

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG**  
**TRƯỜNG ĐẠI HỌC THỦY LỢI**



**BÀI TẬP LỚN**  
**PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**  
**ĐỀ TÀI: ỨNG DỤNG ...**

**Giáo viên hướng dẫn: ThS. Kiều Tuấn Dũng**

**Sinh viên thực hiện:**

<b>STT</b>	<b>Mã sinh viên</b>	<b>Họ và tên</b>	<b>Lớp</b>
1	2251061698	Lê Mạnh Hùng Anh	64CNTT1
2	2251061799	Phan Quang Huy	64CNTT1

**Hà Nội, năm 2025**

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG  
TRƯỜNG ĐẠI HỌC THỦY LỢI**



**BÀI TẬP LỚN**

**PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG  
ĐỀ TÀI: ỨNG DỤNG**

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
1	2251061698	Lê Mạnh Hùng Anh			
2	2251061799	Phan Quang Huy			

**CÁN BỘ CHẤM THI**

---

Hà Nội, năm 2025

## LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, mạng xã hội đã trở thành một phần không thể thiếu trong đời sống hằng ngày của con người. Trong đó, các nền tảng chia sẻ hình ảnh như Instagram đóng vai trò quan trọng trong việc kết nối, giao tiếp và thể hiện bản thân giữa các cá nhân trên toàn thế giới.

Nhận thức được tầm quan trọng đó, nhóm chúng em thực hiện bài tập lớn với đề tài **“Xây dựng ứng dụng chia sẻ hình ảnh – Instagram Clone”**. Đây là một dự án mô phỏng các tính năng cơ bản của ứng dụng Instagram, nhằm giúp sinh viên hiểu rõ hơn về quy trình phát triển một ứng dụng mạng xã hội từ frontend đến backend, đồng thời áp dụng và củng cố kiến thức đã học như lập trình mobile, thiết kế giao diện, quản lý cơ sở dữ liệu, tương tác với server và các vấn đề liên quan đến bảo mật.

Thông qua dự án này, sinh viên không chỉ có cơ hội rèn luyện kỹ năng lập trình mà còn nâng cao khả năng làm việc nhóm, tư duy thiết kế hệ thống và giải quyết vấn đề thực tiễn. Mặc dù ứng dụng chưa thể hoàn thiện đầy đủ như phiên bản gốc, nhưng đó là bước đầu quan trọng để tiếp cận và làm chủ những công nghệ hiện đại trong phát triển phần mềm.

Nhóm chúng em xin chân thành cảm ơn thầy đã hướng dẫn và tạo điều kiện thuận lợi để thực hiện bài tập lớn này. Chúng em rất mong nhận được những góp ý quý báu để hoàn thiện hơn trong các dự án tiếp theo.

# MỤC LỤC

## DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG BIỂU

## DANH MỤC CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	RWD	Responsive Web Design
2		



## Chương 1. TỔNG QUAN VỀ ĐỀ TÀI

### 1.1. Giới thiệu về đề tài

Instagram là một trong những mạng xã hội chia sẻ hình ảnh phổ biến nhất hiện nay, với hàng trăm triệu người dùng trên toàn thế giới. Ứng dụng này cho phép người dùng đăng tải hình ảnh, video, tương tác với nhau thông qua lượt thích, bình luận và theo dõi. Với giao diện trực quan và trải nghiệm người dùng mượt mà, Instagram đã trở thành hình mẫu lý tưởng cho các ứng dụng mạng xã hội hiện đại.

Đề tài "**Instagram Clone**" là một phiên bản mô phỏng các tính năng cơ bản của Instagram, được triển khai dưới dạng một ứng dụng mobile. Trong khuôn khổ bài tập lớn này, nhóm chúng em tập trung xây dựng một hệ thống có thể:

- Đăng ký và đăng nhập người dùng
- Tải lên và hiển thị hình ảnh
- Tương tác với bài viết (like, comment)
- Xem trang cá nhân và danh sách bài viết của người dùng
- Lưu trữ dữ liệu và hình ảnh thông qua backend và dịch vụ đám mây

Dự án được phát triển bằng ngôn ngữ **Kotlin** trên nền tảng **Android**, sử dụng **Firebase** để xử lý các chức năng như xác thực người dùng, lưu trữ hình ảnh, và quản lý cơ sở dữ liệu thời gian thực. Bên cạnh đó, nhóm cũng chú trọng đến giao diện người dùng (UI/UX) và tính bảo mật trong quá trình phát triển ứng dụng.

Thông qua đề tài này, sinh viên có thể hình dung rõ hơn về kiến trúc tổng thể của một ứng dụng mạng xã hội hiện đại, đồng thời áp dụng được nhiều kiến thức liên môn một cách thực tế và hiệu quả.

### 1.2. Mục tiêu của đề tài

- Hiểu và áp dụng các công nghệ phát triển ứng dụng di động.
- Nắm vững cách thiết kế và triển khai backend phục vụ ứng dụng di động.
- Xây dựng các chức năng chính của một mạng xã hội.
- Tăng cường kỹ năng làm việc nhóm, quản lý dự án và lập trình.

### 1.3. Phạm vi của đề tài

- Ứng dụng này chỉ được phát triển để phục vụ nhóm thực hiện dự án và không nhằm mục đích công khai ra toàn cầu.
- Người dùng chính là các thành viên trong nhóm, sử dụng ứng dụng để thực hành và kiểm thử các tính năng của một nền tảng mạng xã hội.
- Hệ thống không có cơ chế quản trị viên mà chỉ có người dùng bình thường.

#### **1.4 Phân chia nhiệm vụ**

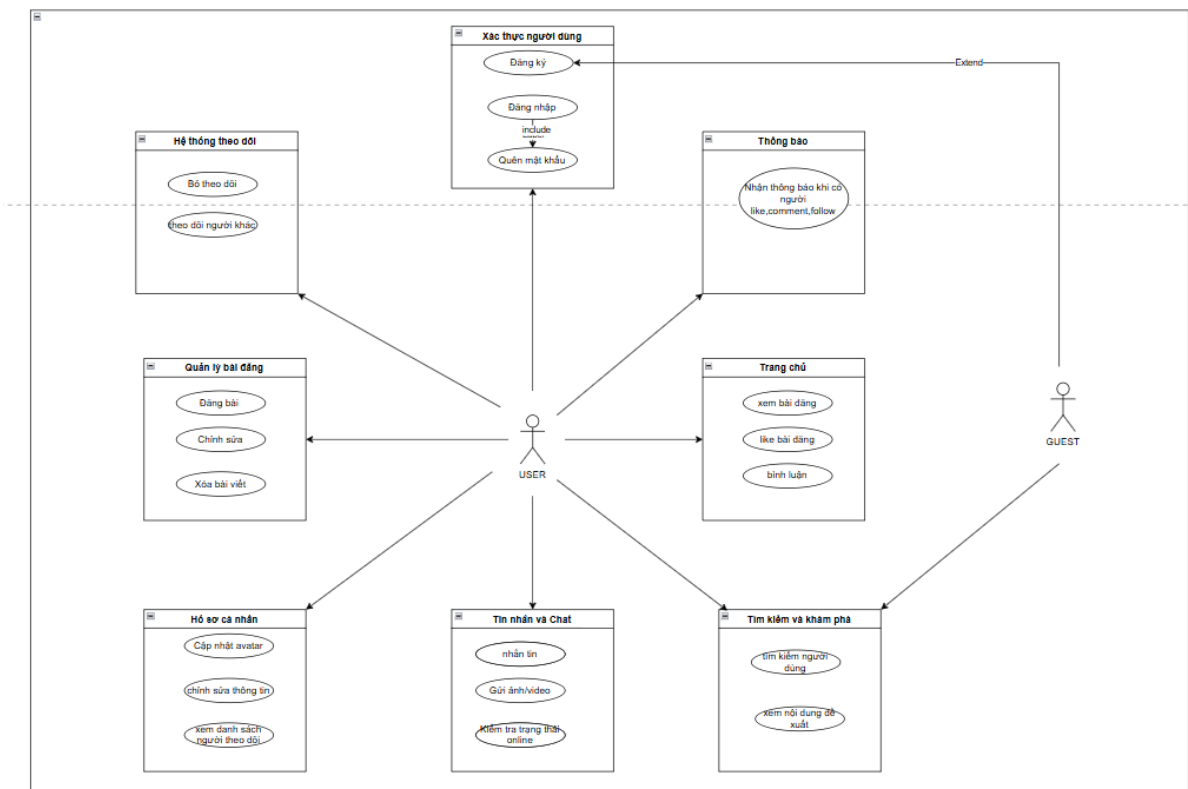
<<Bảng phân chia nhiệm vụ>>

## Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ

### 2.1. Kiến trúc hệ thống

- Firebase Authentication để xác thực người dùng.
- Firestore làm cơ sở dữ liệu chính để lưu trữ thông tin người dùng và bài đăng.
- Firebase Storage để lưu trữ hình ảnh/video.
- Firebase Cloud Messaging để xử lý thông báo đẩy.
- Realtime update với Firebase Realtime Database.

## Use Case Diagram



## Use Case Description – Đăng ký

Use Case Name: Đăng ký	ID: UC01	Importance level: Medium
Primary Actor: App User	Use Case Type: Detail, essential	

<b>Stackholders &amp; Interests:</b>
User – Muốn tạo tài khoản để sử dụng các tính năng của hệ thống.
<b>Brief Description:</b> Use case này mô tả cách người dùng tạo tài khoản mới trên hệ thống Instagram Clone bằng cách cung cấp thông tin cá nhân như tên, email, mật khẩu.
<b>Trigger:</b> User chọn chức năng “Đăng ký” trên trang chính của ứng dụng.
<b>Relationship:</b> <ul style="list-style-type: none"> <li>▪ Association: User</li> <li>▪ Include: (none)</li> <li>▪ Extend: (none)</li> <li>▪ Generalization: (none)</li> </ul>
<b>Action Step:</b> <ol style="list-style-type: none"> <li>1. Người dùng nhập các thông tin đăng ký gồm: tên, email, mật khẩu.</li> <li>2. Hệ thống kiểm tra thông tin đăng ký : <ul style="list-style-type: none"> <li>• Nếu thông tin hợp lệ, hệ thống tạo tài khoản mới.</li> <li>• Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.</li> </ul> </li> <li>3. Nếu đăng ký thành công hệ thống sẽ : <ul style="list-style-type: none"> <li>• Lưu tài khoản vào cơ sở dữ liệu.</li> <li>• Gửi email xác nhận cho người dùng.</li> <li>• Chuyển hướng người dùng đến trang chủ Instagram Clone.</li> </ul> </li> </ol>
<b>Exception:</b> <ul style="list-style-type: none"> <li>▪</li> </ul>
<b>Extension:</b>
<b>Pre-Condition:</b>
<b>Post-Condition:</b>

## Use Case Description – Đăng nhập

Use Case Name: Đăng nhập	ID: UC02	Importance level: Medium
Primary Actor: App User	Use Case Type: Detail, essential	
Stackholders & Interests:		
User – Muốn truy cập vào tài khoản của mình để sử dụng có tính năng của hệ thống.		

<b>Brief Description:</b> Use case này mô tả cách người dùng đăng nhập vào hệ thống Instagram Clone bằng tên đăng nhập và mật khẩu
<b>Trigger:</b> User chọn chức năng “Đăng nhập” trên trang chính của ứng dụng.
<b>Relationship:</b> <ul style="list-style-type: none"> <li>▪ Association: User</li> <li>▪ Include: (none)</li> <li>▪ Extend: (none)</li> <li>▪ Generalization: (none)</li> </ul>
<b>Action Step:</b> <ol style="list-style-type: none"> <li>1. Người dùng đăng nhập tên đăng nhập và mật khẩu.</li> <li>2. Hệ thống kiểm tra thông tin đăng nhập : <ul style="list-style-type: none"> <li>• Nếu thông tin hợp lệ, hệ thống cho phép người dùng truy cập vào trang chủ.</li> <li>• Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.</li> </ul> </li> <li>3. Nếu đăng nhập thành công hệ thống sẽ : <ul style="list-style-type: none"> <li>• Lưu phiên đăng nhập của người dùng.</li> <li>• Chuyển hướng người dùng đến trang chủ Instagram Clone.</li> <li>• Hiển thị thông báo chào mừng.</li> </ul> </li> </ol>
<b>Exception:</b> <ul style="list-style-type: none"> <li>▪</li> </ul>
<b>Extension:</b>
<b>Pre-Condition:</b>
<b>Post-Condition:</b>

## Use Case Description – Quên mật khẩu

<b>Use Case Name: Quên mật khẩu</b>	<b>ID: UC03</b>	<b>Importance level: Medium</b>
<b>Primary Actor: App User</b>	<b>Use Case Type:</b> Detail, essential	
<b>Stackholders &amp; Interests:</b>  User – Muốn khôi phục mật khẩu khi quên.		
<b>Brief Description:</b> Use case này mô tả cách người dùng khôi phục mật khẩu bằng cách nhận email đặt lại mật khẩu.		
<b>Trigger:</b> User chọn chức năng “Quên mật khẩu” trên trang chính của ứng dụng.		
<b>Relationship:</b>  ▪ Association: User		

<ul style="list-style-type: none"> <li>▪ <b>Include: (none)</b></li> <li>▪ <b>Extend: (none)</b></li> <li>▪ <b>Generalization: (none)</b></li> </ul>
<b>Action Step:</b>  <b>1.Người dùng nhập địa chỉ email đã đăng ký.</b>  <b>2.Hệ thống kiểm tra email :</b> <ul style="list-style-type: none"> <li>• Nếu email hợp lệ, hệ thống gửi email đặt lại mật khẩu.</li> <li>• Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.</li> </ul> <b>3.Người dùng truy cập vào email và nhấp vào liên kết đặt lại mật khẩu.</b>  <b>4.Người dùng nhập mật khẩu mới và xác nhận.</b>  <b>5.Hệ thống cập nhật mật khẩu mới và hiển thị thông báo thành công.</b>
<b>Exception:</b>  <ul style="list-style-type: none"> <li>▪</li> </ul>
<b>Extension:</b>
<b>Pre-Condition:</b>
<b>Post-Condition:</b>

## Use Case Description – Đăng bài

<b>Use Case Name: Đăng bài</b>	<b>ID: UC04</b>	<b>Importance level: Medium</b>
<b>Primary Actor: App User</b>	<b>Use Case Type:</b> Detail, essential	
<b>Stackholders &amp; Interests:</b>		
<b>User – Muốn đăng bài, hình ảnh lên Instagram Clone.</b>		
<b>Brief Description:</b> Use case này mô tả cách người dùng tạo bài đăng mới trên Instagram Clone.		
<b>Trigger:</b> User chọn chức năng “Đăng bài” trên giao diện của ứng dụng.		
<b>Relationship:</b>		
<div><ul style="list-style-type: none"><li>▪ <b>Association:</b> User</li><li>▪ <b>Include:</b> (none)</li><li>▪ <b>Extend:</b> (none)</li><li>▪ <b>Generalization:</b> (none)</li></ul></div>		
<b>Action Step:</b>		

<p>1.Người dùng chọn ảnh/video hoặc nội dung bài viết.</p> <p>2.Người dùng có thể thêm mô tả, gắn thẻ bạn bè.</p> <p>3.Người dùng nhấn nút “Đăng bài”.</p> <p>4.Hệ thống kiểm tra nội dung bài viết :</p> <ul style="list-style-type: none"> <li>• Nếu hợp lệ, hệ thống lưu bài viết vào cơ sở dữ liệu và hiển thị trên trang chủ.</li> <li>• Nếu không hợp lệ, hệ thống hiển thị thông báo lỗi.</li> </ul> <p>5.Nếu bài đăng thành công, hệ thống hiển thị thông báo xác nhận.</p>
<p><b>Exception:</b></p> <ul style="list-style-type: none"> <li>▪</li> </ul>
<p><b>Extension:</b></p>
<p><b>Pre-Condition:</b></p>
<p><b>Post-Condition:</b></p>

## Use Case Description – Chỉnh sửa bài đăng

Use Case Name: Chỉnh sửa bài đăng	ID: UC09	Importance level: Medium
Primary Actor: App User	Use Case Type: Detail, essential	
Stackholders & Interests:		
User – Muốn thay đổi nội dung bài đăng sau khi đăng tải.		
Brief Description: Use case này mô tả cách người dùng chỉnh sửa bài đăng của họ.		
Trigger: Người dùng chọn chỉnh sửa một bài đã đăng trước đó.		
Relationship:		
<ul style="list-style-type: none"><li>▪ Association: User</li><li>▪ Include: (none)</li><li>▪ Extend: (none)</li><li>▪ Generalization: (none)</li></ul>		
Action Step:		
1.Người dùng vào hồ sơ cá nhân hoặc trang chủ.		
2.Người dùng chọn một bài đang đã đăng trước đó.		
3.Người dùng nhấn vào tùy chọn “Chỉnh sửa bài đăng”.		

<p>4.Hệ thống hiển thị giao diện chỉnh sửa.</p> <p>5.Người dùng thay đổi nội dung, thêm hoặc xóa hình ảnh/video.</p> <p>6.Người dùng nhấn “Lưu thay đổi”.</p> <p>7.Hệ thống cập nhật bài đăng và hiển thị thông báo thành công.</p> <p>2.Hệ thống tạo thông báo cho người dùng liên quan.</p> <p>2.Hệ thống hiển thị thông báo trong mục “Thông báo”.</p> <p>3.Người dùng có thể nhấn vào thông báo để xem chi tiết.</p>
<p><b>Exception:</b></p> <ul style="list-style-type: none"> <li>▪ Nếu bài đăng không thuộc về người dùng, hệ thống không cho phép chỉnh sửa.</li> <li>▪ Nếu có lỗi mạng trong quá trình lưu, hệ thống hiển thị thông báo lỗi.</li> </ul>
<p><b>Extension:</b></p>
<p><b>Pre-Condition:</b></p>
<p><b>Post-Condition:</b></p>

## Use Case Description – Xóa bài đăng

Use Case Name: Xóa bài đăng	ID: UC09	Importance level: Medium
Primary Actor: App User	Use Case Type: Detail, essential	
Stackholders & Interests:		
User – Muốn xóa bài đăng không còn cần thiết.		
Brief Description: Use case này mô tả cách người dùng xóa một bài đăng khỏi trang cá nhân.		
Trigger: Người dùng chọn “Xóa bài viết” trong menu quản lý bài đăng.		
Relationship:		
<ul style="list-style-type: none"><li>▪ Association: User</li><li>▪ Include: (none)</li><li>▪ Extend: (none)</li><li>▪ Generalization: (none)</li></ul>		
Action Step:		
1.Người dùng truy cập bài đăng của mình.		



<p>2. Người dùng chọn “Xóa bài viết”.</p> <p>3. Hệ thống hiển thị cảnh báo “Xác nhận”.</p> <p>4. Nếu người dùng xác nhận, hệ thống xóa bài đăng khỏi cơ sở dữ liệu.</p> <p>5. Hệ thống hiển thị thông báo xác nhận xóa thành công.</p> <p>2. Hệ thống tạo thông báo cho người dùng liên quan.</p> <p>2. Hệ thống hiển thị thông báo trong mục “Thông báo”.</p> <p>3. Người dùng có thể nhấn vào thông báo để xem chi tiết.</p>
<p><b>Exception:</b></p> <ul style="list-style-type: none"> <li>▪</li> </ul>
<p><b>Extension:</b></p>
<p><b>Pre-Condition:</b></p>
<p><b>Post-Condition:</b></p>

## Use Case Description – Bình luận bài đăng

<b>Use Case Name:</b> Bình luận bài đăng	<b>ID:</b> UC05	<b>Importance level:</b> Medium
<b>Primary Actor:</b> App User	<b>Use Case Type:</b> Detail, essential	
<b>Stackholders &amp; Interests:</b>		
User – Muốn bình luận bài đăng của mình hoặc người khác.		
<b>Brief Description:</b> Use case này mô tả cách người dùng bình luận một bài đăng.		
<b>Trigger:</b> User nhấn vào một bài đăng và chọn chức năng “Bình luận”.		
<b>Relationship:</b>		
<div><div></div><div><ul style="list-style-type: none"><li>▪ <b>Association:</b> User</li><li>▪ <b>Include:</b> (none)</li><li>▪ <b>Extend:</b> (none)</li><li>▪ <b>Generalization:</b> (none)</li></ul></div></div>		
<b>Action Step:</b>		
1.Người dùng nhập nội dung bình luận.		
2.Người dùng nhấn nút “Gửi bình luận”		

<b>3.Hệ thống kiểm tra nội dung bình luận:</b> <ul style="list-style-type: none"> <li>Nếu hợp lệ, bình luận được đăng lên bài viết.</li> <li>Nếu không hợp lệ, hệ thống hiển thị thông báo lỗi.</li> </ul> <b>5.Nếu bình luận thành công, hệ thống hiển thị bình luận trên bài viết.</b>
<b>Exception:</b> <ul style="list-style-type: none"> <li></li> </ul>
<b>Extension:</b>
<b>Pre-Condition:</b>
<b>Post-Condition:</b>

## Use Case Description – Thích bài đăng

Use Case Name: Like bài đăng	ID: UC06	Importance level: Medium
Primary Actor: App User	Use Case Type: Detail, essential	
Stackholders & Interests:		
User – Muốn thể hiện sự yêu thích với đối với bài đăng.		
Brief Description: Use case này mô tả cách người “like” một bài đăng.		
Trigger: User nhấn vào biểu hiện “Thích” dưới bài đăng.		
Relationship:		
<ul style="list-style-type: none"><li>▪ Association: User</li><li>▪ Include: (none)</li><li>▪ Extend: (none)</li><li>▪ Generalization: (none)</li></ul>		
Action Step:		
1.Người dùng nhấn vào biểu tượng “Thích”.		
2.Hệ thống kiểm tra trạng thái like của người dùng trên bài đăng đó:		
<ul style="list-style-type: none"><li>• Nếu chưa like, hệ thống thêm lượt like vào bài đăng.</li><li>• Nếu đã like, hệ thống hủy like.</li></ul>		
3.Hệ thống cập nhật số lượt like và hiển thị thông tin mới.		
4.Nếu người dùng like bài đăng của người khác, hệ thống gửi thông báo đến chủ bài đăng.		

<b>Exception:</b>
▪
<b>Extension:</b>
<b>Pre-Condition:</b>
<b>Post-Condition:</b>

## Use Case Description – Theo dõi người dùng

<b>Use Case Name:</b> Theo dõi người dùng khác	<b>ID:</b> UC07	<b>Importance level:</b> Medium
<b>Primary Actor:</b> App User	<b>Use Case Type:</b> Detail, essential	
<b>Stackholders &amp; Interests:</b>		
User – Muốn theo dõi bài đăng của người khác.		
<b>Brief Description:</b> Use case này mô tả cách người dùng theo dõi người dùng khác trên Instagram Clone.		
<b>Trigger:</b> User nhấn vào nút “Theo dõi” trên hồ sơ của người khác.		
<b>Relationship:</b>		
<ul style="list-style-type: none"><li>▪ <b>Association:</b> User</li><li>▪ <b>Include:</b> (none)</li><li>▪ <b>Extend:</b> (none)</li><li>▪ <b>Generalization:</b> (none)</li></ul>		
<b>Action Step:</b>		
1.Người dùng vào trang cá nhân của người khác.		
2.Người dùng nhấn vào nút “Theo dõi”.		
2.Hệ thống kiểm tra trạng thái theo dõi:		
<ul style="list-style-type: none"><li>• Nếu chưa theo dõi, hệ thống thêm người dùng vào danh sách theo dõi.</li><li>• Nếu đã theo dõi, hệ thống hủy theo dõi.</li></ul>		
3.Hệ thống cập nhật số lượt like và hiển thị thông tin mới.		
4.Nếu người dùng bắt đầu theo dõi, hệ thống gửi thông báo đến người được theo dõi.		
<b>Exception:</b>		
<ul style="list-style-type: none"><li>▪</li></ul>		

<b>Extension:</b>
<b>Pre-Condition:</b>
<b>Post-Condition:</b>

## Use Case Description – Nhắn tin trực tiếp

Use Case Name: Nhắn tin trực tiếp	ID: UC08	Importance level: Medium
Primary Actor: App User	Use Case Type: Detail, essential	
Stackholders & Interests:		
User – Muốn trò chuyện riêng với người khác.		
Brief Description: Use case này mô tả cách người dùng gửi tin nhắn trực tiếp trên Instagram Clone.		
Trigger: User nhấn vào hộp thư và chọn người để nhắn tin.		
Relationship:		
<ul style="list-style-type: none"><li>▪ Association: User</li><li>▪ Include: (none)</li><li>▪ Extend: (none)</li><li>▪ Generalization: (none)</li></ul>		
Action Step:		
<p>1.Người dùng vào phần tin nhắn và chọn một cuộc trò chuyện.</p> <p>2.Người dùng nhập nội dung tin nhắn và nhấn “Gửi”.</p> <p>2.Hệ thống kiểm tra tin nhắn:</p> <ul style="list-style-type: none"><li>• Nếu hợp lệ, tin nhắn được gửi thành công.</li><li>• Nếu không hợp lệ, hệ thống hiển thị thông báo lỗi.</li></ul> <p>3.Hệ thống hiển thị thông báo trong cuộc trò chuyện.</p> <p>4.Hệ thống gửi thông báo đến người nhận tin nhắn.</p>		
Exception:		
<ul style="list-style-type: none"><li>▪</li></ul>		
Extension:		
Pre-Condition:		

**Post-Condition:**

## Use Case Description – Nhận thông báo

Use Case Name: Nhận thông báo	ID: UC09	Importance level: Medium
Primary Actor: App User	Use Case Type: Detail, essential	
Stackholders & Interests:		
User – Muốn nhận thông báo khi có tương tác với bài đăng hoặc tài khoản của mình.		
Brief Description: Use case này mô tả cách người dùng nhận thông báo từ hệ thống.		
Trigger: Hệ thống phát hiện có sự kiện liên quan đến người dùng(like,comment, follow, tin nhắn).		
Relationship:		
<ul style="list-style-type: none"><li>▪ Association: User</li><li>▪ Include: (none)</li><li>▪ Extend: (none)</li><li>▪ Generalization: (none)</li></ul>		
Action Step:		
<div>1.Một sự kiện xảy ra ( ví dụ: ai đó like bài viết, comment, gửi tin nhắn).</div> <div>2.Hệ thống tạo thông báo cho người dùng liên quan.</div> <div>2.Hệ thống hiển thị thông báo trong mục “Thông báo”.</div> <div>3.Người dùng có thể nhấn vào thông báo để xem chi tiết.</div>		
Exception:		
<ul style="list-style-type: none"><li>▪</li></ul>		
Extension:		
Pre-Condition:		
Post-Condition:		

# Use Case Description – Tìm kiếm người dùng

Use Case Name: Tìm kiếm người dùng	ID: UC010	Importance level: High
Primary Actor: App User, Guest	Use Case Type: Detail, essential	
Stackholders & Interests:		
User,Guest – Muốn tìm kiếm tài khoản khác để theo dõi hoặc tương tác		
Brief Description: Use case này mô tả cách người dùng hoặc khách tìm kiếm tài khoản Instagram khác		
Trigger: Người dùng nhập tên hoặc từ khóa vào thanh tìm kiếm.		
Relationship:		
<ul style="list-style-type: none"><li>▪ Association: User,Guest</li><li>▪ Include: (none)</li><li>▪ Extend: (none)</li><li>▪ Generalization: (none)</li></ul>		
Action Step:		
<p>1.Người dùng truy cập trang “Tìm kiếm”.</p> <p>2.Người dùng nhập từ khóa tìm kiếm.</p> <p>3.Hệ thống hiển thị danh sách kết quả phù hợp.</p> <p>4.Người dùng chọn một tài khoản từ danh sách.</p> <p>5.Hệ thống chuyển hướng đến trang hồ sơ của tài khoản đó.</p>		
Exception:		
<ul style="list-style-type: none"><li>▪</li></ul>		
Extension:		
Pre-Condition:		
Post-Condition:		

## Use Case Description – Xem nội dung đề xuất

Use Case Name: Xem nội dung đề xuất	ID: UC11	Importance level: Medium
Primary Actor: App User,Guest	Use Case Type: Detail, essential	
Stackholders & Interests:  User,Guest – Muốn khám phá nội dung mới từ hệ thống.		
Brief Description: Use case này mô tả cách người dùng xem danh sách các bài đăng được đề xuất dựa trên sở thích và tương tác.		
Trigger: Người dùng truy cập tab “Khám phá”.		
Relationship:  <ul style="list-style-type: none"><li>▪ Association: User,Guest</li><li>▪ Include: (none)</li><li>▪ Extend: (none)</li><li>▪ Generalization: (none)</li></ul>		
Action Step:  1.Người dùng truy cập trang “Khám phá”.  2.Hệ thống lấy danh sách bài đăng dựa trên sở thích và xu hướng.  3.Hệ thống hiển thị danh sách bài đăng được đề xuất.  4.Người dùng có thể tương tác với bài đăng.		
Exception:  <ul style="list-style-type: none"><li>▪</li></ul>		
Extension:		
Pre-Condition:		
Post-Condition:		

## Use Case Description – Cập nhật Avatar

Use Case Name: Cập nhật Avatar	ID: UC12	Importance level: Medium
--------------------------------	----------	--------------------------

<b>Primary Actor: App User</b>	<b>Use Case Type:</b> Detail, essential
<b>Stackholders &amp; Interests:</b>	
<b>User – Muốn cập nhật ảnh đại diện cá nhân.</b>	
<b>Brief Description:</b> Use case này mô tả cách người dùng thay đổi ảnh đại diện trên Instagram Clone.	
<b>Trigger:</b> Người dùng truy cập hồ sơ cá nhân và chọn “Cập nhật ảnh đại diện”.	
<b>Relationship:</b> <ul style="list-style-type: none"> <li>▪ <b>Association:</b> User</li> <li>▪ <b>Include:</b> (none)</li> <li>▪ <b>Extend:</b> (none)</li> <li>▪ <b>Generalization:</b> (none)</li> </ul>	
<b>Action Step:</b> <ol style="list-style-type: none"> <li>1. Người dùng vào trang hồ sơ cá nhân.</li> <li>2. Người dùng chọn “Cập nhật ảnh đại diện”.</li> <li>3. Người dùng tải lên ảnh mới hoặc chọn từ thư viện có sẵn.</li> <li>4. Hệ thống kiểm tra và cập nhật ảnh đại diện.</li> <li>5. Hệ thống hiển thị ảnh đại diện mới trên hồ sơ.</li> </ol>	
<b>Exception:</b>	
▪	
<b>Extension:</b>	
<b>Pre-Condition:</b>	
<b>Post-Condition:</b>	

## Use Case Description – Chỉnh sửa thông tin cá nhân

<b>Use Case Name:</b> Chỉnh sửa thông tin cá nhân	<b>ID:</b> UC13	<b>Importance level:</b> Medium
<b>Primary Actor:</b> App User	<b>Use Case Type:</b> Detail, essential	



<b>Stackholders &amp; Interests:</b>
User – Muốn chỉnh sửa thông tin cá nhân như tên, bio, liên kết.
<b>Brief Description:</b> Use case này mô tả cách người dùng chỉnh sửa thông tin cá nhân trên trang hồ sơ.
<b>Trigger:</b> Người dùng truy cập trang hồ sơ và chọn “Chỉnh sửa thông tin”.
<b>Relationship:</b> <ul style="list-style-type: none"> <li>▪ Association: User</li> <li>▪ Include: (none)</li> <li>▪ Extend: (none)</li> <li>▪ Generalization: (none)</li> </ul>
<b>Action Step:</b> <ol style="list-style-type: none"> <li>1. Người dùng vào trang hồ sơ.</li> <li>2. Người dùng chọn “Chỉnh sửa thông tin”.</li> <li>3. Người dùng nhập thông tin mới (tên, bio, liên kết).</li> <li>4. Người dùng nhấn “Lưu thay đổi”.</li> <li>5. Hệ thống cập nhật thông tin và hiển thị trên trang cá nhân.</li> <li>2. Hệ thống tạo thông báo cho người dùng liên quan.</li> <li>2. Hệ thống hiển thị thông báo trong mục “Thông báo”.</li> <li>3. Người dùng có thể nhấn vào thông báo để xem chi tiết.</li> </ol>
<b>Exception:</b> <ul style="list-style-type: none"> <li>▪</li> </ul>
<b>Extension:</b>
<b>Pre-Condition:</b>
<b>Post-Condition:</b>

# Use Case Description – Xem danh sách người theo dõi

Use Case Name: Xem danh sách người theo dõi	ID: UC09	Importance level: Medium
Primary Actor: App User	Use Case Type: Detail, essential	
Stackholders & Interests:		
User – Muốn xem danh sách những người đang theo dõi mình hoặc danh sách những người mình đang theo dõi.		
Brief Description: Use case này mô tả cách người dùng truy cập danh sách người theo trên hồ sơ cá nhân		
Trigger: Người dùng truy cập hồ sơ cá nhân và chọn danh sách người theo dõi.		
Relationship:		
<ul style="list-style-type: none"><li>▪ Association: User</li><li>▪ Include: (none)</li><li>▪ Extend: (none)</li><li>▪ Generalization: (none)</li></ul>		
Action Step:		
<p>1.Người dùng vào trang hồ sơ cá nhân.</p> <p>2.Người dùng nhấn vào mục”Người theo dõi” hoặc “Người đang theo dõi”.</p> <p>3.Hệ thống hiển thị danh sách người theo dõi hoặc danh sách người dùng mà họ đang theo dõi.</p> <p>4.Người dùng có thể nhấn vào từng tài khoản để xem hồ sơ chi tiết.</p>		
Exception:		
<ul style="list-style-type: none"><li>▪</li></ul>		
Extension:		
Pre-Condition:		
Post-Condition:		

## 2.2. Giới thiệu về Công nghệ phát triển

- **Phát triển ứng dụng:** Android Studio với Kotlin.
- **Backend & Cơ sở dữ liệu:** Firebase (Firestore, Firebase Authentication, Firebase Storage, Firebase Cloud Messaging).
- **Realtime & Thông báo:** Firebase Realtime Database, WebSockets.

## **Chương 3. XÂY DỰNG ỨNG DỤNG**

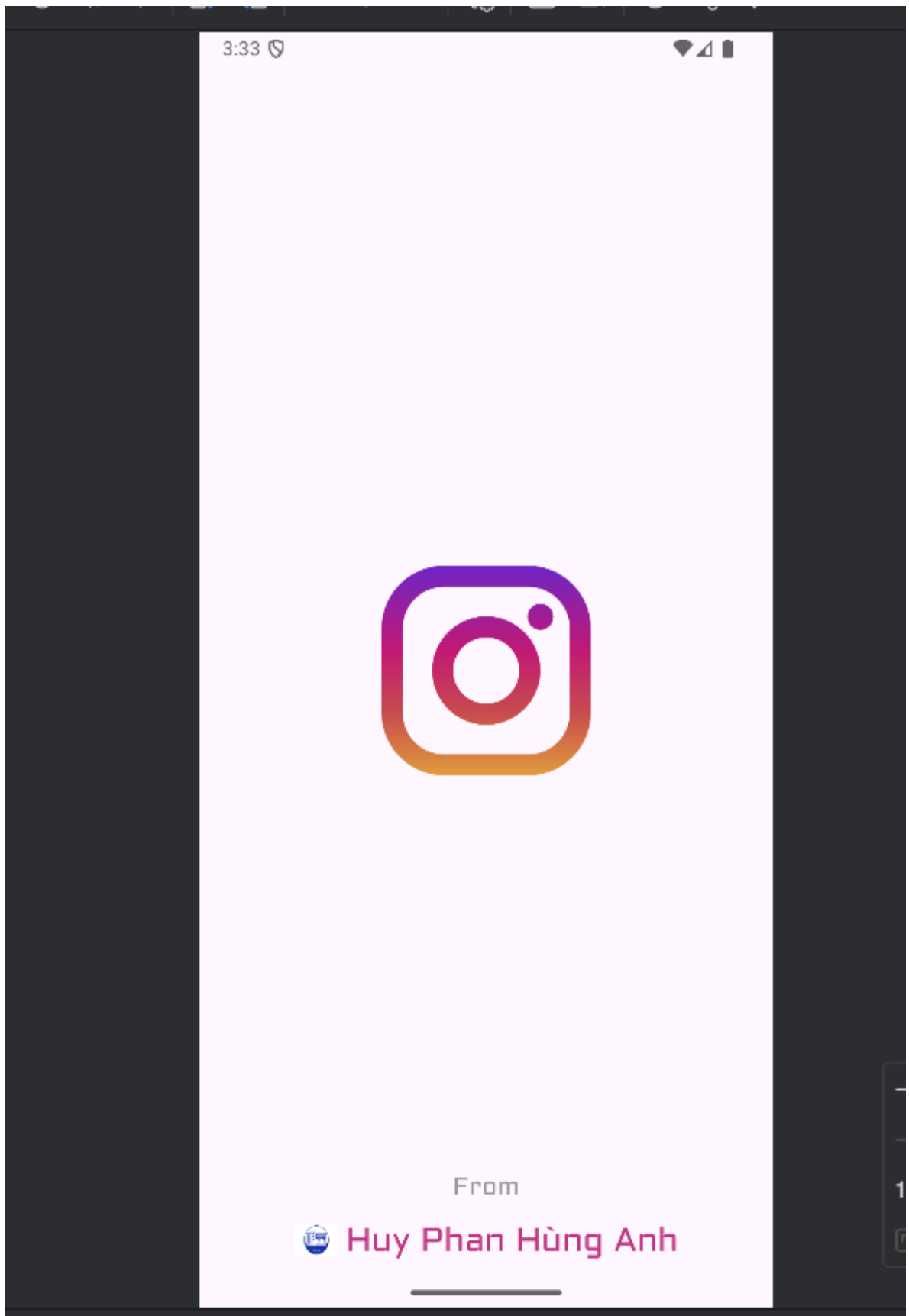
### **3.1. Thiết kế Figma**

<https://www.figma.com/design/SMIs3Pu9cBe5zfGru6vHQ/Instagram-UI-Screens?node-id=0-2&p=f&t=F21YZ17yJ1yEvzgo-0>

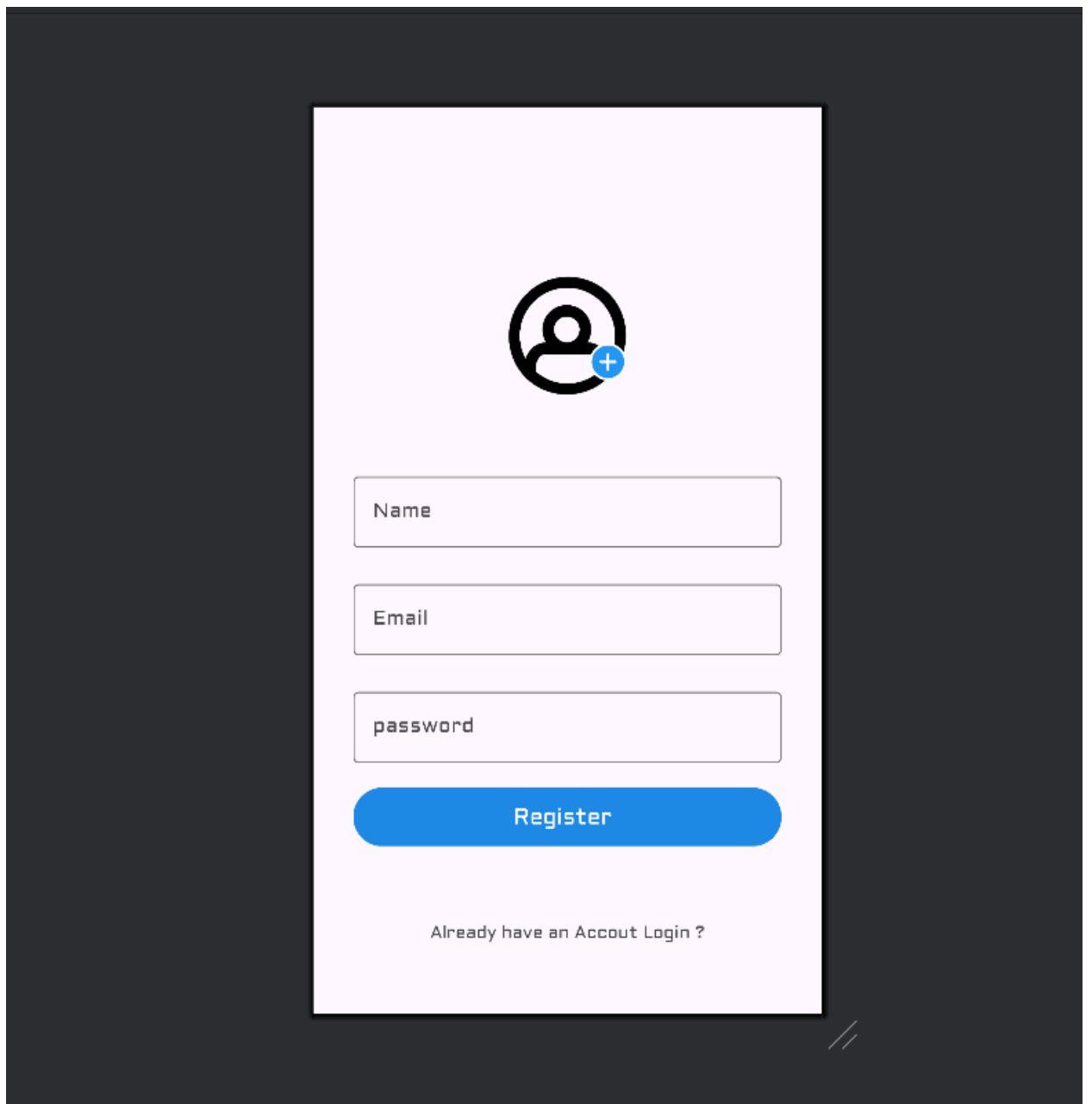
### **3.2. Thiết kế CSDL**

<<Lược đồ>>

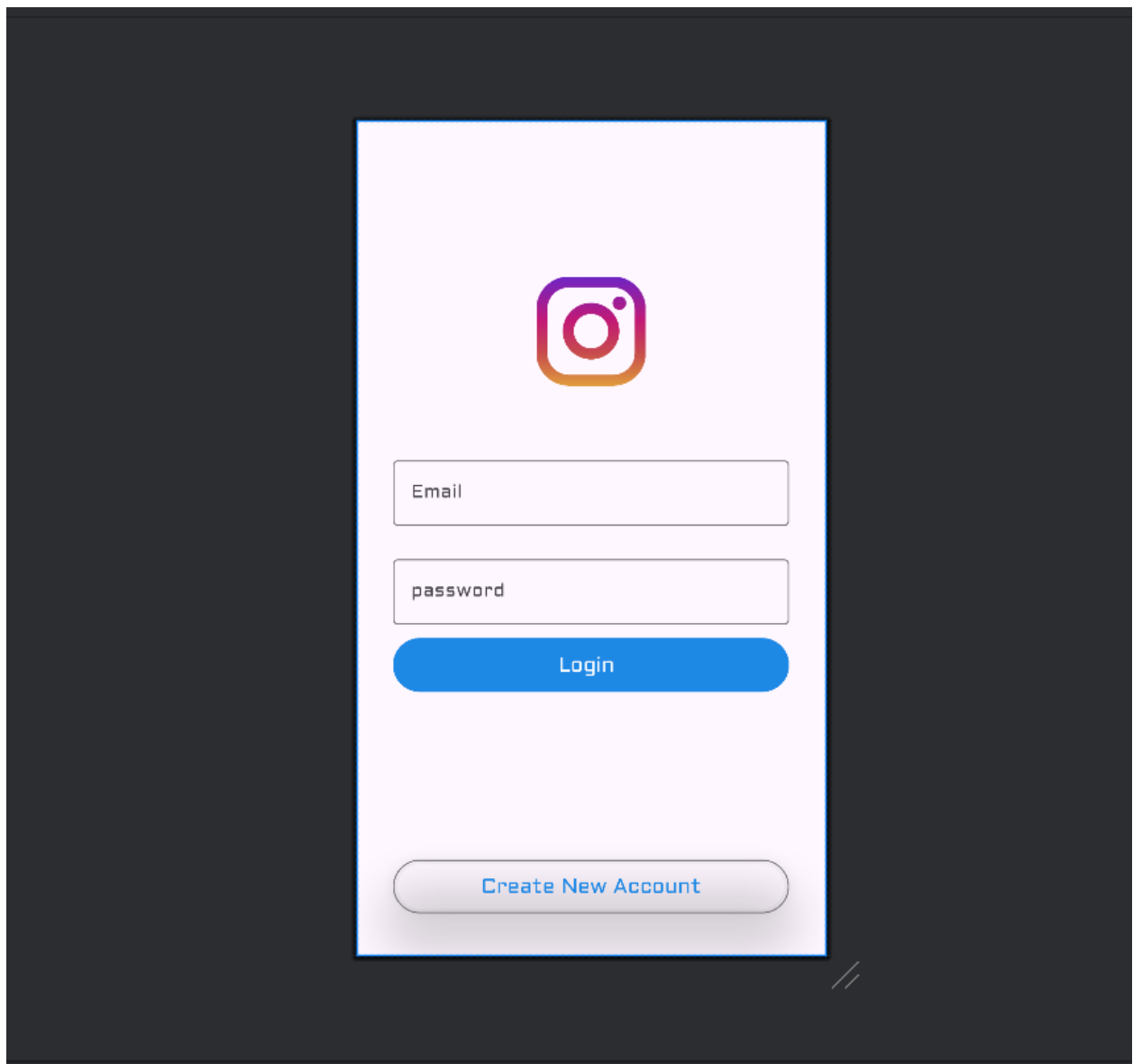
### **3.2. Giao diện ứng dụng**



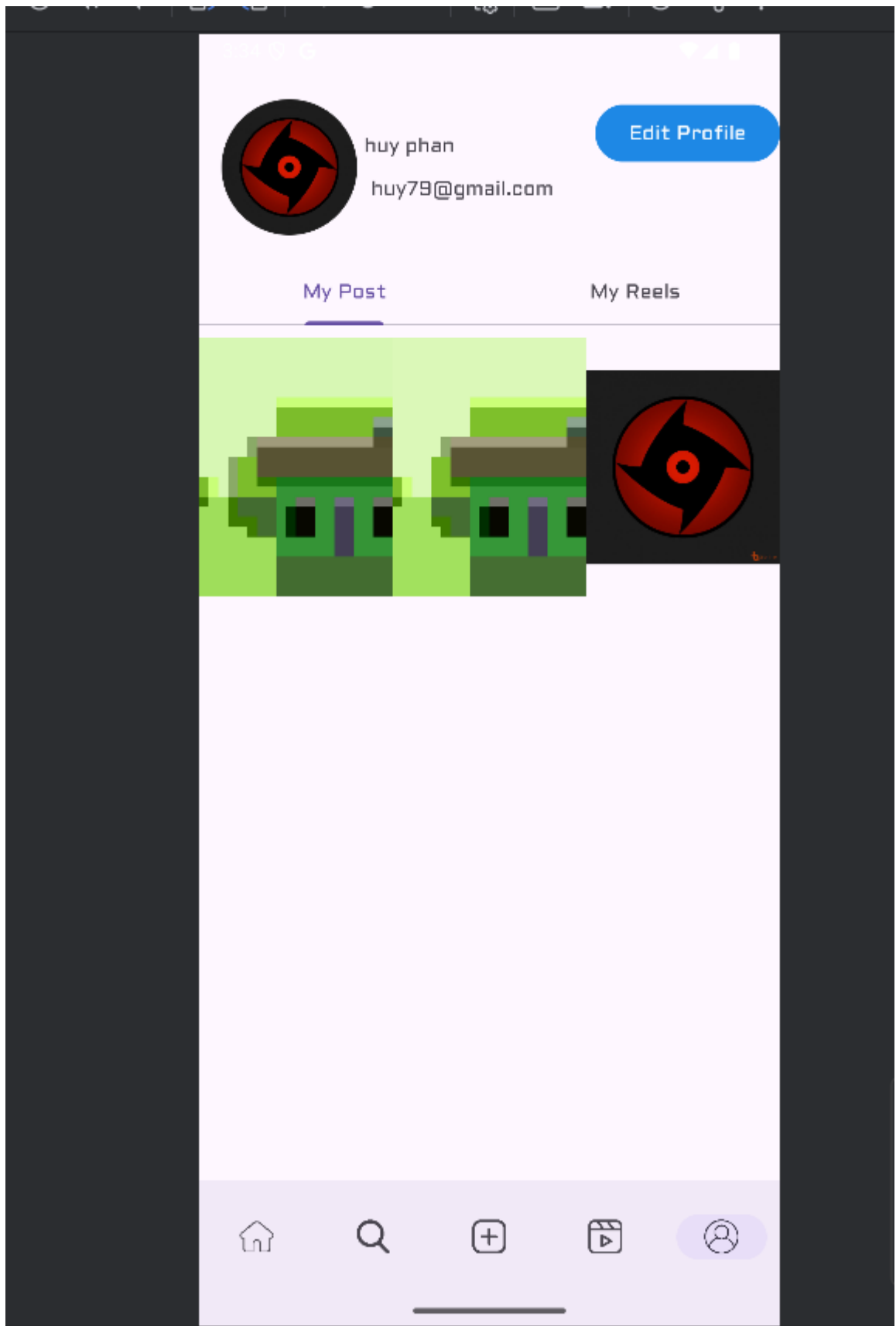
### Giao diện đăng ký người dùng



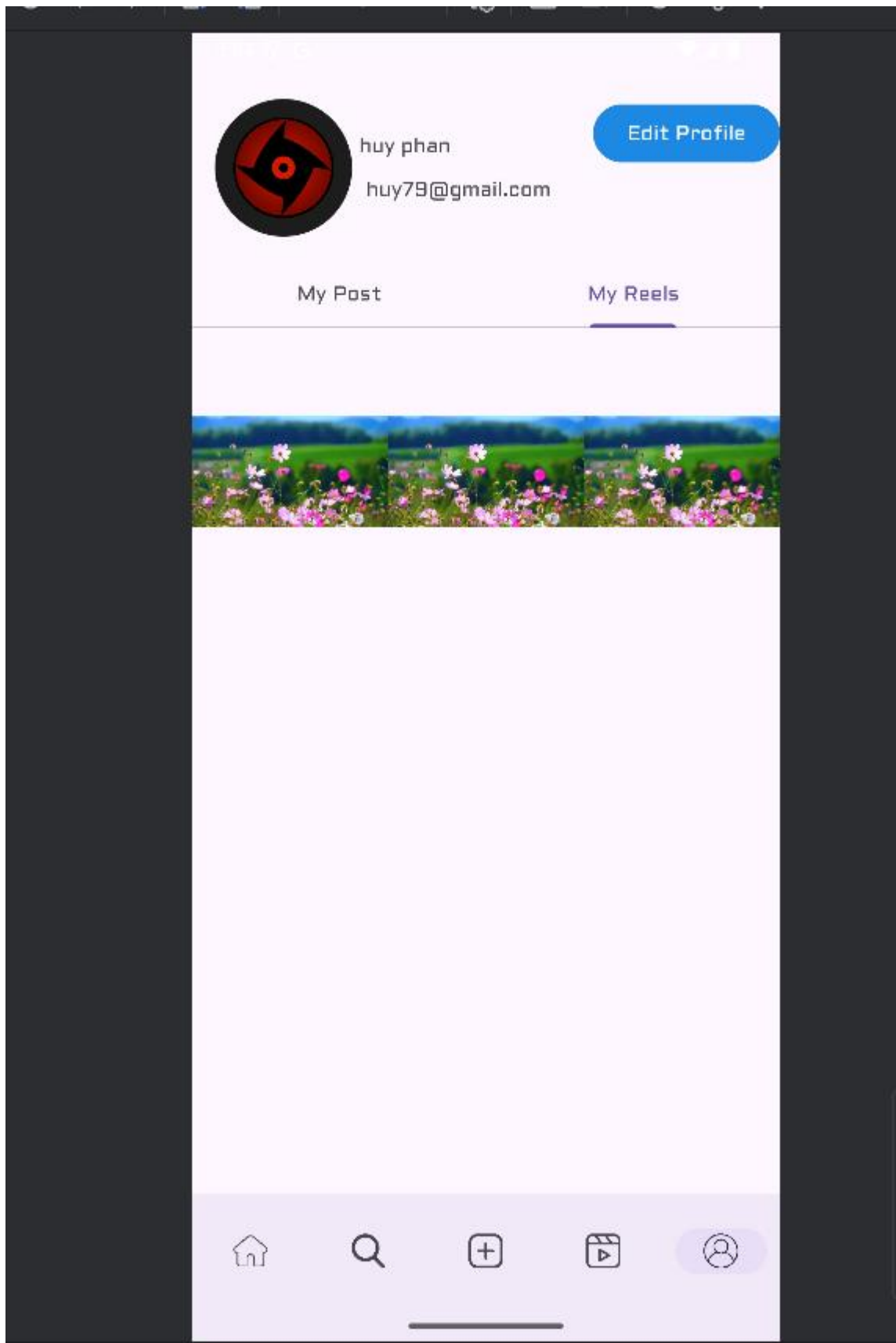
## Giao diện đăng nhập

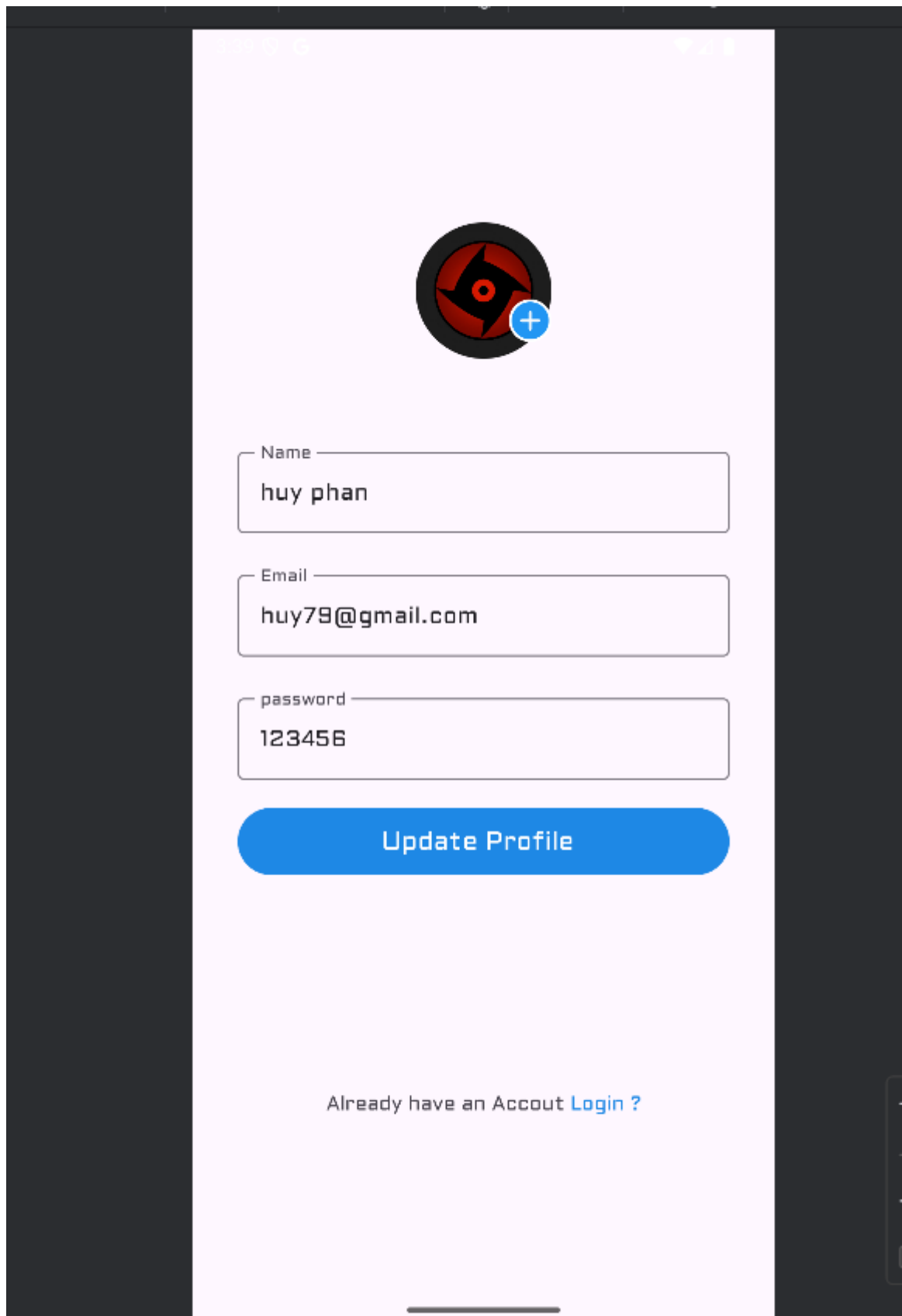


**Giao diện thông tin hồ sơ người dùng**

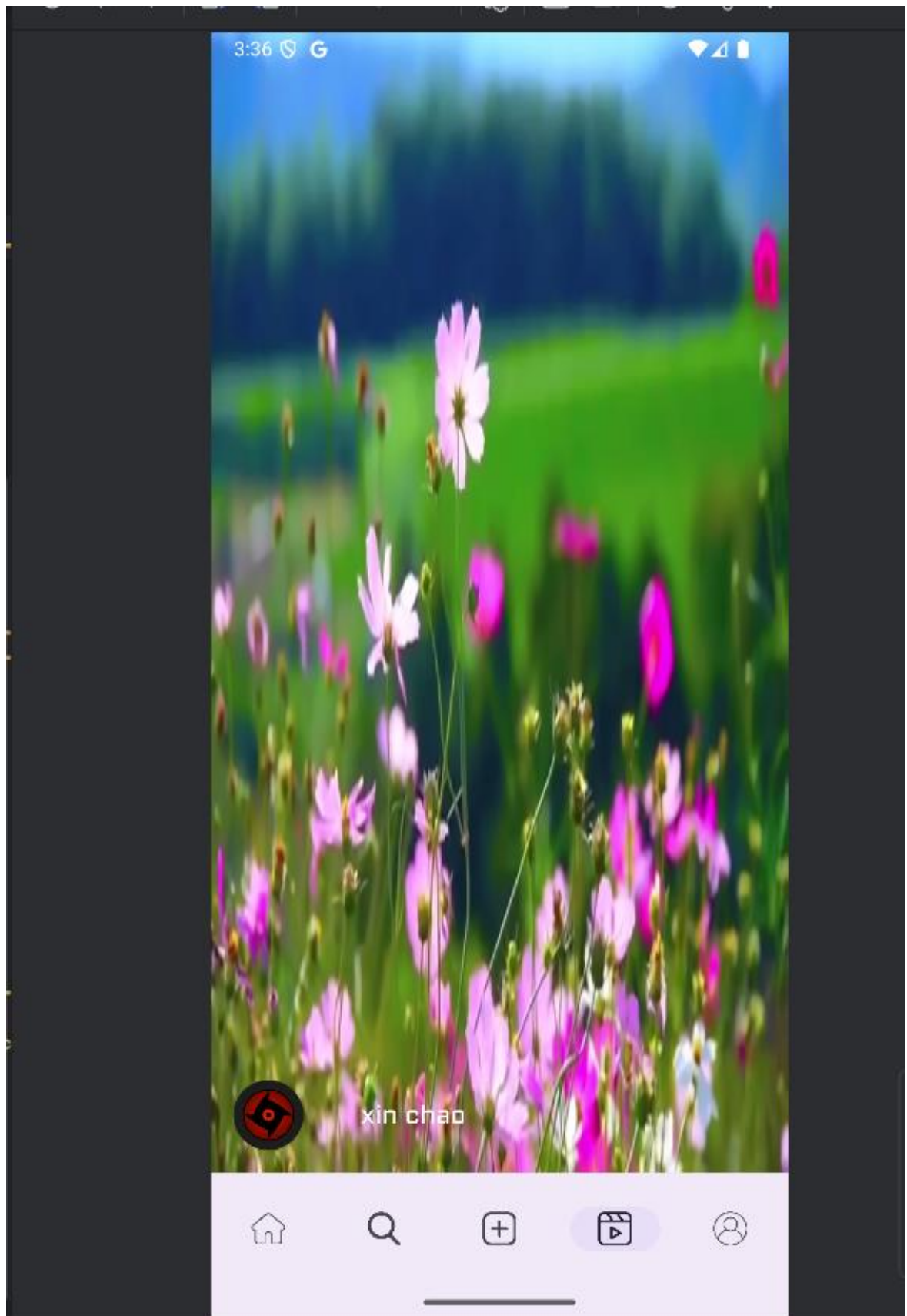




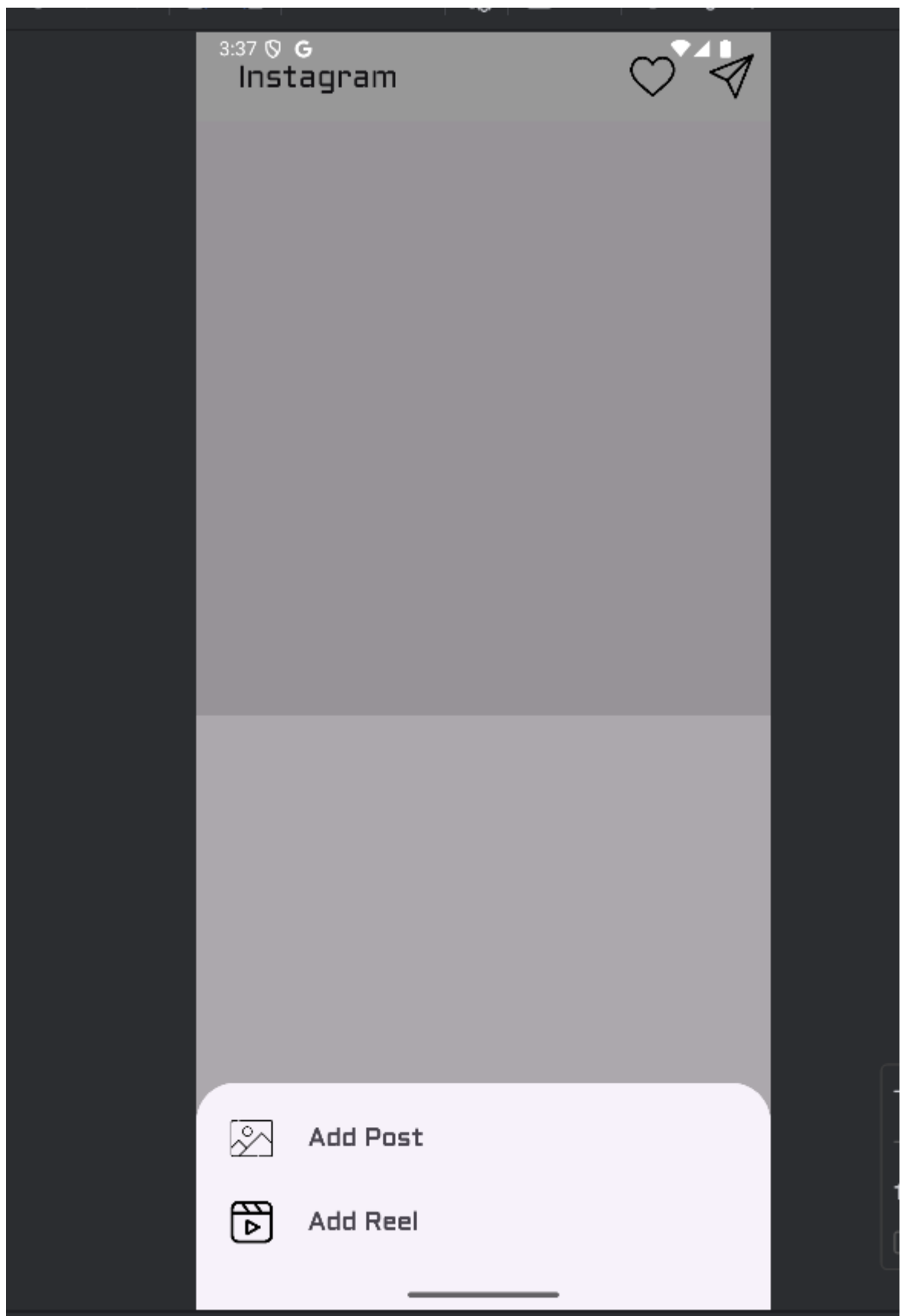




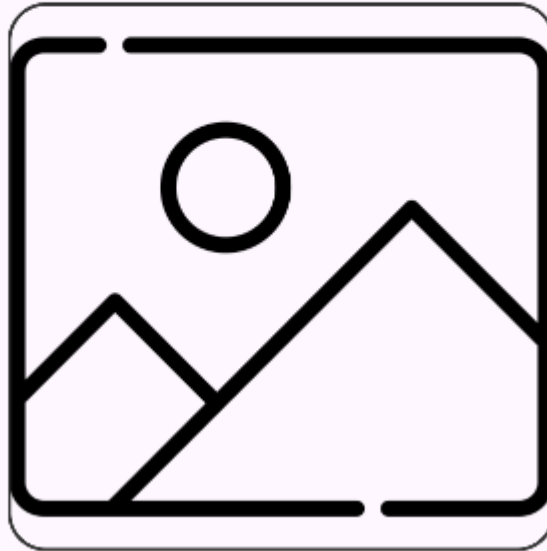
Giao diện các bài đăng Video



**Giao diện đăng bài**



← New Post



Write a caption



**Share To Post**

Your Host May be Appear in Posts and can be seen on the posts tab under your profile

Cancel

Post

← New Reel

Select Reel

Write a caption



Share To Reel

Your Host May be Appear in Posts and can be seen on the posts tab under your profile

Cancel

Post

### 3.4. Code minh họa các chức năng cốt lõi

#### activity\_sign\_up.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SignUpActivity">

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/profile_image"
        android:layout_width="96dp"
        android:layout_height="96dp"
        android:layout_marginTop="136dp"
        android:src="@drawable/user_icon"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/addImage"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:src="@drawable/plus"
        app:civ_border_color="@color/white"
        app:civ_border_width="2dp"
        app:layout_constraintBottom_toBottomOf="@+id/profile_image"
        app:layout_constraintEnd_toEndOf="@+id/profile_image"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="@+id/profile_image"
        app:layout_constraintTop_toTopOf="@+id/profile_image"
        app:layout_constraintVertical_bias="0.82" />

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="60dp"
        android:layout_marginEnd="32dp"
        android:hint="Name"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/profile_image">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="visible" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="24dp"
```

```

        android:layout_marginEnd="32dp"
        android:hint="Email"
        android:visibility="visible"
        app:layout_constraintEnd_toEndOf="@+id/name"
        app:layout_constraintStart_toStartOf="@+id/name"
        app:layout_constraintTop_toBottomOf="@+id/name">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="visible" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="24dp"
        android:layout_marginEnd="32dp"
        android:hint="password"
        app:layout_constraintEnd_toEndOf="@+id/email"
        app:layout_constraintStart_toStartOf="@+id/email"
        app:layout_constraintTop_toBottomOf="@+id/email">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="visible" />

    </com.google.android.material.textfield.TextInputLayout>

    <Button
        android:id="@+id/signUpBtn"
        style="@style/Widget.Material3.Button.Icon"
        android:layout_width="0dp"
        android:layout_height="55dp"
        android:layout_marginTop="16dp"
        android:backgroundTint="@color/blue"
        android:text="Register"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="@+id/password"
        app:layout_constraintStart_toStartOf="@+id/password"
        app:layout_constraintTop_toBottomOf="@+id/password" />

    <TextView
        android:id="@+id/login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Already have an Account Login ?"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/signUpBtn" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

## SignUpActivity.kt

```
package com.neatroots.instagramclone
```



```

import Models.User
import android.content.Intent
import android.os.Bundle
import android.text.Html
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.google.firebase.Firebase
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.auth
import com.google.firebase.firestore.firestore
import com.google.firebase.firestore.toObject
import com.neatroots.instagramclone.databinding.ActivitySignUpBinding
import com.neatroots.instagramclone.utils.USER_NODE
import com.neatroots.instagramclone.utils.USER_PROFILE_FOLDER
import com.neatroots.instagramclone.utils.uploadImage
import com.squareup.picasso.Picasso

class SignUpActivity : AppCompatActivity() {
    val binding by lazy {
        ActivitySignUpBinding.inflate(layoutInflater)
    }
    lateinit var user: User
    private val
launcher=registerForActivityResult (ActivityResultContracts.GetContent()) {
    uri->
    uri?.let {
        uploadImage(uri, USER_PROFILE_FOLDER) {
            if(it!=null) {
                user.image=it
                binding.profileImage.setImageURI(uri)
            }
        }
    }
}
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(binding.root)
        val text = "<font color=#FF000000>Already have an Account</font> <font
color=#1E88E5>Login ?</font>"
        binding.login.setText(Html.fromHtml(text))
        user = User()
        if (intent.hasExtra("MODE")) {
            if (intent.getIntExtra("MODE",-1)==1) {
                binding.signUpBtn.text="Update Profile"
            }
        }
        Firebase.firestore.collection(USER_NODE).document(Firebase.auth.currentUser!!.
        .uid).get().addOnSuccessListener {
            user=it.toObject<User>()!!
            if (!user.image.isNullOrEmpty()) {
                Picasso.get().load(user.image).into(binding.profileImage)
            }
            binding.name.editText?.setText(user.name)
            binding.email.editText?.setText(user.email)
            binding.password.editText?.setText(user.password)
        }
    }
    binding.signUpBtn.setOnClickListener {
        if (intent.hasExtra("MODE")) {

```

```

        if (intent.getIntExtra("MODE",-1)==1){

Firebase.firestore.collection(USER_NODE).document(Firebase.auth.currentUser!!.uid).set(user)

                .addOnSuccessListener {
                    startActivity(
                        Intent(
                            this@SignUpActivity,
                            HomeActivity::class.java
                        )
                    )
                    finish()
                }
            }

        }else {

            if (binding.name.editText?.text.toString().equals("") or
                binding.email.editText?.text.toString().equals("") or
                binding.password.editText?.text.toString().equals("")
            ) {

                Toast.makeText(
                    this@SignUpActivity,
                    "Please fill all the Information",
                    Toast.LENGTH_SHORT
                ).show()
            } else {

FirebaseAuth.getInstance().createUserWithEmailAndPassword(
                binding.email.editText?.text.toString(),
                binding.password.editText?.text.toString()
            ).addOnCompleteListener { result ->

                if (result.isSuccessful) {
                    user.name =
binding.name.editText?.text.toString()
                    user.email =
binding.email.editText?.text.toString()
                    user.password =
binding.password.editText?.text.toString()
                    Firebase.firestore.collection(USER_NODE)

                .document(Firebase.auth.currentUser!!.uid).set(user)
                        .addOnSuccessListener {
                            startActivity(
                                Intent(
                                    this@SignUpActivity,
                                    HomeActivity::class.java
                                )
                            )
                            finish()
                        }
                    }
                } else {
                    Toast.makeText(
                        this@SignUpActivity,
                        result.exception?.localizedMessage,
                        Toast.LENGTH_SHORT
                    ).show()
                }
            }
        }
    }
}

```

```

    }
    binding.addImage.setOnClickListener{
        launcher.launch("image/*")
    }
    binding.login.setOnClickListener {
        startActivity(Intent(this@SignUpActivity,
LoginActivity::class.java))
        finish()
    }
}
}
}

```

## activity\_login.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".LoginActivity">

    <ImageView
        android:id="@+id/profile_image"
        android:layout_width="96dp"
        android:layout_height="96dp"
        android:layout_marginTop="136dp"
        android:src="@drawable/instagram"

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="60dp"
        android:layout_marginEnd="32dp"
        android:hint="Email"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/profile_image">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="visible" />

    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/pass"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="24dp"
        android:layout_marginEnd="32dp"

```

```

        android:hint="password"
        android:visibility="visible"
        app:layout_constraintEnd_toEndOf="@+id/email"
        app:layout_constraintStart_toStartOf="@+id/email"
        app:layout_constraintTop_toBottomOf="@+id/email">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="visible" />

    </com.google.android.material.textfield.TextInputLayout>

    <Button
        android:id="@+id/login_btn"
        style="@style/Widget.Material3.Button.Icon"
        android:layout_width="0dp"
        android:layout_height="55dp"
        android:layout_marginTop="8dp"
        android:backgroundTint="@color/blue"
        android:text="Login"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="@+id/pass"
        app:layout_constraintStart_toStartOf="@+id/pass"
        app:layout_constraintTop_toBottomOf="@+id/pass" />

    <Button
        style="@style/Widget.Material3.Button.OutlinedButton"
        android:layout_width="0dp"
        android:layout_height="55dp"
        android:layout_marginBottom="32dp"
        android:elevation="20dp"
        android:shadowColor="@android:color/transparent"
        android:text="Create New Account"
        android:textColor="@color/blue"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="@+id/login_btn"
        app:layout_constraintStart_toStartOf="@+id/login_btn" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

## LoginActivity.kt

```

package com.neatroots.instagramclone

import Models.User
import android.content.Intent
import android.os.Bundle
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.auth.FirebaseAuth
import com.neatroots.instagramclone.databinding.ActivityLoginBinding

class LoginActivity : AppCompatActivity() {
    private lateinit var binding: ActivityLoginBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        binding = ActivityLoginBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
}

```

```

        binding.loginBtn.setOnClickListener {
            val email = binding.email.editText?.text.toString().trim()
            val password = binding.pass.editText?.text.toString().trim()

            if (email.isEmpty() || password.isEmpty()) {
                Toast.makeText(this, "Please fill all the details",
                    Toast.LENGTH_SHORT).show()
            } else {
                val user = User(email, password)

                FirebaseAuth.getInstance().signInWithEmailAndPassword(user.email!!,
                    user.password!!)
                    .addOnCompleteListener { task ->
                        if (task.isSuccessful) {
                            startActivity(Intent(this,
                                HomeActivity::class.java))
                            finish()
                        } else {
                            Toast.makeText(this,
                                task.exception?.localizedMessage, Toast.LENGTH_SHORT).show()
                        }
                    }
            }
        }
    }
}

```

## KẾT LUẬN

1. Kết quả đạt được
2. Nhược điểm
3. Hướng phát triển

## TÀI LIỆU THAM KHẢO

- [1] David Flanagan, JavaScript: The Definitive Guide, 7th Edition, O'Reilly Media, 2020.
- [2] Adam Freeman, “Pro jQuery”, Apress, 2018.
- [3] Benjamin Jakobus, “Mastering Bootstrap 5”, Packt Publishing, 2018.