

# **BÁO CÁO DỰ ÁN CUỐI KỲ**

**HỌC KỲ 2, NĂM HỌC 2023-2024**

**CT484: PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG**

– **Tên dự án/ứng dụng: Ứng dụng bán sách**

– **Link GitHub mã nguồn:**

**<https://github.com/23-24Sem2-Courses/ct48402-project-huynhhao1111.git>**

– **MSSV 1: B2005702**

– **Họ tên SV 1: Lê Phan Thảo Anh**

– **MSSV 2: B2014654**

– **Họ tên SV 2: Huỳnh Trần Hảo**

– **Lớp học phần: CT449-02**

## MỤC LỤC

I. Tổng quan .....	3
II. Chi tiết các chức năng .....	4
1. Giao diện trang đăng nhập .....	4
2. Giao diện đăng ký .....	6
3. Giao diện trang Home .....	8
4. Giao diện xem chi tiết sách .....	10
* Chức năng thêm số lượng sách vào giỏ hàng .....	11
5. Giao diện giỏ hàng .....	13
6. Giao diện đơn hàng .....	15
7. Giao diện trang quản lý sách .....	17
8. Giao diện trang thêm sách .....	19
9. Giao diện trang sửa sách .....	21
*Chức năng xóa sách .....	22
10. Giao diện trang sách đã yêu thích .....	24
11. Giao diện tìm kiếm sách .....	25
12. Giao diện tài khoản - Đăng xuất .....	27

## I. Tổng quan

– **Miêu tả dự án/ứng dụng:** Ứng dụng sẽ cho phép người dùng đăng ký và đăng nhập tài khoản. Sau khi đăng nhập có thể xem và đăng tải sách, có thể quản lý sách bản thân đã thêm như sửa, xóa và thêm sách mới. Cho phép xem danh sách sách yêu thích của bản thân. Ngoài ra người dùng khi xem chi tiết sách có thể thêm vào giỏ hàng. Xem giỏ hàng, xóa các sách đã thêm khỏi giỏ hàng, đặt hàng và xem đơn hàng của bản thân khi đã mua sách. Tìm kiếm sách một cách nhanh chóng ở trang tìm kiếm.

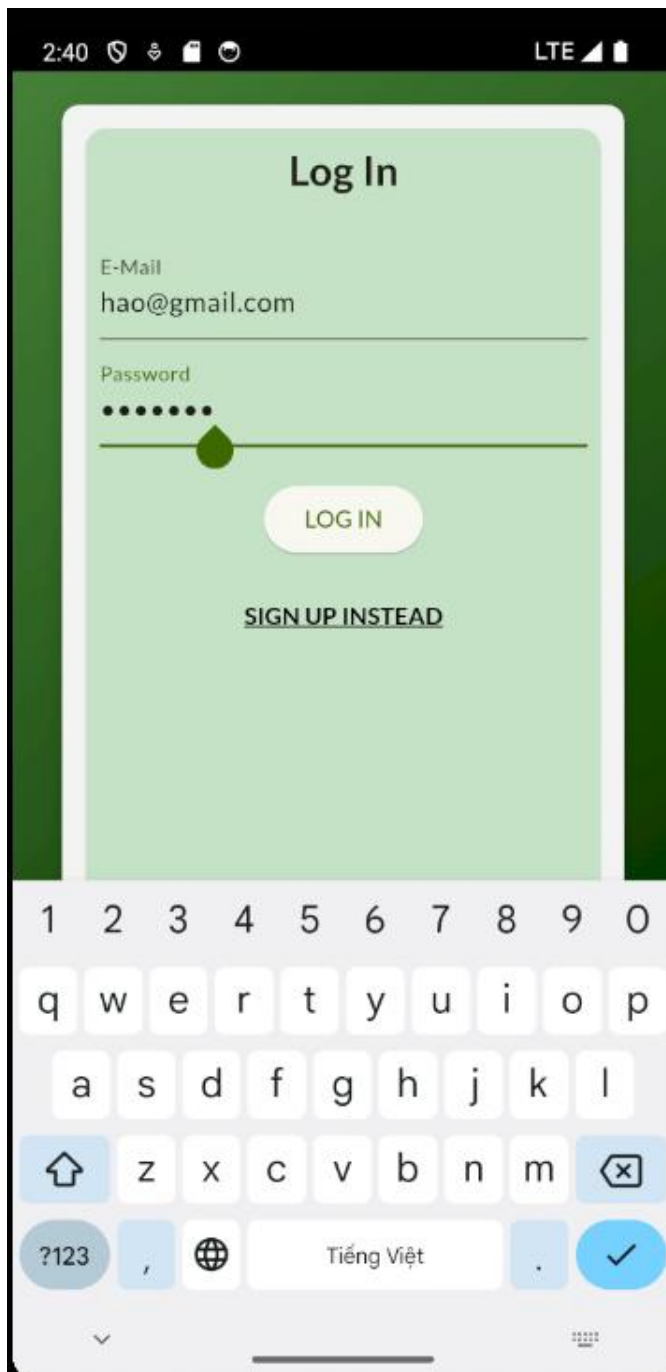
### – Bảng phân công công việc nếu làm nhóm.

Tên thành viên	Công việc	Đóng góp
Lê Phan Thảo Anh	<ul style="list-style-type: none"><li>- Thiết kế giao diện</li><li>- Làm chức năng thêm, sửa, xóa sách</li><li>- Xem chi tiết sách</li><li>- Làm đăng nhập, đăng ký</li><li>- Làm chức năng tìm kiếm sách</li></ul>	100%
Huỳnh Trần Hào	<ul style="list-style-type: none"><li>- Thiết kế giao diện</li><li>- Làm chức năng thêm giỏ hàng và các chức năng ở trang giỏ hàng</li><li>- Làm chức năng trang đơn hàng</li><li>- Thêm sách vào yêu thích</li><li>- Làm đăng xuất giao diện tài khoản</li></ul>	100%

## II. Chi tiết các chức năng

### 1. Giao diện trang đăng nhập

- **Miêu tả giao diện:** Cho phép người dùng đăng nhập vào tài khoản thông qua email và mật khẩu
- **Hình ảnh giao diện:**



**Widget sử dụng cho chức năng/giao diện này:**

- + Scaffold, Stack, Container, SingleChildScrollView, SizedBox, Column, Flexible, Text,

**Các thư viện/plugin sử dụng:**

- Provider: Được sử dụng để quản lý trạng thái ứng dụng và cung cấp dữ liệu cho AuthManager.
- http\_exception.dart: Một tập hợp các lớp để xử lý các ngoại lệ HTTP.
- dialog\_utils.dart: Chứa hàm showErrorDialog để hiển thị thông báo lỗi.
- shared\_preferences: cung cấp cách đơn giản để lưu trữ và truy xuất dữ liệu cục bộ trên thiết bị của người dùng.

**Kiến trúc code và quản lý trạng thái:**

Trạng thái được quản lý thông qua Provider. AuthManager được sử dụng để xử lý các yêu cầu đăng nhập thông qua hàm login.

- + Khi bấm LOGIN sẽ vào trang home cùng với lưu trữ token của người đăng nhập.

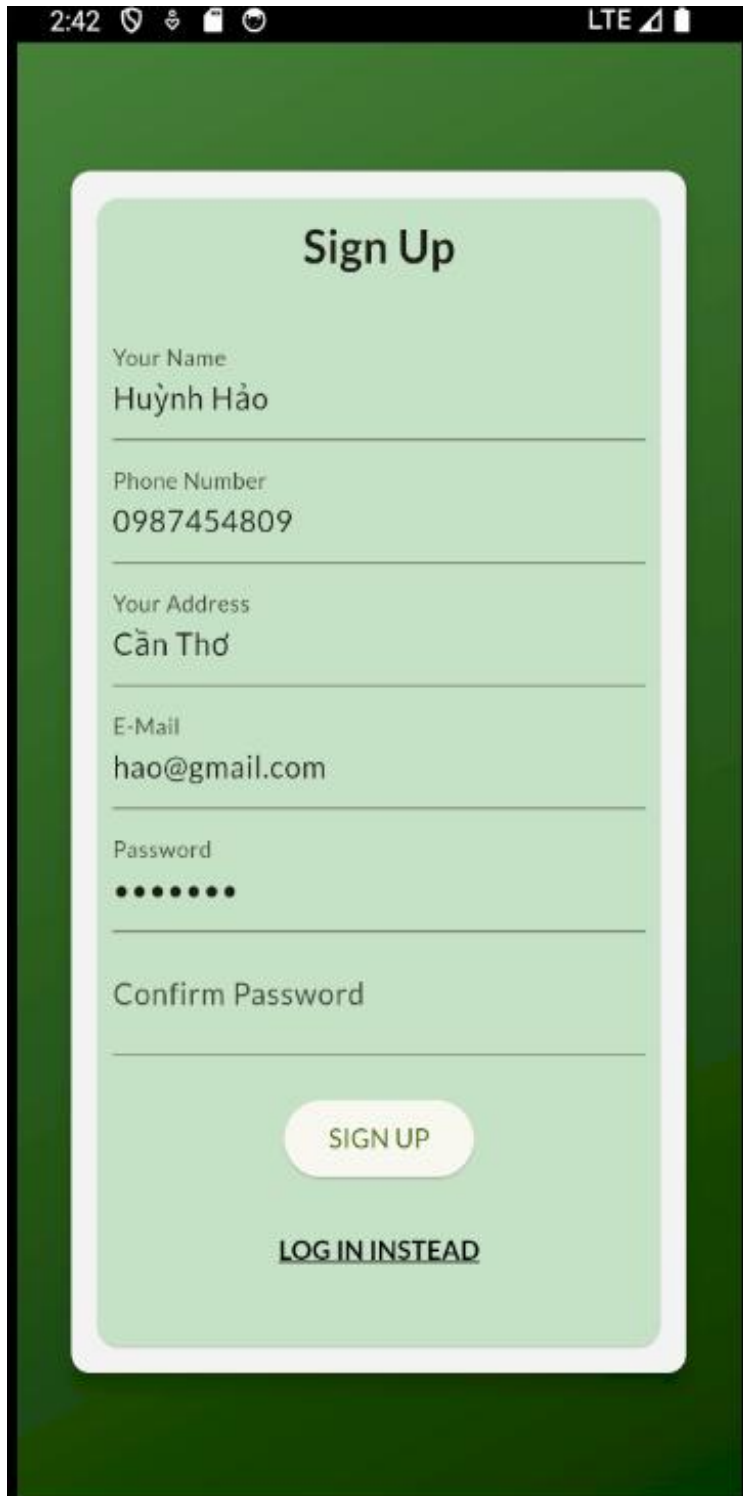
**Đọc và lưu trữ dữ liệu:**

Chức năng này thực hiện đăng nhập người dùng sử dụng dịch vụ Firebase Authentication. API của Firebase Authentication được sử dụng để thực hiện các hành động đăng nhập

- Gọi thông qua hàm login trong file “auth\_service”
- Dữ liệu truy xuất: email, mật khẩu

## 2. Giao diện đăng ký

- **Miêu tả giao diện:** Cho phép người dùng đăng ký tài khoản
- **Hình ảnh giao diện:**



The image shows a mobile application interface for signing up. The screen has a dark green background. In the center, there is a white rounded rectangle containing a light green sign-up form. The form is titled "Sign Up" in bold black text. Below the title, there are five input fields with labels: "Your Name" (filled with "Huỳnh Hào"), "Phone Number" (filled with "0987454809"), "Your Address" (filled with "Cần Thơ"), "E-Mail" (filled with "hao@gmail.com"), and "Password" (filled with seven dots). Below the password field is a "Confirm Password" field. At the bottom of the form, there is a white rounded button labeled "SIGN UP" and a link labeled "LOG IN INSTEAD" in bold black text. The top of the screen shows a status bar with the time "2:42", signal strength, and battery level.

2:42 LTE

### Sign Up

Your Name  
Huỳnh Hào

Phone Number  
0987454809

Your Address  
Cần Thơ

E-Mail  
hao@gmail.com

Password  
•••••••

Confirm Password

SIGN UP

**LOG IN INSTEAD**

**Widget sử dụng cho chức năng/giao diện này:**

- + Scaffold, Stack, Container, SingleChildScrollView, SizedBox, Column, Flexible, Text,

**Các thư viện/plugin sử dụng:**

- Provider: Được sử dụng để quản lý trạng thái ứng dụng và cung cấp dữ liệu cho AuthManager.
- shared\_preferences: cung cấp cách đơn giản để lưu trữ và truy xuất dữ liệu cục bộ trên thiết bị của người dùng.
- http\_exception.dart: Một tập hợp các lớp để xử lý các ngoại lệ HTTP.
- dialog\_utils.dart: Chứa hàm showErrorDialog để hiển thị thông báo lỗi.

**Kiến trúc code và quản lý trạng thái:**

Trạng thái được quản lý thông qua Provider. AuthManager được sử dụng để xử lý các yêu cầu đăng nhập thông qua hàm login.

- + Khi bấm SIGNUP sẽ vào trang home cùng với lưu trữ token của người đã đăng ký.

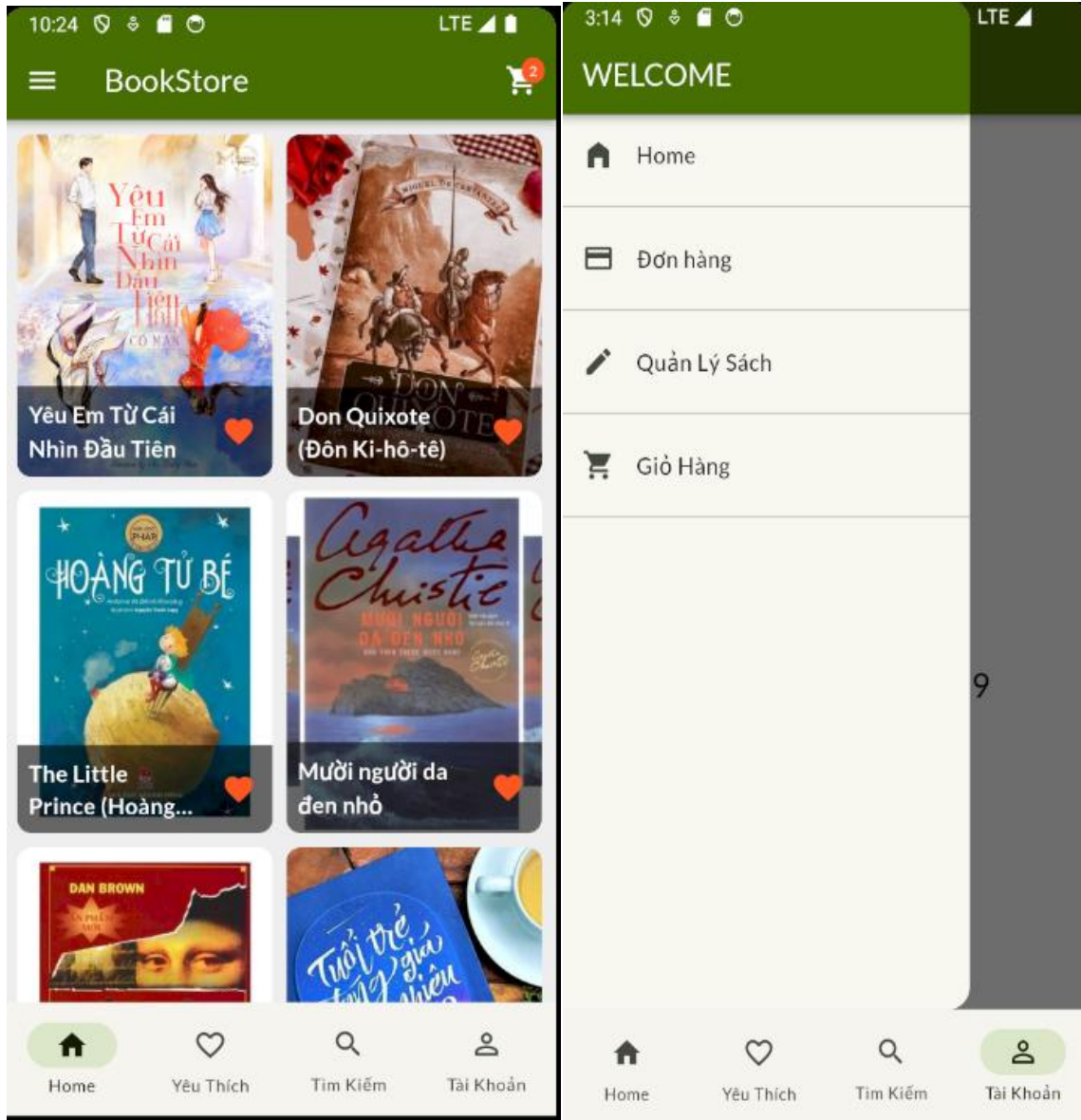
**Đọc và lưu trữ dữ liệu:**

Chức năng này thực hiện đăng ký người dùng sử dụng dịch vụ Firebase Authentication. API của Firebase Authentication được sử dụng để thực hiện các hành động đăng ký. Thông tin lưu dưới dạng Json.

- Gọi thông qua hàm signup trong file “auth\_service”
- Dữ liệu lưu trữ: email, name, address, phone, password

### 3. Giao diện trang Home

- **Miêu tả giao diện:** Trang sẽ hiển thị tất cả các sách ra màn hình. Phía dưới sẽ điều hướng các trang xem được yêu thích, tìm sách và tài khoản. Bên tay trái là Drawer hiển thị các lựa chọn để xem giỏ hàng, đơn hàng, quản lý sách.
- **Hình ảnh giao diện:**





### **Các widget được sử dụng:**

Trong BooksOverviewScreen:

+ Scaffold, AppBar, PopupMenuButton, Consumer, FutureBuilder, ValueListenableBuilder, Center, CircularProgressIndicator, MaterialApp, Text, Icon, Center, PopupMenuButton

Trong AppDrawer:

+ Drawer, Column, Text, Icon, MaterialPageRoute, AppBar, ListTile:

### **Các thư viện/plugin sử dụng:**

- Provider: Được sử dụng để quản lý trạng thái ứng dụng và cung cấp dữ liệu cho.
- flutter\_dotenv: Được sử dụng để đọc biến môi trường từ tệp .env

### **Trạng thái được quản lý thông qua Provider.**

Chức năng này sử dụng thư viện Provider để quản lý trạng thái chia sẻ.

- Khi bấm vào hình trái tim sách sẽ được thêm vào yêu thích, làm thay đổi giao diện trang yêu thích.
- Bấm icon cart sẽ chuyển sang trang giỏ hàng

### **Đọc và lưu trữ dữ liệu:**

Chức năng này lấy dữ liệu từ firebase. Thông tin lưu ở realtime database dưới dạng Json.

- Gọi thông qua hàm fetchBooks trong file “books\_service”
- Nhánh truy xuất: books
- Dữ liệu truy xuất: creatorId, title, author, category, description, imageUrl, price, quantity.

#### 4. Giao diện xem chi tiết sách

- **Miêu tả giao diện:** Xem chi tiết thông tin một quyển sách như tên sách, tác giả, thể loại, số lượng tồn kho, giá tiền và giới thiệu sách.
- **Hình ảnh giao diện:**



#### Các widget được sử dụng trong chức năng/giao diện này:

Scaffold, AppBar, AppDrawer, Consumer, Providers, ValueListenableBuilder, ElevatedButton, IconButton, Icon, Text, Padding, Center, Image.

#### Các thư viện/plugin được sử dụng:

provider: Được sử dụng để quản lý trạng thái chia sẻ giữa các widget, cung cấp dữ liệu từ BooksManager và CartManager đến các widget con.

flutter\_dotenv: Được sử dụng để đọc biến môi trường từ tệp .env

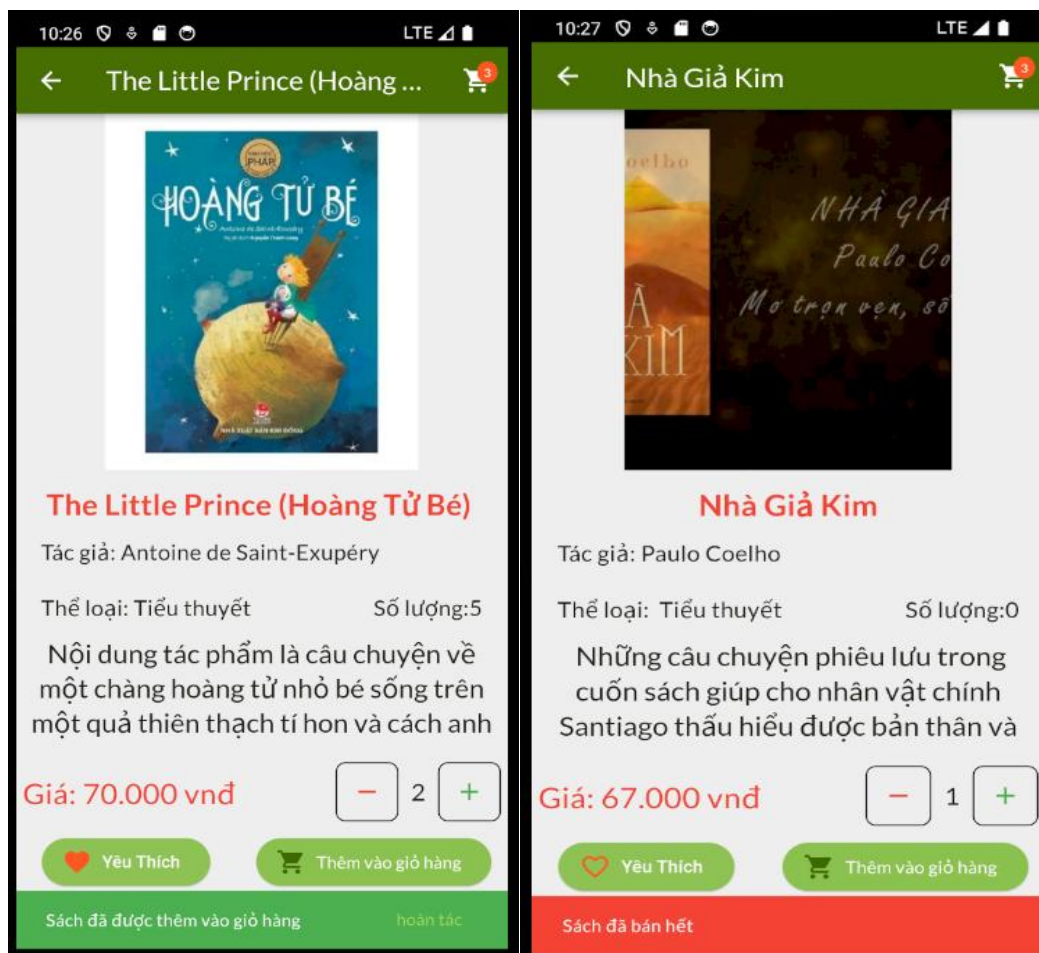
#### Đọc và lưu trữ dữ liệu:

Chức năng này lấy dữ liệu từ firebase. Thông tin sách đã được lưu realtime database dưới dạng Json.

- Đọc dữ liệu cục bộ từ nhánh books
- Nhánh truy xuất: books
- Dữ liệu truy xuất: creatorId, title, author, category, description, imageUrl, price, quantity.

#### \* Chức năng thêm số lượng sách vào giỏ hàng

- **Miêu tả giao diện:** Cho phép người dùng thêm sách vào giỏ hàng cùng với số lượng đã chọn. Nếu chọn quá số lượng hoặc số lượng tồn kho bằng 0 sẽ hiển thị các thông báo.
- **Hình ảnh giao diện:**



**Trạng thái được quản lý:**

Chức năng này sử dụng giải pháp quản lý trạng thái chia sẻ thông qua Provider Pattern:

- Khi bấm vào yêu thích sẽ làm thay đổi giao diện trang yêu thích.
- Khi bấm vào số lượng và thêm vào giỏ hàng truyền dữ liệu vào giỏ hàng, thay đổi con số trên icon cart.

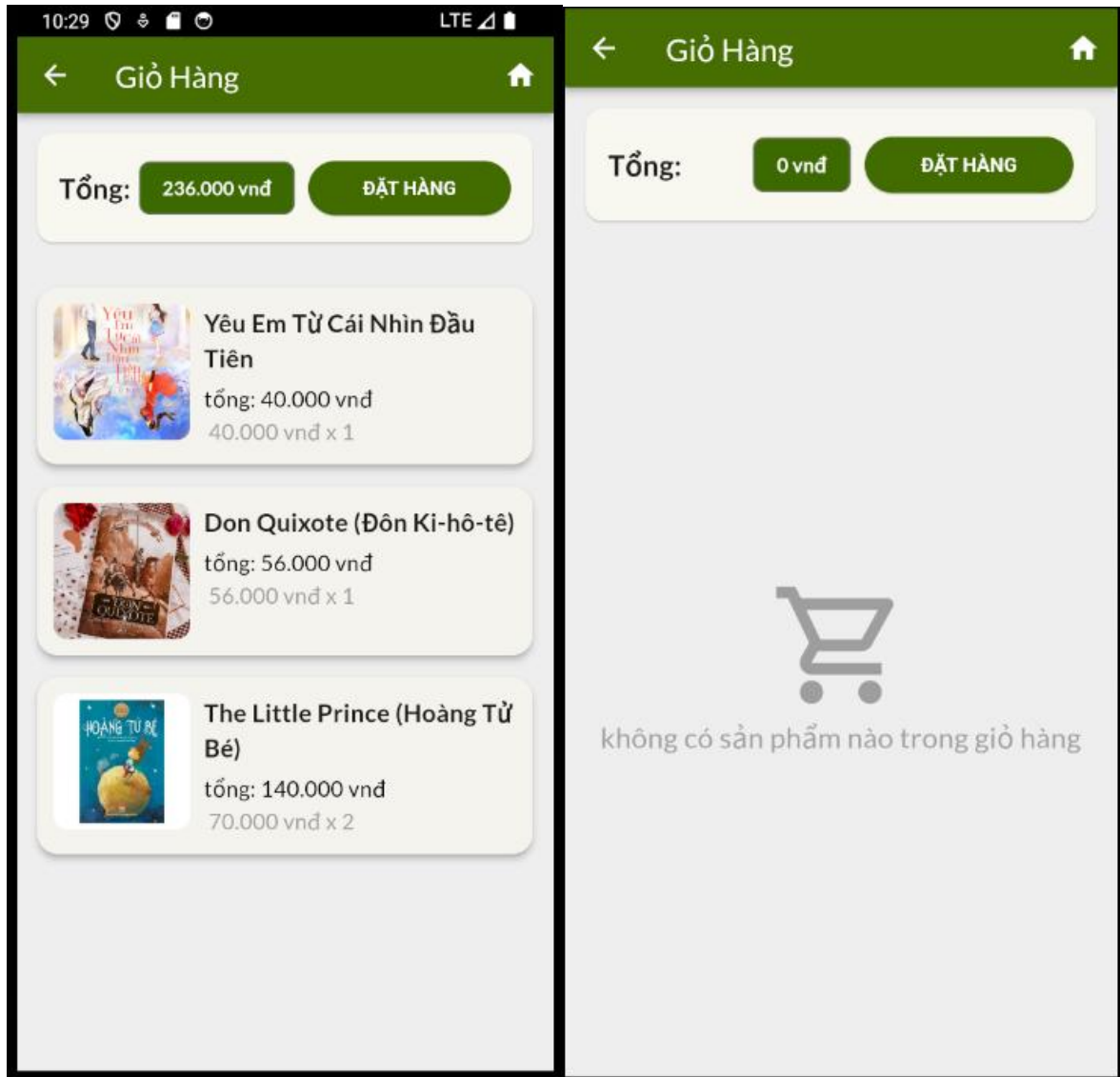
**Đọc và lưu trữ dữ liệu:**

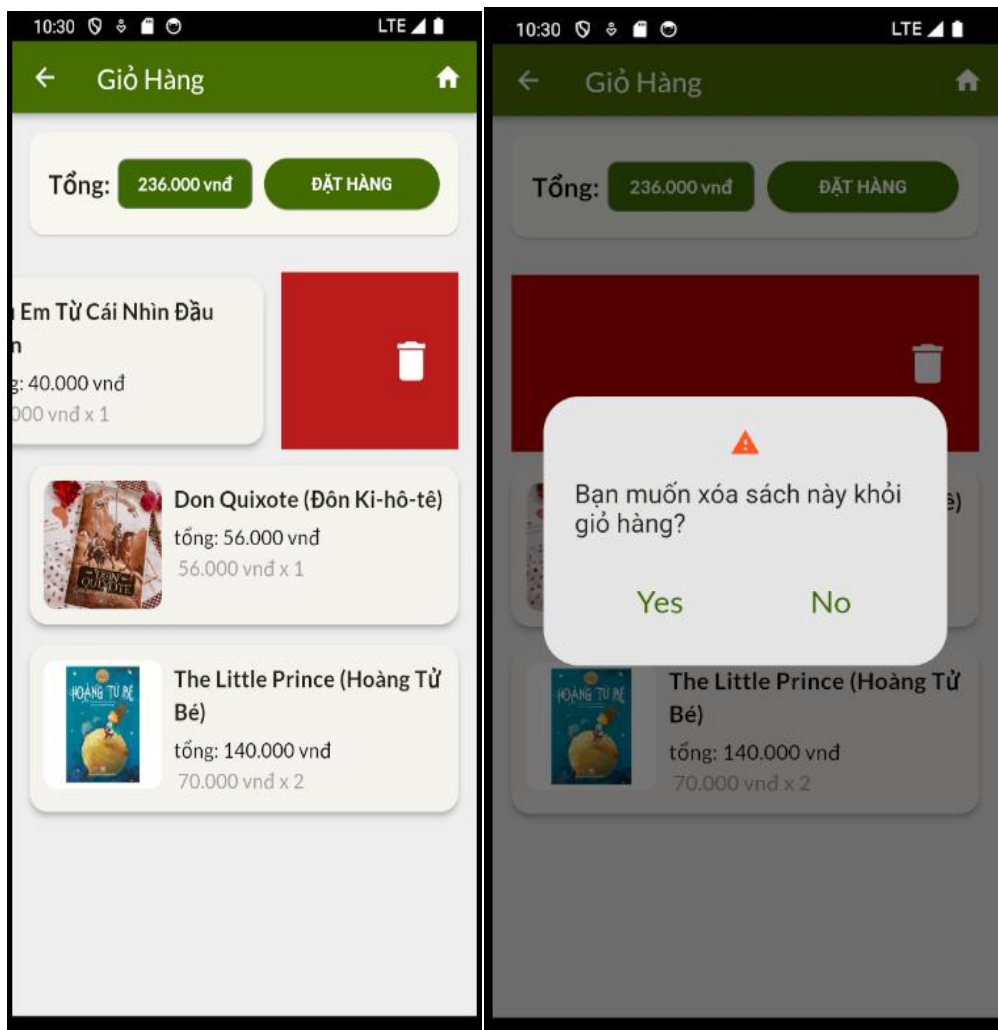
Chức năng này lấy dữ liệu từ firebase. Thông tin sách đã được lưu realtime database dưới dạng Json.

- Gọi các hàm “saveFavoriteStatus” từ file “books\_service” để thêm vào yêu thích. Gọi hàm “addItem” từ file “carts\_service” để thêm sách vào giỏ hàng
- Nhánh truy xuất: books, userFavorites, carts
- Dữ liệu truy xuất:
  - + books: {creatorId, title, author, category, description, imageUrl, price, quantity}
  - + userFavorites
  - + carts : {imageUrl, price, quantity, title}

## 5. Giao diện giỏ hàng

- **Miêu tả giao diện:** Cho phép người dùng xem những quyển sách đã được bỏ vào giỏ hàng, có thể xóa bớt sách ra khỏi giỏ hàng nếu muốn.
- **Hình ảnh giao diện:**





### Các widget được sử dụng trong chức năng/giao diện này:

+ Scaffold,AppBar,Column, Padding, EdgeInsets, Theme, Row, Card, Center,CircularProgressIndicator, ScaffoldMessenger,

### Các thư viện/plugin được sử dụng:

- provider: Được sử dụng để quản lý trạng thái chia sẻ giữa các widget và các phần của ứng dụng.
- flutter\_dotenv: Được sử dụng để đọc các biến môi trường từ tệp .env.

### Trạng thái được quản lý:

Chức năng này sử dụng giải pháp quản lý trạng thái chia sẻ bằng cách sử dụng Provider để cung cấp truy cập đến CartManager, BooksManager, và OrderManager cho các widget cần thiết.

- Khi bấm đặt hàng dữ liệu sẽ thêm vào orders.
- Lướt xóa sản phẩm sẽ làm thay đổi giao diện và dữ liệu trong carts bị xóa đi.

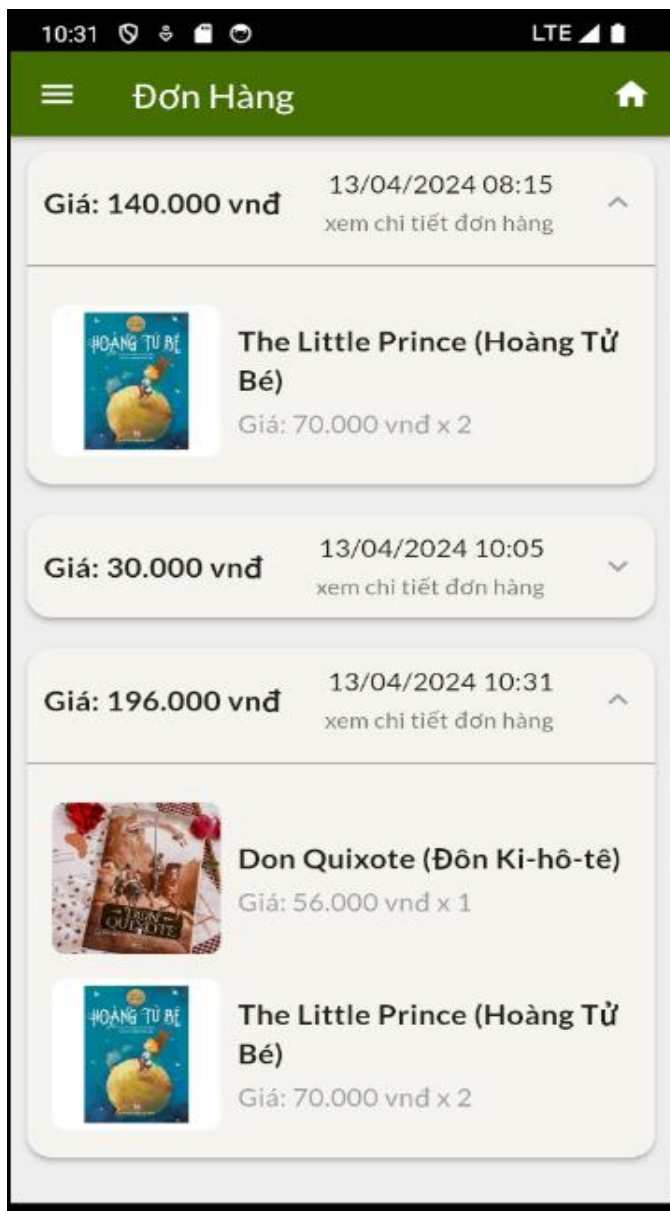
### **Đọc và lưu trữ dữ liệu:**

Chức năng này lấy dữ liệu từ firebase. Thông tin giỏ hàng đã được lưu trên realtime database lưu dưới dạng Json.

- Gọi thông qua hàm “getCart” để lấy thông tin carts và “addItem” để thêm carts, “removeItem” để xóa sản phẩm khỏi carts trong file “carts\_service” và “addOrders” để thêm đơn hàng trong file “orders\_service”.
- Nhánh lưu trữ: carts, order
- Dữ liệu lưu trữ:
  - + carts: {imageUrl, price, quantity, title}
  - + orders: {amount, dateTime, books[]}

## **6. Giao diện đơn hàng**

- **Miêu tả giao diện:** Cho phép người dùng xem các đơn hàng của bản thân gồm những gì, xem được hình ảnh, giá tiền và số lượng sách của đơn hàng đó.
- **Hình ảnh giao diện:**



**Các widget được sử dụng trong chức năng/giao diện này:**

+ Scaffold,AppBar,Column, Center, Text

**Các thư viện/plugin được sử dụng:**

provider: Được sử dụng để quản lý trạng thái chia sẻ giữa các widget và các phần của ứng dụng.

flutter\_dotenv: Được sử dụng để đọc các biến môi trường từ tệp .env

**Đọc và lưu trữ dữ liệu:**

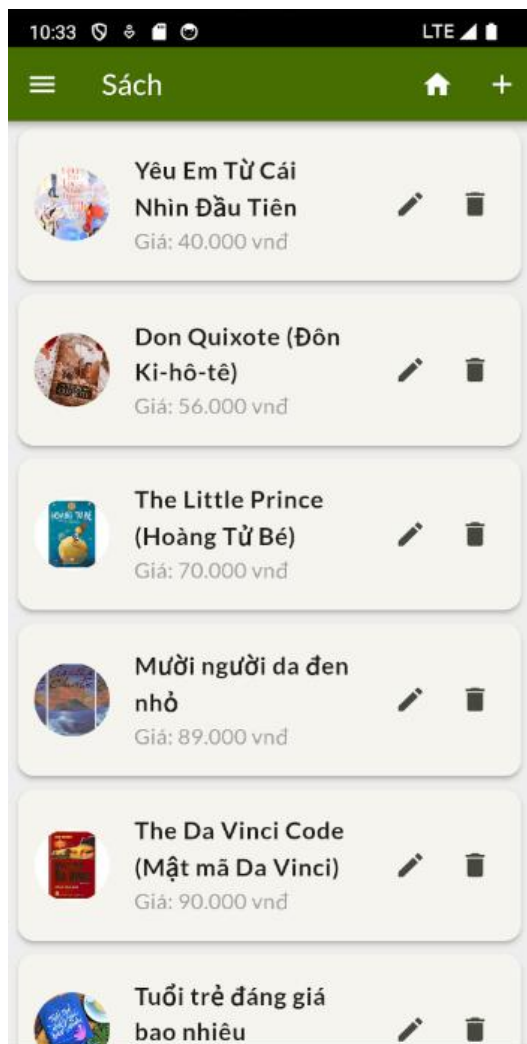


Chức năng này lấy và lưu dữ liệu từ firebase. Thông tin đơn hàng đã được lưu trên realtime database lưu dưới dạng Json.

- Gọi thông qua hàm “fetchOrders” để lấy thông tin đơn hàng trong file “orders\_service”.
- Nhánh lưu trữ: orders
- Dữ liệu lưu trữ: amount, dateTime, books[].

## 7. Giao diện trang quản lý sách

- **Miêu tả giao diện:** Trang hiển thị tất cả các sách người dùng đã thêm trên hệ thống, có thể chọn thêm, sửa, xóa
- **Hình ảnh giao diện:**



**Các widget được sử dụng trong chức năng/giao diện này:**

Scaffold, Text, Image, Icon, AppBar, AppDrawer, FutureBuilder, Consumer, ListView.builder, RefreshIndicator, ListTile, IconButton, CircleAvatar, Row, IconButton, CircleAvatar, ListTile, Card.

**Các thư viện/plugin được sử dụng:**

- provider: Được sử dụng để quản lý trạng thái chia sẻ giữa các widget.
- flutter\_dotenv: Được sử dụng để đọc biến môi trường từ tệp .env.
- http: Được sử dụng để thực hiện các yêu cầu HTTP để giao tiếp với backend.

**Trạng thái được quản lý:**

Chức năng này sử dụng giải pháp quản lý trạng thái chia sẻ thông qua Provider Pattern.

- Khi nhấn vào icon dấu cộng sẽ có thể chuyển sang trang Edit\_book\_screen để thêm sách
- Khi nhấn vào icon cây viết sẽ có thể chuyển sang trang Edit\_book\_screen để sửa sách
- Khi nhấn vào icon thùng rác sẽ có thể xóa sách.


**Đọc và lưu trữ dữ liệu:**

Chức năng này lấy dữ liệu từ firebase. Thông tin sách đã được lưu ở bảng books trên realtime database.

- Gọi thông qua hàm fetchBooks trong file “books\_service”
- Nhánh truy xuất: books
- Dữ liệu truy xuất: creatorId, title, author, category, description, imageUrl, price, quantity.

## 8. Giao diện trang thêm sách

- **Miêu tả giao diện:** Cho phép người dùng thêm sách với những thông tin sau. Nếu thêm thiếu thông tin, hoặc thông tin không đúng sẽ hiển thị các dòng nhắc nhở. Sau khi thêm thành công sẽ chuyển về trang quản lý sách.
- **Hình ảnh giao diện:**

Thêm Sách	
Tên Truyện	Thao Tung Tâm Lý
Giá	52000
Số lượng	4
mô tả	Thomas giới thiệu đến độc giả những hiểu biết liên quan đến thao túng tâm lý và lạm dụng tiềm ẩn.
Tác Giả	Shannon Thomas
Thể Loại	tiểu thuyết
Hình ảnh	
Nhập đường dẫn hình ảnh 2/ps/20240115_c49KkCdj7D.jpeg	



**Các widget được sử dụng trong chức năng/giao diện này:**

Scaffold, AppBar, IconButton, CircularProgressIndicator, Padding, Form, ListView, TextFormField, NavigationBar

**Các thư viện/plugin được sử dụng:**

provider: Được sử dụng để quản lý trạng thái chia sẻ giữa các widget.

flutter\_dotenv: Được sử dụng để đọc biến môi trường từ tệp .env.

http: Được sử dụng để thực hiện các yêu cầu HTTP để giao tiếp với backend.

**Trạng thái được quản lý:**

Chức năng này sử dụng giải pháp quản lý trạng thái chia sẻ thông qua Provider Pattern.

- Khi bấm icon save sẽ trở về trang quản lý và làm thay đổi nó khi đã thêm sách.

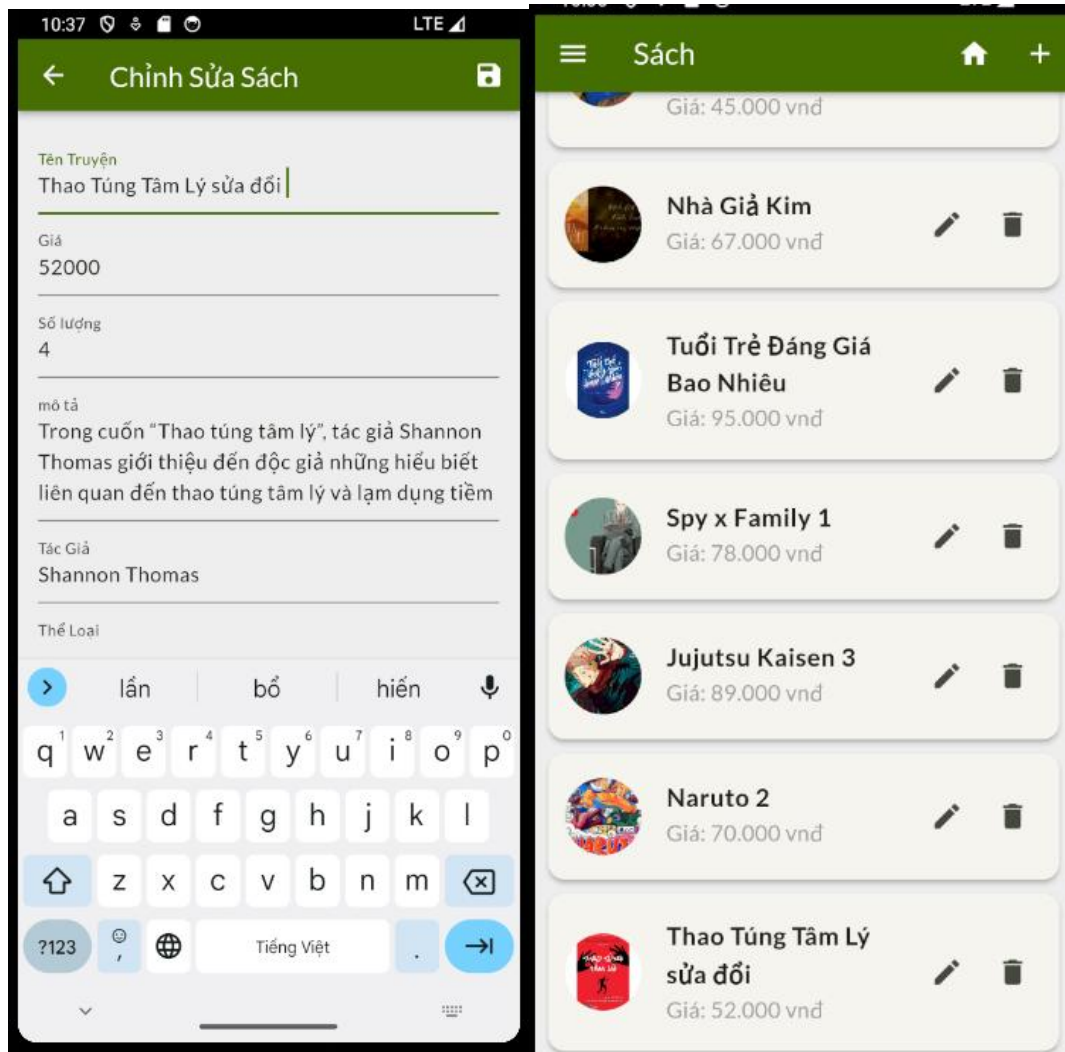
**Đọc và lưu trữ dữ liệu:**

Chức năng này lưu dữ liệu từ firebase. Thông tin sách sẽ được lưu trên realtime database lưu trữ dạng Json.

- Gọi thông qua hàm “addBook” trong file “books\_service”
- Nhánh lưu trữ: books
- Dữ liệu lưu trữ: title, author, category, description, imageUrl, price, quantity.

## 9. Giao diện trang sửa sách

- **Miêu tả giao diện:** Cho phép người dùng chỉnh sửa sách đã thêm. Sau khi sửa thành công sẽ chuyển về trang quản lý sách.
- **Hình ảnh giao diện:**



Các widget được sử dụng trong chức năng/giao diện này:

Scaffold, AppBar, IconButton, CircularProgressIndicator, Padding, Form, ListView, TextFormField, NavigationBar,

### **Các thư viện/plugin được sử dụng:**

provider: Được sử dụng để quản lý trạng thái chia sẻ giữa các widget.

flutter\_dotenv: Được sử dụng để đọc biến môi trường từ tệp .env.

http: Được sử dụng để thực hiện các yêu cầu HTTP để giao tiếp với backend.

### **Trạng thái được quản lý:**

Chức năng này sử dụng giải pháp quản lý trạng thái chia sẻ thông qua Provider Pattern.

- Khi bấm vào icon save thì sẽ chuyển sang và cập nhật lại giao diện trang quản lý.

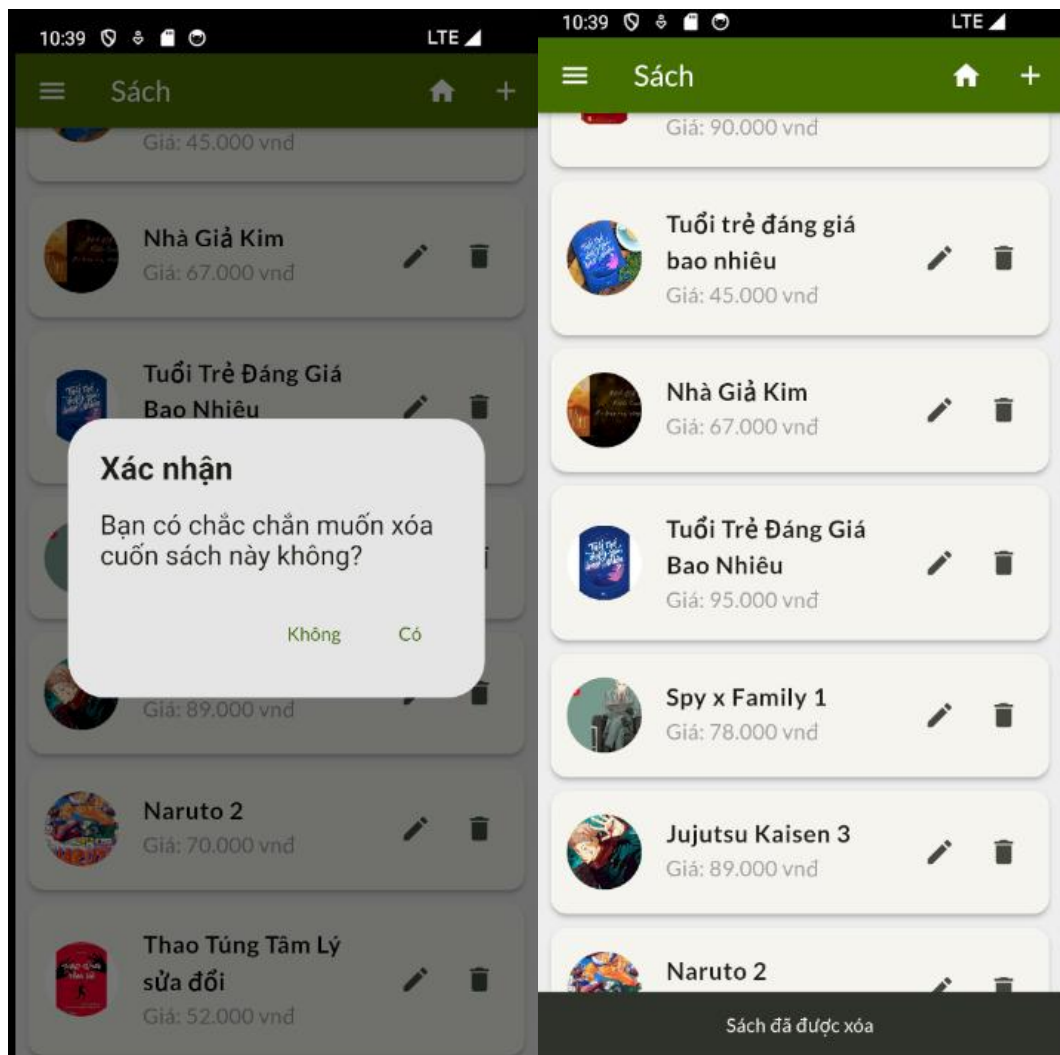
### **Đọc và lưu trữ dữ liệu:**

Chức năng này lấy dữ liệu từ firebase. Thông tin sách đã được lưu trên realtime database dưới dạng Json.

- Gọi thông qua hàm “updateBook” trong file “books\_service”
- Nhánh truy xuất: books
- Dữ liệu : title, author, category, description, imageUrl, price, quantity.

### **\*Chức năng xóa sách**

- **Miêu tả chức năng:** Cho phép người dùng xóa sách đã thêm.
- **Hình ảnh giao diện khi xóa:**



### Trạng thái được quản lý:

Chức năng này sử dụng giải pháp quản lý trạng thái chia sẻ thông qua Provider Pattern.

- Khi bấm icon xóa sẽ load lại trang quản lý sách mất đi sách đó.

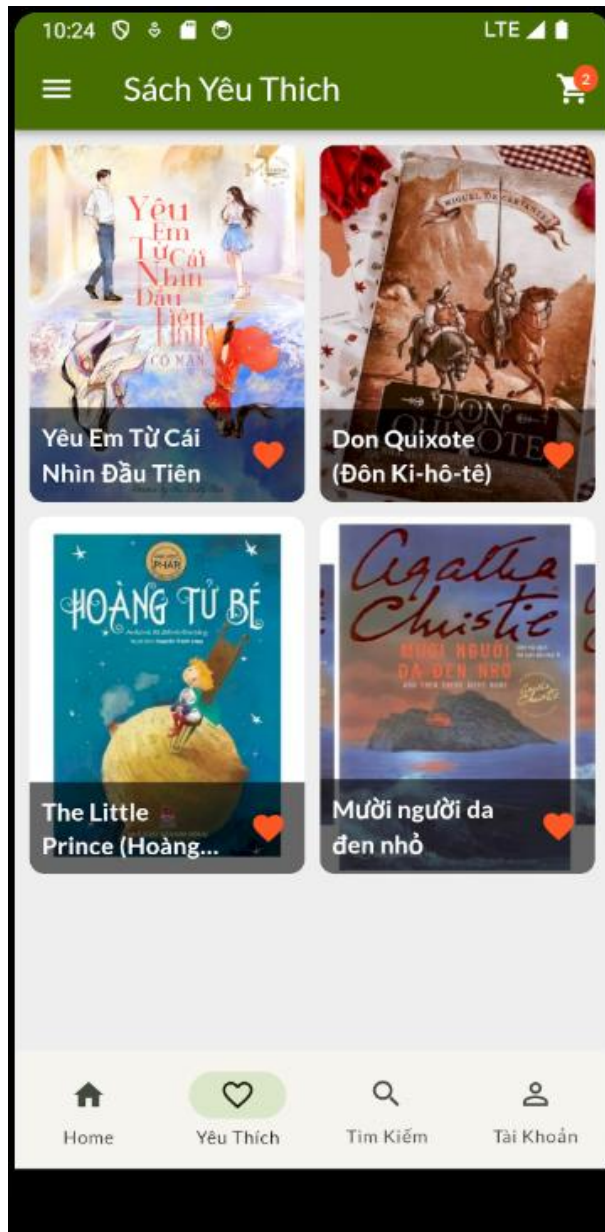
### Đọc và lưu trữ dữ liệu:

Chức năng này lưu dữ liệu từ firebase. Thông tin sách sẽ được lưu trên realtime database lưu trữ dạng Json.

- Gọi thông qua hàm “deleteBook” trong file “books\_service”
- Nhánh lưu trữ: books

## 10. Giao diện trang sách đã yêu thích

- **Miêu tả giao diện:** Cho phép người dùng xem sách mình đã thêm vào yêu thích.
- **Hình ảnh giao diện:**



**Các widget được sử dụng trong chức năng/giao diện này:**

Scaffold, AppBar, Icon, AppDrawer, BookGrid, PopupMenuButton, PopupMenuItem, BottomNavigationBar, Drawer, Column, Text, Icon, MaterialPageRoute, AppBar, ListTile.

**Các thư viện/plugin được sử dụng:**



- provider: Được sử dụng để quản lý trạng thái chia sẻ giữa các widget.

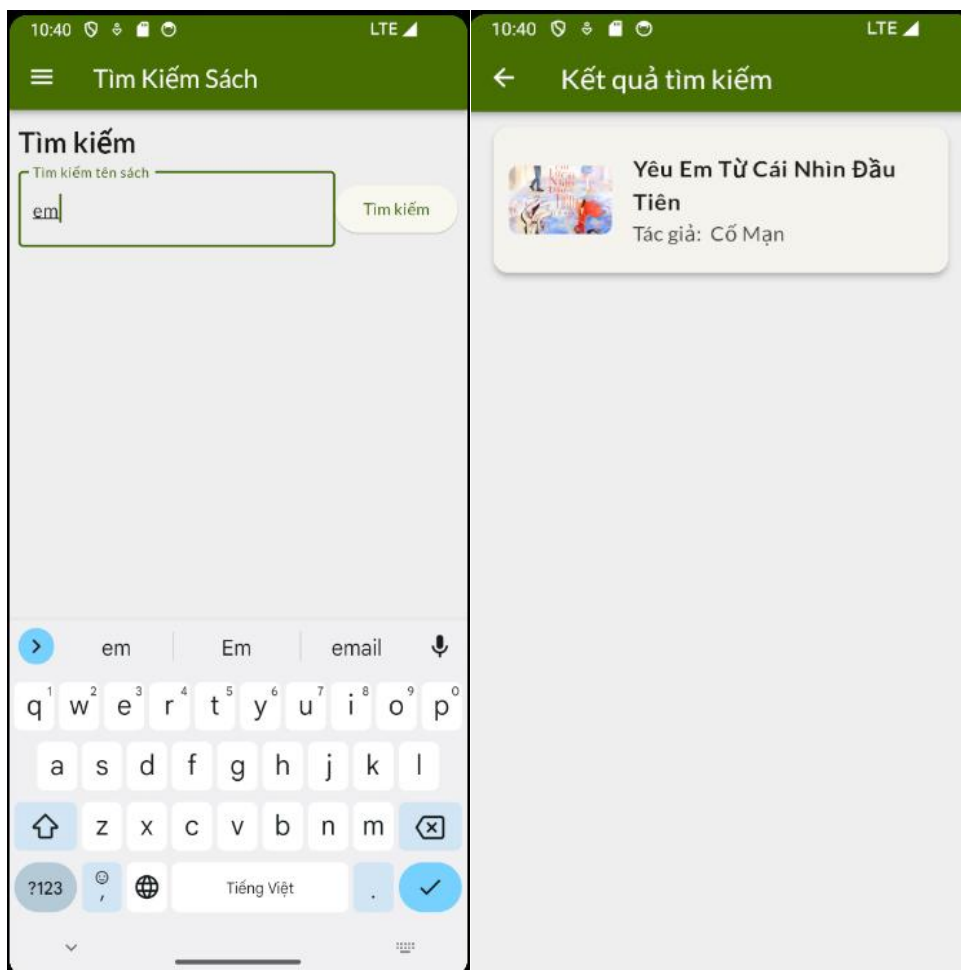
### Đọc và lưu trữ dữ liệu:

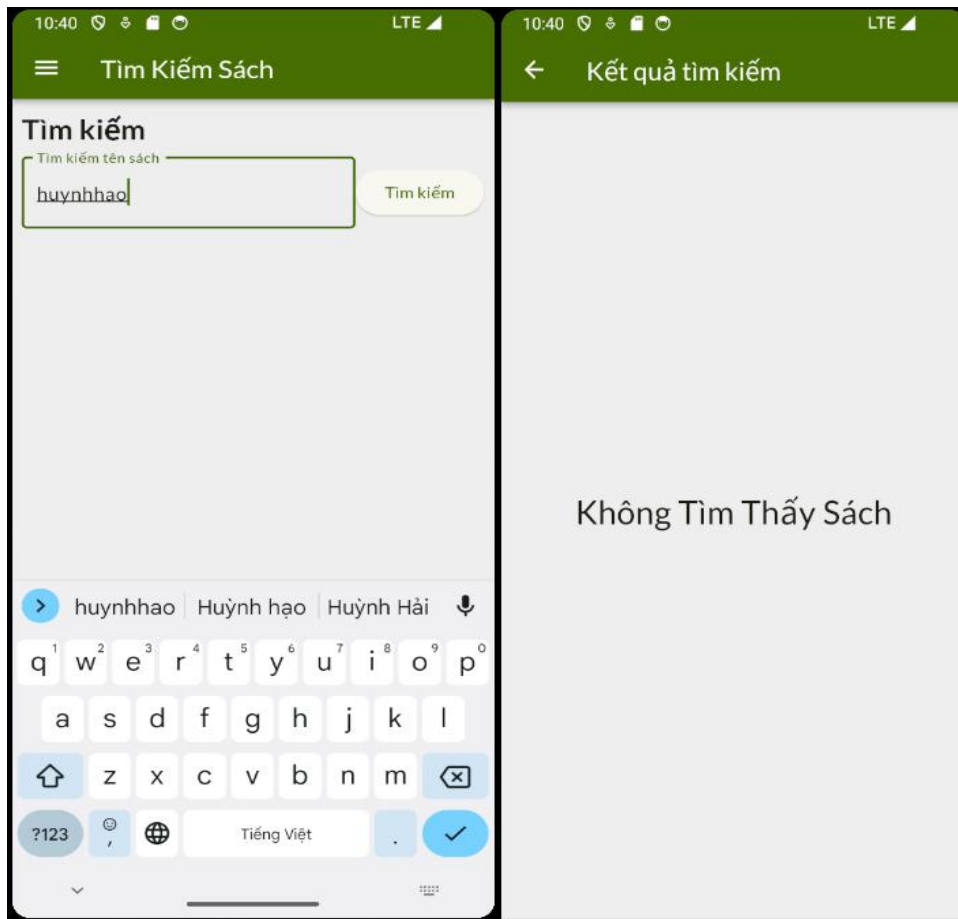
Chức năng này lấy dữ liệu từ firebase. Thông tin sách đã được lưu ở nhánh userFavorite trên realtime database.

- Thông qua biến cục bộ
- Nhánh truy xuất: userFavorite

## 11. Giao diện tìm kiếm sách

- **Miêu tả giao diện:** Cho phép người dùng tìm kiếm sách theo tên sách. Nếu tìm thấy sẽ hiện sách có tên tương ứng, không có sẽ hiển thị dòng “Không tìm thấy sách”.
- **Hình ảnh giao diện:**





### Các widget được sử dụng trong chức năng tìm sách

Scaffold, AppBar, AppDrawer, Column, Padding, Text, TextField, IconButton, ListView, Padding, Expanded, Icon, Center, ClipRRect.

### Thư viện/plugin được sử dụng:

- provider: Được sử dụng để quản lý trạng thái chia sẻ giữa các widget.
- flutter\_dotenv: Được sử dụng để đọc biến môi trường từ tệp .env.
- http: Được sử dụng để thực hiện các yêu cầu HTTP để giao tiếp với backend.

### Trạng thái được quản lý:

Chức năng này sử dụng giải pháp quản lý trạng thái chia sẻ thông qua Provider Pattern.

- Lấy dữ liệu từ booksManager. Khi bấm tìm kiếm sẽ hiển thị sách ra màn hình.

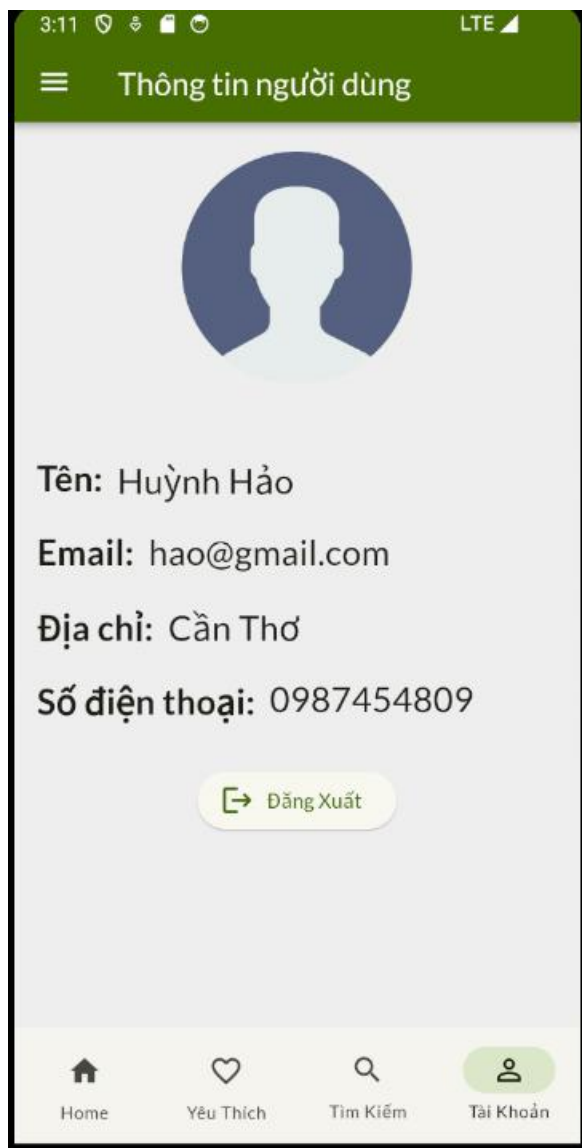
### Đọc và lưu trữ dữ liệu:

Chức năng này lấy dữ liệu từ firebase. Thông tin sách đã được lưu trên realtime database dưới dạng Json.

- Gọi thông qua hàm searchBooksByName trong file “books\_service”
- Nhánh truy xuất: books
- Dữ liệu truy xuất: title.

## 12. Giao diện tài khoản - Đăng xuất

- **Miêu tả giao diện:** Trang hiển thị gmail người dùng, khi bấm đăng xuất sẽ thoát khỏi app.
- **Hình ảnh giao diện:**



**Các widget được sử dụng trong chức năng/giao diện này:**

Scaffold, Text, Image, Icon, AppBar, AppDrawer, Center, NetworkImage, CircleAvatar, Column, Row, ListView.builder, SizedBox

**Các thư viện/plugin được sử dụng:**

provider: Được sử dụng để quản lý trạng thái chia sẻ giữa các widget.

**Đọc và lưu trữ dữ liệu:**

Chức năng này lấy dữ liệu từ firebase. Thông tin được lưu trên dịch vụ Firebase Authentication

- Gọi thông qua hàm “fetchProfile” để lấy thông tin người dùng và “clearSavedAuthToken ” để đăng xuất trong file “auth\_service”
- Dữ liệu truy xuất: name, email, address, phone