**TÀI LIỆU YÊU CẦU DỰ ÁN**

**HỆ THỐNG QUẢN LÝ BÁN HÀNG (POINT OF SALE - POS)**

Phiên bản: 1.1

Ngày: 22/05/2025

Tác giả: [Cao Đăng Khoa]

**MỤC LỤC TÀI LIỆU**

[**1. Giới thiệu tổng quan** 4](#_Toc198804773)

[**1.1. Mục đích và Phạm vi** 4](#_Toc198804774)

[**1.2. Các thuật ngữ và từ viết tắt** 4](#_Toc198804775)

[**2. Phạm vi dự án (Scope)** 5](#_Toc198804776)

[**3. Đối tượng người dùng (User Roles)** 7](#_Toc198804777)

[**3.1. Quản trị viên (Admin)** 7](#_Toc198804778)

[**3.2. Nhân viên Bán hàng (Cashier/Sales Staff)** 7](#_Toc198804779)

[**3.3. Quản lý Kho (Inventory Manager)** 8](#_Toc198804780)

[**3.4. Khách hàng (Customer)** 8](#_Toc198804781)

[**4. Chức năng chính (Functional Requirements)** 8](#_Toc198804782)

[**4.1. Quản lý Bán hàng (Sales Management)** 8](#_Toc198804783)

[**4.2. Quản lý Sản phẩm (Product Management)** 10](#_Toc198804784)

[**4.3. Quản lý Kho (Inventory Management)** 12](#_Toc198804785)

[**4.4. Quản lý Khách hàng (Customer Management)** 12](#_Toc198804786)

[**4.5. Quản lý Nhân viên & Quyền hạn (Employee & Role Management)** 13](#_Toc198804787)

[**4.6. Quản lý Bàn (Table Management - If Applicable)** 14](#_Toc198804788)

[**4.7. Quản lý Khuyến mãi (Discount Management)** 14](#_Toc198804789)

[**4.8. Báo cáo & Thống kê (Reporting & Analytics)** 15](#_Toc198804790)

[**4.9. Cấu hình Hệ thống (System Configuration)** 16](#_Toc198804791)

[**4.10. Xác thực & Bảo mật (Authentication & Security)** 16](#_Toc198804792)

[**5. Yêu cầu phi chức năng (Non-Functional Requirements)** 17](#_Toc198804793)

[**5.1. Hiệu suất (Performance)** 17](#_Toc198804794)

[**5.2. Bảo mật (Security)** 18](#_Toc198804795)

[**5.3. Khả năng mở rộng (Scalability)** 19](#_Toc198804796)

[**5.4. Khả năng sử dụng (Usability)** 19](#_Toc198804797)

[**5.5. Độ tin cậy (Reliability)** 19](#_Toc198804798)

[**5.6. Môi trường triển khai (Deployment Environment)** 20](#_Toc198804799)

# 

# **1. Giới thiệu tổng quan**

## **1.1. Mục đích và Phạm vi**

Hệ thống Quản lý Bán hàng (Point of Sale - POS) là một ứng dụng phần mềm toàn diện được thiết kế để số hóa và tối ưu hóa các quy trình kinh doanh cốt lõi tại các cơ sở bán lẻ như cửa hàng, quán cà phê, nhà hàng. Mục đích chính của hệ thống là cung cấp một giải pháp tích hợp, đáng tin cậy và hiệu quả nhằm hỗ trợ việc quản lý giao dịch bán hàng, hàng tồn kho, thông tin khách hàng, quản lý nhân sự và các hoạt động kinh doanh liên quan. Hệ thống sẽ góp phần nâng cao hiệu suất vận hành, giảm thiểu sai sót do con người, cung cấp dữ liệu kinh doanh tức thời và chính xác để hỗ trợ các quyết định chiến lược.

Với kiến trúc hiện đại, hệ thống sẽ bao gồm:

* Backend API: Được phát triển bằng Java với Spring Boot, cung cấp các dịch vụ web RESTful mạnh mẽ, bảo mật và khả năng mở rộng để xử lý tất cả các logic nghiệp vụ và tương tác với cơ sở dữ liệu.
* Frontend UI: Được xây dựng bằng Vue.js, cung cấp một giao diện người dùng trực quan, phản hồi nhanh (responsive) và thân thiện, tối ưu hóa trải nghiệm cho nhân viên bán hàng và các cấp quản lý.

Dự án này tập trung vào việc xây dựng các module cốt lõi, đủ để vận hành hiệu quả tại một điểm bán hàng duy nhất. Các tính năng mở rộng như tích hợp với hệ thống kế toán bên ngoài, hỗ trợ nhiều chi nhánh, chương trình khách hàng thân thiết phức tạp, và phân tích dữ liệu chuyên sâu sẽ được xem xét và triển khai trong các giai đoạn phát triển tiếp theo của dự án.

## **1.2. Các thuật ngữ và từ viết tắt**

|  |  |
| --- | --- |
| Thuật ngữ/Từ viết tắt | Diễn giải |
| POS | Point of Sale (Hệ thống điểm bán hàng) |
| API | Application Programming Interface (Giao diện lập trình ứng dụng) |
| UI | User Interface (Giao diện người dùng) |
| FR | Functional Requirement (Yêu cầu chức năng) |
| NFR | Non-Functional Requirement (Yêu cầu phi chức năng) |
| CRUD | Create, Read, Update, Delete (Tạo, Đọc, Cập nhật, Xóa) |
| CSDL | Cơ sở dữ liệu |
| ERD | Entity-Relationship Diagram (Sơ đồ thực thể-quan hệ) |
| JWT | JSON Web Token (Một phương pháp xác thực không trạng thái) |
| RBAC | Role-Based Access Control (Kiểm soát truy cập dựa trên vai trò) |
| HTTPS | Hypertext Transfer Protocol Secure (Giao thức truyền tải siêu văn bản an toàn) |
| SQL Injection | Một loại tấn công chèn mã SQL vào các trường nhập liệu |
| XSS | Cross-Site Scripting (Một loại tấn công chéo trang) |
| CSRF | Cross-Site Request Forgery (Một loại tấn công giả mạo yêu cầu chéo trang) |
| LTS | Long Term Support (Phiên bản hỗ trợ dài hạn) |
| ORM | Object-Relational Mapping (Ánh xạ đối tượng quan hệ) |

# **2. Phạm vi dự án (Scope)**

Dự án Hệ thống POS sẽ bao gồm việc phát triển các module và chức năng cốt lõi sau:

* Module Bán hàng: Cung cấp giao diện trực quan cho nhân viên bán hàng để xử lý các giao dịch mua bán, từ việc chọn sản phẩm, áp dụng khuyến mãi đến thanh toán.
* Module Quản lý Sản phẩm: Cho phép quản lý thông tin chi tiết về sản phẩm và danh mục sản phẩm.
* Module Quản lý Kho: Hỗ trợ việc theo dõi nhập/xuất kho và cập nhật tồn kho.
* Module Quản lý Khách hàng: Quản lý thông tin và lịch sử mua hàng của khách hàng.
* Module Quản lý Nhân viên & Phân quyền: Quản lý tài khoản nhân viên và gán quyền hạn truy cập dựa trên vai trò.
* Module Quản lý Bàn (tùy chọn cho nhà hàng/cafe): Quản lý trạng thái và sức chứa của các bàn.
* Module Quản lý Khuyến mãi: Tạo và quản lý các chương trình khuyến mãi.
* Module Báo cáo cơ bản: Cung cấp các báo cáo về doanh thu, sản phẩm bán chạy, và tồn kho.
* Module Cấu hình Hệ thống: Cho phép tùy chỉnh các thông tin cơ bản của cửa hàng.
* Module Xác thực & Bảo mật: Đảm bảo an toàn cho dữ liệu và quyền truy cập của người dùng.
* Module Ghi log hoạt động (Audit Log): Theo dõi các thao tác quan trọng trong hệ thống.

Dự án này không bao gồm các tính năng sau trong giai đoạn hiện tại (sẽ được xem xét trong các giai đoạn mở rộng):

* Tích hợp với hệ thống kế toán hoặc ERP bên ngoài.
* Hỗ trợ đa chi nhánh hoặc quản lý kho nhiều địa điểm.
* Chương trình khách hàng thân thiết nâng cao (tích điểm, đổi quà).
* Giao diện khách hàng tự phục vụ (Kiosk).
* Hệ thống đặt hàng trực tuyến (Online Ordering).
* Quản lý nhà cung cấp và đơn đặt hàng mua (Purchase Orders).
* Phân tích dữ liệu chuyên sâu (Business Intelligence).

# **3. Đối tượng người dùng (User Roles)**

Hệ thống POS được thiết kế để phục vụ nhu cầu của các nhóm người dùng chính sau, với các quyền hạn và trách nhiệm riêng biệt:

## **3.1. Quản trị viên (Admin)**

* Mô tả: Là người dùng có quyền cao nhất trong hệ thống, chịu trách nhiệm quản lý và cấu hình toàn bộ ứng dụng.
* Trách nhiệm chính:
  + Quản lý thông tin và tài khoản của tất cả nhân viên, bao gồm thêm mới, chỉnh sửa, khóa/mở khóa tài khoản.
  + Phân công và quản lý các vai trò (roles) trong hệ thống.
  + Cấu hình các thiết lập chung của cửa hàng (tên, địa chỉ, thuế suất, tiền tệ, v.v.).
  + Truy cập tất cả các loại báo cáo và thống kê.
  + Xem và phân tích nhật ký hoạt động (Audit Logs) của hệ thống.
  + Quản lý toàn bộ danh mục sản phẩm và sản phẩm.
  + Quản lý chương trình khuyến mãi.
* Tương tác: Chủ yếu thông qua một khu vực quản trị chuyên biệt trên giao diện Frontend, yêu cầu xác thực và phân quyền nghiêm ngặt từ Backend.

## **3.2. Nhân viên Bán hàng (Cashier/Sales Staff)**

* Mô tả: Là người dùng trực tiếp tương tác với khách hàng và xử lý các giao dịch bán hàng hàng ngày.
* Trách nhiệm chính:
  + Tạo và quản lý các hóa đơn bán hàng.
  + Thêm, chỉnh sửa số lượng, và xóa sản phẩm khỏi hóa đơn.
  + Áp dụng mã khuyến mãi vào hóa đơn.
  + Xử lý các loại thanh toán (tiền mặt, thẻ, ví điện tử).
  + Tìm kiếm và thêm thông tin khách hàng vào hóa đơn.
  + Quản lý trạng thái bàn (nếu áp dụng).
  + In hóa đơn cho khách hàng.
  + Tra cứu thông tin cơ bản của sản phẩm và giá.
* Tương tác: Sử dụng giao diện bán hàng được tối ưu hóa cho tốc độ và hiệu quả, dễ sử dụng trên thiết bị POS chuyên dụng hoặc máy tính bảng.

## **3.3. Quản lý Kho (Inventory Manager)**

* Mô tả: Là người dùng chịu trách nhiệm chính trong việc quản lý và theo dõi hàng hóa.
* Trách nhiệm chính:
  + Thực hiện các giao dịch nhập kho (nhập hàng từ nhà cung cấp).
  + Thực hiện các giao dịch xuất kho (chuyển hàng, xuất bỏ, xuất nội bộ).
  + Xem và kiểm tra số lượng tồn kho hiện tại của tất cả sản phẩm.
  + Xem lịch sử chi tiết của các giao dịch nhập và xuất kho.
  + Được cảnh báo về các sản phẩm có tồn kho thấp.
* Tương tác: Sử dụng giao diện quản lý kho chuyên biệt, có quyền chỉnh sửa số lượng tồn kho thông qua các giao dịch nhập/xuất.

## **3.4. Khách hàng (Customer)**

* Mô tả: Là người dùng cuối cùng sử dụng dịch vụ của cửa hàng.
* Trách nhiệm chính: Thực hiện giao dịch mua hàng.
* Tương tác: Không tương tác trực tiếp với giao diện quản trị của hệ thống. Thông tin cá nhân và lịch sử mua hàng của họ được lưu trữ và quản lý trong CSDL để phục vụ các chương trình chăm sóc khách hàng và báo cáo kinh doanh.

# **4. Chức năng chính (Functional Requirements)**

Các yêu cầu chức năng mô tả "những gì hệ thống phải làm". Chúng được xây dựng dựa trên cấu trúc CSDL đã định nghĩa và sẽ được triển khai thông qua các API Backend bằng Java (Spring Boot) và giao diện Frontend bằng Vue.js.

## **4.1. Quản lý Bán hàng (Sales Management)**

* FR-SM-001: Tạo Hóa đơn mới:
  + Hệ thống cho phép nhân viên bán hàng khởi tạo một giao dịch bán hàng mới.
  + API Backend sẽ tạo một bản ghi hóa đơn mới với trạng thái ban đầu là 'pending'.
  + Frontend sẽ hiển thị giao diện nhập liệu cho hóa đơn mới.
* FR-SM-002: Thêm/Bớt sản phẩm vào hóa đơn:
  + Thêm sản phẩm: Nhân viên có thể tìm kiếm sản phẩm (bằng barcode, tên sản phẩm) hoặc duyệt qua các danh mục trên giao diện Vue.js. Sau đó, chọn sản phẩm và nhập số lượng để thêm vào hóa đơn.
    - Backend API sẽ kiểm tra số lượng tồn kho của sản phẩm; nếu không đủ, sẽ trả về lỗi và thông báo cho Frontend.
    - Khi thêm, API sẽ tạo một bản ghi invoice\_item mới trong CSDL.
  + Điều chỉnh số lượng: Nhân viên có thể thay đổi số lượng của một sản phẩm đã có trong hóa đơn.
    - Backend API sẽ cập nhật quantity và tự động tính toán lại item\_total trong invoice\_item.
    - Backend API cũng sẽ kiểm tra lại tồn kho sau khi điều chỉnh.
  + Xóa sản phẩm: Nhân viên có thể xóa một sản phẩm đã thêm khỏi hóa đơn.
    - Backend API sẽ xóa bản ghi invoice\_item tương ứng.
* FR-SM-003: Quản lý khách hàng trong hóa đơn:
  + Nhân viên có thể tìm kiếm và gán một khách hàng hiện có vào hóa đơn.
  + Nếu là khách hàng mới hoặc khách vãng lai, hóa đơn có thể được tạo mà không gán customer\_id (NULL).
  + API Backend sẽ liên kết customer\_id với hóa đơn.
* FR-SM-004: Quản lý bàn trong hóa đơn (nếu áp dụng):
  + Nhân viên có thể chọn một bàn khả dụng và gán vào hóa đơn.
  + Khi bàn được gán, trạng thái của bàn trong bảng tables sẽ được cập nhật thành 'occupied' (hoặc tương tự).
  + Khi hóa đơn được thanh toán xong hoặc hủy, trạng thái bàn sẽ được cập nhật lại thành 'available'.
* FR-SM-005: Áp dụng khuyến mãi:
  + Nhân viên có thể nhập mã khuyến mãi (discount code) vào giao diện hóa đơn.
  + Backend API sẽ gọi dịch vụ khuyến mãi để kiểm tra:
    - Tính hợp lệ của mã khuyến mãi (tồn tại, không hết hạn).
    - Loại khuyến mãi và giá trị giảm giá.
    - Điều kiện áp dụng (ví dụ: minimum\_order\_amount).
    - Giới hạn giảm giá tối đa (maximum\_discount\_amount).
  + Nếu hợp lệ, Backend sẽ tính toán discount\_amount và gán discount\_id vào hóa đơn, sau đó cập nhật tổng hóa đơn.
* FR-SM-006: Tính toán và hiển thị tổng tiền hóa đơn:
  + Backend API sẽ tự động tính toán sub\_total (tổng tiền các mặt hàng trước giảm giá và thuế), discount\_amount (số tiền giảm giá thực tế), tax\_amount (số tiền thuế dựa trên tax\_rate của hệ thống và sub\_total), và total\_amount (tổng tiền cuối cùng khách hàng phải trả).
  + Các giá trị này sẽ được cập nhật liên tục và hiển thị trên giao diện Vue.js khi có thay đổi trong invoice\_items hoặc discount\_id.
* FR-SM-007: Xử lý Thanh toán:
  + Hệ thống hỗ trợ nhiều phương thức thanh toán: Tiền mặt, Thẻ (visa/mastercard), Ví điện tử (Momo, ZaloPay, v.v.).
  + Nhân viên chọn phương thức thanh toán và nhập số tiền khách hàng trả.
  + Backend API sẽ xác thực số tiền thanh toán (lớn hơn 0).
  + Hệ thống tính toán và hiển thị số tiền thừa trả lại cho khách hàng.
  + Backend API ghi nhận giao dịch thanh toán vào bảng payments, liên kết với invoice\_id.
  + Sau khi thanh toán hoàn tất, trạng thái của hóa đơn trong bảng invoices sẽ được cập nhật thành 'completed'.
* FR-SM-008: In hóa đơn:
  + Hệ thống cho phép nhân viên in hóa đơn (receipt) cho khách hàng sau khi giao dịch thành công.
  + Frontend sẽ gửi yêu cầu in hóa đơn đến Backend (hoặc một dịch vụ in ấn riêng), với thông tin chi tiết về hóa đơn.
  + Hóa đơn in ra phải bao gồm thông tin cửa hàng, chi tiết sản phẩm, số lượng, giá, tổng tiền, thuế, giảm giá, phương thức thanh toán và cảm ơn khách hàng.

## **4.2. Quản lý Sản phẩm (Product Management)**

* FR-PM-001: Xem danh sách sản phẩm:
  + Hệ thống cho phép người dùng (Admin, Quản lý kho, Nhân viên bán hàng) truy cập danh sách tất cả sản phẩm.
  + API Backend sẽ hỗ trợ phân trang (pagination), tìm kiếm theo tên/barcode, lọc theo danh mục, và sắp xếp theo các tiêu chí (tên, giá, tồn kho).
  + Frontend Vue.js sẽ hiển thị dữ liệu một cách rõ ràng và cung cấp các bộ lọc, chức năng tìm kiếm.
* FR-PM-002: Thêm sản phẩm mới:
  + Admin hoặc Quản lý kho có thể thêm sản phẩm mới thông qua giao diện Frontend.
  + Backend API (Spring Boot) sẽ nhận dữ liệu và thực hiện kiểm tra đầu vào (validation):
    - barcode, name phải là duy nhất và không trống.
    - price phải là số dương.
    - quantity phải là số nguyên không âm.
    - category\_id phải tồn tại trong bảng categories.
  + Backend sẽ tạo bản ghi mới trong bảng products.
* FR-PM-003: Sửa thông tin sản phẩm:
  + Admin hoặc Quản lý kho có thể cập nhật thông tin của một sản phẩm hiện có.
  + Backend API sẽ thực hiện cập nhật bản ghi trong bảng products, bao gồm cả việc tăng version để hỗ trợ optimistic locking.
  + Các ràng buộc validation tương tự như khi thêm mới sẽ được áp dụng.
* FR-PM-004: Quản lý trạng thái sản phẩm:
  + Admin hoặc Quản lý kho có thể chuyển đổi trạng thái is\_active của sản phẩm giữa TRUE (đang kinh doanh) và FALSE (ngừng kinh doanh).
  + Sản phẩm không hoạt động sẽ không hiển thị trong danh sách sản phẩm khi tạo hóa đơn nhưng vẫn được giữ lại trong CSDL cho mục đích lịch sử.
* FR-PM-005: Quản lý Danh mục sản phẩm:
  + Admin có thể thêm, sửa, xóa các danh mục sản phẩm (ví dụ: "Trà sữa", "Cà phê", "Đồ ăn vặt").
  + Backend API sẽ đảm bảo tính duy nhất của tên danh mục.
  + Khi xóa danh mục, hệ thống sẽ cảnh báo nếu có sản phẩm đang thuộc danh mục đó.

## **4.3. Quản lý Kho (Inventory Management)**

* FR-IM-001: Nhập kho:
  + Quản lý kho có thể ghi nhận việc nhập sản phẩm vào kho qua giao diện Frontend.
  + Backend API sẽ nhận product\_id, quantity, note, employee\_id và tạo bản ghi trong bảng stock\_in.
  + Sau khi ghi nhận stock\_in, Backend sẽ tự động tăng số lượng quantity trong bảng products của sản phẩm tương ứng.
* FR-IM-002: Xuất kho:
  + Quản lý kho có thể ghi nhận việc xuất sản phẩm khỏi kho (không phải qua bán hàng).
  + Backend API sẽ nhận product\_id, quantity, note, employee\_id và tạo bản ghi trong bảng stock\_out.
  + Sau khi ghi nhận stock\_out, Backend sẽ tự động giảm số lượng quantity trong bảng products của sản phẩm tương ứng, đảm bảo quantity không bị âm.
* FR-IM-003: Xem lịch sử nhập/xuất kho:
  + Quản lý kho/Admin có thể xem chi tiết tất cả các giao dịch nhập kho và xuất kho, bao gồm sản phẩm, số lượng, ghi chú, và nhân viên thực hiện.
  + Hỗ trợ lọc theo thời gian, sản phẩm, hoặc nhân viên.
* FR-IM-004: Cảnh báo tồn kho thấp (Tính năng mở rộng):
  + Hệ thống sẽ có khả năng cài đặt ngưỡng tồn kho tối thiểu cho mỗi sản phẩm.
  + Khi quantity của một sản phẩm giảm xuống dưới ngưỡng này, hệ thống sẽ hiển thị cảnh báo trên giao diện quản lý kho và/hoặc gửi thông báo (ví dụ: email) đến Quản lý kho/Admin.

## **4.4. Quản lý Khách hàng (Customer Management)**

* FR-CM-001: Xem danh sách khách hàng:
  + Nhân viên/Admin có thể xem danh sách tất cả khách hàng.
  + API Backend hỗ trợ phân trang, tìm kiếm theo tên, số điện thoại, email và sắp xếp.
* FR-CM-002: Thêm khách hàng mới:
  + Nhân viên/Admin có thể thêm thông tin khách hàng mới.
  + Backend API sẽ kiểm tra phone và email là duy nhất (nếu được cung cấp).
  + Tạo bản ghi mới trong bảng customers.
* FR-CM-003: Sửa thông tin khách hàng:
  + Nhân viên/Admin có thể cập nhật thông tin của một khách hàng hiện có.
  + Backend API sẽ cập nhật bản ghi trong bảng customers và tăng version.
* FR-CM-004: Tìm kiếm khách hàng:
  + Hỗ trợ tìm kiếm khách hàng nhanh chóng trong giao diện bán hàng để gán vào hóa đơn.
* FR-CM-005: Xem lịch sử mua hàng của khách hàng:
  + Cho phép xem danh sách các hóa đơn đã đặt của một khách hàng cụ thể, bao gồm tổng tiền và ngày mua.
  + Có thể truy cập chi tiết từng hóa đơn từ danh sách này.

## **4.5. Quản lý Nhân viên & Quyền hạn (Employee & Role Management)**

* FR-ERM-001: Xem danh sách nhân viên:
  + Admin có thể xem thông tin chi tiết của tất cả nhân viên.
  + API Backend hỗ trợ tìm kiếm và phân trang.
* FR-ERM-002: Thêm nhân viên mới:
  + Admin có thể thêm nhân viên mới với các thông tin: Tên, Tên đăng nhập (username), Mật khẩu (được xử lý an toàn), Vai trò (role\_id), Số điện thoại, Email, Trạng thái hoạt động (is\_active).
  + Backend API sẽ kiểm tra tính duy nhất của username, phone, email.
  + Mật khẩu sẽ được băm (hash) và thêm salt trước khi lưu vào CSDL.
* FR-ERM-003: Sửa thông tin nhân viên:
  + Admin có thể cập nhật thông tin của nhân viên hiện có (trừ username).
  + Cho phép đặt lại mật khẩu cho nhân viên.
  + Backend API sẽ cập nhật bản ghi trong bảng employees và tăng version.
* FR-ERM-004: Quản lý vai trò (Roles):
  + Admin có thể thêm, sửa, xóa các vai trò người dùng (ví dụ: "Admin", "Nhân viên", "Quản lý kho").
  + Backend API sẽ đảm bảo tên vai trò là duy nhất.
  + Việc xóa vai trò sẽ bị hạn chế nếu có nhân viên đang sử dụng vai trò đó.
* FR-ERM-005: Phân quyền truy cập (RBAC):
  + Backend API sẽ kiểm tra quyền của người dùng dựa trên role\_id của họ trước khi cho phép truy cập vào các API endpoint hoặc thực hiện các hành động cụ thể.
  + Frontend (Vue.js) sẽ điều chỉnh hiển thị giao diện (menu, nút bấm, form) dựa trên vai trò của người dùng đã đăng nhập để chỉ hiển thị các chức năng mà họ có quyền.

## **4.6. Quản lý Bàn (Table Management - If Applicable)**

* FR-TM-001: Xem danh sách bàn: Nhân viên có thể xem trạng thái hiện tại (available, occupied, reserved, cleaning, out\_of\_service) và sức chứa của tất cả các bàn trong cửa hàng.
* FR-TM-002: Cập nhật trạng thái bàn:
  + Nhân viên hoặc Admin có thể thay đổi trạng thái của bàn thủ công (ví dụ: chuyển từ 'cleaning' sang 'available').
  + Hệ thống sẽ tự động cập nhật trạng thái bàn khi hóa đơn được gán/hủy gán hoặc thanh toán/hủy.
* FR-TM-003: Gán/Hủy gán bàn cho hóa đơn: Hóa đơn có thể được liên kết với một bàn cụ thể khi tạo, và có thể được hủy gán khi kết thúc giao dịch.

## **4.7. Quản lý Khuyến mãi (Discount Management)**

* FR-DM-001: Xem danh sách khuyến mãi: Admin có thể xem tất cả các chương trình khuyến mãi hiện có, bao gồm mã code, mô tả, loại giảm giá, giá trị, ngày bắt đầu/kết thúc, và trạng thái hoạt động.
* FR-DM-002: Thêm khuyến mãi mới:
  + Admin có thể tạo một chương trình khuyến mãi mới với các thông số: Mã code (duy nhất), Mô tả, Loại giảm giá (discount\_type\_id từ bảng discount\_types), Giá trị (ví dụ: 10 cho 10%, hoặc 20000 cho 20k), Ngày bắt đầu, Ngày kết thúc.
  + Cũng có thể thiết lập các điều kiện bổ sung như minimum\_order\_amount (số tiền tối thiểu của hóa đơn để áp dụng) và maximum\_discount\_amount (giới hạn giảm giá tối đa).
  + Backend API sẽ kiểm tra tính hợp lệ của dữ liệu đầu vào.
* FR-DM-003: Sửa thông tin khuyến mãi: Admin có thể cập nhật chi tiết của một chương trình khuyến mãi hiện có (ví dụ: thay đổi giá trị, ngày kết thúc).
* FR-DM-004: Kích hoạt/Vô hiệu hóa khuyến mãi: Admin có thể thay đổi trạng thái active của một khuyến mãi để bật/tắt khả năng áp dụng.

## **4.8. Báo cáo & Thống kê (Reporting & Analytics)**

* FR-RA-001: Báo cáo Doanh thu:
  + Hệ thống cung cấp báo cáo tổng doanh thu theo các khoảng thời gian tùy chỉnh (hàng ngày, hàng tuần, hàng tháng, hàng quý, hàng năm).
  + Hiển thị doanh thu ròng (sau giảm giá và thuế) và doanh thu theo từng phương thức thanh toán.
  + Backend sẽ tổng hợp dữ liệu từ bảng invoices và payments.
* FR-RA-002: Báo cáo Sản phẩm bán chạy:
  + Liệt kê các sản phẩm bán chạy nhất theo số lượng bán ra và/hoặc doanh thu trong một khoảng thời gian nhất định.
  + Backend sẽ tổng hợp dữ liệu từ bảng invoice\_items.
* FR-RA-003: Báo cáo Tồn kho:
  + Tổng quan về số lượng tồn kho hiện tại của từng sản phẩm.
  + Hiển thị tổng giá trị tồn kho của cửa hàng.
  + Báo cáo các sản phẩm dưới ngưỡng tồn kho an toàn.
* FR-RA-004: Lọc và xuất báo cáo:
  + Người dùng có thể áp dụng các bộ lọc (ví dụ: theo ngày, danh mục sản phẩm, nhân viên) để tạo báo cáo tùy chỉnh.
  + Hệ thống cho phép xuất dữ liệu báo cáo ra các định dạng phổ biến như CSV hoặc PDF để phân tích ngoại tuyến.

## **4.9. Cấu hình Hệ thống (System Configuration)**

* FR-SC-001: Cập nhật thông tin cửa hàng: Admin có thể chỉnh sửa các thông tin cơ bản của cửa hàng như tên, địa chỉ, số điện thoại, email, và đường dẫn logo.
* FR-SC-002: Cấu hình thuế: Admin có thể thiết lập tỷ lệ thuế áp dụng cho tất cả các giao dịch bán hàng.
* FR-SC-003: Cấu hình đơn vị tiền tệ: Admin có thể chọn đơn vị tiền tệ mặc định hiển thị trong hệ thống (ví dụ: VND).
* FR-SC-004: Cấu hình tiền tố hóa đơn: Admin có thể thiết lập tiền tố cho mã hóa đơn để dễ dàng nhận diện (ví dụ: "INV-").
* FR-SC-005: Cấu hình ngôn ngữ: Admin có thể chọn ngôn ngữ hiển thị mặc định của hệ thống (ví dụ: Tiếng Việt, Tiếng Anh).

## **4.10. Xác thực & Bảo mật (Authentication & Security)**

* FR-AS-001: Đăng nhập/Đăng xuất an toàn:
  + Hệ thống yêu cầu tất cả người dùng phải xác thực bằng tên đăng nhập và mật khẩu để truy cập các chức năng.
  + Backend (Spring Boot) sẽ sử dụng một thuật toán băm mạnh (ví dụ: BCrypt) và salt để lưu trữ mật khẩu. Mật khẩu không bao giờ được lưu dưới dạng văn bản thuần túy.
  + Sử dụng cơ chế Token-Based Authentication (ví dụ: JWT) để quản lý phiên làm việc của người dùng. Mỗi yêu cầu API từ Frontend sẽ yêu cầu một token hợp lệ.
  + Chức năng đăng xuất sẽ vô hiệu hóa token phiên hiện tại.
  + Hệ thống ghi lại thời gian đăng nhập/đăng xuất, địa chỉ IP và thông tin thiết bị (User Agent) vào bảng user\_sessions để theo dõi và kiểm soát.
* FR-AS-002: Phân quyền truy cập theo vai trò (Role-Based Access Control - RBAC):
  + Mọi yêu cầu API từ Frontend đến Backend (Spring Boot) sẽ được kiểm tra quyền hạn dựa trên vai trò của người dùng (role\_id trong JWT token).
  + Chỉ những người dùng có vai trò phù hợp mới được phép truy cập vào các tài nguyên và thực hiện các hành động cụ thể (ví dụ: chỉ Admin mới có thể thêm/sửa nhân viên, chỉ Quản lý kho mới được nhập/xuất kho).
  + Frontend (Vue.js) sẽ có logic để điều chỉnh hiển thị giao diện (menu, nút bấm, trường nhập liệu) dựa trên quyền của người dùng để tránh hiển thị các chức năng mà người dùng không có quyền.
* FR-AS-003: Phòng chống tấn công phổ biến:
  + SQL Injection: Backend (Spring Boot với JPA/Hibernate hoặc Spring JDBC) sẽ sử dụng Prepared Statements hoặc ORM để ngăn chặn các cuộc tấn công SQL Injection.
  + Cross-Site Scripting (XSS): Frontend (Vue.js) sẽ đảm bảo rằng tất cả dữ liệu hiển thị từ Backend đều được thoát (sanitized) đúng cách. Backend sẽ kiểm tra và làm sạch dữ liệu đầu vào.
  + Cross-Site Request Forgery (CSRF): Backend sẽ triển khai các biện pháp chống CSRF (ví dụ: sử dụng CSRF tokens) để bảo vệ các yêu cầu POST/PUT/DELETE quan trọng.
  + Session Hijacking: Sử dụng HTTPS, JWT token có thời gian sống ngắn, và các biện pháp bảo mật phiên khác để giảm thiểu rủi ro.
* FR-AS-004: Giao tiếp an toàn:
  + Toàn bộ giao tiếp giữa Frontend (Vue.js) và Backend (Spring Boot) phải được mã hóa sử dụng HTTPS để bảo vệ dữ liệu khỏi bị nghe trộm hoặc giả mạo.
* FR-AS-005: Ghi log hoạt động (Audit Log):
  + Backend sẽ tự động ghi lại các hành động quan trọng (như thêm/sửa/xóa sản phẩm, nhân viên, khách hàng, tạo/thanh toán hóa đơn, đăng nhập/đăng xuất) vào bảng audit\_logs.
  + Log sẽ bao gồm thông tin về nhân viên thực hiện, loại hành động, bảng bị ảnh hưởng, ID của bản ghi bị ảnh hưởng, dữ liệu cũ/mới (dạng JSON), và thời gian thực hiện.
  + Chỉ Admin mới có quyền xem các nhật ký này để phục vụ mục đích kiểm toán và khắc phục sự cố.

# **5. Yêu cầu phi chức năng (Non-Functional Requirements)**

Các yêu cầu phi chức năng mô tả "hệ thống phải hoạt động tốt như thế nào".

## **5.1. Hiệu suất (Performance)**

* NFR-PERF-001: Thời gian phản hồi API:
  + Hầu hết các API endpoint đọc (GET) dữ liệu cơ bản (ví dụ: danh sách sản phẩm, chi tiết khách hàng) phải có thời gian phản hồi dưới 300ms.
  + Các API endpoint ghi (POST, PUT, DELETE) dữ liệu cơ bản (ví dụ: thêm sản phẩm, cập nhật thông tin khách hàng) phải có thời gian phản hồi dưới 500ms.
  + Các API liên quan đến tạo/thanh toán hóa đơn phải có thời gian phản hồi dưới 1 giây.
  + Các API tạo báo cáo phức tạp có thể có thời gian phản hồi lên đến 3-5 giây, nhưng cần có cơ chế hiển thị trạng thái "đang tải" cho người dùng.
* NFR-PERF-002: Tải trọng đồng thời:
  + Hệ thống Backend (Spring Boot) phải có khả năng xử lý ít nhất 50 yêu cầu HTTP đồng thời (concurrent requests) mà không làm thời gian phản hồi tăng quá 20% so với hiệu suất cơ bản.
  + Tối ưu hóa các truy vấn CSDL để đảm bảo hiệu suất ngay cả với lượng dữ liệu lớn (ví dụ: hàng chục nghìn sản phẩm, hàng nghìn hóa đơn).
* NFR-PERF-003: Tốc độ tải trang Frontend:
  + Giao diện người dùng (Vue.js) phải tải hoàn chỉnh và sẵn sàng để tương tác trong vòng dưới 2 giây trên kết nối internet tiêu chuẩn (ví dụ: 10 Mbps).
  + Thời gian chuyển đổi giữa các trang trong ứng dụng phải dưới 500ms.

## **5.2. Bảo mật (Security)**

* NFR-SEC-001: Bảo vệ dữ liệu nhạy cảm:
  + Tất cả mật khẩu người dùng phải được băm mạnh mẽ (ví dụ: sử dụng BCrypt) với salt và lưu trữ trong CSDL. Không bao giờ lưu trữ mật khẩu plaintext.
  + Thông tin cá nhân khách hàng (số điện thoại, email, địa chỉ) cần được bảo vệ và chỉ truy cập bởi người dùng được ủy quyền.
* NFR-SEC-002: Kiểm soát truy cập chặt chẽ:
  + Mọi yêu cầu API đến Backend đều phải được xác thực và phân quyền. Các API công khai (ví dụ: đăng nhập) sẽ được giới hạn.
  + Frontend sẽ ẩn hoặc vô hiệu hóa các chức năng mà người dùng không có quyền truy cập.
* NFR-SEC-003: Phòng chống lỗ hổng OWASP Top 10: Hệ thống sẽ được phát triển với các biện pháp phòng ngừa các lỗ hổng bảo mật phổ biến như SQL Injection, Broken Authentication, Sensitive Data Exposure, XSS, Insecure Deserialization, v.v.
* NFR-SEC-004: Giao tiếp mã hóa: Toàn bộ giao tiếp giữa Frontend và Backend, và giữa Backend và CSDL, phải được mã hóa sử dụng HTTPS/SSL/TLS để đảm bảo tính toàn vẹn và bảo mật của dữ liệu.
* NFR-SEC-005: Kiểm tra phiên (Session Management): JWT tokens sẽ có thời gian hết hạn hợp lý và có thể bị thu hồi khi đăng xuất hoặc khi có sự cố bảo mật.

## **5.3. Khả năng mở rộng (Scalability)**

* NFR-SCL-001: Kiến trúc Microservices (hoặc Module hóa rõ ràng): Hệ thống sẽ được thiết kế với các module rõ ràng (hoặc dịch vụ nhỏ) để dễ dàng mở rộng và bảo trì độc lập.
* NFR-SCL-002: Khả năng mở rộng ngang (Horizontal Scalability): Backend (Spring Boot) sẽ được thiết kế để có thể dễ dàng triển khai nhiều instance (phiên bản) trên nhiều server khác nhau, sử dụng Load Balancer để phân phối tải.
* NFR-SCL-003: CSDL linh hoạt: Cấu trúc CSDL được thiết kế chuẩn hóa, cho phép thêm các trường/bảng mới và điều chỉnh mà không gây gián đoạn lớn cho các thành phần hiện có. Sử dụng các kỹ thuật như indexing để tối ưu hiệu suất khi CSDL lớn dần.

## **5.4. Khả năng sử dụng (Usability)**

* NFR-USBL-001: Giao diện trực quan và thân thiện: Giao diện người dùng (Vue.js) phải có thiết kế hiện đại, sạch sẽ, và dễ dàng điều hướng. Các biểu tượng, nhãn và thông báo phải rõ ràng.
* NFR-USBL-002: Tối ưu hóa quy trình bán hàng: Quy trình tạo hóa đơn và thanh toán phải được tối giản, cho phép nhân viên thực hiện nhanh chóng chỉ với vài cú nhấp chuột hoặc thao tác phím.
* NFR-USBL-003: Phản hồi người dùng: Hệ thống phải cung cấp phản hồi tức thì cho các hành động của người dùng (ví dụ: thông báo thành công, lỗi validation, trạng thái tải).
* NFR-USBL-004: Xử lý lỗi thân thiện: Thông báo lỗi phải rõ ràng, dễ hiểu và cung cấp gợi ý về cách khắc phục (thay vì mã lỗi kỹ thuật).

## **5.5. Độ tin cậy (Reliability)**

* NFR-REL-001: Thời gian hoạt động (Uptime): Hệ thống mục tiêu có thời gian hoạt động (uptime) ít nhất 99.9% trong giờ làm việc (ví dụ: tối đa 45 phút ngừng hoạt động mỗi tháng).
* NFR-REL-002: Phục hồi lỗi: Trong trường hợp xảy ra lỗi hệ thống hoặc mất điện đột ngột, dữ liệu phải được phục hồi về trạng thái cuối cùng nhất quán, giảm thiểu mất mát dữ liệu (RPO - Recovery Point Objective) và thời gian phục hồi (RTO - Recovery Time Objective) ở mức thấp nhất có thể.
* NFR-REL-003: Sao lưu dữ liệu tự động: Cơ sở dữ liệu phải được sao lưu tự động và định kỳ (ví dụ: hàng ngày) sang một vị trí an toàn, có khả năng phục hồi dữ liệu khi cần.
* NFR-REL-004: Ghi log và giám sát: Hệ thống sẽ có cơ chế ghi log chi tiết (Backend và Frontend) và giám sát hiệu suất để nhanh chóng phát hiện và khắc phục sự cố.

## **5.6. Môi trường triển khai (Deployment Environment)**

* NFR-DEP-001: Hệ điều hành Server: Ưu tiên Linux (Ubuntu Server LTS hoặc CentOS Stream) cho môi trường sản xuất do tính ổn định và bảo mật cao. Có thể phát triển trên Windows/macOS.
* NFR-DEP-002: Cơ sở dữ liệu: MySQL (phiên bản 8.0 trở lên) hoặc PostgreSQL (phiên bản mới nhất) sẽ là lựa chọn chính. Cần tối ưu hóa CSDL (indexing, query optimization).
* NFR-DEP-003: Công nghệ Backend:
  + Ngôn ngữ: Java 17+ (phiên bản LTS mới nhất).
  + Framework: Spring Boot (phiên bản mới nhất) với Spring Data JPA để tương tác CSDL.
  + Build Tool: Maven hoặc Gradle.
* NFR-DEP-004: Công nghệ Frontend:
  + Framework: Vue.js (phiên bản 3.x).
  + Build Tool: Vite (hoặc Vue CLI nếu cần).
  + Ngôn ngữ: JavaScript/TypeScript.
* NFR-DEP-005: Công cụ kiểm soát phiên bản: Git (được quản lý trên GitHub, GitLab hoặc Bitbucket) cho quản lý mã nguồn và cộng tác.
* NFR-DEP-006: Containerization (Khuyến nghị): Sử dụng Docker để đóng gói ứng dụng Backend và Frontend, giúp việc triển khai và quản lý môi trường nhất quán, dễ dàng hơn.
* NFR-DEP-007: Web Server (cho Frontend): Nginx hoặc Apache để phục vụ các file tĩnh của Frontend và có thể dùng làm Reverse Proxy cho Backend.
* NFR-DEP-008: Môi trường CI/CD (Tùy chọn): Cấu hình các pipeline CI/CD (Continuous Integration/Continuous Deployment) để tự động hóa quá trình build, test và deploy (ví dụ: Jenkins, GitHub Actions, GitLab CI/CD).