|  |  |
| --- | --- |
| logo (CMYK)-01 | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN HỌC PHẦN**

**ỨNG DỤNG QUẢN LÝ BÁN NỘI THẤT**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn : ThS. Lê Nhật Tùng

Sinh viên thực hiện :

Võ Quang Huy

MSSV: 2180607580 Lớp: 21DTHD5

Nguyễn Hoàng Phúc

MSSV: 2180607892 Lớp: 21DTHD5

Ngô Quốc Trọng

MSSV: 2180608144 Lớp: 21DTHD5

TP. Hồ Chí Minh, 2025

Lời cam đoan

Chúng tôi xin trân trọng cam đoan rằng đồ án cơ sở này là thành quả của quá trình làm việc nghiêm túc và nghiên cứu miệt mài của nhóm chúng tôi, được thực hiện dưới sự hướng dẫn tận tình của thầy ThS. Lê Nhật Tùng. Tất cả nội dung trình bày trong báo cáo này, bao gồm các dữ liệu, kết quả phân tích, và các đề xuất, đều là kết quả của sự cố gắng không ngừng của cả nhóm, được xây dựng dựa trên cơ sở lý thuyết vững chắc và các nguồn thông tin đáng tin cậy.

Chúng tôi xin khẳng định rằng không có bất kỳ nội dung nào trong báo cáo này được sao chép từ bất kỳ tài liệu hay nguồn thông tin nào khác mà không có sự trích dẫn và ghi nguồn đầy đủ. Mọi thông tin, dữ liệu, số liệu và kết luận trong báo cáo đều do nhóm chúng tôi trực tiếp thực hiện, tổng hợp và phân tích, không trùng lặp hay sử dụng lại từ các nghiên cứu khác.

Trong quá trình thực hiện đồ án, chúng tôi luôn tuân thủ nghiêm ngặt các nguyên tắc về đạo đức trong nghiên cứu và học thuật. Chúng tôi cam kết rằng báo cáo này không chỉ phản ánh trung thực năng lực nghiên cứu và làm việc của nhóm mà còn thể hiện tinh thần học hỏi và trách nhiệm của từng thành viên.

Chúng tôi hoàn toàn chịu trách nhiệm trước nhà trường và các bên liên quan về tính trung thực, chính xác của toàn bộ nội dung và kết quả trong báo cáo. Nếu phát hiện bất kỳ sai sót hay vi phạm nào, chúng tôi xin chịu mọi hình thức xử lý theo quy định.

TP. Hồ Chí Minh, Ngày 1/1/2025

Sinh viên thực hiện:

Võ Quang Huy

Nguyễn Hoàng Phúc

Ngô Quốc Trọng

MỤC LỤC

[Chương 1. TỔNG QUAN 11](#_Toc186793656)

[1.1 Tổng quan 11](#_Toc186793657)

[1.2 Nhiệm vụ đồ án 12](#_Toc186793658)

[*1.2.1 Mục tiêu nghiên cứu* 12](#_Toc186793659)

[*1.2.2 Đối tượng nghiên cứu và phạm vi giới hạn* 13](#_Toc186793660)

[*1.2.3 Ý nghĩa khoa học và thực tiễn* 14](#_Toc186793661)

[1.3 Cấu trúc đồ án 15](#_Toc186793662)

[Chương 2. CƠ SỞ LÝ THUYẾT 16](#_Toc186793663)

[2.1 Giới thiệu về công nghệ 16](#_Toc186793664)

[*2.1.1 Dart* 16](#_Toc186793665)

[*2.1.2 Flutter* 17](#_Toc186793666)

[*2.1.3 Fire base* 19](#_Toc186793667)

[2.2 Phân tích và thiết kế 21](#_Toc186793668)

[*2.2.1 Đặc tả yêu cầu của phần mềm* 21](#_Toc186793669)

[*2.2.2 Mô hình hóa dữ liệu* 31](#_Toc186793670)

[2.3 Thiết kế cơ sở dữ liệu 39](#_Toc186793671)

[*2.3.1 Cơ sở dữ liệu* 39](#_Toc186793672)

[Chương 3. KẾT QUẢ THỰC NGHIỆM 40](#_Toc186793673)

[3.1 Kết quả 40](#_Toc186793674)

[*3.1.1 Các kết quả* 40](#_Toc186793675)

[Chương 4. KẾT LUẬN VÀ KIẾN NGHỊ 48](#_Toc186793676)

[4.1 Kết luận chung 48](#_Toc186793677)

[4.2 Nội dung đã làm được 49](#_Toc186793678)

[4.3 Hướng phát triển 49](#_Toc186793679)

[TÀI LIỆU THAM KHẢO 52](#_Toc186793680)

MỤC LỤC BẢNG

[Bảng 2.1 Đăng ký 20](#_Toc186793681)

[Bảng 2.2 Đăng nhập 21](#_Toc186793682)

[Bảng 2.3 Tìm kiếm sản phẩm 22](#_Toc186793683)

[Bảng 2.4 Quản lý sản phẩm giỏ hàng 23](#_Toc186793684)

[Bảng 2.5 Thanh toán 24](#_Toc186793685)

[Bảng 2.6 Chỉnh sửa thông tin người dùng 25](#_Toc186793686)

[Bảng 2.7 Quản lý danh mục yêu thích 26](#_Toc186793687)

MỤC LỤC ẢNH

[Hình 2.1 Mô hình ERD 30](#_Toc186793966)

[Hình 2.2 Mô hình Usecase Tổng quát 31](#_Toc186793967)

[Hình 2.3 Mô hình Usecase phân rã cức năng Đăng nhập-đăng xuất 32](#_Toc186793968)

[Hình 2.4 Mô hình Usecase phân rã chức năng Đăng ký 32](#_Toc186793969)

[Hình 2.5 Mô hình Usecase phân rã chức năng Thanh toán 33](#_Toc186793970)

[Hình 2.6 Mô hình Usecase phân rã chức năng Quản lý người dùng 33](#_Toc186793971)

[Hình 2.7 Mô hình Class diagram 34](#_Toc186793972)

[Hình 2.8 Mô hình Activity Đăng nhập 35](#_Toc186793973)

[Hình 2.9 Mô hình Activity đặt sản phẩm 35](#_Toc186793974)

[Hình 2.10 Mô hình Activity Quên mật khẩu 36](#_Toc186793975)

[Hình 2.11 Mô hình cơ sở dữ liệu 37](#_Toc186793976)

**Chương 1. TỔNG QUAN**

## 1.1 Tổng quan

Trong bối cảnh xã hội hiện đại, nhu cầu về trang trí và cải thiện không gian sống ngày càng trở thành một xu hướng quan trọng. Nội thất không chỉ đóng vai trò tiện ích mà còn thể hiện phong cách sống và gu thẩm mỹ của từng cá nhân. Đặc biệt, với sự phát triển nhanh chóng của công nghệ, người tiêu dùng hiện đại mong muốn tìm kiếm các giải pháp mua sắm tiện lợi, hiệu quả và dễ dàng tiếp cận hơn bao giờ hết.

Tuy nhiên, nhiều cửa hàng nội thất vẫn đang gặp phải thách thức trong việc quản lý kho hàng, sản phẩm, đơn hàng và khách hàng một cách đồng bộ và hiệu quả. Việc thiếu các công cụ hỗ trợ quản lý chuyên nghiệp có thể dẫn đến những vấn đề như sai sót trong thông tin sản phẩm, chậm trễ trong xử lý đơn hàng, hoặc không đáp ứng kịp nhu cầu của khách hàng. Điều này không chỉ gây tổn thất về tài chính mà còn ảnh hưởng tiêu cực đến uy tín của doanh nghiệp.

Với sự phổ biến của smartphone và xu hướng số hóa, việc xây dựng một ứng dụng quản lý bán nội thất trên nền tảng di động là một giải pháp cấp thiết. Ứng dụng Flutter được chọn vì đây là framework phát triển đa nền tảng hiện đại, cho phép tạo ra các ứng dụng mượt mà, nhanh chóng và đồng nhất trên cả iOS và Android. Flutter không chỉ giúp giảm thiểu thời gian và chi phí phát triển, mà còn cung cấp các công cụ mạnh mẽ để xây dựng giao diện người dùng hấp dẫn, dễ sử dụng.

Ngoài ra, ứng dụng này hướng đến việc nâng cao trải nghiệm của cả người quản lý lẫn khách hàng. Người quản lý có thể dễ dàng theo dõi tình trạng kho, cập nhật thông tin sản phẩm và xử lý đơn hàng ngay trên điện thoại. Trong khi đó, khách hàng sẽ được tiếp cận thông tin sản phẩm chi tiết, đặt hàng và thanh toán trực tuyến một cách tiện lợi. Điều này không chỉ giúp tối ưu hóa quy trình kinh doanh mà còn tạo ra lợi thế cạnh tranh cho doanh nghiệp trong ngành nội thất đang ngày càng cạnh tranh khốc liệt.

Với những lý do trên, việc phát triển ứng dụng Flutter quản lý bán nội thất không chỉ mang tính thực tiễn cao mà còn là bước đột phá trong việc ứng dụng công nghệ để giải quyết các vấn đề kinh doanh truyền thống, góp phần thúc đẩy sự phát triển của ngành nội thất trong thời đại số hóa.

## 1.2 Nhiệm vụ đồ án

### *1.2.1 Mục tiêu nghiên cứu*

Mục tiêu chính của đề tài là phát triển một ứng dụng quản lý bán nội thất dễ sử dụng, hiệu quả và thân thiện với người dùng, đáp ứng nhu cầu mua sắm và trang trí của đông đảo đối tượng. Các mục tiêu cụ thể bao gồm:

Nghiên cứu và ứng dụng công nghệ Flutter để xây dựng ứng dụng quản lý bán nội thất

* Flutter là một framework hiện đại do Google phát triển, cho phép xây dựng ứng dụng đa nền tảng với mã nguồn duy nhất. Nghiên cứu sẽ tập trung vào việc khai thác các ưu điểm của Flutter, như hiệu năng cao, hỗ trợ giao diện đẹp mắt và linh hoạt, cùng khả năng tích hợp dễ dàng với các công cụ như Firebase. Mục tiêu là xây dựng một ứng dụng tối ưu hóa thời gian phát triển nhưng vẫn đảm bảo chất lượng và hiệu quả.

Phát triển các chức năng cốt lõi: quản lý kho hàng, sản phẩm, đơn hàng và thông tin khách hàng

Quản lý kho hàng: Xây dựng hệ thống lưu trữ và theo dõi thông tin sản phẩm trong kho, bao gồm số lượng tồn kho, ngày nhập hàng và trạng thái hàng hóa. Hệ thống sẽ cung cấp các thông báo khi hàng hóa sắp hết để người quản lý có thể kịp thời bổ sung.

* Quản lý sản phẩm: Tạo giao diện thân thiện để thêm, sửa, xóa và cập nhật thông tin chi tiết về sản phẩm như tên, mô tả, giá cả, hình ảnh và danh mục.
* Quản lý đơn hàng: Phát triển chức năng theo dõi tình trạng đơn hàng từ lúc đặt hàng đến khi giao hàng thành công, bao gồm các trạng thái như đang xử lý, đã giao, hoặc đã hủy.
* Quản lý thông tin khách hàng: Hệ thống lưu trữ thông tin khách hàng, lịch sử giao dịch, và thông tin phản hồi để hỗ trợ dịch vụ chăm sóc khách hàng tốt hơn.

Tích hợp các tính năng mua sắm trực tuyến như giỏ hàng, thanh toán và theo dõi đơn hàng

* Giỏ hàng: Cho phép người dùng thêm, chỉnh sửa và xóa sản phẩm khỏi giỏ hàng một cách dễ dàng. Đồng thời, tích hợp chức năng tính toán tổng chi phí và các chiết khấu nếu có.
* Thanh toán: Tích hợp các phương thức thanh toán trực tuyến phổ biến, chẳng hạn như thẻ ngân hàng, ví điện tử (Momo, ZaloPay), và thanh toán khi nhận hàng (COD).
* Theo dõi đơn hàng: Người dùng có thể theo dõi tiến độ đơn hàng của mình, từ khâu xác nhận đơn, đóng gói, vận chuyển đến khi giao hàng thành công.

Tối ưu hóa giao diện người dùng (UI/UX) và hiệu suất ứng dụng nhằm đảm bảo trải nghiệm mượt mà, hiện đại

* Thẩm mỹ cao: Giao diện đẹp mắt, hiện đại, phù hợp với ngành nội thất.
* Dễ sử dụng: Các thao tác trong ứng dụng sẽ được thiết kế trực quan, đơn giản, phù hợp với mọi đối tượng người dùng.
* Tốc độ và hiệu suất: Tối ưu hóa mã nguồn và sử dụng các công cụ hỗ trợ để đảm bảo ứng dụng hoạt động mượt mà ngay cả với lượng dữ liệu lớn.
* Đề xuất giải pháp cải thiện quy trình quản lý bán hàng, nâng cao hiệu quả kinh doanh cho các doanh nghiệp nội thất
* Đưa ra các giải pháp để tự động hóa một số công việc thủ công, như kiểm kê hàng hóa và xử lý đơn hàng, nhằm giảm thiểu sai sót.
* Cung cấp các báo cáo trực quan về doanh thu, sản phẩm bán chạy, và hành vi mua sắm của khách hàng, từ đó hỗ trợ người quản lý đưa ra các quyết định kinh doanh chính xác hơn.
* Đề xuất cách tích hợp ứng dụng với các nền tảng khác như website, mạng xã hội để mở rộng phạm vi tiếp cận khách hàng.

### *1.2.2 Đối tượng nghiên cứu và phạm vi giới hạn*

Đối tượng nghiên cứu:

* Cửa hàng và doanh nghiệp kinh doanh nội thất: Nghiên cứu các quy trình vận hành và quản lý hiện tại của họ để xác định các vấn đề thường gặp và yêu cầu cụ thể.
* Người dùng ứng dụng: Bao gồm khách hàng mua nội thất trực tuyến và nhân viên quản lý cửa hàng, tập trung vào hành vi sử dụng và nhu cầu thực tế của họ.
* Công nghệ liên quan: Flutter, Firebase, và các công cụ phát triển phần mềm hỗ trợ việc xây dựng ứng dụng đa nền tảng

Phạm vi giới hạn:

* Phát triển ứng dụng trên nền tảng di động (iOS và Android).
* Tập trung vào các chức năng quản lý bán hàng và trải nghiệm mua sắm trực tuyến.
* Sử dụng cơ sở dữ liệu Firebase để lưu trữ và xử lý dữ liệu thời gian thực.
* Giao diện đơn giản, dễ sử dụng, nhưng đảm bảo tính thẩm mỹ và chuyên nghiệp phù hợp với ngành nội thất.
* Giới hạn trong phạm vi các cửa hàng nội thất vừa và nhỏ hoặc các doanh nghiệp nội thất có nhu cầu chuyển đổi số.

### *1.2.3 Ý nghĩa khoa học và thực tiễn*

* Ý nghĩa khoa học:
  + Cung cấp giải pháp công nghệ hiện đại giúp tối ưu hóa quy trình kinh doanh, giảm thiểu sai sót và tăng cường hiệu quả quản lý.
  + Tạo điều kiện cho các doanh nghiệp nội thất nhỏ lẻ tiếp cận với các công cụ số hóa chuyên nghiệp, từ đó nâng cao khả năng cạnh tranh trên thị trường.
  + Mang lại trải nghiệm mua sắm tiện lợi và hiện đại cho khách hàng, tăng mức độ hài lòng và trung thành với thương hiệu.
* Ý nghĩa thực tiễn:
  + Góp phần ứng dụng và khai thác tiềm năng của Flutter trong việc phát triển ứng dụng quản lý và thương mại điện tử.
  + Làm phong phú thêm các nghiên cứu và ứng dụng về quản lý bán hàng trong lĩnh vực nội thất, từ đó mở rộng cơ hội nghiên cứu trong các lĩnh vực liên quan.
  + Là tài liệu tham khảo hữu ích cho các nhà phát triển phần mềm, nghiên cứu viên hoặc sinh viên trong lĩnh vực công nghệ thông tin và quản trị kinh doanh..

## 1.3 Cấu trúc đồ án

Chương 1. TỔNG QUAN: Giới thiệu ngắn gọn về đề tài, nêu tóm tắt những lý thuyết, nghiên cứu trước đây có liên quan đến đề tài.

Chương 2. CƠ SỞ LÝ THUYẾT: Trình bày các khái niệm và phương pháp giải quyết vấn đề của tác giả bao gồm mô tả các công nghệ, hệ thống, các ràng buộc hoặc các giải pháp mới, những mô hình toán, lý giải xây dựng mô hình, ...

Chương 3. KẾT QUẢ THỰC NGHIỆM: Mô tả công việc thực nghiệm đề tài đã tiến hành, các kết quả nghiên cứu lý thuyết, kết quả thực nghiệm đạt được. Đối với các đề tài ứng dụng có kết quả là sản phẩm phần mềm phải có hồ sơ thiết kế, cài đặt, giao diện... theo một trong các mô hình đã học (UML, ...)

Chương 4. KẾT LUẬN VÀ KIẾN NGHỊ: Nêu những kết luận chung, khẳng định những kết quả đạt được, những đóng góp, đề xuất mới và kiến nghị (nếu có).

**Chương 2. CƠ SỞ LÝ THUYẾT**

## 2.1 Giới thiệu về công nghệ

### *2.1.1 Dart*

Dart là một ngôn ngữ lập trình được phát triển bởi Google, lần đầu ra mắt vào năm 2011. Dart được thiết kế để xây dựng các ứng dụng đa nền tảng với hiệu suất cao, bao gồm các ứng dụng web, di động và máy tính để bàn. Ngôn ngữ này nổi bật nhờ khả năng tối ưu hóa và dễ sử dụng, đặc biệt khi kết hợp với Flutter, một framework phổ biến để phát triển giao diện người dùng.

Một số đặc điểm nổi bật:

Dart là viết tắt của không cụm từ chính thức nhưng được Google thiết kế với mục tiêu hiệu quả và hiện đại.

Là một server-side và client-side scripting language, giúp phát triển ứng dụng một cách liền mạch.

Dart script được biên dịch thành mã máy (native code) hoặc JavaScript, giúp tối ưu hóa hiệu suất trên nhiều nền tảng.

Tệp Dart có phần mở rộng mặc định là .dart.

Dart chạy trên nhiều nền tảng như Windows, macOS, Linux, Android và iOS.

Dart tích hợp tốt với Flutter, giúp phát triển giao diện người dùng nhanh chóng và hiệu quả.

Dart hỗ trợ null safety, giúp giảm thiểu các lỗi liên quan đến giá trị null trong mã nguồn.

Dart đang ngày càng phổ biến nhờ sự kết hợp mạnh mẽ với Flutter trong việc phát triển ứng dụng đa nền tảng. Các ứng dụng được xây dựng bằng Dart và Flutter hiện được sử dụng bởi nhiều doanh nghiệp lớn như Alibaba, Google Ads, và BMW. Mặc dù chưa đạt mức phổ biến như các ngôn ngữ lâu đời khác, Dart vẫn thu hút được một cộng đồng lập trình viên đông đảo nhờ những ưu điểm sau:

Hot reload: Cho phép lập trình viên thấy ngay các thay đổi trong ứng dụng mà không cần khởi động lại toàn bộ.

Cộng đồng hỗ trợ mạnh mẽ: Dart có một hệ sinh thái đang phát triển, với tài liệu phong phú, thư viện mã nguồn mở, và sự hỗ trợ từ Google.

Thư viện phong phú: Dart cung cấp nhiều thư viện và gói (package) để hỗ trợ phát triển nhanh chóng và hiệu quả.

Ưu điểm dart:

* Hiệu suất cao: Dart biên dịch trực tiếp sang mã máy, mang lại hiệu năng tốt trên các nền tảng di động và máy tính.
* Tính đa nền tảng: Một đoạn mã Dart có thể chạy trên nhiều nền tảng mà không cần điều chỉnh.
* Null safety: Giúp lập trình viên tránh các lỗi phổ biến liên quan đến giá trị null.
* Dễ học: Cú pháp của Dart thân thiện với các ngôn ngữ như Java, C#, và JavaScript.
* Hỗ trợ mạnh mẽ từ Flutter: Dart là ngôn ngữ chính của Flutter, giúp tăng tốc độ phát triển ứng dụng.

Dart là một ngôn ngữ lập trình hiện đại, mạnh mẽ và đa năng, đặc biệt phù hợp với việc phát triển ứng dụng đa nền tảng. Với sự hỗ trợ từ Google và cộng đồng lập trình viên ngày càng đông đảo, Dart tiếp tục là lựa chọn lý tưởng cho các nhà phát triển muốn xây dựng ứng dụng hiệu quả và dễ bảo trì.

### *2.1.2 Flutter*

Flutter là một framework mã nguồn mở được phát triển bởi Google, lần đầu ra mắt vào năm 2017. Flutter cho phép các nhà phát triển xây dựng ứng dụng giao diện người dùng (UI) chất lượng cao trên nhiều nền tảng, bao gồm iOS, Android, web và máy tính để bàn, từ một mã nguồn duy nhất. Đây là công nghệ nổi bật nhờ khả năng tạo ra giao diện đẹp, mượt mà và hiệu suất cao.

Một số đặc điểm nổi bật:

* Flutter là framework UI dành cho phát triển ứng dụng đa nền tảng.
* Sử dụng ngôn ngữ Dart làm ngôn ngữ chính để phát triển.
* Flutter cung cấp tính năng hot reload, cho phép các nhà phát triển thấy ngay các thay đổi trong mã nguồn mà không cần khởi động lại ứng dụng.
* Cung cấp bộ công cụ giao diện (UI toolkit) phong phú, với các widget được tối ưu hóa cho cả thiết kế Material Design (Google) và Cupertino (iOS).
* Hỗ trợ xây dựng ứng dụng trên nhiều nền tảng, bao gồm Android, iOS, web, macOS, Windows và Linux.
* Flutter không phụ thuộc vào các thành phần giao diện gốc của hệ điều hành (native components) mà tự vẽ toàn bộ giao diện, giúp đồng bộ trải nghiệm trên mọi nền tảng.

Flutter nhanh chóng trở thành một trong những framework phát triển ứng dụng đa nền tảng phổ biến nhất hiện nay. Nhiều công ty lớn đã ứng dụng Flutter trong các sản phẩm của họ, bao gồm Google Ads, Alibaba, BMW, và Tencent. Một số điểm mạnh của Flutter về cộng đồng và sự phổ biến:

* Cộng đồng đông đảo: Flutter có cộng đồng lập trình viên toàn cầu với tài liệu phong phú và sự hỗ trợ từ Google.
* Hệ sinh thái phát triển mạnh: Nhiều gói (package) mã nguồn mở được chia sẻ để hỗ trợ các tính năng từ cơ bản đến nâng cao.
* Hỗ trợ lâu dài: Google tiếp tục cập nhật và cải tiến Flutter để đáp ứng nhu cầu của thị trường.

Ưu điểm flutter:

* Hiệu suất cao: Flutter biên dịch trực tiếp mã nguồn Dart sang mã máy (native code), mang lại hiệu suất vượt trội.
* Đa nền tảng: Xây dựng ứng dụng cho nhiều nền tảng từ một mã nguồn duy nhất, tiết kiệm thời gian và chi phí phát triển.
* Giao diện đẹp và linh hoạt: Với hệ thống widget mạnh mẽ, Flutter cho phép tạo ra giao diện người dùng đẹp, nhất quán và tùy chỉnh dễ dàng.
* Hot reload: Tăng tốc độ phát triển bằng cách cho phép xem ngay các thay đổi trong thời gian thực.
* Cộng đồng mạnh mẽ: Flutter có một cộng đồng lập trình viên sôi động, giúp giải quyết vấn đề và cung cấp tài nguyên hỗ trợ nhanh chóng.

Flutter là một công nghệ tiên tiến và hiện đại, giúp đơn giản hóa quy trình phát triển ứng dụng đa nền tảng với giao diện đẹp và hiệu suất cao. Với sự hỗ trợ từ Google và cộng đồng lập trình viên toàn cầu, Flutter tiếp tục là lựa chọn hàng đầu cho các nhà phát triển muốn xây dựng ứng dụng nhanh chóng, hiệu quả và đồng nhất trên nhiều nền tảng.

### *2.1.3 Fire base*

Firebase là một nền tảng phát triển ứng dụng toàn diện (Backend-as-a-Service, BaaS) do Google phát triển. Firebase cung cấp các công cụ và dịch vụ mạnh mẽ giúp các nhà phát triển xây dựng, cải thiện và mở rộng ứng dụng trên nhiều nền tảng như iOS, Android, web và máy tính để bàn. Được ra mắt lần đầu vào năm 2011 và chính thức được Google mua lại vào năm 2014, Firebase đã trở thành một phần không thể thiếu trong hệ sinh thái phát triển ứng dụng hiện đại.

Một số đặc điểm nổi bật:

* Realtime Database: Lưu trữ và đồng bộ hóa dữ liệu trong thời gian thực giữa các client và server.
* Cloud Firestore: Cung cấp cơ sở dữ liệu NoSQL hiệu năng cao, dễ dàng mở rộng và tích hợp tốt với các dịch vụ khác của Firebase.
* Authentication: Cung cấp các phương thức xác thực người dùng như email, Google, Facebook, GitHub, Twitter và số điện thoại.
* Cloud Storage: Giải pháp lưu trữ mạnh mẽ và an toàn cho các tập tin (hình ảnh, video, v.v.).
* Cloud Messaging: Hỗ trợ gửi thông báo đẩy (push notifications) tới người dùng trên các thiết bị và nền tảng khác nhau.
* Crashlytics: Công cụ theo dõi và báo cáo lỗi giúp cải thiện hiệu suất và ổn định của ứng dụng.
* Analytics: Cung cấp thông tin chi tiết về hành vi người dùng, hiệu suất ứng dụng và doanh thu.
* Hosting: Dịch vụ lưu trữ nội dung web tĩnh và ứng dụng web một cách nhanh chóng và an toàn

Firebase là một trong những nền tảng phát triển ứng dụng phổ biến nhất nhờ vào sự hỗ trợ toàn diện từ Google và tính tiện lợi trong sử dụng. Firebase đang được sử dụng bởi nhiều công ty lớn và các nhà phát triển cá nhân trên toàn thế giới nhờ những lợi thế sau:

* Dễ tích hợp: Firebase dễ dàng tích hợp với các framework phổ biến như Flutter, React Native, và Angular.
* Tài liệu phong phú: Firebase cung cấp tài liệu chi tiết và các khóa học trực tuyến giúp nhà phát triển dễ dàng tiếp cận.
* Cộng đồng lớn: Cộng đồng hỗ trợ rộng lớn giúp giải quyết các vấn đề nhanh chóng và chia sẻ tài nguyên hữu ích.

Ưu điểm của Firebase:

* Hỗ trợ đa nền tảng: Firebase có thể được sử dụng trên iOS, Android, web, và máy tính để bàn.
* Tiết kiệm thời gian và chi phí: Firebase loại bỏ nhu cầu phải xây dựng cơ sở hạ tầng phía sau (backend), giúp nhà phát triển tập trung vào giao diện và trải nghiệm người dùng.
* Cập nhật thời gian thực: Firebase Realtime Database và Firestore cho phép đồng bộ hóa dữ liệu tức thời, rất hữu ích cho các ứng dụng chat, bản đồ, và cộng tác.
* Bảo mật mạnh mẽ: Firebase sử dụng cơ chế xác thực và các quy tắc bảo mật linh hoạt để bảo vệ dữ liệu.

Khả năng mở rộng tốt: Firebase có thể mở rộng để hỗ trợ các ứng dụng từ nhỏ đến lớn, phục vụ hàng triệu người dùng.

Firebase là một giải pháp toàn diện và mạnh mẽ dành cho các nhà phát triển muốn xây dựng và quản lý ứng dụng hiệu quả trên nhiều nền tảng. Với sự hỗ trợ từ Google, các dịch vụ đa dạng và cộng đồng lớn mạnh, Firebase là một lựa chọn lý tưởng cho các dự án ứng dụng hiện đại, từ quy mô nhỏ đến lớn.

## 2.2 Phân tích và thiết kế

### *2.2.1 Đặc tả yêu cầu của phần mềm*

Bảng 2.1 Đăng ký

|  |  |
| --- | --- |
| **Function (Chức năng)** | Đăng ký |
| **Description (Mô tả)** | Người dùng đăng ký tài khoản để đăng nhập và ứng dụng. |
| **Inputs (Đầu vào)** | Người dùng nhập các thông tin họ và tên, số điện thoại, mật khẩu, xác nhận mật khẩu. |
| **Source** |  |
| **Outputs (Đầu ra/ kết quả)** | Tạo được tài khoản từ thông tin cung cấp |
| **Destination** | Truy cập vào màn hình đăng nhập. |
| **Action** | Người dùng nhập thông tin vào form đăng ký sau đó hệ thống sẽ kiểm tra tài khoản có tạo được hay không, nếu không sẽ yêu cầu người dùng nhập lại thông tin chính xác |
| **Requirements (Yêu cầu)** | Người dùng nhập đầy đủ thông tin |
| **Pre-condition** |  |
| **Post-condition** |  |
| **Side effects** | Không. |

Bảng 2.2 Đăng nhập

|  |  |
| --- | --- |
| **Function (Chức năng)** | Đăng nhập |
| **Description (Mô tả)** | Người dùng đăng nhập vào hệ thống sau khi đã đăng ký tài khoản |
| **Inputs (Đầu vào)** | Số điện thoại, mật khẩu |
| **Source** | Được lấy dữ liệu từ cơ sở dữ liệu SQLite và Firebase |
| **Outputs (Đầu ra/ kết quả)** | Đúng số điện thoại đã đăng ký trước đó |
| **Destination** | Truy cập vào màn hình Home |
| **Action** | Người dùng đăng nhập bằng số điện thoại đã đăng ký. Nếu tài khoản đúng thì đăng nhập thành công vào ứng dụng, nếu đăng nhập sai thì yêu cầu nhập lại thông tin số điện thoại và mật khẩu. |
| **Requirements (Yêu cầu)** | Người dùng nhập số điện thoại và mật khẩu |
| **Pre-condition** | Số điện thoại đúng định dạng |
| **Post-condition** | Số điện thoại, mật khẩu phải chính xác |
| **Side effects** | Không. |

Bảng 2.3 Tìm kiếm sản phẩm

|  |  |
| --- | --- |
| **Function (Chức năng)** | Tìm kiếm sản phẩm |
| **Description (Mô tả)** | 1. Tìm kiếm sản phẩm theo tên của sản phẩm 2. Người dùng sẽ nhập tên sản phẩm mà họ muốn tìm trong hệ thóng 3. Nếu trong hệ thống có sản phẩm giống tên với tên người dùng nhập, hệ thống sẽ trả lại kết quả là các sản phẩm đó, còn không thì hệ thống không trả lại gì. |
| **Inputs (Đầu vào)** | 1. Các ký tự |
| **Source** |  |
| **Outputs (Đầu ra/ kết quả)** |  |
| **Destination** |  |
| **Action** | Người dùng chọn nhấp vào thanh tìm kiếm ở Home. Sau đó, người dùng nhập tên sản phẩm mà mình muốn tìm, nếu trong hệ thống có sản phẩm đó thì sẽ trả lại kết quả còn không thì sẽ không trả lại kết quả. |
| **Requirements (Yêu cầu)** |  |
| **Pre-condition** | Người dùng phải nhập các ký tự |
| **Post-condition** | Nếu tên sản phẩm người dùng nhập trùng với tên sản phẩm muốn tìm thì trả về các sản phẩm ddos |
| **Side effects** | Không. |

Bảng 2.4 Quản lý sản phẩm giỏ hàng

|  |  |
| --- | --- |
| **Function (Chức năng)** | Quản lý sản phẩm giỏ hàng |
| **Description (Mô tả)** | Thêm, chỉnh sửa, xóa sản phẩm ra khỏi giỏ hàng. |
| **Inputs (Đầu vào)** |  |
| **Source** |  |
| **Outputs (Đầu ra/ kết quả)** | Sản phẩm được thêm vào giỏ hàng, chỉnh sửa số lượng sản phẩm, xóa sản phẩm ra khỏi giỏ hàng. |
| **Destination** |  |
| **Action** | Người dùng chọn xem sản phẩm muốn mua và thêm sản phẩm đó vào giỏ hàng, trong giỏ hàng người dùng có thể chỉnh sửa số lượng sản phẩm, xóa sản phẩm ra khỏi giỏ hàng. |
| **Requirements (Yêu cầu)** |  |
| **Pre-condition** |  |
| **Post-condition** |  |
| **Side effects** | Không. |

Bảng 2.5 Thanh toán

|  |  |
| --- | --- |
| **Function (Chức năng)** | Thanh toán |
| **Description (Mô tả)** | Người dùng có thể thanh toán sản phẩm mình muốn mua. |
| **Inputs (Đầu vào)** |  |
| **Source** |  |
| **Outputs (Đầu ra/ kết quả)** |  |
| **Destination** |  |
| **Action** | Người dùng chọn sản phẩm muốn mua sau đó vào giỏ hàng chỉnh sửa số lượng muốn mua và tiến hành thanh toán, nhập các thông tin cá nhân(địa chỉ giao hàng, họ tên, …) sau đó sẽ tiến hành thanh toán |
| **Requirements (Yêu cầu)** |  |
| **Pre-condition** |  |
| **Post-condition** |  |
| **Side effects** | Không. |

Bảng 2.6 Chỉnh sửa thông tin người dùng

|  |  |
| --- | --- |
| **Function (Chức năng)** | Chỉnh sửa thông tin người dùng |
| **Description (Mô tả)** | Người dùng muốn chỉnh sửa các thông tin(có thể thay đổi) của tài khoản. |
| **Inputs (Đầu vào)** | Thông tin muốn thay đổi. |
| **Source** | Được lấy dữ liệu từ cơ sở dữ liệu Firebase. |
| **Outputs (Đầu ra/ kết quả)** | Thông tin được thay đổi và cập nhật lên cơ sở dữ liệu Firebase. |
| **Destination** |  |
| **Action** | Người dùng nhập vào những thông tin mà mình muốn thay đổi. Nếu thông tin hợp lệ thì hệ thống sẽ tiến hành cập nhật dữ liệu mới của tài khoản lên cơ sở dữ liệu và thông báo thành công, nếu không hợp lệ sẽ thông báo và trả về thông tin cũ của tài khoản. |
| **Requirements (Yêu cầu)** | Người dùng nhập thông tin mới vào. |
| **Pre-condition** | Thông tin thỏa yêu cầu hệ thống. |
| **Post-condition** | Thông tin mới phải khác với thông tin hiện tại. |
| **Side effects** | Không. |

Bảng 2.7 Quản lý danh mục yêu thích

|  |  |
| --- | --- |
| **Function (Chức năng)** | Quản lý danh mục yêu thích |
| **Description (Mô tả)** | Thêm, sửa, xóa sản phẩm vào danh mục yêu thích. |
| **Inputs (Đầu vào)** |  |
| **Source** |  |
| **Outputs (Đầu ra/ kết quả)** | Sản phẩm được thêm vào danh mục yêu thích hoặc xóa ra khỏi danh mục yêu thích. |
| **Destination** |  |
| **Action** | Người dùng chọn sản phẩm muốn thêm vào danh mục yêu thích và nhấn biểu tượng trái tim kế bên sản phẩm, sản phẩm đó sẽ được thêm vào danh mục yêu thích. Người dùng muốn xóa sản phẩm ra khỏi danh mục yếu thích chỉ cần nhấn vào biểu tượng thùng rác của sản phẩm đó. |
| **Requirements (Yêu cầu)** |  |
| **Pre-condition** |  |
| **Post-condition** |  |
| **Side effects** | Không. |

#### 2.2.1.2 Yêu cầu chức năng

Chức năng chính của ứng dụng

Đăng nhập ( Login )

Xác thực biểu mẫu: Kiểm tra và xác thực dữ liệu người dùng nhập vào.

Hiển thị hộp thoại lỗi: Thông báo lỗi khi có ngoại lệ từ FirebaseAuthException.

Đăng nhập bằng Firebase Auth: Sử dụng Firebase Authentication để xác thực tài khoản người dùng.

Đăng ký ( Register )

Xác thực biểu mẫu: Kiểm tra và xác thực dữ liệu đầu vào của người dùng.

Hiển thị hộp thoại lỗi: Thông báo lỗi khi có ngoại lệ từ FirebaseAuthException.

Tạo tài khoản với Firebase Auth: Đăng ký người dùng mới trong Firebase Authentication.

Xác minh tài khoản: Sử dụng OTP qua điện thoại thông qua Firebase Auth để xác minh tài khoản.

Trang chính ( Home page )

Hiển thị banner: Hiển thị các banner quảng cáo.

Hiển thị danh mục (category): Liệt kê các danh mục sản phẩm.

Hiển thị sản phẩm mới (New Product): Hiển thị danh sách sản phẩm mới.

Hiển thị sản phẩm giá giảm (Discount Price Product): Hiển thị danh sách sản phẩm đang giảm giá.

Hiển thị sản phẩm bán chạy (Best Seller Product): Liệt kê các sản phẩm bán chạy nhất.

Hiển thị sản phẩm đánh giá cao (Top Review Product): Hiển thị sản phẩm có đánh giá tốt nhất.

Thêm vào yêu thích (Add to Favorite): Thêm sản phẩm vào danh sách yêu thích.

Thêm vào giỏ hàng (Add to Cart): Thêm sản phẩm vào giỏ hàng.

Chi tiết sản phẩm ( Product Detail )

Thêm vào giỏ hàng (Add to Cart): Lưu sản phẩm vào giỏ hàng của người dùng.

Thêm vào yêu thích (Add to Favorite): Lưu sản phẩm vào danh sách yêu thích.

Hiển thị toàn màn hình (Show Full Screen Product): Xem chi tiết sản phẩm ở chế độ toàn màn hình.

Hiển thị chi tiết (Show Detail): Hiển thị thông tin chi tiết sản phẩm.

Xem sản phẩm theo màu sắc (Show Product Follow Color): Xem các phiên bản sản phẩm theo màu sắc.

Trang giỏ hàng ( Cart Page )

Xóa sản phẩm (Delete Item): Loại bỏ sản phẩm khỏi giỏ hàng.

Thay đổi số lượng (Change Amount): Thay đổi số lượng sản phẩm trong giỏ hàng.

Hiển thị sản phẩm trong giỏ (Show Cart Item): Hiển thị danh sách sản phẩm trong giỏ hàng.

Tính tổng giá tiền (Calculate Price All Item): Tính tổng giá tiền của tất cả sản phẩm trong giỏ.

Trang yêu thích ( Favorite Page )

Xóa sản phẩm (Delete Item): Loại bỏ sản phẩm khỏi danh sách yêu thích.

Hiển thị danh sách yêu thích (Show Favorite Item): Hiển thị danh sách sản phẩm đã thêm vào yêu thích.

Trang thanh toán ( Checkout Page )

Nhập thông tin khách hàng (Enter Information Customer): Nhập thông tin cá nhân để giao hàng.

Chọn phương thức thanh toán (Enter Payment Method): Lựa chọn phương thức thanh toán.

Cập nhật sản phẩm (Update Item Product): Cập nhật lại danh sách sản phẩm trong giỏ hàng trước khi thanh toán.

Xem lại thông tin và đặt hàng (Review Information and Order): Kiểm tra thông tin đơn hàng và xác nhận.

Trang tìm kiếm ( Search Page )

Tìm kiếm theo tên sản phẩm (Search by Name Product): Tìm kiếm sản phẩm bằng tên.

Hiển thị sản phẩm theo danh mục (Show Product by Category): Hiển thị danh sách sản phẩm theo danh mục.

Lọc sản phẩm (Filter Product): Lọc sản phẩm theo tiêu chí.

Trang đơn hàng ( Order Page )

Hiển thị đơn hàng (Show Order): Hiển thị các đơn hàng với trạng thái:

CHECKING: Đang chờ xác nhận.

DELIVERING: Đang giao hàng.

COMPLETE: Đã giao hàng thành công.

Hiển thị chi tiết đơn hàng (Show Order Detail): Xem thông tin chi tiết về từng đơn hàng.

Hủy đơn hàng (Cancel Order): Hủy các đơn hàng có trạng thái CHECKING.

Trang cài đặt ( Setting Page )

Trang cá nhân (Profile Page):

Hiển thị thông tin người dùng.

Cập nhật thông tin cá nhân.

Cập nhật hình ảnh đại diện.

Trang giỏ hàng (Cart Page): Truy cập và quản lý giỏ hàng từ đây.

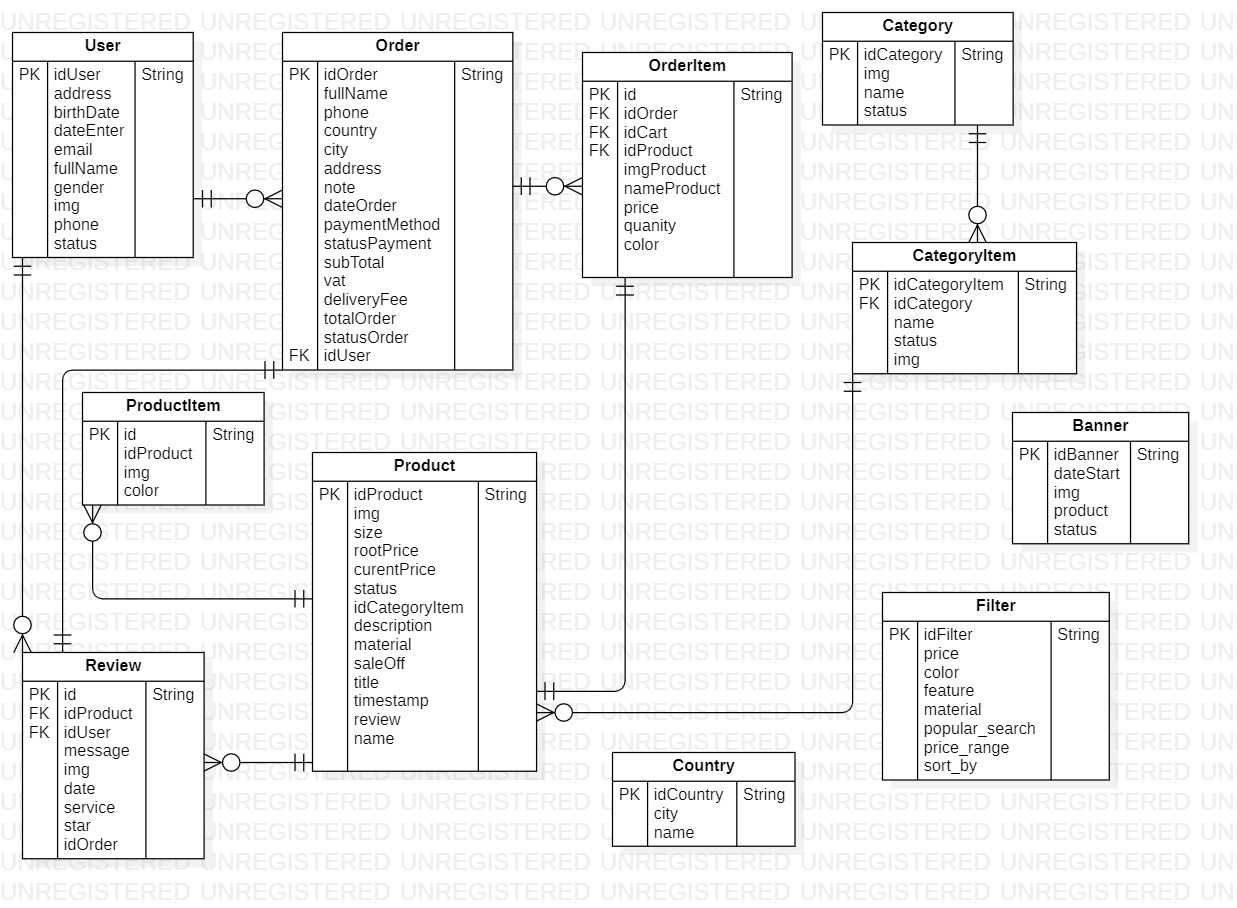
Trang yêu thích (Favorite Page): Truy cập và quản lý danh sách yêu thích.

Trang đơn hàng (Order Page): Truy cập và quản lý các đơn hàng.

Đăng xuất (Log out): Thoát khỏi tài khoản hiện tại.

### *2.2.2 Mô hình hóa dữ liệu*

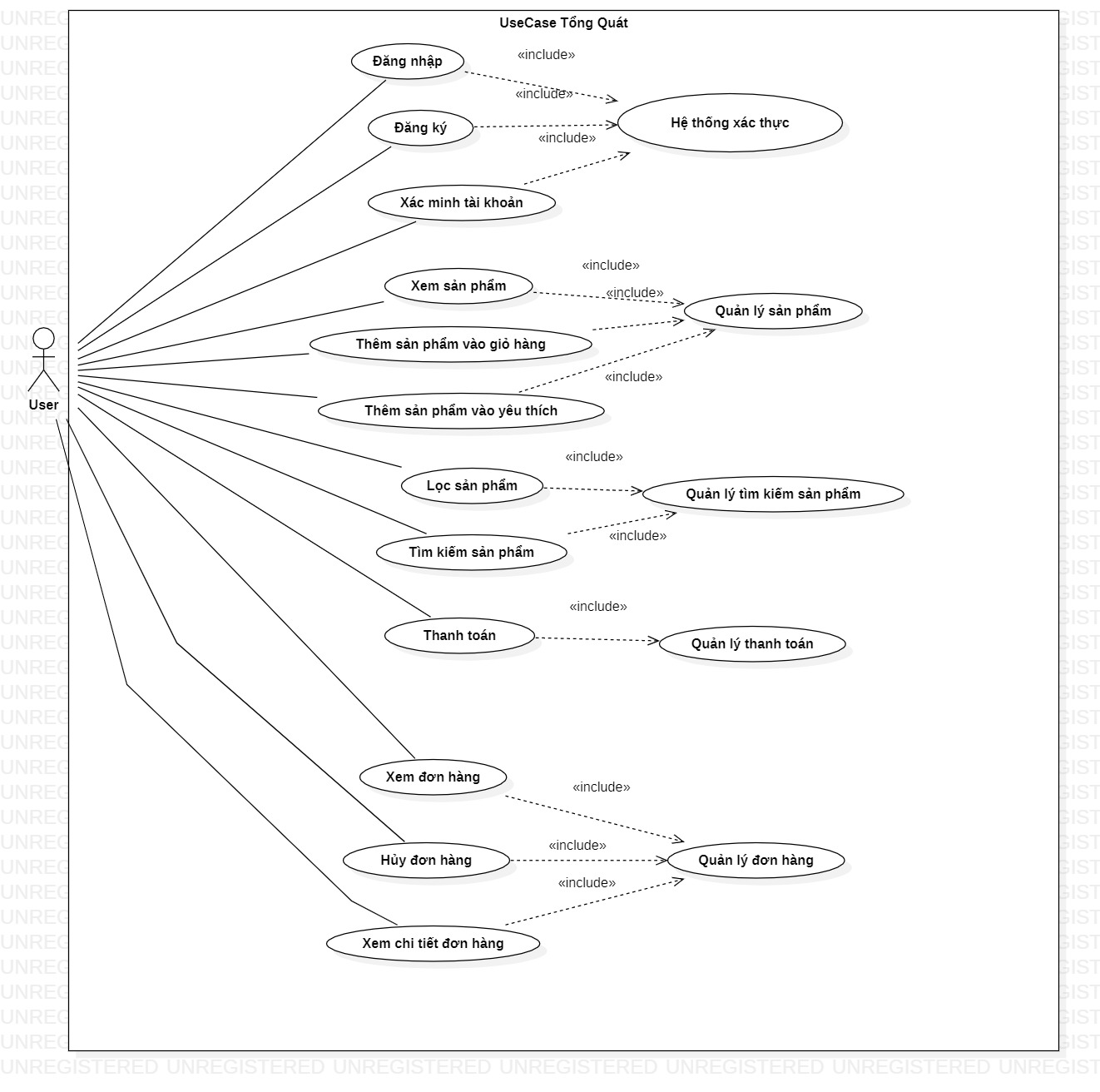
#### 2.2.2.1 Mô hình ERD



Hình 2.1 Mô hình ERD

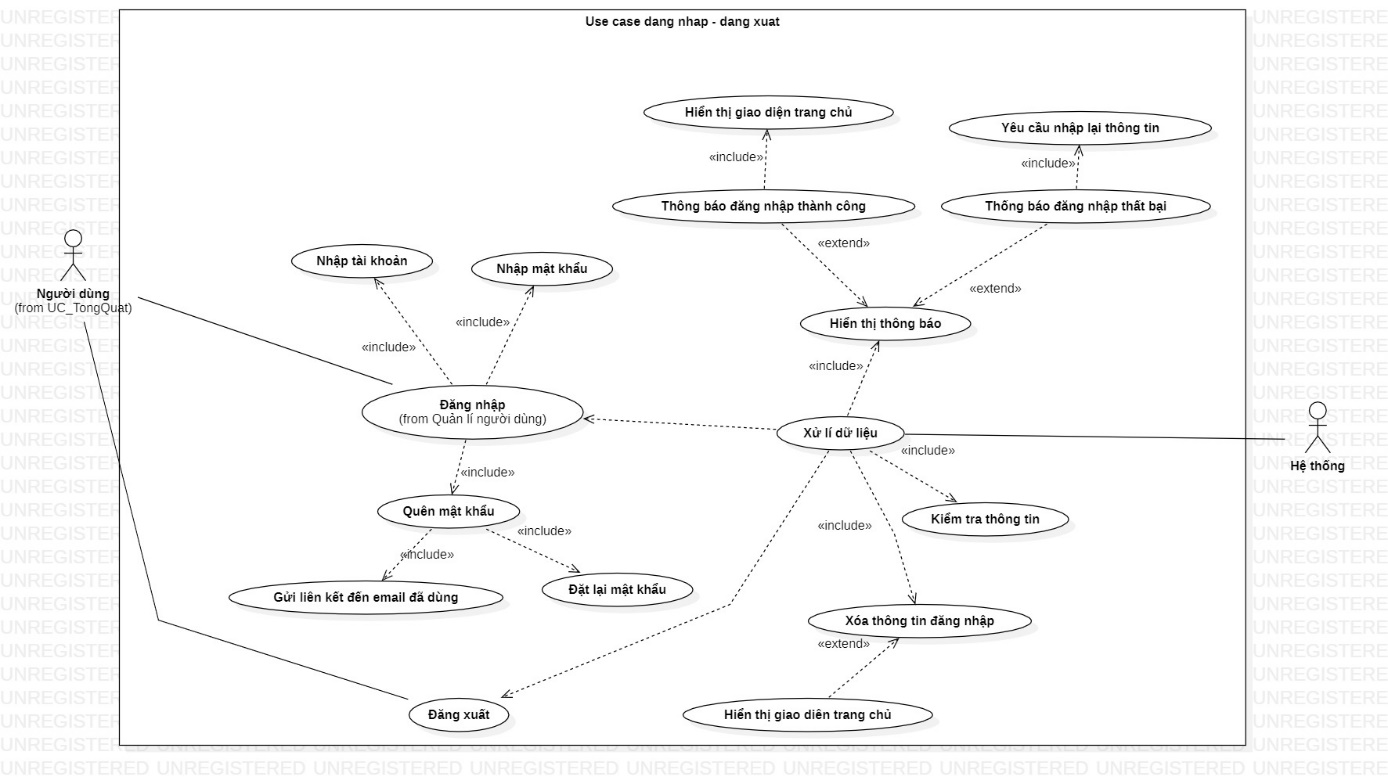
#### 2.2.2.2 Mô hình Use case

##### 2.2.2.2.1 Use case tổng quát



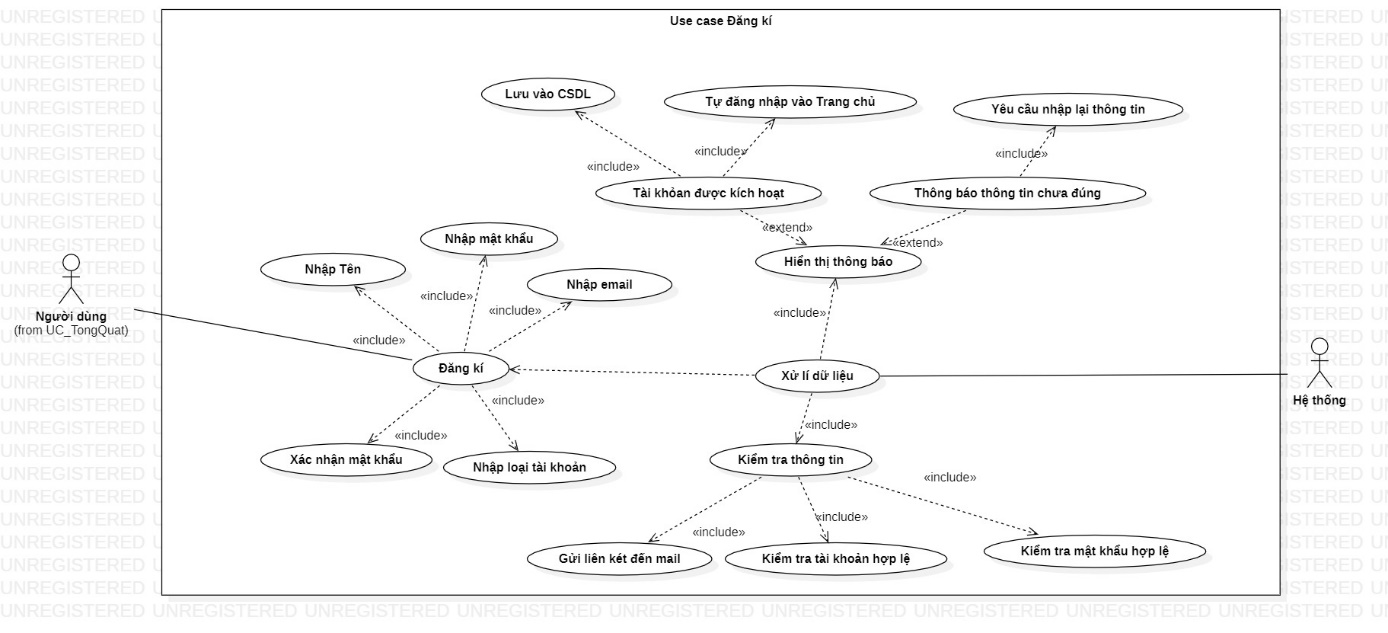
Hình 2.2 Mô hình Usecase Tổng quát

##### 2.2.2.2.2 Use case phân rã chức năng Đăng nhập - đăng suất



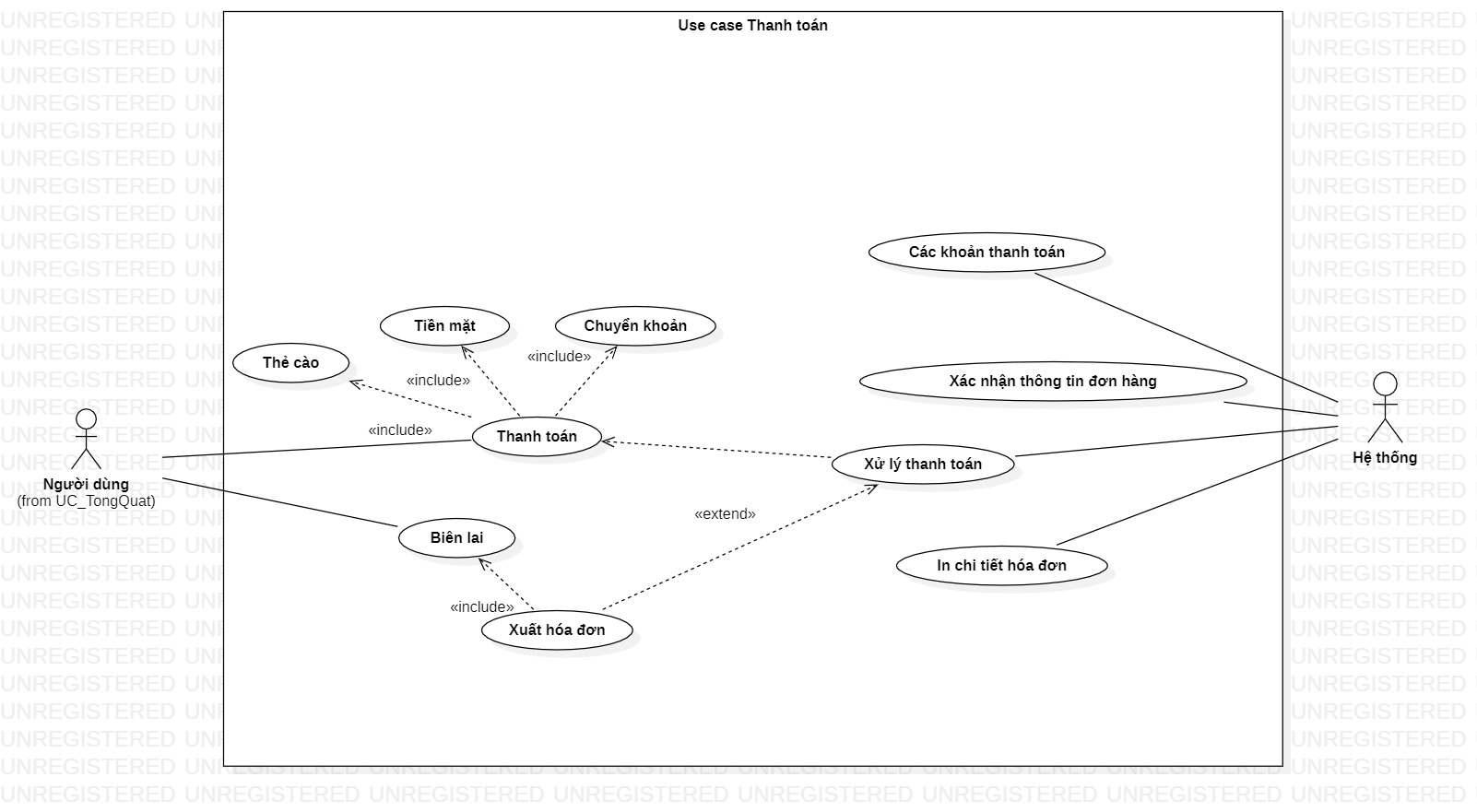
Hình 2.3 Mô hình Usecase phân rã cức năng Đăng nhập-đăng xuất

##### 2.2.2.2.3 Use case phân rã chức năng Đăng ký



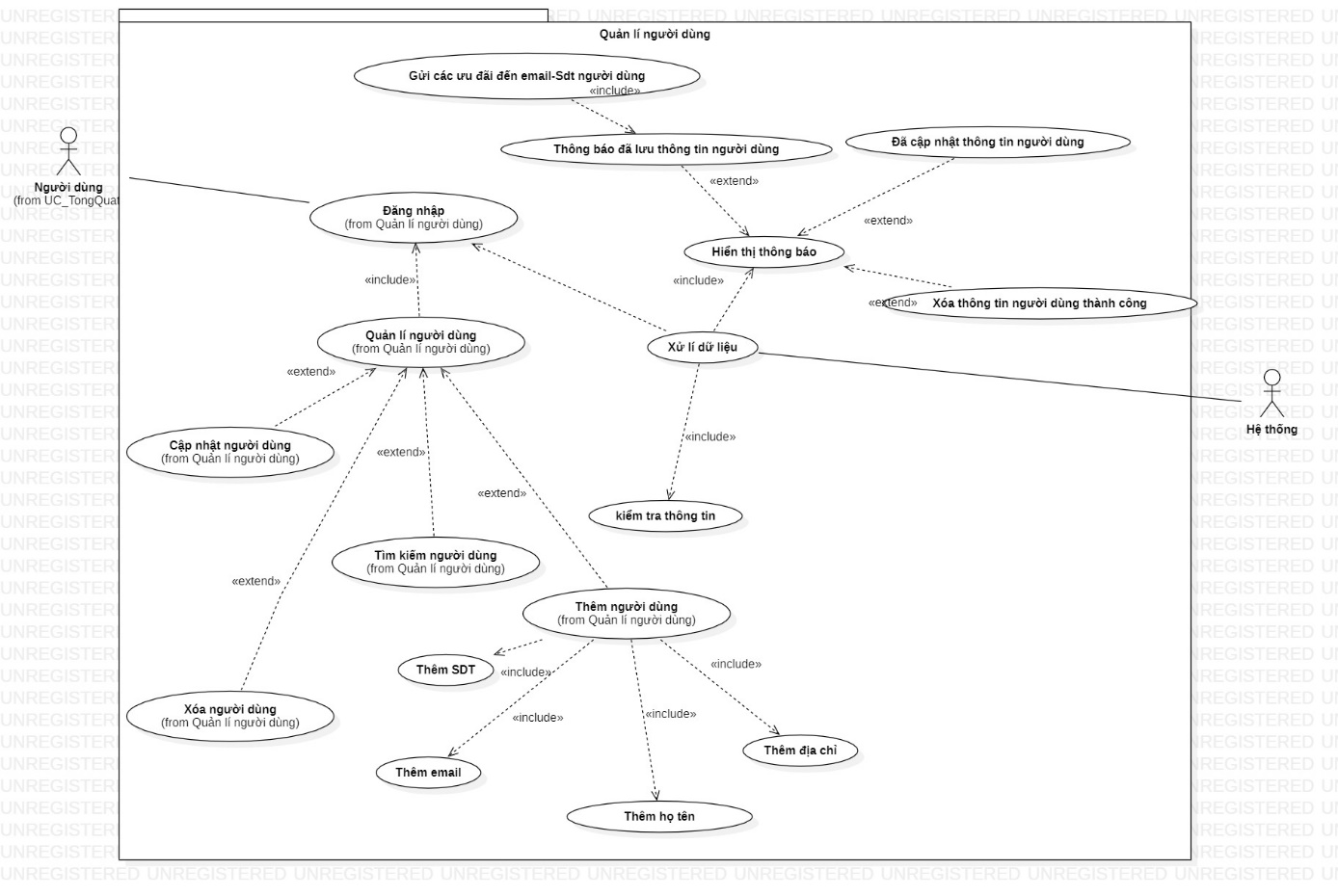
Hình 2.4 Mô hình Usecase phân rã chức năng Đăng ký

##### 2.2.2.2.4 Use case phân rã chức năng Thanh Toán



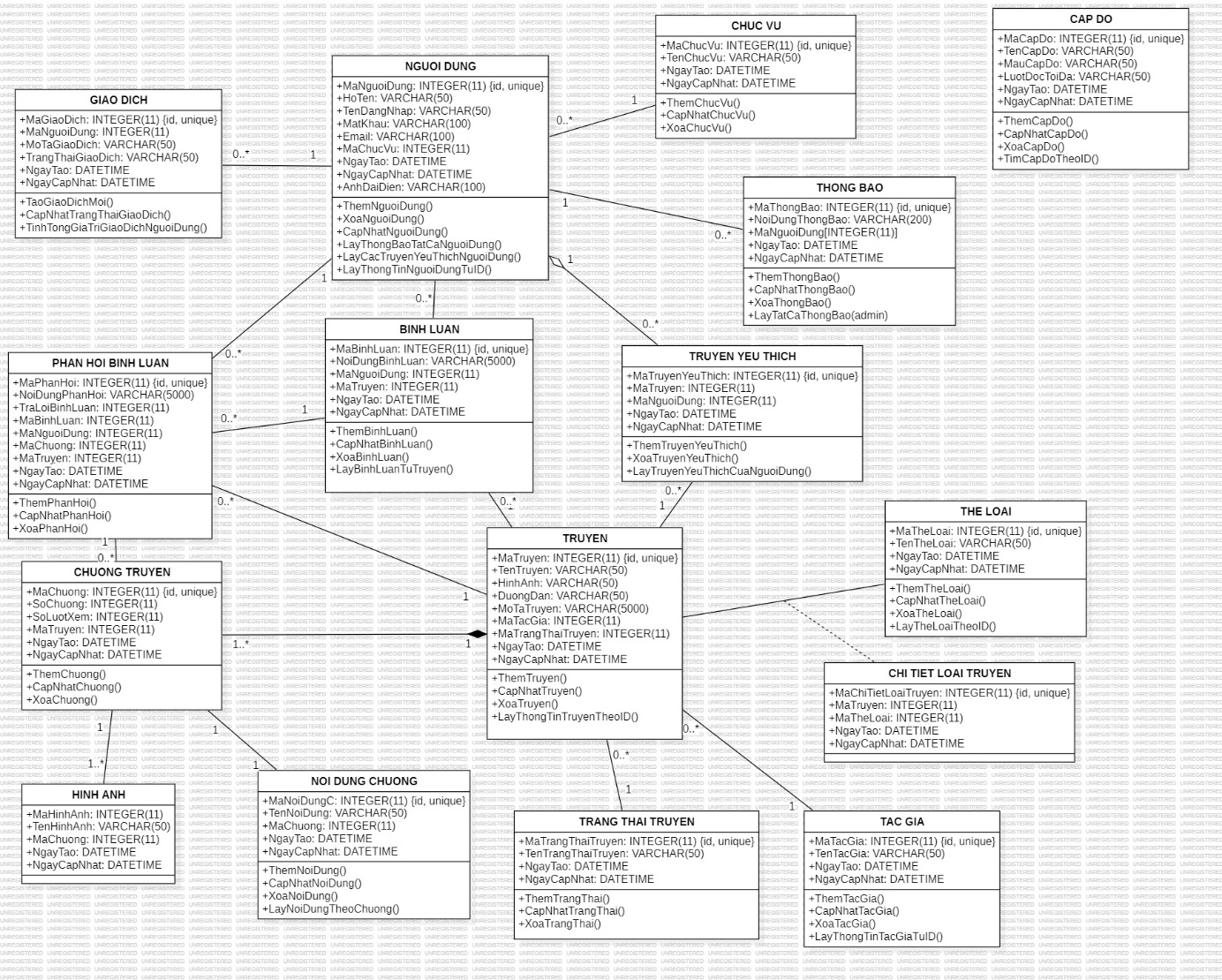
Hình 2.5 Mô hình Usecase phân rã chức năng Thanh toán

##### 2.2.2.2.5 Use case phân rã chức năng Quản lý người dùng



Hình 2.6 Mô hình Usecase phân rã chức năng Quản lý người dùng

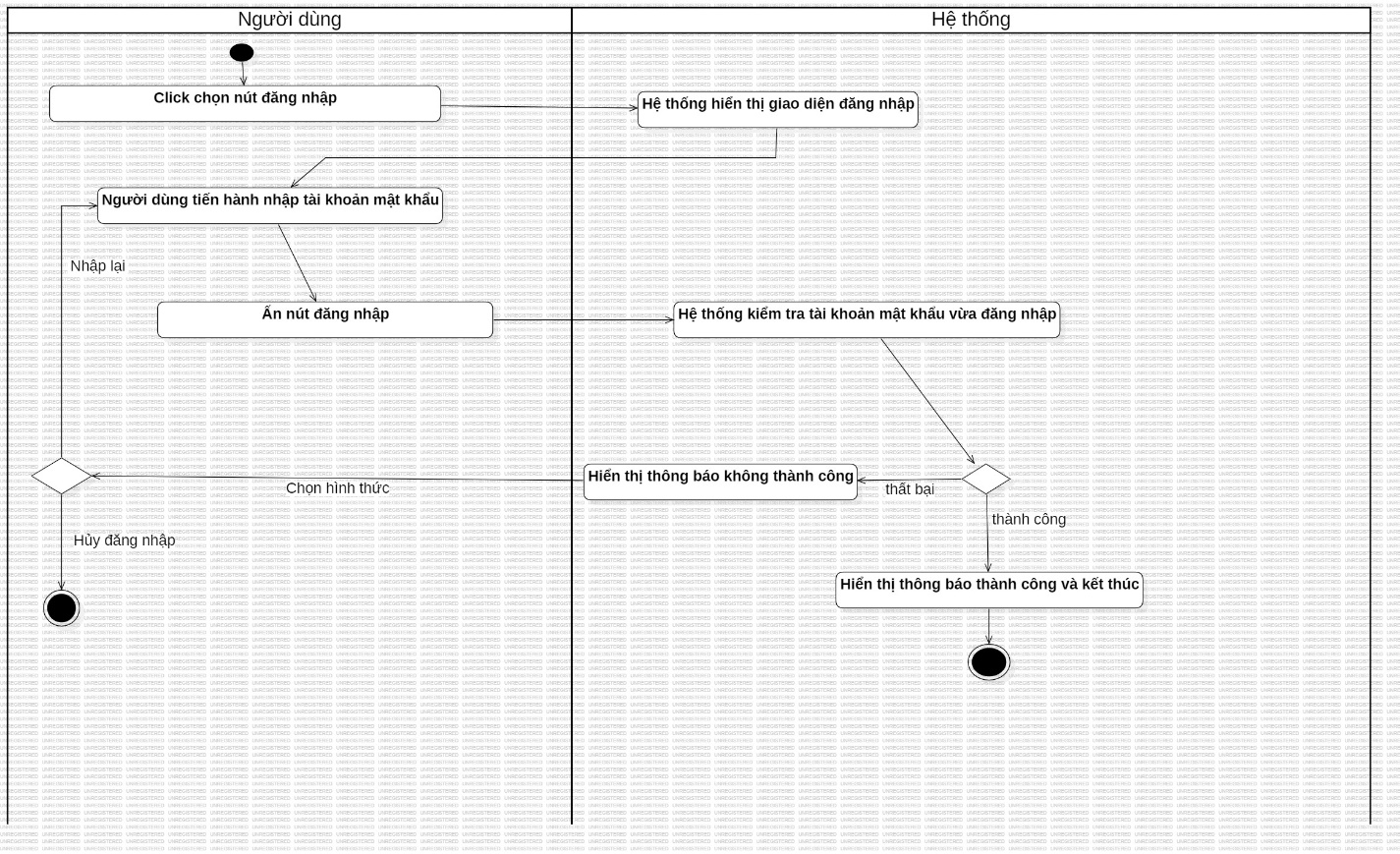
#### 2.2.2.3 Mô hình Class diagram //chưa vẽ



Hình 2.7 Mô hình Class diagram

#### 2.2.2.4 Mô hình Activity diagram

2.2.2.4.1 Activity đăng nhập



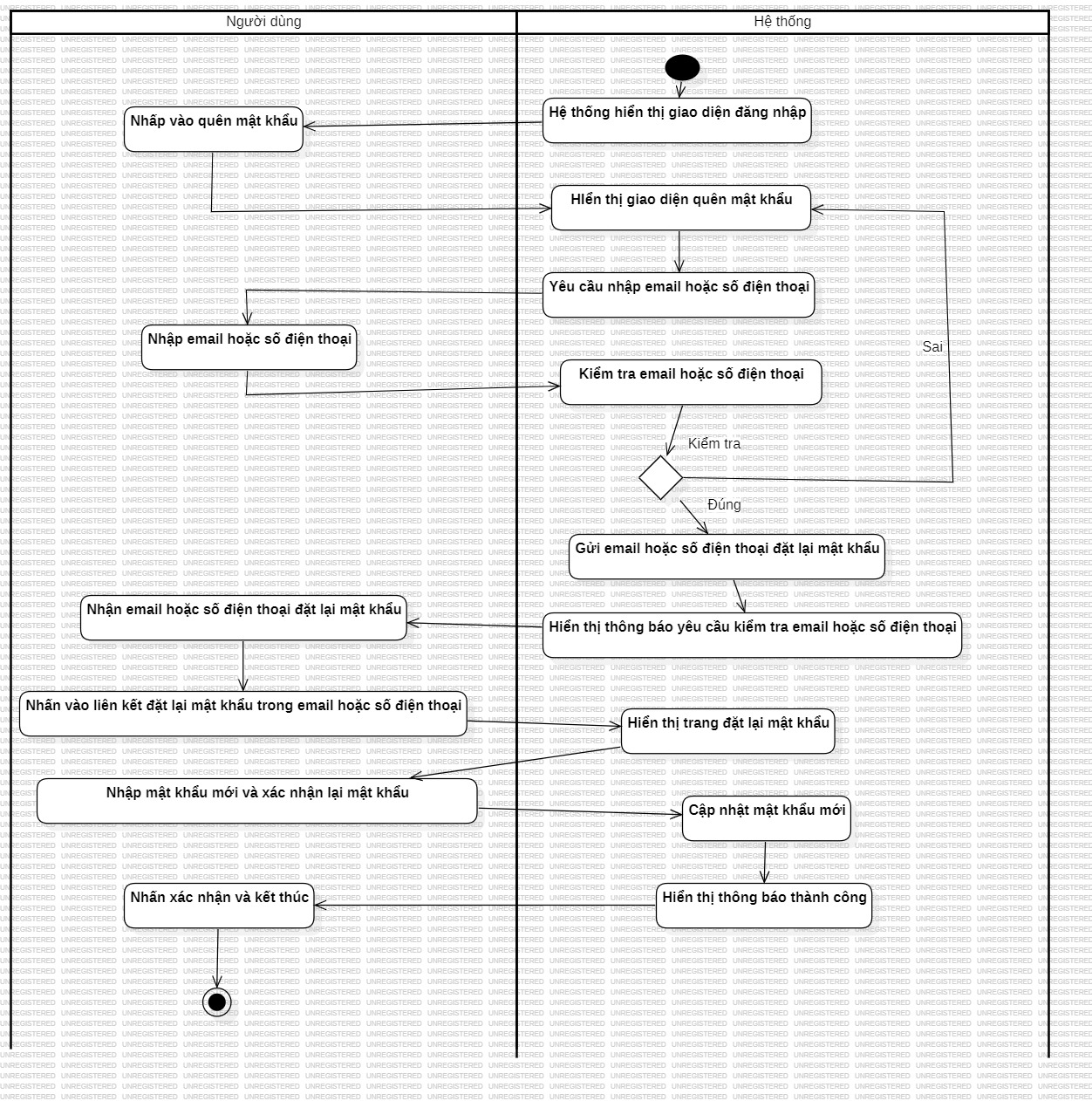
Hình 2.8 Mô hình Activity Đăng nhập

##### 2.2.2.4.2 Activity đặt sản phẩm



Hình 2.9 Mô hình Activity đặt sản phẩm

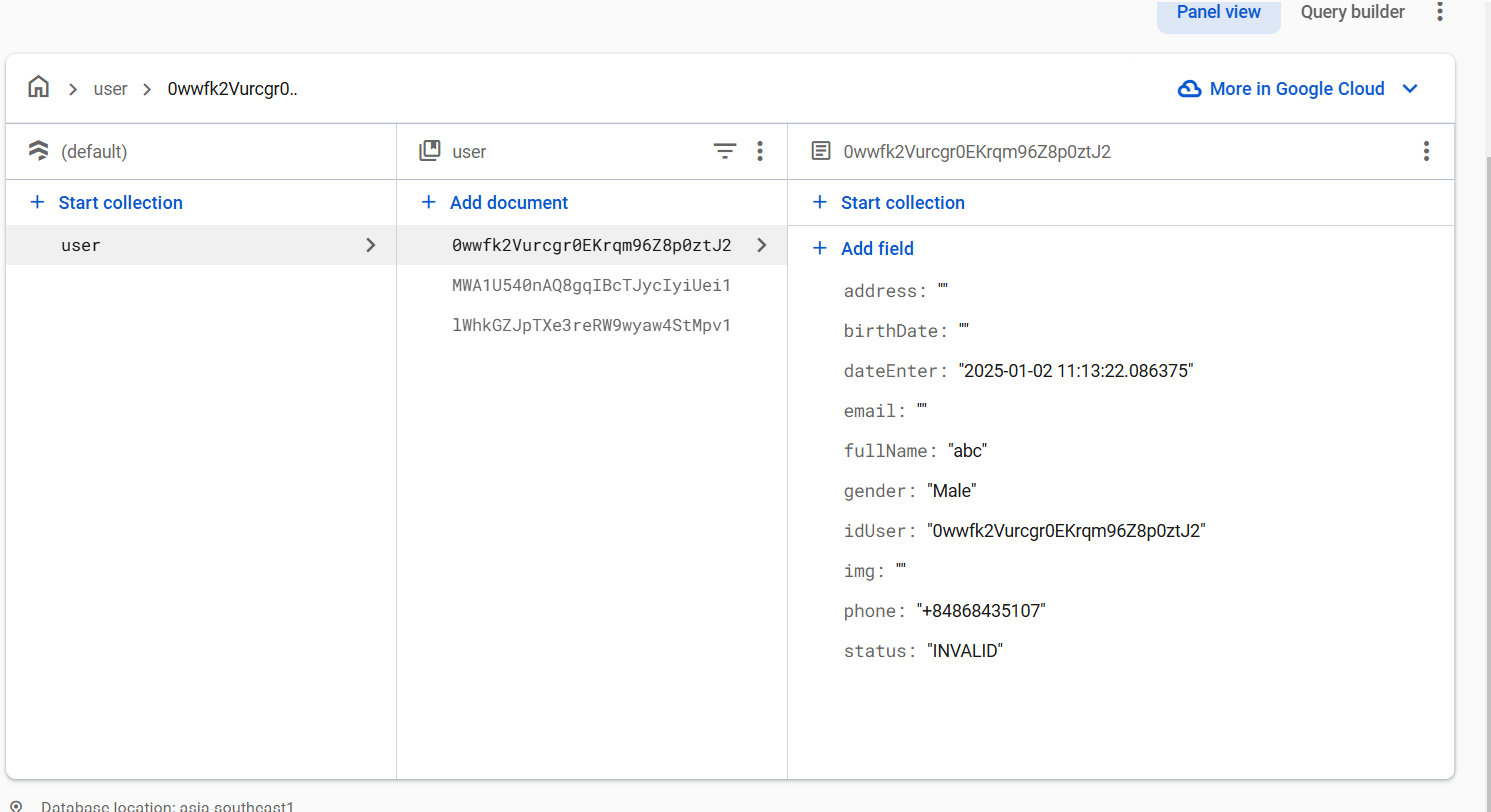
##### 2.2.2.4.4 Activity quên mật khẩu

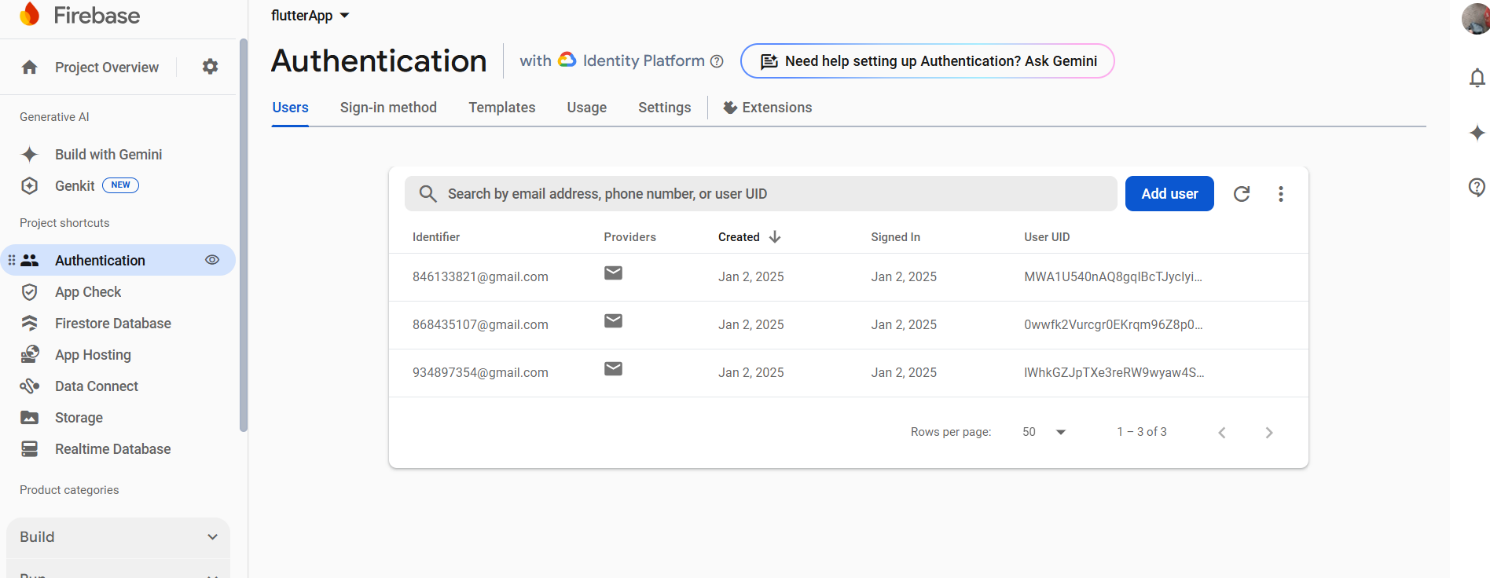


Hình 2.10 Mô hình Activity Quên mật khẩu

## 2.3 Thiết kế cơ sở dữ liệu

### *2.3.1 Cơ sở dữ liệu*





Hình 2.11 Mô hình cơ sở dữ liệu

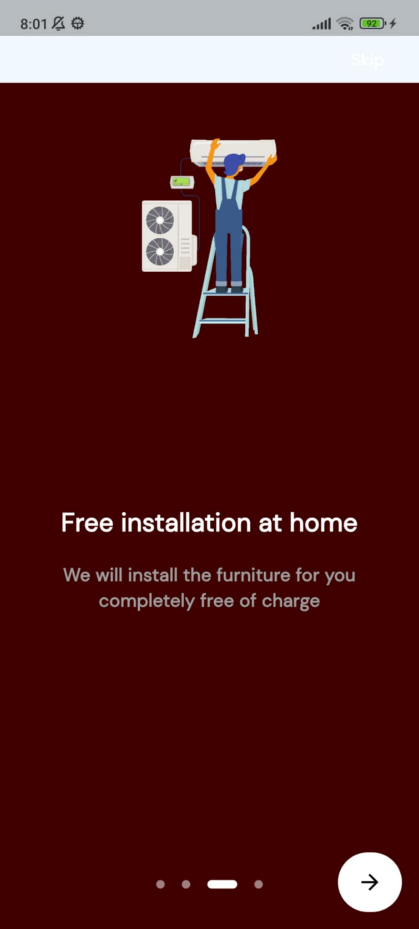
**Chương 3. KẾT QUẢ THỰC NGHIỆM**

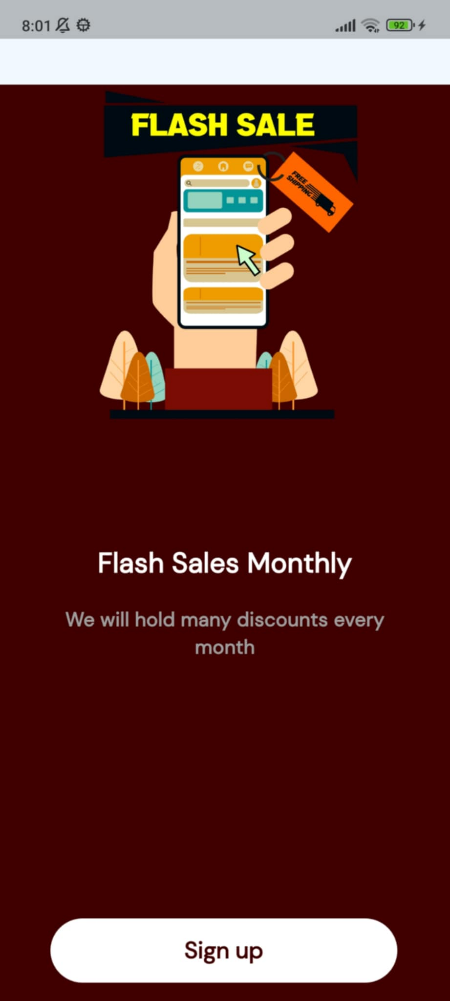
## 3.1 Kết quả

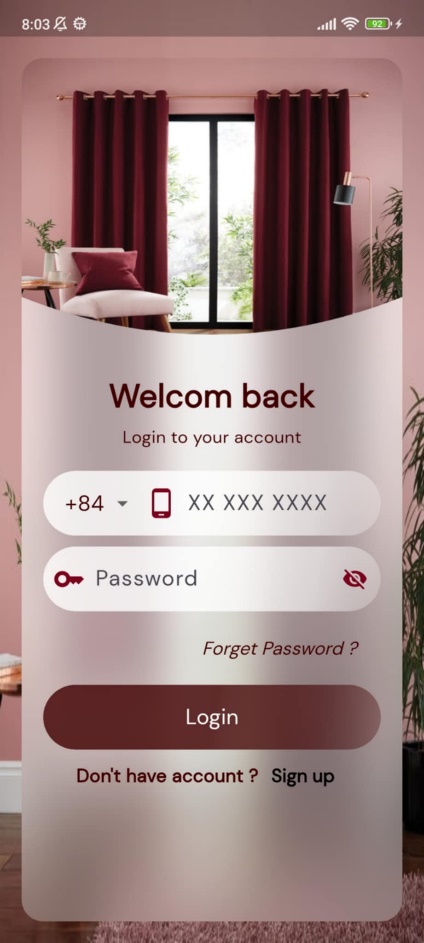
### *3.1.1 Các kết quả*

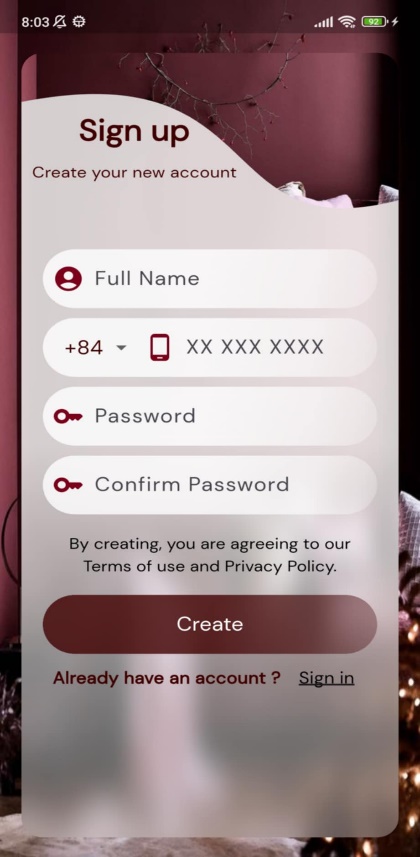


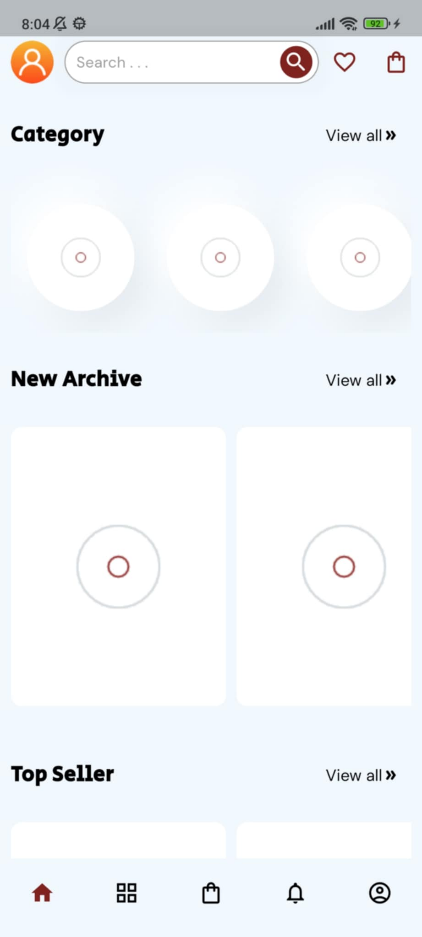




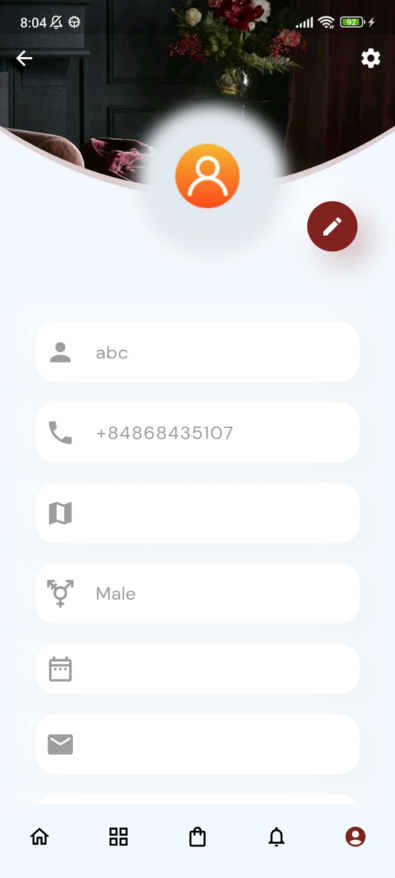


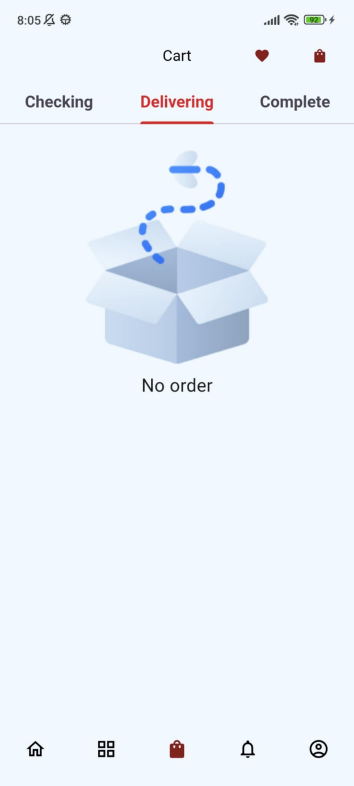




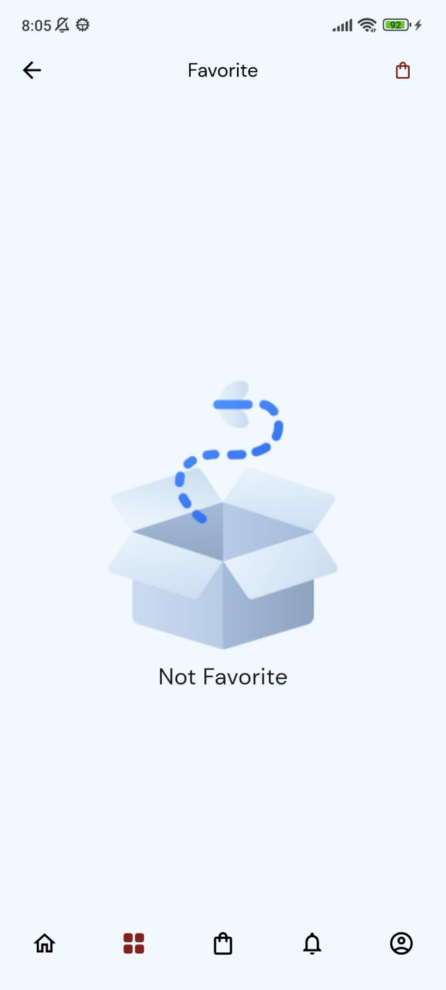


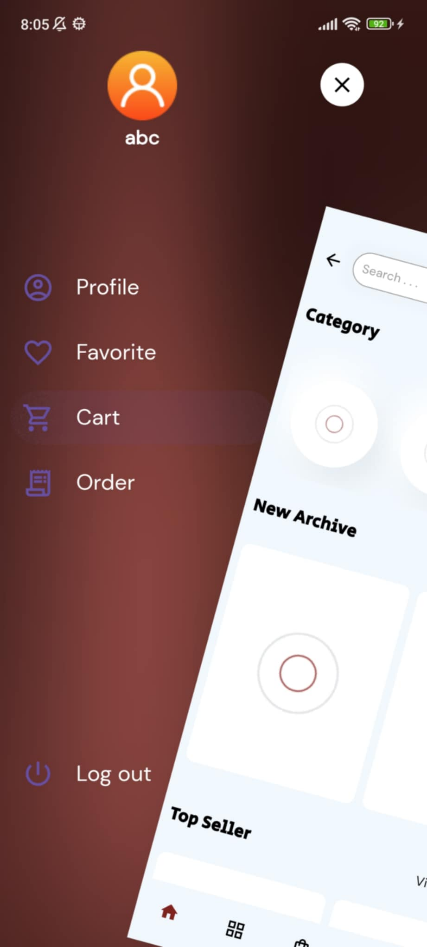


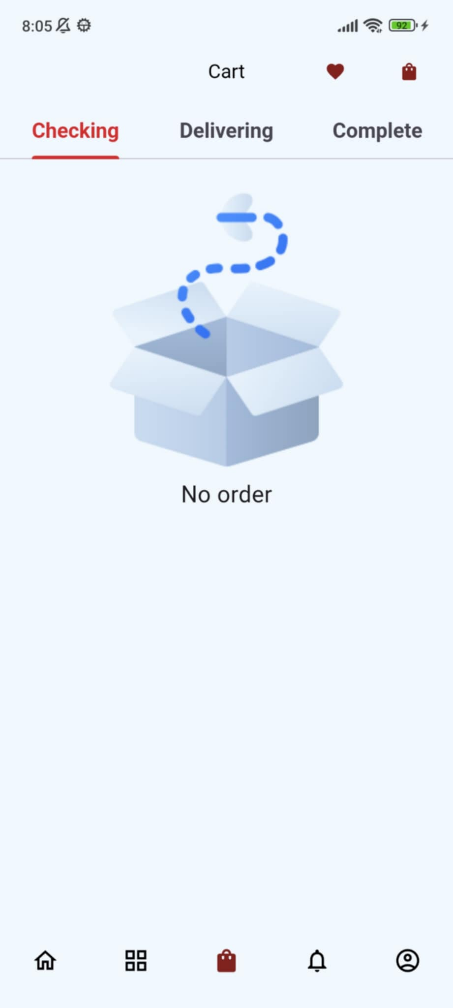


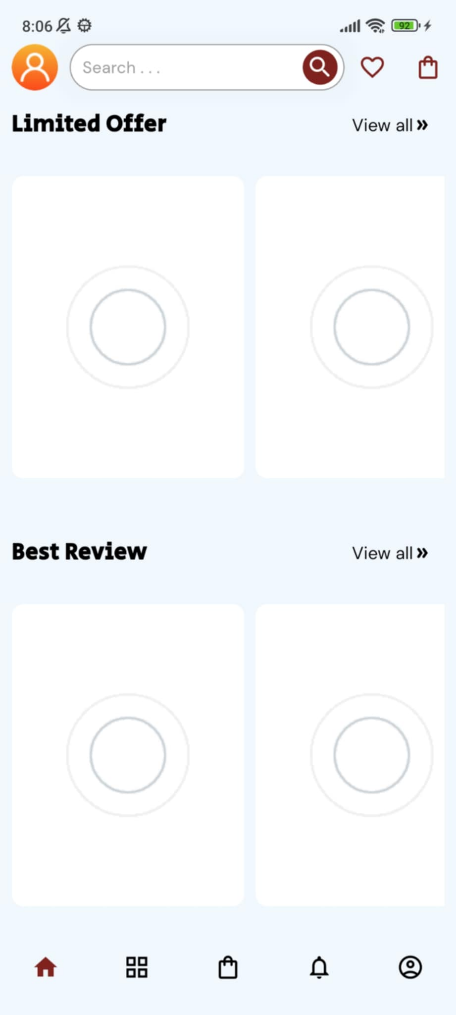












**Chương 4. KẾT LUẬN VÀ KIẾN NGHỊ**

## 4.1 Kết luận chung

Trong thời đại công nghệ số hóa phát triển mạnh mẽ, việc xây dựng nền tảng bán nội thất hiện đại trên Flutter không chỉ đáp ứng nhu cầu của thị trường mà còn góp phần định hình lại cách người tiêu dùng tiếp cận và mua sắm sản phẩm nội thất. Nền tảng này mang đến sự kết hợp hoàn hảo giữa công nghệ tiên tiến và trải nghiệm người dùng tối ưu, tạo ra một không gian mua sắm thông minh, tiện lợi và hiện đại. Ứng dụng trên Flutter đã chứng minh được tính hiệu quả nhờ vào khả năng xây dựng đa nền tảng, tối ưu hóa chi phí phát triển và vận hành, đồng thời đảm bảo giao diện đồng nhất trên cả hai hệ điều hành Android và iOS. Những tính năng tích hợp như quản lý sản phẩm, danh mục, tài khoản người dùng và báo cáo thống kê không chỉ hỗ trợ doanh nghiệp trong việc tổ chức và theo dõi vận hành mà còn cải thiện sự tương tác giữa người bán và khách hàng. Điều này tạo nên một hệ sinh thái kinh doanh toàn diện, nơi mọi quy trình đều được tự động hóa và tối ưu hóa.

Về phía khách hàng, ứng dụng mang lại trải nghiệm mua sắm liền mạch với các tính năng nổi bật như tìm kiếm sản phẩm, lưu danh sách yêu thích, đồng bộ hóa dữ liệu trên nhiều thiết bị, và nhận thông báo cập nhật sản phẩm mới. Những tiện ích này không chỉ tạo sự hài lòng mà còn thúc đẩy khách hàng quay lại sử dụng dịch vụ, đồng thời xây dựng mối quan hệ lâu dài giữa doanh nghiệp và người tiêu dùng.

Việc sử dụng Flutter trong phát triển nền tảng bán nội thất cũng tạo ra sự linh hoạt trong việc mở rộng tính năng, cập nhật giao diện và triển khai các cải tiến kỹ thuật. Điều này đặc biệt quan trọng trong bối cảnh thị trường liên tục thay đổi, đòi hỏi các doanh nghiệp phải nhanh chóng thích nghi với xu hướng và nhu cầu mới của khách hàng.

Từ góc độ kinh doanh, nền tảng này còn cung cấp các giải pháp thúc đẩy doanh thu và tối ưu hóa lợi nhuận thông qua các gói dịch vụ linh hoạt như miễn phí và VIP. Việc cung cấp quyền truy cập ưu tiên, loại bỏ quảng cáo, và các ưu đãi độc quyền giúp doanh nghiệp không chỉ duy trì nguồn thu ổn định mà còn tăng cường sức cạnh tranh trên thị trường.

Hơn thế nữa, ứng dụng bán nội thất hiện đại còn góp phần thúc đẩy văn hóa mua sắm trực tuyến, giúp khách hàng tiếp cận các sản phẩm chất lượng với sự tiện nghi tối đa. Điều này không chỉ nâng cao giá trị thương hiệu của doanh nghiệp mà còn tạo nên tác động tích cực đến thói quen tiêu dùng của người dân, góp phần thúc đẩy sự phát triển của ngành nội thất số.

Tóm lại, việc đầu tư xây dựng và phát triển nền tảng bán nội thất trên Flutter là một bước đi chiến lược, mở ra cơ hội không giới hạn cho cả doanh nghiệp và khách hàng. Đây không chỉ là công cụ hỗ trợ kinh doanh mà còn là giải pháp toàn diện giúp doanh nghiệp vượt qua thách thức, nâng cao hiệu quả hoạt động và đáp ứng nhu cầu ngày càng đa dạng của thị trường. Với sự đầu tư hợp lý và tầm nhìn dài hạn, nền tảng này sẽ trở thành một phần không thể thiếu trong hệ sinh thái kinh doanh nội thất trực tuyến, góp phần đưa ngành nội thất số lên một tầm cao mới trong tương lai.

## 4.2 Nội dung đã làm được

Thiết kế giao diện ứng dụng trực quan, đảm bảo tính thẩm mỹ và trải nghiệm người dùng cao.

Xây dựng các thành phần giao diện như danh mục sản phẩm, chi tiết sản phẩm, giỏ hàng, và thanh toán.

Sử dụng Flutter Widgets để tối ưu hóa hiệu năng và đồng nhất giao diện trên cả hai nền tảng Android và iOS.

## 4.3 Hướng phát triển

Để đáp ứng tốt hơn nhu cầu của thị trường và khách hàng, ứng dụng bán nội thất hiện đại trên nền tảng Flutter cần được phát triển thêm nhiều tính năng và giải pháp mới. Điều này không chỉ nâng cao trải nghiệm người dùng mà còn giúp doanh nghiệp tối ưu hóa vận hành, gia tăng khả năng cạnh tranh và tạo dựng nền tảng phát triển bền vững. Một trong những hướng đi quan trọng là tích hợp trí tuệ nhân tạo (AI) và học máy (Machine Learning) vào hệ thống. AI có thể phân tích dữ liệu khách hàng để hiểu rõ hơn về sở thích, nhu cầu và hành vi mua sắm, từ đó gợi ý các sản phẩm phù hợp, cá nhân hóa trải nghiệm mua sắm và nâng cao sự hài lòng của người dùng. Bên cạnh đó, AI còn giúp dự đoán xu hướng thị trường và hỗ trợ doanh nghiệp xây dựng chiến lược kinh doanh hiệu quả hơn.

Việc nâng cấp hệ thống quản lý sản phẩm và tài khoản người dùng cũng là một yếu tố cần thiết. Hệ thống quản lý sản phẩm cần linh hoạt, cho phép doanh nghiệp cập nhật thông tin, trạng thái tồn kho và giá cả dễ dàng. Đồng thời, tính năng quản lý tài khoản người dùng cần hỗ trợ lưu lịch sử mua hàng, đồng bộ hóa dữ liệu trên nhiều thiết bị và cung cấp danh sách sản phẩm yêu thích. Bên cạnh đó, tích hợp khả năng theo dõi chi tiết đơn hàng sẽ mang lại sự minh bạch và tiện lợi cho cả doanh nghiệp và khách hàng. Những cải tiến này không chỉ tối ưu hóa quy trình vận hành mà còn nâng cao sự hài lòng của khách hàng.

Phát triển tính năng thanh toán trực tuyến và dịch vụ VIP cũng đóng vai trò quan trọng trong việc cải thiện trải nghiệm người dùng. Hệ thống thanh toán cần hỗ trợ đa dạng phương thức như thẻ tín dụng, ví điện tử và các cổng thanh toán quốc tế, đảm bảo sự an toàn và tiện lợi trong giao dịch. Dịch vụ VIP cần cung cấp các đặc quyền như truy cập sớm vào sản phẩm mới, loại bỏ quảng cáo và nhận các ưu đãi độc quyền, giúp tăng giá trị dịch vụ và tạo nguồn thu ổn định cho doanh nghiệp.

Bên cạnh đó, việc phát triển công cụ tương tác và xây dựng cộng đồng người dùng sẽ giúp tạo sự gắn kết lâu dài giữa khách hàng và doanh nghiệp. Hệ thống bình luận, đánh giá sản phẩm và chương trình khách hàng thân thiết như tích điểm thưởng hoặc giảm giá là những yếu tố quan trọng để khuyến khích khách hàng quay lại mua sắm. Ngoài ra, tính năng chia sẻ sản phẩm lên mạng xã hội sẽ mở rộng khả năng tiếp cận khách hàng mới và tăng độ nhận diện thương hiệu.

Đảm bảo an ninh và bảo mật thông tin người dùng cũng là một trong những yếu tố then chốt cần được chú trọng. Hệ thống cần tích hợp mã hóa dữ liệu và chứng chỉ SSL để bảo vệ thông tin cá nhân và giao dịch. Đồng thời, các biện pháp phòng chống xâm nhập sẽ giúp ngăn chặn các mối đe dọa từ bên ngoài, đảm bảo sự ổn định và an toàn cho nền tảng.

Cuối cùng, tăng cường chiến lược marketing và quảng bá sẽ giúp ứng dụng mở rộng phạm vi tiếp cận. Sử dụng các công cụ marketing như email marketing, thông báo đẩy và phân tích dữ liệu người dùng để xây dựng chiến dịch quảng cáo chính xác sẽ mang lại hiệu quả cao trong việc thu hút khách hàng. Với những cải tiến này, ứng dụng bán nội thất hiện đại trên Flutter không chỉ cải thiện trải nghiệm người dùng mà còn tạo ra giá trị lâu dài cho doanh nghiệp, từ đó xây dựng nền tảng phát triển bền vững và đáp ứng nhu cầu ngày càng đa dạng của thị trường.

**TÀI LIỆU THAM KHẢO**

[1] https://flutter.dev/docs.

[2] [https://stackoverflow.com](https://stackoverflow.com/).

[3] [https://techcrunch.com](https://techcrunch.com/).

[4] https://github.com/

[5] Trần Minh C (2021). Tích hợp trí tuệ nhân tạo vào ứng dụng di động. Nhà xuất bản Khoa học và Kỹ thuật.

[6]Nguyễn Văn D (2023). Phát triển hệ thống thanh toán trực tuyến. Nhà xuất bản Thông tin và Truyền thông.

[7][https://techcrunch.com](https://techcrunch.com/).

[8] https://www.youtube.com/@TITVvn