

# BÁO CÁO DỰ ÁN CUỐI KỲ

HỌC KỲ 1, NĂM HỌC 2024-2025

CT484: PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG

- Tên dự án/ứng dụng: Ứng dụng quản lý mượn sách
- Link GitHub mã nguồn: [CT48402-Project](#)
- Link Youtube: <https://youtu.be/OJkspoATN6M>
- MSSV 1: B2111797
- Họ tên SV 1: Lê Văn Hậu
- MSSV 2: B2111827
- Họ tên SV 2: Phan Văn Trường
- Lớp học phần: CT482-02

## I. Tổng quan

- **Miêu tả dự án/ứng dụng:** Ứng dụng quản lý mượn sách thường được sử dụng ở những nơi như thư viện giúp người quản lý thư viện có thể quản lý việc cho mượn sách một cách dễ dàng hơn. Ứng dụng quản lý các đối tượng như sách, thành viên thư viện và phiếu mượn.
- Bảng phân công công việc nếu làm nhóm.

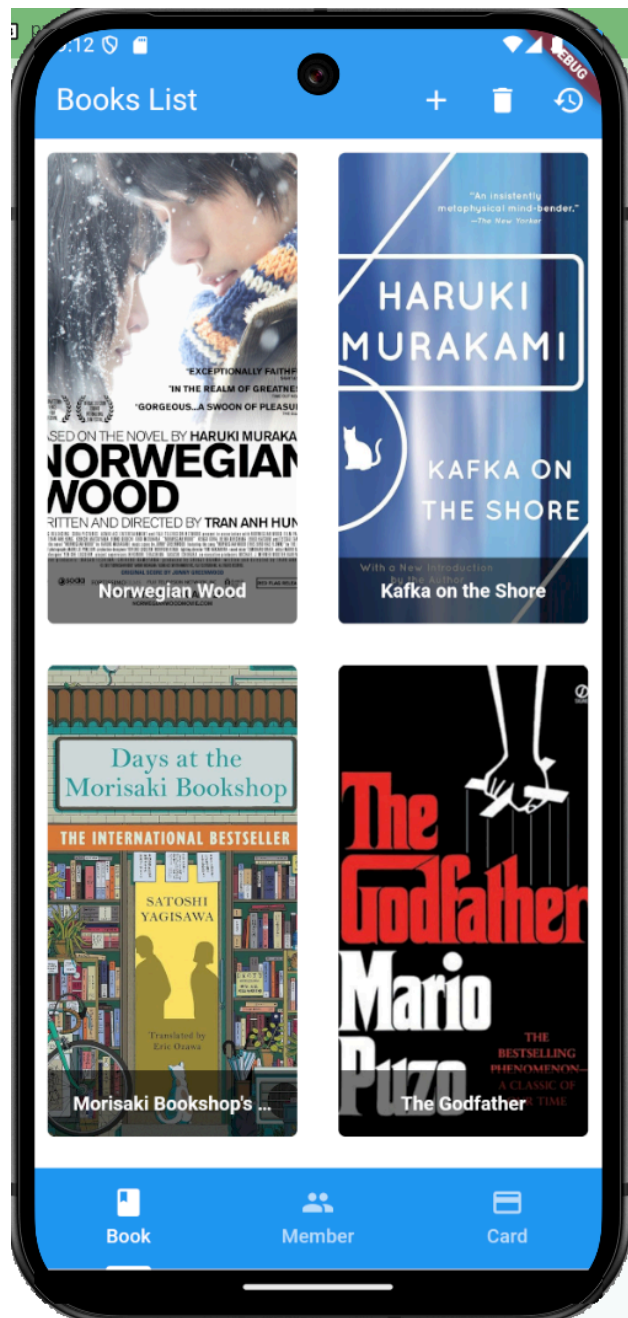
Công việc (bao gồm cả UI và quản lý state, dữ liệu,... nếu có)	Lê Văn Hậu	Phan Văn Trường
Đăng nhập		x
Quản lý sách		x
Quản lý thành viên	x	
Quản lý phiếu mượn	x	

## II. Chi tiết các chức năng

### 1. Giao diện 1: Hiện thị danh sách tất cả sách của thư viện.

- **Miêu tả giao diện:** giao diện hiển thị tất cả các sách được lưu từ cơ sở dữ liệu.

- Ảnh giao diện:



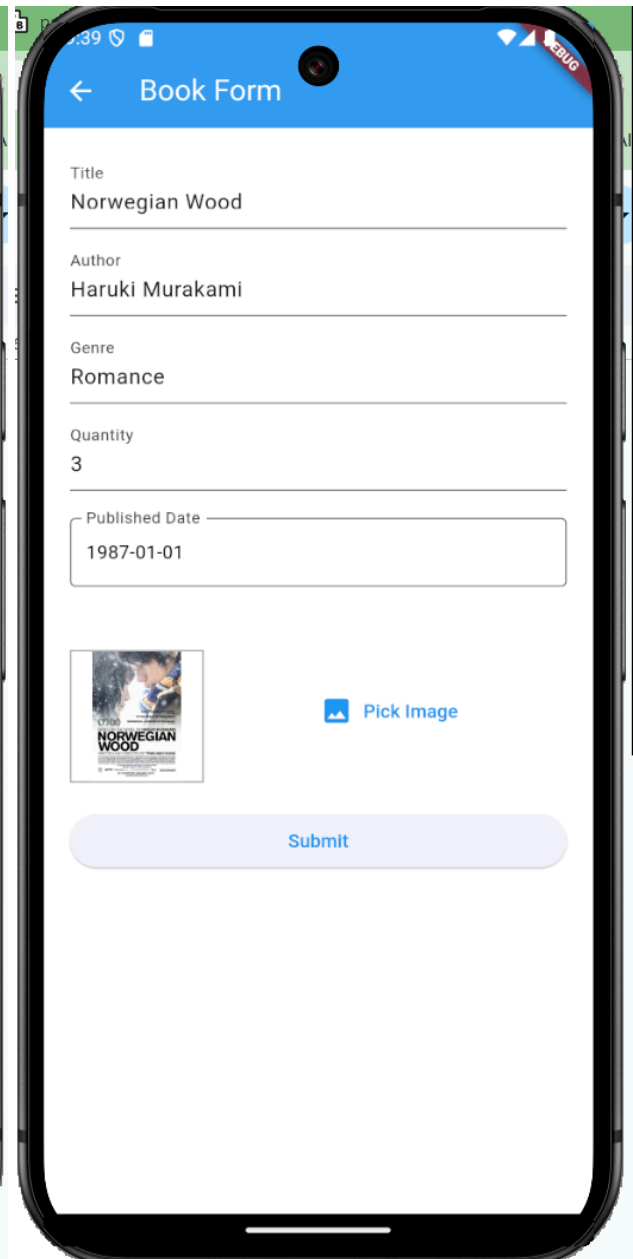
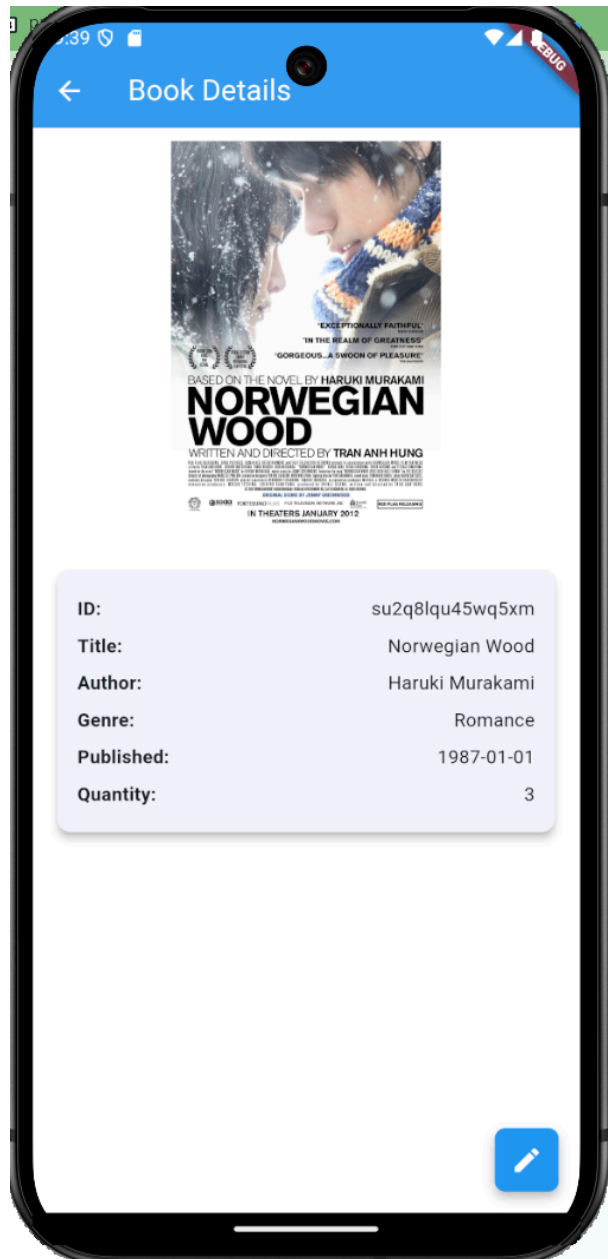
- Chi tiết cài đặt:

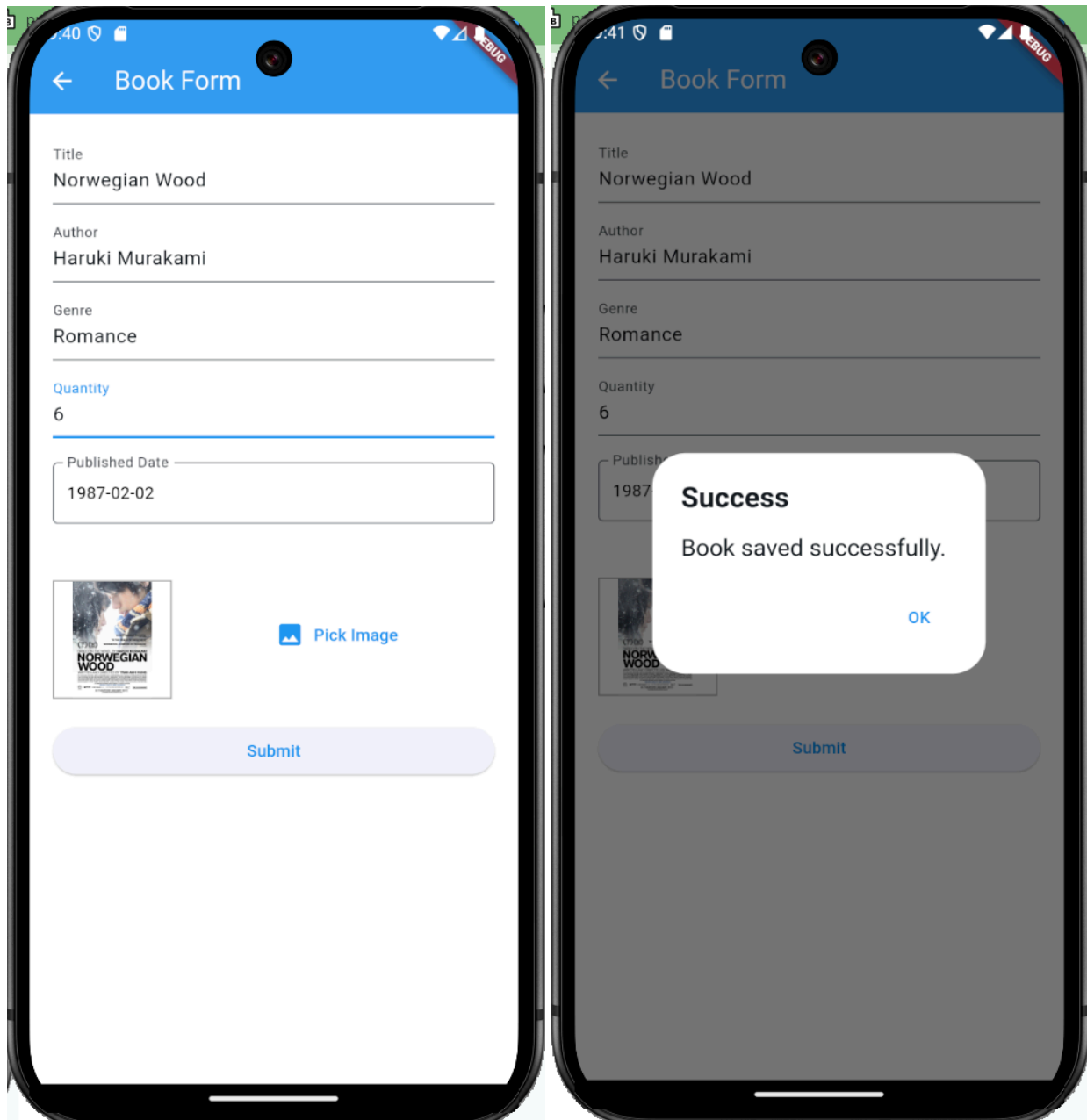
- + các widget sử dụng : AddBookButton, DeleteBookButton, RestoreBookButton, Text, Icon Button, CircularProgressIndicator, FutureBuilder, BooksGrid.
- + Thư viện:

- + Provider: Dùng để quản lý trạng thái ứng dụng, cung cấp cơ chế quản lý trạng thái giữa các widget trong ứng dụng. Trong code này, nó được sử dụng để truy cập vào BooksManager, giúp quản lý sách.
- + http: thư viện được sử dụng để thực hiện các yêu cầu HTTP (GET, POST, PUT, DELETE...) trong ứng dụng Flutter. Nó giúp giao tiếp với các API web hoặc máy chủ từ xa, cho phép bạn gửi và nhận dữ liệu dưới dạng JSON hoặc các định dạng khác.
- + PocketBase: là một backend-as-a-service (BaaS) đơn giản nhưng mạnh mẽ, cung cấp các dịch vụ như cơ sở dữ liệu, xác thực người dùng, và lưu trữ tệp tin. Thư viện pocketbase trong Flutter giúp kết nối và tương tác với PocketBase backend.
- + Quản lý chia sẻ: sử dụng Provider để quản lý trạng thái chia sẻ. Provider được sử dụng trong BooksListView để truy cập vào BooksManager thông qua `context.read<BooksManager>()`. Điều này cho phép bạn truy cập vào các phương thức và dữ liệu của BooksManager để thực hiện các thao tác lấy dữ liệu từ hệ thống.
- + Sử dụng dịch vụ lưu trữ Pocketbase. Lấy dữ liệu sách đã lưu từ pocketbase về và hiển thị với BookGrid.

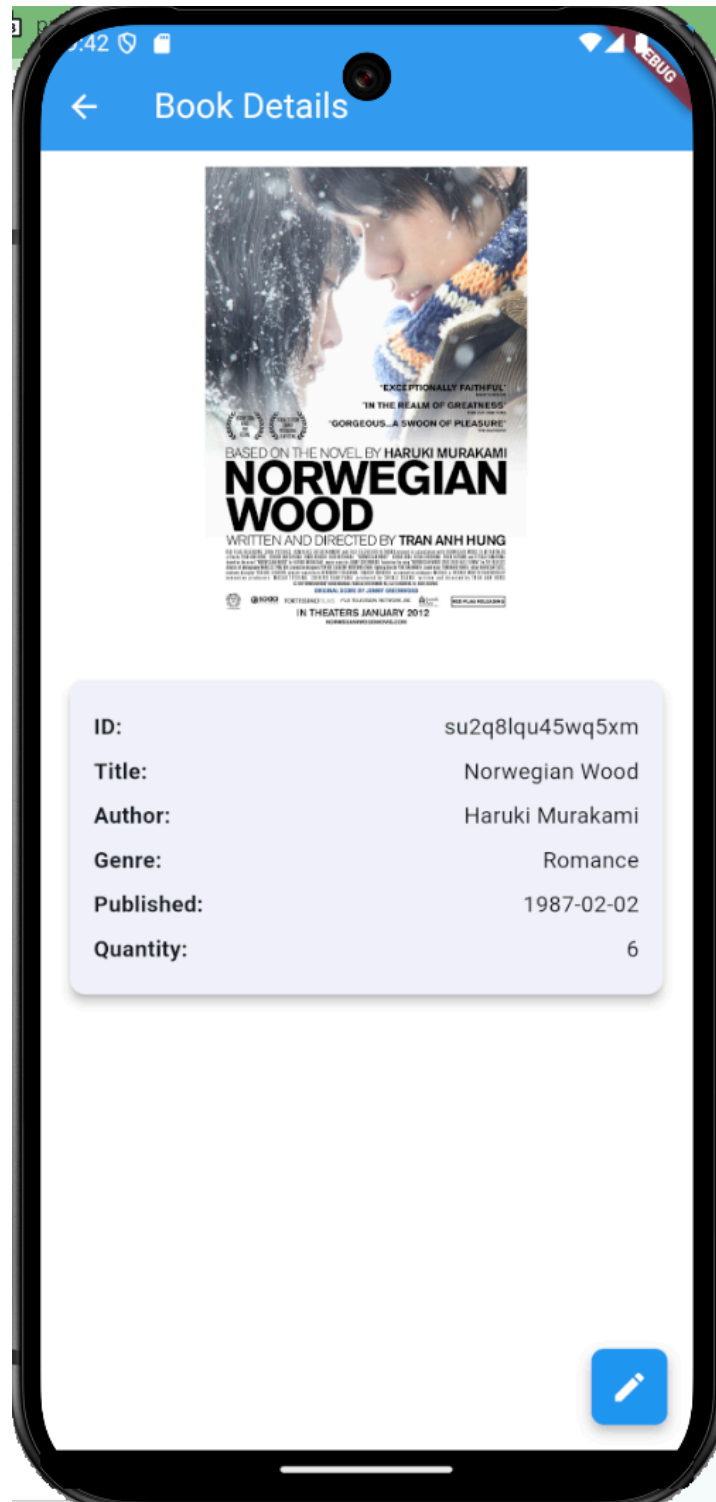
## 2. Giao diện 2: Chỉnh sửa thông tin sách

- **Miêu tả giao diện:** khi người dùng chọn icon Edit trong trang thông tin chi tiết sách sẽ chuyển tới giao diện chỉnh sửa thông tin sách.
- **Ảnh giao diện:**





+ Chỉnh sửa thành công



- Chi tiết cài đặt:

- + Các widget sử dụng: Padding, ListView, TextFormField, ElevatedButton, buildBookPreview hiển thị ảnh đã được lưu theo sách, buildDatePicker hiển thị khung chọn ngày tháng năm, TextFormField, SizedBox, InputDecorator, Center, FittedBox, Expanded, Image.file, AlertDialog.
- + Thư viện:
  - + image\_picker: Dùng để chọn ảnh từ thư viện hoặc máy ảnh.
  - + intl: Cung cấp khả năng định dạng ngày giờ, tiền tệ, và các yếu tố ngôn ngữ khác.
  - + Provider: Dùng để quản lý trạng thái ứng dụng, cung cấp cơ chế quản lý trạng thái giữa các widget trong ứng dụng. Trong code này, nó được sử dụng để truy cập vào BooksManager, giúp quản lý và chỉnh sửa sách từ danh sách quản lý.
  - + http: thư viện được sử dụng để thực hiện các yêu cầu HTTP (GET, POST, PUT, DELETE...) trong ứng dụng Flutter. Nó giúp giao tiếp với các API web hoặc máy chủ từ xa, cho phép bạn gửi và nhận dữ liệu dưới dạng JSON hoặc các định dạng khác.
  - + PocketBase: là một backend-as-a-service (BaaS) đơn giản nhưng mạnh mẽ, cung cấp các dịch vụ như cơ sở dữ liệu, xác thực người dùng, và lưu trữ tệp tin. Thư viện pocketbase trong Flutter giúp kết nối và tương tác với PocketBase backend.
- + Quản lý chia sẻ: sử dụng Provider để quản lý trạng thái chia sẻ. Provider được sử dụng trong BookForm để truy cập vào BooksManager thông qua Provider.of<BooksManager>(context, listen: false). Điều này cho phép bạn truy cập vào các phương thức và dữ liệu của BooksManager để thực hiện các thao tác như chỉnh sửa sách trong hệ thống.

- + Sử dụng dịch vụ lưu trữ Pocketbase. Khi chỉnh sửa sách sẽ thực hiện gọi api từ pocketbase để chỉnh sửa sách.

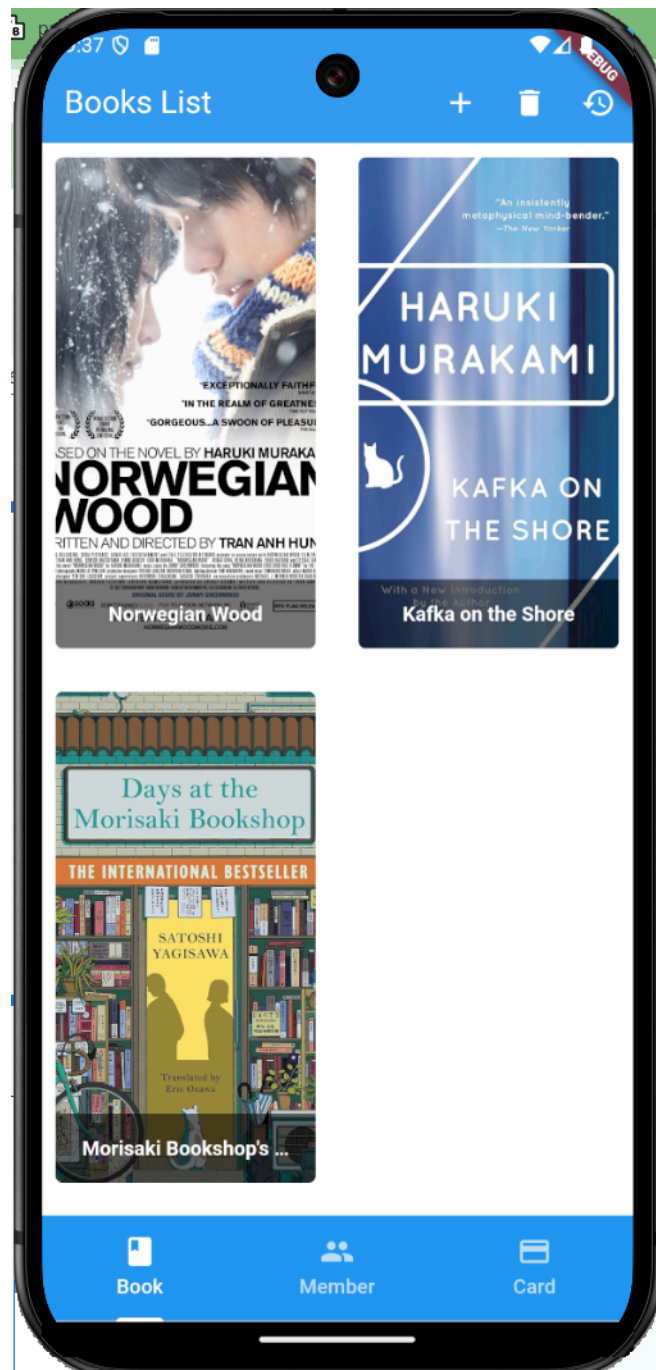
### 3. Giao diện 3: Xóa sách

- **Miêu tả giao diện:** khi người dùng chọn icon delete sẽ chuyển sang giao diện xóa sách.
- **Ảnh giao diện:**





- + Xóa thành công.




- Chi tiết cài đặt:

- + Các widget sử dụng: ListView, Card, Checkbox, ElevatedButton, FutureBuilder, Center, CircularProgressIndicator, Column, Expanded, RoundedRectangleBorder, SnackBar.

- + Thư viện:
  - + Provider: Dùng để quản lý trạng thái ứng dụng, cung cấp cơ chế quản lý trạng thái giữa các widget trong ứng dụng. Trong code này, nó được sử dụng để truy cập vào BooksManager, giúp quản lý và chỉnh sửa sách từ danh sách quản lý.
  - + http: thư viện được sử dụng để thực hiện các yêu cầu HTTP (GET, POST, PUT, DELETE...) trong ứng dụng Flutter. Nó giúp giao tiếp với các API web hoặc máy chủ từ xa, cho phép bạn gửi và nhận dữ liệu dưới dạng JSON hoặc các định dạng khác.
  - + PocketBase: là một backend-as-a-service (BaaS) đơn giản nhưng mạnh mẽ, cung cấp các dịch vụ như cơ sở dữ liệu, xác thực người dùng, và lưu trữ tệp tin. Thư viện pocketbase trong Flutter giúp kết nối và tương tác với PocketBase backend.
- + Quản lý chia sẻ: sử dụng Provider để quản lý trạng thái chia sẻ. Provider được sử dụng trong DeleteBookScreen để truy cập vào BooksManager thông qua context.read<BooksManager>(). Điều này cho phép bạn truy cập vào các phương thức và dữ liệu của BooksManager để thực hiện các thao tác như chỉnh sửa sách (thay đổi deleted) trong hệ thống.
- + Sử dụng dịch vụ lưu trữ Pocketbase. Khi xóa sẽ thực hiện gọi api từ pocketbase để thay đổi deleted của sách.

#### 4. Giao diện 4: Thêm thành viên

- **Miêu tả giao diện:** hiển thị form thêm mới thành viên và thực hiện submit form để thêm thành viên.
- **Ảnh giao diện:**

10:06 100% 

← Add Member

First Name

Last Name

Phone Number

Email


Citizen ID

Package


Total (Money)

0 VND

Enter a URL

 Take Picture

Add Member

10:14 100% 

← Add Member

First Name

Hau Van

Last Name

Le

Phone Number

0939111111

Email

hau111@gmail.com

Citizen ID


092203000111


Package

6 Months

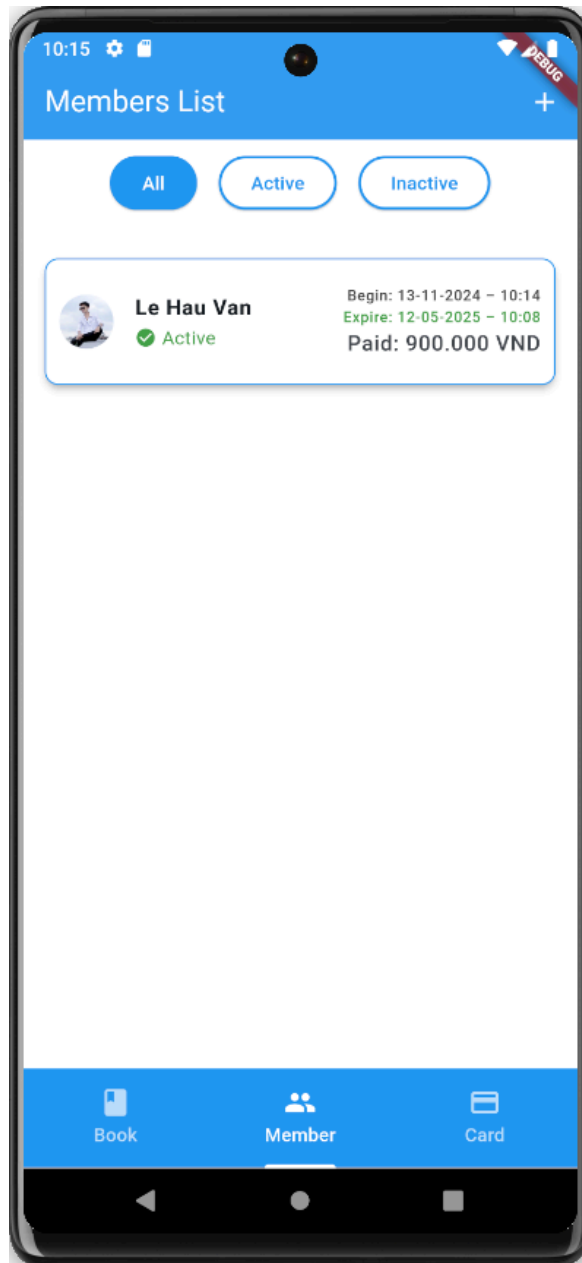
Total (Money)

900.000 VND



 Take Picture

Add Member



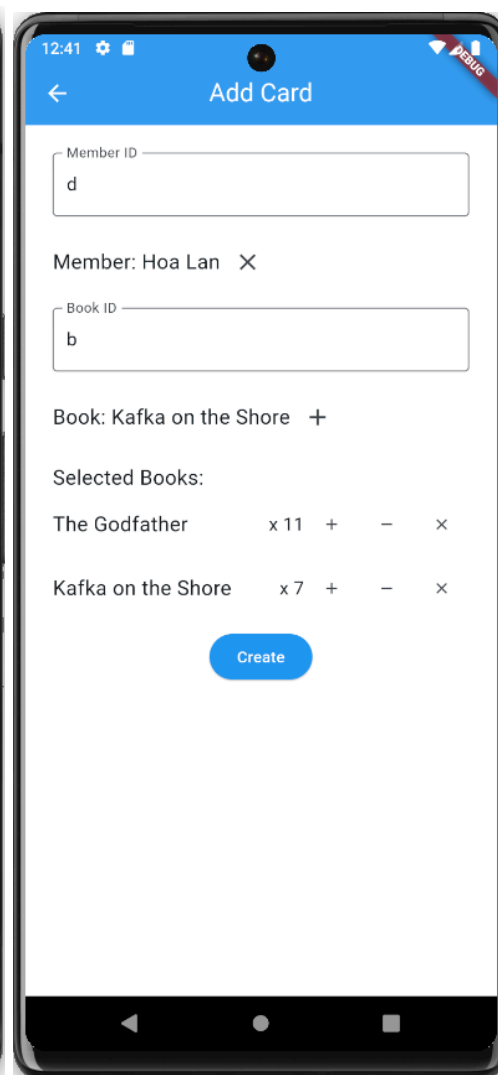
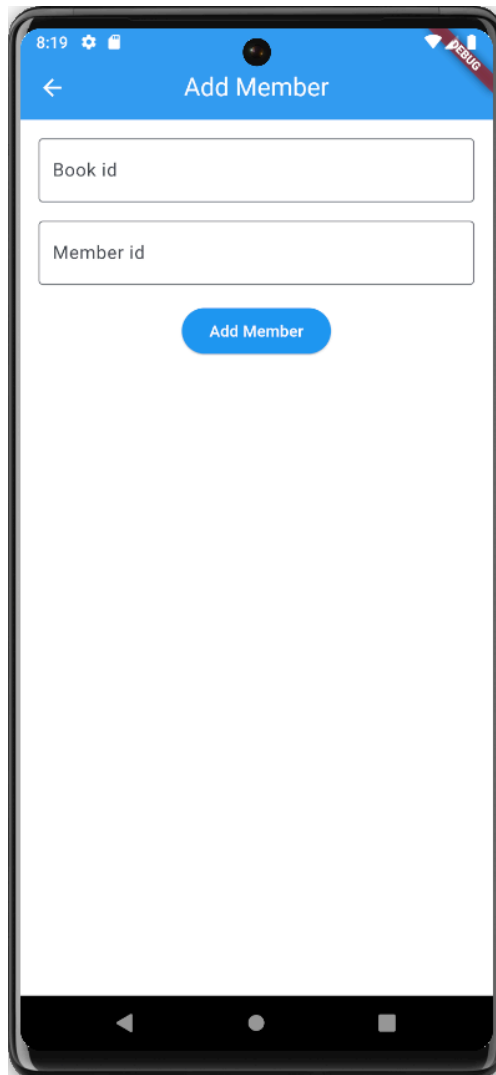
- **Chi tiết cài đặt:**

- + Các widget sử dụng: Stack, SingleChildScrollView, Form, Column, TextFormField, SizedBox, DropdownButtonFormField, TextEditingController, ElevatedButton, Container, CircularProgressIndicator, Text, Row, FittedBox, Image.network, Image.file, Expanded, SizedBox, TextButton.icon, Icon, AlertDialog, ActionButton, Navigator,...

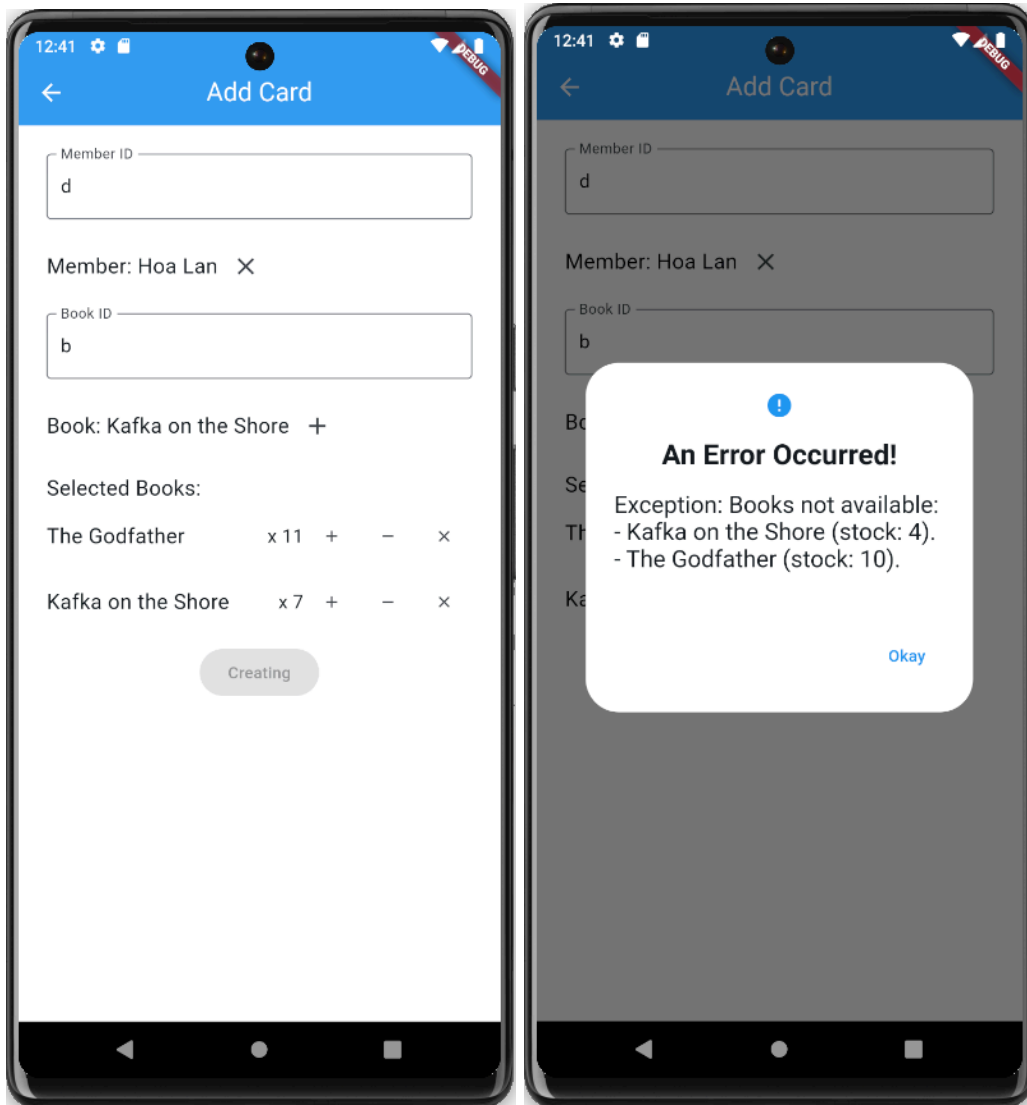
- + Thư viện:
  - + `image_picker`: Dùng để chọn ảnh từ thư viện hoặc máy ảnh.
  - + `intl`: Cung cấp khả năng định dạng ngày giờ, tiền tệ, và các yếu tố ngôn ngữ khác.
  - + `Provider`: Dùng để quản lý trạng thái ứng dụng, cung cấp cơ chế quản lý trạng thái giữa các widget trong ứng dụng. Trong code này, nó được sử dụng để truy cập vào `MembersManager`, giúp quản lý và thêm mới thành viên vào danh sách quản lý.
  - + `http`: thư viện được sử dụng để thực hiện các yêu cầu HTTP (GET, POST, PUT, DELETE...) trong ứng dụng Flutter. Nó giúp giao tiếp với các API web hoặc máy chủ từ xa, cho phép bạn gửi và nhận dữ liệu dưới dạng JSON hoặc các định dạng khác.
  - + `PocketBase`: là một backend-as-a-service (BaaS) đơn giản nhưng mạnh mẽ, cung cấp các dịch vụ như cơ sở dữ liệu, xác thực người dùng, và lưu trữ tệp tin. Thư viện `pocketbase` trong Flutter giúp kết nối và tương tác với `PocketBase` backend.
- + Quản lý chia sẻ: sử dụng `Provider` để quản lý trạng thái chia sẻ. `Provider` được sử dụng trong `MemberAddForm` để truy cập vào `MembersManager` thông qua `context.read<MembersManager>()`. Điều này cho phép bạn truy cập vào các phương thức và dữ liệu của `MembersManager` để thực hiện các thao tác như thêm thành viên mới vào hệ thống.
- + Sử dụng dịch vụ lưu trữ `Pocketbase`. Khi thêm thành viên sẽ thực hiện gọi api từ `pocketbase` để thêm mới thành viên.

## 5. Giao diện 5: Thêm phiếu mượn

- **Miêu tả giao diện:** hiển thị form thêm mới phiếu mượn.
- **Ảnh giao diện:**







Lập phiếu mượn với số lượng > số lượng tồn -> thất bại



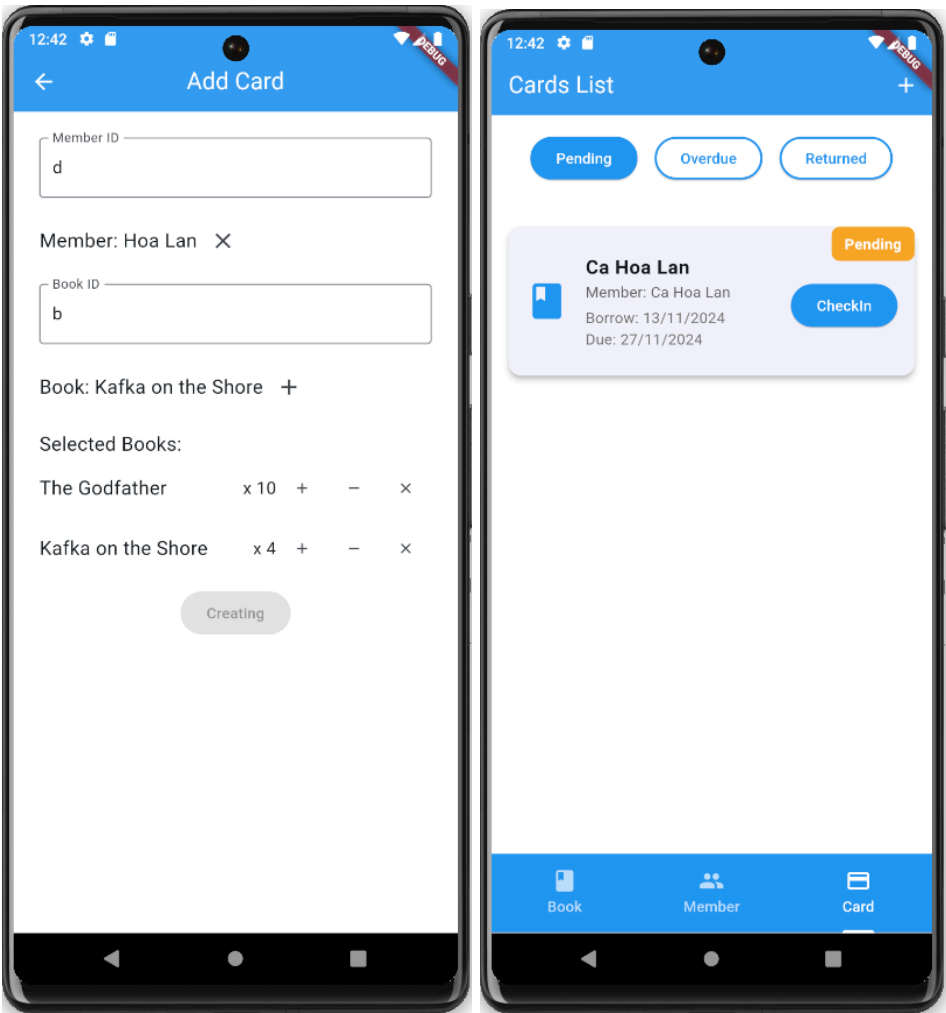
Trước khi lập phiếu mượn

Collections / books ⚙️ 🔄 </> API Preview + New record

🔍 Search term or filter like created > "2022-01-01"...

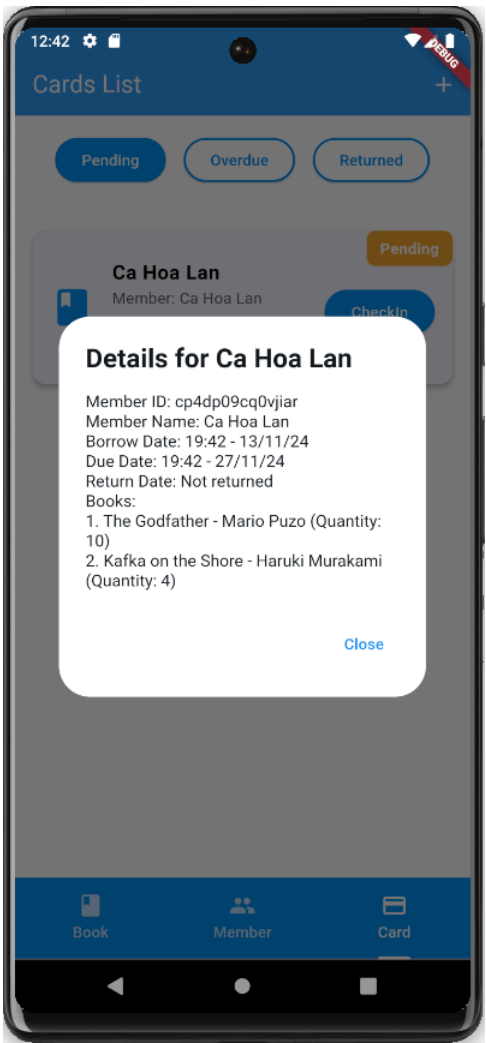
<input type="checkbox"/>	id	T title	T author	T genre	🖼️ imageUrl	# deleted	👤 userid	# quantity	📅 published	📅 created ↓	📅 updated	...
<input type="checkbox"/>	khqeyvpx3ma98j	The Godfather	Mario Puzo	Fiction		0	truong1212@gmail.com	10	1969-03-08 00:00:00 UTC	2024-11-08 13:43:21 UTC	2024-11-13 04:58:55 UTC	→
<input type="checkbox"/>	q1yk3nsu0sb8wu	Morisaki Bookshop's Dreams	Yagisawa Satoshi	Slice of Life		0	truong1212@gmail.com	2	2010-11-08 00:00:00 UTC	2024-11-08 13:41:43 UTC	2024-11-13 05:01:17 UTC	→
<input type="checkbox"/>	bv983gvc02iz912	Kafka on the Shore	Haruki Murakami	Magical Realism		0	truong1212@gmail.com	4	2002-01-02 00:00:00 UTC	2024-11-08 13:40:25 UTC	2024-11-13 05:01:15 UTC	→
<input type="checkbox"/>	su2q8lqu45wq5xm	Norwegian Wood	Haruki Murakami	Romance		0	truong1212@gmail.com	5	1987-02-02 00:00:00 UTC	2024-11-08 13:38:54 UTC	2024-11-13 04:49:52 UTC	→

Lập phiếu mượn với số lượng < hoặc = số lượng tồn -> thành công









Xem chi tiết thông tin phiếu mượn



Số lượng sách đã bị trừ tương ứng với số sách đã mượn

<input type="checkbox"/>	id	T title	T author	T genre	imageUrl	# deleted	userid	# quantity	published	created ↓	updated	...
<input type="checkbox"/>	khqeyevpx3ma98j	The Godfather	Mario Puzo	Fiction		0	truong1212@gmail.com	0	1969-03-08 00:00:00 UTC	2024-11-08 13:43:21 UTC	2024-11-13 05:42:42 UTC	→
<input type="checkbox"/>	q1yk3nsu0sb8wu	Morisaki Bookshop's Dreams	Yagisawa Satoshi	Slice of Life		0	truong1212@gmail.com	2	2010-11-08 00:00:00 UTC	2024-11-08 13:41:43 UTC	2024-11-13 05:01:17 UTC	→
<input type="checkbox"/>	bv983gvc02iz912	Kafka on the Shore	Haruki Murakami	Magical Realism		0	truong1212@gmail.com	0	2002-01-02 00:00:00 UTC	2024-11-08 13:40:25 UTC	2024-11-13 05:42:41 UTC	→
<input type="checkbox"/>	su2q8lqu45wq5xm	Norwegian Wood	Haruki Murakami	Romance		0	truong1212@gmail.com	5	1987-02-02 00:00:00 UTC	2024-11-08 13:38:54 UTC	2024-11-13 04:49:52 UTC	→

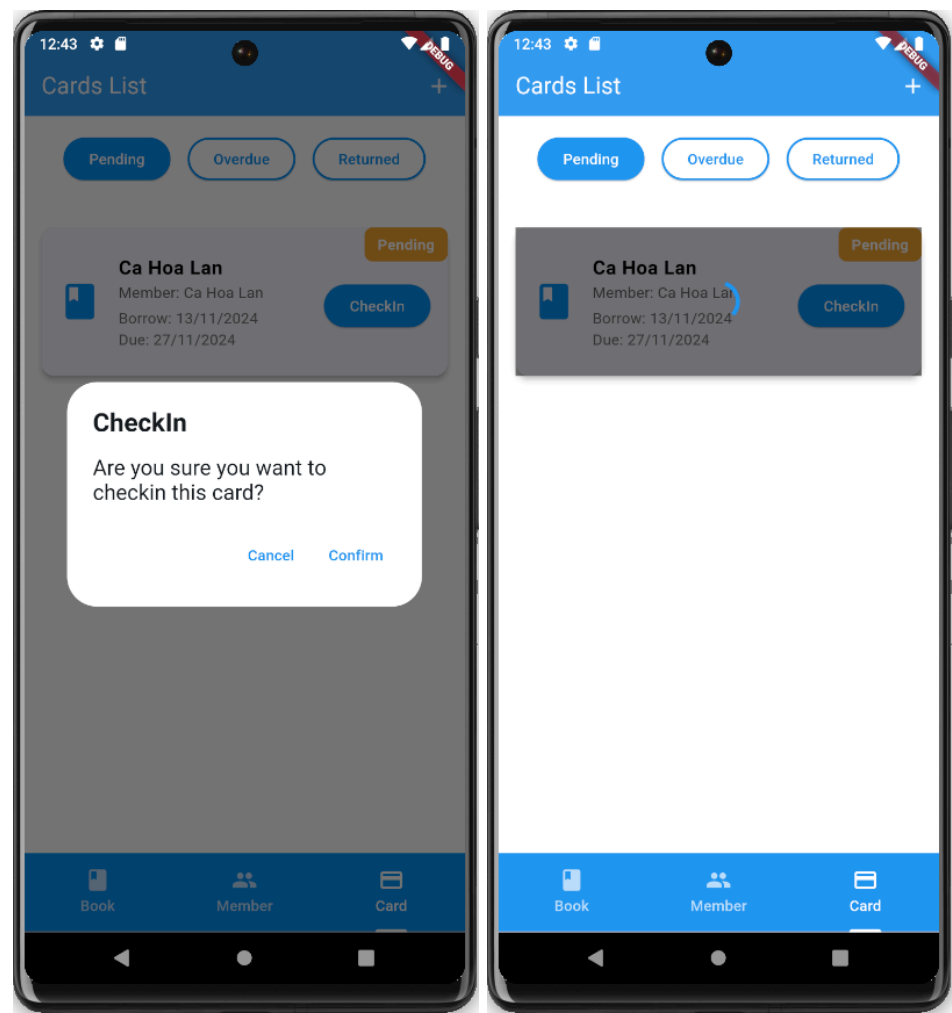
- Chi tiết cài đặt:

- + Widget sử dụng: Scaffold, TextFormField, SingleChildScrollView, Form, Row, ElevatedButton, Tooltip, AppBar, Text, Padding, IconButton, Row, Column, Tooltip, ElevatedButton, SingleChildScrollView, Flexible, Icon, IconButton, CircularProgressIndicator, SnackBar,...
- + Thư viện sử dụng:
  - + Provider: Dùng để quản lý trạng thái ứng dụng, cung cấp cơ chế quản lý trạng thái giữa các widget trong ứng dụng
  - + http: thư viện được sử dụng để thực hiện các yêu cầu HTTP (GET, POST, PUT, DELETE...) trong ứng dụng Flutter. Nó giúp giao tiếp với các API web hoặc máy chủ từ xa, cho phép bạn gửi và nhận dữ liệu dưới dạng JSON hoặc các định dạng khác.
  - + PocketBase: là một backend-as-a-service (BaaS) đơn giản nhưng mạnh mẽ, cung cấp các dịch vụ như cơ sở dữ liệu, xác thực người dùng, và lưu trữ tệp tin. Thư viện pocketbase trong Flutter giúp kết nối và tương tác với PocketBase backend.
- + Quản lý chia sẻ: Thư viện quản lý trạng thái, được sử dụng để truy cập vào các đối tượng quản lý như BooksManager, MembersManager, và CardsManager thông qua context.read<T>(). Cung cấp cơ chế quản lý trạng thái giữa các widget.
- + Sử dụng dịch vụ lưu trữ Pocketbase. Khi lập phiếu mượn sẽ thực hiện gọi api từ pocketbase để lưu lại phiếu mượn.

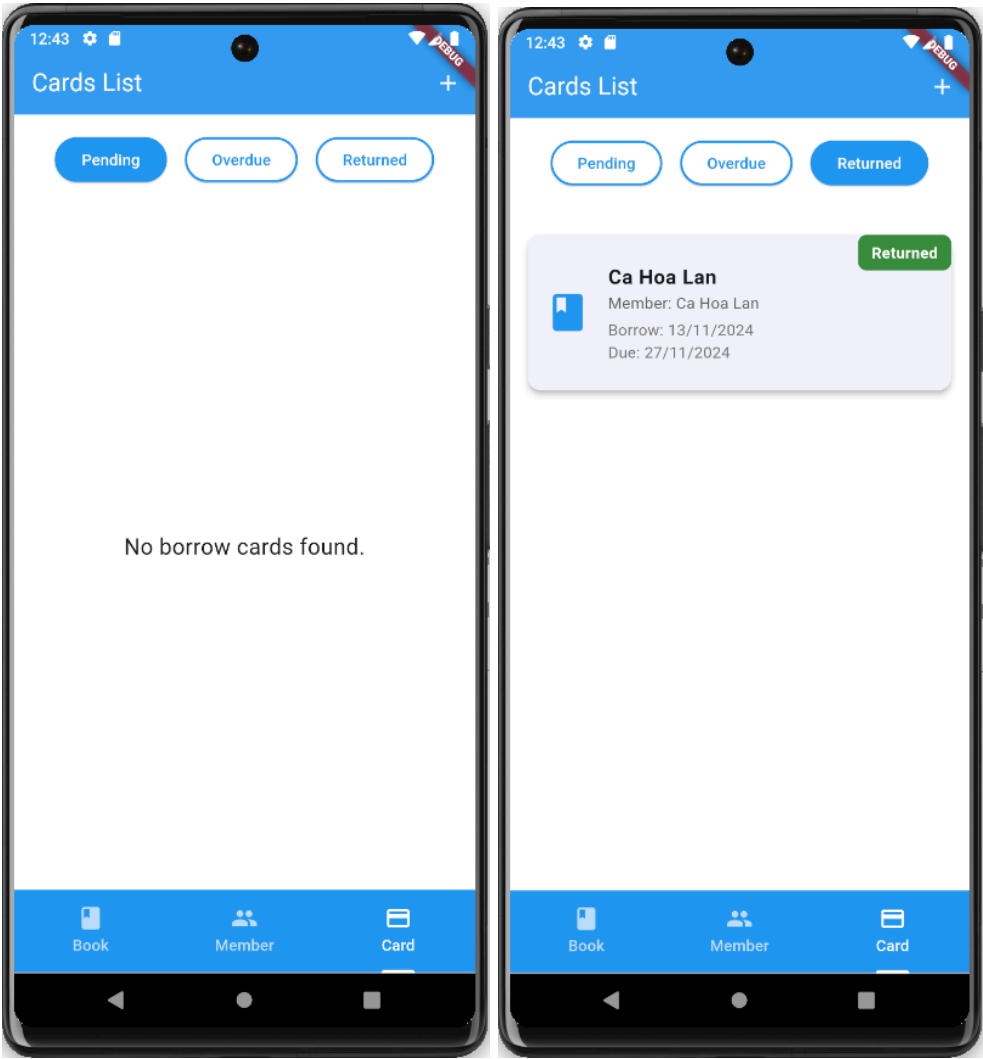
## 6. Giao diện 6: Trả sách

- **Miêu tả giao diện:** hiển thị hộp thoại xác nhận sách trả về. Ấn xác nhận sẽ thực hiện thay đổi trạng thái trả sách vào csdl.
- **Ảnh giao diện:**



Thực hiện xác nhận trả sách.





Trả sách thành công




















Số lượng sách được cập nhật lại

Collections / books  

 API Preview  New record

Q Search term or filter like created > "2022-01-01"...

<input type="checkbox"/>	id	T title	T author	T genre	 imageUrl	# deleted	 userid	# quantity	 published	 created ↓	 updated	...
<input type="checkbox"/>	 khqeyevpx3ma98j	The Godfather	Mario Puzo	Fiction		0	 truong1212@gmail.com	10	1969-03-08 00:00:00 UTC	2024-11-08 13:43:21 UTC	2024-11-13 05:43:11 UTC	→
<input type="checkbox"/>	 q1yk3nsu0sb8wu	Morisaki Bookshop's Dreams	Yagisawa Satoshi	Slice of Life		0	 truong1212@gmail.com	2	2010-11-08 00:00:00 UTC	2024-11-08 13:41:43 UTC	2024-11-13 05:01:17 UTC	→
<input type="checkbox"/>	 bv983gvc02iz912	Kafka on the Shore	Haruki Murakami	Magical Realism		0	 truong1212@gmail.com	4	2002-01-02 00:00:00 UTC	2024-11-08 13:40:25 UTC	2024-11-13 05:43:10 UTC	→
<input type="checkbox"/>	 su2q8lqu45wq5xm	Norwegian Wood	Haruki Murakami	Romance		0	 truong1212@gmail.com	5	1987-02-02 00:00:00 UTC	2024-11-08 13:38:54 UTC	2024-11-13 04:49:52 UTC	→

- **Chi tiết cài đặt:**

- Widget sử dụng :StatefulWidget, FutureBuilder, Text, Consumer, CircularProgressIndicator, ListTile, Icon, Column, SizedBox, Stack, Positioned, ElevatedButton, AlertDialog, TextButton,...
- Thư viện sử dụng:
  - provider: Thư viện này cung cấp cơ chế quản lý trạng thái giữa các widget. Trong đoạn mã này, CardsManager được cung cấp thông qua context.read<CardsManager>() và Consumer<CardsManager>() giúp theo dõi các thay đổi trong trạng thái của CardsManager. Khi CardsManager thay đổi, CardsList sẽ được cập nhật tự động mà không cần phải gọi lại setState().
  - FutureBuilder: Mặc dù không phải là một thư viện riêng, FutureBuilder là một widget của Flutter để xử lý dữ liệu bất đồng bộ (asynchronous). Trong đoạn mã này, FutureBuilder được sử dụng để thực hiện công việc tải danh sách thẻ từ CardsManager và hiển thị các trạng thái của quá trình tải (đang tải, lỗi, thành công).
  - http: thư viện được sử dụng để thực hiện các yêu cầu HTTP (GET, POST, PUT, DELETE...) trong ứng dụng Flutter. Nó giúp giao tiếp với các API web hoặc máy chủ từ xa, cho phép bạn gửi và nhận dữ liệu dưới dạng JSON hoặc các định dạng khác.
  - PocketBase: là một backend-as-a-service (BaaS) đơn giản nhưng mạnh mẽ, cung cấp các dịch vụ như cơ sở dữ liệu, xác thực người dùng, và lưu trữ tệp tin. Thư viện pocketbase trong Flutter giúp kết nối và tương tác với PocketBase backend.

- Quản lý chia sẻ: sử dụng Provider để quản lý trạng thái của danh sách thẻ (CardsManager) và cập nhật giao diện khi có sự thay đổi trong dữ liệu thẻ.
- Sử dụng dịch vụ lưu trữ Pocketbase. Khi cập nhật trạng thái trả của phiếu mượn sẽ thực hiện gọi api từ pocketbase để lưu lại thay đổi của phiếu mượn.