ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

Logo

Description automatically generated

**BÁO CÁO CUỐI KỲ**

**Đề tài: Quản lý nhà sách**

**Môn học**: Các Công Nghệ Lập Trình Hiện Đại

**Lớp**: DCT121C3

**Giảng viên**: Đỗ Như Tài

**Các thành viên trong nhóm**:

3121411071 – Nguyễn Quang Hiếu

3121411106 – Đoàn Minh Khôi

3121411149 – Trần Minh Chiến

**Năm học**: 2024-2025 / HKII

**BẢNG PHÂN CÔNG**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Nhiệm vụ** | **Phần trăm hoàn thành** |
| **Nguyễn Quang Hiếu** |  |  |
| **Đoàn Minh Khôi** |  |  |
| **Trần Minh Chiến** |  |  |

Table of Contents

[Chương 1: Tổng quan đề tài 5](#_Toc198496271)

[**1.1.** **Mô tả dự án** 5](#_Toc198496272)

[**1.2.** **Khảo sát hiện trạng** 6](#_Toc198496273)

[1.2.1. Hiện trạng tổ chức: 6](#_Toc198496274)

[Bảng câu hỏi phỏng vấn 6](#_Toc198496275)

[1.2.2. Hiện trạng nghiệp vụ (chức năng hệ thống) 8](#_Toc198496276)

[**1.3.** **Quy trình nghiệp vụ:** 8](#_Toc198496277)

[1.3.1. Quản lý nhập hàng: 8](#_Toc198496278)

[1.3.2. Quy trình quản lý kho: 9](#_Toc198496279)

[1.3.3. Quy trình bán hàng: 10](#_Toc198496280)

[1.3.4. Quy trình quản lý khách hàng: 11](#_Toc198496281)

[**1.4.** **Yêu cầu hệ thống** 12](#_Toc198496282)

[1.4.1. Yêu cầu chức năng: 12](#_Toc198496283)

[1.4.2. Yêu cầu phi chức năng: 12](#_Toc198496284)

[**1.5.** **Đề xuất giải pháp:** 13](#_Toc198496285)

[Chương 2: Tìm hiểu công nghệ. 15](#_Toc198496286)

[1. Quy trình phát triển phần mềm: 15](#_Toc198496287)

[2. Kế hoạch phát triển hệ thống: 16](#_Toc198496288)

[3. Cấu hình môi trường 18](#_Toc198496289)

[4. Chi tiết 19](#_Toc198496290)

[1. Frontend 19](#_Toc198496291)

[2. Backend 22](#_Toc198496292)

[3. Quản Lý Mã Nguồn 26](#_Toc198496293)

[4. Docker 27](#_Toc198496294)

[Chương 3: Phân tích hệ thống 28](#_Toc198496295)

[**2.1.** **Ngữ cảnh nghiệp vụ và quy trình nghiệp vụ** 28](#_Toc198496298)

[2.1.1. Ngữ cảnh nghiệp vụ: 28](#_Toc198496299)

[2.1.2. Quy trình nghiệp vụ: 30](#_Toc198496300)

[**2.2.** **Mô hình ngữ cảnh và mô hình khái niệm** 32](#_Toc198496301)

[2.2.1. Mô hình ngữ cảnh (Context Diagram) 32](#_Toc198496302)

[2.2.2. Mô hình khái niệm 32](#_Toc198496303)

[**2.3.** **Mô hình Use case** 33](#_Toc198496304)

[2.3.1. Use case Khách hàng 33](#_Toc198496305)

[2.3.2. Use case Nhân viên/ Quản trị viên 34](#_Toc198496306)

[**2.4.** **User Story** 34](#_Toc198496307)

[2.4.1. Khách hàng: 34](#_Toc198496308)

[2.4.2. Quản lý: 34](#_Toc198496309)

[**2.5.** **Thiết kế giao diện ở mức ý niệm** 35](#_Toc198496310)

[**2.6.** **Thiết kế dữ liệu ở mức ý niệm** 35](#_Toc198496311)

[**2.7.** **Đặc tả Use case và mô hình** 35](#_Toc198496312)

[CHƯƠNG 4: THIẾT KẾ HỆ THỐNG 39](#_Toc198496313)

[4.1.1 Sơ đồ khối ý niệm của hệ thống 39](#_Toc198496314)

[4.1.2 Lý do chọn lựa công nghệ 40](#_Toc198496315)

[4.2 Khung nhìn mức thành phần (Decomposition View – C4 Model) 41](#_Toc198496316)

[4.2.1 System Context Diagram 41](#_Toc198496317)

[4.2.2 Container Diagram 42](#_Toc198496318)

[4.2.3 Component Diagram 43](#_Toc198496319)

[4.3 Khung nhìn mức dữ liệu (Data/Transaction View) 43](#_Toc198496320)

[4.3.1 Mô hình dữ liệu (ERD) 43](#_Toc198496321)

[4.3.2 Luồng giao dịch đặt hàng 44](#_Toc198496322)

[4.4 Triển khai hệ thống và CI/CD 44](#_Toc198496323)

[4.4.1 Triển khai bằng Docker 44](#_Toc198496324)

[4.4.2 Quy trình CI/CD với GitHub Actions 45](#_Toc198496325)

[4.5 Giám sát hệ thống (Resilience View) 45](#_Toc198496326)

# 

# Chương 1: Tổng quan đề tài

* 1. **Mô tả dự án**

Trong thời đại công nghệ số phát triển mạnh mẽ hiện nay, việc quản lý thông tin bằng phương pháp thủ công đã bộc lộ nhiều hạn chế như dễ sai sót, mất thời gian và khó mở rộng khi quy mô tăng lên. Đối với các thư viện, nhà sách hoặc hệ thống cho thuê sách, việc quản lý số lượng lớn đầu sách, thông tin người dùng, tình trạng mượn trả và các giao dịch liên quan là một bài toán phức tạp nếu không có sự hỗ trợ từ các hệ thống tự động hóa. Chính vì vậy, nhu cầu xây dựng một hệ thống quản lý sách hiệu quả, tiện lợi và dễ sử dụng là vô cùng cần thiết.

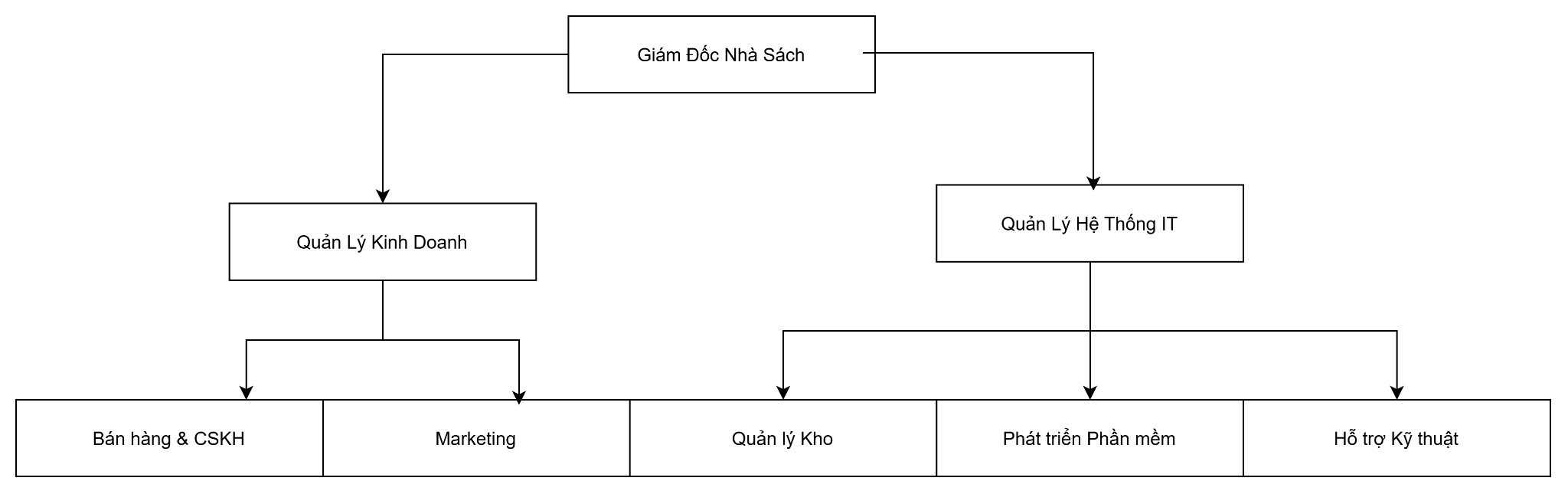
Hệ thống này không chỉ giúp tối ưu hóa quy trình quản lý sách mà còn hỗ trợ tra cứu nhanh chóng, giảm thiểu thời gian xử lý nghiệp vụ và nâng cao trải nghiệm của người dùng. Việc áp dụng công nghệ thông tin trong quản lý sách góp phần hiện đại hóa mô hình quản lý truyền thống, đồng thời đảm bảo tính chính xác và hiệu quả trong công việc.

* 1. **Khảo sát hiện trạng** 
     1. Hiện trạng tổ chức:

Bảng câu hỏi phỏng vấn

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Nội dung phỏng vấn** | **Người trả lời** | **Nội dung trả lời** |
| 1 | Hỏi, tìm hiểu về hiện trạng tổng quan về hệ thống quản lý sách | |  | | --- | | Nguyễn Quang Hiếu |  |  | | --- | |  | | Hiện tại, chúng tôi có khoảng hơn 5000 đầu sách trong hệ thống, được phân loại theo thể loại, tác giả và nhà xuất bản. Hệ thống quản lý vẫn còn thủ công một phần, cần xây dựng hệ thống CSDL tập trung. |
| 2 | Quy trình nhập, xuất sách | |  | | --- | | Nguyễn Quang Hiếu |  |  | | --- | |  | | Sách được nhập vào kho từ các nhà xuất bản và nhà phân phối. Chúng tôi ghi nhận thông tin sách vào hệ thống một cách thủ công, cần tự động hóa việc cập nhật thông tin và kiểm kê. |
| 3 | Công nghệ và phần mềm đang sử dụng | Đoàn Minh Khôi | Chúng tôi đang sử dụng một phần mềm quản lý kho hàng cũ, chạy trên nền tảng Windows. Hệ thống cần được nâng cấp lên công nghệ web-based và cloud-based. |
| 4 | Hệ thống bán hàng và bảo mật | Đoàn Minh Khôi | Mỗi giao dịch được ghi nhận trên hệ thống POS riêng biệt, chưa đồng bộ với hệ thống quản lý kho. Bảo mật dữ liệu khách hàng và giao dịch cần được cải thiện. |
| 5 | Quản lý dữ liệu khách hàng và marketing | Nguyễn Ngọc Hạnh Nguyên | Dữ liệu khách hàng được lưu trữ phân tán, cần xây dựng hệ thống CRM tích hợp và công cụ phân tích dữ liệu để tối ưu chiến dịch marketing. |

**Sơ đồ tổ chức nội bộ:**

**Mô tả sơ đồ:**

* Giám đốc nhà sách: Người điều hành toàn bộ hoạt động.
* Quản lý kinh doanh: Phụ trách các hoạt động bán hàng, CSKH, marketing.
* Bộ phận Bán hàng & CSKH: Tiếp xúc trực tiếp với khách hàng, giải quyết khiếu nại.
* Bộ phận Marketing: Quảng bá thương hiệu, thực hiện chương trình khuyến mãi.
  + Quản lý hệ thống IT: Giám sát công nghệ, đảm bảo hạ tầng kỹ thuật vận hành ổn định.
* Bộ phận Phát triển phần mềm: Quản lý hệ thống POS, kho, CRM, kế toán.
* Bộ phận Hỗ trợ kỹ thuật: Bảo trì máy tính, thiết bị POS, bảo mật dữ liệu.
* Bộ phận Quản lý kho: Theo dõi nhập hàng, kiểm kê sách, báo cáo số liệu tồn kho.
  + 1. Hiện trạng nghiệp vụ (chức năng hệ thống)

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Nghiệp vụ** | **Người thực hiện** | **Ghi chú** |
| 1 | Quản lý sách | Bộ phận Quản lý kho | Quản lý danh mục sách, phân loại, cập nhật thông tin sách mới, theo dõi sách bán chạy. Cần tin học hóa quy trình phân loại và cập nhật. |
| 2 | Quản lý kho | Bộ phận Quản lý kho | Nhập xuất sách, kiểm kê hàng tồn, sắp xếp kho, báo cáo sách tồn kho. Cần xây dựng hệ thống quản lý kho thông minh với công nghệ RFID. |
| 3 | Bán hàng và chăm sóc khách hàng | Bộ phận Kinh doanh & CSKH | Xử lý giao dịch, quản lý thông tin khách hàng, giải quyết khiếu nại. Cần phát triển hệ thống CRM tích hợp. |
| 4 | Marketing và truyền thông | Bộ phận Marketing | Quảng bá sách mới, tổ chức sự kiện, thiết kế chương trình khuyến mãi. Cần xây dựng hệ thống phân tích dữ liệu lớn. |
| 5 | Phát triển và bảo trì hệ thống CNTT | Bộ phận IT | Quản lý cơ sở hạ tầng IT, phát triển và bảo trì phần mềm, đảm bảo an ninh dữ liệu. Cần áp dụng DevOps và CI/CD. |

* 1. **Quy trình nghiệp vụ:**
     1. Quản lý nhập hàng:

Mục tiêu: Đảm bảo cung ứng sách kịp thời, chất lượng và đúng nhu cầu khách hàng.

* + - * Xác định nhu cầu nhập hàng
  + Phân tích doanh số bán hàng trong quá khứ để dự đoán nhu cầu nhập sách.
  + Nghiên cứu thị trường, xu hướng đọc sách hiện tại để cập nhật sách mới, sách bán chạy.
  + Thu thập phản hồi từ khách hàng về các thể loại sách hoặc tựa sách được quan tâm.
    - * Lựa chọn nhà cung cấp
  + Tìm kiếm và đánh giá các nhà xuất bản, nhà phân phối sách có uy tín trên thị trường.
  + So sánh các yếu tố: giá cả, chính sách chiết khấu, thời gian giao hàng và điều kiện thanh toán.
  + Xây dựng mối quan hệ bền vững với các nhà cung cấp để đảm bảo quá trình nhập hàng ổn định.
    - * Đặt hàng và nhận hàng
  + Nhân viên lập đơn đặt hàng, bao gồm các thông tin: tên sách, số lượng, giá dự kiến, ngày giao hàng.
  + Theo dõi tiến trình thực hiện đơn hàng và cập nhật trạng thái.
  + Khi hàng được giao đến, tiến hành kiểm tra số lượng và chất lượng thực tế so với đơn hàng.
  + Cập nhật thông tin sách nhập vào hệ thống quản lý kho.
    - * Quản lý nhà cung cấp
  + Đánh giá định kỳ hiệu quả và độ tin cậy của từng nhà cung cấp dựa trên các tiêu chí: đúng hẹn, chất lượng sách, hỗ trợ sau bán.
  + Duy trì liên hệ thường xuyên để cập nhật sản phẩm mới, chương trình ưu đãi.
  + Xử lý các phát sinh trong quá trình nhập hàng: sai sót, chậm giao, chất lượng không đảm bảo.
    1. Quy trình quản lý kho:

Mục tiêu: Đảm bảo kho sách được tổ chức khoa học, dữ liệu chính xác, hỗ trợ tốt cho việc bán hàng và nhập hàng.

* + - * Sắp xếp hàng hóa trong kho
  + Sắp xếp sách theo thể loại, tên tác giả, nhà xuất bản hoặc mã phân loại.
  + Đảm bảo không gian kho được tổ chức hợp lý, dễ tìm kiếm, tránh nhầm lẫn.
  + Áp dụng mã vạch hoặc mã QR để hỗ trợ nhận diện và truy xuất nhanh thông tin sách.
    - * Kiểm kê kho định kỳ
  + Thực hiện kiểm kê theo chu kỳ (tháng/quý) để đối chiếu số liệu thực tế và hệ thống.
  + Xử lý các trường hợp lệch tồn kho do mất mát, hư hỏng hoặc lỗi hệ thống.
  + Cập nhật lại số liệu tồn kho trên phần mềm quản lý để đảm bảo độ chính xác.
    - * Quản lý hàng tồn kho
  + Theo dõi lượng sách tồn để kịp thời bổ sung hoặc xử lý sách tồn lâu.
  + Thiết lập ngưỡng tồn kho tối thiểu và tối đa cho từng đầu sách để tránh thiếu hụt hoặc tồn dư.
  + Triển khai các chương trình khuyến mãi, giảm giá để đẩy mạnh tiêu thụ sách tồn kho.
    1. Quy trình bán hàng:

Mục tiêu: Tối ưu trải nghiệm khách hàng, tăng doanh số và vận hành trơn tru quy trình giao dịch.

* + - * Tư vấn và hỗ trợ khách hàng
  + Nhân viên bán hàng có kiến thức về các thể loại và nội dung sách.
  + Chủ động tư vấn sách phù hợp với nhu cầu, độ tuổi, sở thích khách hàng.
  + Hỗ trợ tìm kiếm và cung cấp thông tin nhanh chóng, chính xác.
    - * Xử lý thanh toán
  + Hệ thống hỗ trợ nhiều phương thức thanh toán: tiền mặt, thẻ ngân hàng, ví điện tử.
  + Đảm bảo quy trình thanh toán diễn ra thuận tiện, bảo mật và nhanh chóng.
  + Xuất hóa đơn đầy đủ cho khách hàng sau khi hoàn tất giao dịch.
    - * Đóng gói và giao hàng
  + Kiểm tra kỹ sản phẩm trước khi đóng gói.
  + Sử dụng bao bì phù hợp để đảm bảo sách không bị hư hỏng trong quá trình vận chuyển.
  + Theo dõi tiến trình giao hàng và cập nhật thông tin đến khách hàng.
    1. Quy trình quản lý khách hàng:

Mục tiêu: Tăng mức độ hài lòng và xây dựng mối quan hệ lâu dài với khách hàng.

* + - * Xây dựng cơ sở dữ liệu khách hàng
  + Thu thập thông tin cơ bản: họ tên, địa chỉ, số điện thoại, email.
  + Lưu lại lịch sử mua hàng, sản phẩm yêu thích và mức độ tương tác của khách hàng.
    - * Chăm sóc khách hàng
  + Gửi thông báo về sách mới, ưu đãi qua email hoặc tin nhắn định kỳ.
  + Xử lý nhanh chóng các khiếu nại, thắc mắc từ khách hàng với tinh thần cầu thị và chuyên nghiệp.
  + Thiết lập và triển khai chương trình khách hàng thân thiết để tăng độ gắn bó.
    - * Thu thập phản hồi khách hàng
  + Triển khai các khảo sát online hoặc trực tiếp để ghi nhận ý kiến về chất lượng dịch vụ và sản phẩm.
  + Phân tích phản hồi để điều chỉnh chiến lược kinh doanh, nâng cao trải nghiệm khách hàng.
  1. **Yêu cầu hệ thống**
     1. Yêu cầu chức năng:
* Quản lý danh mục và phân loại sách:
  + Thêm, sửa, xóa danh mục sách (thể loại, chủ đề).
  + Gán sách vào danh mục tương ứng để dễ tìm kiếm và lọc.
  + Tìm kiếm sách theo danh mục, tên sách, tác giả, nhà xuất bản, năm xuất bản.
* Cập nhật và quản lý thông tin sách:
  + Thêm sách mới: nhập thông tin chi tiết (tên, tác giả, giá, ảnh bìa, mô tả...).
  + Cập nhật thông tin sách (giá, mô tả, số lượng tồn...).
  + Xóa sách không còn kinh doanh khỏi hệ thống.
* Quản lý kho sách:
  + Ghi nhận nhập kho: thêm mới sách, số lượng, nhà cung cấp, ngày nhập.
  + Ghi nhận xuất kho khi có giao dịch bán hàng.
  + Theo dõi và cập nhật số lượng sách tồn kho theo thời gian thực.
  + Thực hiện kiểm kê kho định kỳ.
* Xử lý bán hàng:
  + Tạo đơn hàng mới với thông tin sách, khách hàng và tổng giá trị đơn hàng.
  + Cập nhật trạng thái đơn hàng (đang xử lý, đã giao, đã hủy).
  + Tích hợp thanh toán (tiền mặt/thẻ/ví điện tử).
  + In hoặc gửi hóa đơn điện tử cho khách hàng.
* Quản lý thông tin khách hàng:
  + Lưu trữ thông tin khách hàng: họ tên, địa chỉ, email, số điện thoại.
  + Theo dõi lịch sử mua hàng của từng khách.
  + Phân loại khách hàng (mới, thân thiết, VIP...).
* Báo cáo và thống kê:
  + Báo cáo doanh thu theo ngày, tháng, năm.
  + Thống kê sách bán chạy, sách tồn kho, đơn hàng theo trạng thái.
  + Báo cáo hiệu quả nhà cung cấp.
    1. Yêu cầu phi chức năng:

Hệ thống không chỉ cần đáp ứng các chức năng mà còn phải đảm bảo:

* Hiệu năng
  + Hệ thống phải xử lý tối thiểu 100 yêu cầu/phút với độ trễ nhỏ hơn 2 giây.
  + Tìm kiếm sách trả kết quả trong vòng 1 giây.
  + Tính sẵn sàng
  + Hệ thống cần hoạt động 24/7 với thời gian gián đoạn không quá 1% mỗi tháng.
* Khả năng mở rộng
  + Có thể dễ dàng mở rộng thêm danh mục, loại sách, người dùng hoặc tích hợp với các hệ thống khác (ví dụ: vận chuyển, thanh toán).
* Bảo mật
  + Phân quyền người dùng rõ ràng: quản trị viên, nhân viên, khách hàng.
  + Dữ liệu người dùng được mã hóa khi lưu trữ và truyền tải.
  + Kiểm soát đăng nhập bằng mật khẩu mạnh, xác thực OTP khi cần.
* Dễ sử dụng
  + Giao diện người dùng thân thiện, hỗ trợ tiếng Việt.
  + Thiết kế hỗ trợ thao tác nhanh, phù hợp cho nhân viên kho, nhân viên bán hàng và khách hàng online.
* Khả năng sao lưu và phục hồi
  + Dữ liệu được sao lưu tự động hàng ngày.
  + Có chức năng phục hồi hệ thống trong trường hợp mất mát dữ liệu.
  1. **Đề xuất giải pháp:**

Để giải quyết những hạn chế trong quy trình quản lý và kinh doanh sách hiện nay (thủ công, phân tán dữ liệu, khó kiểm soát hàng tồn, thiếu thống kê kịp thời...), nhóm đề xuất xây dựng một hệ thống phần mềm quản lý bán sách tích hợp, hoạt động trên nền web, với các đặc điểm và lợi ích sau:

* Hệ thống tích hợp các chức năng cốt lõi:
  + Hệ thống sẽ tích hợp đầy đủ các nghiệp vụ chính trong quản lý nhà sách bao gồm: Quản lý sách và kho hàng; quản lý bán hàng, hóa đơn và giao dịch; quản lý khách hàng, thẻ thành viên; quản lý khuyến mãi; thống kê và báo cáo; phân quyền và quản lý nhân viên.
  + Điều này giúp đồng bộ dữ liệu giữa các bộ phận, giảm thiểu sai sót, nâng cao hiệu quả quản lý.
* Công nghệ đề xuất
  + Giao diện người dùng (Frontend): Sử dụng ReactJS giúp phát triển giao diện linh hoạt, thân thiện với người dùng, dễ bảo trì và mở rộng.
  + Máy chủ xử lý (Backend): Sử dụng Node.js (Express) để xây dựng API, cho khả năng xử lý nhanh, phù hợp với các ứng dụng thời gian thực.
  + Cơ sở dữ liệu: Sử dụng PostgreSQL hoặc MongoDB tùy thuộc vào yêu cầu dữ liệu (có thể kết hợp cả hai để xử lý dữ liệu có cấu trúc và phi cấu trúc).
  + Quản lý trạng thái kho và đơn hàng theo thời gian thực: Sử dụng WebSocket hoặc polling để cập nhật số liệu tức thì.
  + Bảo mật hệ thống: Mã hóa mật khẩu người dùng, xác thực JWT, phân quyền người dùng rõ ràng.
* Hình thức triển khai
  + Hệ thống hoạt động trên nền tảng web, người dùng có thể truy cập qua trình duyệt mà không cần cài đặt phần mềm.
  + Triển khai trên nền tảng cloud (như AWS, Heroku hoặc Docker-based VPS) để đảm bảo khả năng mở rộng và tính sẵn sàng cao.
* Lợi ích kỳ vọng
  + Tăng hiệu quả quản lý: Giảm thời gian xử lý đơn hàng, kiểm kê kho, báo cáo.
  + Nâng cao trải nghiệm khách hàng: Cung cấp thông tin rõ ràng, hỗ trợ chương trình thành viên và khuyến mãi hấp dẫn.
  + Dễ dàng mở rộng và tích hợp: Có thể mở rộng thêm chi nhánh, kênh bán hàng (online/offline) hoặc tích hợp API với đơn vị vận chuyển, thanh toán...
  + Hỗ trợ ra quyết định: Dữ liệu thống kê chi tiết giúp nhà quản lý dễ dàng phân tích và đưa ra các chiến lược kinh doanh hiệu quả.

# Chương 2: Tìm hiểu công nghệ.

1. Quy trình phát triển phần mềm:

Hệ thống được phát triển theo mô hình Agile – Scrum, với mục tiêu đảm bảo linh hoạt trong thay đổi yêu cầu, phản hồi nhanh từ người dùng và cải tiến liên tục. Mỗi vòng lặp phát triển (Sprint) kéo dài từ 1 đến 2 tuần và bao gồm đầy đủ các giai đoạn: phân tích, thiết kế, lập trình, kiểm thử và triển khai.

Các bước trong quy trình phát triển:

* Thu thập và phân tích yêu cầu:
  + Phỏng vấn người dùng, khảo sát thực tế để xác định rõ nghiệp vụ.
  + Ghi nhận yêu cầu chức năng và phi chức năng.
  + Lập danh sách backlog sản phẩm (Product Backlog).
* Thiết kế hệ thống:
  + Thiết kế kiến trúc hệ thống theo mô hình client-server.
  + Xây dựng các sơ đồ ngữ cảnh, use-case, ERD, luồng xử lý và UI mockup.
  + Thiết kế API kết nối giữa frontend (ReactJS) và backend (Spring Boot).
* Phát triển phần mềm theo Sprint:
  + Mỗi Sprint tập trung phát triển một nhóm chức năng cụ thể.
  + Sử dụng Git để quản lý phiên bản mã nguồn, nhánh phát triển riêng cho từng tính năng.
  + Lập trình frontend bằng ReactJS (HTML/CSS/JS) trong Visual Studio Code.
  + Xây dựng backend với Spring Boot (Java), kết nối cơ sở dữ liệu PostgreSQL.
* Kiểm thử và hiệu chỉnh:
  + Kiểm thử đơn vị (unit test) và kiểm thử tích hợp (integration test).
  + Sửa lỗi, hiệu chỉnh theo phản hồi người dùng hoặc nhóm test.
  + Kiểm thử API bằng Postman hoặc Swagger UI.
* Triển khai và bàn giao:
  + Triển khai hệ thống lên môi trường thật (có thể sử dụng Docker, cloud server, hoặc Heroku).
  + Đào tạo người dùng sử dụng hệ thống, bàn giao tài liệu hướng dẫn và mã nguồn.

1. Kế hoạch phát triển hệ thống:

Bảng tổ chức công việc chi tiết

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Timeframe** | **Estimate** |
| 1. Project Kickoff & Planning | Khởi Động và Lập Kế Hoạch Dự Án | 1 week: 06/03 - 13/03 |  |
| 1.1 | Họp khởi động dự án | 06/03 - 13/03 | 4h |
| 1.2 | Thu thập yêu cầu nhanh | 06/03 - 13/03 | 6h |
| 1.3 | Phỏng vấn các bên liên quan chính | 06/03 - 13/03 | 8h |
| 1.4 | Lập tài liệu yêu cầu ưu tiên | 06/03 - 13/03 | 6h |
| 1.5 | Hoàn thiện kế hoạch dự án | 06/03 - 13/03 | 6h |
| 2. Design & Setup | Thiết Kế và Cài Đặt | 1 week: 13/03 - 20/03 |  |
| 2.1 | Thiết kế kiến trúc kỹ thuật đơn giản hóa | 13/03 - 20/03 | 8h |
| 2.2 | Thiết kế lược đồ cơ sở dữ liệu | 13/03 - 20/03 | 8h |
| 2.3 | Thiết kế UI/UX quan trọng và wireframe | 13/03 - 20/03 | 8h |
| 2.4 | Thiết lập môi trường phát triển | 13/03 - 20/03 | 4h |
| 2.5 | Thiết lập công cụ quản lý dự án | 13/03 - 20/03 | 4h |
| 3. Core Development | Phát triển cốt lõi | 5 weeks: 20/03 - 24/04 |  |
| 3.1. Foundation | Nền Tảng | 1 week: 20/03 - 27/03 |  |
| 3.1.1 | Triển khai cơ sở dữ liệu | 20/03 - 27/03 | 8h |
| 3.1.2 | Hệ thống xác thực | 20/03 - 27/03 | 8h |
| 3.1.3 | Cấu trúc API cốt lõi | 20/03 - 27/03 | 8h |
| 3.1.4 | Thiết lập ứng dụng cơ bản | 20/03 - 27/03 | 8h |
| 3.2. Inventory Management | Quản Lý kho hàng | 1.5 weeks: 27/03 - 05/04 |  |
| 3.2.1 | Quản lý danh mục sách và metadata | 27/03 - 05/04 | 12h |
| 3.2.2 | Chức năng theo dõi tồn kho | 27/03 - 05/04 | 8h |
| 3.2.3 | Quản lý danh mục cơ bản | 27/03 - 05/04 | 6h |
| 3.2.3 | Quản lý danh mục cơ bản | 27/03 - 05/04 | 6h |
| 3.2.4 | Tích hợp mã vạch đơn giản | 27/03 - 05/04 | 6h |
| 3.3. Sales Management | Quản Lý Bán Hàng | 1.5 weeks: 05/04 - 15/04 |  |
| 3.3.1 | Giao diện điểm bán hàng | 05/04 - 15/04 | 12h |
| 3.3.2 | Xử lý thanh toán cơ bản | 05/04 - 15/04 | 8h |
| 3.3.3 | Tạo hóa đơn | 05/04 - 15/04 | 6h |
| 3.3.4 | Hoạt động bán hàng thiết yếu | 05/04 - 15/04 | 6h |
| 3.4. Customer Management | Quản Lý Khách Hàng | 1 week: 15/04 - 24/04 |  |
| 3.4.1 | Cơ sở dữ liệu khách hàng | 15/04 - 24/04 | 8h |
| 3.4.2 | Theo dõi lịch sử mua hàng | 15/04 - 24/04 | 8h |
| 3.4.3 | Tương tác khách hàng cơ bản | 15/04 - 24/04 | 8h |
| 4. Essential Features | Tính năng thiết yếu | 1.5 weeks: 24/04 - 03/05 |  |
| 4.1 | Bảng điều khiển báo cáo cơ bản | 24/04 - 03/05 | 10h |
| 4.2 | Phân tích kinh doanh quan trọng | 24/04 - 03/05 | 6h |
| 4.3 | Quản lý người dùng và vai trò | 24/04 - 03/05 | 8h |
| 4.4 | Điểm tích hợp hệ thống | 24/04 - 03/05 | 8h |
| 5. Testing & QA | Kiểm thử và đảm bảo chất lượng | 1 week: 03/05 - 10/05 |  |
| 5.1 | Kiểm thử chức năng của tất cả các module | 03/05 - 10/05 | 12h |
| 5.2 | Kiểm thử tích hợp | 03/05 - 10/05 | 8h |
| 5.3 | Kiểm thử chấp nhận người dùng | 03/05 - 10/05 | 6h |
| 5.4 | Sửa lỗi và hoàn thiện | 03/05 - 10/05 | 12h |
| 5.5 | Tối ưu hóa hiệu suất | 03/05 - 10/05 | 4h |
| 6. Deployment & Launch | Triển Khai và Ra Mắt | 0.5 week: 10/05 - 15/05 |  |
| 6.1 | Thiết lập môi trường sản xuất | 10/05 - 15/05 | 4h |
| 6.2 | Di chuyển dữ liệu | 10/05 - 15/05 | 4h |
| 6.3 | Đào tạo nhân viên | 10/05 - 15/05 | 6h |
| 6.4 | Hoạt động go-live | 10/05 - 15/05 | 4h |
| 6.5 | Hỗ trợ sau triển khai | 10/05 - 15/05 | 4h |
|  | Total |  | 400h |

1. Cấu hình môi trường

* Hệ điều hành: Windows 11
* IDE: Visual Studio Code
* Frontend:
  + Ngôn ngữ: JavaScript, JSX
  + Framework: ReactJS
  + Quản lý gói: npm/yarn
  + Công cụ hỗ trợ:
    - React Router (điều hướng)
    - Axios (gọi API)
    - TailwindCSS / Bootstrap (thiết kế UI)
* Backend:
  + Ngôn ngữ: Java
  + Framework: Spring Boot
  + Các thành phần tích hợp:
  + Spring Web (REST API)
  + Spring Data JPA (ORM)
  + Spring Security (phân quyền người dùng)
  + Build tool: Maven hoặc Gradle
* Quản lý mã nguồn – Git:
  + Công cụ: Git CLI, GitHub Desktop hoặc tích hợp sẵn trong VS Code
  + Nền tảng: GitHub / GitLab / Bitbucket
  + Mô hình nhánh: Git Flow (main / develop / feature)
  + Công cụ hỗ trợ: Docker : đóng gói và triển khai hệ thống

## Chi tiết

### Frontend

#### 1.1. Ngôn Ngữ và Framework

- JavaScript/JSX

* ES6+ features
* Arrow functions
* Destructuring
* Async/await
* Template literals
* Modules

- ReactJS

* Functional Components
* Hooks (useState, useEffect, useContext)
* Custom Hooks
* Context API
* React.memo và useMemo
* Error Boundaries

#### 1.2. Quản Lý Gói

- npm/yarn

* Package.json
* Dependencies
* DevDependencies
* Scripts
* Lock files

#### 1.3. Công Cụ Hỗ Trợ

React Router

- Route Configuration

<Routes>

<Route path="/" element={<Home />} />

<Route path="/books" element={<BookList />} />

<Route path="/books/:id" element={<BookDetail />} />

<Route path="/cart" element={<Cart />} />

<Route path="/checkout" element={<Checkout />} />

</Routes>

```

* Navigation

useNavigate hook

Link component

Programmatic navigation

* Route Guards
* Protected routes
* Authentication checks
* Nested Routes
* Layout components
* Outlet component

Axios

- API Configuration

const api = axios.create({

baseURL: 'http://localhost:8080/api',

timeout: 5000,

headers: {

'Content-Type': 'application/json'

}

});

* Interceptors
* Request interceptor
* Response interceptor
* Error handling
* API Calls
* GET requests
* POST requests
* PUT requests
* DELETE requests
* Error Handling
* Global error handling
* Custom error messages
* Retry mechanism

TailwindCSS

- Configuration

module.exports = {

content: ['./src/\*\*/\*.{js,jsx,ts,tsx}'],

theme: {

extend: {

colors: {

primary: '#1a365d',

secondary: '#2d3748'

}

}

},

plugins: []

}

* Utility Classes
* Flexbox
* Grid
* Spacing
* Typography
* Colors
* Custom Components
* Buttons
* Cards
* Forms
* Navigation
* Responsive Design
* Breakpoints
* Mobile-first approach
* Custom media queries

### Backend

#### 2.1. Ngôn Ngữ và Framework

* Java
* Java 17
* Lambda expressions
* Stream API
* Optional class
* Records
* Pattern matching
* Spring Boot
* Auto-configuration
* Starter dependencies
* Embedded servers
* Actuator
* DevTools

#### 2.2. Các Thành Phần Tích Hợp

## Spring Web

* REST Controllers

@RestController

@RequestMapping("/api/books")

public class BookController {

@GetMapping

public List<Book> getAllBooks() { ... }

@GetMapping("/{id}")

public Book getBook(@PathVariable Long id) { ... }

@PostMapping

public Book createBook(@RequestBody Book book) { ... }

}

* Request Mapping
* Path variables
* Request parameters
* Request body
* Response entities
* Exception Handling
* Global exception handler
* Custom exceptions
* Error responses

## Spring Data JPA

* Entity Classes

@Entity

@Table(name = "books")

public class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@Column(nullable = false)

private String title;

@ManyToOne

@JoinColumn(name = "author\_id")

private Author author;

}

* Repositories
* CRUD operations
* Custom queries
* Query methods
* Pagination
* Relationships
* One-to-One
* One-to-Many
* Many-to-Many
* Transactions
* Transaction management
* Isolation levels
* Propagation behaviors

## Spring Security

* Security Configuration

@Configuration

@EnableWebSecurity

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) {

http

.authorizeRequests()

.antMatchers("/api/public/\*\*").permitAll()

.antMatchers("/api/admin/\*\*").hasRole("ADMIN")

.anyRequest().authenticated()

.and()

.csrf().disable()

.sessionManagement()

.sessionCreationPolicy(SessionCreationPolicy.STATELESS);

return http.build();

}

}

* Authentication
* JWT authentication
* OAuth2
* Basic authentication
* Authorization
* Role-based access
* Method security
* Custom security rules
* Password Encryption
* BCrypt
* Password encoding
* Salt generation

#### 2.3. Build Tool

* Maven/Gradle
* Dependencies
* Plugins
* Build lifecycle
* Profiles
* Properties

### Quản Lý Mã Nguồn

#### 3.1. Git

* Công Cụ
* Git CLI
* GitHub Desktop
* VS Code Integration
* Nền Tảng
* GitHub
* GitLab
* Bitbucket

#### 3.2. Mô Hình Nhánh

- Git Flow

main

└── develop

├── feature/user-authentication

├── feature/book-management

└── feature/order-processing

* Branch Types
* main: Production code
* develop: Development code
* feature: New features
* release: Release preparation
* hotfix: Bug fixes

#### 3.3. Quy Trình Làm Việc

1. Tạo nhánh feature từ develop
2. Phát triển tính năng
3. Tạo pull request
4. Code review
5. Merge vào develop
6. Tạo release từ develop
7. Merge vào main

### Docker

#### 4.1. Containerization

- Dockerfile

# Backend

FROM openjdk:17-jdk-slim

WORKDIR /app

COPY target/\*.jar app.jar

EXPOSE 8080

ENTRYPOINT ["java", "-jar", "app.jar"]

# Frontend

FROM node:16-alpine

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

RUN npm run build

EXPOSE 3000

CMD ["npm", "start"]

#### 4.2. Docker Compose

* docker-compose.yml

version: '3.8'

services:

backend:

build: ./backend

ports:

- "8080:8080"

depends\_on:

- db

- redis

frontend:

build: ./frontend

ports:

- "3000:3000"

depends\_on:

- backend

db:

image: postgres:13

environment:

POSTGRES\_DB: bookstore

POSTGRES\_USER: admin

POSTGRES\_PASSWORD: secret

redis:

image: redis:6

ports:

- "6379:6379"

#### 4.3. Triển Khai

* Build images
* Push to registry
* Pull on server
* Run containers
* Health checks
* Logging
* Monitoring

# Chương 3: Phân tích hệ thống



## **Ngữ cảnh nghiệp vụ và quy trình nghiệp vụ**

* + 1. Ngữ cảnh nghiệp vụ:

Hệ thống quản lý nhà sách là một giải pháp tổng thể nhằm số hóa toàn bộ hoạt động kinh doanh sách của một nhà sách truyền thống, mở rộng khả năng tương tác với khách hàng trực tuyến và nâng cao hiệu quả quản trị nội bộ.

Trong bối cảnh hiện nay, khi nhu cầu mua sắm trực tuyến ngày càng tăng và hoạt động kinh doanh cần được kiểm soát chặt chẽ, hệ thống quản lý nhà sách không chỉ đóng vai trò là một công cụ bán hàng trực tuyến, mà còn là nền tảng hỗ trợ vận hành xuyên suốt từ quản lý kho, chăm sóc khách hàng, khuyến mãi đến phân tích kinh doanh.

Các bên liên quan chính:

|  |  |
| --- | --- |
| Người dùng | Vai trò và nhu cầu |
| Khách hàng | Tìm kiếm, xem chi tiết, mua sách, đánh giá sản phẩm, theo dõi đơn hàng. |
| Nhân viên bán hàng | Hỗ trợ khách hàng trong quá trình mua sắm, xử lý đơn hàng, giải đáp thắc mắc |
| Quản trị viên | Quản lý sách, đơn hàng, kho hàng, khách hàng, chương trình khuyến mãi và theo dõi doanh thu. |
| Hệ thống | Tự động hóa các quy trình xử lý đơn hàng, cập nhật kho, tính khuyến mãi, lưu trữ dữ liệu và xuất báo cáo. |

Tác nhân và tương tác:

* + Khách hàng tương tác với hệ thống thông qua website/app để tìm kiếm, đặt hàng và theo dõi trạng thái mua sắm.
  + Quản trị viên và nhân viên bán hàng thao tác qua giao diện quản trị để quản lý sản phẩm, xử lý đơn hàng, cập nhật dữ liệu.
  + Hệ thống xử lý nghiệp vụ tự động, cung cấp dữ liệu thời gian thực cho các tác nhân và đảm bảo các quy trình diễn ra thông suốt.

Tổng quan dòng nghiệp vụ chính:

* Tìm kiếm & mua sách:

Người dùng tìm kiếm sách → xem chi tiết → thêm vào giỏ → thanh toán → hệ thống xử lý đơn hàng → xuất hóa đơn → cập nhật kho → thông báo trạng thái.

* Quản lý kho & sách:

Quản trị viên nhập sách mới → hệ thống cập nhật danh mục → theo dõi tồn kho → cảnh báo nếu sắp hết → xuất kho theo đơn hàng.

* Báo cáo & phân tích:

Hệ thống tổng hợp số liệu theo thời gian → hiển thị biểu đồ doanh thu, số đơn hàng, sách bán chạy.

* Chăm sóc khách hàng:

Người dùng gửi phản hồi hoặc đánh giá → quản trị viên/CSKH tiếp nhận → xử lý khiếu nại nếu có.

* + 1. Quy trình nghiệp vụ:

Hệ thống quản lý nhà sách bao gồm nhiều quy trình nghiệp vụ phối hợp với nhau nhằm đảm bảo hoạt động kinh doanh vận hành trơn tru, hiệu quả, đồng thời mang lại trải nghiệm tốt cho khách hàng. Các quy trình chính bao gồm:

* + - 1. Quy trình Quản lý sách

Mục tiêu:

* Duy trì thông tin sách đầy đủ, chính xác để phục vụ tra cứu, bán hàng và thống kê.

Các bước:

* Tiếp nhận sách mới từ nhà cung cấp.
* Cập nhật thông tin chi tiết sách vào hệ thống (tên sách, tác giả, thể loại, giá, mô tả, ảnh bìa…).
* Phân loại sách theo thể loại, chủ đề hoặc bộ sưu tập.
* Duyệt nội dung sách hiển thị trên website.
* Cập nhật sách khi có thay đổi thông tin.
* Xóa sách nếu ngừng kinh doanh hoặc lỗi hệ thống.
  + - 1. Quy trình Nhập – Xuất kho

Mục tiêu:

* Đảm bảo hàng hóa (sách) được nhập, xuất và kiểm kê chính xác theo thời gian thực.

Các bước:

* Nhập kho:
  + Xác định nhu cầu và lập đơn đặt hàng.
  + Kiểm tra sách khi nhà cung cấp giao hàng.
  + Cập nhật số lượng và thông tin vào hệ thống kho.
* Xuất kho:
  + Xử lý đơn hàng khách hàng đặt.
  + Trừ số lượng tương ứng trong hệ thống kho.
  + Ghi nhận thông tin hóa đơn và tình trạng giao hàng.
* Kiểm kê:
  + Thực hiện kiểm kê định kỳ.
  + Đối chiếu thực tế và dữ liệu hệ thống.
  + Báo cáo chênh lệch và điều chỉnh kho.
    - 1. Quy trình Bán hàng

Mục tiêu:

* Cung cấp cho khách hàng quy trình mua sách dễ dàng, nhanh chóng và đáng tin cậy.

Các bước:

* Khách hàng tìm kiếm sách trên hệ thống (theo tên, thể loại, tác giả...).
* Xem chi tiết sách (thông tin, tình trạng còn hàng, đánh giá...).
* Thêm vào giỏ hàng và tiến hành thanh toán.
* Chọn phương thức thanh toán (tiền mặt, chuyển khoản, ví điện tử).
* Hệ thống xử lý đơn hàng và cập nhật trạng thái.
* In hóa đơn và cập nhật tồn kho.
* Giao hàng hoặc thông báo nhận sách tại cửa hàng.
  + - 1. Quy trình Quản lý khách hàng

Mục tiêu:

* Xây dựng mối quan hệ lâu dài với khách hàng, nâng cao trải nghiệm mua sắm.

Các bước:

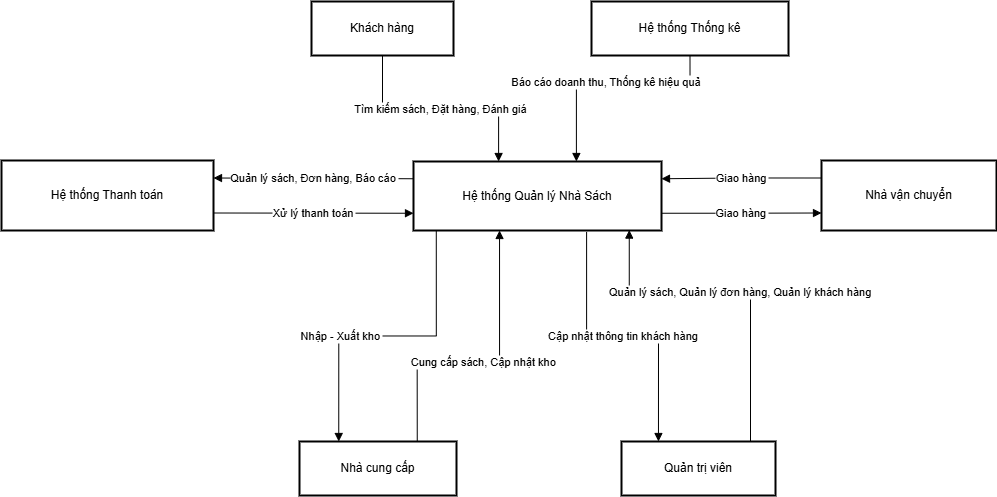
* Khách hàng đăng ký tài khoản hoặc mua hàng không cần đăng ký.
* Lưu trữ thông tin cá nhân và lịch sử mua hàng.
* Cập nhật hồ sơ khách hàng.
* Gửi thông báo khuyến mãi, sách mới qua email/SMS.
* Ghi nhận và xử lý phản hồi, khiếu nại.
* Quản lý điểm thưởng, thẻ thành viên (nếu có).
  + - 1. Quy trình Báo cáo – Thống kê

Mục tiêu:

* Cung cấp thông tin tổng hợp để hỗ trợ quản lý và ra quyết định.

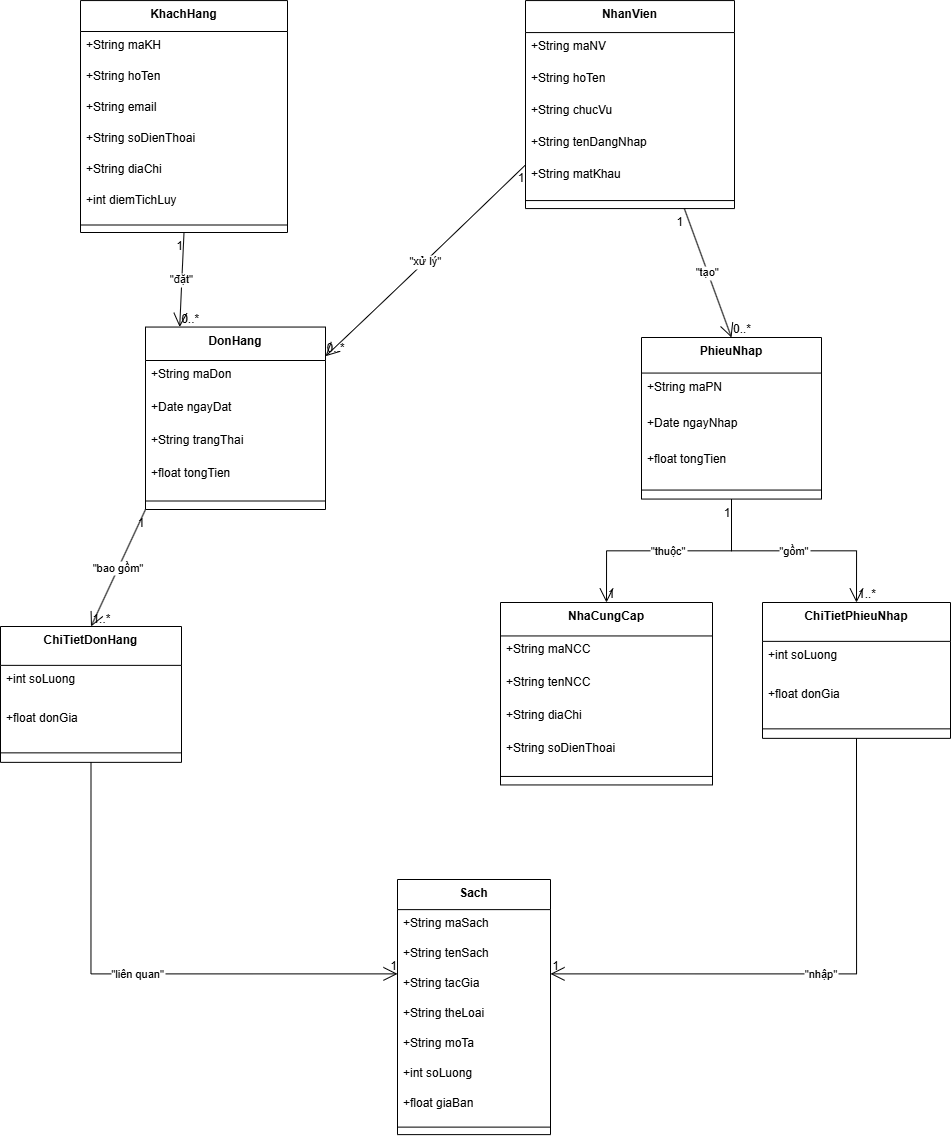
Các bước:

* Hệ thống thu thập dữ liệu bán hàng, kho, khách hàng theo thời gian thực.
* Xuất báo cáo doanh thu ngày/tháng/năm.
* Báo cáo sách bán chạy, hàng tồn kho.
* Xuất báo cáo đơn hàng, hiệu quả khuyến mãi.
* Hỗ trợ xuất dữ liệu Excel hoặc PDF để lưu trữ hoặc phân tích.
  1. **Mô hình ngữ cảnh và mô hình khái niệm**
     1. Mô hình ngữ cảnh (Context Diagram)



Giải thích các thành phần trong mô hình ngữ cảnh:

* Khách hàng: Thực hiện các tác vụ như tìm kiếm sách, đặt hàng, đánh giá sản phẩm, thanh toán đơn hàng.
* Quản trị viên: Quản lý sách, đơn hàng, khách hàng và các báo cáo.
* Nhà cung cấp: Cung cấp sách và cập nhật thông tin kho.
* Nhà vận chuyển: Thực hiện giao hàng cho khách hàng.
* Hệ thống Thanh toán: Xử lý và xác nhận các giao dịch thanh toán.
* Hệ thống Thống kê: Cung cấp các báo cáo về doanh thu, hiệu quả khuyến mãi, tồn kho và các số liệu khác.
  + 1. Mô hình khái niệm



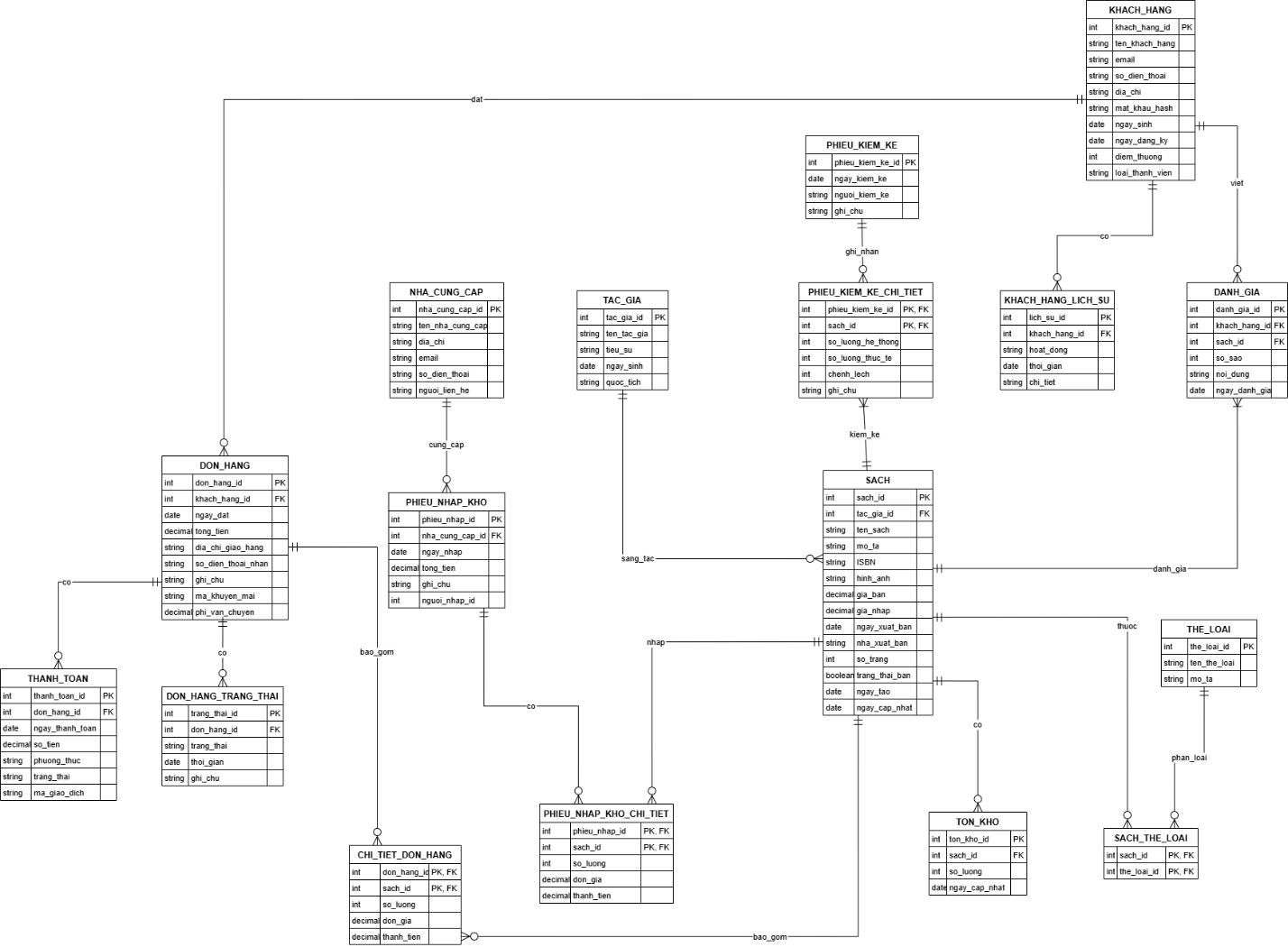
* 1. **Mô hình Use case** 
     1. Use case Khách hàng

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use Case** | **Mô tả chức năng** |
| UC01 | Đăng ký tài khoản | Người dùng tạo tài khoản mới để sử dụng hệ thống. |
| UC02 | Đăng nhập | Người dùng đăng nhập vào hệ thống với tài khoản đã có. |
| UC03 | Tìm kiếm sách | Tìm kiếm sách theo tên, thể loại, tác giả. |
| UC04 | Xem chi tiết sách | Hiển thị thông tin chi tiết sách (giá, mô tả, còn hàng hay không...). |
| UC05 | Thêm sách vào giỏ hàng | Lưu sách đã chọn vào giỏ hàng tạm thời. |
| UC06 | Đặt hàng | Tiến hành tạo đơn đặt hàng từ giỏ hàng. |
| UC07 | Thanh toán | Thực hiện thanh toán đơn hàng bằng nhiều hình thức. |
| UC08 | |  | | --- | |  |  |  | | --- | | Theo dõi đơn hàng | | Xem trạng thái đơn hàng: chờ xử lý, đang giao, đã hoàn thành... |

* + 1. Use case Nhân viên/ Quản trị viên

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use Case** | **Mô tả chức năng** |
| UC09 | |  | | --- | |  |  |  | | --- | | Quản lý sách | | Thêm, sửa, xóa thông tin sách. |
| UC10 | Quản lý đơn hàng | Xem và cập nhật trạng thái đơn hàng. |
| UC11 | Quản lý kho | Nhập, xuất sách, kiểm kê hàng tồn. |
| UC12 | Quản lý khách hàng | Cập nhật thông tin khách hàng, lịch sử mua hàng. |
| UC13 | Quản lý nhân viên | Thêm, phân quyền, vô hiệu hóa tài khoản nhân viên. |
| UC14 | Báo cáo thống kê | Xem và xuất báo cáo doanh thu, đơn hàng, sách bán chạy.. |

* 1. **User Story**
     1. Khách hàng:
        + Tôi muốn tìm kiếm sách theo tên, thể loại, tác giả để dễ dàng tìm được sách mình quan tâm.
        + Tôi muốn xem thông tin chi tiết về sách (giá cả, tình trạng còn hàng, nội dung tóm tắt).
        + Tôi muốn thêm sách vào giỏ hàng và thanh toán.
        + Tôi muốn theo dõi trạng thái đơn hàng của mình.
        + Tôi muốn đánh giá và viết nhận xét về sách đã mua.
     2. Quản lý:
        + Tôi muốn thêm/sửa/xóa thông tin sách trong kho.
        + Tôi muốn quản lý đơn hàng và cập nhật trạng thái giao hàng.
        + Tôi muốn xem báo cáo doanh thu theo ngày/tháng/năm.
        + Tôi muốn quản lý thông tin khách hàng.
        + Tôi muốn theo dõi hàng tồn kho và nhận cảnh báo khi sắp hết hàng.
  2. **Thiết kế giao diện ở mức ý niệm**
  3. **Thiết kế dữ liệu ở mức ý niệm**

****

* 1. **Đặc tả Use case và mô hình**

|  |  |  |
| --- | --- | --- |
|  | | |
| Use Case Number: | UC1 | |
| Use Case Name: | Quản lý sách | |
| Actor (s): | Admin | |
| Maturity: | Focused | |
| Summary: | Cho phép Admin thêm thông tin sách mới, xóa sách và chỉnh sửa thông tin sách. | |
| Basic Course of Events: | Actor Action | System Response |
| 1. Admin vào trang quản lý thông tin sách. |  |
|  | 2. Hệ thống chuyển tiếp đến trang quản lý sách. |
| 3. Admin tùy chọn chức năng thêm. A1, A2, A3, A4 |  |
|  | 4. Hệ thống hiện ra giao diện điền thông tin sách. |
| 5. Admin nhập thông tin sách. E1 |  |
|  | 6. Hệ thống lưu thông tin vào CSDL. |
| Alternative Paths: | A1.  Admin nhấn vào biểu tượng thùng rác.  Hệ thống xóa sách khỏi CSDL. | |
|  | A2.  Admin nhấn vào biêu thượng .  Hệ thống hiện ra giao diện sửa thông tin sách.  Admin sửa thông tin sách  Hệ thống hiển thị thông báo sửa thành công và cập nhật CSDL. | |
|  | A3.  Admin nhấn vào biểu tượng  Hệ thống truy xuất dữ liệu thông qua ID của sách.  Hệ thống hiển thị giao diện chi tiết sách. | |
|  | A4.  Admin nhấn vào thanh tìm kiếm.  Admin nhập từ khóa tìm kiếm thông tin sách.  Hệ thống truy xuất từ cơ sở dữ liệu.  Hệ thống trả về thông tin sách. | |
| Exception Paths: | E1. Hệ thống xác thực thông tin tạo sách không thành công và hiển thị thông báo. | |
| Extension Points: | None | |
| Triggers: | Admin muốn thực hiện các thao tác thêm, sửa, xóa sách. | |
| Assumptions: | None | |
| Preconditions: | Người dùng phải là Admin hệ thống. | |
| Post Conditions: | Admin thực hiện các thao tác thêm, xóa, sửa thành công. | |
| Reference : | None | |
| Author(s): | Trần Minh Chiến | |
| Date: | 10/05/2025 | |
| Activity Diagram: | | |

# CHƯƠNG 4: THIẾT KẾ HỆ THỐNG

4.1. Tổng quan thiết kế hệ thống

### 4.1.1 Sơ đồ khối ý niệm của hệ thống

Hệ thống được thiết kế theo mô hình ba lớp truyền thống: giao diện người dùng (frontend), dịch vụ xử lý nghiệp vụ (backend), và hệ quản trị cơ sở dữ liệu (database). Sơ đồ sau mô tả kiến trúc tổng quan (được thể hiện bằng ngôn ngữ Mermaid trong tài liệu gốc):

Ảnh có chứa văn bản, hàng, biểu đồ, Sơ đồ

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa biểu đồ, hàng, Sơ đồ, văn bản

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa hàng, biểu đồ, Song song, Kế hoạch

Nội dung do AI tạo ra có thể không chính xác.

Giải thích: Người dùng tương tác với giao diện web thông qua trình duyệt. Frontend được xây dựng bằng ReactJS, giao tiếp với backend qua REST API. Backend xử lý logic nghiệp vụ, xác thực và phân quyền thông qua JWT. Dữ liệu được lưu trữ trong PostgreSQL. Redis được sử dụng để tăng tốc truy vấn và giảm tải. Hệ thống giám sát và ghi log nhằm đảm bảo ổn định và dễ dàng bảo trì.

### 4.1.2 Lý do chọn lựa công nghệ

- ReactJS: Thư viện JavaScript phổ biến, dễ phát triển giao diện người dùng hiện đại.  
- Spring Boot: Framework Java mạnh mẽ, phù hợp cho việc xây dựng RESTful API.  
- PostgreSQL: Hệ quản trị CSDL quan hệ mạnh, hỗ trợ tốt các truy vấn phức tạp.  
- Redis: Bộ nhớ đệm giúp cải thiện tốc độ truy xuất dữ liệu.  
- JWT: Giải pháp xác thực nhẹ, không cần lưu trạng thái trên server.  
- Docker: Cho phép đóng gói toàn bộ hệ thống thành container, thuận tiện triển khai.  
- GitHub Actions: Tự động hóa quy trình kiểm thử và triển khai (CI/CD).

Kết luận: Kiến trúc hệ thống đảm bảo phân tách rõ ràng các vai trò chức năng, đồng thời tận dụng các công nghệ hiện đại nhằm tối ưu hiệu suất, khả năng mở rộng và bảo trì.

## 4.2 Khung nhìn mức thành phần (Decomposition View – C4 Model)

Để mô tả rõ cấu trúc hệ thống, chúng tôi sử dụng mô hình C4 (Context, Container, Component, Code). Mỗi mức mô hình cung cấp góc nhìn khác nhau về hệ thống.

Ảnh có chứa văn bản, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

### 4.2.1 System Context Diagram

Ảnh có chứa văn bản, hàng, ảnh chụp màn hình, biểu đồ

Nội dung do AI tạo ra có thể không chính xác.

### 4.2.2 Container Diagram

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Nội dung do AI tạo ra có thể không chính xác.

Mô tả: Web App cung cấp giao diện tương tác với người dùng. API đảm nhận logic nghiệp vụ và kiểm soát quyền truy cập. Redis hỗ trợ cache dữ liệu giúp tăng hiệu suất. Tất cả container được triển khai thông qua Docker để đảm bảo nhất quán môi trường.

### 4.2.3 Component Diagram

Ảnh có chứa văn bản, hàng, biểu đồ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Controllers: tiếp nhận và xử lý yêu cầu từ frontend. Services: chứa logic nghiệp vụ. Repositories: truy vấn dữ liệu từ cơ sở dữ liệu.  
Kết luận: Việc chia hệ thống thành nhiều mức theo mô hình C4 giúp dễ dàng quản lý, mở rộng và đảm bảo trách nhiệm rõ ràng giữa các thành phần.

## 4.3 Khung nhìn mức dữ liệu (Data/Transaction View)

### 4.3.1 Mô hình dữ liệu (ERD)

Ảnh có chứa văn bản, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

### 4.3.2 Luồng giao dịch đặt hàng

Ảnh có chứa văn bản, ảnh chụp màn hình, biên lai, hàng

Nội dung do AI tạo ra có thể không chính xác.

Kết luận: Mô hình dữ liệu được thiết kế chuẩn hóa và luồng giao dịch đảm bảo an toàn, toàn vẹn dữ liệu khi nhiều người dùng thao tác đồng thời.

## 4.4 Triển khai hệ thống và CI/CD

### 4.4.1 Triển khai bằng Docker

version: '3.8' services:  
 frontend:  
 build: ./frontend ports:  
 - "3000:3000" backend:  
 build: ./backend ports:  
 - "8080:8080" depends\_on:  
 - db db:  
 image: postgres:14 environment:  
 POSTGRES\_USER: admin POSTGRES\_PASSWORD: secret POSTGRES\_DB: bookstore

### 4.4.2 Quy trình CI/CD với GitHub Actions

CI/CD giúp tự động hóa quá trình kiểm thử, build và triển khai ứng dụng sau mỗi lần cập nhật mã nguồn. Quy trình cơ bản:  
- Khi push code lên nhánh chính (main), GitHub Actions tự động:  
- Kiểm tra cú pháp và chạy unit test  
- Build Docker image  
- Triển khai ứng dụng lên server thử nghiệm hoặc production.

Kết luận: Với Docker và CI/CD, hệ thống đảm bảo được triển khai nhất quán và nhanh chóng, giảm rủi ro do thao tác thủ công.

## 4.5 Giám sát hệ thống (Resilience View)

Để đảm bảo hệ thống hoạt động ổn định, các cơ chế giám sát và ghi log được tích hợp:  
- Logging: Spring Boot sử dụng Logback để ghi log hệ thống.  
- Tracing: Áp dụng OpenTelemetry hoặc Zipkin để theo dõi request xuyên suốt các thành phần.  
- Monitoring: Prometheus thu thập số liệu từ các container (CPU, bộ nhớ, số request lỗi…). Grafana được dùng để trực quan hóa các chỉ số này.  
- Alerting: Khi một chỉ số vượt ngưỡng, hệ thống gửi email cảnh báo tới quản trị viên.

Ảnh có chứa văn bản, biểu đồ, hàng, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Kết luận: Việc áp dụng cơ chế giám sát giúp kịp thời phát hiện sự cố và tối ưu hiệu suất vận hành trong môi trường thực tế.