

# 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 chia sẻ hình ảnh
- **Link GitHub mã nguồn:** <https://github.com/23-24Sem2-Courses/ct48401-project-minhdubai31>
- **MSSV:** Trà Hoàng Minh
- **Họ tên SV:** B2014671
- **Lớp học phần:** CT48401

## I. Tổng quan

### Miêu tả dự án/ứng dụng:

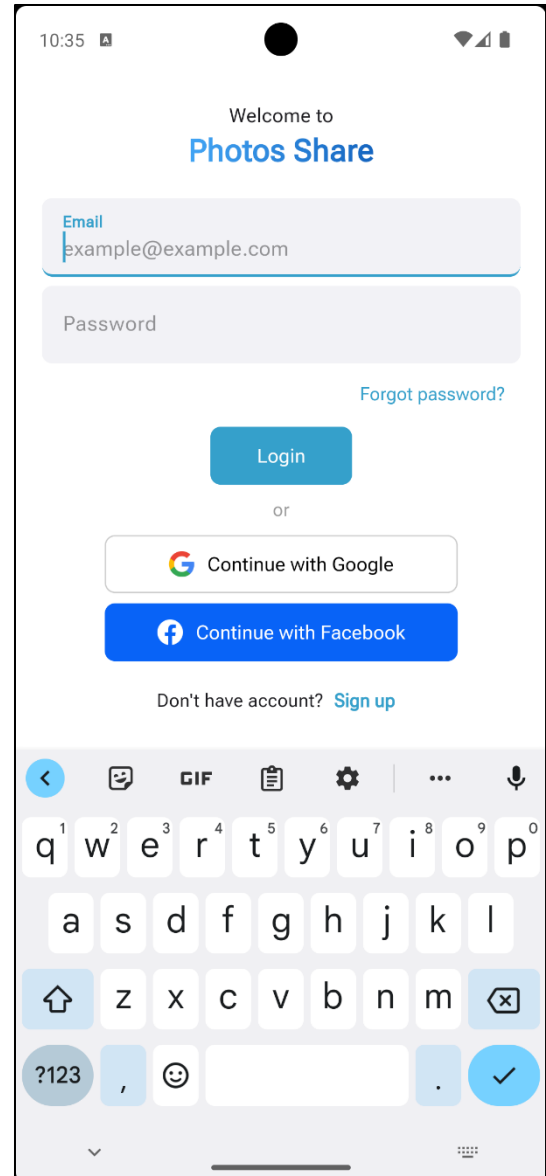
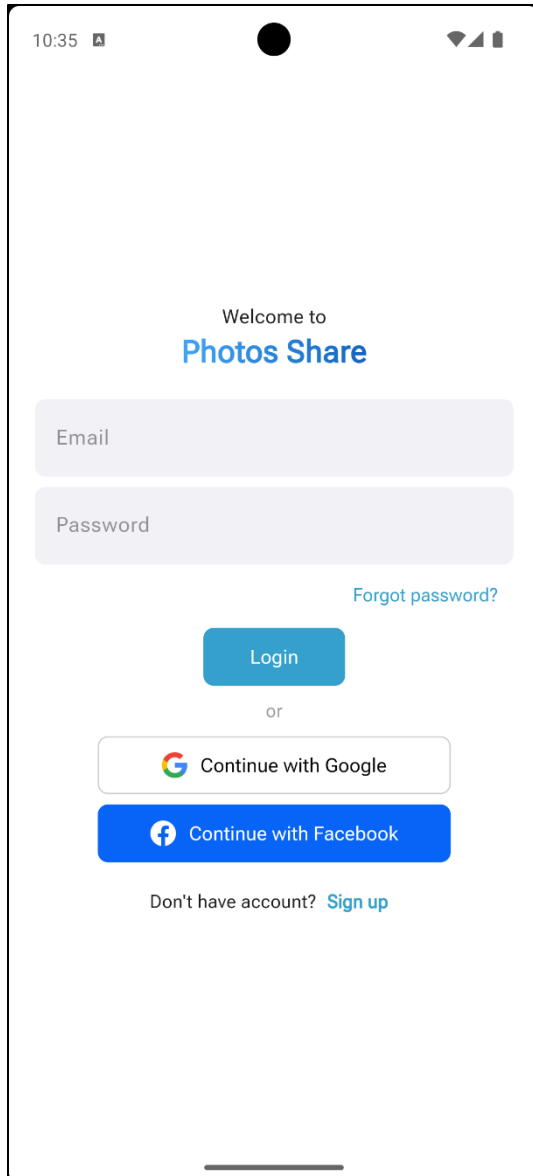
Đây là một ứng dụng chia sẻ hình ảnh lấy cảm hứng từ ứng dụng Instagram của Meta. Cho phép người dùng đăng nhập/đăng ký tài khoản và bắt đầu upload hình ảnh của bản thân chia sẻ với mọi người trên Internet. Khi đó, những người khác sẽ có thể xem và bình luận ở bài đăng của bạn. Mọi cập nhật đều là realtime nên mọi người có thể theo dõi nội dung một cách nhanh chóng.

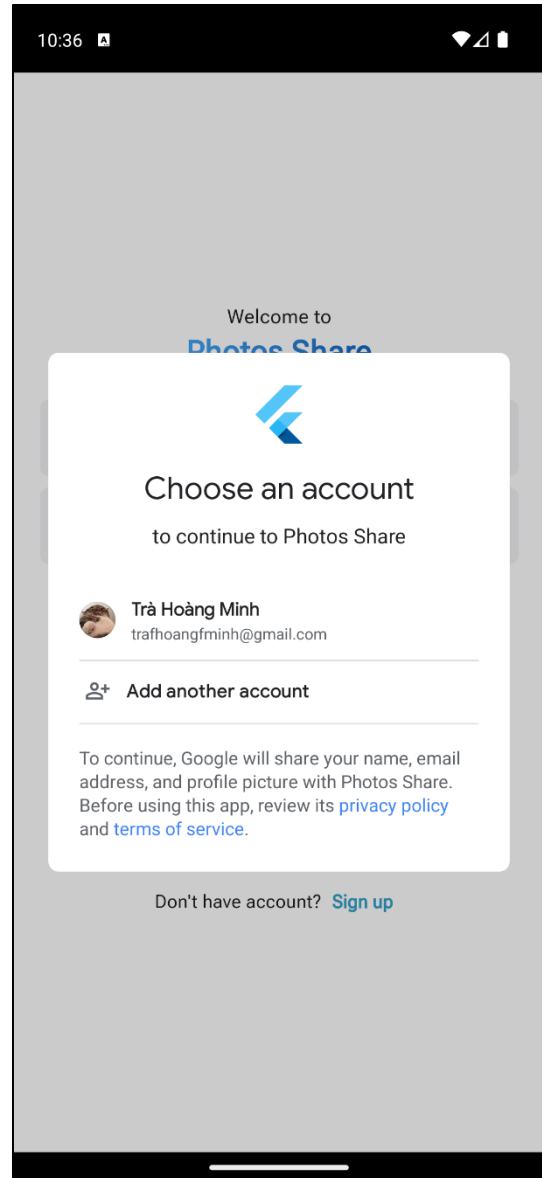
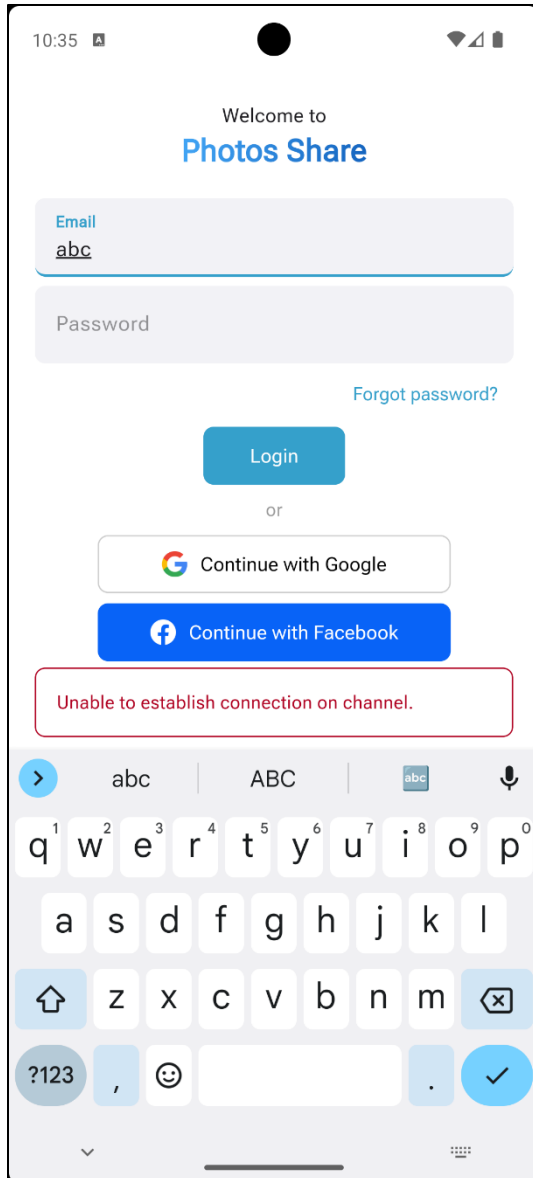
## II. Chi tiết các chức năng

### 1. Chức năng/giao diện 1: Trang đăng nhập

**Miêu tả chức năng/giao diện:** Cho phép người dùng đã có tài khoản đăng nhập bằng email và mật khẩu, hoặc có thể tiếp tục với Google hoặc Facebook (tự động tạo tài khoản mới nếu tài khoản chưa có trong cơ sở dữ liệu). Nếu có lỗi xảy ra, đẩy thông báo popup cho người dùng. Có thể từ trang này di chuyển đến trang đăng ký và trang khôi phục mật khẩu.

**Ảnh chức năng/giao diện:**





## Chi tiết cài đặt:

+ Các widget được sử dụng gồm:

- `AnnotatedRegion<SystemUiOverlayStyle>`: Ôm lấy toàn bộ ứng dụng để điều chỉnh màu sắc của thanh trạng thái và thanh điều hướng bằng thuộc tính value nhận vào một `SystemUiOverlayStyle`.
- `Opacity`: ôm lấy toàn bộ giao diện để điều chỉnh độ mờ của giao diện khi đang đăng nhập bằng Google/Facebook (để tránh popup đăng nhập của Google cùng màu trắng với giao diện gây khó nhìn).
- Widget tự custom `GradientText`: Kết hợp `ShaderMask` và `Text` để tạo hiệu ứng gradient cho chữ, với đầu vào là `String` text và `Gradient` gradient.

- Widget tự custom CustomTextField: Là widget TextField với nhiều thông số được thiết lập sẵn để có giao diện phù hợp với ứng dụng.
- Widget tự custom CustomButton: Là widget TextButton với các thông số được thiết lập sẵn, có thể chuyển đổi child từ dạng Text sang CircularProgressIndicator (loading).
- TextButton: dùng cho **Sign up** và **Forgot password?** giúp di chuyển qua lại giữa các trang.
- Widget tự custom OAuthButton: Là TextButton với child là SizedBox ôm lấy Row chứa AssetImage là logo của Google/Facebook và Text **Continue with Google/Facebook**. Có trạng thái loading khi đang đăng nhập.

+ Plugin/thư viện được sử dụng:

- Bộ Firebase SDKs để sử dụng các dịch vụ của Firebase (Firebase core, FirebaseCLI).
- Plugin firebase\_auth để điều khiển việc đăng nhập/đăng ký, quản lý trạng thái đăng nhập trong xuyên suốt ứng dụng. *(dùng cho toàn bộ ứng dụng)*
- Plugin google\_sign\_in để hỗ trợ đăng nhập bằng Google.
- Plugin flutter\_facebook\_auth để hỗ trợ đăng nhập bằng Facebook.
- Plugin google\_fonts để sử dụng font có trên Google Fonts. *(dùng cho toàn bộ ứng dụng)*
- Plugin flex\_color\_scheme để áp dụng các colorScheme có sẵn. *(dùng cho toàn bộ ứng dụng)*
- Plugin another\_flushbar để hiện thông báo popup tới người dùng.

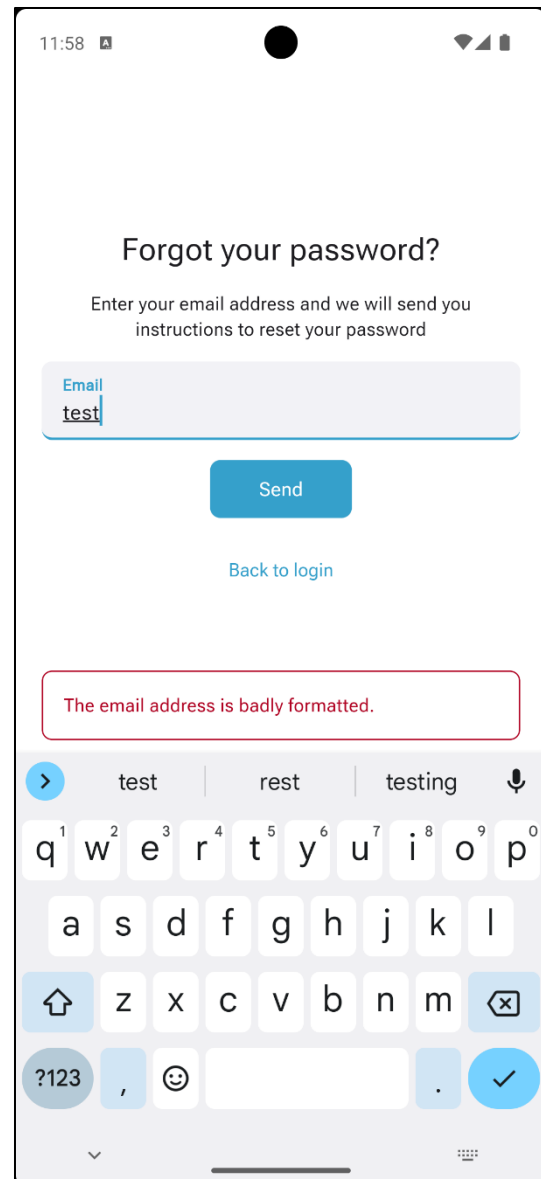
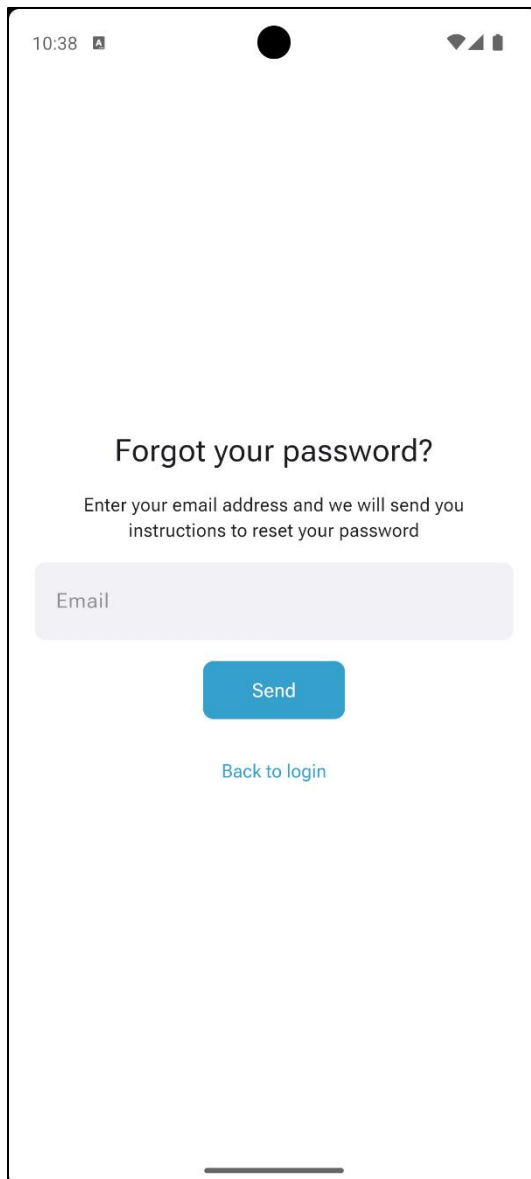
+ Sử dụng StreamBuilder để theo dõi sự thay đổi của trạng thái đăng nhập của người dùng với stream **authStateChange** được hỗ trợ bởi Firebase Authentication.

+ Chức năng thực hiện lấy dữ liệu người dùng nhập và gửi lên trên Firebase để xử lý và trả về kết quả đăng nhập thích hợp.

## 2. Chức năng/giao diện 2: Trang khôi phục mật khẩu

**Miêu tả chức năng/giao diện:** Cho phép người dùng đặt lại mật khẩu bằng cách nhập email đã đăng ký. Firebase sẽ gửi một email đến mail người dùng, kèm theo đó là đường link để người dùng đặt lại mật khẩu.

**Ảnh chức năng/giao diện:**



### Chi tiết cài đặt:

Quá trình cài đặt tương tự như Trang đăng nhập.

Sử dụng phương thức **sendPasswordResetEmail** có sẵn trong FirebaseAuth để gửi email thay đổi mật khẩu tới email đã đăng ký.

### 3. Chức năng/giao diện 3: Trang đăng ký

**Miêu tả chức năng/giao diện:** Cho phép người dùng tạo tài khoản mới bằng cách nhập một số trường thông tin. Hoặc đăng ký bằng Google/Facebook tương tự như trang đăng nhập.

**Ảnh chức năng/giao diện:**

10:38

Welcome to  
**Photos Share**

Name

Email

Password

Confirm password

Sign up

or

Continue with Google

Continue with Facebook

Already have an account? [Sign in](#)

10:39

Welcome to  
**Photos Share**

Name  
Minh

Email  
trafhoangminh@gmail.com

Password  
.....

Confirm password  
.....

or

Continue with Facebook

Account created successfully

Continue with Facebook

### Chi tiết cài đặt:

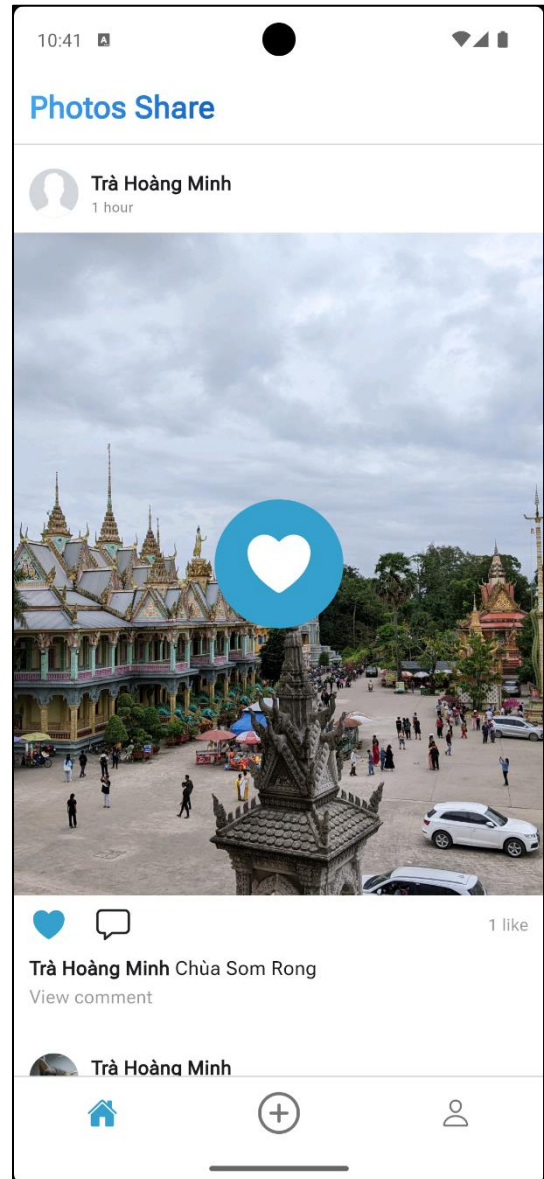
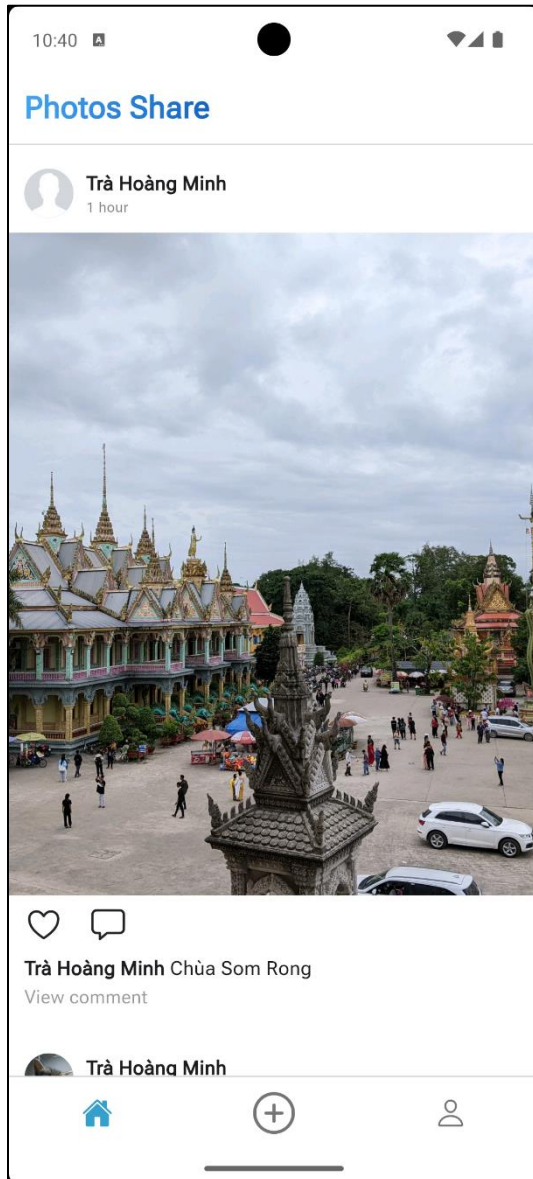
Quá trình cài đặt tương tự như Trang đăng nhập và Trang khôi phục mật khẩu.

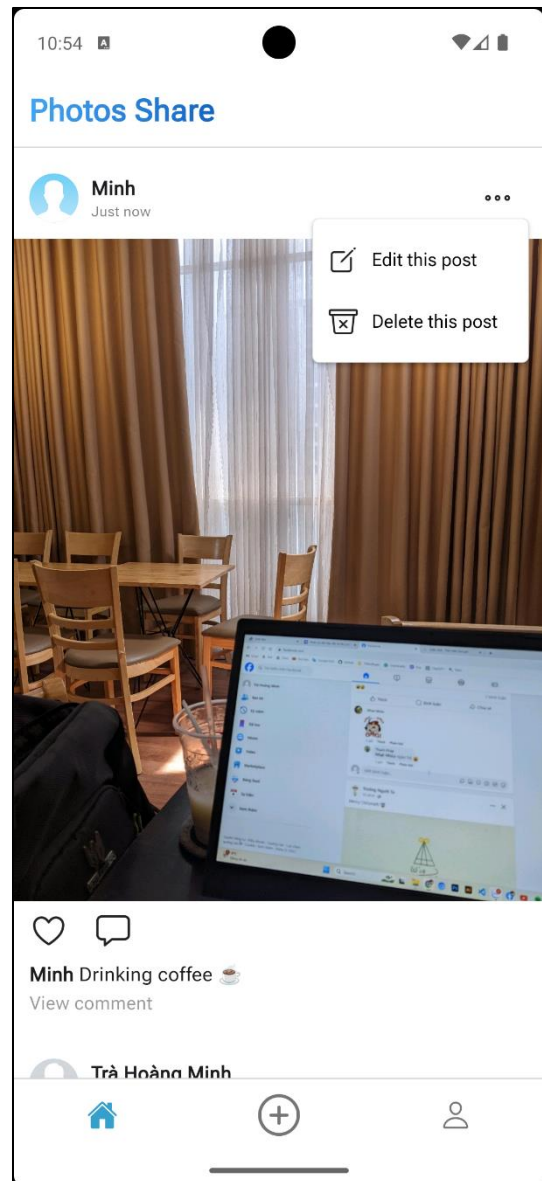
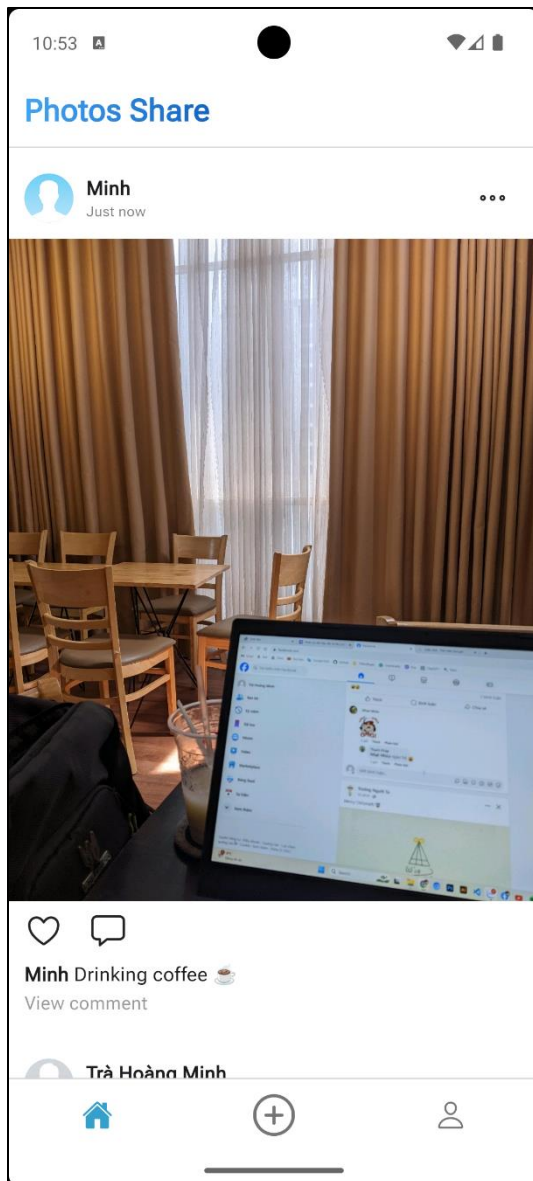
### 4. Chức năng/giao diện 4: Giao diện chính – Tab Home

**Miêu tả chức năng/giao diện:** Hiển thị các bài post mà những người dùng đã chia sẻ. Mỗi bài đăng sẽ bao gồm tên, avatar của người đăng, thời gian đã đăng, hình ảnh, nút yêu thích, số lượt thích, số bình luận và nút bình luận để truy cập vào **giao diện bình luận**. Người dùng có thể tap 2 lần vào hình hoặc nhấn trái tim để yêu thích bài đăng, tap 1 lần để truy cập vào **giao diện Photo Viewer**. Người dùng cũng có thể tap vào tên hoặc avatar trên bài đăng để truy cập vào **trang Cá nhân** của người đó. Đối với người dùng là chủ nhân của

bài đăng sẽ có thêm nút 3 chấm ở góc trên bên phải của bài đăng để mở ra popup menu cho phép **xóa** hoặc **chỉnh sửa bài đăng**.

**Ảnh chức năng/giao diện:**





### Chi tiết cài đặt:

+ Các widget được sử dụng gồm: (và các widget đã liệt kê ở những giao diện trước)

- Widget tự custom UserAvatarAndName: Kết hợp nhiều widget trong đó có CircleAvatar để hiển thị tên, avatar, thời gian đăng (đối với tab Home) và hiển thị tên, avatar, tiểu sử (đối với tab Cá nhân).
- Stack, Positioned, AspectRatio để hiển thị hình ảnh, kết hợp với GestureDetector, AnimationScale để làm hiệu ứng khi tap 2 lần thả tim bài viết.
- Widget TabView kết hợp với BottomNavigationBar của Scaffold để làm giao diện tab cho phép chuyển qua lại tab Home và **tab Cá nhân** của người dùng hiện tại bằng thao tác trượt ngang hoặc thao tác trên thanh điều hướng phía dưới.



- Widget PopupMenuButton để tạo menu popup.
- Widget ListView để hiển thị các bài post có thể cuộn được.

+ Các plugin/thư viện được sử dụng gồm:

- Plugin ionicons dùng để sử dụng các icon phù hợp hơn với giao diện.
- Plugin cloud\_firestore và firebase\_storage để viết các service phục vụ việc load hình và upload hình từ Firebase storage, cập nhật thông tin từ Firestore.
- Plugin cached\_network\_image để hiển thị hình từ internet và kết hợp lưu lại cache để có thể load lại nhanh hơn khi cần hiển thị hình ảnh cũ.

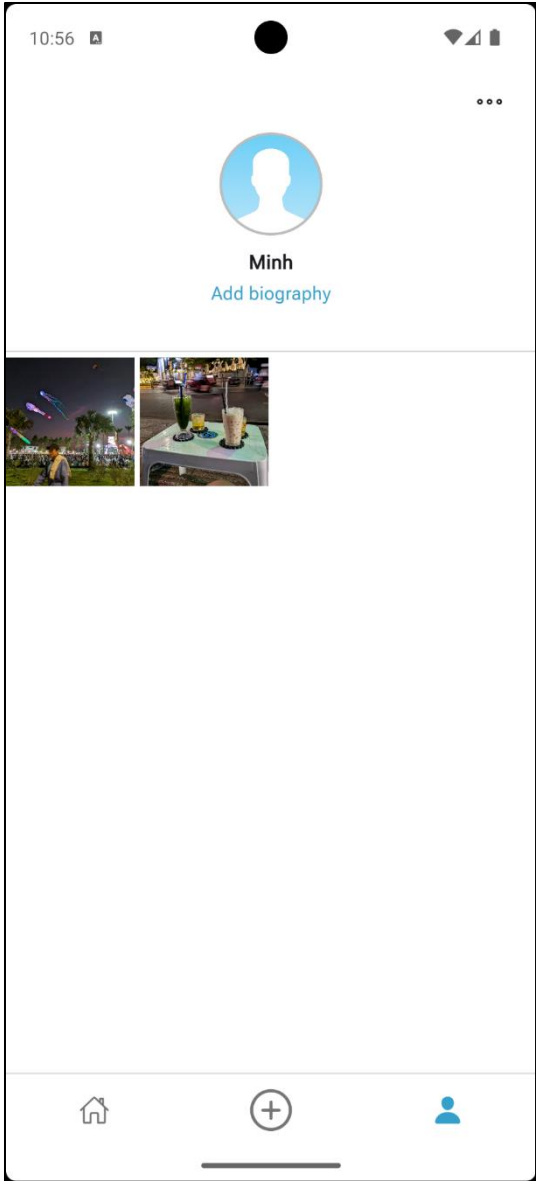
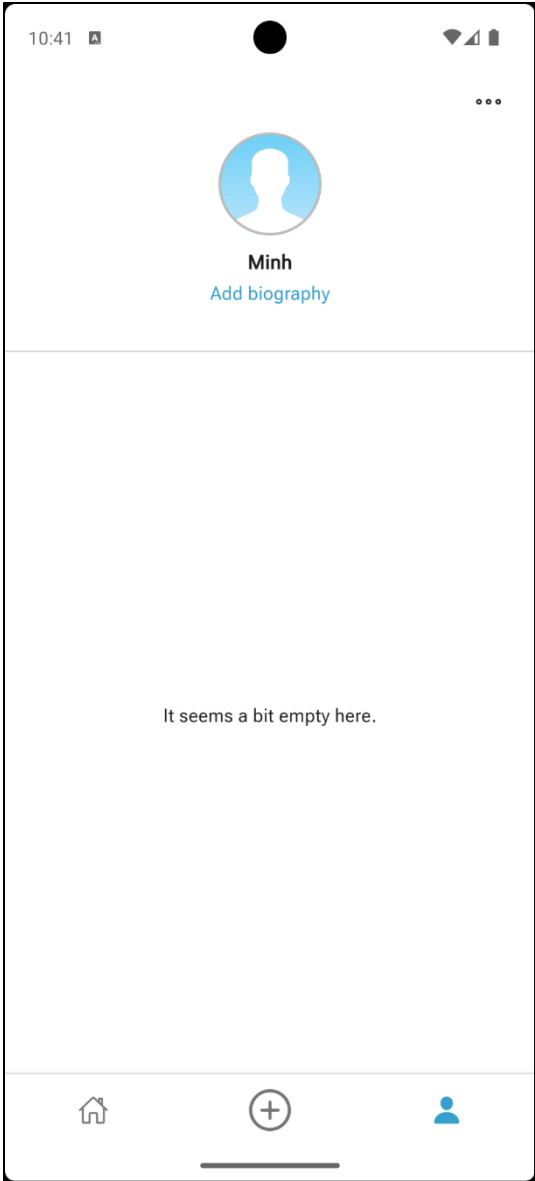
+ Việc hiển thị cũng như theo dõi sự thay đổi của các bài post được tiến hành thông qua StreamBuilder với stream được cung cấp từ plugin của Firebase storage.

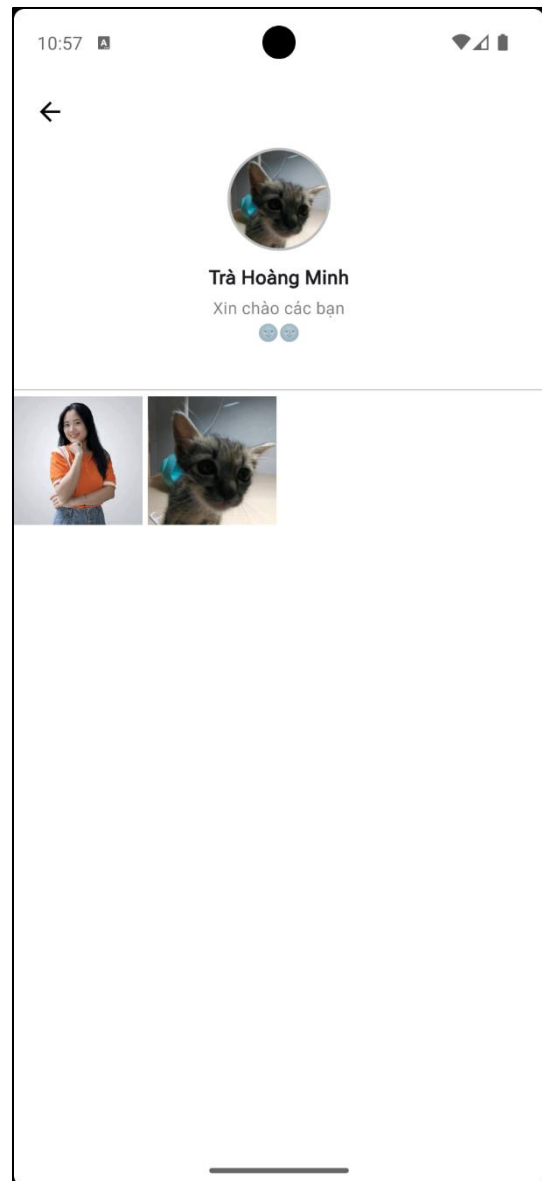
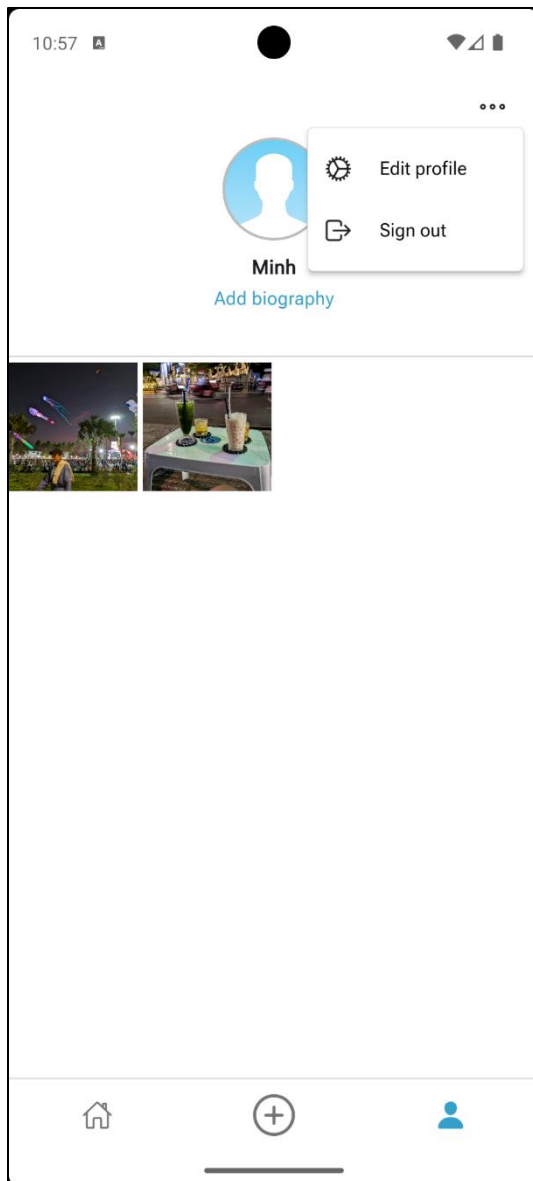
+ Thông tin về bài viết được lấy về dưới dạng JSON từ dịch vụ database đám mây Firestore.

## **5. Chức năng/giao diện 5: Giao diện chính – Tab Cá nhân và Trang Cá nhân**

**Miêu tả chức năng/giao diện:** Hiển thị các thông tin cá nhân của người dùng như avatar, tên, tiểu sử và danh sách các tấm hình đã chia sẻ dưới dạng lưới. Người dùng có thể tap vào từng hình để xem bài đăng đầy đủ. Nếu đang ở trang của bản thân sẽ có thêm nút 3 chấm popup menu cho phép người dùng đăng xuất hoặc truy cập vào trang **Chỉnh sửa thông tin cá nhân**.

**Ảnh chức năng/giao diện:**





### Chi tiết cài đặt:

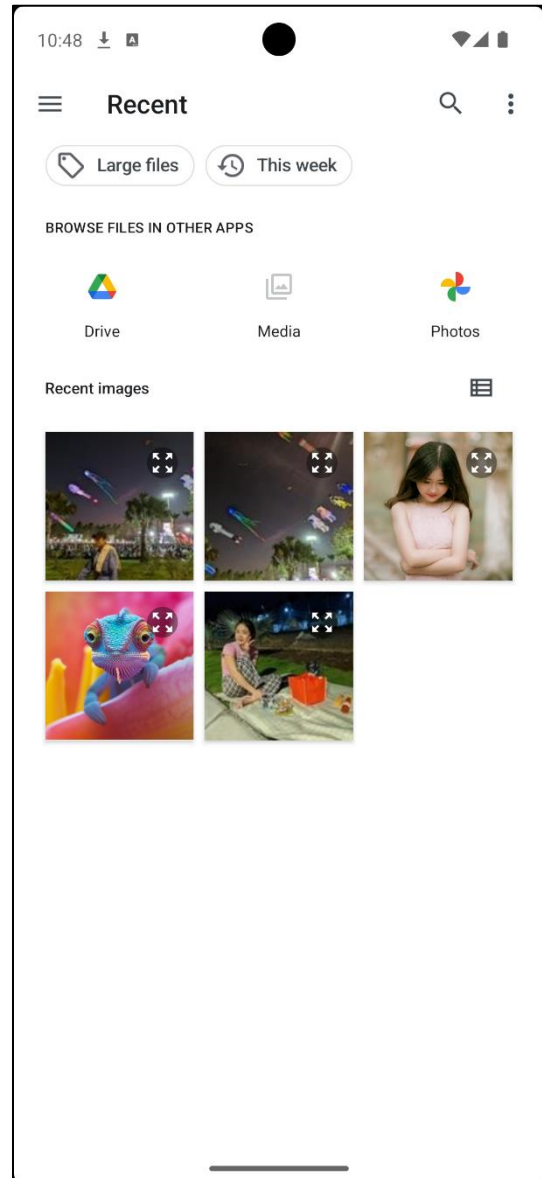
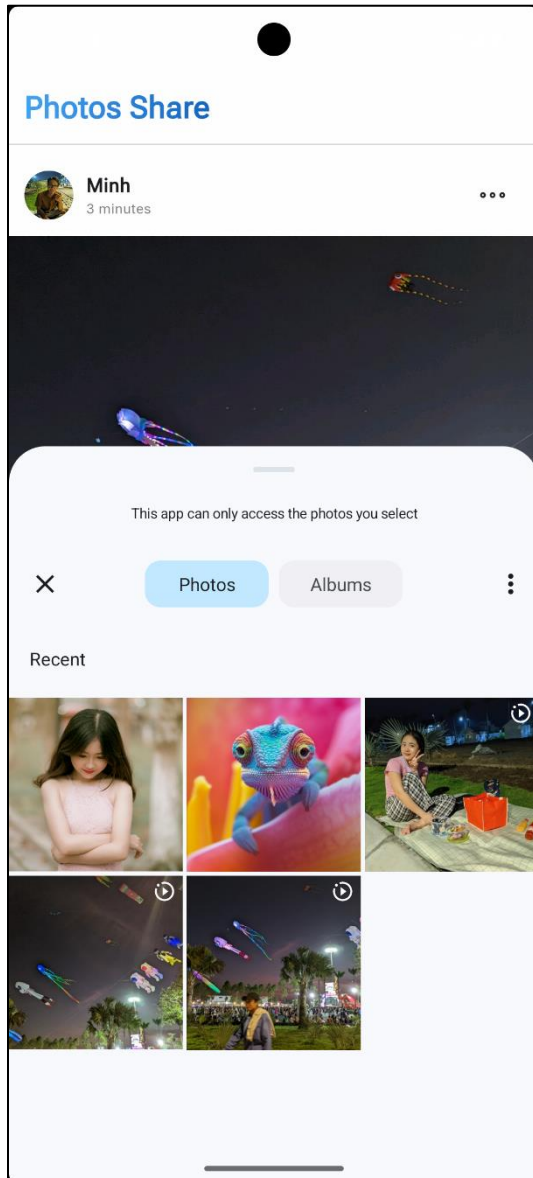
Các widget cũng như cách thức hoạt động tương tự như giao diện tab Home. Ngoài ra có sử dụng thêm widget GridView để hiển thị hình ảnh bài post dưới dạng lưới và Divider để vẽ đường kẻ ngang.

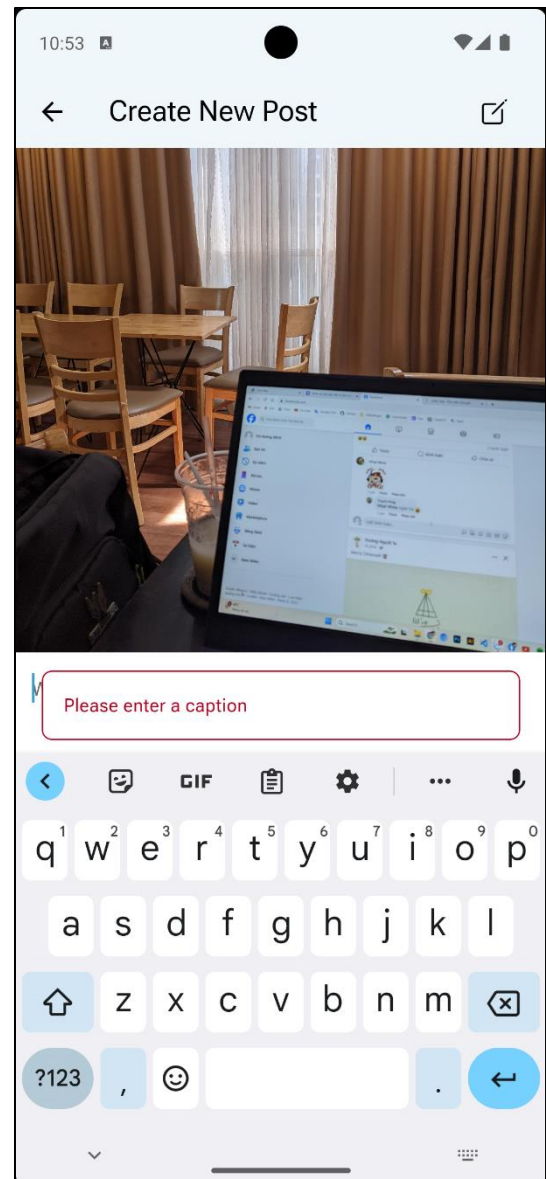
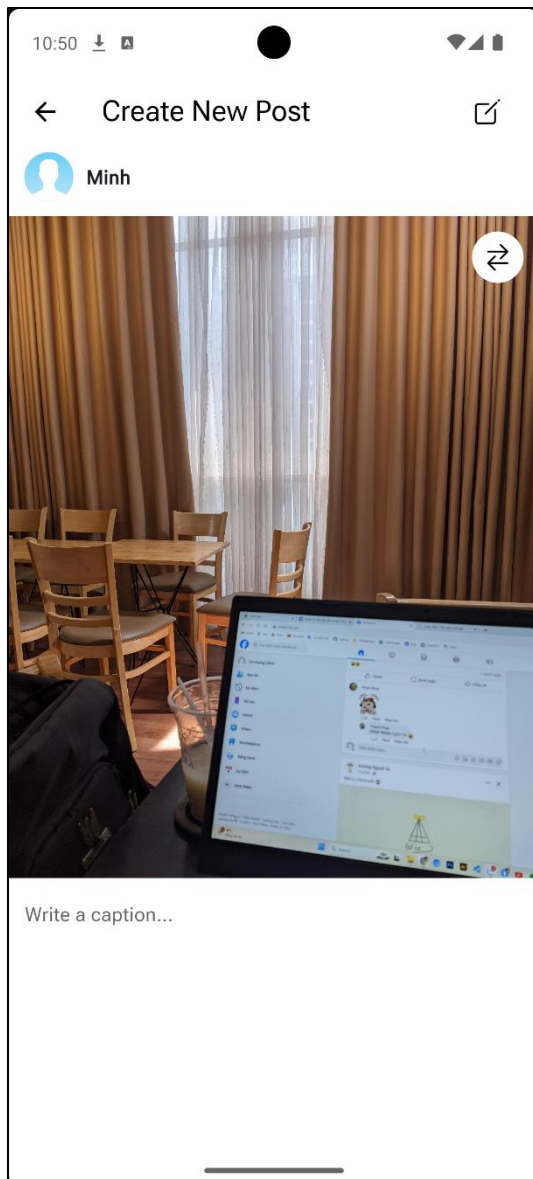
### 6. Chức năng/giao diện 6: Giao diện trang Đăng/Chỉnh sửa bài viết

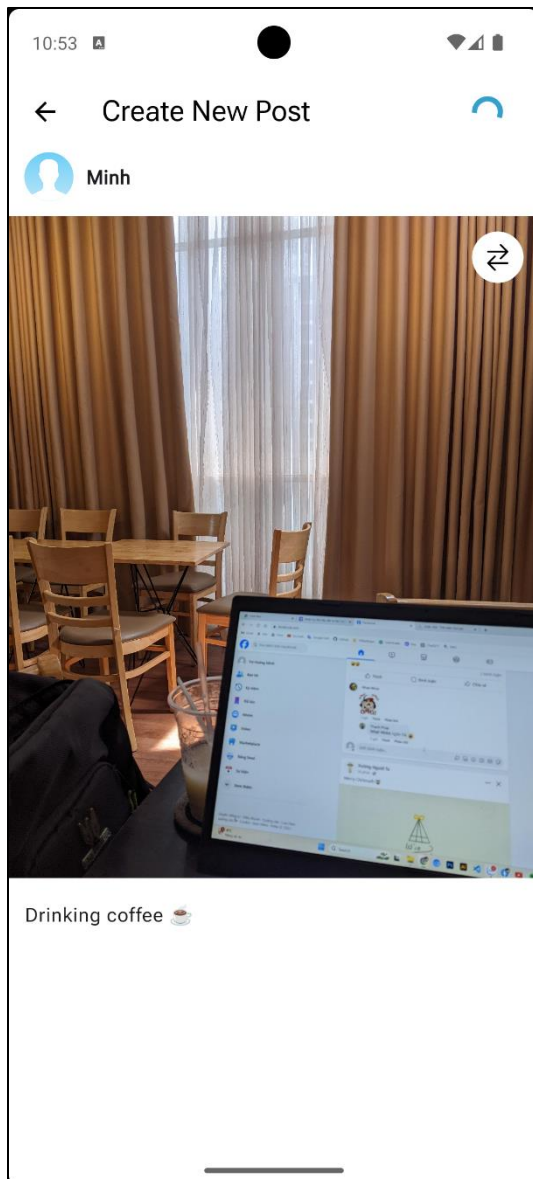
**Miêu tả chức năng/giao diện:** Người dùng có thể đăng bài viết mới bằng cách tap vào icon dấu cộng ở trên thanh điều hướng bên dưới, sau đó chọn một hình ảnh mà mình muốn chia sẻ, sau đó viết caption cho bài đăng. Người dùng cũng có thể chỉnh sửa bài viết đã đăng từ popup menu như đã được đề cập ở trên. Hình ảnh trước khi được upload lên server Firebase storage sẽ được nén lại ở độ phân giải tối đa  $1200 \times 1500$  bằng plugin

flutter\_image\_compress để tăng tốc độ upload cũng như giảm dung lượng lưu trữ trên cloud. Hình ảnh cũ sau khi đã được thay thế sẽ được xóa khỏi cloud.

### Ảnh chức năng/giao diện:







### Chi tiết cài đặt:

+ Các widget được sử dụng:

- Widget UserAvatarAndName như đã đề cập ở phần trước.
- Stack, Positioned, IconButton, Container, CachedNetworkImage (plugin cached\_network\_image) để hiển thị hình và nút thay đổi hình chồng lên nhau.
- TextField dùng cho caption của bài viết.

+ Plugin/thư viện được sử dụng:

- Plugin image\_picker dùng để chọn hình ảnh có trong máy.
- Plugin cached\_network\_image tương tự như các giao diện khác.

- Plugin flutter\_image\_compress dùng để nén hình ảnh về độ phân giải và kích thước nhỏ hơn trước khi upload lên cloud storage.
- Plugin firebase\_storage để viết service hỗ trợ upload file và xóa file cũ trên Firebase storage.

+ Thông tin bài đăng sẽ được lưu dưới dạng JSON và cập nhật lên Firestore. Các thông tin được lưu trữ bao gồm:

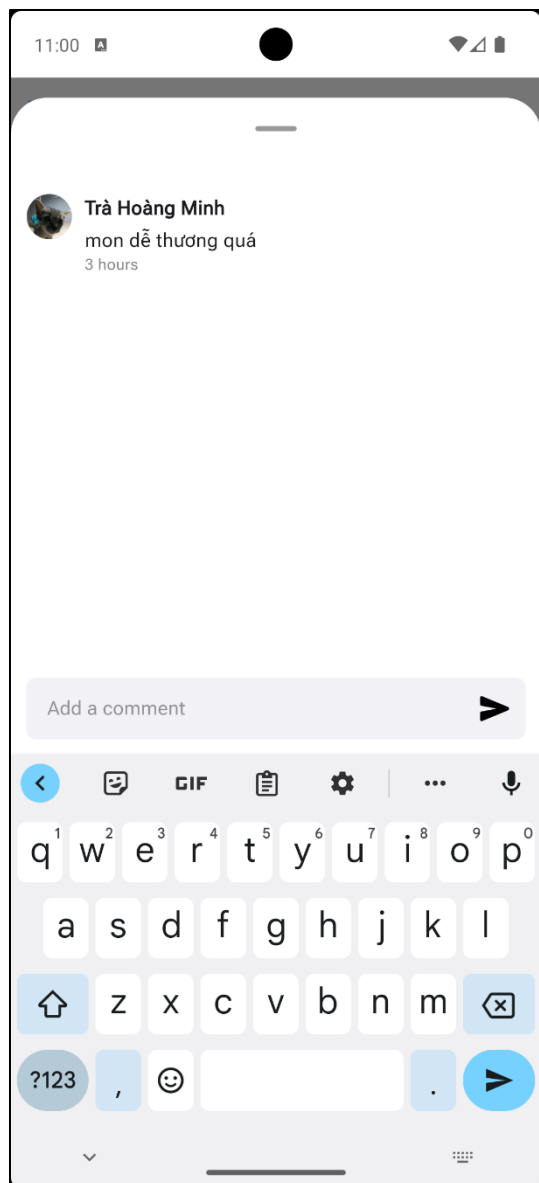
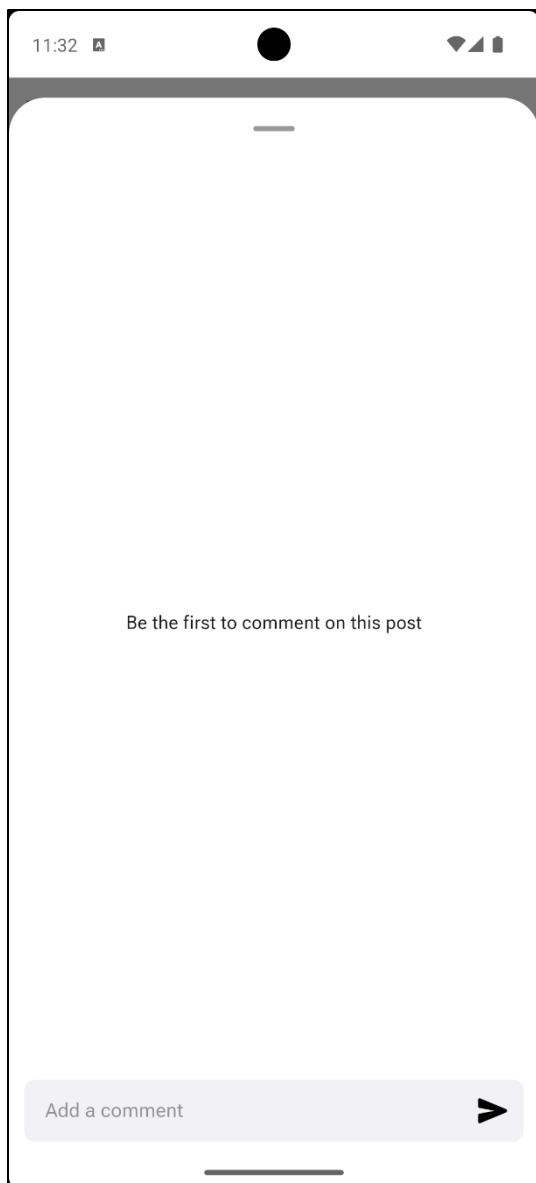
- ID của người dùng hiện tại (chủ nhân bài đăng).
- Caption bài đăng.
- Đường dẫn đến ảnh.
- Tham chiếu đến ảnh trên Firebase storage (phục vụ cho việc xóa ảnh về sau).
- Thời gian đăng.
- Danh sách ID của người dùng đã yêu thích bài viết.

+ Việc cập nhật trạng thái giữa các trang tương tự như các giao diện trên.

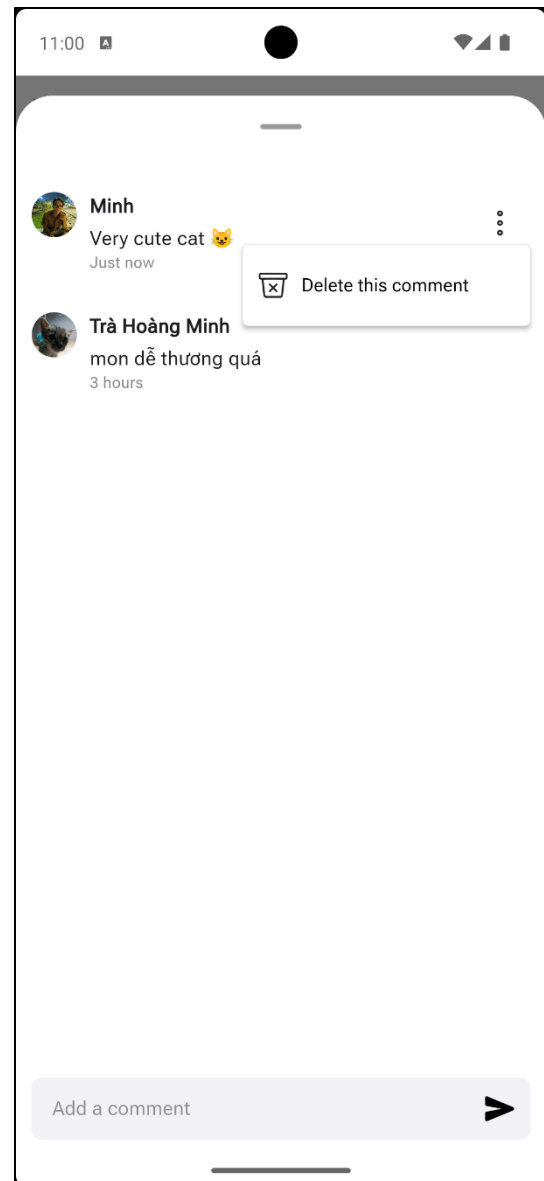
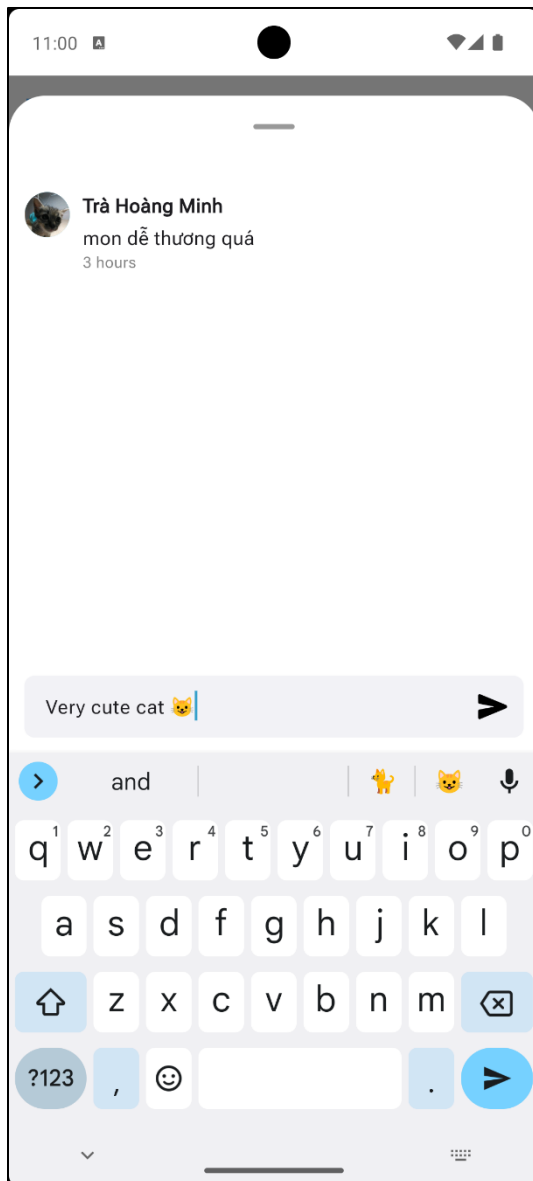
## **7. Chức năng/giao diện 7: Chức năng Bình luận**

**Miêu tả chức năng/giao diện:** Khi người dùng bấm vào nút bình luận hoặc số lượng bình luận của một bài đăng sẽ hiển thị những bình luận của bài đăng đó. Ở giao diện này người dùng có thể để lại bình luận vào bài đăng. Và nếu người dùng là chủ nhân của bình luận thì sẽ có hiển thị thêm nút 3 chấm popup menu để người dùng có thể xóa bình luận đi.

**Ảnh chức năng/giao diện:**







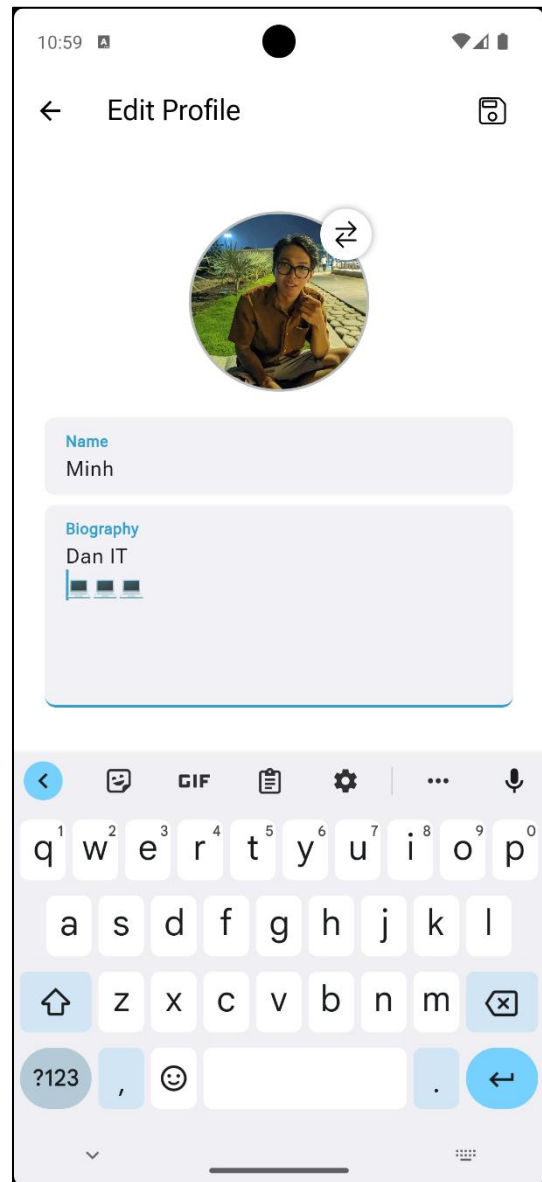
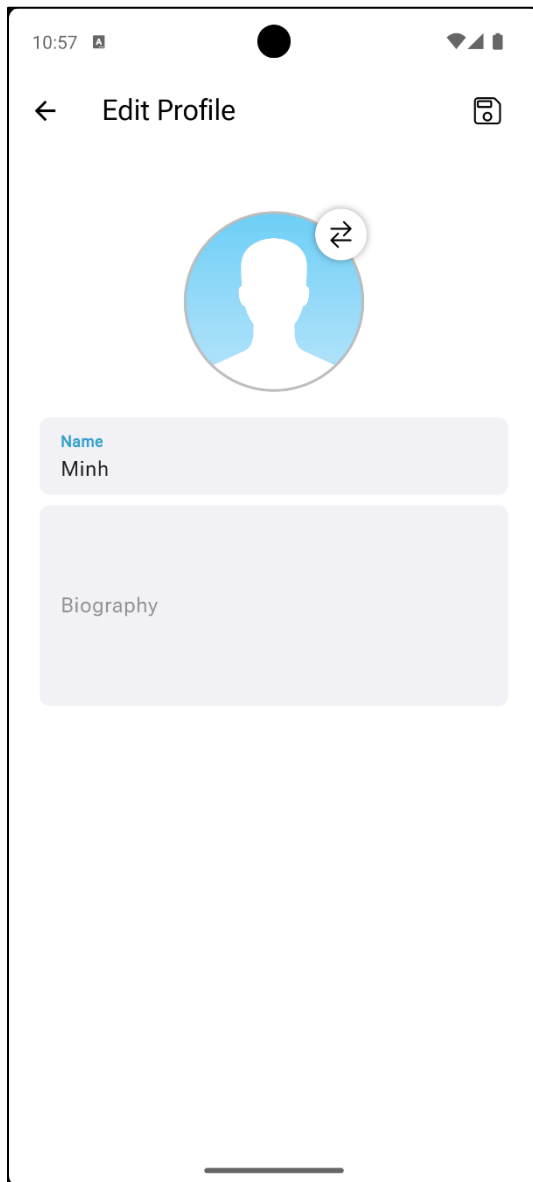
### Chi tiết cài đặt:

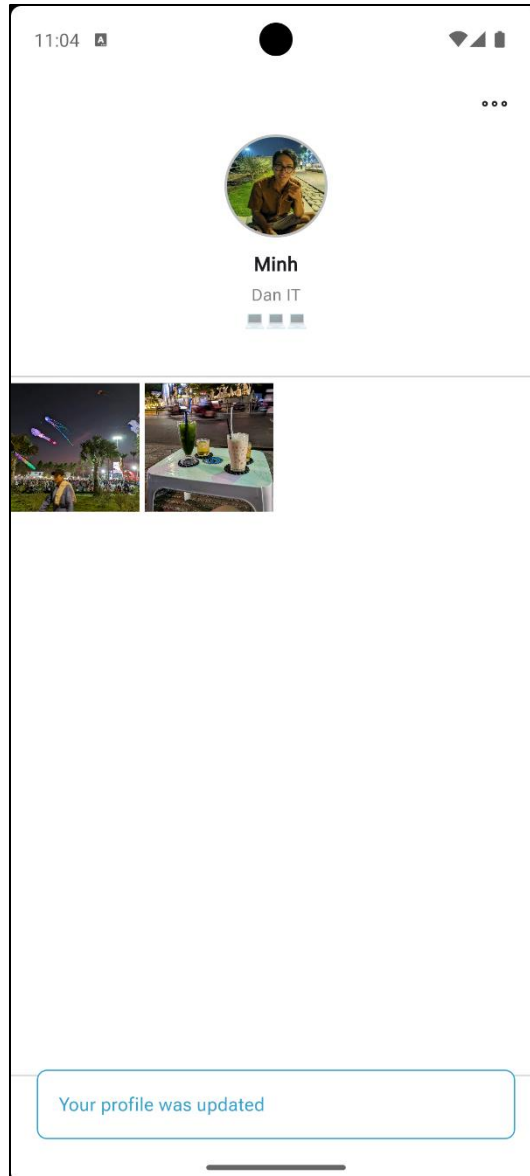
- + Các widget được sử dụng gồm các widget đã được sử dụng ở các giao diện khác như PopupMenuButton, TextField, Row, Column, CircleAvatar.
- + Để hiển thị bình luận dưới dạng nổi phía trên giao diện khác (modal) có sử dụng function showModalBottomSheet có trong flutter.
- + Việc tạo mới, đồng bộ dữ liệu ở trang này tương tự các trang trên, sẽ sử dụng Firestore để lưu trữ thông tin lên cloud database.

## 8. Chức năng/giao diện 8: Chức năng/giao diện cập nhật thông tin cá nhân

**Miêu tả chức năng/giao diện:** Cho phép người dùng cập nhật một số thông tin cá nhân của bản thân như ảnh đại diện, tên và tiểu sử của bản thân (phần giới thiệu nhỏ phía dưới tên người dùng). Ảnh đại diện trước khi upload lên cloud sẽ được flutter\_image\_compress nén lại thành kích thước tối đa  $500 \times 500$  để giảm dung lượng.

**Ảnh chức năng/giao diện:**





### Chi tiết cài đặt:

+ Một số widget được sử dụng gồm:

- Tận dụng custom widget CustomTextField.
- CircleAvatar, Stack, Positioned với tác dụng như các giao diện trên.

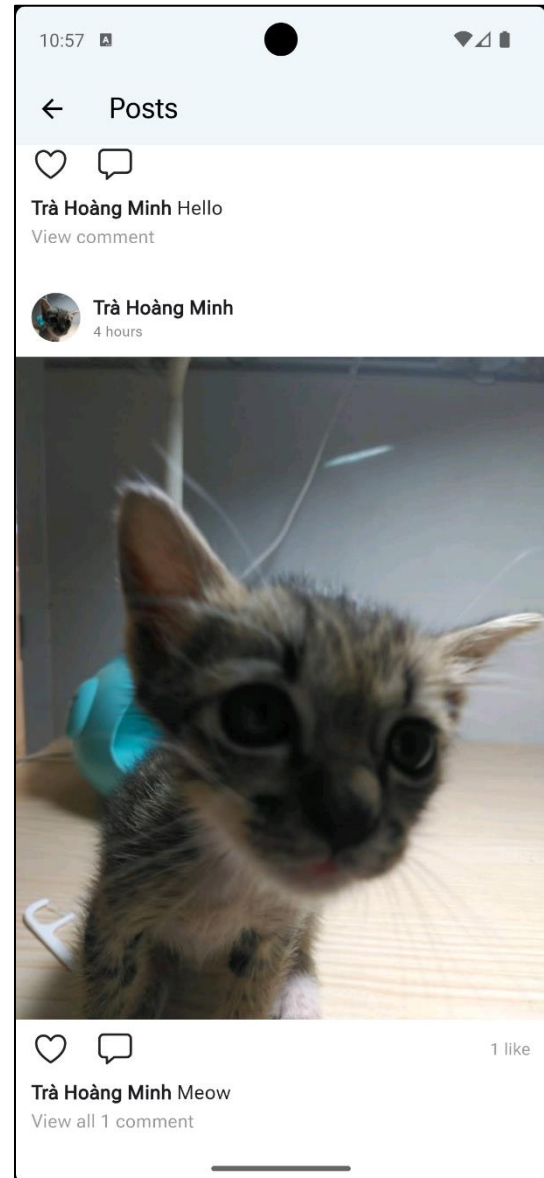
+ Các chức năng, cách lưu trữ dữ liệu được cài đặt tương tự như chức năng đăng bài.

### 9. Chức năng/giao diện 9: Giao diện Bài post cá nhân

**Miêu tả chức năng/giao diện:** Khi người dùng tap vào ảnh trên Trang Cá nhân, sẽ được dẫn đến trang bài post cá nhân và tự động scroll đến bài đăng đã tap vào. Thông tin hiển

thị và các chức năng ở trang này tương tự như trang Giao diện chính – Tab Home, nhưng chỉ hiển thị bài đăng của một cá nhân.

### Ảnh chức năng/giao diện:



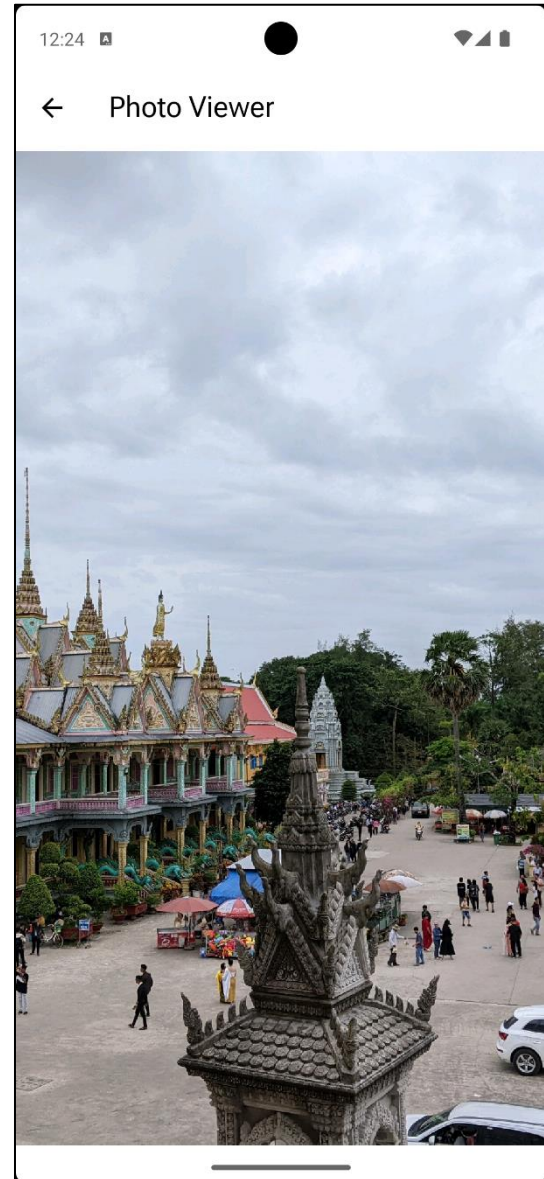
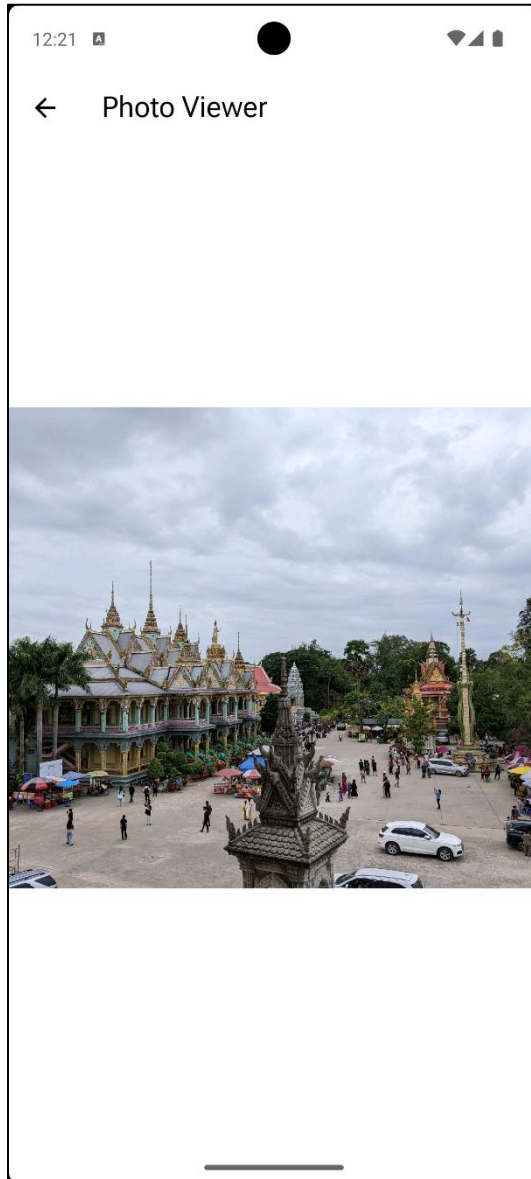
### Chi tiết cài đặt:

Quá trình cài đặt tương tự như Giao diện chính – Tab Home. Tuy nhiên, thay vì sử dụng ListView, ScrollablePositionedList (có trong plugin scrollable\_positioned\_list) được sử dụng để thay thế nhằm hỗ trợ cài đặt chức năng scroll tới bài đăng mà người dùng đã tap vào một cách dễ dàng hơn.

## 10. Chức năng/giao diện 10: Giao diện Photo Viewer

**Miêu tả chức năng/giao diện:** Khi người dùng tap 1 lần vào hình trong bài post, người dùng sẽ được chuyển đến giao diện Photo Viewer. Ở đây người dùng có thể thu phóng ảnh để theo dõi ảnh một cách đầy đủ hơn.

**Ảnh chức năng/giao diện:**



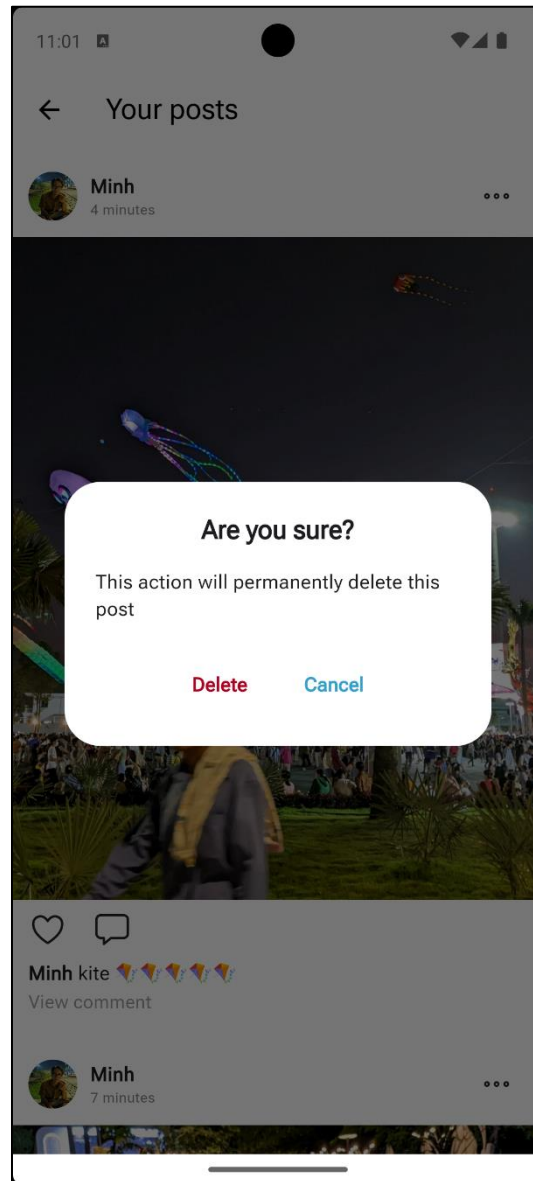
**Chi tiết cài đặt:**

Sử dụng plugin `photo_view` để hỗ trợ thu phóng ảnh. Tuy cũng có thể sử dụng widget `InteractiveView` có sẵn trong flutter nhưng để đơn giản hóa việc cài đặt nên plugin `photo_view` đã được sử dụng ở phần này.

## 11. Chức năng/giao diện 11: Giao diện Xác nhận xóa bài đăng

**Miêu tả chức năng/giao diện:** Xác nhận việc người dùng có chắc chắn khi muốn xóa bài đăng không. Người dùng có thể hủy hoặc xác nhận xóa để tiến hành quá trình xóa bài. Thông tin về bài đăng và ảnh sẽ được xóa khỏi cơ sở dữ liệu trên Firebase.

**Ảnh chức năng/giao diện:**



**Chi tiết cài đặt:**

Dùng function showDialog kết hợp với widget AlertDialog, TextButton để hiển thị ô xác nhận. Khi người dùng xác nhận hàm xóa bài đăng sẽ được kích hoạt và tiến hành xóa thông tin bài đăng trên cloud. Firestore sẽ tự động đồng bộ dữ liệu và cập nhật lại thông tin ở các trang.