Mục lục

**1. Đặc tả yêu cầu người dùng**

**1.1 Mục tiêu của hệ thống**

**1.2. Phạm vi của hệ thống**

**1.3 Đối tượng người dùng**

**1.4 Kịch bản sử dụng (Use Cases)**

* Mô tả các tình huống thực tế
* Minh hoạ qua sơ đồ Use Case

**2. Phân tích yêu cầu**

**2.1 Tổng quan về yêu cầu**

* Tóm tắt yêu cầu từ phần đặc tả
* Các giả định ban đầu

**2.2 Mô hình hóa yêu cầu**

* Sơ đồ Use Case chi tiết
* Sơ đồ lớp (Class Diagram)
* Sơ đồ luồng dữ liệu (DFD - Data Flow Diagram)

**2.3 Đánh giá yêu cầu**

* Độ ưu tiên của các yêu cầu
* Đánh giá tính khả thi

**2.4 Rủi ro và hạn chế**

* Rủi ro trong việc phát triển và thực hiện yêu cầu
* Hạn chế kỹ thuật hoặc tổ chức

Nội dung

**Chương 1: Đặc tả Yêu Cầu Người Dùng**

**1.1. Mục tiêu của hệ thống**

**1.2. Phạm vi của hệ thống**

**1.3. Đối tượng người dùng**

**1.4 Kịch bản sử dụng (Use Cases)**

**2. Phân tích yêu cầu**

**2.1 Tổng quan về yêu cầu**

**2.2 Mô hình hóa yêu cầu**

* + .

**2.3 Đánh giá yêu cầu**

**2.6. Các trường hợp ngoại lệ**

Trong quá trình vận hành, hệ thống quản lý bán hàng có thể gặp phải các tình huống bất thường. Việc xác định trước các trường hợp này và đưa ra cách xử lý phù hợp sẽ giúp đảm bảo hệ thống hoạt động ổn định, giảm thiểu rủi ro và tăng độ tin cậy.

**Các trường hợp ngoại lệ và cách xử lý:**

1. **Sản phẩm hết hàng khi thêm vào đơn hàng:**
   * **Mô tả:** Khi nhân viên cố gắng thêm một sản phẩm vào đơn hàng nhưng số lượng sản phẩm trong kho không đủ đáp ứng.
   * **Xử lý:**
     + Hiển thị thông báo lỗi: "Sản phẩm không đủ số lượng trong kho."
     + Tự động cập nhật trạng thái sản phẩm (nếu số lượng = 0 → trạng thái "Hết hàng").
     + Gợi ý cho nhân viên nhập thêm hàng vào kho.
2. **Lỗi kết nối cơ sở dữ liệu:**
   * **Mô tả:** Hệ thống không thể truy cập vào cơ sở dữ liệu, khiến toàn bộ các chức năng như xem sản phẩm, tạo đơn hàng, hoặc báo cáo bị gián đoạn.
   * **Xử lý:**
     + Ghi log lỗi để kiểm tra và khắc phục sau.
     + Hiển thị thông báo: "Lỗi kết nối hệ thống. Vui lòng thử lại sau."
     + Hướng dẫn nhân viên liên hệ bộ phận kỹ thuật nếu lỗi kéo dài.
3. **Nhập sai dữ liệu đầu vào:**
   * **Mô tả:** Người dùng nhập dữ liệu không hợp lệ khi tạo sản phẩm, đơn hàng hoặc tài khoản (ví dụ: giá sản phẩm âm, số lượng không phải số nguyên).
   * **Xử lý:**
     + Hiển thị thông báo: "Dữ liệu không hợp lệ. Vui lòng kiểm tra và thử lại."
     + Ngăn người dùng lưu thông tin sai vào hệ thống.
     + Cung cấp các hướng dẫn cụ thể để sửa lỗi.
4. **Trùng mã sản phẩm hoặc đơn hàng:**
   * **Mô tả:** Nhân viên vô tình nhập trùng mã sản phẩm hoặc đơn hàng trong hệ thống.
   * **Xử lý:**
     + Hiển thị thông báo: "Mã này đã tồn tại. Vui lòng nhập mã khác."
     + Không cho phép lưu thông tin mới nếu mã bị trùng.
     + Đề xuất tự động tạo mã mới (nếu cần).
5. **Đơn hàng thanh toán thất bại:**
   * **Mô tả:** Nhân viên xử lý thanh toán nhưng gặp lỗi (ví dụ: lỗi máy POS, thanh toán không thành công).
   * **Xử lý:**
     + Hiển thị thông báo: "Thanh toán thất bại. Vui lòng kiểm tra phương thức thanh toán."
     + Ghi log chi tiết lỗi để kiểm tra sau.
     + Cho phép nhân viên chọn phương thức thanh toán khác hoặc tạm giữ đơn hàng.
6. **Sản phẩm bị xoá trong khi đơn hàng đang xử lý:**
   * **Mô tả:** Nhân viên xóa một sản phẩm khỏi hệ thống trong khi sản phẩm này đã được thêm vào một đơn hàng đang xử lý.
   * **Xử lý:**
     + Hiển thị cảnh báo: "Sản phẩm đang được sử dụng trong đơn hàng. Không thể xoá."
     + Yêu cầu nhân viên cập nhật lại đơn hàng trước khi xóa sản phẩm.
7. **Lỗi phân quyền:**
   * **Mô tả:** Nhân viên cố gắng truy cập hoặc thực hiện các chức năng vượt quyền hạn (ví dụ: nhân viên cố xoá tài khoản Admin).
   * **Xử lý:**
     + Hiển thị thông báo: "Bạn không có quyền thực hiện thao tác này."
     + Ghi log hành động để quản trị viên kiểm tra.
8. **Số lượng lớn giao dịch cùng lúc (quá tải hệ thống):**
   * **Mô tả:** Nhiều giao dịch cùng lúc khiến hệ thống bị chậm hoặc ngưng hoạt động.
   * **Xử lý:**
     + Hạn chế số lượng yêu cầu xử lý đồng thời.
     + Hiển thị thông báo: "Hệ thống đang bận. Vui lòng thử lại sau vài phút."
     + Tăng cường khả năng xử lý bằng cách tối ưu cơ sở dữ liệu hoặc sử dụng các giải pháp phân tải.
9. **Lỗi khi tạo báo cáo doanh thu:**
   * **Mô tả:** Admin yêu cầu tạo báo cáo doanh thu nhưng hệ thống không tìm thấy dữ liệu trong khoảng thời gian yêu cầu.
   * **Xử lý:**
     + Hiển thị thông báo: "Không có dữ liệu trong khoảng thời gian này."
     + Hướng dẫn Admin kiểm tra lại ngày/tháng yêu cầu báo cáo.
10. **Lỗi mất kết nối khi thao tác trên hệ thống:**
    * **Mô tả:** Nhân viên đang thao tác thì mất kết nối Internet, gây gián đoạn.
    * **Xử lý:**
      + Tự động lưu tạm thời dữ liệu thao tác (dùng local storage hoặc session).
      + Hiển thị thông báo: "Mất kết nối. Vui lòng kiểm tra lại mạng."
      + Tự động đồng bộ lại dữ liệu khi kết nối được khôi phục.

**2.4 Rủi ro và hạn chế**

Chương 3. Thiết kế chương trình  
**2.5. Sơ đồ thực thể ERD (Entity Relationship Diagram)**

Sơ đồ ERD mô tả các thực thể chính trong hệ thống, các thuộc tính của từng thực thể, và mối quan hệ giữa chúng. Đây là một công cụ quan trọng trong việc thiết kế cơ sở dữ liệu, giúp đảm bảo hệ thống được triển khai một cách logic và nhất quán.

**Thực thể chính trong hệ thống**

1. **Sản phẩm (Product):**
   * **Thuộc tính:**
     + product\_id (PK): Mã sản phẩm.
     + product\_name: Tên sản phẩm.
     + price: Giá sản phẩm.
     + quantity: Số lượng tồn kho.
     + description: Mô tả sản phẩm.
     + image\_url: Đường dẫn hình ảnh sản phẩm.
2. **Đơn hàng (Order):**
   * **Thuộc tính:**
     + order\_id (PK): Mã đơn hàng.
     + order\_date: Ngày tạo đơn hàng.
     + total\_amount: Tổng tiền đơn hàng.
     + customer\_name: Tên khách hàng (nếu cần lưu).
     + employee\_id (FK): Mã nhân viên xử lý đơn hàng.
3. **Chi tiết đơn hàng (OrderDetail):**
   * **Thuộc tính:**
     + order\_detail\_id (PK): Mã chi tiết đơn hàng.
     + order\_id (FK): Mã đơn hàng.
     + product\_id (FK): Mã sản phẩm.
     + quantity: Số lượng sản phẩm trong đơn hàng.
     + price: Giá sản phẩm tại thời điểm mua.
4. **Nhân viên (Employee):**
   * **Thuộc tính:**
     + employee\_id (PK): Mã nhân viên.
     + employee\_name: Tên nhân viên.
     + role: Vai trò (Admin/Nhân viên).
     + email: Email nhân viên.
     + password: Mật khẩu.
5. **Báo cáo doanh thu (RevenueReport):**
   * **Thuộc tính:**
     + report\_id (PK): Mã báo cáo.
     + report\_date: Ngày tạo báo cáo.
     + total\_revenue: Tổng doanh thu trong khoảng thời gian.
     + branch\_id (FK): Mã chi nhánh (nếu áp dụng cho nhiều cơ sở).

**Mối quan hệ giữa các thực thể**

1. **Quan hệ giữa Product và OrderDetail:**
   * Một sản phẩm có thể xuất hiện trong nhiều chi tiết đơn hàng (1:N).
   * Một chi tiết đơn hàng chỉ liên kết với một sản phẩm duy nhất.
2. **Quan hệ giữa Order và OrderDetail:**
   * Một đơn hàng có thể chứa nhiều chi tiết đơn hàng (1:N).
   * Một chi tiết đơn hàng chỉ thuộc về một đơn hàng.
3. **Quan hệ giữa Employee và Order:**
   * Một nhân viên có thể xử lý nhiều đơn hàng (1:N).
   * Một đơn hàng chỉ được xử lý bởi một nhân viên.
4. **Quan hệ giữa Branch (nếu có) và RevenueReport:**
   * Một chi nhánh có thể có nhiều báo cáo doanh thu (1:N).
   * Một báo cáo doanh thu thuộc về một chi nhánh duy nhất.

**Mô tả sơ đồ ERD (bằng văn bản để bạn có thể vẽ):**

1. Thực thể Product liên kết với OrderDetail qua thuộc tính product\_id.
2. Thực thể Order liên kết với OrderDetail qua thuộc tính order\_id.
3. Thực thể Employee liên kết với Order qua thuộc tính employee\_id.
4. Thực thể RevenueReport liên kết với Branch qua thuộc tính branch\_id (nếu có đa chi nhánh).
5. OrderDetail là bảng trung gian chứa thông tin liên kết giữa Order và Product.

**Ví dụ thuộc tính cụ thể cho từng thực thể:**

* **Product:**
  + product\_id: P001
  + product\_name: Laptop XYZ
  + price: 15,000,000 VND
  + quantity: 100
* **Order:**
  + order\_id: O1001
  + order\_date: 18/12/2024
  + total\_amount: 25,000,000 VND
* **OrderDetail:**
  + order\_detail\_id: OD1001
  + order\_id: O1001
  + product\_id: P001
  + quantity: 2
  + price: 15,000,000 VND

**2.7. Đề xuất giải pháp kỹ thuật**

Để đảm bảo hệ thống quản lý bán hàng vận hành hiệu quả, đáp ứng các yêu cầu chức năng và phi chức năng đã đặt ra, các giải pháp kỹ thuật được đề xuất như sau:

**1. Kiến trúc hệ thống**

* **Mô hình MVC (Model-View-Controller):**  
  Hệ thống sẽ sử dụng mô hình MVC để tách biệt giữa giao diện người dùng (View), logic xử lý (Controller), và dữ liệu (Model). Điều này giúp tăng khả năng bảo trì, mở rộng và tái sử dụng mã nguồn.
* **Kiến trúc client-server:**
  + **Frontend:** Giao diện người dùng sẽ chạy trên trình duyệt web hoặc thiết bị di động, được xây dựng bằng các công nghệ hiện đại như ReactJS hoặc VueJS.
  + **Backend:** Xử lý logic nghiệp vụ và giao tiếp với cơ sở dữ liệu, sử dụng framework như Node.js, Django, hoặc Laravel.
* **Hệ thống phân tầng (3-layer architecture):**  
  Hệ thống được chia thành các tầng chính:
  + **Presentation Layer:** Cung cấp giao diện cho người dùng (web/app).
  + **Business Logic Layer:** Chứa các quy tắc nghiệp vụ của hệ thống.
  + **Data Access Layer:** Giao tiếp với cơ sở dữ liệu.

**2. Giao tiếp giữa Frontend và Backend**

* **RESTful API:**  
  Hệ thống sẽ sử dụng REST API để giao tiếp giữa frontend và backend. Các endpoint REST API sẽ tuân thủ các nguyên tắc chuẩn để đảm bảo tính nhất quán và dễ bảo trì.
  + **Ví dụ:**
    - GET /products - Lấy danh sách sản phẩm.
    - POST /orders - Tạo đơn hàng mới.
    - PUT /products/{id} - Cập nhật thông tin sản phẩm.
    - DELETE /users/{id} - Xóa tài khoản người dùng.
* **Giải pháp nâng cao:**  
  Đối với các giao dịch yêu cầu xử lý theo thời gian thực (real-time), có thể sử dụng WebSocket hoặc gRPC để cải thiện hiệu suất và giảm độ trễ.

**3. Cơ sở dữ liệu**

* **Công cụ:**
  + **PostgreSQL:** Được lựa chọn vì khả năng xử lý mạnh mẽ, hỗ trợ các chức năng nâng cao như indexing, transaction, và bảo mật tốt.
  + **Redis (Caching):** Dùng để lưu trữ dữ liệu tạm thời, tăng tốc các truy vấn thường xuyên (ví dụ: danh sách sản phẩm phổ biến).
* **Thiết kế cơ sở dữ liệu:**
  + Xây dựng cơ sở dữ liệu quan hệ (RDBMS) với các bảng chính như:
    - Bảng **Products (Sản phẩm):** Chứa thông tin sản phẩm.
    - Bảng **Orders (Đơn hàng):** Quản lý thông tin đơn hàng.
    - Bảng **Users (Người dùng):** Lưu thông tin tài khoản quản trị và nhân viên.
    - Bảng **Inventory (Tồn kho):** Theo dõi số lượng sản phẩm tại các cơ sở.
* **Backup dữ liệu:**
  + Sử dụng các công cụ như pg\_dump hoặc cơ chế tự động của PostgreSQL để sao lưu định kỳ dữ liệu.
  + Lưu trữ bản sao lưu trên dịch vụ cloud (AWS S3, Google Cloud Storage) để đảm bảo an toàn.

**4. Công nghệ giao diện người dùng**

* **Frontend Frameworks:**
  + Sử dụng **ReactJS** hoặc **VueJS** để xây dựng giao diện người dùng hiện đại, dễ sử dụng và tương tác tốt.
  + Sử dụng **Bootstrap** hoặc **Tailwind CSS** để tăng tốc quá trình thiết kế giao diện và đảm bảo tính tương thích trên nhiều thiết bị (responsive).
* **Tính năng tối ưu giao diện:**
  + Tích hợp các biểu đồ và báo cáo động (sử dụng thư viện như Chart.js hoặc Highcharts) để hiển thị doanh thu và phân tích dữ liệu.

**5. Hiệu suất và khả năng mở rộng**

* **Load Balancer:**  
  Sử dụng công cụ như Nginx hoặc HAProxy để phân phối tải đều cho các server backend.
* **CDN (Content Delivery Network):**  
  Tích hợp CDN (như Cloudflare) để tăng tốc tải tài nguyên tĩnh như hình ảnh, CSS, và JavaScript.
* **Scaling:**
  + **Horizontal Scaling:** Thêm nhiều server khi số lượng giao dịch tăng cao.
  + **Vertical Scaling:** Nâng cấp cấu hình server nếu cần xử lý nhiều giao dịch phức tạp.

**6. Bảo mật**

* **Mã hóa dữ liệu:**
  + Sử dụng HTTPS để mã hóa giao tiếp giữa client và server, đảm bảo dữ liệu người dùng không bị đánh cắp.
  + Mã hóa dữ liệu nhạy cảm (như mật khẩu) trước khi lưu vào cơ sở dữ liệu (sử dụng thuật toán bcrypt).
* **Phân quyền:**
  + Xây dựng hệ thống phân quyền chặt chẽ giữa các nhóm người dùng (Super Admin, Admin, Nhân viên).
  + Sử dụng JWT (JSON Web Token) để xác thực người dùng khi truy cập API.
* **Tường lửa ứng dụng (WAF):**  
  Tích hợp các công cụ như ModSecurity để phát hiện và ngăn chặn các cuộc tấn công phổ biến như SQL Injection, XSS.

**7. Kiểm thử và triển khai**

* **Kiểm thử:**
  + Thực hiện kiểm thử tự động (Automated Testing) với các công cụ như Selenium hoặc Jest để đảm bảo hệ thống hoạt động ổn định.
  + Kiểm thử tải (Load Testing) để đánh giá hiệu suất khi có nhiều người dùng truy cập cùng lúc.
* **Triển khai:**
  + Sử dụng Docker để đóng gói và triển khai ứng dụng một cách nhanh chóng và nhất quán trên nhiều môi trường (dev, test, prod).
  + Sử dụng các công cụ CI/CD (Jenkins, GitHub Actions) để tự động hóa quá trình build và triển khai.