**CHƯƠNG 1. GIỚI THIỆU DỰ ÁN**

**1.1 Quản lý cho thuê phòng**

Ứng dụng Quản lý và Cho thuê phòng được thiết kế tỉ mỉ nhằm hỗ trợ việc quản lý dễ dàng và hiệu quả, đồng thời tạo sự kết nối giữa chủ nhà và người thuê. Hệ thống này cho phép quản lý thông tin phòng, hỗ trợ tìm kiếm, đặt lịch xem phòng, và thực hiện giao dịch thuê một cách minh bạch và tiện lợi.

**1.2 Phạm vi hệ thống**

Hệ thống Quản lý và Cho thuê phòng bao gồm các thành phần chính sau:

1. Đăng nhập và Quản lý tài khoản
2. Đăng tin cho thuê phòng
3. Tìm kiếm và lọc phòng
4. Quản lý lịch hẹn
5. Hệ thống đặt cọc và thanh toán
6. Hỗ trợ người dùng

**1.3 Mô tả chi tiết hệ thống**

Hệ thống được phát triển với 17 tính năng chính, phục vụ hai nhóm người dùng chính: Chủ nhà và Người thuê.  
**Các tính năng chính:**  
*Đối với Chủ nhà:*

* Đăng tin cho thuê: Cho phép chủ nhà đăng thông tin chi tiết về phòng, bao gồm hình ảnh, giá thuê, địa chỉ và tiện nghi.
* Quản lý phòng trống: Cập nhật và quản lý trạng thái phòng để tránh đặt trùng.
* Xác nhận thông tin: Xác nhận thông tin với người thuê để đảm bảo tính chính xác.
* Quản lý hợp đồng: Tạo và quản lý hợp đồng thuê một cách dễ dàng.
* Quản lý thanh toán: Theo dõi và nhận thanh toán từ người thuê.

*Đối với Người thuê:*

* Tìm kiếm phòng: Tìm kiếm nhanh dựa trên các tiêu chí như vị trí, giá thuê, diện tích và tiện nghi.
* Hiển thị thông tin chi tiết: Xem đầy đủ thông tin về phòng, bao gồm hình ảnh và mô tả chi tiết.
* Liên hệ trực tiếp: Kết nối trực tiếp với chủ nhà trong ứng dụng.
* Đặt lịch xem phòng: Đặt lịch xem phòng trực tuyến.
* Đánh giá và bình luận: Chia sẻ trải nghiệm sau khi thuê phòng.

**Các tính năng bổ sung:**

* Hệ thống đặt cọc an toàn
* Danh sách yêu thích
* Thông báo phòng mới
* Nhắc nhở thanh toán
* Báo cáo sự cố
* Hỗ trợ trực tuyến
* Thông tin khu vực xung quanh

**1.4 Mục tiêu ứng dụng**

Mục tiêu chính của ứng dụng là:

* Tạo nền tảng đáng tin cậy kết nối chủ nhà và người thuê
* Đơn giản hóa quá trình tìm kiếm và thuê phòng
* Đảm bảo minh bạch và an toàn trong giao dịch
* Cung cấp trải nghiệm người dùng thuận tiện và hiệu quả

Ứng dụng được thiết kế nhằm giải quyết các thách thức trong việc tìm kiếm và thuê phòng, mang đến giải pháp toàn diện cho cả chủ nhà và người thuê.

**CHƯƠNG 2. KHẢO SÁT LÝ THUYẾT**

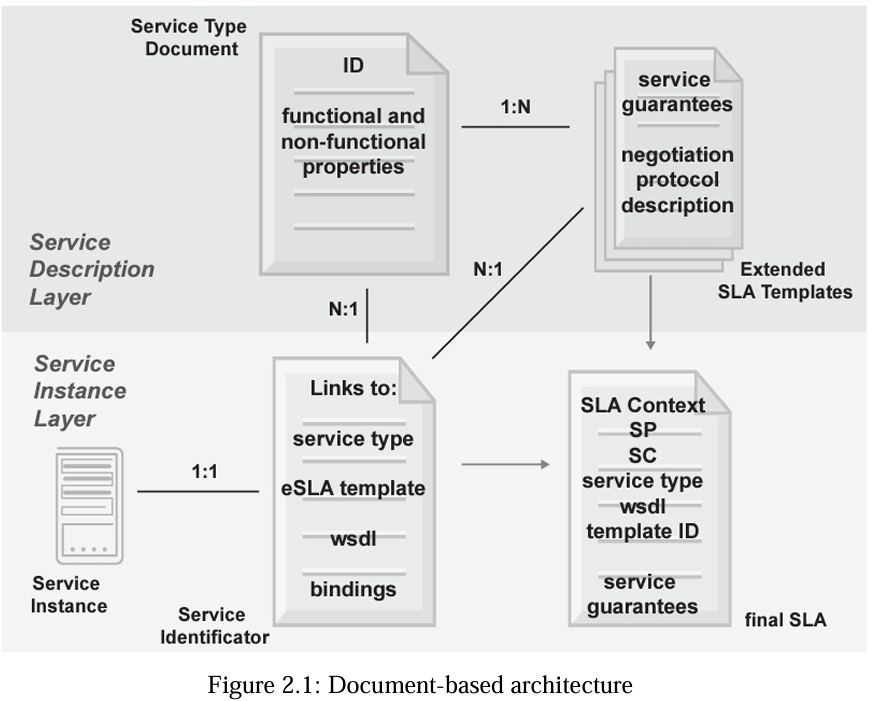
**2.1 Firebase Firestore là gì?**

Firestore là một cơ sở dữ liệu NoSQL tiên tiến, được lưu trữ trên nền tảng đám mây của Google, cho phép các nhà phát triển dễ dàng thao tác với dữ liệu. Nó sử dụng các tính năng trực quan và quen thuộc để lưu trữ, đồng bộ hóa và truy vấn dữ liệu, phục vụ phát triển ứng dụng web và di động.

Firestore cung cấp sự tiện lợi và khả năng mở rộng với khả năng **bộ nhớ đệm thông minh** giúp lưu trữ các truy vấn thường được sử dụng nhất, qua đó giảm độ trễ khi truy cập dữ liệu.

Các nhóm phát triển có thể hưởng lợi từ khả năng sử dụng trực quan thông qua việc tích hợp cả tính năng thời gian thực (real-time sync) và hỗ trợ ngoại tuyến trong các truy vấn.

Tính năng thời gian thực giúp dữ liệu luôn được cập nhật trên tất cả các thiết bị kết nối ngay khi có thay đổi. Điều này đảm bảo rằng người dùng luôn có quyền truy cập phiên bản mới nhất của tài liệu ở bất kỳ đâu.



**Trong Hình 2.1**, chúng ta có thể thấy rằng:

* **Bộ sưu tập (Collections):** Bộ sưu tập chứa các tài liệu cùng loại. Ví dụ, bộ sưu tập **"users"** chứa các tài liệu đại diện cho người dùng.
* **Tài liệu (Documents):** Mỗi tài liệu là một bản ghi dữ liệu, có thể chứa nhiều trường (fields) khác nhau. Ví dụ, một tài liệu trong bộ sưu tập **"users"** có thể chứa các trường như **"name"**, **"email"**, **"address"**, v.v.
* **Trường (Fields):** Trường chứa các giá trị dữ liệu. Ví dụ, trường **"name"** chứa tên của người dùng.
* **Tài liệu lồng nhau (Nested Documents):** Các tài liệu có thể được lồng nhau để tạo ra các cấu trúc dữ liệu phức tạp hơn.

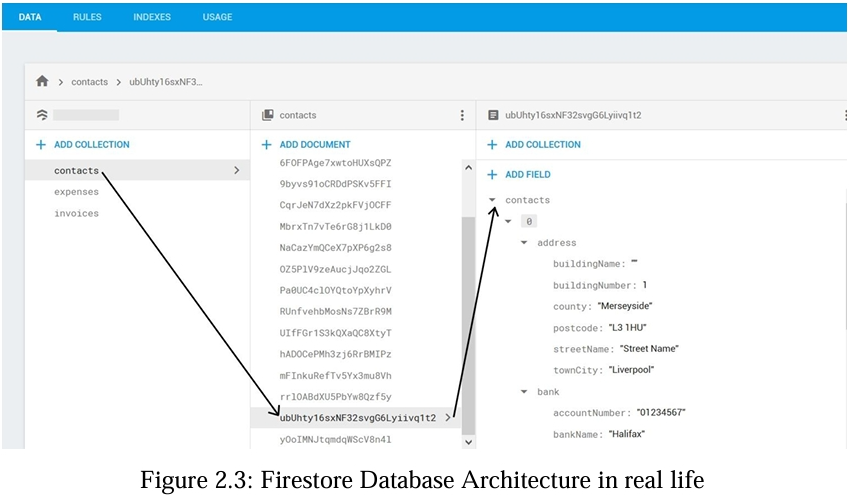
**2.2 Firebase Realtime Database là gì?**

Realtime Database là một dịch vụ cơ sở dữ liệu được lưu trữ trên đám mây do Google Firebase cung cấp, giúp đồng bộ hóa dữ liệu giữa thiết bị người dùng và đám mây mà không cần làm mới trang thủ công.

Cơ sở dữ liệu này cho phép xây dựng các ứng dụng mạnh mẽ và phức tạp mà không cần lo lắng về mạng hoặc lưu trữ dữ liệu. Nó cũng hỗ trợ ứng dụng web và di động tương tác, giúp việc quản lý dữ liệu và tạo trải nghiệm thời gian thực cho người dùng trở nên dễ dàng hơn.

Điều này hỗ trợ các doanh nghiệp tích hợp cập nhật thời gian thực vào ứng dụng của họ, đồng thời giữ cho khách hàng được thông báo kịp thời nếu có bất kỳ thay đổi nào liên quan đến sản phẩm hoặc dịch vụ.

**2.3 Kiến trúc cơ sở dữ liệu Firestore**



Trong **Hình 2.3**, chúng ta có thể thấy rằng mỗi dự án Firebase có thể chứa nhiều cơ sở dữ liệu Firestore. Mỗi cơ sở dữ liệu bao gồm các **bộ sưu tập (collections)**, các bộ sưu tập chứa **tài liệu (documents)**, và mỗi tài liệu là một bản ghi dữ liệu.

Các định nghĩa bổ sung về các từ khóa:

* **Bộ sưu tập (Collections):** Một nhóm các tài liệu có liên quan. Mỗi tài liệu thuộc về một bộ sưu tập, và một bộ sưu tập có thể chứa nhiều tài liệu. Bộ sưu tập là đơn vị tổ chức dữ liệu chính trong Firestore.
* **Tài liệu (Documents):** Là đơn vị lưu trữ chính trong Firestore, chứa dữ liệu ứng dụng theo dạng cặp khóa-giá trị (key-value). Mỗi tài liệu có một ID duy nhất và có thể chứa nhiều trường (fields) cũng như các **bộ sưu tập con (sub-collections)**.
* **Trường (Fields):** Là các thuộc tính trong mỗi tài liệu, tương tự như các cột trong cơ sở dữ liệu truyền thống. Một trường có thể chứa các kiểu dữ liệu cơ bản (chuỗi, số, boolean) hoặc các kiểu dữ liệu phức tạp (mảng, bản đồ).
* **Bộ sưu tập con (Sub-collections):** Là các bộ sưu tập nằm trong một tài liệu, tạo nên cấu trúc phân cấp giữa các bộ sưu tập và tài liệu. Firestore hỗ trợ bộ sưu tập con để lưu trữ dữ liệu chi tiết liên quan đến từng tài liệu.

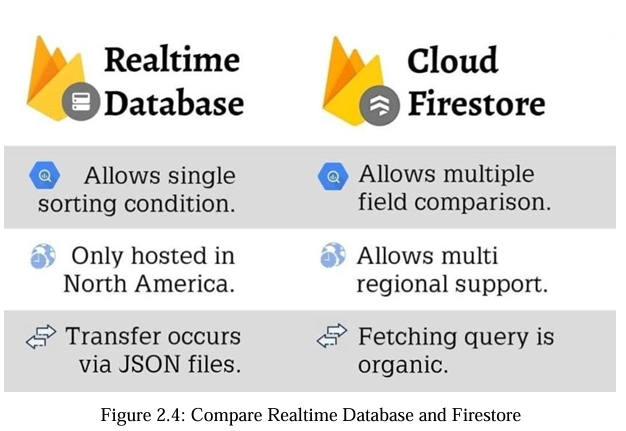
**2.4 Xác thực Firebase (Firebase Authentication)**

Firebase Authentication là một dịch vụ xác thực người dùng do Firebase cung cấp, giúp ứng dụng quản lý và bảo mật tài khoản người dùng một cách dễ dàng.

Mặc dù dự án này không yêu cầu Firebase Authentication, nhưng đây là một giải pháp phổ biến, phù hợp để triển khai các chức năng đăng nhập, đăng ký và quản lý phiên làm việc.

Firebase Authentication hỗ trợ nhiều phương thức như email/mật khẩu, số điện thoại, tài khoản Google, Facebook, và các phương thức khác, cung cấp tính linh hoạt và bảo mật cho người dùng khi truy cập ứng dụng.

**2.5 So sánh Realtime Database và Firestore**



Từ **Hình 2.4**, có thể thấy rằng **Firestore** sử dụng cấu trúc **tài liệu - bộ sưu tập (document-collection)**, trong khi **Realtime Database** sử dụng cấu trúc **cây JSON (JSON tree)**.

Hơn nữa:

* **Firestore** hỗ trợ các truy vấn mạnh mẽ hơn và xử lý dữ liệu quy mô lớn với cấu trúc phân cấp, trong khi **Realtime Database** chủ yếu phù hợp với các ứng dụng có cấu trúc dữ liệu đơn giản.
* Ngoài ra, Firestore cho phép **lọc** và **sắp xếp dữ liệu** dựa trên nhiều điều kiện mà không cần tải toàn bộ tập dữ liệu về thiết bị khách hàng.

**2.6 Chức năng truy vấn và lọc trong Firestore**

Firestore hỗ trợ các truy vấn phức tạp, giúp dễ dàng lọc và sắp xếp dữ liệu dựa trên các tiêu chí khác nhau. Điều này đặc biệt hữu ích trong việc:

* Quản lý danh sách sinh viên.
* Tìm kiếm sinh viên.
* Sắp xếp danh sách theo nhiều tiêu chí khác nhau.

Firestore cho phép:

* **Truy vấn có điều kiện (Conditional Queries):** Lọc dữ liệu dựa trên các điều kiện cụ thể (ví dụ: tìm sinh viên theo tên hoặc ID).
* **Sắp xếp dữ liệu (Sorting Data):** Sắp xếp kết quả theo một hoặc nhiều trường dữ liệu.
* **Truy vấn kết hợp (Combined Queries):** Kết hợp nhiều điều kiện để tạo ra kết quả như mong muốn.

Firestore hỗ trợ truy vấn thời gian thực, nghĩa là khi dữ liệu thay đổi, các thiết bị kết nối sẽ được thông báo ngay lập tức và có thể cập nhật dữ liệu một cách tự động.

**2.7 Cập nhật và đồng bộ dữ liệu thời gian thực**

Firestore cung cấp khả năng **đồng bộ hóa thời gian thực**, cho phép các thay đổi được phản ánh ngay lập tức trên tất cả các thiết bị được kết nối.

Tính năng này đặc biệt hữu ích khi xây dựng ứng dụng quản lý sinh viên thời gian thực, vì dữ liệu sẽ được đồng bộ hóa ngay khi có thay đổi, đảm bảo rằng quản trị viên, quản lý và nhân viên luôn có thông tin mới nhất.

Firestore hỗ trợ **Listeners**, cho phép ứng dụng lắng nghe các thay đổi trong một bộ sưu tập hoặc tài liệu cụ thể. Khi dữ liệu thay đổi, Listener sẽ tự động cập nhật giao diện người dùng với dữ liệu mới.

**2.8 Bảo mật và kiểm soát truy cập trong Firestore**

Firestore sử dụng **Firestore Security Rules** để bảo vệ dữ liệu, cho phép bạn định nghĩa quyền truy cập cho từng tài liệu và bộ sưu tập dựa trên các điều kiện cụ thể. Các quy tắc này được áp dụng theo thời gian thực và có thể tùy chỉnh theo yêu cầu của ứng dụng.

Để triển khai yêu cầu của hệ thống quản lý, Firestore Security Rules có thể được thiết lập như sau:

* **Admin:** Toàn quyền truy cập dữ liệu, bao gồm thêm, xóa và cập nhật thông tin.
* **Quản lý (Manager):** Quyền hạn chế để thêm, chỉnh sửa và xem thông tin sinh viên, nhưng không được thay đổi cài đặt hệ thống hoặc cấu hình bảo mật.
* **Nhân viên (Employee):** Chỉ có quyền xem dữ liệu, không thể chỉnh sửa, ngoại trừ việc cập nhật ảnh đại diện cá nhân.

**Custom Claims:** Khi kết hợp với Firebase Authentication, Firestore hỗ trợ **Custom Claims** để cấp quyền truy cập dựa trên vai trò người dùng. Ví dụ, Custom Claims có thể gắn nhãn người dùng là **Admin**, **Quản lý**, hoặc **Nhân viên** và áp dụng các quyền truy cập tương ứng.

**2.9 Nhập và xuất dữ liệu trong Firestore**

* **Nhập dữ liệu:** Firestore cho phép nhập dữ liệu từ các tệp JSON hoặc CSV thông qua các công cụ hỗ trợ hoặc lập trình. Trong ứng dụng quản lý sinh viên, Admin có thể nhập danh sách sinh viên hoặc chứng chỉ từ các tệp để dễ dàng quản lý.
* **Xuất dữ liệu:** Firestore cho phép xuất dữ liệu dưới các định dạng như JSON hoặc CSV. Điều này hữu ích cho việc tạo báo cáo, thống kê hoặc lưu trữ dữ liệu ngoại tuyến khi cần.

Để đáp ứng yêu cầu xuất dữ liệu sinh viên và chứng chỉ, Firestore có thể được kết hợp với các thư viện như **papaparse** (cho CSV) hoặc **exceljs** để xuất dữ liệu sang tệp Excel hoặc CSV một cách dễ dàng.

**2.10 Firestore SDK và REST API**

Firestore cung cấp **Firebase SDKs** cho nhiều nền tảng như web, iOS và Android, giúp tích hợp và phát triển dễ dàng với Firestore. Các SDK của Firestore hỗ trợ đầy đủ các chức năng như thêm, chỉnh sửa, xóa và đồng bộ hóa dữ liệu thời gian thực.

Ngoài ra, Firestore còn cung cấp **REST API**, cho phép truy cập và quản lý dữ liệu từ các ứng dụng máy chủ hoặc các môi trường không hỗ trợ SDK.

**2.11 Lập chỉ mục trong Firestore (Firestore Indexing)**

Firestore tự động lập chỉ mục cho từng trường trong tài liệu để hỗ trợ các truy vấn hiệu quả. Tuy nhiên, đối với các truy vấn phức tạp, cần tạo **chỉ mục tùy chỉnh (custom indexes)** để tối ưu hóa hiệu suất khi truy vấn các tập dữ liệu lớn như danh sách sinh viên hoặc chứng chỉ.

* **Chỉ mục kết hợp (Composite Index):** Firestore cho phép tạo các chỉ mục kết hợp để hỗ trợ các truy vấn đa điều kiện, ví dụ: sắp xếp sinh viên theo tên và tuổi.

**2.12 Công cụ phát triển và quản lý Firestore**

* **Firebase Console:** Giao diện quản trị Firestore trực tuyến, cho phép nhà phát triển quản lý cơ sở dữ liệu, xem và chỉnh sửa dữ liệu, thiết lập quy tắc bảo mật, và theo dõi các chỉ số hệ thống.
* **Firebase Emulator Suite:** Bộ công cụ mô phỏng của Firebase, cho phép phát triển và kiểm tra Firestore mà không cần kết nối với cơ sở dữ liệu thực. Điều này đặc biệt hữu ích để kiểm tra các quy tắc bảo mật và xác thực trước khi triển khai lên môi trường sản xuất.

**CHƯƠNG 3. ỨNG DỤNG QUẢN LÝ CHO THUÊ PHÒNG TRÊN NỀN TẢNG DI ĐỘNG**

**3.1 Phân tích yêu cầu**

Ứng dụng di động Quản lý Cho thuê Phòng được thiết kế để giúp quản lý thông tin cho thuê phòng một cách hiệu quả và đơn giản. Ứng dụng tập trung vào việc xử lý thông tin chi tiết về phòng, thông tin của chủ nhà và người thuê, với trọng tâm là tính dễ sử dụng và thân thiện với người dùng. Ứng dụng cho phép nhanh chóng cập nhật, bổ sung danh sách phòng và hỗ trợ quản lý hồ sơ người dùng cùng kiểm soát truy cập dựa trên vai trò gồm ba nhóm: Chủ nhà, Người thuê, và Quản trị viên, đảm bảo trải nghiệm tùy chỉnh cho từng nhóm người dùng.

**3.2 Phạm vi hệ thống**

Hệ thống Quản lý Cho thuê Phòng trên di động cung cấp các tính năng chính sau:

1. Đăng nhập và xác thực
2. Quản lý người dùng
3. Quản lý danh sách phòng
4. Hệ thống đặt phòng và giữ chỗ
5. Kiểm soát truy cập dựa trên vai trò

Hệ thống tích hợp một cơ chế đăng nhập an toàn và thân thiện, thiết lập một hệ thống phân cấp quyền truy cập rõ ràng cho từng vai trò. Người dùng là Quản trị viên được trang bị các công cụ quản lý mạnh mẽ, bao gồm khả năng cập nhật ảnh đại diện, theo dõi danh sách người dùng đầy đủ, thêm người dùng mới và quản lý các thông tin cá nhân như tên, thông tin liên lạc và trạng thái tài khoản.

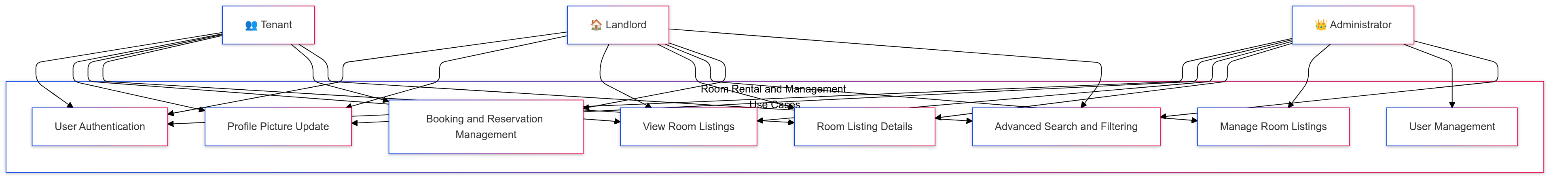
Đối với quản lý danh sách phòng, Chủ nhà và Quản trị viên được cung cấp các công cụ trực quan để xử lý danh sách phòng. Người dùng có thể xem danh sách phòng chi tiết, thêm hoặc cập nhật thông tin phòng và sử dụng các chức năng sắp xếp, tìm kiếm đa tiêu chí để truy xuất thông tin đầy đủ về từng phòng, bao gồm hình ảnh, giá cả và tiện nghi.

Ngoài ra, hệ thống hỗ trợ nhập và xuất danh sách phòng và thông tin người thuê, tăng cường khả năng truy cập và tích hợp dữ liệu. Quản trị viên có thể tải lên và tải xuống dữ liệu danh sách dưới dạng file Excel hoặc CSV để thuận tiện cho người dùng.

Hệ thống thiết lập kiểm soát truy cập dựa trên vai trò cho ba nhóm người dùng chính:

* **Chủ nhà:** Hoàn toàn kiểm soát danh sách phòng, quản lý đặt phòng và hồ sơ cá nhân.
* **Người thuê:** Có khả năng tìm kiếm, xem và đặt phòng, quản lý hồ sơ cá nhân.
* **Quản trị viên:** Quản lý toàn bộ hệ thống, kiểm soát người dùng và giám sát tổng thể nền tảng.

**3.3 Biểu đồ Use Case UML**



Hình 3.3: Biểu đồ Use Case

**Trường hợp Sử dụng 1: Xác thực người dùng**

* **Tác nhân:** Quản trị viên, Chủ nhà, Người thuê
* **Mô tả:** Người dùng đăng nhập vào hệ thống, vai trò của họ được xác định dựa trên thông tin đăng nhập.
* **Điều kiện tiên quyết:** Hệ thống đang hoạt động, người dùng có thông tin đăng nhập hợp lệ.
* **Điều kiện sau cùng:** Người dùng truy cập hệ thống sau khi đăng nhập thành công.

**Trường hợp Sử dụng 2: Cập nhật ảnh đại diện**

* **Tác nhân:** Quản trị viên, Chủ nhà, Người thuê
* **Mô tả:** Người dùng đã đăng nhập có thể cập nhật ảnh đại diện của mình.
* **Điều kiện tiên quyết:** Người dùng đã đăng nhập vào hệ thống.
* **Điều kiện sau cùng:** Ảnh đại diện của người dùng được cập nhật thành công.

**Trường hợp Sử dụng 3: Xem danh sách phòng**

* **Tác nhân:** Quản trị viên, Chủ nhà, Người thuê
* **Mô tả:** Người dùng truy cập danh sách đầy đủ các phòng hiện có trong hệ thống.
* **Điều kiện tiên quyết:** Người dùng đã đăng nhập.
* **Điều kiện sau cùng:** Người dùng xem được thông tin tổng quan về danh sách phòng.

**Trường hợp Sử dụng 4: Chi tiết phòng cho thuê**

* **Tác nhân:** Quản trị viên, Chủ nhà, Người thuê
* **Mô tả:** Người dùng mở trang chi tiết hiển thị đầy đủ thông tin về một phòng cụ thể.
* **Điều kiện tiên quyết:** Người dùng đã đăng nhập vào hệ thống.
* **Điều kiện sau cùng:** Người dùng xem được thông tin chi tiết về phòng đã chọn.

**Trường hợp Sử dụng 5: Quản lý danh sách phòng**

* **Tác nhân:** Chủ nhà, Quản trị viên
* **Mô tả:** Người dùng quản lý hồ sơ phòng bằng cách thêm, cập nhật, xóa và nhập/xuất dữ liệu danh sách phòng.

**Trường hợp Sử dụng 6: Quản lý đặt phòng và giữ chỗ**

* **Tác nhân:** Người thuê, Chủ nhà
* **Mô tả:** Quản lý các yêu cầu đặt phòng, bao gồm tạo, sửa đổi và hủy yêu cầu.
* **Các hành động chính:**

1. Tạo đặt phòng mới
2. Chỉnh sửa yêu cầu đặt phòng
3. Hủy đặt phòng
4. Xác nhận/Từ chối yêu cầu đặt phòng

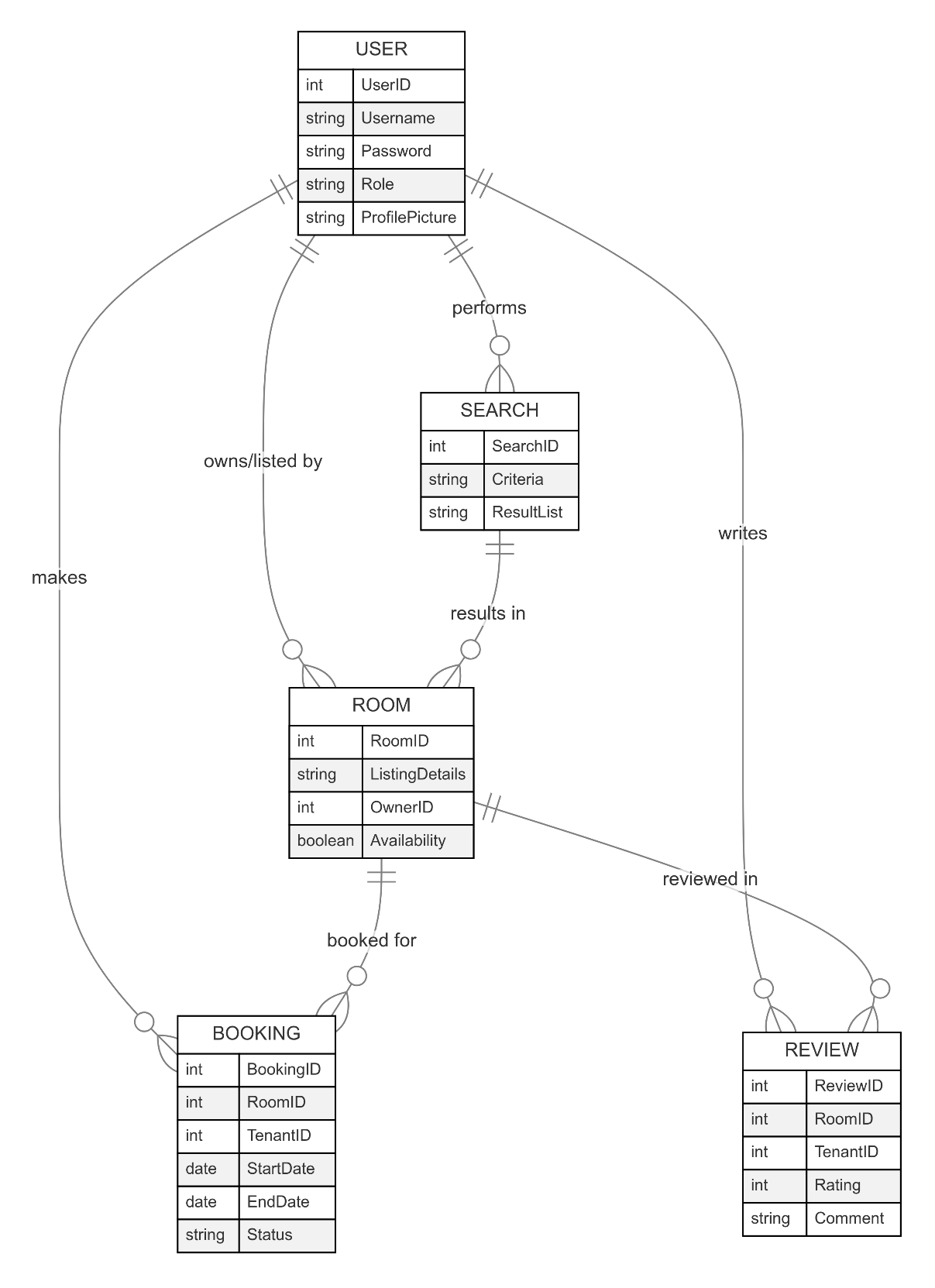
**Trường hợp Sử dụng 7: Quản lý người dùng**

* **Tác nhân:** Quản trị viên
* **Mô tả:** Quản lý toàn diện tài khoản người dùng, bao gồm tạo mới, chỉnh sửa và kiểm soát truy cập.

**Trường hợp Sử dụng 8: Tìm kiếm và Lọc nâng cao**

* **Tác nhân:** Người thuê, Chủ nhà, Quản trị viên
* **Mô tả:** Chức năng tìm kiếm nâng cao để tìm phòng dựa trên nhiều tiêu chí như vị trí, giá cả, tiện nghi và trạng thái phòng.

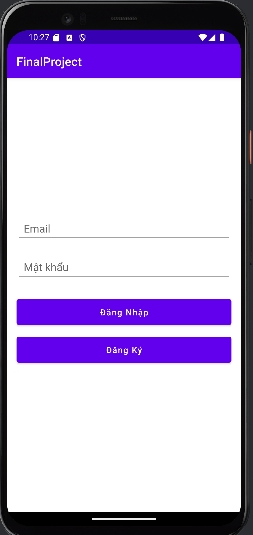
**3.4 Biểu đồ Quan hệ Thực thể UML (ERD)**



Hình 3.4: Biểu đồ ERD

**3.5 Triển khai**

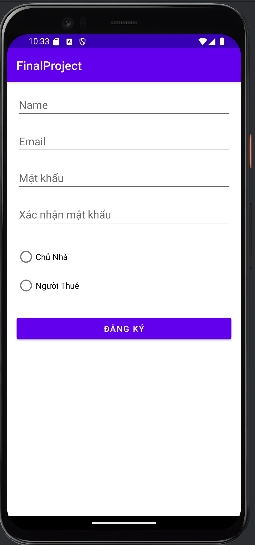
##### *3.5.1 Giao diện đăng nhập*



Hình 3.5.1: Giao diện đăng nhập

Màn hình đăng nhập là giao diện ban đầu khi khởi động hệ thống. Giao diện này yêu cầu người dùng nhập email và mật khẩu. Khi nhập đúng thông tin, người dùng sẽ được gán vai trò cụ thể tương ứng với email và mật khẩu đó. Các vai trò gồm: Chủ nhà hoặc Người thuê.

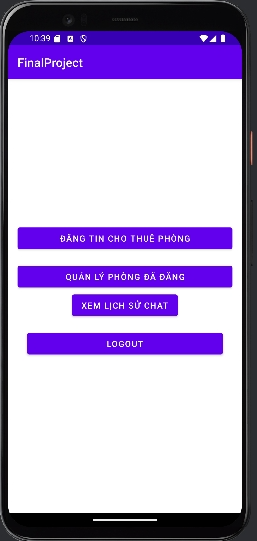
##### *3.5.2 Giao diện đăng ký*



Hình 3.5.2: Giao diện đăng ký

Trang đăng ký yêu cầu người dùng nhập các thông tin như họ tên, email, mật khẩu, xác nhận mật khẩu và chọn một trong hai vai trò: Chủ nhà hoặc Người thuê.

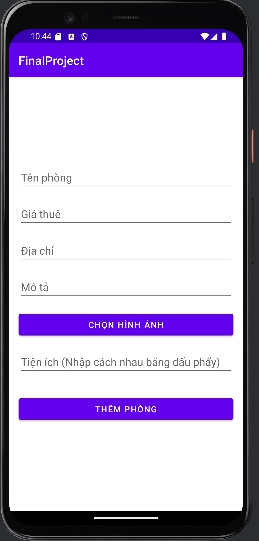
##### *3.5.3 Trang chủ dành cho Chủ nhà*



Hình 3.5.3: Trang chủ của Chủ nhà

Trang chủ dành cho Chủ nhà bao gồm các tính năng: Đăng phòng cho thuê, Quản lý phòng đã đăng, Xem lịch sử trò chuyện, và Đăng xuất.

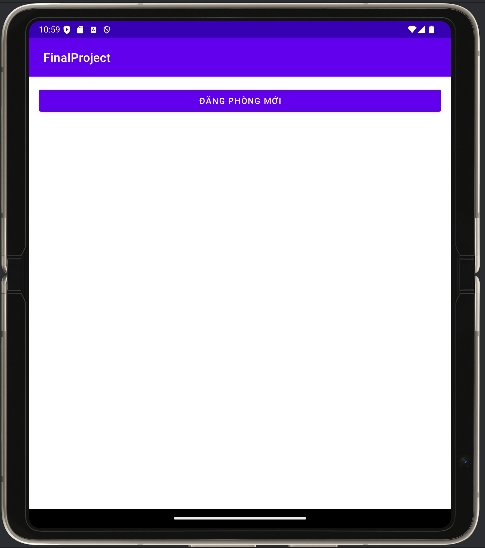
##### *3.5.4 Trang Đăng phòng cho thuê*



Hình 3.5.4: Trang đăng phòng cho thuê

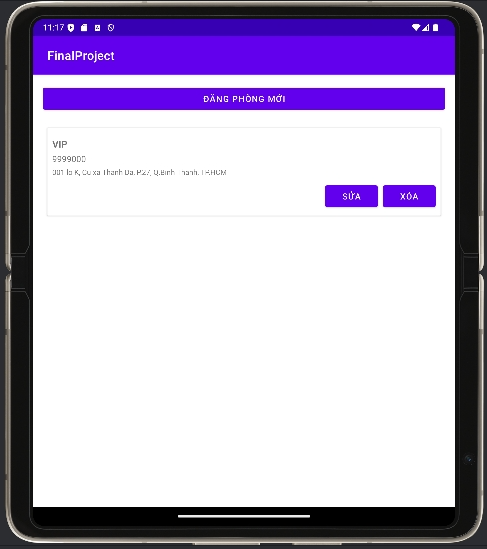
Trang Đăng phòng yêu cầu Chủ nhà nhập các trường dữ liệu: Tên phòng, Giá thuê, Địa chỉ, Mô tả, Chọn ảnh và Tiện nghi (ngăn cách bằng dấu phẩy).

##### *3.5.5 Trang Quản lý phòng đã đăng*



Hình 3.5.5.a: Trang quản lý khi chưa có phòng nào được đăng

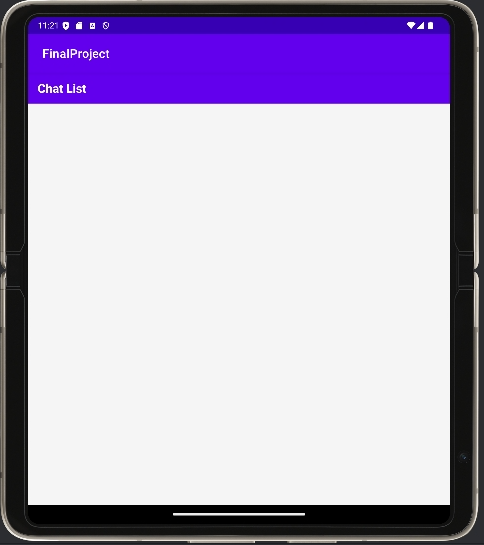
Khi chưa có phòng nào được đăng, trang quản lý phòng sẽ hiển thị nút "Đăng phòng mới" (New Room Post) để người dùng nhấn và được chuyển hướng đến trang Đăng phòng.



Hình 3.5.5.b: Trang quản lý khi đã có phòng được đăng

Trang quản lý phòng khi đã có phòng được đăng sẽ có hai nút: Chỉnh sửa và Xóa. Nút Chỉnh sửa khi nhấn sẽ chuyển hướng đến trang Cập nhật thông tin phòng. Nút Xóa khi nhấn sẽ xóa vĩnh viễn phòng đó.

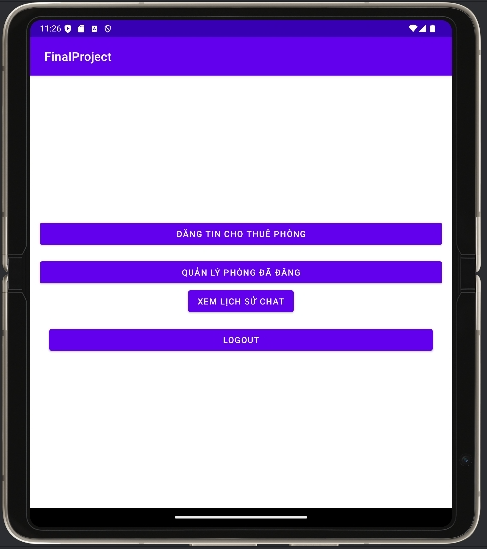
##### *3.5.6 Xem lịch sử trò chuyện*



Hình 3.5.6: Trang lịch sử trò chuyện

Trang lịch sử trò chuyện cho phép xem lại các tin nhắn giữa Chủ nhà và Người thuê.

##### *3.5.7 Nút Đăng xuất*



Hình 3.5.7: Nút Đăng xuất trên Trang chủ

Khi người dùng (Chủ nhà hoặc Người thuê) nhấn nút Đăng xuất (LOGOUT), họ sẽ thoát khỏi phiên làm việc.

**CHƯƠNG 4. KẾT LUẬN**

**4.1 Kết quả đạt được**

Hệ thống Quản lý Thuê Phòng đã được phát triển thành công theo các yêu cầu đã đề ra, cung cấp một khung làm việc toàn diện để quản lý thông tin cho thuê phòng và tạo điều kiện thuận lợi cho sự tương tác giữa chủ nhà và người thuê. Ứng dụng cho phép chủ nhà quản lý và cập nhật danh sách phòng hiệu quả, trong khi người thuê có thể dễ dàng tìm kiếm, xem và đặt phòng dựa trên sở thích của họ. Cơ chế xác thực dựa trên vai trò đã được triển khai một cách mạnh mẽ, thiết lập các mức độ truy cập bảo mật và rõ ràng cho Chủ Nhà, Người Thuê, và Quản Trị Viên, đảm bảo trải nghiệm người dùng hiệu quả và có tổ chức.

**4.2 Hạn chế**

Hạn chế chính của hệ thống hiện tại là thiếu các tính năng nâng cao có thể cải thiện thêm chức năng và tính linh hoạt. Mặc dù hệ thống cung cấp một nền tảng vững chắc cho việc quản lý thuê phòng, nhưng vẫn còn cơ hội để cải thiện, bao gồm:

1. **Tùy chọn tìm kiếm và lọc hạn chế**: Hệ thống hiện chỉ cung cấp các khả năng tìm kiếm cơ bản dựa trên vị trí, giá cả và tiện nghi. Việc tích hợp các công cụ tìm kiếm và lọc tinh vi hơn, cho phép người dùng kết hợp nhiều tiêu chí, sẽ nâng cao hiệu quả trong việc tìm phòng phù hợp.
2. **Thiếu hình ảnh hóa dữ liệu**: Hệ thống chưa có phần dành riêng để trực quan hóa dữ liệu thuê thông qua biểu đồ, đồ thị và các cách biểu diễn dữ liệu tương tác khác. Việc tích hợp các tính năng này sẽ giúp người dùng, đặc biệt là quản trị viên và chủ nhà, có được cái nhìn sâu sắc hơn và đưa ra quyết định chính xác hơn.
3. **Thiếu báo cáo và phân tích**: Hệ thống hiện tại không cung cấp các khả năng báo cáo hoặc phân tích toàn diện. Việc giới thiệu các tính năng tạo báo cáo chi tiết về tỷ lệ lấp đầy, xu hướng doanh thu và các chỉ số hiệu suất quan trọng khác sẽ nâng cao khả năng hỗ trợ ra quyết định của hệ thống.

**4.3 Đề xuất cải tiến trong tương lai**

Để khắc phục các hạn chế đã được nhận diện và nâng cao thêm chức năng của ứng dụng Quản lý Thuê Phòng, các đề xuất cải tiến sau được đề xuất:

1. **Tìm kiếm và lọc nâng cao**: Phát triển một mô-đun tìm kiếm nâng cao cho phép người dùng lọc và sắp xếp danh sách phòng dựa trên nhiều tiêu chí như vị trí, khoảng giá, kích thước, tiện nghi, và tình trạng sẵn có. Điều này sẽ cải thiện đáng kể trải nghiệm người dùng khi tìm kiếm phòng phù hợp nhất.
2. **Hình ảnh hóa dữ liệu và phân tích**: Tích hợp một phần hoặc mô-đun dành riêng để trình bày dữ liệu thuê phòng thông qua các cách biểu diễn trực quan, chẳng hạn như biểu đồ, đồ thị, và bảng điều khiển. Điều này sẽ giúp quản trị viên, chủ nhà và cả người thuê có được những thông tin giá trị và đưa ra các quyết định sáng suốt hơn.
3. **Báo cáo toàn diện**: Triển khai một khung báo cáo mạnh mẽ tạo ra các báo cáo chi tiết về các chỉ số chính như tỷ lệ lấp đầy, doanh thu, xu hướng đặt phòng, và hoạt động người dùng. Các báo cáo này sẽ hỗ trợ chủ nhà và quản trị viên giám sát hiệu suất kinh doanh cho thuê và đưa ra quyết định dựa trên dữ liệu.
4. **Thông báo và nhắc nhở tự động**: Giới thiệu các tính năng tự động gửi thông báo và nhắc nhở đến người dùng, chẳng hạn như ngày đến hạn thanh toán, danh sách phòng mới phù hợp với sở thích của người thuê, hoặc cảnh báo bảo trì cho chủ nhà. Điều này sẽ nâng cao trải nghiệm người dùng tổng thể và giúp hợp lý hóa các nhiệm vụ quản lý cho thuê khác nhau.
5. **Công cụ giao tiếp và cộng tác tích hợp**: Cải tiến hệ thống bằng cách tích hợp các tính năng giao tiếp và cộng tác như nhắn tin trong ứng dụng, chia sẻ tệp, và quản lý công việc. Điều này sẽ cải thiện sự phối hợp giữa chủ nhà và người thuê, thúc đẩy trải nghiệm thuê phòng mượt mà và hiệu quả hơn.

Bằng cách triển khai các cải tiến này, ứng dụng Quản lý Thuê Phòng sẽ trở thành một công cụ toàn diện và mạnh mẽ hơn, mang đến cho người dùng trải nghiệm phong phú trong việc quản lý tài sản cho thuê và thúc đẩy sự kết nối mạnh mẽ hơn giữa chủ nhà và người thuê.

**1. AddRoomActivity.java**

* Thuộc tính năng: **Đăng tin cho thuê phòng**.
* Vai trò: Hoạt động cho phép chủ nhà nhập thông tin phòng trọ như hình ảnh, giá thuê, địa chỉ, và tiện ích đi kèm.

**2. BookedDatesAdapter.java**

* Thuộc tính năng: **Lịch hẹn xem phòng & quản lý lịch trống**.
* Vai trò: Adapter để hiển thị danh sách các ngày đã được đặt hoặc không khả dụng.

**3. BookingHistoryActivity.java**

* Thuộc tính năng: **Lịch hẹn xem phòng**.
* Vai trò: Hiển thị lịch sử các lần đặt phòng hoặc lịch hẹn của người dùng.

**4. BookingSuccessActivity.java**

* Thuộc tính năng: **Lịch hẹn xem phòng**.
* Vai trò: Hiển thị màn hình xác nhận sau khi người dùng đặt lịch thành công.

**5. ChatActivity.java**

* Thuộc tính năng: **Liên hệ trực tiếp**.
* Vai trò: Cung cấp giao diện trò chuyện giữa người thuê và chủ nhà.

**6. ChatListActivity.java**

* Thuộc tính năng: **Liên hệ trực tiếp**.
* Vai trò: Hiển thị danh sách các cuộc trò chuyện hiện có.

**7. ContactAdapter.java**

* Thuộc tính năng: **Liên hệ trực tiếp**.
* Vai trò: Adapter hỗ trợ hiển thị danh bạ hoặc danh sách liên hệ giữa các bên.

**8. DatePickerActivity.java**

* Thuộc tính năng: **Lịch hẹn xem phòng**.
* Vai trò: Giao diện cho người dùng chọn ngày và giờ đặt lịch.

**9. DepositActivity.java**

* Thuộc tính năng: **Hệ thống đặt cọc**.
* Vai trò: Giao diện cho phép người dùng thực hiện các giao dịch đặt cọc.

**10. EditRoomActivity.java**

* Thuộc tính năng: **Quản lý phòng trọ**.
* Vai trò: Cho phép chủ nhà chỉnh sửa thông tin phòng trọ đã đăng.

**11. HostActivity.java**

* Thuộc tính năng: **Quản lý phòng trọ**.
* Vai trò: Giao diện chính cho chủ nhà để xem và quản lý các phòng trọ của họ.

**12. ImagePagerAdapter.java**

* Thuộc tính năng: **Hiển thị thông tin chi tiết**.
* Vai trò: Hỗ trợ hiển thị các hình ảnh của phòng trọ theo dạng trình chiếu.

**13. LoginActivity.java**

* Thuộc tính năng: **Xác nhận thông tin**.
* Vai trò: Xử lý đăng nhập cho người dùng (cả chủ nhà và người thuê).

**14. ManageRoomAdapter.java**

* Thuộc tính năng: **Quản lý phòng trọ**.
* Vai trò: Adapter hỗ trợ hiển thị danh sách các phòng trọ mà chủ nhà đang quản lý.

**15. ManageRoomsActivity.java**

* Thuộc tính năng: **Quản lý phòng trọ**.
* Vai trò: Giao diện cho phép chủ nhà xem tổng quan các phòng trọ.

**16. Message.java / MessageAdapter.java**

* Thuộc tính năng: **Liên hệ trực tiếp**.
* Vai trò: Xử lý và hiển thị các tin nhắn trong ứng dụng.

**17. MyAppGlideModule.java**

* Thuộc tính năng: **Hiển thị hình ảnh**.
* Vai trò: Cấu hình Glide để tải và hiển thị hình ảnh.

**18. RenterActivity.java**

* Thuộc tính năng: **Tìm kiếm phòng trọ**.
* Vai trò: Giao diện chính cho người thuê để tìm và quản lý thông tin phòng trọ.

**19. Review.java / ReviewAdapter.java**

* Thuộc tính năng: **Đánh giá và nhận xét**.
* Vai trò: Xử lý và hiển thị các nhận xét, đánh giá của người thuê.

**20. Room.java**

* Thuộc tính năng: **Hiển thị thông tin chi tiết**.
* Vai trò: Mô tả cấu trúc dữ liệu của một phòng trọ (giá, địa chỉ, tiện ích, v.v.).

**21. RoomAdapter.java**

* Thuộc tính năng: **Tìm kiếm phòng trọ**.
* Vai trò: Hiển thị danh sách phòng trọ trong kết quả tìm kiếm.

**22. RoomDetailsActivity.java**

* Thuộc tính năng: **Hiển thị thông tin chi tiết**.
* Vai trò: Giao diện chi tiết về một phòng trọ cụ thể.

**23. RoomVerificationActivity.java**

* Thuộc tính năng: **Xác nhận thông tin**.
* Vai trò: Hỗ trợ xác minh thông tin phòng trọ do chủ nhà cung cấp.

**24. SavedRoomsActivity.java**

* Thuộc tính năng: **Tạo danh sách ưa thích**.
* Vai trò: Hiển thị danh sách các phòng trọ mà người thuê đã lưu.

**25. SearchRoomActivity.java**

* Thuộc tính năng: **Tìm kiếm phòng trọ**.
* Vai trò: Giao diện để tìm kiếm phòng trọ dựa trên các tiêu chí.

**26. SignupActivity.java**

* Thuộc tính năng: **Xác nhận thông tin**.
* Vai trò: Xử lý đăng ký tài khoản cho người dùng.