

TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA
-----oOo-----



**BÁO CÁO BÀI TẬP LỚN
PHÂN TÍCH THIẾT KẾ PHẦN MỀM**

Course/Học phần:	Phân tích thiết kế phần mềm
Instructor/Giảng viên:	ThS. Vũ Quang Dũng
Group/Tên nhóm:	1
Class/Lớp:	N06
Topic/Chủ đề:	Xây dựng app quản lý thư viện

HÀ NỘI - 2025

DANH SÁCH THÀNH VIÊN TRONG NHÓM 1

TT	Họ và tên	Email	Vai trò	Mã sinh viên	Điểm đóng góp (theo thang điểm 10)	Điểm cuối cùng(theo thang điểm 10)
1	Quách Gia Bảo	22010039@st.phenikaa-uni.edu.vn	Trưởng nhóm	22010039	10/10	
2	Trần Đăng Khoa	22010303@st.phenikaa-uni.edu.vn	Thành viên	22010303	10/10	
3	Tô Thái An	22010038@st.phenikaa-uni.edu.vn	Thành viên	22010038	10/10	
4	Hoàng Văn Bắc	22010356@st.phenikaa-uni.edu.vn	Thành viên	22010356	10/10	
5	Hà Thành Đạt	22010041@st.phenikaa-uni.edu.vn	Thành viên	22010041	10/10	

LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, việc ứng dụng công nghệ vào quản lý thư viện trở nên cần thiết nhằm tối ưu hóa quy trình, tiết kiệm thời gian và nâng cao hiệu suất làm việc. Hệ thống quản lý thư viện truyền thống thường gặp nhiều hạn chế như khó khăn trong việc tra cứu, cập nhật dữ liệu, và quản lý mượn trả sách. Do đó, một ứng dụng quản lý thư viện hiện đại, linh hoạt và dễ sử dụng sẽ giúp cải thiện đáng kể hiệu quả hoạt động của thư viện.

Với mục tiêu này, nhóm chúng tôi đã phát triển một ứng dụng quản lý thư viện sử dụng **Flutter** làm công nghệ chính để xây dựng giao diện đa nền tảng, kết hợp với **Firestore** để lưu trữ và đồng bộ dữ liệu. Ứng dụng cung cấp các tính năng quan trọng như quản lý sách, quản lý người dùng, theo dõi mượn trả sách và tìm kiếm thông tin nhanh chóng.

Báo cáo này sẽ trình bày chi tiết về quá trình xây dựng ứng dụng, bao gồm phân tích yêu cầu, thiết kế hệ thống, triển khai và đánh giá hiệu quả. Hy vọng rằng sản phẩm này sẽ đóng góp một giải pháp hữu ích cho công tác quản lý thư viện và mở ra cơ hội phát triển thêm nhiều tính năng trong tương lai.

MỤC LỤC

DANH SÁCH THÀNH VIÊN TRONG NHÓM 1	1
LỜI MỞ ĐẦU	2
MỤC LỤC	3
NỘI DUNG CHÍNH	4
I. Giới thiệu	4
1. Tổng quan về ứng dụng	4
2. Mục tiêu phát triển ứng dụng	4
3. Công nghệ sử dụng	5
4. Các chức năng chính	5
5. Lợi ích của ứng dụng	8
6. Đối tượng sử dụng	8
II. MÔ HÌNH HỆ THỐNG	8
1. Kiến trúc tổng quan	8
2. Các thành phần chính của hệ thống	9
3. Cơ sở dữ liệu	11
4. Lưu đồ hoạt động chính	13
5. Cấu trúc thư mục	14
III. PHÂN TÍCH HỆ THỐNG	17
1. Sơ đồ Use/Case	17
a. Sơ đồ Use/Case người dùng	17
b. Sơ đồ Use/Case quản trị viên	17
2. Các chức năng của ứng dụng	18
a. Chức năng cho người dùng	18
b. Chức năng cho quản trị viên	32
IV. KẾT LUẬN	48
1. Kết quả thu được	48
2. Hạn chế của ứng dụng	49
3. Hướng phát triển đề tài	50
4. Tổng kết	51
TÀI LIỆU THAM KHẢO	51

NỘI DUNG CHÍNH

I. Giới thiệu

1. Tổng quan về ứng dụng

Ứng dụng quản lý thư viện là một giải pháp phần mềm được phát triển trên nền tảng **Flutter** và sử dụng **Firebase** để lưu trữ dữ liệu. Ứng dụng hỗ trợ các thư viện, trường học, trung tâm học tập hoặc cá nhân trong việc quản lý sách, người dùng, cũng như hoạt động mượn và trả sách một cách hiệu quả.

Trong bối cảnh công nghệ số phát triển mạnh mẽ, việc ứng dụng các giải pháp phần mềm vào quản lý thư viện giúp tối ưu hóa quy trình làm việc, giảm thiểu sai sót, nâng cao hiệu suất và cải thiện trải nghiệm của người dùng. Ứng dụng này không chỉ giúp tổ chức tài liệu khoa học mà còn cung cấp các tính năng tự động hóa để quản lý hoạt động mượn/trả một cách tiện lợi.

2. Mục tiêu phát triển ứng dụng

Ứng dụng được xây dựng nhằm giải quyết một số vấn đề chính trong quản lý thư viện truyền thống, bao gồm:

- Khó khăn trong việc theo dõi số lượng sách và tình trạng mượn/trả.
- Việc ghi chép và quản lý thủ công dễ dẫn đến sai sót.
- Khả năng tìm kiếm sách hạn chế khi số lượng sách lớn.
- Khó kiểm soát thông tin độc giả và lịch sử mượn sách.

Mục tiêu của ứng dụng là:

- Cung cấp một nền tảng quản lý thư viện **hiện đại, trực quan và dễ sử dụng**.
- Giúp **quản lý danh mục sách** một cách khoa học, dễ dàng tìm kiếm và cập nhật.

- Hỗ trợ **quản lý người dùng** (độc giả), bao gồm thông tin cá nhân và lịch sử mượn sách.
- Hỗ trợ **quản lý mượn/trả sách** với các tính năng tự động cập nhật trạng thái sách và nhắc nhở hạn trả.
- Ứng dụng công nghệ **đồng bộ dữ liệu theo thời gian thực** bằng Firebase, giúp thông tin luôn chính xác và kịp thời.

3. Công nghệ sử dụng

- **Flutter**: Là một framework phát triển ứng dụng đa nền tảng (Android, iOS, Web) của Google, giúp tối ưu hiệu suất và giao diện.
- **Firebase**: Dịch vụ backend mạnh mẽ hỗ trợ lưu trữ dữ liệu (Cloud Firestore), xác thực người dùng (Firebase Authentication), và thông báo (Firebase Cloud Messaging).
- **Provider / Riverpod**: Quản lý trạng thái giúp ứng dụng hoạt động mượt mà và dễ bảo trì.
- **Dart**: Ngôn ngữ lập trình chính của Flutter, giúp tối ưu hiệu suất và tốc độ xử lý.

4. Các chức năng chính

a. Chức năng của quản trị viên

Quản lý danh mục sách:

- Thêm, sửa, xóa thông tin sách (tên sách, tác giả, thể loại, năm xuất bản, số lượng, v.v.).
- Hỗ trợ tìm kiếm và lọc sách theo nhiều tiêu chí.

Quản lý người dùng (độc giả):

- Đăng ký tài khoản và quản lý thông tin cá nhân.
- Theo dõi lịch sử mượn/trả sách của từng độc giả.

Quản lý mượn và trả sách:

- Lưu trữ thông tin mượn/trả sách theo thời gian thực.
- Tự động cập nhật trạng thái sách (có sẵn, đã mượn, quá hạn).
- Nhắc nhở độc giả về hạn trả sách qua thông báo.

Đồng bộ dữ liệu real-time:

- Nhờ sử dụng Firebase, tất cả thông tin được cập nhật ngay lập tức trên nhiều thiết bị.
- Giúp quản trị viên dễ dàng theo dõi và quản lý thư viện từ xa.

Hệ thống báo cáo và thống kê:

- Cung cấp báo cáo về số lượng sách, sách đang mượn, sách quá hạn, số lượng độc giả đang hoạt động.
- Hỗ trợ xuất dữ liệu để phục vụ việc phân tích và đánh giá.

b. Chức năng của người dùng

Đăng ký tài khoản

- Cho phép người dùng đăng ký tài khoản mới để sử dụng hệ thống.
- Yêu cầu xác thực email trước khi kích hoạt tài khoản.
- Quản trị viên có thể xét duyệt tài khoản trước khi cho phép sử dụng.

Đăng nhập

- Cho phép người dùng đăng nhập vào hệ thống bằng email và mật khẩu.
- Hỗ trợ tính năng quên mật khẩu và đặt lại qua email.
- Tự động khóa tài khoản nếu nhập sai mật khẩu quá 5 lần liên tiếp.

Tìm kiếm sách

- Hỗ trợ tìm kiếm sách theo tên, tác giả, thể loại hoặc ISBN.
- Cung cấp bộ lọc nâng cao theo năm xuất bản, thể loại, tình trạng sách.

- Hiện thị danh sách kết quả theo thứ tự liên quan nhất.

Xem chi tiết sách

- Hiện thị thông tin chi tiết của sách, bao gồm tên, tác giả, mô tả, số lượng còn lại.
- Cung cấp ảnh bìa sách và vị trí trên kệ thư viện.
- Cho phép xem đánh giá và nhận xét từ những người đã mượn trước đó.

Đặt mượn sách

- Cho phép người dùng gửi yêu cầu mượn sách nếu còn số lượng.
- Tự động kiểm tra giới hạn số lượng sách được mượn (tối đa 3 cuốn).
- Gửi yêu cầu đến nhân viên thư viện để xét duyệt.

Kiểm tra trạng thái mượn sách

- Hiện thị danh sách sách đã yêu cầu mượn kèm trạng thái (chờ duyệt, đã duyệt, từ chối).
- Cập nhật trạng thái theo thời gian thực sau khi nhân viên xử lý.
- Hiện thị lý do nếu yêu cầu bị từ chối.

Nhận thông báo nhắc nhở

- Tự động gửi thông báo trước 3 ngày khi đến hạn trả sách.
- Gửi thông báo vào ngày hết hạn và khi sách quá hạn.
- Cảnh báo nếu người dùng có quá nhiều sách mượn quá hạn.

Trả sách

- Cho phép người dùng báo trả sách qua ứng dụng.
- Cập nhật trạng thái sách sau khi nhân viên xác nhận đã nhận lại.
- Ghi nhận lịch sử mượn và trả sách.

Xem lịch sử mượn sách

- Cung cấp danh sách sách đã từng mượn kèm ngày mượn và ngày trả.
- Lưu trữ lịch sử mượn sách tối đa 2 năm.

Đánh giá sách

- Cho phép người dùng chấm điểm và để lại nhận xét về sách đã đọc.
- Hạn chế chỉ cho phép đánh giá sách sau khi đã trả.
- Hiện thị đánh giá trung bình từ tất cả người dùng.

5. Lợi ích của ứng dụng

- ✓ **Tiện lợi và dễ sử dụng:** Giao diện thân thiện, hỗ trợ đa nền tảng, giúp quản trị viên và người dùng dễ dàng thao tác.
- ✓ **Tăng hiệu suất quản lý:** Giảm thiểu công việc thủ công, tiết kiệm thời gian và hạn chế sai sót.
- ✓ **Truy cập mọi lúc, mọi nơi:** Dữ liệu được lưu trữ trên đám mây, đảm bảo tính linh hoạt và an toàn.
- ✓ **Nâng cao trải nghiệm người dùng:** Hỗ trợ tìm kiếm sách nhanh chóng, nhắc nhở tự động và thao tác mượn/trả đơn giản.

6. Đối tượng sử dụng

Ứng dụng phù hợp với nhiều đối tượng khác nhau, bao gồm:

- **Thư viện trường học, trung tâm học tập:** Giúp quản lý sách giáo khoa, tài liệu học tập cho học sinh/sinh viên.
- **Thư viện công cộng:** Tối ưu hóa quy trình quản lý sách và hỗ trợ độc giả.
- **Cá nhân đam mê đọc sách:** Hỗ trợ quản lý bộ sưu tập sách cá nhân một cách khoa học.

II. MÔ HÌNH HỆ THỐNG

1. Kiến trúc tổng quan

Ứng dụng quản lý thư viện được thiết kế theo mô hình **Client-Server** để đảm bảo tính linh hoạt, hiệu suất và khả năng mở rộng. Mô hình này bao gồm hai thành phần chính:

Client (Frontend - Ứng dụng Flutter):

- Giao diện người dùng (UI) chạy trên thiết bị di động hoặc web.
- Kết nối với cơ sở dữ liệu và dịch vụ backend để lấy và hiển thị dữ liệu.
- Hỗ trợ các chức năng như đăng nhập, tra cứu sách, mượn trả sách, nhận thông báo.

Server (Backend - Firebase Services):

- Lưu trữ dữ liệu người dùng, sách, lịch sử mượn trả.
- Xử lý xác thực và phân quyền người dùng.
- Gửi thông báo nhắc nhở trả sách và cập nhật thông tin thư viện.

2. Các thành phần chính của hệ thống

a. Front-end(Flutter)

Frontend là phần mà người dùng tương tác trực tiếp, gồm các tính năng chính sau:

Quản lý người dùng

- Đăng ký, đăng nhập qua **Firebase Authentication** (Email/Password, Google Sign-In).
- Quản lý thông tin cá nhân (họ tên, email, lịch sử mượn sách).

Quản lý sách

- Hiển thị danh sách sách với tìm kiếm và bộ lọc.
- Xem chi tiết từng cuốn sách (tác giả, mô tả, tình trạng).
- Quản trị viên có thể thêm, sửa, xóa sách.

Mượn & Trả sách

- Người dùng có thể đặt mượn sách trực tiếp từ ứng dụng.
- Kiểm tra tình trạng sách (còn hay đã được mượn).
- Lịch sử mượn sách của người dùng.

Hệ thống thông báo

- Gửi thông báo nhắc nhở trả sách.
- Thông báo về sách mới hoặc các sự kiện trong thư viện.

b. Back-end(Firebase Services)

Phía backend sử dụng **Firebase** với các dịch vụ chính:

Firebase Authentication

- Xác thực người dùng bằng email/password hoặc Google Sign-In.
- Phân quyền: Người dùng (độc giả) và quản trị viên (admin).

Firebase Firestore (Cơ sở dữ liệu)

Dữ liệu được lưu trữ dưới dạng NoSQL. Ví dụ:

/users (Collection)

```

|— userId (Document)
  |— name: "Nguyễn Văn A"
  |— email: "nguyenvana@example.com"
  |— borrowedBooks: ["bookId_1", "bookId_2"]
  
```

/books (Collection)

```

|— bookId (Document)
  |— title: "Lập trình Flutter"
  
```

└─ **author: "John Doe"**
└─ **category: "Công nghệ"**
└─ **availability: true**

/borrow_records (Collection)

└─ **recordId (Document)**
└─ **userId: "userId_123"**
└─ **bookId: "bookId_456"**
└─ **borrowDate: "2025-03-01"**
└─ **returnDate: "2025-03-10"**
└─ **status: "borrowed"**

Firebase Cloud Storage

- Lưu trữ ảnh bìa sách, ảnh đại diện người dùng.

Firebase Cloud Functions (nếu cần)

- Xử lý thông báo tự động (ví dụ: nhắc nhở trả sách).

3. Cơ sở dữ liệu

Cấu trúc dữ liệu Firestore dạng NoSQL, có thể tổ chức như sau:

/users (Collection)

└─ **userId (Document)**
└─ **name: String**
└─ **email: String**
└─ **borrowedBooks: [bookId]**

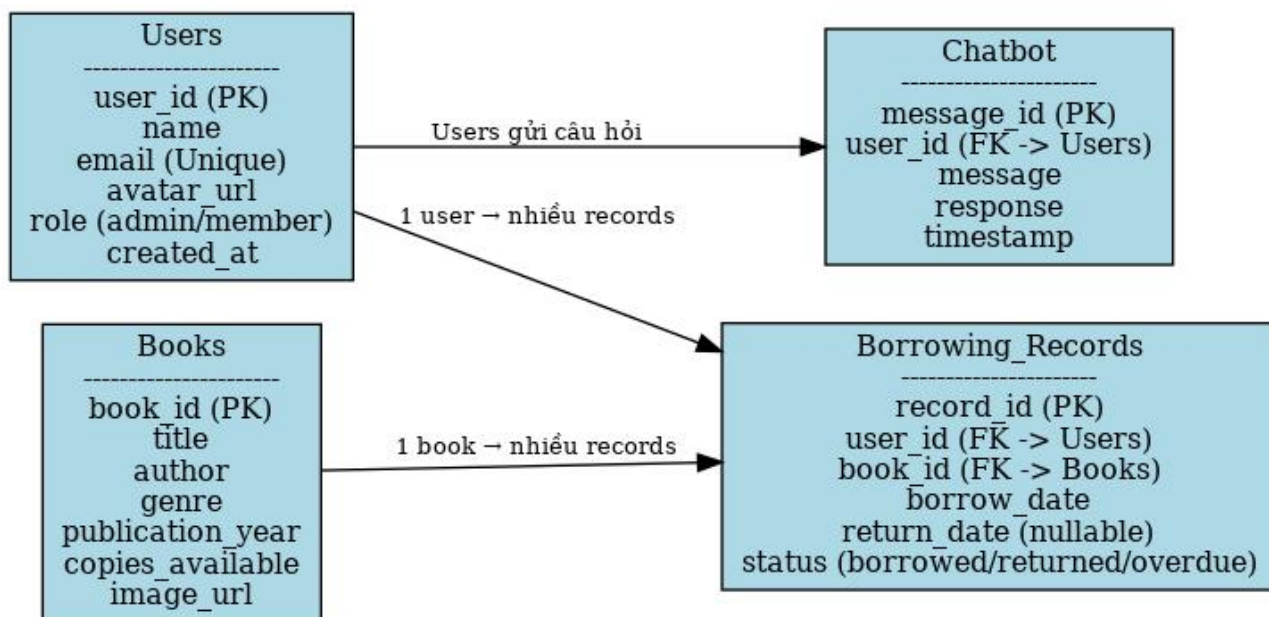
/books (Collection)

- |— bookId (Document)**
 - |— title: String**
 - |— author: String**
 - |— category: String**
 - |— availability: Boolean**

/borrow_records (Collection)

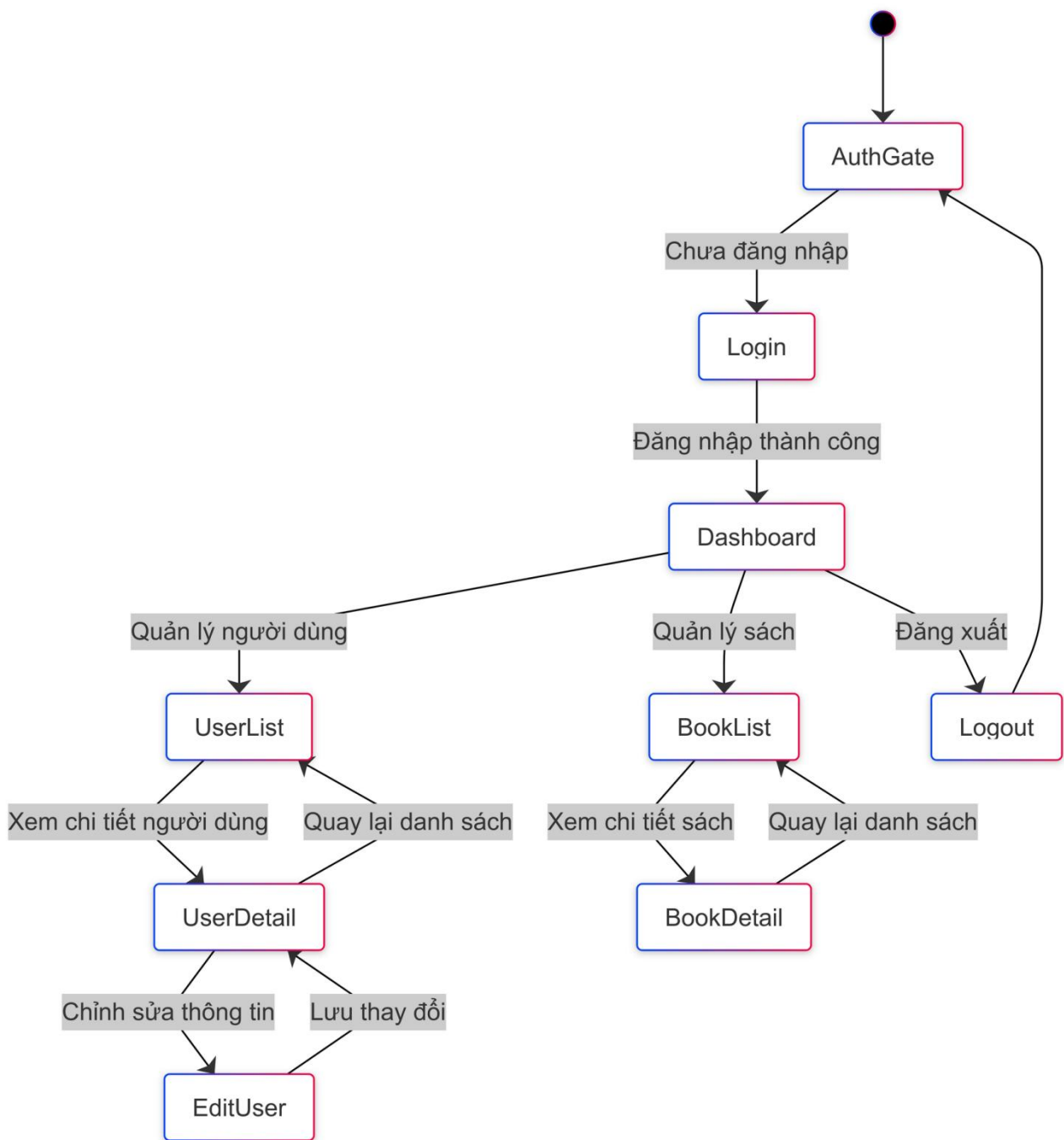
- |— recordId (Document)**
 - |— userId: String**
 - |— bookId: String**
 - |— borrowDate: Timestamp**
 - |— returnDate: Timestamp**
 - |— status: String (borrowed/returned)**

Liên kết giữa các bảng:



4. Lưu đồ hoạt động chính

Quản trị viên:



Thành viên:

5. Cấu trúc thư mục

lib/

| — *main.dart* *# Điểm khởi chạy chính của ứng dụng*

|

| — *core/* *# Cấu hình chung cho ứng dụng*

- | | — *app_constants.dart* # Các hằng số chung
- | | — *app_routes.dart* # Định nghĩa các route điều hướng
- | | — *theme.dart* # Cấu hình giao diện (màu sắc, font,...)
- |
- | — *models/* # Định nghĩa các model dữ liệu
- | | — *book_model.dart* # Model cho sách
- | | — *user_model.dart* # Model cho người dùng
- | | — *borrow_record_model.dart* # Model cho lịch sử mượn sách
- |
- | — *services/* # Xử lý logic kết nối dữ liệu
- | | — *auth_service.dart* # Quản lý đăng nhập, đăng ký
- | | — *book_service.dart* # Quản lý sách (CRUD)
- | | — *borrow_service.dart* # Quản lý mượn/trả sách
- | | — *notification_service.dart* # Quản lý thông báo Firebase
- |
- | — *controllers/* # Điều khiển dữ liệu từ service đến UI
- | | — *auth_controller.dart* # Kiểm soát xác thực người dùng
- | | — *book_controller.dart* # Kiểm soát danh sách sách
- | | — *borrow_controller.dart* # Kiểm soát mượn/trả sách
- |
- | — *views/* # Giao diện ứng dụng
- | | — *auth/* # Màn hình xác thực
- | | | — *login_screen.dart*


```

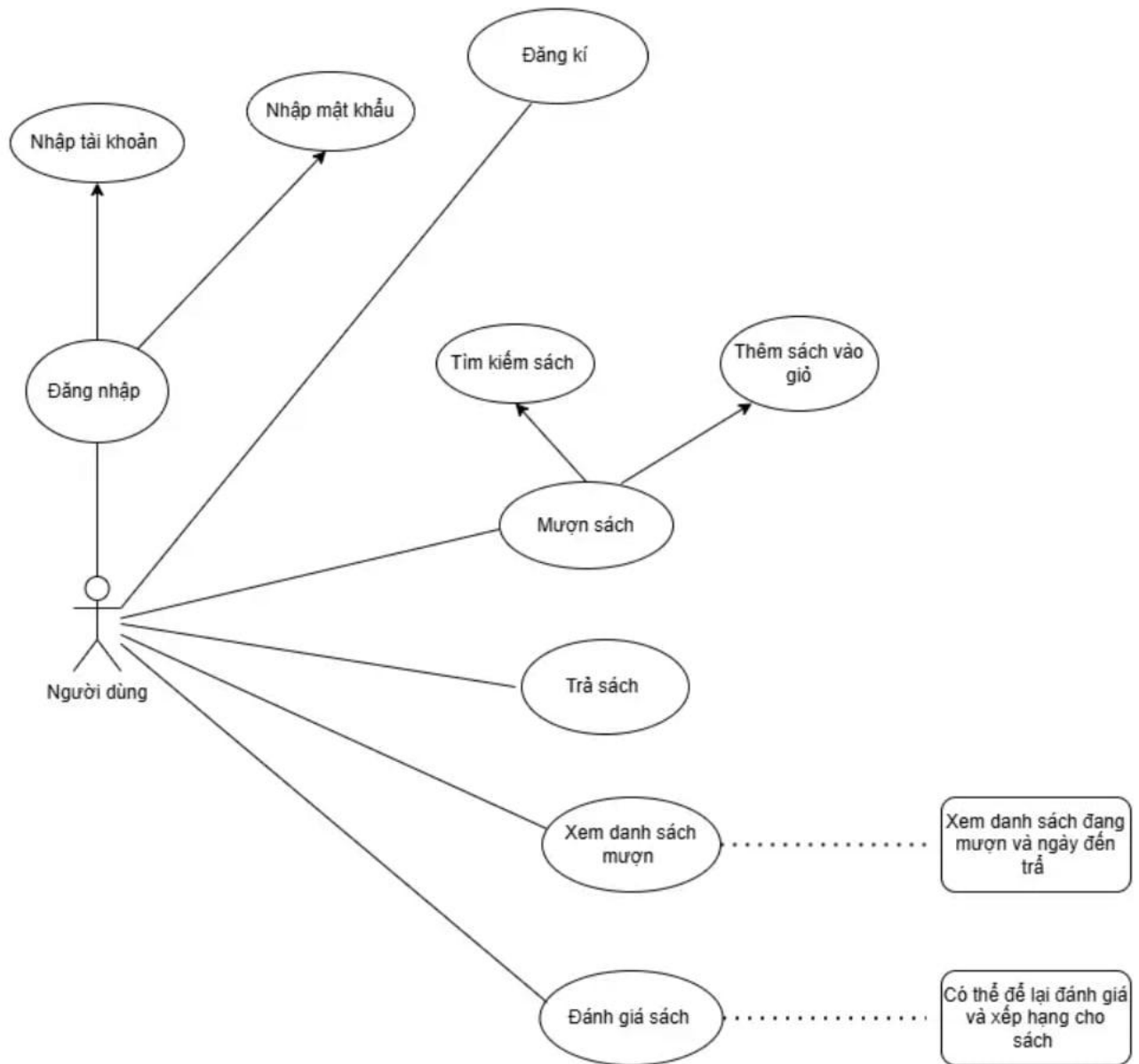
| | | └── register_screen.dart
| |
| | └── home/           # Màn hình chính
| | | └── home_screen.dart
| | | └── book_list.dart
| |
| |
| | └── book/           # Màn hình quản lý sách
| | | └── book_detail.dart
| | | └── book_add.dart
| |
| |
| | └── borrow/         # Màn hình mượn/trả sách
| | | └── borrow_screen.dart
| | | └── return_screen.dart
| |
|
| └── widgets/          # Các widget dùng chung
| | └── book_card.dart   # Hiển thị sách theo dạng card
| | └── custom_button.dart # Nút bấm tùy chỉnh
| | └── input_field.dart  # Trường nhập dữ liệu
| |
|
└── utils/              # Các tiện ích hỗ trợ
    | └── validators.dart   # Hàm kiểm tra dữ liệu đầu vào
    | └── date_helper.dart  # Xử lý định dạng ngày tháng
    | └── logger.dart       # Ghi log thông tin (debug)

```

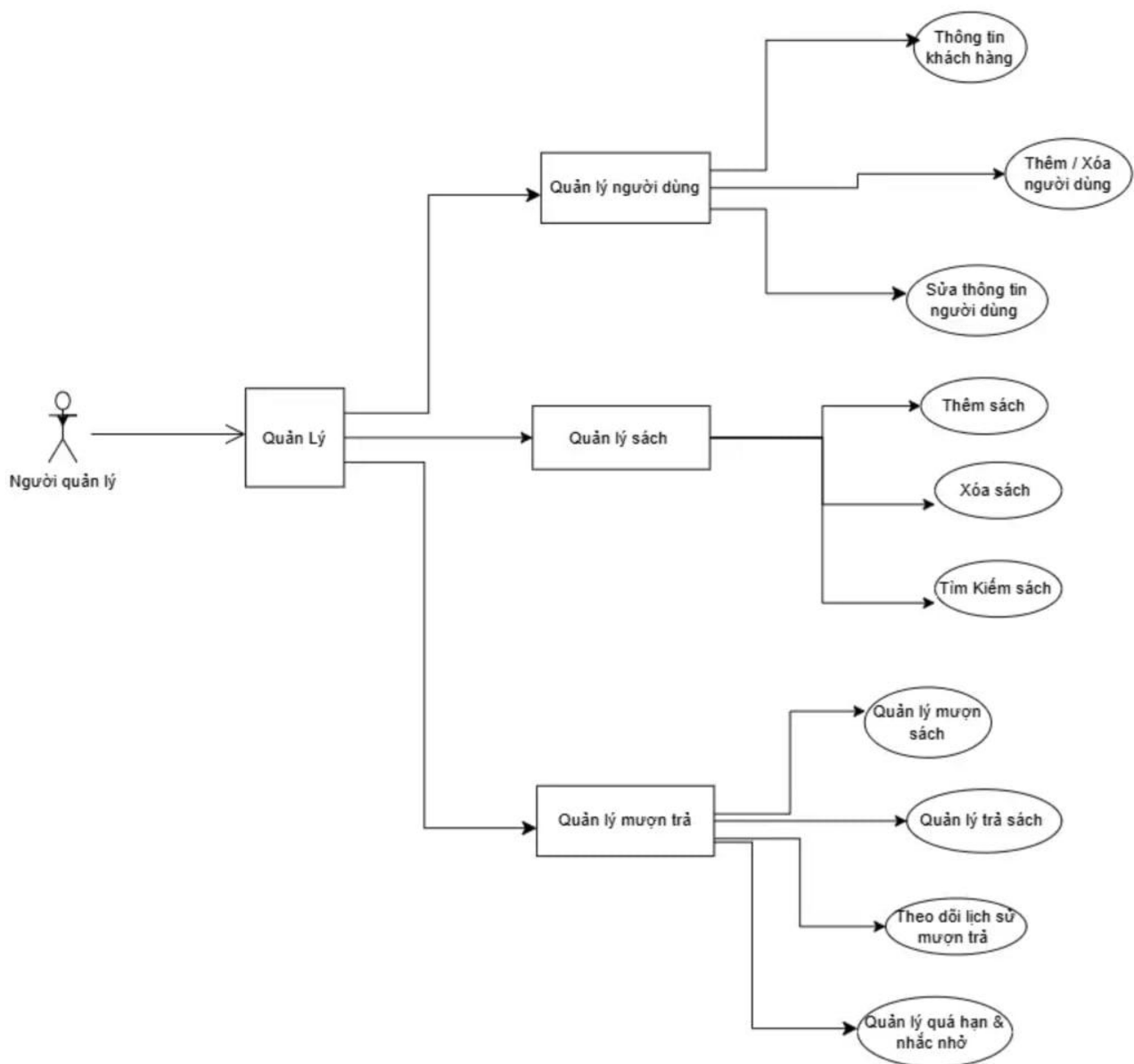
III. PHÂN TÍCH HỆ THỐNG

1. Sơ đồ Use/Case

a. Sơ đồ Use/Case người dùng



b. Sơ đồ Use/Case quản trị viên

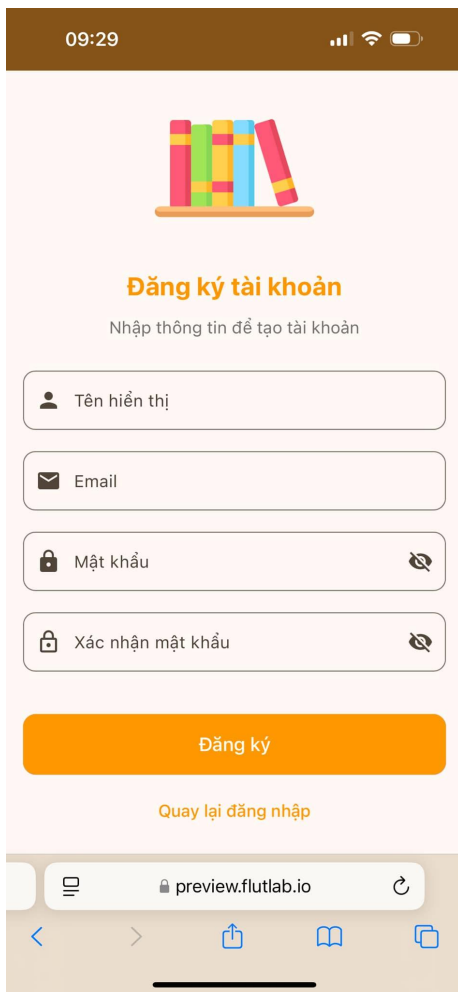


2. Các chức năng của ứng dụng

a. Chức năng cho người dùng

- Đăng ký:

Màn hình giao diện:



09:29

Đăng ký tài khoản

Nhập thông tin để tạo tài khoản

Tên hiển thị

Email

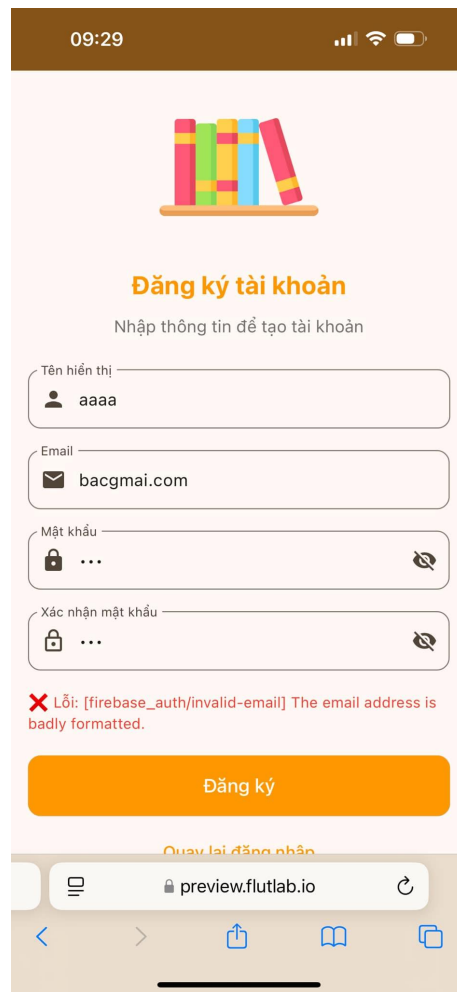
Mật khẩu

Xác nhận mật khẩu

Đăng ký

[Quay lại đăng nhập](#)

preview.flutlab.io



09:29

Đăng ký tài khoản

Nhập thông tin để tạo tài khoản

Tên hiển thị
aaaa

Email
bacgmai.com

Mật khẩu
...

Xác nhận mật khẩu
...

Lỗi: [firebase_auth/invalid-email] The email address is badly formatted.

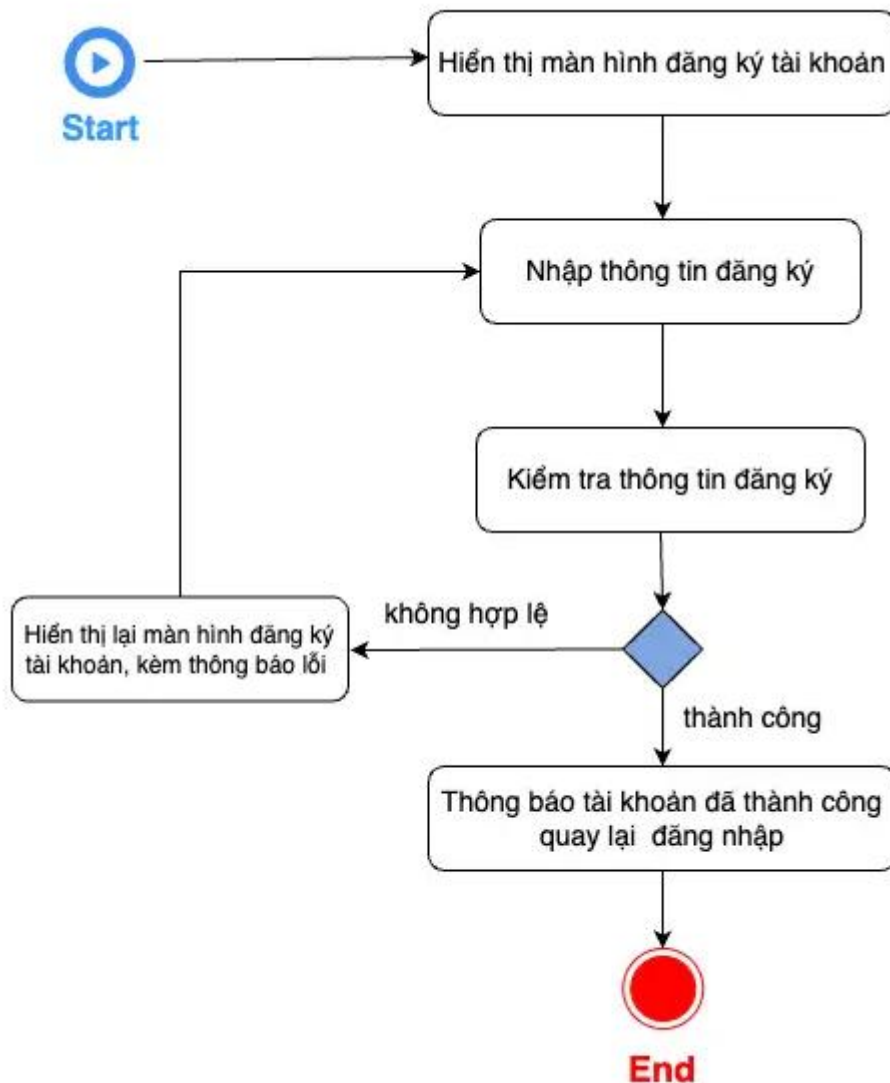
Đăng ký

[Quay lại đăng nhập](#)

preview.flutlab.io

Activity Flow:

Step	Description
1	Hiển thị màn hình đăng ký tài khoản cho người dùng.
2	Người dùng nhập thông tin đăng ký vào biểu mẫu.
3	Hệ thống kiểm tra thông tin đăng ký.
4	<ul style="list-style-type: none"> Nếu thông tin không hợp lệ: Hệ thống hiển thị lại màn hình đăng ký kèm –thông báo lỗi để người dùng chỉnh sửa. Nếu thông tin hợp lệ: Thông báo tài khoản đã được đăng ký thành công, hướng dẫn người dùng quay lại màn hình đăng nhập.



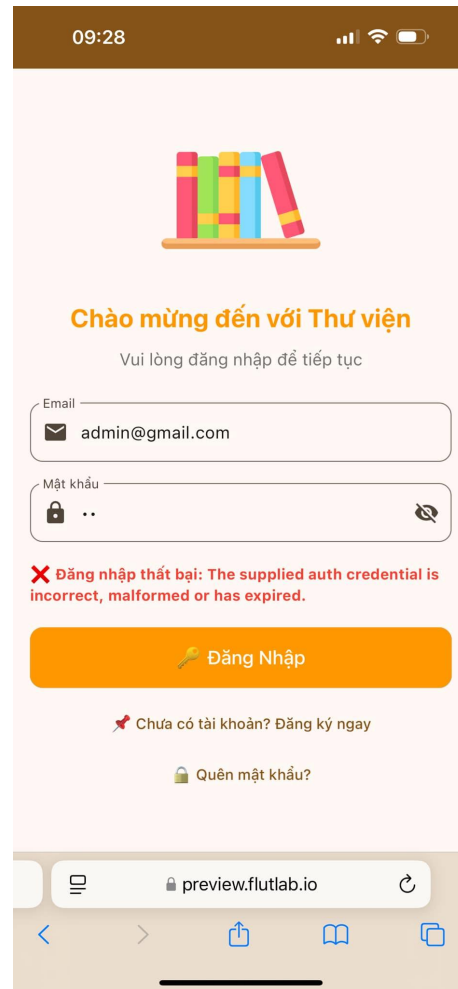
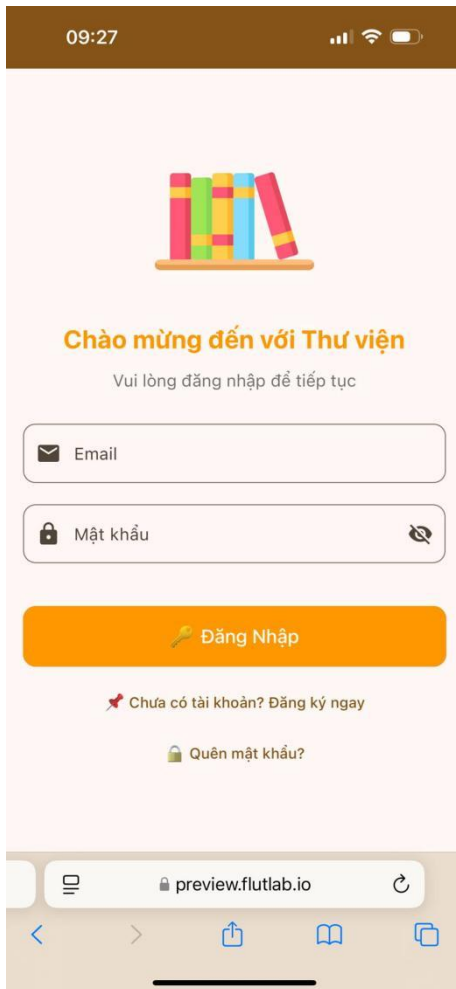
Bussiness Rules:

Rule	Description
Quy tắc nhập thông tin đăng ký	<ul style="list-style-type: none"> · Người dùng cần cung cấp các thông tin bắt buộc: <ul style="list-style-type: none"> • Họ và tên (bắt buộc, tối đa 50 ký tự, không chứa ký tự đặc biệt). • Email (bắt buộc, đúng định dạng email, không trùng với tài khoản đã tồn tại). • Số điện thoại (tùy chọn, nếu nhập phải đúng định dạng và không trùng lặp). • Mật khẩu (bắt buộc, phải tuân theo chính sách bảo

	<p>mật).</p> <ul style="list-style-type: none"> · Mật khẩu phải thỏa mãn yêu cầu: <ul style="list-style-type: none"> • Tối thiểu 8 ký tự. • Chứa ít nhất 1 chữ hoa, 1 chữ thường, 1 số và 1 ký tự đặc biệt. · Nếu hệ thống hỗ trợ đăng ký bằng Google, người dùng có thể chọn đăng ký bằng tài khoản Google thay vì nhập thông tin thủ công.
Quy tắc xác minh tài khoản	<ul style="list-style-type: none"> · Sau khi đăng ký, hệ thống sẽ gửi email xác minh. · Tài khoản chỉ có thể đăng nhập sau khi đã xác minh email. · Liên kết xác minh email sẽ hết hạn sau 24 giờ. · Nếu liên kết xác minh hết hạn, người dùng có thể yêu cầu gửi lại email xác minh.
Quy tắc xử lý lỗi	<ul style="list-style-type: none"> · Nếu email đã tồn tại, hiển thị thông báo: "Email đã được sử dụng. Vui lòng chọn email khác". · Nếu số điện thoại đã tồn tại, hiển thị thông báo: "Số điện thoại đã được đăng ký". · Nếu mật khẩu không hợp lệ, hiển thị thông báo hướng dẫn tạo mật khẩu mạnh. · Nếu hệ thống gặp lỗi khi gửi email xác minh, hiển thị thông báo: "Không thể gửi email xác minh. Vui lòng thử lại sau".
Quy tắc bảo mật	<ul style="list-style-type: none"> · Mật khẩu được lưu dưới dạng mã hóa, không lưu trữ dưới dạng plaintext. · Số điện thoại (nếu có) có thể được sử dụng để xác thực hai yếu tố (2FA). · Nếu hệ thống hỗ trợ Captcha, người dùng phải vượt qua Captcha để hoàn tất đăng ký (chống bot).

- Đăng nhập:

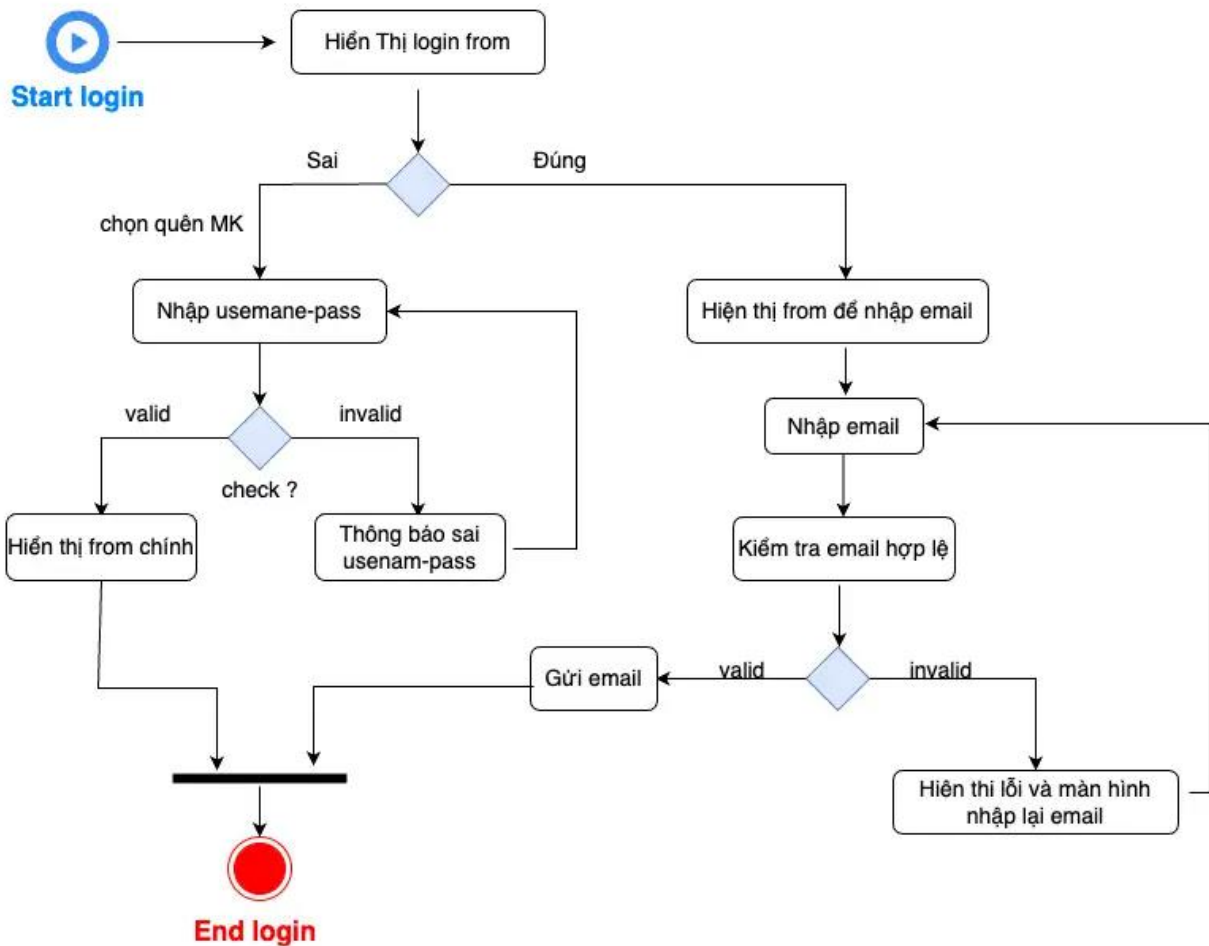
Màn hình giao diện:



Activity Flow:

Step	Description
1	Người dùng truy cập chức năng "Quản lý thành viên".
2	Chọn một trong hai hành động chính: <ul style="list-style-type: none"> - Thêm thành viên. - Xem thông tin thành viên.
3	<ul style="list-style-type: none"> - Nếu chọn Thêm thành viên: <ul style="list-style-type: none"> + Nhập thông tin thành viên. + Người dùng có thể chọn "Hủy" để thoát hoặc "Thêm" để hoàn thành

	việc thêm thành viên.
4	<ul style="list-style-type: none"> - Nếu chọn Xem thông tin thành viên: + Người dùng có các tùy chọn: - Đặt lại mật khẩu: Gửi email xác nhận và hoàn thành việc đặt lại mật khẩu. - Chỉnh sửa thông tin: Nhập thông tin chỉnh sửa và chọn "Lưu". - Xóa thành viên: Xác nhận việc xóa thành viên.
5	Sau khi hoàn thành các thao tác, hệ thống sẽ kết thúc quy trình.

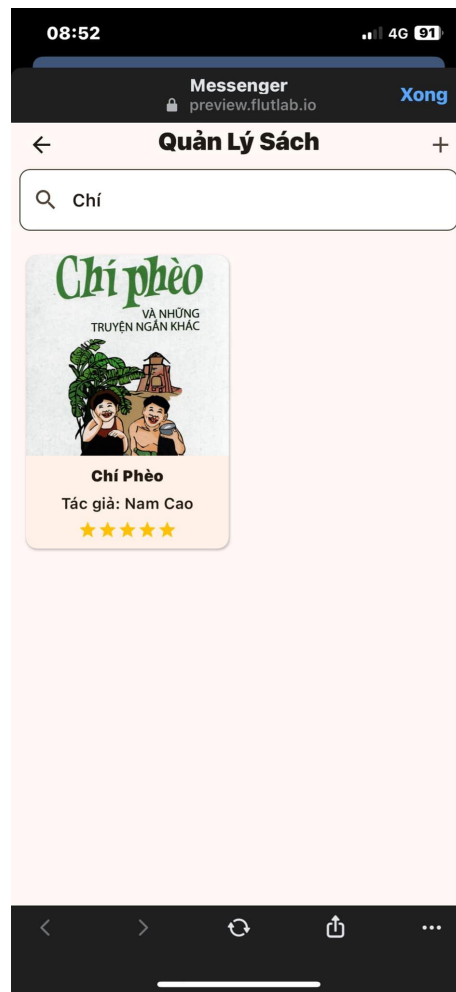
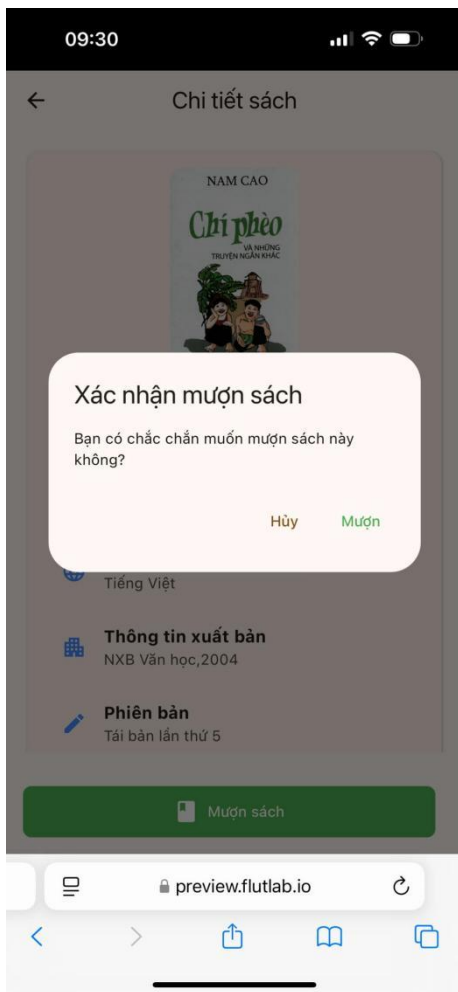


Bussiness Rules:

Rule	Description
Quy tắc xác thực	<ul style="list-style-type: none"> · Hệ thống chỉ cho phép người dùng đã đăng ký đăng nhập. · Tên đăng nhập có thể là email hoặc số điện thoại đã đăng ký. · Mật khẩu phải hợp lệ theo chính sách bảo mật của hệ thống. · Hỗ trợ đăng nhập bằng tài khoản Google (nếu có tích hợp).
Quy tắc bảo mật	<ul style="list-style-type: none"> · Sau 5 lần nhập sai mật khẩu liên tiếp, tài khoản sẽ bị khóa tạm thời trong 15 phút. · Mật khẩu không được lưu trữ dưới dạng văn bản thuần (plaintext). · Hệ thống sử dụng Firestore Authentication để xác thực người dùng. · Nếu bật Xác thực hai yếu tố (2FA), hệ thống sẽ yêu cầu nhập mã OTP gửi qua email/số điện thoại.
Quy tắc xử lý lỗi	<ul style="list-style-type: none"> · Nếu người dùng nhập sai tài khoản/mật khẩu, hiển thị thông báo lỗi: "Tài khoản hoặc mật khẩu không đúng". · Nếu tài khoản bị khóa tạm thời, hiển thị thông báo: "Tài khoản bị khóa trong 15 phút do nhập sai quá nhiều lần". · Nếu tài khoản chưa được kích hoạt, yêu cầu người dùng xác minh email trước khi đăng nhập. · Nếu có lỗi kết nối mạng, hiển thị thông báo: "Không thể kết nối. Vui lòng kiểm tra lại mạng của bạn".
Quy tắc duy trì phiên đăng nhập	<ul style="list-style-type: none"> · Người dùng có thể chọn "Ghi nhớ đăng nhập", hệ thống sẽ duy trì phiên bằng Firestore Authentication (ID Token hoặc Session Token). · Phiên đăng nhập sẽ tự động hết hạn sau 30 ngày nếu không có hoạt động. · Nếu người dùng đăng xuất thủ công, hệ thống sẽ xóa thông tin phiên khỏi thiết bị.

- Mục lục:

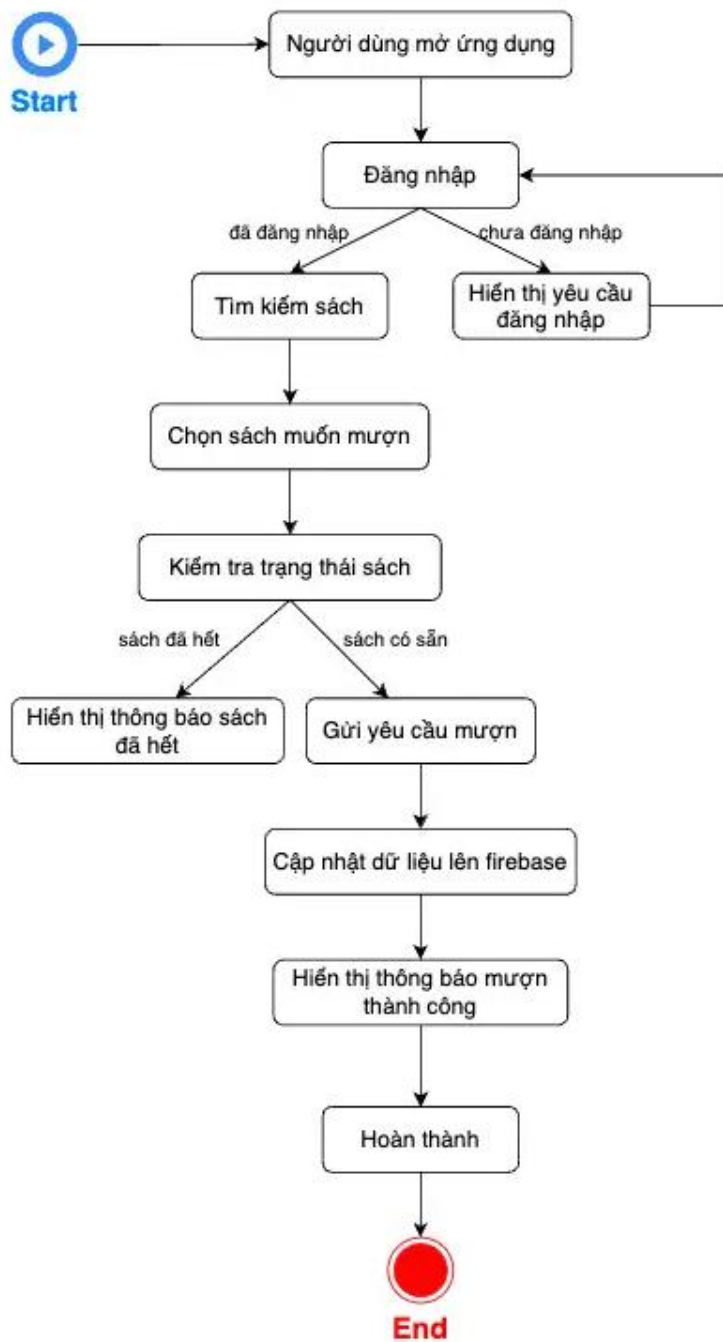
Màn hình giao diện:



Activity Flow:

Step	Description
1	Người dùng mở ứng dụng và đăng nhập. Nếu chưa đăng nhập, hệ thống hiển thị yêu cầu đăng nhập.
2	Sau khi đăng nhập thành công, người dùng tìm kiếm sách muốn mượn.
3	Người dùng chọn sách muốn mượn trong danh sách kết quả tìm kiếm.
4	Hệ thống kiểm tra trạng thái sách: <ul style="list-style-type: none"> - Nếu sách đã hết: Hiển thị thông báo sách đã hết. - Nếu sách có sẵn: Gửi yêu cầu mượn

5	Hệ thống cập nhật dữ liệu mượn sách lên Firebase.
6	Hiển thị thông báo mượn sách thành công.
7	Quy trình hoàn tất.



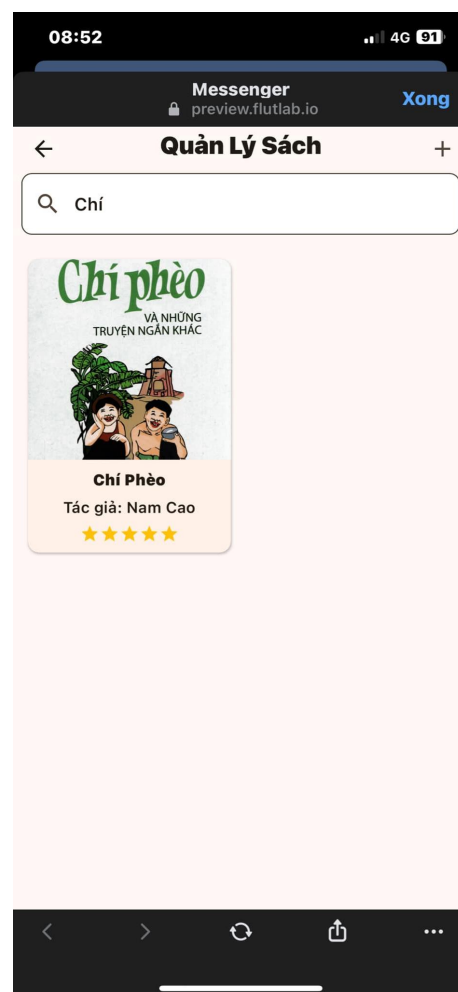
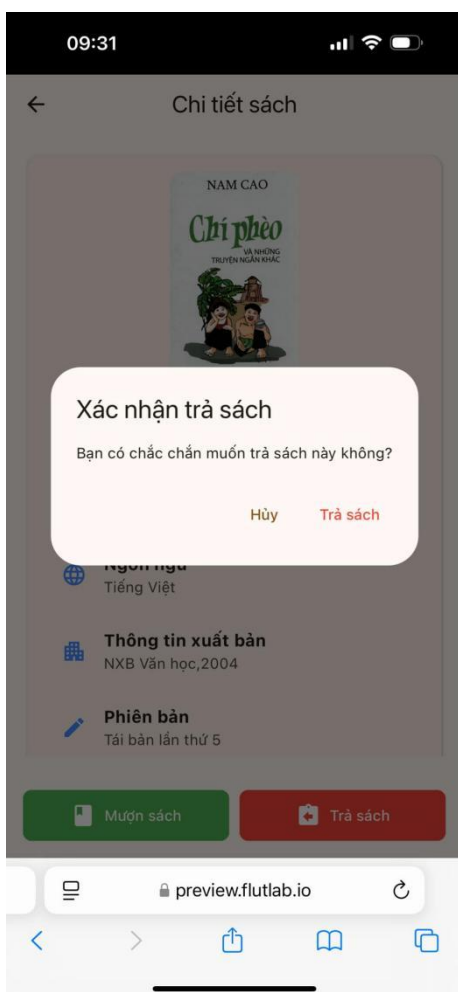
Business Rule:

Rule	Description
Quy tắc đủ điều kiện mượn sách	<ul style="list-style-type: none">· Chỉ thành viên đã đăng ký và được kích hoạt mới có thể mượn sách.· Thành viên không bị khóa tài khoản mới được phép mượn sách.· Thành viên không có sách quá hạn chưa trả mới được phép mượn thêm sách.· Số lượng sách còn lại trong kho phải lớn hơn 0 để có thể mượn.
Quy tắc số lượng và thời hạn mượn	<ul style="list-style-type: none">· Mỗi thành viên được mượn tối đa 5 quyển sách cùng lúc.· Thời hạn mượn sách: 14 ngày tính từ ngày mượn.· Có thể gia hạn tối đa 1 lần cho mỗi quyển sách, thêm 7 ngày nếu sách chưa có người đặt trước.
Quy tắc đặt trước sách	<ul style="list-style-type: none">· Nếu sách đã hết, thành viên có thể đặt trước.· Khi sách có sẵn, hệ thống gửi thông báo và giữ sách trong 48 giờ trước khi chuyển cho người đặt tiếp theo.· Thành viên không đến nhận trong thời gian giữ, lượt đặt trước bị hủy.
Quy tắc quá hạn và trễ hạn sách	<ul style="list-style-type: none">· Nếu sách quá hạn, hệ thống sẽ gửi nhắc nhở tự động qua email hoặc thông báo ứng dụng.· Nếu sách quá hạn trên 7 ngày, tài khoản bị tạm ngưng mượn sách mới cho đến khi trả sách.· Nếu sách quá hạn trên 30 ngày, hệ thống có thể áp dụng phí phạt hoặc yêu cầu đền bù (nếu có quy định).
Quy tắc xử lý ghi nhận lịch sử mượn	Hệ thống lưu lại lịch sử mượn của từng thành viên, bao gồm: <ul style="list-style-type: none">• Ngày mượn• Ngày phải trả• Tình trạng sách khi mượn/trả• Trạng thái quá hạn/phạt nếu có

<p>Quy tắc xử lý lỗi</p>	<ul style="list-style-type: none"> · Nếu sách không có sẵn, hiển thị: "Sách này hiện đã hết. Bạn có muốn đặt trước không?" · Nếu thành viên vượt quá số lượng mượn tối đa, hiển thị: "Bạn đã đạt giới hạn mượn sách. Vui lòng trả sách trước khi mượn thêm." · Nếu thành viên có sách quá hạn chưa trả, hiển thị: "Bạn có sách quá hạn. Vui lòng trả sách trước khi mượn tiếp." · Nếu có lỗi kết nối khi thực hiện mượn sách, hiển thị: "Không thể xử lý yêu cầu. Vui lòng kiểm tra lại kết nối."
---------------------------------	---

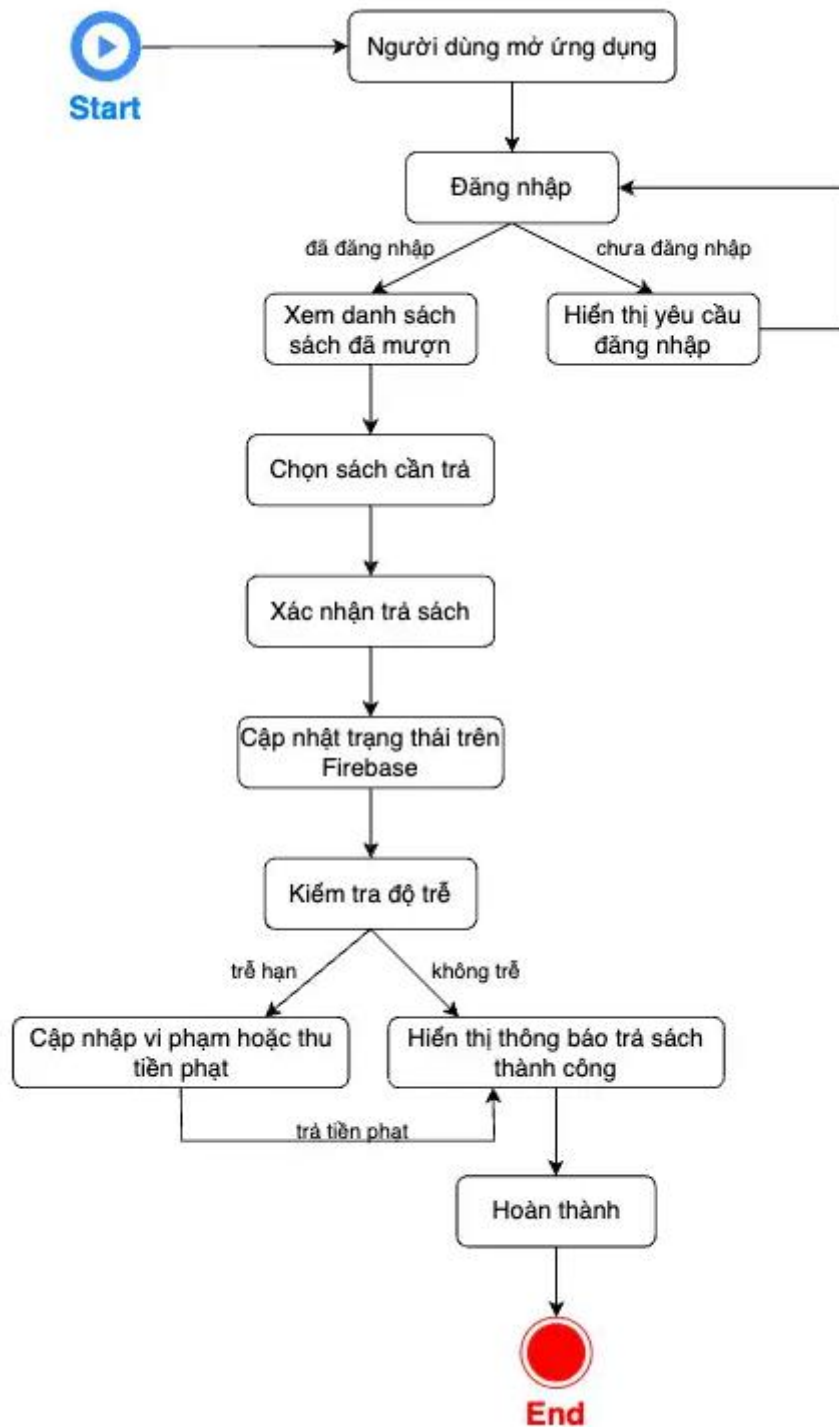
- Trả sách:

Màn hình giao diện:



Activity Flow:

Step	Description
1	Người dùng mở ứng dụng và đăng nhập. Nếu chưa đăng nhập, hệ thống hiển thị yêu cầu đăng nhập.
2	Sau khi đăng nhập thành công, hệ thống cho phép người dùng xem danh sách sách đã mượn.
3	Người dùng chọn sách cần trả từ danh sách.
4	Người dùng xác nhận trả sách. Hệ thống sẽ cập nhật trạng thái trả sách trên Firebase.
5	Hệ thống kiểm tra độ trễ: <ul style="list-style-type: none">- Nếu trả trễ: Ghi nhận vi phạm hoặc tính phạt- Nếu không trả trễ: Hiển thị thông báo trả sách thành công
6	Quy trình hoàn tất.



Business Rule:

Rule	Description
Quy tắc đủ điều kiện trả sách	<ul style="list-style-type: none"> Chỉ những sách được mượn trong hệ thống mới có thể trả. Người dùng chỉ có thể trả những sách do chính họ mượn.

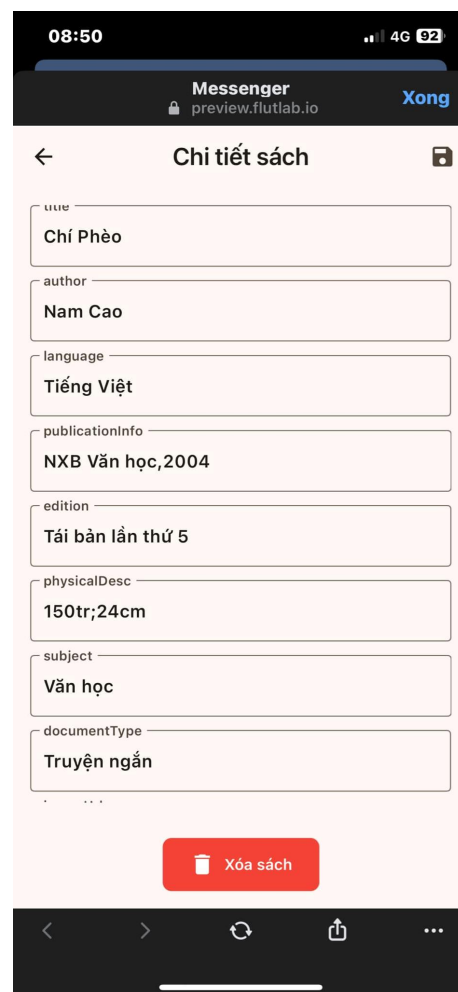
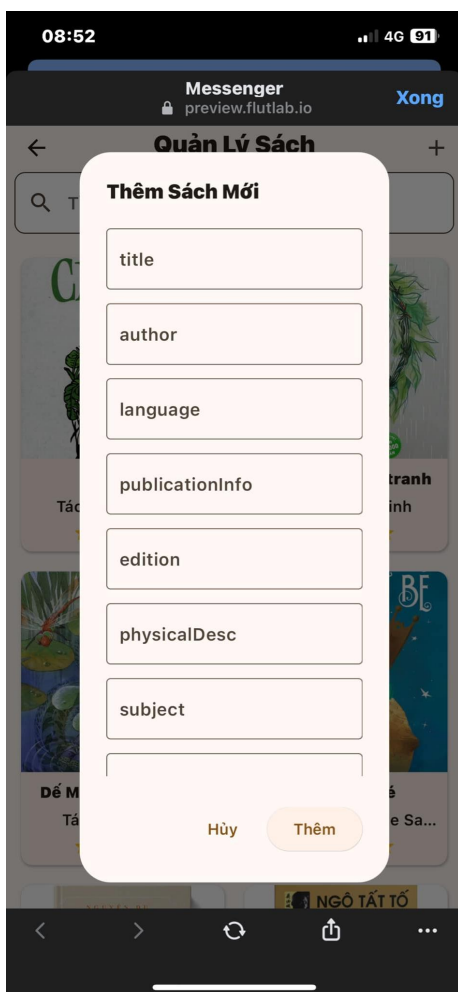
	<ul style="list-style-type: none"> · Nếu sách đã bị quá hạn, hệ thống sẽ kiểm tra và xử lý phí phạt (nếu có). · Nếu sách bị mất hoặc hư hỏng, thành viên phải bồi thường theo quy định trước khi hoàn tất trả sách.
Quy tắc xác nhận trả sách	<ul style="list-style-type: none"> · Thành viên có thể trả sách theo một trong hai cách: <ul style="list-style-type: none"> • Trực tiếp tại thư viện (quét mã sách hoặc nhập mã thủ công). • Qua kiosk/trạm trả sách tự động (nếu thư viện hỗ trợ). · Nhân viên thư viện cần kiểm tra tình trạng sách trước khi xác nhận trả. · Khi sách được trả thành công, hệ thống sẽ: <ul style="list-style-type: none"> • Tăng số lượng sách có sẵn. • Xóa trạng thái “Đang mượn” của sách khỏi tài khoản người dùng. • Ghi nhận lịch sử trả sách.
Quy tắc xử lý sách quá hạn	<ul style="list-style-type: none"> · Nếu sách trả trễ hạn dưới 7 ngày, chỉ gửi cảnh báo nhưng không áp dụng hình phạt. · Nếu sách trả trễ hạn từ 7 đến 30 ngày, tài khoản bị tạm ngưng mượn sách trong 7 ngày. · Nếu sách trả trễ hạn hơn 30 ngày, hệ thống có thể: <ul style="list-style-type: none"> • Áp dụng phí phạt tùy theo số ngày trễ hạn. • Yêu cầu thành viên mua sách thay thế nếu sách không thể sử dụng lại. • Hạn chế quyền mượn sách của thành viên nếu tái phạm nhiều lần.
Quy tắc ghi nhận lịch sử trả sách	<ul style="list-style-type: none"> · Mỗi lần trả sách, hệ thống lưu lại thông tin: <ul style="list-style-type: none"> • Ngày trả • Tình trạng sách khi trả • Trạng thái quá hạn hoặc phạt (nếu có) · Thành viên có thể xem lịch sử trả sách trong tài khoản của

	mình.
Quy tắc xử lý lỗi	<ul style="list-style-type: none"> · Nếu sách không tồn tại trong hệ thống, hiển thị: "Sách này không hợp lệ hoặc chưa được mượn từ hệ thống." · Nếu sách đã bị mất nhưng chưa được xử lý, hiển thị: "Sách này đã bị báo mất. Vui lòng liên hệ thư viện để giải quyết." · Nếu có lỗi kết nối khi xác nhận trả sách, hiển thị: "Không thể xử lý yêu cầu. Vui lòng thử lại sau."

b. Chức năng cho quản trị viên

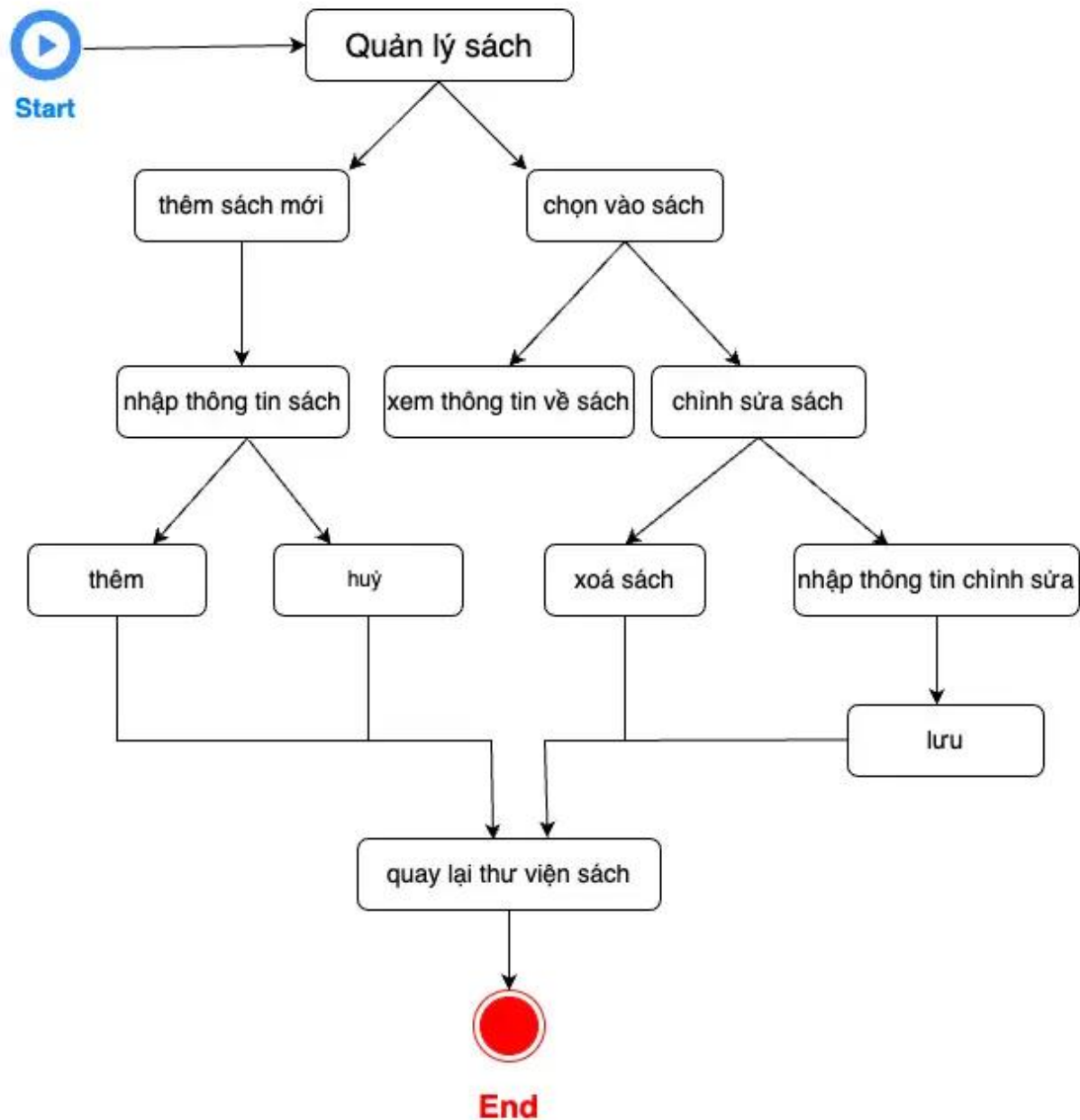
- Quản lý sách:

Màn hình giao diện:



Activity Flow:

Step	Description
1	Người dùng truy cập chức năng "Quản lý sách".
2	Người dùng chọn một trong hai hành động chính: - Thêm sách mới. - Chọn vào sách.
3	- Nếu chọn Thêm sách mới: + Nhập thông tin sách. + Người dùng có thể chọn "Thêm" để hoàn tất hoặc "Hủy" để thoát.
4	- Nếu chọn Chọn vào sách: + Người dùng có các tùy chọn: - Xem thông tin về sách: Chỉ hiển thị thông tin chi tiết của sách. - Chỉnh sửa sách: Nhập thông tin cần chỉnh sửa và chọn "Lưu". - Xóa sách: Xác nhận việc xóa sách khỏi hệ thống
5	Sau khi hoàn thành thao tác, hệ thống quay lại thư viện sách. Kết thúc quy trình quản lý sách.



Business Rule:

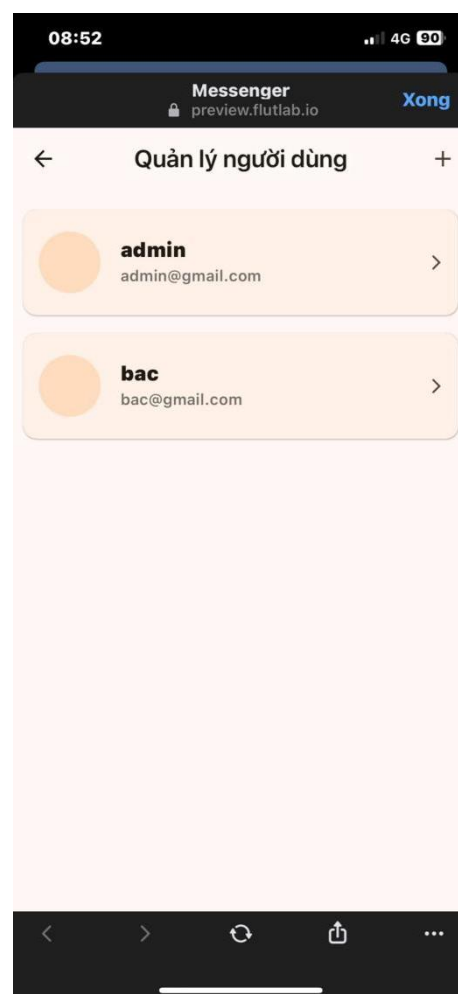
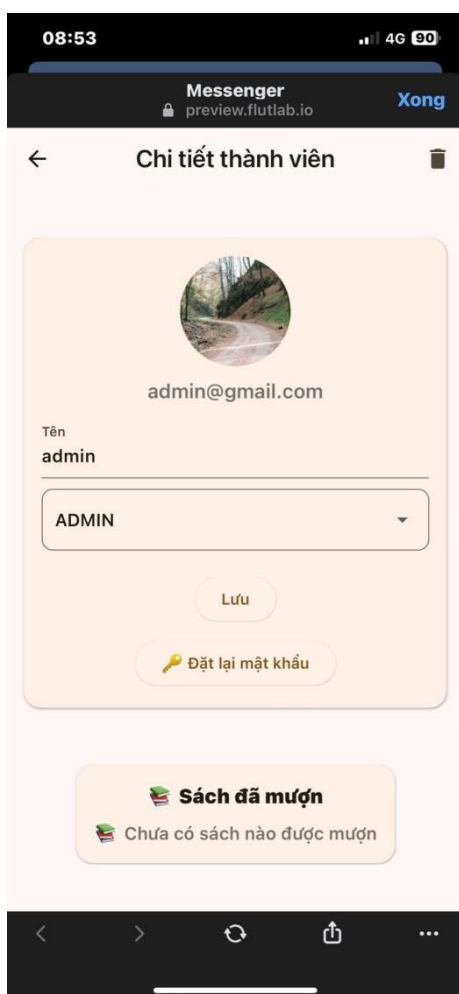
Rule	Description
Quy tắc nhập và cập nhật thông tin sách	<ul style="list-style-type: none"> · Quản trị viên hoặc nhân viên thư viện có quyền thêm, chỉnh sửa, xóa sách. · Thông tin bắt buộc khi thêm sách: <ul style="list-style-type: none"> • Mã sách (duy nhất, tự động sinh hoặc nhập thủ công). • Tên sách (tối đa 255 ký tự).

	<ul style="list-style-type: none"> • Tác giả (tối đa 100 ký tự, có thể có nhiều tác giả). • Thể loại (có thể chọn từ danh sách hoặc thêm mới). • Nhà xuất bản (tối đa 100 ký tự). • Năm xuất bản (chỉ nhận năm hợp lệ). • Số lượng sách có sẵn (số nguyên dương). <p>· Các thông tin tùy chọn:</p> <ul style="list-style-type: none"> • Mô tả sách (giới hạn 1000 ký tự). • Ảnh bìa sách (hỗ trợ upload ảnh). • Vị trí lưu trữ trong thư viện (kệ, khu vực, mã vị trí).
Quy tắc quản lý số lượng sách	<p>· Khi sách được mượn, hệ thống sẽ tự động giảm số lượng sách có sẵn.</p> <p>· Khi sách được trả, hệ thống sẽ tự động tăng số lượng sách có sẵn.</p> <p>· Nếu số lượng sách = 0, không cho phép mượn thêm.</p> <p>· Nếu sách bị mất/hỏng, quản trị viên có thể cập nhật số lượng và tình trạng sách.</p>
Quy tắc tìm kiếm và lọc sách	<p>· Người dùng có thể tìm kiếm sách theo:</p> <ul style="list-style-type: none"> • Tên sách, tác giả, mã sách, thể loại, nhà xuất bản. <p>· Hỗ trợ lọc sách theo:</p> <ul style="list-style-type: none"> • Thể loại, năm xuất bản, tình trạng sách (còn/trả hết), vị trí lưu trữ. <p>· Tìm kiếm phải có tốc độ phản hồi nhanh và hiển thị kết quả theo phân trang nếu danh sách quá dài.</p>
Quy tắc xóa sách	<p>· Chỉ quản trị viên có quyền xóa sách.</p> <p>· Không thể xóa sách nếu sách đang được mượn.</p> <p>· Khi xóa sách, hệ thống phải hiển thị cảnh báo xác nhận trước khi thực hiện.</p>
Quy tắc xử lý lỗi	<p>· Nếu người dùng nhập thiếu thông tin bắt buộc, hiển thị thông báo lỗi.</p> <p>· Nếu nhập trùng Mã sách, hiển thị thông báo: "Mã sách đã</p>

	<p>tồn tại. Vui lòng chọn mã khác."</p> <ul style="list-style-type: none"> · Nếu ảnh bìa không đúng định dạng (JPEG, PNG, WebP), hiển thị thông báo lỗi. · Nếu có lỗi kết nối khi tải danh sách sách, hiển thị thông báo: "Không thể tải danh sách sách. Vui lòng kiểm tra kết nối mạng."
--	---

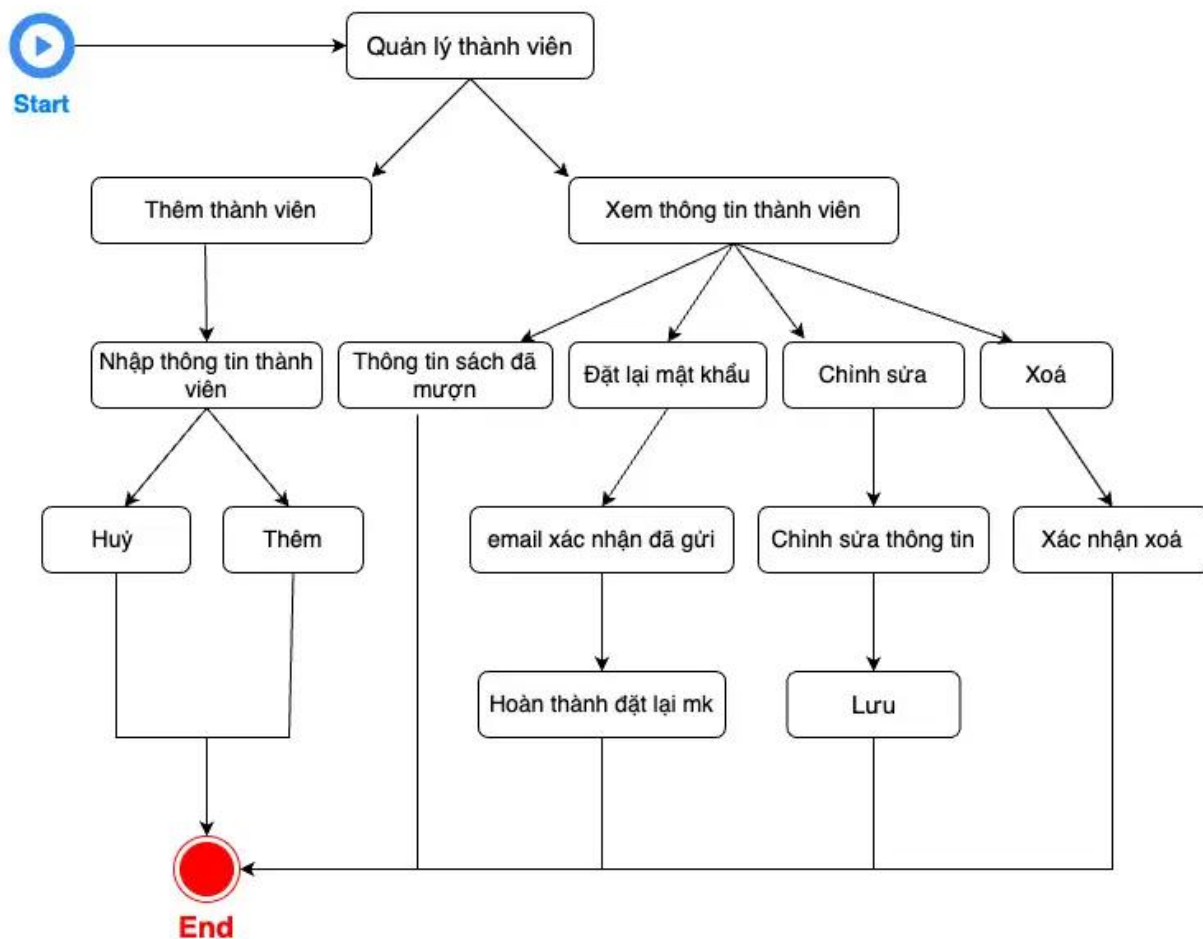
- Quản lý thành viên:

Màn hình giao diện:



Activity Flow:

Step	Description
1	Người dùng truy cập chức năng "Quản lý thành viên".
2	Chọn một trong hai hành động chính: <ul style="list-style-type: none">- Thêm thành viên.- Xem thông tin thành viên.
3	<ul style="list-style-type: none">- Nếu chọn Thêm thành viên:<ul style="list-style-type: none">+ Nhập thông tin thành viên.+ Người dùng có thể chọn "Hủy" để thoát hoặc "Thêm" để hoàn thành việc thêm thành viên.
4	<ul style="list-style-type: none">- Nếu chọn Xem thông tin thành viên:<ul style="list-style-type: none">+ Người dùng có các tùy chọn:<ul style="list-style-type: none">- Đặt lại mật khẩu: Gửi email xác nhận và hoàn thành việc đặt lại mật khẩu.- Chỉnh sửa thông tin: Nhập thông tin chỉnh sửa và chọn "Lưu".- Xóa thành viên: Xác nhận việc xóa thành viên.
5	Sau khi hoàn thành các thao tác, hệ thống sẽ kết thúc quy trình.



Business Rule:

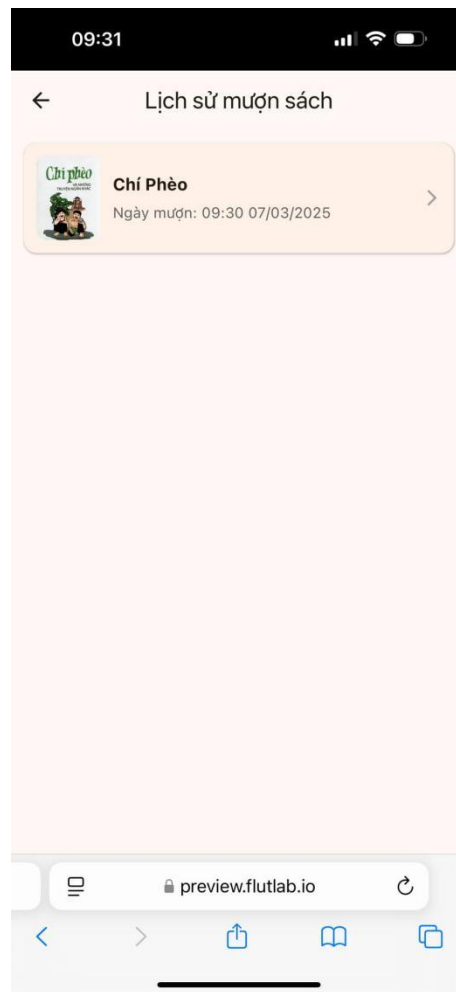
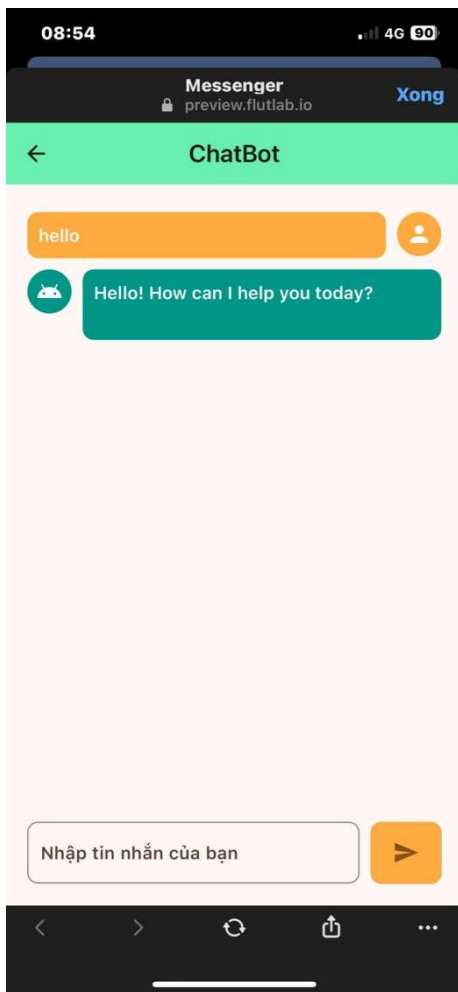
Rule	Description
Quy tắc phân quyền người dùng	<p>Hệ thống có 3 loại người dùng chính:</p> <ol style="list-style-type: none"> Quản trị viên (Admin): Toàn quyền quản lý hệ thống, bao gồm quản lý sách, người dùng, và thiết lập hệ thống. Nhân viên thư viện: Quản lý sách, xử lý mượn/trả sách, hỗ trợ thành viên. Thành viên (Người mượn sách): Có thể tìm kiếm, mượn, trả sách và xem lịch sử mượn.
Quy tắc đăng ký và kích hoạt tài khoản	<ul style="list-style-type: none"> Quản trị viên hoặc nhân viên thư viện có quyền thêm, chỉnh sửa, xóa sách. Thông tin bắt buộc khi thêm sách:

	<ul style="list-style-type: none"> • Mã sách (duy nhất, tự động sinh hoặc nhập thủ công). • Tên sách (tối đa 255 ký tự). • Tác giả (tối đa 100 ký tự, có thể có nhiều tác giả). • Thể loại (có thể chọn từ danh sách hoặc thêm mới). • Nhà xuất bản (tối đa 100 ký tự). • Năm xuất bản (chỉ nhận năm hợp lệ). • Số lượng sách có sẵn (số nguyên dương). <p>· Các thông tin tùy chọn:</p> <ul style="list-style-type: none"> • Mô tả sách (giới hạn 1000 ký tự). • Ảnh bìa sách (hỗ trợ upload ảnh). • Vị trí lưu trữ trong thư viện (kệ, khu vực, mã vị trí).
Quy tắc quản lý thông tin người dùng	<ul style="list-style-type: none"> · Khi sách được mượn, hệ thống sẽ tự động giảm số lượng sách có sẵn. · Khi sách được trả, hệ thống sẽ tự động tăng số lượng sách có sẵn. · Nếu số lượng sách = 0, không cho phép mượn thêm. · Nếu sách bị mất/hỏng, quản trị viên có thể cập nhật số lượng và tình trạng sách.
Quy tắc đặt lại mật khẩu	<ul style="list-style-type: none"> · Chỉ quản trị viên có quyền xóa sách. · Không thể xóa sách nếu sách đang được mượn. · Khi xóa sách, hệ thống phải hiển thị cảnh báo xác nhận trước khi thực hiện.
Quy tắc xóa tài khoản	<ul style="list-style-type: none"> · Nếu người dùng nhập thiếu thông tin bắt buộc, hiển thị thông báo lỗi. · Nếu nhập trùng Mã sách, hiển thị thông báo: "Mã sách đã tồn tại. Vui lòng chọn mã khác." · Nếu ảnh bìa không đúng định dạng (JPEG, PNG, WebP), hiển thị thông báo lỗi. · Nếu có lỗi kết nối khi tải danh sách sách, hiển thị thông báo: "Không thể tải danh sách sách. Vui lòng kiểm tra kết nối mạng."

Quy tắc khóa/tạm ngưng tài khoản	<ul style="list-style-type: none"> · Quản trị viên có thể tạm ngưng/kích hoạt lại tài khoản người dùng. · Khi tài khoản bị khóa: <ul style="list-style-type: none"> • Người dùng không thể đăng nhập hoặc mượn sách. • Hiện thị thông báo "Tài khoản của bạn đã bị khóa. Vui lòng liên hệ quản trị viên." · Nếu một tài khoản vi phạm quy định nhiều lần (ví dụ: mượn sách nhưng không trả), quản trị viên có thể cấm vĩnh viễn tài khoản đó.
Quy tắc tìm kiếm/lọc người dùng	<ul style="list-style-type: none"> · Quản trị viên có thể tìm kiếm người dùng theo: <ul style="list-style-type: none"> • Tên, email, số điện thoại, vai trò (Admin/Nhân viên/Thành viên). · Hỗ trợ lọc danh sách theo: <ul style="list-style-type: none"> • Trạng thái tài khoản (Hoạt động/Bị khóa). • Ngày đăng ký.
Quy tắc xử lý lỗi	<ul style="list-style-type: none"> · Nếu người dùng nhập thông tin không hợp lệ, hiện thị thông báo lỗi. · Nếu email đã tồn tại, hiện thị: "Email này đã được sử dụng. Vui lòng chọn email khác." · Nếu tài khoản bị khóa, hiện thị: "Tài khoản của bạn đã bị khóa. Liên hệ quản trị viên để biết thêm chi tiết." · Nếu lỗi kết nối khi tải danh sách người dùng, hiện thị: "Không thể tải danh sách người dùng. Vui lòng kiểm tra kết nối mạng."

- Quản lý mượn trả:

Màn hình giao diện:



Activity Flow:

Step	Description
1	Admin mở ứng dụng và đăng nhập. Nếu chưa đăng nhập, hệ thống hiển thị yêu cầu đăng nhập.
2	Sau khi đăng nhập thành công, admin xem danh sách sách đã được mượn.
3	Admin xem danh sách người mượn và ngày hết hạn trả sách.
4	Admin thực hiện một trong hai hành động sau:

	-Xác nhận trả sách thủ công nếu cần: Hệ thống cập nhật trạng thái sách -Kiểm tra sách trễ hạn
5	Nếu không trễ hạn: Không thực hiện gì
6	Nếu có trễ hạn: -Cập nhật vi phạm hoặc thu phí phạt -Gửi thông báo nhắc nhở
7	Quy trình kết thúc



Business Rule:

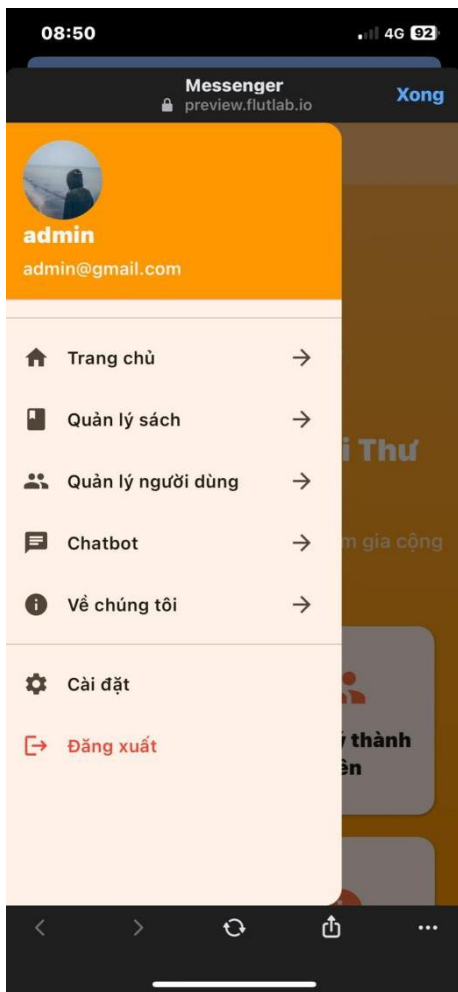
Rule	Description
Quy tắc phân quyền quản lý	<ul style="list-style-type: none"> · Nhân viên thư viện và quản trị viên có quyền quản lý mượn/trả sách. · Thành viên chỉ có thể xem lịch sử mượn/trả của chính họ,

	<p>không có quyền thay đổi dữ liệu.</p> <ul style="list-style-type: none"> · Quản trị viên có quyền cập nhật, chỉnh sửa thông tin mượn/trả trong trường hợp đặc biệt (sai sót dữ liệu, lỗi hệ thống).
Quy tắc kiểm soát mượn sách	<ul style="list-style-type: none"> · Thành viên phải đủ điều kiện mượn (không bị khóa tài khoản, không có sách quá hạn, chưa vượt quá số lượng mượn tối đa). · Số lượng sách mượn tối đa: 5 quyển/lần. · Thời hạn mượn: 14 ngày, có thể gia hạn 1 lần thêm 7 ngày nếu sách chưa có người đặt trước. · Nếu sách hết, thành viên có thể đặt trước và nhận thông báo khi sách có sẵn.
Quy tắc kiểm soát trả sách	<ul style="list-style-type: none"> · Sách chỉ được trả nếu đang trong trạng thái "Đang mượn" bởi thành viên. · Trả sách có thể thực hiện tại thư viện hoặc qua kiosk/trạm trả sách (nếu hỗ trợ). · Nhân viên kiểm tra tình trạng sách trước khi xác nhận trả. · Khi sách được trả thành công, hệ thống: <ul style="list-style-type: none"> • Cập nhật số lượng sách có sẵn. • Xóa trạng thái “Đang mượn” của sách khỏi tài khoản thành viên. • Lưu lịch sử mượn/trả.
Quy tắc xử lý quá hạn và phạt	<ul style="list-style-type: none"> · Nếu sách trả quá hạn dưới 7 ngày, chỉ gửi cảnh báo nhưng không phạt. · Nếu sách quá hạn từ 7 đến 30 ngày, tài khoản bị tạm ngưng mượn sách trong 7 ngày. · Nếu sách quá hạn hơn 30 ngày, hệ thống có thể: <ul style="list-style-type: none"> • Áp dụng phí phạt theo số ngày trễ. • Yêu cầu thành viên mua sách thay thế nếu sách không thể sử dụng lại. • Hạn chế quyền mượn sách nếu tái phạm nhiều lần.

Quy tắc ghi nhận lịch sử mượn/trả	<ul style="list-style-type: none"> · Hệ thống ghi nhận mọi giao dịch mượn/trả, bao gồm: <ul style="list-style-type: none"> • Tên sách, mã sách • Tên người mượn • Ngày mượn, hạn trả, ngày trả thực tế • Tình trạng sách khi trả • Trạng thái quá hạn hoặc phạt (nếu có) · Thành viên có thể xem lịch sử của họ trong tài khoản. · Nhân viên thư viện có thể lọc danh sách mượn/trả theo ngày, trạng thái, người dùng.
Quy tắc xử lý lỗi	<ul style="list-style-type: none"> · Nếu sách không tồn tại trong hệ thống, hiển thị: "Sách không hợp lệ hoặc chưa được mượn từ hệ thống." · Nếu thành viên không đủ điều kiện mượn, hiển thị thông báo lỗi cụ thể (quá hạn, đã mượn đủ số lượng, tài khoản bị khóa). · Nếu có lỗi kết nối khi cập nhật dữ liệu, hiển thị: "Không thể xử lý yêu cầu. Vui lòng thử lại sau."

- **Thông kê số lượng:**

Màn hình giao diện:



Activity Flow:

Step	Description
1	Admin mở ứng dụng và đăng nhập. Nếu chưa đăng nhập, hệ thống hiển thị yêu cầu đăng nhập.
2	Sau khi đăng nhập thành công, admin chọn mục Thống kê trong menu.
3	Hệ thống hiển thị các tùy chọn thống kê, bao gồm: <ul style="list-style-type: none"> - Tổng số sách trong thư viện - Số lượng sách đã mượn - Số lượng sách trễ hạn - Số lượng thành viên đang hoạt động

	- Doanh thu từ phí phạt (nếu có)
4	Admin chọn một tùy chọn thống kê cụ thể.
5	Hệ thống hiển thị biểu đồ hoặc danh sách chi tiết về thông tin đã chọn.
6	Nếu cần, admin có thể lọc theo ngày, tháng, năm hoặc xuất dữ liệu thống kê.
7	Quy trình kết thúc.

Business Rule:

Rule	Description
Quy tắc chung	<ul style="list-style-type: none"> · Người dùng phải đăng nhập vào hệ thống để sử dụng chức năng thống kê. · Chỉ quản trị viên hoặc nhân viên thư viện mới có quyền xem thống kê chi tiết. · Dữ liệu thống kê được tự động cập nhật theo dữ liệu mới nhất trong Firebase. · Các số liệu thống kê chỉ hiển thị dữ liệu hợp lệ (không tính sách bị xóa hoặc mượn/trả không hợp lệ).
Quy tắc lấy dữ liệu thống kê	<ul style="list-style-type: none"> · Thống kê số lượng sách theo thể loại: <ul style="list-style-type: none"> • Chỉ tính sách còn tồn kho hoặc có trạng thái "đang mượn". · Thống kê số lượt mượn/trả: <ul style="list-style-type: none"> • Chỉ tính giao dịch có trạng thái "đã trả" hoặc "đang mượn". • Không tính giao dịch bị hủy hoặc lỗi. · Thống kê người dùng mượn sách nhiều nhất: <ul style="list-style-type: none"> • Chỉ tính người dùng có ít nhất một lượt mượn thành

	<p>công.</p> <ul style="list-style-type: none"> · Thống kê sách được mượn nhiều nhất: <ul style="list-style-type: none"> • Chỉ tính sách có số lần mượn > 0 trong khoảng thời gian được chọn.
Quy tắc lọc dữ liệu	<ul style="list-style-type: none"> · Mặc định hệ thống hiển thị thống kê trong 30 ngày gần nhất. · Người dùng có thể lọc theo các khoảng thời gian: <ul style="list-style-type: none"> • 7 ngày qua • Tháng này • Năm nay • Khoảng thời gian tùy chỉnh
Quy tắc xuất báo cáo	<ul style="list-style-type: none"> · Báo cáo có thể được xuất dưới dạng PDF hoặc Excel (CSV). · Khi xuất báo cáo, hệ thống sẽ ghi lại lịch sử xuất báo cáo kèm thời gian và người thực hiện. · Báo cáo chỉ chứa dữ liệu đã lọc theo thời gian mà người dùng đã chọn.

IV. KẾT LUẬN

1. Kết quả thu được

Ứng dụng quản lý thư viện được phát triển bằng Flutter và Firebase đã hoàn thành các mục tiêu đề ra, mang lại nhiều tiện ích cho việc quản lý sách và người dùng. Cụ thể, ứng dụng đã đạt được những kết quả quan trọng sau:

- **Quản lý sách hiệu quả:** Hệ thống cho phép thêm, chỉnh sửa, xóa và tìm kiếm sách một cách nhanh chóng, giúp người dùng dễ dàng quản lý danh mục sách trong thư viện.

- **Hỗ trợ mượn/trả sách:** Chức năng quản lý mượn/trả sách giúp theo dõi tình trạng sách trong thời gian thực, cập nhật lịch sử mượn và trả sách, đảm bảo tính minh bạch và chính xác trong việc quản lý thư viện.
- **Quản lý người dùng:** Hệ thống hỗ trợ đăng ký, đăng nhập, phân quyền giữa quản trị viên và độc giả, giúp việc quản lý người dùng trở nên hiệu quả và bảo mật hơn.
- **Lưu trữ dữ liệu trên đám mây:** Firebase Firestore được sử dụng làm cơ sở dữ liệu chính, giúp lưu trữ và truy xuất dữ liệu theo thời gian thực, đảm bảo tính ổn định và khả năng mở rộng của hệ thống.
- **Thông báo & nhắc nhở:** Ứng dụng tích hợp Firebase Cloud Messaging (FCM) để gửi thông báo nhắc nhở người dùng về thời hạn trả sách, giúp giảm thiểu tình trạng trễ hạn.
- **Giao diện thân thiện, đa nền tảng:** Sử dụng Flutter giúp ứng dụng có giao diện đẹp, dễ sử dụng và hoạt động mượt mà trên cả hai hệ điều hành Android và iOS.

Những kết quả trên cho thấy ứng dụng đã đáp ứng tốt nhu cầu của một hệ thống quản lý thư viện hiện đại, mang lại trải nghiệm thuận tiện cho người dùng và hỗ trợ tối ưu cho hoạt động của thư viện.

2. Hạn chế của ứng dụng

Mặc dù đã đạt được nhiều kết quả tích cực, ứng dụng vẫn còn một số hạn chế cần được cải thiện trong tương lai để nâng cao hiệu suất và trải nghiệm người dùng:

- **Chưa hỗ trợ ngoại tuyến:** Hiện tại, người dùng cần có kết nối internet để truy cập dữ liệu, điều này gây bất tiện trong một số trường hợp không có mạng.
- **Hiệu suất khi xử lý dữ liệu lớn:** Khi số lượng sách và người dùng tăng lên, Firebase Firestore có thể gặp độ trễ nếu truy vấn không được tối ưu hóa, ảnh hưởng đến tốc độ phản hồi của hệ thống.

- **Chưa tích hợp AI đề xuất sách:** Hệ thống chưa có tính năng gợi ý sách dựa trên sở thích của người dùng, trong khi đây là một chức năng hữu ích giúp cá nhân hóa trải nghiệm đọc.
- **Thiếu tính năng quét mã vạch:** Hiện tại, việc nhập sách vẫn được thực hiện thủ công, chưa hỗ trợ quét mã ISBN để tự động nhận diện thông tin sách, gây mất thời gian cho người quản lý thư viện.

Những hạn chế này là cơ sở để định hướng cho các cải tiến trong tương lai, giúp ứng dụng trở nên hoàn thiện hơn.

3. Hướng phát triển đề tài

Để nâng cao chất lượng ứng dụng, trong các phiên bản tiếp theo, có thể phát triển thêm các tính năng và cải tiến sau:

- **Hỗ trợ ngoại tuyến:** Tích hợp SQLite hoặc Hive để lưu trữ dữ liệu cục bộ, giúp người dùng có thể truy cập thông tin ngay cả khi không có kết nối internet. Dữ liệu sẽ được đồng bộ hóa tự động khi mạng được khôi phục.
- **Tối ưu hóa hiệu suất:** Cải thiện truy vấn Firestore bằng cách sử dụng pagination (phân trang) để giảm tải dữ liệu được tải về một lúc, đồng thời áp dụng bộ nhớ đệm để tăng tốc độ truy xuất dữ liệu.
- **Tích hợp AI đề xuất sách:** Sử dụng Machine Learning để phân tích lịch sử mượn sách và đề xuất sách phù hợp với sở thích của từng người dùng, giúp nâng cao trải nghiệm đọc.
- **Bổ sung tính năng quét mã vạch:** Áp dụng Google ML Kit để nhận diện mã ISBN từ bìa sách, giúp tự động điền thông tin sách thay vì nhập thủ công, giúp tiết kiệm thời gian và giảm sai sót.
- **Nâng cấp giao diện:** Cải thiện UI/UX bằng cách áp dụng Flutter animations và Material 3 để tạo ra trải nghiệm trực quan, sinh động hơn, đồng thời tối ưu hóa thiết kế giúp người dùng thao tác dễ dàng hơn.

Với những hướng phát triển này, ứng dụng hứa hẹn sẽ ngày càng hoàn thiện, cung cấp trải nghiệm tốt hơn cho người dùng và đáp ứng nhu cầu của các thư viện hiện đại.

4. Tổng kết

Ứng dụng quản lý thư viện được phát triển bằng Flutter và Firebase đã đáp ứng được các yêu cầu quan trọng trong việc quản lý sách, người dùng và quá trình mượn/trả sách. Nhờ vào khả năng lưu trữ dữ liệu trên đám mây và hỗ trợ đa nền tảng, ứng dụng mang lại sự tiện lợi, linh hoạt cho cả quản trị viên và độc giả.

Tuy nhiên, vẫn còn một số hạn chế cần được khắc phục, như khả năng hoạt động ngoại tuyến, hiệu suất khi xử lý dữ liệu lớn, thiếu tính năng quét mã vạch và AI đề xuất sách. Những điểm này sẽ là trọng tâm để cải tiến trong các phiên bản tiếp theo, nhằm nâng cao trải nghiệm người dùng và tối ưu hóa hệ thống.

Trong tương lai, việc bổ sung các công nghệ như SQLite/Hive để hỗ trợ ngoại tuyến, tối ưu hóa truy vấn Firestore, tích hợp AI và quét mã vạch sẽ giúp ứng dụng trở nên toàn diện hơn. Đồng thời, nâng cấp UI/UX với Material 3 và Flutter animations sẽ giúp ứng dụng trở nên chuyên nghiệp và thân thiện hơn với người dùng.

Nhìn chung, ứng dụng đã hoàn thành tốt các mục tiêu đề ra và có tiềm năng phát triển mạnh mẽ trong thời gian tới, góp phần hiện đại hóa hoạt động quản lý thư viện và mang lại nhiều giá trị cho người dùng.

TÀI LIỆU THAM KHẢO

1. Barry Burd (2023). Flutter For Dummies.
2. Mark Clow (2021). Flutter Cookbook: Over 100 recipes to build modern, fast, and dynamic mobile applications with Flutter.
3. Laurence Moroney (2017). The Definitive Guide to Firebase.
4. Gautam Reddy & S. Patel (2022). "A Comparative Study of Mobile App Development Frameworks: Flutter vs React Native"

