

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

----- ☎ -----



ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:

**PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI
ĐỘNG NGHE SÁCH NÓI VÀ ĐỌC SÁCH ĐIỆN TỬ**

| | |
|----------------------|----------------------|
| Giảng viên hướng dẫn | : TS. Dương Trần Đức |
| Sinh viên thực hiện | : Lê Thanh Bình |
| Mã sinh viên | : B19DCCN067 |
| Lớp | : D19CNPM03 |
| Khoá | : 2019 – 2024 |
| Hệ | : Đại học chính quy |

Hà Nội - 2023

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

----- ☎ -----



ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:

**PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI
ĐỘNG NGHE SÁCH NÓI VÀ ĐỌC SÁCH ĐIỆN TỬ**

| | |
|----------------------|----------------------|
| Giảng viên hướng dẫn | : TS. Dương Trần Đức |
| Sinh viên thực hiện | : Lê Thanh Bình |
| Mã sinh viên | : B19DCCN067 |
| Lớp | : D19CNPM03 |
| Khoá | : 2019 – 2024 |
| Hệ | : Đại học chính quy |

Hà Nội - 2023

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM
(CỦA GIẢNG VIÊN PHẢN BIỆN)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:.....(bằng chữ.....)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

Hà Nội, ngày.... tháng.... năm 2023

Giảng viên phản biện
(Ký và ghi rõ họ tên)

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM
(CỦA GIẢNG VIÊN HƯỚNG DẪN)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:.....(bằng chữ.....)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

Hà Nội, ngày.... tháng.... năm 2023

Giảng viên phản biện
(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Với lòng kính trọng và biết ơn sâu sắc, lời đầu tiên em xin gửi lời cảm ơn chân thành đến các thầy cô của Học Viện Công Nghệ Bưu Chính Viễn Thông, đặc biệt là các thầy cô Khoa Công Nghệ Thông Tin 1, những người đã chia sẻ kiến thức, kinh nghiệm và hỗ trợ không ngừng nghỉ, giúp đỡ em trong suốt quá trình học tập vừa qua.

Tiếp theo, em xin gửi lời cảm ơn chân thành và biết ơn sâu sắc đến thầy TS. Dương Trần Đức đã dành thời gian để hướng dẫn, chỉ bảo và đề xuất hướng giải quyết vấn đề khi em gặp khó khăn trong suốt quá trình thực hiện đồ án này.

Cuối cùng, không thể thiếu lòng biết ơn tới gia đình, bạn bè và những người đã luôn động viên, hỗ trợ và cho em niềm tin để hoàn thành đồ án một cách tốt nhất.

Dù đã cố gắng hết sức, em nhận thấy rằng đồ án vẫn còn một số hạn chế và điều chưa hoàn hảo. Quãng thời gian này không đơn giản chỉ là việc nghiên cứu và thực hiện mà còn là những bài học quý báu về sự kiên trì, sự sáng tạo và khả năng vượt qua khó khăn. Sự hỗ trợ, định hướng và ý kiến đóng góp từ quý thầy cô, các bạn, và mọi người sẽ giúp em tiếp tục phát triển và hoàn thiện hơn nữa trong tương lai.

Với tất cả sự kính trọng, em xin chân thành cảm ơn!

Hà Nội, tháng 12 năm 2023

Sinh viên thực hiện

Lê Thanh Bình

MỤC LỤC

| | |
|--|-----------|
| DANH MỤC BẢNG..... | 11 |
| PHẦN MỞ ĐẦU..... | 12 |
| 1. PHÁT BIỂU BÀI TOÁN..... | 12 |
| 2. PHẠM VI NGHIÊN CỨU..... | 13 |
| 3. LÝ DO CHỌN ĐỀ TÀI..... | 13 |
| 4. MỤC TIÊU ĐỒ ÁN..... | 14 |
| 5. CẦU TRÚC ĐỒ ÁN..... | 14 |
| CHƯƠNG 1: CƠ SỞ LÝ THUYẾT..... | 15 |
| 1. GIỚI THIỆU..... | 15 |
| 1.1. Định nghĩa và tính chất cơ bản..... | 15 |
| 1.2. Ứng dụng và phát triển..... | 15 |
| 1.3. Phân loại và các kiểu định dạng..... | 15 |
| 2. CÁC CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG..... | 16 |
| 2.1. React Native..... | 16 |
| 2.2. NestJS..... | 18 |
| 3. CƠ SỞ LÝ THUYẾT HỆ THỐNG KHUYẾN NGHỊ..... | 23 |
| 3.1. Giới thiệu..... | 23 |
| 3.2. Hệ thống khuyến nghị – Recommender Systems..... | 23 |
| 3.3. Tìm hiểu các phương pháp, kỹ thuật trong Recommender Systems..... | 27 |
| CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG..... | 31 |
| 1. YÊU CẦU HỆ THỐNG..... | 31 |
| 1.1. Sơ đồ tổng quan usecase của hệ thống..... | 31 |
| 1.2. Một số chức năng chính của hệ thống..... | 35 |
| 2. KIẾN TRÚC HỆ THỐNG..... | 47 |
| 2.1. Sơ đồ mô tả kiến trúc hệ thống..... | 47 |
| 2.2. Trích lớp thực thể..... | 48 |
| 2.3. Biểu đồ lớp thực thể pha phân tích..... | 49 |
| 2.4. Biểu đồ lớp thực thể pha thiết kế..... | 50 |
| 2.5. Thiết kế Database..... | 51 |

| | |
|---|-----------|
| 2.6. Activity Diagram..... | 52 |
| CHƯƠNG 3: PHÁT TRIỂN ỨNG DỤNG..... | 57 |
| 1. PHÁT TRIỂN CHỨC NĂNG ĐỌC SÁCH ĐIỆN TỬ..... | 57 |
| 2. PHÁT TRIỂN CHỨC NĂNG NGHE SÁCH TRỰC TUYẾN..... | 61 |
| 3. PHÁT TRIỂN CHỨC NĂNG ĐĂNG KÝ, ĐĂNG NHẬP..... | 64 |
| 3.1. Chức năng Đăng nhập..... | 64 |
| 3.2. Chức năng Đăng ký..... | 66 |
| 3.3. Chức năng Đăng nhập với tài khoản thứ 3..... | 68 |
| 4. PHÁT TRIỂN CHỨC NĂNG CHỈNH SỬA THÔNG TIN TÀI KHOẢN.. | 72 |
| 5. MỘT SỐ GIAO DIỆN CHỨC NĂNG KHÁC CỦA ỨNG DỤNG..... | 74 |
| 6. HỆ THỐNG KHUYẾN NGHỊ..... | 76 |
| 6.1. Các thư viện sử dụng và xử lý dữ liệu:..... | 76 |
| 6.2. Most popular items..... | 77 |
| 6.3. Item based..... | 79 |
| 6.4. User based..... | 80 |
| 6.5. Content based..... | 82 |
| TỔNG KẾT..... | 84 |
| 1. KẾT QUẢ ĐẠT ĐƯỢC..... | 84 |
| 2. NHỮNG HẠN CHẾ..... | 84 |
| 3. HƯỚNG PHÁT TRIỂN TIẾP THEO..... | 84 |
| TÀI LIỆU THAM KHẢO..... | 85 |

DANH MỤC ẢNH

| | |
|---|----|
| Hình 1.1: Sơ đồ kiến trúc của NestJS..... | 18 |
| Hình 1.2: ACID and BASE transaction..... | 22 |
| Hình 1.3: Ví dụ về Quá tải thông tin (Information Overload)..... | 24 |
| Hình 1.4: Users, Items and Ratings..... | 25 |
| Hình 1.5: Ví dụ ma trận xếp hạng user-item trong đề xuất phim..... | 26 |
| Hình 1.6: Các phương pháp, kỹ thuật trong Recommender System..... | 27 |
| Hình 1.7: Minh họa hệ thống khuyến nghị dựa trên Collaborative Filtering..... | 28 |
| Hình 1.8: Minh họa hệ thống khuyến nghị Content-based..... | 29 |
| Hình 1.9: Minh họa model-based CB-RS..... | 30 |
| Hình 2.1: Usecase tổng quan hệ thống..... | 34 |
| Hình 2.2: Biểu đồ usecase chức năng Đăng ký..... | 35 |
| Hình 2.3: Biểu đồ usecase chức năng Đăng nhập với các bên thứ 3..... | 36 |
| Hình 2.4: Biểu đồ use case chức năng Đăng nhập..... | 38 |
| Hình 2.5: Biểu đồ usecase chức năng Đọc sách điện tử..... | 39 |
| Hình 2.6: Biểu đồ usecase chức năng Nghe sách trực tuyến..... | 40 |
| Hình 2.7: Biểu đồ usecase chức năng Chỉnh sửa thông tin tài khoản..... | 42 |
| Hình 2.8: Biểu đồ usecase chức năng Đánh dấu sách..... | 43 |
| Hình 2.9: Biểu đồ usecase chức năng Thêm sách yêu thích..... | 45 |
| Bảng 10: Kịch bản chức năng Thêm sách yêu thích..... | 45 |
| Hình 2.10: Biểu đồ usecase chức năng Lịch sử truy cập..... | 46 |
| Hình 2.11: Biểu đồ lớp thực thể pha phân tích..... | 49 |
| Hình 2.12: Biểu đồ lớp thực thể pha thiết kế..... | 50 |
| Hình 2.13: Cơ sở dữ liệu..... | 51 |
| Hình 2.14: Biểu đồ hoạt động module Đăng nhập..... | 52 |

| | |
|--|----|
| Hình 2.15: Biểu đồ hoạt động module Đăng ký..... | 52 |
| Hình 2.16: Biểu đồ hoạt động module Đăng nhập tài khoản với các bên thứ 3..... | 53 |
| Hình 2.17: Biểu đồ hoạt động module Đọc sách điện tử..... | 53 |
| Hình 2.18: Biểu đồ hoạt động module Nghe sách trực tuyến..... | 54 |
| Hình 2.19: Biểu đồ hoạt động module Đánh dấu sách..... | 54 |
| Hình 2.20: Biểu đồ hoạt động module Chỉnh sửa thông tin tài khoản..... | 55 |
| Hình 2.21: Biểu đồ hoạt động module Thêm sách yêu thích..... | 56 |
| Hình 2.22: Biểu đồ hoạt động module Truy cập lịch sử sách..... | 56 |
| Hình 3.1: Giao diện Đọc sách điện tử (1)..... | 57 |
| Hình 3.2: Giao diện Đọc sách điện tử (2)..... | 57 |
| Hình 3.3: Giao diện Đọc sách điện tử (3)..... | 57 |
| Hình 3.4: Giao diện Đọc sách điện tử (4)..... | 57 |
| Hình 3.4: Giao diện Đọc sách điện tử (5)..... | 58 |
| Hình 3.5: Giao diện Đọc sách điện tử (6)..... | 58 |
| Hình 3.6: Ví dụ kết quả trả về sau lấy thông tin sách điện tử thành công..... | 60 |
| Hình 3.7: Giao diện Nghe sách trực tuyến (1)..... | 61 |
| Hình 3.8: Giao diện Nghe sách trực tuyến (2)..... | 61 |
| Hình 3.9: Ví dụ kết quả trả về sau lấy thông tin sách điện tử thành công..... | 63 |
| Hình 3.10: Giao diện Đăng nhập (1)..... | 64 |
| Hình 3.11: Giao diện Đăng nhập (2)..... | 64 |
| Hình 3.12: Ví dụ kết quả trả về sau khi đăng nhập thành công..... | 65 |
| Hình 3.13: Giao diện Đăng ký (1)..... | 66 |
| Hình 3.14: Giao diện Đăng ký (2)..... | 66 |
| Hình 3.15: Ví dụ kết quả trả về sau khi đăng ký thành công..... | 67 |
| Hình 3.16: Firebase console (1)..... | 68 |

| | |
|--|----|
| Hình 3.17: Firebase console (2)..... | 68 |
| Hình 3.18: Giao diện Đăng nhập..... | 69 |
| Hình 3.19: Giao diện đăng nhập bằng tài khoản Google..... | 69 |
| Hình 3.20: Ví dụ kết quả trả về sau khi đăng nhập thành công..... | 71 |
| Hình 3.21: Giao diện Thông tin tài khoản..... | 72 |
| Hình 3.22: Giao diện Chỉnh sửa thông tin..... | 72 |
| Hình 3.23: Giao diện Thư viện (1)..... | 74 |
| Hình 3.24: Giao diện Thư viện (2)..... | 74 |
| Hình 3.25: Giao diện Thông tin sách (1)..... | 74 |
| Hình 3.26: Giao diện Thông tin sách (2)..... | 74 |
| Hình 3.27: Giao diện Tìm kiếm sách (1)..... | 75 |
| Hình 3.28: Giao diện Tìm kiếm sách (2)..... | 75 |
| Hình 3.29: Giao diện Lịch sử truy cập..... | 75 |
| Hình 3.30: Giao diện Sách yêu thích..... | 75 |
| Hình 3.32: Cấu trúc dữ liệu từ file csv mẫu..... | 77 |
| Hình 3.33: Kết quả 10 sách phổ biến nhất..... | 78 |
| Hình 3.34: Kết quả những sách tương tự với tên sách cụ thể..... | 80 |
| Hình 3.35: Kết quả những sách tương tự với sách yêu thích của người dùng khác..... | 82 |
| Hình 3.36: Kết quả khuyến nghị dựa trên nội dung sách..... | 83 |

DANH MỤC BẢNG

| | |
|---|-----------|
| Bảng 1: Bảng mô tả các tác nhân của hệ thống..... | 21 |
| Bảng 2: Bảng mô tả các tác nhân của hệ thống..... | 31 |
| Bảng 3: Bảng danh sách các chức năng chính..... | 31 |

| | |
|---|----|
| Bảng 4: Kịch bản chức năng Đăng ký..... | 35 |
| Bảng 5: Kịch bản chức năng Đăng ký với các bên thứ 3..... | 37 |
| Bảng 6: Kịch bản chức năng Đăng nhập..... | 38 |
| Bảng 7: Kịch bản chức năng Đọc sách điện tử..... | 39 |
| Bảng 9: Kịch bản chức năng Chỉnh sửa thông tin tài khoản..... | 42 |
| Bảng 10: Kịch bản chức năng Đánh dấu sách..... | 43 |
| Bảng 12: Kịch bản chức năng Lịch sử truy cập..... | 46 |

PHẦN MỞ ĐẦU

1. PHÁT BIỂU BÀI TOÁN

Công nghệ di động ngày nay đã thay đổi cách chúng ta tiếp cận thông tin và kiến thức, đặc biệt là trong việc tiêu thụ nội dung văn hóa và giáo dục. Sự tiện lợi của thiết bị di động đã tạo ra một sự đổi mới về cách chúng ta tiếp cận sách, từ việc dịch chuyển từ sách giấy sang sách điện tử cho đến trải nghiệm độc đáo của sách nói thông qua các ứng dụng di động.

Trong xã hội hiện đại, nhu cầu về việc sử dụng ứng dụng di động để nghe sách đang ngày càng tăng cao. Cuộc sống hối hả và lối sống di chuyển thường xuyên đã tạo ra nhu cầu cấp thiết cho việc tiếp tục trải nghiệm văn hóa đọc sách mà không đòi hỏi quá nhiều thời gian tập trung. Điều này thúc đẩy sự phát triển của ứng dụng sách nói và sách điện tử trên thiết bị di động, nơi mà người dùng không chỉ đọc sách điện tử mà còn được tận hưởng trải nghiệm âm thanh sống động và hấp dẫn từ sách nói. Một số kênh, trang mạng trực tuyến có lượng độc giả lớn ở Việt Nam hiện nay như Kho sách nói, Cafe sách, Hẻm Radio, Góc sách, Queen Voice... còn đầu tư mạnh để tạo dựng các tác phẩm đọc công phu, sống động như thật bằng việc phân vai cho các nhân vật trong sách/truyện để tạo nên sức hút, lôi kéo độc giả.

Tuy nhiên, sự bùng nổ của thông tin trong thời đại số cũng đồng nghĩa với việc người dùng phải đối mặt với khó khăn trong việc lựa chọn nội dung phù hợp. Số lượng sách và thông tin ngày càng tăng lên, nhưng thời gian có hạn của mỗi người khiến việc tìm kiếm nội dung phù hợp trở nên một thách thức đáng kể. Điều này làm nảy sinh nhu cầu về các công cụ hỗ trợ, như hệ thống khuyến nghị trong các ứng dụng di động, giúp người dùng lọc và nhận khuyến nghị tùy chỉnh dựa trên sở thích cá nhân. Điều này không chỉ giúp giảm bớt áp lực mà còn tiết kiệm thời gian và tăng trải nghiệm người dùng khi tìm kiếm nội dung phù hợp.

Với tầm quan trọng của vấn đề này, việc nghiên cứu và phát triển hệ thống sách nói và sách điện tử trên thiết bị di động trở nên cần thiết hơn bao giờ hết. Điều này không chỉ là để đáp ứng nhu cầu đa dạng của người đọc mà còn để tạo ra những trải nghiệm đọc sách mới lạ, sáng tạo hơn. Việc tận dụng công nghệ và phát triển ứng dụng không chỉ làm tăng cường trải nghiệm người dùng mà còn mở ra cơ hội cho việc khám phá thế giới văn hóa và kiến thức một cách linh hoạt và đa dạng hơn.

2. PHẠM VI NGHIÊN CỨU

Phạm vi nghiên cứu sẽ tập trung vào việc phát triển hệ thống ứng dụng về sách được tích hợp 3 phần chính bao gồm: Cửa hàng thương mại điện tử về sách, Thư viện quản lý nghe sách nói và đọc sách điện tử và Quản lý các sản phẩm. Cụ thể như sau các phần nghiên cứu sau:

- Em Lê Thanh Bình nghiên cứu về đề tài: Phát triển ứng dụng nghe sách nói và đọc sách điện tử kết hợp hệ thống khuyến nghị về sách trên thiết bị di động.
- Bạn Đinh Như Cương nghiên cứu về đề tài: Phát triển ứng dụng thương mại điện tử cho sản phẩm sách nói và sách điện tử.
- Bạn Hoàng Anh Đức nghiên cứu về đề tài: Phát triển ứng dụng Web quản lý sách nói và sách điện tử. Đồng bộ dữ liệu giữa sách nói và sách điện tử.

3. LÝ DO CHỌN ĐỀ TÀI

Sau đây là những lý do em chọn nghiên cứu về đề tài: Phát triển ứng dụng nghe sách nói và đọc sách điện tử kết hợp hệ thống khuyến nghị về sách trên thiết bị di động.

- **Ứng dụng thực tiễn:** Đề tài này không chỉ tập trung vào một lĩnh vực mà còn liên quan mật thiết đến nhu cầu và lối sống của nhiều người, tạo ra tiềm năng thị trường rộng lớn và áp dụng linh hoạt trên nhiều nền tảng.
- **Tìm hiểu các phương pháp giải quyết vấn đề quá tải thông tin:** Hệ thống khuyến nghị giúp người dùng dễ dàng tìm kiếm sách phù hợp với sở thích cá nhân, giảm áp lực lựa chọn trong bối cảnh thông tin ngày càng tăng.
- **Khả năng học hỏi và phát triển kỹ năng:** Việc thực hiện đồ án này sẽ giúp em phát triển kỹ năng lập trình, phân tích dữ liệu, thiết kế hệ thống và tối ưu hóa trải nghiệm người dùng trên thiết bị di động.
- **Mở rộng khả năng tiếp cận thông tin và giáo dục:** Việc phát triển ứng dụng trên thiết bị di động mang lại khả năng tiếp cận sách nhanh chóng và thuận tiện cho mọi người, không phụ thuộc vào vị trí địa lý hay thiết bị đọc cụ thể.
- **Sự phù hợp với thị trường công nghệ hiện nay:** Đề tài này liên quan mật thiết đến các công nghệ đang phát triển mạnh mẽ, mở ra cơ hội việc làm và sự phát triển nghề nghiệp trong lĩnh vực Công nghệ thông tin.
- **Tác động đến cộng đồng:** Tạo ra một sản phẩm thể mang lại lợi ích và tác động tích cực đến cộng đồng người dùng, cải thiện cách họ tiếp cận và tận hưởng sách.
- **Thúc đẩy sự phát triển và sáng tạo:** Bằng cách kết hợp giữa nội dung văn bản và âm thanh, ứng dụng này không chỉ cung cấp thông tin mà còn tạo ra một trải nghiệm đọc sách độc đáo, khuyến khích sự sáng tạo và tư duy sáng tạo.

4. MỤC TIÊU ĐỒ ÁN

- Xây dựng một ứng dụng thực tế, tích hợp cả khả năng nghe sách nói và đọc sách điện tử, mang lại trải nghiệm đọc sách đa dạng và phong phú cho người sử dụng cùng với nhiều nền tảng khác nhau trên thiết bị di động như Android, IOS thông qua việc sử dụng React Native.
- Thiết kế hệ thống và cơ sở dữ liệu cho ứng dụng sách đã được xác định để tương thích với các chức năng. Việc xác định cấu trúc dữ liệu và cách tổ chức hệ thống đã được tối ưu hóa, đảm bảo khả năng mở rộng và linh hoạt trong việc đáp ứng nhu cầu người dùng cũng như quản lý dữ liệu một cách hiệu quả và an toàn.
- Xây dựng hệ thống để kết nối và xử lý các yêu cầu, cũng như để lưu trữ thông tin từ ứng dụng và người dùng sử dụng các công nghệ đang được phổ biến hiện nay như NestJS và Database PostgreSQL.
- Tìm hiểu về các phương pháp khuyến nghị (recommendation system) trong hệ thống sách nói và sách điện tử. Nghiên cứu này sẽ đề xuất và đánh giá các phương pháp khuyến nghị giúp cá nhân hóa trải nghiệm đọc sách dựa trên hành vi của người dùng với 2 phương pháp phổ biến sau:
 - o Phương pháp Collaborative Filtering Recommendation (Lọc cộng tác).
 - o Phương pháp Content-based Recommendation (Dựa trên nội dung).

5. CẤU TRÚC ĐỒ ÁN

Nội dung của đồ án được xây dựng thành các chương như sau:

Chương 1. Cơ sở lý thuyết

Chương 2. Phân tích thiết kế hệ thống

Chương 3. Phát triển ứng dụng

Kết luận và hướng phát triển tiếp theo

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1. GIỚI THIỆU

1.1. Định nghĩa và tính chất cơ bản

- Theo Bách khoa toàn thư mở Wikipedia [1] [2]:

○ Sách nói là một hình thức truyền tải thông tin nằm trong một cuốn sách. Giống như với sách in trên giấy hay sách điện tử, thông tin được giữ nguyên và thể hiện một cách chân thực. Người dùng sách nói sẽ nghe một người đọc sách thành tiếng thông qua các phương tiện truyền thông hay các ứng dụng thông minh.

○ Sách điện tử (e-book hay eBook), là một quyển sách được xuất bản và phát hành cho các thiết bị kỹ thuật số, bao gồm văn bản, hình ảnh hoặc cả hai, có thể đọc được trên màn hình phẳng của máy tính hoặc các thiết bị điện tử khác.

- Tính chất chung của cả hai loại sách này đó là sự linh hoạt, tiện lợi và khả năng tương tác với người dùng.

- Trên cơ sở sự phổ biến của thiết bị di động và internet, sách nói và sách điện tử đã trở thành phương tiện học tập và giải trí được ưa chuộng. Việc tiếp cận nhanh chóng, khả năng tùy chỉnh theo sở thích cá nhân cùng với khả năng tương tác cao làm cho cả hai loại sách này trở nên hấp dẫn và tiềm năng cho sự phát triển cá nhân và giáo dục.

1.2. Ứng dụng và phát triển

- Sự phát triển của ứng dụng sách nói và sách điện tử trên thiết bị di động đang mở ra một thế giới mới của tiện ích và tiếp cận kiến thức. Với giao diện thân thiện trên di động, việc đọc và nghe sách trở nên thuận tiện hơn bao giờ hết.

- Khả năng tương tác, nhận xét, và chia sẻ nội dung trên các nền tảng mạng xã hội từ thiết bị di động đã làm tăng cường trải nghiệm người dùng đáng kể.

1.3. Phân loại và các kiểu định dạng

- **Sách nói (Audiobooks):**

○ **Audiobook:** Đây là định dạng phổ biến nhất của sách nói, chúng thường là phiên bản audio của sách in. Các file này thường có định dạng MP3 hoặc M4B (được sử dụng cho các audiobook có nhiều chương, cho phép người nghe dễ dàng chuyển đổi giữa các chương).

○ **Podcast:** Một loại sách nói có thể bao gồm các chủ đề đa dạng, từ chính trị đến giáo dục, và thường được phân phối thông qua các nền tảng podcast như Apple

Podcasts, Spotify, hay SoundCloud. Định dạng này có thể là **MP3, AAC, ID3**.

- **Đọc trực tiếp (Live readings):** Đây là các phiên bản ghi âm của các buổi đọc trực tiếp các tác phẩm văn học, thường được chia sẻ trên các nền tảng trực tuyến như YouTube hoặc các diễn đàn âm thanh.

- **Sách điện tử (eBooks):**

- **eBook:** Đây là dạng phổ biến nhất của sách điện tử, thường là file dưới định dạng EPUB, PDF, MOBI, AZW, và KF8. Mỗi định dạng này có tính linh hoạt khác nhau trong việc hiển thị nội dung trên các thiết bị khác nhau (như Kindle, iPad, hoặc máy tính).

- **PDF:** Định dạng PDF cung cấp tính năng tương tác, cho phép người đọc tương tác với nội dung như tìm kiếm, ghi chú, hoặc thậm chí làm việc với các phần mềm bôi sáng, ghi chú trực tiếp trong file PDF.

- **Đa phương tiện:** Các sách điện tử có thể chứa đựng nhiều phương tiện khác nhau bên cạnh văn bản, bao gồm hình ảnh, video, âm thanh, và các tính năng tương tác khác.

2. CÁC CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG

2.1. React Native

2.1.1. Giới thiệu

- Theo trang chủ của React Native [4]: React Native là các đoạn code đã được viết sẵn (framework) do công ty công nghệ Facebook phát triển. Các lập trình viên React Native là người sử dụng những framework này để phát triển nên các hệ thống, nền tảng ứng dụng trên các hệ điều hành như IOS và Android. Ngôn ngữ lập trình được sử dụng nhiều nhất là Javascript.

- Sự ra đời của React Native đã giải quyết được bài toán về hiệu năng và sự phức tạp khi trước đó người ta phải dùng nhiều loại ngôn ngữ native cho mỗi nền tảng di động. Chính vì thế lập trình React Native sẽ giúp tiết kiệm được phần lớn thời gian và công sức khi thiết kế và xây dựng nên một ứng dụng đa nền tảng. Javascript phù hợp với rất nhiều nền tảng khác nhau.

2.1.2. Đặc điểm của React Native

- React Native hoạt động trên ba thread:

- Chuỗi giao diện người dùng (UI Thread): Chuỗi ứng dụng chính có quyền truy cập vào giao diện người dùng của ứng dụng.

- Shadow Thread: Thread nền sử dụng thư viện React để tính toán bố cục của ứng

dụng.

- Chuỗi JavaScript (JavaScript Thread): Chuỗi thực thi có chứa mã React (JavaScript).
- React Native sẽ hoạt động bằng cách tích hợp Main Thread và JS Thread cho một ứng dụng mobile. Main Thread sẽ có vai trò cập nhật giao diện người dùng (UI) và xử lý tương tác của người dùng. JS Thread là nơi thực thi và xử lý mã Javascript.

○ Main Thread và JS Thread hoạt động độc lập vì vậy cần sử dụng một cầu nối (Bridge) để giúp 2 thread này tương tác với nhau. Cầu nối cho phép chúng chuyển đổi dữ liệu giữa thread này và thread kia, giúp chúng giao tiếp mà không phải phụ thuộc vào nhau.

2.1.3. Lý do lựa chọn React Native

- **Khả năng tái sử dụng:** Ưu điểm lớn nhất của React Native là các nhà phát triển không cần phải tạo các mã riêng biệt cho các nền tảng khác nhau (Android và iOS). Trên thực tế, khoảng 90% mã có thể được sử dụng lại giữa hai nền tảng, giúp tăng tốc độ phát triển và mang đến hiệu quả đáng kể. Việc tái sử dụng code cũng rút ngắn thời gian đưa sản phẩm ra thị trường và cần ít nỗ lực bảo trì hơn.

- **Live reload:** Tính năng live reload của React Native cho phép xem và làm việc với các thay đổi ngay lập tức. Bạn có thể thực hiện các bản sửa lỗi trong code ngay khi ứng dụng đang tải, tính năng live reload sẽ tự động refresh ứng dụng để bạn có thể thấy những thay đổi của đoạn code.

- **Cải thiện giao diện và trải nghiệm người dùng:** React Native sử dụng thư viện React JavaScript để xây dựng các giao diện app cực nhanh và đáp ứng nhiều nhu cầu. Ngoài ra, framework này có khả năng kết xuất tuyệt vời và sử dụng cách tiếp cận dựa trên thành phần giúp dễ dàng tạo ra các ứng dụng với giao diện người dùng đơn giản, hoặc phức tạp.

- **Tối ưu hoá hiệu suất:** Vì có thể sử dụng lại code trong React Native, nó giúp tiết kiệm chi phí phát triển lên đến 40%. Ngoài ra, React Native hỗ trợ phát triển trên 2 nền tảng (Android và IOS). Trên hết, có rất nhiều thành phần đã được xây dựng sẵn trong React Native giúp đẩy nhanh quá trình phát triển.

- **Plugin của bên thứ ba:** Việc xây dựng một ứng dụng từ đầu có thể rất tốn kém, vì vậy React Native cung cấp một số tùy chọn plugin của bên thứ ba bao gồm các native mô-đun và JavaScript-based để các nhà phát triển sử dụng. Các plugin của bên thứ ba giúp nâng cao hiệu suất của ứng dụng và tiết kiệm thời gian phát triển app.

- **Cộng đồng phát triển:** Việc có một công ty mẹ như Facebook giúp React Native nhận được sự quan tâm của nhiều chuyên gia, kỹ sư và những người đam mê công nghệ trên toàn thế giới. Đó là lý do React Native có một cộng đồng phát triển mạnh mẽ, có thể dễ dàng tìm kiếm sự hỗ trợ cũng như chia sẻ kiến thức của mình. Ngoài ra, React Native là một framework mã nguồn mở nên mọi người đều có thể đóng góp để giúp framework này phát triển hơn nữa trong tương lai.

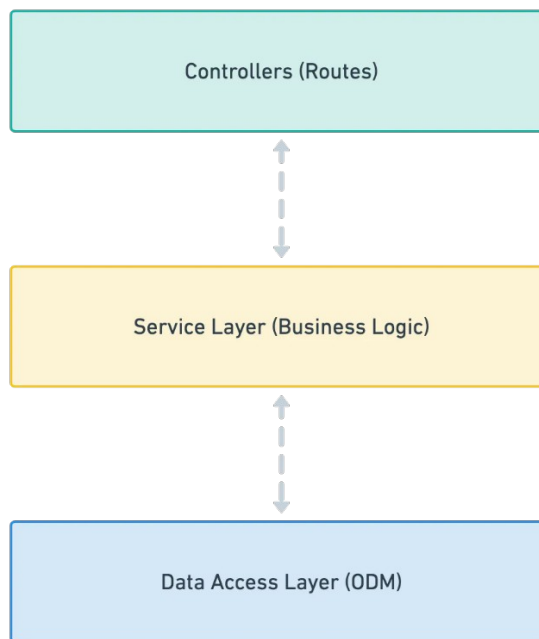
2.2. NestJS

2.2.1. Giới thiệu

- Theo trang chủ của NestJS [5]: NestJS là một framework mã nguồn mở để phát triển ứng dụng server-side (backend applications) bằng ngôn ngữ TypeScript hoặc JavaScript. Nó được xây dựng trên cơ sở của Node.js và sử dụng các khái niệm từ TypeScript để tạo ra một môi trường phát triển hiện đại và mạnh mẽ cho việc xây dựng các ứng dụng web và API.

- Mục tiêu chính của NestJS là cung cấp một cấu trúc ứng dụng rõ ràng và dễ quản lý, giúp tăng tính bảo trì và sự tổ chức trong mã nguồn. Để đạt được điều này, NestJS triển khai mô hình kiến trúc lõi (core architecture) dựa trên các nguyên tắc của Angular, đặc biệt là sử dụng Dependency Injection (DI) và Modules (Các module).

2.2.2. Đặc điểm của NestJS



Hình 1.1: Sơ đồ kiến trúc của NestJS

- Cấu trúc của NestJS được xây dựng dựa trên mô hình kiến trúc lõi (core architecture) giúp tạo ra một ứng dụng server-side (backend application) rõ ràng, dễ quản lý và dễ mở rộng. Cấu trúc NestJS thường được tổ chức thành các phần chính sau:

- **Module (Các module):** Module là một phần cơ bản trong cấu trúc NestJS. Mỗi ứng dụng NestJS bao gồm ít nhất một module gốc (root module) và có thể có nhiều

module con. Module là nơi tổ chức các thành phần của ứng dụng như Controllers, Providers và các thành phần khác. Mỗi module đại diện cho một phần chức năng cụ thể của ứng dụng.

- **Controller (Bộ điều khiển):** Controllers là thành phần chịu trách nhiệm xử lý các yêu cầu HTTPS từ phía client và trả về kết quả tương ứng. Controllers là nơi xử lý các request và trả về các response. Các phương thức của controller được chú thích (decorated) bằng các decorator như `@Get()`, `@Post()`, `@Put()`... để chỉ định các route và phương thức HTTPS tương ứng.

- **Provider (Các nhà cung cấp):** Providers là thành phần chịu trách nhiệm cung cấp các dịch vụ cho ứng dụng. Đây có thể là các service, repository, logger... Providers sử dụng dependency injection để chèn vào các thành phần khác và có thể được sử dụng bởi các controllers hoặc các providers khác.

- **Middleware (Trung gian):** Middleware là các hàm xử lý mà NestJS sử dụng để xử lý các yêu cầu HTTP trước khi chúng đến các route xử lý chính. Middleware có thể được sử dụng để thực hiện các thao tác chung như xác thực, ghi log, xử lý lỗi...

- **Filter (Bộ lọc):** Filters được sử dụng để xử lý các exception (ngoại lệ) xảy ra trong ứng dụng. Filters cho phép bạn xử lý và thay đổi response trước khi gửi về client khi có exception xảy ra.

- **Guard (Bảo vệ):** Guards được sử dụng để kiểm tra xem một yêu cầu có thể được xử lý hoặc không. Guards cho phép bạn thực hiện các kiểm tra xác thực hoặc kiểm tra quyền trước khi xử lý một yêu cầu.

- **Interceptor (Bộ chặn):** Interceptors là các hàm xử lý mà NestJS sử dụng để chặn và thay đổi response trước khi nó được gửi về client. Interceptors có thể được sử dụng để thực hiện các thao tác chung trên response trước khi nó đi ra ngoài.

- **Exception (Ngoại lệ):** Exception handling (xử lý ngoại lệ) là một phần quan trọng của cấu trúc NestJS. Exception handling cho phép bạn xử lý các exception xảy ra trong ứng dụng và trả về các thông báo lỗi thích hợp cho client.

2.2.3. Lý do lựa chọn NestJS

- Node.js và TypeScript: NestJS được xây dựng trên Node.js và hỗ trợ TypeScript một cách mạnh mẽ. TypeScript giúp dễ dàng quản lý mã nguồn, giúp tăng tính ổn định và dễ bảo trì của ứng dụng.

- Cấu trúc module và dependency injection: Như đã trình bày ở trên về đặc điểm của NestJS, sử dụng cấu trúc module giúp tổ chức code logic một cách rõ ràng, dễ dàng

mở rộng và tái sử dụng. Dependency injection cũng là một điểm mạnh, giúp quản lý các thành phần của ứng dụng một cách hiệu quả.

- RESTful API và WebSockets: NestJS cung cấp các công cụ mạnh mẽ cho việc xây dựng RESTful APIs và hỗ trợ WebSockets, rất hữu ích trong việc tương tác thời gian thực với thiết bị di động.
- Bảo mật: Thư viện bảo mật của NestJS giúp bảo vệ ứng dụng khỏi các lỗ hổng bảo mật thông thường, một yếu tố quan trọng đối với ứng dụng có thể chứa thông tin cá nhân của người dùng.
- Bảo mật: Thư viện bảo mật của NestJS Hỗ trợ testing: NestJS hỗ trợ các công cụ và phương pháp testing linh hoạt, giúp đảm bảo tính ổn định và chất lượng của ứng dụng giúp bảo vệ ứng dụng khỏi các lỗ hổng bảo mật thông thường, một yếu tố quan trọng đối với ứng dụng có thể chứa thông tin cá nhân của người dùng.

2.2.4. Giới thiệu

- Theo trang chủ PostgreSQL [3]: PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ và đối tượng (object-relational database management system) miễn phí và nguồn mở (RDBMS) tiên tiến nhất hiện nay. khả năng mở rộng cao và tuân thủ các tiêu chuẩn kỹ thuật. Nó được thiết kế để xử lý một loạt các khối lượng công việc lớn, từ các máy tính cá nhân đến kho dữ liệu hoặc dịch vụ Web có nhiều người dùng đồng thời.
- PostgreSQL linh động có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.
- PostgreSQL là hệ thống quản lý cơ sở dữ liệu đầu tiên triển khai tính năng kiểm soát đồng thời nhiều phiên bản (MVCC).
- PostgreSQL là một phần mềm mã nguồn mở miễn phí bởi vậy PostgreSQL có thể được dùng, sửa đổi và phổ biến bởi bất kỳ ai cho bất kỳ mục đích nào.

2.2.5. Đặc điểm của PostgreSQL

- **Cung cấp nhiều kiểu dữ liệu:** PostgreSQL cung cấp đa dạng kiểu dữ liệu như nguyên hàm (các nguyên số, boolean, số, chuỗi), cấu trúc (UUID, phạm vi, ...), hình học, document, ...
- **Bảo đảm toàn vẹn dữ liệu:** Dữ liệu trong PostgreSQL đảm bảo tính toàn vẹn bằng cách ràng buộc loại từ, Primary Keys, Foreign Keys, khóa khuyến nghị, khóa hàm số, ...
- **Tính năng thiết lập linh hoạt:** Người dùng được thiết lập danh mục từ đơn giản đến phức tạp, tối ưu hóa tốc độ truy cập, hỗ trợ thống kê trên nhiều cột, ...

- **Chức năng bảo mật:** PostgreSQL hỗ trợ xây dựng hàng rào bảo mật, xác thực mạnh (SCRAM-SHA-256, SSPI, LDAP, GSSAPI, Certificate, ...), hệ thống kiểm soát truy cập kỹ càng, bảo mật cấp độ cột – hàng, kết nối cơ sở dữ liệu hoặc luồng khác với giao diện SQL chuẩn.

2.2.6. SQL và NoSQL

- PostgreSQL là một hệ quản trị cơ sở dữ liệu (Hệ quản trị cơ sở dữ liệu quan hệ - RDBMS), chủ yếu dựa trên mô hình quan hệ và sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để tương tác với dữ liệu. Tuy nhiên, PostgreSQL cũng hỗ trợ một số tính năng NoSQL, như hỗ trợ JSON, các loại dữ liệu phức tạp, và các tính năng linh hoạt giúp xử lý dữ liệu đa dạng một cách hiệu quả. Do đó, nó có thể kết hợp giữa tính linh hoạt của NoSQL và sức mạnh của SQL, tạo ra một sự lựa chọn đa năng cho các ứng dụng khác nhau.

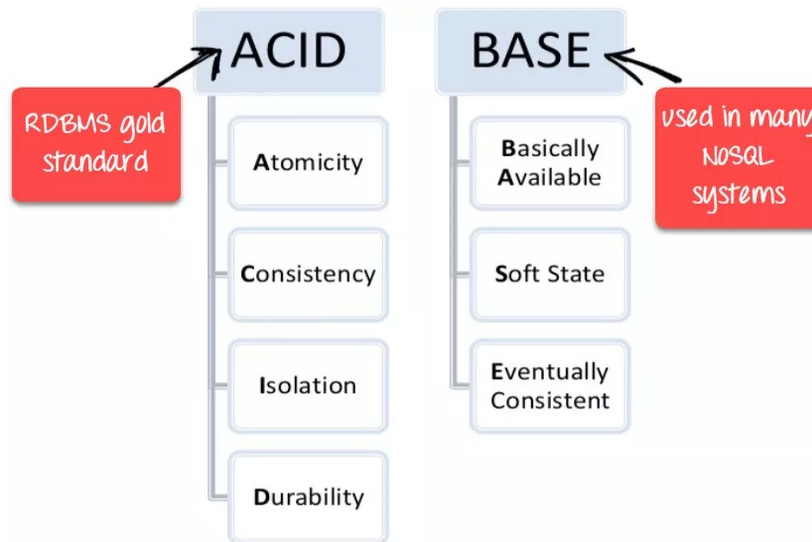
| | SQL | NoSQL |
|----------------|---|--|
| Định nghĩa | Cơ sở dữ liệu SQL chủ yếu được gọi là RDBMS hoặc Cơ sở dữ liệu quan hệ | Cơ sở dữ liệu NoSQL chủ yếu được gọi là cơ sở dữ liệu không liên quan hoặc phân tán |
| Thiết kế | RDBMS truyền thống sử dụng cú pháp và truy vấn SQL để phân tích và lấy dữ liệu để có thêm thông tin chi tiết. Chúng được sử dụng cho các hệ thống OLAP. | Hệ thống cơ sở dữ liệu NoSQL bao gồm nhiều loại công nghệ cơ sở dữ liệu khác nhau. Các cơ sở dữ liệu này được phát triển để đáp ứng nhu cầu trình bày cho sự phát triển của ứng dụng hiện đại. |
| Ngôn ngữ Query | Structured query language (SQL) | Không có ngôn ngữ query |
| Kiểu | SQL databases là cơ sở dữ liệu dựa trên bảng | NoSQL databases có thể dựa trên tài liệu, cặp khóa-giá trị, cơ sở dữ liệu biểu đồ |
| Schema | SQL databases có lược đồ được xác định trước | NoSQL databases sử dụng lược đồ động cho dữ liệu phi cấu trúc. |
| Hiệu năng | SQL hoạt động tốt và nhanh thì | Nhanh hơn SQL NoSQL thì |

| | | |
|--|--|---|
| | việc desgin tốt là cực kì quan trọng và ngược lại. | denormalized cho phép bạn lấy được tất cả thông tin về một item cụ thể với các codition mà không cần JOIN liên quan hoặc truy vấn SQL phức tạp. |
|--|--|---|

Bảng 1: Bảng mô tả các tác nhân của hệ thống

2.2.7. Lý do lựa chọn PostgreSQL

- **Tính linh hoạt và đa dạng:** PostgreSQL hỗ trợ nhiều loại dữ liệu, bao gồm cả dữ liệu cấu trúc và bán cấu trúc. Khả năng lưu trữ và xử lý dữ liệu JSON, XML, và các loại dữ liệu phức tạp khác làm cho nó phù hợp với nhiều loại ứng dụng khác nhau.
- **Hiệu suất mạnh mẽ:** PostgreSQL được tối ưu hóa để cung cấp hiệu suất tốt đối với các tập dữ liệu lớn và trong các truy vấn phức tạp. Cơ sở dữ liệu này cung cấp các công cụ và cấu trúc tối ưu hóa để tối đa hóa hiệu suất.
- **Tính bảo mật:** Với khả năng kiểm soát truy cập cấp độ người dùng và quyền hạn cụ thể, cùng với khả năng mã hóa dữ liệu, PostgreSQL là một trong những hệ quản trị cơ sở dữ liệu an toàn và tin cậy.
- **Hỗ trợ ACID:** PostgreSQL tuân thủ nguyên tắc ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo tính nhất quán, an toàn và tin cậy của giao dịch trong cơ sở dữ liệu.



Hình 1.2: ACID and BASE transaction

- **Mở rộng dễ dàng:** PostgreSQL hỗ trợ mô hình mở rộng theo chiều ngang (horizontal

scaling), cho phép việc mở rộng dữ liệu và tải lên cơ sở dữ liệu dễ dàng hơn khi cần thiết.

- **Cộng đồng mạnh mẽ và hỗ trợ tốt:** Với cộng đồng người dùng rộng lớn và sự hỗ trợ nhiệt tình từ cộng đồng mã nguồn mở, người dùng PostgreSQL có thể tìm thấy nhiều tài liệu, hướng dẫn và hỗ trợ từ cộng đồng.

- **Tính ổn định và phát triển liên tục:** PostgreSQL có một chu kỳ phát triển rõ ràng, với các phiên bản mới được cập nhật thường xuyên để cải thiện tính năng và khả năng của nó.

3. CƠ SỞ LÝ THUYẾT HỆ THỐNG KHUYẾN NGHỊ

3.1. Giới thiệu

Sự xuất hiện gần đây của các mạng xã hội và các thiết bị di động phổ biến đã góp phần tạo ra một lượng thông tin khổng lồ trên Internet. Thế giới đã bước vào kỷ nguyên Quá tải thông tin (Information Overload): nhiều thông tin được tạo ra hơn những gì chúng ta thực sự có thể tiêu thụ và xử lý. Để hiểu ý nghĩa của nó trong thực tế, theo thống kê năm 2021 được cung cấp bởi Google Analytics, comScore, Nielsen... cho thấy chỉ trong một phút có khoảng 4,497,420 lượt tìm kiếm được thực hiện trên Google, hơn 1,440,000 lượt chia sẻ hình ảnh hoặc video mới được thực hiện trên Pinterest, khoảng 500 giờ video mới được tải lên và có khoảng 694,444 giờ video được xem trên Youtube.

Với khả năng cho phép người dùng tìm thấy bất kỳ thứ gì họ đang tìm kiếm trong bộ thông tin và dữ liệu khổng lồ này. Tuy nhiên trên thực tế con người không thể xử lý được nhiều thông tin như vậy nếu không có sự hỗ trợ của bất kỳ công cụ lọc tự động nào. Hệ thống khuyến nghị (RS) là một nhóm các công cụ lọc thông tin đã được chứng minh là phương tiện có giá trị trong việc hỗ trợ người dùng tìm kiếm, theo cách cá nhân hóa, những gì có liên quan đến họ trong không gian đào tạo phức tạp như vậy. Chúng cung cấp cho người dùng quyền truy cập được cá nhân hóa vào các tài nguyên lớn. Mặt khác, trong hơn 20 năm qua chúng ta đã giúp cho sự phổ biến của các công cụ lọc thông tin mới này, cụ thể là các công cụ khuyến nghị đã được chứng minh là rất hữu ích trong việc hỗ trợ người dùng giải quyết các công việc, đưa ra quyết định hằng ngày kể cả trong các tình huống phức tạp.

3.2. Hệ thống khuyến nghị – Recommender Systems

3.2.1. Định nghĩa

Hệ thống khuyến nghị (RS) là các công cụ và kỹ thuật phần mềm cung cấp đề xuất

về các mục sẽ hữu ích cho người dùng. Những đề xuất như vậy có thể liên quan đến các quá trình đưa ra quyết định khác nhau, chẳng hạn như người dùng nên kết nối với mạng xã hội nào, mua sản phẩm gì, nghe nhạc gì hoặc xem phim gì. Các sản phẩm mua sắm, âm nhạc, phim ảnh đều là ví dụ về các mặt hàng trong các tình huống đề xuất cụ thể. Ngày nay, hầu hết mọi dịch vụ trực tuyến đều có tính năng đề xuất. Pandora, Netflix, LinkedIn, Youtube và nhiều công ty, tập đoàn lớn khác sử dụng các chức năng đề xuất trong hệ thống của họ để thu hút người dùng và cung cấp cho người dùng dịch vụ tốt hơn.

Mục đích chính của RS là giúp người dùng thỏa mãn nhu cầu mong muốn của họ khi xử lý lượng thông tin khổng lồ. Để đạt được điều này, RS cố gắng lựa chọn tập hợp con các mục phù hợp nhất với sở thích và thị hiếu của người dùng. Theo một số tài liệu, hệ thống khuyến nghị được định nghĩa: “Thuật ngữ hệ thống khuyến nghị chỉ ra rằng bất kỳ hệ thống nào tạo ra các đề xuất được cá nhân hóa làm đầu ra hoặc có tác dụng hướng dẫn người dùng theo cách cá nhân hóa đến các đối tượng thú vị hoặc hữu ích trong một không gian rộng lớn có thể tùy chọn”.



Hình 1.3: Ví dụ về Quá tải thông tin (Information Overload)

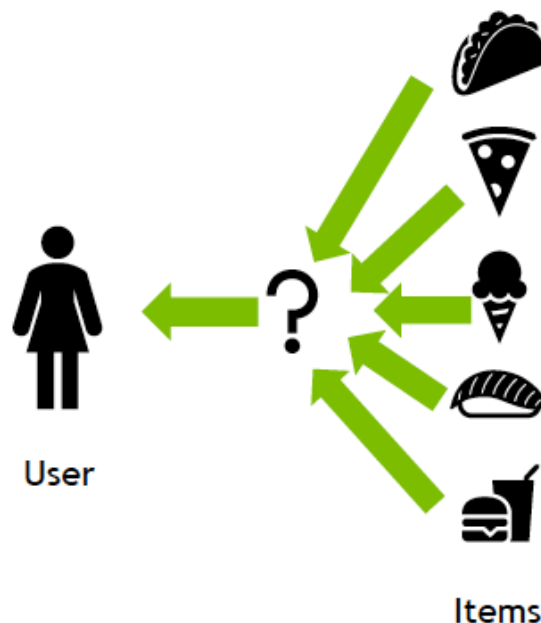
Bên trên là một ví dụ về vấn đề Quá tải Thông tin diễn hình được mô tả trong đó người dùng xem một bộ phim và không biết nên chọn phim nào. Với các tình huống thực tế, người dùng bị choáng ngợp với hàng nghìn/triệu mặt hàng, thì việc đưa ra lựa chọn đúng đắn nếu không có bất kỳ sự trợ giúp nào.

Về nguyên tắc, mục đích chính của cả hệ thống khuyến nghị và hệ thống tìm kiếm là đáp ứng nhu cầu thông tin của người dùng. Tuy nhiên, có khá nhiều khác biệt cơ bản giữa hai công nghệ này. Theo Jeffrey O'Brien, một nhà báo của tạp chí Fortune, khi nói về các hệ thống khuyến nghị, đã trích dẫn “Họ nói rằng Web đang rời khỏi kỉ nguyên tìm kiếm và bước vào kỷ nguyên khám phá. Có gì khác biệt? Tìm kiếm là những gì bạn làm khi bạn đang tìm kiếm thứ gì đó. Khám phá là khi có điều gì đó tuyệt vời mà bạn không biết là có tồn tại hoặc không biết cách yêu cầu, đã tìm thấy bạn”. So với các hệ thống tìm kiếm, hệ thống khuyến nghị cung cấp khả năng cho người dùng khám phá các tài nguyên mới mà ban đầu họ có thể chưa nghĩ đến mà không cần phải xác định rõ ràng nhu cầu của mình.

3.2.2. Users, Items and Ratings

- Theo công thức chính thức của bài toán khuyến nghị được định nghĩa như sau. Gọi U là tập hợp người dùng và I là tập hợp các sản phẩm trong hệ thống và 2 tập hợp này có thể rất lớn. Cho $f: U \times I \rightarrow R$ (trong đó R là tập hợp có thứ tự toàn phần) là hàm đo lường mức độ hữu ích của mục $i \in I$ đối với người dùng $u \in U$.
- Khi đó, bài toán khuyến nghị bao gồm việc tìm cho mỗi người dùng u sản phẩm $i^{max,u} \in I$ để tối đa hoá hàm f : [11]

$$\forall u \in U, i^{max,u} = \arg \max_{i \in I} f(u, i)$$



Hình 1.4: Users, Items and Ratings

- **Users:** Người dùng (Users) là những tác nhân của hệ thống được cung cấp các đề xuất. Người dùng có thể được biểu diễn theo nhiều cách khác nhau tùy thuộc vào kỹ thuật đề xuất được sử dụng để tính toán các đề xuất. Để cung cấp các đề xuất được cá nhân hóa, hệ thống phải lập mô hình và duy trì thông tin về sở thích của họ. Trong RS dựa trên nội dung, sở thích của người dùng có thể được biểu diễn theo cách minh bạch hơn bằng các vector thuộc tính/thuật ngữ theo cách tiếp cận dựa trên heuristic, bằng mô hình theo cách tiếp cận dựa trên mô hình hoặc bằng các công cụ biểu diễn tri thức.
- **Items:** Sản phẩm (Items) là thuật ngữ chung được sử dụng để biểu thị tài nguyên mà hệ thống đề xuất cho người dùng. Các mặt hàng có thể được đặc trưng bởi độ phức tạp và giá trị hoặc tiện ích của chúng. Ví dụ về các Items có độ phức tạp và giá trị thấp là: tin tức, trang Web, sách, phim. Trong khi ví dụ về các mặt hàng phức tạp hơn và có giá trị cao hơn có thể bao gồm từ điện thoại di động, máy tính xách tay đến dịch vụ tài chính, công việc và du lịch. Tùy thuộc vào hệ thống và kỹ thuật đề xuất, nội dung Items có thể có ít nhiều cấu trúc phức tạp.

| | | | | |
|--|---|---|--|---|
| |  |  |  |  |
| John  | 5 | 1 | 3 | 5 |
| Tom  | ? | ? | ? | 2 |
| Alice  | 4 | ? | 3 | ? |

Hình 1.5: Ví dụ ma trận xếp hạng user-item trong đề xuất phim

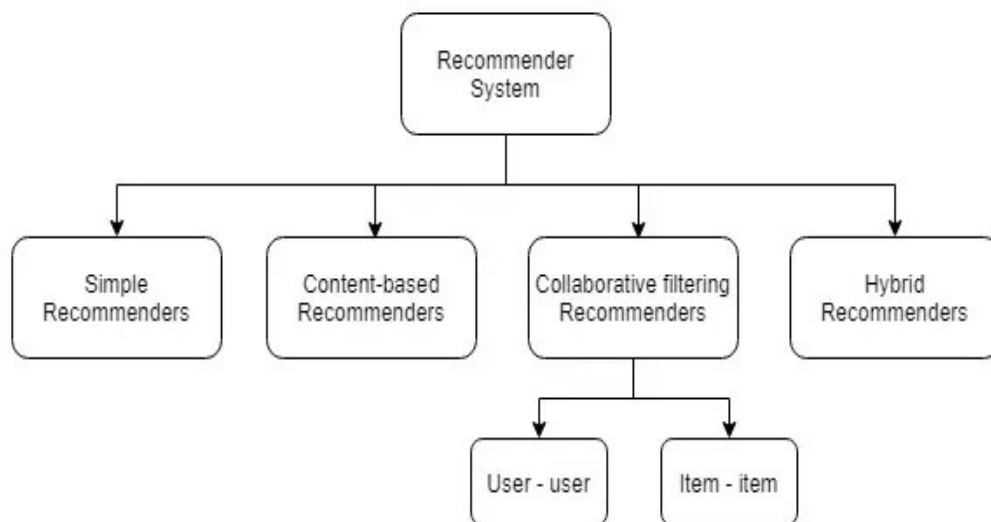
- **Ratings:** Điều quan trọng nhất mà RS dựa vào là tính sẵn có của thông tin cập nhật về sở thích của người dùng dưới dạng phản hồi của người dùng. Tùy thuộc vào cách thu thập thông tin đó, phản hồi của người dùng có thể được phân loại là rõ ràng hoặc ẩn ý. Người dùng được yêu cầu đưa ra ý kiến của mình về một mục theo thang đánh giá có thể là số (ví dụ: 1-5 sao) hoặc thứ tự (rất đồng ý, đồng ý, trung lập, không đồng ý, rất không đồng ý) hoặc cũng có thể

là nhị phân (thích/không thích). Mặc dù trường hợp phản hồi rõ ràng phổ biến hơn chủ yếu là do có sẵn nhiều bộ dữ liệu có xếp hạng, nhưng trên thực tế, trường hợp hệ thống thu thập phản hồi ngầm từ người dùng lại phổ biến hơn. Một hệ thống có thể suy ra sở thích của người dùng bằng cách theo dõi hành vi của người dùng mà không gây phiền toái cho người dùng.

- **Từ dự đoán xếp hạng đến xếp hạng:** Trong việc xây dựng bài toán khuyến nghị nêu trên, hệ thống chủ yếu được xem như một hệ thống dự đoán chính xác các xếp hạng. Vấn đề như vậy được gọi dự đoán xếp hạng. Tuy nhiên, mục tiêu cuối cùng của hệ thống trong hầu hết các tình huống là cung cấp cho người dùng danh sách đề xuất được xếp hạng, cụ thể là đề xuất N hàng đầu. Như trong nhiều hệ thống thương mại, các đề xuất tốt nhất được hiển thị nhưng giá trị xếp hạng dự đoán thì không.

3.2.3. Các phương pháp, kỹ thuật trong Recommender System

Tùy thuộc vào cách ước tính hàm tiện ích và tính sẵn có của dữ liệu về đặc điểm của các sản phẩm, có nhiều loại kỹ thuật, phương pháp khuyến nghị khác nhau. Hai cái chính là: lọc cộng tác (Collaborative filtering) và dựa trên nội dung (Content-based). Bên cạnh hai cách tiếp cận này, còn có các cách tiếp cận khác như dựa trên tri thức (knowledge-based), nhân khẩu học và dựa vào cộng đồng (community-based). Một loại hệ thống khuyến nghị thường được sử dụng trong thực tế là các hệ thống tư vấn kết hợp (Hybrid recommenders) với các chiến lược khác nhau để cải thiện hiệu suất riêng biệt của chúng và đạt được chất lượng đề xuất cao hơn.

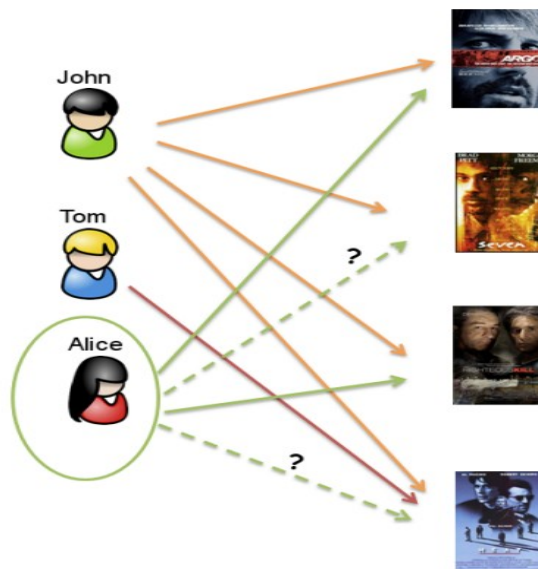


Hình 1.6: Các phương pháp, kỹ thuật trong Recommender System

3.3. Tìm hiểu các phương pháp, kỹ thuật trong Recommender Systems

3.3.1. Collaborative Filtering Recommendation

Collaborative Filtering (Lọc cộng tác) là quá trình lọc hoặc đánh giá các sản phẩm bằng cách sử dụng ý kiến của người khác. Theo cách tiếp cận này, các đề xuất được cá nhân hóa cho người dùng mục tiêu được tạo ra bằng cách sử dụng ý kiến của những người dùng có sở thích tương tự với sở thích của người dùng mục tiêu. Giả định chính trong phương pháp này là người dùng có sở thích tương tự trong quá khứ sẽ có sở thích tương tự trong tương lai.



Hình 1.7: Minh họa hệ thống khuyến nghị dựa trên Collaborative Filtering

Khác với bất kỳ kỹ thuật nào khác, dữ liệu đầu vào duy nhất mà CF-RS là ma trận xếp hạng user-item. Hình ảnh trên biểu diễn một ví dụ đơn giản về trường hợp lọc cộng tác tương ứng với ma trận xếp hạng xếp hạng user-item. Nếu chúng ta coi *Alice* là người dùng mục tiêu, như đã nói trước đó, thì các đề xuất sẽ được tạo dựa trên xếp hạng do những người dùng khác có cùng sở thích đưa ra. Trong trường hợp cụ thể này, cả *John* và *Alice* đều có sở thích giống nhau vì cả hai đều đánh giá *Argo and Righteous Kill* tương tự nhau. Hệ thống có thể khai thác xếp hạng của *John* để ước tính xếp hạng chưa biết của *Alice*. Trực giác cơ bản đằng sau phương pháp này là vì *John* thực sự thích *Heat* nên *Alice* cũng có thể thích nó.

Có hai loại phương pháp lọc cộng tác chính: dựa trên bộ nhớ (memory-based) và dựa trên mô hình (model-based). CF dựa trên bộ nhớ sử dụng một loại phương pháp Machine Learning cụ thể là thuật toán lân cận (k-NN) gần nhất. Đặc tính chính của phương pháp này là nó không yêu cầu bất kỳ giai đoạn xây dựng mô hình sơ bộ nào vì các dự đoán được thực hiện bằng cách tổng hợp xếp hạng của những nước láng giềng gần nhất. Ngược lại, các kỹ thuật dựa trên mô hình trước tiên sẽ học một mô hình dự đoán mà cuối cùng được sử dụng để đưa ra dự đoán.

Các phương pháp tiếp cận dựa trên bộ nhớ có thể được phân loại dựa trên người dùng hoặc dựa trên sản phẩm. Cách tiếp cận dựa trên người dùng bao gồm việc dự đoán mức độ liên quan của một mặt hàng đối với người dùng mục tiêu bằng cách kết hợp tuyến tính các xếp hạng của người hàng xóm, được đánh giá cao bởi sự tương

đồng giữa người dùng mục tiêu và những người hàng xóm đó. Xem xét độ lệch xếp hạng so với phương tiện xếp hạng của người dùng và hàng xóm (\bar{r}_u). Dự đoán cho người dùng đang hoạt động u và mục tiêu i được tính như sau: [11]

$$r_{u,i} = \bar{r}_u + \frac{\sum_{j=1}^K (r_{u_j,i} - \bar{r}_u) \cdot w_{u,u_j}}{\sum_{j=1}^K w_{u,u_j}}$$

Trong đó K là số lượng hàng xóm của người dùng u và w_{u,u_j} là trọng số tương tự giữa người dùng đang hoạt động u và hàng xóm u_j được xác định bởi hệ số tương quan Pearson dưới đây: [11]

$$w_{u,u_j} = \frac{\sum_i (r_{u,i} - \bar{r}_u) \cdot (r_{u_j,i} - \bar{r}_{u_j})}{\sqrt{\sum_i (r_{u,i} - \bar{r}_u)^2 + \sum_i (r_{u_j,i} - \bar{r}_{u_j})^2}}$$

Cách tiếp cận item-based CF dựa trên việc sử dụng cùng các kỹ thuật dựa trên tương quan hoặc dựa trên *cosine* để tính toán sự tương đồng giữa các vật phẩm thay vì người dùng.

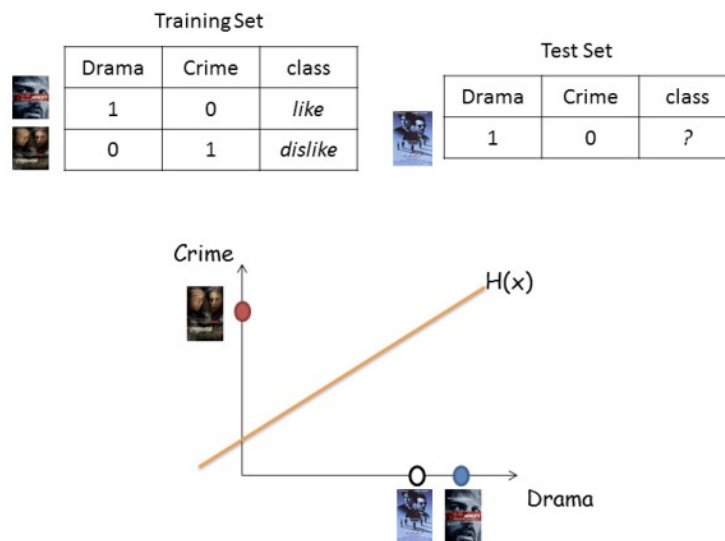
3.3.2. Content-based Recommendation

Content-based RS dựa trên nội dung đề xuất một mục cho người dùng, mô tả về mục đó và hồ sơ sở thích của người dùng. Quy trình cơ bản được thực hiện bởi người giới thiệu dựa trên nội dung bao gồm việc khớp các thuộc tính của hồ sơ người dùng trong đó các tùy chọn và sở thích được lưu trữ, với các thuộc tính của một đối tượng nội dung.



Hình 1.8: Minh họa hệ thống khuyến nghị Content-based

Khác với lọc cộng tác, phương pháp đề xuất này phụ thuộc vào tính sẵn có của các nội dung mô tả các sản phẩm. Các tính năng như vậy có thể được trích xuất từ các mô tả sản phẩm không có cấu trúc hoặc bán cấu trúc bằng cách sử dụng các kỹ thuật Xử lý ngôn ngữ tự nhiên (NLP) thích hợp hoặc có thể được lấy từ dữ liệu có cấu trúc như trường hợp dữ liệu dạng bảng trong cơ sở dữ liệu quan hệ. Hình ảnh dưới đây cho thấy một ví dụ về cách tiếp cận dựa trên nội dung có liên quan đến người dùng *Alice*.



Hình 1.9: Minh họa model-based CB-RS

Như chúng ta có thể thấy, khác với trường hợp CF ở cách tiếp cận này, phim được cung cấp các thuộc tính, chẳng hạn như diễn viên, thể loại, Sự khác biệt là chỉ người dùng mục tiêu mới được xem xét trong quá trình đề xuất. Trực giác cơ bản đằng sau cách tiếp cận này là vì *Alice* thích *Argo* nên cô ấy có thể thích *Heat* vì cả hai đều thuộc thể loại Drama.

Dưới đây là hai phương pháp đề xuất dựa trên nội dung chính: heuristic-based hoặc model-based:

- Các phương pháp tiếp cận sử dụng hàm heuristic có nguồn gốc từ Information Retrieval and Information Filtering (Truy xuất thông tin và Lọc thông tin). Các sản phẩm được đề xuất dựa trên sự so sánh giữa nội dung của chúng và hồ sơ người dùng.
- Các phương pháp tiếp cận dựa trên mô hình sử dụng kỹ thuật Machine Learning để tìm hiểu mô hình sở thích của người dùng bằng cách phân tích đặc điểm nội dung của các mục mà người dùng đã xếp hạng.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1. YÊU CẦU HỆ THỐNG

1.1. Sơ đồ tổng quan usecase của hệ thống

1.1.1. Các tác nhân:

- Tác nhân tham gia vào hệ thống:
- Người dùng (User)
- Người quản lý (Admin)

Bảng 2: Bảng mô tả các tác nhân của hệ thống

| STT | Tác nhân | Mô tả |
|-----|------------|--|
| 1 | Người dùng | Người sử dụng ứng dụng để mua sách, theo dõi đơn hàng, đọc sách, nghe sách điện tử và quản lý sách của mình. |
| 2 | Quản lý | Người quản lý phụ trách sách, tải sách mới lên, quản lý đơn hàng. |

1.1.2. Danh sách các chức năng chính

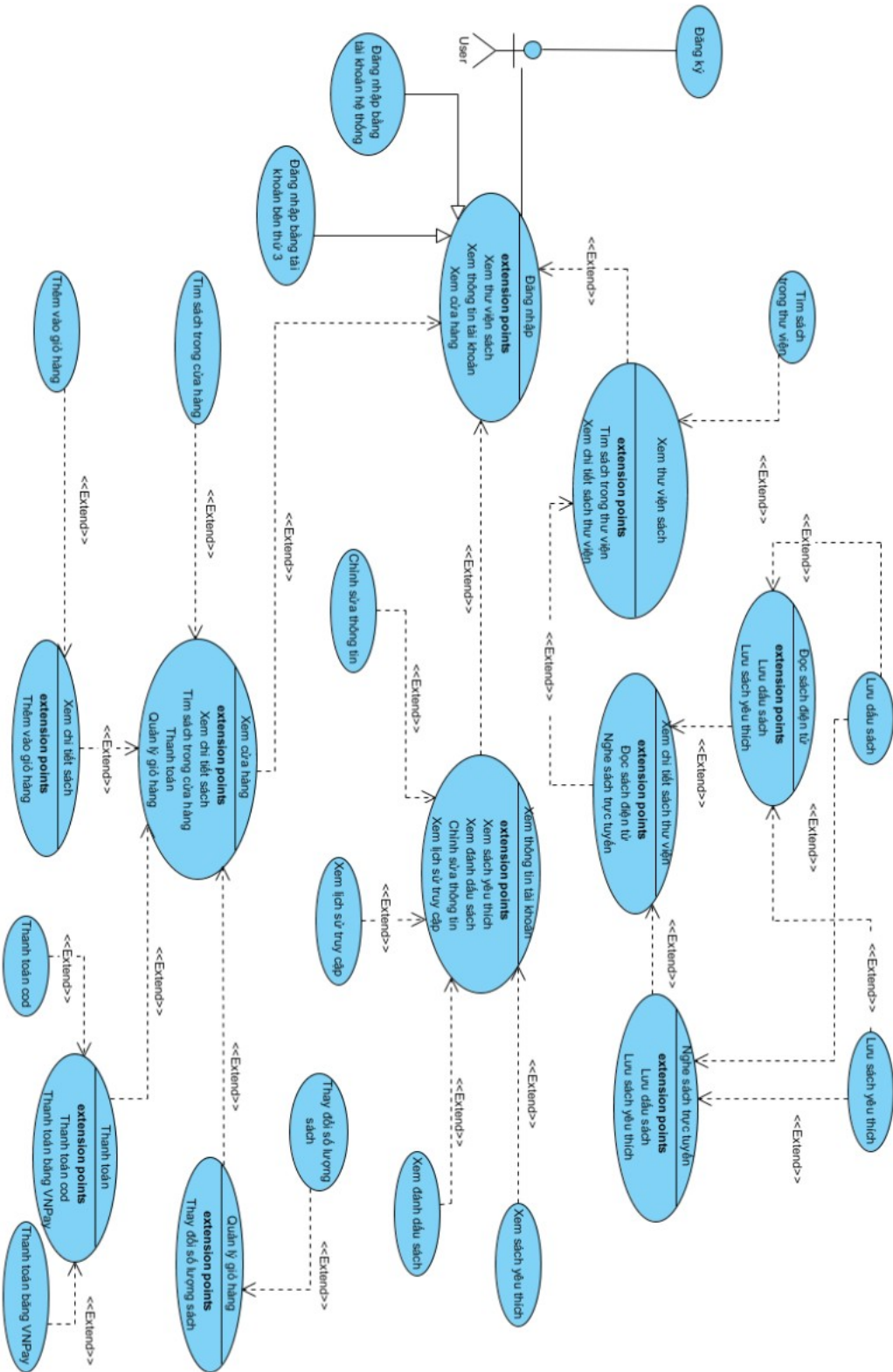
Bảng 3: Bảng danh sách các chức năng chính

| STT | Tên chức năng | Mô tả |
|-----|-------------------------------|--|
| 1 | Đăng ký | Cho phép người dùng đăng ký tài khoản để sử dụng hệ thống |
| 2 | Đăng nhập | Cho phép người dùng đăng nhập vào hệ thống |
| 3 | Đăng ký với tài khoản thứ 3 | Cho phép người dùng đăng ký tài khoản bằng tài khoản của bên thứ 3 |
| 4 | Đăng nhập với tài khoản thứ 3 | Cho phép người dùng đăng nhập bằng tài khoản của bên thứ 3. |
| 5 | Tìm kiếm sách | Cho phép người dùng tìm kiếm sách trong thư |

| | | |
|----|-------------------------------|--|
| | | viện. |
| 6 | Đọc sách điện tử | Cho phép người dùng đọc sách điện tử trong thư viện. |
| 7 | Nghe sách nói | Cho phép người dùng nghe sách nói trong thư viện. |
| 8 | Thêm dấu sách | Cho phép người dùng lưu vị trí dấu sách ở trong sách điện tử và sách nói. |
| 9 | Thêm sách yêu thích | Cho phép người dùng thêm sách yêu thích vào mục sở thích của bản thân. |
| 10 | Xem dấu sách | Cho phép người dùng xem danh sách các dấu sách đã được lưu. |
| 11 | Xem sách yêu thích | Cho phép người dùng xem được danh sách sách yêu thích đã được lưu. |
| 12 | Lịch sử truy cập | Cho phép người dùng truy cập vào lịch sử những sách đã đọc thời gian gần nhất. |
| 13 | Thông tin chi tiết sách | Cho phép người dùng xem thông tin chi tiết của sách bao gồm nội dung, tác giả, ... |
| 14 | Chỉnh sửa thông tin tài khoản | Cho phép người dùng thay đổi thông tin cơ bản của tài khoản cá nhân. |
| 15 | Thay đổi mật khẩu | Cho phép người dùng thay đổi mật khẩu của tài khoản |
| 16 | Chỉnh phong chữ | Cho phép người dùng thay đổi kích cỡ phong chữ khi đang đọc sách điện tử. |
| 17 | Nổi bật văn bản | Cho phép người dùng đánh dấu vị trí những chỗ nổi bật trong sách điện tử. |
| 18 | Thay đổi cách đọc sách | Cho phép người dùng thay đổi cách đọc sách điện tử ở dạng cuộn và dạng trang sách. |
| 19 | Danh mục chương | Cho phép người dùng lựa chọn danh mục các |

| | | |
|----|------------------------------|---|
| | | chương trong sách điện tử và sách nói. |
| 20 | Giao diện đọc sách | Cho phép người dùng thay đổi giao diện đọc sách điện tử với các chế độ nền sáng, nền tối. |
| 21 | Phóng to | Cho phép người dùng phóng to màn hình khi đang đọc sách điện tử. |
| 22 | Toàn màn hình | Cho phép người dùng đọc sách điện tử trong chế độ toàn màn hình. |
| 23 | Chỉnh sửa màu chữ và màu nền | Cho phép người dùng thay đổi màu chữ và màu nền trong giao diện đọc sách điện tử. |
| 24 | Chuyển sách | Cho phép người dùng chuyển sang chương khác của sách nói hoặc tự động chuyển khi chương kết thúc. |
| 25 | Tua nhanh chậm | Cho phép người dùng tua nhanh hoặc chậm khi đang nghe sách nói. |
| 26 | Thay đổi tốc độ đọc | Cho phép người dùng thay đổi tốc độ đọc của sách nói. |
| 27 | Lặp lại | Cho phép người dùng lặp lại chương sách nói đang được bật. |
| 28 | Hẹn giờ | Cho phép người dùng hẹn giờ để tự động tắt sách nói đang bật. |
| 29 | Đồng bộ sách nói | Cho phép người dùng chuyển đến vị trí sách đang nghe sang vị trí sách điện tử tương ứng. |
| 30 | Đồng bộ sách điện tử | Cho phép người dùng chuyển vị trí sách điện tử đang đọc sang vị trí sách nói tương ứng. |

1.1.3. Biểu đồ Usecase tổng quát



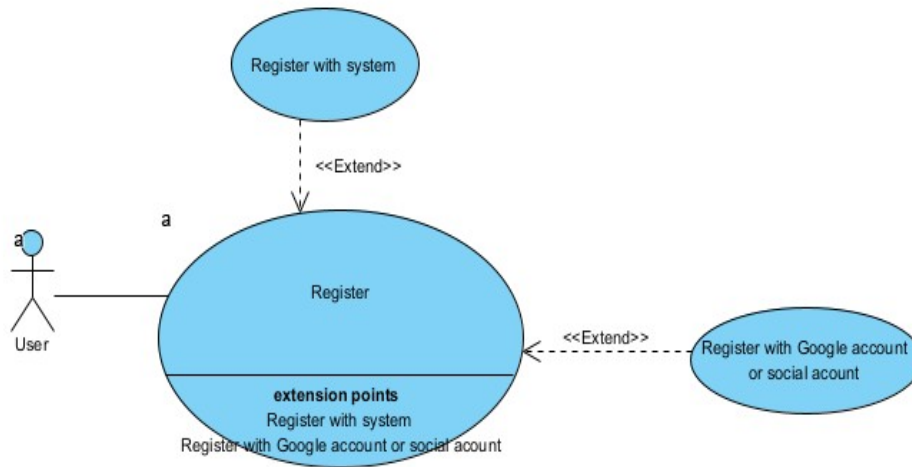
Hình 2.1: Usecase tổng quan hệ thống

1.2. Một số chức năng chính của hệ thống

1.2.1. Đăng ký (Register)

- Mô tả chức năng: Người dùng mới có thể tạo tài khoản bằng cách cung cấp tên, năm sinh, số điện thoại, địa chỉ email hợp lệ và mật khẩu an toàn. Chức năng này cho phép người dùng truy cập vào ứng dụng và sử dụng được đầy đủ các chức năng của ứng dụng.
- Biểu đồ use case:

Hình 2.2: Biểu đồ usecase chức năng Đăng ký



Bảng 4: Kịch bản chức năng Đăng ký

| | |
|---|---|
| Tên usecase | Đăng ký |
| Tác nhân chính | Người dùng (User) |
| Điều kiện | Người dùng chưa đăng ký tài khoản. |
| Đảm bảo tối thiểu | Người dùng được đăng ký lại |
| Đảm bảo thành công | Người dùng đăng ký tài khoản thành công vào ứng dụng hệ thống |
| Chuỗi sự kiện chính: | |
| 1. Người dùng mở ứng dụng và chọn chức năng "Đăng Ký" từ màn hình chính của ứng dụng. | |

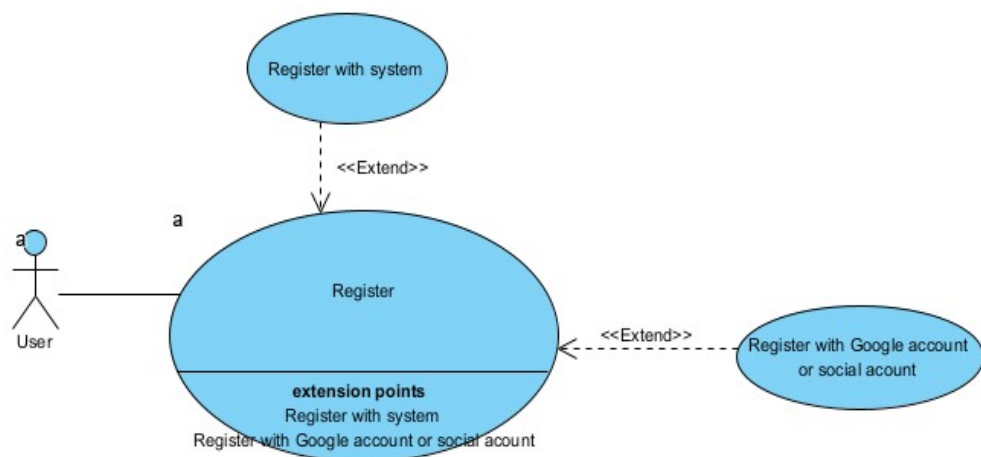
2. Người dùng nhập thông tin cá nhân bao gồm tên, năm sinh, số điện thoại, địa chỉ email, mật khẩu và xác nhận mật khẩu.
3. Người dùng bấm nút “Đăng ký”.
4. Hệ thống xác thực, lưu trữ thông tin và người dùng được chuyển tiếp sang giao diện Đăng nhập.

Ngoại lệ

- 4.1. Người dùng không nhập đủ thông tin yêu cầu
 - 4.1.1. Hệ thống thông báo yêu cầu người dùng phải nhập đủ các trường thông tin cần thiết.
- 4.2. Người dùng nhập email đã tồn tại
 - 4.2.1. Hệ thống thông báo email đã tồn tại và yêu cầu người dùng nhập lại.
- 4.3. Người dùng nhập mật khẩu không đủ 6 ký tự.
 - 4.3.1. Hệ thống thông báo lỗi và yêu cầu người dùng nhập lại.

1.2.2. Đăng nhập tài khoản với các bên thứ 3

- Mô tả chức năng: Người dùng mới có thể tạo tài khoản bằng cách sử dụng các tài khoản thứ 3 như Google, Facebook, ... để đăng ký vào tài khoản hệ thống. Chức năng này cho phép người dùng truy cập vào ứng dụng và sử dụng được đầy đủ các chức năng của ứng dụng.
- Biểu đồ use case:



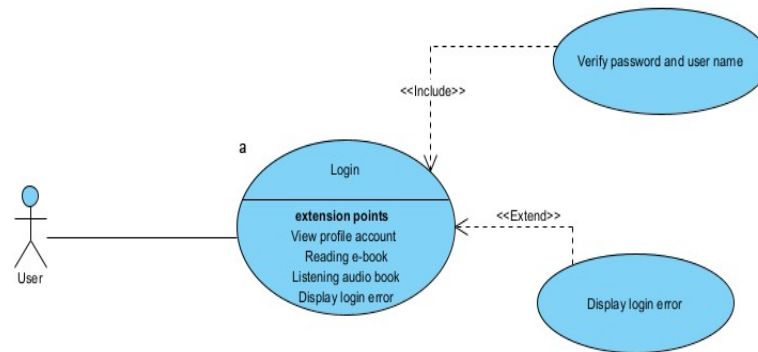
Hình 2.3: Biểu đồ usecase chức năng Đăng nhập với các bên thứ 3

Bảng 5: Kịch bản chức năng Đăng ký với các bên thứ 3

| | |
|---|--|
| Tên usecase | Đăng nhập với các bên thứ 3 |
| Tác nhân chính | Người dùng (User) |
| Điều kiện | Người dùng chưa đăng nhập tài khoản. |
| Đảm bảo tối thiểu | Người dùng được đăng nhập lại |
| Đảm bảo thành công | Người dùng đăng ký tài khoản thành công và đăng nhập được vào ứng dụng |
| Chuỗi sự kiện chính: | |
| <ol style="list-style-type: none"> 1. Người dùng mở ứng dụng. 2. Người dùng chọn đăng nhập qua tài khoản Google hoặc tài khoản mạng xã hội (Google, Facebook ...). 3. Hệ thống hiển thị màn hình đăng nhập với bên tài khoản thứ 3. 4. Người dùng nhập thông tin tài khoản, mật khẩu và đăng nhập 5. Hệ thống bên thứ 3 xác thực thông tin và gửi thông tin trả lại hệ thống. Hệ thống xác thực thông tin, lưu trữ thông tin và người dùng được chuyển sang giao diện Đăng nhập. | |
| Ngoại lệ | |
| <ol style="list-style-type: none"> 5.1. Người dùng không nhập đủ thông tin yêu cầu <ol style="list-style-type: none"> 5.1.1. Hệ thống thông báo yêu cầu người dùng phải nhập đủ các trường thông tin cần thiết. 5.2. Hệ thống bên thứ 3 xác thực sai thông tin <ol style="list-style-type: none"> 5.2.1. Hệ thống yêu cầu người dùng nhập lại thông tin đăng nhập với các tài khoản thứ 3. | |

1.2.3. Đăng nhập (Login)

- Mô tả chức năng: Người dùng có thể đăng nhập vào ứng dụng bằng tài khoản và mật khẩu đã đăng ký vào hệ thống ứng dụng. Chức năng này cho phép người dùng truy cập vào ứng dụng và sử dụng được đầy đủ các chức năng của ứng dụng.
- Biểu đồ use case



Hình 2.4: Biểu đồ use case chức năng Đăng nhập

Bảng 6: Kịch bản chức năng Đăng nhập

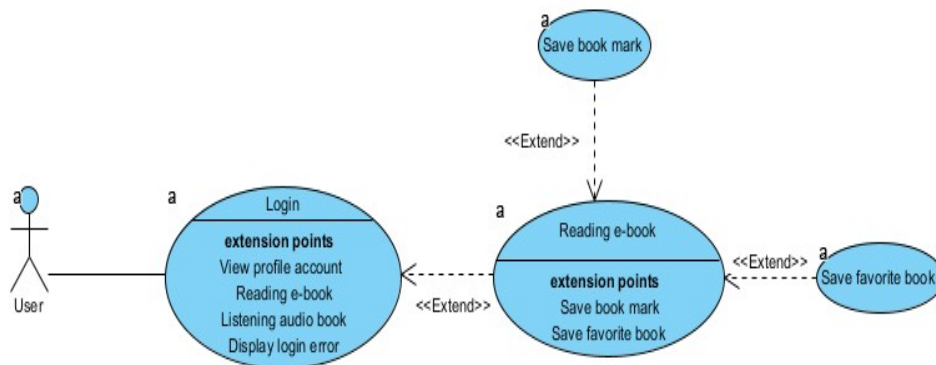
| | |
|--|---|
| Tên usecase | Đăng nhập |
| Tác nhân chính | Người dùng (User) |
| Điều kiện | Người dùng chưa đăng nhập vào hệ thống. |
| Đảm bảo tối thiểu | Người dùng được đăng nhập lại |
| Đảm bảo thành công | Người dùng đăng nhập thành công vào ứng dụng hệ thống |
| Chuỗi sự kiện chính: | |
| <ol style="list-style-type: none"> 1. Người dùng mở ứng dụng. 2. Hệ thống hiển thị chức năng "Đăng Nhập". 3. Người dùng nhập tài khoản, mật khẩu và bấm nút đăng nhập. 4. Hệ thống kiểm tra và xác thực thông tin đăng nhập. Hệ thống xác thực, thông báo người dùng đăng nhập thành công và chuyển đến giao diện chính của ứng dụng | |

Ngoại lệ

- 4.1. Người dùng không nhập đủ thông tin yêu cầu
 - 4.1.1. Hệ thống thông báo yêu cầu người dùng phải nhập đủ các trường thông tin cần thiết.
- 4.2. Người dùng nhập sai định dạng của email.
 - 4.2.1. Hệ thống thông báo sai email định dạng và yêu cầu người dùng nhập lại.
- 4.3. Người dùng nhập sai thông tin đăng nhập.
 - 4.3.1. Hệ thống thông báo lỗi và yêu cầu người dùng nhập lại.

1.2.4. Đọc sách điện tử (Reading e-book)

- Mô tả chức năng: Người dùng có thể chọn cuốn sách từ thư viện ở trên giao diện chính của ứng dụng và mở nó để đọc trực tiếp trên màn hình điện thoại của mình. Giao diện thông tin và đánh giá sách được hiển thị. Người dùng có thể tùy chỉnh cỡ chữ, màu sắc, độ sáng và đánh dấu trang sách hiện tại để tối ưu hóa trải nghiệm đọc sách.
- Use case:



Hình 2.5: Biểu đồ usecase chức năng Đọc sách điện tử

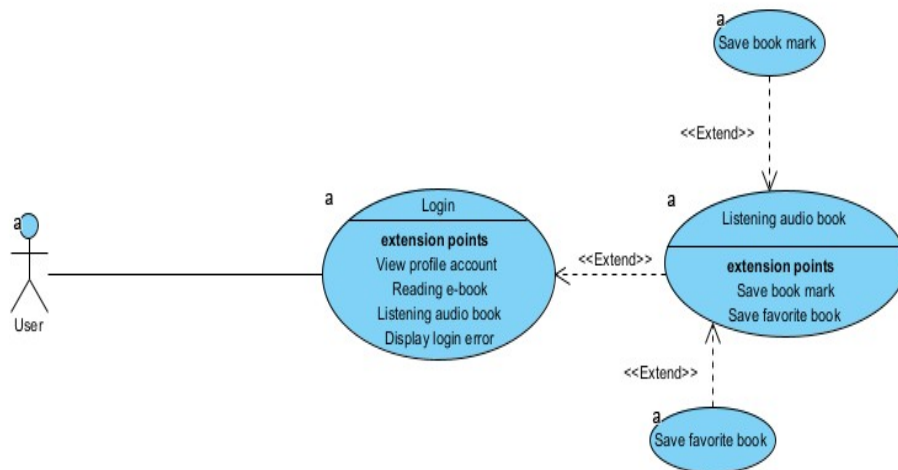
Bảng 7: Kịch bản chức năng Đọc sách điện tử

| | |
|-----------------------|--------------------------------------|
| Tên usecase | Đọc sách điện tử |
| Tác nhân chính | Người dùng (User) |
| Điều kiện | - Người dùng đăng nhập vào hệ thống. |

| | |
|--|---|
| | - Người dùng có ít nhất 1 quyển sách ebook trong thư viện của mình. |
| Đảm bảo thành công | Người dùng đọc được sách điện tử. |
| Chuỗi sự kiện chính: | |
| <ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng. 2. Hệ thống hiển thị giao diện chính của ứng dụng. Hệ thống hiển thị danh sách các sách điện tử được phân loại chia theo thể loại, sách yêu thích và sách khuyến nghị. 3. Người dùng bấm chọn vào một quyển sách. 4. Giao diện thông tin và đánh giá sách được hiển thị. 5. Người dùng bấm nút Đọc sách. 6. Hệ thống hiển thị nội dung sách và người dùng có thể cuộn trang, và đánh dấu trang sách hiện tại, thêm sách vào mục yêu thích... | |

1.2.5. Nghe sách (Listening audio book)

- Mô tả chức năng: Người dùng có thể chọn chức năng audiobook từ thư viện giao diện chính của ứng dụng và nghe nó như một tệp âm thanh. Người dùng có thể điều chỉnh âm lượng, tạm dừng, tua nhanh, tua chậm và chọn từng phần muốn nghe tùy ý. Ngoài ra người dùng có thể tạm dừng và lưu lại phần đang nghe.
- Use case:



Hình 2.6: Biểu đồ usecase chức năng Nghe sách trực tuyến

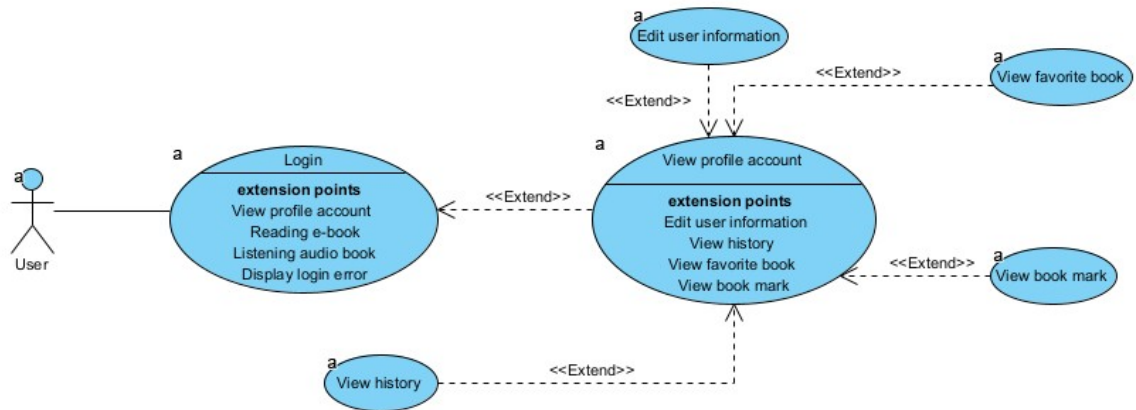
Bảng 8: Kịch bản chức năng Nghe sách

| | |
|--|---|
| Tên usecase | Nghe sách |
| Tác nhân chính | Người dùng (User) |
| Điều kiện | <ul style="list-style-type: none"> - Người dùng đăng nhập vào hệ thống. - Người dùng có ít nhất 1 quyển sách trong thư viện của mình. |
| Đảm bảo thành công | Người dùng nghe được sách từ ứng dụng hệ thống. |
| <p>Chuỗi sự kiện chính:</p> <ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng. 2. Hệ thống hiển thị giao diện chính của ứng dụng. Hệ thống hiển thị danh sách các sách điện tử được phân loại chia theo thể loại, sách yêu thích và sách khuyến nghị. 3. Người dùng bấm chọn vào một quyển sách. 4. Giao diện thông tin và đánh giá sách được hiển thị. 5. Người dùng bấm nút Nghe sách. 6. Ứng dụng bắt đầu phát âm thanh và người dùng có thể sử dụng các nút điều chỉnh như tăng/giảm âm lượng, tạm dừng, tua nhanh/tua chậm, chuyển đoạn qua các phần hoặc đánh dấu lưu phần đang nghe hiện tại. | |

1.2.6. *Chỉnh sửa thông tin tài khoản (Profile)*

- Mô tả chức năng: Người dùng có thể xem và chỉnh sửa thông tin cá nhân tài khoản của họ từ hồ sơ người dùng.
- Use case:

Hình 2.7: Biểu đồ usecase chức năng Chỉnh sửa thông tin tài khoản



Bảng 9: Kịch bản chức năng Chỉnh sửa thông tin tài khoản

| | |
|--|--|
| Tên usecase | Chỉnh sửa thông tin tài khoản |
| Tác nhân chính | Người dùng (User) |
| Điều kiện | Người dùng đăng nhập vào hệ thống. |
| Đảm bảo thành công | Người dùng chỉnh sửa được thông tin tài khoản và cá nhân của mình. |
| Chuỗi sự kiện chính: | |
| 1. Người dùng đăng nhập thành công vào ứng dụng. | |
| 2. Hệ thống hiển thị giao diện chính của ứng dụng. | |
| 3. Người dùng chọn chức năng “Thông tin tài khoản” ở trên giao diện chính. | |
| 4. Hệ thống hiển thị trang quản lý thông tin cá nhân của người. | |
| 5. Người dùng chọn chức năng chỉnh sửa thông tin cá nhân. | |
| 6. Hệ thống hiển thị trang sửa thông tin cá nhân với các ô nhập. | |
| 7. Người dùng thực hiện điền thông tin cá nhân muốn sửa và bấm nút Xác nhận. | |
| 8. Hệ thống xác thực thông tin được nhập, lưu trữ vào cơ sở dữ liệu và chuyển tiếp | |

sang màn hình Thông tin tài khoản.

Ngoại lệ

1.1. Người dùng không nhập đủ thông tin yêu cầu

1.1.1. Hệ thống thông báo yêu cầu người dùng phải nhập đủ các trường thông tin cần thiết.

1.1.2. Người dùng nhập sai định dạng của các trường.

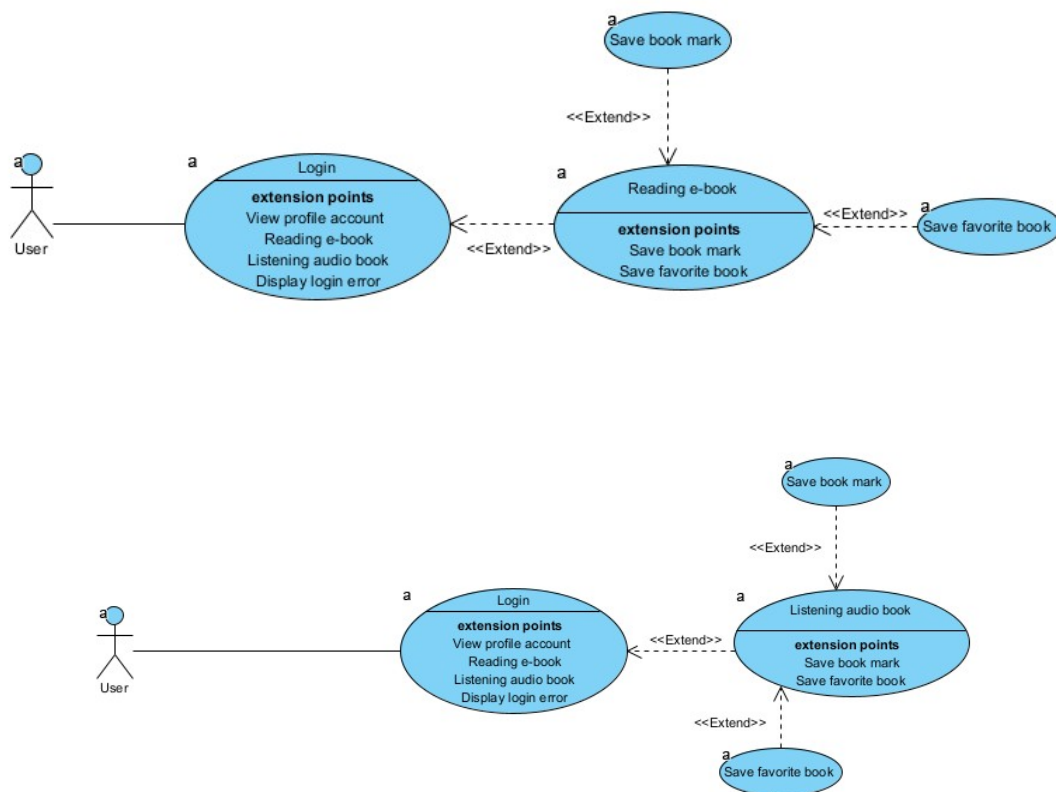
1.1.3. Hệ thống thông báo sai định dạng ở các trường ô nhập đó và yêu cầu người dùng nhập lại.

1.2.7. Đánh dấu sách (Book mark)

- Mô tả chức năng: Người dùng có thể đánh dấu trang hoặc vị trí trong sách ebook hoặc audiobook để tìm lại dễ dàng sau này.

- Use case:

Hình 2.8: Biểu đồ usecase chức năng Đánh dấu sách



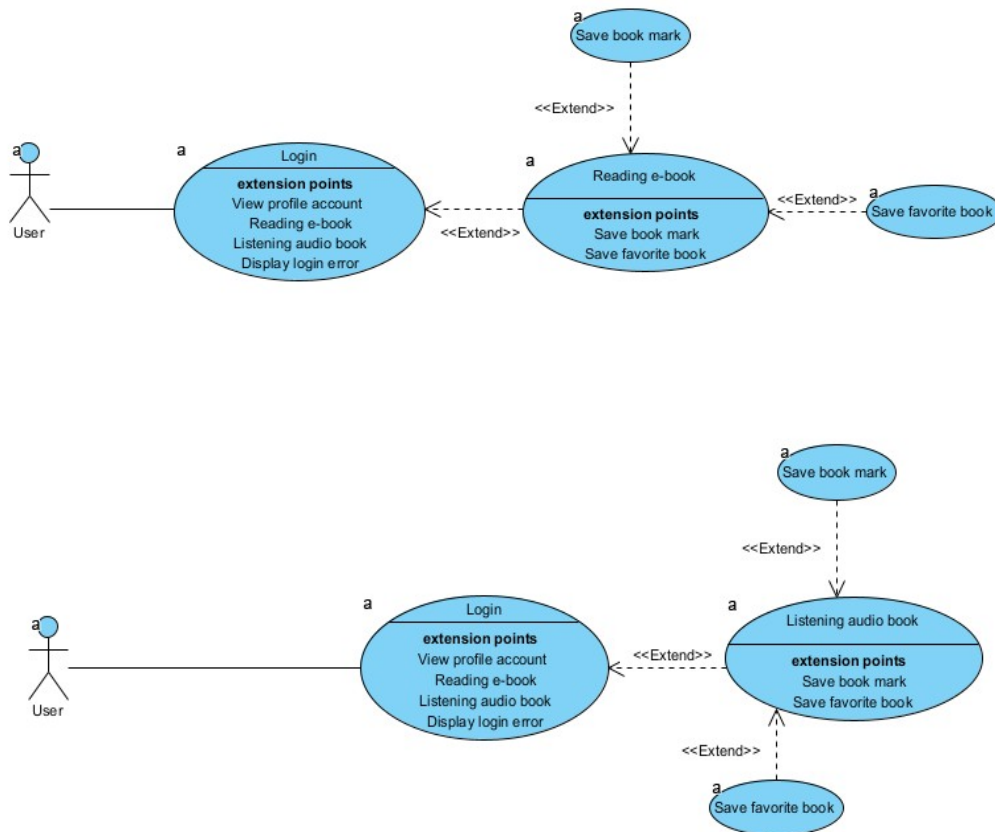
Bảng 10: Kịch bản chức năng Đánh dấu sách

| | |
|---|---|
| Tên usecase | Đánh dấu sách |
| Tác nhân chính | Người dùng (User) |
| Điều kiện | - Người dùng đăng nhập vào hệ thống. - Người dùng có ít nhất 1 quyển sách trong thư viện của mình. |
| Đảm bảo thành công | Người lưu được trạng thái sách đang đọc của mình vào hệ thống |
| <p>Chuỗi sự kiện chính:</p> <ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng. 2. Hệ thống hiển thị giao diện chính của ứng dụng. Hệ thống hiển thị danh sách các sách điện tử được phân loại chia theo thể loại, sách yêu thích và sách khuyến nghị. 3. Người dùng bấm chọn vào một quyển sách. 4. Giao diện thông tin và đánh giá sách được hiển thị. 5. Người dùng bấm Đọc/Nghe sách. 6. Hệ thống chuyển tiếp sang giao diện ứng dụng dùng để đọc/nghe sách 7. Người dùng chọn biểu tượng Đánh dấu sách trên giao diện. 8. Hệ thống xác nhận và lưu lại thông tin số trang hoặc vị trí audio book. | |

1.2.8. Thêm sách yêu thích (Favorite book)

- Mô tả chức năng: Người dùng có thể thêm sách vào danh sách yêu thích để dễ dàng truy cập sau này.
- Use case:

Hình 2.9: Biểu đồ usecase chức năng Thêm sách yêu thích



Bảng 11: Kịch bản chức năng Thêm sách yêu thích

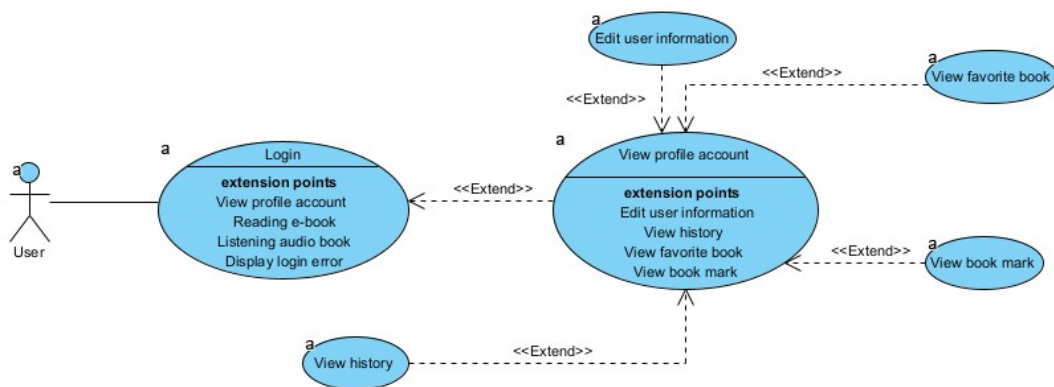
| | |
|---------------------------|---|
| Tên usecase | Thêm sách yêu thích |
| Tác nhân chính | Người dùng (User) |
| Điều kiện | <ul style="list-style-type: none"> - Người dùng đăng nhập vào hệ thống. - Người dùng có ít nhất 1 quyển sách trong thư viện của mình. |
| Đảm bảo thành công | Người thêm được sách vào danh mục |

| | |
|--|--------------------|
| | yêu thích của mình |
| <p>Chuỗi sự kiện chính:</p> <ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng. 2. Hệ thống hiển thị giao diện chính của ứng dụng. Hệ thống hiển thị danh sách các sách điện tử được phân loại chia theo thể loại, sách yêu thích và sách khuyến nghị. 3. Người dùng bấm chọn vào một quyển sách. 4. Giao diện thông tin và đánh giá sách được hiển thị. 5. Người dùng bấm Đọc/Nghe sách. 6. Hệ thống chuyển tiếp sang giao diện ứng dụng dùng để đọc/nghe sách 7. Người dùng chọn biểu tượng Yêu thích sách trên giao diện. 8. Hệ thống xác nhận và lưu lại thông tin sách và lưu lại vào danh mục sách yêu thích. | |

1.2.9. Lịch sử truy cập (History)

- Mô tả chức năng: Người dùng sau khi đọc hoặc nghe sách sẽ hệ thống sẽ tự động lưu lại thông tin về cuốn sách và thông tin vị trí thời gian đọc/nghe lần cuối của người dùng để dễ dàng truy cập sau này.
- Use case:

Hình 2.10: Biểu đồ usecase chức năng Lịch sử truy cập



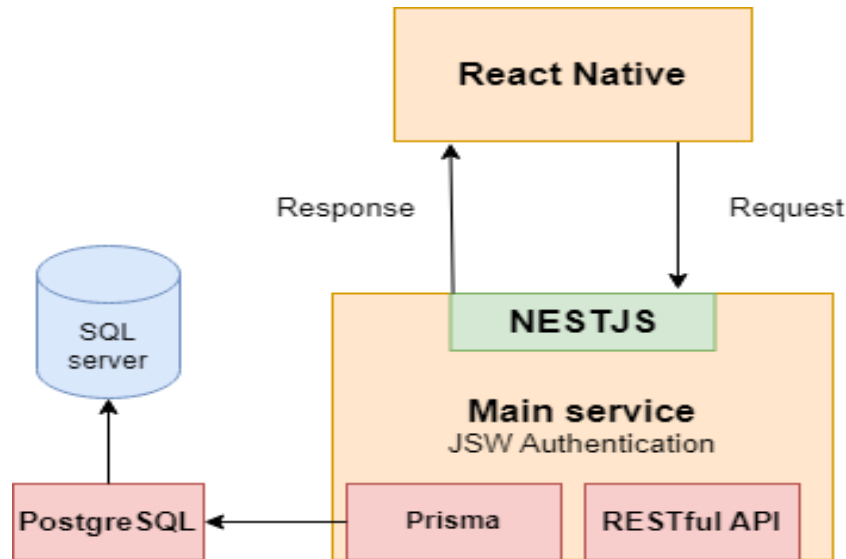
Bảng 12: Kịch bản chức năng Lịch sử truy cập

| | |
|--------------------|------------------|
| Tên usecase | Lịch sử truy cập |
|--------------------|------------------|

| | |
|--|---|
| Tác nhân chính | Người dùng (User) |
| Điều kiện | - Người dùng đăng nhập vào hệ thống. - Người dùng có ít nhất 1 quyển sách trong thư viện của mình. |
| Đảm bảo thành công | Hệ thống lưu được thông tin lần cuối người dùng truy cập vào cuốn sách |
| Chuỗi sự kiện chính: | |
| <ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng. 2. Hệ thống hiển thị giao diện chính của ứng dụng. 3. Người dùng chọn chức năng “Lịch sử” ở trên giao diện chính. 4. Hệ thống hiển thị danh sách những sách đã đọc/nghe gần đây của người dùng được sắp xếp theo thời gian. 5. Người dùng bấm chọn vào một quyển sách. 6. Giao diện thông tin và đánh giá sách được hiển thị. 7. Người dùng tiếp tục có thể chọn chức năng đọc/nghe sách và chuyển màn hình ứng dụng đến chức năng tương ứng. 8. Hệ thống hiển thị thông tin lần cuối người dùng truy cập. 9. Người dùng lựa chọn Đọc tiếp 10. Hệ thống tự động chuyển vị trí cuốn sách đến vị trí lần cuối được truy cập. | |

2. KIẾN TRÚC HỆ THỐNG

2.1. Sơ đồ mô tả kiến trúc hệ thống



2.2. Trích lớp thực thể

- Các đối tượng thực thể cần xử lý:

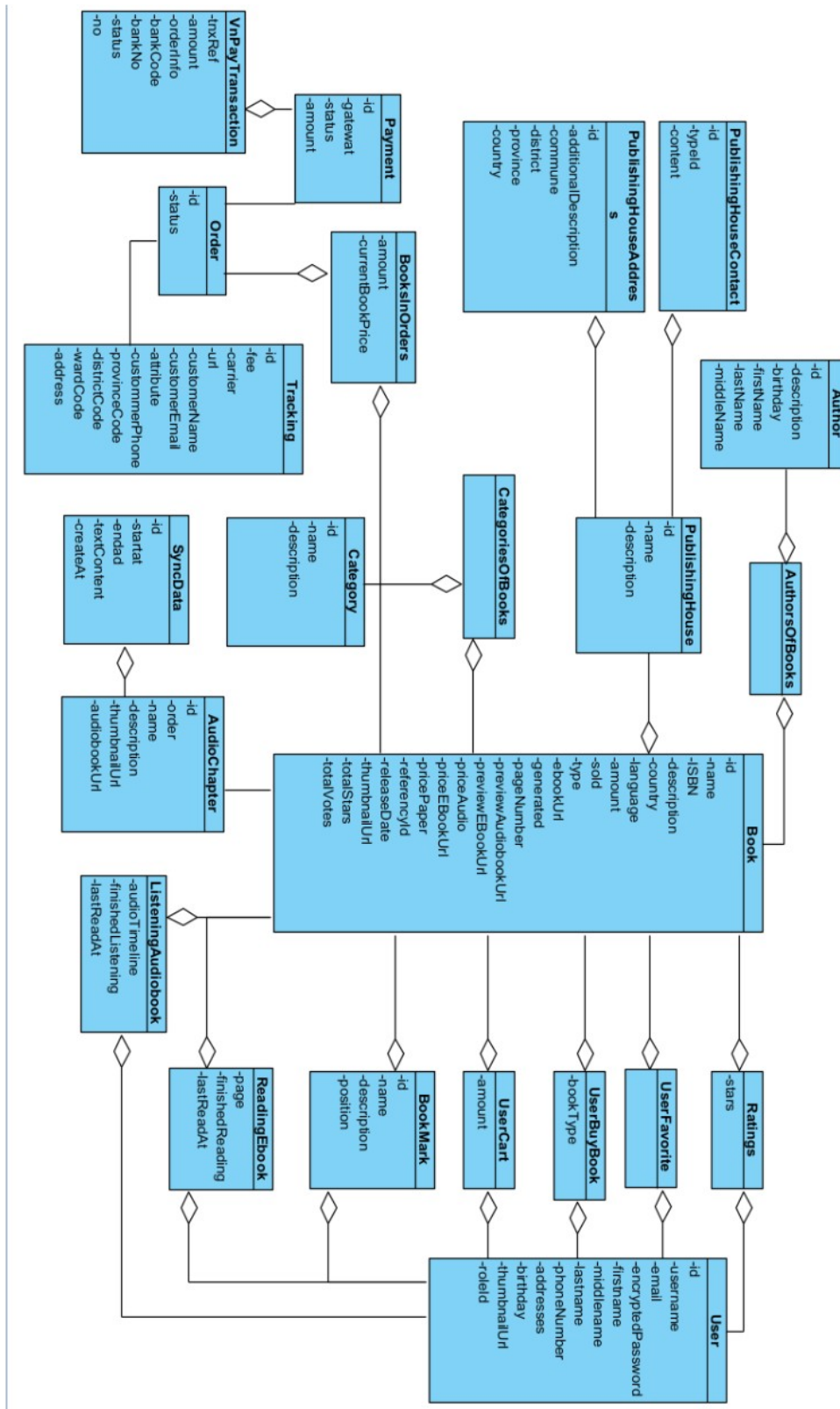
- Người dùng → Lớp thực thể User.
- Quyền sách → Lớp thực thể Book.
- Tác giả → Lớp thực thể Author.
- Nhà xuất bản → Lớp thực thể Publishing house.
- Đơn hàng → Lớp thực thể Order.
- Sách yêu thích → Lớp thực thể User favorite.
- Giỏ hàng → Lớp thực thể User cart.
- Danh mục → Lớp thực thể Category.
- Dấu sách → Lớp thực thể Book mark.
- Đánh giá → Lớp thực thể Ratings.
- Thanh toán → Lớp thực thể Payment.
- Theo dõi đơn hàng → Lớp thực thể Tracking.

- Quan hệ số lượng giữa các lớp thực thể

- Một Book có thể được viết bởi nhiều Authors, một Author có thể viết nhiều Books nên là quan hệ n - n, tách thành một lớp kết nối giữa Books và Authors.
- Một Book có thể được mua bởi nhiều Users, một Users có thể mua nhiều Books nên là quan hệ n - n, tách thành một lớp kết nối mua sách giữa Books và Users.
- Một Book có thể nằm trong nhiều Orders, một Order có thể chứa nhiều Books nên là quan hệ n - n, tách thành một lớp kết nối giữa Books và Orders.

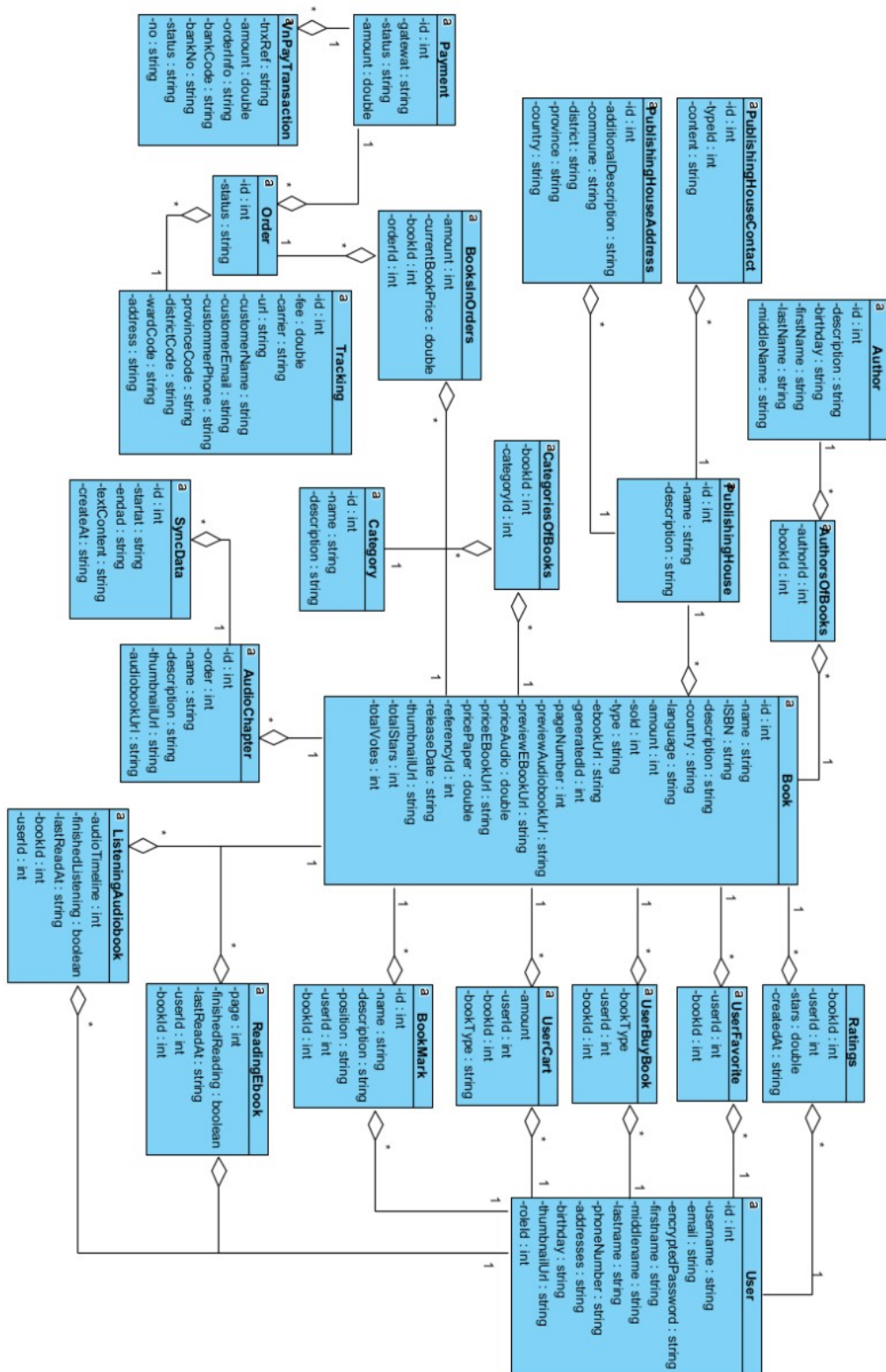
- Một Book có thể có nhiều Categories và một Category có thể có trong nhiều Books nên là quan hệ n-n, tách thành một lớp kết nối giữa Books và Category.
- Một Book có thể được đọc/nghe bởi nhiều Users, một User có thể đọc/nghe nhiều Books nên là quan hệ n-n, nên tách thành các lớp kết nối.
- Một Book có thể được đánh giá bởi nhiều Users, một User có thể đánh giá nhiều Books nên là quan hệ n-n, tách thành một lớp kết nối.
- Một Book có thể được yêu thích bởi nhiều Users, một Users có thể yêu thích nhiều Books nên là quan hệ n-n, tách thành một lớp kết nối.
- Một Book có thể được đọc bởi nhiều Users, một Users có thể đọc nhiều Books nên là quan hệ n-n, tách thành một lớp kết nối.

2.3. Biểu đồ lớp thực thể pha phân tích



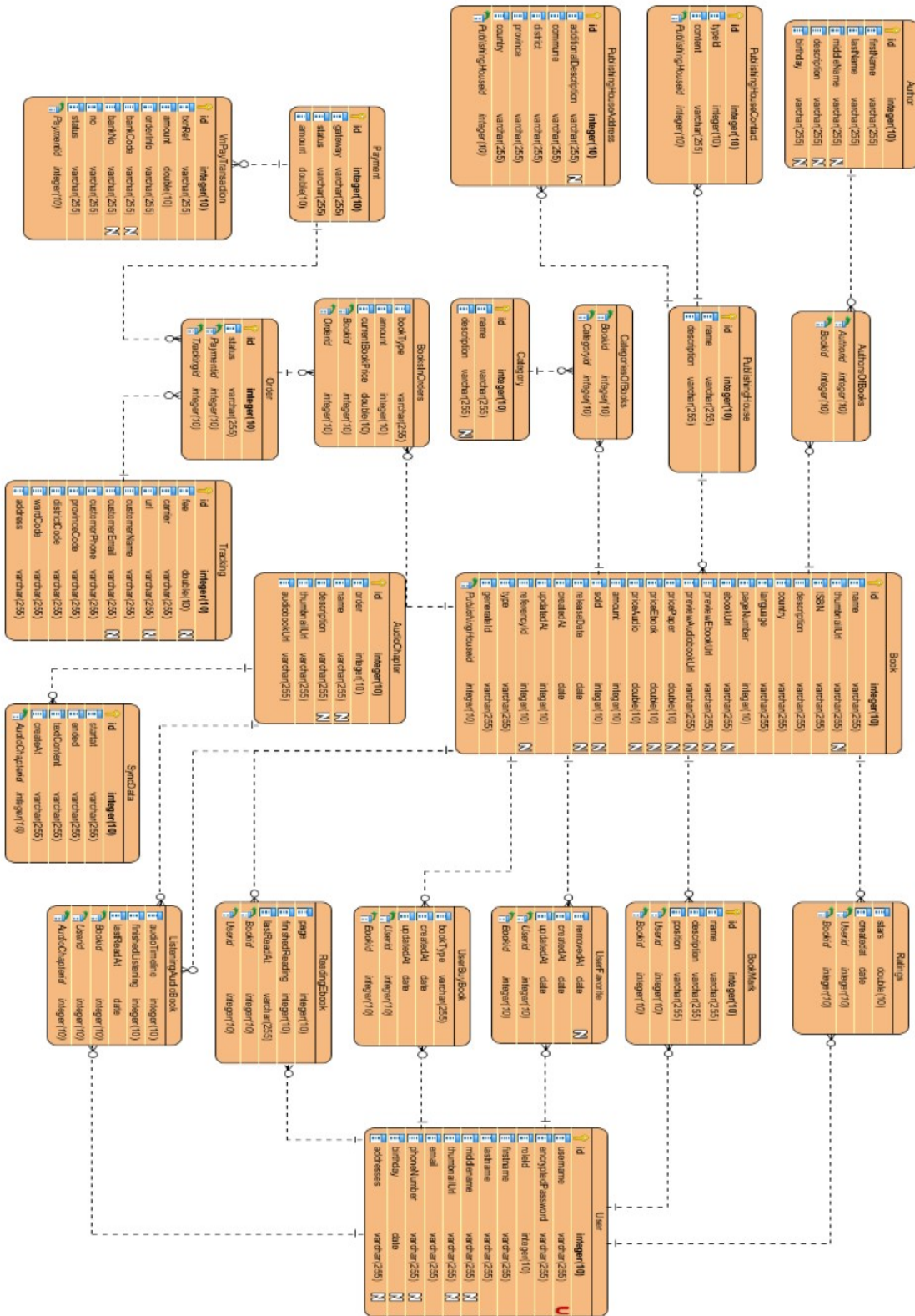
Hình 2.11: Biểu đồ lớp thực thể pha phân tích

2.4. Biểu đồ lớp thực thể pha thiết kế



Hình 2.12: Biểu đồ lớp thực thể pha thiết kế

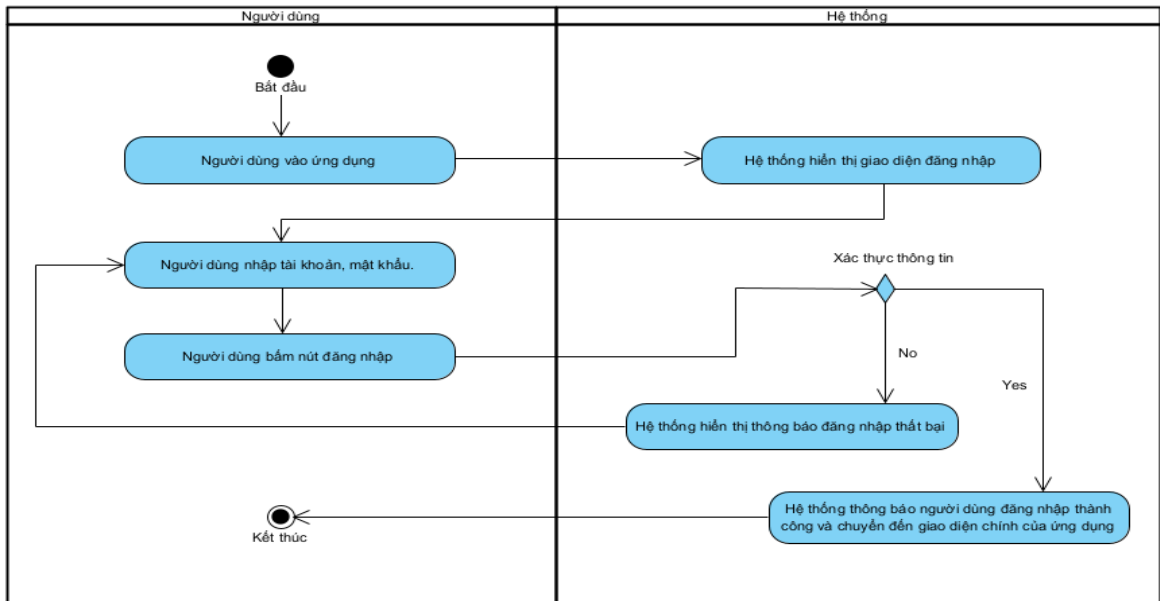
2.5. Thiết kế Database



Hình 2.13: Cơ sở dữ liệu

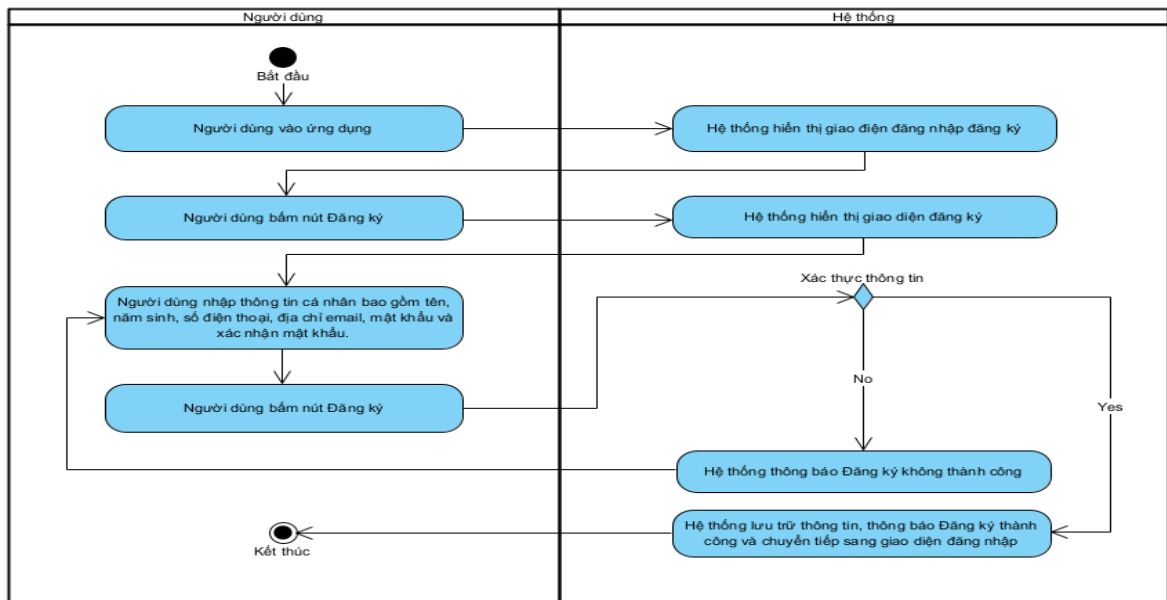
2.6. Activity Diagram

2.6.1. Biểu đồ hoạt động Module Đăng nhập



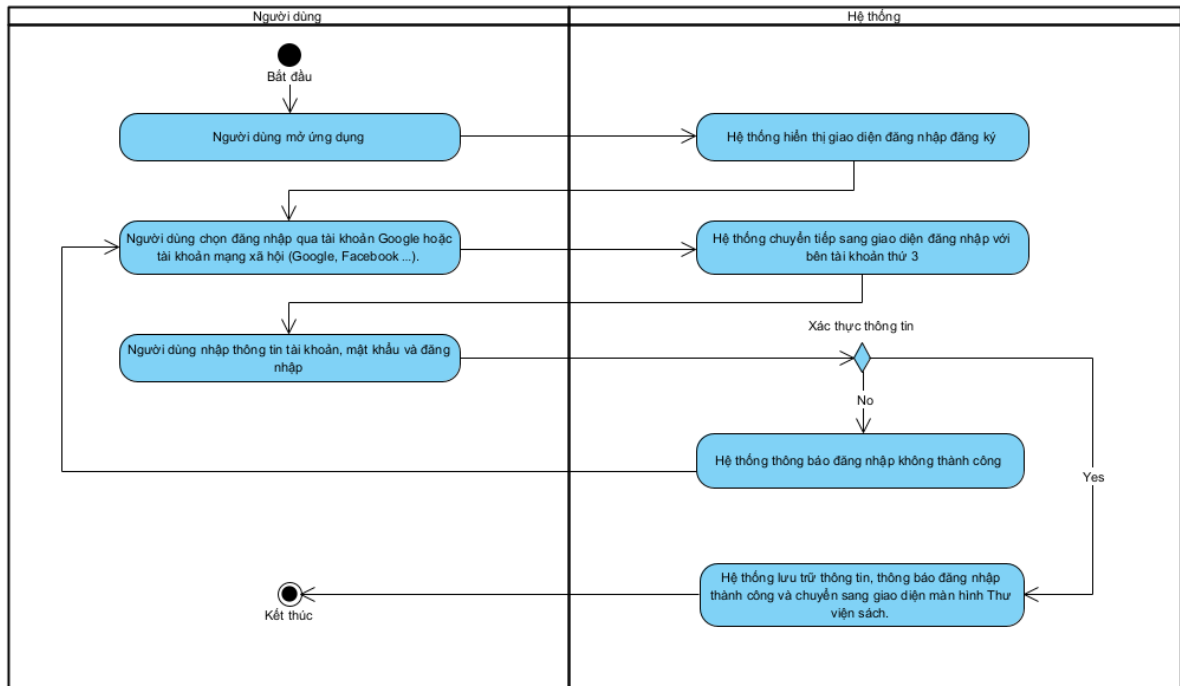
Hình 2.14: Biểu đồ hoạt động module Đăng nhập

2.6.2. Biểu đồ hoạt động Module Đăng ký



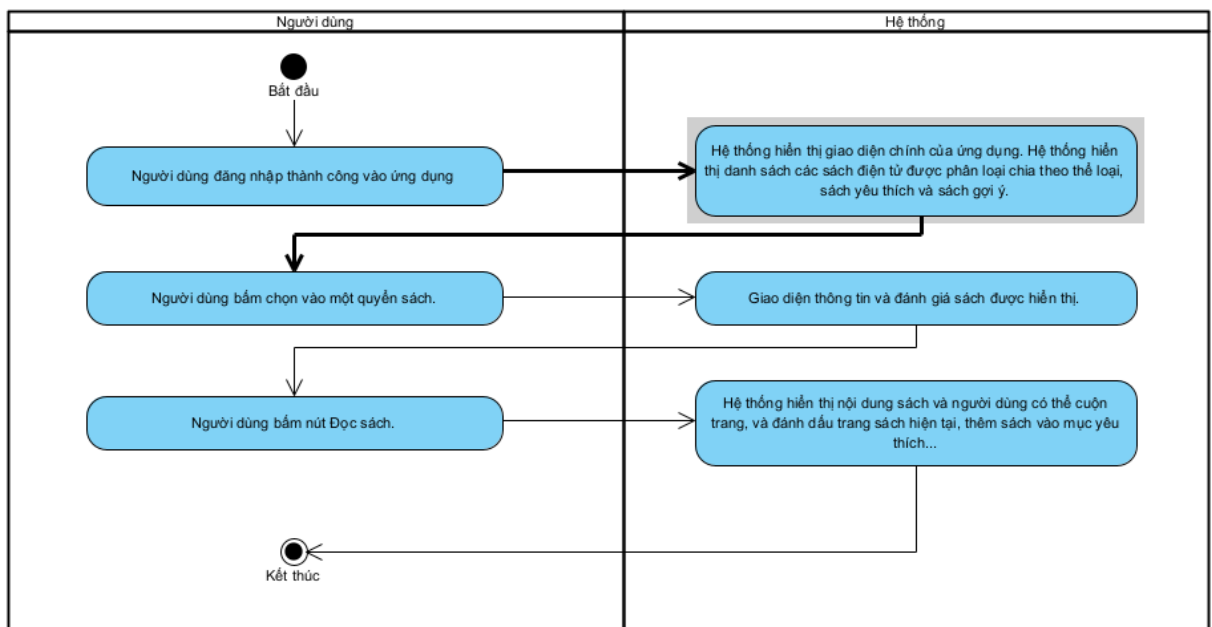
Hình 2.15: Biểu đồ hoạt động module Đăng ký

2.6.3. Biểu đồ hoạt động Module Đăng nhập tài khoản với bên thứ 3



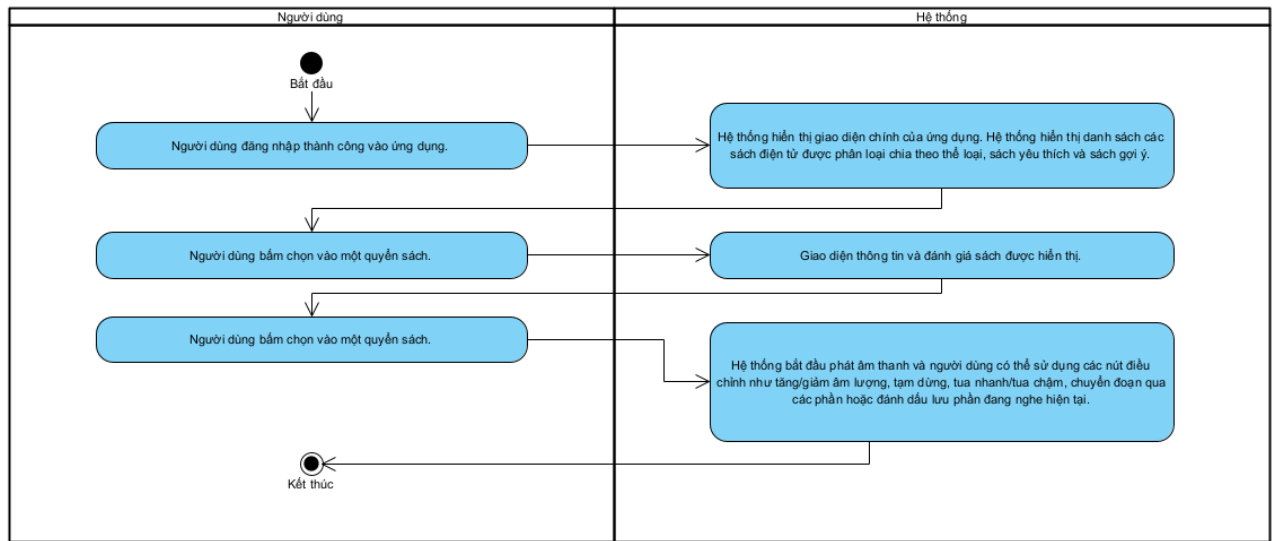
Hình 2.16: Biểu đồ hoạt động module Đăng nhập tài khoản với các bên thứ 3

2.6.4. Biểu đồ hoạt động Module Đọc sách điện tử



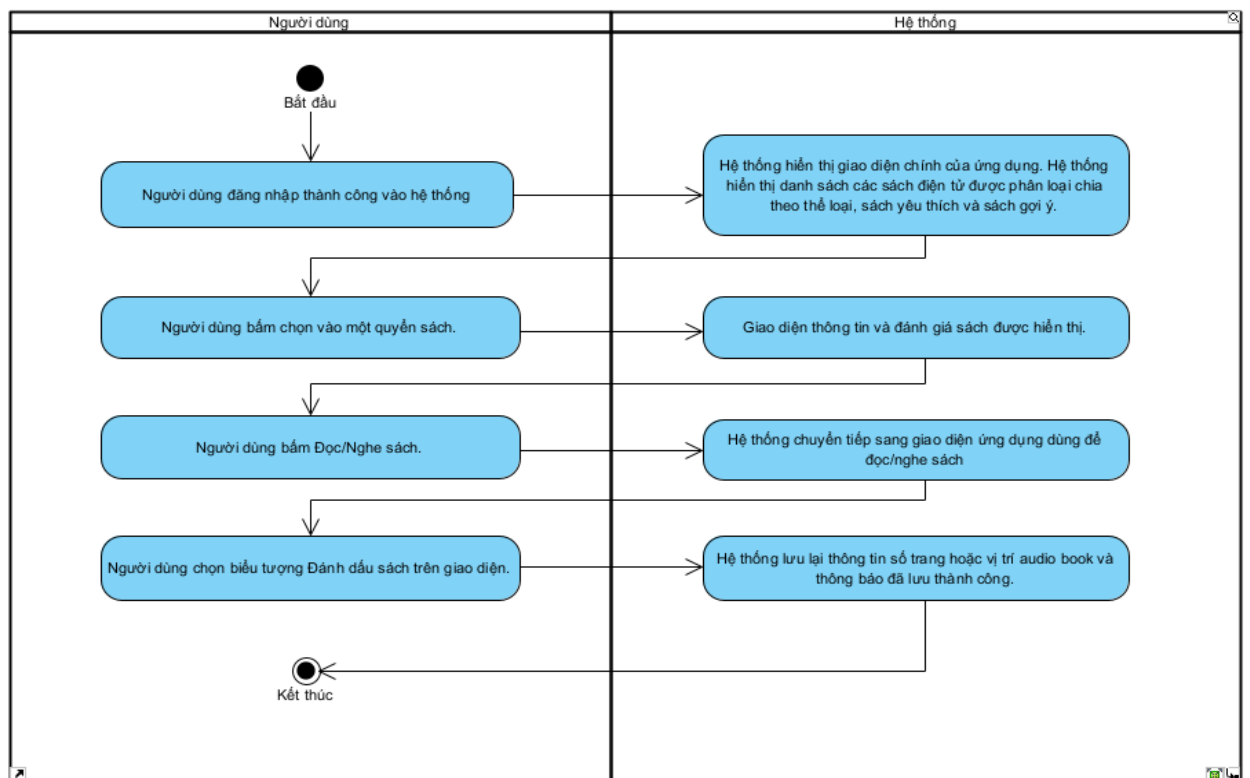
Hình 2.17: Biểu đồ hoạt động module Đọc sách điện tử

2.6.5. Biểu đồ hoạt động Module Nghe sách trực tuyến



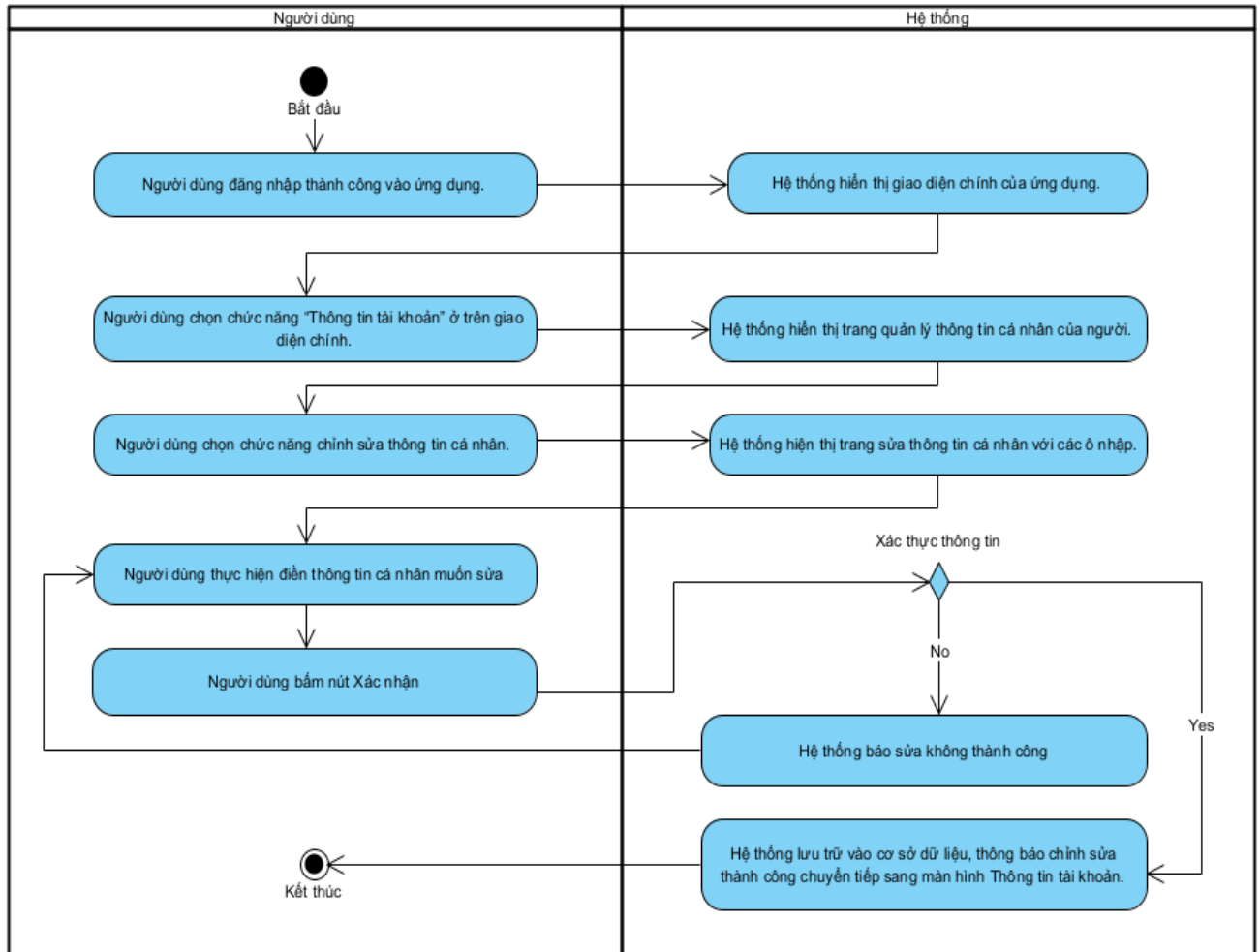
Hình 2.18: Biểu đồ hoạt động module Nghe sách trực tuyến

2.6.6. Biểu đồ hoạt động Module Đánh dấu sách



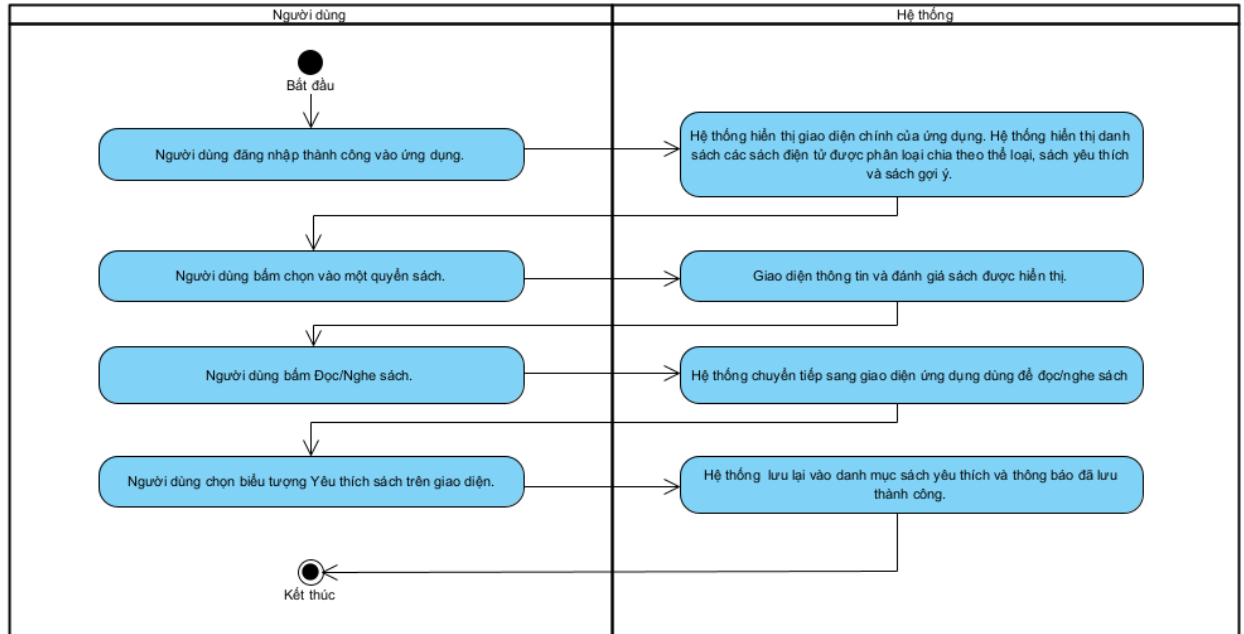
Hình 2.19: Biểu đồ hoạt động module Đánh dấu sách

2.6.7. Biểu đồ hoạt động Module *Chỉnh sửa thông tin tài khoản*



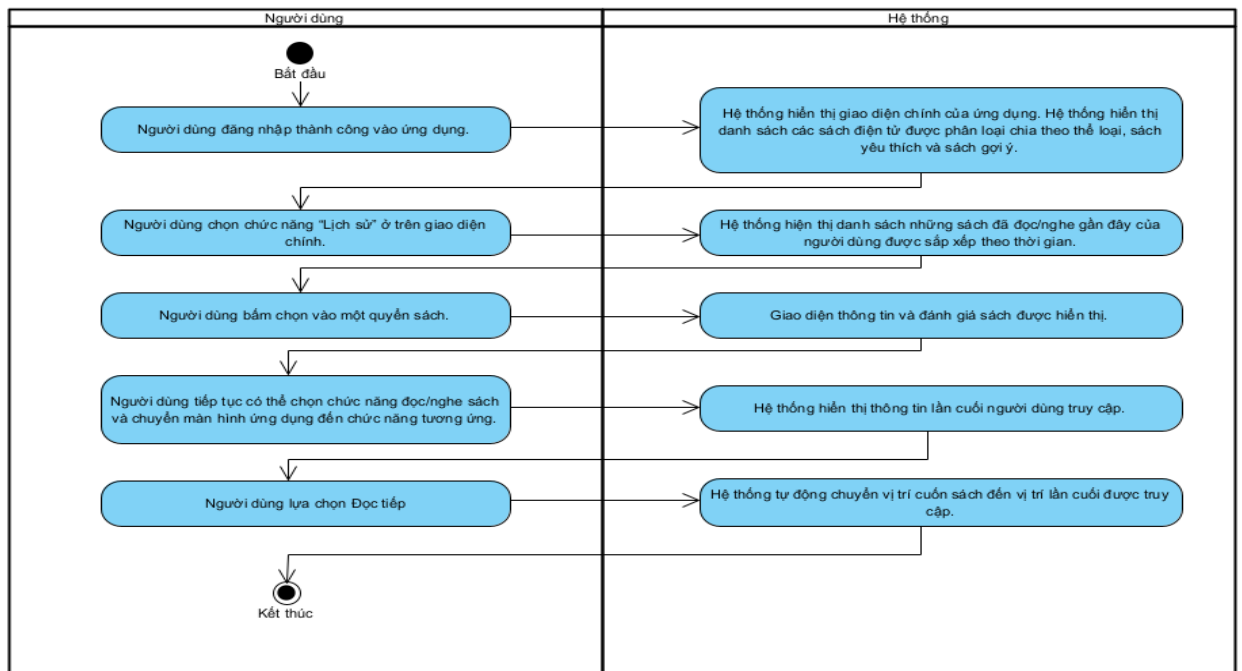
Hình 2.20: Biểu đồ hoạt động module *Chỉnh sửa thông tin tài khoản*

2.6.8. Biểu đồ hoạt động Module Thêm sách yêu thích



Hình 2.21: Biểu đồ hoạt động module Thêm sách yêu thích

2.6.9. Biểu đồ hoạt động Module Truy cập lịch sử



Hình 2.22: Biểu đồ hoạt động module Truy cập lịch sử sách

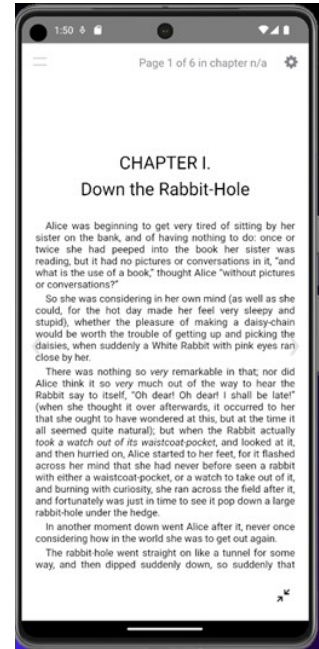
CHƯƠNG 3: PHÁT TRIỂN ỨNG DỤNG

1. PHÁT TRIỂN CHỨC NĂNG ĐỌC SÁCH ĐIỆN TỬ

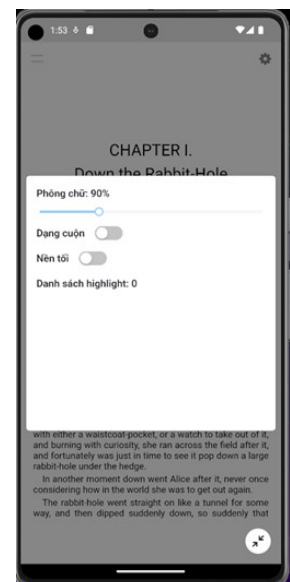
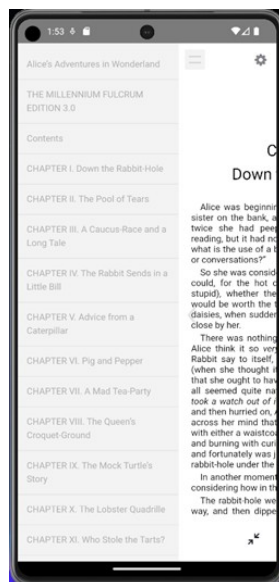
- Giao diện người dùng:



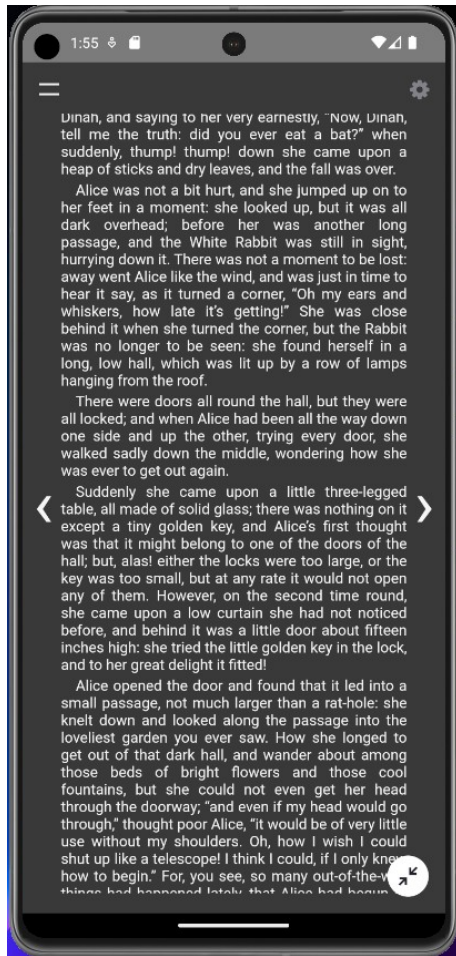
Hình 3.1: Giao diện Đọc sách điện tử (1)



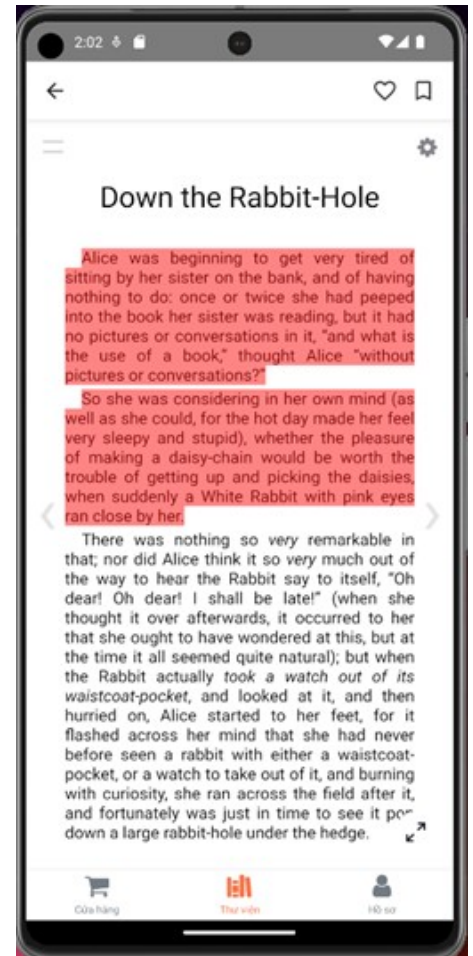
Hình 3.2: Giao diện Đọc sách điện tử (2)



Hình 3.3: Giao diện Đọc sách điện tử (3)



Hình 3.4: Giao diện Đọc sách điện tử (4)



Hình 3.4: Giao diện Đọc sách điện tử (5)

Hình 3.5: Giao diện Đọc sách điện tử (6)

- Sử dụng thư viện thứ 3 ReactReader để đọc các định dạng chuẩn file của sách điện tử như epub, pdf, ... Kết hợp các chức năng để hỗ trợ người dùng trong ứng dụng đọc sách như: Lựa chọn chương, Thay đổi phông chữ, Highlight trong sách, Thay đổi giữa đọc theo trang sách và kiểu cuộn, Nền sáng/tối ...

<ReactReader

```
url="https://react-reader.metabits.no/files/alice.epub"
readerStyles={theme === 'dark' ? darkReaderTheme :
lightReaderTheme}
location={location}
locationChanged={(loc: string) => {
  setLocation(loc)
  window.localStorage.setItem('location_book', loc);
  if (renditionRef.current && toc.current) {
```

```

        const { displayed, href } =
        renditionRef.current.location.start
        const chapter = toc.current.find((item) => item.href
        === href)
        setPage(`Page ${displayed.page} of ${displayed.total}
        in chapter ${chapter ? chapter.label : 'n/a'
        }`
        )
    }
    }}
    tocChanged={({_toc: NavItem[]}) => (toc.current = _toc)}
    getRendition={({_rendition: Rendition}) => {
        renditionRef.current = _rendition
        setRendition(_rendition);
        _rendition.hooks.content.register((contents: Contents) => {
            const body =
contents.window.document.querySelector('body')
            if (body) {
                body.oncontextmenu = () => {
                    return false
                }
            }
        })
        renditionRef.current?.themes.fontSize(`$
{fontSizePer}%`);
    }}

```

```

        epubOptions={getLocalStorage('is_scroll', false) ? { flow:
        'scrolled', manager: 'continuous' } : {}}
    />

```

- API GET lấy thông tin của các sách điện tử với tham số là userId để có thể truy cập vào database lấy những sách điện tử đã được người dùng mua và thêm vào thư viện.

```

async findAllBookBuyByUser(userId: number) {
    const user = await this.prisma.user.findUnique({ where: { id:
    userId } });
    if (!user) {
        return new FailureResponse(
            STATUS_CODE.NOT_FOUND,
            'User ' + RESPONSE_MESSAGE.NOT_FOUND,
        );
    }
    const books = await this.prisma.userBuyBook.findMany({
        where: {
            userId: userId,

```



```

    },
    include: {
      book: {
        include: {
          authorsBooks: {
            include: {
              author: true,
            },
          },
        },
      },
    },
  },
});
if (!books) {
  return new FailureResponse(
    STATUS_CODE.NOT_FOUND,
    'Books ' + RESPONSE_MESSAGE.NOT_FOUND,
  );
}
return new SuccessResponseArray(STATUS_CODE.OK,
RESPONSE_MESSAGE.OK, books);
}

```

Response body

```

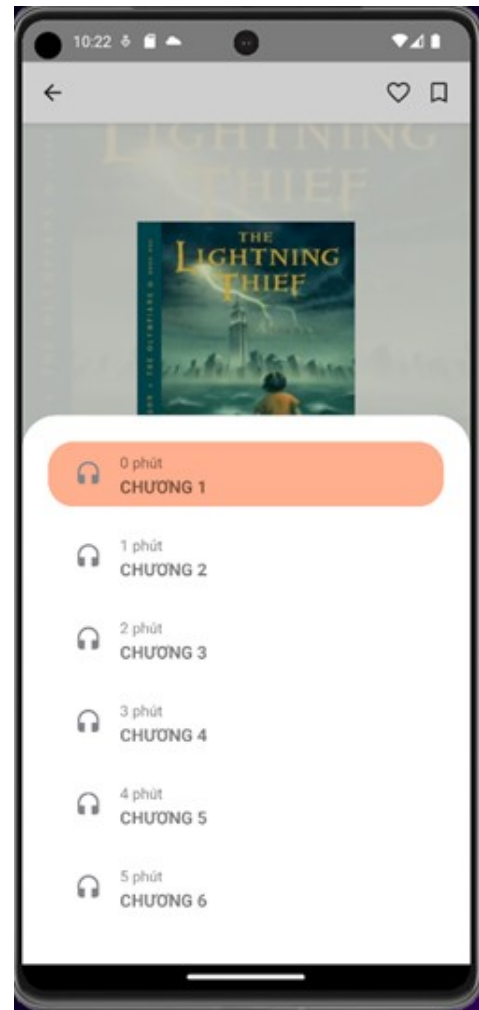
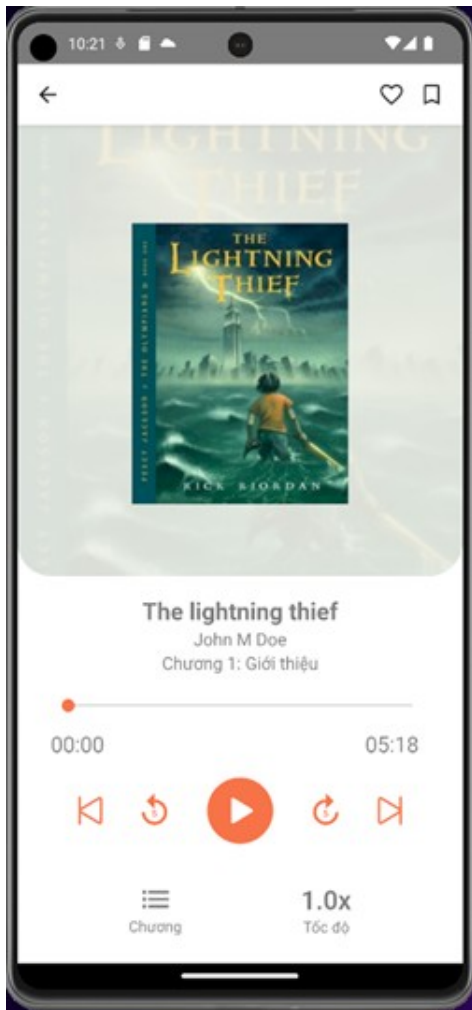
{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "count": 4,
    "items": [
      {
        "userId": 4,
        "bookId": 1,
        "bookType": "EBOOK",
        "createdAt": "2023-01-01T00:00:00.000Z",
        "updatedAt": "2023-11-16T16:56:25.881Z",
        "book": {
          "id": 1,
          "publishingHouseId": 1,
          "name": "The lightning thief",
          "thumbnailUrl": "https://archive.org/services/img/pdfy-vJFcaYG7Xau9y5ZZ",
          "ISBN": "123",
          "description": "Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature, going back to 45 BC. Lorem Ipsum is the only Lorem Ipsum text on the Internet, and it has been the standard Lorem Ipsum text since the 1960s. This is the first line of Lorem Ipsum, written by the most famous living Lorem Ipsum writer, Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from the dictionary and found the root of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of 'de Finibus Bonorum et Malorum' by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, 'Lorem ipsum dolor sit amet...', has been repeated ad nauseam ever since. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of Garamont type, set it type and printed the first Lorem Ipsum specimen. Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature, going back to 45 BC. Lorem Ipsum is the only Lorem Ipsum text on the Internet, and it has been the standard Lorem Ipsum text since the 1960s. This is the first line of Lorem Ipsum, written by the most famous living Lorem Ipsum writer, Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from the dictionary and found the root of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of 'de Finibus Bonorum et Malorum' by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, 'Lorem ipsum dolor sit amet...', has been repeated ad nauseam ever since. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of Garamont type, set it type and printed the first Lorem Ipsum specimen.",
          "country": "VN",
          "language": "VN",
          "pageNumber": 210,
          "ebookUrl": null,
          "previewEbookUrl": null,
        },
      }
    ]
  }
}

```

Hình 3.6: Ví dụ kết quả trả về sau lấy thông tin sách điện tử thành công

2. PHÁT TRIỂN CHỨC NĂNG NGHE SÁCH TRỰC TUYẾN

- Chức năng nghe sách trực tuyến: Người dùng có thể chọn chức năng audiobook từ thư viện giao diện chính của ứng dụng và nghe nó như một tệp âm thanh. Người dùng có thể điều chỉnh âm lượng, tạm dừng, tua nhanh, tua chậm và chọn từng phần muốn nghe tùy ý. Ngoài ra người dùng có thể tạm dừng và lưu lại phần đang nghe.
- Giao diện người dùng:



Hình 3.7: Giao diện Nghe sách trực tuyến (1) Hình 3.8: Giao diện Nghe sách trực tuyến (2)

- Sử dụng thư viện thứ 3 React Native Sound để đọc các định dạng chuẩn file của audio book như MP3, AAC, ID3... Kết hợp các chức năng để hỗ trợ người dùng trong ứng dụng đọc sách như: Lựa chọn chương, Thay đổi tốc độ, Tua nhanh chậm...

```
// Load the sound file when the component mounts
const newSound = new Sound('audio1.mp3', Sound.MAIN_BUNDLE, error
=> {
  if (error) {
    console.log('Error loading sound: ', error);
    return;
  }
  setSound(newSound);
});

// Release the sound and clear the interval when the component
unmounts
return () => {
  if (newSound) {
    newSound.release();
  }
};
}, []);
```

- Tương tự đối với đọc sách điện tử, API GET lấy thông tin của các sách điện tử với tham số là userId để có thể truy cập vào database lấy những sách điện tử đã được người dùng mua và thêm vào thư viện.

```
async findAllBookBuyByUser(userId: number) {
  const user = await this.prisma.user.findUnique({ where: { id:
userId } });
  if (!user) {
    return new FailureResponse(
      STATUS_CODE.NOT_FOUND,
      'User ' + RESPONSE_MESSAGE.NOT_FOUND,
    );
  }
  const books = await this.prisma.userBuyBook.findMany({
    where: {
      userId: userId,
    },
    include: {
      book: {
        include: {
```

```

        authorsBooks: {
            include: {
                author: true,
            },
        },
    },
},
});
if (!books) {
    return new FailureResponse(
        STATUS_CODE.NOT_FOUND,
        'Books ' + RESPONSE_MESSAGE.NOT_FOUND,
    );
}
return new SuccessResponseArray(STATUS_CODE.OK,
RESPONSE_MESSAGE.OK, books);
}

```

Response body

```

{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "count": 4,
    "items": [
      {
        "userId": 4,
        "bookId": 1,
        "bookType": "EBOOK",
        "createdAt": "2023-01-01T00:00:00.000Z",
        "updatedAt": "2023-11-16T16:56:25.881Z",
        "book": {
          "id": 1,
          "publishingHouseId": 1,
          "name": "The lightning thief",
          "thumbnailUrl": "https://archive.org/services/img/pdfy-vJFcaYG7Xau9y5ZZ",
          "ISBN": "123",
          "description": "Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature, discovered by William
          chard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from the
          f the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of 'de Finibus Bonorum et Malorum'
          by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, 'Lorem ipsum
          line in section 1.10.32",
          "country": "VN",
          "language": "VN",
          "pageNumber": 210,
          "ebookUrl": null,
          "previewEbookUrl": null,
        }
      }
    ]
  }
}

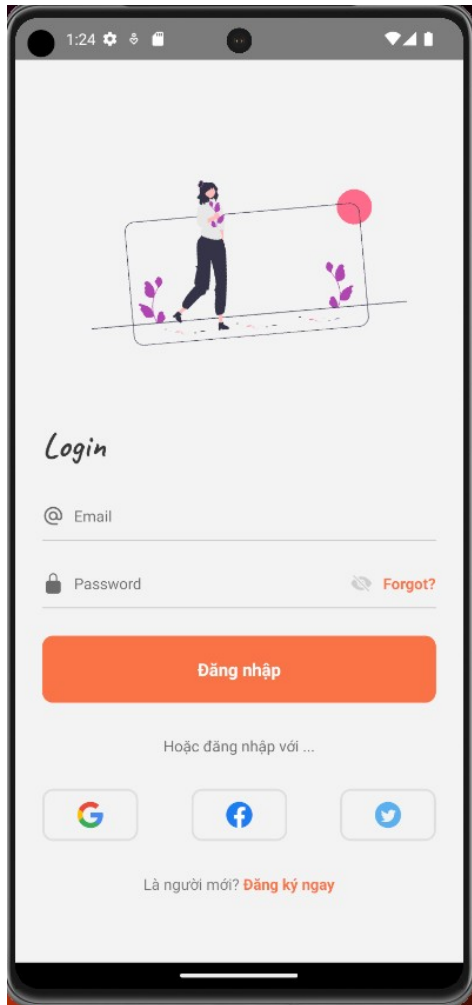
```

Hình 3.9: Ví dụ kết quả trả về sau lấy thông tin sách điện tử thành công

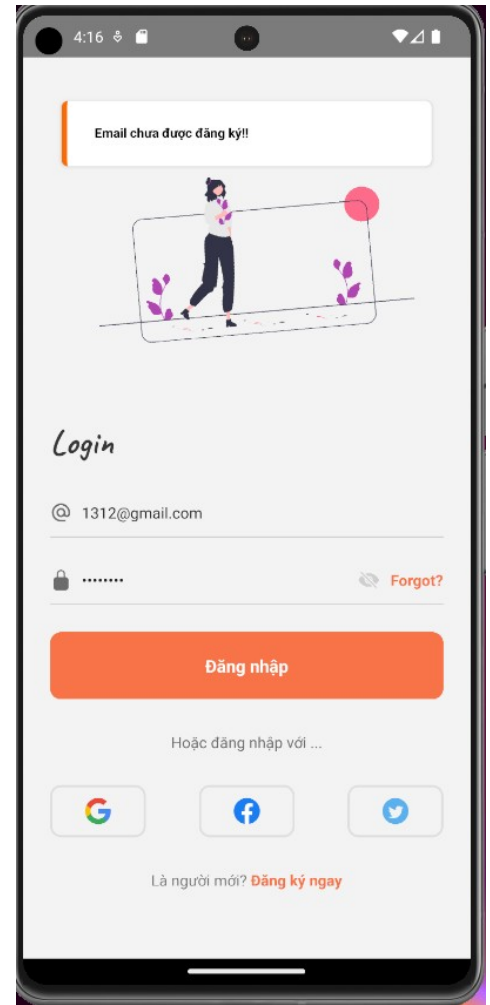
3. PHÁT TRIỂN CHỨC NĂNG ĐĂNG KÝ, ĐĂNG NHẬP

3.1. Chức năng Đăng nhập

- Chức năng Đăng nhập: Cho phép người dùng sử dụng tài khoản đã được đăng ký trong hệ thống để đăng nhập vào ứng dụng.
- Giao diện người dùng:



Hình 3.10: Giao diện Đăng nhập (1)



Hình 10.11: Giao diện Đăng nhập (2)

- Người dùng điền đầy đủ thông tin và nhấn nút đăng nhập. Hàm "handleLogin" trong ứng dụng sẽ xử lý xác minh các thông tin, sau khi xác nhận thành công sẽ gửi API POST Login đến server yêu cầu đăng nhập.
- Sau khi server nhận được yêu cầu từ API Login sẽ xử lý các thông tin và truy xuất vào database để có thể trả về kèm theo thông tin của "User" và các "token" theo chuẩn

JWT.

```

async login(email: string, password: string): Promise<AuthEntity> {
  let user = await this.prisma.user.findFirst({ where: { email } });
  if (!user) {
    throw new NotFoundException(`Email chưa được đăng ký!!`);
  }
  const isPasswordValid = await bcrypt.compare(
    password,
    user.encryptedPassword,
  );
  if (!isPasswordValid) {
    throw new UnauthorizedException('Mật khẩu không đúng!!');
  }
  delete user.encryptedPassword;
  return {
    accessToken: this.jwtService.sign(
      { userId: user.id, role: user.roleId },
      { expiresIn: '1d' },
    ),
    refreshToken: this.jwtService.sign(
      { userId: user.id, role: user.roleId },
      { expiresIn: '7d' },
    ),
    user: user,
  };
}

```

Response body

```

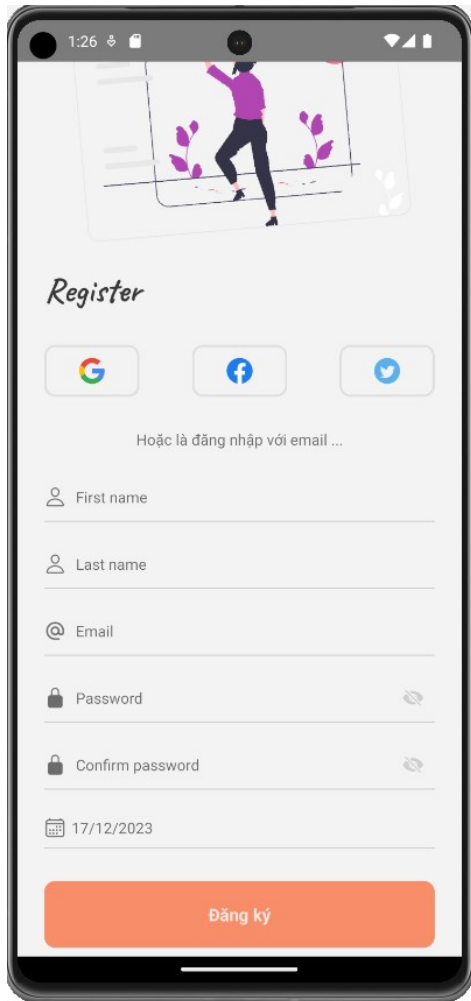
{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmVudGUiOiJlbnV0VSIiwiaWF0IjoxNzAyOTc3MTAyLCJleHAiOjE3MDM1ODU0E3",
    "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmVudGUiOiJlbnV0VSIiwiaWF0IjoxNzAyOTc3MTAyLCJleHAiOjE3MDM1ODU0E3",
    "user": {
      "id": 4,
      "username": "binhbeobeu3@gmail.com",
      "roleId": "USER",
      "firstName": "Le",
      "lastName": "Thanh Binh",
      "middleName": null,
      "thumbnailUrl": null,
      "email": "binhbeobeu3@gmail.com",
      "phoneNumber": "1111",
      "birthday": "2001-08-15T17:00:00.000Z",
      "createdAt": "2023-11-17T07:57:06.707Z",
      "updatedAt": "2023-11-24T08:53:29.321Z",
      "addresses": null
    }
  }
}

```

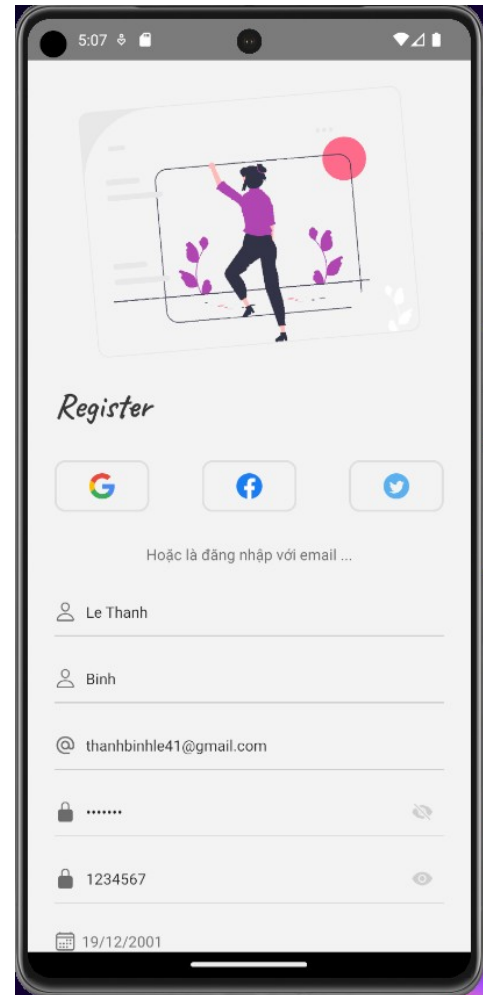
Hình 3.12: Ví dụ kết quả trả về sau khi đăng nhập thành công

3.2. Chức năng Đăng ký

- Chức năng đăng ký: Người dùng mới có thể tạo tài khoản bằng cách cung cấp tên, năm sinh, số điện thoại, địa chỉ email hợp lệ và mật khẩu an toàn. Chức năng này cho phép người dùng truy cập vào ứng dụng và sử dụng được đầy đủ các chức năng của ứng dụng.
- Giao diện người dùng:



Hình 3.13: Giao diện Đăng ký (1)

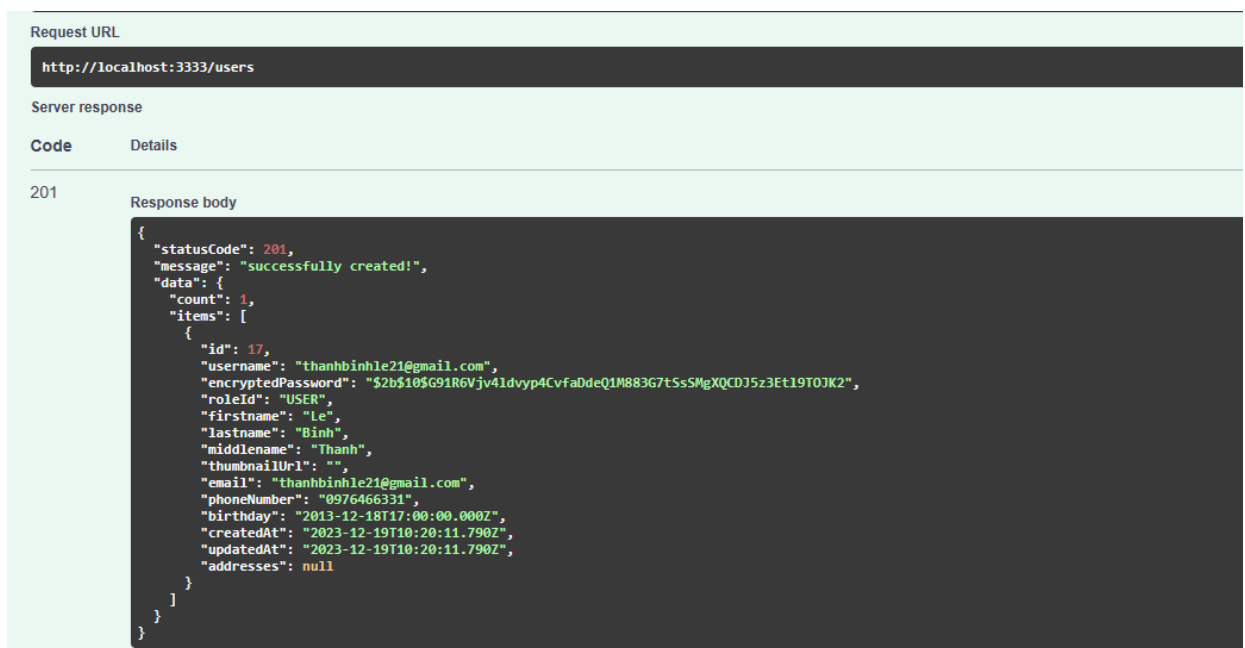


Hình 3.14: Giao diện Đăng ký (2)

- Người dùng điền đầy đủ thông tin và nhấn nút đăng ký. Hàm "handleSignUp" trong ứng dụng sẽ xử lý xác minh các thông tin, sau khi xác nhận thành công sẽ gửi API POST SignUp đến server yêu cầu đăng ký tài khoản mới trong hệ thống.
- Sau khi server nhận được yêu cầu từ API SignUp sẽ xử lý các thông tin và tạo mới

user để lưu vào database sau đó trả về kèm theo thông tin của “User”.

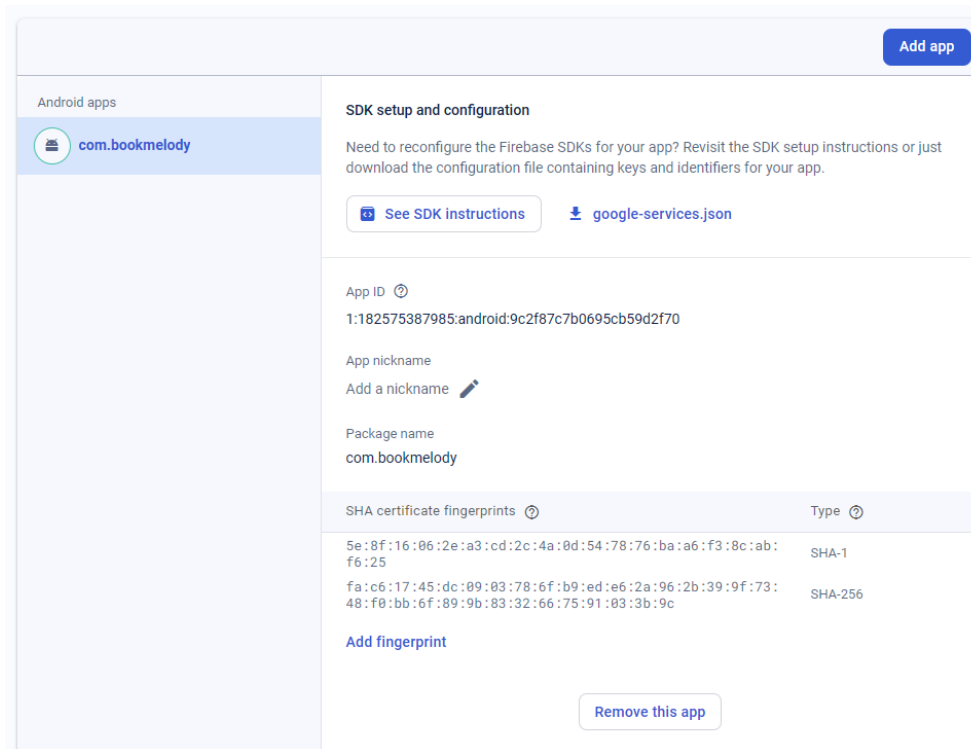

```
async createNewUser(createUserRequest: CreateUserRequest) {  
  const userDto = await new CreateUserDto().ParseFromRequest(  
    createUserRequest,  
  );  
  let user;  
  try {  
    user = await this.prisma.user.create({  
      data: {  
        ...userDto,  
      },  
    });  
  } catch (e: unknown) {  
    if (typeof e === 'string') {  
      console.log(e.toUpperCase());  
    } else if (e instanceof Error) {  
      console.log(e.message);  
    }  
  }  
  return user;  
}
```



Hình 3.15: Ví dụ kết quả trả về sau khi đăng ký thành công

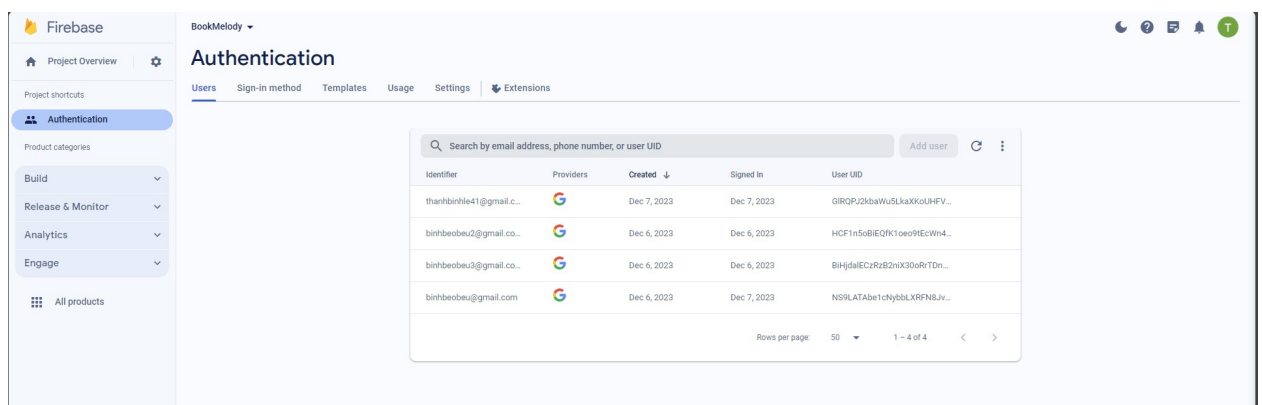
3.3. Chức năng Đăng nhập với tài khoản thứ 3

- Chức năng đăng nhập với tài khoản thứ 3: Người dùng mới có thể tạo tài khoản bằng cách sử dụng các tài khoản thứ 3 như Google, Facebook, ... để đăng ký vào tài khoản hệ thống. Chức năng này cho phép người dùng truy cập vào ứng dụng và sử dụng được đầy đủ các chức năng của ứng dụng.
- Firebase console trong Android apps



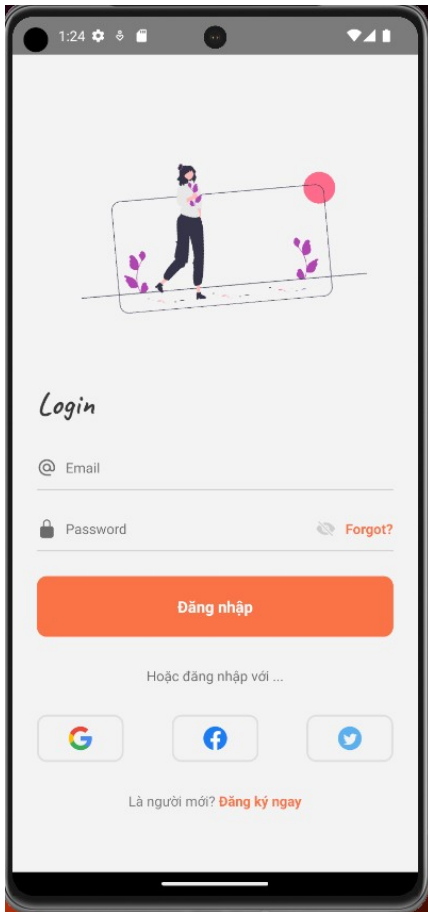
Hình 3.16: Firebase console (1)

- Các tài khoản Google đăng nhập thông qua Firebase

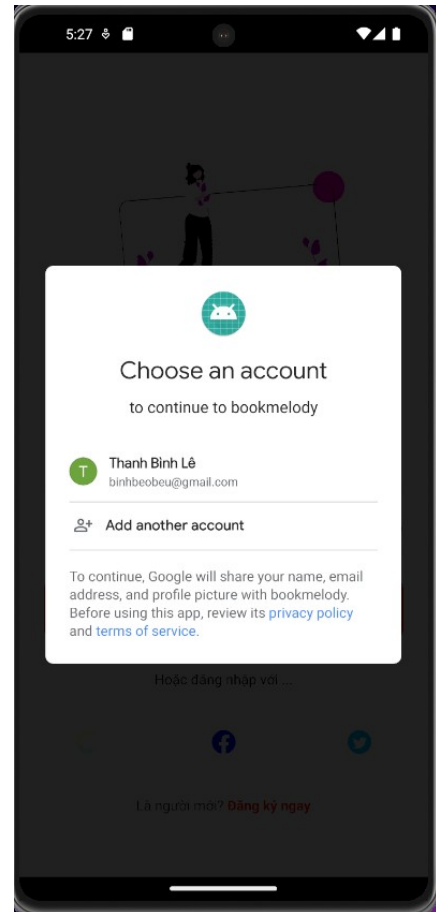


Hình 3.17: Firebase console (2)

- Giao diện người dùng:



Hình 3.18: Giao diện Đăng nhập



Hình 3.19: Giao diện đăng nhập bằng tài khoản Google

- Người dùng ấn vào icon Google sau đó ứng dụng sẽ gọi thư viện của Firebase để yêu cầu đăng nhập trên thiết bị di động thông qua hàm “signInGoogle”. Sau khi đăng nhập thành công Firebase sẽ lấy thông tin tài khoản Google gửi trả về cho ứng dụng.

```
export const signInGoogle = async () => {
  try {
    await GoogleSignin.hasPlayServices({showPlayServicesUpdateDialog: true});
    // Get the users ID token
    const {idToken} = await GoogleSignin.signIn();
    // Create a Google credential with the token
```

```

    const googleCredential =
auth.GoogleAuthProvider.credential(idToken);
    // Sign-in the user with the credential
    const user = await auth().signInWithCredential(googleCredential);
    return user;
  } catch (error: any) {
    if (error.code === statusCodes.SIGN_IN_CANCELLED) {
      // user cancelled the login flow
    } else if (error.code === statusCodes.IN_PROGRESS) {
      // operation (e.g. sign in) is in progress already
    } else if (error.code === statusCodes.PLAY_SERVICES_NOT_AVAILABLE)
{
      // play services not available or outdated
    } else {
      // some other error happened
      console.log(error);
    }
    return null;
  }
};

```

- Sau khi ứng dụng nhận được thông tin tài khoản của Google trả về từ Firebase, sẽ lấy thông tin tài khoản đó đăng nhập với hệ thống để có thể lấy được các JWT “token” vào ứng dụng.
- Server nhận được thông tin tài khoản Google sẽ thực hiện kiểm tra nếu chưa có sẽ lưu trữ thông tin vào trong database sau đó gửi trả lại về ứng dụng thông tin của “User” và các “token” theo chuẩn JWT.

```

async loginSocial(createUserRequest: CreateUserRequest) {
  const { email, password } = createUserRequest
  let user = await this.prisma.user.findFirst({ where: { email } });
  if (user) {
    const isPasswordValid = await bcrypt.compare(
      password,
      user.encryptedPassword,
    );
    if (!isPasswordValid) {
      throw new UnauthorizedException('Lỗi login tài khoản thứ 3!!!');
    }
    delete user.encryptedPassword;
    return {
      accessToken: this.jwtService.sign({ userId: user.id },
{ expiresIn: '1d' }),

```

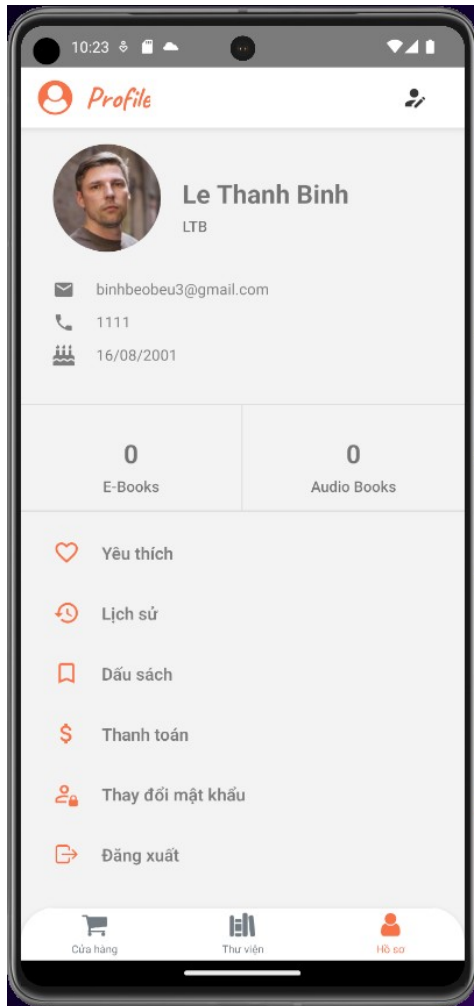
```
        refreshToken: this.jwtService.sign({ userId: user.id },
{ expiresIn: '7d' })),
        user: user,
    };
}
else {
    const userDto = await new
CreateUserDto().ParseFromRequest(createUserRequest);
    let user = null;
    try {
        user = await this.prisma.user.create({
            data: {
                ...userDto,
            },
        });
    } catch (e: unknown) {
        if (typeof e === 'string') {
            console.log(e.toUpperCase());
        } else if (e instanceof Error) {
            console.log(e.message);
        }
    }
    return {
        accessToken: this.jwtService.sign({ userId: user.id },
{ expiresIn: '1d' })),
        refreshToken: this.jwtService.sign({ userId: user.id },
{ expiresIn: '7d' })),
        user: user,
    };
}
}
```

```
Response body
{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJ0sInJvbGUiOiJVVU0VSIiwiaWF0IjoxNzAyOTc3MTAyLCJleHAiOjE3MDMwNjM1M",
    "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJ0sInJvbGUiOiJVVU0VSIiwiaWF0IjoxNzAyOTc3MTAyLCJleHAiOjE3MDM1ODE5",
    "user": {
      "id": 4,
      "username": "binhbeobeu3@gmail.com",
      "roleId": "USER",
      "firstname": "Le",
      "lastname": "Thanh Binh",
      "middlename": null,
      "thumbnailUrl": null,
      "email": "binhbeobeu3@gmail.com",
      "phoneNumber": "1111",
      "birthday": "2001-08-15T17:00:00.000Z",
      "createdAt": "2023-11-17T07:57:06.707Z",
      "updatedAt": "2023-11-24T08:53:29.321Z",
      "addresses": null
    }
  }
}
```

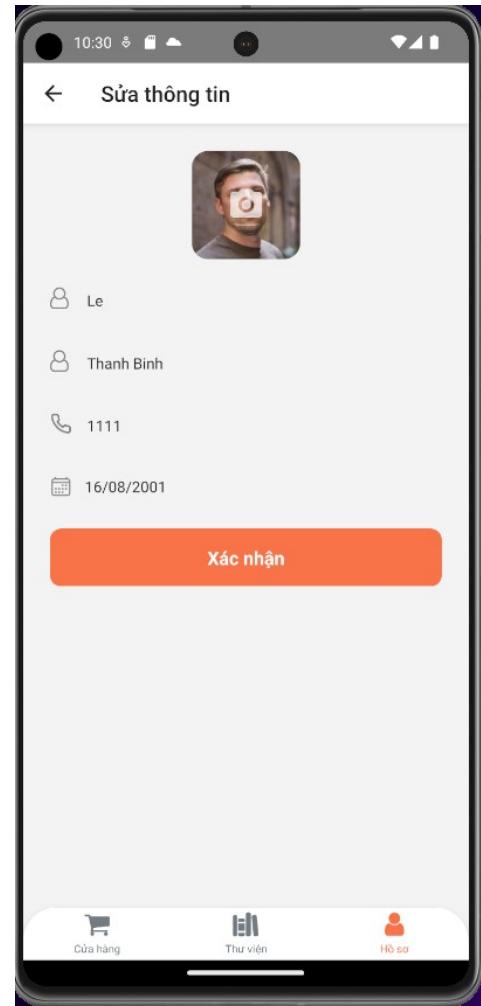
Hình 3.20: Ví dụ kết quả trả về sau khi đăng nhập thành công

4. PHÁT TRIỂN CHỨC NĂNG CHỈNH SỬA THÔNG TIN TÀI KHOẢN

- Giao diện người dùng:



Hình 3.21: Giao diện Thông tin tài khoản



Hình 3.22: Giao diện Chỉnh sửa thông tin

- Sau khi người dùng nhấp vào icon chỉnh sửa Hồ sơ trên thanh tác vụ, người dùng nhập đầy đủ các thông tin tài khoản cần được chỉnh sửa và bấm nút xác nhận. Ứng dụng sẽ gọi đến hàm “onEditProfile” để xác thực các thông tin sau đó gửi API Patch đến sever để yêu cầu chỉnh sửa.

```
const onEditProfile = async () => {
  if (!currentUser) {
    return;
  }
  if (profileData.firstname === '' && profileData.lastname === '') {
    showToast('info', 'Vui lòng nhập tên!!!');
  }
}
```

```

    return;
  }

  dispatch(loadingSliceActions.setIsLoading(true));
  const payload: Partial<IUser> = {
    ...profileData,
    id: currentUser.id,
    thumbnailUrl: selectedImage ? selectedImage :
profileData.thumbnailUrl,
  };
  const res = await dispatch(editMyProfileApi(payload));

  if (res.payload && res.payload.statusCode === 200) {
    dispatch(authSliceActions.setCurrentUser(res.payload.data));
    setTimeout(() => {
      dispatch(loadingSliceActions.setIsLoading(false));
      showToast('success', 'Sửa thông tin thành công!!');
      props.navigation.goBack();
    }, 500);
  }
};

```

- Server nhận được thông tin chỉnh sửa tài khoản sẽ xác thực thông tin và lưu vào database và trả về thông báo thành công cho ứng dụng.

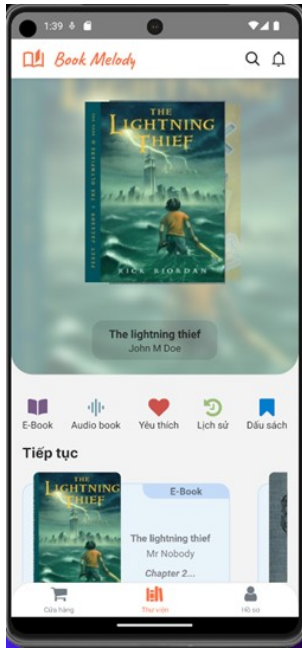
```

async update(id: number, updateUserDto: UpdateUserDto) {
  const newUser = await this.prisma.user
    .update({ where: { id: id }, data: updateUserDto })
    .catch((e) => {
      return new FailureResponse(
        STATUS_CODE.SERVER_INTERNAL_ERROR,
        RESPONSE_MESSAGE.SERVER_INTERNAL_ERROR,
        e,
      );
    });

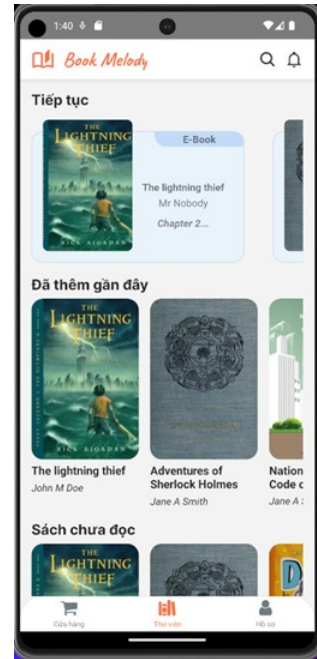
  if (!newUser) {
    return;
  }
  return new SuccessResponseItem(
    STATUS_CODE.OK,
    RESPONSE_MESSAGE.OK,
    newUser,
  );
}

```

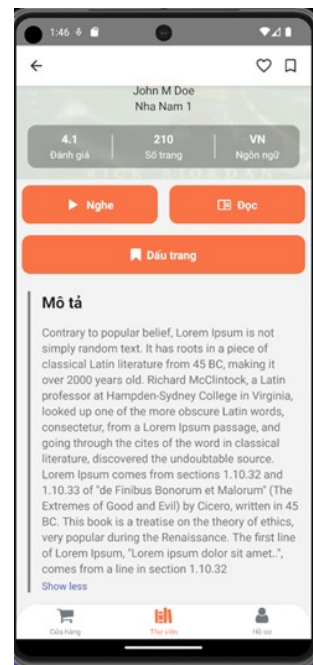
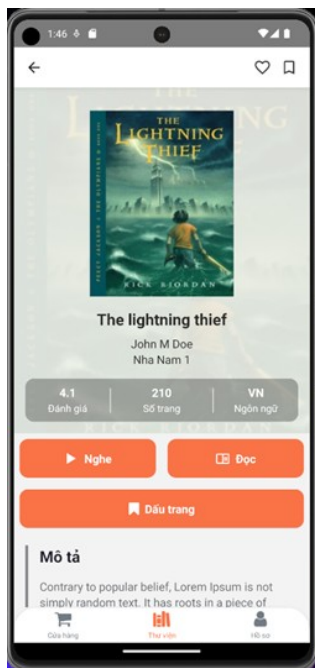

5. MỘT SỐ GIAO DIỆN CHỨC NĂNG KHÁC CỦA ỨNG DỤNG



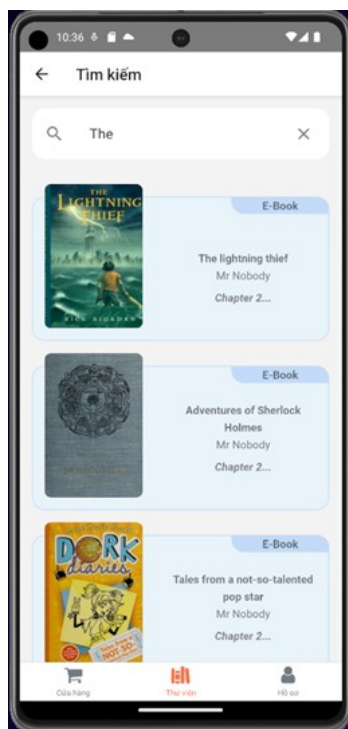
Hình 3.23: Giao diện Thư viện (1)



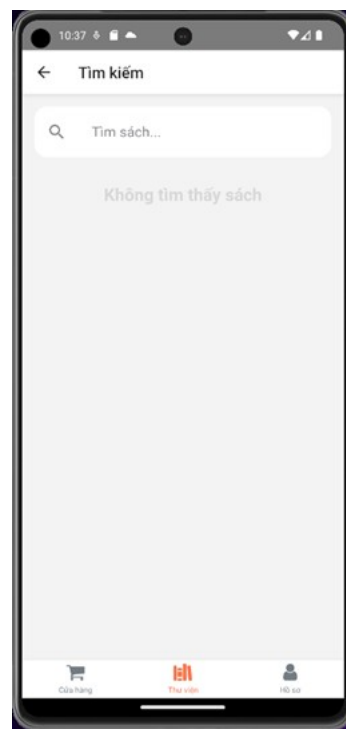
Hình 3.24: Giao diện Thư viện (2)



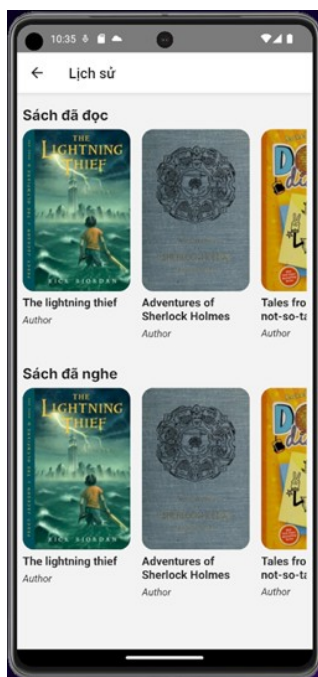
Hình 3.25: Giao diện Thông tin sách (1)



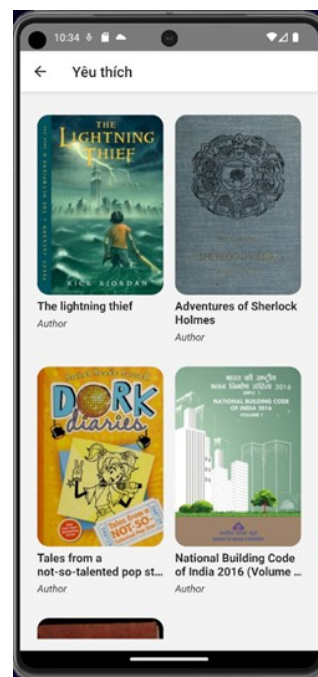
Hình 3.26: Giao diện Thông tin sách (2)



Hình 3.27: Giao diện Tìm kiếm sách (1)



Hình 3.28: Giao diện Tìm kiếm sách (2)



Hình 3.29: Giao diện Lịch sử truy cập

Hình 3.30: Giao diện Sách yêu thích

6. HỆ THỐNG KHUYẾN NGHỊ

Hệ thống khuyến nghị là một công cụ thông minh được thiết kế để cá nhân hóa trải nghiệm người dùng. Như đã trình bày về cơ sở lý thuyết ở Chương 1, tiếp theo hãy áp dụng hệ thống khuyến nghị vào ứng dụng nghe và đọc sách sau đây:

6.1. Các thư viện sử dụng và xử lý dữ liệu:

- **Pandas:** Thư viện cung cấp cấu trúc, công cụ phân tích và xử lý dữ liệu đặc biệt.
- **Numpy:** Sử dụng để tính toán, các mảng đa chiều và các chức năng hỗ trợ khác.
- **Matplotlib:** Thư viện vẽ biểu đồ, đồ thị và hình ảnh để trực quan hoá dữ liệu.
- **Cosine Similarity:** công cụ trong thư viện scikit-learn giúp tính toán độ tương đồng cosine giữa các vector. Trong trường hợp này, được sử dụng để đo độ tương đồng giữa các đặc trưng của các sách để xác định sự tương đồng giữa chúng.
- **Regular Expression:** Thư viện cung cấp công cụ để xử lý và phân tích các biểu thức chính quy trong Python.
- **PIL:** Là 1 thư viện mạnh mẽ để mở, chỉnh sửa và lưu trữ các hình ảnh.
- **Random:** Thư viện này cung cấp các công cụ để làm việc với số ngẫu nhiên.
- **CountVectorizer:** Là một công cụ trong scikit-learn giúp chuyển đổi một danh sách các văn bản thành một ma trận các đặc trưng dựa trên số lần xuất hiện của từng từ trong các văn bản.
- **NLTK:** Natural Language Toolkit là một thư viện cung cấp các công cụ và tài nguyên cho xử lý ngôn ngữ tự nhiên trong Python. Nó bao gồm các công cụ như tokenization, stopwords, và các nguồn dữ liệu cho việc phân tích ngôn ngữ.
- **Stopwords:** NLTK cung cấp danh sách các stopwords, các từ phổ biến như "và", "là", "của", không mang ý nghĩa đặc biệt trong việc phân tích ngôn ngữ, thường được loại bỏ trong quá trình xử lý văn bản.
- Có 3 bảng dữ liệu chính sau bao gồm Ratings, Users và Books.

| User-ID | ISBN | Book-Rating | User-ID | Location | Age |
|---------|--------|-------------|---------|------------------------------------|------|
| 0 | 276725 | 034545104X | 0 | nyc, new york, usa | NaN |
| 1 | 276726 | 0155061224 | 1 | stockton, california, usa | 18.0 |
| 2 | 276727 | 0446520802 | 2 | moscow, yukon territory, russia | NaN |
| 3 | 276729 | 052165615X | 3 | porto, v.n.gaia, portugal | 17.0 |
| 4 | 276729 | 0521795028 | 4 | farnborough, hants, united kingdom | NaN |

| ISBN | Book-Title | Book-Author | Year-Of-Publication | Publisher | Image-URL-S | |
|------|------------|---|----------------------|-----------|-------------------------|---|
| 0 | 0195153448 | Classical Mythology | Mark P. O. Morford | 2002 | Oxford University Press | http://images.amazon.com/images/P/0195153448.0... |
| 1 | 0002005018 | Clara Callan | Richard Bruce Wright | 2001 | HarperFlamingo Canada | http://images.amazon.com/images/P/0002005018.0... |
| 2 | 0060973129 | Decision in Normandy | Carlo D'Este | 1991 | HarperPerennial | http://images.amazon.com/images/P/0060973129.0... |
| 3 | 0374157065 | Flu: The Story of the Great Influenza Pandemic... | Gina Bari Kolata | 1999 | Farrar Straus Giroux | http://images.amazon.com/images/P/0374157065.0... |
| 4 | 0393045218 | The Mummies of Urumchi | E. J. W. Barber | 1999 | W. W. Norton & Company | http://images.amazon.com/images/P/0393045218.0... |

Hình 3.32: Cấu trúc dữ liệu từ file csv mẫu

- Clean data và xử lý dữ liệu Books và Ratings như loại bỏ dữ liệu thiếu, loại bỏ cột không cần thiết, xử lý tiêu đề sách và loại bỏ các đánh giá không hợp lý.

| Book-Title | Book-Author | Publisher | Image-URL-L | User-ID | Book-Rating |
|----------------|----------------------|-----------------------|---|---------|-------------|
| 1 Clara Callan | Richard Bruce Wright | HarperFlamingo Canada | http://images.amazon.com/images/P/0002005018.0... | 8 | 5 |
| 3 Clara Callan | Richard Bruce Wright | HarperFlamingo Canada | http://images.amazon.com/images/P/0002005018.0... | 11676 | 8 |
| 5 Clara Callan | Richard Bruce Wright | HarperFlamingo Canada | http://images.amazon.com/images/P/0002005018.0... | 67544 | 8 |
| 8 Clara Callan | Richard Bruce Wright | HarperFlamingo Canada | http://images.amazon.com/images/P/0002005018.0... | 116866 | 9 |
| 9 Clara Callan | Richard Bruce Wright | HarperFlamingo Canada | http://images.amazon.com/images/P/0002005018.0... | 123629 | 9 |

6.2. Most popular items

- Tính số lượt đánh giá và điểm đánh giá trung bình cho từng cuốn sách sau đó sẽ kết hợp thông tin đánh giá và đưa ra giá trị trung bình của điểm đánh giá và ngưỡng số lượt đánh giá đáng tin cậy. Kết quả đầu ra cuối cùng đó là những sách phổ biến nhất.
- Dựa trên công thức tính trọng số để có thể tính ra “*weighted_avg*” [10]:

$$weighted\ Avg = \frac{(v \cdot R) + (m \cdot C)}{v + m}$$

- o v là số lượt đánh giá cho sách.
- o R là điểm đánh giá trung bình của sách.

- C là điểm đánh giá trung bình của tất cả các sách trong tập dữ liệu.
- m là ngưỡng số lượng đánh giá đáng tin cậy (thường được chọn dựa trên phân vị nhất định của số lượt đánh giá).
- Xác định danh sách Sách phổ biến, chỉ số phổ biến dựa trên hàm `weighted_rate` (thường được sử dụng để đánh giá sự phổ biến của một sản phẩm dựa trên số lượng và chất lượng của đánh giá.) và kết quả được sắp xếp theo chỉ số này.

```

def popular_books(df, n=100):
    rating_count=df.groupby("BookTitle").count()["BookRating"]
    .reset_index()
    rating_count.rename(columns={"BookRating": "NumberOfVotes"},
inplace=True)

    rating_average=df.groupby("BookTitle")["BookRating"].mean()
    .reset_index()
    rating_average.rename(columns={"BookRating": "AverageRatings"},
inplace=True)

    popularBooks=rating_count.merge(rating_average, on="Book-Title")

    C=popularBooks["AverageRatings"].mean()
    m=popularBooks["NumberOfVotes"].quantile(0.90)

    def weighted_rate(x):
        v=x["NumberOfVotes"]
        R=x["AverageRatings"]

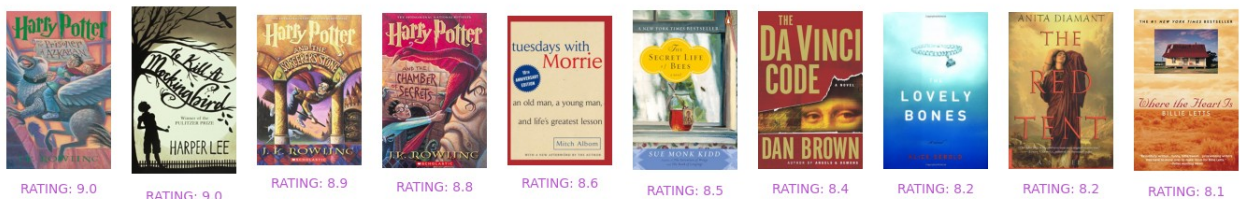
        return ((v*R) + (m*C)) / (v+m)

    popularBooks=popularBooks[popularBooks["NumberOfVotes"] >=250]
    popularBooks["Popularity"]=popularBooks.apply(weighted_rate, axis=1)

    popularBooks=popularBooks.sort_values(by="Popularity", ascending=False)
    return popularBooks[["Book-
Title", "NumberOfVotes", "AverageRatings", "Popularity"]].reset_index(drop
=True).head(n)

```

MOST POPULAR 10 BOOKS



Hình 3.33: Kết quả 10 sách phổ biến nhất

6.3. Item based

- Item-based collaborative filtering được sử dụng để dự đoán sở thích hoặc ưa thích của người dùng dựa trên lịch sử hoặc hành vi của họ trong việc sử dụng các sản phẩm hoặc dịch vụ. Do đó để tính toán độ tương tự giữa các cuốn sách cần sử dụng “*Pivot table*” trong tập dữ liệu hiện tại.
- Bảng pivot (pivot table) là một công cụ mạnh mẽ trong xử lý dữ liệu, thường được sử dụng để tóm tắt, phân tích và tổng hợp dữ liệu từ một bảng dữ liệu lớn. Công cụ này cho phép chuyển đổi và biến đổi dữ liệu từ dạng dài và không cấu trúc thành dạng rộng và có cấu trúc hơn để dễ dàng hiểu và phân tích.
- Ở đây, em sẽ sử dụng hàm “*pivot_table*” được tích hợp sẵn trong Pandas để tính toán độ tương tự giữa các cuốn sách và đưa ra kết quả:

```
def item_based(bookTitle):
    common_books_pivot = common_books.pivot_table(=["User-
ID"], columns=["Book-Title"], values="Book-Rating")
    title = common_books_pivot[bookTitle]
    recommendation_df = pd.DataFrame(
        common_books_pivot.corrwith(title).sort_values(ascending=False)
    ).reset_index(drop=False)

    if bookTitle in recommendation_df["Book-Title"].values:
        recommendation_df = recommendation_df[
            recommendation_df["Book-Title"] != bookTitle
        ]

    less_rating = recommendation_df[
        recommendation_df["Book-Title"].apply(
            lambda x: df[df["Book-Title"] == x]["Book-Rating"].mean() <
5
        )
    ]
    recommendation_df = recommendation_df[
        ~recommendation_df["Book-Title"].isin(less_rating["Book-
Title"])
    ].head(5)
    recommendation_df.columns = ["Book-Title", "Correlation"]
```

```
[ ] item_based("Harry Potter and the Sorcerer s Stone Harry Potter Paperback")
```

WOULD YOU LIKE TO TRY THESE BOOKS?



Hình 3.34: Kết quả những sách tương tự với tên sách cụ thể

6.4. User based

- User-based collaborative filtering là một phương pháp trong hệ thống đề xuất (recommendation system) được sử dụng để dự đoán sở thích hoặc ưa thích của người dùng dựa trên sự tương đồng giữa các người dùng trong hệ thống.
- Ở đây, em tiếp tục sử dụng “*pivote_table*” để tìm chỉ số của id trong `users_pivot.index` (các User-ID trong pivot table), để sử dụng cho tính toán độ tương tự sau đó sử dụng ma trận tương tự giữa các người dùng dựa trên pivot table `users_pivot` theo công thức [9]:

$$\text{similarity} = \cos(\theta) = \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}}$$

```
new_df = df[df["User-ID"].map(df["User-ID"].value_counts()) > 150]
users_pivot = new_df.pivot_table(
    index=["User-ID"], columns=["Book-Title"], values="Book-Rating"
).fillna(0)
def users_choice(id):
    users_fav = new_df[new_df["User-ID"] == id].nlargest(5, "Book-Rating")
    return users_fav
def user_based(new_df, id):
    index = np.where(users_pivot.index == id)[0][0]
    similarity = cosine_similarity(users_pivot)
    similar_users = list(enumerate(similarity[index]))
```



```

similar_users = sorted(similar_users, key=lambda x: x[1],
reverse=True) [0:5]
user_rec = []
for i in similar_users:
    data = df[df["User-ID"] == users_pivot.index[i[0]]]
    user_rec.extend(list(data.drop_duplicates("User-ID") ["User-
ID"].values))
return user_rec

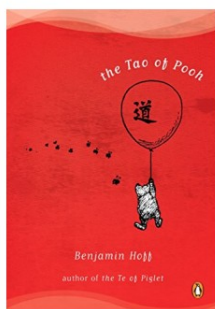
def common(new_df, user, user_id):
x = new_df[new_df["User-ID"] == user_id]
recommend_books = []
user = list(user)
for i in user:
    y = new_df[(new_df["User-ID"] == i)]
    books = y.loc[~y["Book-Title"].isin(x["Book-Title"]), :]
    books = books.sort_values(["Book-Rating"], ascending=False)
[0:5]
    recommend_books.extend(books["Book-Title"].values)
return recommend_books[0:5]

user_id = random.choice(new_df["User-ID"].values)
user_choice_df = pd.DataFrame(users_choice(user_id))
user_favorite = users_choice(user_id)

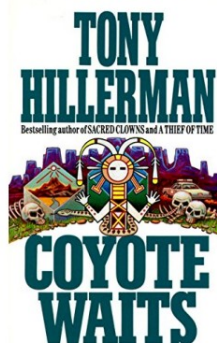
user_based_rec = user_based(new_df, user_id)
books_for_user = common(new_df, user_based_rec, user_id)
books_for_userDF = pd.DataFrame(books_for_user, columns=["Book-Title"])

```

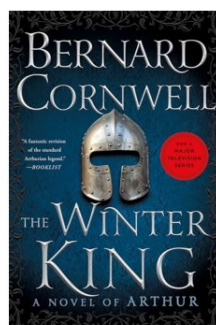
YOUR FAVORITE BOOKS



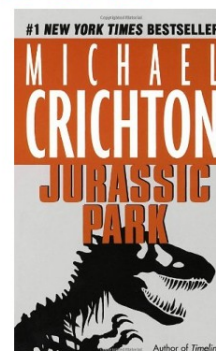
RATING: 8.1



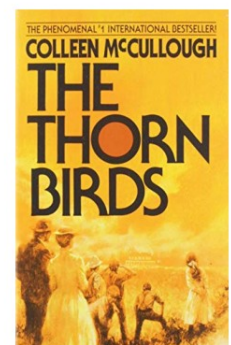
RATING: 8.5



RATING: 10.0

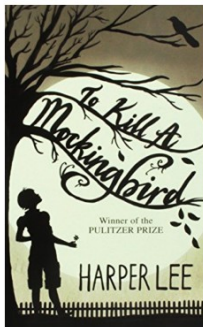


RATING: 8.1

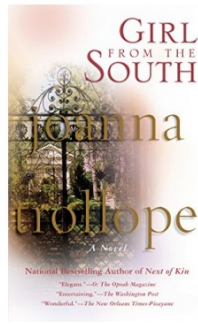


RATING: 8.4

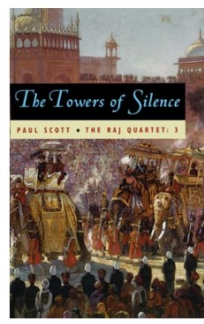
YOU MAY ALSO LIKE THESE BOOKS



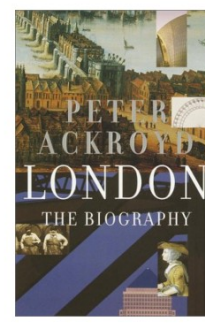
RATING: 9.1



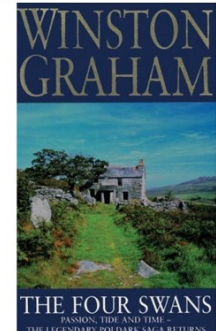
RATING: 10.0



RATING: 10.0



RATING: 10.0



RATING: 10.0

Hình 3.35: Kết quả những sách tương tự với sách yêu thích của người dùng khác

6.5. Content based

- Content-based filtering là một phương pháp trong hệ thống đề xuất (recommendation system) dựa trên việc phân tích đặc trưng (feature) của các sản phẩm để đưa ra các đề xuất cho người dùng dựa trên sự tương đồng về nội dung của các sản phẩm đó.
- Ở đây đối với sách cần khuyến nghị sử dụng `CountVectorizer()` để chuyển đổi các thông tin về sách thành vector. Sau đó tính toán ma trận độ tương tự giữa các cuốn sách dựa trên các vector đã tạo theo hàm cosine. Cuối cùng, xác định chỉ mục của `bookTitle` trong DataFrame `common_books` để sử dụng cho việc tìm sách tương tự [9]

$$\text{similarity} = \cos(\theta) = \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}}$$

```
def content_based(bookTitle):
    bookTitle = str(bookTitle)
    if bookTitle in df["Book-Title"].values:
        rating_count = pd.DataFrame(df["Book-Title"].value_counts())
        rare_books = rating_count[rating_count["Book-Title"] <=
150].index
        common_books = df[~df["Book-Title"].isin(rare_books)]

        if bookTitle in rare_books:
            most_common = (
                pd.Series(common_books["Book-
Title"]).unique()).sample(3).values)
            print("No Recommendations for this Book ☹️ \n ")
        else:
```

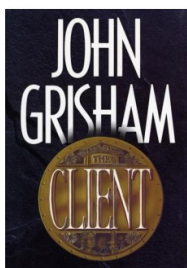
```

common_books = common_books.drop_duplicates(subset=["Book-
Title"])
common_books.reset_index(inplace=True)
common_books["index"] = [i for i in
range(common_books.shape[0])]
targets = ["Book-Title", "Book-Author", "Publisher"]
common_books["all_features"] = [
    ".join(common_books[targets].iloc[i,].values)
    for i in range(common_books[targets].shape[0])]
vectorizer = CountVectorizer()
common_booksVector =
vectorizer.fit_transform(common_books["all_features"])
similarity = cosine_similarity(common_booksVector)
index = common_books[common_books["Book-Title"] ==
bookTitle][
    "index"
].values[0]
similar_books = list(enumerate(similarity[index]))
similar_booksSorted = sorted(
    similar_books, key=lambda x: x[1], reverse=True
)[1:6]
books = []
for i in range(len(similar_booksSorted)):
    books.append(
        common_books[common_books["index"] ==
similar_booksSorted[i][0]][
            "Book-Title"
        ].item()

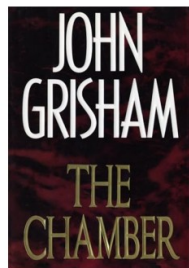
```

▶ content_based("The Firm")

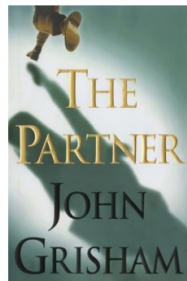
YOU MAY ALSO LIKE THESE BOOKS



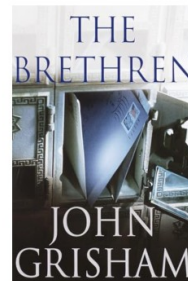
RATING: 7.9



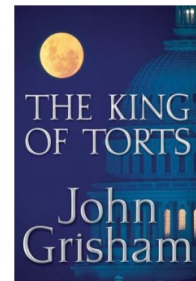
RATING: 7.4



RATING: 7.7



RATING: 7.4



RATING: 7.5

Hình 3.36: Kết quả khuyến nghị dựa trên nội dung sách

TỔNG KẾT

1. KẾT QUẢ ĐẠT ĐƯỢC

- Hệ thống đã đáp ứng được các chức năng cơ bản của một ứng dụng đọc sách điện tử và nghe sách trực tuyến trên thiết bị di động.
- Phát triển ứng dụng dưới hai nền tảng di động chính Android/IOS.
- Giao diện đã được thiết kế để tối ưu hóa trải nghiệm đọc và nghe, cung cấp các tính năng như đánh dấu trang, tùy chỉnh âm lượng, và tốc độ đọc.
- Khả năng khuyến nghị sách thông qua việc áp dụng các hệ thống khuyến nghị sử dụng để hiểu sâu hơn về sở thích và hành vi đọc của người dùng.

2. NHỮNG HẠN CHẾ

- Cần cải thiện hiệu suất ứng dụng trên các nền tảng và thiết bị khác nhau để đảm bảo trải nghiệm mượt mà và tối ưu hóa tương thích.
- Hệ thống còn đơn giản, cần phát triển thêm nhiều tính năng để có thể đáp ứng nhu cầu thực tế.
- Cần cải thiện hiệu suất mạng, đặc biệt là khi người dùng đọc sách điện tử hoặc nghe sách trực tuyến.
- Phát triển cơ chế tải trước nội dung để cải thiện trải nghiệm đọc sách và nghe sách offline trên thiết bị di động.

3. HƯỚNG PHÁT TRIỂN TIẾP THEO

- Xây dựng mạng xã hội Sách giúp tăng cường tính tương tác như nhận xét, thảo luận trực tuyến, đăng các bài viết về Sách để người dùng có thể chia sẻ ý kiến và trao đổi về các tác phẩm.
- Phát triển chức năng đọc sách offline hoàn thiện hơn, cho phép người dùng truy cập nội dung một cách nhanh chóng mà không cần kết nối internet.
- Tích hợp các công nghệ như chatbot để tư vấn hỗ trợ chọn sách, mua sách, áp dụng công nghệ AR/VR để tạo trải nghiệm đọc sách tương tác và hấp dẫn hơn ...
- Xây dựng chức năng đọc sách với hình ảnh động, hoạt ảnh và video hỗ trợ việc học tập và giáo dục trẻ em.
- Phát triển ứng dụng trên các nền tảng mới như smart TV, các thiết bị thông minh như đồng hồ thông minh, từ đó mở rộng phạm vi sử dụng của ứng dụng.
- Mở rộng tính năng chia sẻ trên các nền tảng xã hội phổ biến để người dùng có thể dễ

dành chia sẻ trải nghiệm và sách yêu thích của người dùng.

TÀI LIỆU THAM KHẢO

- [1] “Sách điện tử”. [Online]. Available:
https://vi.wikipedia.org/wiki/Sách_điện_tử
- [2] “Sách nói”. [Online]. Available:
https://vi.wikipedia.org/wiki/Sách_nói
- [3] “PostgreSQL: The World's Most Advanced Open Source Relational Database”. [Online]. Available:
- [4] “React Native: Setting up the development environment”. [Online]. Available:
<https://reactnative.dev/docs/environment-setup>
- [5] “A progressive Node.js framework for building efficient, reliable and scalable server-side applications”. [Online]. Available:
<https://docs.nestjs.com>
- [6] “Next-generation & fully type-safe ORM for NestJS”. [Online]. Available:
<https://www.prisma.io/nestjs>
- [7] Trần Đình Quế. Phân tích và thiết kế hệ thống thông tin. Học viện Công nghệ Bưu chính Viễn thông. 2018
<https://www.postgresql.org/docs>
- [8] “Epubjs - Enhanced eBooks in the browser.”. [Online]. Available:
<https://idpf.org/epub/30/spec/epub30-overview.html>
- [9] “Độ tương tự cosin” . [Online]. Available:
https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99_t%C6%B0%C6%A1ng_t%E1%BB%B1_cosin
- [10] “Weighted arithmetic mean”. [Online]. Available:
https://en.wikipedia.org/wiki/Weighted_arithmetic_mean
- [11] (PDF) Recommender Systems and Linked Open Data. [Online]. Available
https://www.researchgate.net/publication/300646445_Recommender_Systems_and_Linked_Open_Data [accessed Nov 25 2023].