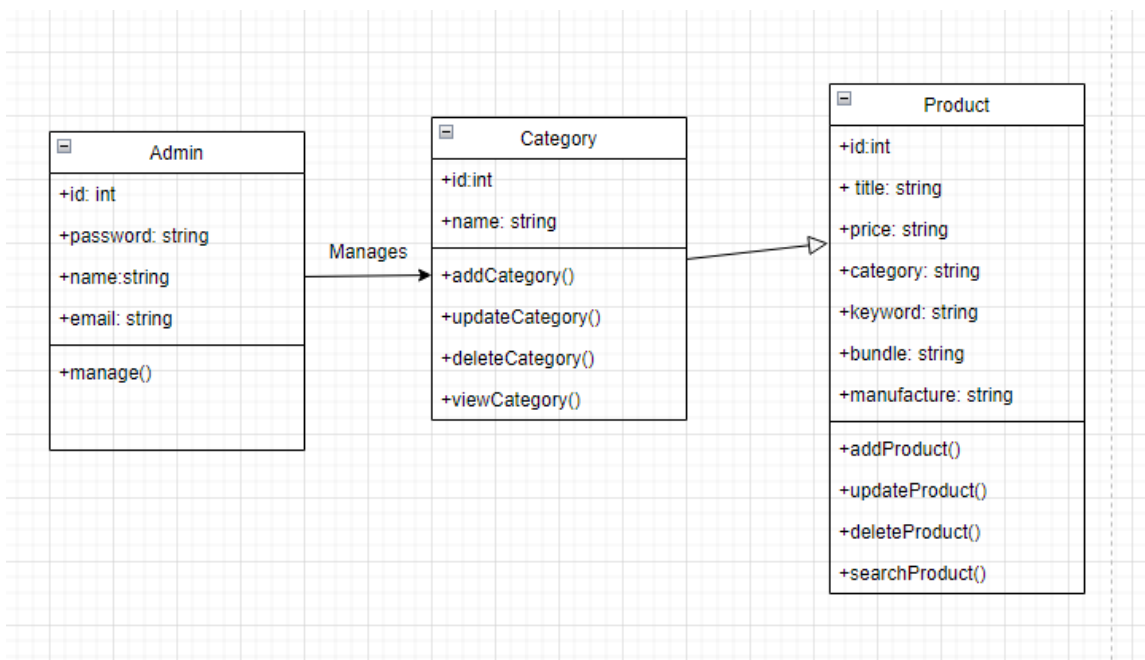


Figure 45. Manage category class diagram (Admin page)

### b3) Assign Product to Bundle Relation

In this section, Admin has the right to create bundles for products. It's called relation.

Relation attributes: relation\_id, relation\_title, relation\_product, relation\_bundle..

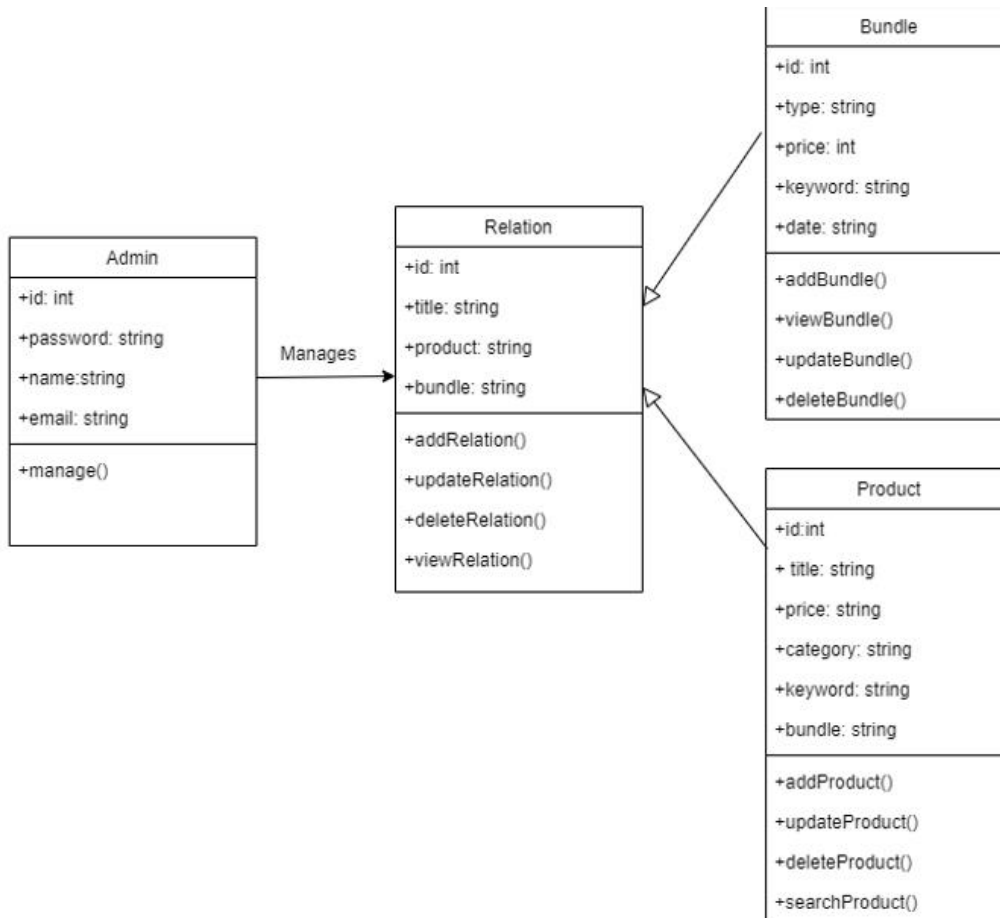


Figure 46. Assign product to bundle relation class diagram (Admin page)

### b4) Manage Manufacturer

In this section, Admin manages the manufactures of the products. Admin can: add, view, delete, update manufactures.

Manufactures attributes: id, title

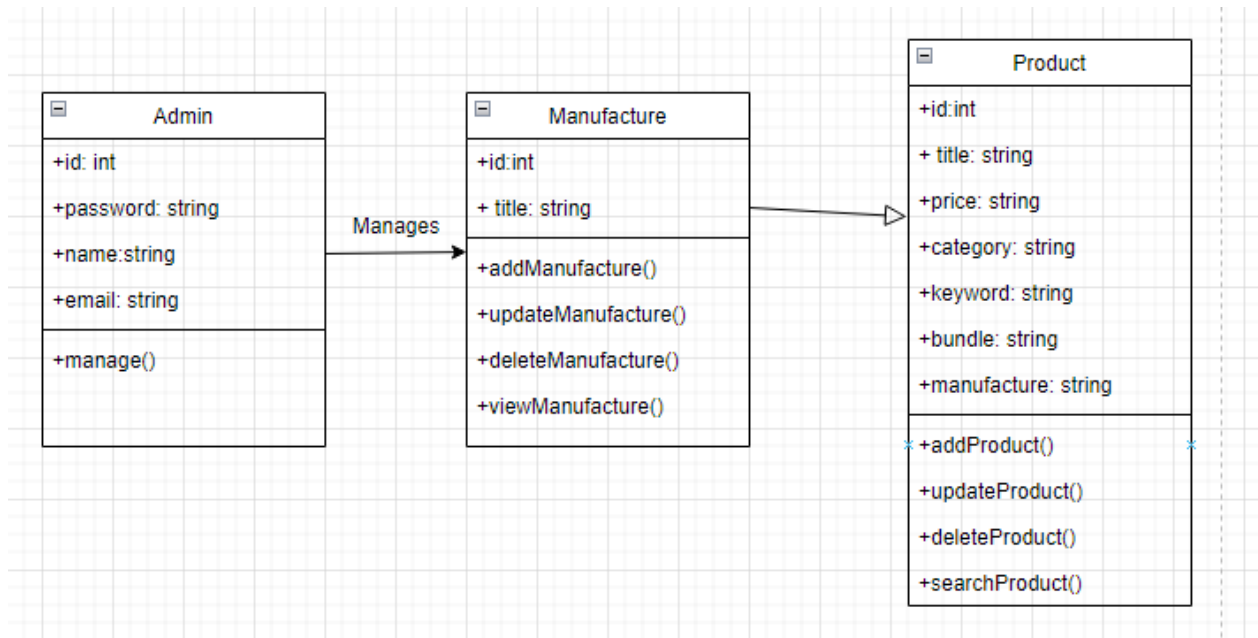


Figure 47. Manage manufacturer class diagram (Admin page)

### b5) Manage Coupon

Admin has the right to manage coupons of all products. He can: add, view, delete, update coupons.

Coupon attributes: id, title, price, code,...

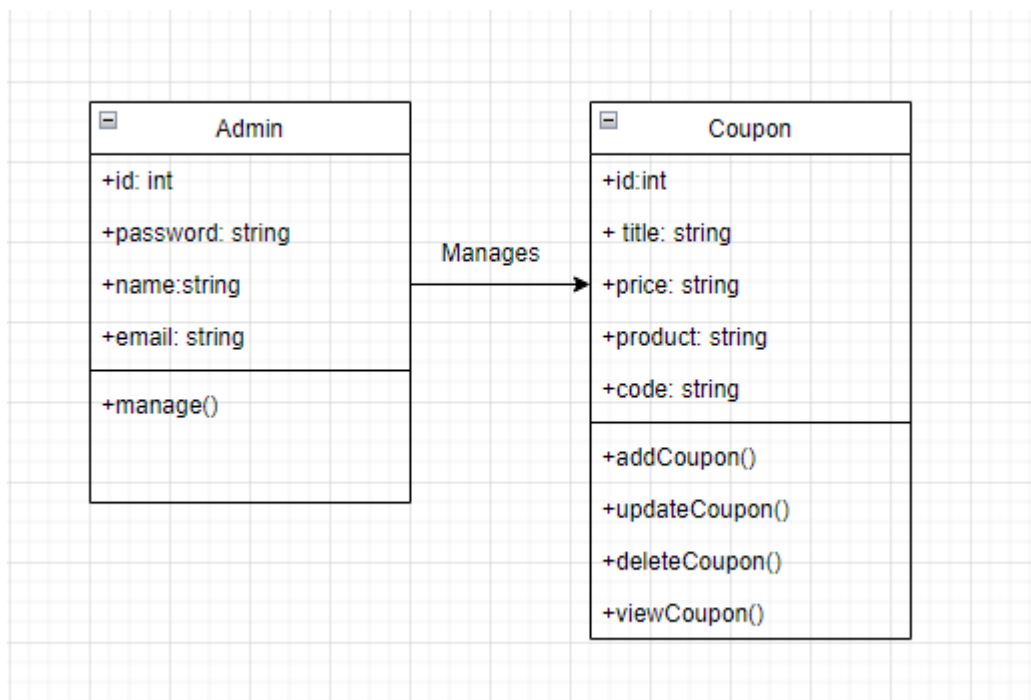


Figure 48. Manage coupon class diagram (Admin page)



## b6) Manage User

Admin has control over all users on the site. Admin can: view, delete, update, add users.

User attributes: id, name, email. phone, email, job, password....

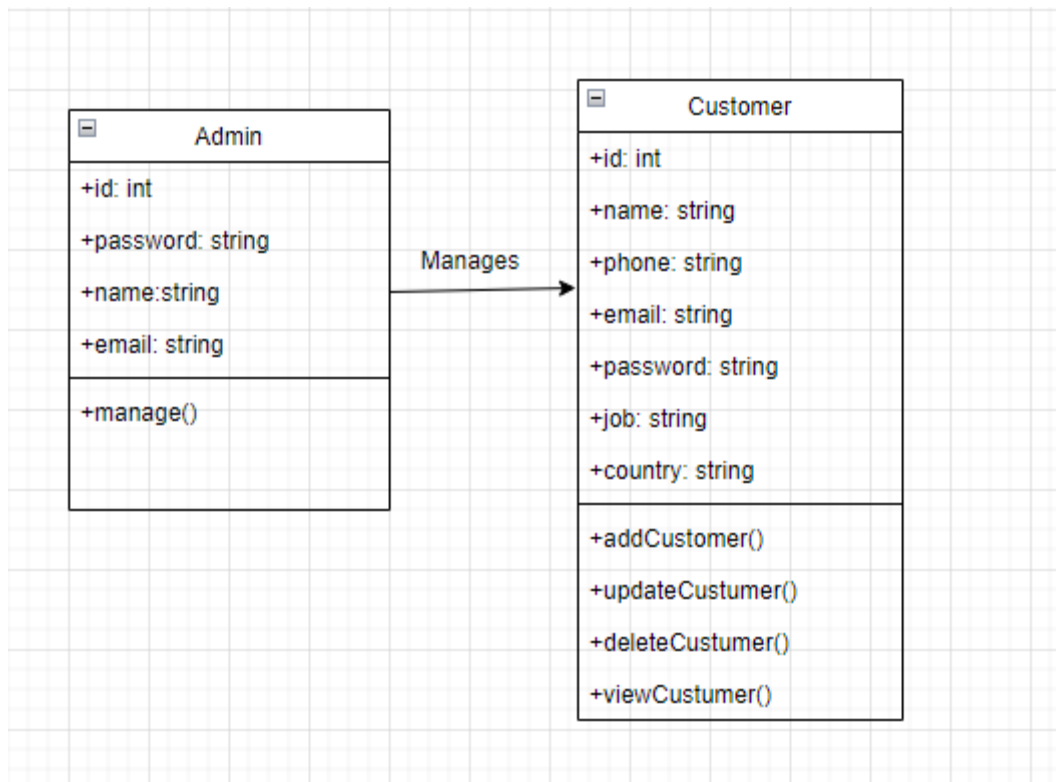


Figure 49. Manage user class diagram (Admin page)

## 4. Database diagram

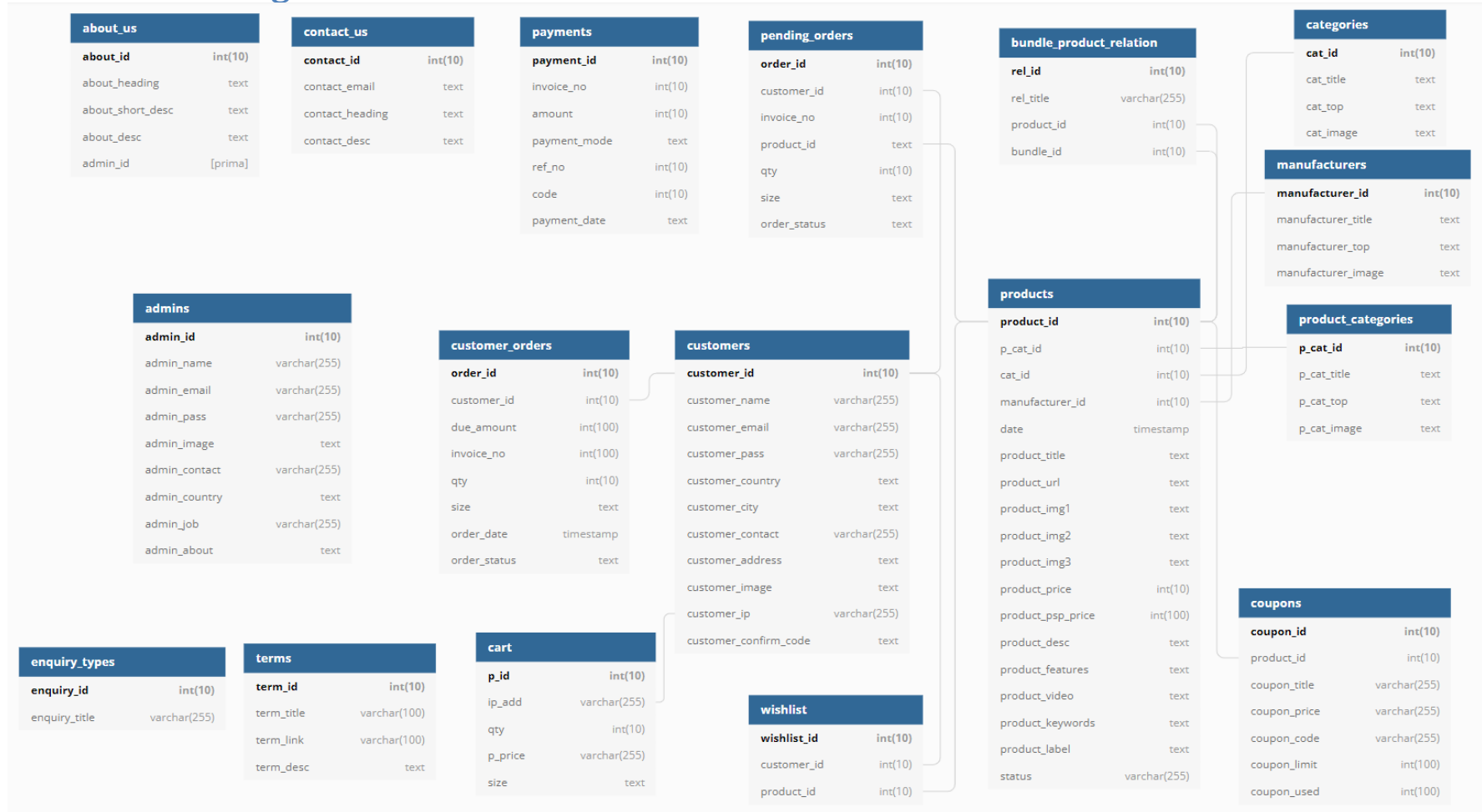


Figure 50. Database diagram

### *a) Roles of tables*

- **about-us:** contains information that will be displayed on the about us page.
- **admins:** contains all the information about the admin account.
- **bundle\_product\_relation:** contains information that assigns products to the bundle relationship.
- **cart:** contains information about the products in the shopping cart.
- **categories:** contains all the information about the categories of products.
- **Contact-us:** contains information that will be displayed on the contact us page.
- **coupons:** contains all the information about the coupon of the product.
- **customers:** contains all the information about the customer account.
- **customer\_orders:** contains all the information about the customer's order.
- **enquiry\_types:** contains all the information about the type of inquiry contact us.
- **manufacturers:** contains all the information about the manufacturers of products.
- **payments:** contains all information about completed payment orders.
- **pending\_orders:** contains all information about pending orders.
- **products:** contains all the information about the product.
- **product\_categories:** contains all information about additional product categories.
- **terms:** contains all information about terms of purchase.
- **wishlist:** contains all information about future wishlist products.

## *b) Relationships*

- **products – categories:** One-to-many: A category has many products but one product must belong to just one category.
- **products – manufacturers:** One-to-many: A manufacturer has many products but one product must belong to just one manufacturer.
- **products – product\_categories:** One-to-many: an additional product category has many products but one product must belong to just one additional product category.
- **products – coupons:** One-to-many: A coupon has many products but one product must belong to just one coupon.
- **products – bundle\_product\_relation:** One-to-many: A bundle has many products but one product must belong to just one bundle.
- **products – wishlist:** Many-to-many: A wish list can have many products and a product can belong to many wish lists.
- **products – pending\_orders:** Many-to-many: A pending order can have many products and a product can belong to any pending orders.
- **customers – wishlist:** One-to-many: A customer has many wish lists but one wish list must belong to just one customer.
- **customers – cart:** One-to-one: A customer has only one shopping cart.
- **customers – customer\_orders:** One-to-many: A customer has many orders but one order must belong to just one customer.
- **customers – pending\_orders:** One-to-many: A customer has many pending orders but one pending order must belong to just one customer.
- **customers – payments:** One-to-many: A customer has many pending orders but one pending order must belong to just one customer.

### III/ System architecture

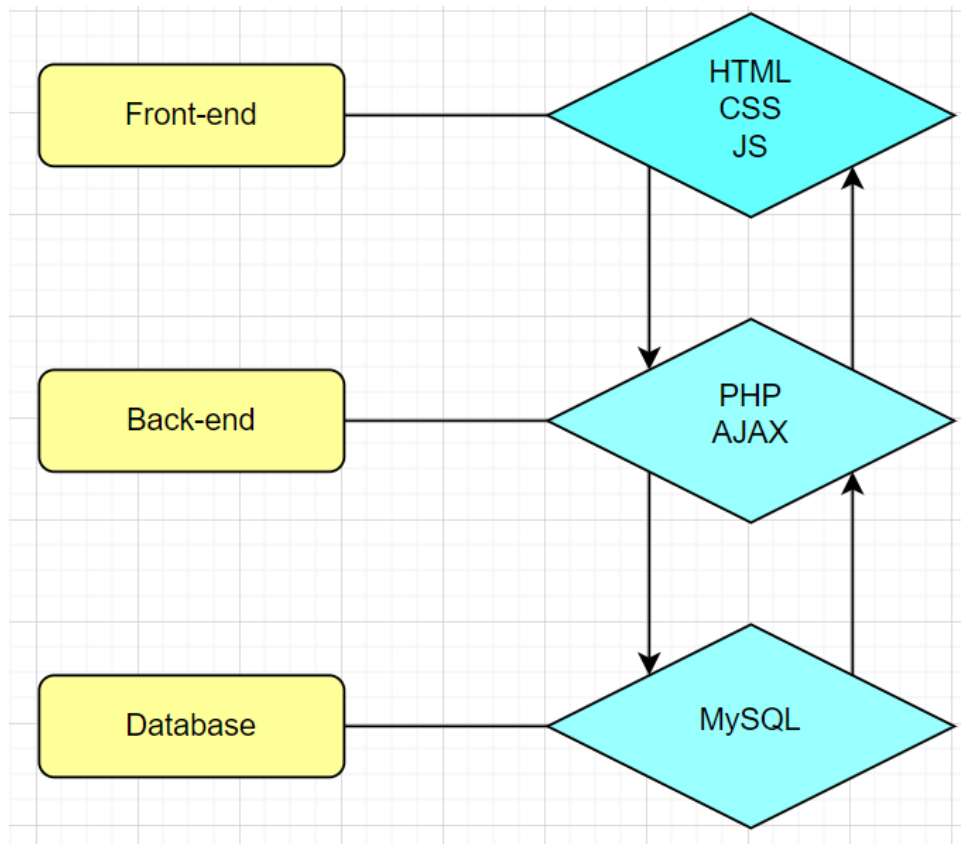


Figure 51. System architecture diagram

A website is full-side if both the Front-end, Back-end, and Database are developed completely.

- **Front-end:** Interacts with users, retrieves requests and sends them to the server-side, and receives the response from the Server.

- + The Front-end of the system includes 2 main pages: a landing page for customers and an admin page for administrators. Both pages are built with HTML, CSS, and JS.

- + Some main pages of the Landing page are Home page, Product page, Product View page, About Us page, Term page, Cart page, and My Account page.

- + Some main pages of the Admin page are Dashboard page, Product page, Bundle page, Manufacturer page, Category page, Customer page, Order page, Payment page, User page.

- **Back-end:** Receives the request from the client, performs functions, and takes data from the Database to send the response to the Client.

+ In this part we design using AJAX and PHP: PHP we use to connect the database and API. AJAX is used to control the navigation of the website.

- **Database:** Provide necessary data for the project. We use MySQL to store data for the system.

## C/ TOOLS & TECHNICAL CHOICES

### I/ PHP

We applied PHP for the backend part. PHP is a server scripting language for creating dynamic and interactive web pages. PHP is convenient because it is open-source and platform-independent. Furthermore, PHP supports several types of databases and for this system, we chose a MySQL database to work with.

### II/ MySQL

MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold vast amounts of information in a corporate network. In particular, a relational database is a digital store collecting data and organizing it according to the relational model. In this model, tables consist of rows and columns, and relationships between data elements all follow a strict logical structure. An RDBMS is simply the set of software tools used to implement, manage, and query such a database.

### III/ API

Application Programming Interface (API) is a software interface that allows two applications to interact with each other without any user intervention. API is a collection of software functions and procedures. In simple terms, API means a software code that can be accessed or executed. API is defined as a code that helps two different software's to communicate and exchange data with each other.

### IV/ HTML

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.

Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.

As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like

headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

## V/ CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations.

## VI/ JS

JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client-side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

## VII/ AJAX

Ajax is a set of web development techniques that uses various web technologies on the client-side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously without interfering with the display and behavior of the existing page.



# D/ RESULT AND DISCUSSION

## I/ Results

In this project, we have succeeded in implementing these main functions:

- **Login function:** Customer and admin can log in to the system. They can enter their email and password in a login format and then access the system by clicking on the “Login” button. In case the email or password does not match, the system returns an error message.

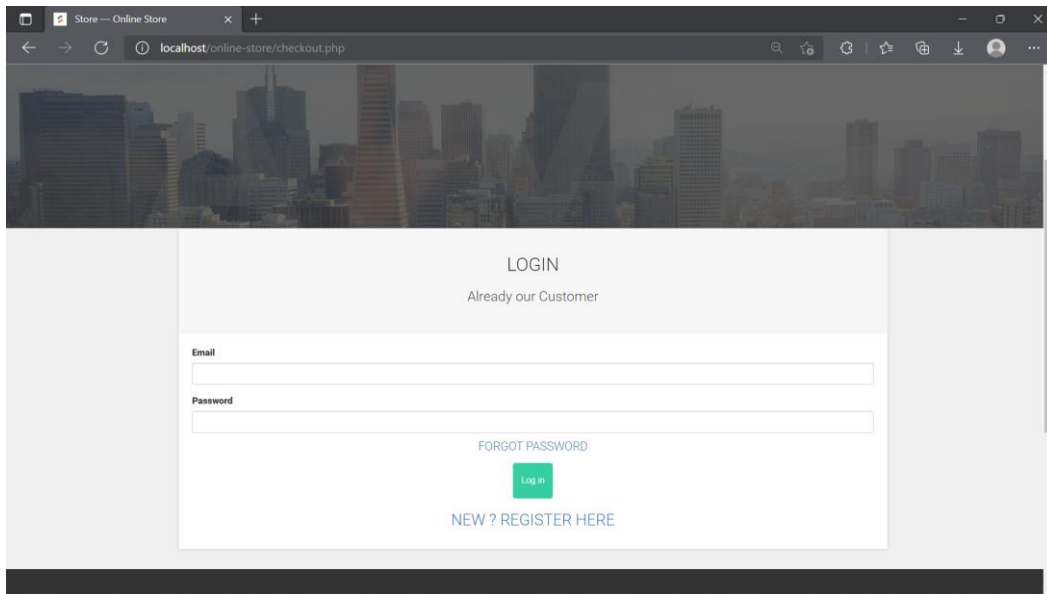


Figure 52. Login page (landing page)

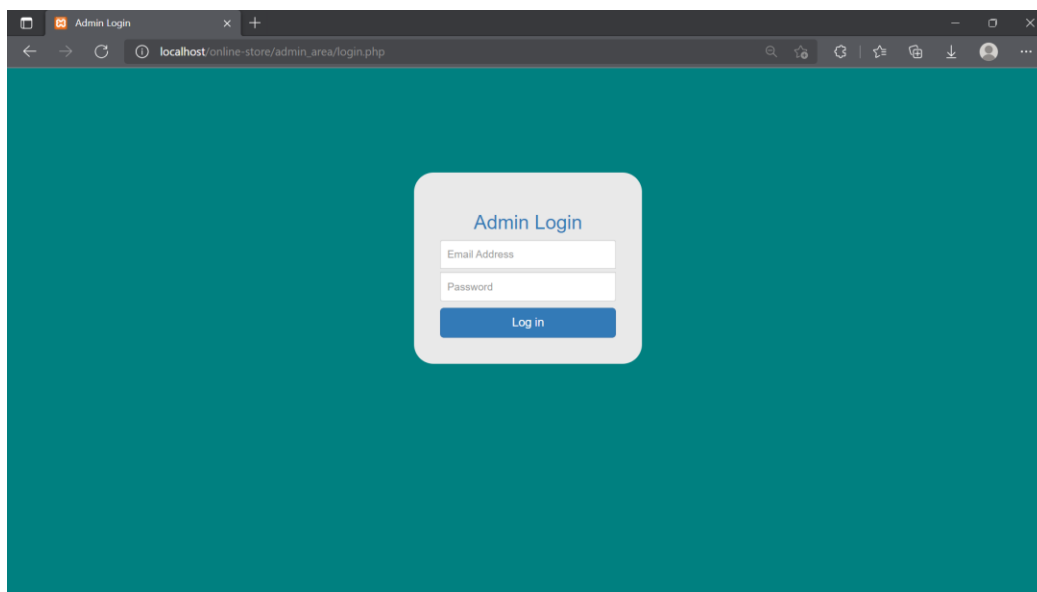


Figure 53. Login page (admin page)

- **View product and bundle:** Customers can easily view all products, a product or a bundle.

They are visible on the Shop page of the website

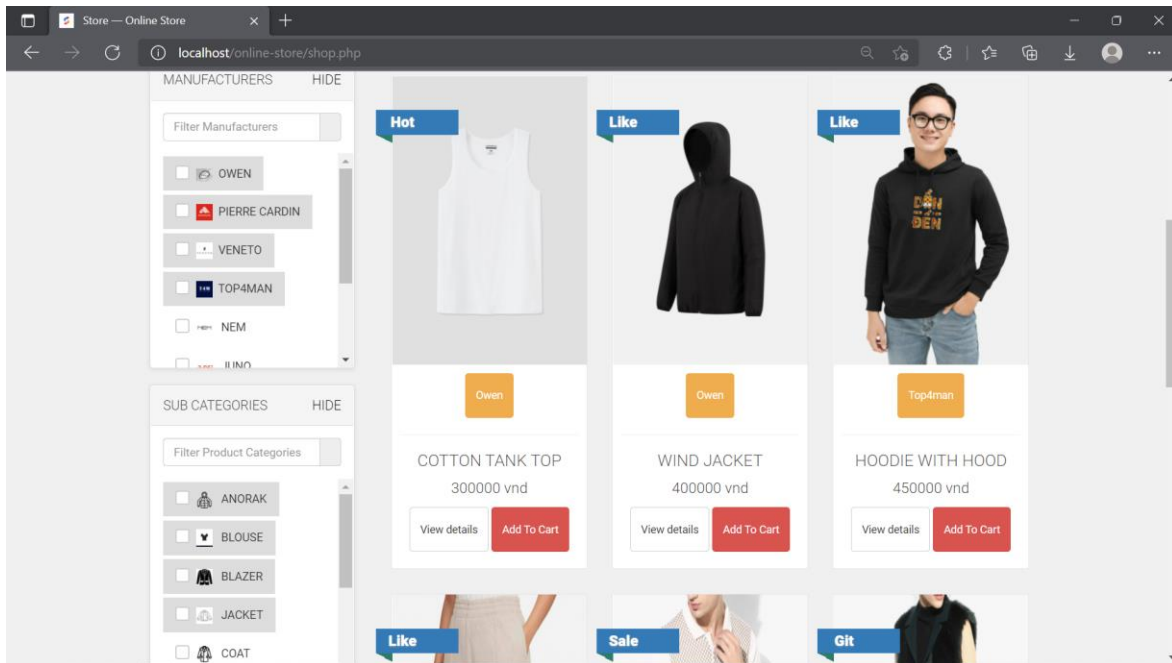


Figure 54. Shop page (landing page)

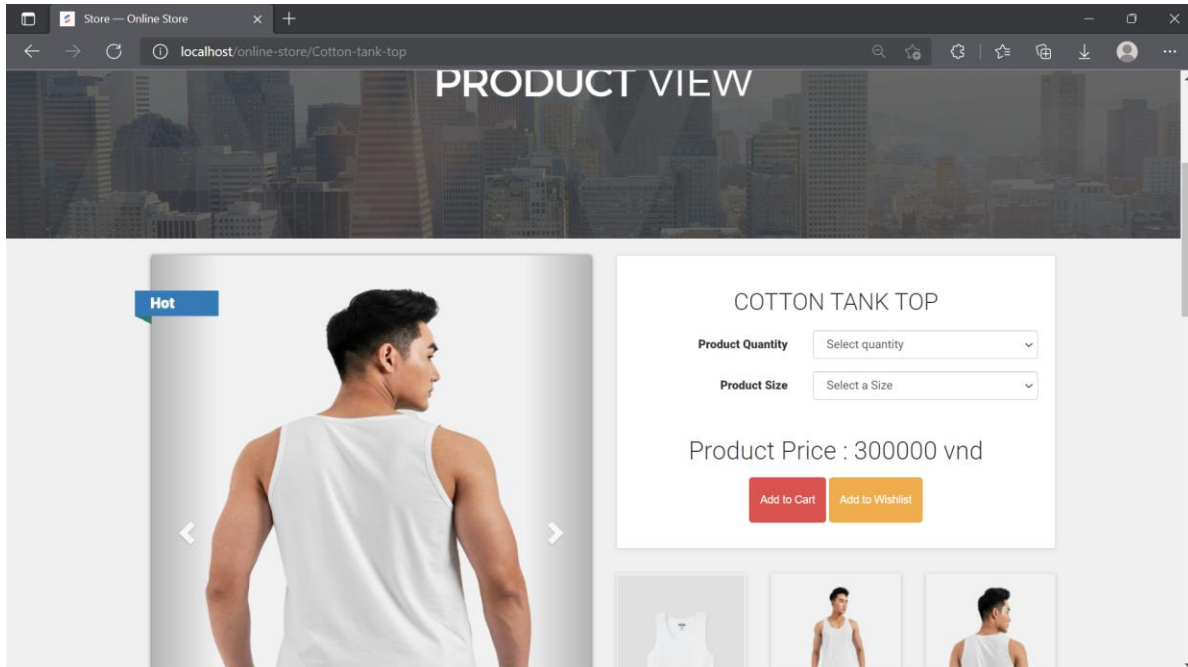


Figure 55. Product view page (landing page)

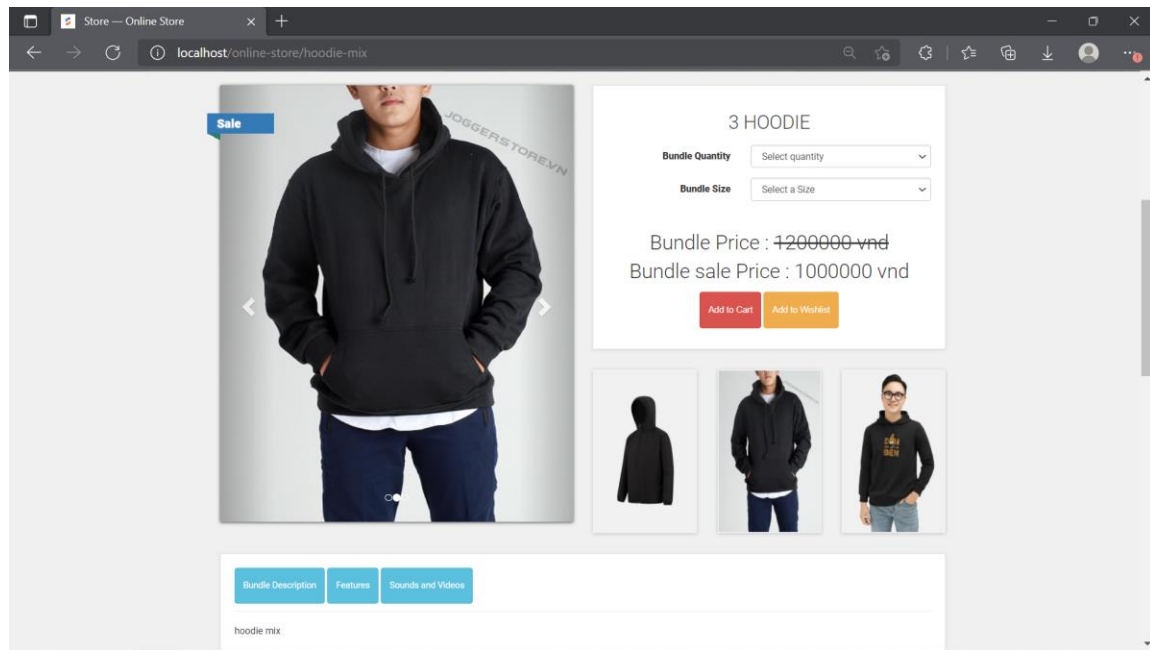


Figure 56. Bundle view page (landing page)

- **Buy product:** Customer accounts can also buy any product or bundle by simple operations.

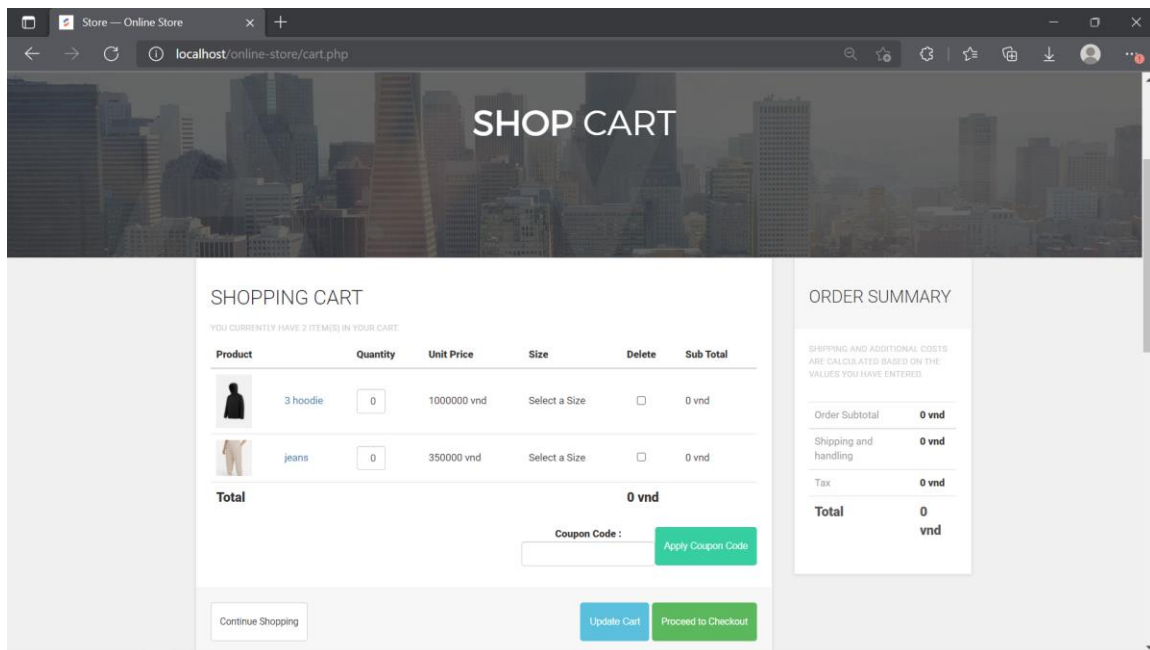


Figure 57. Cart page (landing page)

- **Payment:** Customer accounts can also pay by cash or bank card.

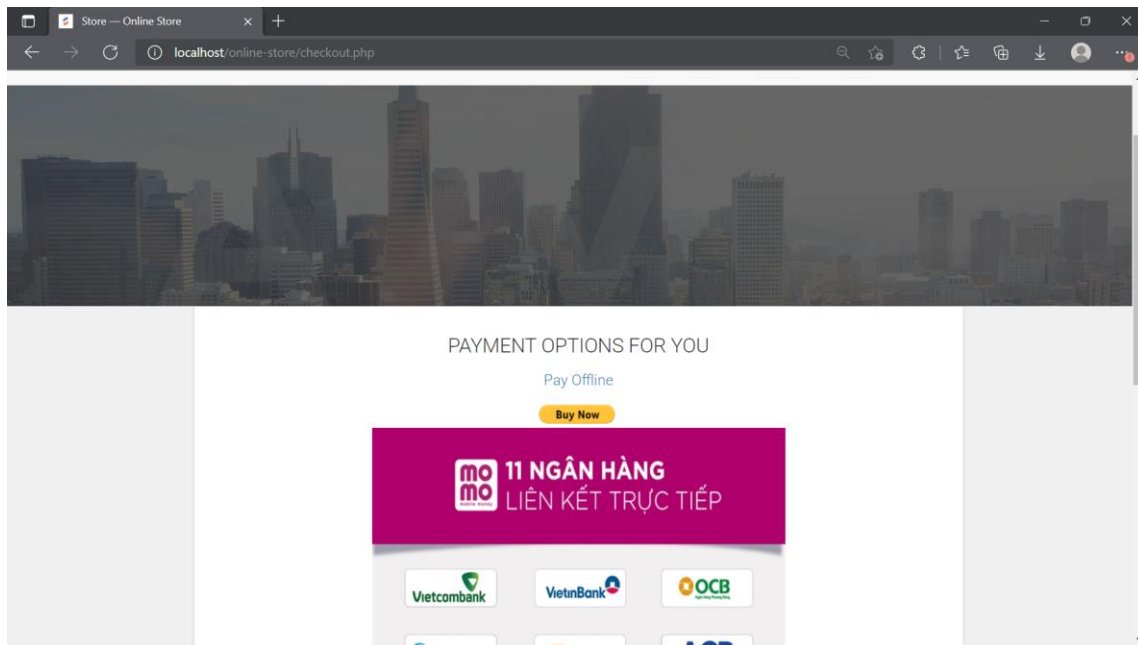


Figure 58. Payment page (landing page)

- **Management function:**

+ **Manage account:** Customers can manage their account, from order placed, wish list, edit account information, delete account. Administrator accounts can view and delete with customer accounts, edit their own account or add another admin account to the system.

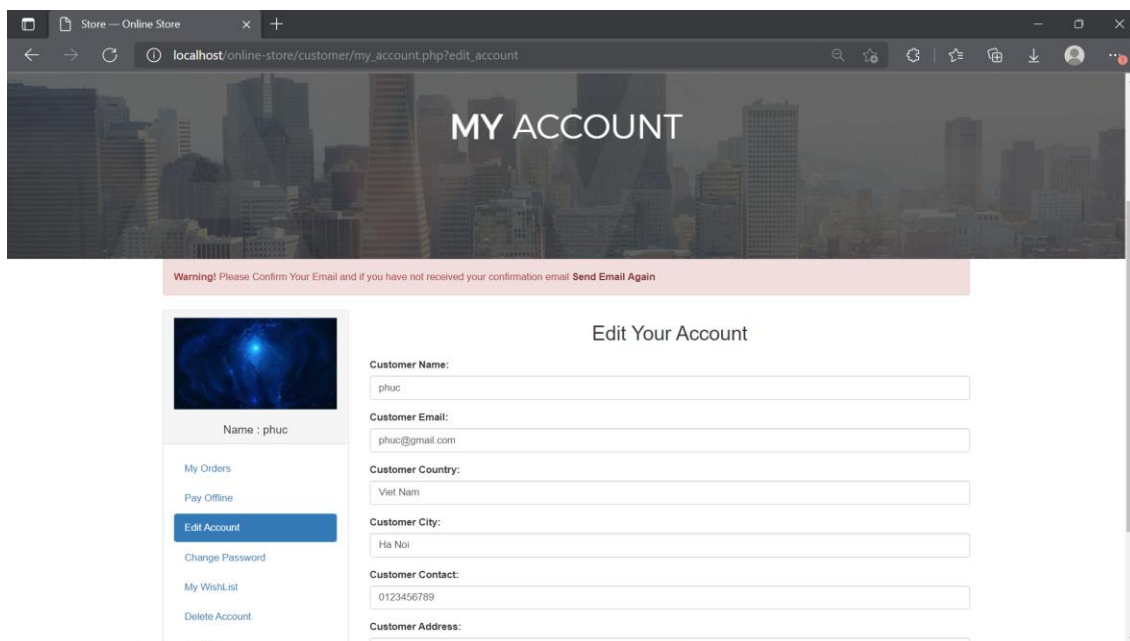


Figure 59. My account page (landing page)

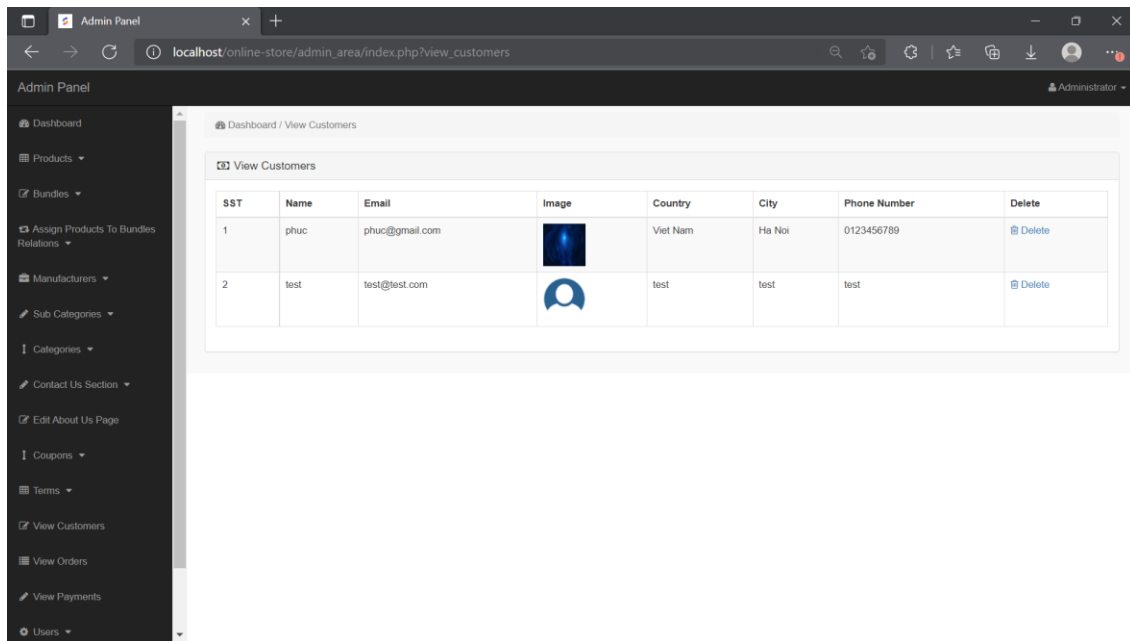


Figure 60. View customer page (admin page)

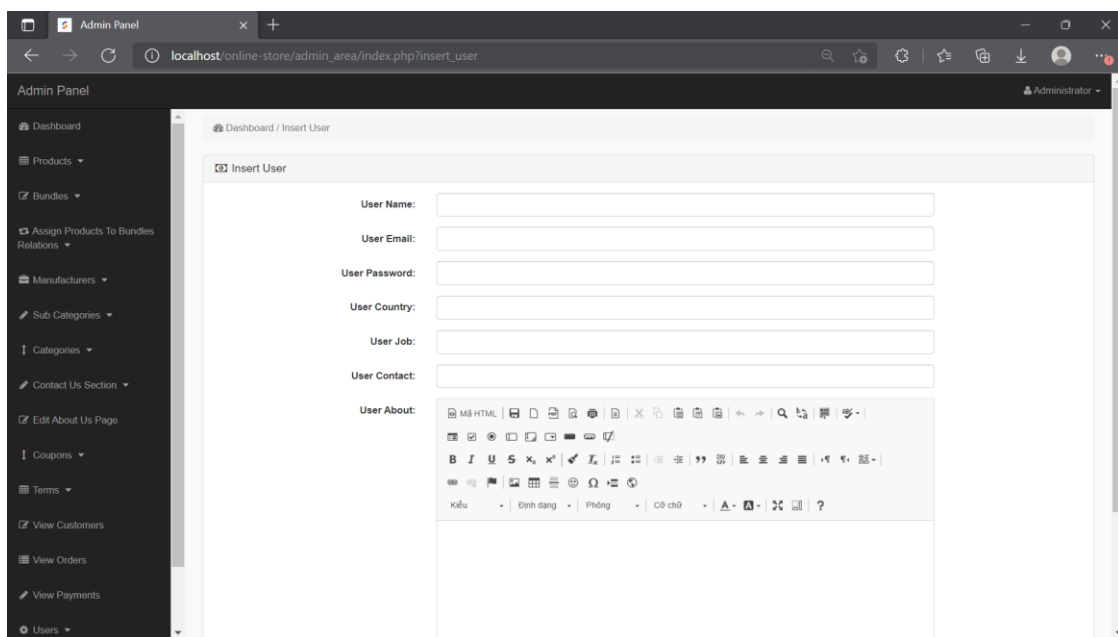


Figure 61. Insert user page (admin page)

+ **Manage products, bundles, categories, manufacturers:** administrator accounts can manage information of products, bundles, categories, manufacturers.

Admin Panel

Dashboard / View Products

View Products

SST	Title	Image	Price	Sold	Keywords	Date	Delete	Edit
1	hoodie		300000 vnd	0	hoodie	2022-02-20 04:03:28	Delete	Edit
2	ao thun		200000 vnd	1	ao thun	2022-02-20 04:12:57	Delete	Edit
3	Khaki : Shirt Jacket		270000 vnd	0	shirt	2022-02-20 04:17:43	Delete	Edit
4	Jeans Basic Slim		250000 vnd	0	jean	2022-02-20 11:20:00	Delete	Edit
5	Black Blouse Versace Coat1		400000 vnd	0	coat	2022-02-20 11:39:52	Delete	Edit
6	ao len		500000 vnd	0	len	2022-02-20 11:41:27	Delete	Edit
7	jeans		350000 vnd	0	jean	2022-02-20 11:42:49	Delete	Edit
8	Hoodie with hood		450000 vnd	1	hoodie	2022-02-20 12:02:40	Delete	Edit

Figure 62. Product page (admin page)

Admin Panel

Dashboard / View Bundles

View Bundles

SST	Title	Image	Price	Sold	Keywords	Date	Delete	Edit
1	3 hoodie		1200000 vnd	0	jacket	2022-02-20 15:14:48	Delete	Edit

Figure 63. Bundle page (admin page)

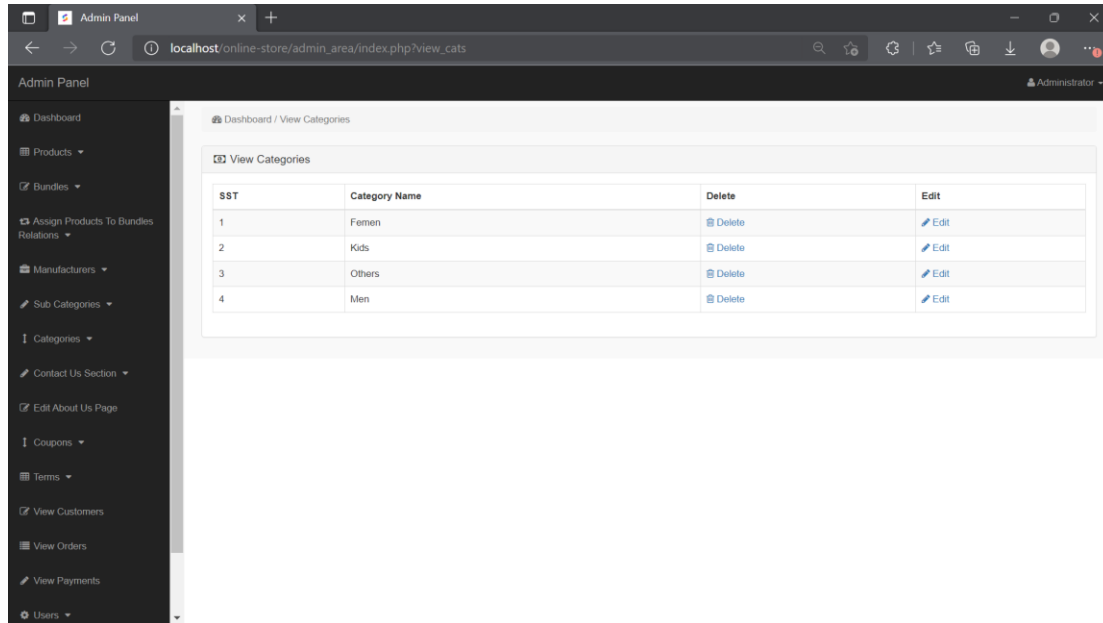


Figure 64. Categories page (admin page)

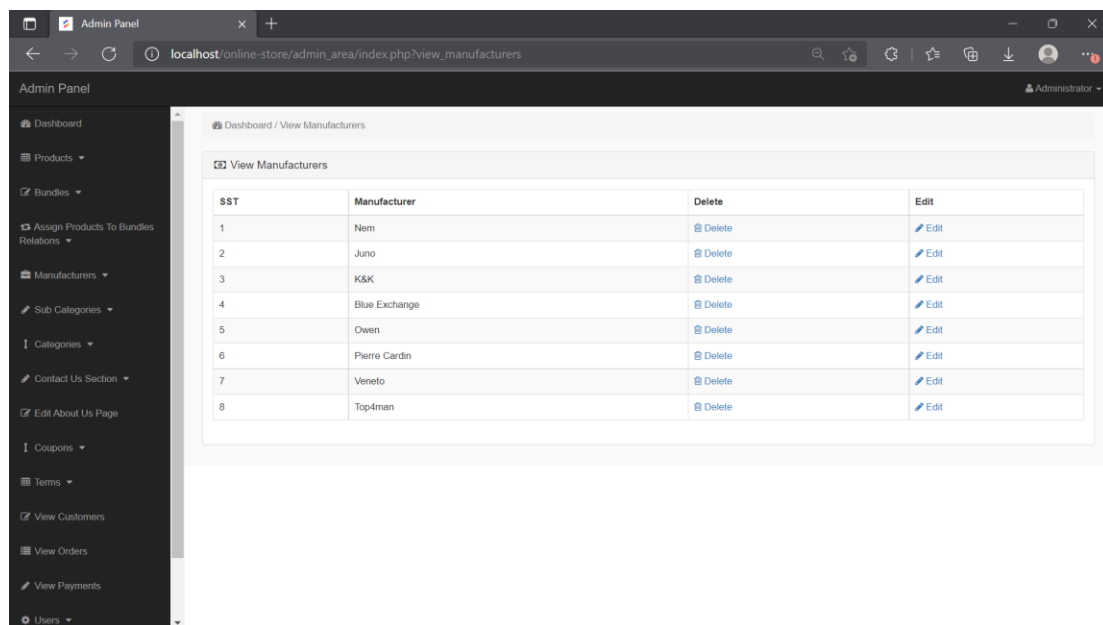


Figure 65. Manufacturer page (admin page)

## II/ Discussion

**Human:** Due to our poor experience with web services and UI/UX, our website could not be optimized and smooth in terms of operation. Moreover, there were also a lot of difficulties when we had to learn new techniques and finish the system at the same time.

**Pandemic:** The COVID-19 pandemic was also a problem for us. While living in different cities, it was hard for us to communicate, convey ideas, and support each other when there were obstacles.

**Time:** The Group Project happened during a time when there was an overwhelming amount of other projects and examinations from the university. Therefore, even with the appropriate time limit, we found it quite challenging to fully concentrate on this project and enhance the system more.

## E/ CONCLUSION & FUTURE WORK

### I/ Conclusion

To sum up, we have successfully built almost all essential functions of the Online Store Management System. Thanks to a friendly interface, users can easily interact with the system.

According to the initial use case diagram that we designed, some main functions of the system are finished:

- *Customers and administrators can log in to the system.*
- *Customers can view products, add products to wishlist or cart, payments, manage the account.*
- *Administrators can create, delete, inspect, and update products, bundles, categories, manufacturers, coupons.*

### II/ Future work

To enhance this web application, we are planning to do the following tasks in the future:

- *More products can be added.*
- *Add product reviews and ratings.*
- *Better interface and easy to use.*
- *Comments can be edited.*
- *Ameliorate the system security.*