ĐẠI HỌC PHENIKAA

TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA

BÀI TẬP LỚN CUỐI KỲ

LẬP TRÌNH THIẾT BỊ DI ĐỘNG

MÔ PHỎNG GIAO DIỆN APP INSTAGRAM

Giảng viên hướng dẫn: ThS. Nguyễn Xuân Quế

Sinh viên thực hiện: Đỗ Thị Mỹ Hạnh (22010128)

Khoá: K16 – 2022 – 2026

Lớp tín chỉ: Đồ án liên ngành-1-1-25(N04)

Chương trình đào tạo: Công nghệ thông tin

Hà Nội, tháng 11 năm 2025

MỤC LỤC

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất đến **ThS. Nguyễn Xuân Quế**, người đã tận tình hướng dẫn, đồng hành và hỗ trợ em trong suốt quá trình thực hiện đồ án "Instagram UI Clone" - dự án mô phỏng giao diện người dùng của ứng dụng Instagram trên nền tảng di động.

Trong suốt thời gian triển khai dự án, thầy không chỉ truyền đạt cho em những kiến thức chuyên môn quý báu về phát triển ứng dụng di động với Flutter và Dart, mà còn giúp định hướng tư duy thiết kế UI/UX, khơi dậy tinh thần sáng tạo, sự tỉ mỉ và tinh thần trách nhiệm trong từng chi tiết giao diện. Những góp ý sâu sắc và đầy tâm huyết của thầy đã giúp em nhìn nhận và giải quyết các thách thức trong việc tái tạo một giao diện phức tạp, mượt mà như Instagram, từ đó từng bước cải tiến và hoàn thiện sản phẩm. Em vô cùng trân trọng sự kiên nhẫn, tận tâm và niềm tin mà thầy đã dành cho mình.

Dự án Instagram UI Clone không chỉ là kết quả của một quá trình ứng dụng kiến thức và kỹ năng lập trình di động, mà còn là hành trình tìm tòi, học hỏi quý giá. Qua đó, em đã có cơ hội hiểu sâu sắc hơn trong việc nắm vững cách xây dựng các thành phần UI phức tạp, quản lý trạng thái cục bộ và tối ưu hóa trải nghiệm người dùng trên thiết bị di động.

Mặc dù đã luôn cố gắng hết mình, nhưng với kiến thức và kinh nghiệm còn hạn chế, dự án của em chắc chắn vẫn không tránh khỏi những thiếu sót nhất định. Em rất mong thầy thông cảm và ghi nhận sự nỗ lực chân thành mà em đã dành cho dự án này.

Một lần nữa, em xin gửi đến thầy lời cảm ơn sâu sắc nhất. Kính chúc thầy luôn mạnh khỏe, hạnh phúc, và tiếp tục truyền cảm hứng đến nhiều thế hệ sinh viên trên con đường học thuật và sáng tạo.

I. PHẦN I – SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. Giới thiệu chung

1.1. Mục tiêu

Dự án Instagram UI Clone được xây dựng nhằm mô phỏng lại giao diện người dùng của ứng dụng Instagram, sử dụng ngôn ngữ Dart và framework Flutter. Đây là một dự án thực hành tập trung vào phát triển giao diện người dùng (UI/UX) trên nền tảng di động.

Mục tiêu chính của dự án là:

- Rèn luyện kỹ năng thiết kế giao diện Flutter chuyên sâu: Áp dụng các widget, layout, animation của Flutter để tái tạo một giao diện phức tạp và trực quan như Instagram.
- Hiểu luồng hoạt động của các màn hình trong ứng dụng mạng xã hội: Nắm bắt cách người dùng tương tác và di chuyển giữa các tính năng cơ bản như xem bài viết, story, reels, tìm kiếm và trang cá nhân.
- Tạo ra một ứng dụng có giao diện hoàn thiện, tương tác mượt mà: Đảm bảo trải nghiệm người dùng gần nhất với ứng dụng gốc, dù không có cơ sở dữ liệu hoặc chức năng backend thực tế.
- Thực hành quản lý trạng thái và dữ liệu giả lập: Sử dụng mock data để cung cấp nội dung cho giao diện, đồng thời tìm hiểu các phương pháp quản lý trạng thái cục bộ trong Flutter.

1.2. Phạm vi dự án

Dự án này tập trung vào việc phát triển một ứng dụng di động có phạm vi rõ ràng như sau:

- Nền tảng mục tiêu: Ứng dụng di động chạy được trên Android.
- Trọng tâm: Tập trung hoàn toàn vào việc hiển thị giao diện, thiết kế layout và luồng điều hướng giữa các màn hình. Mọi yếu tố hình ảnh và tương tác UI đều được ưu tiên cao nhất.
- Dữ liệu: Sử dụng mock data (dữ liệu giả lập) được khai báo trực tiếp trong mã nguồn cho tất cả các nội dung như bài viết, thông tin người dùng, story, và video reels. Điều này giúp tách biệt việc phát triển UI khỏi các ràng buộc của backend.
- Giới hạn chức năng:
    + Không có chức năng đăng nhập/đăng ký người dùng thật. Người dùng sẽ được coi là đã đăng nhập vào một tài khoản giả lập.
    + Không có khả năng lưu trữ dữ liệu lâu dài hoặc tương tác với cơ sở dữ liệu thực. Mọi thay đổi trên UI (ví dụ: thích bài viết, thêm bình luận) chỉ mang tính chất hiển thị tạm thời và không được lưu lại.

+ Không có khả năng gọi API bên ngoài để lấy dữ liệu động hoặc thực hiện các thao tác backend.

1.3. Đối tượng sử dụng

Dự án này hướng đến các đối tượng chính sau:

- Sinh viên, học viên học Flutter: Là một dự án thực hành lý tưởng để áp dụng kiến thức về Flutter vào một ứng dụng thực tế, phức tạp hơn các ví dụ cơ bản.
- Các nhà phát triển Flutter muốn tìm hiểu về cấu trúc UI/UX của Instagram: Có thể tham khảo mã nguồn để hiểu cách tái tạo các thành phần giao diện phức tạp.
- Người dùng muốn trải nghiệm giao diện Instagram mô phỏng: Cung cấp một ứng dụng có giao diện quen thuộc để người dùng có thể tương tác mà không cần tài khoản Instagram thật.

1.4. Công cụ và môi trường phát triển

Dự án sử dụng các công cụ và môi trường phát triển tiêu chuẩn cho một ứng dụng Flutter:

Thành phần	Phiên bản / Công nghệ	Mục đích
Ngôn ngữ	Dart	Ngôn ngữ lập trình chính cho Flutter.
Framework	Flutter 3.x (hoặc phiên bản mới nhất)	SDK phát triển UI đa nền tảng.
IDE	Visual Studio Code (VS Code)	Môi trường phát triển tích hợp, hỗ trợ Dart/Flutter.
Hệ điều hành	Windows 11	Nền tảng phát triển và chạy giả lập/thiết bị thật.
Thư viện phụ trợ	video_player	Để phát video trong các màn hình Reels.
Quản lý phiên bản	Git (GitHub)	Quản lý mã nguồn

2. Tổng quan hệ thống

2.1. Kiến trúc tổng thể

Ứng dụng được thiết kế với kiến trúc rõ ràng, sử dụng Bottom Navigation Bar làm trung tâm điều hướng giữa các màn hình chính. Toàn bộ các màn hình này được quản lý bởi một DashboardScreen (hoặc MainScreen) để đảm bảo luồng điều hướng mượt mà và quản lý trạng thái thanh điều hướng.

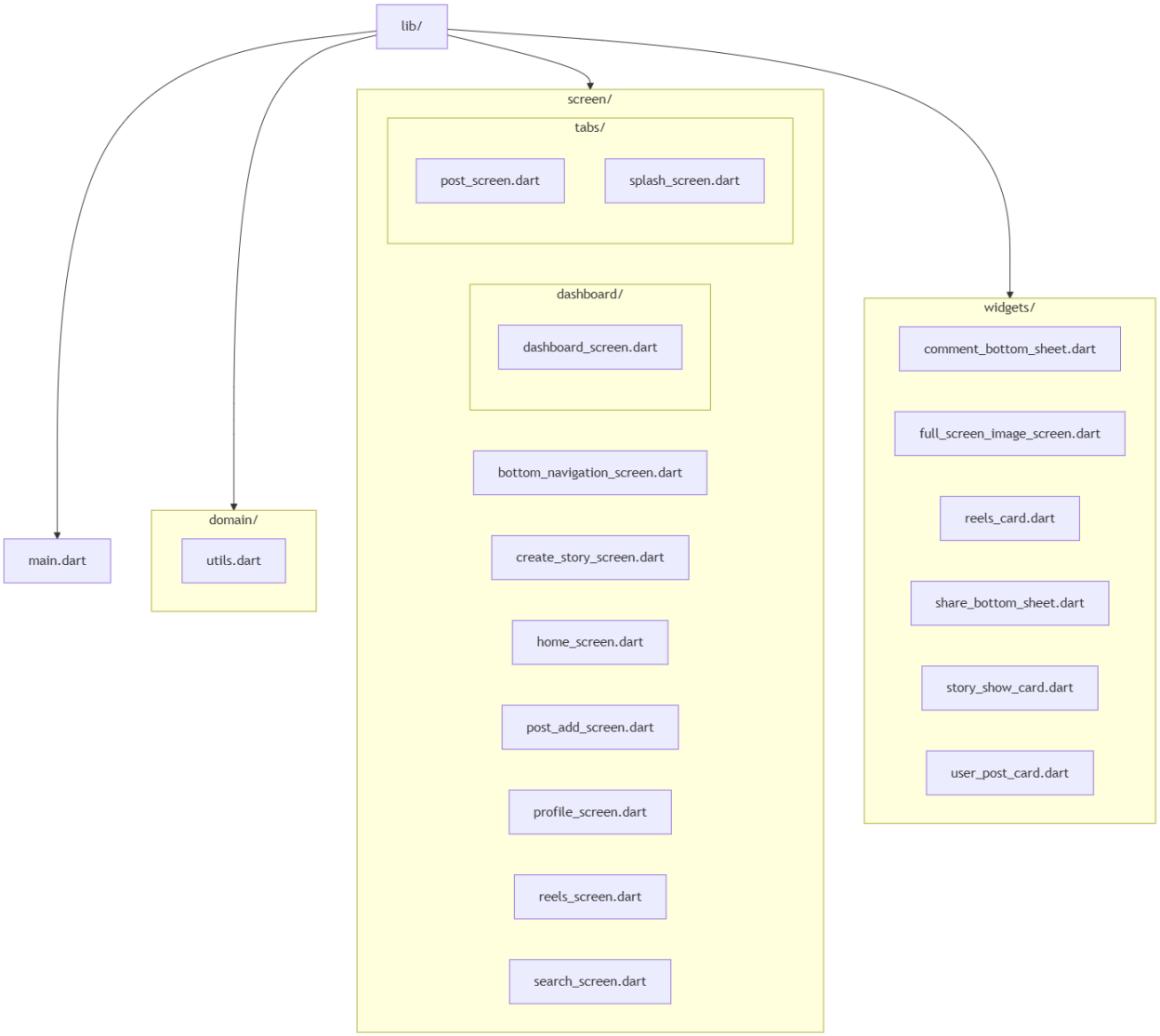
Ứng dụng bao gồm 5 màn hình chính, mỗi màn hình tương ứng với một biểu tượng trên thanh điều hướng dưới cùng:

1. Home Screen: Nơi hiển thị các bài viết và story từ những người dùng đang theo dõi (giả lập).

2. Search Screen: Hiển thị lưới các bài viết khám phá (explore grid) và thanh tìm kiếm (chức năng tìm kiếm giả lập).
3. Add Post Screen: Màn hình để người dùng "đăng tải" một bài viết mới (chức năng giả lập).
4. Reels Screen: Hiển thị các video ngắn dạng cuộn dọc.
5. Profile Screen: Trang cá nhân của người dùng hiện tại, hiển thị thông tin và các bài viết đã đăng.

Mỗi màn hình chính này lại có thể chứa hoặc gọi tới các widget con riêng biệt, được thiết kế để tái sử dụng và đảm bảo tính module hóa:

- **user_post_card.dart**: Widget này hiển thị một bài đăng hoàn chỉnh trên màn hình Home, bao gồm thông tin người đăng, hình ảnh/video, chú thích và các nút tương tác.
- **reels_card.dart**: Widget này dùng để hiển thị một video Reels với các nút tương tác và thông tin người dùng, được sử dụng trong màn hình Reels.
- **comment_bottom_sheet.dart**: Đây là một popup dạng sheet từ dưới lên, hiển thị danh sách bình luận cho một bài đăng.
- **story_show_card.dart**: Widget này hiển thị story toàn màn hình, cho phép người dùng xem và chuyển đổi giữa các story. Nó thường được gọi từ màn hình Home khi người dùng nhấn vào một story.
- **share_bottom_sheet.dart**: Một popup dạng sheet từ dưới lên, mô phỏng giao diện chia sẻ nội dung bài viết đến các ứng dụng khác.
- **full_screen_image_screen.dart**: Widget này có chức năng hiển thị một hình ảnh ở chế độ toàn màn hình, có thể được sử dụng để xem chi tiết ảnh từ bài đăng hoặc các phần khác của ứng dụng.
- Và nhiều widget UI nhỏ hơn có thể được định nghĩa trực tiếp trong các màn hình hoặc là một phần của các widget lớn(ví dụ: các nút like, comment, icon, text fields,…).



Hình 1.2.1. Sơ đồ cấu trúc thư mục

3. Các chức năng chính của hệ thống

Bảng dưới đây liệt kê các chức năng chính mà ứng dụng Instagram UI Clone sẽ mô phỏng, cùng với mô tả ngắn gọn và các file/màn hình liên quan.

STT	Chức năng	Mô tả	File / Màn hình
1	Trang chủ (Home)	Hiển thị danh sách các bài viết (post) của người dùng theo dõi và Story của họ ở đầu trang.	home_screen.dart, user_post_card.dart, story_circle_widget.dart
2	Tìm kiếm (Search)	Hiển thị một lưới các ảnh/video khám phá (explore grid) giả lập, và thanh tìm kiếm (UI mock-up).	search_screen.dart, explore_grid_item.dart

3	Tạo bài đăng (Add Post)	Mô phỏng quy trình chọn ảnh từ thư viện và thêm chú thích (caption) cho bài viết mới.	post_add_screen.dart
4	Reels	Hiển thị danh sách các video ngắn dạng cuộn dọc (swipe vertical), với các nút tương tác giả lập.	reels_screen.dart, reels_card.dart
5	Hồ sơ (Profile)	Hiển thị thông tin cá nhân của người dùng, avatar, bio, và các bài viết/reels đã đăng.	profile_screen.dart, profile_post_grid.dart
6	Story	Cho phép xem các Story của người dùng theo dạng toàn màn hình, có thể vuốt để chuyển đổi.	story_show_card.dart
7	Bình luận	Hiển thị danh sách bình luận giả lập cho một bài viết hoặc video Reels, dưới dạng Bottom Sheet.	comment_bottom_sheet.dart
8	Điều hướng chung	Quản lý chuyển đổi giữa 5 màn hình chính thông qua Bottom Navigation Bar.	dashboard_screen.dart, custom_bottom_nav_bar.dart
9	Chia sẻ bài viết	Mô phỏng giao diện chia sẻ bài viết qua các ứng dụng khác (dạng Bottom Sheet).	share_bottom_sheet.dart

4. Mô tả chi tiết từng chức năng

4.1. Home Screen

Màn hình Home là trung tâm của ứng dụng, nơi người dùng sẽ dành phần lớn thời gian để xem nội dung.

- Cấu trúc: Sử dụng ListView.builder hiệu quả để hiển thị một danh sách cuộn vô hạn các bài đăng. Phần đầu của màn hình sẽ chứa một ListView.builder theo chiều ngang để hiển thị Story.
- Thành phần chính:
    + Story Bar: Một hàng các avatar hình tròn ở đầu trang, mỗi avatar đại diện cho một Story. Khi nhấn vào, sẽ mở StoryShowCard.
    + Post Feed: Mỗi bài đăng là một widget UserPostCard riêng biệt, chứa:
        - Avatar và tên người đăng.
        - Hình ảnh hoặc video của bài đăng.
        - Các nút tương tác: Like, Comment, Share, Save.
        - Số lượt thích và chú thích (caption) của bài đăng.

        - Phần hiển thị bình luận gần đây nhất.

- Tương tác:
    + Nhấn vào avatar người đăng hoặc story circle sẽ mở StoryShowCard toàn màn hình.
    + Nhấn vào nút bình luận hoặc phần hiển thị bình luận sẽ mở CommentBottomSheet.
    + Nhấn vào nút chia sẻ sẽ mở ShareBottomSheet.

4.2. Search Screen

Màn hình Search cung cấp một trải nghiệm khám phá nội dung tương tự Instagram.

- Cấu trúc: Chủ yếu sử dụng GridView.count hoặc StaggeredGridView (nếu có) để hiển thị một lưới các ảnh/video giả lập, tạo cảm giác như một trang "Khám phá" thực thụ.
- Thành phần chính:
    + Search Bar: Một TextField được thiết kế để giống thanh tìm kiếm của Instagram, nhưng chức năng tìm kiếm thực tế chỉ là mock UI.
    + Explore Grid: Lưới các hình ảnh và video thu nhỏ, khi nhấn vào có thể dẫn đến một màn hình chi tiết bài đăng (chức năng này có thể được đơn giản hóa hoặc giả lập).
- Tương tác:
    + Người dùng có thể nhập văn bản vào thanh tìm kiếm, nhưng không có kết quả tìm kiếm thực tế.
    + Nhấn vào một ảnh trong lưới sẽ hiển thị hiệu ứng nhấn hoặc có thể điều hướng đến một màn hình chi tiết bài đăng đơn giản.

4.3. Add Post Screen

Màn hình này mô phỏng quy trình người dùng đăng một bài viết mới.

- Cấu trúc: Giao diện được thiết kế tối giản, tập trung vào khung ngắm camera (mô phỏng) và các điều khiển chụp/quay.
- Thành phần chính:
    + Header Bar: Chứa nút "Close" (để quay lại), và các nút điều khiển camera như "Flash" và "Settings" (chỉ là UI).
    + Side Controls: Một hàng dọc các biểu tượng (Text, Loop, Grid, Arrow Down) ở bên trái, mô phỏng các công cụ chỉnh sửa hoặc chế độ camera.
    + Capture Button Area: Phần dưới cùng của màn hình bao gồm:

    - Gallery Previews: Các hình tròn nhỏ hiển thị ảnh từ thư viện (mô phỏng) ở hai bên nút chụp.
    - Capture Button: Một nút lớn hình tròn ở giữa, mô phỏng nút chụp ảnh hoặc quay video.
    - Mode Selector: Các tùy chọn "POST", "STORY", "REEL" để chuyển đổi giữa các chế độ tạo nội dung (hiện tại "STORY" đang được chọn).
  + Rotate Camera Button: Một nút ở góc dưới bên phải để chuyển đổi giữa camera trước/sau (mô phỏng).

- Tương tác:
  + Người dùng có thể nhấn nút "Close" để thoát màn hình.
  + Các nút điều khiển camera (flash, settings, side controls, rotate camera) chỉ mang tính chất hiển thị, không có chức năng thực tế.
  + Nút chụp ảnh trung tâm và các ảnh preview thư viện cũng chỉ là UI mô phỏng.
  + Người dùng có thể "chọn" giữa các chế độ "POST", "STORY", "REEL" (thông qua _buildTextOption), nhưng hiện tại chỉ thay đổi trạng thái UI.

4.4. Reels Screen

Màn hình Reels mang đến trải nghiệm xem video ngắn theo chiều dọc.

- Cấu trúc: Màn hình ReelsScreen hiện tại chỉ đóng vai trò là container cho một ReelsCard. Widget ReelsCard là nơi chứa toàn bộ logic và giao diện để hiển thị và quản lý việc chuyển đổi giữa các video ngắn.
- Thành phần chính:
  + Reels Card (ReelsCard.dart): Đây là widget chính, được thiết kế để chiếm toàn bộ màn hình và tự quản lý danh sách các video. Mỗi ReelsCard chứa:
    - VideoPlayer để phát video (từ danh sách shortVideo giả lập được định nghĩa trong ReelsCard).
    - Thông tin người đăng
    - Caption của video
    - Một nút âm nhạc hoặc âm thanh
    - Các nút tương tác ở bên phải: Like, Comment, Share, Refresh và More.
    - Một khung hình nhỏ ở dưới cùng bên phải mô phỏng avatar người đăng nhạc hoặc một item đặc trưng.

- Tương tác:
    + Vuốt lên hoặc xuống: Người dùng có thể vuốt dọc trên video để chuyển đổi giữa các video ngắn khác nhau trong danh sách nội bộ của ReelsCard.
    + Nhấn vào video: Nhấn vào video sẽ tạm dừng hoặc tiếp tục phát video.

4.5. Profile Screen

Màn hình Profile hiển thị thông tin cá nhân của người dùng hiện tại.

- Cấu trúc: Chia làm các phần rõ ràng: header với thông tin người dùng, sau đó là các tab để hiển thị bài viết và Reels.
- Thành phần chính:
    + Profile Header:
        o Ảnh đại diện (avatar) lớn.
        o Tên người dùng (username), số lượng bài viết, người theo dõi, đang theo dõi (mock data).
        o Phần mô tả cá nhân (bio).
        o Nút "Edit Profile" (chỉ là UI).
    + Tab Bar: Chứa 2 tab chính:
        o Grid View: Hiển thị các bài đăng của người dùng dưới dạng lưới 3x3 (sử dụng GridView.builder).
        o Reels View: Hiển thị các video Reels của người dùng dưới dạng lưới.
- Tương tác:
    + Người dùng có thể chuyển đổi giữa các tab "Bài đăng" và "Reels" để xem nội dung tương ứng.
    + Nhấn vào một ảnh/video trong lưới sẽ hiển thị hiệu ứng nhấn hoặc có thể điều hướng đến chi tiết bài đăng/Reels.

5. Mock Data

Dữ liệu giả lập (mock data) là một phần thiết yếu của dự án, đóng vai trò quan trọng trong việc cho phép ứng dụng hiển thị nội dung phong phú và đầy đủ mà không cần phụ thuộc vào kết nối backend. Để tối ưu hóa tốc độ phát triển và sự linh hoạt ban đầu, dữ liệu giả lập được định nghĩa trực tiếp trong các file .dart chứa giao diện người dùng (UI), như các file widget hoặc screen, nơi chúng được sử dụng. Cách tiếp cận này giúp các thành phần UI có thể nhanh chóng hiển thị dữ liệu tĩnh, tạo điều kiện thuận lợi cho việc phát triển và kiểm thử giao diện.

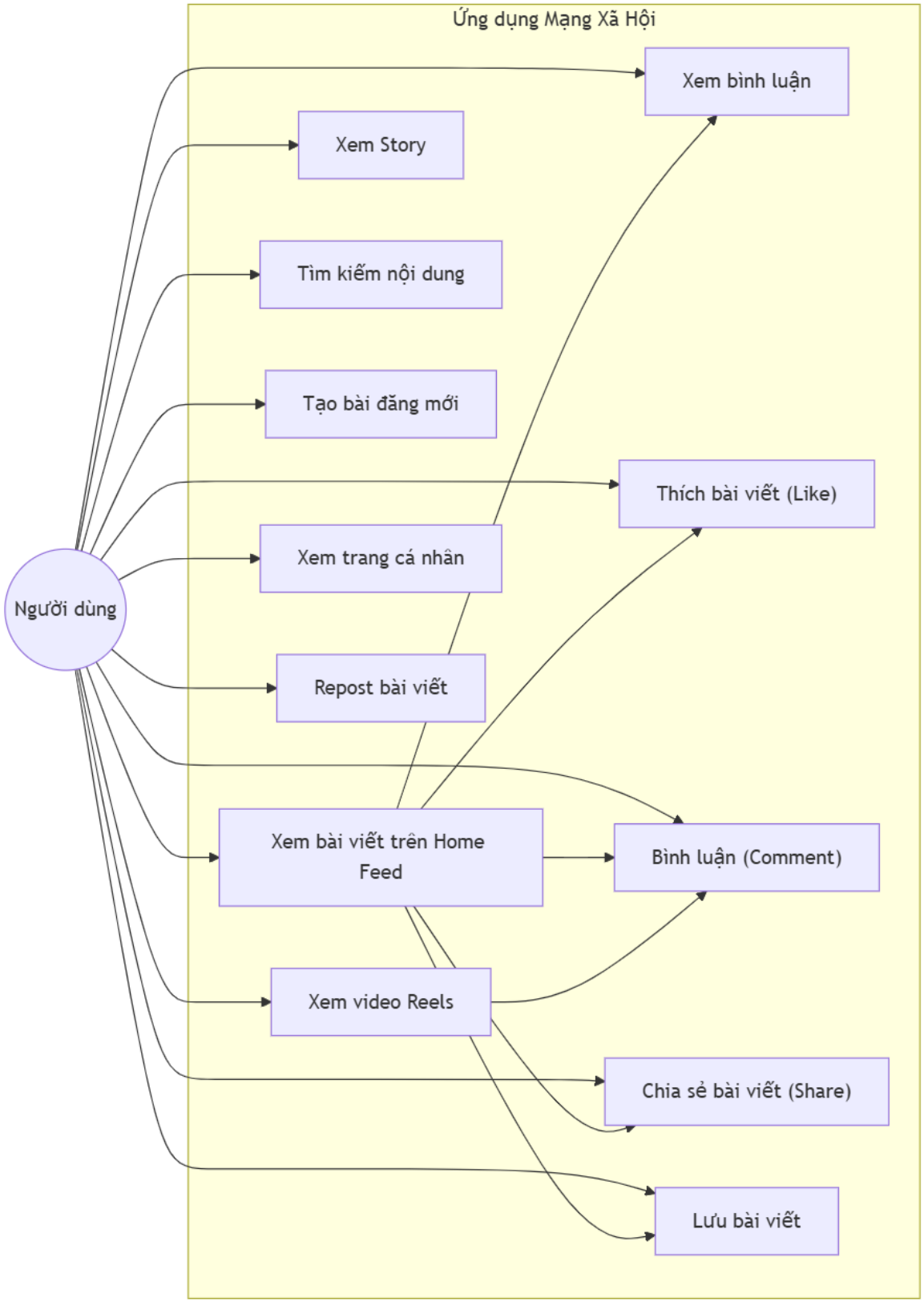
6. Yêu cầu phi chức năng

Các yêu cầu phi chức năng (Non-Functional Requirements) mô tả cách hệ thống phải hoạt động, thay vì mô tả những gì hệ thống phải làm. Chúng rất quan trọng để đảm bảo chất lượng của ứng dụng.

Tiêu chí	Mô tả chi tiết
Hiệu năng (Performance)	Tốc độ phản hồi: Ứng dụng phải có thời gian tải màn hình nhanh chóng (dưới 1-2 giây) và chuyển đổi giữa các màn hình mượt mà, không bị giật lag, đặc biệt khi cuộn danh sách lớn (Home Feed, Reels). Sử dụng tài nguyên: Tối ưu hóa việc sử dụng CPU, RAM và pin để đảm bảo ứng dụng hoạt động ổn định trên nhiều thiết bị di động, không gây nóng máy hoặc hao pin quá mức.
Khả năng sử dụng (Usability)	Giao diện người dùng (UI): Giao diện phải được thiết kế sao cho giống với ứng dụng Instagram bản thật nhất có thể về bố cục, màu sắc, font chữ và phong cách các icon. Mục tiêu là tạo ra một trải nghiệm người dùng quen thuộc và trực quan. Trải nghiệm người dùng (UX): Các tương tác như vuốt, chạm, nhấn phải có phản hồi rõ ràng và tự nhiên. Điều hướng giữa các màn hình phải dễ dàng và dễ hiểu.
Khả năng bảo trì (Maintainability)	Cấu trúc mã nguồn: Mã nguồn phải được tổ chức rõ ràng theo module, tuân thủ các nguyên tắc thiết kế tốt (SOLID, DRY). Các thành phần UI phải được tách biệt thành các widget nhỏ, tái sử dụng được, giúp dễ dàng debug, sửa lỗi và nâng cấp. Tính dễ đọc: Mã nguồn phải được viết rõ ràng, có comment đầy đủ ở những phần phức tạp, tuân thủ các quy tắc định dạng (ví dụ: flutter format) để bất kỳ nhà phát triển nào cũng có thể hiểu và làm việc với nó.
Khả năng mở rộng (Extensibility)	Chia module: Cấu trúc dự án phải cho phép dễ dàng mở rộng các tính năng mới (ví dụ: thêm tính năng nhắn tin, video call) mà không ảnh hưởng lớn đến các phần hiện có của ứng dụng. Tái sử dụng: Các widget và logic UI được thiết kế để có thể tái sử dụng ở nhiều màn hình hoặc ngữ cảnh khác nhau, giảm trùng lặp mã và tăng hiệu quả phát triển.
Khả năng cài đặt (Installability)	Ứng dụng phải có khả năng build thành file APK (Android) và chạy ổn định trên phiên bản hệ điều hành di động phổ biến (từ Android 5.0 trở lên). Kích thước ứng dụng: Kích thước file cài đặt (bundle size) của ứng dụng phải được giữ ở mức hợp lý, tối ưu hóa tài nguyên hình ảnh và font chữ để giảm dung lượng.
Bảo mật (Security - UI level)	(Lưu ý: Do không có backend, yêu cầu bảo mật chủ yếu liên quan đến việc xử lý dữ liệu giả lập và tránh các lỗ hổng UI cơ bản.) Xử lý dữ liệu giả lập an toàn: Đảm bảo rằng mock data không chứa bất kỳ thông tin nhạy cảm nào và được xử lý cục bộ.

7. Use Case Diagram (Mô hình ca sử dụng)

Sơ đồ Use Case giúp hình dung các chức năng chính của hệ thống từ góc nhìn của người dùng. Trong dự án Instagram UI Clone, người dùng (Actor) tương tác với các chức năng hiển thị giao diện.



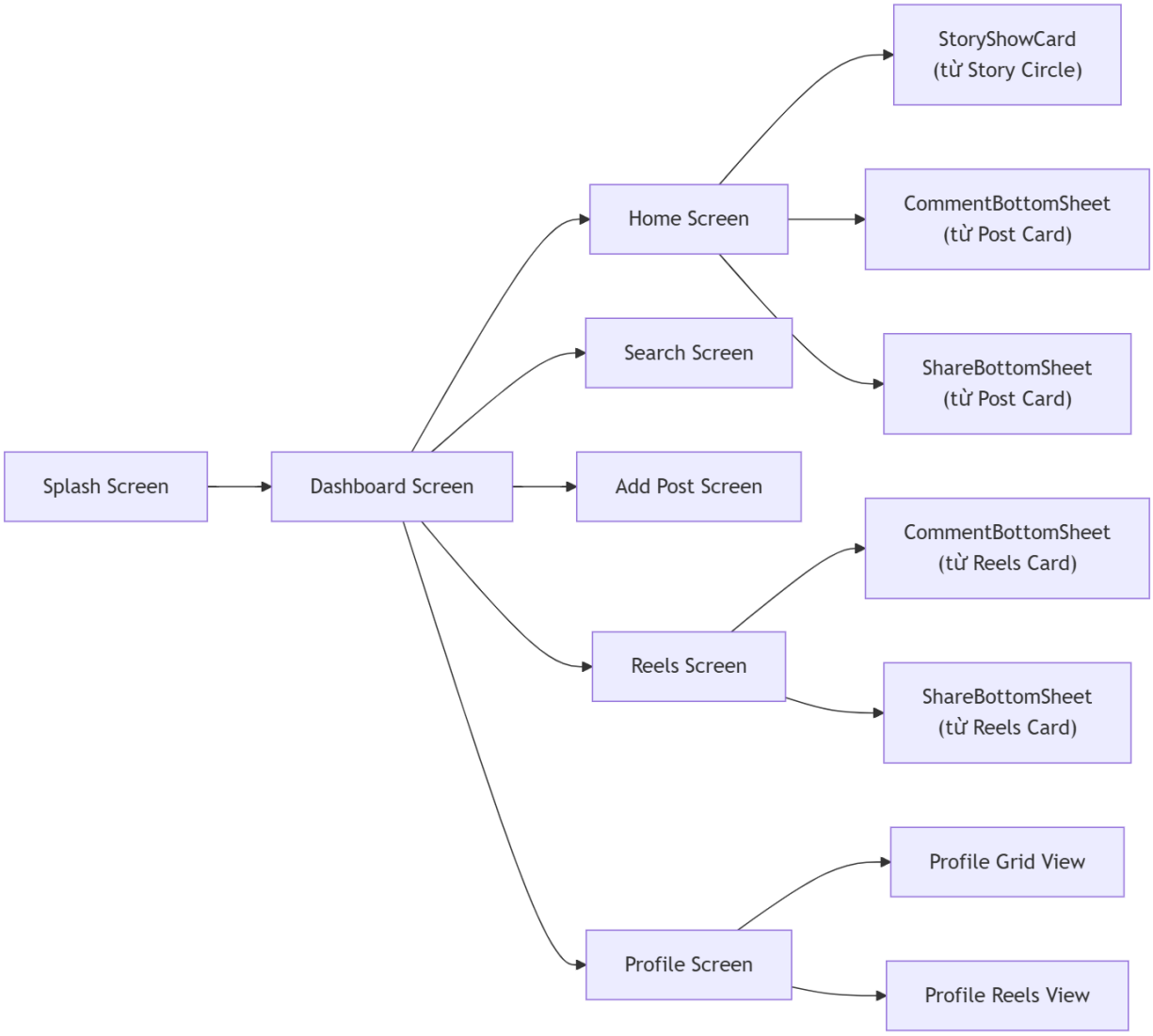
Hình 1.7. Use case

Mô tả các Use Case:

- Xem bài viết trên Home Feed: Người dùng có thể cuộn qua danh sách các bài viết, xem ảnh/video, đọc caption, và xem các tương tác giả lập.
- Xem Story: Người dùng có thể nhấn vào avatar Story và xem các Story toàn màn hình, vuốt để chuyển đổi.
- Tìm kiếm nội dung: Người dùng có thể truy cập màn hình tìm kiếm, xem lưới ảnh khám phá và nhập văn bản vào thanh tìm kiếm (không có chức năng tìm kiếm backend).
- Tạo bài đăng mới: Người dùng có thể truy cập màn hình thêm bài viết, mô phỏng việc chọn ảnh và thêm caption, sau đó "đăng" bài (chỉ là thông báo trên UI).
- Xem video Reels: Người dùng có thể cuộn dọc qua các video ngắn trên màn hình Reels, xem video, thông tin người đăng và các tương tác giả lập.
- Xem trang cá nhân: Người dùng có thể xem trang cá nhân của mình, bao gồm avatar, thông tin bio và các bài viết/Reels đã đăng.
- Xem bình luận: Từ bài viết, người dùng có thể mở Bottom Sheet để xem danh sách bình luận giả lập.

8. User Flow (Luồng người dùng)

Luồng người dùng mô tả trình tự các màn hình mà người dùng sẽ trải nghiệm khi tương tác với ứng dụng.



Hình 1.8. Use flow

Giải thích luồng:

1. Splash Screen: Màn hình khởi động ban đầu (nếu có), thường là logo ứng dụng.
2. Dashboard Screen: Màn hình chính chứa BottomNavigationBar và quản lý việc chuyển đổi giữa 5 màn hình con.
3. Home Screen:
    + Từ Home, người dùng có thể nhấn vào Story Circle để xem StoryShowCard.
    + Từ một UserPostCard, người dùng có thể mở CommentBottomSheet hoặc ShareBottomSheet.

4. Search Screen: Màn hình tìm kiếm độc lập.
5. Add Post Screen: Màn hình để tạo bài đăng mới.
6. Reels Screen:
    + Từ Reels, người dùng có thể vuốt lên/xuống để chuyển Reels.
    + Từ một ReelsCard, người dùng có thể mở CommentBottomSheet hoặc ShareBottomSheet.
7. Profile Screen:
    + Từ Profile, người dùng có thể chuyển đổi giữa Grid View (bài đăng) và Reels View (video ngắn của cá nhân).

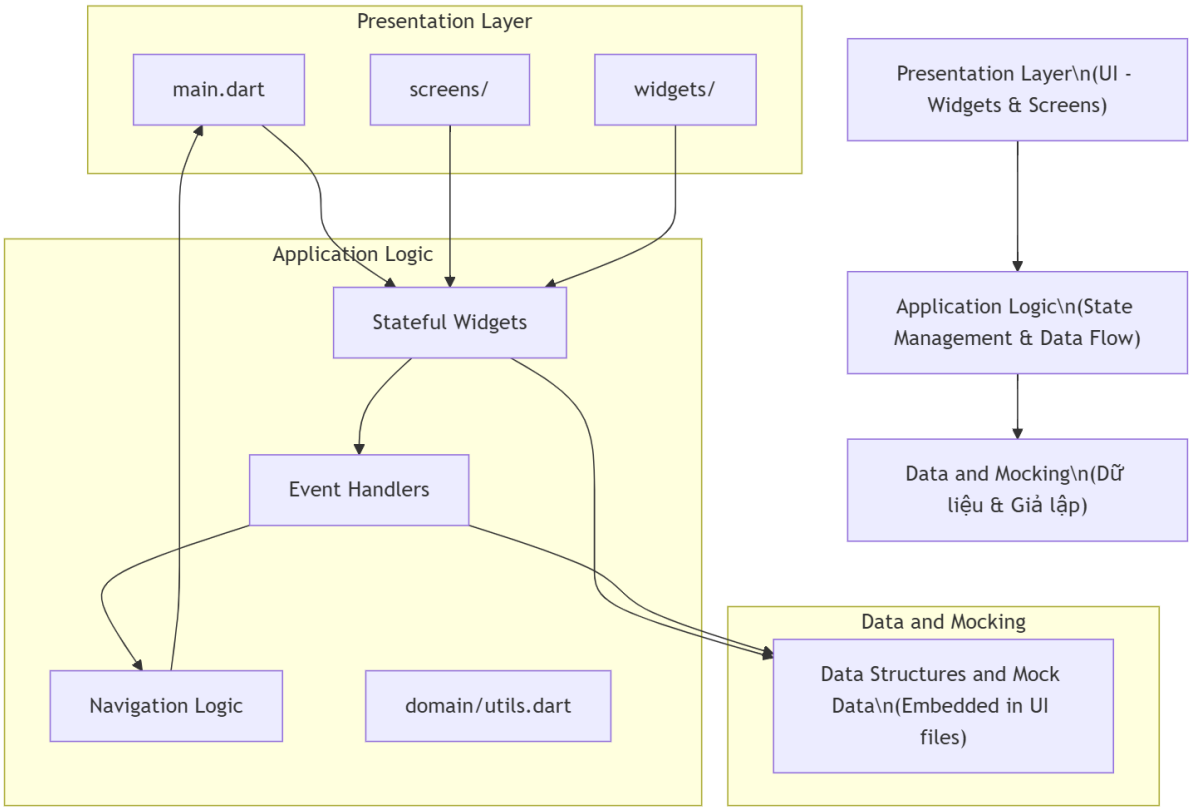
II. PHẦN II – SOFTWARE ARCHITECTURE DESIGN (SAD)

1. Kiến trúc hệ thống

Ứng dụng được tổ chức theo mô hình kiến trúc phân tầng giao diện (Layered UI Architecture), một cách tiếp cận phổ biến trong phát triển Flutter để tách biệt các trách nhiệm và dễ dàng bảo trì.

- Main Layer (main.dart): Đây là điểm khởi đầu của ứng dụng, nơi cấu hình MaterialApp và định nghĩa theme, routes ban đầu. Nó quản lý luồng khởi động và điều hướng tổng thể.
- Screen Layer (lib/screens/): Chứa các màn hình chính của ứng dụng, mỗi màn hình là một widget StatefulWidget hoặc StatelessWidget lớn. Các màn hình này chịu trách nhiệm bố cục tổng thể và quản lý các widget con.
    + bottom_navigation_screen.dart, dashboard_screen.dart: Quản lý BottomNavigationBar và việc hiển thị các màn hình con.
    + home_screen.dart, search_screen.dart, add_post_screen.dart, reels_screen.dart, profile_screen.dart: Các màn hình tương ứng với các tab chính.
- Widget Layer (lib/widgets/): Chứa các thành phần giao diện nhỏ hơn, có thể tái sử dụng trên nhiều màn hình. Đây là nơi chứa các widget tùy chỉnh như UserPostCard, ReelsCard, StoryShowCard, CommentBottomSheet, v.v. Việc tách biệt này giúp tránh trùng lặp DRY và tăng tính module hóa.
- Utils Layer (lib/domain/utils/): Chứa các tiện ích, hằng số, định nghĩa màu sắc, font chữ, các hàm helper chung cho toàn ứng dụng.

Sơ đồ cấu trúc tầng (3-Layer Diagram)

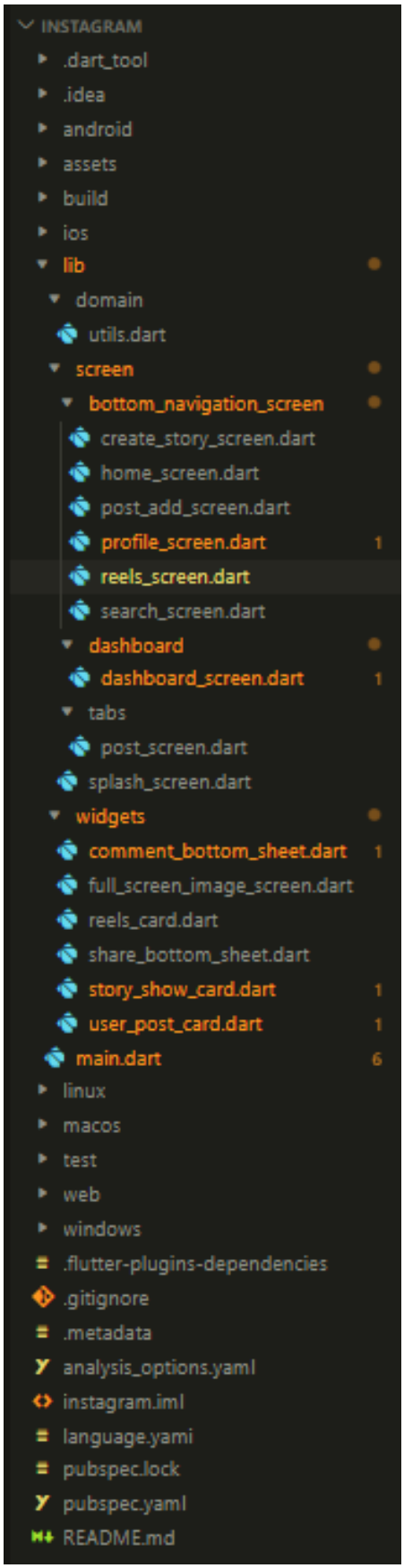


Giải thích:

- Presentation Layer: Đây là lớp tương tác trực tiếp với người dùng, chịu trách nhiệm hiển thị giao diện và nhận đầu vào. Nó bao gồm main.dart, các screens/ và widgets/ tái sử dụng.
- Application Logic: Lớp này xử lý logic nghiệp vụ cục bộ, quản lý trạng thái của các widget, và điều phối các tương tác người dùng. Trong dự án này, nó chủ yếu nằm trong các StatefulWidget và các hàm xử lý sự kiện.
- Data and Mocking Layer: Lớp này chịu trách nhiệm về dữ liệu. Trong dự án này, các cấu trúc dữ liệu và dữ liệu giả lập (mock data) được định nghĩa trực tiếp và nhúng vào các file UI (screens/widgets) mà chúng được sử dụng. Điều này giúp tối ưu hóa tốc độ phát triển giao diện ban đầu.

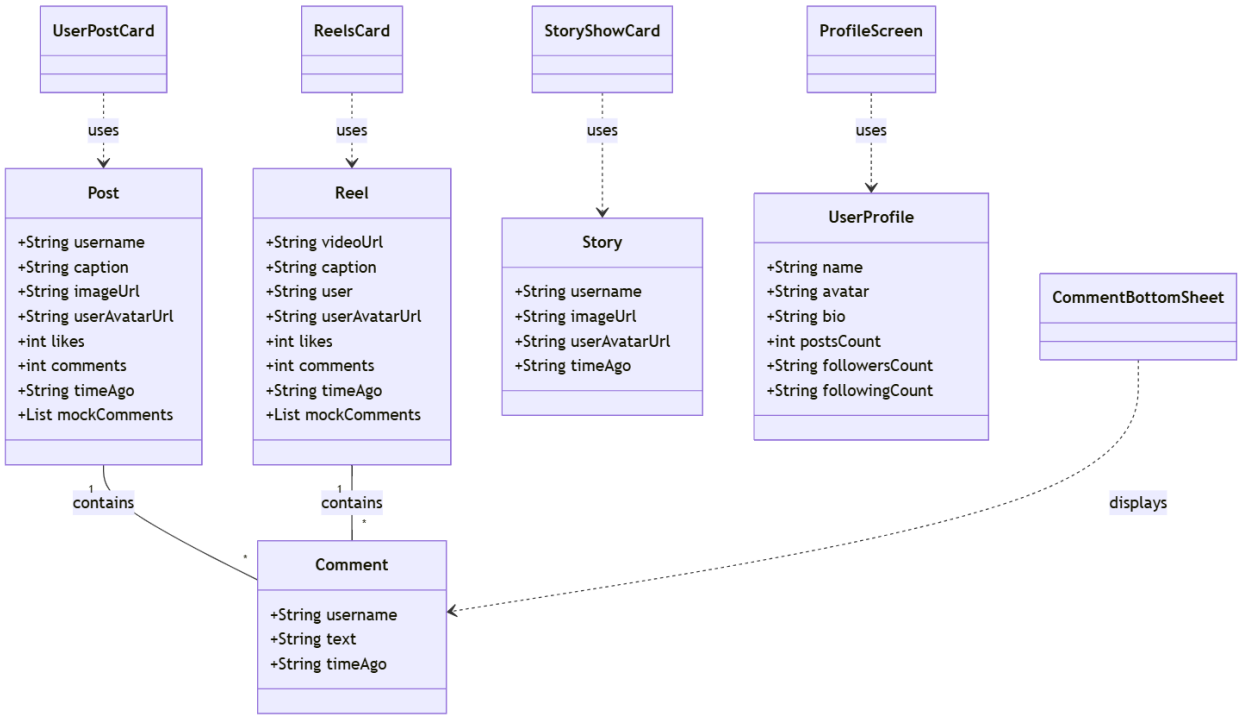
2. Cấu trúc thư mục (Project Structure)

Cấu trúc thư mục chi tiết giúp tổ chức mã nguồn một cách hợp lý, dễ dàng tìm kiếm và bảo trì.


INSTAGRAM
.dart_tool
.idea
android
assets
build
ios
lib
domain
utils.dart
screen
bottom_navigation_screen
create_story_screen.dart
home_screen.dart
post_add_screen.dart
profile_screen.dart 1
reels_screen.dart
search_screen.dart
dashboard
dashboard_screen.dart 1
tabs
post_screen.dart
splash_screen.dart
widgets
comment_bottom_sheet.dart 1
full_screen_image_screen.dart
reels_card.dart
share_bottom_sheet.dart
story_show_card.dart 1
user_post_card.dart 1
main.dart 6
linux
macos
test
web
windows
.flutter-plugins-dependencies
.gitignore
.metadata
analysis_options.yaml
instagram.iml
language.yaml
pubspec.lock
pubspec.yaml
README.md

3. Mô tả lớp (Class Diagram UML)

Sơ đồ lớp (Class Diagram) mô tả cấu trúc tĩnh của hệ thống, bao gồm các lớp, thuộc tính và mối quan hệ giữa chúng. Mặc dù đây là một dự án UI, việc định nghĩa các lớp dữ liệu giúp cấu trúc mock data một cách rõ ràng.



Giải thích các lớp:

- Post: Đại diện cho một bài đăng trên Home Feed. Chứa thông tin về người đăng, nội dung, hình ảnh, số lượt thích, bình luận và danh sách bình luận giả lập.
- Reel: Đại diện cho một video ngắn trên màn hình Reels. Chứa URL video, caption, thông tin người dùng và các tương tác.
- Story: Đại diện cho một Story của người dùng. Chứa URL hình ảnh/video, thông tin người đăng.
- UserProfile: Đại diện cho thông tin cá nhân của một người dùng. Chứa tên, avatar, bio, số lượng bài viết, người theo dõi và đang theo dõi.
- Comment: Đại diện cho một bình luận. Chứa tên người bình luận, nội dung bình luận và thời gian.
- Mối quan hệ:

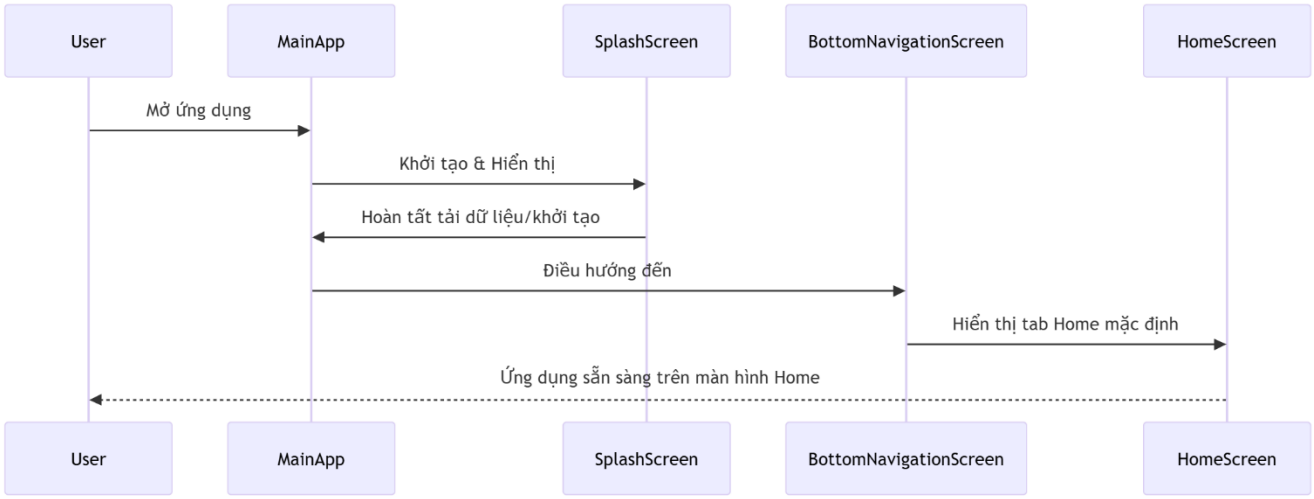
+ Post và Reel có mối quan hệ một-nhiều với Comment (một bài đăng/Reel có thể có nhiều bình luận).

+ Các widget UI như UserPostCard, ReelsCard, StoryShowCard, ProfileScreen, CommentBottomSheet sử dụng các đối tượng từ các lớp model tương ứng để hiển thị dữ liệu.

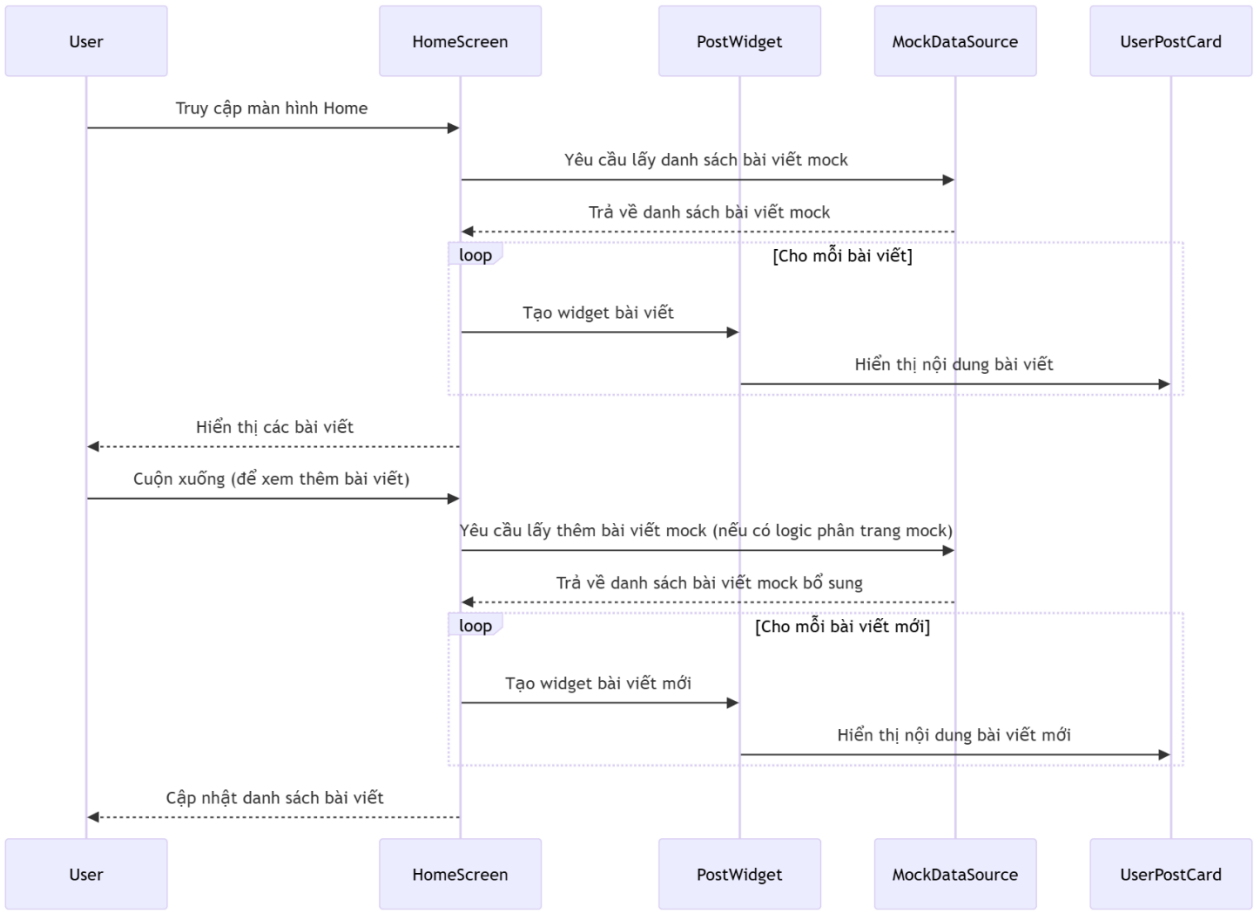
4. Sequence Diagram – Mô tả luồng hoạt động

Sơ đồ trình tự (Sequence Diagram) mô tả sự tương tác giữa các đối tượng theo thứ tự thời gian, cho thấy cách các chức năng được thực hiện.

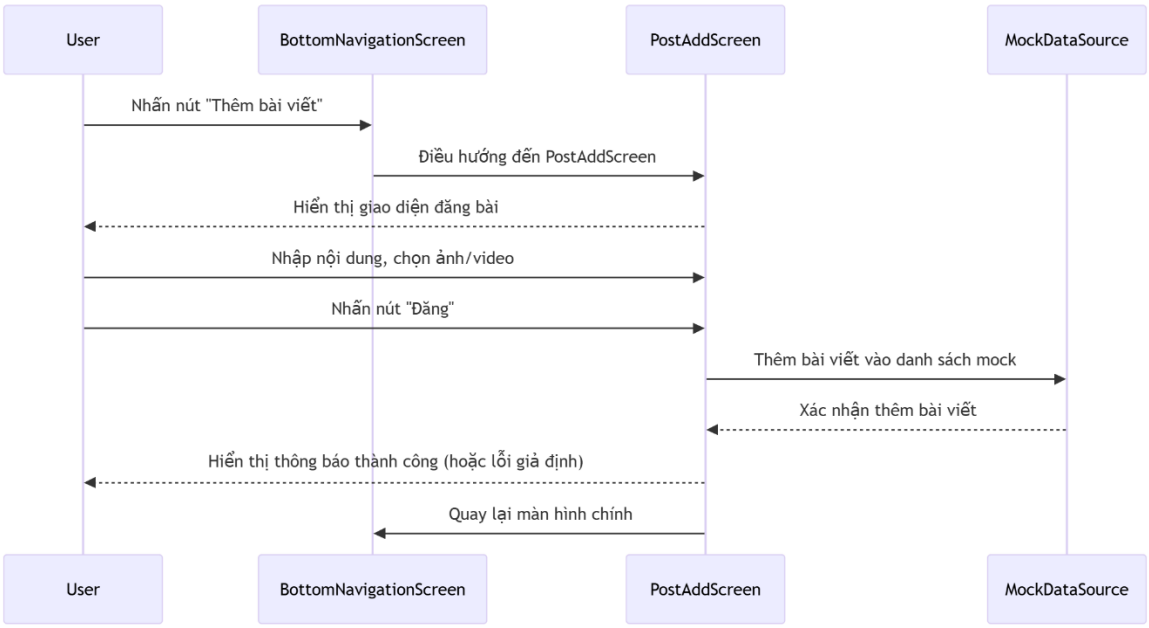
4.1. Luồng chung của ứng dụng



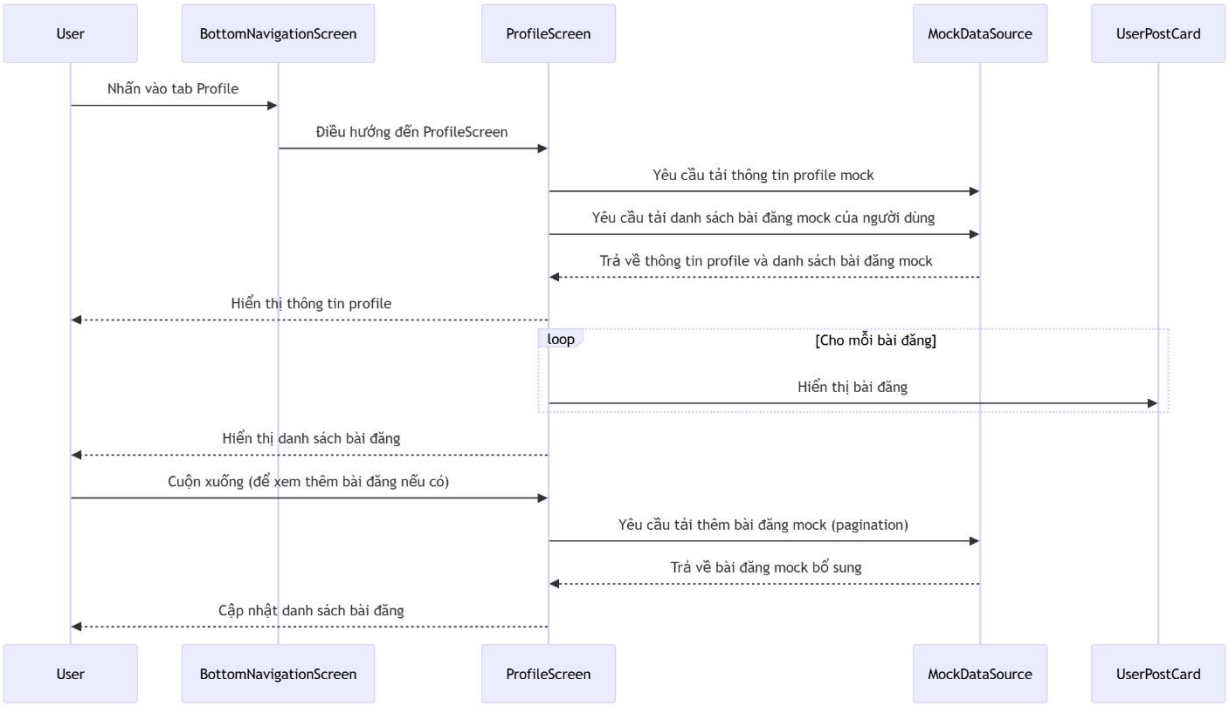
4.2. Luồng xem bài viết



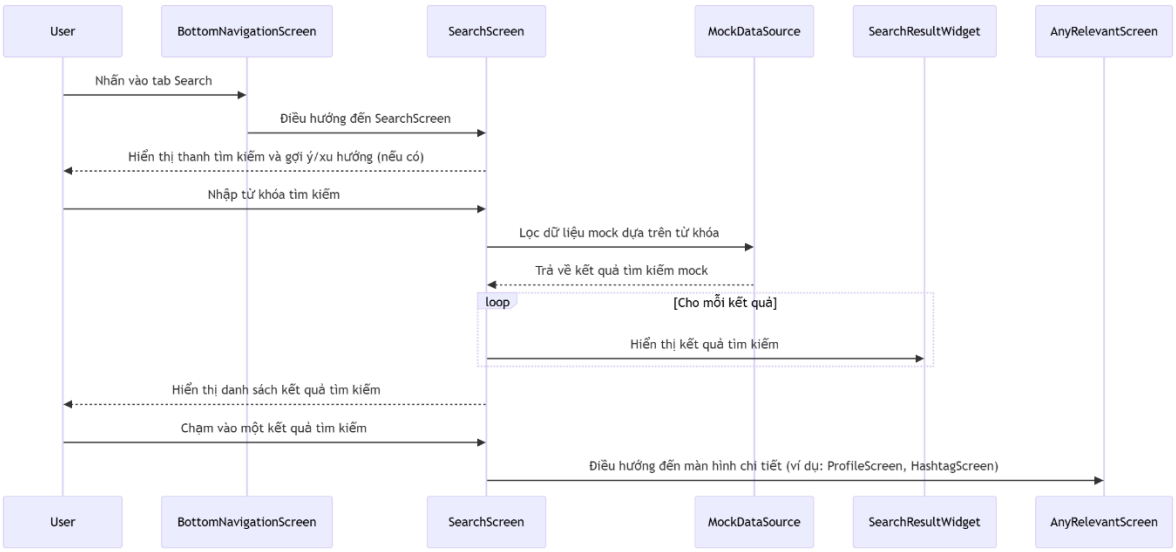
4.3. Luồng đăng bài viết mới



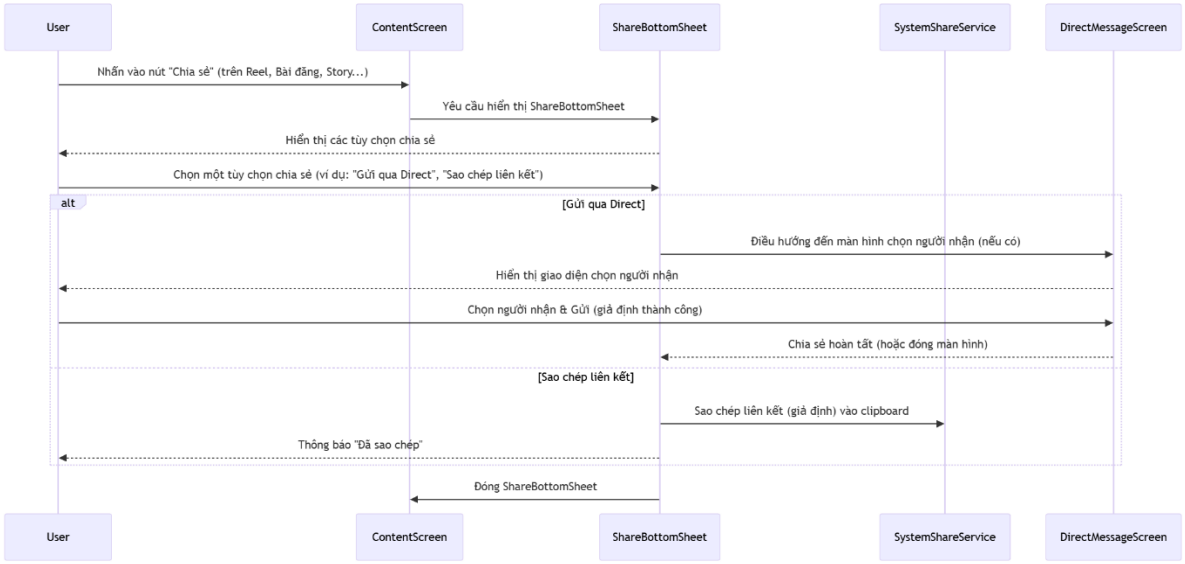
4.4. Luồng xem profile



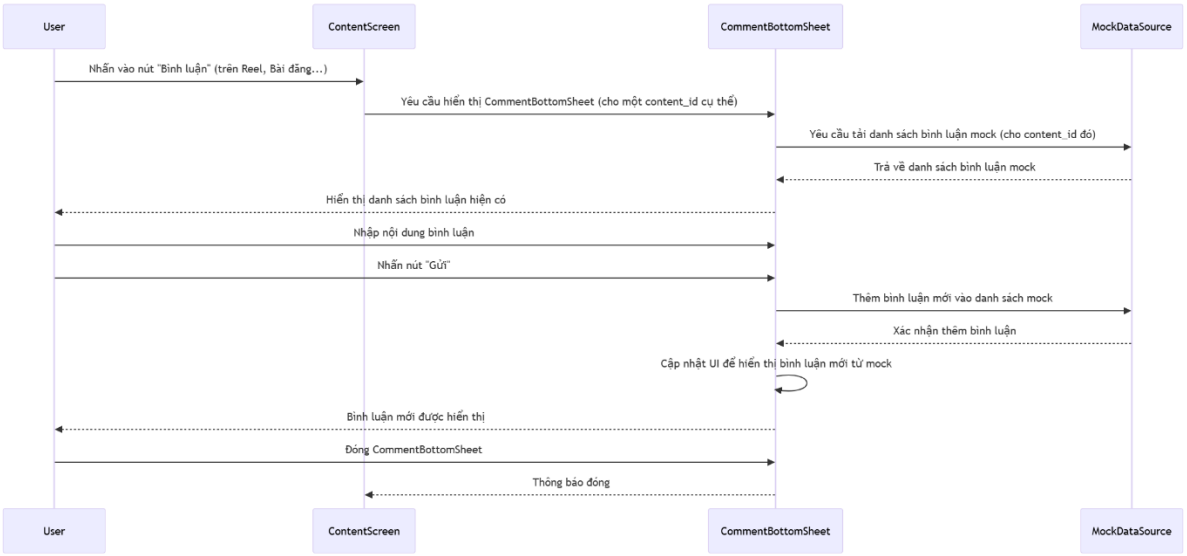
4.5. Luồng tìm kiếm



4.6. Luồng chia sẻ

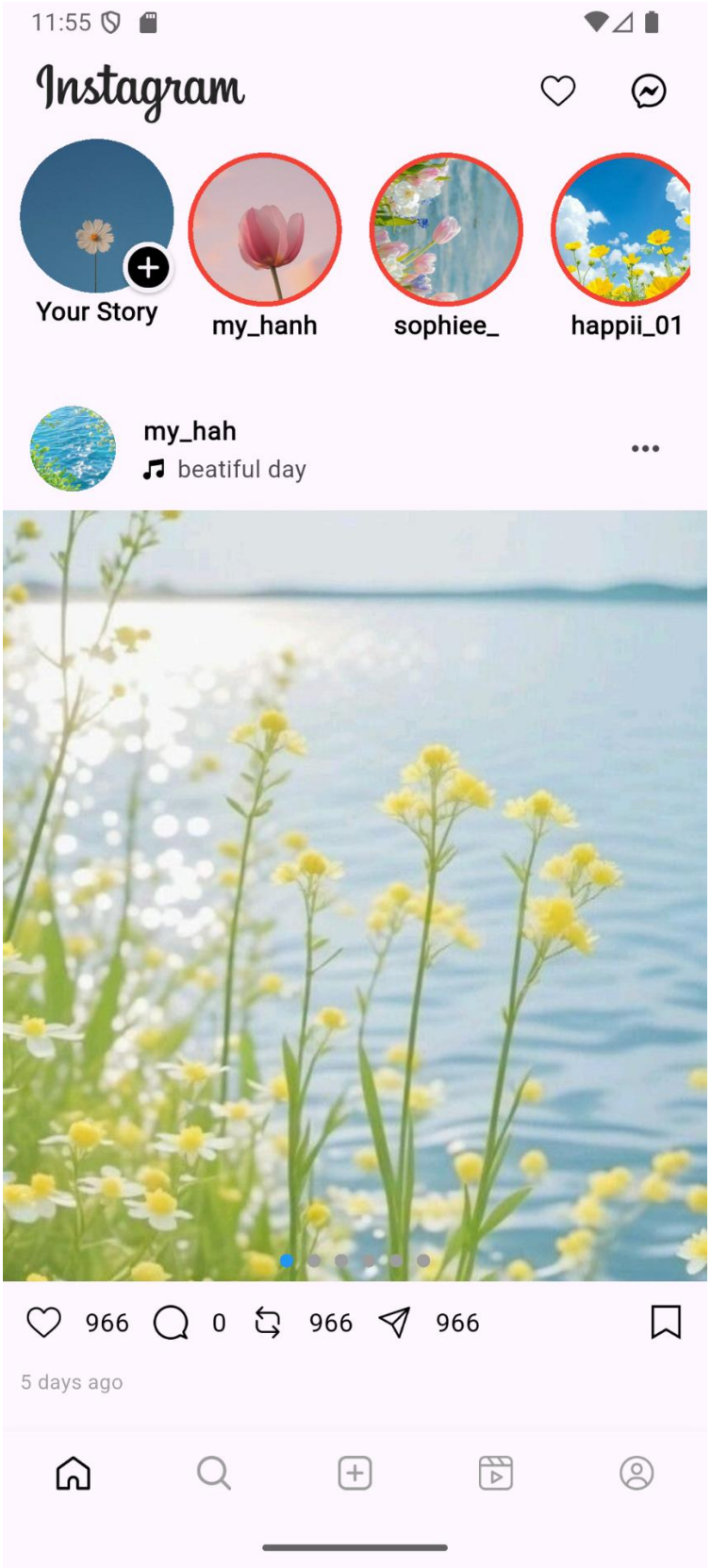


4.7. Luồng bình luận

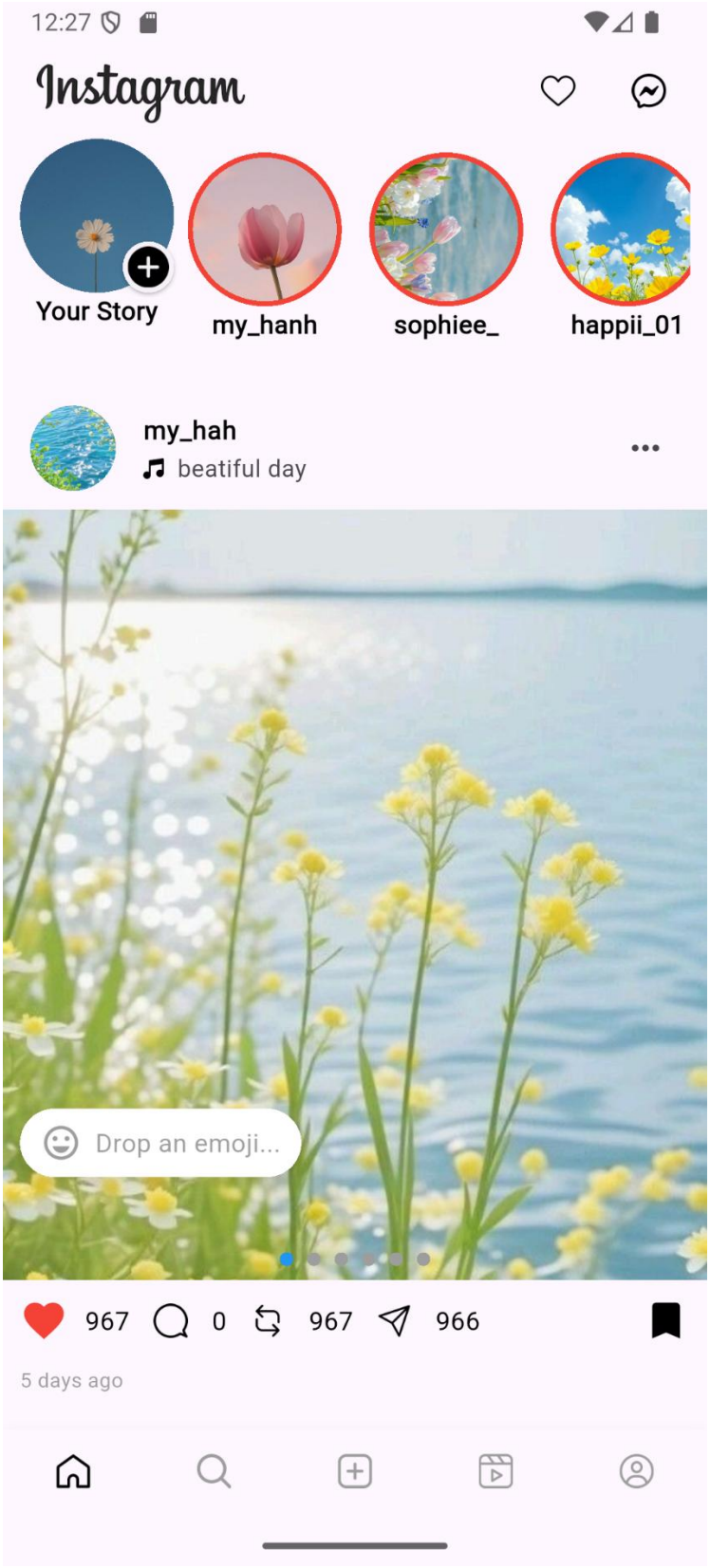


5. Thiết kế giao diện (UI Design Layouts)

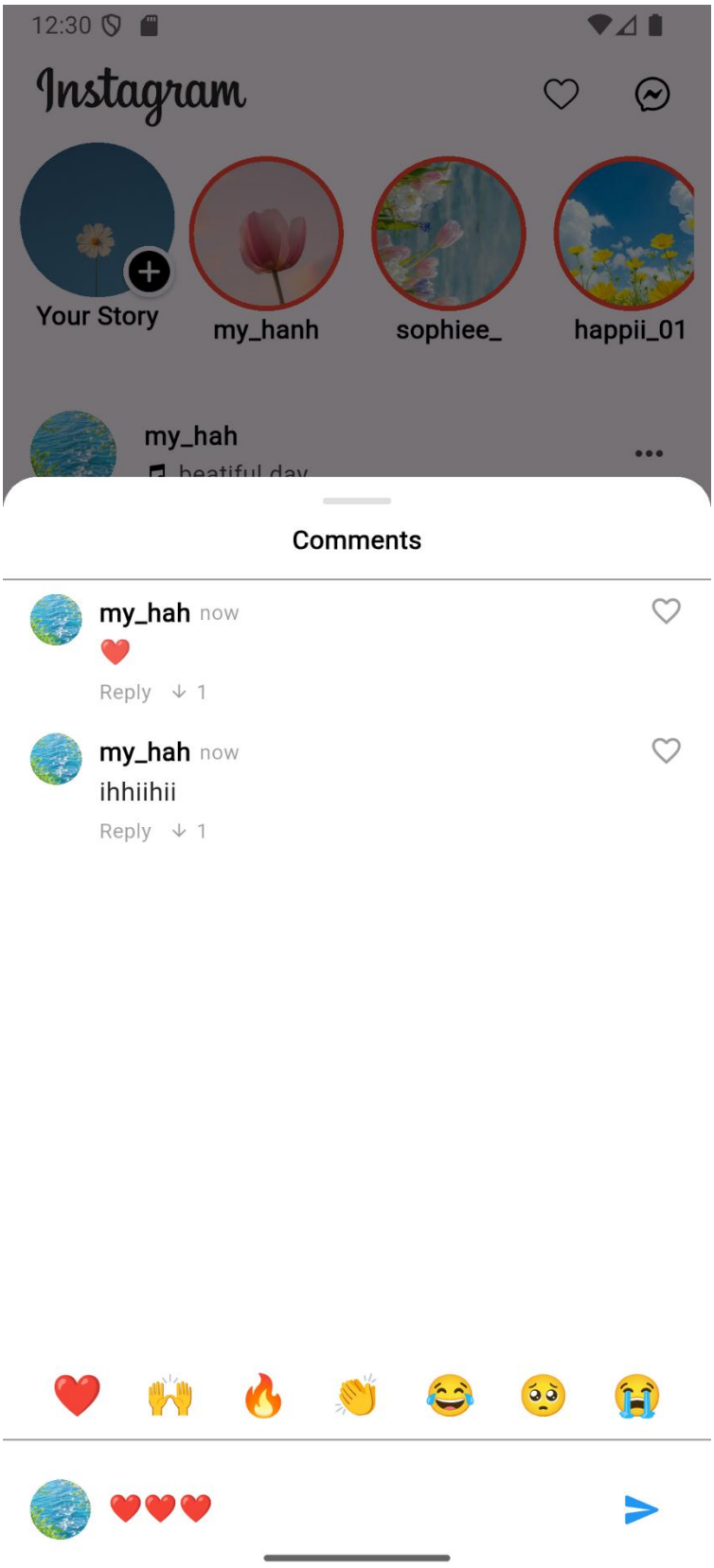
Phần này sẽ trình bày các ảnh chụp màn hình thực tế của ứng dụng Instagram UI Clone đã được xây dựng. Các hình ảnh này là bằng chứng trực quan về việc tái tạo giao diện người dùng của Instagram.



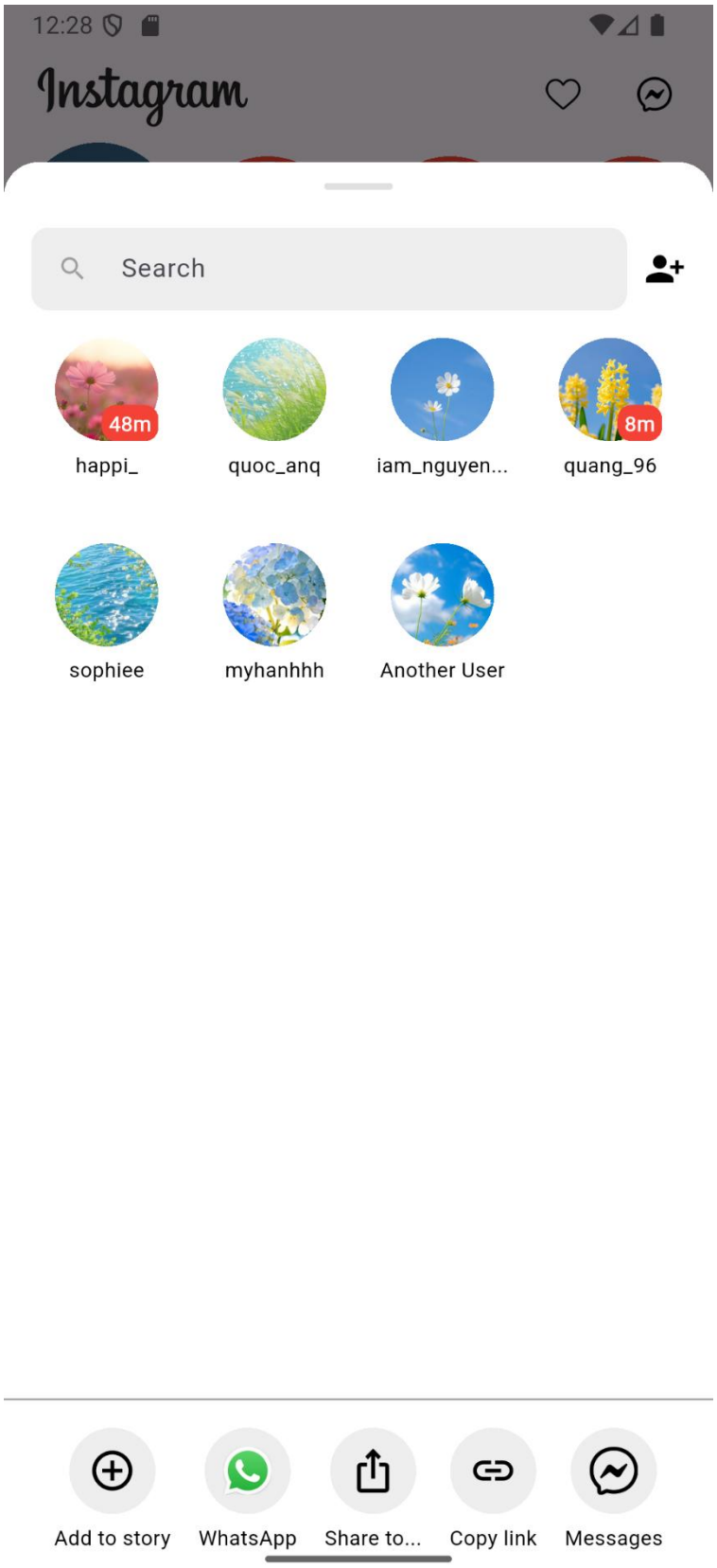
Hình 5.1. Màn hình Home (Home Screen)



Hình 5.2. Giao diện like, repost, đã lưu bài viết

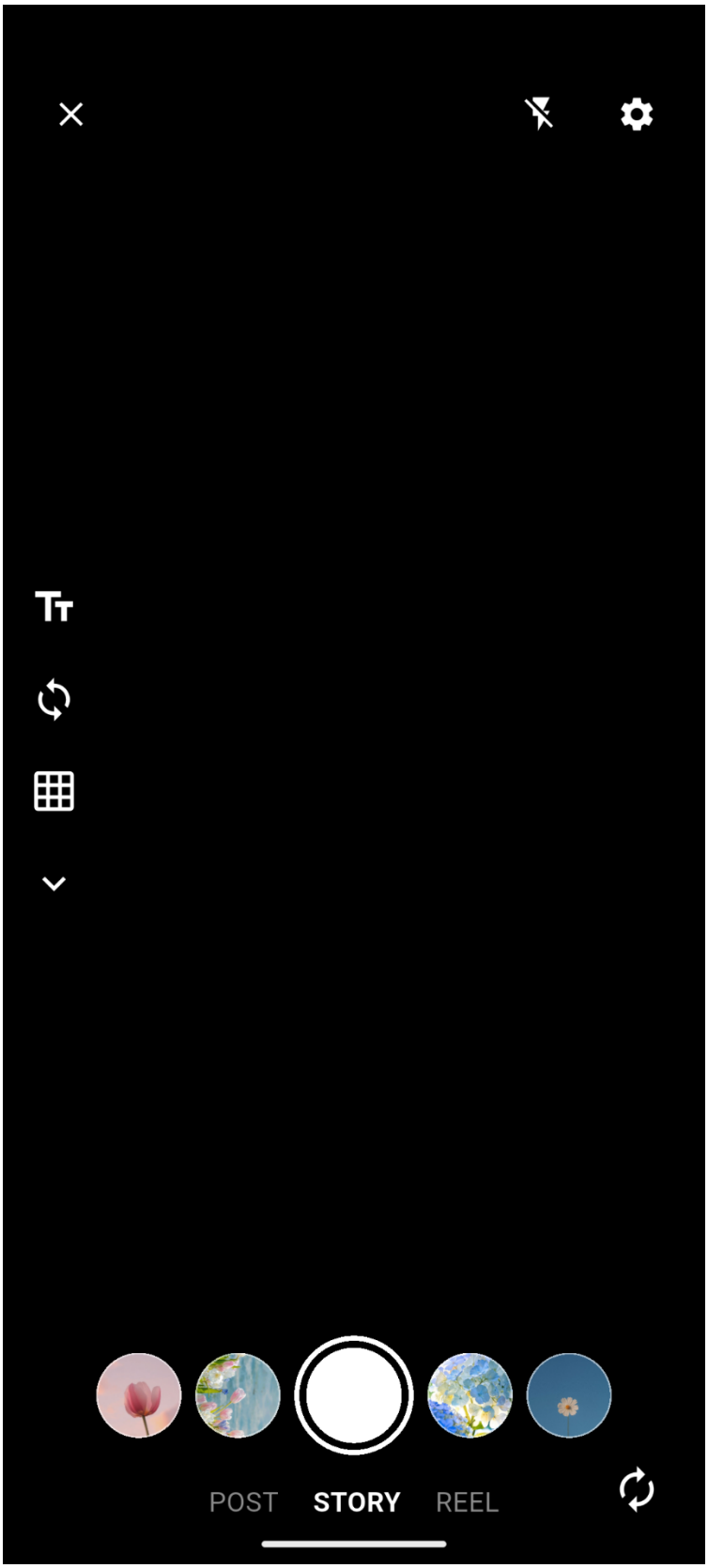


Hình 5.3. Giao diện bình luận bài viết

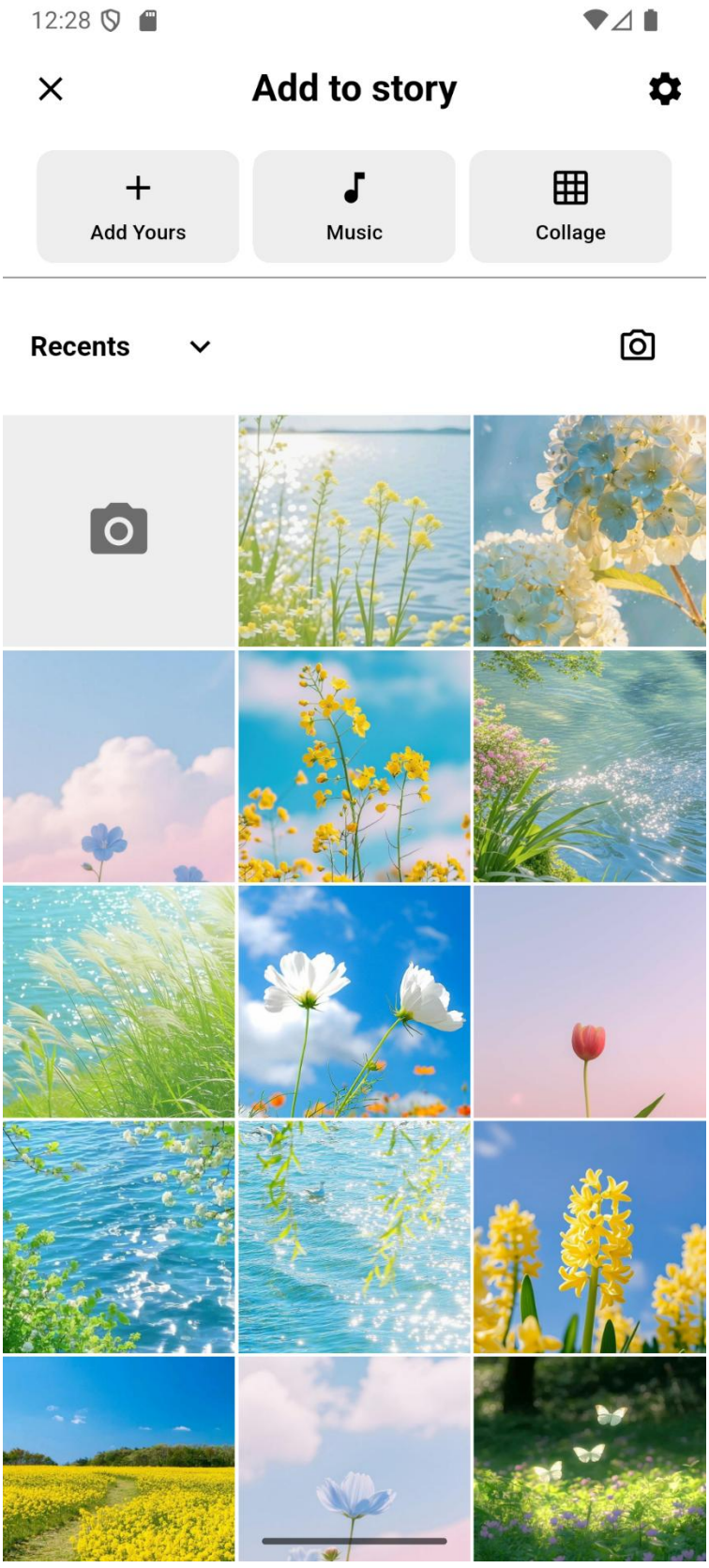


Hình 5.4. Giao diện chia sẻ bài viết

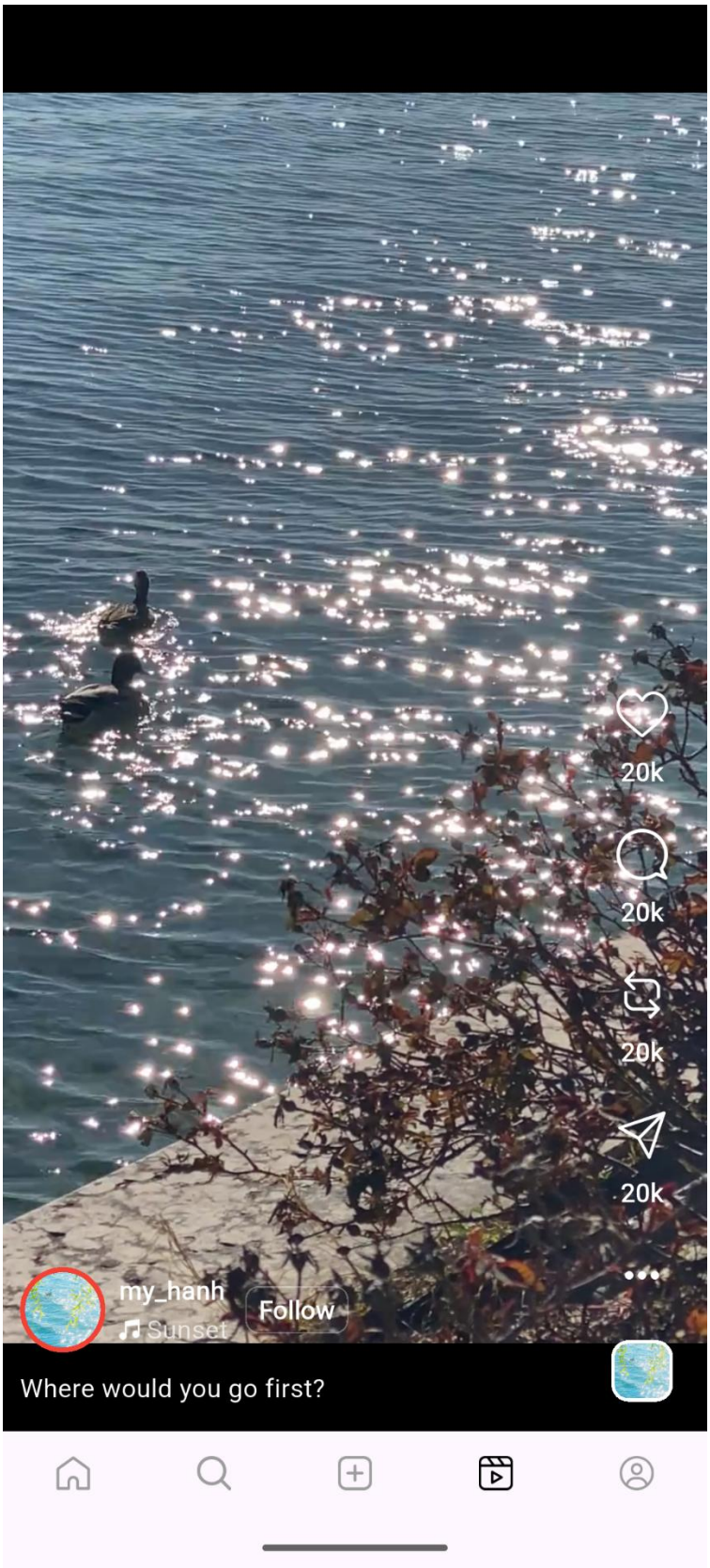
Hình 5.5. Màn hình Tìm kiếm (Search Screen).



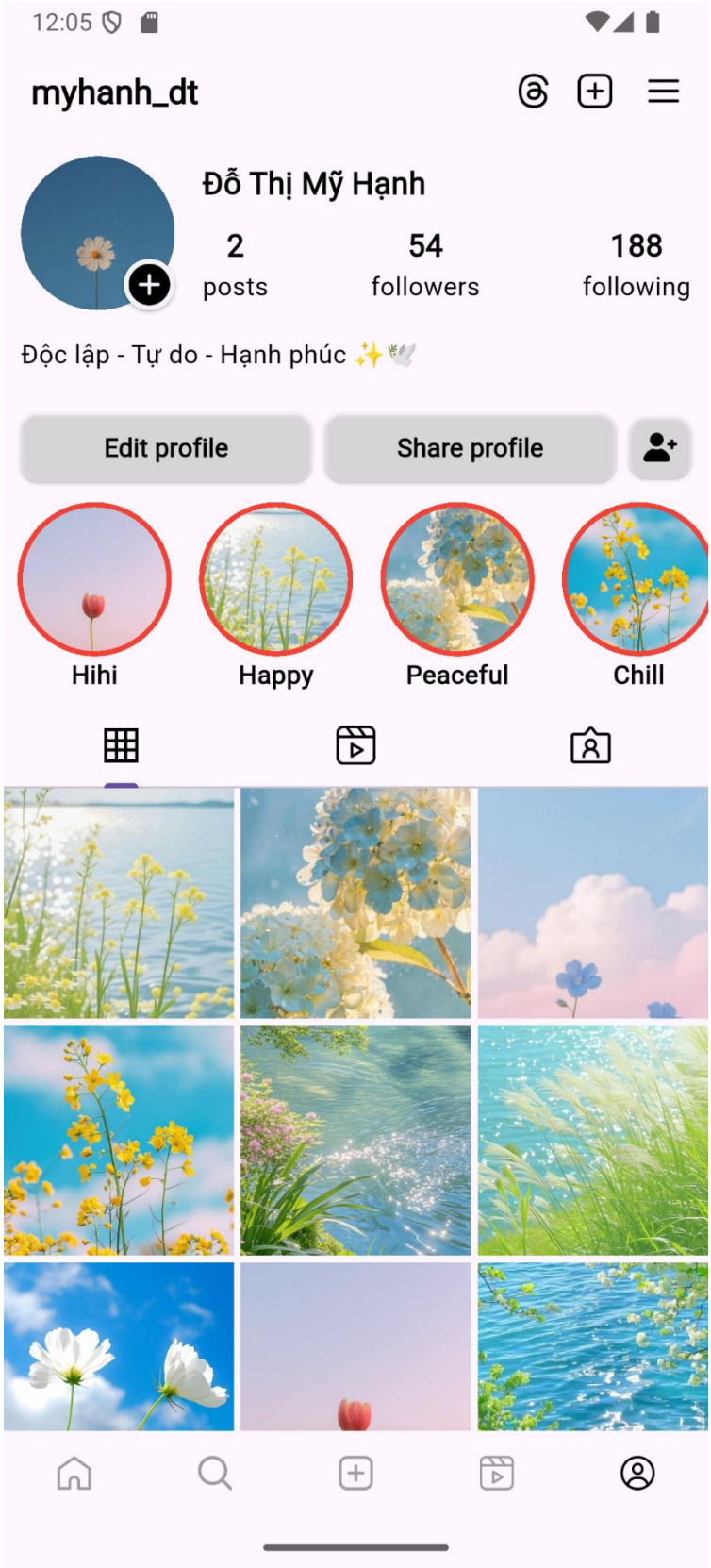
Hình 5.6. Màn hình Tìm kiếm (Search Screen), chụp ảnh mới



Hình 5.7. Màn hình Thêm bài đăng (Add Post Screen), chọn ảnh từ điện thoại



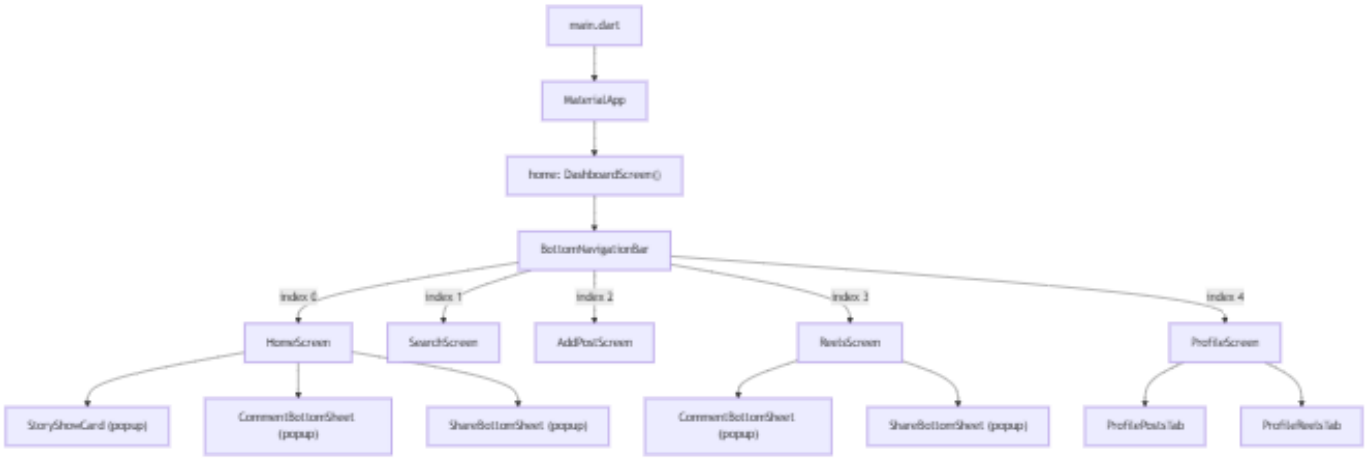
Hình 5.8. Màn hình Reels (Reels Screen)



Hình 5.9. Màn hình Hồ sơ (Profile Screen)

6. Navigation Flow Diagram

Sơ đồ luồng điều hướng này minh họa cách các màn hình được liên kết với nhau, đặc biệt là mối quan hệ giữa main.dart, dashboard_screen.dart và các màn hình chính.



Giải thích:

- main.dart khởi tạo MaterialApp và đặt DashboardScreen làm màn hình chính.
- DashboardScreen chứa BottomNavigationBar và quản lý việc hiển thị các màn hình con tương ứng với mỗi tab.
- Từ HomeScreen, ReelsScreen và các màn hình khác, các widget popup như StoryShowCard, CommentBottomSheet, ShareBottomSheet có thể được mở lên thông qua Navigator.push hoặc showModalBottomSheet.
- ProfileScreen sử dụng TabBarView để chuyển đổi giữa ProfilePostsTab và ProfileReelsTab.

7. Quản lý Widget tái sử dụng

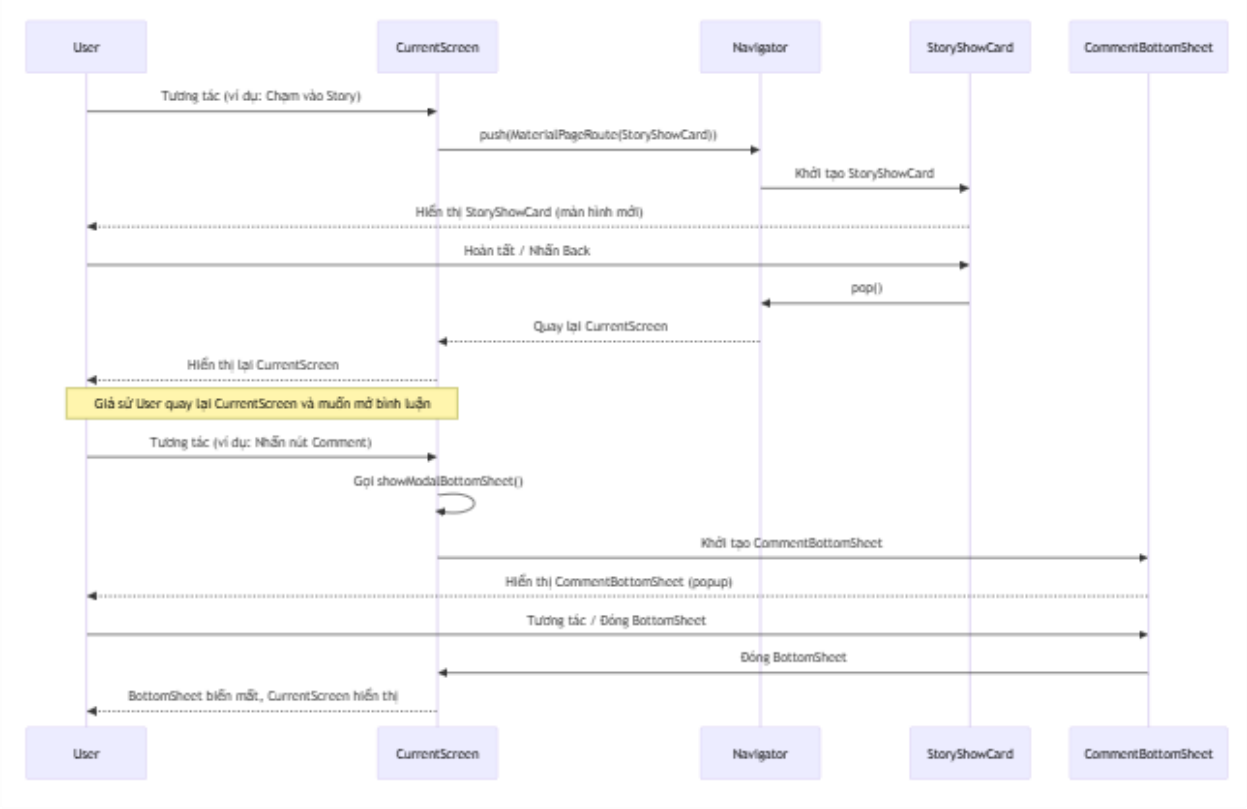
Việc thiết kế các widget có khả năng tái sử dụng là một nguyên tắc cốt lõi trong Flutter để đảm bảo tính module hóa, dễ bảo trì và giảm trùng lặp mã.

Widget	Chức năng	Ví dụ sử dụng
UserPostCard	Hiển thị một bài đăng hoàn chỉnh (ảnh, caption, like, comment, avatar).	Được sử dụng nhiều lần trong HomeScreen để hiển thị feed.
StoryCircleWidget	Hiển thị ảnh đại diện và tên người dùng cho Story.	Được sử dụng trong HomeScreen cho thanh Story.
StoryShowCard	Hiển thị Story toàn màn hình, có thể vuốt để xem Story kế tiếp.	Được gọi từ HomeScreen khi nhấn vào StoryCircleWidget.

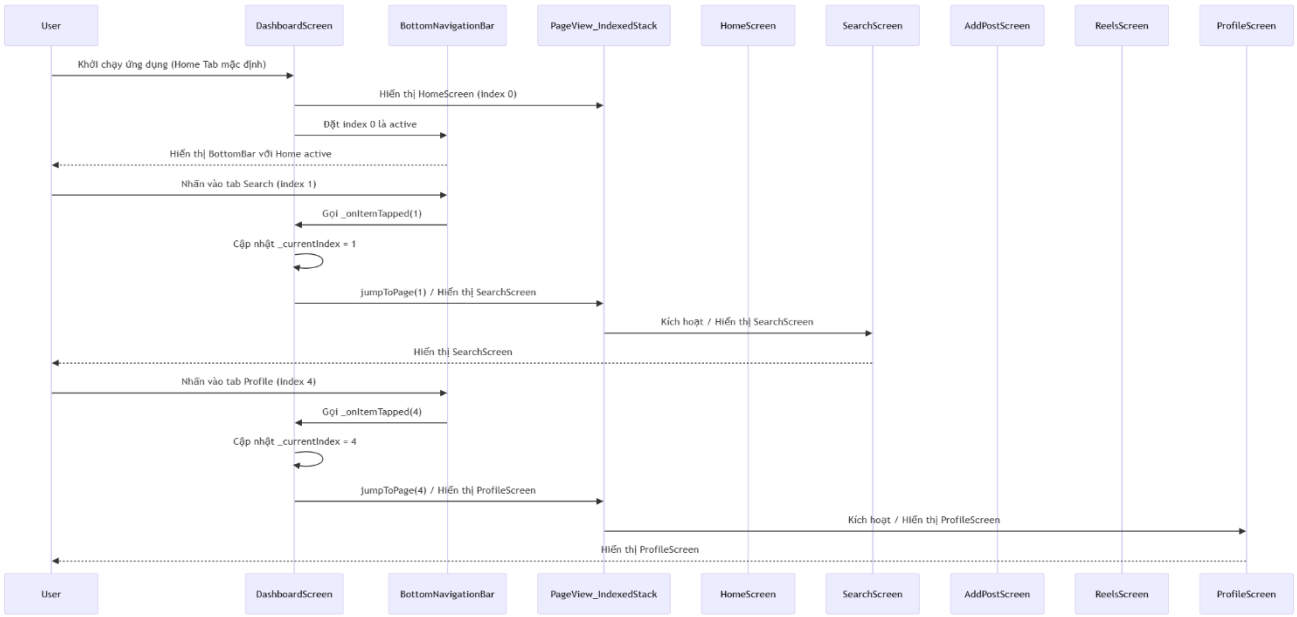
ReelsCard	Hiển thị một video Reels với các thông tin người dùng và nút tương tác.	Được sử dụng nhiều lần trong ReelsScreen (PageView).
CommentBottomSheet	Một Bottom Sheet hiển thị danh sách các bình luận giả lập.	Được gọi từ UserPostCard (HomeScreen) và ReelsCard.
ShareBottomSheet	Một Bottom Sheet mô phỏng giao diện chia sẻ (ví dụ: qua tin nhắn, các app).	Được gọi từ UserPostCard (HomeScreen) và ReelsCard.
ProfileHeader	Hiển thị avatar, tên người dùng, bio và các số liệu thống kê trên Profile.	Được sử dụng trong ProfileScreen.
ExploreGridItem	Hiển thị một ô ảnh/video trong lưới khám phá.	Được sử dụng trong SearchScreen (GridView).
CustomAppBar	AppBar tùy chỉnh cho các màn hình.	Có thể được sử dụng trên nhiều screens để đảm bảo phong cách nhất quán.

8. Logic điều hướng (Routing Logic)

Trong Flutter, điều hướng giữa các màn hình chủ yếu được thực hiện bằng Navigator. Dự án này sử dụng Navigator.push và Navigator.pop cho các luồng điều hướng không phải là tab chính, và quản lý trạng thái index cho BottomNavigationBar.



Biểu đồ tuần tự cho Điều hướng chung



Biểu đồ Tuần tự cho Quản lý Bottom Navigation Bar

9. Đánh giá và hướng phát triển

9.1. Kết quả đạt được

Dự án “Instagram UI Clone” đã đạt được các mục tiêu ban đầu và mang lại những kết quả quan trọng:

- Hoàn thiện giao diện chính của ứng dụng Instagram: Thành công trong việc tái tạo một cách chi tiết và trung thực các màn hình Home, Search, Add Post, Reels, Profile, cùng với các thành phần như Story, Comment Bottom Sheet, Share Bottom Sheet.
- Có thể chạy và tương tác cơ bản giữa các màn hình: Ứng dụng hoạt động mượt mà trên cả Android, cho phép người dùng điều hướng giữa các tab, cuộn qua các bài viết/Reels, và mở các pop-up tương tác.
- Học được cách chia nhỏ widget, quản lý trạng thái đơn giản và tổ chức mã nguồn khoa học: Dự án là cơ hội tuyệt vời để thực hành việc thiết kế các widget tái sử dụng, hiểu về lifecycle của StatefulWidget và cách quản lý trạng thái cục bộ với setState().
- Sử dụng hiệu quả mock data: Chứng minh khả năng xây dựng giao diện người dùng hoàn chỉnh mà không cần phụ thuộc vào backend, giúp tăng tốc độ phát triển UI.
- Rèn luyện kỹ năng xử lý assets: Quản lý hình ảnh và video trong thư mục assets/ và hiển thị chúng trong ứng dụng.

9.2. Hướng phát triển

Để biến dự án UI Clone này thành một ứng dụng hoạt động đầy đủ, các hướng phát triển sau đây là cần thiết và có thể được triển khai trong tương lai:

- Kết nối Firebase Authentication:
    + Triển khai chức năng đăng ký, đăng nhập và đăng xuất người dùng thực.
    + Sử dụng Firebase Auth để quản lý trạng thái xác thực người dùng.
- Tạo Database Firestore để lưu bài đăng, story, reels:
    + Thay thế mock data bằng dữ liệu thực được lưu trữ trên Firestore.
    + Triển khai các thao tác CRUD (Create, Read, Update, Delete) cho bài viết, story, reels, bình luận.
    + Thiết lập mối quan hệ giữa người dùng và nội dung của họ.
- Tích hợp Image Picker và upload ảnh thật:
    + Cho phép người dùng chọn ảnh từ thư viện hoặc chụp ảnh mới.
    + Upload ảnh lên Firebase Storage và lưu URL vào Firestore.
- Thêm Dark Mode / Light Mode:
    + Triển khai khả năng chuyển đổi giao diện giữa chế độ sáng và tối, tăng tính cá nhân hóa cho người dùng.
- Thêm Notification và Profile Edit:
    + Phát triển hệ thống thông báo giả lập hoặc tích hợp Firebase Cloud Messaging cho thông báo đẩy.
    + Cho phép người dùng chỉnh sửa thông tin cá nhân (bio, avatar) và cập nhật lên Firestore.
- Thêm chức năng Follow/Unfollow:
    + Triển khai logic theo dõi/bỏ theo dõi người dùng, cập nhật trạng thái trong Firestore.
- Tích hợp API thực:
    + Kết nối với các API bên ngoài cho các tính năng như bản đồ, vị trí, hoặc dịch thuật.
- Tối ưu hóa hiệu năng và trải nghiệm người dùng:
    + Cải thiện tốc độ tải ảnh, video, tối ưu hóa việc sử dụng bộ nhớ.

+ Thêm các animation và transition mượt mà hơn.

+ Xử lý lỗi và hiển thị thông báo thân thiện với người dùng.

10. Kết luận

Dự án "Instagram UI Clone" là một minh chứng rõ ràng cho khả năng của Flutter trong việc xây dựng các ứng dụng di động có giao diện phức tạp và đẹp mắt. Dự án này không chỉ giúp nhóm hoàn thiện kỹ năng lập trình UI/UX mà còn cung cấp một cái nhìn sâu sắc về quy trình phát triển phần mềm:

- Xác định yêu cầu (SRS): Từ việc hiểu rõ mục tiêu và phạm vi của một ứng dụng phổ biến.
- Phân tích thiết kế (SAD): Đến việc lên kế hoạch kiến trúc, cấu trúc thư mục, và các sơ đồ tương tác.
- Xây dựng và tổ chức mã nguồn khoa học: Áp dụng các nguyên tắc thiết kế tốt để tạo ra mã nguồn dễ đọc, dễ bảo trì và dễ mở rộng.
- Hiểu cấu trúc của một ứng dụng mạng xã hội phổ biến: Nắm bắt các thành phần cốt lõi và luồng tương tác mà người dùng mong đợi.

Mặc dù chưa có backend và cơ sở dữ liệu thực, dự án đã thể hiện đầy đủ quy trình phát triển phần mềm theo hướng giao diện – lấy người dùng làm trung tâm, đặt nền tảng vững chắc cho việc phát triển các ứng dụng di động phức tạp hơn trong tương lai. Đây là một bước đệm quan trọng để tiến tới xây dựng các ứng dụng Flutter hoàn chỉnh, có khả năng kết nối với dữ liệu và tương tác với người dùng theo thời gian thực.