**TRƯỜNG ĐẠI HỌC CẦN THƠ**

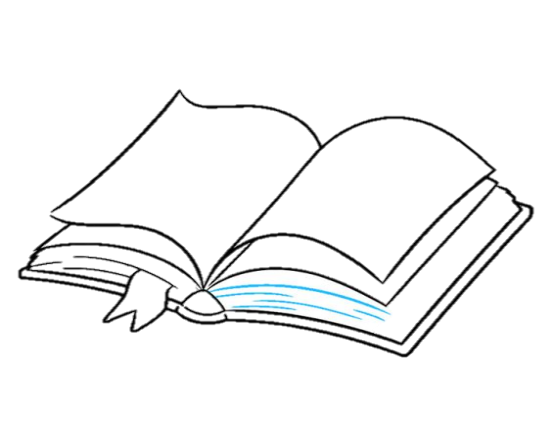
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO BÀI TẬP LỚN**

**PHÁT TRIỂN ỨNG DỤNG WEB**

**Nhóm học phần: CT449**



**ĐỀ TÀI: QUẢN LÝ MƯỢN SÁCH**

**Giảng viên hướng dẫn:**

**Ths. Nguyễn Minh Trung**

**Sinh viên thực hiện: Nguyễn Tấn Đức**

**MSSV: B2017035**

**Lớp: DI20Z6A3**

**Khóa: 46  
  
 Cần Thơ, tháng 25/4/2024**

**LỜI CẢM ƠN**

Để hoàn thành được bài báo cáo này, em xin chân thành cảm ơn Ban Giám hiệu, các khoa, phòng và quý thầy, cô của trường Công nghệ thông tin và Truyền thông, những người đã tận tình giúp đỡ và tạo điều kiện cho em trong quá trình học tập. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến thầy Ths. Nguyễn Minh Trung - người đã trực tiếp giảng dạy và hướng dẫn em thực hiện bài báo cáo này bằng tất cả lòng nhiệt tình và sự quan tâm sâu sắc.

Trong quá trình thực hiện bài báo cáo này, do hiểu biết còn nhiều hạn chế nên bài làm khó tránh khỏi những thiếu sót. Em rất mong nhận được những lời góp ý của thầy để bài báo cáo ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

**MỤC LỤC**

[**LỜI CẢM ƠN** i](#_Toc151694568)

[DANH MỤC SƠ ĐỒ ii](#_Toc151694569)i

[DANH MỤC HÌNH ẢNH v](#_Toc151694571)

[TÓM TẮT v](#_Toc151694572)i

[PHẦN GIỚI THIỆU 1](#_Toc151694573)

[1. Đặt vấn đề 1](#_Toc151694574)

[2. Mục tiêu đề tài 2](#_Toc151694575)

[PHẦN NỘI DUNG 3](#_Toc151694576)

[Chương 1. Đặc tả yêu cầu 4](#_Toc151694577)

[Chương 2. Cơ sở lý thuyết 5](#_Toc151694578)

[1. Tổng quan về MEVN STACK: 5](#_Toc151694579)

[2. MongoDB: 6](#_Toc151694580)

[3. Restful API: 6](#_Toc151694581)

[4. Express.js: 7](#_Toc151694582)

[5. Vue.js: 8](#_Toc151694583)

[6. Node.js: 8](#_Toc151694584)

[7. Tổng quan về Mongoose: 9](#_Toc151694586)

[8. VS CODE: 9](#_Toc151694586)

[Chương 3. Các sơ đồ của Website 10](#_Toc151694587)

[1. Sơ đồ Use case 10](#_Toc151694588)

[2. Cở sở dữ liệu 20](#_Toc151694591)

[Mô hình cơ cở dữ liệu CDM 20](#_Toc151694592)

[Chương 4. Kết quả đạt được 15](#_Toc151694595)

DANH MỤC SƠ ĐỒ

[Sơ đồ 1.1: Sơ đồ use case tổng quát 10](#_Toc151693979)

[Sơ đồ 1.2: Sơ đồ use case Quản lý độc giả 12](#_Toc151693980)

[Sơ đồ 1.3: Sơ đồ use case Quản lý NXB 13](#_Toc151693981)

[Sơ đồ 1.4: Sơ đồ use Quản lí tài khoản khách hàng 14](#_Toc151693982)

[Sơ đồ 1.5: Sơ đồ use case: Quản lí sách 15](#_Toc151693983)

[Sơ đồ 1.6: Sơ đồ use Đăng ký mượn sách 17](#_Toc151693982)

[Sơ đồ 1.7: Sơ đồ use case: Đăng ký trả sách 18](#_Toc151693983)

[Sơ đồ 1.8: Sơ đồ use tìm kiếm sách 19](#_Toc151693982)

DANH MỤC HÌNH ẢNH

[Hình 1: Mô hình CDM 12](#_Toc151694160)

[Hình 2: Giao diện Tìm kiếm sách 23](#_Toc151694161)

[Hình 3: Giao diện Đăng ký mượn 23](#_Toc151694162)

[Hình 4: Giao diện Đăng ký trả 23](#_Toc151694163)

[Hình 5: Giao diện Quản lý sách 24](#_Toc151694164)

[Hình 6: Giao diện Quản lý NXB 24](#_Toc151694165)

[Hình 7: Giao diện Hiển thị danh sách NXB 24](#_Toc151694166)

[Hình 8: Giao diện Quản lí Độc giả 25](#_Toc151694167)

[Hình 9: Giao diện Hiển thị danh sách Độc giả 25](#_Toc151694168)

[Hình 10: Giao diện Hiển thị danh sách Độc giả đang mượn 25](#_Toc151694168)

TÓM TẮT

* Công nghệ thông tin đang tiến bộ mạnh mẽ, mang lại nhiều lợi ích cho xã hội và đời sống con người. Việc sử dụng internet và các hình thức kinh doanh trực tuyến ngày càng phổ biến, làm cho việc tra cứu tìm kiếm trở nên tiện lợi hơn bao giờ hết. Do đó, nhiều thư viện đang áp dụng hệ thống quản lý mượn sách trực tuyến để người đọc có thể dễ dàng tìm kiếm thông tin và thực hiện các giao dịch mượn sách một cách thuận tiện.
* Mục tiêu của dự án "**Xây dựng Hệ thống Quản lý Mượn Sách**" là phát triển một hệ thống dựa trên công nghệ MEVN STACK (MongoDB, Express, Vue.js, Node.js). Để đạt được mục tiêu này, cần có sự nghiên cứu và hiểu biết sâu sắc về công nghệ MEVN STACK được sử dụng rộng rãi trong phát triển web hiện nay.
* Dự án mục tiêu đề ra là tạo ra một hệ thống quản lý mượn sách với giao diện thân thiện, dễ sử dụng, giúp quá trình mượn sách diễn ra nhanh chóng, an toàn và hiệu quả.

PHẦN GIỚI THIỆU

* 1. Đặt vấn đề
* Đọc sách không chỉ là một sở thích cá nhân mà còn là một đề tài đầy đa chiều và tranh cãi trong cộng đồng yêu sách.
* Giá trị Văn hóa và Tri thức: Các cuốn sách không chỉ là tài liệu đọc mà còn là biểu hiện của văn hóa và tri thức, là nguồn cảm hứng cho người đọc.
* Việc áp dụng công nghệ thông tin vào quản lý thư viện giúp tăng cường tính hiệu quả và sự đa dạng của dịch vụ. Sự ra đời của hệ thống quản lý thư viện trực tuyến giúp giải quyết vấn đề mượn sách, cải thiện quy trình cho người quản lý thư viện và đáp ứng nhu cầu đọc sách của người đọc một cách thuận tiện hơn.
  1. Mục tiêu đề tài

- Mục tiêu tổng quát:

Xây dựng hệ thống quản lý mượn sách trực tuyến với các chức năng hoàn chỉnh, dễ sử dụng và thuận tiện cho cả người quản lý thư viện và người đọc, giúp người đọc có thể mượn sách một cách thuận tiện và nhanh chóng, đồng thời giúp thư viện quản lý sách và độc giả một cách hiệu quả..

- Mục tiêu cụ thể:

Hệ thống Quản lý mượn sách gồm có 2 website: **Trang Người Dùng** và **Trang Quản Trị**

**Trang Người Dùng ( Độc Giả ) :**

- Gồm 3 chức năng:

Liệt kê các sách và Tìm kiếm:

+ Hiển thị danh sách các sách có trong thư viện.

+ Cho phép người đọc tìm kiếm sách theo tiêu chí như tên sách, tác giả, thể loại.

+ Xem thông tin chi tiết về mỗi cuốn sách bao gồm tóm tắt nội dung, tác giả, số lượng sách còn lại.

Đăng ký mượn sách:

+ Cho phép người đọc đăng ký mượn sách.

+ Chọn sách cần mượn từ danh sách **Sách** có sẵn.

+ Chọn thời gian mượn và thông tin liên lạc.

+ Xác nhận đăng ký mượn sách.

- Đăng ký trả sách:

+ Cho phép người đọc đăng ký trả sách.

+ Xác nhận trả sách và cập nhật trạng thái mượn trả sách trong hệ thống.

+ Cập nhật thông tin về số lượng sách còn lại sau khi trả sách.

- Trang Quản Trị có những chức năng:

1. Quản lý Sách.
2. Quản lý Nhà Xuất Bản.
3. Quản lý Độc Giả.
4. Quản lý Mượn sách.

- Chức năng:

1. Quản lý Sách:

+ Thêm sách mới vào hệ thống với thông tin như tên sách, tác giả, số lượng, đơn giá,….

+ Chỉnh sửa thông tin về sách như cập nhật lại thông tin, số lượng ,tên,……

+ Xóa sách khỏi hệ thống nếu cần thiết, ví dụ như sách đã bị hỏng hoặc đã quá lỗi thời.

1. Quản lý Nhà Xuất Bản:

+ Thêm mới thông tin về các NXB, bao gồm tên, địa chỉ, thông tin liên hệ.

+ Hiệu chỉnh thông tin về các NXB như: cập nhật thông liên hệ hoặc địa chỉ mới.

+ Xóa thông tin về các NXB không còn hoạt động hoặc không còn liên quan đến hệ thống.

1. Quản lý Độc Giả:

+ Thêm mới thông tin về các độc giả bao gồm: tên, địa chỉ, thông tin liên hệ,….

+ Hiệu chỉnh thông tin về các độc giả như: cập nhật thông liên hệ mới.

+ Xóa thông tin về các độc giả không còn sử dụng dịch vụ thư viện hoặc có lỗi trong thông tin.

1. Quản lý Mượn sách:

+ Ghi nhận việc mượn sách cho mỗi độc giả, bao gồm thông tin về sách mượn, ngày mượn và ngày trả.

+ Cập nhật thông tin về việc trả sách bao gồm: ngày trả, tình trạng của sách trả.

PHẦN NỘI DUNG

CHƯƠNG I: ĐẶC TẢ YÊU CẦU

Việc xây dựng "Hệ thống quản lý mượn sách" nhằm tạo ra một môi trường thuận tiện cho việc quản lý và sử dụng sách trong một thư viện hoặc cơ sở lưu trữ sách.

Giao diện và chức năng cho người quản lý (Admin):

* Quản lý danh sách sách trong thư viện, bao gồm thêm mới, chỉnh sửa và xóa sách.
* Quản lý thông tin về người mượn, bao gồm quản lý thông tin tài khoản và lịch sử mượn sách.

Quản lý quy trình mượn và trả sách, bao gồm việc ghi nhận thông tin mượn sách, trả sách và tính phí phạt (nếu có).

* Giao diện và chức năng cho người mượn sách:

+ Xem danh sách sách có sẵn trong thư viện.

+ Tìm kiếm sách theo tiêu chí cụ thể.

+ Xem thông tin chi tiết về từng cuốn sách, bao gồm tên sách, tác giả, mô tả, hình ảnh.

+ Thực hiện việc mượn sách.

+ Thực hiện việc trả sách và cập nhật thông tin về sách đã trả.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

* 1. Tổng quan về MEVN STACK:

Khái niệm:

MEVN STACK là sự kết hợp giữa của **MongoDB, Express.js, Vue.js, Node.js** và khiến cho việc xây dựng những ứng dụng web trở nên mạnh mẽ và đơn giản hơn bao giờ hết.

Ưu và nhược điểm của công nghệ MEVN STACK:

- Ưu điểm:

* **Dễ dàng xây dựng ứng dụng web động**: MEVN stack cung cấp một giải pháp đầy đủ cho việc xây dựng các ứng dụng web động. Bằng cách kết hợp MongoDB để lưu trữ dữ liệu, Express.js để quản lý tương tác với dữ liệu, Vue.js để tạo giao diện người dùng và Node.js để xây dựng máy chủ, MEVN stack cung cấp một cách tiếp cận đơn giản và hiệu quả để phát triển ứng dụng web động.
* **Tính linh hoạt**: Mỗi thành phần trong MEVN stack đều rất linh hoạt và cho phép các nhà phát triển tùy chỉnh ứng dụng của mình theo nhu cầu của họ. MongoDB cung cấp một cách tiếp cận linh hoạt cho việc lưu trữ dữ liệu, Express.js cho phép các nhà phát triển xây dựng các API RESTful động, Vue.js cho phép các nhà phát triển tạo giao diện người dùng phức tạp và Node.js cho phép các nhà phát triển xây dựng ứng dụng web động đa nền tảng.
* **Hiệu năng cao**: MEVN stack sử dụng Node.js để xây dựng máy chủ, cung cấp khả năng xử lý đồng thời cao. Điều này giúp các ứng dụng xây dựng bằng MEVN stack có thể xử lý nhiều yêu cầu cùng một lúc, tăng hiệu suất và tốc độ phản hồi.
* **Cộng đồng lớn**: MEVN stack được sử dụng rộng rãi trong cộng đồng phát triển web, vì vậy có rất nhiều tài liệu và hỗ trợ từ cộng đồng. Điều này giúp các nhà phát triển dễ dàng tìm kiếm thông tin và giải quyết các vấn đề khi phát triển ứng dụng.

- Nhược điểm:

* **Không phù hợp cho các ứng dụng lớn**: MEVN stack không phù hợp cho các ứng dụng lớn và phức tạp, do các công nghệ trong stack này chủ yếu được sử dụng cho các ứng dụng web đơn giản và vừa phải. Nếu bạn đang phát triển một ứng dụng web lớn và phức tạp hơn, bạn có thể cần sử dụng các công nghệ khác hoặc tùy chỉnh MEVN stack để đáp ứng nhu cầu của mình.
* **Cần có kiến thức về nền tảng JavaScript**: Để phát triển các ứng dụng với MEVN stack, bạn cần có kiến thức về nền tảng JavaScript, bao gồm cả các framework Vue.js và Node.js. Nếu bạn không có kinh nghiệm với JavaScript, việc học và sử dụng MEVN stack có thể khó khăn.
* **Độ phức tạp của MongoDB**: Mặc dù MongoDB là một cơ sở dữ liệu NoSQL rất phổ biến, nhưng nó có một số đặc điểm phức tạp, đặc biệt là khi sử dụng các tính năng phức tạp như đa lưu trữ hoặc quản lý giao dịch. Điều này có thể khiến cho việc phát triển và bảo trì ứng dụng với MEVN stack trở nên khó khăn hơn.
* **Khả năng mở rộng giới hạn**: MEVN stack cung cấp một giải pháp đơn giản và nhanh chóng cho việc xây dựng ứng dụng web động, tuy nhiên, khả năng mở rộng của nó có thể bị giới hạn. Nếu ứng dụng của bạn phát triển mạnh mẽ và có nhu cầu mở rộng, bạn có thể cần phải tùy chỉnh MEVN stack hoặc chuyển sang các công nghệ khác để đáp ứng nhu cầu của mình.
  1. MongoDB:

A logo of a computer chip

Description automatically generated

- Là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, phát triển bởi MongoDB . Các ứng dụng được phát triển trên MongoDB có thể được triển khai trên các nền tảng khác nhau, bao gồm các môi trường đám mây và máy chủ riêng.

- MongoDB được sử dụng rộng rãi trong các ứng dụng web, đặc biệt là các ứng dụng có nhu cầu xử lý dữ liệu lớn và phức tạp, như các ứng dụng thương mại điện tử, các hệ thống giám sát và các ứng dụng chạy trên đám mây.

* 1. Restful API:

Khái niệm:

RESTful API (Representational State Transfer) là một kiểu kiến trúc ứng dụng mạng dựa trên giao thức HTTP, giúp các ứng dụng giao tiếp với nhau thông qua các phương thức HTTP như GET, POST, PUT, DELETE. RESTful API giúp đơn giản hóa việc truyền tải dữ liệu giữa client và server dưới dạng JSON hoặc XML.

Ưu điểm và hạn chế:

- Ưu điểm:

* Dễ hiểu và sử dụng: RESTful API sử dụng các phương thức HTTP thông dụng, giúp dễ dàng triển khai và tích hợp.
* Tính nhất quán: RESTful API đảm bảo tính nhất quán trong việc truy cập và quản lý dữ liệu.
* Khả năng mở rộng: RESTful API hỗ trợ khả năng mở rộng tốt, giúp dễ dàng thêm các tính năng mới.
* Bảo mật: RESTful API hỗ trợ nhiều cơ chế xác thực và ủy quyền, giúp bảo vệ dữ liệu và tài nguyên.
* Ngôn ngữ và nền tảng độc lập: RESTful API có thể hoạt động trên nhiều ngôn ngữ và nền tảng khác nhau.

- Hạn chế:

1. Trạng thái phiên (stateless): RESTful API không lưu trữ trạng thái phiên giữa các yêu cầu, điều này có thể gây khó khăn trong việc xử lý các tác vụ phức tạp.
2. Tốc độ truy vấn: RESTful API có thể chậm hơn so với các giao thức khác như gRPC hay GraphQL, đặc biệt khi cần truy vấn nhiều dữ liệu liên quan.
3. Over-fetching và under-fetching: RESTful API có thể trả về nhiều dữ liệu hơn hoặc ít hơn so với nhu cầu của client, dẫn đến lãng phí tài nguyên và băng thông.
4. Khó khăn trong phiên bản hóa: Việc cập nhật và quản lý phiên bản của RESTful API có thể gây khó khăn, khiến cho việc duy trì và nâng cấp hệ thống trở nên phức tạp.
   1. Express.js:
5. - Express.js là một framework web được xây dựng trên nền tảng Node.js. Nó được phát triển nhằm giúp cho việc xây dựng ứng dụng web với Node.js trở nên đơn giản và nhanh chóng hơn. Express.js cung cấp cho các nhà phát triển các tính năng và công cụ để xử lý các yêu cầu HTTP, xử lý các phản hồi, quản lý session và cookie, và nhiều tính năng khác.
6. - Express.js được thiết kế để đơn giản hóa việc xây dựng các ứng dụng web và API với Node.js. Nó cho phép các nhà phát triển tạo ra các định tuyến (routing) dựa trên các yêu cầu HTTP, tạo middleware để xử lý các yêu cầu, và cung cấp các công cụ để quản lý các kết nối và phiên làm việc.

5. Vue.js:

- Vue.js là một framework JavaScript mã nguồn mở được sử dụng để xây dựng các ứng dụng web động và giao diện người dùng (UI). Nó được thiết kế để cung cấp một cách tiếp cận dễ dàng cho việc xây dựng các ứng dụng phức tạp và tương tác với người dùng.

- Vue.js cho phép các nhà phát triển tạo ra các thành phần (components) tái sử dụng để xây dựng giao diện người dùng và quản lý trạng thái ứng dụng trong một cách hiệu quả. Nó cũng cung cấp các công cụ để liên kết dữ liệu (data binding) và xử lý sự kiện (event handling), giúp cho việc phát triển các ứng dụng web trở nên đơn giản và nhanh chóng hơn.

- Vue.js có thiết kế linh hoạt và có thể tích hợp vào các ứng dụng web hiện có một cách dễ dàng. Nó hỗ trợ các tính năng như hot-reload (tự động tải lại khi có thay đổi), Vuex (trạng thái quản lý), Vue Router (định tuyến) và nhiều tính năng khác để giúp cho việc phát triển các ứng dụng web trở nên dễ dàng và linh hoạt hơn.

6. Node.js:

- Node.js là một nền tảng phát triển được xây dựng trên Chrome's V8 JavaScript engine để cho phép các nhà phát triển sử dụng JavaScript để phát triển các ứng dụng trên môi trường máy chủ. Nó cho phép các nhà phát triển xây dựng các ứng dụng web và các dịch vụ mạng khác một cách dễ dàng và hiệu quả.

- Node.js cho phép các nhà phát triển xây dựng các ứng dụng web với hiệu suất cao bằng cách sử dụng JavaScript để xử lý các yêu cầu từ phía máy khách và chuyển tiếp chúng đến máy chủ. Nó cũng cung cấp các tính năng để xử lý các yêu cầu không đồng bộ (asynchronous), giúp cho các ứng dụng web trở nên nhanh chóng và hiệu quả hơn.

- Node.js có thiết kế linh hoạt và có thể tích hợp vào các ứng dụng web hiện có một cách dễ dàng. Nó có thể được sử dụng để xây dựng các ứng dụng web đơn giản hoặc các ứng dụng web phức tạp, và hỗ trợ các tính năng như mã nguồn mở, quản lý gói, và phát triển cộng đồng.

* 1. Tổng quan về Mongoose:
* Mongoose là một thư viện ODM (Object-Document Mapping) cho MongoDB và Node.js, cho phép lập trình viên dễ dàng tương tác với cơ sở dữ liệu MongoDB thông qua các đối tượng JavaScript.
* Mongoose cung cấp các tính năng như:
* **Validation**: Kiểm tra tính hợp lệ của dữ liệu trước khi lưu vào cơ sở dữ liệu.
* **Query building**: Xây dựng các truy vấn đến cơ sở dữ liệu MongoDB một cách dễ dàng.
* **Middleware**: Cho phép định nghĩa các hàm middleware được thực hiện trước hoặc sau các hoạt động như lưu hoặc xóa đối tượng.
* **Schema definition**: Cho phép định nghĩa các lược đồ (schema) cho các đối tượng được lưu trữ trong cơ sở dữ liệu.
* **Population**: Cho phép lấy dữ liệu từ các bảng liên quan đến nhau (references) trong cơ sở dữ liệu MongoDB.
  1. VS CODE:

A logo for a software company

Description automatically generated

- Visual Studio Code chính là ứng dụng cho phép biên tập, soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế website một cách nhanh chóng. Visual Studio Code hay còn được viết tắt là VS code. Trình soạn thảo này vận hành mượt mà trên cacs nền tảng như Windows, macOS, Linux. Hơn thế nữa, VS code còn cho khả năng tương thích với những thiết bị máy tính có cấu hình tầm trung vẫn có thể sử dụng dễ dàng

CHƯƠNG III: CÁC SƠ ĐỒ CỦA WEBSITE

* 1. Sơ đồ Use case

Sơ đồ Use case tổng quát của hệ thống

Sơ đồ 1.1: Sơ đồ use case tổng quát

A diagram of a company

Description automatically generated

* + 1. Đặc tả sơ đồ chức năng use case tổng quát

Bảng 1.1: Đặc tả sơ đồ chức năng use case tổng quát

|  |  |  |
| --- | --- | --- |
| STT | Tên use case | Diễn giải |
| 1 | Đăng ký mượn sách | Cho phép Độc giả chọn sách, cung cấp thông tin đăng ký, và nhận sách từ thư viện sau khi hệ thống xác nhận việc mượn.. |
| 2 | Đăng ký trả sách | Độc giả có thể đăng ký trả lại sách sau khi đọc xong. |
| 3 | Đăng nhâp. | Vào được giao diện của hệ thống mượn sách. |
| 4 | Tìm kiếm sách | Giúp độc giả có thể tìm kiếm các sách theo từ khóa bất kì khi nhập vào. |
| 5 | Quản lý sách | Hệ thống cho phép quản trị viên thực hiện các hoạt động liên quan đến quản lý sách trong thư viện như hiệu chỉnh và cập nhật. |
| 6 | Quản lý độc giả | Hệ thống cho phép quản trị viên thực hiện các hoạt động quản lý độc giả, bao gồm thêm mới, hiệu chỉnh và xóa thông tin. |
| 7 | Quản lý mượn sách | Hệ thống cho phép quản trị viên và thủ thư thực hiện các hoạt động liên quan đến việc mượn và trả sách cho độc giả. |
| 8 | Quản lý Nhà sản xuất | Hệ thống cho phép quản trị viên thực hiện các hoạt động quản lý thông tin về các nhà xuất bản. |

1.2 Sơ đồ use case quản lý độc giả

A diagram of a diagram

Description automatically generated

**Sơ đồ 1.2: Sơ đồ use case Quản lí Độc giả.**

|  |  |
| --- | --- |
| **Tác Nhân** | **Admin** |
| **Mô tả** | Quản lý thông tin về các độc giả trong hệ thống thư viện. |
| **Điều kiện** | Nhân viên hoặc quản trị viên đang đăng nhập vào hệ thống. |
| **Luồng sự kiện chính** | 1. Nhân viên thêm mới thông tin độc giả vào hệ thống. 2. Hệ thống lưu trữ thông tin mới và cập nhật cơ sở dữ liệu. 3. Nhân viên thư viện hiệu chỉnh thông tin độc giả đã có trong hệ thống. 4. Hệ thống cập nhật thông tin và lưu trữ các thay đổi. 5. thư viện xóa thông tin về độc giả khỏi hệ thống khi cần thiết. 6. Hệ thống xóa thông tin và cập nhật cơ sở dữ liệu. |
| **Luồng sự kiện rẽ nhánh** | - Nếu thông tin độc giả không hợp lệ hoặc bị thiếu, hệ thống thông báo lỗi và yêu cầu nhập lại. |
| **Kết quả** | Các hoạt động quản lý độc giả được thực hiện một cách chính xác và hiệu quả. |

**1.3 Sơ đồ use case quản lý** **Nhà xuất bản**

**A diagram of a person with text

Description automatically generated**

**Sơ đồ 1.3: Sơ đồ use case Quản lí Nhà xuất bản.**

|  |  |
| --- | --- |
| **Tác Nhân** | **Admin** |
| **Mô tả** | - Quản trị viên thực hiện các hoạt động liên quan đến quản lý thông tin về các nhà xuất bản trong hệ thống thư viện. |
| **Điều kiện** | - Quản trị viên đã đăng nhập vào hệ thống. |
| **Luồng sự kiện chính** | 1. Quản trị viên chọn chức năng "Quản lý Nhà Xuất Bản". 2. Hệ thống hiển thị danh sách các nhà xuất bản có trong cơ sở dữ liệu 3. Quản trị viên thực hiện các hành động như thêm mới,   sửa đổi hoặc xóa thông tin về nhà xuất bản.   1. Hệ thống cập nhật cơ sở dữ liệu sau mỗi hành động của quản trị viên 2. Quản trị viên hoàn tất công việc và thoát khỏi chức năng "Quản lý Nhà Xuất Bản". |
| **Luồng sự kiện rẽ nhánh** | + Nếu quản trị viên chọn thêm mới nhà xuất bản:   1. Quản trị viên nhập thông tin mới cho nhà xuất bản (tên, địa chỉ, liên hệ). 2. Hệ thống kiểm tra thông tin nhập và thêm nhà xuất bản mới vào cơ sở dữ liệu.   + Nếu quản trị viên chọn sửa đổi thông tin nhà xuất bản:   1. Quản trị viên chọn nhà xuất bản cần sửa đổi. 2. Quản trị viên cập nhật thông tin mới cho nhà xuất bản. 3. Hệ thống cập nhật thông tin và lưu vào cơ sở dữ liệu.   + Nếu quản trị viên chọn xóa thông tin nhà xuất bản:   1. Quản trị viên chọn nhà xuất bản cần xóa. 2. Hệ thống xác nhận và loại bỏ thông tin nhà xuất bản khỏi cơ sở dữ liệu. |
| **Kết quả** | - Các hoạt động quản lý nhà xuất bản được thực hiện một cách chính xác và hiệu quả, và cơ sở dữ liệu được cập nhật đồng bộ với các thay đổi. |

**1.4 Sơ đồ use case quản lý** **Sách.**

A diagram of a person with text

Description automatically generated

**Sơ đồ 1.4: Sơ đồ use case Quản lí Sách.**

|  |  |
| --- | --- |
| **Tác nhân** | **Admin** |
| **Mô tả** | Quản lý thông tin về sách trong thư viện. |
| **Điều kiện** | Đăng nhập vào hệ thống quản trị thư viện. |
| **Luồng sự kiện chính** | 1. Nhân viên thêm sách mới vào hệ thống. 2. Nhân viên hiệu chỉnh thông tin sách hiện có. 3. Nhân viên xóa sách khỏi hệ thống. |
| **Luồng sự kiện rẽ nhánh** | - Nếu thông tin sách không hợp lệ:  + Thủ thư thông báo lỗi và yêu cầu nhập lại thông tin. |
| **Kết quả** | Cập nhật thông tin về sách trong cơ sở dữ liệu thư viện. |

**1.5 Sơ đồ use case quản lý** **mượn** **sách.**

**A diagram of a diagram

Description automatically generated**

**Sơ đồ 1.5: Sơ đồ use case Quản lí mượn sách.**

|  |  |
| --- | --- |
| **Tác nhân** | **Admin** |
| **Mô tả** | - Quản lý thông tin về sách được mượn trong thư viện. |
| **Điều kiện** | - Nhân viên thư viện đã đăng nhập vào hệ thống và có quyền quản lý mượn sách. |
| **Luồng sự kiện chính** | 1. Nhân viên thư viện chọn tùy chọn "Quản lý Mượn Sách" trong giao diện quản trị. 2. Hệ thống hiển thị danh sách các yêu cầu mượn sách chờ xử lý. 3. Nhân viên thư viện chọn một yêu cầu mượn sách từ danh sách. 4. Hệ thống hiển thị thông tin chi tiết về yêu cầu mượn sách và sách tương ứng. 5. Nhân viên thư viện kiểm tra thông tin, số lượng sách có sẵn và tình trạng của sách. 6. Nếu thông tin hợp lệ và số lượng sách đủ, nhân viên thư viện ghi nhận việc mượn sách và cập nhật trạng thái của yêu cầu. 7. Hệ thống ghi nhận thông tin mượn sách và cập nhật số lượng sách trong kho. 8. Hệ thống thông báo cho độc giả về việc mượn sách thành công. |
| **Luồng sự kiện rẽ nhánh** | - Nếu thông tin không hợp lệ hoặc số lượng sách không đủ:  + Nhân viên thư viện từ chối yêu cầu mượn sách và cung cấp lý do từ chối.  + Hệ thống cập nhật trạng thái của yêu cầu và thông báo cho độc giả về lý do từ chối. |
| **Kết quả** | - Yêu cầu mượn sách được xử lý thành công và thông tin được ghi nhận trong hệ thống.  - Số lượng sách trong kho được cập nhật.  - Độc giả nhận được thông báo về việc mượn sách thành công hoặc từ chối. |

**1.6 Sơ đồ use case đăng ký** **mượn** **sách.**

A black and white diagram

Description automatically generated

**Sơ đồ 1.6: Sơ đồ use case đăng ký mượn sách.**

|  |  |
| --- | --- |
| **Tác nhân** | **Admin** |
| **Mô tả** | Độc giả đăng ký mượn sách từ thư viện. |
| **Điều kiện** | Sách cần mượn phải có sẵn trong kho sách của thư viện và chưa được mượn hết. |
| **Luồng sự kiện chính** | 1. Độc giả chọn sách cần mượn và cung cấp thông tin mượn sách (tên, số lượng,…) 2. Hệ thống kiểm tra thông tin độc giả và sách. 3. Nếu thông tin hợp lệ, hệ thống ghi nhận việc mượn sách và cập nhật trạng thái của sách trong kho. 4. Hệ thống thông báo cho độc giả về việc mượn sách thành công. |
| **Luồng sự kiện rẽ nhánh** | - Nếu thông tin không hợp lệ:  + Nếu sách không có sẵn trong kho hoặc đã được mượn hết, hệ thống thông báo cho độc giả và yêu cầu chọn sách khác. |
| **Kết quả** | Sách được mượn và ghi nhận trong hệ thống.  Độc giả nhận được thông báo xác nhận việc mượn sách thành công. |

**1.7 Sơ đồ use case trả sách.**

A diagram of a diagram

Description automatically generated

**Sơ đồ 1.7: Sơ đồ use case trả sách.**

|  |  |
| --- | --- |
| **Tác nhân** | **Admin** |
| **Mô tả** | Độc giả đăng ký trả sách cho thư viện. |
| **Điều kiện** | Độc giả đã mượn sách từ thư viện và cần trả sách trước khi hết hạn mượn. |
| **Luồng sự kiện chính** | 1. Độc giả đến bàn làm việc của nhân viên và thông báo muốn trả sách. 2. Nhân viên nhận sách từ độc giả và kiểm tra thông tin và tình trạng sách cần trả. 3. Nhân viên cập nhật thông tin trả sách trong hệ thống và ghi lại ngày trả. 4. Hệ thống cập nhật số lượng sách trong kho và kiểm tra tình trạng sách trả. |
| **Luồng sự kiện rẽ nhánh** | - Nếu sách trả đúng hạn:  + Nhân viên cung cấp cảm ơn cho độc giả.  - Nếu sách trả trễ hạn:  + Nhân viên áp dụng các quy định phạt trễ hạn trả sách.  + Hệ thống cập nhật thông tin về phạt trễ hạn vào tài khoản của độc giả. |
| **Kết quả** | - Sách được trả và thông tin trả sách được ghi nhận đúng trong hệ thống.  - Độc giả nhận được thông báo cảm ơn hoặc thông tin về phạt trễ hạn (nếu có). |

1.8 Sơ đồ use case tìm kiếm sách

A black and white image of a circle with a arrow pointing to the center

Description automatically generated

**Sơ đồ 1.8: Sơ đồ use case Tìm kiếm sách.**

|  |  |
| --- | --- |
| **Tác nhân** | **Admin** |
| **Mô tả** | Người dùng (độc giả) có thể tìm kiếm sách trong hệ thống thư viện dựa trên tiêu chí như tên sách, tác giả, thể loại hoặc từ khóa. |
| **Điều kiện** | Người dùng đã đăng nhập vào hệ thống thư viện. |
| **Luồng sự kiện chính** | 1. Người dùng nhập từ khóa hoặc thông tin liên quan vào ô tìm kiếm trên giao diện. 2. Hệ thống hiển thị kết quả tìm kiếm dựa trên thông tin được nhập. 3. Người dùng xem danh sách các sách phù hợp và chọn sách cần xem chi tiết. |
| **Luồng sự kiện rẽ nhánh** | Nếu không có kết quả tìm kiếm: Hệ thống hiển thị thông báo không tìm thấy kết quả và yêu cầu người dùng thử lại với tiêu chí tìm kiếm khác. |
| **Kết quả** | Kết quả tìm kiếm, người dùng có thể xem chi tiết về sách và thực hiện các hành động mượn sách. |

* 1. Cở sở dữ liệu

Mô hình cơ cở dữ liệu CDM

A diagram of a computer

Description automatically generated

Hình 1: Mô hình CDM

Mô tả các lớp:

Bảng 2.1: Mô tả lớp

|  |  |  |
| --- | --- | --- |
| STT | Tên bảng | Diễn giải |
| 1 | DocGia | Danh sách thông tin độc giả |
| 2 | Sach | Danh sách về thông tin Sách |
| 3 | TheoDoiMuonSach | Trạng thái về mượn sách |
| 4 | NhaXuatBan | Danh sách Nhà xuất bản |
| 5 | NhanVien | Danh sách thông tin nhân viên |

- Mô tả Độc giả:

Bảng 2.2: Mô tả DocGia

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Giá trị mặc định** | **Diễn giải** |
| MaDocGia | String |  | Mã độc giả |
| HoLot | String |  | Họ lót người dùng |
| Ten | String |  | Tên người dùng |
| NgaySinh | Date |  | Ngày sinh |
| Phai | String |  | Giới tính |
| DiaChi | String |  | Địa chỉ |
| DienThoai | String |  | Số điện thoại |

**- Mô tả Sách:**

Bảng 2.3: Mô tả Sach

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Giá trị mặc định** | **Diễn giải** |
| MaSach | String |  | Mã sách |
| TenSach | String |  | Tên sách |
| DonGia | Number |  | Giá tiền |
| SoQuyen | Number |  | Số lượng sách |
| NamXuatBan | Number |  | Năm xuất bản |
| MaNXB | String |  | Mã Nhà Xuất Bản |
| NguonGoc | String |  | Nguồn gốc |
| TacGia | String |  | Tác giả |

**- Mô tả Nhân Viên:**

Bảng 2.4: Mô tả NhanVien

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Giá trị mặc định** | **Diễn giải** |
| MSNV | String |  | Mã số nhân viên |
| HoTenNV | String |  | Họ và tên nhân viên |
| Password | String |  | Mật khẩu |
| ChucVu | String |  | Chức vụ |
| DiaChi | String |  | Địa chỉ |
| SoDienThoai | String |  | Số điện thoại |

- **Mô tả Nhà xuất bản:**

Bảng 2.5: Mô tả NhaXuatBan

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Giá trị mặc định** | **Diễn giải** |
| MaNXB | String |  | Mã NXB |
| TenNXB | String |  | Tên của Nhà xuất bản |
| DiaChi | String |  | Địa chỉ |

- **Mô tả Theo dõi mượn sách:**

Bảng 2.6: Mô tả TheoDoiMuonSach

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Giá trị mặc định** | **Diễn giải** |
| MaDocGia | String |  | Mã độc giả |
| MaSach | String |  | Mã sách |
| NgayMuon | Date |  | Ngày mượn sách |
| NgayTra | Date |  | Ngày trả sách |

CHƯƠNG IV: KẾT QUẢ ĐẠT ĐƯỢC

Website được xây dựng với giao diện thân thiện và đơn giản, mang tới trải nghiệm tốt cho khách hàng.

- **Website ở phía độc giả**:

**+ Tìm kiếm các sách:**

A screenshot of a computer

Description automatically generated

Hình 2: Giao diện Tìm kiếm

**+ Đăng ký mượn sách**:

A screenshot of a computer

Description automatically generated

Hình 3: Giao diện Đăng ký mượn sách

**+ Đăng ký trả sách**:

A screenshot of a computer

Description automatically generated

Hình 4: Giao diện Đăng ký trả sách

**Website của Quản trị.**

**+ Giao diện quản lý sách :**

A screenshot of a computer

Description automatically generated

Hình 5: Giao diện quản lí sách

**+ Giao diện quản lý Nhà xuất bản:**

**A screenshot of a computer

Description automatically generated**

Hình 6: Giao diện quản lí Nhà xuất bản

**A screenshot of a computer

Description automatically generated**

Hình 7: Giao diện Hiển thị danh sách NXB

**+ Giao diện quản lý Độc giả:**

**A screenshot of a computer

Description automatically generated**

Hình 8: Giao diện quản lí Độc giả

A screenshot of a computer

Description automatically generated

Hình 9: Giao diện danh sách các Độc giả

A screenshot of a computer

Description automatically generated

Hình 10: Giao diện danh sách các Độc giả đang mượn sách

**Đường link đến github mã nguồn dự án:**