

TIỂU LUẬN KẾT THÚC HỌC PHẦN CÔNG NGHỆ JAVA

**Thiết kế và xây dựng ứng dụng quản lý nhà hàng bằng Java**

Lớp: 10ĐH\_CNTT2

Sinh viên thực hiện:

1. Lưu Thy Thy 1050080078

Giảng viên hướng dẫn: Đặng Đức Trung

Tp. Hồ Chí Minh, tháng 12 năm 2023

**PHIẾU CHẤM TIỂU LUẬN**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **NỘI DUNG** | **ĐIỂM TỐI ĐA** | **ĐIỂM ĐẠT ĐƯỢC** |
| 1 | Tiểu luận viết đúng quy định về định dạng và đúng các thành phần nội dung (Bìa, mục lục, danh mục hình, danh mục bảng, mở đầu, chương 1. Cơ sở lý thuyết, Chương 2. Phân tích thiết kế phần mềm, Chương 3. Kết quả thử nghiệm, Phần Kết luận và hướng phát triển, Danh mục Tài liệu tham khảo) | 1 |  |
| 2 | Tiểu luận có cơ sở lý thuyết đúng | 0.5 |  |
| 3 | Tiểu luận có phân tích, thiết kế hệ thống cơ bản |  |  |
|  | * Use Case tổng quát | 0.5 |  |
|  | * Class Diagram | 0.5 |  |
|  | * Database Diagram | 0.5 |  |
| 4 | Tiểu luận có trình bày kết quả cài đặt thử nghiệm | 1 |  |
| 5 | Dự án có trang chủ với giao diện hợp lý | 1 |  |
| 6 | Dự án có trang thông tin với dữ liệu truy vấn được từ cơ sở dữ liệu | 1 |  |
| 7 | Dự án có phần quản lý dữ liệu với các chức năng |  |  |
|  | * Thêm dữ liệu | 1 |  |
|  | * Sửa dữ liệu | 1 |  |
|  | * Xóa dữ liệu | 0.5 |  |
|  | Tìm kiếm theo tiêu thông tin cung cấp | 0.5 |  |
|  | Có form chức năng thực hiện được chức năng chính của dự án | 1 |  |
| **TỔNG CỘNG** | | **10** |  |

|  |  |
| --- | --- |
| ĐIỂM SỐ | ĐIỂM CHỮ |
|  |  |

GIẢNG VIÊN CHẤM

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 5](#_Toc152947150)

[**CHƯƠNG 1. CƠ SỞ LÝ THUYẾT** 7](#_Toc152947151)

[**1.1.** **Tổng quan về ngôn ngữ lập trình Java** 7](#_Toc152947152)

[**1.2.** **Tổng quan về Angular và Spring Boot** 10](#_Toc152947153)

[**1.2.1.** **Angular** 10](#_Toc152947154)

[**1.2.2.** **Spring Boot** 12](#_Toc152947155)

[**1.3.** **Tổng quan về môi trường phát triển phần mềm IntelliJ** 14](#_Toc152947156)

[**1.4.** **Tổng quan về Cơ sở dữ liệu quan hệ** 17](#_Toc152947157)

[**1.5.** **Tổng quan về Hệ quản trị cơ sở dữ liệu MySQL** 19](#_Toc152947158)

[**1.6.** **Tổng quan về Xampp** 20](#_Toc152947159)

[**1.7.** **Tổng quan về Postman** 22](#_Toc152947160)

[**1.8.** **Tổng quan về máy chủ Tomcat** 23](#_Toc152947161)

[**1.9.** **Tổng quan về nghiệp vụ của đề tài** 24](#_Toc152947162)

[**1.9.1.** **Nghiệp vụ là gì ?** 24](#_Toc152947163)

[**1.9.2.** **Nghiệp vụ của Thiết kế và xây dựng ứng dụng quản lý nhà hàng bằng Java** 25](#_Toc152947164)

[**CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ** 26](#_Toc152947165)

[**2.1. Phân tích yêu cầu nghiệp vụ** 26](#_Toc152947166)

[2.1.1. Quản lý đăng kí 26](#_Toc152947167)

[2.1.2. Quản lý đăng nhập 26](#_Toc152947168)

[2.1.3. Quản lý danh mục món ăn 26](#_Toc152947169)

[2.1.4. Quản lý thực đơn 26](#_Toc152947170)

[2.1.5. Quản lý đặt hàng 26](#_Toc152947171)

[2.1.6. Quản lý người dùng 27](#_Toc152947172)

[**2.2. Thiết kế biểu đồ chức năng** 27](#_Toc152947173)

[**2.3. Thiết kế biểu đồ UseCase** 28](#_Toc152947174)

[2.3.1. Biểu đồ UseCase tổng quát 28](#_Toc152947175)

[2.3.2. Biểu đồ UseCase Quản lý danh mục món ăn 28](#_Toc152947176)

[2.3.3. Biểu đồ UseCase Quản lý thực đơn 28](#_Toc152947177)

[2.3.4. Biểu đồ UseCase Quản lý đặt hàng 29](#_Toc152947178)

[2.3.5. Biểu đồ UseCase Quản lý người dùng 29](#_Toc152947179)

[**2.4. Thiết kế biểu đồ hoạt động các Case (Activity Diagrams)** 30](#_Toc152947180)

[2.4.1. Biểu đồ hoạt động đăng kí 30](#_Toc152947181)

[2.4.2. Biểu đồ hoạt động đăng nhập 30](#_Toc152947182)

[2.4.3. Biểu đồ hoạt động tìm kiếm 31](#_Toc152947183)

[2.4.4. Biểu đồ hoạt động đặt hàng 31](#_Toc152947184)

[2.4.5. Biểu đồ hoạt động đổi mật khẩu 32](#_Toc152947185)

[**2.5. Thiết kế biểu đồ lớp (Class Diagram)** 32](#_Toc152947186)

[2.5.1. User và UserDao, Category và CategoryDao, Product và ProductDao, Bill và BillDao 32](#_Toc152947187)

[2.5.2. JWT 33](#_Toc152947188)

[2.5.3. Rest và RestImpl 33](#_Toc152947189)

[2.5.4. Service và ServiceImpl 33](#_Toc152947190)

[**2.6. Thiết kế cơ sở dữ liệu** 34](#_Toc152947191)

[2.6.1. ERD 34](#_Toc152947192)

[2.6.2. Thiết kế chi tiết các Tables 34](#_Toc152947193)

[**CHƯƠNG 3. CÀI ĐẶT THỰC NGHIỆM** 36](#_Toc152947194)

[**3.1. Cài đặt cơ sở dữ liệu** 36](#_Toc152947195)

[**3.2. Cài đặt phần mềm ứng dụng** 37](#_Toc152947196)

[**KẾT LUẬN** 44](#_Toc152947197)

[**TÀI LIỆU THAM KHẢO** 45](#_Toc152947198)

# **LỜI MỞ ĐẦU**

Ngày nay, các nhà hàng ngày càng sử dụng nhiều công nghệ thông tin để nâng cao hiệu quả hoạt động. Một ứng dụng quản lý nhà hàng tốt sẽ giúp các nhà hàng quản lý tốt hơn các hoạt động của mình, từ đó nâng cao chất lượng dịch vụ và tăng doanh thu. Ứng dụng quản lý nhà hàng là một ứng dụng thực tế, có nhu cầu sử dụng cao.

Tính thực tiễn: Ứng dụng quản lý nhà hàng là một ứng dụng thực tế, có nhu cầu sử dụng cao. Ngày nay, các nhà hàng ngày càng sử dụng nhiều công nghệ thông tin để nâng cao hiệu quả hoạt động. Một ứng dụng quản lý nhà hàng tốt sẽ giúp các nhà hàng quản lý tốt hơn các hoạt động của mình, từ đó nâng cao chất lượng dịch vụ và tăng doanh thu.

Tính ứng dụng: Ngôn ngữ lập trình Java là một ngôn ngữ lập trình phổ biến, được sử dụng rộng rãi trong nhiều lĩnh vực, trong đó có lĩnh vực phát triển ứng dụng quản lý. Việc sử dụng Java để phát triển ứng dụng quản lý nhà hàng sẽ giúp ứng dụng có tính ổn định và khả năng mở rộng cao.

Tính khả thi: Đề tài này là một đề tài khả thi, có thể được thực hiện trong thời gian và nguồn lực cho phép. Sinh viên có thể tham khảo các tài liệu tham khảo và các ứng dụng quản lý nhà hàng hiện có để thiết kế và xây dựng ứng dụng của mình.

Mục tiêu nghiên cứu của tiểu luận này là thiết kế và xây dựng một ứng dụng quản lý nhà hàng sử dụng ngôn ngữ lập trình Java. Ứng dụng này sẽ tập trung vào việc tự động hóa và tối ưu hóa quy trình quản lý nhà hàng. Tạo ra một ứng dụng quản lý nhà hàng hoàn chỉnh và hiệu quả sử dụng ngôn ngữ lập trình Java, giúp cải thiện quá trình quản lý và tạo ra trải nghiệm tốt hơn cho cả nhân viên và khách hàng của nhà hàng.

Đối tượng nghiên cứu của tiểu luận này là các nhà hàng và quán ẩm, đặc biệt là các doanh nghiệp trong ngành thực phẩm và dịch vụ ẩm thực. Tiểu luận tập trung vào việc phân tích, thiết kế và xây dựng ứng dụng quản lý nhà hàng sử dụng ngôn ngữ lập trình Java, và do đó, đối tượng nghiên cứu chính là những người quản lý nhà hàng, nhân viên và khách hàng liên quan đến việc quản lý và sử dụng ứng dụng này.Xác định phạm vi nghiên cứu. Những người quản lý nhà hàng sẽ trở thành đối tượng chính trong việc đánh giá tính khả thi và hiệu quả của ứng dụng

Việc lựa chọn phương pháp nghiên cứu cho tiểu luận "Thiết kế và xây dựng ứng dụng quản lý nhà hàng bằng Java" là một bước quan trọng để đảm bảo rằng nghiên cứu được tiến hành một cách hiệu quả và đạt được mục tiêu đã đề ra.

Trong việc thiết kế và xây dựng ứng dụng quản lý nhà hàng bằng Java, mục tiêu của nghiên cứu chủ yếu tập trung vào việc phát triển một ứng dụng thực tế và hiệu quả. Vì vậy, phương pháp nghiên cứu sẽ chủ yếu là Phát triển và thử nghiệm nguyên mẫu (Prototyping).

Phát triển và thử nghiệm nguyên mẫu (Prototyping): Chọn phương pháp này bởi vì chúng ta đang tạo một ứng dụng thực tế. Bằng cách xây dựng một phiên bản nguyên mẫu của ứng dụng, chúng ta có thể nhanh chóng kiểm tra và xác minh tính khả thi của các yêu cầu đã xác định. Thông qua việc kiểm tra nguyên mẫu với đối tượng nghiên cứu (những người quản lý nhà hàng, nhân viên và khách hàng), chúng ta có thể thu thập phản hồi quan trọng về giao diện, tính năng và hiệu suất.

Cấu trúc nội dung của tiểu luận tuân theo một cấu trúc logic và có tính chất trình tự để giúp đọc giả hiểu rõ quá trình thiết kế và phát triển ứng dụng quản lý nhà hàng. Dưới đây là cấu trúc nội dung :

Lời mở đầu

Chương 1: Cơ sở lý thuyết

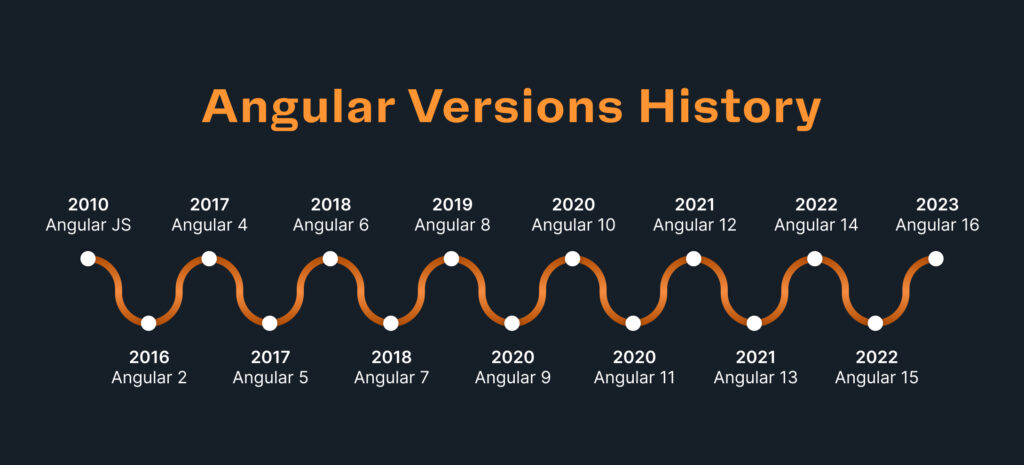
Chương 2: Phân tích và thiết kế

Chương 3: Cài đặt thử nghiệm

Kết luận

# **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

* 1. **Tổng quan về ngôn ngữ lập trình Java**
* Ngôn ngữ lập trình Java là một ngôn ngữ lập trình phổ biến và mạnh mẽ, được phát triển ban đầu bởi Sun Microsystems (nay là Oracle Corporation). Được tạo ra bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak. Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.
* Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.
* Một số tổng quan về ngôn ngữ Java:
* Độc lập nền tảng (Platform Independence): Một trong những đặc điểm nổi bật của Java là khả năng hoạt động trên nhiều hệ điều hành và kiến trúc máy tính khác nhau. Điều này là do mã nguồn Java được biên dịch thành mã bytecode, được thực thi trên máy ảo Java (JVM - Java Virtual Machine).
* Lập trình hướng đối tượng (Object-Oriented Programming - OOP): Java là một ngôn ngữ lập trình hướng đối tượng, có thể thực hiện các khái niệm quan trọng như lớp, đối tượng, kế thừa, đa hình, và đóng gói. Điều này giúp xây dựng ứng dụng có cấu trúc rõ ràng và dễ bảo trì.
* Bảo mật: Java có mô hình bảo mật mạnh mẽ. Ứng dụng Java chạy trong môi trường cách ly, và nó kiểm soát truy cập vào tài nguyên hệ thống, ngăn chặn nhiều loại tấn công bảo mật thông qua môi trường chạy. Java cung cấp một môi trường quản lý chương trình với nhiều mức khác nhau.
  + Mức 1 : Chỉ có thể truy xuất dữ liệu cũng như phương phức thông qua giao diện mà lớp cung cấp.
  + Mức 2 : Trình biên dịch kiểm soát các đoạn mã sao cho tuân thủ các quy tắc của ngôn ngữ lập trình Java trước khi thông dịch.
  + Mức 3 : Trình thông dịch sẽ kiểm tra mã byte code xem các đoạn mã này có đảm bảo được các quy định, quy tắc trước khi thực thi.
  + Mức 4: Java kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.
* Đa luồng (Multithreading): Java hỗ trợ đa luồng, cho phép ứng dụng thực hiện nhiều tác vụ đồng thời. Điều này làm cho ứng dụng Java phù hợp với việc xây dựng ứng dụng đa nhiệm và đa người dùng.
* Khả năng mở rộng (Scalability): Java có khả năng mở rộng tốt, từ ứng dụng nhỏ đơn giản cho đến các hệ thống lớn phức tạp.
* Thư viện tiêu chuẩn (Standard Library): Java đi kèm với một thư viện tiêu chuẩn (Java Standard Library) phong phú, bao gồm các lớp và giao diện cho xử lý chuỗi, I/O, mạng, đồ họa, và nhiều tính năng khác.
* Ngôn ngữ thông dịch: được chia ra làm 2 loại (tùy theo các hiện thực hóa ngôn ngữ đó) là ngôn ngữ thông dịch và ngôn ngữ biên dịch.
  + Thông dịch (Interpreter) : Nó dịch từng lệnh rồi chạy từng lệnh, lần sau muốn chạy lại thì phải dịch lại.
  + Biên dịch (Compiler): Code sau khi được biên dịch sẽ tạo ra 1 file thường là .exe, và file .exe này có thể đem sử dụng lại không cần biên dịch nữa.
* Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ vừa biên dịch vừa thông dịch. Cụ thể là: khi viết mã, hệ thống tạo ra một tệp .java. Khi biên dịch mã nguồn của chương trình sẽ được biên dịch ra mã byte code. Máy ảo Java (Java Virtual Machine) sẽ thông dịch mã byte code này thành machine code (hay native code) khi nhận được yêu cầu chạy chương trình.
* Các phiên bản của Java:
* Java Standard Edition (Java SE) – Là một nền tảng cơ bản cho phép phát triển giao diện điều khiển, các ứng dụng mạng và các ứng dụng dạng Win Form.
* Java Enterprise Edition (Java EE) – Được xây dựng trên nền tảng Java SE, giúp phát triển các ứng dụng web, các ứng dụng ở cấp doanh nghiệp, …
* Java Mobile Edition (Java ME) – Là một nền tảng cho phép phát triển các ứng dụng nhúng vào các thiết bị điện tử như mobile,…
* Các thành phần của Java SE Platform:
* JRE (Java Runtime Environment): cung cấp JVM (Java Virtual Machine) và thư viện được sử dụng để chạy chương trình Java.
* JDK (Java Development Kit): được biết đến như bộ cung cụ phát triển Java, bao gồm: trình biên dịch và trình gỡ rối được sử dụng để phát triển các ứng dụng Java.
* Ưu điểm:
* Java là nền tảng độc lập vì chúng ta có thể chạy mã Java trên bất kỳ máy nào mà không cần cài đặt bất kỳ phần mềm đặc biệt nào, JVM thực hiện điều đó.
* Java là hướng đối tượng vì các lớp và đối tượng của nó.
* Lý do chính để Java được bảo mật là con trỏ, Java không sử dụng con trỏ.
* Trong Java, chúng ta có thể thực thi nhiều chương trình đồng thời, do đó, có thể đạt được đa luồng.
* Java là một ngôn ngữ cấp cao giúp nó dễ hiểu.
* Quản lý bộ nhớ hiệu quả được thực hiện bằng Ngôn ngữ lập trình Java.
* Tự động quản lý bộ nhớ: Java sử dụng Garbage Collection để quản lý bộ nhớ, giúp ngăn ngừa rò rỉ bộ nhớ và giảm nguy cơ lỗi bộ nhớ.
* Nhược điểm:
* Là một ngôn ngữ cấp cao, nó phải xử lý các cấp độ biên dịch và trừu tượng của một máy ảo. Java thể hiện hiệu suất kém, nguyên nhân chính là do bộ thu gom rác, cấu hình bộ nhớ đệm không hợp lệ và bế tắc giữa các quy trình.
* Java có rất ít trình tạo GUI – Swing, SWT, JSF và JavaFX trong số những trình xây dựng phổ biến hơn.
* Khởi động chậm: JVM có thể cần thời gian khởi động, làm cho ứng dụng Java có thời gian phản hồi ban đầu chậm hơn.
* Cú pháp dài: Java có cú pháp dài hơn so với một số ngôn ngữ khác, điều này có thể đòi hỏi thời gian và công sức hơn trong việc viết mã.
* Giới hạn phát triển trên các thiết bị nhúng: Java không phải lúc nào cũng phù hợp cho phát triển trên các thiết bị nhúng hoặc có tài nguyên hạn chế.
* Các ứng dụng của Java:
* Lập trình web: Java là một ngôn ngữ lập trình phía máy chủ rất nổi tiếng trong việc xây dựng các ứng dụng web, với các framework phổ biến nhất là Spring, Struts, Hibernate, Apache Hadoop và JSF. Các trang web lớn như LinkedIn, AliExpress, Amazon… đều sử dụng Java để lập trình.
* Ứng dụng Android: Java là ngôn ngữ lập trình chính thức để phát triển Android, nó chiếm đến 46,2% tổng số ứng dụng Android hiện nay. Java được sử dụng bởi Android Studio, là IDE chính thức để phát triển các ứng dụng Android. Vì vậy, nếu làm việc với ngôn ngữ này, sẽ thấy việc phát triển Android trở nên dễ dàng hơn rất nhiều.
* Ứng dụng desktop: Java cũng phổ biến trong việc tạo giao diện đồ họa người dùng (GUI) cho các ứng dụng desktop. Java có rất nhiều framework sẵn có (như Swing, AWT, JavaFX, Griffon…), giúp cho việc phát triển GUI trở nên dễ dàng. Do đó, nếu đang định tạo một ứng dụng dành cho máy tính để bàn thì Java là một sự lựa chọn tuyệt vời.
* Video games: muốn học đồ họa và phát triển games, hãy chọn Java vì nó có rất nhiều framework và thư viện có sẵn để sử dụng như LibGDX và OpenGL.
* Big Data: ngoài Python thì còn có nhiều ngôn ngữ lập trình phổ biến khác cũng được sử dụng trong Big Data. Mặc dù vậy, khi nói đến Công nghệ Big Data, đa phần các lập trình viên đều ưa chuộng Java hơn. Điều này là do hầu hết các công cụ Big Data nổi tiếng (như Hadoop, Deeplearning4j) đều sử dụng Java.
* Internet vạn vật (IoT): ác nhà phát triển thường xem Java là ngôn ngữ tốt nhất để phát triển thiết bị IoT. Về cơ bản, Java là một ngôn ngữ lập trình không có nền tảng. Vì vậy, người ta chỉ cần tạo mã một lần và nó sẽ hoạt động trên nhiều thiết bị khác nhau, từ đó giúp giảm bớt khối lượng công việc tổng thể.
  1. **Tổng quan về Angular và Spring Boot và Rest API**
     1. **Angular**
* Angular là một mã nguồn mở viết bởi TypeScript và được sử dụng để xây dựng giao diện web (front – end). Angular được hình thành, phát triển từ khoảng năm 2009 và đang phát triển cho đến ngày nay bởi Google. Đây được xem là framework front end mạnh mẽ và chuyên biệt dành cho các nhà phát triển sử dụng HTML cao cấp.
* Hiện nay, Angular được các công ty lớn lựa chọn sử dụng như: Upwork, Forbes, General Motors, . .. Đây sẽ là cơ hội nghề nghiệp vô cùng lớn nếu sử dụng thành thạo Angular. Tuy nhiên, trước đó cần nắm được các khái niệm cơ bản như JavaScript, CSS và HTML, cách thao tác với kiến trúc Model-View-Controller (MVC).
* Lịch sử phát triển của Angular: phần mềm đã được nghiên cứu, xây dựng và chính thức ra mắt vào năm 2009 bởi Misko Hevery cùng với một người bạn khác là Adam Abrons. Angular được xem là một dự án riêng cho đến khi Misko Hevery tham gia vào dự án Google Feedback với vai trò là lập trình viên bán thời gian. Trong thời gian khoảng 6 tháng, Misko và 2 người khác cũng đã viết khoảng 17.000 dòng mã khác nhau cho dự án Google Feedback.



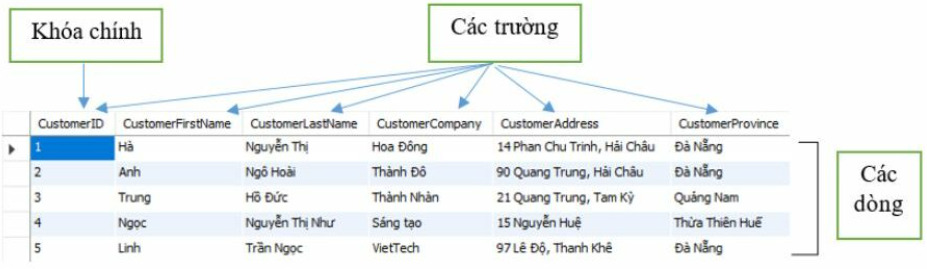
* Các đặc trưng của Angular:
* Angular có khả năng tạo ra các ứng dụng client-side dựa trên mô hình Model-View-Controller (MVC).
* Angular được các lập trình viên tận dụng để có thể xây dựng dựa trên JavaScript.
* Các mã JavaScript có thể được tự động hoá xử lý sao cho tương thích với từng trình duyệt nhất nhờ khả năng tương thích cao của Angular.
* Angular được dùng rộng rãi khi có mã nguồn mở hoàn toàn miễn phí.
* Các tính năng của Angular:
* Controller: tính năng tự động đồng bộ dữ liệu giữa đối tối tượng $scope. Với tính năng này, bên view sẽ dùng toàn bộ dữ liệu có sẵn tại scope để thực hiện hiển thị tương ứng.
* Data-binding: tính năng hỗ trợ đồng bộ hoá dữ liệu giữa hai chiều view và model khi view có thay đổi.
* Service: được coi là singleton object chuyên dùng để sử dụng các mô hình dữ liệu có sẵn như: ($http, $controller, $sce, $complie, $document, $parse, $httpBackend, . ..) chỉ khởi tạo 1 lần duy nhất.
* Filter: có khả năng lọc phần tử có trong item ở từng mảng và trả kết quả cho từng item mới.
* Directive: được dùng để tạo ra các thẻ HTML riêng và thường có các directive mặc định sau: ngModel, ngBing, . ..
* Temple: có chức năng thu thập dữ liệu từ controller và được xem là một thành phần của view.
* Routing: khả năng điều hướng, di chuyển giữa các action trong controller.
* MVC và MVVN: là mô hình được thiết kế với mục tiêu phân loại những ứng dụng có nhiều thành phần mà thường gắn liền với MVC.
* Deeplink: Liên kết sâu sẽ hỗ trợ lập trình viên trong việc xác định trạng thái của ứng dụng URL và có thể bookmark với nhiều kết quả tìm kiếm khác. Hầu hết có thể phục hồi được từ những địa chỉ URL có cùng trạng thái từ ứng dụng này.
* Dependency Injection: Thường được sử dụng trong phần mềm AngularJS hỗ trợ xây dựng những ứng dụng có nhiều khả năng mở rộng, dễ dàng theo dõi và kiểm soát.
* Ưu điểm:
* Angular được các chuyên gia đánh giá cao, mã nguồn này hỗ trợ các Single Page Application lm việc đơn giản, nhanh chóng.
* Nhờ tính năng Binding data dựa trên các nền tảng HTML nên code front-end sẽ cực kỳ thân thiện với người dùng.
* Có thể sử dụng Unit Test.
* Component giúp tái sử dụng tốt hơn.
* Angular có khả năng mở rộng giúp cho lập trình viên có thể chỉnh sửa code được dễ dàng hơn và với nhiều tính năng hơn. Từ đó giúp giảm chi phí lập trình và cải thiện năng suất làm việc.
* AngularJS tương thích với nhiều hệ điều hành khác nhau. Có thể dùng tốt trên nhiều loại thiết bị khác nhau kể cả trong máy tính để bàn và trên Laptop.
* Nhược điểm:
* Tính an toàn: bản chất của Angular là một framework front-end. Thông thường, tính an toàn của front-end sẽ không cao bằng back-end. Chính vì vậy phải có một hệ thống check dữ liệu sao cho việc trả về kết quả tối ưu nhất khi dùng API.
* Khả năng bảo mật: website có thể trở nên không bảo mật nếu dùng một số trình duyệt có tính năng Disable JavaScript.
  + 1. **Spring Boot**
* SpringBoot là một dự án phát triển bằng JAV (ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hoá việc phát triển một ứng dụng với Spring, chủ yếu tập trung vào quá trình phát triển business cho ứng dụng.
* Là một Java framework siêu lớn và đa năng, làm được tất cả mọi thứ. Nó được phân chia làm nhiều module, mỗi module làm một chức năng, ví dụ Spring Core, Web, Data access, AOP, . .. Spring được phát triển dựa trên 2 khái niệm nền tảng là Dependency injection và AOP (Aspect Oriented Programming).



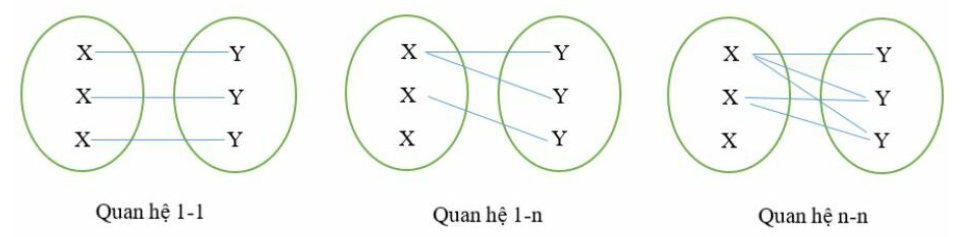
* Để phát triển một ứng dụng web cơ bản HelloWorld với Spring framework sẽ cần tối thiểu 5 công đoạn sau:
* Tạo một project chạy Maven với các dependency thiết yếu của Spring MVC và Servlet API.
* Một tập tin web.xml để cấu hình DispatcherServlet của Spring MVC.
* Một tập tin cấu hình của Spring MVC.
* Một class Controller trả về một trang "Hello World" khi có request đến.
* Cuối cùng là phải có một web server thích hợp để đưa ứng dụng lên host.



* Giờ đây với Spring Boot, có thể tạo dự án Spring một cách dễ dàng và cấu hình cũng đơn giản với Sublime Text để code mà không phải cài đặt Eclipse hay NetBeans
* Các tính năng của Spring boot:
* Tạo các ứng dụng Spring độc lập
* Nhúng trực tiếp Tomcat, Jetty hoặc Undertow (không cần thiết phải deploy ra file WAR)
* Các starter dependency giúp việc cấu hình Maven đơn giản hơn
* Tự động cấu hình Spring khi cần thiết
* Không sinh code cấu hình và không bắt buộc phải cấu hình qua XML.
* Các kiến thức cần thiết khi học Spring boot:
* Java Core
  + Java cơ bản: Hàm, biến, vòng lặp, . ..
  + Hướng đối tượng OOP trong Java
  + Biết cách dùng các collection API thông dụng và quan trọng.
  + Ngoài ra còn có những tính năng khác mà cũng nên biết mặc dù spring boot ít khi dùng đến như Asynchronous, Stream API, Multithreading và File IO.
* Template engine
  + Sử dụng template engine để xử lý view trong các ứng dụng MVC để pass data vào view và trả về một trang HTML cơ bản trong spring boot. Ngoài ra, có thể tìm hiểu thêm về việc sử dụng Thymeleaf và JSP trong spring boot.
* Package manager
  + Sử dụng template engine để quản lý view trong các ứng dụng MVC giúp pass data vào view và trả về một trang HTML cơ bản trong spring boot. Ngoài ra, cũng nên tìm hiểu thêm về việc cài đặt Thymeleaf và JSP trong spring boot.
  + Để đảm bảo rằng thư viện được cài đặt đúng, thì packagemanager là thứ không thể thiếu được khi code dự án spring boot. Java có hai package manager tương tự với NPM và Yarn của Javascript là Gradle và Maven. Nên tìm hiểu kỹ cách sử dụng của 2 loại package manager trên.
* JPA/MongoDB
  + Cách thiết lập cấu hình database trong spring boot.
  + Cấu trúc các lớp và interface của JPA/MongoDB.
  + Các thao tác CRUD cơ bản.
  + Tự động tạo các hàm và query dựa trên tên method.
  + Sử dụng MongoTemplate để viết các câu query phức tạp.
  + Phân trang và sắp xếp lại nội dung query.
* Ưu điểm:
* Phát triển các ứng dụng dựa trên Spring một cách tiết kiệm chi phí và dễ dàng.
* Tự động cài đặt tất cả các components cho một ứng dụng Spring cấp sản xuất.
* Các máy chủ nhúng được tích hợp sẵn (Tomcat, Jetty và Undertow), dẫn đến việc phát triển ứng dụng trở nên nhanh chóng và dễ dàng hơn bao giờ hết.
* Nhiều lựa chọn mở rộng, hỗ trợ nhà phát triển làm việc với dữ liệu được nhúng bên trong cơ sở dữ liệu.
* Dễ dàng truy cập cơ sở dữ liệu thông qua các dịch vụ hàng chờ như MySQL, Oracle, MongoDB, Redis, ActiveMQ, . ..
* Tích hợp với hệ sinh thái Spring.
* Giảm thiểu thời gian phát triển code, cải thiện tốc độ phát triển chung của toàn hệ thống.
* Dễ dàng cài đặt các module khác như Sping-MVC, Spring Data, Spring Sercurity, Spring Cloud
* Nó cung cấp các HTTPs servers như Tomcat, Jety, . . giúp việc phát triển, kiểm thử, deploy ứng dụng dễ dàng.
* Cung cấp công cụ CLI (Command Line Interface) hỗ trợ quá trình phát triển và test phần mềm dễ dàng từ command line
* Ngoài ra cũng có nhiều plugins được triển khai tự động và các công cụ hỗ trợ Build như Maven hoặc Gradle.
* Nhược điểm:
* Thiếu kiểm soát. Do style cố định, Spring Boot tạo ra nhiều tác động không được kiểm soát liên quan đến các tệp thực thi lớn.
* Quá trình chuyển đổi dự án Spring cũ hoặc hiện có sang các ứng dụng SpringBoot nhiều khó khăn và tốn thời gian.
* Không thích hợp xây dựng các ứng dụng quy mô lớn. Hoạt động đồng thời với các microservices, theo nhiều nhà phát triển, SpringBoot không thích hợp để phát triển các ứng dụng đa nền tảng.
  + 1. **Rest API**
* API REST (còn được gọi là API RESTful) là giao diện lập trình ứng dụng (API) tuân thủ các ràng buộc và quy ước của kiến ​​trúc REST được sử dụng trong giao tiếp giữa máy khách và máy chủ. REST là viết tắt của Representational State Transfer và được phát triển bởi nhà khoa học máy tính Roy Fielding.
* API (Application Programming Interface) là một bộ quy tắc và cơ chế mà theo đó một ứng dụng hoặc thành phần tương tác với một ứng dụng hoặc thành phần khác. API có thể trả về dữ liệu cần cho ứng dụng của mình ở các loại dữ liệu phổ biến như JSON hoặc XML.
* REST (Representational State Transfer) là một hình thức chuyển đổi cấu trúc dữ liệu, một kiểu kiến ​​trúc để viết API. Nó sử dụng một phương thức HTTP đơn giản để liên lạc giữa máy với máy. Vì vậy, thay vì sử dụng URL để xử lý thông tin người dùng, REST gửi yêu cầu HTTP như GET, POST, DELETE,... tới URL để xử lý dữ liệu.
* RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.
* Các phương thức cơ bản (Methods):
* GET: Trả về một Resource hoặc một danh sách Resource.
* POST: Tạo mới một Resource.
* PUT: Cập nhật thông tin cho Resource (toàn bộ resource).
* PATCH: Cật nhật thông tin cho resourse (một phần resource).
* DELETE: Xoá một Resource.
* Header: Authentication và quy định kiểu dữ liệu trả về
* Khi một yêu cầu trong REST cần xác thực quyền truy cập, thông tin bổ sung phải được sử dụng trong tiêu đề. Thông tin ủy quyền bao gồm, ví dụ: mã thông báo người dùng. Hiện tại có ba cơ chế xác thực chính:
  + HTTP Basic
  + JSON Web Token (JWT)
  + OAuth2
* Ngoài ra, header còn giúp client chỉ định loại nội dung được trả về từ máy chủ: Content-Type. Việc này được thực hiện thông qua phần Accept (Chấp nhận) trong tiêu đề. Giá trị của nó thường thuộc loại MIME:
  + image – image/png, image/jpeg, image/gif
  + audio – audio/wav, audio/mpeg
  + video – video/mp4, video/ogg
* Status Code :
* 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
* 201 Created – Trả về khi một Resouce vừa được tạo thành công.
* 204 No Content – Trả về khi Resource xoá thành công.
* 304 Not Modified – Client có thể sử dụng dữ liệu cache, resource server không đổi gì.
* 400 Bad Request – Request không hợp lệ
* 401 Unauthorized – Request cần có xác thực.
* 403 Forbidden – bị từ chối không cho phép.
* 404 Not Found – Không tìm thấy resource từ URI
* 405 Method Not Allowed – Phương thức không cho phép với user hiện tại.
* 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.
* 415 Unsupported Media Type – Không hỗ trợ kiểu Resource này.
* 422 Unprocessable Entity – Dữ liệu không được xác thực
* 429 Too Many Requests – Request bị từ chối do bị giới hạn
  1. **Tổng quan về môi trường phát triển phần mềm IntelliJ**
* IntelliJ là một IDE dựa trên Java hàng đầu được các doanh nghiệp sử dụng. Là công cụ lập trình và phát triển bởi JetBrains.Môi trường phát triển này cung cấp các chức năng cấp doanh nghiệp đặc biệt, bao gồm lập chỉ mục (index) mã nguồn mở, tái cấu trúc và phân tích code động.
* JetBrains được thành lập và chính thức đưa vào sử dụng năm 2000, trụ sở tại Prague, nước Áo. Nhóm thành viên của JetBrains ban đầu gồm có 7 chuyên gia lập trình, phát triển, đến nay 2022 JetBrains đã trở thành một tập đoàn với gần 2000 nhân viên. JetBrains thuộc Top 100 công ty phần mềm tốt nhất toàn cầu.
* JetBrains là thương hiệu hàng đầu thế giới cung cấp giải pháp công cụ lập trình và phát triển với 12,8 triệu người dùng trên 220 quốc gia. Tính đến năm 2022, JetBrains đã cung cấp cho ngành lập trình thế giới 34 công cụ khác nhau, trên 7000 plugins và đạt được hơn 110 giải thưởng uy tín thế giới về các sản phẩm công cụ do hãng cung cấp tới người dùng và tổ chức.
* IDE này có hai biến thể: phiên bản cộng đồng miễn phí và phiên bản nâng cao. Phiên bản nâng cao được thiết kế riêng biệt phù hợp với nhu cầu của doanh nghiệp.
* Từng đoạt giải thưởng Java IDEA, IntelliJ IDEA được thiết kế nhằm tối ưu hoá hiệu suất ngôn ngữ lập trình với khả năng biên tập và phân tích code thông minh, tập hợp các công cụ hỗ trợ mạnh mẽ cho một loạt các ngôn ngữ lập trình.
* IntelliJ IDEA cung cấp code thông minh, công cụ phân tích code và tập hợp mạnh mẽ của refactorings hỗ trợ một loạt các ngôn ngữ lập trình, các khuôn khổ và thuật toán.
* Cung cấp các công cụ mạnh mẽ cho việc lập trình Java, JavaScript/ActionScript/Flex, HTML/XHTML/CSS, XML/XSL, PHP, Ruby/JRuby, Groovy, SQL, FreeMarker/Velocity, JSP, JSF và JSF 2.0, EJB, AJAX, GWT, GWT 2.0, Google App Engine, Struts, Struts 2, JBoss Seam, Spring, Spring 2.5 and 3.0, Spring Security 2.0, Hibernate/JPA and JPA 2.0, Tapestry, Web Beans, Bean Validation, OSGi, Android, và nhiều hơn thế nữa.
* Lịch sử ra đời:
* 2000: IntelliJ IDEA được phát hành lần đầu tiên vào tháng 1 năm 2001. JetBrains, một công ty phần mềm có trụ sở tại Nga, là người sáng lập và phát triển nó.
* Ban đầu: IntelliJ IDEA đã nhanh chóng thu hút sự chú ý với các tính năng nổi bật như trí tuệ biên dịch, refactorings thông minh và hỗ trợ mã hóa chất lượng cao.
* Các phiên bản phát triển: Các phiên bản phát triển tiếp theo của IntelliJ IDEA tiếp tục cải tiến với nhiều tính năng và công cụ mới, bao gồm hỗ trợ cho nhiều ngôn ngữ lập trình và framework khác nhau.
* 2003: IntelliJ IDEA 3.0 được phát hành, mở rộng khả năng hỗ trợ cho các ngôn ngữ khác ngoài Java.
* 2005: Phiên bản IntelliJ IDEA 5.0 giới thiệu hỗ trợ cho các dự án phức tạp và cải tiến hiệu suất.
* 2008: IntelliJ IDEA 7.0 có sự thêm vào nhiều tính năng như Spring, Hibernate và nhiều framework khác.
* 2010: Phiên bản IntelliJ IDEA 9.0 đưa ra sự hỗ trợ đặc biệt cho Java EE 6 và một số tính năng mới khác.
* 2012: IntelliJ IDEA 11.0 có sự thêm vào tính năng hỗ trợ cho các ngôn ngữ như Scala, Clojure, và Groovy.
* 2014: IntelliJ IDEA 13.0 ra mắt với nhiều cải tiến như hỗ trợ Docker, AngularJS, và nhiều tính năng khác.
* 2016: IntelliJ IDEA 2016.2 bổ sung nhiều tính năng như hỗ trợ JavaScript và TypeScript.
* Các phiên bản mới của IntelliJ IDEA tiếp tục được phát triển với nhiều tính năng và cải tiến mới, cũng như việc mở rộng hỗ trợ cho nhiều ngôn ngữ lập trình và framework khác nhau.
* Các tính năng của IntelliJ:
* Luồng dữ liệu phân tích (Analysis Data Flow): IntelliJ tạo một danh sách các ký hiệu một cách thông minh trong khi nhập code có thể dùng bằng cách nhấn phím Enter, giúp giảm đáng kể chu kỳ thời gian phát triển code.
* Điều hướng đến Bản sao (Navigate to Duplicates): tính năng này tìm thấy các phần dư thừa trong chương trình và gạch dưới chúng. Ví dụ, nó giúp cho lập trình viên biết nếu đã khai báo cùng một biến hai lần. Nhờ đó, sẽ không cần phải tìm các phần thừa theo cách thủ công.
* Khắc phục nhanh: IntelliJ hiển thị biểu tượng hình bóng đèn khi mắc lỗi khi viết code và tự động sửa lỗi khi nhấn vào biểu tượng đó.
* Kiểm soát phiên bản: tính năng kiểm soát phiên bản giúp theo dõi dự án bằng cách mở rộng hỗ trợ cho GitHub. Nó là cứu cánh cho các nhà phát triển nguồn mở sử dụng GitHub để phát triển dự án của họ.
* Ưu điểm:
* Tính Năng Phong Phú: IntelliJ IDEA cung cấp một loạt các tính năng mạnh mẽ như refactorings thông minh, tự động hoàn thành mã, hỗ trợ độ debug và tính năng đa nhiệm.
* Hỗ Trợ Ngôn Ngữ Đa Dạng: Ngoài Java, IntelliJ IDEA hỗ trợ nhiều ngôn ngữ khác như Kotlin, Scala, Groovy, Python, JavaScript, TypeScript, HTML, CSS và nhiều ngôn ngữ khác.
* Hỗ Trợ Frameworks Phổ Biến: Nó cung cấp hỗ trợ tích hợp cho nhiều frameworks phổ biến như Spring, Hibernate, AngularJS, và nhiều framework khác.
* Tích Hợp Với Công Cụ Quản Lý Phiên Bản: IntelliJ IDEA tích hợp tốt với các hệ thống quản lý phiên bản như Git, SVN, và Mercurial.
* Hỗ Trợ Maven và Gradle: IntelliJ IDEA có tích hợp mạnh mẽ với các công cụ quản lý dự án như Maven và Gradle.
* Hiệu Suất Cao: Nó được thiết kế để cung cấp hiệu suất cao và khả năng mở rộng cho các dự án lớn và phức tạp.
* Hỗ trợ nhiều hệ thống, đồng thời hỗ trợ mở rộng database editor và UML designer
* Cho các tính năng chạy thử UI, bảo vệ code
* Hỗ trợ nhiều công cụ phát triển và gỡ lỗi cho hầu hết những servers
* Trình soạn thảo văn bản thông minh cho ngôn ngữ lập trình Java, CSS, HTML
* Tích hợp kiểm soát các phiên bản
* Nhược điểm:
* Giá Cả: IntelliJ IDEA Ultimate, phiên bản có đầy đủ tính năng, có một giá phải trả, đặc biệt là đối với cá nhân và các doanh nghiệp nhỏ.
* Yêu Cầu Tài Nguyên Hệ Thống: Do tính năng và tính năng đa dạng, IntelliJ IDEA yêu cầu một số tài nguyên hệ thống đáng kể, đặc biệt là khi làm việc trên các dự án lớn.
* Có quá nhiều phím tắt cần nhớ, UI giao tiếp cũng có những vấn đề chưa được cải thiện
  1. **Tổng quan về Cơ sở dữ liệu quan hệ**
* Cơ sở dữ liệu quan hệ (Relational Database) là một hệ thống lưu trữ dữ liệu dựa trên mô hình dữ liệu quan hệ. Đây là một hệ thống quản lý cơ sở dữ liệu (DBMS) sử dụng để tổ chức và lưu trữ dữ liệu dưới dạng bảng và các mối quan hệ giữa các bảng
* Một số tổng quan về cơ sở dữ liệu quan hệ
* Bảng (Table): Dữ liệu trong cơ sở dữ liệu quan hệ được tổ chức thành các bảng. Mỗi bảng có một tên và chứa hàng và cột, tạo thành một lưới dữ liệu có cấu trúc.
* Hàng (Row): Hàng trong một bảng chứa dữ liệu cho một bản ghi cụ thể hoặc một mục trong bảng đó.
* Cột (Column): Cột trong một bảng đại diện cho một loại dữ liệu cụ thể. Mỗi cột có một tên đại diện cho loại thông tin mà nó chứa.
* Khóa chính (Primary Key): Một cột hoặc một tập hợp các cột trong bảng được sử dụng để xác định một bản ghi duy nhất trong bảng. Khóa chính giúp duyệt và tìm kiếm dữ liệu nhanh chóng.
* Khóa ngoại (Foreign Key): Khóa ngoại là một cột trong bảng tham chiếu đến khóa chính của một bảng khác. Nó xác định mối quan hệ giữa các bảng và cho phép lấy thông tin từ nhiều bảng khác nhau.



* Mối quan hệ: sự kết nối giữa hai bảng để xác định mối liên quan giữa các trường dữ liệu. Biểu hiện ở 3 dạng sau:
  + Mối quan hệ 1-1: Mỗi bảng ghi chỉ có một và một bảng tướng ứng mà thôi
  + Mối quan hệ 1-n: Mối quan hệ này khá phổ biến trong cơ sở dữ liệu. Trong mối quan hệ này thì 1 bảng ghi ở bảng này có thể có nhiều bảng ghi tương ứng ở bảng kia.
  + Mối quan hệ n-n: Trong mối quan hệ này thì 1 bảng ghi ở bảng này có thể có nhiều bảng ghi tương ứng ở bảng kia và ngược lại.



* Ưu điểm:
* Cấu trúc dữ liệu rõ ràng: Dữ liệu được tổ chức trong bảng và cấu trúc dữ liệu rất rõ ràng, giúp quản lý và tìm kiếm dữ liệu dễ dàng.
* Kiểm soát dữ liệu: Các ràng buộc, như khóa chính và khóa ngoại, giúp bảo vệ tính toàn vẹn và chất lượng dữ liệu.
* Mối quan hệ phức tạp: Cơ sở dữ liệu quan hệ cho phép xây dựng mối quan hệ phức tạp giữa các bảng, cho phép tạo mô hình dữ liệu phức tạp.
* Truy vấn mạnh mẽ: SQL (Structured Query Language) được sử dụng để truy vấn và tương tác với dữ liệu trong cơ sở dữ liệu quan hệ.
* Hỗ trợ đa người dùng: Cơ sở dữ liệu quan hệ hỗ trợ nhiều người dùng cùng lúc và quản lý các giao dịch.
* Nhược điểm:
* Hiệu suất hạn chế: Đối với các ứng dụng có khối lượng lớn dữ liệu hoặc yêu cầu hiệu suất cao, cơ sở dữ liệu quan hệ có thể đối mặt với giới hạn hiệu suất.
* Phức tạp trong việc thay đổi cấu trúc: Khi cấu trúc dữ liệu cần thay đổi, việc thực hiện điều này có thể đòi hỏi sự quản lý và kiểm soát cẩn thận để không làm hỏng ứng dụng hiện có.
* Khó khắc phục lỗi: Đối với các cơ sở dữ liệu lớn, việc xác định và sửa lỗi có thể trở nên phức tạp.
* Spring Boot và MySQL:
* Spring Boot cung cấp tích hợp đơn giản với cơ sở dữ liệu MySQL bằng ứng dụng Spring Data JPA. Nó sử dụng Hibernate như nền tảng của JPA để thực thi các tác vụ liên quan đến cơ sở dữ liệu.
* Các thuộc tính cấu hình cơ sở dữ liệu (bao gồm url, username, password) thông thường được cài đặt trong file application.properties hoặc application.yml của ứng dụng Spring Boot.
* Spring Boot tự động thiết lập một vài bean liên quan đến JPA và Hibernate, giảm thiểu cài đặt cấu hình và khiến cho việc xây dựng, update cơ sở dữ liệu trở nên đơn giản.
* Spring Boot và JPA
* Spring Boot sử dụng Spring Data JPA để giúp việc lập trình với JPA trở nên đơn giản hơn. Spring Data JPA là một dự án con của Spring Data, có một vài cải tiến và thuận tiện khi dùng JPA.
* Spring Boot cung cấp các chức năng để tự động xây dựng bảng dựa trên các đối tượng Java (entities) sử dụng các annotation như @Entity, @Table, @Column, và các annotation khác của JPA.
* Các repository Spring Data JPA giúp thực hiện các thao tác đơn giản như tạo, chỉnh sửa, xoá, và truy xuất dữ liệu mà không phải viết code SQL.
  1. **Tổng quan về Hệ quản trị cơ sở dữ liệu MySQL**
* MySQL là một hệ quản trị cơ sở dữ liệu (DBMS) phổ biến và mã nguồn mở được sử dụng rộng rãi trên toàn thế giới và hỗ trợ đa số các ngôn ngữ lập trình
* Hệ quản trị cơ sở dữ liệu MySQL được hiểu như là chương trình dùng để quản lý hệ thống cơ sở dữ liệu, trong đó, cơ sở dữ liệu là một hệ thống lưu trữ thông tin được sắp xếp rõ ràng, phân lớp ngăn nắp giúp truy cập dữ liệu một cách thuận lợi và nhanh chóng nhất.
* MySQL là cơ sở dữ liệu tốc độ cao, ổn định hoạt động trên nhiều hệ điều hành, cung cấp một hệ thống lớn các hàm tiện ích rất mạnh
* MySQL là một hệ quản trị cơ sở dữ liệu quan hệ phát triển bởi Oracle Corporation. Nó hỗ trợ nhiều loại dữ liệu, bao gồm số nguyên, chuỗi, ngày tháng, hình ảnh và nhiều loại dữ liệu khác.
* Nó hỗ trợ nhiều hệ điều hành, bao gồm Windows, macOS và các phiên bản của Linux.
* MySQL được sử dụng cho việc bỗ trợ PHP, Perl và nhiều ngôn ngữ khác. Là nơi lưu trữ những thông tin trên các trang web viết bằng framework PHP hay Perl…
* Đặc điểm của MySQL:
* Là một trong những phần mềm quản trị CSDL dạng server based, hệ gần giống với SQL server of Microsoft.
* MySQL là phần mềm quản lý dữ liệu thông qua CSDL. Và mỗi một CSDL đều có bảng quan hệ chứa dữ liệu riêng biệt.
* MySQL có cơ chế quản lý sử dụng riêng giúp cho mỗi người sử dụng đều có thể quản lý cùng lúc một hay nhiều CSDL khác nhau. Và mỗi người dùng đều có 1 username và password để truy nhập và truy xuất đến CSDL. Khi truy vấn đến CSDL của MySQL, phải cung cấp tài khoản và mật khẩu có quyền sử dụng cơ sở dữ liệu đó.
* MySQL tương thích tốt với môi trường PHP, giúp hệ thống hoạt động mạnh mẽ. Có thể viết hoặc chỉnh sửa code MySQL trên các IDE dành cho PHP
* Sử dụng MySQL:
* MySQL có phiên bản dòng lệnh (Command-Line) cho việc tương tác trực tiếp với cơ sở dữ liệu.
* Có nhiều công cụ quản lý cơ sở dữ liệu dựa trên giao diện đồ họa cho MySQL, bao gồm phpMyAdmin, MySQL Workbench và nhiều công cụ khác.
* Nhiều ngôn ngữ lập trình hỗ trợ kết nối và tương tác với cơ sở dữ liệu MySQL, chẳng hạn như Java, Python, PHP và nhiều ngôn ngữ khác.
* Ứng dụng của MySQL:
* MySQL được sử dụng rộng rãi trong các ứng dụng web, đặc biệt là trong việc lưu trữ và quản lý dữ liệu trên các trang web động.
* Nó cũng được sử dụng trong các ứng dụng di động, ứng dụng máy tính cá nhân và ứng dụng doanh nghiệp để lưu trữ và truy xuất dữ liệu.
* MySQL cũng có thể được triển khai trên các máy chủ dùng cho mục đích xử lý dữ liệu lớn (Big Data) và phân tích dữ liệu.
* Ưu điểm:
* Miễn phí và mã nguồn mở.
* Hiệu suất cao và xử lý tải lớn.
* Hỗ trợ đa nền tảng.
* Hỗ trợ giao dịch và tính toàn vẹn dữ liệu.
* Nhược điểm:
* Có giới hạn trong việc xử lý các cấu trúc dữ liệu phức tạp.
* Hiệu suất có thể giảm đi khi cơ sở dữ liệu trở nên quá lớn.
* Các tính năng cao cấp có thể yêu cầu phiên bản thương mại (Oracle MySQL).
  1. **Tổng quan về Xampp**
* XAMPP là một phần mềm mã nguồn mở được sử dụng để cài đặt và quản lý môi trường phát triển web trên máy tính cá nhân hoặc máy chủ. Tên "XAMPP" xuất phát từ viết tắt của "X" (để cho biết nó là có sẵn cho nhiều nền tảng), "Apache" (máy chủ web), "MySQL" (hệ quản trị cơ sở dữ liệu), "PHP" (ngôn ngữ lập trình phía máy chủ), và "Perl" (ngôn ngữ lập trình kịch bản). Dưới đây là một tổng quan về XAMPP:
* Là một trong những phần mềm được nhiều lập trình viên sử dụng để thiết lập website theo ngôn ngữ PHP. XAMPP có công dụng thiết lập web server có cài đặt sẵn các công cụ như PHP, Apache, MySQL…
* Xampp sở hữu thiết kế giao diện thân thiện với người dùng, cho phép các lập trình viên có thể đóng mở hoặc reboot các tính năng của server mọi lúc.
* Xampp được đánh giá là một trong những phần mềm trọng yếu trong việc xây dựng website dành cho các lập trình viên PHP
* Có thể tương thích với nhiều hệ điều hành như Windows, Mac, Linux, Solaris…
* Các thành phần của XAMPP:
* Apache: là một website software sử dụng source code mở hoàn toàn miễn phí, được sử dụng bởi 46% trên tổng số các website trên toàn cầu.
* PHP : PHP (viết tắt của Hypertext Pre-processor, tiền thân là Personal Home Page) là một ngôn ngữ lập trình kịch bản, được ứng dụng trong các loại website (tĩnh và động) và ứng dụng website. PHP có thể được nhúng trong HTML, được ứng dụng trong quản lý các loại nội dung website
* MySql : là một hệ thống quản lý cơ sở dữ liệu liên kết có ưu điểm nhanh chóng, dễ dùng cho các lập trình viên.
* Perl : có tên đầy đủ là Practical Extraction and Report Language, là một trong những ngôn ngữ lập trình hiện đại hoạt động song song với PHP trong Xampp có khả năng lọc bỏ những dữ liệu thừa và cung cấp những dữ liệu quan trọng trong việc thiết lập website
* Chức năng của XAMPP:
* XAMPP cung cấp một cách dễ dàng để cài đặt và cấu hình môi trường phát triển web trên máy tính. Nó bao gồm các thành phần quan trọng như máy chủ web Apache, hệ quản trị cơ sở dữ liệu MySQL, PHP và Perl.
* Ngoài ra, XAMPP cũng bao gồm phpMyAdmin (giao diện quản lý cơ sở dữ liệu MySQL), phpSQLiteAdmin (quản lý cơ sở dữ liệu SQLite), Mercury Mail (máy chủ email), FileZilla FTP Server (máy chủ FTP), Tomcat (máy chủ ứng dụng Java), và nhiều công cụ khác.
* Ứng dụng của XAMPP:
* Phát triển và kiểm thử ứng dụng web trên máy tính cá nhân.
* Tạo môi trường phát triển web để thử nghiệm các ứng dụng trước khi triển khai trên máy chủ thật.
* Học và thử nghiệm các công nghệ web như PHP, MySQL, và Apache.
* Phát triển trang web cục bộ trước khi đưa lên một máy chủ sản xuất.
* Ưu điểm:
* Dễ cài đặt và sử dụng: XAMPP có giao diện cài đặt và quản lý dễ sử dụng, giúp nhanh chóng tạo ra môi trường phát triển web trên máy tính cá nhân.
* Đa nền tảng: XAMPP hỗ trợ nhiều hệ điều hành, bao gồm Windows, macOS và Linux.
* Môi trường phát triển đầy đủ: XAMPP bao gồm tất cả các thành phần cần thiết để phát triển và kiểm thử ứng dụng web, bao gồm máy chủ web, cơ sở dữ liệu, và ngôn ngữ lập trình phía máy chủ.
* Tùy chỉnh: có thể tùy chỉnh các cài đặt của môi trường phát triển theo nhu cầu của bản thân
* Mã nguồn mở: XAMPP là phần mềm mã nguồn mở, miễn phí và được cộng đồng phát triển.
* Nhược điểm:
* Không phù hợp cho môi trường sản xuất: XAMPP thường được sử dụng cho mục đích phát triển và kiểm thử, không phù hợp cho môi trường sản xuất thực tế.
* Bảo mật yếu: mặc dù có thể tùy chỉnh cài đặt để làm cho nó an toàn hơn, XAMPP mặc định không cấu hình với mức độ bảo mật cao và có thể dễ dàng bị tấn công nếu không cấu hình cẩn thận.
  1. **Tổng quan về Postman**
* Postman hiện là một trong những công cụ phổ biến nhất để kiểm tra API. Nó bắt đầu vào năm 2012 như một dự án phụ của Abhinav Asthana với mục tiêu đơn giản hóa quy trình làm việc API trong thử nghiệm và phát triển. API là viết tắt của Application Programming Interface, cho phép các ứng dụng phần mềm giao tiếp với nhau thông qua lệnh gọi API.
* Postman là một công cụ cho phép làm việc với API, phổ biến nhất là REST. Postman hiện là một trong những công cụ phổ biến nhất để thử nghiệm API. Người đưa thư cho phép gọi API nghỉ ngơi mà không cần viết một dòng mã nào.
* Người đưa thư hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, v.v.). Cũng có thể lưu lại lịch sử yêu cầu của mình, điều này rất thuận tiện và cho phép sử dụng lại khi cần.
* Các chức năng cơ bản của Postman:
* Cho phép gửi yêu cầu HTTP bằng các phương thức GET, POST, PUT và DELETE.
* Cho phép POST dữ liệu ở dạng key-value, văn bản (text) hoặc JSON.
* Xem kết quả trả về dưới dạng văn bản, hình ảnh, XML và JSON.
* Hỗ trợ ủy quyền (Oauth1, 2).
* Có thể thay đổi tiêu đề yêu cầu.
* Các tính năng của Postman:
* New: Cho phép người dùng tạo request, environment hoặc collection.
* Import: Cho phép người dùng import collection hoặc environment. Một số tùy chọn khác còn được sử dụng để import từ file folder, paste từ text thuần hoặc thậm chí là link.
* Open new: Cho phép người dùng mở một tab mới, cửa sổ runner hoặc cửa sổ postman.
* Runner: Cho phép người dùng kiểm tra một cách tự động thông qua Runner và collection.
* My workspace: Cho phép người dùng tạo cửa sổ làm việc cá nhân hoặc theo nhóm.
* Invite: Cho phép người dùng mời các thành viên làm việc và cộng tác cùng nhau.
* History: Cho phép người dùng lần theo những request đã thực hiện từ trước.
* Collections: Cho phép người dùng tổ chức các thử nghiệm. Mỗi collection sẽ chứa các thư mục con với nhiều yêu cầu khác nhau, có thể là request hoặc thư mục trùng lặp.
* Tab request: Hiển thị tiêu đề request mà người dùng làm việc. Theo đó, nó sẽ mặc định “untitled Request” đồng thời hiển thị cho các request không có tiêu đề khác.
* Request URL: Đây còn được gọi là điểm cuối, là nơi cho phép người dùng xác định liên kết đến nơi mà API sẽ thực hiện giao tiếp.
* HTTP Request: Khi người dùng click vào HTTP Request thì danh sách hiển thị sẽ được thả xuống với các request khác, có thể là post, copy, delete, hoặc get,…
* Save: Trong trường hợp thay đổi request, người dùng chỉ cần nhấp vào Save, các thay đổi sẽ tự động lưu và không bị ghi đè.
* Params: Cho phép người dùng vẽ các tham số cần thiết cho một request.
* Headers: Cho phép người dùng tiến hành thiết lập các header, chẳng hạn như nội dung JSON tùy thuộc theo cách tổ chức của bản thân.
* Body: Cho phép người dùng tùy chỉnh các chi tiết trong phần request. Đây là phần thường được dùng nhiều nhất trong request Post.
* Tests: Đây là những script được thực hiện khi tiến hành request. Thế nhưng, nó cần phải có các thử nghiệm như thiết lập điểm checkpoint để kiểm tra trạng thái. Khi đó, những dữ liệu nhận được sẽ đáp ứng mong đợi của người dùng đồng thời sở hữu các thử nghiệm khác.
* Pre-request script: Đây là các tập lệnh sẽ được thực thi trước khi request. Đa phần, nó sẽ cho môi trường được sử dụng nhằm đảm bảo việc kiểm tra, giúp chúng có thể chạy trong môi trường chính xác nhất.
  1. **Tổng quan về máy chủ Tomcat**
* Apache Tomcat là một loại web server HTTP được phát triển bởi Apache Software Foundation, nó có khả năng hỗ trợ mạnh cho các ứng dụng Java thay vì các website tĩnh khác. DO vậy, nó có khả năng chạy trên nhiều bản Java chuyên biệt như: Java Servlet, JavaServer Pages (JSP), Java EL và WebSocket.
* Chức năng của Tomcat:
* Tomcat là một máy chủ ứng dụng web chuyên dụng được thiết kế để thực thi ứng dụng web phát triển bằng ngôn ngữ lập trình Java, như Servlet và JSP.
* Nó cung cấp môi trường thực thi ứng dụng web Java, cho phép chạy mã Java trên máy chủ và phản hồi các yêu cầu từ trình duyệt web.
* Đặc điểm:
* Mã nguồn mở: Tomcat là phần mềm mã nguồn mở, có sẵn miễn phí và được phát triển bởi cộng đồng.
* Hỗ trợ Servlet và JSP: Nó hỗ trợ Servlet API và JSP, cho phép phát triển ứng dụng web động bằng Java.
* Dễ cài đặt và cấu hình: Tomcat có quá trình cài đặt và cấu hình đơn giản, giúp nhanh chóng thiết lập môi trường phát triển.
* Hiệu suất tốt: Tomcat được tối ưu hóa để cung cấp hiệu suất cao cho các ứng dụng web Java.
* Môi trường phát triển: Nó cung cấp môi trường phát triển cục bộ cho việc phát triển và kiểm thử ứng dụng trước khi triển khai trên máy chủ sản xuất.
* Sử dụng:
* Máy chủ Tomcat có thể sử dụng cho phát triển ứng dụng web Java trên máy tính cá nhân.
* Nó cũng có thể triển khai trên các máy chủ sản xuất để chạy ứng dụng web thực tế.
* Tomcat thường được sử dụng trong kết hợp với các máy chủ web như Apache HTTP Server để cung cấp môi trường phát triển đa nhiệm, với Tomcat xử lý ứng dụng web Java và Apache xử lý các tập tin tĩnh và yêu cầu web khác.
* Ưu điểm:
* Miễn phí và mã nguồn mở.
* Sở hữu tính ổn định cao
* Có khả năng hỗ trợ cho các hệ điều hành như: windows, linux, Mac OS, BSD,…
* Sở hữu những chức năng bổ sung như: tomcat manager application, specialized realm implementation và tomcat valves.
* Hỗ trợ nhiều ngôn ngữ lập trình Java, Servlet và JSP.
* Dễ cài đặt và cấu hình.
* Hiệu suất tốt cho ứng dụng web Java.
* Tích hợp tốt với các công cụ phát triển Java.
* Nhược điểm:
* Thích hợp cho ứng dụng web Java, không phải cho mọi loại ứng dụng.
* Yêu cầu kiến thức về Java và ứng dụng web để tận dụng hết tiềm năng của nó.
* Có thể gặp phải một số vấn đề về hiệu năng website với lượng truy cập lớn.
  1. **Tổng quan về nghiệp vụ của đề tài**
     1. **Nghiệp vụ là gì ?**
* Nghiệp vụ là một thuật ngữ thường được sử dụng trong lĩnh vực kinh doanh và quản lý để chỉ một loạt các hoạt động hoặc công việc cụ thể liên quan đến một lĩnh vực hoặc một ngành nghề cụ thể. Nó bao gồm tất cả các công việc, quy trình, và hoạt động cần thiết để thực hiện một nhiệm vụ hoặc mục tiêu cụ thể. Nghiệp vụ thường bao gồm các bước, quy trình, và quy định cụ thể mà nhân viên hoặc chuyên gia trong ngành nghề đó cần tuân theo.
* Một số tổng quan về nghiệp vụ:
* Mô tả chi tiết: Nghiệp vụ cung cấp mô tả chi tiết về các hoạt động và quy trình trong một ngành nghề hoặc lĩnh vực cụ thể. Nó giúp định nghĩa rõ ràng nhiệm vụ và trách nhiệm của từng công việc.
* Quy trình làm việc: Nó mô tả các bước cụ thể mà nhân viên hoặc chuyên gia cần thực hiện để hoàn thành một nhiệm vụ. Quy trình làm việc thường có thể được biểu đồ hóa để hỗ trợ trong việc hiểu và thực hiện chúng.
* Chuẩn mực và quy định: Nghiệp vụ thường liên quan đến việc tuân thủ các chuẩn mực, quy định và hướng dẫn cụ thể của ngành nghề. Điều này có thể bao gồm các quy tắc an toàn, chuẩn mực chất lượng, quy định pháp lý và các yêu cầu khác.
* Người thực hiện: Nghiệp vụ định rõ vai trò và trách nhiệm của từng người trong tổ chức hoặc ngành nghề. Nó mô tả người cần thực hiện công việc và khả năng cần thiết cho công việc đó.
* Mục tiêu: Nghiệp vụ thường liên quan đến việc đạt được các mục tiêu cụ thể, bao gồm việc cung cấp sản phẩm hoặc dịch vụ, tối ưu hóa hiệu suất và đảm bảo tuân thủ các quy tắc và quy định.
  + 1. **Nghiệp vụ của Thiết kế và xây dựng ứng dụng quản lý nhà hàng bằng Java**
* Nghiệp vụ của đề tài "Thiết kế và xây dựng ứng dụng quản lý nhà hàng bằng Java" là phát triển một ứng dụng máy tính sử dụng ngôn ngữ lập trình Java để quản lý các hoạt động của một nhà hàng.
* Nhiệm vụ:
* Phát triển mã nguồn ứng dụng sử dụng ngôn ngữ lập trình Java.
* Tạo một giao diện dành cho nhân viên nhà hàng để quản lý hệ thống.
* Tạo các chức năng quản lý như thêm, sửa đổi và xóa
* Kết nối với cơ sở dữ liệu để lưu trữ thông tin
* Thiết kế giao diện

# **CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ**

## **2.1. Phân tích yêu cầu nghiệp vụ**

### 2.1.1. Quản lý đăng kí

|  |  |
| --- | --- |
| Chức năng | Đăng kí |
| Nội dung | Cho phép đăng kí tài khoản |
| Cách xử lý | 1.Điền các thông tin hệ thống yêu cầu để đăng kí tài khoản  2.Hệ thống sẽ kiểm tra dữ liệu, nếu bị trùng thông tin sẽ thông báo đăng kí thất bại, ngược lại đăng kí thành công |
| Kết quả | Đăng kí thất bại hoặc đăng kí thành công |

### 2.1.2. Quản lý đăng nhập

|  |  |
| --- | --- |
| Chức năng | Đăng nhập |
| Nội dung | Cho phép đăng nhập tài khoản |
| Cách xử lý | 1.Điền các thông tin hệ thống yêu cầu để đăng nhập tài khoản  2.Hệ thống sẽ kiểm tra dữ liệu, nếu thành công sẽ chuyển sang trang quản lý, ngược lại thông báo đăng nhập thất bại |
| Kết quả | Đăng nhập thất bại hoặc đăng nhập thành công |

### 2.1.3. Quản lý danh mục món ăn

|  |  |
| --- | --- |
| Chức năng | Quản lý danh mục |
| Nội dung | Quản lý danh mục món ăn |
| Cách xử lý | 1.Thêm danh mục món ăn  2.Sửa danh mục món ăn  3.Xóa danh mục món ăn |
| Kết quả | Hiển thị đầy đủ và chính xác các thông tin của danh mục món ăn |

### 2.1.4. Quản lý thực đơn

|  |  |
| --- | --- |
| Chức năng | Quán lý thực đơn |
| Nội dung | Quản lý các món ăn trong thực đơn |
| Cách xử lý | 1.Thêm món ăn  2.Sửa món ăn  3.Xóa món ăn  4.Tìm món ăn |
| Kết quả | Hiển thị đầy đủ và chính xác các thông tin của món ăn |

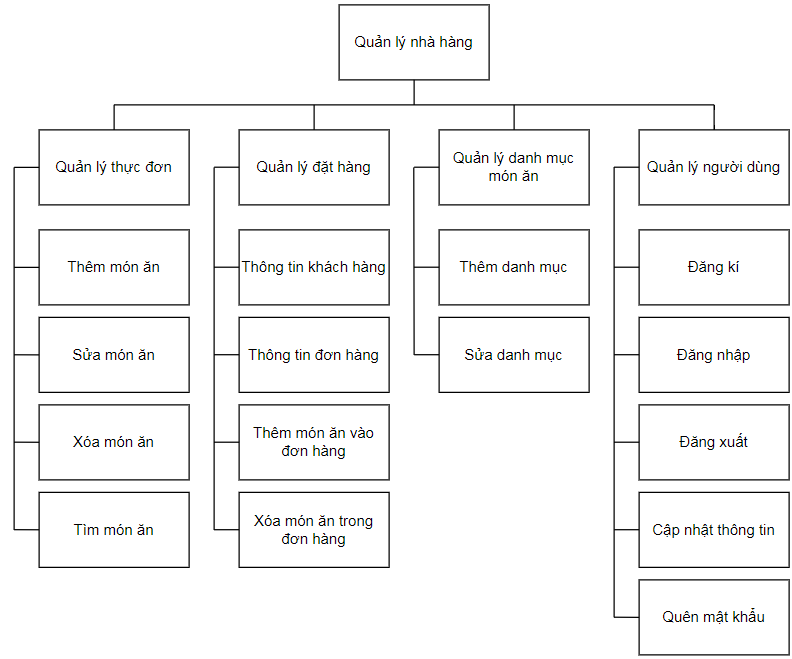
### 2.1.5. Quản lý đặt hàng

|  |  |
| --- | --- |
| Chức năng | Đặt hàng |
| Nội dung | Thông tin về đơn hàng |
| Cách xử lý | 1.Thông tin khách hàng  2.Thông tin đơn hàng  3.Thêm món ăn vào đơn hàng  4.Xóa món ăn ra đơn hàng  5.Sửa đơn hàng |
| Kết quả | Tiếp nhận và xử lý đơn đặt hàng một cách hiệu quả  Tổng số tiền được tính chính xác và có sẵn để thanh toán |

### 2.1.6. Quản lý người dùng

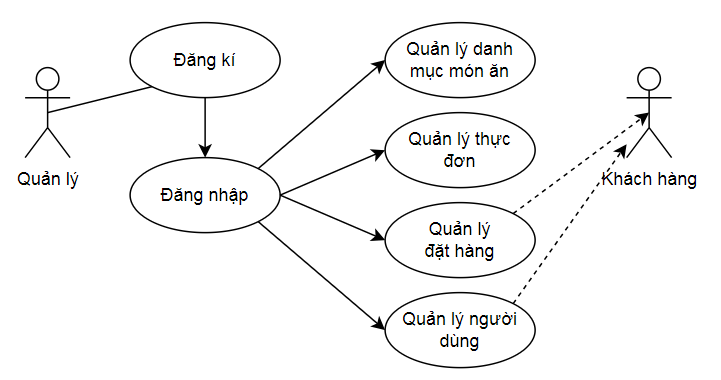
|  |  |
| --- | --- |
| Chức năng | Quản lý người dùng |
| Nội dung | Thông tin người dùng |
| Cách xử lý | 1.Đăng kí  2.Đăng nhập  4.Đăng xuất  5.Cập nhật thông tin  6.Quên mật khẩu |
| Kết quả | Cập nhật và hiển thị thông tin về người dùng |

## **2.2. Thiết kế biểu đồ chức năng**

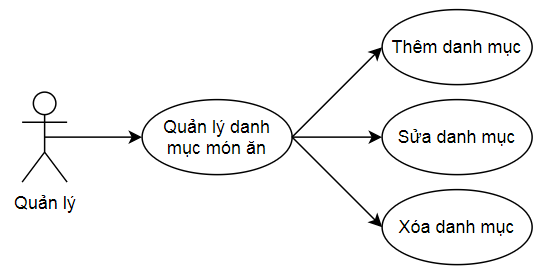


## **2.3. Thiết kế biểu đồ UseCase**

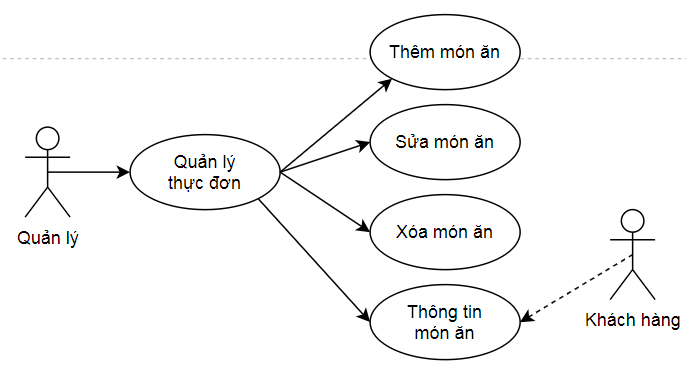
### 2.3.1. Biểu đồ UseCase tổng quát



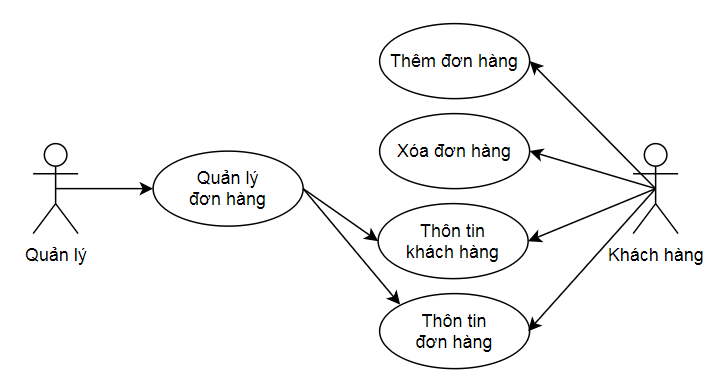
### 2.3.2. Biểu đồ UseCase Quản lý danh mục món ăn



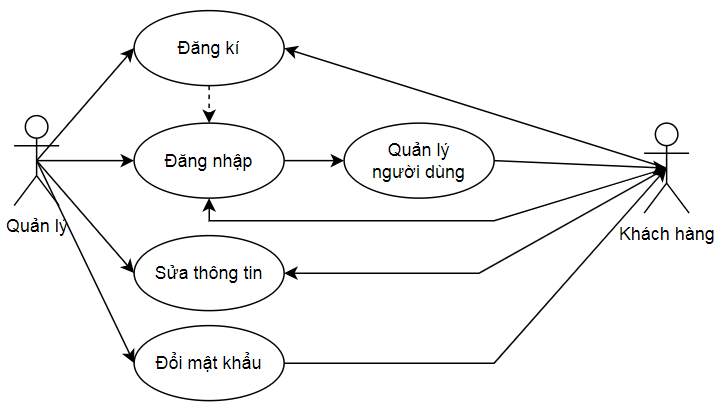
### 2.3.3. Biểu đồ UseCase Quản lý thực đơn



### 2.3.4. Biểu đồ UseCase Quản lý đặt hàng

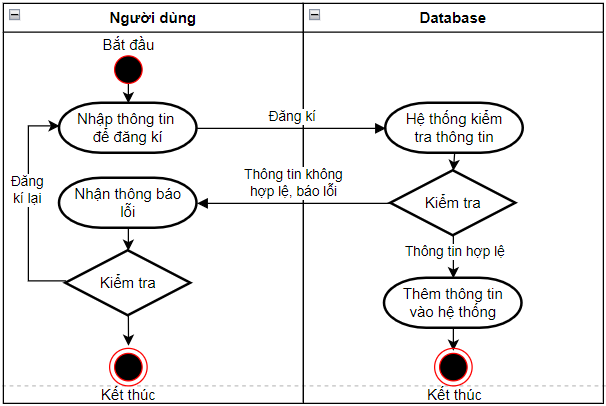


### 2.3.5. Biểu đồ UseCase Quản lý người dùng

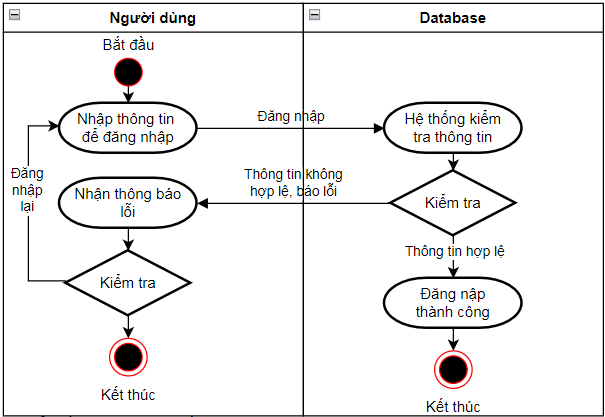


## **2.4. Thiết kế biểu đồ hoạt động các Case (Activity Diagrams)**

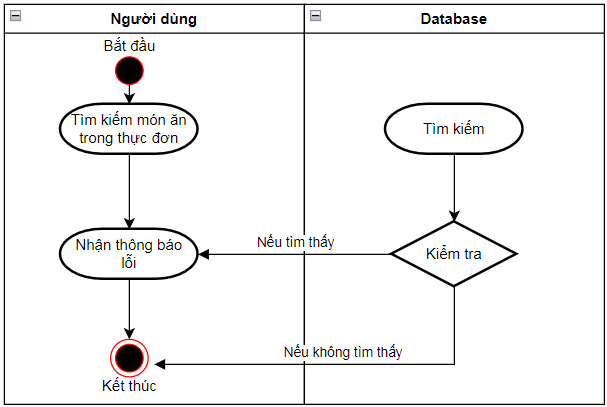
### 2.4.1. Biểu đồ hoạt động đăng kí



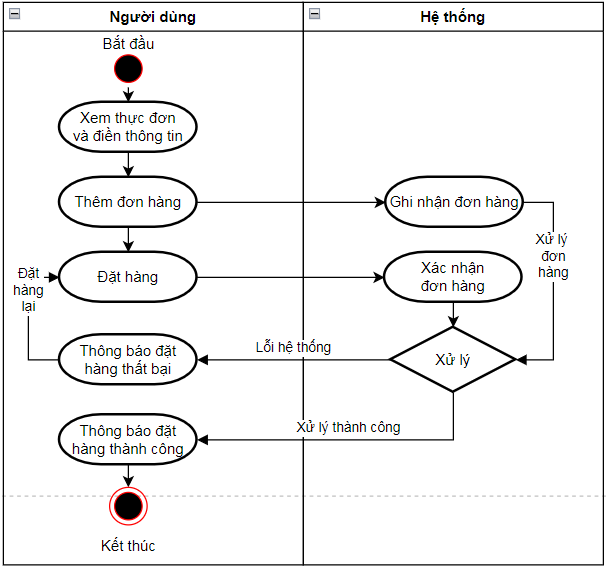
### 2.4.2. Biểu đồ hoạt động đăng nhập



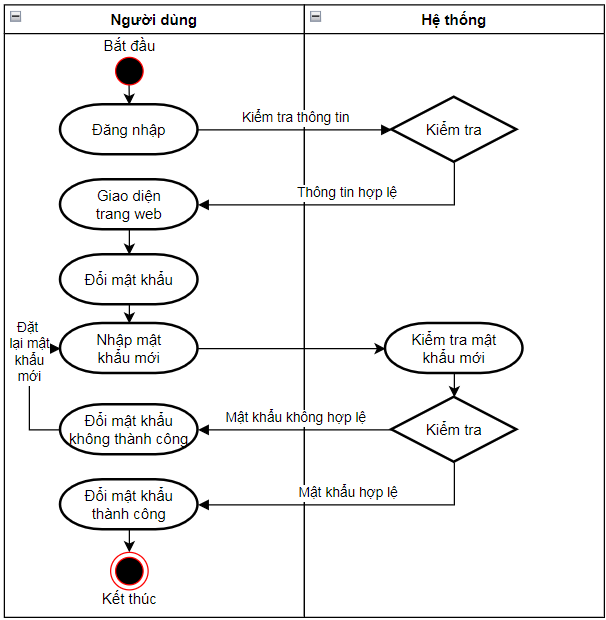
### 2.4.3. Biểu đồ hoạt động tìm kiếm



### 2.4.4. Biểu đồ hoạt động đặt hàng

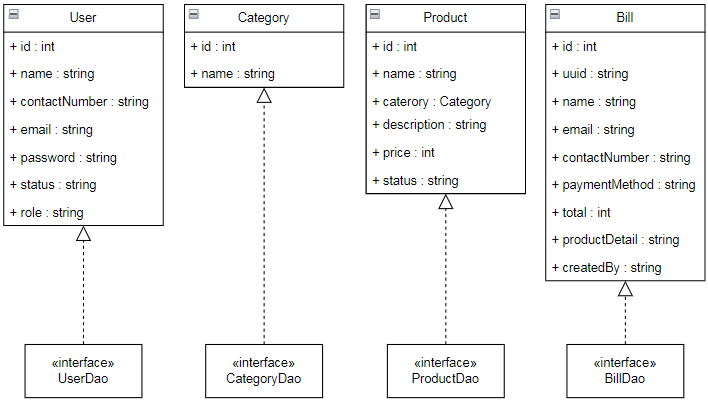


### 2.4.5. Biểu đồ hoạt động đổi mật khẩu

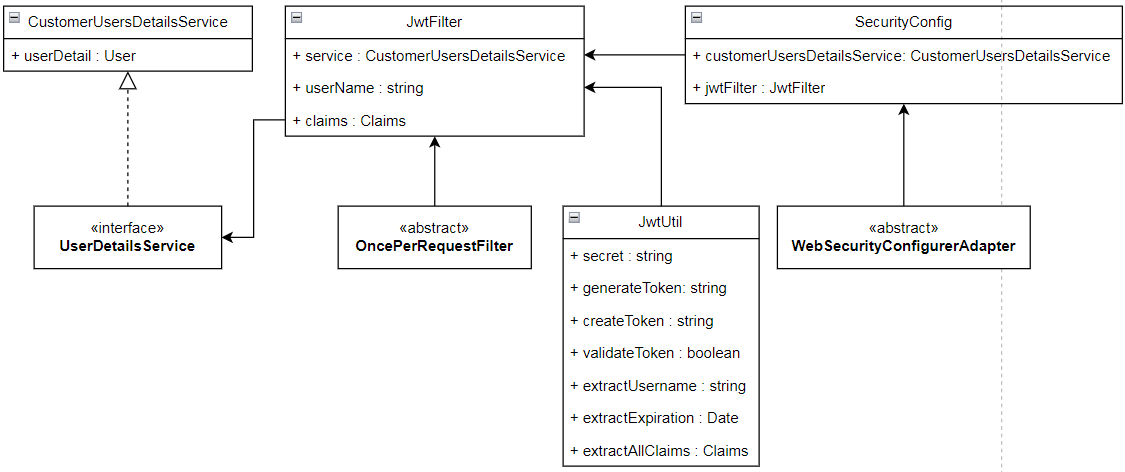


## **2.5. Thiết kế biểu đồ lớp (Class Diagram)**

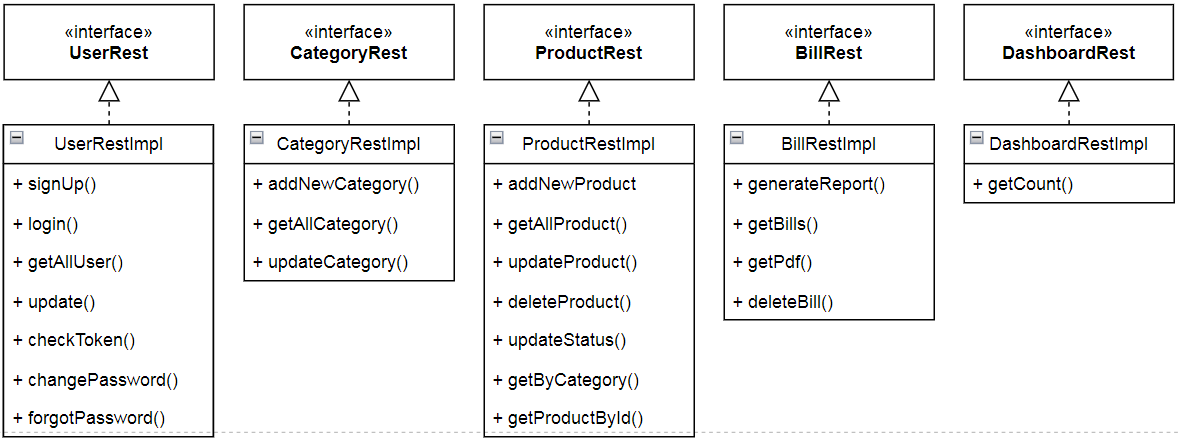
### 2.5.1. User và UserDao, Category và CategoryDao, Product và ProductDao, Bill và BillDao



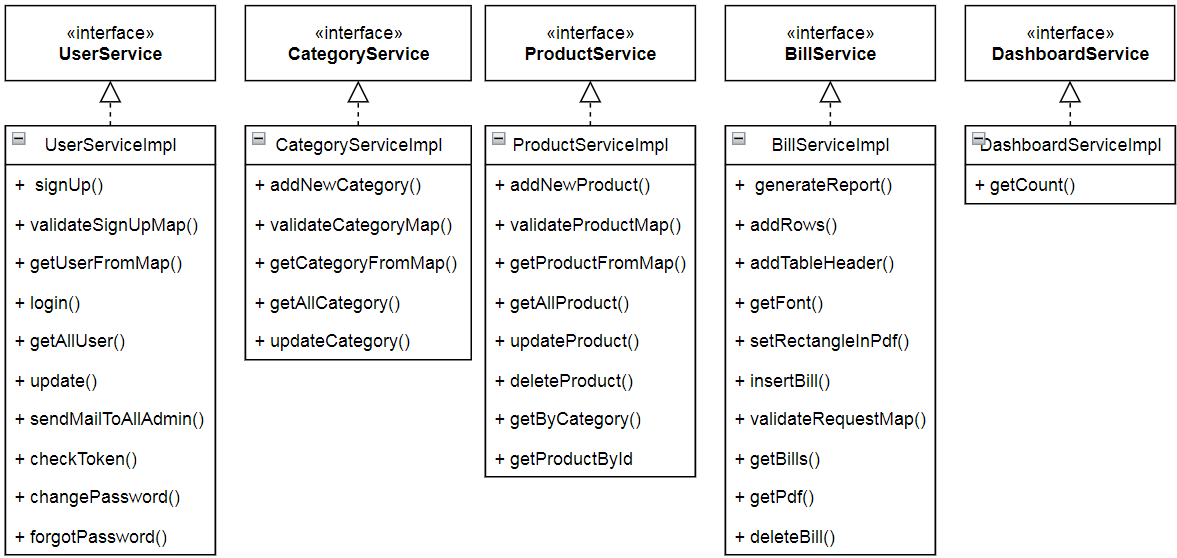
### 2.5.2. JWT



### 2.5.3. Rest và RestImpl

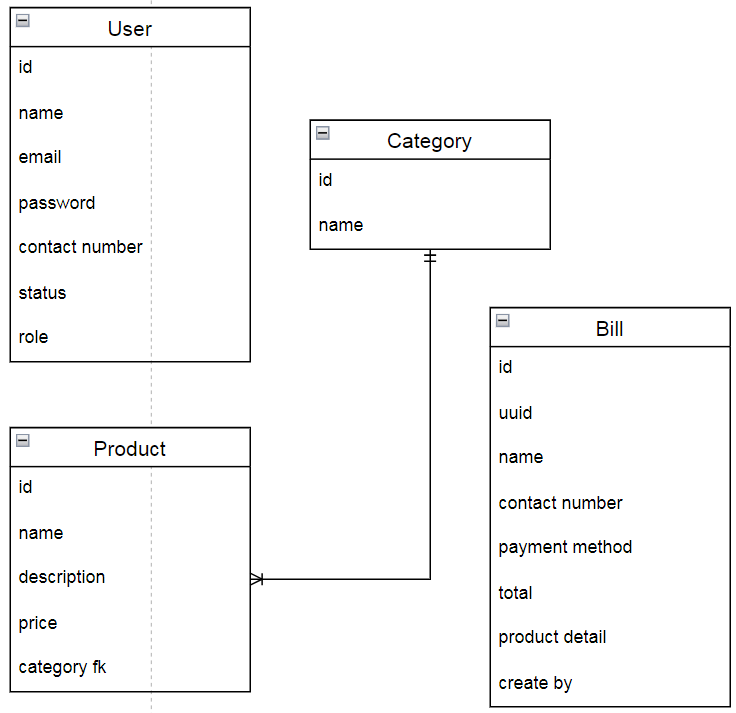


### 2.5.4. Service và ServiceImpl



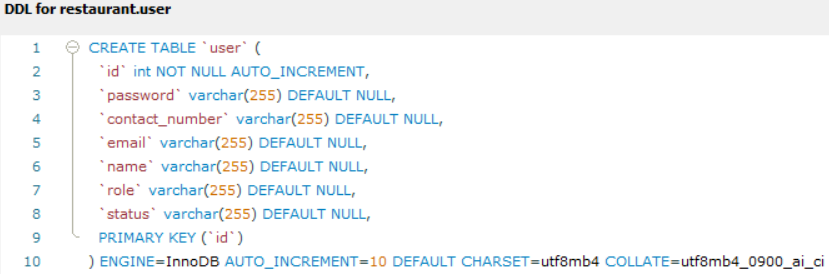
## **2.6. Thiết kế cơ sở dữ liệu**

### 2.6.1. ERD



### 2.6.2. Thiết kế chi tiết các Tables

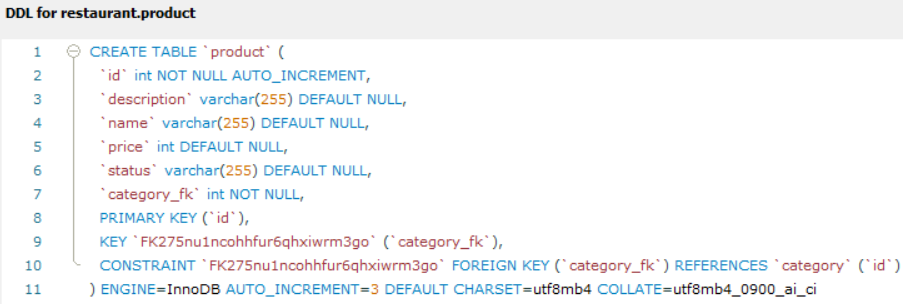
* Table User



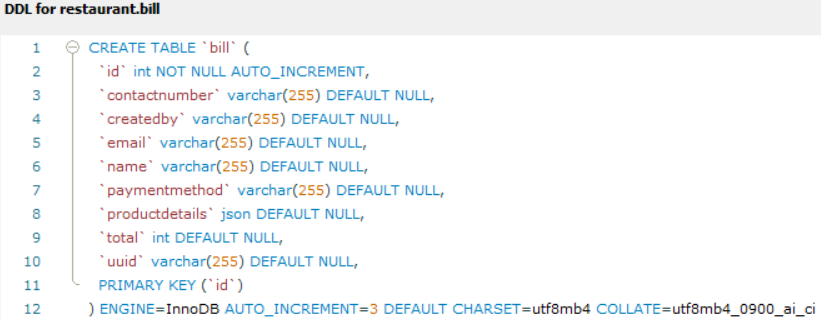
* Table Category



* Table Product

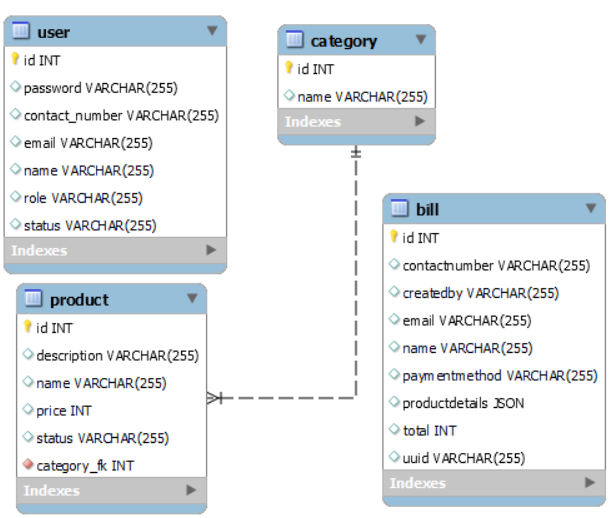


* Table Bill

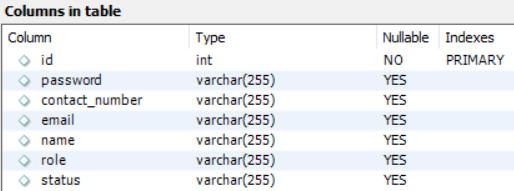


# **CHƯƠNG 3. CÀI ĐẶT THỰC NGHIỆM**

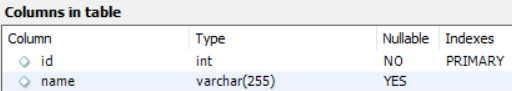
## **3.1. Cài đặt cơ sở dữ liệu**



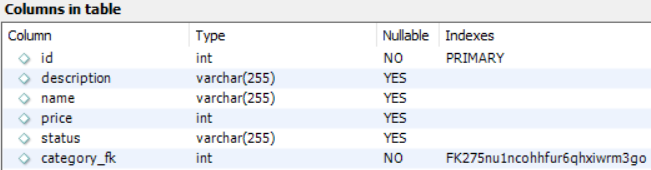
Hình 3.1. Cơ sở dữ liệu của ứng dụng được tổ chức trên hệ quản trị



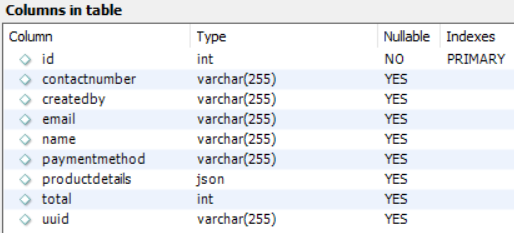
Hình 3.2. Bảng dữ liệu User



Hình 3.3. Bảng dữ liệu Category

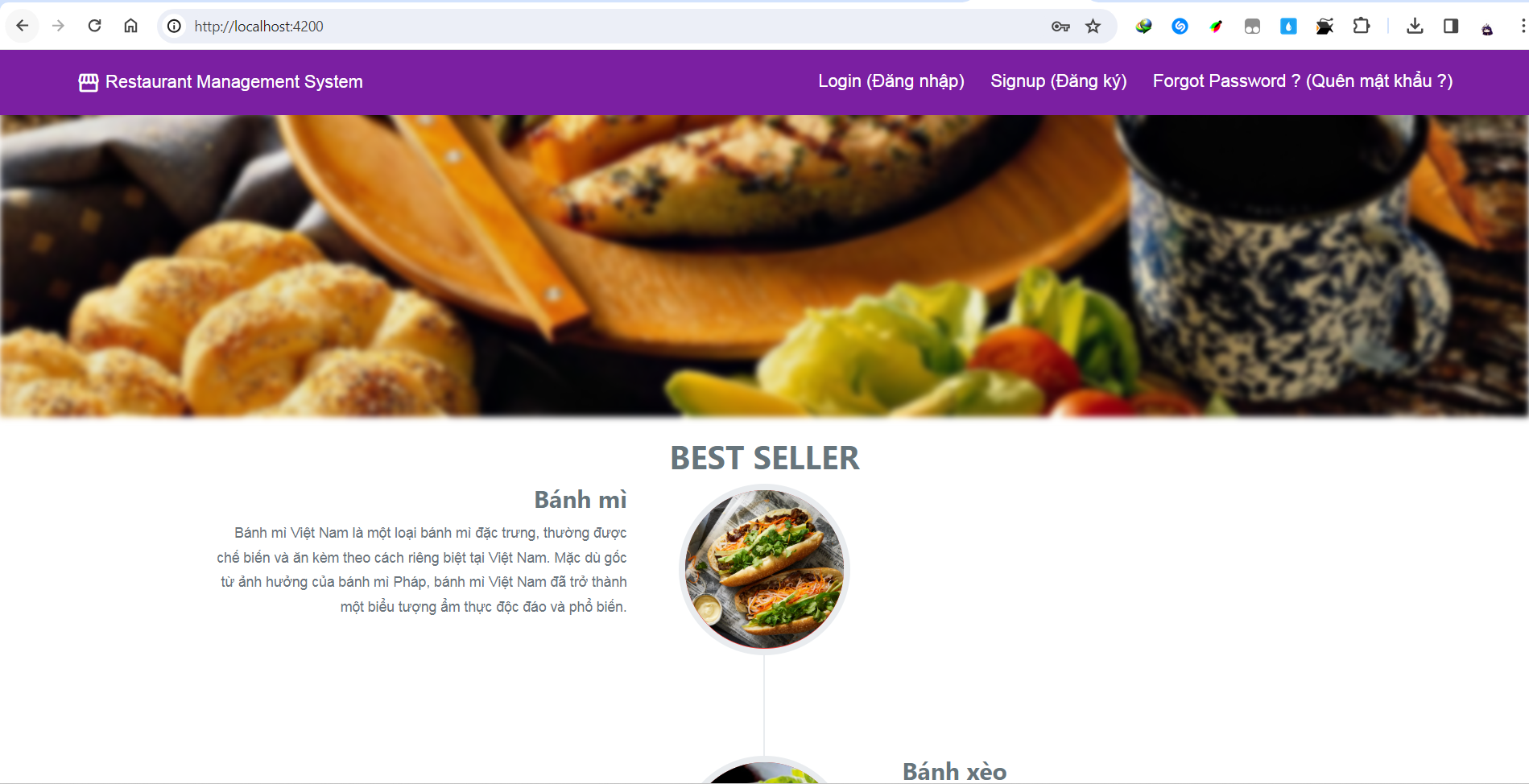


Hình 3.4. Bảng dữ liệu Product

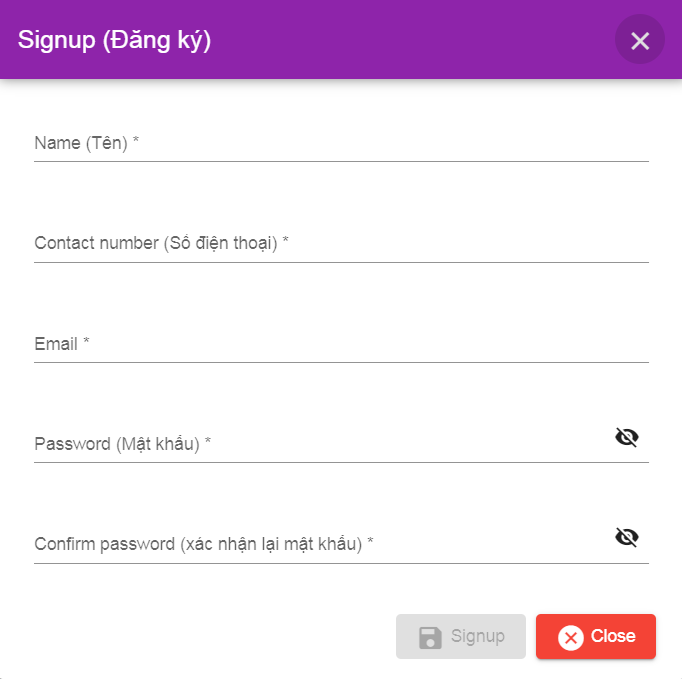


Hình 3.5. Bảng dữ liệu Bill

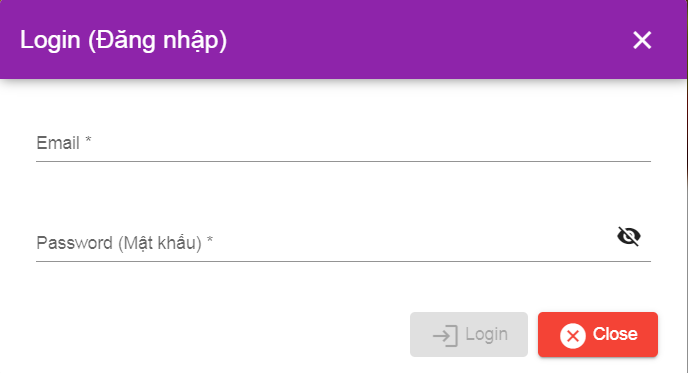
## **3.2. Cài đặt phần mềm ứng dụng**



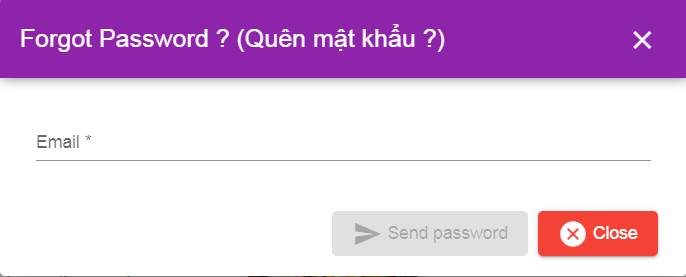
Hình 3.4. Giao diện trang chủ



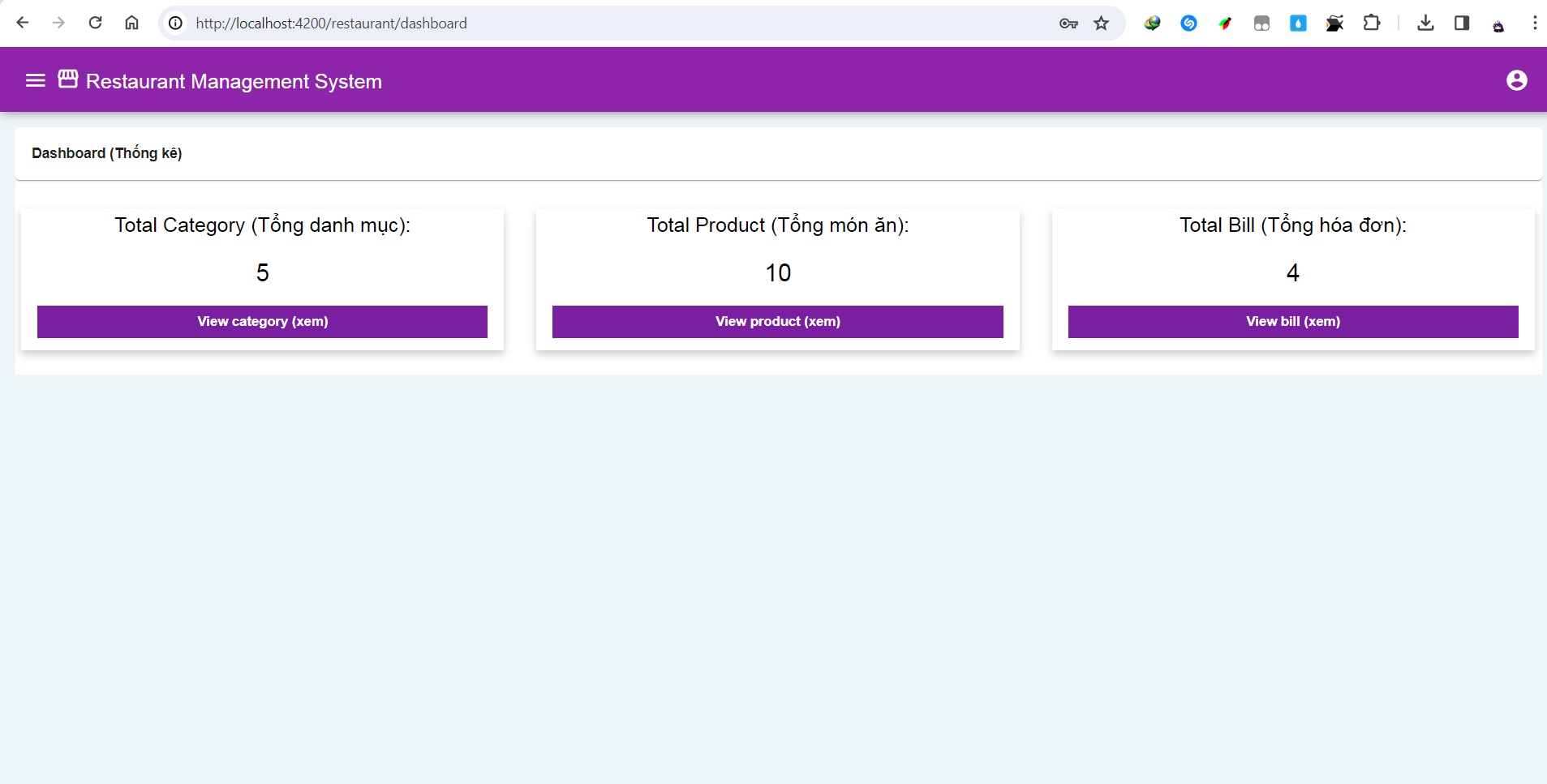
Hình 3.5. Giao diện đăng ký



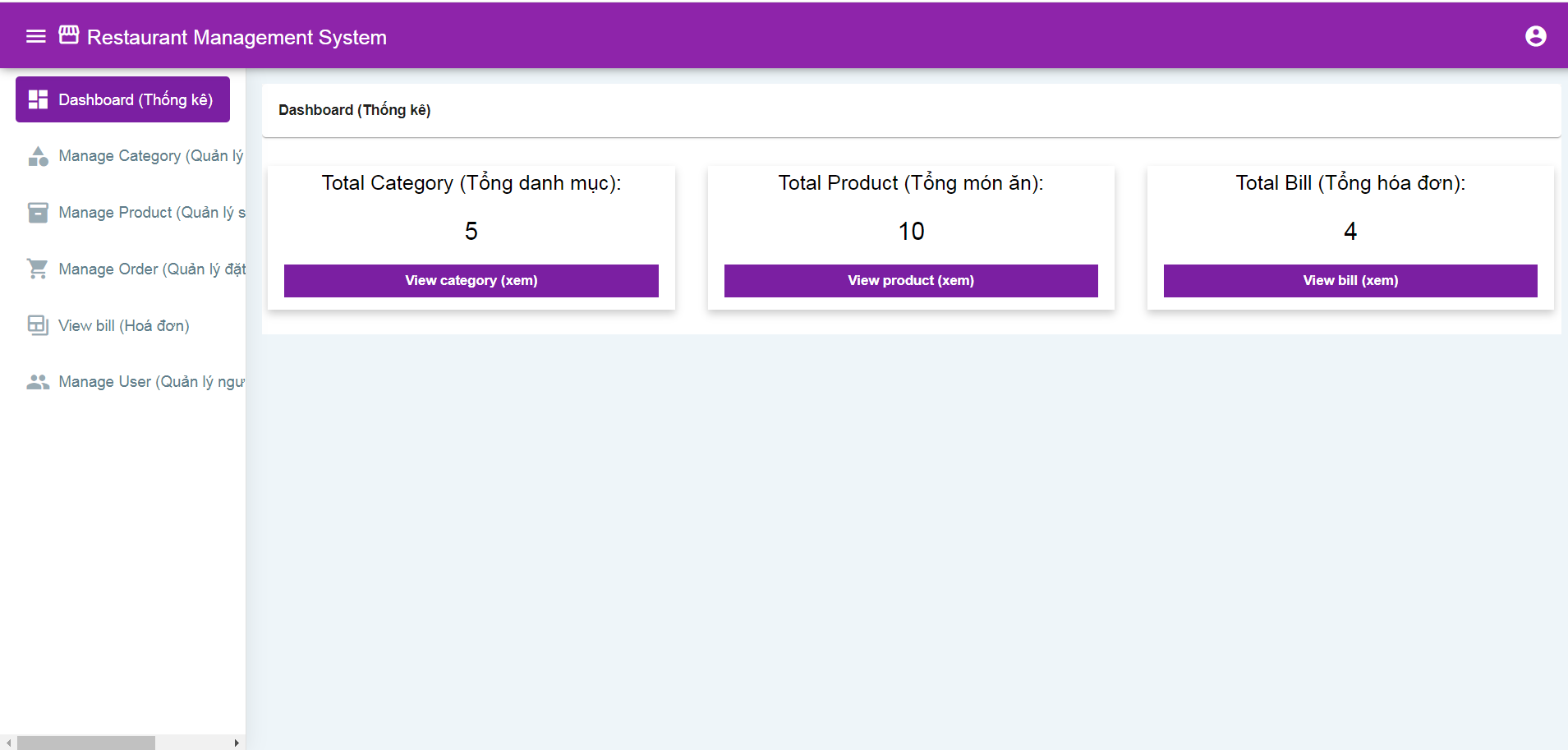
Hình 3.6. Giao diện đăng nhập



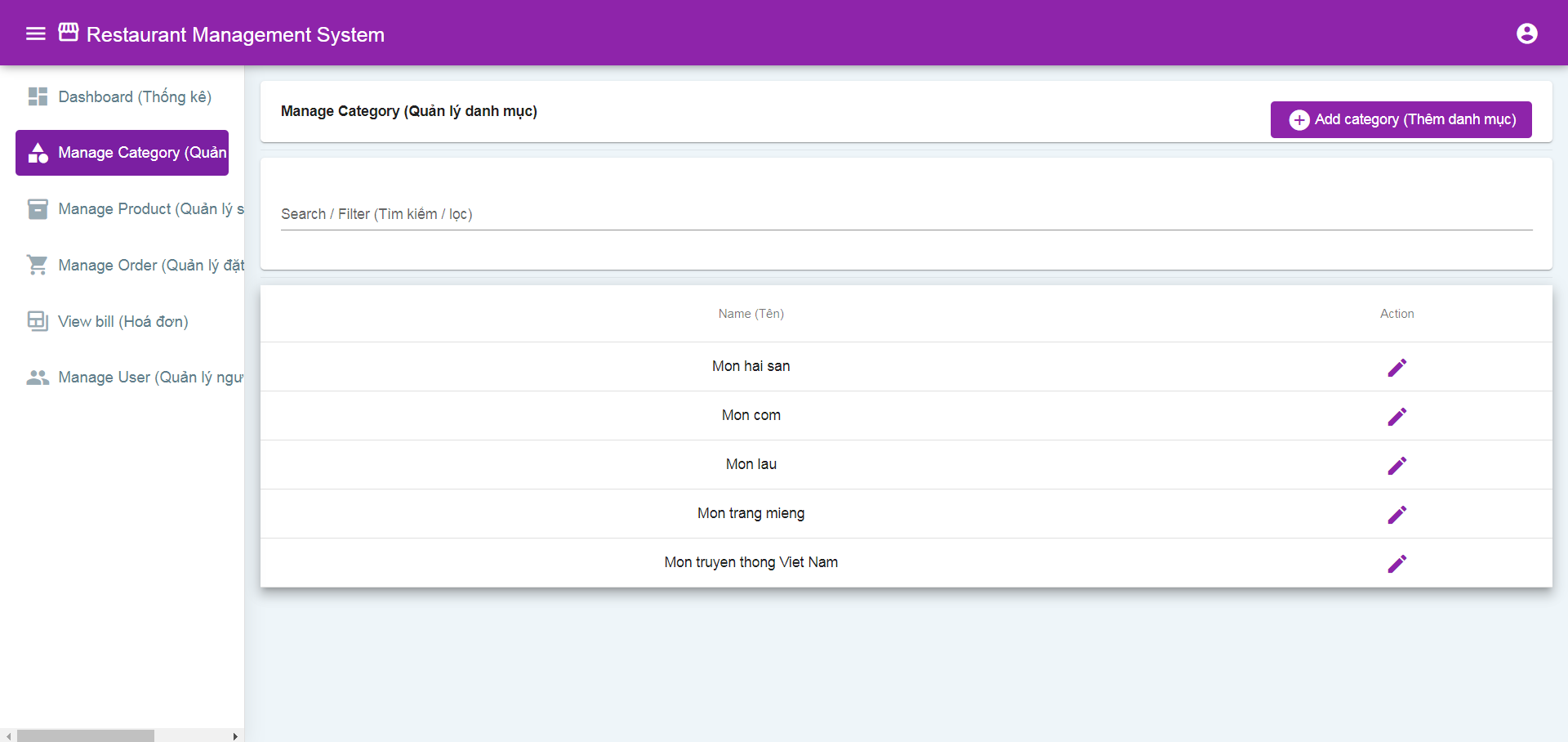
Hình 3.7. Giao diện trang quên mật khẩu



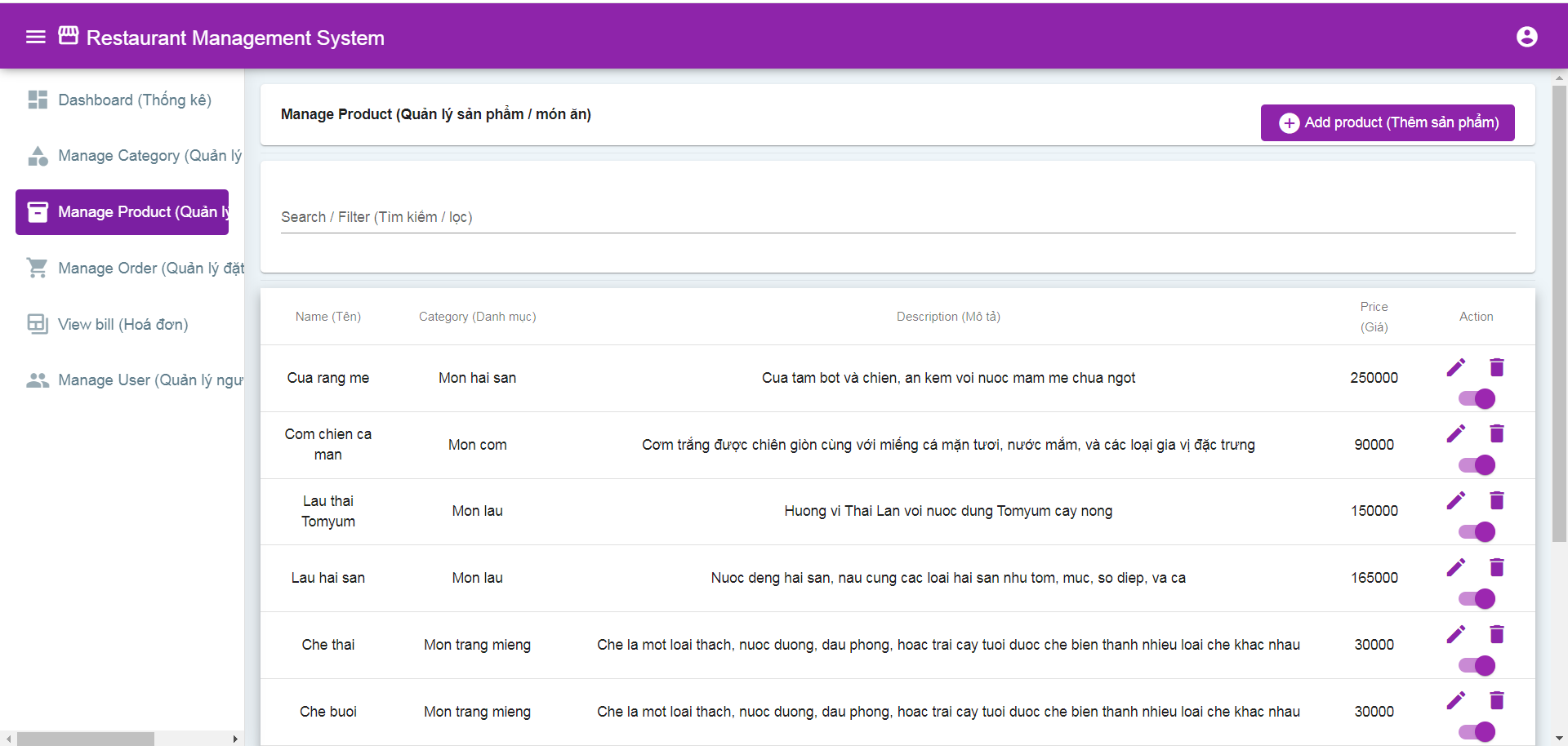
Hình 3.8. Giao diện trang quản lý



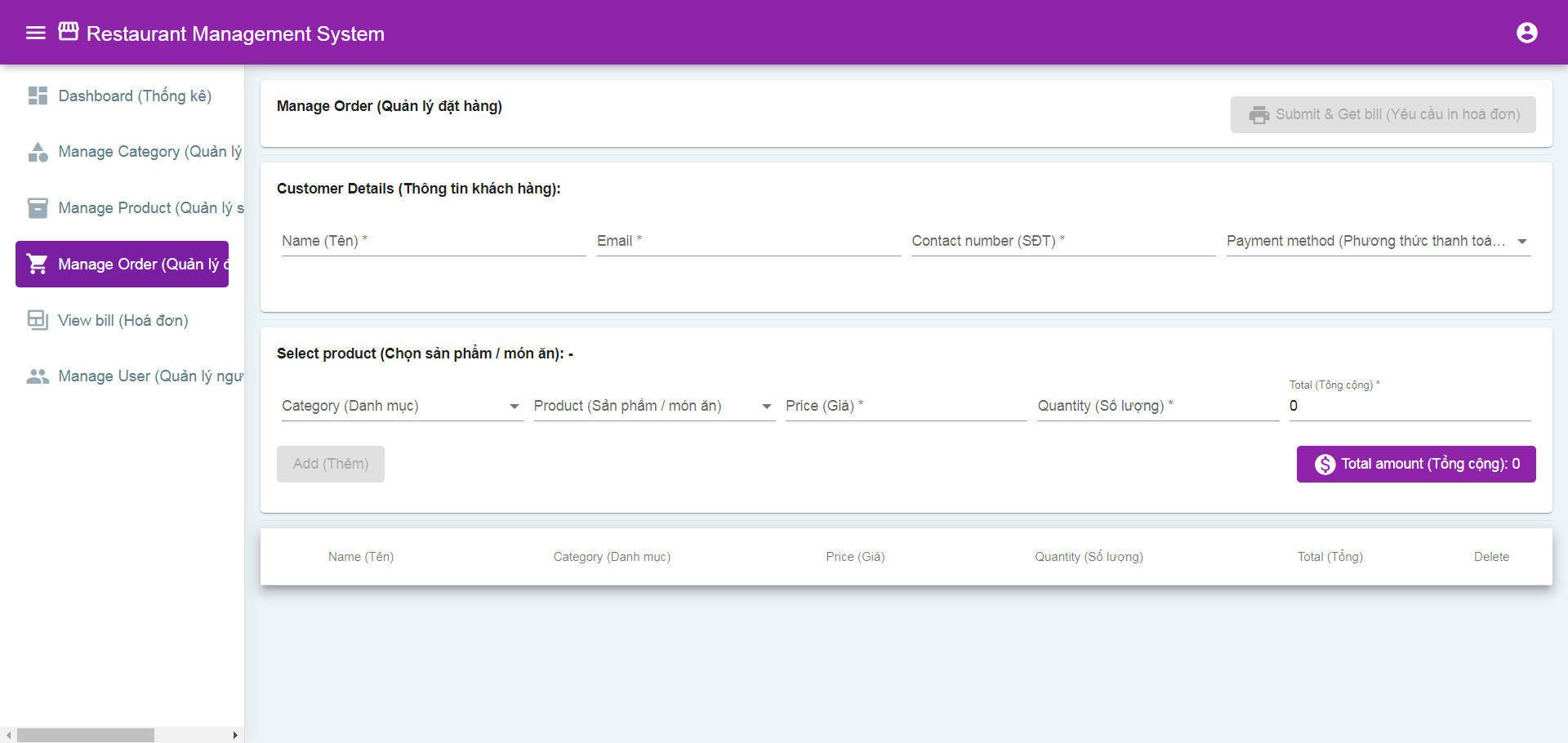
Hình 3.9. Giao diện trang thống kê



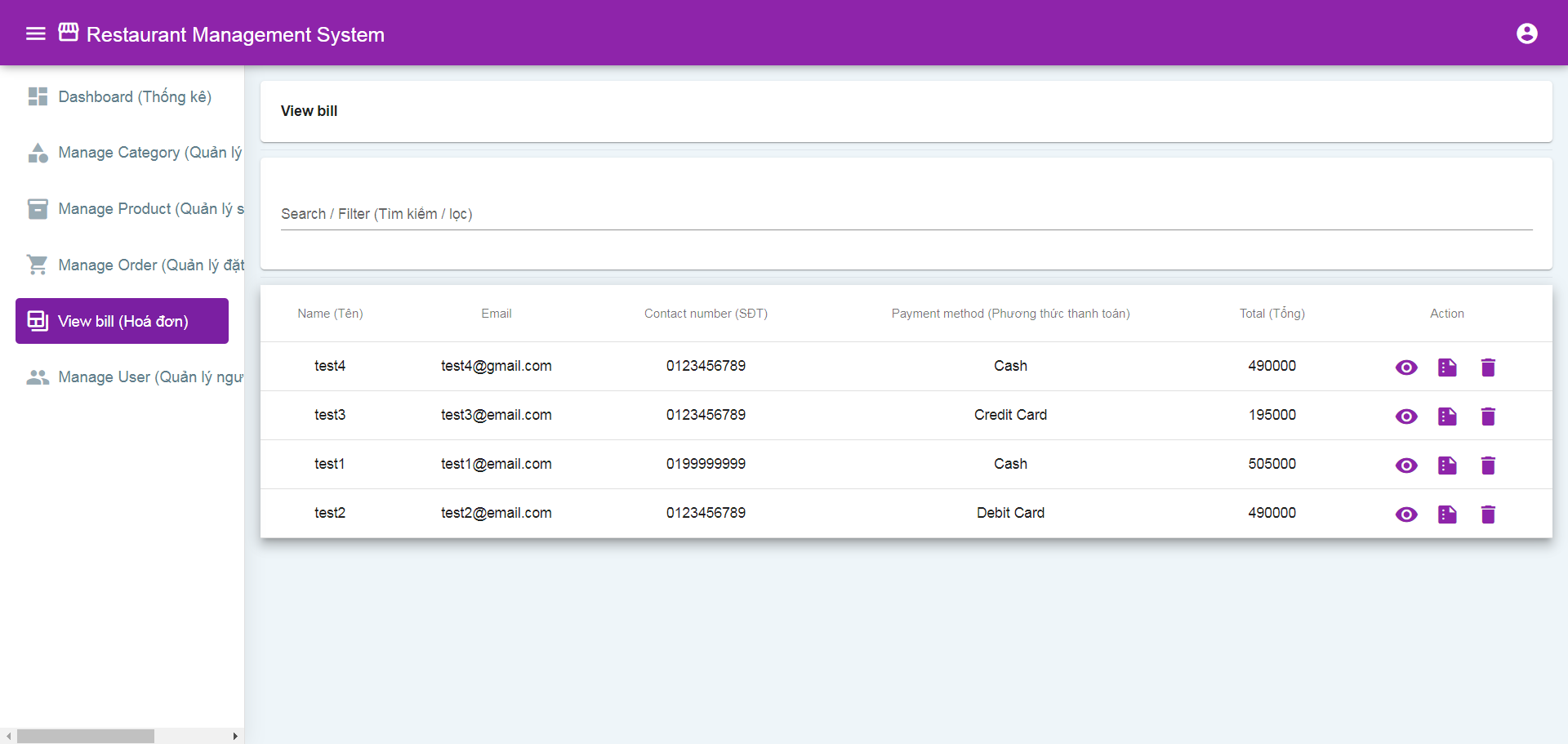
Hình 4.0. Giao diện trang quản lý danh mục



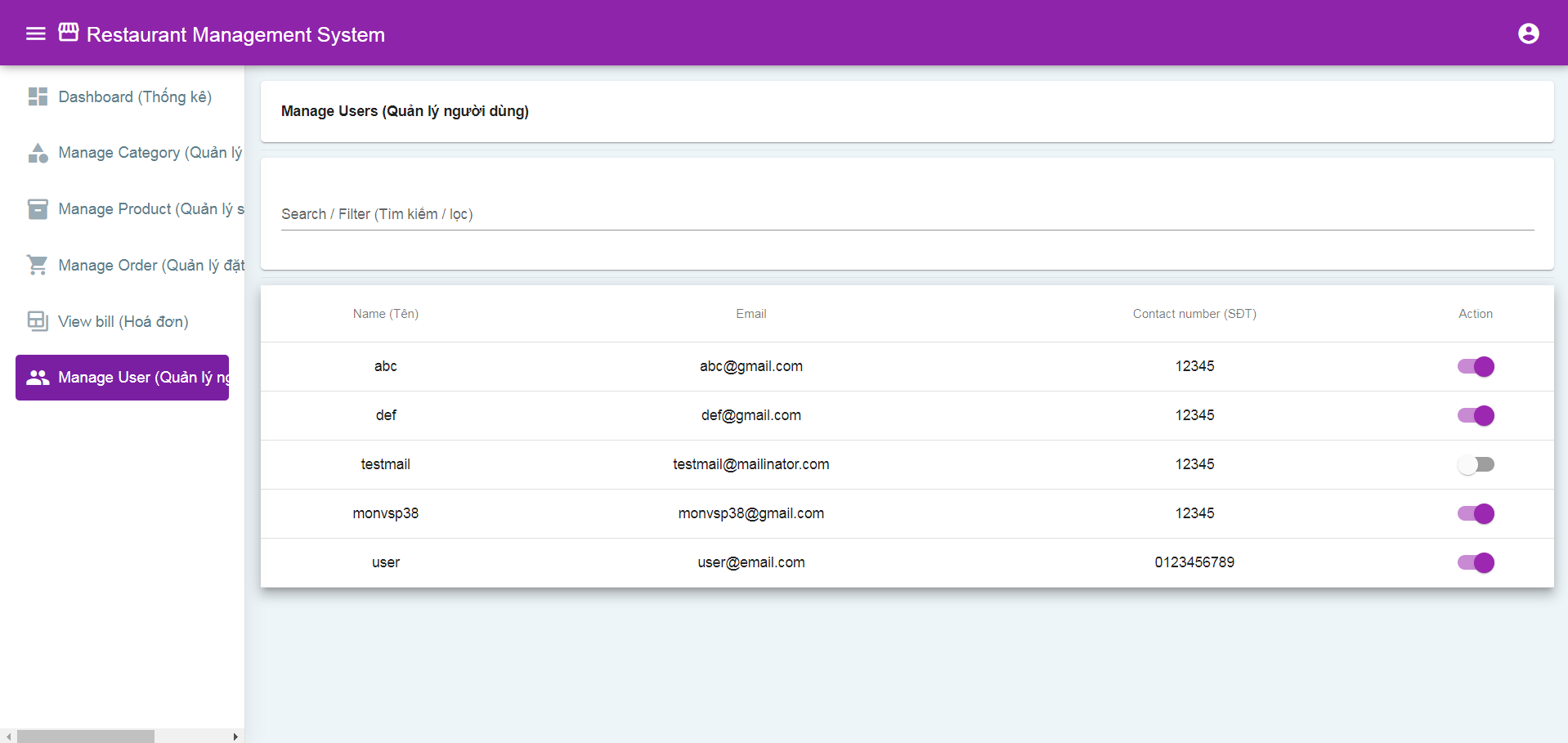
Hình 4.1. Giao diện trang quản lý sản phẩm



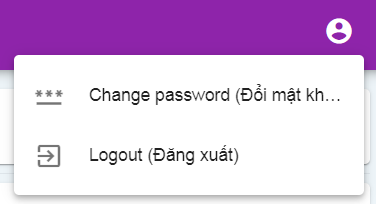
Hình 4.2. Giao diện trang quản lý đặt hàng



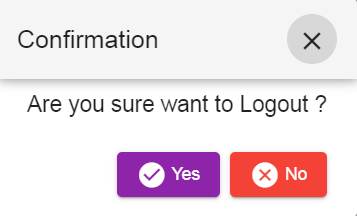
Hình 4.3. Giao diện trang hoá đơn đặt hàng



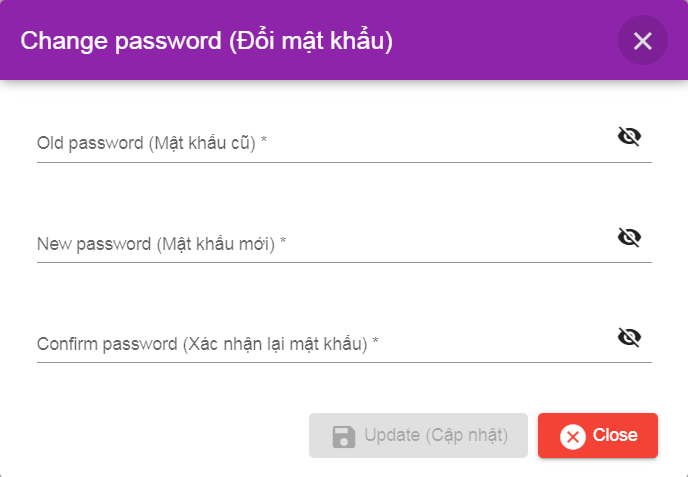
Hình 4.4. Giao diện trang quản lý người dùng



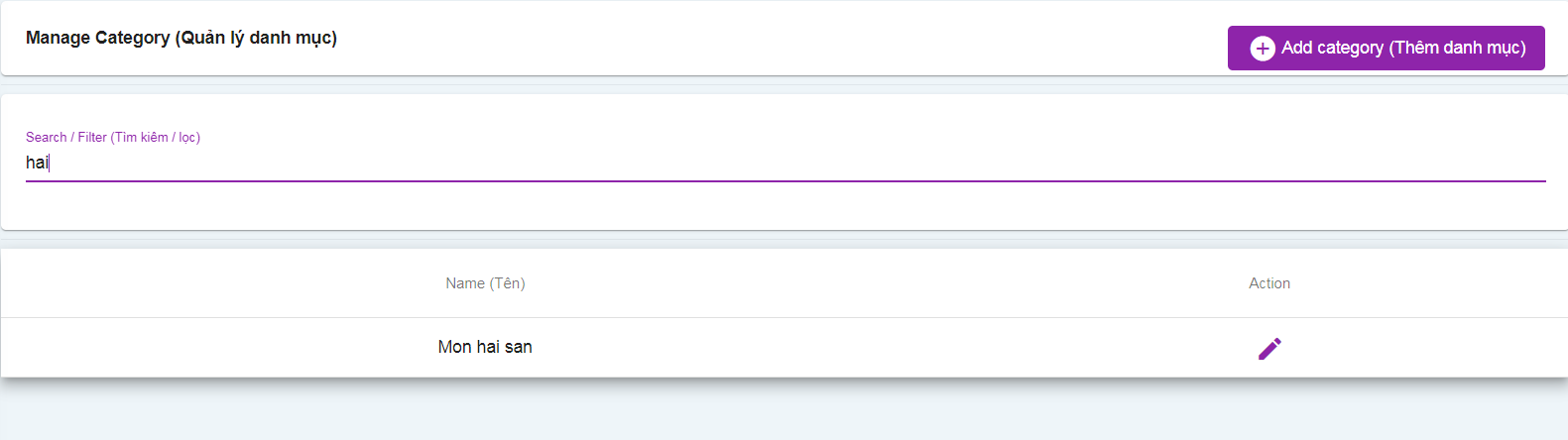
Hình 4.4. Giao diện đăng xuất và đổi mật khẩu



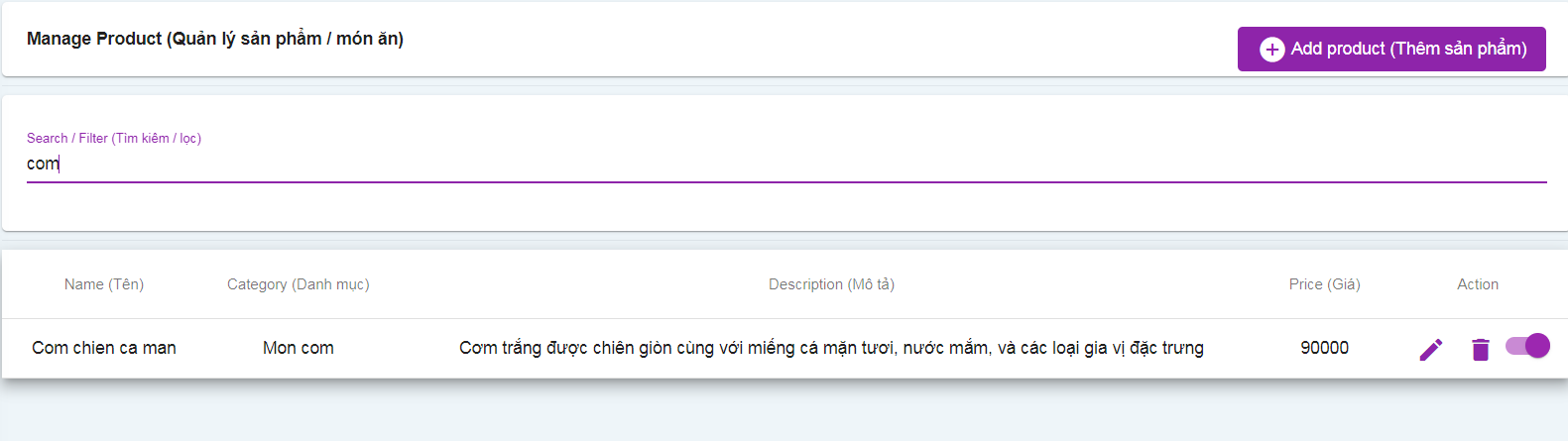
Hình 4.5. Giao diện đăng xuất



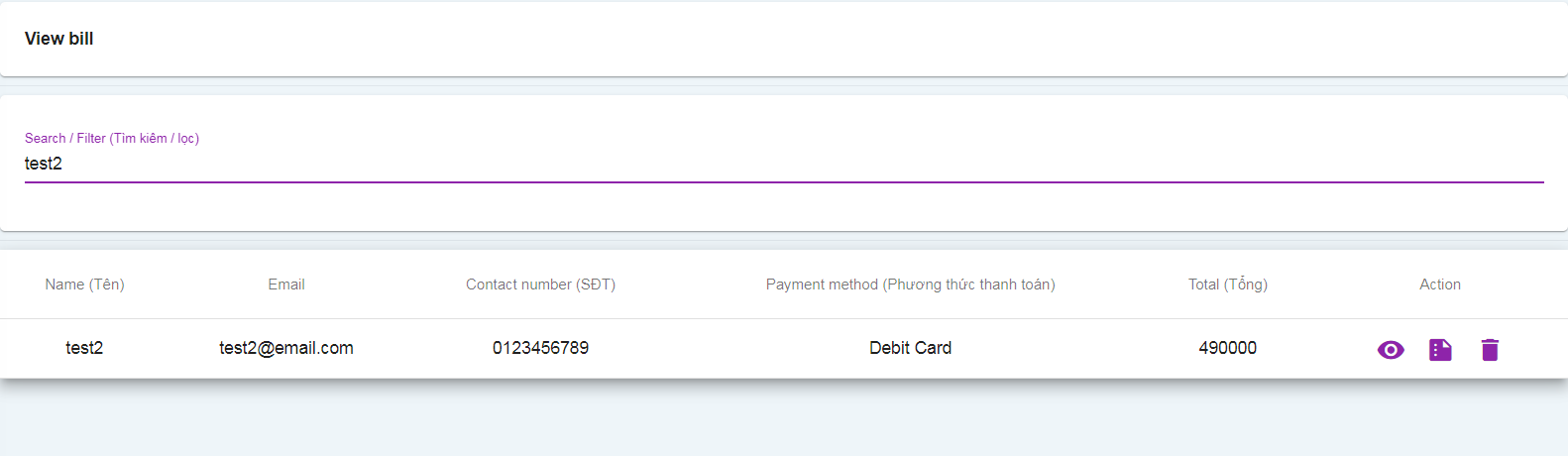
Hình 4.6. Giao diện đổi mật khẩu



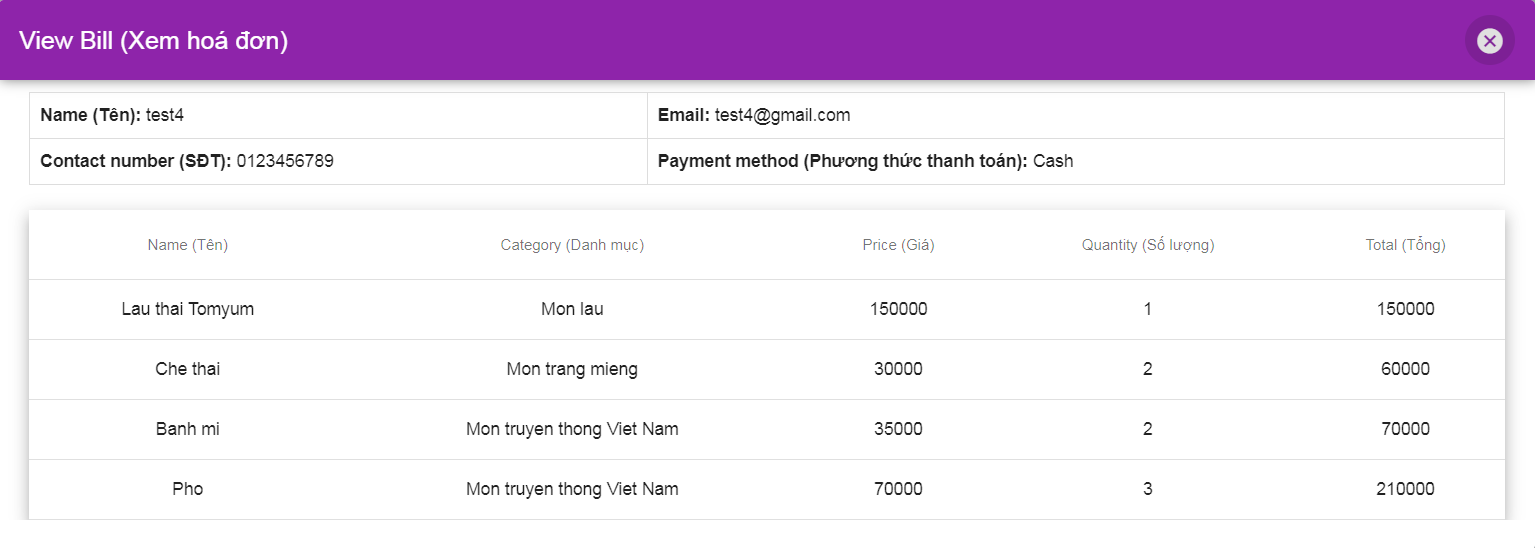
Hình 4.7. Giao diện tìm kiếm danh mục



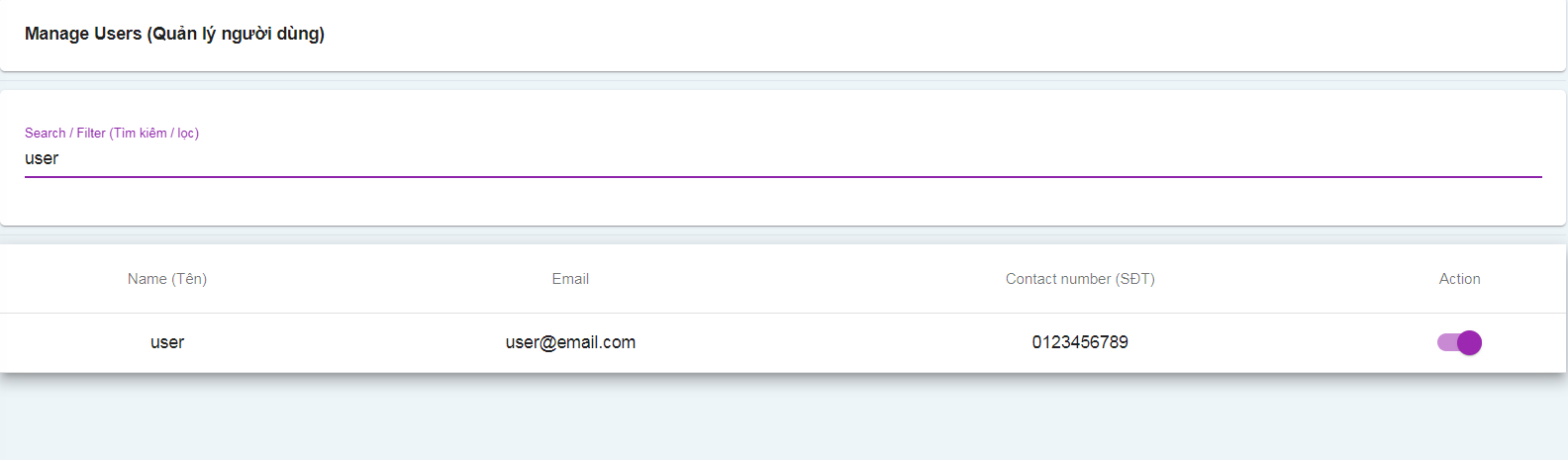
Hình 4.8. Giao diện tìm kiếm sản phẩm



Hình 4.9. Giao diện tìm kiếm hoá đơn



Hình 4.10. Giao diện tìm kiếm hoá đơn



Hình 4.11. Giao diện tìm kiếm người dùng

# **KẾT LUẬN**

1. **Kết quả đạt được**

* Thiết kế thành công phần mềm có giao diện đơn giản, trực quan, dễ tiếp cận.
* Xây dựng được các chức năng cơ bản của một phần mềm quản lý.
* Nắm được cơ bản các kiến thức về Java Spring Boot, Angular, MySQL,…
* Mô hình hóa các tính năng quản lý nhà hàng như:
* Sử dụng Spring Boot để xây dựng ứng dụng Java và Hibernate làm framework ORM (Object-Relational Mapping) để tương tác với cơ sở dữ liệu MySQL.

1. **Hạn chế**

* Cơ sở dữ liệu còn ít và đơn giản: có thể không đáp ứng được nhu cầu ngày càng tăng về lưu trữ và truy xuất dữ liệu.
* Chưa tối ưu được Code chương trình: code hiện tại có thể không được viết tối ưu, dẫn đến tình trạng chậm và tăng tiêu tốn tài nguyên hệ thống.
* Còn thiếu nhiều chức năng quản lý khác
* Giao diện người dùng: còn sơ sài và đơn giản

1. **Đề xuất**

* Xây dựng và phát triển cơ sở dữ liệu
* Xây dựng và thiết kế các biểu đồ nhằm giúp cho việc thống kê trở nên trực quan hơn.
* Thêm các chức năng quản lý khác như: Quản Lý Nhân Viên, Quản lý Bàn Ăn, Quản Lý Khách Hàng, Quản Lý Nguyên Liệu và Kho, Quản Lý Đặt Chỗ, …
* Tối ưu hoá đoạn code để chương trình chạy nhanh hơn
* Tăng cường bảo mật hệ thống để đảm bảo an toàn thông tin của người dùng và dữ liệu quan trọng.
* Thực hiện quản lý lỗi định kỳ và kiểm thử bảo mật để phát hiện và khắc phục những lỗ hổng có thể gặp.
* Tích hợp các công nghệ mới, cụ thể là các cải tiến trong Java Spring Boot, Angular, MySQL để cải thiện hiệu suất và tính năng của trang website.

# **TÀI LIỆU THAM KHẢO**

[1] Đặng Văn Đức, Phân tích thiết kế hướng đối tượng bằng UML (Thực hành với Rational Rose), NXB Khoa học và Kỹ thuật, 2002

[2] Nguyễn Văn Vy, Phân tích và thiết kế hệ thống thông tin quản lý, NXB khoa học tự nhiên và công nghệ, 2007.

[3] Lê Văn Phùng, Cơ sở dữ liệu quan hệ và công nghệ phân tích thiết kế, NXB thông tin và truyền thông, 2010.

[4] Craig Walls ,Spring Boot in Action, Manning Publications, 2015

[6] Yakov Fain & Anton Moiseev , Angular Development with Typescript, Manning Publications, 2018

[7] Gion Kunz, Mastering Angular: A hands-on guide to architecting, designing, and building your own Angular applications, Packt Publishing, 2019

[8] Thorben Janssen, Spring Data JPA: Getting Started and Query Methods, Thorben Janssen, 2020

[9] Karthik Appigatla, MySQL 8 Cookbook, Packt Publishing, 2018

[10] Vũ Tiến Linh, Lập trình Java với Spring Boot, NXB Thông tin và Truyền thông, 2020

[11] Vũ Đức Tâm, Angular 2+ - Học cùng Examples, NXB Thông tin và Truyền thông, 2017

[12] Lê Việt Khánh, Lập trình web với AngularJS và Spring Boot, NXB Thông tin và Truyền thông, 2016

[13] Trần Đức Minh, MySQL - Hệ quản trị cơ sở dữ liệu phổ biến, NXB Thống kê, 2015

[14] Trần Anh Minh, Spring Framework - Tổng quan về Spring Framework và mô hình MVC, NXB Thông tin và Truyền thông, 2015

[15] John Doe, Postman: A Comprehensive Guide to API Testing, Postman, Inc., 2020

[16] Jane Smith ,Mastering IntelliJ IDEA: A Comprehensive Guide, JetBrains, sáng tạo viên của IntelliJ IDEA, 2019

[17] Michael Johnson, Mastering Visual Studio Code: Tips and Tricks for Productive Development, Microsoft Corporation, 2020

[18] Leonard Richardson, Sam Ruby, RESTful Web Services, O'Reilly Media, 2007

[19] Matthias Biehl. RESTful API Design (API-University Series), CreateSpace Independent Publishing Platform, 2013

[20] Trần Duy Thanh, Tìm hiểu và thực hành MySQL, Công ty TNHH Thương mại và Dịch vụ Tin học Hoàng Việt, 2017

[21] Jeffrey Zeldman, Ethan Marcotte, Designing with Web Standards, New Riders, 2009