

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 mua sắm trang sức
- Link GitHub mã nguồn: <https://github.com/23-24Sem2-Courses/ct48403-project-nguyenmaixuanhuong>
- MSSV 1: B2014572
- Họ tên SV 1: Nguyễn Mai Xuân Hương
- MSSV 2: B2014594
- Họ tên SV 2: Đỗ Huỳnh Như
- Lớp học phần: CT484_03

I. Tổng quan

- **Miêu tả dự án/ứng dụng:** vài câu miêu tả tổng quan về dự án/ứng dụng.

Ứng dụng mua sắm trang sức giúp khách hàng xem và tìm kiếm các sản phẩm, cung cấp thông tin chi tiết. Khách hàng có thể thêm vào mục yêu thích và giỏ hàng, cũng như xóa sản phẩm khi cần. Sau khi thêm vào giỏ hàng, khách hàng có thể xem lại và xác nhận đơn hàng, bổ sung thông tin cá nhân trước khi mua hàng. Cũng có thể xem lại thông tin các đơn hàng đã đặt trước đó.

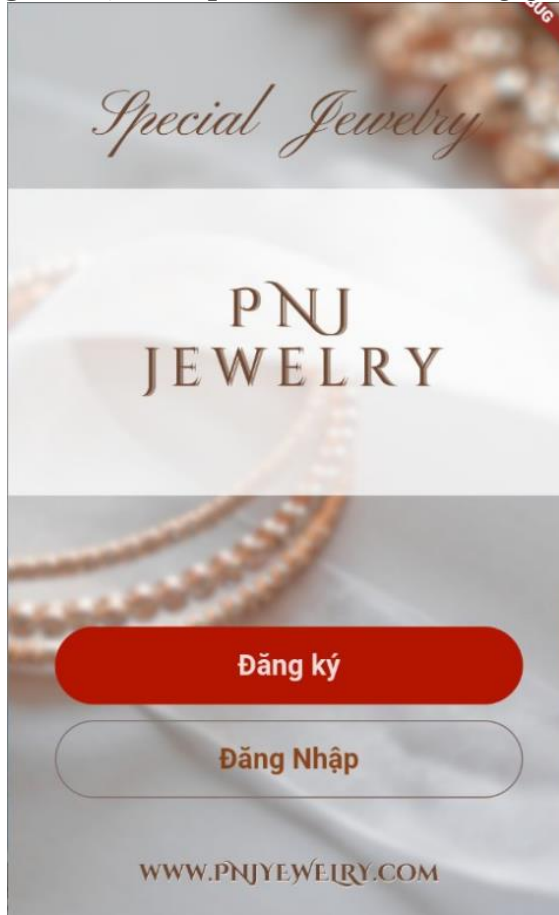
- **Bảng phân công công việc nếu làm nhóm.**

STT	Tên Thành Viên	Công Việc
1	Đỗ Huỳnh Như	<ul style="list-style-type: none">- Thêm và hiển thị trang sản phẩm yêu thích- Xây dựng giao diện trang Home- Xây dựng trang chi tiết sản phẩm- Xây trang tìm kiếm- Xây dựng trang Order- Xây dựng trang quản lý sản phẩm
2	Nguyễn Mai Xuân Hương	<ul style="list-style-type: none">- Xây dựng đăng ký, đăng nhập- Xây dựng trang Home- Xây dựng trang tìm Kiếm- Xây dựng trang Cart- Xây dựng trang Order- Xây dựng trang quản lý sản phẩm

II. Chi tiết các chức năng

1. Chức năng/giao diện 1: Trang Welcome

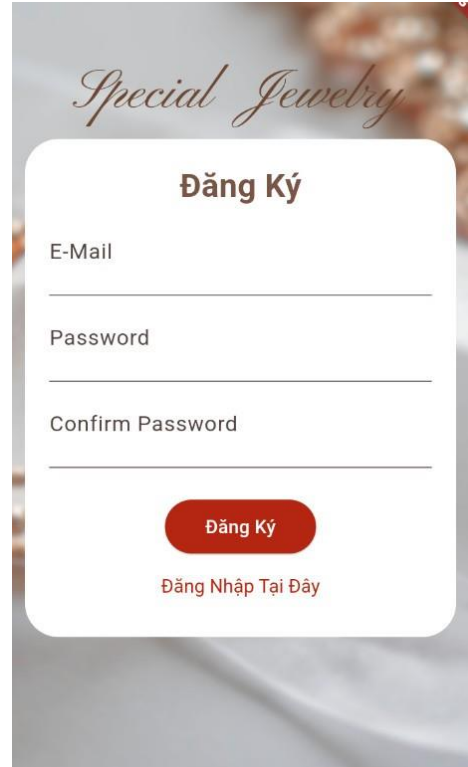
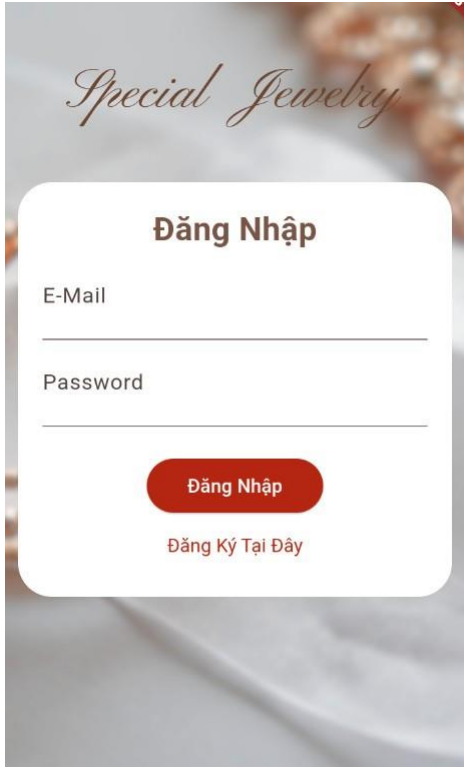
- **Miêu tả chức năng/giao diện:** Người dùng khi truy cập vào ứng dụng sẽ hiện thị trang welcome đầu tiên. Ở trang này người dùng có thể chọn đăng nhập hoặc đăng ký để vào trang chủ của cửa hàng.
- **Ảnh chức năng/giao diện:** chụp (các) ảnh chức năng/giao diện 1



- **Chi tiết cài đặt:** trình bày cách đặt của chức năng/giao diện 1. Cụ thể, sinh viên trả lời các câu hỏi sau:
 - + Các widget được sử dụng trong giao diện này là: Container, Stack, Positioned, filleButton, DecorationImage, OutlineButton, Text

2. Chức năng/giao diện 2: Trang đăng ký, đăng nhập

- **Miêu tả chức năng/giao diện:** Người dùng có thể đăng nhập tài khoản bằng email và password hoặc đăng ký nếu chưa có tài khoản.
- **Ảnh chức năng/giao diện:**



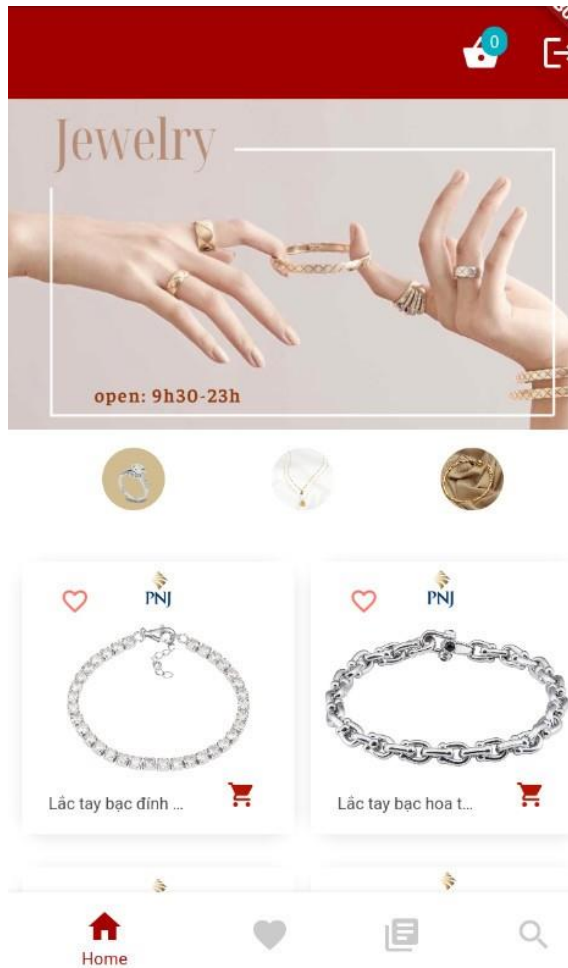
- **Chi tiết cài đặt:**
 - + Các widget được sử dụng trong giao diện này là: Scaffold, Container, Form, Center, Card, Column, SingleChildScrollView, Text, Padding, TextFormField, ValueListenableBuilder, CircularProgressIndicator, InputDecoration, Icon, IconButton, SizedBox
 - + Sử dụng thư viện và plugin:
 - **Lotties:** là một công cụ mạnh mẽ trong Flutter cho phép bạn thêm các animation phức tạp vào ứng dụng của mình một cách dễ dàng
 - **Provider:** giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http:** giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert :** giúp chuyển đổi thành kiểu Json và ngược lại;
 - **shared_preferences:** giúp cung cấp một không gian lưu trữ cục bộ để lưu lại token của người dùng.
 - + Sử dụng giải pháp quản lý trạng thái là: Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng AuthManager (with ChangeNotifier) lưu trữ

thông tin xác thực người dùng và thực hiện việc xác thực cũng như đăng ký. Tại chức năng này sử dụng `context.read<AuthManager>()` để truy cập đối tượng `AuthManager` được cung cấp.

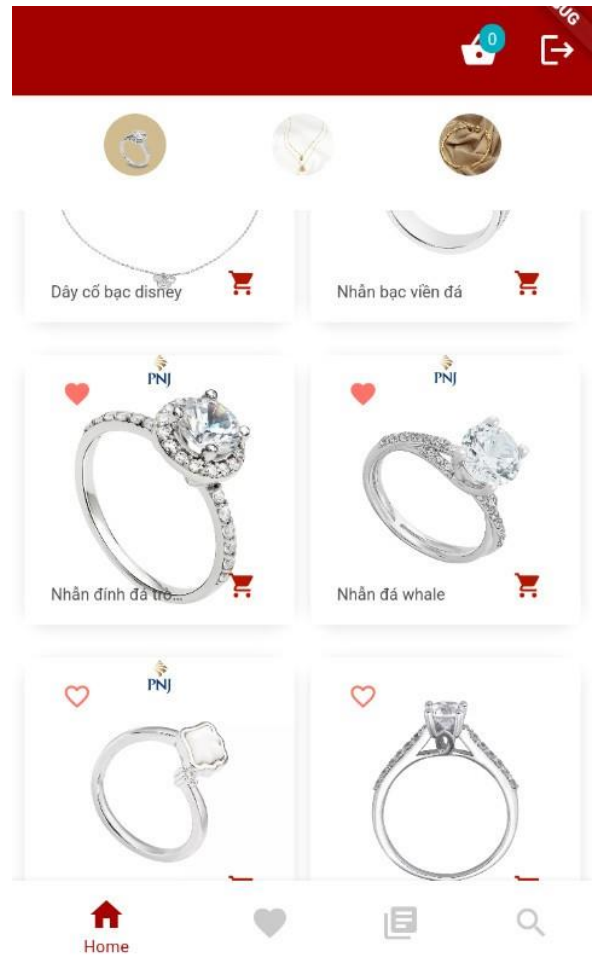
+ Chức năng này thực hiện việc xác thực và lưu trữ thông tin người dùng thông qua dịch vụ Firebase Authentication.

3. Chức năng/giao diện 3: Trang chủ (Home)

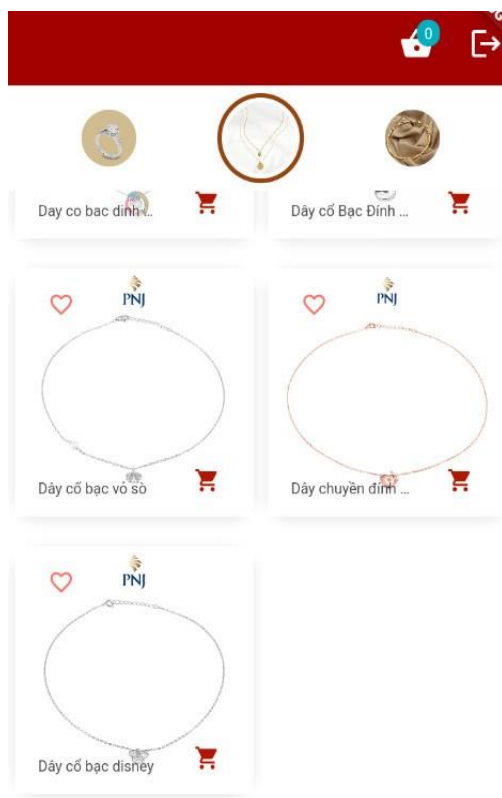
- **Miêu tả chức năng/giao diện:** Khách hàng có thể xem các sản phẩm có tại cửa hàng, cũng như thêm sản phẩm vào danh mục yêu thích hoặc thêm vào giỏ hàng
- **Ảnh chức năng/giao diện:**



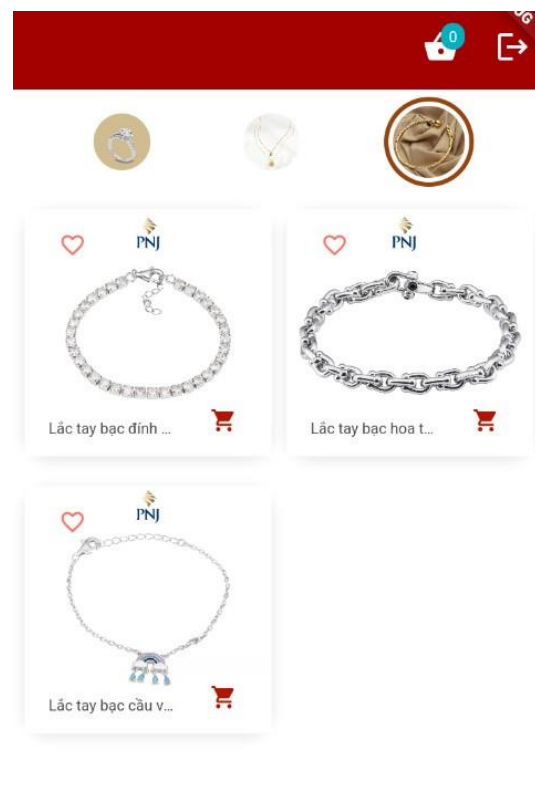
<Giao diện chính>



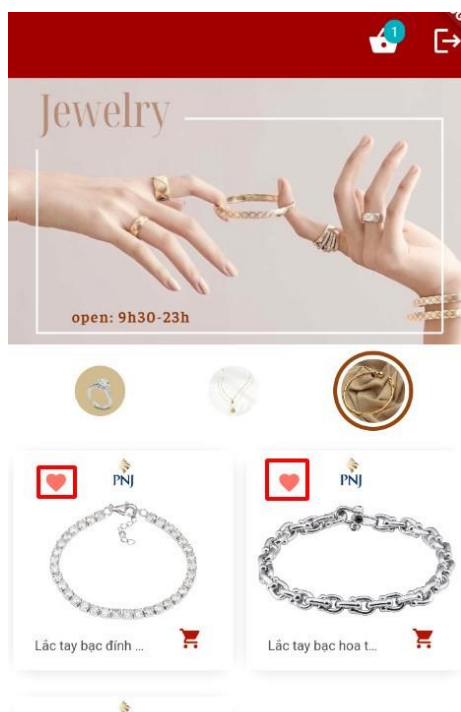
<Giao diện khi cuộn>



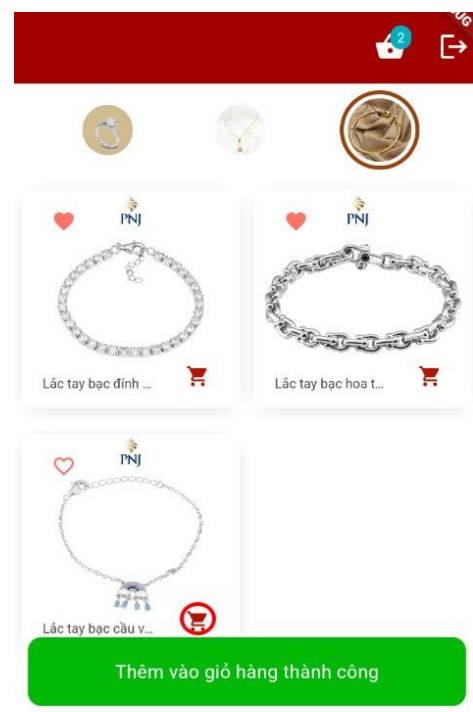
<Giao diện theo loại Dây chuyền>



<Giao diện theo loại Vòng>



<Thêm vào sản phẩm yêu thích>



<Thêm vào giỏ hàng>

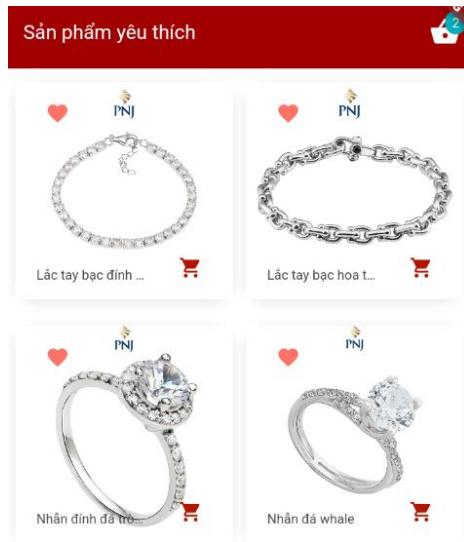
- **Chi tiết cài đặt:**

- + Các widget được sử dụng trong giao diện này là: Scaffold, Container, Center, Card, Column, Text, SizedBox, Padding, ValueListenableBuilder, CircularProgressIndicator, Icon, IconButton, NestedScrollView, GestureDetector, bottomNavigationBar, GridView, GridTileBar, Image, GridTile, SnackBar
 - NestedScrollView là một widget mạnh mẽ được sử dụng để tạo ra các giao diện cuộn lồng nhau (nested scrolling interfaces), tức là các giao diện có thể cuộn được bên trong nhau. NestedScrollView thường được sử dụng khi ta muốn kết hợp một ScrollView bên trong một SliverAppBar. Trong đó SliverAppBar làm đầu vào cho thanh đầu trang (app bar) SliverList, SliverGrid, hoặc bất kỳ Sliver widget nào khác làm nội dung cuộn bên dưới.
- + Sử dụng thư viện và plugin: **provider, http, convert**
 - **Provider:** giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http:** giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert** :giúp chuyển đổi thành kiểu Json và ngược lại;
- + Sử dụng giải pháp quản lý trạng thái là:
 - Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductManager và CartManager (with ChangeNotifier) để quản lý hiển thị danh sách các sản phẩm trang sức, thêm các sản phẩm vào mục yêu thích và lọc sản phẩm theo từng loại .
 - Bên cạnh đó quản lý việc hiển thị số lượng sản phẩm trong giỏ hàng cũng như thêm sản phẩm vào giỏ hàng.
 - Tại chức năng này sử dụng context.select<ProductsManager, List<Product>> () {} và context.watch< CartManager >() để truy cập đối tượng ProductManager và CartManager được cung cấp.
 - Sử dụng StatefullWidget với biến trạng thái _nameCatoegory để lọc và hiển thị các sản phẩm theo loại (nhẫn, dây chuyền, vòng hoặc tất cả) và biến showFavorite để hiển thị sản phẩm yêu thích.
- + Chức năng này đọc dữ liệu các sản phẩm trang sức từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON là

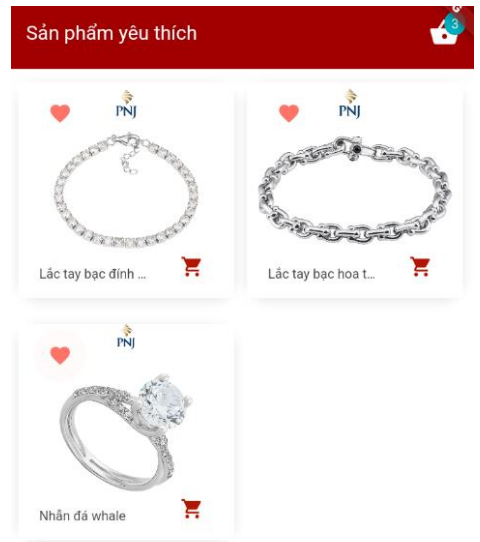
```
{ "products":  
  [ "id":  
    { "description": ,  
      "imageUrl": ,  
      "nameProduct":  
      , "price": ,  
      "nameCategory"  
      _isFavorite: , } ] }
```

4. Chức năng/giao diện 4: Giao diện trang sản phẩm yêu thích

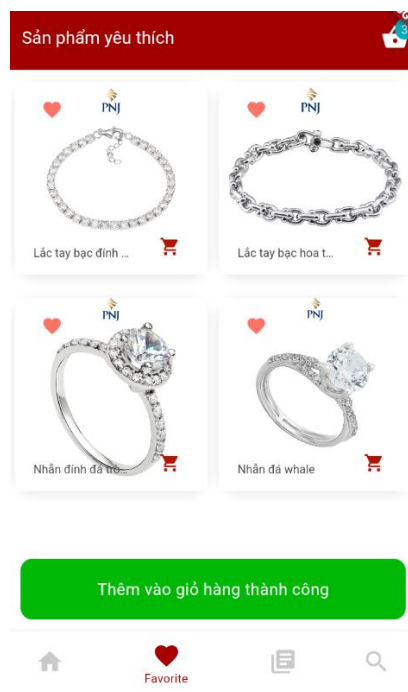
- **Miêu tả chức năng/giao diện:** Giao diện này hiển thị các sản phẩm mà người dùng đã thêm vào danh mục yêu thích.
- **Ảnh chức năng/giao diện:**



<Hiện sản phẩm yêu thích>



<Xóa 1 sản phẩm yêu thích>



<Thêm sản phẩm vào giỏ hàng>

- **Chi tiết cài đặt:**

- + Các widget được sử dụng trong giao diện này là: Scaffold, Container, Center, Card, Column, Text, SizedBox, Padding, FutureBuilder, ValueListenableBuilder, Icon, IconButton, GestureDetector, bottomNavigationBar, GridView, GridTileBar, Image, GridTile, SnackBar, CircularProgressIndicator
- + Sử dụng thư viện và plugin: **provider, http, convert**
 - **Provider:** giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http:** giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert:** giúp chuyển đổi thành kiểu Json và ngược lại;
- + Sử dụng giải pháp quản lý trạng thái là:
 - Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductManager và CartManager (with ChangeNotifier) để quản lý hiển thị các sản phẩm được lưu vào mục yêu thích cũng như xóa phẩm ra khỏi mục yêu thích.
 - Bên cạnh đó quản lý việc hiển thị số lượng sản phẩm trong giỏ hàng cũng như thêm sản phẩm vào giỏ hàng.
 - Tại chức năng này sử dụng context.select<ProductsManager, List<Product>> () {} và context.watch< CartManager >() để truy cập đối tượng ProductManager và CartManager được cung cấp.
 - Sử dụng StatefullWidget với _showFavorite để hiển thị sản phẩm yêu thích.
- + Chức năng này đọc dữ liệu các sản phẩm trang sức từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON là

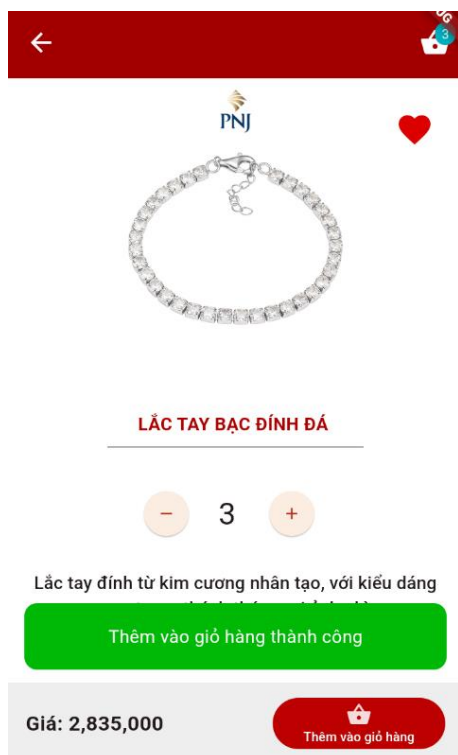
```
{ "products":  
  [ "id":  
    { "description":  
      "imageUrl":  
      "nameProduct":  
      , "price":  
      "nameCategory"  
      _isFavorite: } ] }
```


5. Chức năng/giao diện 5: Giao diện trang chi tiết sản phẩm

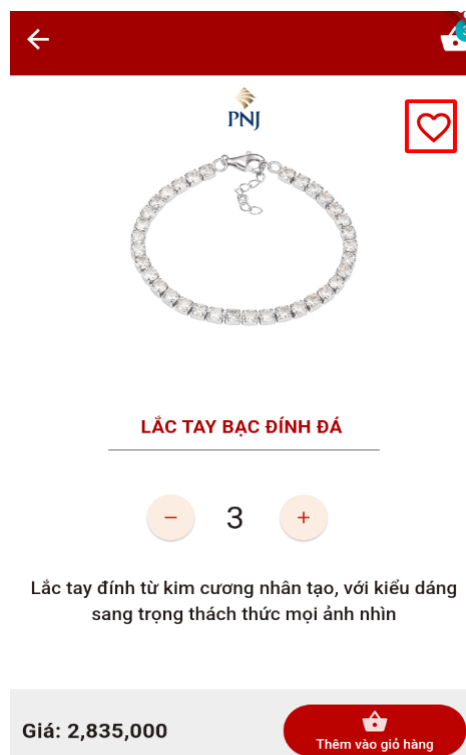
- **Miêu tả chức năng/ giao diện:** Giao diện hiển thị chi tiết các thông tin của sản phẩm và nút điều chỉnh số lượng tùy thích để thêm vài giỏ hàng và cũng có thể thêm sản phẩm vào mục yêu thích
- **Ảnh chức năng/ giao diện:**



<Giao diện mặc định trang chi tiết>



<Giao diện khi tùy chỉnh số lượng>

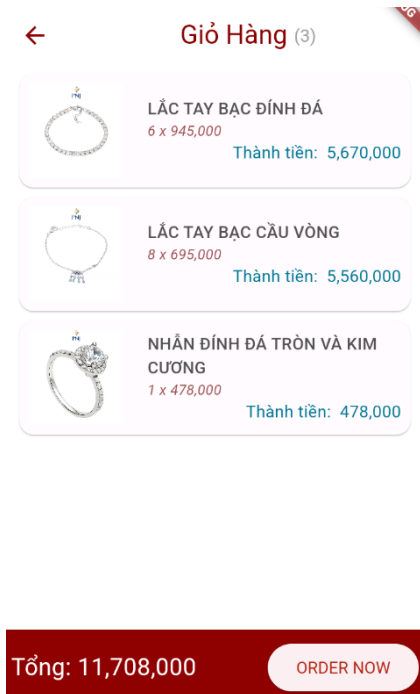


- **Chi tiết cài đặt:**

- + Các widget được sử dụng trong giao diện này là: Scaffold, AppBar, Container, Center, Card, Column, Text, SizedBox, Padding, FutureBuilder, ValueListenableBuilder, Icon, IconButton, bottomNavigationBar, Image, GridTile, SnackBar, Stack, Positioned, Row, FilledButton
- + Sử dụng thư viện và plugin: **provider, http, convert**
 - **Provider:** giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http:** giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert:** giúp chuyển đổi thành kiểu Json và ngược lại;
 - **Lotties:** là một công cụ mạnh mẽ trong Flutter cho phép bạn thêm các animation phức tạp vào ứng dụng của mình một cách dễ dàng
- + Sử dụng giải pháp quản lý trạng thái là:
 - Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductManager và CartManager (with ChangeNotifier) để quản lý hiển thị thông tin sản phẩm chi tiết.
 - Bên cạnh đó quản lý việc hiển thị số lượng sản phẩm trong giỏ hàng cũng như thêm sản phẩm vào giỏ hàng theo số lượng tùy chỉnh.
 - Tại chức năng này sử dụng context.watch<CartManager>() để truy cập đối tượng ProductManager và CartManager được cung cấp.

6. Chức năng/giao diện 6: Giao diện trang giỏ hàng (Cart)

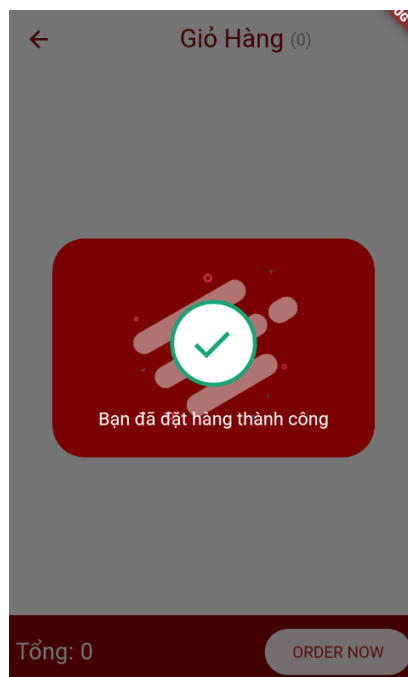
- **Miêu tả chức năng/ giao diện:** Giao diện hiển thị chi tiết các sản phẩm được thêm vào giỏ hàng và tổng tiền của các sản phẩm.
- **Ảnh chức năng/ giao diện:**



<Trang hiển thị sản phẩm trong giỏ hàng>



<Form xác nhận thông đặt hàng >



<Giao diện đặt hàng thành công>

- **Chi tiết cài đặt:**

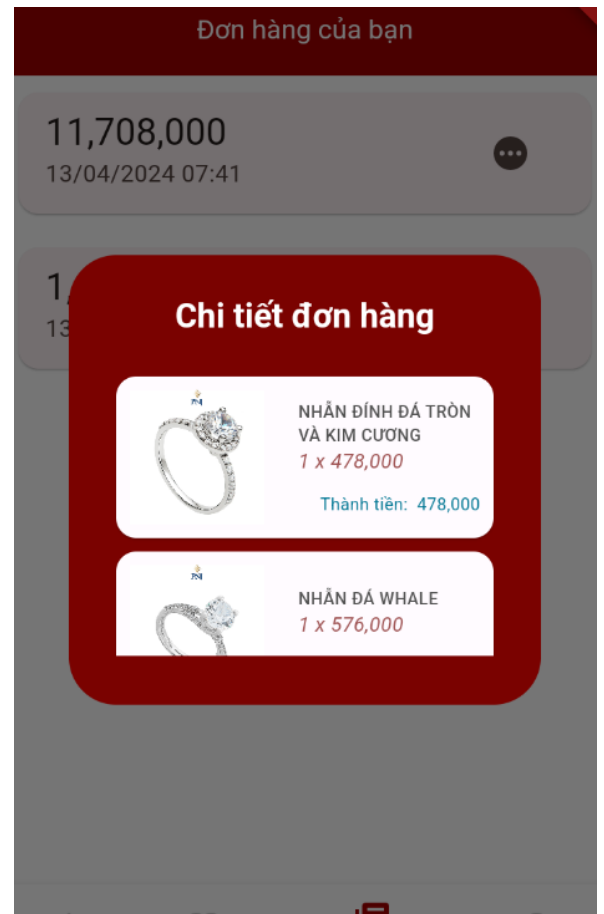
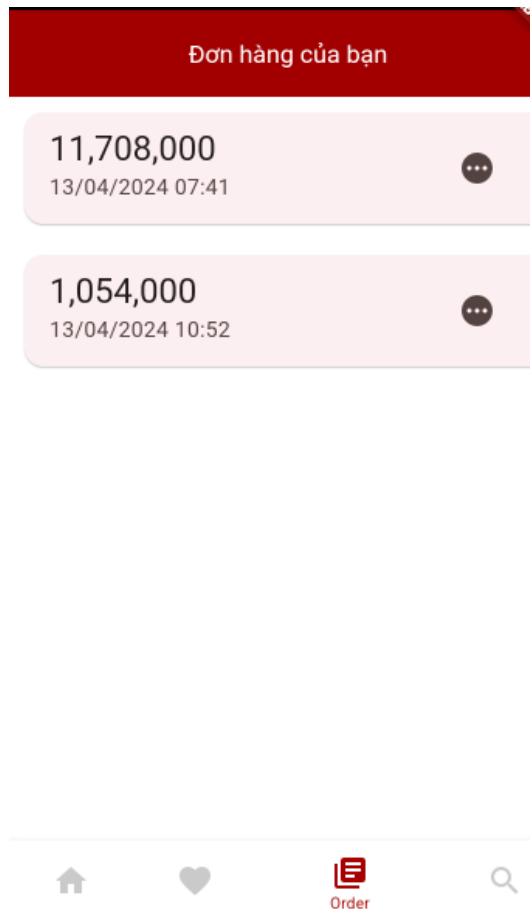
- + Các widget được sử dụng trong giao diện này là: Scaffold, AppBar, Container, Center, Card, Column, Text, SizedBox, Padding, FutureBuilder, ValueListenableBuilder, Icon, IconButton, bottomNavigationBar, Image, GridTile, SnackBar, Stack, Positioned, Row, FilledButton, ListView, GestureDetector, showDialog, Expanded, OutlineInputBorder, Form, Dismissible
- + Sử dụng thư viện và plugin: **provider, http, convert**
 - **Provider:** giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http:** giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert:** giúp chuyển đổi thành kiểu Json và ngược lại;
 - **Lotties:** là một công cụ mạnh mẽ trong Flutter cho phép bạn thêm các animation phức tạp vào ứng dụng của mình một cách dễ dàng
- + Sử dụng giải pháp quản lý trạng thái là:
 - Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng OrdersManager và CartManager (with ChangeNotifier) để quản lý hiển thị các CartItem trong trang Cart và xác nhận thông tin để tạo OrderItem.
 - Tại chức năng này sử dụng context.watch<CartManager>() để truy cập đối tượng CartManager được cung cấp.
- + Chức năng này đọc dữ liệu các sản phẩm trong giỏ hàng từ dịch vụ lưu trữ Firebase

Realtime Database. Với cấu trúc JSON là:

```
“carts”:  
  [“idUser”:  
    [“id”: {  
      “id”:  
      “nameProduct”:  
      “imageUrl”:  
      “quantity”:  
      “price”:  
    }  
  ]  
]
```

7. Chức năng/giao diện 7: Giao diện trang đơn hàng (Order)

- **Miêu tả chức năng/ giao diện:** Giao diện hiển thị thông tin các đơn hàng đã đặt của người dùng và có thể xem chi tiết từng đơn hàng.
- **Ảnh chức năng/ giao diện:**



- Chi tiết cài đặt:

- + Các widget được sử dụng trong giao diện này là: Scaffold, AppBar, Container, Center, Card, Column, Text, SizedBox, Padding, FutureBuilder, ValueListenableBuilder, Icon, IconButton, bottomNavigationBar, Image, Lotties, Stack, Positioned, Row, FilledButton
- + Sử dụng thư viện và plugin: **provider**, **http**, **convert**
 - **Provider:** giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http:** giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert:** giúp chuyển đổi thành kiểu Json và ngược lại;
 - **Lotties:** là một công cụ mạnh mẽ trong Flutter cho phép bạn thêm các animation phức tạp vào ứng dụng của mình một cách dễ dàng
- + Sử dụng giải pháp quản lý trạng thái là:

- Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductsManager (with ChangeNotifier) để quản lý hiển thị thông tin sản phẩm ứng với nội dung tìm kiếm.
 - Ta sử dụng một biến nameProduct để quản lý nội dung muốn tìm kiếm
 - Tại chức năng này sử dụng Cosumner để lắng nghe sự thay đổi của ProductsManager.
- + Chức năng này đọc dữ liệu các đơn hàng từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON là:

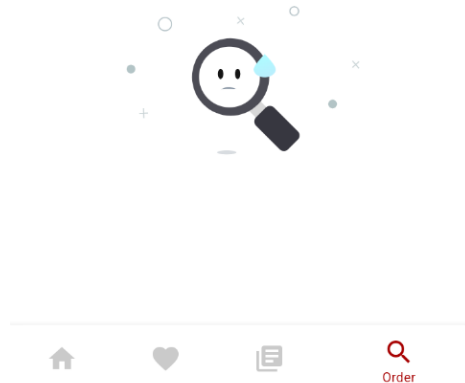
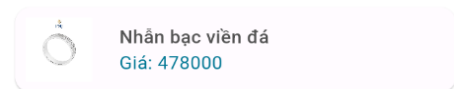
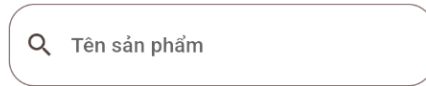
```

“orders”: [
  “idUser”: [
    “idorderItem”: {
      “address”:
      “amount”:
      “dateTime”:
      “id”:
      “phoneNumber”:
      “products”: [
        {
          “id”:
          “nameProduct”:
          “imageUrl”:
          “quantity”:
          “price”:
        }
      ]
    }
  ]
]

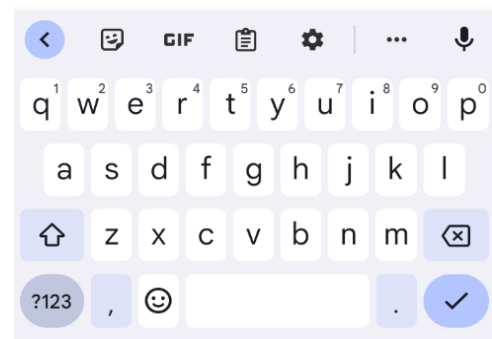
```

8. Chức năng/giao diện 8: Giao diện trang tìm kiếm (search)

- **Miêu tả chức năng/ giao diện:** Giao diện giúp người dùng tìm kiếm các sản phẩm theo tên sản phẩm, có thể nhấn vào kết quả tìm kiếm để truy cập vào trang chi tiết của sản phẩm đó.
- **Ảnh chức năng/ giao diện:**



<Giao diện khi chưa tìm kiếm>



<Giao diện khi tìm kiếm>

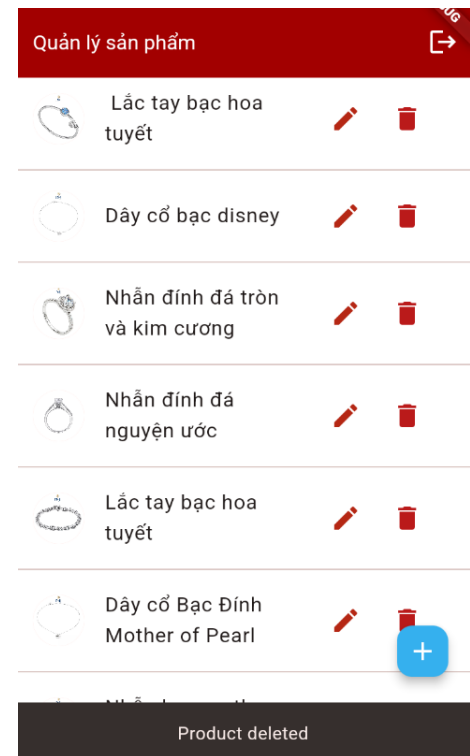
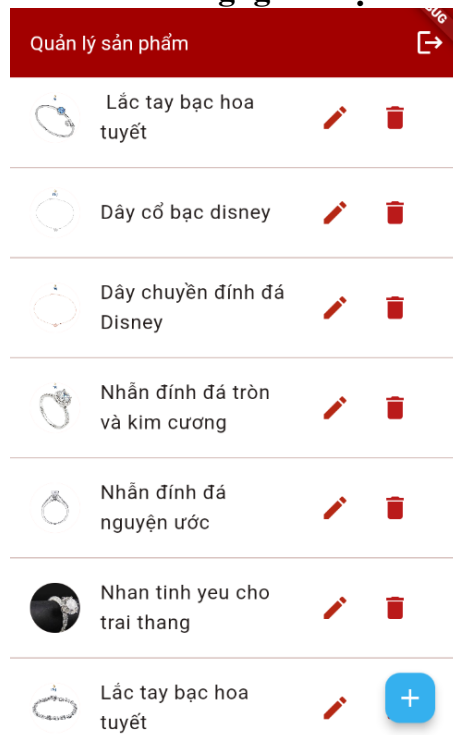
- **Chi tiết cài đặt:**
 - + Các widget được sử dụng trong giao diện này là: Scaffold, AppBar, Container, Center, Card, Column, Text, Padding, FutureBuilder, Icon, InputDecoration, bottomNavigationBar, Image, Lotties, Row, FilledButton, SizeBox, Align, Expanded, ListView,
 - + Sử dụng thư viện và plugin: **provider**, **http**, **convert**
 - **Provider:** giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http:** giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert:** giúp chuyển đổi thành kiểu Json và ngược lại;
 - + Sử dụng giải pháp quản lý trạng thái là:
 - Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng OrdersManager (with ChangeNotifier) để quản lý hiển thị thông tin các đơn hàng và chi tiết đơn hàng.
 - Tại chức năng này sử dụng Cosumner để lắng nghe sự thay đổi của OrdersManager.

- + Chức năng này đọc dữ liệu các sản phẩm trang sức từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON là:

```
{ "products":  
  [{"id":  
    { "description":  
      "imageUrl":  
      "nameProduct":  
      , "price":  
      "nameCategory"  
      _isFavorite: } ] }
```


9. Chức năng/giao diện 9: Giao diện trang quản lý sản phẩm

- **Miêu tả chức năng/ giao diện:** Giao diện dùng để hiển thị các sản phẩm trong cửa hàng. Ta có thể thêm chỉnh sửa hoặc xóa sản phẩm.
- **Ảnh chức năng/ giao diện:**



<Xóa sản phẩm>

- Chi tiết cài đặt

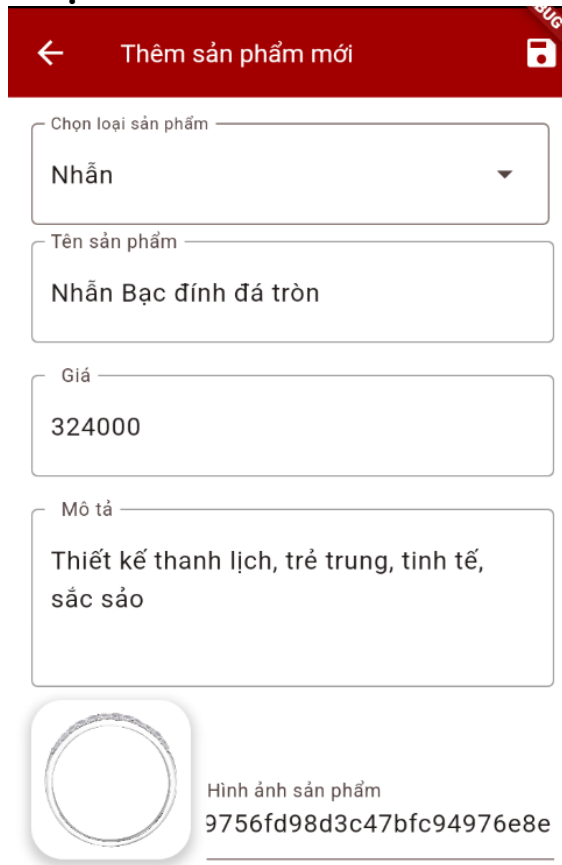
- + Các widget được sử dụng trong giao diện này là: Scaffold, AppBar, Container, Center, Card, Column, Text, SizedBox, Padding, FutureBuilder, Icon, IconButton, Row, FilledButton, FloatingActionButton, RefreshIndicator, ListView, ListTile, CircleAvatar, SnackBar
- + Sử dụng thư viện và plugin: **provider, http, convert**
 - **Provider:** giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http:** giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert:** giúp chuyển đổi thành kiểu Json và ngược lại;
- + Sử dụng giải pháp quản lý trạng thái là:
 - Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductsManager (with ChangeNotifier) để quản lý hiển thị thông tin các sản phẩm trong cửa hàng
 - Ta sử dụng Navigator để điều hướng qua các trang thêm và chỉnh sửa sản phẩm

- Tại chức năng này sử dụng Cosumner để lắng nghe sự thay đổi của ProductsManager.
- + Chức năng này đọc dữ liệu các sản phẩm trang sức từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON là:

```
{ "products":  
  [{"id":  
    { "description":  
      "imageUrl":  
      "nameProduct":  
      , "price":  
      "nameCategory"  
      _isFavorite: } ] }
```

10. Chức năng/giao diện 10: Giao diện trang thêm sản phẩm mới

- **Miêu tả chức năng/ giao diện:** Giao diện này giúp cho admin thêm sản phẩm mới vào cửa hàng
- **Ảnh chức năng/ giao diện:**



← Thêm sản phẩm mới

Chọn loại sản phẩm
Nhẫn

Tên sản phẩm
Nhẫn Bạc đính đá tròn

Giá
324000

Mô tả
Thiết kế thanh lịch, trẻ trung, tinh tế, sắc sảo

Hình ảnh sản phẩm
9756fd98d3c47bfc94976e8e

- Chi tiết cài đặt:

- + Các widget được sử dụng trong giao diện này là: Scaffold, AppBar, Container, Center, Card, Column, Text, SizedBox, Padding, Icon, IconButton, Image, Row, ListView, Form, TextFormField, InputDecoration, ClipRRect, Expanded, DropdownMenu,
- + Sử dụng thư viện và plugin: **provider**, **http**, **convert**
 - **Provider:** giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http:** giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert:** giúp chuyển đổi thành kiểu Json và ngược lại;
- + Sử dụng giải pháp quản lý trạng thái là:
 - Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductsManager (with ChangeNotifier) để quản lý thêm sản phẩm vào cửa hàng

- + Chức năng này đọc dữ liệu các sản phẩm trang sức từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON là:

```
{ "products":  
  [{"id":  
    { "description":  
      "imageUrl":  
      "nameProduct":  
      , "price":  
      "nameCategory"  
      _isFavorite: } ] }
```

11. Chức năng/giao diện 11: Giao diện trang **Chỉnh sửa sản phẩm**

- **Miêu tả chức năng/ giao diện:** Giao diện này giúp cho admin chỉnh sửa sản phẩm có trong cửa hàng
- **Ảnh chức năng/ giao diện:**

The image shows two side-by-side screenshots of a mobile application interface for editing a product. Both screens have a red header bar with a back arrow and a save icon. The left screen is titled 'Thêm sản phẩm mới' (Add new product) and the right is 'Chỉnh sửa sản phẩm' (Edit product). Both screens show a form with the following fields:

- Chọn loại sản phẩm** (Select product type): A dropdown menu with 'Nhẫn' (Ring) selected.
- Tên sản phẩm** (Product name): A text field containing 'Nhẫn Bạc đính đá tròn' (Silver Ring with round stones).
- Giá** (Price): A text field. In the left screen, it contains '324000'. In the right screen, it contains '999000'. A red arrow points from the price field in the left screen to the price field in the right screen.
- Mô tả** (Description): A text field containing 'Thiết kế thanh lịch, trẻ trung, tinh tế, sắc sảo' (Elegant, youthful, sophisticated, and sharp design).
- Hình ảnh sản phẩm** (Product image): A circular image placeholder. In the left screen, it shows a placeholder image with the URL '9756fd98d3c47bfc94976e8e'. In the right screen, it shows a placeholder image with the URL 'https://down-vn.img.suserco'.

- **Chi tiết cài đặt:**

- + Các widget được sử dụng trong giao diện này là: Scaffold, AppBar, Container, Center, Card, Column, Text, SizedBox, Padding, Icon, IconButton, Image, Row, ListView, Form, TextFormField, InputDecoration, ClipRRect, Expanded, DropdownMenu,
- + Sử dụng thư viện và plugin: **provider**, **http**, **convert**
 - **Provider**: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget;
 - **http**: giúp thực hiện các yêu cầu thông qua giao thức http
 - **convert**: giúp chuyển đổi thành kiểu Json và ngược lại;
- + Sử dụng giải pháp quản lý trạng thái là:
 - Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductsManager (with ChangeNotifier) để quản lý chỉnh sửa sản phẩm trong cửa hàng
- + Chức năng này đọc dữ liệu các sản phẩm trang sức từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON là:

```
{ "products":  
  [{"id":  
    { "description":  
      "imageUrl":  
      "nameProduct":  
      , "price":  
      "nameCategory"  
      _isFavorite: } ] }
```