HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

Restaurant Order and Management Website

LE THANH CHINH
chinh.lt198209@sis.hust.edu.vn

Major: Information technology

Supervisor: MSc Le Ba Vui _____

Department: Computer Engineering

School: Information and Communications Technology

HANOI, 06/2024

PLEDGE

Student's Full Name: . . . . . .
Student ID: . . . . . .
Contact Phone Number: . . . . . .
Email: . . . . . .
Class: . . . . . .
Program: . . . . . .

I, Le Thanh Chinh, commit that the Graduation Thesis (GT) is my own research work under the guidance of MSc Le Ba Vui. The results presented in the GT are truthful and are my own achievements, not copied from any other works. All references in the GT, including images, tables, data, and quotations, are clearly and fully cited in the bibliography. I take full responsibility for any violations of the school's regulations concerning plagiarism.

Hanoi,

Student

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to MSc Lê Bá Vui. His dedicated guidance and support throughout the process of completing my thesis have been invaluable. His important directions and shared knowledge greatly contributed to the successful completion of my work.

I also extend my sincere thanks to Hanoi University of Science and Technology and all the professors who have created an excellent learning environment and built a solid knowledge foundation for me. The beneficial lessons and practical sessions at enterprises provided by the professors have not only deepened my academic understanding but also equipped me with essential life skills.

Furthermore, I wish to express my heartfelt appreciation to my family and relatives, who have always encouraged and provided the best conditions for me throughout my academic journey. I am also grateful to my friends and collaborators who have accompanied me, sharing and overcoming challenges together during our time at the university.

Given the limited time and my own expertise, I am aware that my thesis may still contain shortcomings. I eagerly look forward to receiving feedback and evaluations from my professors to further improve and perfect my work.

Thank you very much!

ABSTRACT

The management of restaurants is currently a crucial and complex issue, particularly when conducted manually. With the increasing number of restaurants and the expansion of new branches, the demand for an efficient management system has become increasingly important. To address this issue, I have chosen to develop a restaurant management website with functions such as management, meal ordering, online ordering, and online table reservations. Through market research, it is evident that existing solutions have addressed basic restaurant needs but still exhibit certain limitations.

Based on market survey results, I found that developing a restaurant management website is an essential need. By surveying the needs of restaurants, I gained a clearer understanding of the challenges they face, which enabled me to devise ideas to address these issues effectively. My solution involves creating a restaurant management website with key functions such as restaurant management, meal ordering, online ordering, online table reservations, restaurant introduction, and a blog. These functions not only help optimize operational processes but also enhance customer experience.

The primary contribution of this thesis is to provide a comprehensive solution to the current challenges in restaurant management. The result of this project is the successful development of a restaurant management website, which not only meets the present needs but also has the potential for further expansion and improvement. This website is designed to enhance operational efficiency and service quality. It promises to become an indispensable tool, aiding restaurants in management, optimizing processes, and improving customer experience, thereby contributing to the sustainable growth of the food service industry.

Student

(Signature and full name)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abbreviation	English name	Vietnamese name
API	Application Programming Interface	Giao diện lập trình ứng dụng
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
CNTT	Information technology	Công nghệ thông tin
GT	Graduation thesis	Đồ án tốt nghiệp
SV	Sinh viê	Sinh viên
UML	Unified Modeling Language	Ngôn ngữ mô hình hóa
SQL	Structured Query Language	Ngôn ngữ truy vấn cấu trúc dữ liệu
CSS	Cascading Style Sheets	Ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (html)
CLI	Command Line Interface	Giao diện dòng lệnh
HTTPS	Hypertext Transfer Protocol Secure	Một phần mở rộng của Hypertext Transfer Protocol (HTTP).
XML	eXtensible Markup Language	Ngôn ngữ đánh dấu mở rộng
Js	JavaScript	Ngôn ngữ được tích hợp và nhúng vào (html) giúp website trở nên sống động hơn
Ts	TypeScript	Ngôn ngữ lập trình mã nguồn mở

CHAPTER 1. INTRODUCTION

1.1 Motivation

The development of the Restaurant Ordering and Management Website stemmed from the urgent need in the restaurant industry for streamlined and efficient operations. Nowadays, restaurants face many challenges in managing orders, optimizing customer service and improving operational efficiency. Without an effective management system, it is easy for errors to occur during order processing, prolonged service time, and reduced service quality, directly affecting customer satisfaction and customer satisfaction. restaurant revenue.

Implementing an online ordering and management system is of significant importance for restaurants. Such a system not only optimizes the order management process and reduces errors, but also improves the customer experience by providing quick and convenient service. Additionally, it allows restaurants to track and analyze business data, leading to more informed strategic decisions.

The importance of this project extends beyond the individual restaurants; Its benefits can be realized in many different fields. From coffee shops to food retailers, any business involved in food service can take advantage of an effective order management system to streamline operations and improve service quality. and improve customer satisfaction.

Addressing the need for an effective order and management system promises many tangible benefits, including enhanced operational efficiency, improved customer satisfaction, and reduced errors and decision-making. data-based determination. In summary, the development of a Restaurant Ordering and Management Website is essential to meet the needs of the restaurant industry, with potential applications in many other fields, contributing to the modernization and enhancement of services. overall service.

1.2 Objectives and scope of the graduation thesis

Website to bring convenience and optimize costs for restaurants. Therefore, to solve the problem mentioned in section 1.1, I built Restaurant Order and Management Website aims to: (i) provide a convenient management system for users, (ii) order, reserve tables and pay online, (iii) make statistics and checks flexibly, (iv) ensure sensitive data confidentiality. With this system, managers and staff will be extremely convenient in managing dishes and revenue, and customers will be able to order, reserve tables and pay online.

In addition, the system is also a simple employee management system, integrating a blog function to increase interaction with customers. With the above solutions, the system has overcome the problems that restaurants are facing. Based on these features, restaurants can easily manage all restaurant work and easily come up with appropriate solutions and strategies to increase revenue and bring customer satisfaction. In addition, customers can also more easily access the restaurant through online ordering, table booking, and blog functions.

1.3 Tentative solution

Because the product is written on a website platform, I chose the language (i) Java, combined with interface design using (ii) Angular. To be able to do this, I need to thoroughly learn about Java Core, OOP, Spring Framework, etc. Details of the technology used will be presented in Chapter 3.

To solve the problem of creating a Restaurant Order and Management Website, you need to learn about restaurant operations, and then apply this practice to your work. Details are presented in Chapter 2.

To be able to manage users, I have divided roles into three groups: Group 1 - administrators, group 2 - restaurant staff, and group 3 - customers.

To manage the restaurant, I have designed specific functions such as (i) member management, (ii) food management, (iii) order management, (iv) article management, (v)) ordering food, reserving a table, paying online, (vi) statistical functions. These functions ensure convenience for managers and employees to manage the restaurant conveniently while providing convenience to customers.

1.4 Thesis organization

The remainder of this graduate project report is organized as follows:

Chapter 2: REQUIREMENT SURVEY AND ANALYSIS: Chapter 2 presents a survey of the problems that restaurants are facing today, the current situation and the necessity of a restaurant management website. Analyze existing solutions, their capabilities and limitations, and analyze requirements and present the main functions of the website.

Chapter 3: TECHNOLOGY USED: Chapter 3 introduces the technologies used to build websites, highlights of that technology, and reasons for choosing that technology to build a restaurant management website to address the requirements identified above. chapter 2.

Chapter 4: DESIGN, IMPLEMENTATION, AND EVALUATION: Chapter 4 presents how to deploy and develop applications. The main contents covered in this

chapter include (i) architectural design, (ii) detailed design, (iii) application construction. In the architectural design section, the selected software architecture and overall application design are presented. The detailed design part includes interface design, class design, and database of the application. The application building section shows the results achieved by GT.

Chapter 5: SOLUTION AND CONTRIBUTION: Chapter 5 presents outstanding solutions and contributions. This chapter highlights the outstanding contributions that GT brings. These contributions include GT contribution when applied in practice, solving the limitations of current restaurant management applications.

Chapter 6: CONCLUSION AND FUTURE WORK: Chapter 6 presents the achieved results, provides conclusions and future development directions for the software.

CHAPTER 2. REQUIREMENT SURVEY AND ANALYSIS

In this chapter, I will present a survey of actual restaurant websites. From there, I give an overview of the functions and detailed specifications of some main functions of this website.

2.1 Status survey

Food and beverage (F&B) is one of the largest industries and has a close relationship with the lives of billions of people around the world. The Business Research Company's F&B research report released in January 2023 said that the F&B market is expected to recover after the "black swan" event - COVID-19 and the scale is expected to increase to 9.225 billion USD by 2027 with an average growth rate of 6.3%.[1]

The F&B industry in general and the restaurant business in particular are growing rapidly. More and more restaurants are opening, leading to an increasing need for restaurant management. Besides, the trend of kitchen automation is being used by many restaurants to optimize all stages from ordering to serving. Serve food to customers. In addition, ordering food online brings many conveniences to customers.

Currently, there are quite a few Restaurant Order and Management Websites available on the market such as godmother, . These applications all bring convenience and ease of use to users. Due to the scope of GT, I would like to partially re-implement the utilities of the above software and develop more special features for users.

[1] https://vietstock.vn/2023/08/xu-huong-phat-trien-ben-vung-trong-nganh-thuc-pham-va-do-uong-735-1117379.htm

2.2 Functional Overview

2.2.1 General use case diagram

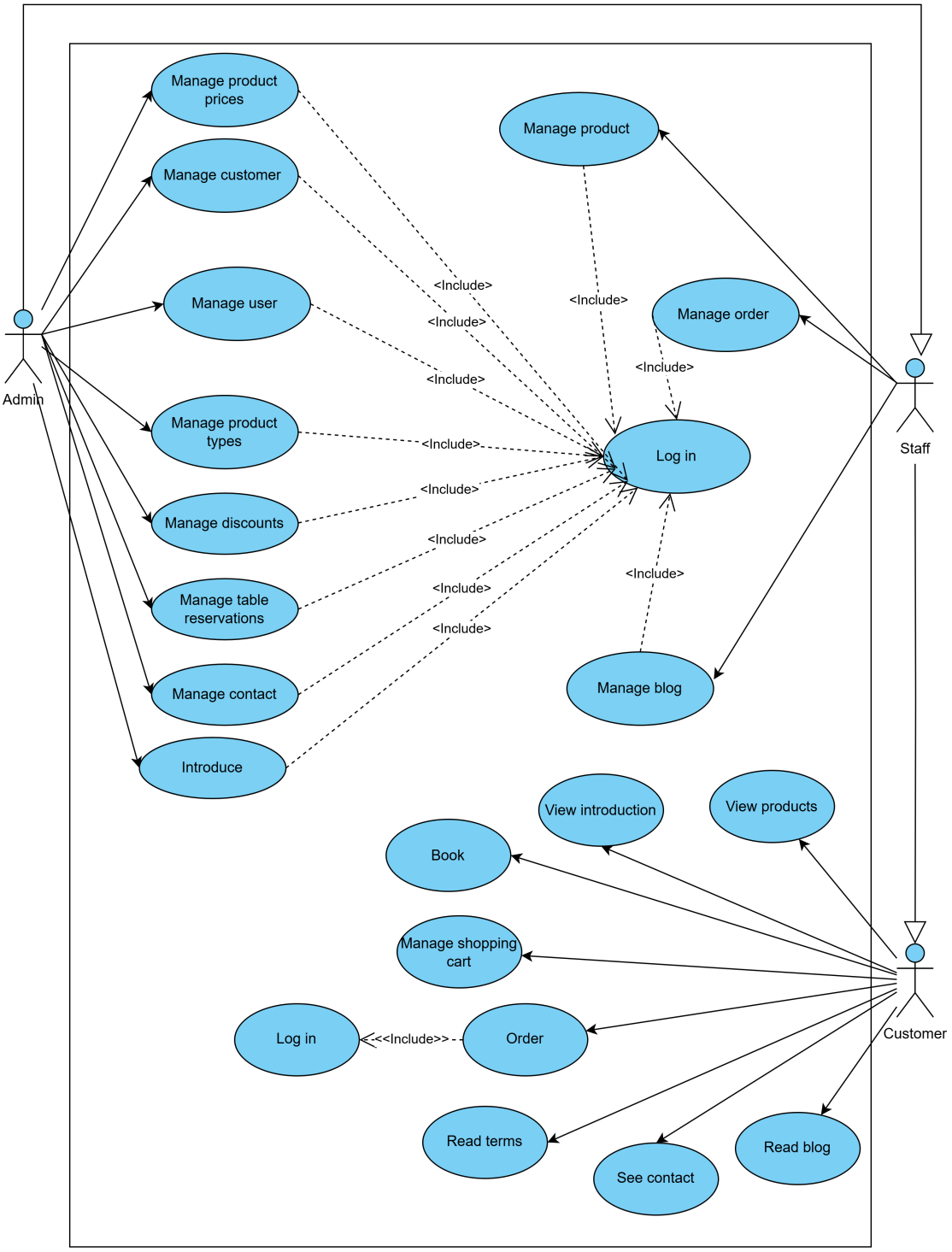

Figure 2.1: Use Case diagram overviews the management system

Restaurant Order and Management Websites software includes different components and functions. **Figure 2.1** shows the actors and main functions of the system.

Main actors: admin, staff,customer.

- admin is the coordinator and holds the role of restaurant manager.
- Employees are users of certain functions such as order management, product management, and article management.
- Employees are users of certain functions such as order management, product management, and article management.

2.2.2 Detailed use case diagram

2.2.2.1 Product management use case diagram

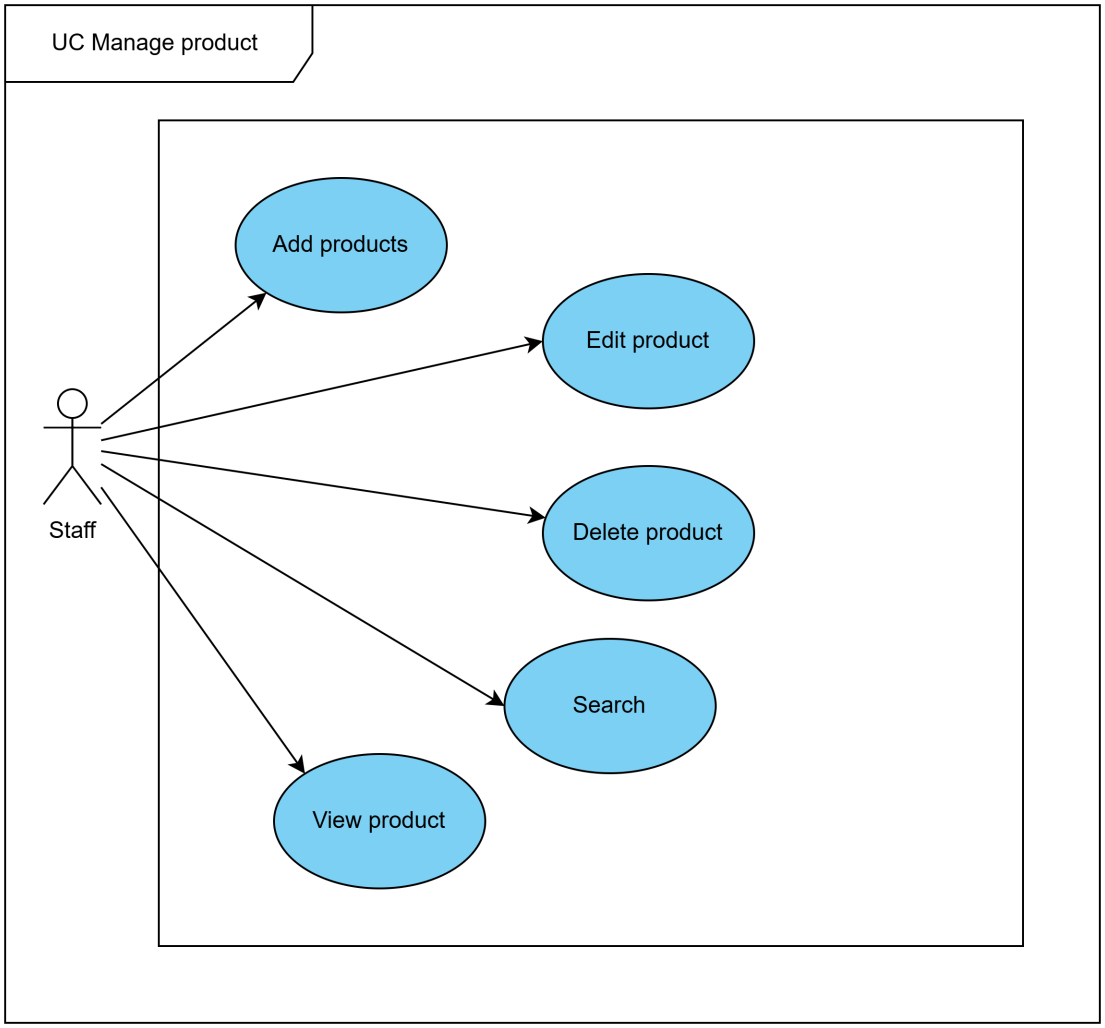

Figure 2.2: Product management use case diagram

Figure 2.2 describes a system that allows product management with the main actor being the employee. Staff will update all product information. When new products become available, employees must first add the new products. If the product has information errors, staff will update the product information. For products that are no longer for sale, staff can delete the product. Additionally, employees can search for products and view product details.

2.2.2.2 User management use case diagram

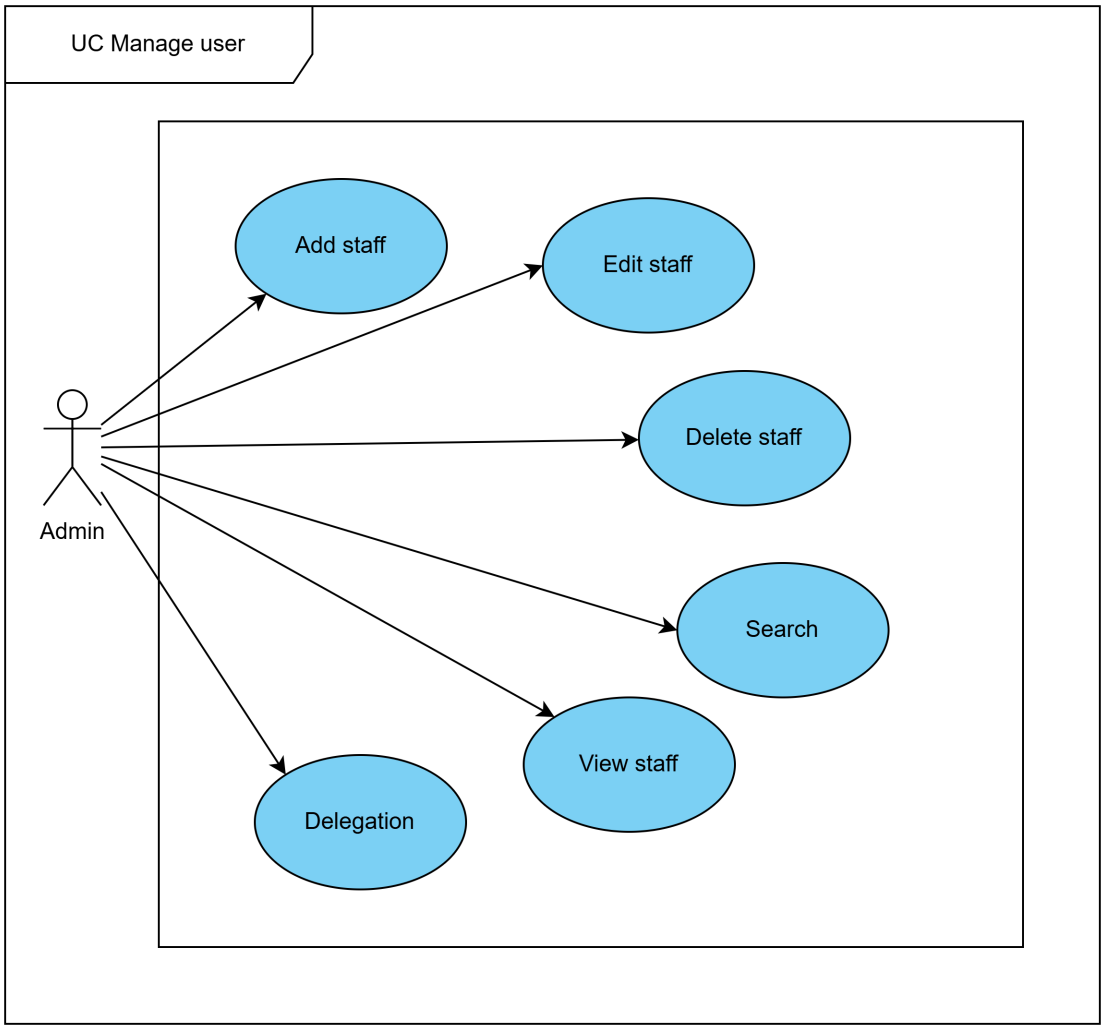

Figure 2.3: User management use case diagram

Figure 2.3 describes the User Management function with the main agent being admin. Similar to the Product Management function, the admin will have full user management functions such as adding, editing, deleting, and updating users. Each admin will have an account and password to log into the system.

2.2.2.3 Order management use case diagram

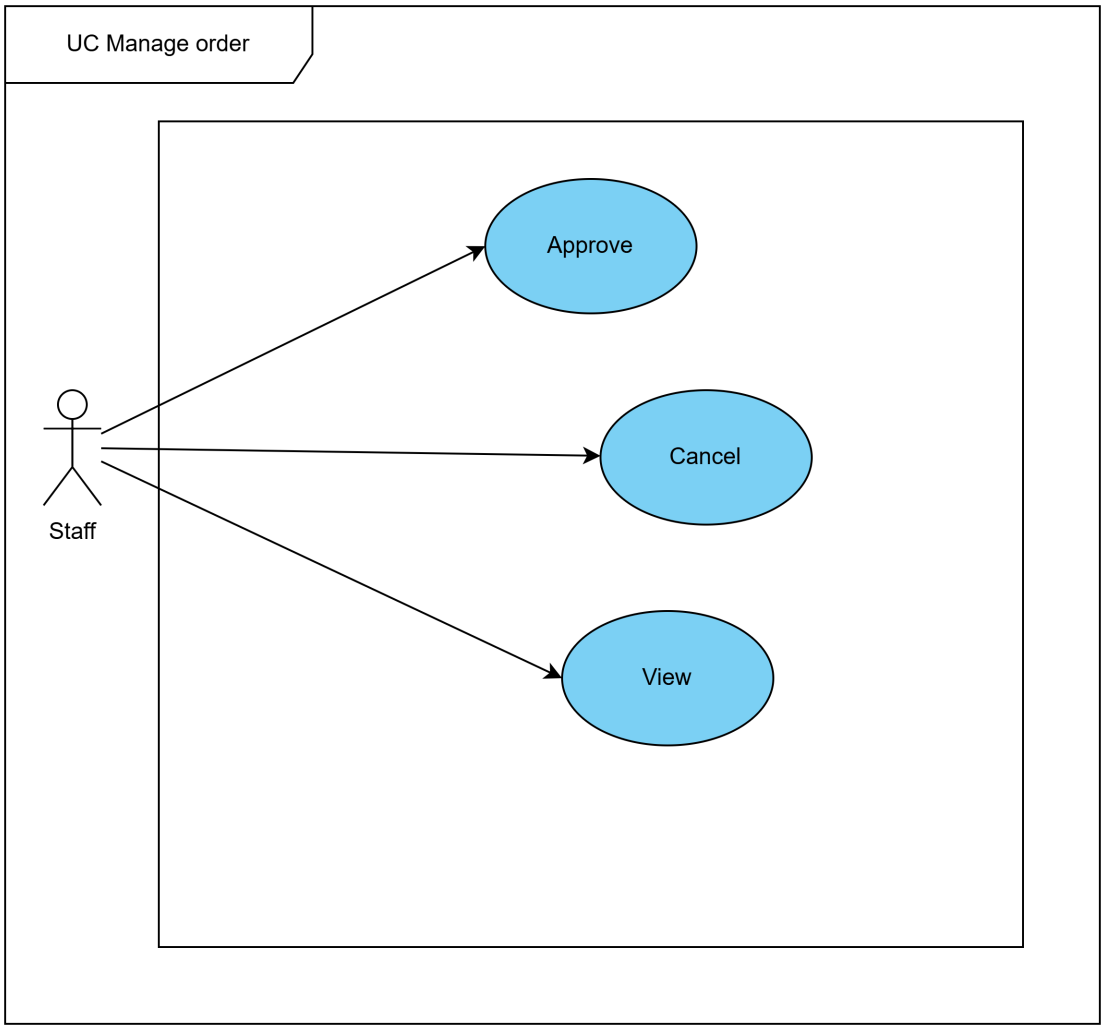

Figure 2.4: Order management use case diagram

Figure 2.4 describe the Order Management function with the main actor being the Employee. The customer will create an order and the employee will receive that order. Staff has the function of approving orders, canceling orders, and viewing orders

2.2.2.4 Product view use case diagram

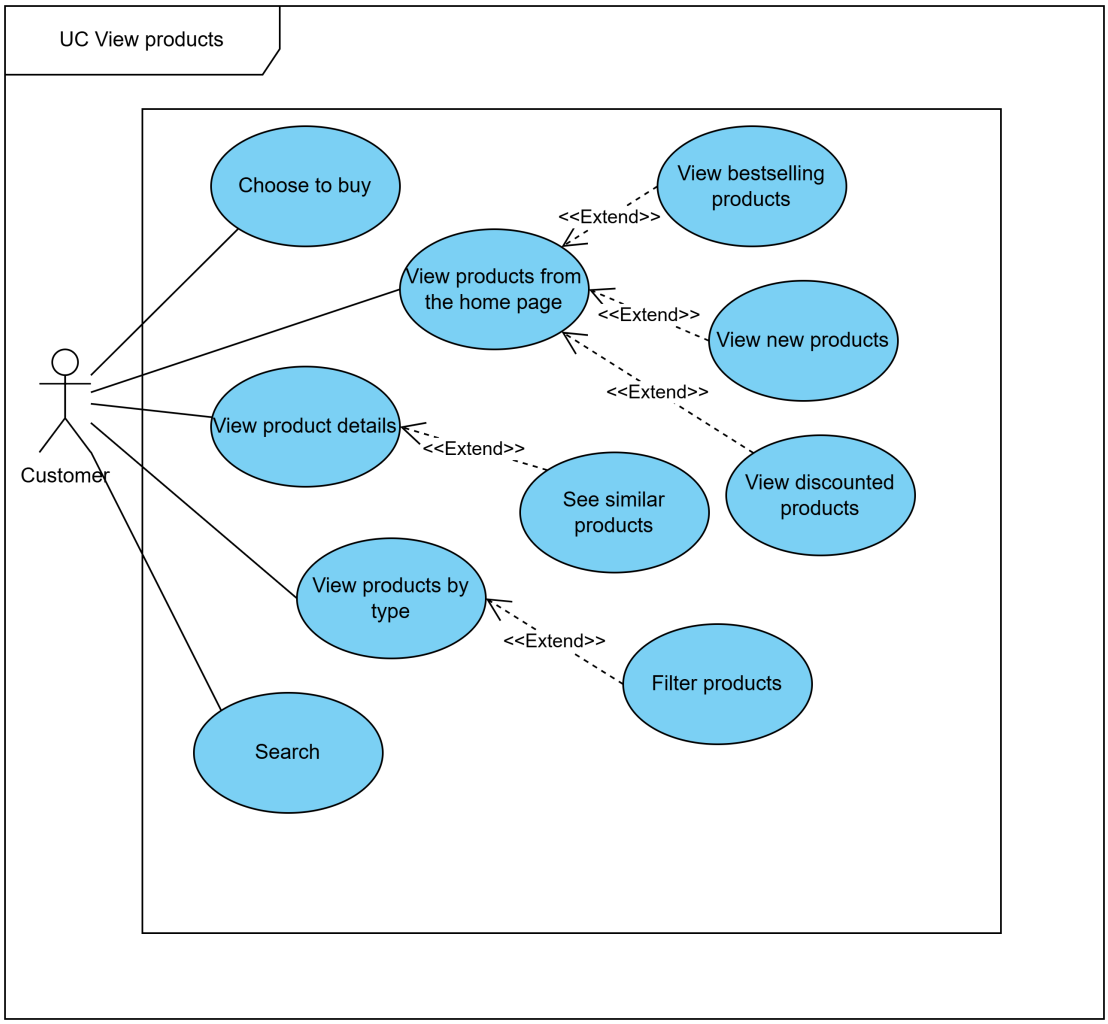

Figure 2.5: Product view use case diagram

Figure 2.5 describe the product viewing function with the main actor being the customer. First, when accessing the website, customers will be taken to the home page, where customers can view products on the home page including categories such as best-selling products, new products, discounted products, view products. by type. In addition, customers can search for products, view product details, view similar products, and choose to buy.

2.2.2.5 Shopping cart management use case diagram

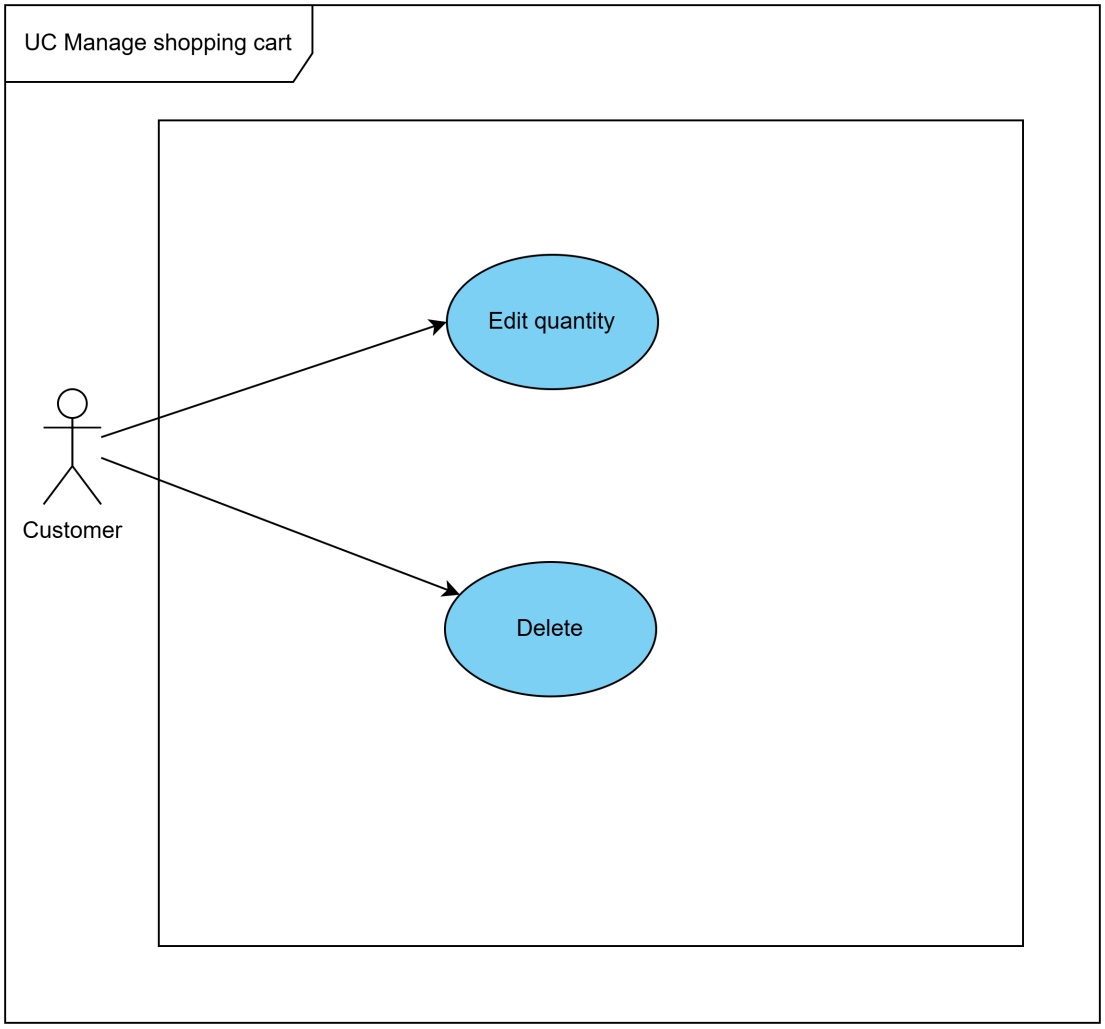

Figure 2.6: Shopping cart management use case diagram

Figure 2.6 describe the shopping cart management function with the main actor being the customer. Customers can manage their shopping cart by editing quantities and deleting products.

2.2.3 Business process

Restaurant Order and Management Websites are responsible for managing dishes, managing staff, managing news, managing orders, ordering food, booking tables online, and paying online.

Initially, the software development team will create an admin. This person is responsible for managing the restaurant and coordinating users. Admin will then proceed to grant permissions to employees. Employees have the role of food management, order management, table reservation management, and article management.

Each admin and employee must have an account including Username and Password to log into the system. For each role, the system will automatically navigate to the functions corresponding to that person's role.

After admin and staff add dishes to the menu. Customers can search for dishes from the menu to order, and staff will perform the ordering function for customers eating at the restaurant, in addition, customers can order online and pay online or pay when To receive goods, customers will have to log in if they want to use this function. After successfully ordering, the system will display the order, and the staff will be responsible for updating the order status. For customers ordering online, when the order is canceled or successfully delivered, the system will send an email to the customer.

In addition, customers can book a table online. In case the table is sold out, the staff will cancel the table and the system will send an email to the customer. Customers can view the restaurant's articles and policies on the restaurant's homepage.

2.3 Functional description

Table 2.1: List of Use Cases

System	Use Case Group	Use Case Code	Use Case Name
Restaurant Order and Management Websites	Overviews	UC001	Log in
		UC002	Manage order
		UC003	Manage product
		UC004	Manage blog
		UC005	Manage product prices
		UC006	Manage customer
		UC007	Manage product
		UC008	Manage user
		UC009	Manage product types
		UC010	Manage discounts
		UC011	Manage table reservations
		UC012	Manage contact

Continued on next page

Table 2.1: List of Use Cases (Continued)

System	Use Case Group	Use Case Code	Use Case Name
		UC013	Introduce
		UC014	Log in
		UC015	**Order**
		UC016	View products
		UC017	View introduction
		UC018	Book
		UC019	Manage shopping cart
		UC020	Read terms
		UC021	Read blog
		UC022	See contact
	Manage product	UC023	**Add products**
		UC024	Edit product
		UC025	Delete product
		UC026	**Search**
		UC027	View product
	Manage user	UC028	Add staff
		UC029	Edit staff
		UC030	Delete staff
		UC031	Search
		UC032	View staff
		UC033	Delegation
	Manage order	UC034	**Approve**
		UC035	Cancel
		UC035	View

Continued on next page

Table 2.1: List of Use Cases (Continued)

System	Use Case Group	Use Case Code	Use Case Name
	Product view	UC035	View products from the home page
		UC036	View bestselling products
		UC037	View new products
		UC038	View discounted products
		UC039	Search
		UC040	View product details
		UC041	See similar products
		UC042	Choose to buy
		UC043	View products by type
		UC044	Filter products
	Manage shopping cart	UC045	**Edit quantity**
		UC046	Delete

Table 2.2: Describe the use case of ordering

Use Case Code	UC015	Use Case Name	Order
Actors	Client		
Preconditions	User confirms to place order		
Basic flow		Actor	Action
	1	Client	User requests to place an order

Continued on next page

Table 2.2: Describe the use case of ordering (Continued)

Use Case Code	UC015	Use Case Name	Order
Basic flow	2	System	The system displays the payment interface
	3	Client	The user enters purchase information including user information, delivery address, and delivery phone number
	4	Client	The user confirms and requests the order, if not placed then moves to a Alternate flows 4a
	5	System	The system checks order information. If the information is valid, go to step 6, if not, go to Alternate flows 5a
	6	System	The system checks whether the customer uses the voucher or not. If the voucher is used, money is deducted. Otherwise, money is not deducted.
	7	System	The system stores customer information.
	8	System	The system automatically generates order codes.
	9	System	The system saves order information and a list of ordered items into the orders database table
	10	System	The system displays the order as successful, if not then goes to Alternate flows 10a
Alternate flows		Actor	Action
	4a		User did not place an order
	1	Client	Users want to stop trading, select the cancel button
	2	System	The system displays the shopping cart interface
	5a		Invalid order information

Continued on next page

Table 2.2: Describe the use case of ordering (Continued)

Use Case Code	UC015	Use Case Name	Order
Alternate flows	1	System	The system displays the information input form and incorrectly entered fields, return to step 3
	10a		Order failed
	1	System	The system displays a message that the order creation was unsuccessful
	2	System	The system displays the information input form and incorrectly entered fields, return to step 3
	3	Client	The customer re-enters information
Post-condition			If the order is successful, an order will be saved in the system. In other cases, the system requests an error display

Table 2.3: Describe the use case for adding products

Use Case Code	UC023	Use Case Name	Add products
Actors	staff		
Preconditions	Employees add products		
Basic flow		Actor	Action
	1	Staff	The employee chooses the function to add products
	2	System	The system displays the product adding form
	3	System	The system displays information by category
	4	Staff	Employee enters product information
	5	Staff	The employee presses the save button
	6	System	The system checks the information, if not valid, switches to Alternate flows 5a. If valid, go to step 7
	7	System	The system saves information to the system
Alternate flows		Actor	Action
	5a		Invalid information
	1	System	Error message
Post-condition			If the product is successfully added, a notification will be displayed. In other cases, the system displays an error message

Table 2.4: Describe the use case for Search product

Use Case Code	UC026	Use Case Name	Search product
Actors	staff		
Preconditions	Employees search for products		
Basic flow		Actor	Action
	1	Staff	Employees enter the product they want to search for in the search bar
	2	System	The system displays the products to search for
Alternate flows	none		
Post-condition			If the product search is successful, the product will be displayed. In other cases, the system will display the product not found

Table 2.5: Describe the use case for Approve order

Use Case Code	UC034	Use Case Name	Approve order
Actors	staff		
Preconditions	Staff approves orders		
Basic flow		Actor	Action
	1	Staff	The employee chooses the order approval function
	2	Staff	The employee requested to approve the order
	3	System	The system displays a confirmation message of order approval, the order is approved in the system. If the order is not approved, it will be transferred to Alternate flow 3a
Alternate flows		Actor	Action
	3a		Staff does not approve orders
	1	Staff	The employee does not want to approve that order, select the cancel button
	2	System	Exit the order approval form
Post-condition			If an order is approved, one order will be approved. In other cases, the system requests an error display

Table 2.6: Describe the use case for Edit quantity

Use Case Code	UC045	Use Case Name	Edit quantity.
Actors	Customer		
Preconditions	Customer change product quantity.		
Basic flow		Actor	Action
	1	Customer	The Customer changes the number of products in the shopping cart on the shopping cart interface of the website
	2	System	The system will change the number of products in the shopping cart. If the quantity reaches 0, it will go to flow 2a
Alternate flows		Actor	Action
	2a		User reduces product to 0
	1	Customer	User reduces product to 0
	2	System	Remove product from cart
Post-condition			The system changes the number of products in the shopping cart.

2.4 Non-functional requirement

The application is designed with a mechanism to decentralize users according to each role and function, ensuring that each user can only access and use features consistent with their permissions. Structurally, building the application in separate modules makes it easy to upgrade when necessary.

2.4.1 Efficiency

Fast speed: Ensures that web pages load within up to 10 seconds, even during high traffic.

2.4.2 Possibility

Suitable for customer needs: Regularly survey and collect feedback from users to ensure the website's features and content meet the needs and desires of customers.

Ease of use: The user interface is intuitive and easy to navigate, supports multiple languages, and provides customer support through multiple channels such as live chat, email, and phone.

2.4.3 Reliability

SSL certificate: The website must be protected with a valid SSL certificate, ensuring all communication between the user and the server is encrypted.

2.4.4 Maintainability

Modular structure: Using modular architecture to make maintenance and updates easier, parts can be replaced or upgraded without affecting the entire system.

2.4.5 Environment

Browser and internet: Ensure that users only need to install a browser and have an internet connection to access and use the full functionality of the website.

Browser compatibility: Test and ensure the website works well on popular browsers such as Chrome, Firefox, Safari, and Edge.

2.4.6 Security

Information security: Use security measures such as data encryption, firewalls, and intrusion detection systems to protect user information.

Authentication and authorization: Implement strong authentication and access management systems to ensure only authorized users can access sensitive parts of the system.

2.4.7 Portability

Cross-platform: Ensures that the system can run on many different environments such as desktop and mobile, with a compatible and optimized user interface for each device.

Responsive design: Use responsive design so the website automatically adjusts the interface to suit the device's screen size.

2.4.8 Interactive abillity

Works well: Website functions need to operate correctly and efficiently, ensuring users receive quick feedback after each operation.

User experience: Optimize user experience with a friendly interface, clear interactive elements and ease of use.

2.4.9 Regulations

Legal compliance: Ensure the website complies with state regulations on information security, privacy, and other technical standards.

Clear policies: Provide clear and easy-to-understand privacy policies and terms of use so that users clearly understand their rights and obligations when using the website.

2.4.10 Load

Concurrent processing ability: Ensure the system can handle 1000 visits at the same time without affecting performance and user experience.

3.1 Frontend

3.1.1 Angular

Angular is an open source code written in TypeScript and used to design web interfaces (front-end). Angular was built and developed in 2009 and is maintained until now by Google. This is considered a powerful and specialized front end framework for programmers using advanced HTML.[1]

Angular is widely used because it has many advantages over other frameworks, including:[1]

- **TypeScript:** Angular uses TypeScript, an extended version of JavaScript with static data types, to help develop web applications in a structured, easy to manage and maintain manner.
- **Two-way data binding:**Angular provides two-way data binding between the model and the user interface (view). This synchronizes data automatically between components, saving developers time and effort.
- **Dependency Injection:** Angular uses Dependency Injection (DI) to manage and deliver objects to application components dynamically. This helps create code that is easy to maintain and reuse.
- **Routing:** Angular provides a powerful routing system, allowing for easy and flexible transitions between pages, improving user experience.
- **Forms:** Angular provides powerful features for handling and validating input from users, making form management easy and efficient.
- **Community and ecosystem:** Angular has a large and robust community, which comes with a rich ecosystem of supporting libraries and tools. This helps developers easily find solutions and receive support when needed.

3.1.2 RxJS

RxJS is Asynchronous programming library for JavaScript. RxJS is used in Angular to process data streams such as events from the server or user interactions.

3.1.3 Angular CLI

Angular CLI is a tool developed to run on the application window interface (command line interface or CLI) to help create projects, manage files in projects

[1]https://topdev.vn/blog/angular-la-gi/

and perform many different tasks such as testing. (test), bundle and deploy the project quickly and efficiently.[2]

AngularCli helps developers create an application faster and easier through simple commands and, importantly, helps in the future maintenance/upgrade of the project. The current version of Cli is 1.0.0 released on 2017-03. -24 and is still being improved by developers

To sum up, Angular CLI is Command line tool to create, build, and test Angular applications quickly and easily.

3.1.4 Angular Material

Angular's UI library is designed according to Google's Material Design principles. It provides standard user interface components to create applications with beautiful and easy-to-use user interfaces.

3.2 Back-end

3.2.1 Java

Java is a modern, high-level, object-oriented, secure, and powerful programming language. and is a Platform.[3]

Platform: Any hardware or software environment in which a program runs is known as a Platform. With its own runtime environment (JRE) and API, Java is called Platform.

Java programming language was originally developed by Sun Microsystems initiated by James Gosling and released in 1995. The latest version of Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity its multiple configurations have been built to suit a variety of platforms. For example: J2EE for enterprise applications, J2ME for mobile applications.

The new J2 versions have been renamed Java SE, Java EE, and Java ME. Java's motto is "Write Once, Run Anywhere" - write once, run many places, meaning you only need to write once on windows for example, then you can run the same program on Linux, Android, other devices. suffer from J2ME...

Java programming language has the following features:

- Object Oriented - In Java, everything is an Object. Java is easily extensible and maintainable because it is built on the Object model.
- Platform independent - Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine language, rather bytecode - platform independent create. This byte-

Figure 3.1: Java logo

code is interpreted by the virtual machine (JVM) on whatever platform it is running on.

- Simplicity - Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it will be very easy to become a master of Java.
- Security - With Java's secure feature, it allows developing virus-free, tamper-free systems. Authentication techniques are based on public key encryption.
- Architecture - neutral - The Java compiler generates an architecture-neutral object file format, making the compiled code executable on multiple processors, in the presence of a Java operating system.
- Portable - The centralized architecture and no implementation dependency aspect of this specification makes Java portable. Compilers in Java are written in ANSI C, which is a POSIX subset.
- Robust - Java makes efforts to eliminate error-prone situations by error checking at compile time and error checking at runtime.
- Multithreading - With Java's multithreading feature it is possible to write programs that can perform multiple tasks simultaneously. This design feature allows developers to build interactive applications that can run more smoothly.
- Interpreted - Java bytecode is translated directly to the original computers and is not stored anywhere.
- High Performance - With the use of Just-In-Time compiler, Java allows high

performance implementation.

- Distributed - Java is designed for the distributed environment of the Internet.
- Dynamic - Java is more dynamic than C or C++ because it is designed to adapt to the evolving environment. Java programs can carry large amounts of information at runtime that can be used to verify and resolve accesses to objects at runtime.

There are 4 main types of Java apps:

- **Standalone App** is also known as Desktop App or Window-based App. To create this type of application, people often use AWT, Swing or JavaFX framework.
- **Web App** is an application that runs on a server and creates dynamic pages. Currently, servlet, jsp, struts, jsf, spring... are the technologies used to create Web App in java.
- **An application** like Banking App has the advantages of high security, load balancing and clustering. In java, EJB is used to create Enterprise Apps.
- **Mobile App** is an application created for mobile devices. Currently Android and Java ME are used to run these applications.

3.2.2 Spring Framework

Spring Framework makes it easy for programmers to create Java Enterprise (Java EE) applications and provides everything you need to use the Java language. Being open source, Spring has a large and active community that provides continuous feedback based on many real-life use cases. This has helped Spring develop successfully and become a source of knowledge that any programmer needs to understand.

Spring Framework is divided into modules. Applications can select the required module. The focus is on core container modules, including the configuration model and dependency injection mechanism.

In addition, Spring Framework provides platform support for many different application architectures, including messaging, data streaming, and web.

Some advantages of Spring Framework can be mentioned as follows:[2]

- **Pre-defined Templates:** Spring framework contains many types of templates for Hibernate, JDBC and JPA technologies. With the help of this method, programmers do not need to define complex code. For example: JdbcTemplate

[2]https://itviec.com/blog/spring-framework-la-gi/

– Here, we don't need to write logic to create a statement, confirm transaction, create connection and handle exception. This helps save time.

- **Easy and simple to test:** You can easily test the entire application just by using the Spring framework with dependency injection mechanism. EJB or Struts applications require a server to execute the application.
- **Non-intrusive:** Following the Plain Old Java Object (POJO) technique, Spring is easy to implement because it does not force the programmer to inherit specific classes or implement on any interface.
- **Rapid development:** With the help of dependency injection, framework integration and support for Java EE (Enterprise Edition) based application development become easy.
- **Strong abstraction support:** Spring supports strong abstraction capabilities for Java EE-based provisions such as JMS, JDBC, JPA, and JTA.
- **Neat:** Spring web framework is an MVC web framework that offers a great alternative to application development web frameworks using Struts or other popular web frameworks.
- **Spring provides a suitable API:** Spring translates technology specific exceptions from JDBC, Hibernate or JDO into unified and uncontrolled exceptions.
- **Lightweight IoC (Inversion of Control):** IoC is lightweight, especially compared to EJB containers. This advantage helps create and deploy applications on computers with limited memory and CPU resources.
- **Continuous transaction management** : Spring provides an interface that can help scale down to local transactions (e.g., using a database) and scale up to global transactions (e.g., JTA) .

Security: Spring Security is an important part of the Spring ecosystem, providing security features such as authentication, authorization, and session management.

3.2.3 Spring Boot

Spring Boot is one of the modules of the Spring framework that specializes in providing RAD (Rapid Application Development) features that allow the creation and development of independent applications based on Spring quickly.[3]

Spring Boot was born with the purpose of eliminating the complicated configuration of Spring, it does not require XML configuration and improves productivity

[3]https://stringee.com/vi/blog/post/java-spring-boot-la-gi

for developers. With the presence of Spring Boot, the Spring ecosystem has become stronger, more popular and more effective than ever.

With its very clear purpose, Spring Boot has overcome Spring's configuration limitations.

Those advantages include:

- Full convergence of Spring framework features.
- Simplify configuration and build standalone applications that can run in "java -jar" thanks to dependency starters.
- Easy to deploy because server applications are embedded directly into the application to avoid difficulties when deploying to production environments without the need to download WAR files.
- Low configuration, automatic support at any time for Sping-like functionality such as increased productivity, reduced coding time and no XML configuration required.
- Provides many external plugins, metrics, and application configurations.

RESTful APIs: Spring Boot provides good support for building RESTful APIs through Spring MVC, allowing data access from other applications via HTTP protocol.

- **Swagger/OpenAPI:** Used to create and document APIs. Swagger helps developers understand how to use APIs easily and conveniently.

4.1 Architecture design

The website is designed according to an architecture combining three tier and MVC models[4] Combining the two models, we get a complete Spring Boot application, including the following components:

- Controller: returns View (containing available data, HTML page form), or Model expressed as API for View (View written specifically in React, Vue, or Angular).
- Service: contains calculation and processing code. When the Controller requests, the corresponding Service will receive and return data to the Controller (returning the Model). The Controller will send back the View as above.
- Repository: Service can also interact with other services, or use Repository to call the DB. Repository is the one that directly interacts, reads and writes data in the DB and returns it to the service.

To design the interface, I use Angular as a fond-end. Details will be presented below

4.1.1 Software architecture selection

4.1.1.1 Front-end architecture

Document Object Model (DOM): The DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

The W3C DOM standard is separated into 3 different parts:

- Core DOM - standard model for all document types
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

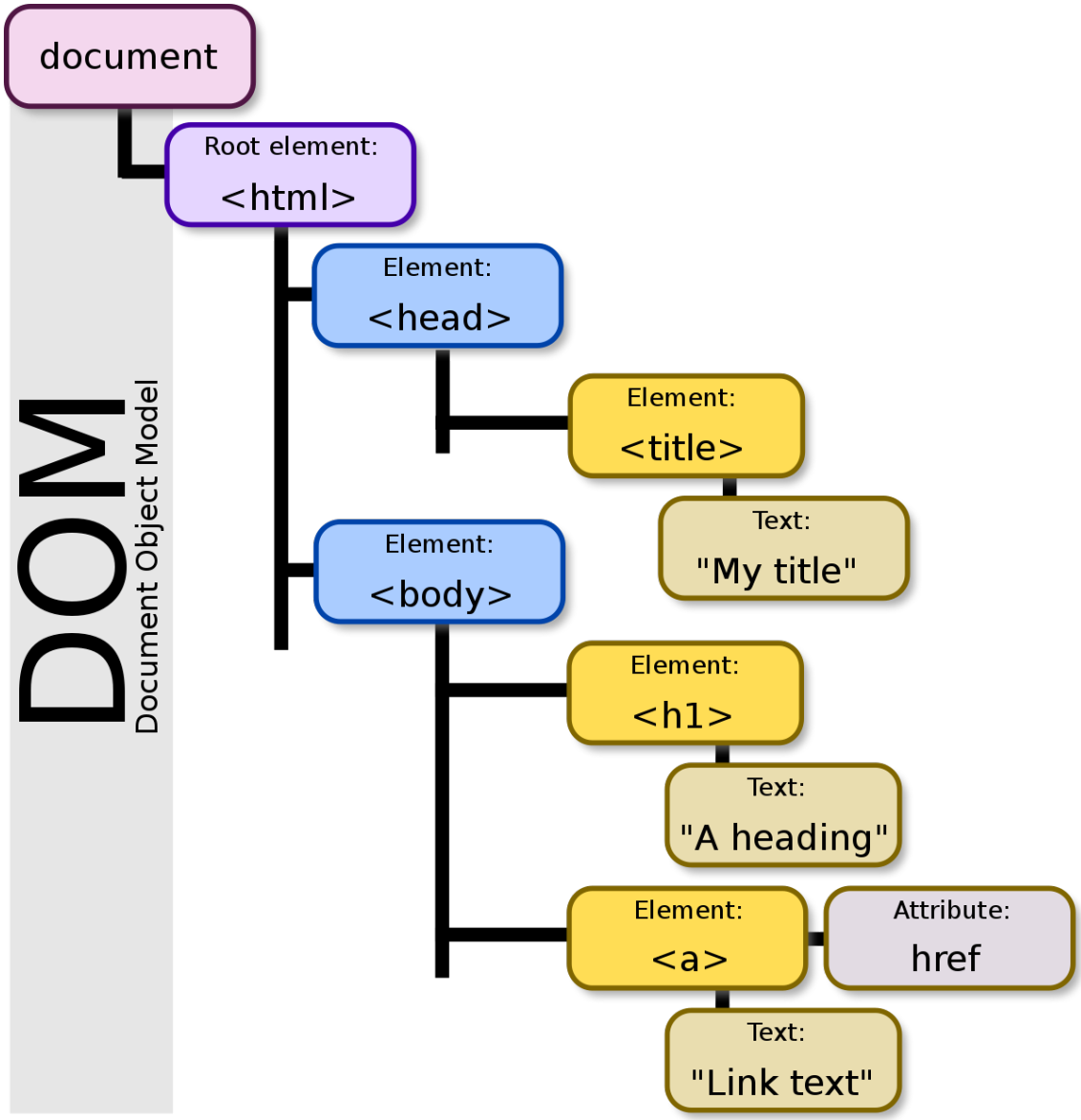

Figure 4.1: DOM structure tree

MVVM architecture:

MVVM in Angular is an acronym for Model-View-ViewModel, a software architecture used in application programming.

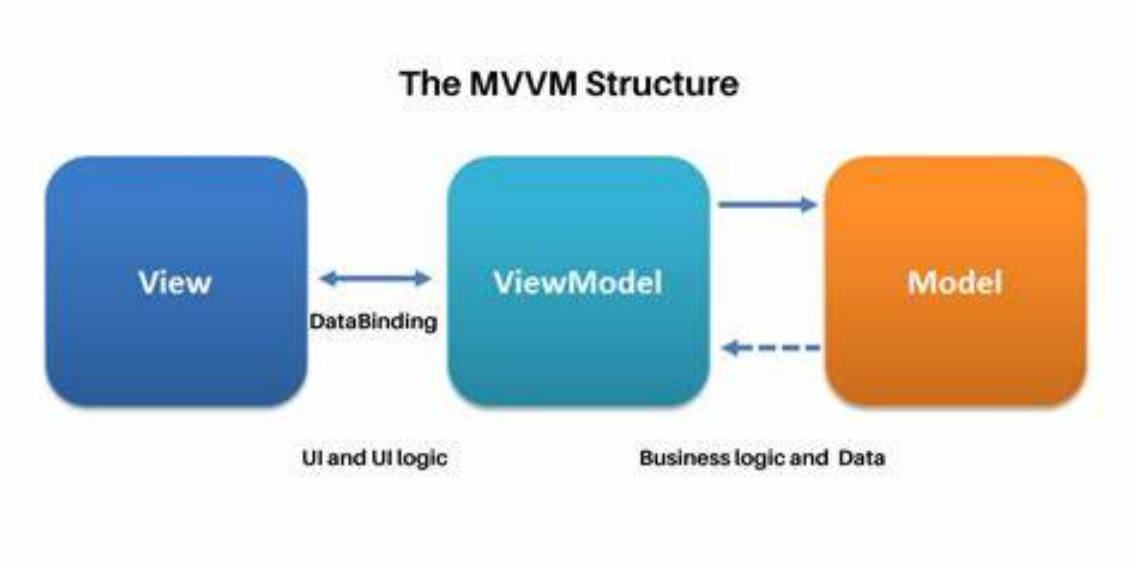

Figure 4.2: The MVVM Structure

MVVM is divided into three main components:

- Model (M): Represents the data and logic of the application. The model represents the objects and state of the system.
- View (V): Represents the user interface, i.e. what the user sees and interacts with. Views represent interface elements such as buttons, input fields, and other interactive elements.
- ViewModel (VM): Acts as an intermediary between Model and View. The ViewModel provides the data and behavior necessary for the View to display and interact with the Model. The ViewModel is unaware of the existence of the View, which separates the application logic and user interface.

Organizing and managing code according to MVVM architecture helps create applications that are easy to maintain, flexible and efficient in data processing and user interface. This makes Angular one of the popular and powerful frameworks for building modern and interactive web applications.

4.1.1.2 Back-end architecture

Java Spring Boot MVC:

Java Spring Boot MVC is a combination of Three-Tier and MVC models. This system includes the following components:

- Controller: Receives requests and data from the user, then returns a View (an HTML page containing data) or a Model that is converted into an API to provide to the View (View is written specifically with frameworks such as React, Vue, or Angular).

- Service: Contains code that performs calculations and processing. When receiving a request from the Controller, the Service will process and return data to the Controller (in the form of Model). The Controller will forward this data to the View as described.
- Repository: Service can interact with other services or use Repository to access the database. Repository is the component that interacts directly with the database, including reading and writing data, then returning results to the Service.
- Model: Are objects that contain data after being processed by the Service and returned to the Controller.
- View: There are two types of View: The traditional type returns an HTML page that already contains data. In this case, the Controller will insert data into the View and return it (Spring MVC uses tools like JSP or Thymeleaf to do this). The View type is developed independently, not part of the Spring Boot project. Common in systems that use APIs. View is developed using frameworks such as React, Angular,... and Controller will provide Model data through API to View, and receive requests from View via API.

4.1.2 Overall design

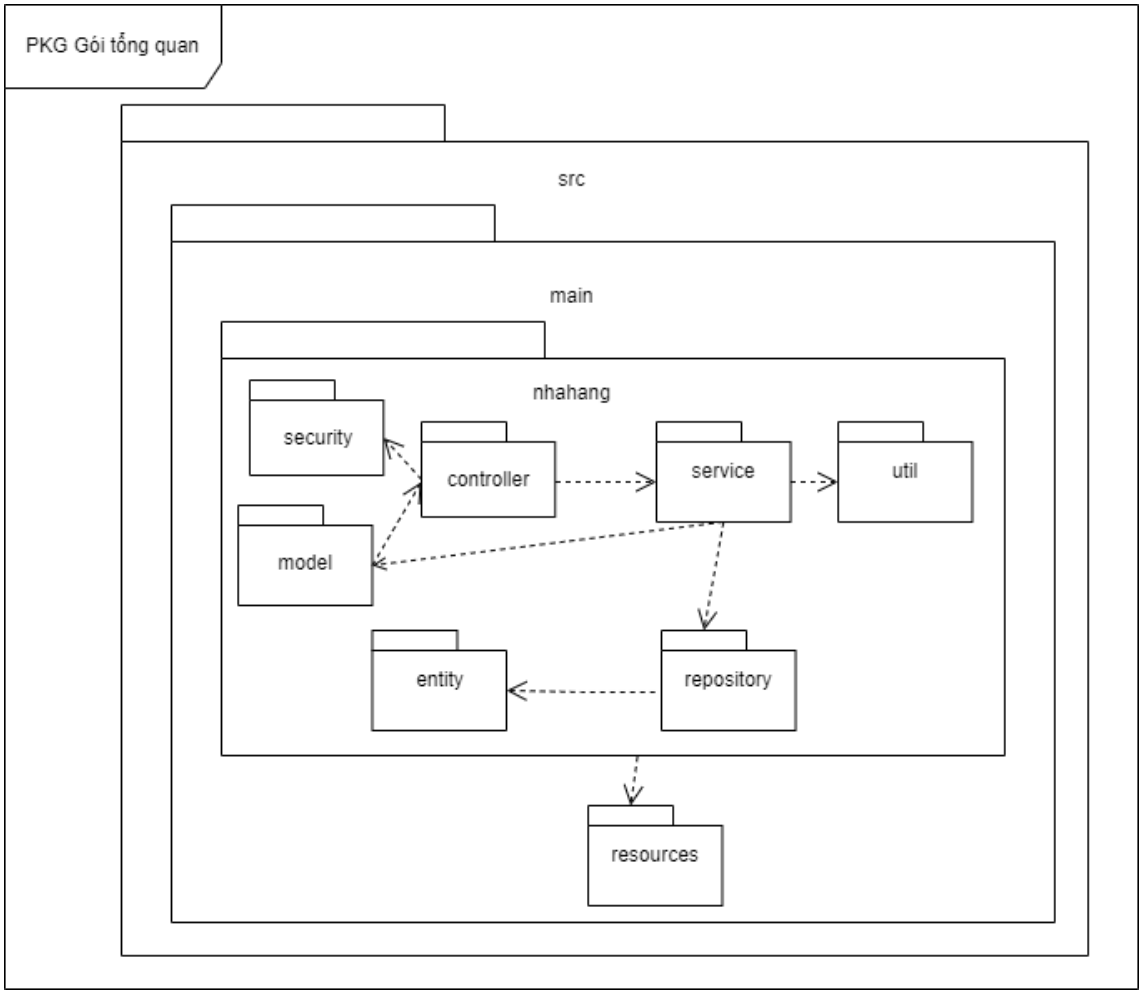

Figure 4.3: Package diagram of the program

The Controller receives the request and starts calling the Service's methods.

Service receives request from Controller. With simple requests, the Service can process and return results immediately. However, if you need to manipulate the database, the Service will call the Repository to retrieve data.

Repository receives requests from Service and performs operations with the database. Data retrieved from the database will be converted by the system into Java objects, called Entities.

Service receives Entities from Repository, processes them and transforms them into Models. This model is then returned to the Controller.

Controller receives Model and returns it to View. There are two ways to do this:

- Use template engine to insert Model data into HTML page and return a complete HTML page to the client.

- Send data via API for View to analyze and display (how to display is determined by View).

When the View is finished displaying, the user will see a list of users on the website.

4.1.3 Detailed package design

4.1.3.1 Design package details for order function

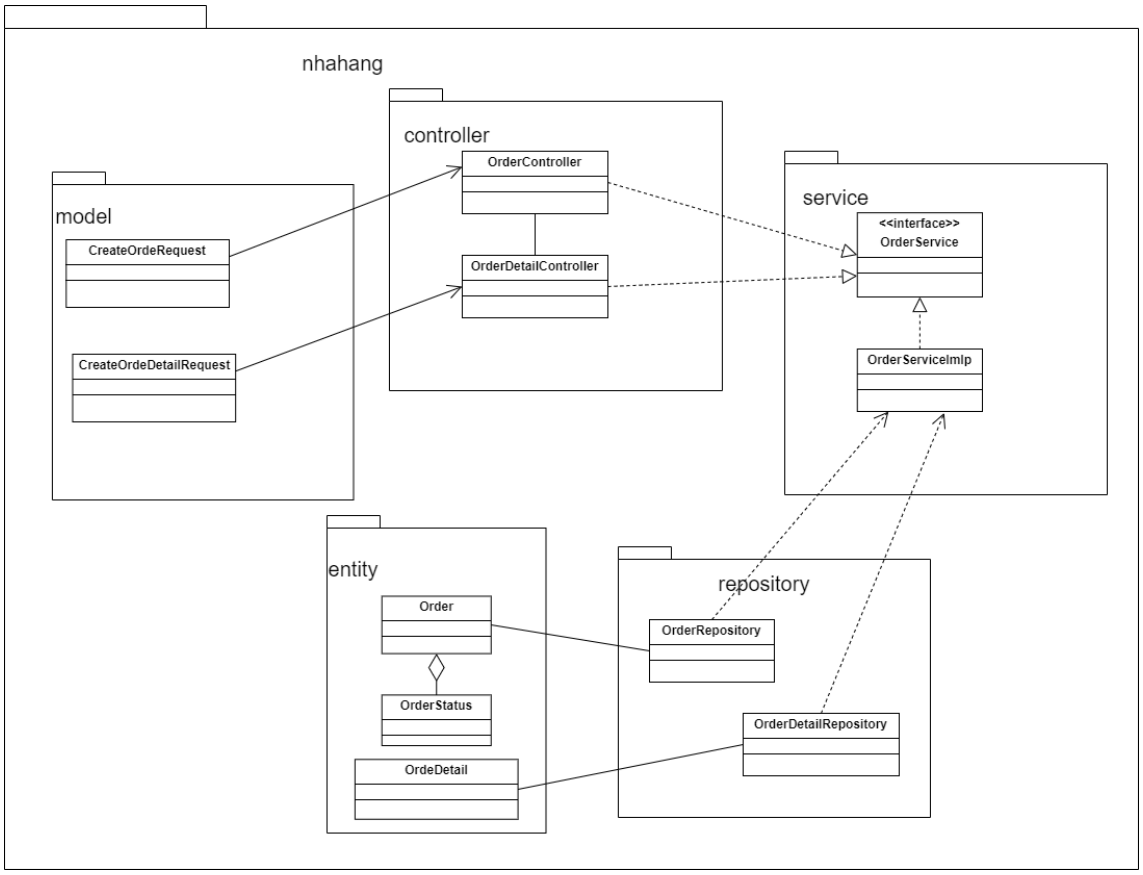

Figure 4.4: Design package details for order function

Figure 4.4 The picture describes the details of the order package. This package includes CreateOrderRequest and CreateOrderRequest which contain methods that ensure the data is valid before processing. OrderController and OrderDetail-Cotroller contain methods for handling user requests. OrderService - An interface containing the rules of the order function. OrderServiceImpl - contains methods that implement order functions. OrderRepository - contains methods to query the order's database. OrderDetalRepository - contains methods to query the order detai database. Order, OrderStatus, OrderDetail contain entity objects

4.2 Detailed design

4.2.1 User interface design

The interface is arranged in a layout that is close to the user. The information in the interface is designed to fully satisfy the user.

The application is designed with high interactivity, responding quickly when users perform operations. The interface is friendly and provides clear suggestions through icons and navigation bars, making it easy for users to use without having to learn much.

Regarding color, the application uses colors appropriate to each function, such as red for error and blue for success. The overall color tone of the application is pleasant, avoiding using too many colors that distract the user's eyes. The application is designed to be user-friendly, limit the use of too many colors, and prioritize a simple but attractive interface.Website use multiple images to attract customer attention

The program's interface is designed to suit all types of screens. Does not require a screen with too large resolution or high color coverage

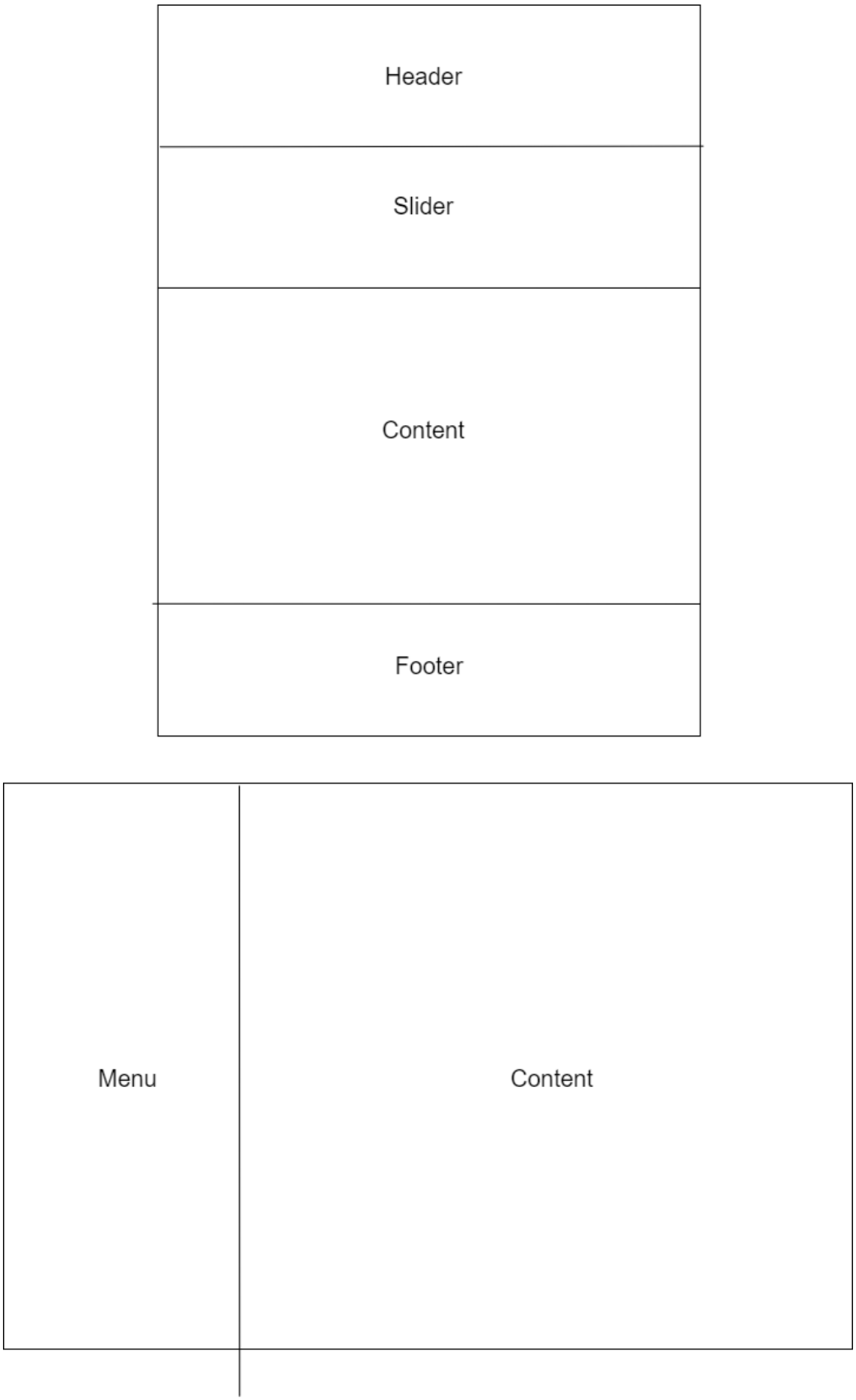

Figure 4.5: Typical interface frame of the web

4.2.2 Layer design

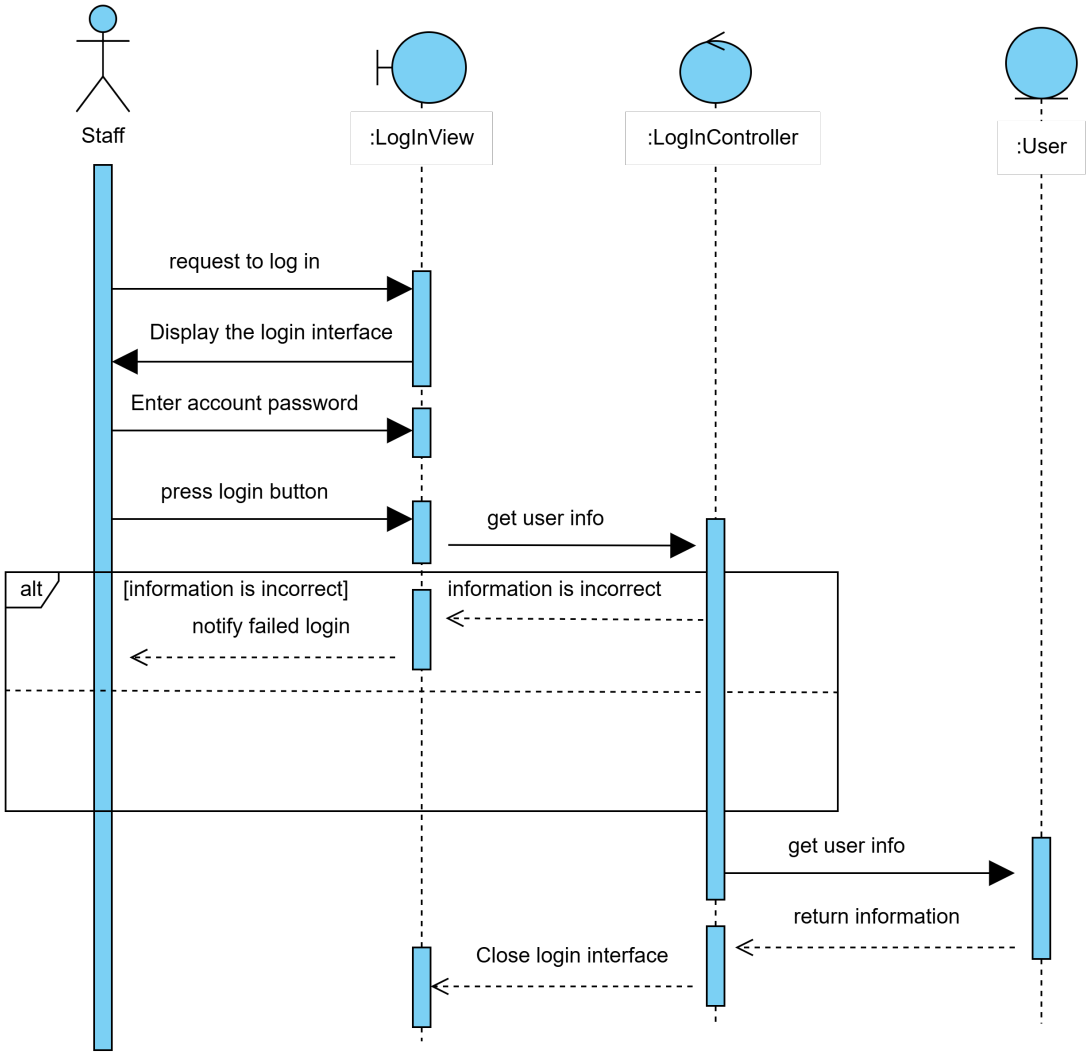

Figure 4.6: Login sequence diagram

Figure 4.6 describes the message flow for the login function. When users log in by filling in information about their account and password. When the user presses the login button, the system will retrieve the user's data to match with the database. When logged in successfully, the system will automatically search for the login user's role to navigate to the interface with functions suitable for the user.

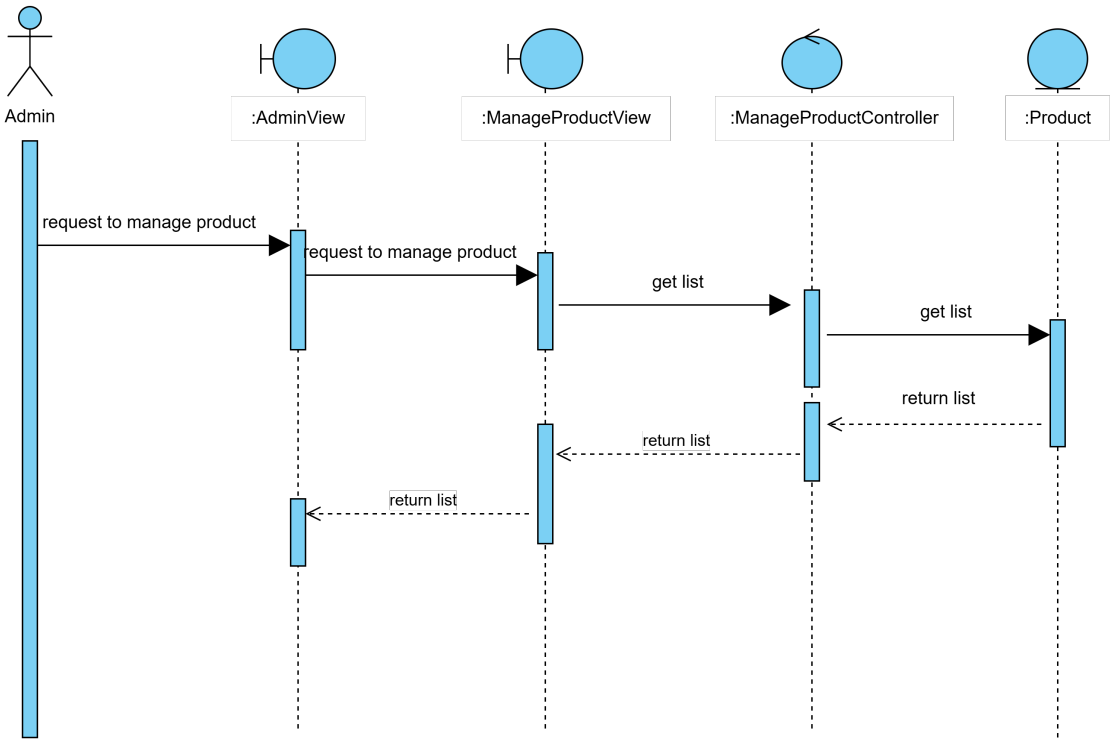

Figure 4.7: Product management sequence diagram

Figure 4.7 describes the message flow for product management. When a user requests product management, the system will retrieve a list of products from the database and then display it to the user.

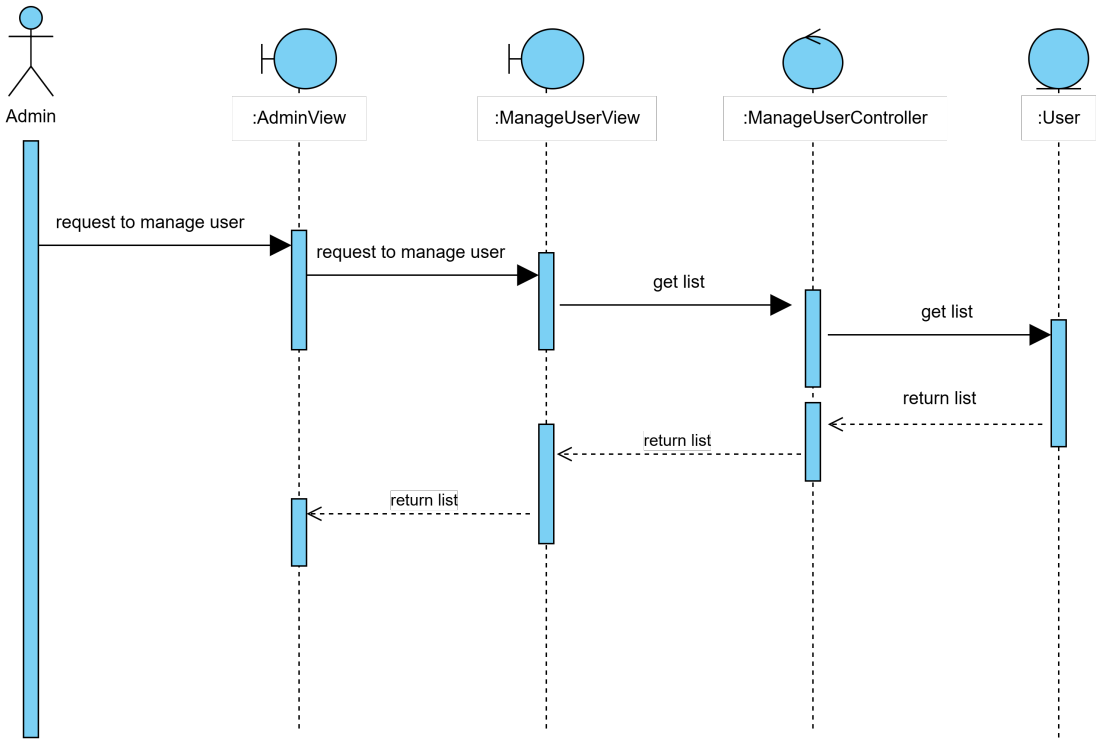

Figure 4.8: User management sequence diagram

Figure 4.8 describes the message flow for the user management function. When an employee requests to manage users, the system will retrieve a list of users and display it to the employee

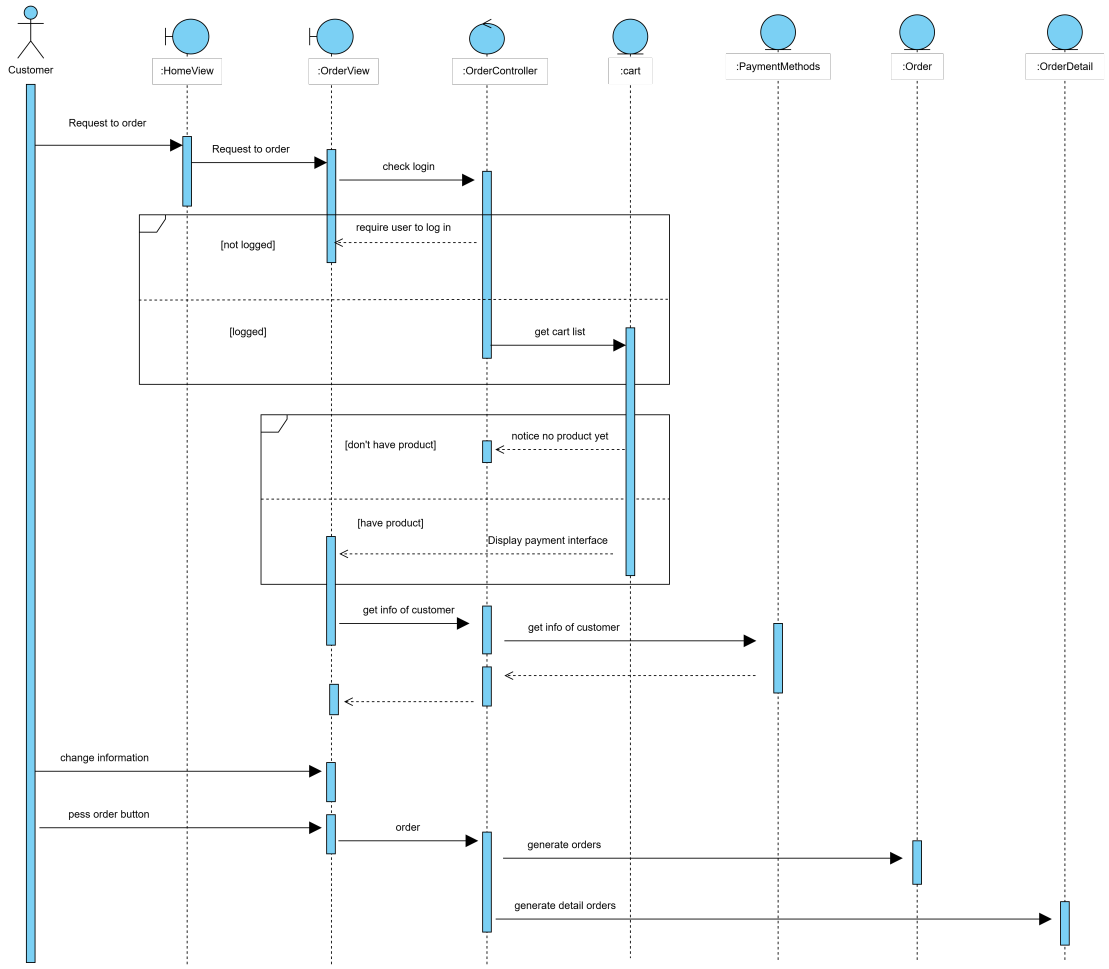

Figure 4.9: Order sequence diagram

Figure 4.9 describes the message flow for the order function. When a customer requests to place an order, the system will check whether the customer is logged in or not. If not, request a login. If so, the system will get the shopping cart list from the database. If there are no products in the cart, the system will notify the user. If there are already products in the cart, the system will display the payment interface. After the customer presses the order button, the system will generate an order.

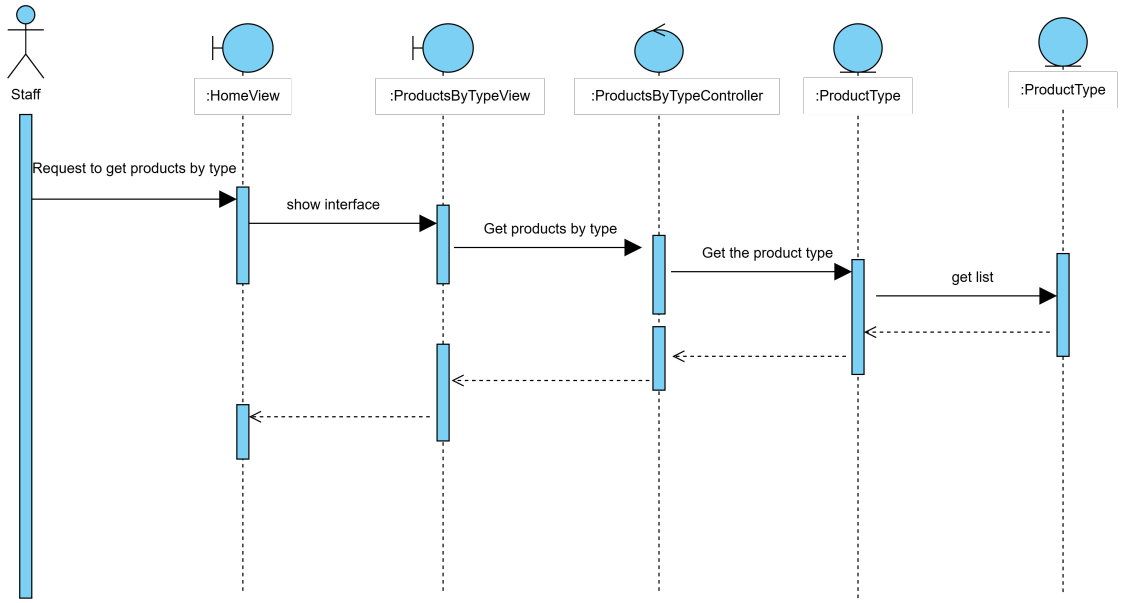

Figure 4.10: Product view sequence diagram

Figure 4.10 describes the message flow for the Product view function. After the user presses the view product button, the system will retrieve product information from the database and display it to the user.

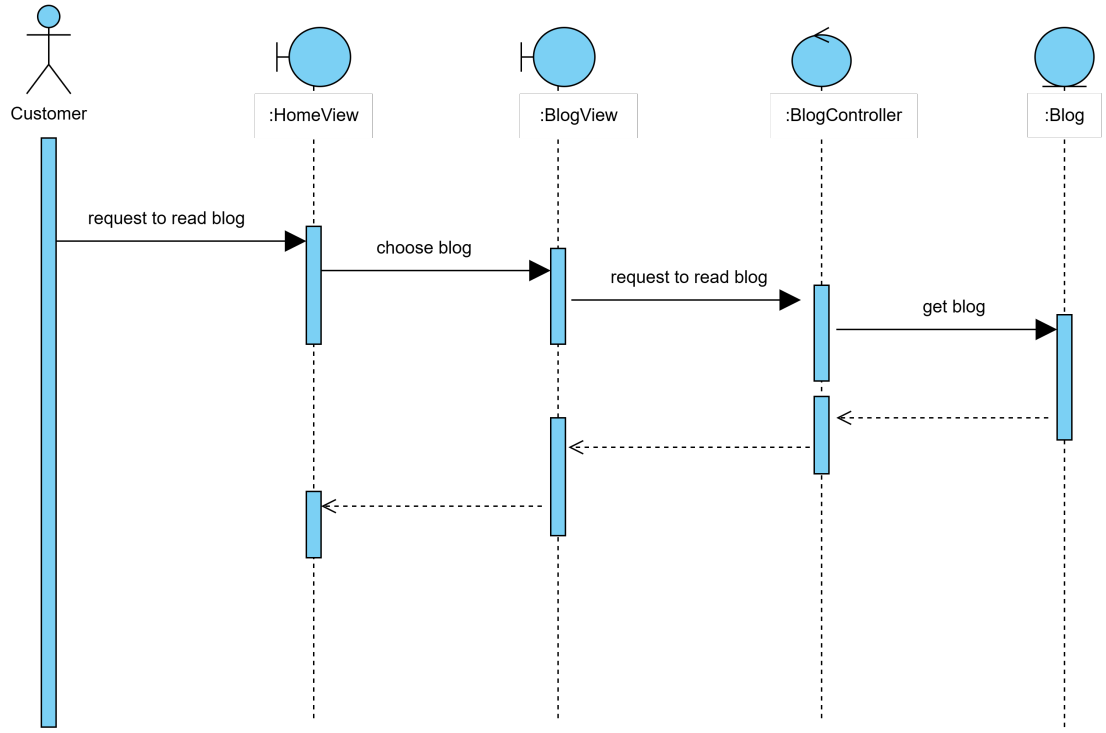

Figure 4.11: Blog reading sequence diagrams

Figure 4.11 describes the message flow for the reading blog function. after the

user presses the blog button and selects the blog the user wants to read. The system will retrieve the blog's information and display it to the user

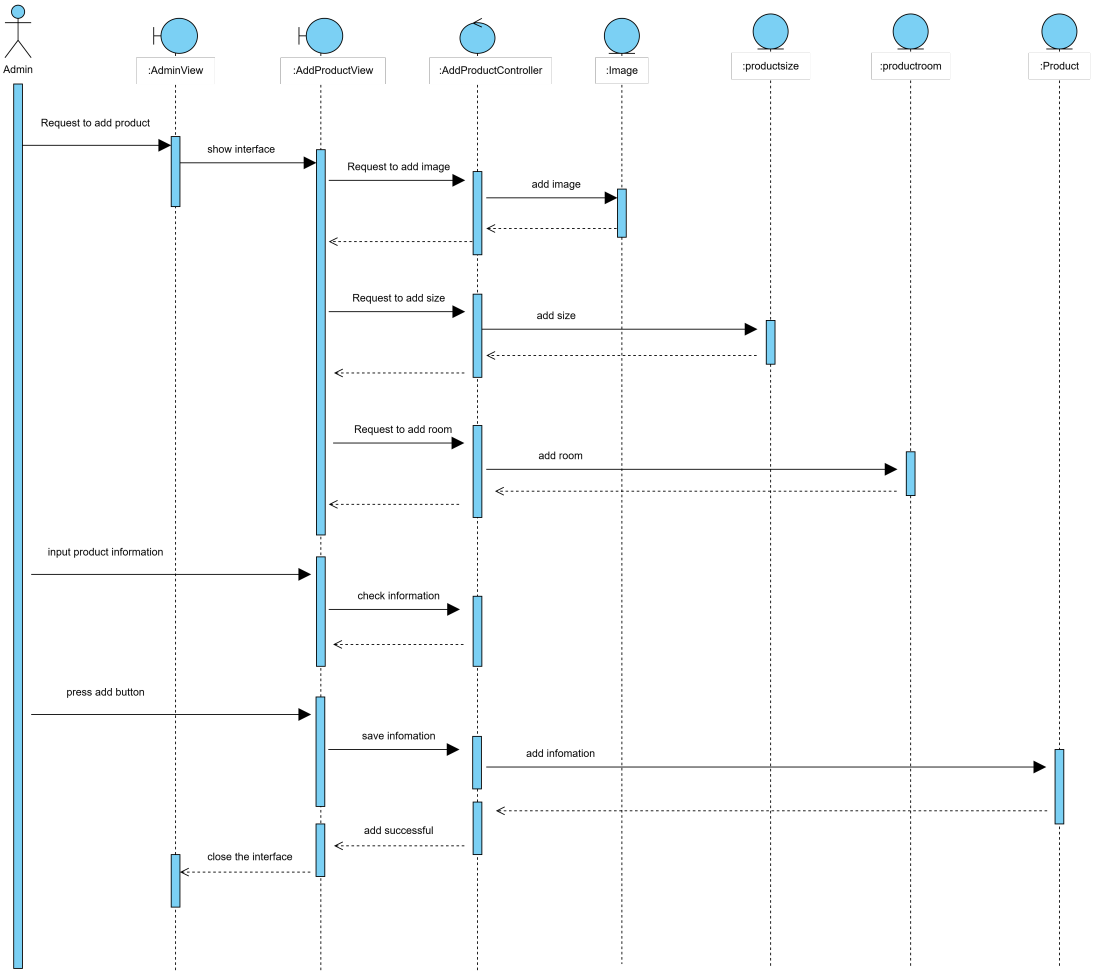

Figure 4.12: Adding products sequence diagrams

Figure 4.12 describes the message flow for the add product function. After the admin requests more products, the system will display the interface and ask the admin to fill in the necessary information. After filling in all information, the admin presses the add product button, the system will add product information and database. After adding information successfully, the system displays a success message.

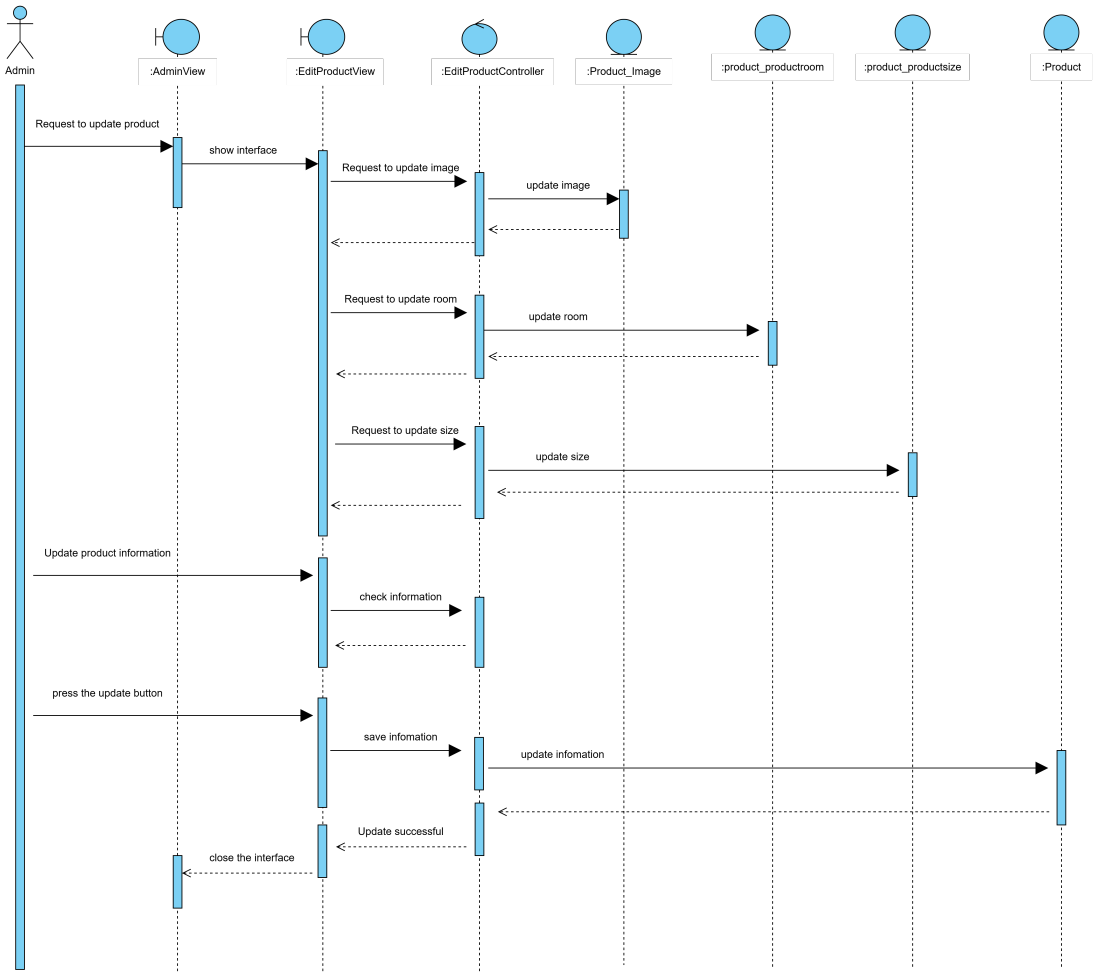

Figure 4.13: Repair products sequence diagrams

Figure 4.13 describes the message flow for the product update function. The admin requests to update the product, the system requires the admin to fill in the product information. After entering admin information, press the product update button, the system will update the information into the database and notify you of success.

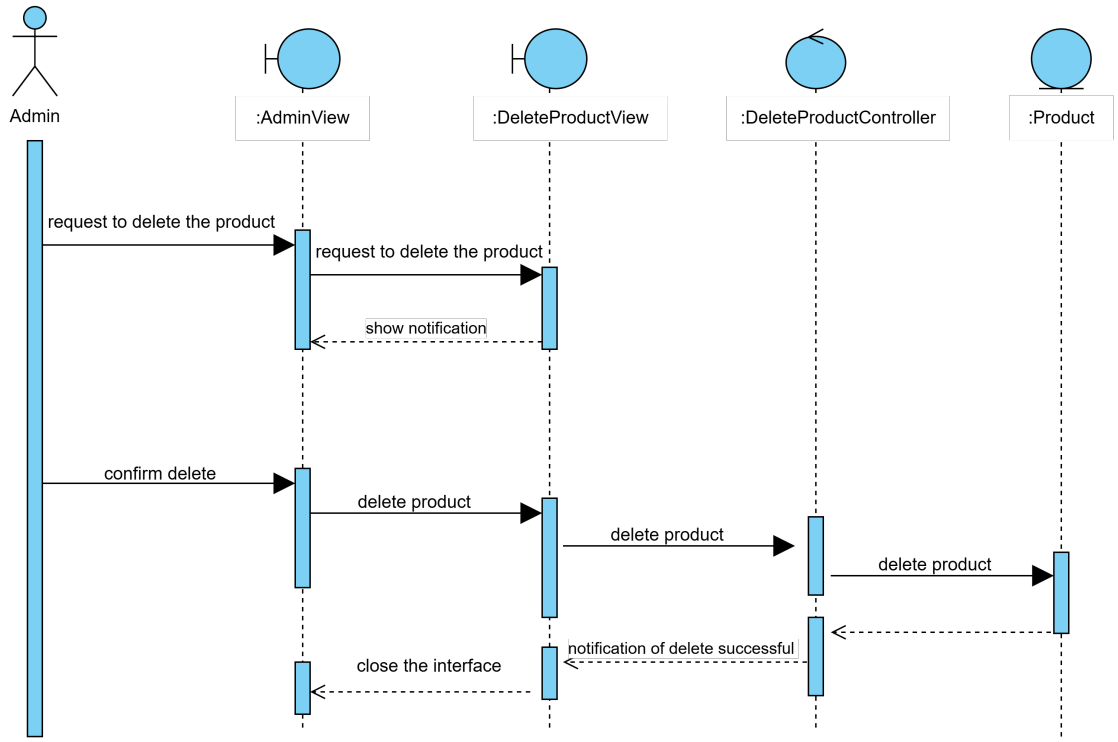

Figure 4.14: Delete products sequence diagrams

Figure 4.14 describes the message flow for the product deletion function. After the admin requests to delete the product, the system displays a notification. Admin presses the delete product button, the system deletes product information from the database and displays a notification

4.2.3 Database design

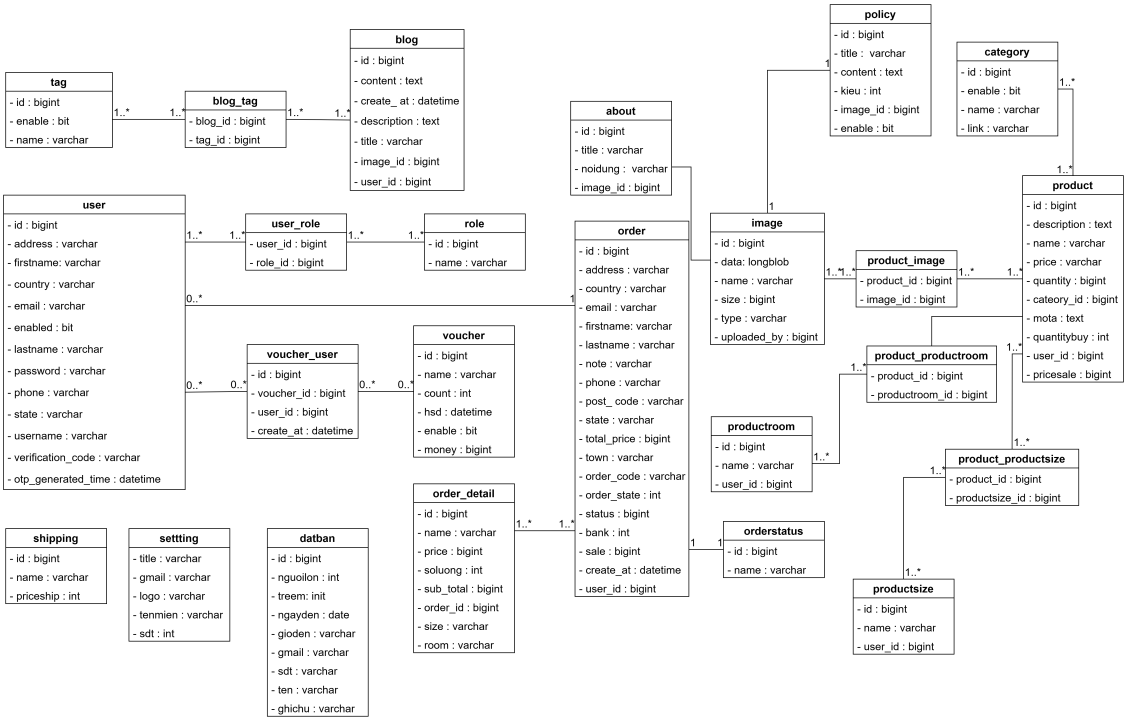

Figure 4.15: Detailed database design

4.2.4 database specification

Table 4.1: Product database specification

	Attribute name	Datatypes	Constraint
1	ID	bigint	primary key
2	description	text	
3	name	varchar	
4	price	varchar	not null
5	quantity	bigint	not null
6	cateory_id	bigint	Foreign Key
7	mota	text	
8	quantitybuy	int	
9	user_id	bigint	Foreign Key
10	pricesale	bigint	

Table 4.2: Product image database specification

	Attribute name	Datatypes	Constraint
1	product_id	bignit	primary key
2	image_id	bignit	Foreign Key

Table 4.3: About database specification

	Attribute name	Datatypes	Constraint
1	id	bignit	primary key
2	title	varchar	
3	noidung	varchar	
4	image_id	bignit	Foreign Key

Table 4.4: Image database specification

	Attribute name	Datatypes	Constraint
1	ID	bigint	primary key
2	data	longblob	
3	name	varchar	
4	size	bigint	not null
5	type	varchar	
6	uploaded_by	bigint	Foreign Key

Table 4.5: Product productroom database specification

	Attribute name	Datatypes	Constraint
1	product_id	bignit	primary key
2	productroom_id	bignit	Foreign Key

Table 4.6: Product room database specification

	Attribute name	Datatypes	Constraint
1	id	bignit	primary key
2	name	varchar	
3	user_id	bignit	Foreign Key

Table 4.7: Product productsize database specification

	Attribute name	Datatypes	Constraint
1	product_id	bignit	primary key
2	productsize_id	bignit	Foreign Key

Table 4.8: Product size database specification

	Attribute name	Datatypes	Constraint
1	id	bignit	primary key
2	name	varchar	
3	user_id	bignit	Foreign Key

Table 4.9: Book database specification

	Attribute name	Datatypes	Constraint
1	ID	bigint	primary key
2	nguoilon	text	
3	treem	varchar	
4	ngayden	varchar	
5	gioden	bigint	
6	gmail	bigint	
7	sdt	text	
8	ten	int	
9	ghichu	bigint	

Table 4.10: Orders database specification

	Attribute name	Datatypes	Constraint
1	ID	bigint	primary key
2	address	varchar	
3	country	varchar	
4	email	varchar	
5	firstname	varchar	
6	lastname	varchar	
7	note	varchar	
8	phone	varchar	
9	post_code	varchar	not null
10	state	varchar	
11	total_price	bigint	not null
12	town	varchar	
13	user_id	bigint	Foreign Key
14	order_code	varchar	
15	order_state	int	not null
16	status	bigint	
17	bank	int	
18	sale	bigint	
19	create_at	datetime	

Table 4.11: Order detail database specification

	Attribute name	Datatypes	Constraint
1	ID	bigint	primary key
2	name	varchar	
3	price	bigint	not null
4	soluong	ỉnt	not null
5	sub_total	bigint	not null
6	order_id	bigint	Foreign Key
7	size	varchar	
8	room	varchar	

Table 4.12: Order status database specification

	Attribute name	Datatypes	Constraint
1	id	bignit	primary key
2	name	varchar	

Table 4.13: User database specification

	Attribute name	Datatypes	Constraint
1	ID	bigint	primary key
2	address	varchar	
3	country	varchar	
4	email	varchar	Foreign Key
5	enabled	bit	not null
6	firstname	varchar	
7	lastname	varchar	
8	password	varchar	
9	phone	varchar	
10	state	varchar	
11	username	bigint	Foreign Key
12	verification_code	varchar	
13	otp_generated_time	datetime	

Table 4.14: User role database specification

	Attribute name	Datatypes	Constraint
1	user_id	bignit	primary key
2	tole_id	bignit	Foreign Key

Table 4.15: Role database specification

	Attribute name	Datatypes	Constraint
1	id	bignit	primary key
2	name	varchar	

Table 4.16: Blog database specification

	Attribute name	Datatypes	Constraint
1	ID	bigint	primary key
2	content	text	
3	create_at	datetune	
4	description	text	
5	title	varchar	
6	image_id	bigint	Foreign Key
7	user_id	bigint	Foreign Key

Table 4.17: Blog tag database specification

	Attribute name	Datatypes	Constraint
1	blog_id	bignit	primary key
2	tag_id	bignit	Foreign Key

Table 4.18: Tag database specification

	Attribute name	Datatypes	Constraint
1	id	bignit	primary key
2	enable	bit	not null
3	name	varchar	

Table 4.19: Voucher database specification

	Attribute name	Datatypes	Constraint
1	ID	bigint	primary key
2	name	varchar	Foreign Key
3	count	int	
4	hsd	datetime	
5	enable	bit	
6	money	bigint	

Table 4.20: Policy database specification

	Attribute name	Datatypes	Constraint
1	ID	bigint	primary key
2	title	varchar	
3	content	text	
4	kieu	int	
5	image_id	bigint	Foreign Key
6	enable	bit	

Table 4.21: Setting database specification

	Attribute name	Datatypes	Constraint
1	title	varchar	
2	gmail	varchar	
3	logo	varchar	
4	tenmien	varchar	
5	sdt	int	

Table 4.22: Category database specification

	Attribute name	Datatypes	Constraint
1	id	bignit	primary key
2	enable	bit	not null
3	name	varchar	
4	link	varchar	

Table 4.23: Voucher user database specification

	Attribute name	Datatypes	Constraint
1	id	bignit	primary key
2	voucher_id	bignit	Foreign Key
3	user_id	bignit	Foreign Key
4	create_at	datetime	

Table 4.24: Shipping database specification

	Attribute name	Datatypes	Constraint
1	id	bignit	primary key
2	name	varchar	
3	priceship	int	

4.3 Application Building

4.3.1 Libraries and Tools

Table 4.25: List of libraries and tools used

Purpose	Tools	URL address
Programming	JDK 11	https://www.oracle.com/java /technologies/downloads
Programming IDE	Visual Studio Code 1.90	https://code.visualstudio.com /download
front-end programming library	NodeJS v20.15	https://nodejs.org/en/download/.
front-end programming library	Angular	https://v17.angular.io/cli
back-end programming library	java spring boot	https://marketplace.visualstudio. com/items?itemName=vmware. vscode-boot-dev-pack
Tool to create local host	Xampp v8.2.12	https://www.apachefriends.org/ download.html
UML design	draw.io	https://app.diagrams.net/

4.3.2 Achievement

After researching, analyzing and implementing, I have built a Restaurant Order and Management Website with the main functions of product management, order management, table reservation management, and order management. User roles are described as follows: For Customers:

- Access to restaurant introduction information:Restaurant overview, Menu items, best sellers,...,Promotions, events.
- Online table reservation
- Ordering and pay dishes
- Access to the restaurant's blog

For Staff:

- View dishes ordered by customers
- Track and update dish status (not prepared, in progress, completed, served)
- Notification when dish status changes to 'completed'
- Notification when customers request assistance or payment
- Post and edit blogs.

For Management:

- Statistics
- Manage dishes
- Manage order
- Manage blogs
- Manage members

Table 4.26: Statistics of application information

Information	Statistical
Number of packages in the source code	27 packages
Total number of application files	540 file
Project capacity	1.06 GB
Application source code capacity	1.5 MB
The total number of lines of code	23,499

4.3.3 Illustration of main functions

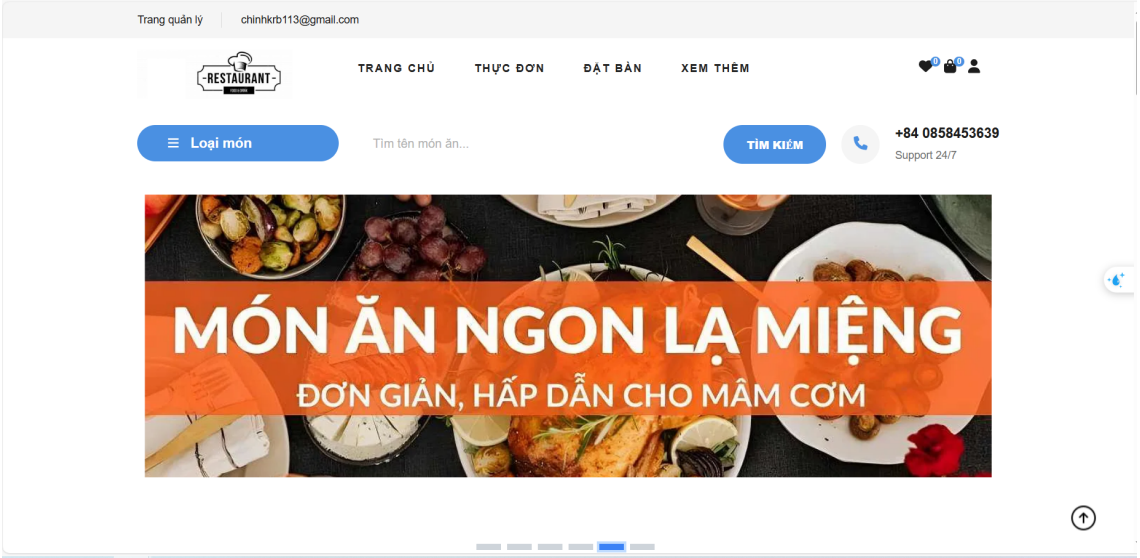

Figure 4.16: Home interface

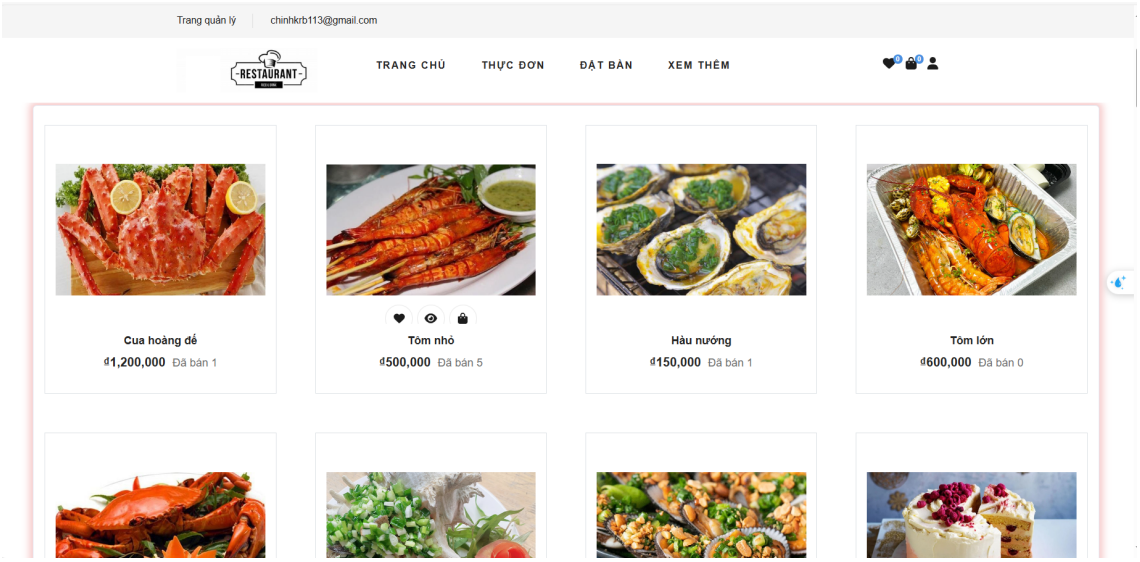

Figure 4.17: Menu interface

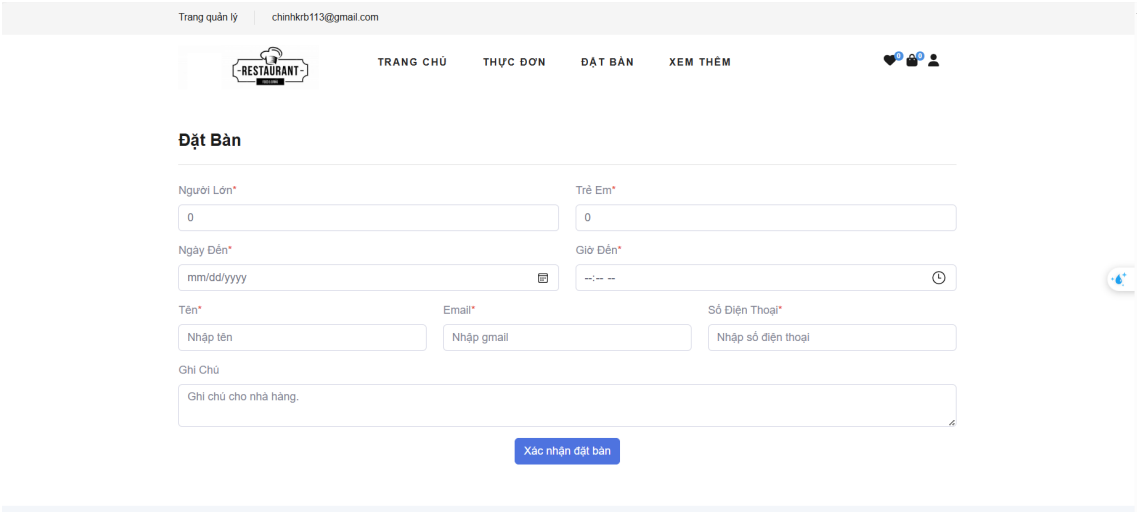

Figure 4.18: Book table interface

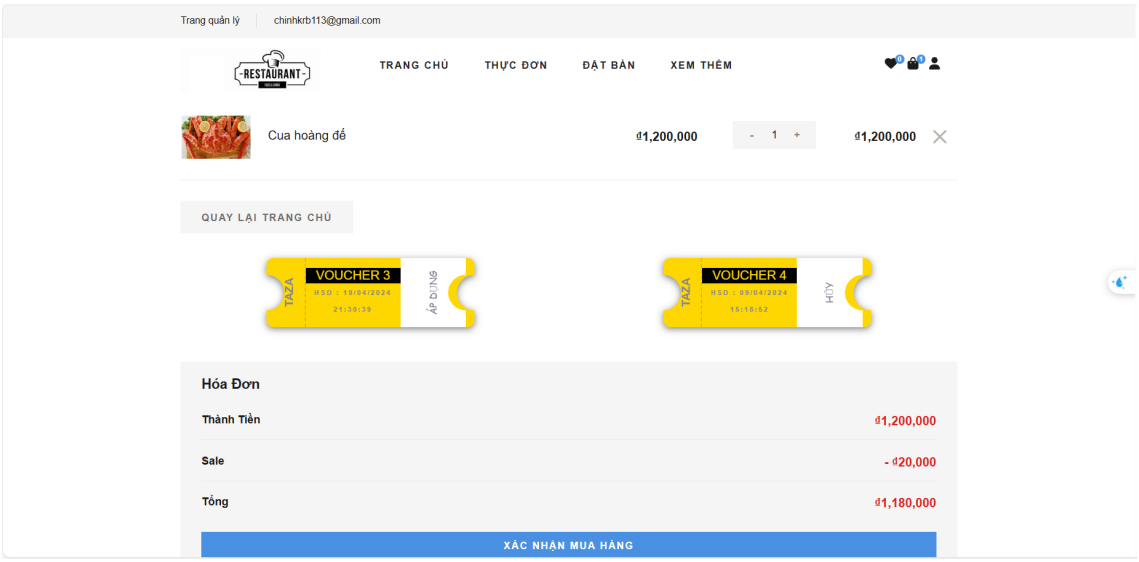

Figure 4.19: Cart interface

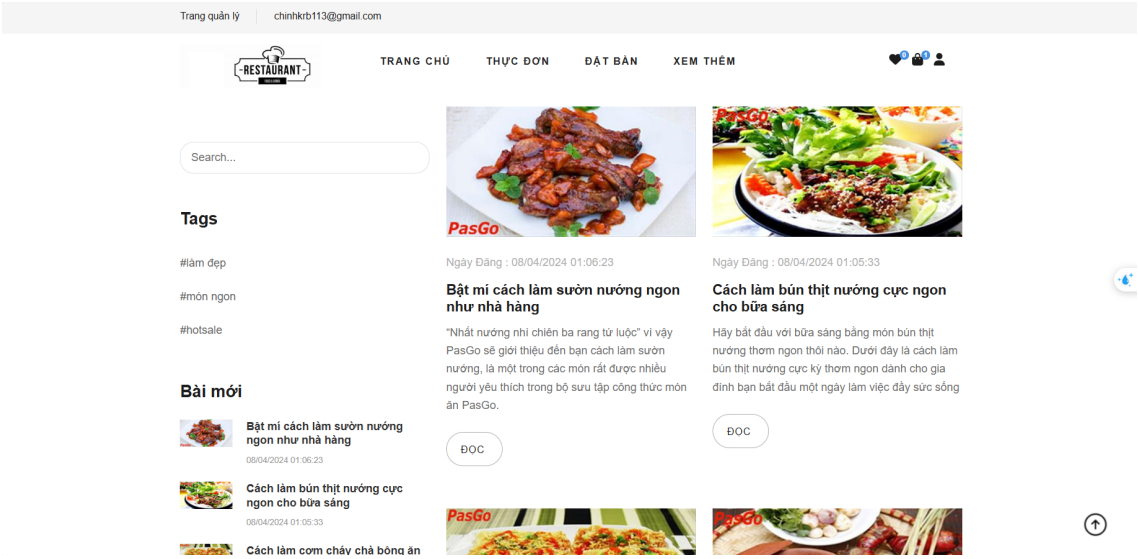

Figure 4.20: Blog interface

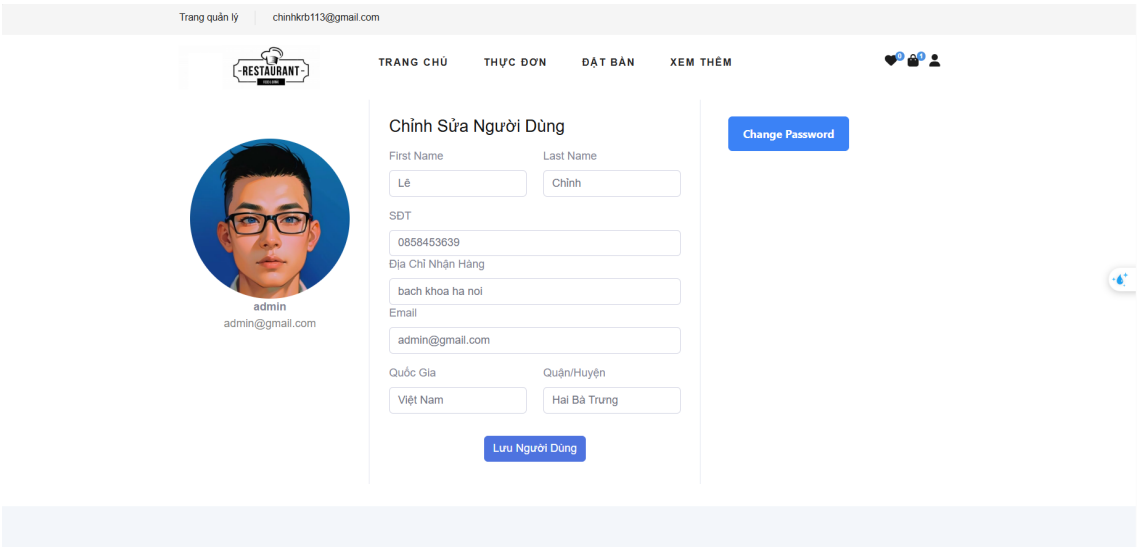

Figure 4.21: User information interface

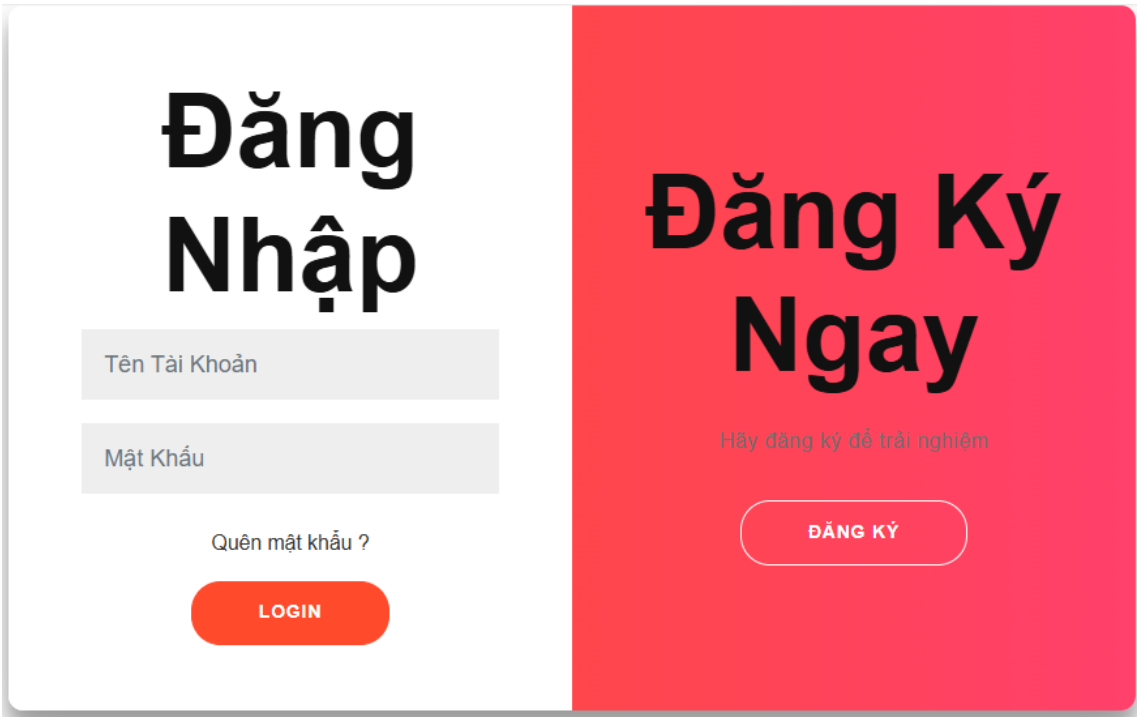

Figure 4.22: Login interface

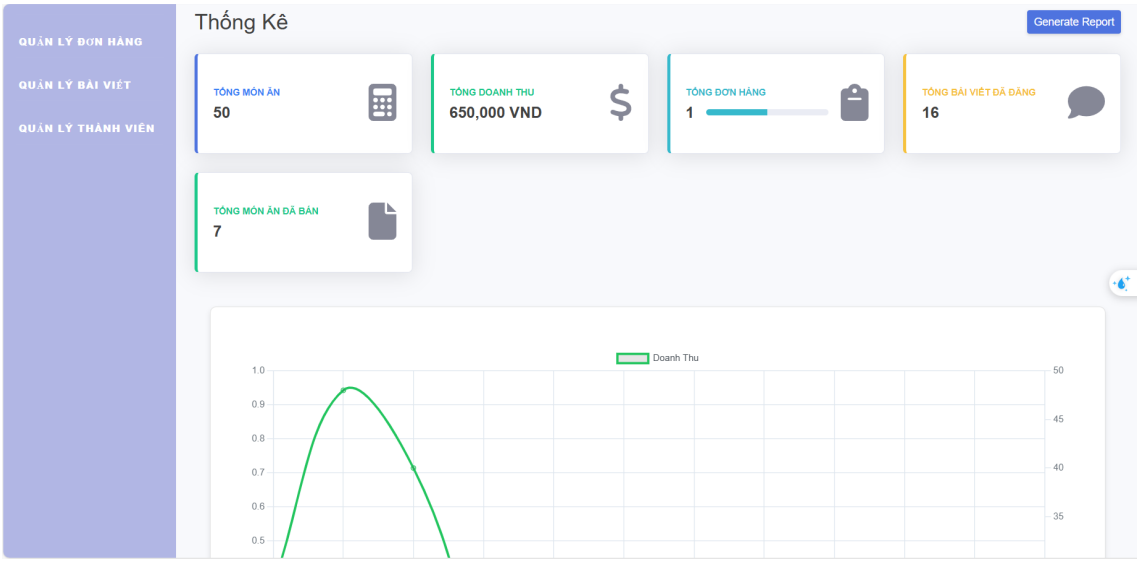

Figure 4.23: Statistical interface

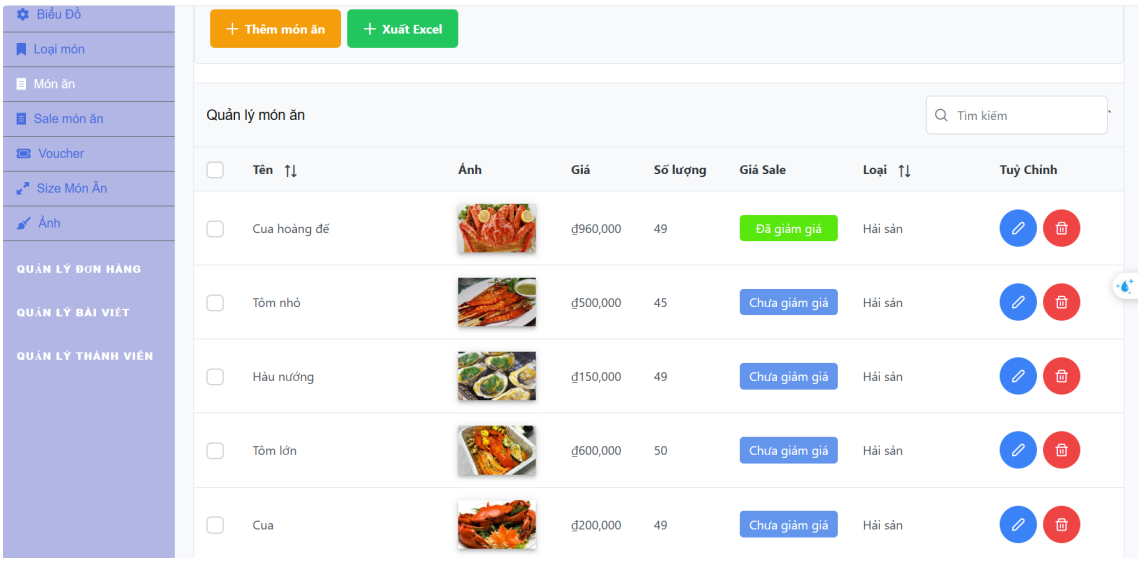

Figure 4.24: Food management interface

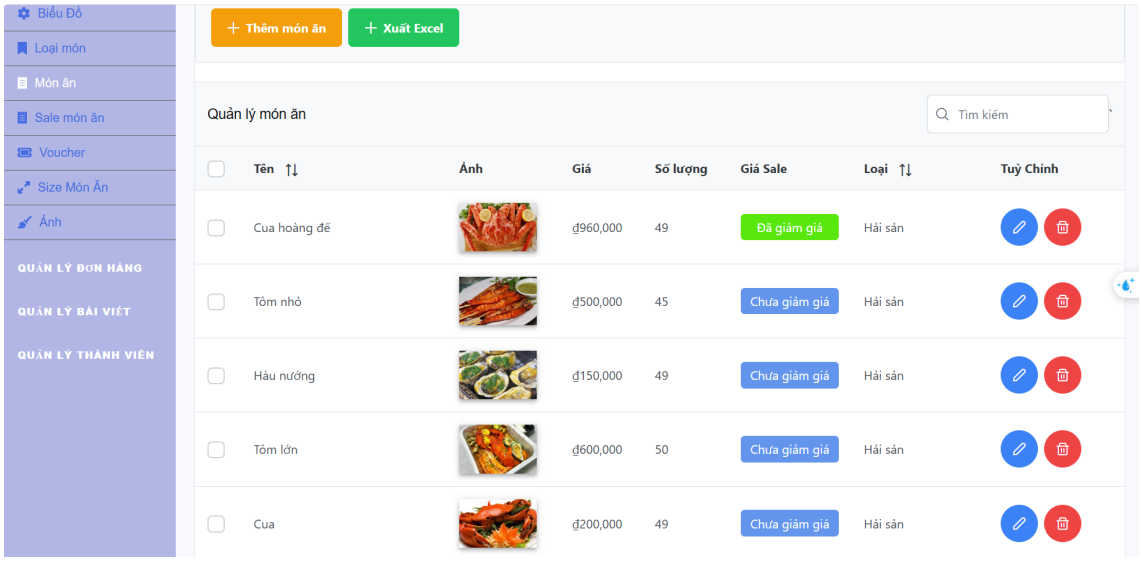

Figure 4.25: Discount interface

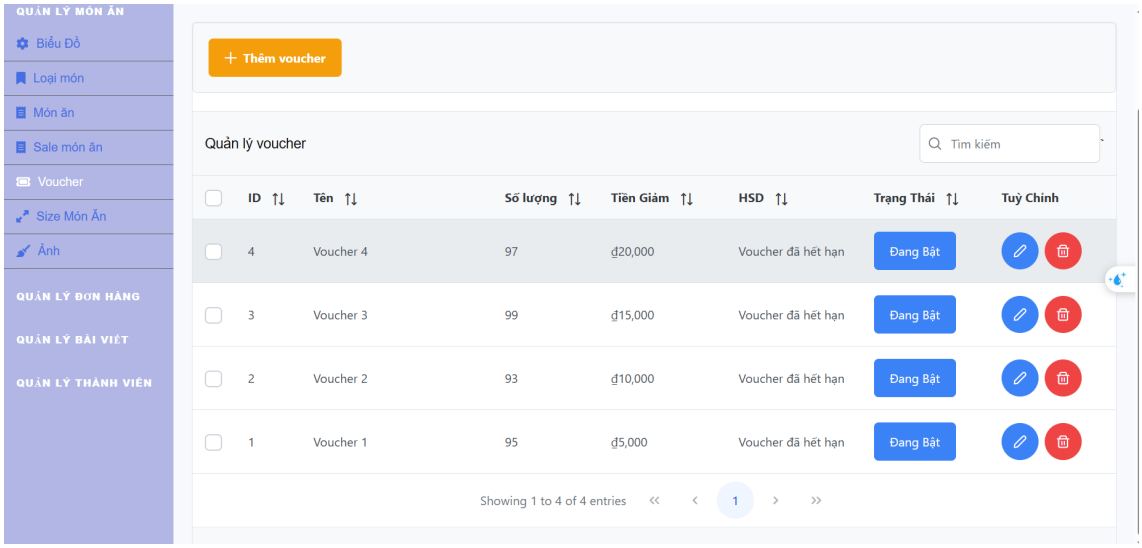

Figure 4.26: Voucher interface

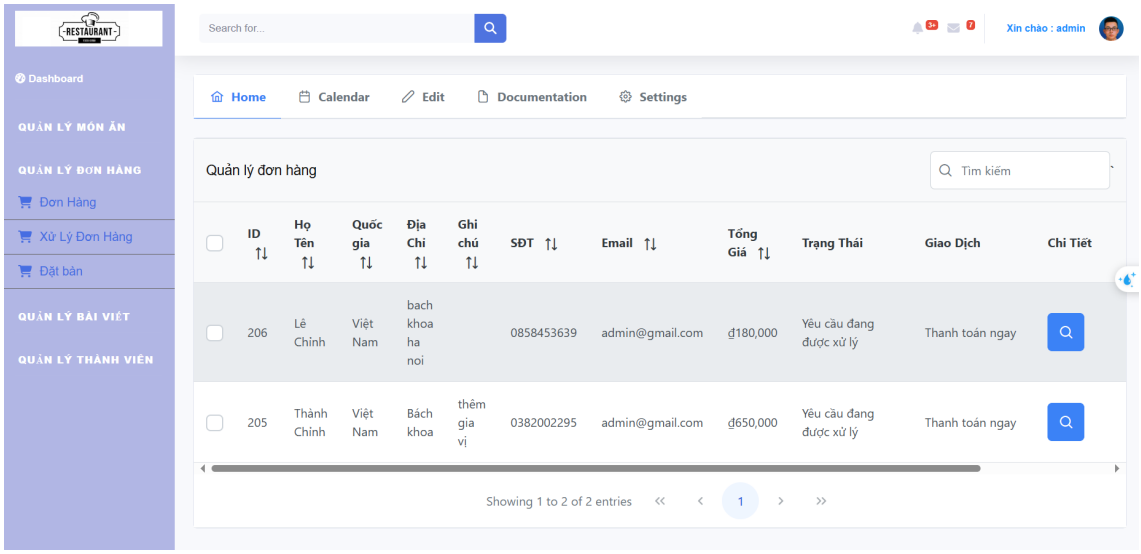

Figure 4.27: Order management interface

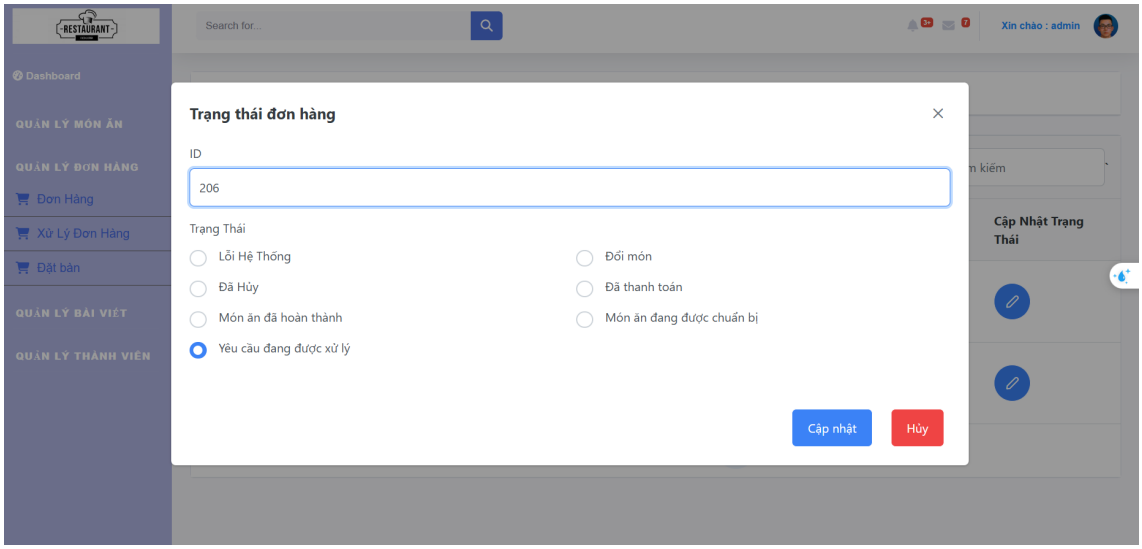

Figure 4.28: Order status interface

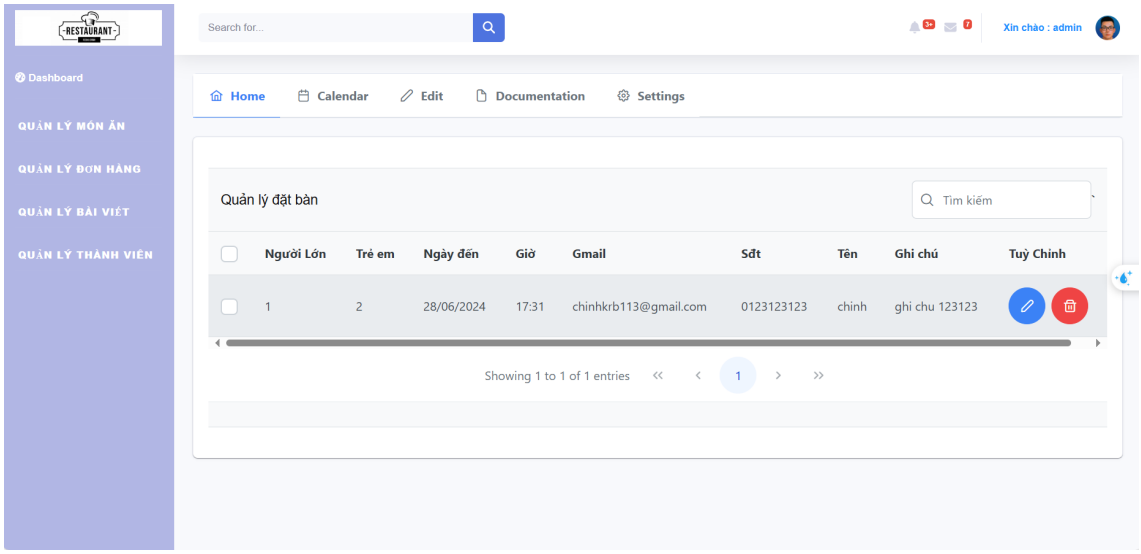

Figure 4.29: Manage table reservations interface

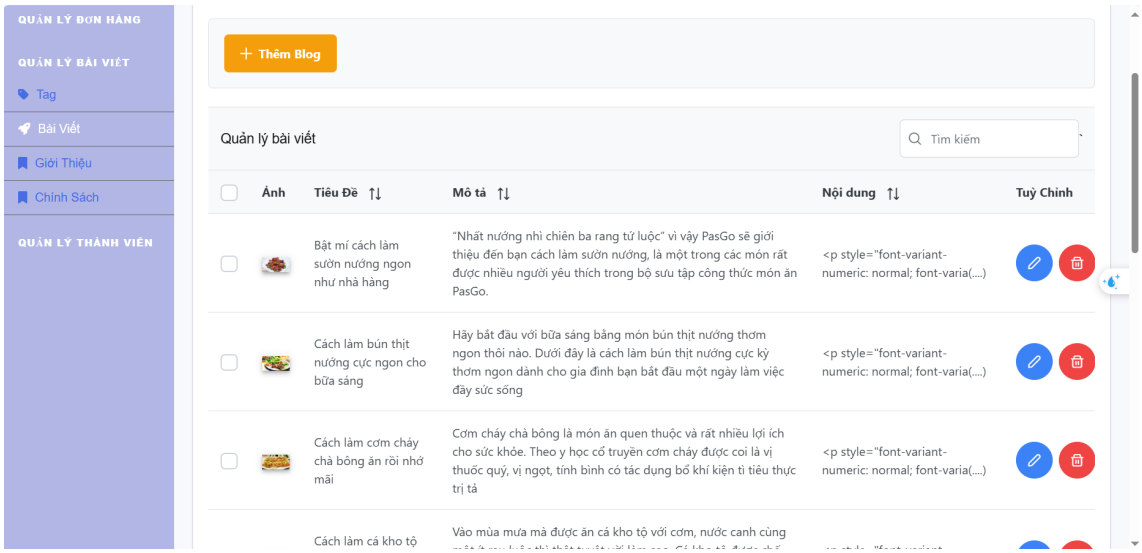

Figure 4.30: Manage blog interface

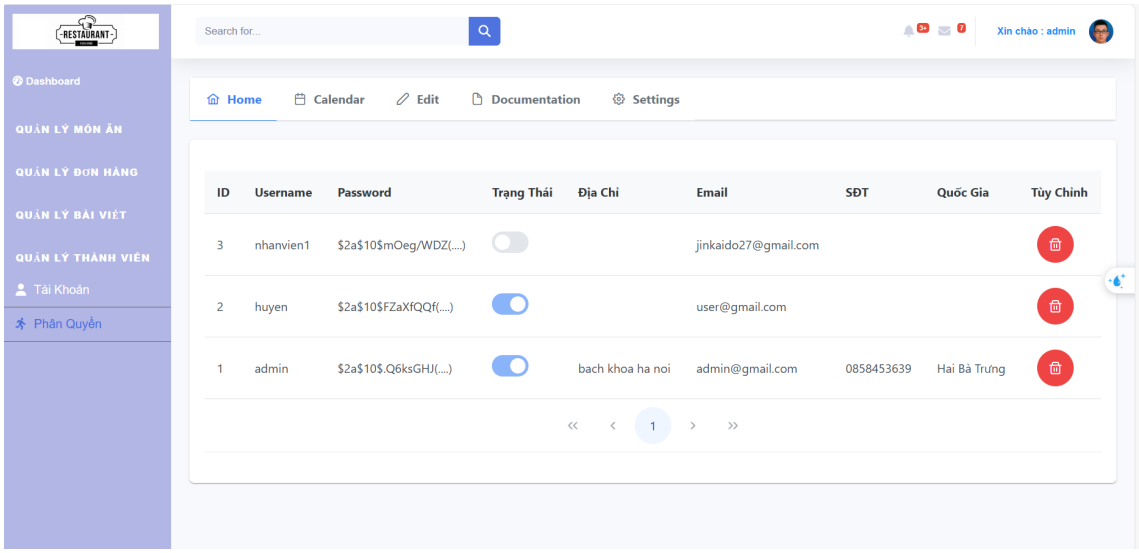

Figure 4.31: Manage user interface

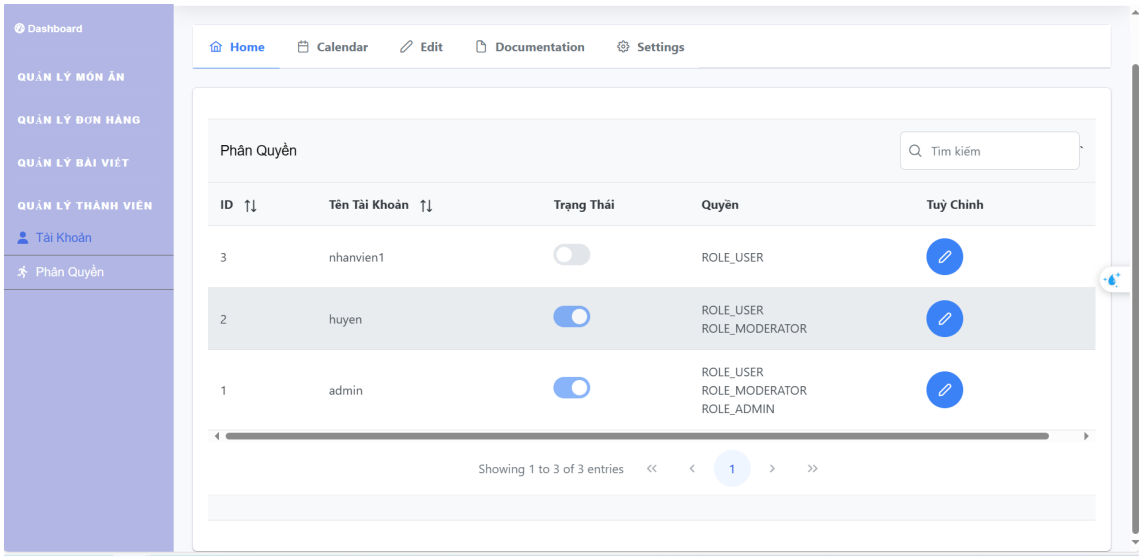

Figure 4.32: User authorization interface

4.4 Testing

4.4.1 Compatibility testing

Table 4.27: Compatibility test statistics table

Device	Specifications	Display	Function	Connect
Dell Ínspiron 7559	Configuration uses Skylake Core i5-6300HQ chip, GeForce GTX 960M discrete graphics with 4GB GDDR5 graphics memory, 8GB DDR3 bus 1,600 MHz RAM and 15.6 inch screen with Full HD resolution	Pass	Pass	Pass

4.4.2 Black box Testing

4.4.2.1 Test the login function

Table 4.28: Test the login function

ID	Test Case Title	Test Case Procedure	Expected Output	Actual Result	Status
1	Enter a valid user and password and check their existence in the database.	1. Open the login page 2. Enter a valid user and password 3. Click the login button	- The log in successfully, you will be redirected to the dashboard page - User and password exist in database	- The log in successfully, you will be redirected to the dashboard page - User and password exist in database	Pass
2	Enter invalid user and password and check if they exist in the database.	1. Open the login page 2. Enter invalid user and password 3. Click the login button	- The login fails, a notification will be displayed - User and password do not exist in the database	- The login fails, a notification will be displayed - User and password do not exist in the database	Pass
3	Enter user and leave password blank	1. Open the login page 2. Enter user and leave password blank 3. Click the login button	- The login fails, a notification will be displayed - User or password is incorrect	- The login fails, a notification will be displayed - User or password is incorrect	Pass
4	Enter password and leave user blank	1. Open the login page 2. Enter password and leave user blank 3. Click the login button	- The login fails, a notification will be displayed - User or password is incorrect	- The login fails, a notification will be displayed - User or password is incorrect	Pass

4.4.2.2 Test product management functions

Table 4.29: Test the function of adding products

ID	Test Case Title	Test Case Procedure	Expected Output	Actual Result	Status
1	Enter product information	1. Open the modal to add products 2. Enter product information 3. Click the submit button	- Display a notification of successful saving - Display the product just added to the list.	- Display a notification of successful saving - Display the product just added to the list.	Pass
2	Do not enter product information	1. Open the modal to add products 2. Leave the information fields blank 3. Click the submit button	- Display unsuccessful save message - Display validation on the product adding modal.	- Display unsuccessful save message - Display validation on the product adding modal.	Pass
3	Enter missing product information	1. Open the modal to add products 2. Enter one missing product information field. 3. Click the submit button	- Display unsuccessful save message - Display validation on the product adding modal.	- Display unsuccessful save message - Display validation on the product adding modal.	Pass

Table 4.30: Test the product repair function

ID	Test Case Title	Test Case Procedure	Expected Output	Actual Result	Status
1	Update product information without leaving fields blank.	1. Open the product update modal page 2. Edit information 3. Click the submit button	- Display notification of successful saving - Display newly updated products on the list.	- Display notification of successful saving - Display newly updated products on the list.	Pass
2	Update product information leaving the information fields blank.	1. Open the product update modal 2. Leave the information fields blank 3. Click the submit button	- Display unsuccessful save message - Display validation on the product adding modal.	- Display unsuccessful save message - Display validation on the product adding modal.	Pass
3	Enter missing product information	1. Open the product update modal 2. Enter one missing product information field. 3. Click the submit button	- Display unsuccessful save message - Display validation on the product adding modal.	- Display unsuccessful save message - Display validation on the product adding modal.	Pass

Table 4.31: Test the product search function

ID	Test Case Title	Test Case Procedure	Expected Output	Actual Result	Status
1	Enter keywords and search requirements.	Search product	- Display the products you are looking for on the list.	- Display the products you are looking for on the list.	Pass

4.5 Deployment

To use the trial version, follow these steps:

- Clone project
- Environment settings, configuration (download the technologies presented in

the table 4.25)

- Create a database through Xampp (the query is stored in the db folder), turn on MySQL mode.

5.1 The interface is easy to use and convenient to use

An easy-to-use and convenient interface is an important factor in website design, helping users have a better experience and easily achieve their goals. Besides, the easy-to-use and convenient interface not only helps users perform tasks more effectively but also increases satisfaction levels and reduces discomfort during product use.

The interface is designed according to the principles:

Simplicity: This principle emphasizes simplifying the user interface and experience to make the product easy to use and understand:

- Eliminate unnecessary elements, focus on the main function.
- Use a clear, easy-to-understand layout.
- Simplify the operation process.

Consistency: Ensure that the user interface and experience are maintained consistently across the entire product:

- Use consistent colors, fonts, and layout throughout the entire website/application.
- Maintain logic in naming and arranging functions.
- Create a seamless experience for users.

Clarity: The design needs to be clear and easy to understand so that users can learn and use the product easily:

- Use easy-to-read fonts, contrasting colors, and clear icons. Clearly define function points and tasks for easy user interaction.

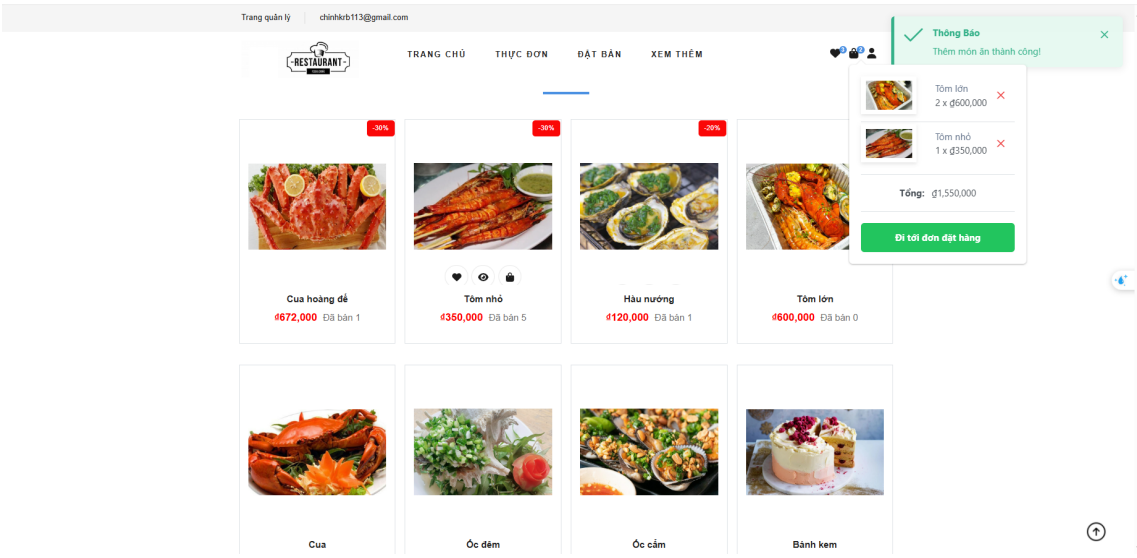

Figure 5.1: Menu interface

Website uses a simple, straightforward, easy-to-use interface, Website will help users have no difficulty the first time using it.

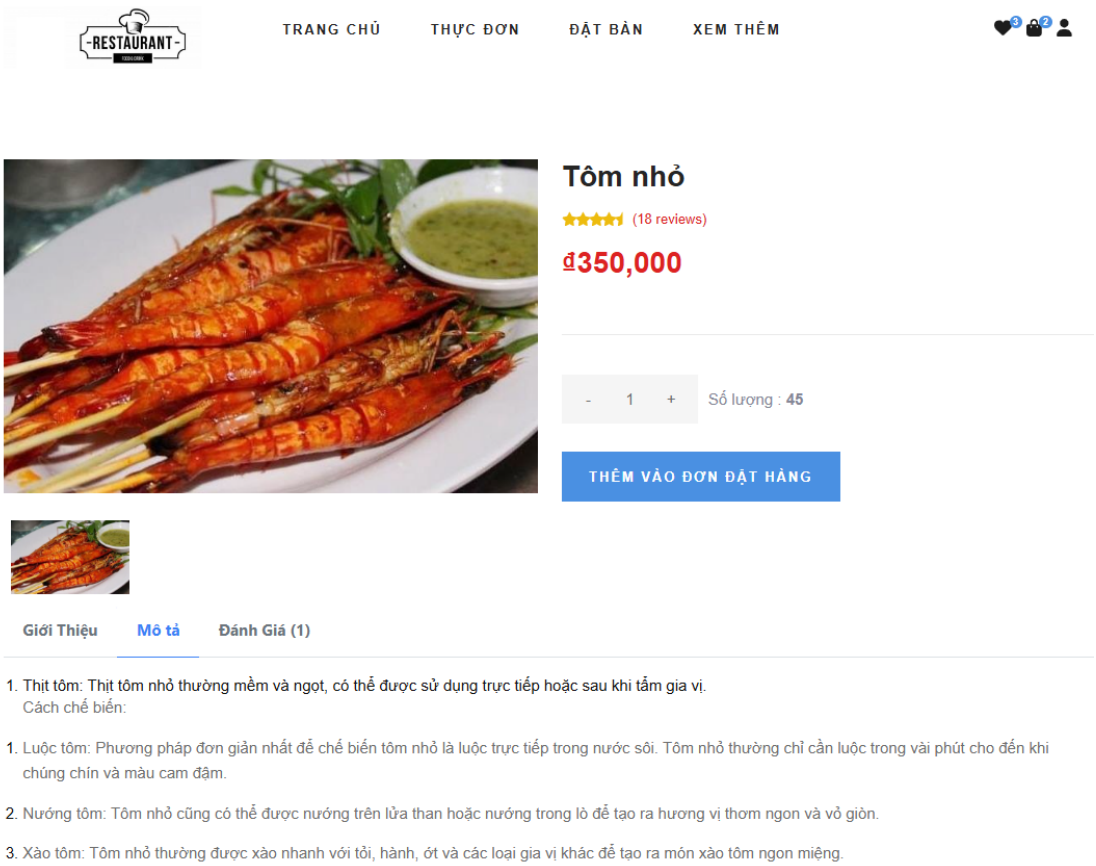

Figure 5.2: Detailed dish interface

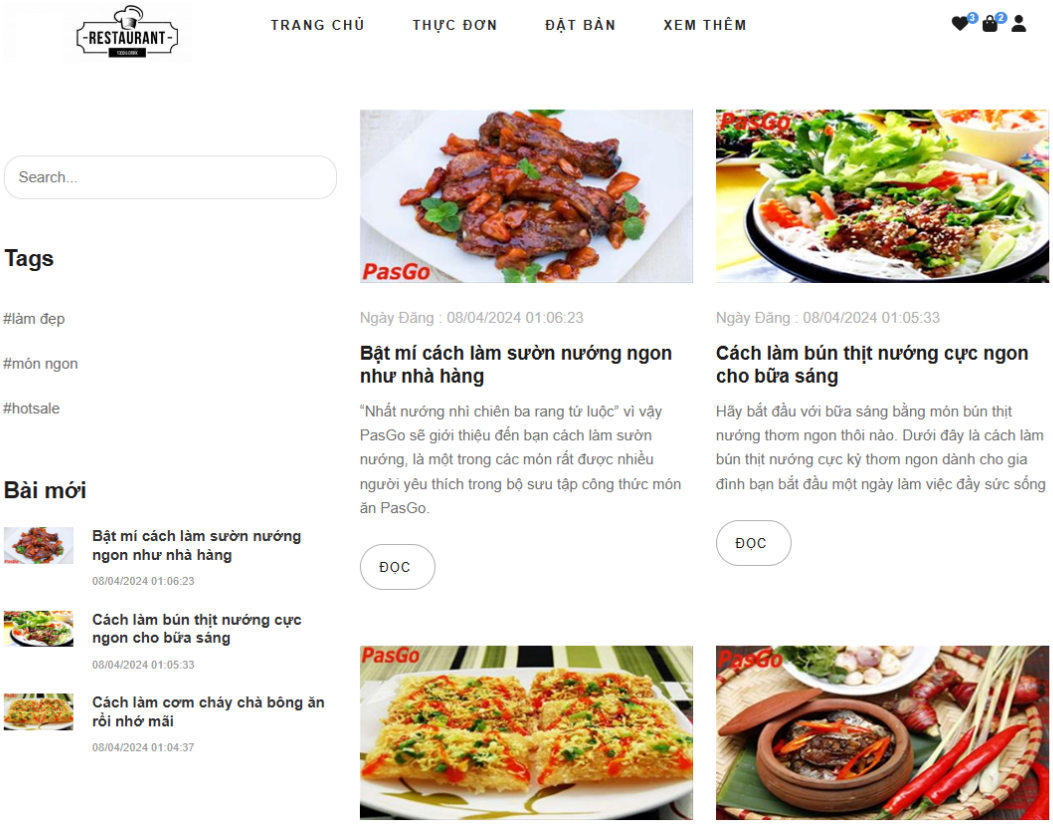

Figure 5.3: Blog viewing interface

The content is presented logically, easy to read, and easy to remember

Figure 5.1, Figure 5.2 and Figure 5.3 Describes interfaces that bring convenience and friendly to users. Functions and content are displayed clearly and specifically, without overlapping other content.

5.2 Secure sensitive data

5.2.1 Problem

In an increasingly digital world, with strong developments in technology fields, sensitive data security has become an important and challenging issue. Sensitive data can include personal information, financial data, and other types of important information. Protecting this data is important not only for individuals but also for organizations and businesses. Once this data is stolen or illegally disclosed, it can cause enormous damage to the business. Addressing the issue of sensitive data security is an important and necessary factor.

Risks and consequences of sensitive data breaches: The risks of sensitive data breaches are increasing with the rise of cyberattacks, from the theft of personal information to attacks on organizational systems. The consequences of these breaches can be severe, including financial loss, reputational damage and violation

of individual privacy rights.

Factors that contribute to security threats: Many factors contribute to threats to sensitive data:

Cyber attacks: Hackers use sophisticated techniques to infiltrate and steal data. Human error: Employees may accidentally expose information via email, USB drives, or other means. Malware: Malware, ransomware, and other malicious software can damage and steal data.

Security measures To protect sensitive data, a comprehensive security strategy is necessary, including:

- **Data encryption:** Use encryption methods to protect data during transmission and storage.
- **Strong authentication**: Apply multi-factor authentication to ensure only authorized people can access data.
- **Employee training:** Improve employee security awareness and skills to minimize human error.
- **Monitoring and early detection:** Use monitoring and analytics tools to detect early signs of attacks and respond promptly.

Conclude Securing sensitive data is a complex issue that requires special attention from both individuals and organizations. Implementing effective security measures and complying with legal regulations will help minimize risks and protect important information in the best possible way. Nowadays, more and more modern technologies are being developed, ensuring the security of sensitive data is not just an option but a necessary requirement.

5.2.2 Solutions and results

Use JWT to:

- **Authentication:** This is the most common case where JWT is used. Once the user has logged into the system, subsequent requests from the user will contain additional JWT code. This allows users to be granted access to the urls, services, and resources that the Token allows. This method is not affected by Cross-Origin Resource Sharing (CORS) because it does not use cookies.
- **Information exchange:** JSON Web Token is a pretty good way to securely transmit information between members, thanks to its signature. The recipient can know who the sender is through the signature. And the signature is created by combining the header and payload, so through that we can confirm whether

the signature has been forged or not.

Use Spring Security to: encrypt passwords (which basically means don't store passwords in clear data).

5.2.2.1 JWT

JSON Web Token (JWT) is an open standard (RFC 7519) that defines how to securely communicate between members using a JSON object. This information can be authenticated and trusted thanks to its "signature". The signature portion of the JWT will be re-encrypted using HMAC or RSA.[5]

Below is a JSON Web Token: [1]

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiJuaHMzMTA4IiwiZXhwIjoxNTU4MDYzODM3fQ.
449KVmOFWcpOUjnYGm-f1QWhY8N-DerKDfTK0JQm1Nc
```

Figure 5.4: JSON Web Token

The above JWT includes 3 parts:

- Header (eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9)
- Payload (eyJzdWIiOiJuaHMzMTA4IiwiZXhwIjoxNTU4MDYzODM3fQ)
- Signature (449KVmOFWcpOUjnYGm-f1QWhY8N-DerKDfTK0JQm1Nc)

separated by "." (dot).

It looks complicated, but its structure follows the following format:

<base64-encoded **header**>.<base64-encoded **payload**>.<HMACSHA256(base64-encoded **signature**)>

[1] https://viblo.asia/p/jwt-tu-co-ban-den-chi-tiet-LzD5dXwe5jY

H

```
String header = "{\"alg\":\"HS256\",\"typ\":\"JWT\"}";
System.out.println(Base64.getUrlEncoder().encodeToString(header.getBytes()));
```

Figure 5.7: Encoding Header

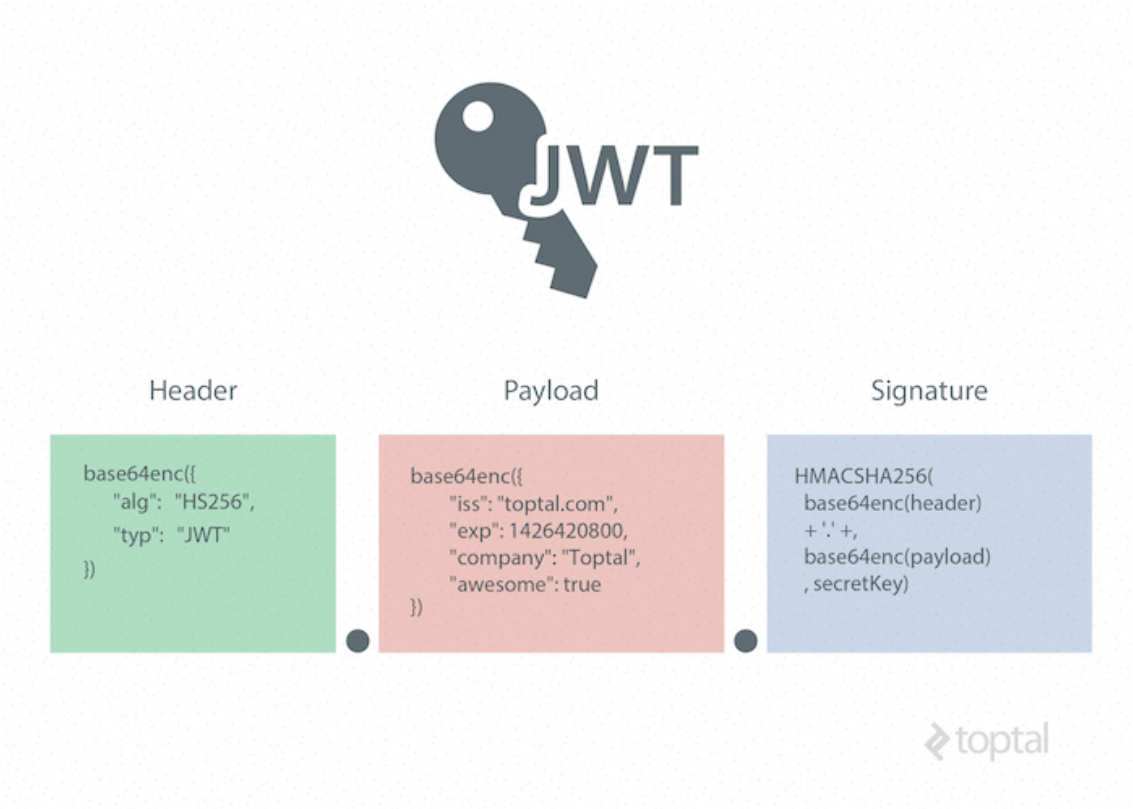

Figure 5.5: Format JWT

Header includes two main parts:

- **typ** - Token type (default is JWT - This information indicates that this is a JWT Token)
- **alg** - Algorithm used for encryption (HMAC SHA256 - HS256 or RSA).

For example:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Figure 5.6: Example Header JWT

The above JSON fragment after base64url encoding will become as follows:

Output: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

The second part of the token is Payload, which contains the content of the information (claim). The transmitted information can be a description of an entity (for example, a user) or it can also be additional information for the Header. In general, they are divided into 3 types: reserved, public and private.

Reserved: is the information specified in the IANA JSON Web Token Claims registry. None of this information is mandatory. However, depending on each application you implement, please bind the mandatory requirements to the necessary information

- **iss (issuer):** token issuer (optional)
- **subject**sub (): topic of the token (optional)
- **aud (audience):** token user (optional)
- **exp (expired time):** time the token will expire (optional)
- **nbf (not before time):** token will not be valid before this time
- **iat (issued at):** the time the token was issued, calculated according to UNIX time
- **jti:** JWT ID

Public: The key can be defined according to the wishes of the JWT user. However, to avoid duplication, tokens should be specified in the IANA JSON Web Token Registry or a URI containing an unduplicated namespace.

Private: Additional information is used to pass between member computers

For example:

```
{
  "sub": "nhs3108",
  "exp": 1558065429
}
```

Figure 5.8: Example Playload JWT

The above JSON fragment after base64url encoding will become as follows:

```
String payload = "{"sub":"nhs3108","exp":1558063837}";
System.out.println(Base64.getUrlEncoder().encodeToString(payload.getBytes()));
```

Figure 5.9: Encoding Payload

Output: eyJzdWIiOiJuaHMzMTA4IiwiZXhwIjoxNTU4MDYzODM3fQ

The signature is created by combining the Header + Payload parts, then encoding it with a certain encoding algorithm, the more complex the better, for example HMAC SHA-256

With signature being the combination of header and payload In the above two parts, we have:

```
String header = "{\"alg\":\"HS256\",\"typ\":\"JWT\"}";
String encodedHeader = Base64.getUrlEncoder().encodeToString(header.getBytes());

String payload = "{"sub":"nhs3108","exp":1558063837}";
String encodedPayload = Base64.getUrlEncoder().encodeToString(payload.getBytes());

String signature = encodedHeader + "." + encodedPayload;

String encodedSignature = HMACSHA256.encode(signature, scretKey);

System.out.println(encodedSignature);
```

Figure 5.10: Signature JWT

Output: 449KVmOFWcpOUjnYGm-f1QWhY8N-DerKDfTK0JQm1Nc

Summarizing the 3 parts, we have the following JWT string:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJuaHMzMTA4IiwiZXhwIjoxNTU4M
f1QWhY8N-DerKDfTK0JQm1Nc

JWT processing:

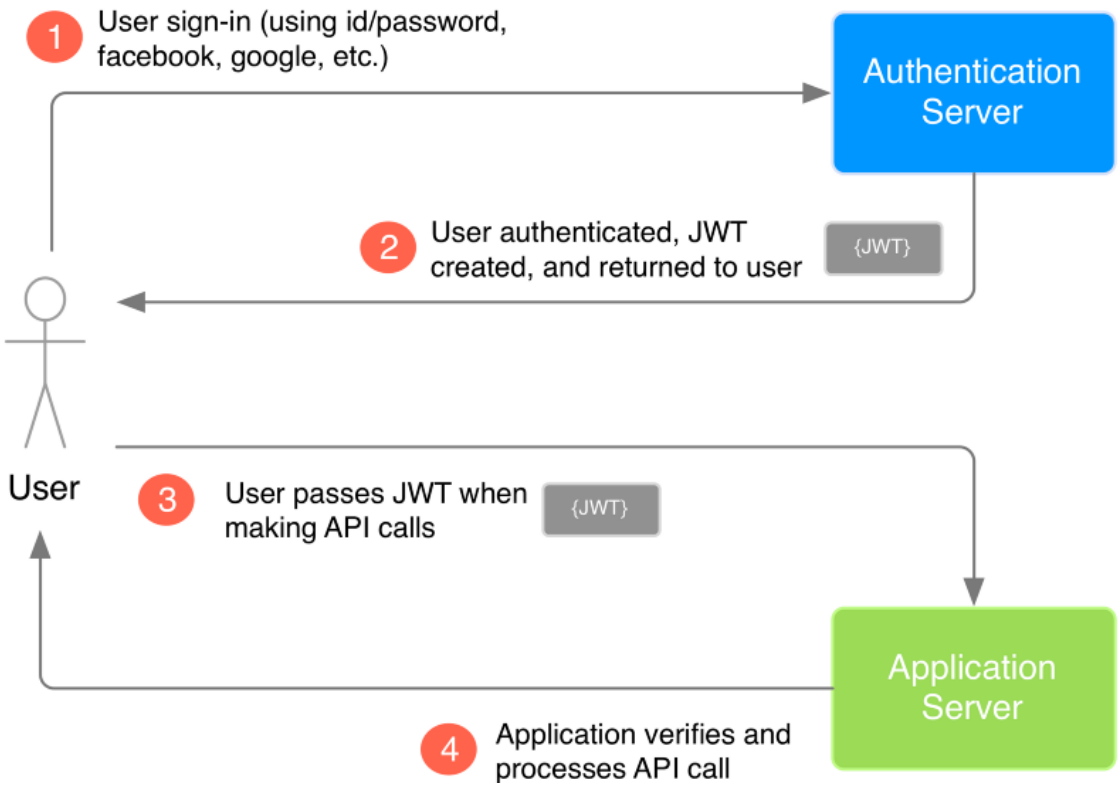

Figure 5.11: JWT processing

Looking at the diagram, we can see the flow as follows:

- User performs login by sending id/password or using social network accounts to the Authentication Server (Authentication Server)
- Authentication Server receives data that the User sends to serve user authentication. In case of success, the Authentication Server will create a JWT and return it to the user via response.
- The user receives the JWT returned by the Authentication Server as a "key" to execute further "commands" to the Application Server.
- Application Server, before making a request from the User, will verify the JWT sent. If OK, continue executing the called request.

5.2.2.2 Spring Security

To understand Bcrypt, we need to learn about "hashing", "streching" and "salting".

Hashing:

Hashing is a technique used to convert data from one format to another, usually a fixed sequence of characters (called a hash). Hashing is often used in many fields of information technology, especially in security and cryptography. Here are some

important points about hashing:[6]

Characteristics of Hashing:

- **One-way conversion:** One of the main characteristics of hashing is that the process is one-way. This means that from the original data (also called input or plaintext), we can create a hash value, but we cannot conversely find the original data from that hash value.
- **Fixed size:** Regardless of the size of the original data, the generated hash value always has a fixed size. For example, the SHA-256 hash function always produces a string of 256 bits (32 bytes).
- **Data Integrity:** Hashing is used to check data integrity. If data is changed even slightly, its hash value will be completely different. This helps detect data alteration or tampering.
- **Processing speed:** Hashing is typically very fast, which makes it an effective method for processing and verifying data.

For example:

"**HelloWorld**" after hashing using the md5 hash algorithm will produce results similar to "**68e109f0f40ca72a15e05cc22786f8e6**".

With current computer technology, decoding the hash function is impossible. However, if we are not careful, there are still ways a counterfeiter can get his hands on the password if he gets his hands on the hash code. Two common attack methods are:

- **Rainbow table attacks:** attacks using a pre-computed password dictionary.
- **Brute Force attacks:** attack every possible situation.

To solve this problem, we will learn about Salting and Stretching.

Salting:

Salting is another layer of security added to the hash to make it more secure, unique strings of characters are added to the password before hashing to change the hash completely.

Characteristics of Salting:

- **Add a random value:** add a random value, called "salt", to the password before it is hashed. Each password will have a different salt, making the hash value of the same password different each time.
- **Salt is stored along with the hash:** After creating the value both the hash

and salt values are stored in the database. When it is necessary to verify the password, the salt will be used again to recalculate the hash value from the user's password.

- **Increased complexity:** Adding salt increases the complexity of hash values, making it impractical to create rainbow tables (tables containing precomputed hashes from common passwords), because Each hash value in the rainbow table will only work with a specific salt.

Stretching:

Stretching is a technique in password security that increases the time needed to calculate the hash value of a password. This makes brute-force attacks (trial-and-error attacks) and other attacks more difficult because each password check takes more time. Stretching is usually done by repeating the hashing process many times.

Characteristics of Stretching:

- **Repeatedly:** Stretching involves applying a hash function to the result of a previous hash multiple times. The number of iterations is chosen so that the computation becomes slow but does not cause undue difficulty for legitimate users.
- **Increase computation time:** The main goal of stretching is to increase the computation time of each hash value, making brute-force attacks much slower.
- **Use strong hash functions:** Stretching often uses strong cryptographic hash functions such as SHA-256 or SHA-512, or special hash functions designed for password security such as PBKDF2, bcrypt, or scrypt.

Bcrypt:

Bcrypt is a password encryption function designed by Niels Provos and David Mazières based on the Blowfish encryption algorithm and first introduced at USENIX in 1999. It is a combination of all three concepts we just discussed. understand "hashing" "streching" and "salting".

Normally a bcrypt hashing sequence will look like this:

$[algorithm]$[cost]$[22 characters salt][31 characters hash]

For example: "123456" using Bcrypt will give the result:

$2a$10$.Q6ksGHJEJkctPv98RxI4.gMiJBAwNm6jB6SCXYVL7dKbpO3kjpHC

In there:

- **2a**: hash algorithm

- **10**: cost (2Î0 or 1024 iterations)
- **.Q6ksGHJEJkctPv98RxI4.**: 16-byte (128-bit) salt base64 encoded to 22 characters.
- **gMiJBAwNm6jB6SCXYVL7dKbpO3kjpHC**: 24-byte (192-bit) hash code base64 encoded to 31 characters.

So with the hash code as above, our password will run a total of 2 to the power of 10 times (1024 loops) using the salt **.Q6ksGHJEJkctPv98RxI4.**, the password "**123456**" combined with the blowfish encryption algorithm **$2a$** to give us the result.

We test hashing 123456 twice in a short period of time, then we get the following results:

```
Băm sử dụng Bcrypt của chuỗi 123456 lần 1 là:
$2a$10$.Q6ksGHJEJkctPv98RxI4.gMiJBAwNm6jB6SCXYVL7dKbpO3kjpHC

Băm sử dụng Bcrypt của chuỗi 123456 lần 2 là:
$2a$10$uw.SfEGLEPT1WnGnADshZ.QwE4MrYLtxmjfcb9ap3phX5od0l5A7q
```

Figure 5.12: Hash 123456 twice in a row

Figure 5.12 shows us that with the same plaintext, hashing twice consecutively or any hashing time will give us different ciphertexts. Therefore, it will be difficult for attackers to crack this ciphertext, improving the safety of storing data in the database.

After successfully hashing the string 123456, the software uses a method to check the plaintext with the ciphertext and does not hash it again. Result:

```
Kiểm tra 123456:

$2a$10$.Q6ksGHJEJkctPv98RxI4.gMiJBAwNm6jB6SCXYVL7dKbpO3kjpHC
True
$2a$10$uw.SfEGLEPT1WnGnADshZ.QwE4MrYLtxmjfcb9ap3phX5od0l5A7q
True

Kiểm tra 111111:
False
```

Figure 5.13: The result matches 123456 and a strange plaintext 111111

As a result, we protect our users' data

Figure 5.14: Password field in the database

Figure 5.14 shows that the password is encrypted in the database even though the password may be the same.

5.3 Send email

5.3.1 Problem

In the modern business context, especially in the food service (F&B) industry, updating order status is not only part of the management process but also a key factor in customer care. client. This ensures that customers are always informed about the status of their order, from the time the order is placed until the product is received. This not only helps improve customer experience but also minimizes issues related to customer management and support. This essay will analyze the important reasons for updating restaurant order statuses.

- **Provide Transparent and Timely Information** Providing transparent and timely information helps customers track the processing and delivery process easily. This creates trust and satisfaction from customers, showing that the restaurant pays attention to their needs and expectations.
- **Minimize Customer Anxiety** Notification about the stages of the order (preparation, delivery, completion) helps customers feel more secure. This peace of mind not only creates a positive shopping experience but also increases customer trust and loyalty.
- **Improve Customer Experience** Enhance customer experience by continuously and accurately updating order status. This shows professionalism and thoughtfulness on the part of the restaurant, encouraging customers to come back and use the service more times.
- **Minimize Complaints and Support Customers** Complete information about order status reduces the need to contact customer service. This reduces the load on the support department, allowing them to focus on more important matters, while creating a positive impression of the restaurant.

Besides,There are also some cases where restaurants need to notify customers

such as: When users register an account on the website, authentication measures need to be taken to avoid bad guys attacking by registering fake or spam accounts.Other case, Notify customers when they reserve a table but the table is sold out.

5.3.2 Solutions and results

Introducing Spring Email

Concept: Spring Email is a library in Spring Framework, providing developers with support classes and tools to create and send emails in Spring Boot Java applications. This library uses SMTP and MIME protocols to create and send emails, allowing us to attach attachments, customize email content, and send emails to multiple recipients at once.[2]

Here are the main advantages of Spring Boot Email:

- **Easy integration and configuration:** Spring Boot provides default configurations for sending emails through application.properties or application.yml files, making configuration simple and flexible.
- **Supports many types of emails:** Spring Boot supports sending simple emails, emails with file attachments, and emails with HTML content, meeting the application's rich email sending needs.
- **Using JavaMailSender:** Spring Boot uses JavaMailSender to send emails, this is an interface that helps interact with email servers easily, providing methods to set up and send emails effectively.
- **Integrates well with Spring Framework**: Spring Boot is a part of Spring Framework, so it integrates well with other Spring features and components, such as Dependency Injection, Spring MVC, and Spring Data, making application development a breeze. more effective way.
- **Strong Community Support:** Spring Boot is one of the most popular frameworks in the Java community, thus has strong support from the community, ensuring continuous support and updates for issues and features. email-related functionality.
- **Security and reliability:** Spring Boot supports security mechanisms such as TLS and SSL to protect email sending, ensuring safety and reliability when transmitting data over the network.

[2]https://freetuts.net/spring-email-trong-spring-boot-5796.html

Figure 5.15: Order cancellation email

Figure 5.16: Order delivered successfully email

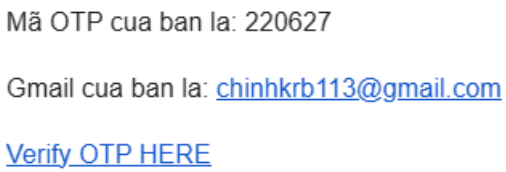

Figure 5.17: Email otp

Figure 5.18: Cancel reservation email

Figure 5.15, Figure 5.16, Figure 5.17, Figure 5.18 describes the content of the email the user receives

5.4 Statistical

5.4.1 Problem

In restaurant management, using statistics not only helps improve operational efficiency but also brings many different benefits. Statistics provide an overview of business activities, thereby helping managers make accurate and timely decisions.

- **Raw Materials and Inventory Management** Statistics help track daily, weekly, and monthly material usage, thereby optimizing ordering and minimizing waste. Effective inventory management helps ensure that there are always enough ingredients for dishes without overages or shortages.
- **Analysis of Revenue and Profit** Collecting and analyzing daily, weekly, and monthly sales data helps managers better understand business performance. This allows them to identify best-selling dishes, peak hours, and high-sales days, thereby flexibly adjusting business strategies.
- **Understand Customer Behavior** Statistics provide information about customer behavior and preferences, including favorite dishes, frequency of restaurant visits, and average spending. This information helps restaurants customize menus and marketing strategies to fit customer needs.
- **Employee Performance Evaluation** Statistics can be used to track employee performance, including number of orders processed, service time, and customer satisfaction. From there, managers can detect problems and take measures to improve employee performance.
- **Forecasting and Planning** Statistics help forecast future business trends based on historical data. This assists managers in planning marketing campaigns, promotions, and adjusting business strategies to meet market needs.

5.4.2 Solutions and results

My restaurant management website allows statistics on the first page of the management page. In addition, the website also allows exporting excel files for managers to easily manage and make flexible statistics

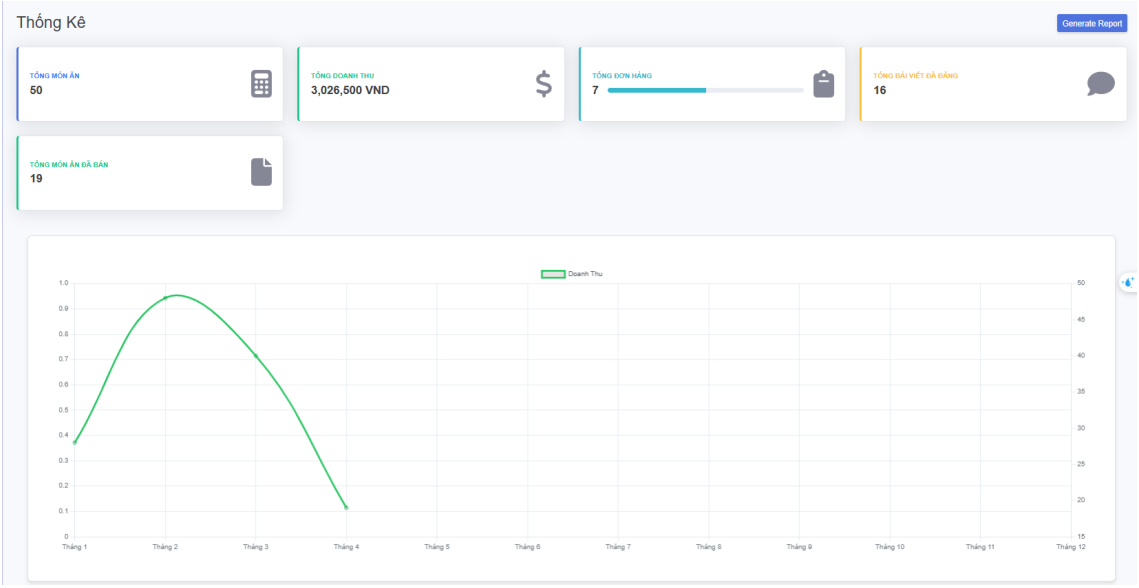

Figure 5.19: Interface of the statistics page

Figure 5.19 describes the interface of the statistics page

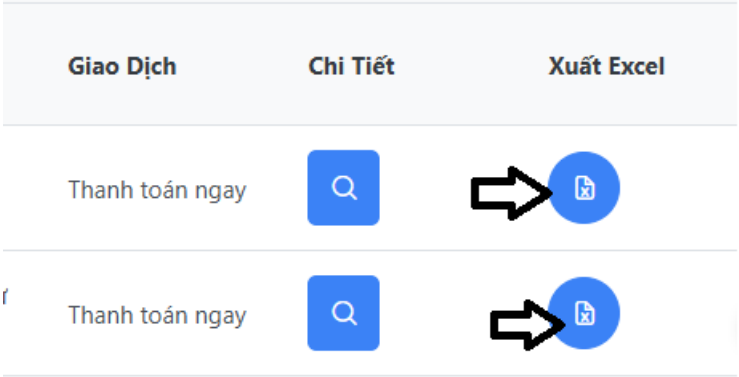

Figure 5.20: Statistics using excel

	A	B	C	D	E
1	ID	Name	Price	Quantity	Subtotal
2	338	Cua	200000	1	200000
3	337	Ốc đêm	50000	1	50000
4	336	Bún Thái	50000	1	50000
5	335	Chè Bưởi	1500	1	1500
6	334	Kem Bơ	10000	1	10000
7	333	Ốc cẩm	100000	1	100000
8	332	Ốc đêm	50000	1	50000
9	331	Cua	200000	1	200000
10	330	Tôm nhỏ	350000	1	350000
11	329	Tôm lớn	600000	2	1200000
12	328	Cua	200000	1	200000
13	327	Tôm nhỏ	500000	1	500000
14	326	Hàu nướng	150000	1	150000

Figure 5.21: Statistics using excel

Figure 5.20 and Figure 5.21 describing statistical functions in Excel

CHAPTER 6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

After the process of researching, analyzing and building with the dedicated guidance of ThS. Le Ba Vui, I have built the Restaurant Order and Management Website. Restaurant websites have very diverse and rich functions, including features for customers, employees and management. This helps it outperform many other products on the market in the same field. For example, online table reservation and online food ordering and payment features increase convenience and customer experience. Management features for chefs and wait staff also provide greater efficiency and interoperability between parts of the restaurant. Compared to other products, this website provides a comprehensive platform to manage restaurant operations from ordering to data management and statistics. Here I would like to summarize what the website has done:

- **Provide information about restaurants to customers:** Restaurant overview, menu list with featured dishes and detailed information about each dish. Promotions, special events, restaurant activities.
- **Book a table and order online:** Customers can easily book a table online. Order and pay for food online to save waiting time at the restaurant.
- **Restaurant blog:** Articles about dishes, events and other restaurant-related information. Ability to create, edit, and manage blog posts.
- **Management for employees:** View and update information about dishes ordered by customers. Track and update the status of dishes. Food management, table reservation management, blog management.
- **Management and statistics for management:** Revenue statistics. Manage information about restaurant dishes and members. Manage and update content on the restaurant's blog.

During the process of developing the restaurant website, I have achieved some significant achievements such as successfully implementing online table booking and ordering features, helping to improve operational efficiency and customer experience. Besides, the website also provides management functions for employees and managers. However, building software in a short time will still have many limitations and shortcomings. The software continues to receive reviews from users to build and develop into a more complete version.

6.2 Future work

The website is still under development, features are being completed such as: (i) email sending, (ii) table and order management, (iii) blog.

In the future, I will continue to improve the functions. In addition, I will develop additional functions such as: (i) improving user interface, (ii) advanced analysis and statistics, (iii) menu management, (iv) function automation , (v) performance optimization. Hopefully these features will help the website solve restaurant problems quickly and effectively, bringing a good experience to users.

REFERENCE

[1] *Angular documentation*. [Online]. Available: https://angular.dev/overview (visited on 06/20/2024).

[2] *Java spring boot*. [Online]. Available: https://kungfutech.edu.vn/khoa-hoc/spring-boot?fbclid=IwZXh0bgNhZW0CMTAAAR2dNqo2FRPsbETrWZW4v1pCf2iS1mPkqyWSHG7IOX-EbTbfjZtYJLcYU_aem_ZmFrZWR1bW15MT (visited on 05/30/2024).

[3] *Mysql documentation*. [Online]. Available: https://dev.mysql.com/doc/ (visited on 05/30/2024).

[4] K. S. P. Reddy, *Beginning Spring Boot 2: Applications and microservices with the Spring framework*. Apress, 2017.

[5] *Json web tokens*. [Online]. Available: https://ponyfoo.com/articles/json-web-tokens-vs-session-cookies (visited on 06/20/2024).

[6] *Java hashing using md5, sha, pbkdf2, bcrypt and scrypt*. [Online]. Available: https://howtodoinjava.com/java/java-security/how-to-generate-secure-password-hash-md5-sha-pbkdf2-bcrypt-examples/ (visited on 06/21/2024).