

TRƯỜNG ĐẠI HỌC QUY NHƠN
KHOA CÔNG NGHỆ THÔNG TIN

-----○○○●○○-----



BÁO CÁO BÀI TẬP NHÓM

HỌC PHẦN: KIẾN TRÚC HƯỚNG DỊCH VỤ

TÊN ĐỀ TÀI: Sử dụng kiến trúc SOA để phân tích, xây dựng và quản lý siêu thị mini

NHÓM SINH VIÊN THỰC HIỆN

| TT | Họ và tên | MSSV |
|----|----------------------|------------|
| 1 | Trương Thanh Thu Huệ | 4451050838 |
| 2 | Trần Thị Thanh Kiều | 4451051111 |
| 3 | Hán Thị Triều | 4451051008 |

GVHD: ThS. Trần Hoàng Việt

Bình Định, tháng 01, năm 2025

MỤC LỤC

| | |
|---|-----------|
| 1. Bài toán..... | 3 |
| 1.1 Phát biểu bài toán:..... | 3 |
| 2.1 Xác định mục tiêu của hệ thống | 3 |
| 2.1.1 Các yêu cầu chức năng..... | 3 |
| 2.2 Xây dựng biểu đồ chức năng dựa trên mục tiêu của hệ thống | 5 |
| 2.3 Phân rã chức năng con (dịch vụ) dựa trên phân rã mục tiêu con của hệ thống | 5 |
| 2.3.1 Quản lý người dùng(Port 8000):..... | 5 |
| 2.3.2 Quản lý sản phẩm(Port 8001):..... | 6 |
| 2.3.3 Quản lý đơn hàng (Port 8002)..... | 7 |
| 2.4 Biểu đồ luồng dữ liệu (theo chức năng)..... | 8 |
| 2.4.1 Đăng nhập, đăng ký | 8 |
| 2.4.2 Quản lý người dùng | 9 |
| 2.4.3 Quản lý sản phẩm..... | 9 |
| 3. Phân tích và thiết kế dữ liệu của hệ thống..... | 10 |
| 3.1 Mô hình thực thể liên kết của từng dịch vụ..... | 10 |
| 3.2 Mô hình quan hệ của từng dịch vụ..... | 11 |
| 3.3 Bảng dữ liệu..... | 12 |
| 4. Giao diện của hệ thống | 14 |
| 4.1 Xây dựng giao diện API cho từng dịch vụ | 14 |
| 5. Kết luận | 27 |
| 1.1 Kết quả đạt được..... | 27 |
| 1.2 Hạn chế | 27 |
| 1.3 Hướng mở rộng..... | 28 |
| 1.4 Link git..... | 28 |

1. Bài toán

1.1 Phát biểu bài toán:

Bài toán đặt ra là xây dựng một hệ thống quản lý cho một siêu thị mini. Hệ thống cần hỗ trợ quản lý các nghiệp vụ như quản lý sản phẩm, tài khoản người dùng (admin, thu ngân), thanh toán và hóa đơn quầy thu ngân.

Hệ thống sẽ quản lý các thành phần và hoạt động sau:

- Sản phẩm: Lưu trữ thông tin sản phẩm chi tiết như giá cả, tồn kho, mã sản phẩm, mô tả, hình ảnh.
- Tài khoản: Quản lý người dùng (Admin, thu ngân) với phân quyền rõ ràng.
- Thanh toán, hóa đơn: Thanh toán và lưu lại chi tiết hóa đơn.

2. Phân tích chức năng của hệ thống

2.1 Xác định mục tiêu của hệ thống

2.1.1 Các yêu cầu chức năng

Yêu cầu chức năng là các yêu cầu cụ thể liên quan đến những gì hệ thống cần thực hiện để đáp ứng nhu cầu người dùng. Các chức năng chính của hệ thống quản lý siêu thị mini có thể bao gồm:

- **Đăng ký và đăng nhập người dùng:**
 - Cho phép người dùng đăng ký tài khoản mới với các thông tin như tên đăng nhập, mật khẩu, email.
 - Cho phép người dùng đăng nhập vào hệ thống sử dụng tên đăng nhập và mật khẩu.
 - Hệ thống kiểm tra thông tin đăng nhập và phân quyền truy cập dựa trên vai trò của người dùng (Employee, Admin).
- **Quản lý người dùng:**
 - Xóa người dùng
 - Tìm người dùng theo tên, vai trò.
- **Quản lý sản phẩm:**
 - Quản lý thông tin sản phẩm (Mã sản phẩm, tên sản phẩm, giá, số lượng tồn kho, mô tả,).
 - Thêm, sửa, xóa sản phẩm trong hệ thống.
 - Tìm kiếm sản phẩm theo mã sản phẩm, tên sản phẩm,...
- **Quản lý hoá đơn:**
 - Tạo hóa đơn bán hàng cho các giao dịch khi khách hàng thanh toán.
 - Tính toán tổng số tiền phải thanh toán.

2.1.2 Các yêu cầu phi chức năng

Yêu cầu phi chức năng đề cập đến các đặc điểm về chất lượng của hệ thống, đảm bảo hệ thống hoạt động hiệu quả và ổn định.

- **Hiệu suất:**

- **Tốc độ:** Hệ thống phải có thể xử lý yêu cầu nhanh chóng. Thời gian đáp ứng mỗi yêu cầu API không quá 2 giây.
- **Khả năng mở rộng:** Hệ thống phải có thể mở rộng dễ dàng khi số lượng người dùng hoặc dữ liệu tăng lên. Cần có khả năng xử lý hàng ngàn yêu cầu đồng thời.

- **Bảo mật:**

- **Xác thực và phân quyền:** Cần có cơ chế bảo mật như xác thực người dùng thông qua JWT (JSON Web Token) và phân quyền người dùng dựa trên vai trò.
- **Mã hóa dữ liệu:** Mật khẩu và thông tin nhạy cảm phải được mã hóa khi lưu trữ.

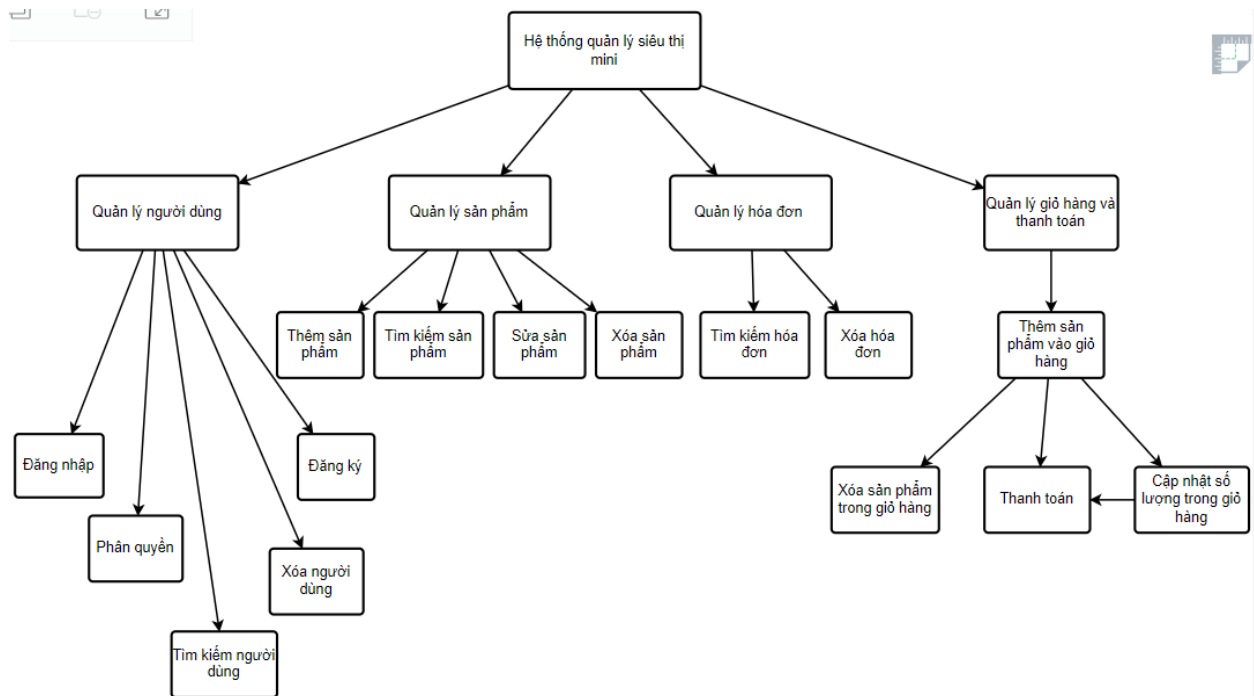
- **Tính mở rộng và khả năng bảo trì:**

- **Tái sử dụng mã nguồn:** Mã nguồn cần phải được cấu trúc rõ ràng, dễ bảo trì, dễ mở rộng.
- **Tài liệu API:** API phải có tài liệu rõ ràng và đầy đủ, cho phép các lập trình viên dễ dàng tích hợp hệ thống với các ứng dụng khác.

- **Tính tương thích:**

- **Tương thích với trình duyệt:** Frontend phải hoạt động tốt trên các trình duyệt phổ biến như Chrome, Cốc cốc,...

2.2 Xây dựng biểu đồ chức năng dựa trên mục tiêu của hệ thống



2.3 Phân rã chức năng con (dịch vụ) dựa trên phân rã mục tiêu con của hệ thống

2.3.1 Quản lý người dùng (Port 8000):

- **Mục tiêu:** Quản lý tài khoản người dùng (thu ngân, admin), bao gồm đăng ký, đăng nhập và phân quyền.
- **Các chức năng con (dịch vụ):**
 - **Đăng ký:**
 - + **Mô tả:** Cho phép người dùng mới đăng ký tài khoản, cung cấp thông tin như tên đăng nhập, email, mật khẩu.
 - + **Dịch vụ:** POST <http://localhost:8000/api/users/register>.
 - + **Chức năng:** Kiểm tra thông tin hợp lệ của thông tin nhập vào, lưu trữ vào cơ sở dữ liệu.
 - **Đăng nhập:**
 - + **Mô tả:** Cung cấp chức năng đăng nhập cho người dùng đã đăng ký. Hệ thống sẽ xác thực thông tin đăng nhập và cung cấp quyền truy cập tương ứng.
 - + **Dịch vụ:** POST <http://localhost:8000/api/users/login>.
 - + **Chức năng:** Xác thực tên đăng nhập và mật khẩu, thiết lập token JWT để truy cập hệ thống.
 - **Phân quyền người dùng:**
 - + **Mô tả:** Gán quyền cho người dùng khi đăng nhập (admin có quyền quản lý, thu ngân có quyền bán hàng, thanh toán hóa đơn).

+ **Chức năng:** Kiểm tra và xác định quyền truy cập của người dùng dựa trên vai trò của họ.

- **Lấy tất cả danh sách người dùng:**

+ **Mô tả:** Chức năng này cho phép admin hoặc hệ thống lấy tất cả thông tin người dùng hiện có trong hệ thống.

+ **Dịch vụ:** GET <http://localhost:8000/api/users/users>.

+ **Chức năng:** Truy vấn cơ sở dữ liệu và trả về danh sách người dùng, bao gồm tên, email, vai trò.

- **Xóa người dùng:**

+ **Mô tả:** Chức năng này cho phép admin xóa người dùng khỏi hệ thống nếu họ không còn cần thiết.

+ **Dịch vụ:** DELETE <http://localhost:8000/api/users/users/:id>

+ **Chức năng:** Admin có thể chọn người dùng để xóa khỏi hệ thống dựa trên id.

- **Tìm kiếm người dùng:**

+ **Mô tả:** Cho phép admin tìm kiếm người dùng theo mã người dùng.

+ **Dịch vụ:** GET <http://localhost:8000/api/users/:id>

+ **Chức năng:** Tìm kiếm người dùng một cách nhanh chóng cho admin.

2.3.2 Quản lý sản phẩm(Port 8001):

- **Mục tiêu:** Quản lý danh sách sản phẩm của siêu thị, bao gồm thêm, sửa, xóa, tìm kiếm sản phẩm.

- **Các chức năng con (dịch vụ):**

- **Thêm sản phẩm mới:**

+ **Mô tả:** Cho phép admin thêm sản phẩm mới vào hệ thống. Cung cấp thông tin như tên sản phẩm, giá cả, mô tả, số lượng tồn kho và hình ảnh.

+ **Dịch vụ:** POST <http://localhost:8001/api/products>

+ **Chức năng:** Kiểm tra tính hợp lệ của thông tin sản phẩm, lưu trữ sản phẩm vào cơ sở dữ liệu, cập nhật số lượng tồn kho.

- **Sửa sản phẩm:**

+ **Mô tả:** Cho phép admin chỉnh sửa số lượng sản phẩm.

+ **Dịch vụ:** PUT <http://localhost:8001/api/products/:id>

+ **Chức năng:** Kiểm tra dữ liệu mới, cập nhật giá và số lượng sản phẩm trong cơ sở dữ liệu.

- **Xóa sản phẩm:**

+ **Mô tả:** Cho phép admin xóa sản phẩm khỏi hệ thống.

+ **Dịch vụ:** DELETE <http://localhost:8001/api/product/:id>

+ **Chức năng:** Xóa sản phẩm khỏi cơ sở dữ liệu.

- **Tìm kiếm sản phẩm:**
 - + **Mô tả:** Cho phép admin tìm kiếm sản phẩm theo mã sản phẩm.
 - + **Dịch vụ:** GET <http://localhost:8001/api/product/:id>
 - + **Chức năng:** Tìm kiếm sản phẩm một cách nhanh chóng.
- **Lấy tất cả danh sách sản phẩm.**
 - + **Mô tả:** Truy xuất danh sách tất cả các sản phẩm hiện có.
 - + **Dịch vụ:** GET <http://localhost:8001/api/product>
 - + **Chức năng:** Truy vấn cơ sở dữ liệu và trả về danh sách sản phẩm, bao gồm mã sản phẩm, tên, giá, số lượng tồn kho, mô tả.
- **Cập nhật số lượng sản phẩm.**
 - + **Mô tả:** Cập nhật số lượng sản phẩm khi nhân viên thực hiện thanh toán.
 - + **Dịch vụ:** PUT <http://localhost:8001/api/products/:id/decrease-quantity>
 - + **Chức năng:** Truy vấn cơ sở dữ liệu và trả về số lượng sản phẩm đã được cập nhật.

2.3.3 Quản lý đơn hàng (Port 8002)

- **Mục tiêu:** Hỗ trợ trong việc tạo hóa đơn và thanh toán.
- **Các chức năng con (dịch vụ):**
 - **Lấy tất cả danh sách hóa đơn:**
 - + **Mô tả:** Chức năng này cho phép admin hoặc hệ thống lấy toàn bộ thông tin các hóa đơn đã được tạo trong hệ thống.
 - + **Dịch vụ:** GET <http://localhost:8002/api/invoices>.
 - + **Chức năng:** Truy xuất danh sách tất cả các hóa đơn đã được tạo trong hệ thống. Các thông tin gồm: Mã hóa đơn, ngày tạo, tổng tiền.
 - **Lấy danh sách hóa đơn theo id thu ngân:**
 - + **Mô tả:** Chức năng này cho phép thu ngân lấy toàn bộ thông tin các hóa đơn mà thu ngân đã tạo.
 - + **Dịch vụ:** GET <http://localhost:8002/api/invoices/user/:id>
 - + **Chức năng:** Truy xuất danh sách tất cả các hóa đơn với id của thu ngân đã tạo trong hệ thống. Các thông tin gồm: Mã hóa đơn, ngày tạo, tổng tiền.
 - **Xóa hóa đơn:**
 - + **Mô tả:** Cho phép admin xóa hóa đơn khỏi hệ thống.
 - + **Dịch vụ:** DELETE <http://localhost:8002/api/invoices/:id>
 - + **Chức năng:** Khi admin xác định hóa đơn cần xóa, hệ thống sẽ yêu cầu xác nhận và sau đó xóa hóa đơn khỏi cơ sở dữ liệu.
 - **Xem chi tiết hóa đơn:**

+ **Mô tả:** Cung cấp chức năng để cả admin và thu ngân có thể xem chi tiết một hóa đơn, bao gồm thông tin: Mã hóa đơn, id người tạo, ngày tạo, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, tổng tiền.

+ **Dịch vụ:** GET http://localhost:8002/api/invoices/:invoice_id/details.

+ **Chức năng:** Khi click xem chi tiết của 1 hóa đơn thì hiển thị thông tin chi tiết của hóa đơn đó.

- **Tìm kiếm hóa đơn:**

+ **Mô tả:** Cho phép admin tìm kiếm hóa đơn dựa trên các tiêu chí như mã hóa đơn.

+ **Chức năng:** Admin tìm kiếm hóa đơn theo mã hóa đơn, hệ thống sẽ trả về danh sách các hóa đơn phù hợp với tiêu chí tìm kiếm.

- **Thanh toán:**

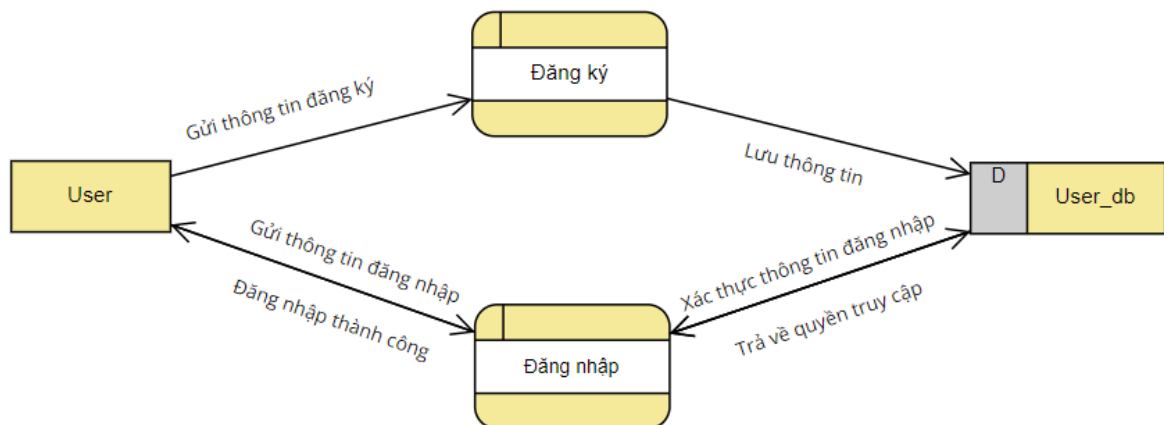
+ **Mô tả:** Cho phép admin tìm kiếm hóa đơn dựa trên các tiêu chí như mã hóa đơn.

+ **Dịch vụ:** POST <http://localhost:8002/api/invoices>

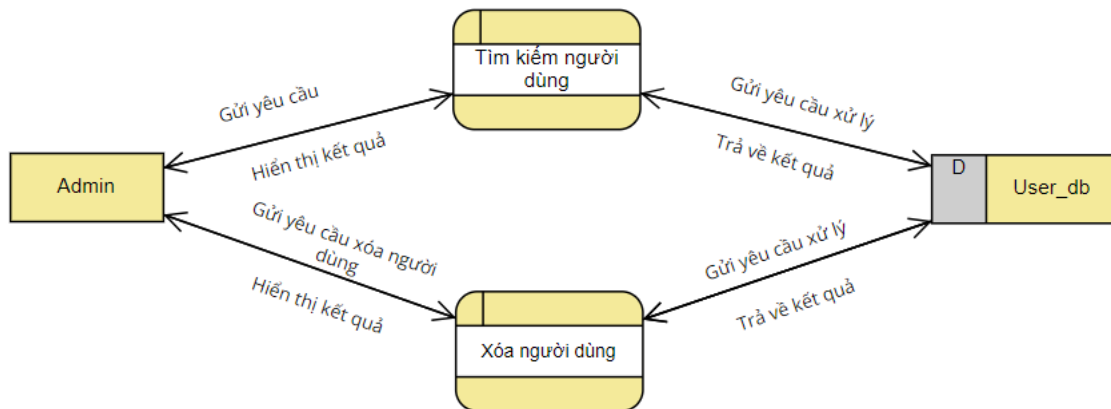
+ **Chức năng:** Admin tìm kiếm hóa đơn theo mã hóa đơn, hệ thống sẽ trả về danh sách các hóa đơn phù hợp với tiêu chí tìm kiếm.

2.4 Biểu đồ luồng dữ liệu (theo chức năng)

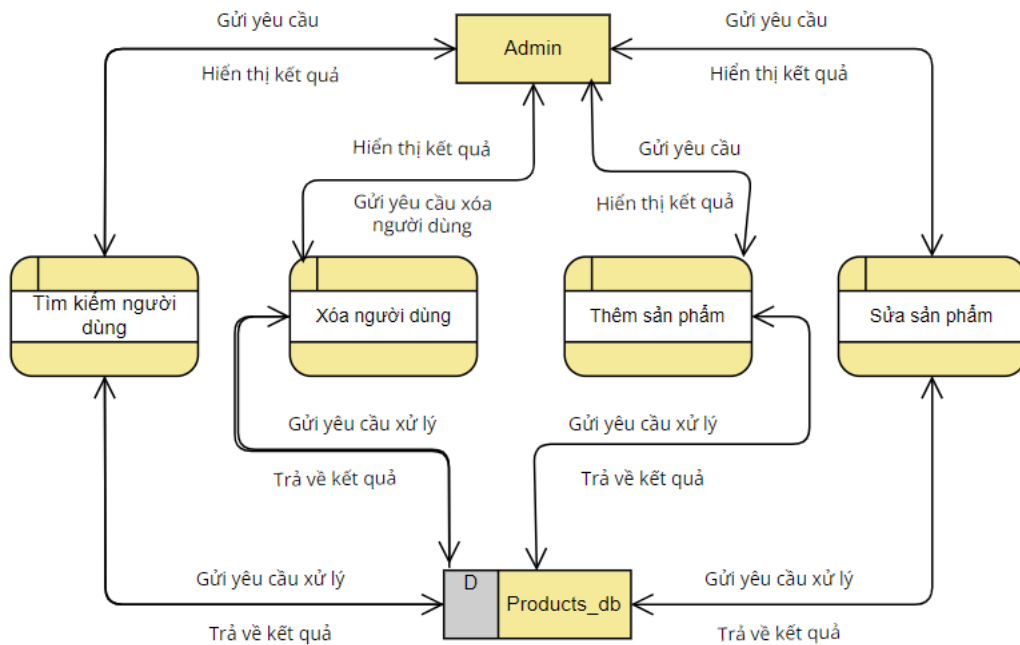
2.4.1 Đăng nhập, đăng ký



2.4.2 Quản lý người dùng



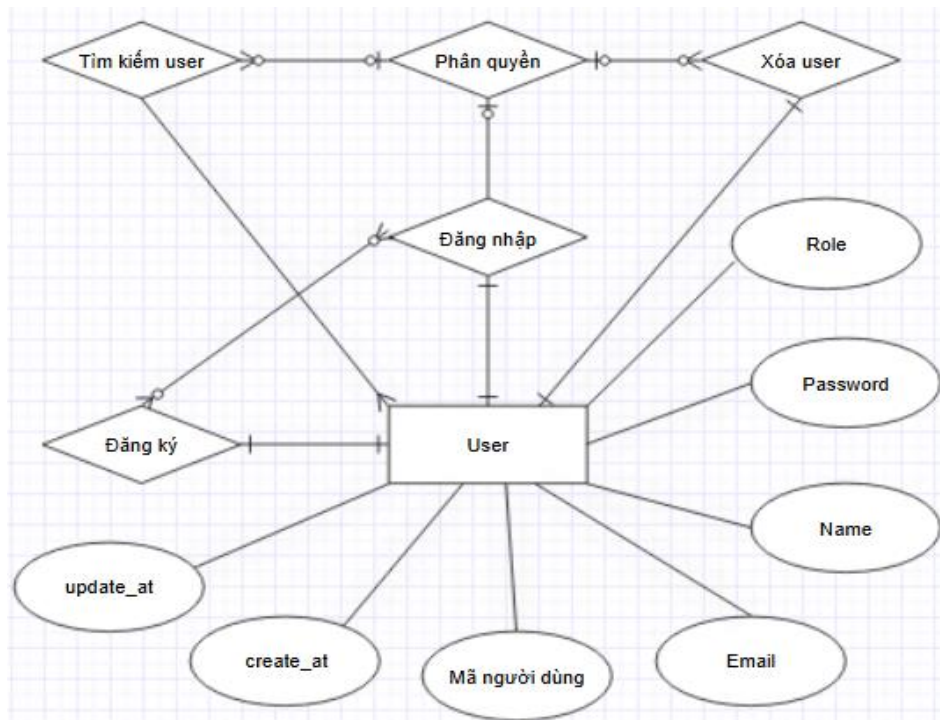
2.4.3 Quản lý sản phẩm



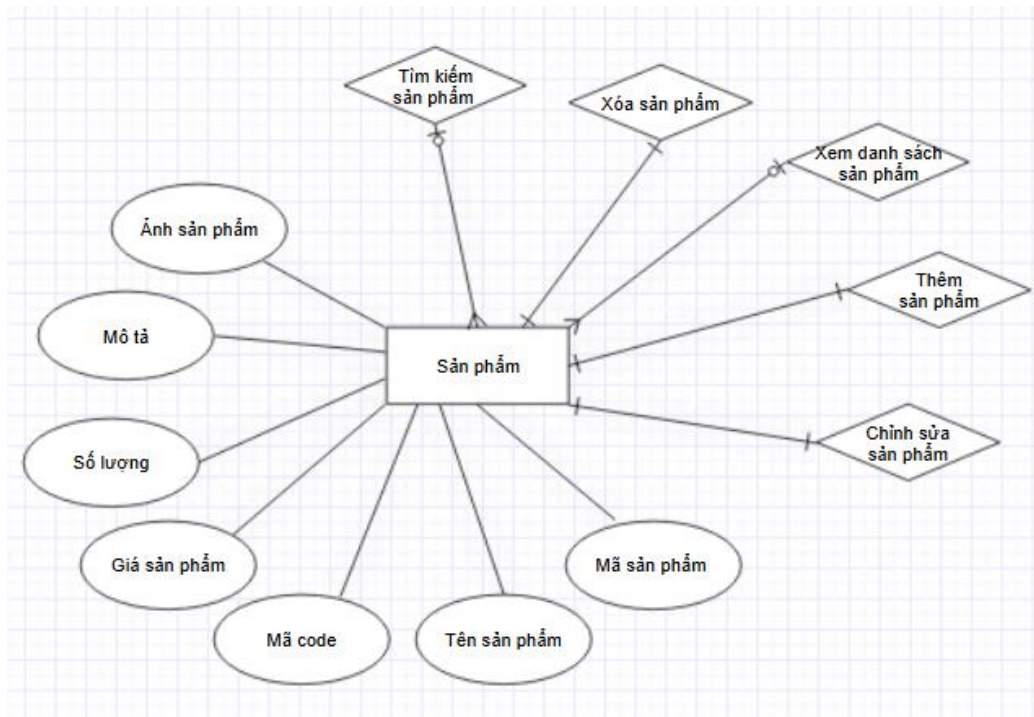
Phân tích và thiết kế dữ liệu của hệ thống

1.1 Mô hình thực thể liên kết của từng dịch vụ

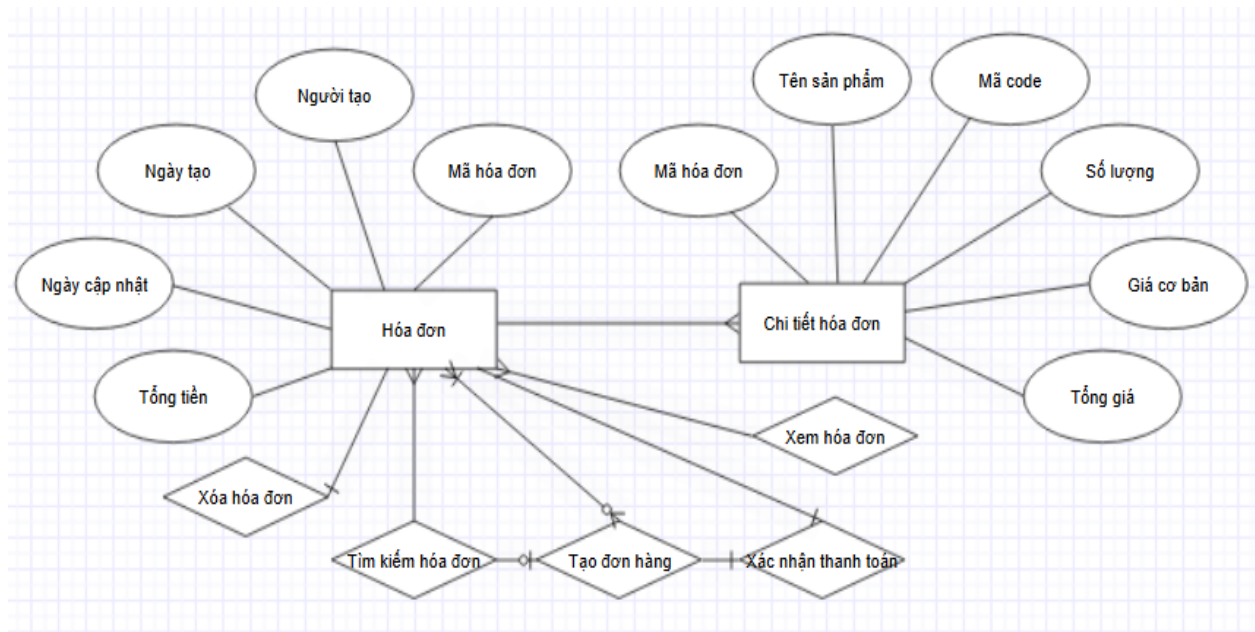
- Dịch vụ: Quản lý người dùng



- Dịch vụ: Quản lý sản phẩm

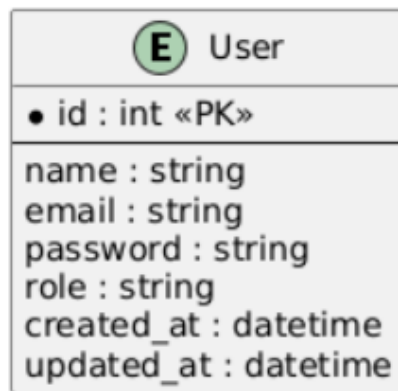


- Dịch vụ: Quản lý đơn hàng

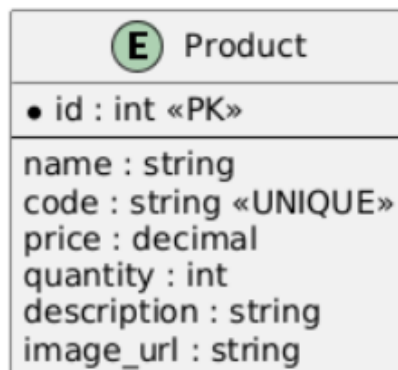


1.2 Mô hình quan hệ của từng dịch vụ

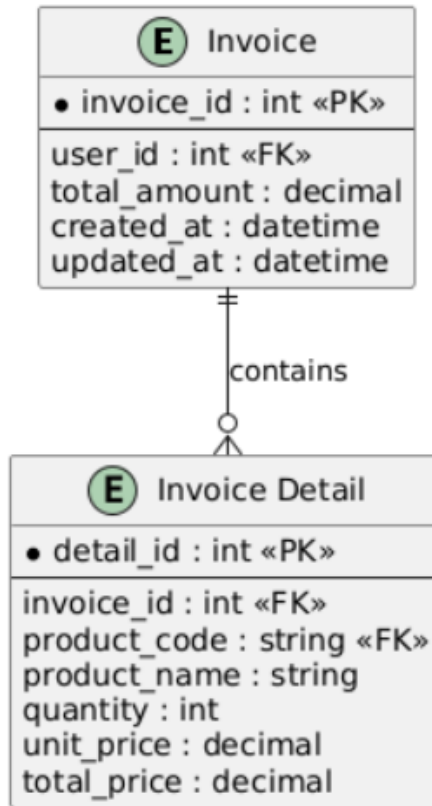
- Dịch vụ: Quản lý người dùng



- Dịch vụ: Quản lý sản phẩm



- Dịch vụ: Quản lý đơn hàng



1.3 Bảng dữ liệu

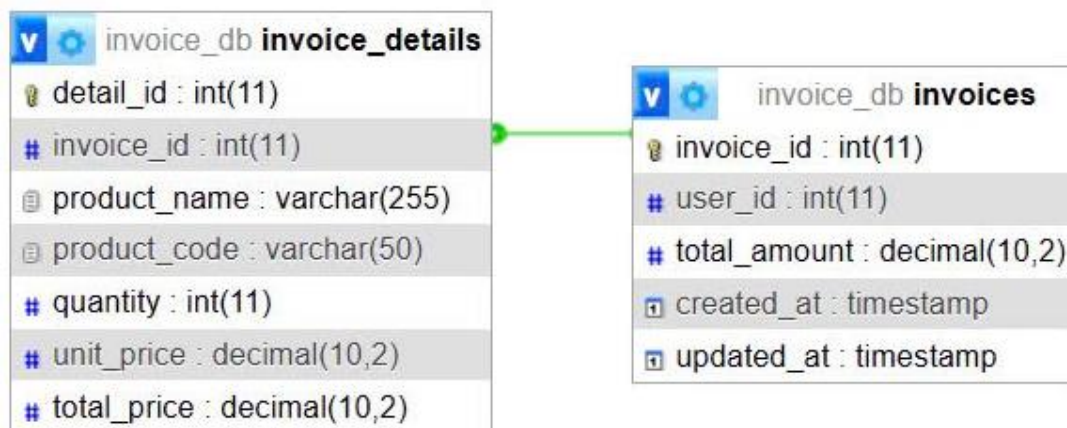
- Dịch vụ quản lý người dùng (port 8000):

| | user_db users |
|---|---------------------------------|
| 🔑 | id : int |
| 📄 | name : varchar(100) |
| 🔑 | email : varchar(100) |
| 📄 | password : varchar(255) |
| 🔍 | role : enum('Admin','Employee') |
| 📅 | created_at : timestamp |
| 📅 | updated_at : timestamp |

- Dịch vụ quản lý sản phẩm (port 8001):

| product_db | products |
|-------------|---------------|
| id | int |
| name | varchar(255) |
| code | varchar(50) |
| price | decimal(10,2) |
| quantity | int |
| description | mediumtext |
| image_url | varchar(255) |

- Dịch vụ quản lý đơn hàng (port 8002):

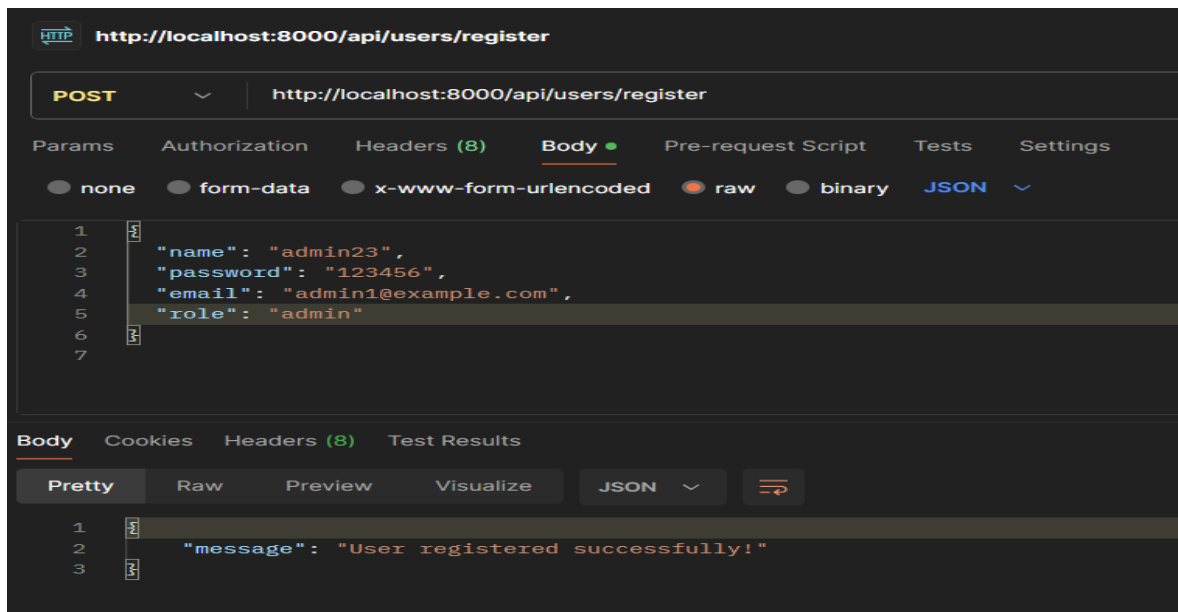


Giao diện của hệ thống

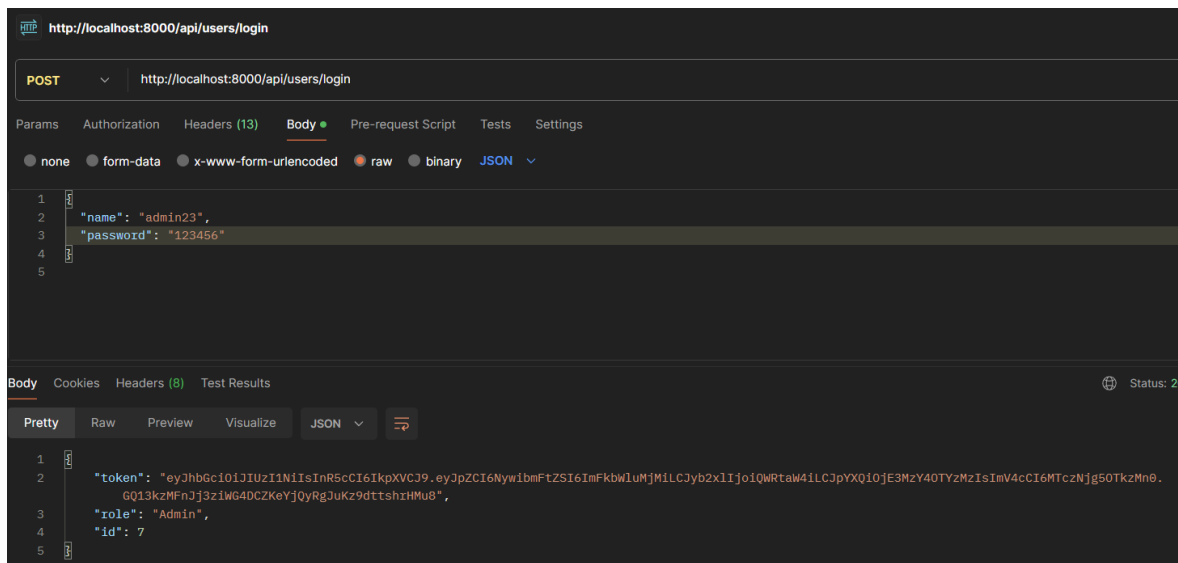
4.1 Xây dựng giao diện API cho từng dịch vụ

4.1.1 Dịch vụ quản lý người dùng (port 8000):

- Đăng ký:
+ Dịch vụ: POST <http://localhost:8000/api/users/register>.

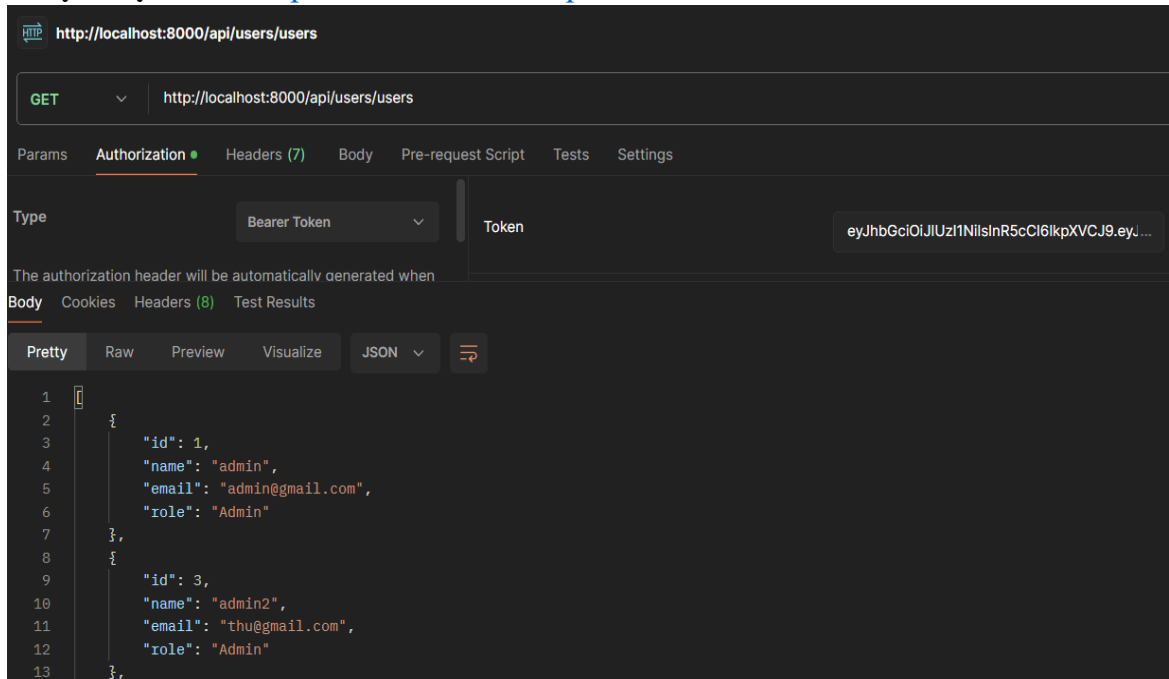


- Đăng nhập:
+ Dịch vụ: POST <http://localhost:8000/api/users/login>.



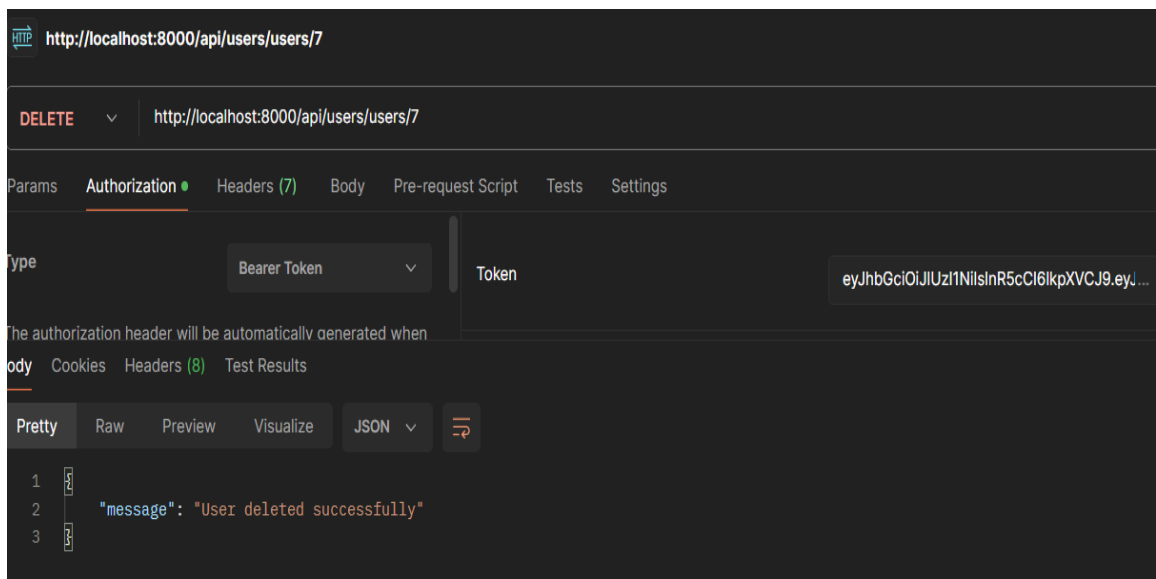
- **Lấy tất cả danh sách người dùng:**

+ **Dịch vụ:** GET <http://localhost:8000/api/users/users>



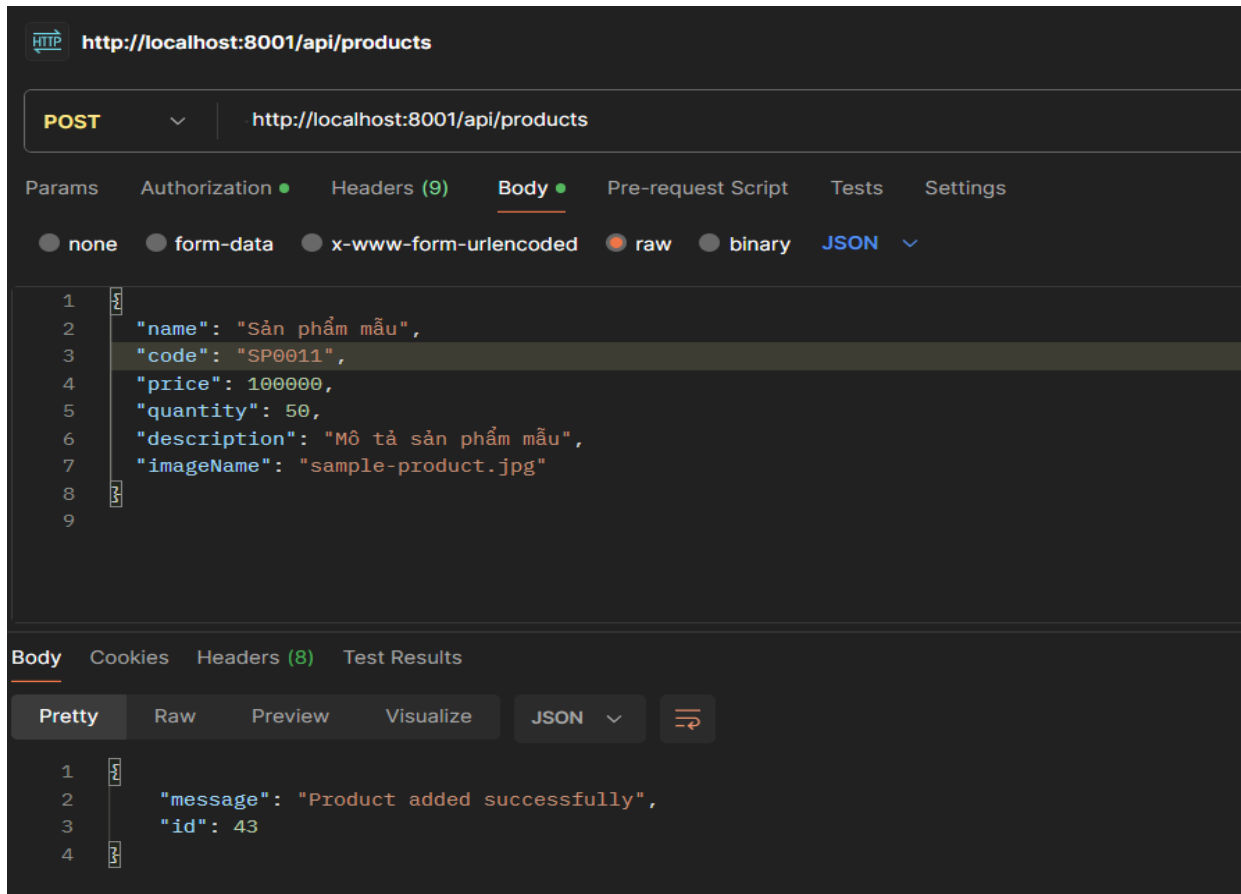
- **Xóa người dùng:**

+ **Dịch vụ:** DELETE <http://localhost:8000/api/users/users/:id>



4.1.1 Dịch vụ quản lý sản phẩm (port 8001):

- Thêm sản phẩm mới:
+ Dịch vụ: POST <http://localhost:8001/api/products>



- Sửa sản phẩm:
+ Dịch vụ: PUT <http://localhost:8001/api/products/:id>

The screenshot shows a REST client interface with a PUT request to `http://localhost:8001/api/products/43`. The request body is a JSON object: `{"price": 120000, "quantity": 30}`. The response body is a JSON object: `{"message": "Product updated successfully"}`.

Request:

- Method: PUT
- URL: `http://localhost:8001/api/products/43`
- Body:

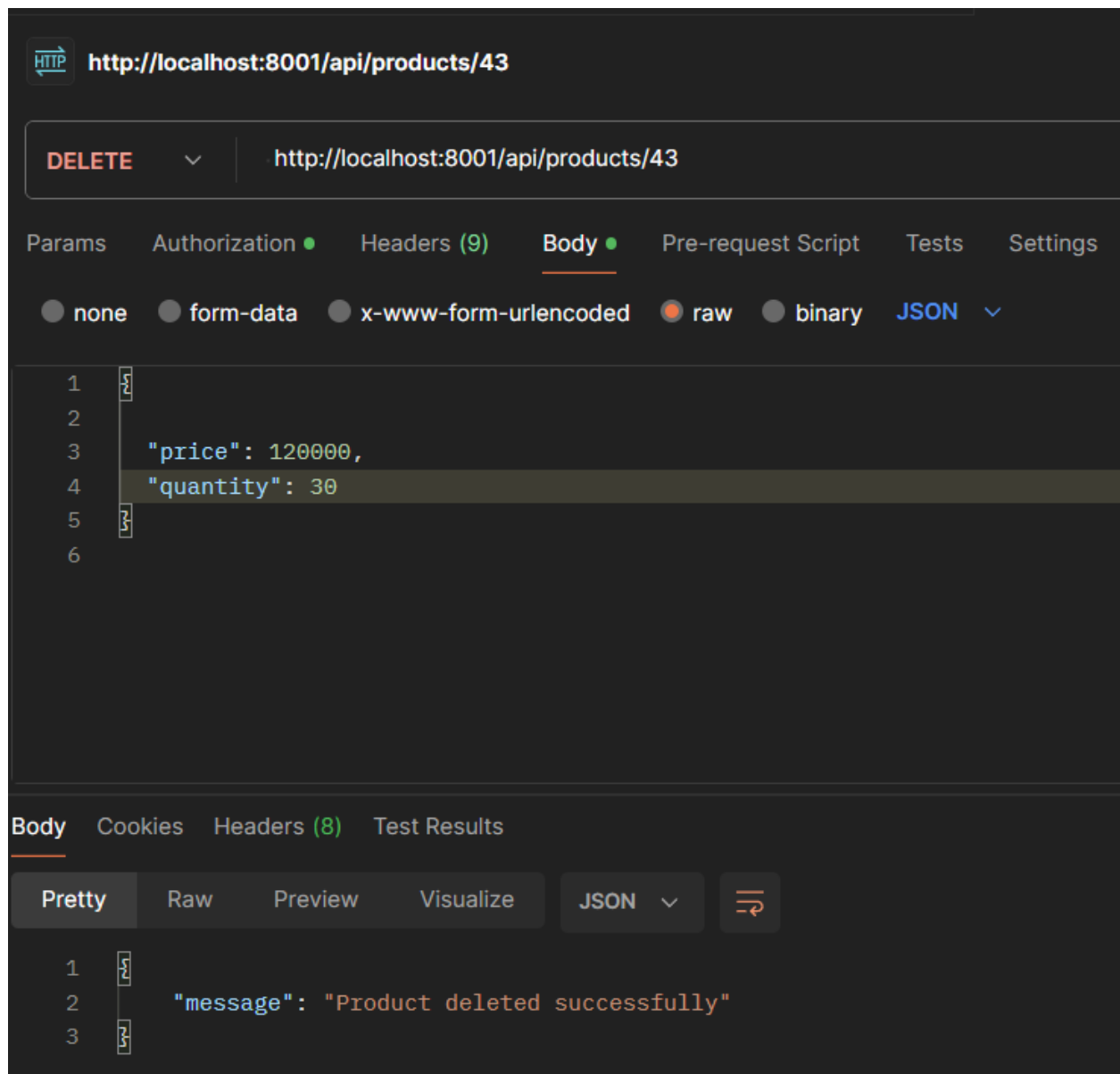
```
1 {  
2     
3   "price": 120000,  
4   "quantity": 30  
5 }  
6
```

Response:

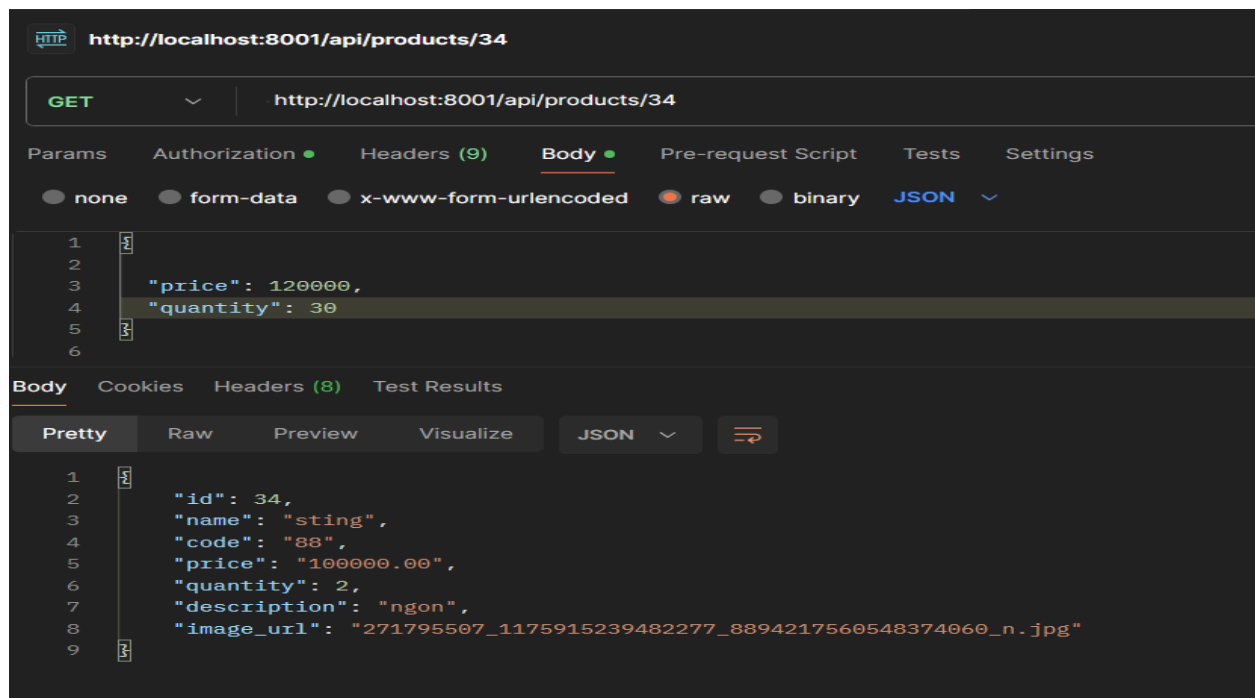
- Body:

```
1 {  
2   "message": "Product updated successfully"  
3 }
```

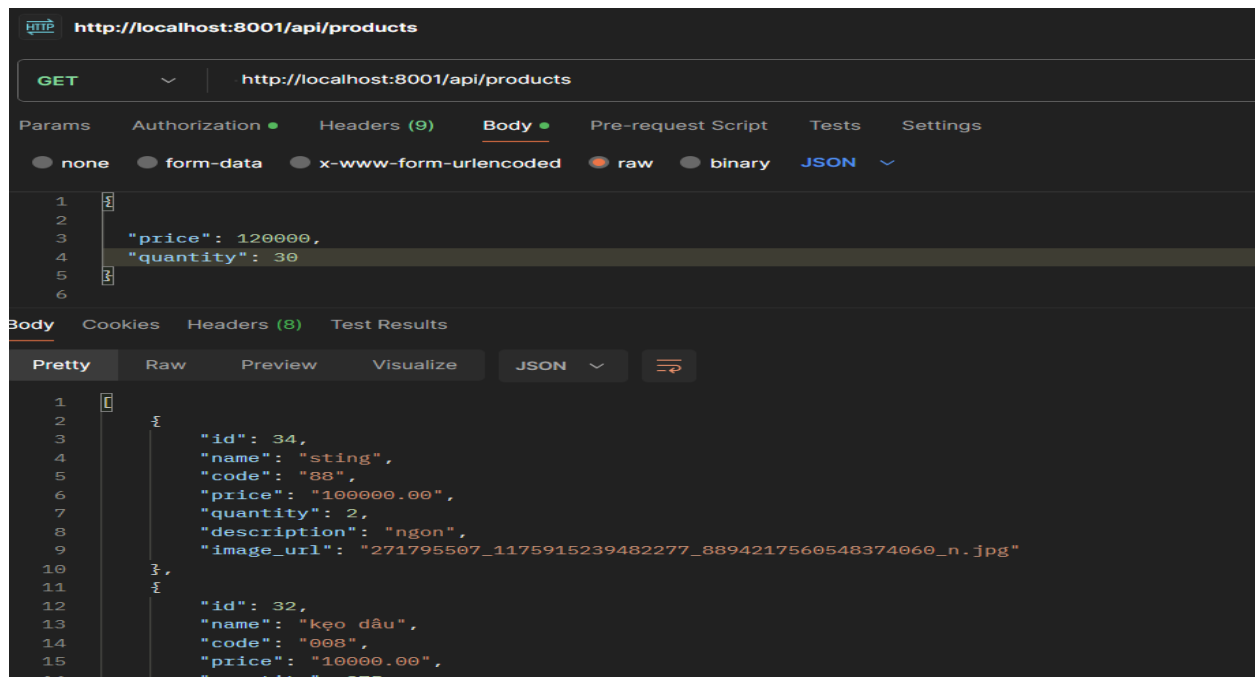
- Xóa sản phẩm:
+ Dịch vụ: DELETE <http://localhost:8001/api/product/:id>



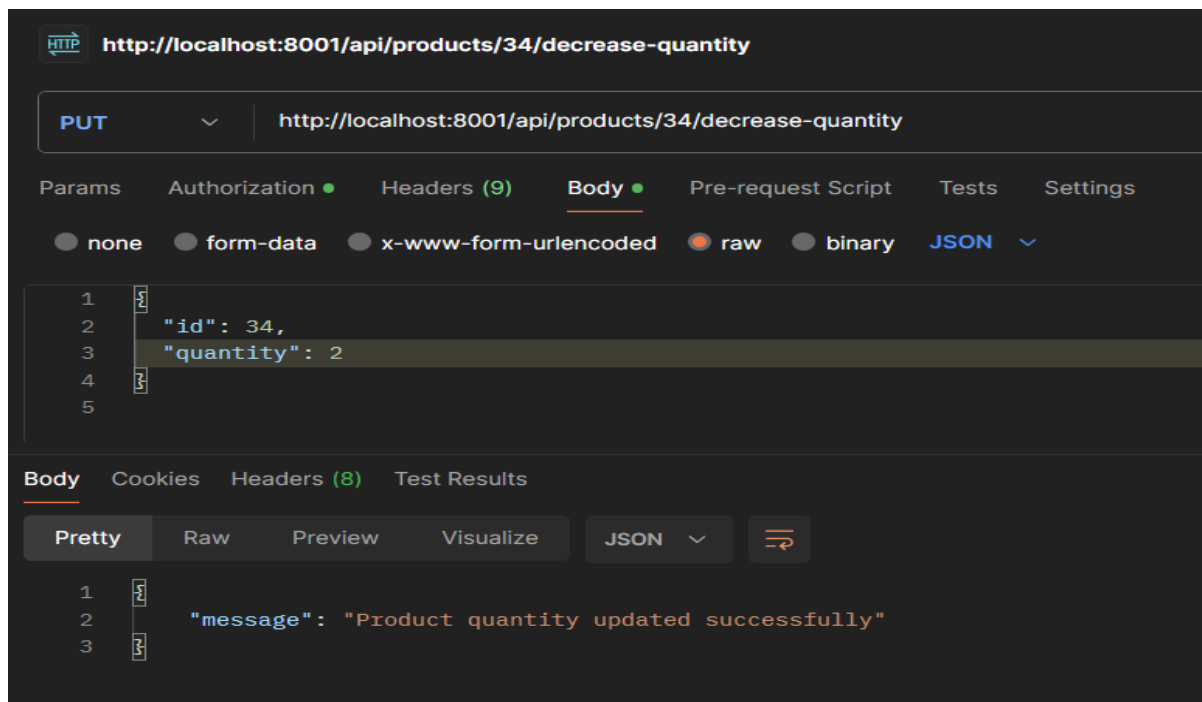
- Tìm kiếm sản phẩm:
+ Dịch vụ: GET <http://localhost:8001/api/product/:id>



- Lấy tất cả danh sách sản phẩm.
+ Dịch vụ: GET <http://localhost:8001/api/product>

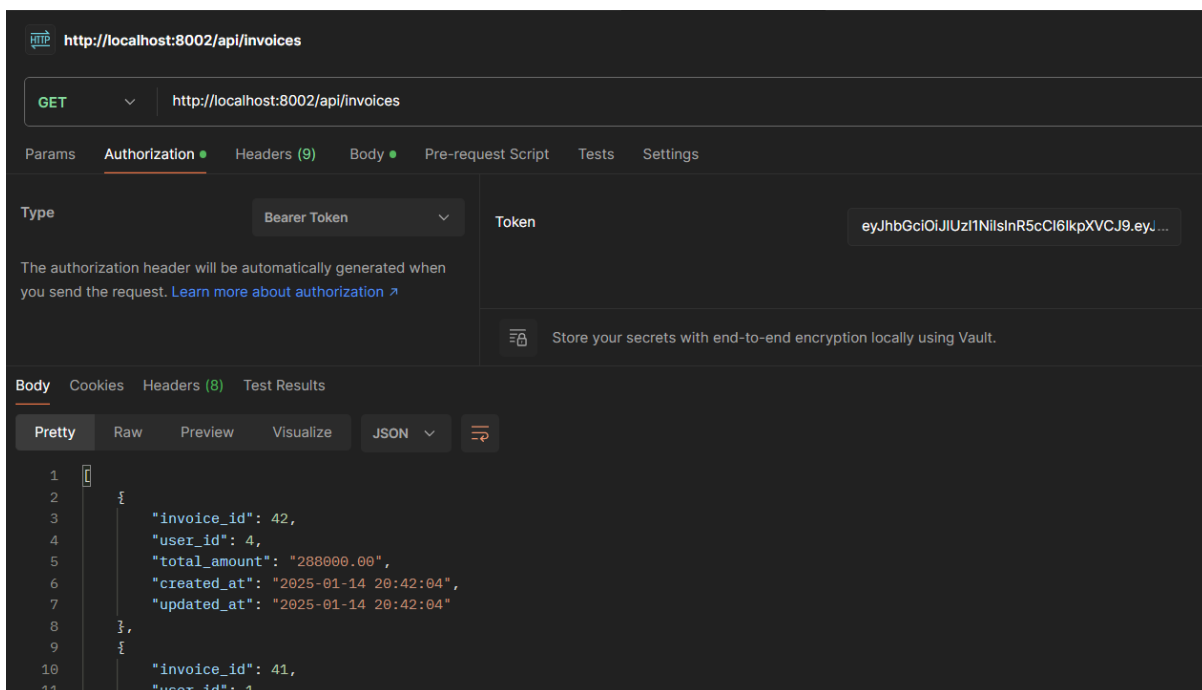


- Cập nhật số lượng sản phẩm.
+ Dịch vụ: PUT <http://localhost:8001/api/products/:id/decrease-quantity>

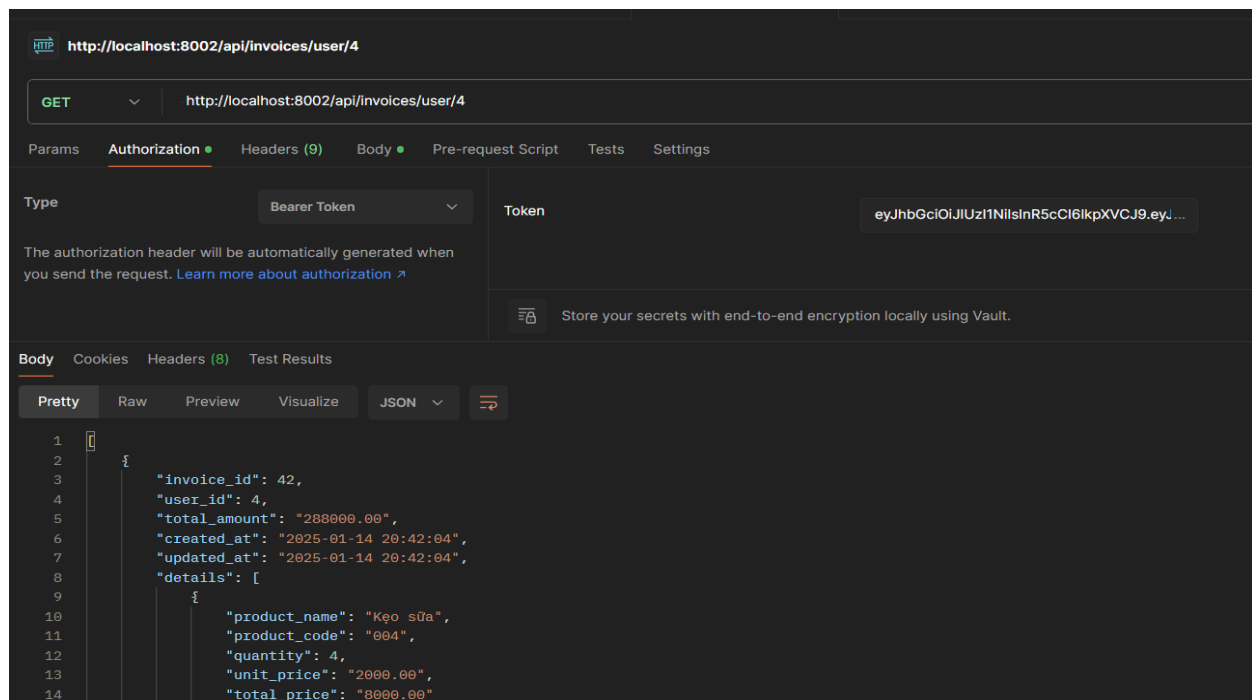


4.1.1 Dịch vụ quản lý đơn hàng(port 8002):

- Lấy tất cả danh sách hóa đơn:
+ Dịch vụ: GET <http://localhost:8002/api/invoices>.



- **Lấy danh sách hóa đơn theo id thu ngân:**
+ Dịch vụ: GET <http://localhost:8002/api/invoices/user/:id>



HTTP <http://localhost:8002/api/invoices/user/4>

GET <http://localhost:8002/api/invoices/user/4>

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Type Bearer Token Token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Store your secrets with end-to-end encryption locally using Vault.

Body Cookies Headers (8) Test Results

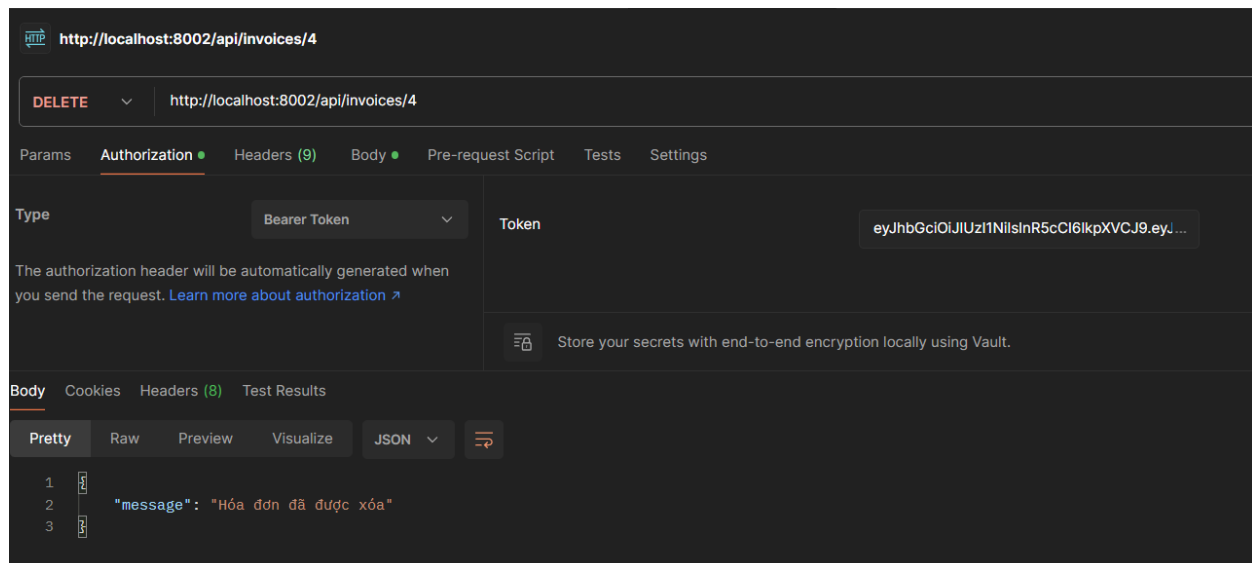
Pretty Raw Preview Visualize JSON

```

1 {
2   "invoice_id": 42,
3   "user_id": 4,
4   "total_amount": "288000.00",
5   "created_at": "2025-01-14 20:42:04",
6   "updated_at": "2025-01-14 20:42:04",
7   "details": [
8     {
9       "product_name": "Kẹo sữa",
10      "product_code": "004",
11      "quantity": 4,
12      "unit_price": "2000.00",
13      "total_price": "8000.00"
14    }
15  ]
16 }

```

- **Xóa hóa đơn:**
+ Dịch vụ: DELETE <http://localhost:8002/api/invoices/:id>



HTTP <http://localhost:8002/api/invoices/4>

DELETE <http://localhost:8002/api/invoices/4>

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Type Bearer Token Token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Store your secrets with end-to-end encryption locally using Vault.

Body Cookies Headers (8) Test Results

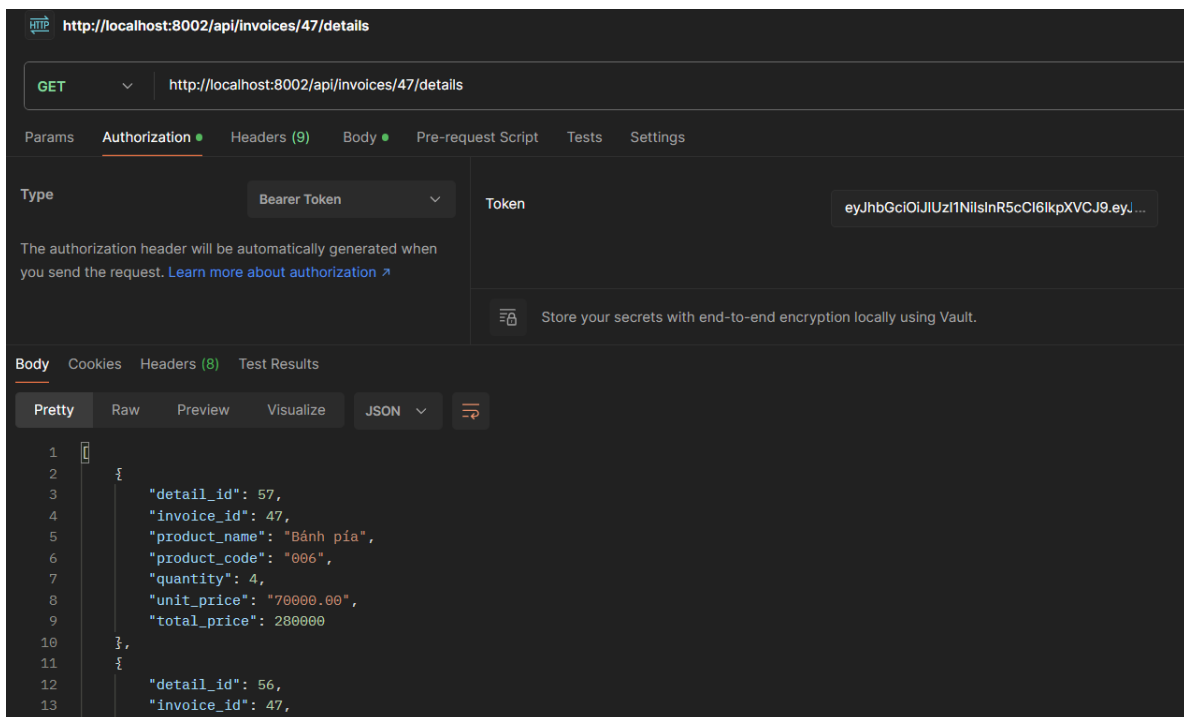
Pretty Raw Preview Visualize JSON

```

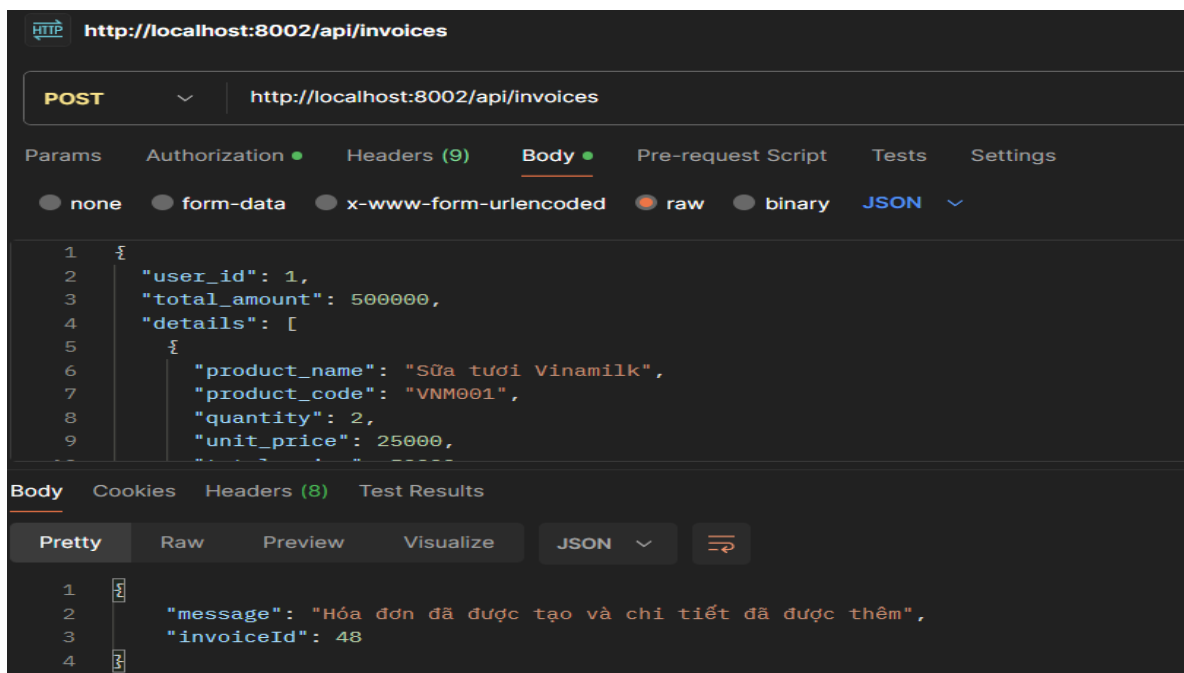
1 {
2   "message": "Hóa đơn đã được xóa"
3 }

```

- Xem chi tiết hóa đơn:
+ Dịch vụ: GET http://localhost:8002/api/invoices/:invoice_id/details.

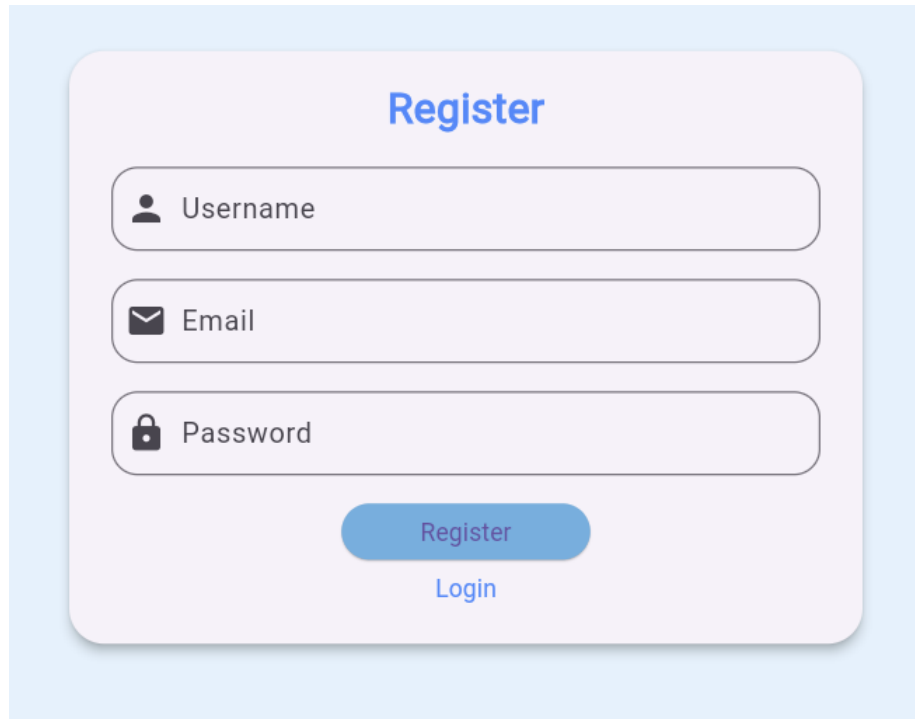


- Thanh toán:
+ Dịch vụ: POST <http://localhost:8002/api/invoices>



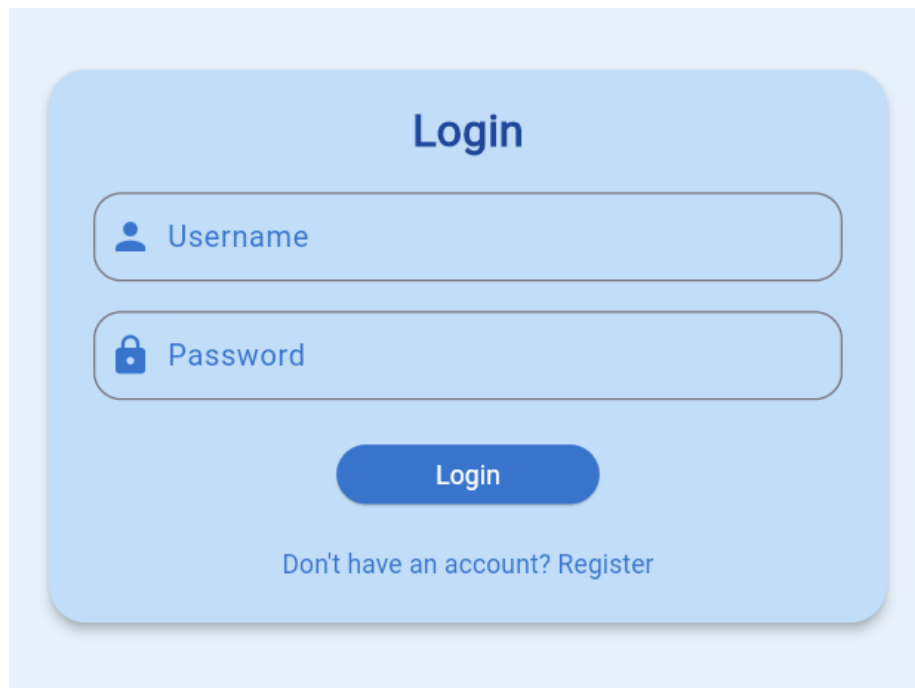
4.2 Xây dựng giao diện người dùng (Web, Mobile, Windows)

- **Giao diện đăng ký:**



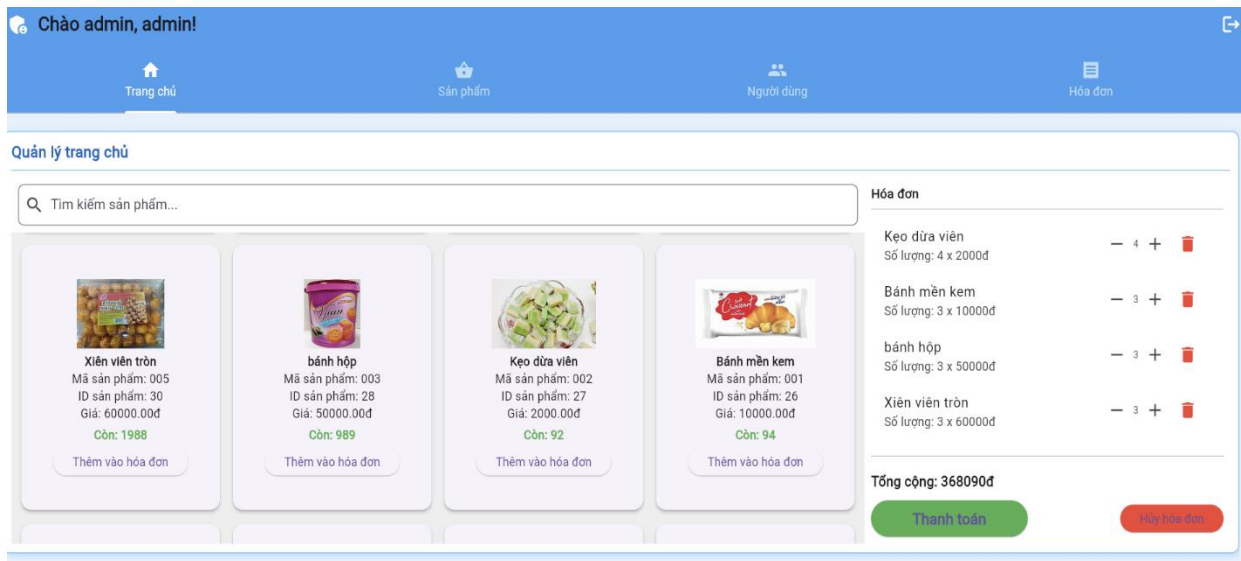
A UI mockup of a registration form. The form is titled "Register" in blue text. It contains three input fields: "Username" with a person icon, "Email" with an envelope icon, and "Password" with a lock icon. Below the fields are two buttons: a blue "Register" button and a light blue "Login" link.

- **Giao diện đăng nhập:**

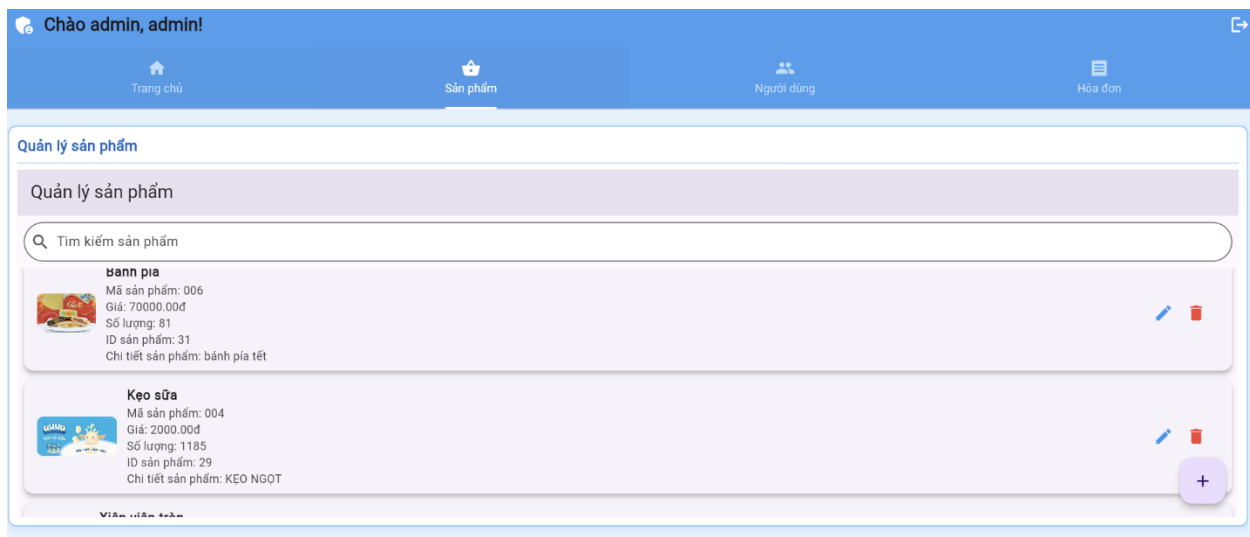


A UI mockup of a login form. The form is titled "Login" in blue text. It contains two input fields: "Username" with a person icon and "Password" with a lock icon. Below the fields is a blue "Login" button. At the bottom, there is a link that says "Don't have an account? Register".

- **Giao diện admin:**



- **Giao diện quản lý sản phẩm:**



- **Giao diện quản lý user:**

Chào admin, admin!

Trang chủ

Sản phẩm

Người dùng

Hóa đơn

Quản lý người dùng

🔍 Tìm kiếm người dùng

Chọn vai trò

1admin
Email: admin@gmail.com
Role: Admin

3admin2
Email: thu@gmail.com
Role: Admin

4Employee
Email: thuthue20031022@gmail.com

- **Giao diện quản lý hóa đơn cho admin:**

Chào admin, admin!

Trang chủ

Sản phẩm

Người dùng

Hóa đơn

Quản lý hóa đơn

Quản lý hóa đơn

🔍 Tìm kiếm hóa đơn

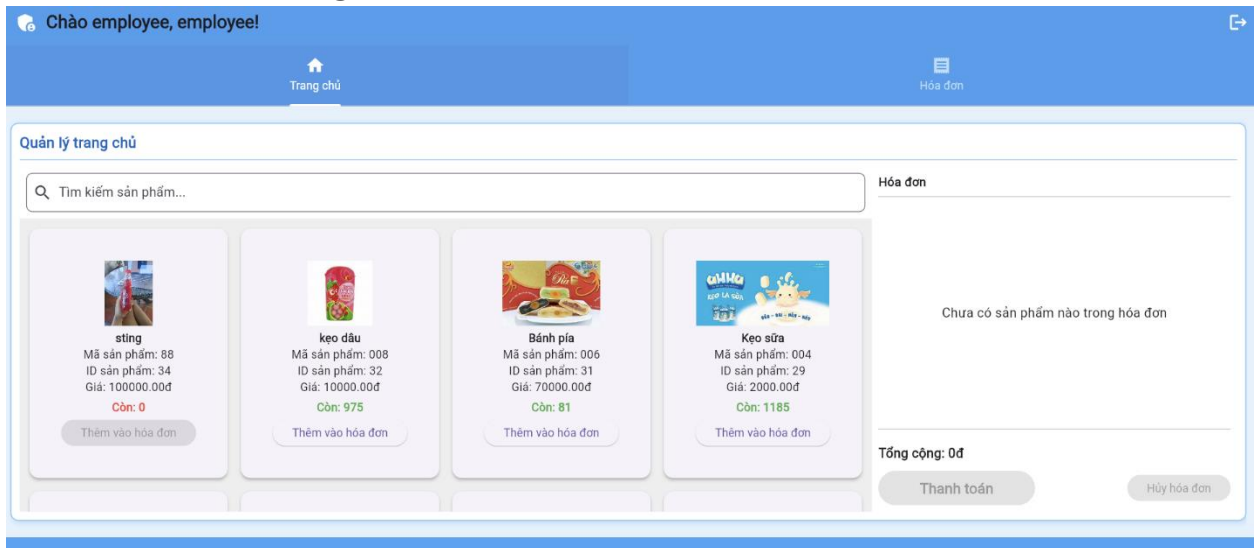
Hóa đơn #48
Người tạo: 1
Ngày tạo: 2025-01-15 06:37:58
Tổng tiền: 500000.00 đ

Hóa đơn #42
Người tạo: 4
Ngày tạo: 2025-01-14 20:42:04
Tổng tiền: 288000.00 đ

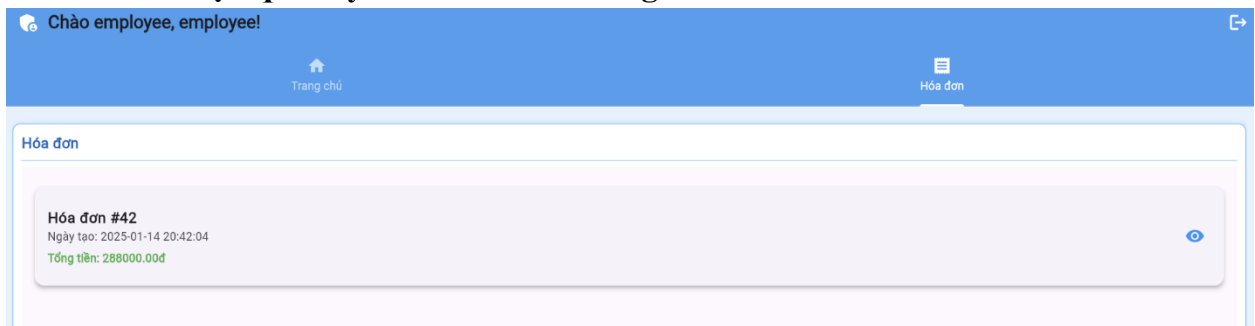
Hóa đơn #41

25

- **Giao diện thu ngân:**



- **Giao diện quản lý hóa đơn cho thu ngân:**



- **Giao diện xem chi tiết hóa đơn:**

Chi tiết hóa đơn

Mã hóa đơn: 42

Người tạo: 4

Ngày tạo: 2025-01-14 20:42:04

Chi tiết sản phẩm:

| | |
|--|---------|
| Kẹo sữa | 8000đ |
| Mã sản phẩm: 004 - Số lượng: 4 x 2000.00đ | |
| Bánh pía | 280000đ |
| Mã sản phẩm: 006 - Số lượng: 4 x 70000.00đ | |

Tổng tiền: 288000.00 đ

Đóng

Kết luận

1.1 Kết quả đạt được

- Áp dụng thành công **Kiến trúc hướng dịch vụ SOA** trong phát triển giao diện web.
- Ứng dụng quản lý bán hàng đã hoàn thành với giao diện đơn giản và các chức năng cơ bản hoạt động tốt.
- Hỗ trợ nhiều chức năng phù hợp với thực tế của việc tìm kiếm, thêm, xóa, sửa và quản lý sản phẩm, đơn hàng và người dùng.

1.2 Hạn chế

- Chưa tập trung quá nhiều về giao diện thân thiện, thuận tiện cho người dùng sử dụng.
- Do thời gian làm bài tập có hạn, khả năng còn hạn chế và một số vấn đề khó khăn nảy sinh trong quá trình làm bài tập nên còn nhiều vấn đề vẫn chỉ còn trên ý tưởng, chưa được thực thi.
- Còn hạn chế về tính năng nâng cao như tích hợp thanh toán online.

1.3 Hướng mở rộng

- Mở rộng thêm chức năng thanh toán trực tuyến.
- Phát triển thêm việc tạo tài khoản cho nhân viên thay vì tự đăng ký, để Admin quản lý nhân viên tốt hơn.
- Tích hợp quét mã sản phẩm, thay vì thêm sản phẩm vào hóa đơn theo cách thêm và tìm kiếm thông thường.
- Tích hợp báo cáo trực quan bằng biểu đồ nâng cao.

1.4 Link git

<https://github.com/TruongHue/KienTrucHuongDichVu.git>

LỜI CẢM ƠN

Trước tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc tới thầy **Trần Hoàng Việt**, giảng viên hướng dẫn, người đã tận tình chỉ bảo và đồng hành cùng chúng em trong suốt quá trình thực hiện đề tài "**Quản lý siêu thị mini theo kiến trúc hướng dịch vụ**". Sự hướng dẫn tận tâm, những ý kiến góp ý quý báu cùng sự động viên của thầy là nguồn động lực lớn giúp chúng em hoàn thành tốt bài báo cáo này.

Đề tài được thực hiện trong khuôn khổ học phần **Kiến trúc hướng dịch vụ**, thuộc chương trình đào tạo chuyên ngành **Công nghệ phần mềm**, Khoa **Công nghệ thông tin**, Trường **Đại học Quy Nhơn**. Chúng em xin chân thành cảm ơn các thầy cô trong khoa và nhà trường đã tạo điều kiện thuận lợi, cung cấp tài liệu, kiến thức và một môi trường học tập chuyên nghiệp, giúp chúng em có cơ hội nghiên cứu và phát triển đề tài này.

Bên cạnh đó, chúng em cũng xin cảm ơn Khoa **Công nghệ thông tin**, Trường **Đại học Quy Nhơn**, đã tổ chức học phần "**Kiến trúc hướng dịch vụ**", giúp chúng em không chỉ nắm vững lý thuyết mà còn có cơ hội áp dụng những kiến thức đã học vào thực tiễn, qua đó hiểu sâu hơn về việc xây dựng và triển khai hệ thống theo kiến trúc dịch vụ (SOA).

Cuối cùng, mặc dù đã cố gắng hoàn thiện bài báo cáo, nhưng với thời gian và năng lực còn hạn chế, chắc chắn không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp từ thầy cô và các bạn để có thể cải thiện và phát triển đề tài tốt hơn trong tương lai.

Một lần nữa, chúng em xin gửi lời cảm ơn chân thành tới tất cả những ai đã hỗ trợ và đồng hành cùng chúng em trong suốt hành trình này!