|  |  |
| --- | --- |
| ĐINH KHẮC DIÊN | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------**    **ĐỒ ÁN TỐT NGHIỆP**  NGÀNH CÔNG NGHỆ THÔNG TIN    **XÂY DỰNG ỨNG DỤNG HẸN HÒ TRÊN NỀN TẢNG ANDROID SỬ DỤNG MAUI VÀ ASP.NET CORE WEB API**  **CBHD: TS. Phạm Văn Hà**  **Sinh viên: Đinh Khắc Diên**  **Mã số sinh viên: 2021603238**  **Hà Nội – Năm 2025** |
| NGÀNH HỆ CÔNG NGHỆ THÔNG TIN |

**LỜI CẢM ƠN**

Trong thời gian hoàn thành đồ án tốt nghiệp ngành công nghệ thông tin, em đã nhận được nhiều sự hỗ trợ, giúp đỡ và khuyến khích từ rất nhiều người. Những người này đã đóng góp rất nhiều cho sự thành công của đồ án. Trong lời cảm ơn này, em xin gửi lời tri ân chân thành đến các tổ chức, cá nhân và gia đình của em.

Đầu tiên, em xin gửi lời cảm ơn đến các thầy, các cô của khoa công nghệ thông tin – trường Đại học Công Nghiệp Hà Nội, những người đã truyền đạt cho em kiến thức, kỹ năng và tinh thần làm việc chuyên nghiệp trong lĩnh vực công nghệ thông tin. Tiếp đó, em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn của em, TS. Phạm Văn Hà, thầy là người đã hỗ trợ, giúp đỡ em trong suốt quá trình thực hiện đồ án. Những lời khuyên, sự chỉ dẫn và hướng dẫn của thầy đã giúp em vượt qua được những khó khăn trong quá trình nghiên cứu và hoàn thành đồ án một cách hiệu quả.

Em cũng muốn gửi lời cảm ơn đến gia đình của mình, những người đã luôn ủng hộ và động viên em trong suốt thời gian nghiên cứu và hoàn thành đồ án. Họ là nguồn động viên, niềm tin và nguồn sức mạnh to lớn giúp em vượt qua những thử thách trong quá trình hoàn thành dự án.

Cuối cùng, mình xin gửi lời cảm ơn đến bạn bè, những người đã đồng hành và chia sẻ những khoảnh khắc vui buồn trong quá trình thực hiện đồ án. Cảm ơn vì đã luôn ở bên cạnh và giúp đỡ mình trong suốt thời gian này.

Em xin chân thành cảm ơn!

MỤC LỤC

[DANH MỤC HÌNH ẢNH 1](#_Toc198672887)

[MỞ ĐẦU 4](#_Toc198672888)

[1. Lý do chọn đề tài 4](#_Toc198672889)

[2. Mục tiêu của đề tài 4](#_Toc198672890)

[3. Nội dung nghiên cứu 5](#_Toc198672891)

[4. Phạm vi đề tài 5](#_Toc198672892)

[5. Bố cục của đề tài 5](#_Toc198672893)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 6](#_Toc198672894)

[1.1. Khảo sát 6](#_Toc198672895)

[1.2. Giới thiệu về công nghệ và ngôn ngữ sử dụng 8](#_Toc198672896)

[1.2.1. Giới thiệu về công nghệ MAUI 8](#_Toc198672897)

[1.2.2. Giới thiệu về công nghệ ASP.NET Core Web API 9](#_Toc198672898)

[1.2.3. Giới thiệu mô hình MVVM(Model – View – ViewModel) 11](#_Toc198672899)

[1.2.4. Giới thiệu về ngôn ngữ lập trình C# 12](#_Toc198672900)

[1.2.5. Hệ quản trị cơ sở dữ liệu quan hệ SQL Server 14](#_Toc198672901)

[1.3. Kết luận chương 1 16](#_Toc198672902)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 17](#_Toc198672903)

[2.1. Xác định các tác nhân của hệ thống 17](#_Toc198672904)

[2.2. Xác định các chức năng của hệ thống 17](#_Toc198672905)

[2.3. Biểu đồ Use case 19](#_Toc198672906)

[2.3.1. Biểu đồ Use case chính 19](#_Toc198672907)

[2.3.2. Biểu đồ Use case thứ cấp 20](#_Toc198672908)

[2.4. Mô tả chi tiết các Use case 21](#_Toc198672909)

[2.4.1. Use case “Xem danh sách ghép đôi” 21](#_Toc198672910)

[2.4.2. Use case “Tìm kiếm người dùng” 23](#_Toc198672911)

[2.4.3. Use case “Tương tác hồ sơ” 26](#_Toc198672912)

[2.4.4. Use case “Báo cáo người dùng” 29](#_Toc198672913)

[2.4.5. Use case “Quản lý tin nhắn” 33](#_Toc198672914)

[2.4.6. Use case “Quản lý thông báo” 37](#_Toc198672915)

[2.4.7. Use case “Quản lý thông tin cá nhân” 40](#_Toc198672916)

[2.4.8. Use case “Quản lý người dùng” 44](#_Toc198672917)

[2.4.9. Use case “Quản lý báo cáo người dùng” 48](#_Toc198672918)

[2.5. Cơ sở dữ liệu 52](#_Toc198672919)

[2.5.1. Sơ đồ dữ liệu quan hệ 52](#_Toc198672920)

[2.5.2. Chi tiết các bảng cơ sở dữ liệu 54](#_Toc198672921)

[2.6. Thiết kế giao diện chung cho User 59](#_Toc198672922)

[2.6.1. Giao diện Đăng nhập, Đăng ký 59](#_Toc198672923)

[2.6.2. Giao diện Trang chủ 60](#_Toc198672924)

[2.6.3. Giao diện Xem hồ sơ người dùng 61](#_Toc198672925)

[2.6.4. Giao diện Nhắn tin 61](#_Toc198672926)

[2.6.5. Giao diện Tìm kiếm người dùng 62](#_Toc198672927)

[2.7. Kết luận chương 2 62](#_Toc198672928)

[CHƯƠNG 3: KẾT QUẢ VÀ KIỂM THỬ 63](#_Toc198672929)

[3.1. Kết quả sản phẩm 63](#_Toc198672930)

[3.1.1. Trang màn hình giao diện của người người dùng 63](#_Toc198672931)

[3.1.2. Trang màn hình giao diện của người quản trị 69](#_Toc198672932)

[3.2. Kiểm thử ứng dụng 72](#_Toc198672933)

[3.2.1. Mục đích 72](#_Toc198672940)

[3.2.2. Định nghĩa và từ viết tắt 72](#_Toc198672941)

[3.2.3. Lịch trình kiểm thử 72](#_Toc198672942)

[3.2.4. Phạm vi kiểm thử 72](#_Toc198672943)

[3.2.5. Tài nguyên kiểm thử 73](#_Toc198672944)

[3.2.6. Chiến lược kiểm thử 73](#_Toc198672945)

[3.2.7. Điều kiện chấp nhận kiểm thử 74](#_Toc198672946)

[3.2.8. Thực hiện kiểm thử 75](#_Toc198672947)

[3.2.9. Kết quả kiểm thử 78](#_Toc198672948)

[3.3. Kết luận chương 3 79](#_Toc198672949)

[KẾT LUẬN 80](#_Toc198672950)

[1. Kết quả đạt được 80](#_Toc198672951)

[2. Hạn chế 80](#_Toc198672952)

[3. Hướng phát triển 80](#_Toc198672953)

[TÀI LIỆU THAM KHẢO 81](#_Toc198672954)

# DANH MỤC HÌNH ẢNH

[Hình 2‑1: Các tác nhân của hệ thống 17](#_Toc198673008)

[Hình 2‑2: Biểu đồ Use case chính 19](#_Toc198673009)

[Hình 2‑3: Biểu đồ Use case thứ cấp phía người dùng 20](#_Toc198673010)

[Hình 2‑4: Biểu đồ Use case thứ cấp phía người quản trị 20](#_Toc198673011)

[Hình 2‑5: Use case “Xem danh sách ghép đôi” 21](#_Toc198673012)

[Hình 2‑6: Biểu đồ trình tự UC “Xem danh sách ghép đôi” 22](#_Toc198673013)

[Hình 2‑7: Biểu đồ lớp chi tiết use case “Xem danh sách ghép đôi” 23](#_Toc198673014)

[Hình 2‑8: Use case “Tìm kiếm người dùng” 23](#_Toc198673015)

[Hình 2‑9: Biểu đồ trình tự UC “Tìm kiếm người dùng” 25](#_Toc198673016)

[Hình 2‑10: Biểu đồ lớp chi tiết use case “Tìm kiếm người dùng” 26](#_Toc198673017)

[Hình 2‑11: Use case “Tương tác hồ sơ” 26](#_Toc198673018)

[Hình 2‑12: Biểu đồ trình tự UC” Tương tác hồ sơ” 28](#_Toc198673019)

[Hình 2-13: Biểu đồ lớp chi tiết use case “Tương tác hồ sơ” 29](#_Toc198673020)

[Hình 2‑14: Use case “Báo cáo người dùng” 29](#_Toc198673021)

[Hình 2‑15: Biểu đồ trình tự UC” Báo cáo người dùng” 31](#_Toc198673022)

[Hình 2‑16: Biểu đồ lớp chi tiết use case “Báo cáo người dùng” 32](#_Toc198673023)

[Hình 2‑17: Use case “Quản lý tin nhắn” 33](#_Toc198673024)

[Hình 2‑18: Biểu đồ trình tự UC “Quản lý tin nhắn” 36](#_Toc198673025)

[Hình 2‑19: Biểu đồ lớp chi tiết use case “Quản lý tin nhắn” 37](#_Toc198673026)

[Hình 2‑20: Use case “Quản lý thông báo” 37](#_Toc198673027)

[Hình 2‑21: Biểu đồ trình tự UC “Quản lý thông báo” 39](#_Toc198673028)

[Hình 2‑22: Biểu đồ lớp chi tiết use case “Quản lý thông báo” 40](#_Toc198673029)

[Hình 2‑23: Use case “Quản lý thông tin cá nhân” 41](#_Toc198673030)

[Hình 2‑24: Biểu đồ trình tự UC “Quản lý thông tin cá nhân” 43](#_Toc198673031)

[Hình 2‑25: Biểu đồ lớp chi tiết use case “Quản lý thông tin cá nhân” 44](#_Toc198673032)

[Hình 2‑26: Use case “Quản lý người dùng” 44](#_Toc198673033)

[Hình 2‑27: Biểu đồ trình tự UC “Quản lý người dùng” 47](#_Toc198673034)

[Hình 2‑28: Biểu đồ lớp chi tiết use case “Quản lý người dùng” 48](#_Toc198673035)

[Hình 2-29: Use case “Quản lý báo cáo người dùng” 48](#_Toc198673036)

[Hình 2-30: Biểu đồ trình tự UC “Quản lý báo cáo người dùng” 50](#_Toc198673037)

[Hình 2‑31: Biểu đồ lớp chi tiết use case “Quản lý báo cáo người dùng” 51](#_Toc198673038)

[Hình 2‑32: Thiết kế giao diện Đăng ký 59](#_Toc198673039)

[Hình 2‑33: Thiết kế giao diện Đăng nhập 60](#_Toc198673040)

[Hình 2‑34: Thiết kế giao diện Trang chủ 60](#_Toc198673041)

[Hình 2‑35: Thiết kế giao diện Xem hồ sơ người dùng 61](#_Toc198673042)

[Hình 2-36: Thiết kế giao diện Nhắn tin 61](#_Toc198673043)

[Hình 2‑37: Thiết kế giao diện Tìm kiếm người dùng 62](#_Toc198673044)

[Hình 3‑1: Giao diện trang chủ ứng dụng (Swipe page) 63](#_Toc198673045)

[Hình 3‑2: Giao diện hiển thị hồ sơ cá nhân 64](#_Toc198673046)

[Hình 3‑3: Giao diện danh sách người dùng đã ghép đôi 64](#_Toc198673047)

[Hình 3-4: Giao diện nhắn tin 65](#_Toc198673048)

[Hình 3‑5: Giao diện thông báo 66](#_Toc198673049)

[Hình 3‑6: Giao diện quản lý thông tin cá nhân 66](#_Toc198673050)

[Hình 3‑7: Giao diện tìm kiếm người dùng 67](#_Toc198673051)

[Hình 3‑8: Giao diện nhắn tin chat bot 67](#_Toc198673052)

[Hình 3‑9: Giao diện đăng nhập 67](#_Toc198673053)

[Hình 3‑10: Giao diện đăng ký 68](#_Toc198673054)

[Hình 3‑11: Giao diện báo cáo người dùng 69](#_Toc198673055)

[Hình 3-12: Giao diện quản lý người dùng 70](#_Toc198673056)

[Hình 3‑13: Giao diện quản lý báo cáo 71](#_Toc198673057)

[Hình 3‑14: Giao diện đăng nhập 75](#_Toc198673058)

[Hình 3‑15: Giao diện danh sách ghép đôi 77](#_Toc198673059)

[**MỤC LỤC** 3](#_Toc198673060)

[Bảng 2‑1: Bảng mô tả các tác nhân của hệ thống 17](#_Toc198673061)

[Bảng 2‑2: Bảng sơ đồ dữ liệu quan hệ 52](#_Toc198673062)

[Bảng 2‑3: Bảng cơ sở dữ liệu User 54](#_Toc198673063)

[Bảng 2‑4: Bảng cơ sở dữ liệu Swipe 55](#_Toc198673064)

[Bảng 2‑5: Bảng cơ sở dữ liệu Match 55](#_Toc198673065)

[Bảng 2‑6: Bảng cơ sở dữ liệu Message 55](#_Toc198673066)

[Bảng 2‑7: Bảng cơ sở dữ liệu Photo 55](#_Toc198673067)

[Bảng 2‑8: Bảng cơ sở dữ liệu Notification 56](#_Toc198673068)

[Bảng 2‑9: Bảng cơ sở dữ liệu OTP 57](#_Toc198673069)

[Bảng 2‑10: Bảng cơ sở dữ liệu RefreshToken 57](#_Toc198673070)

[Bảng 2‑11: Bảng cơ sở dữ liệu Interest 57](#_Toc198673071)

[Bảng 2‑12: Bảng cơ sở dữ liệu User\_Interest 58](#_Toc198673072)

[Bảng 2‑13: Bảng cơ sở dữ liệu Report 58](#_Toc198673073)

[Bảng 2‑14: Bảng cơ sở dữ liệu Preference 58](#_Toc198673074)

[Bảng 2‑15: Bảng cơ sở dữ liệu Prefernce\_Interest 59](#_Toc198673075)

[Bảng 3-1: Bảng định nghĩa và viết tắt 72](#_Toc198673076)

[Bảng 3-2: Bảng lịch trình kiểm thử 72](#_Toc198673077)

[Bảng 3-3: Bảng tài nguyên phần cứng 73](#_Toc198673078)

[Bảng 3-4: Bảng tài nguyên phần mềm 73](#_Toc198673079)

[Bảng 3-5: Bảng công cụ kiểm thử 73](#_Toc198673080)

[Bảng 3-6: Loại kiểm thử 74](#_Toc198673081)

[Bảng 3-7: Bảng phân loại lỗi 74](#_Toc198673082)

[Bảng 3-8: Bảng test case Đăng nhập 76](#_Toc198673083)

[Bảng 3-9: Bảng test case Hiển thị danh sách ghép đôi 77](#_Toc198673084)

# MỞ ĐẦU

## **1. Lý do chọn đề tài**

Hiện nay, trong thời đại công nghệ phát triển cực kỳ nhanh chóng, chuyển đổi số đang là xu hướng tất yếu đã và đang thay đổi cách con người giao tiếp với nhau. Hơn nữa trong bối cảnh công nghệ phát triển mạnh mẽ và phổ biến của Internet, việc sử dụng các ứng dụng cải thiện giao tiếp, tìm kiếm các mối quan hệ ngày càng trở nên phổ biến và đang dần thay thế cho các cách thức truyền thống.

Có một bộ phận không nhỏ những người được cho là hướng nội như bản thân em hoặc đơn giản là không thích tìm kiếm các mối quan hệ theo cách truyền thống. Việc có một cách tiếp cận mới là nhu cầu không thể thiếu.

Trong xã hội ngày càng ưu tiên công việc hiện nay, một bộ phận người mong muốn việc tạo dựng các mối quan hệ mới, tìm kiếm bạn đời trở nên dễ dàng hơn, không tốn quá nhiều thời gian và họ có thể tận dụng quãng thời gian giải lao ngắn ngủi qua điện thoại.

Lý do chọn xây dựng ứng dụng hẹn hò Loner là bởi vì việc muốn mang lại trải nghiệm khác lạ khi muốn mở rộng quan hệ mới, cung cấp cách tiếp cận dễ dàng, đơn giản hơn với một bộ phận người vừa mong muốn tìm bạn đời mà không muốn mất quá nhiều thời gian.

Đồng thời, việc áp dụng công nghệ vào việc tìm kiếm bạn đời giúp mọi người có thể tiếp xúc được nhiều người có cùng sở thích, tính cách hơn hoặc đơn giản là chủ động tìm kiếm theo ý thích của bản thân.

Việc lựa chọn nền tảng Android để phát triển cũng phụ thuộc vào tình hình hiện tại khi mà Android là hệ điều hành di động phổ biến nhất tại Việt Nam cũng như trên toàn thế giới.

Vì vậy, việc xây dựng ứng dụng hẹn hò Loner là điều tất yếu với bối cảnh xã hội hiện đại ngày nay.

## **2. Mục tiêu của đề tài**

Đề tài: “Xây dựng ứng dụng hẹn hò trên nền tảng Android sử dụng MAUI và ASP.Net Core Web Api” đáp ứng được các mục tiêu:

Giải quyết những nhu cầu thực tiễn của một bộ phận người “hướng nội”, ngại giao tiếp theo cách truyền thống. Giúp việc mở rộng các mối quan hệ, tìm kiếm bạn đời trở nên dễ dàng hơn, tiết kiệm thời gian hơn. Cung cấp nền tảng giúp người dùng dễ dàng tìm kiếm, kết nối và tương tác với những người phù hợp.

## **3. Nội dung nghiên cứu**

- Tìm hiểu về nhu cầu, mong muốn và thói quen của người dùng độc thân trong việc tìm kiếm các mối quan hệ trực tuyến.

- Nghiên cứu các ứng dụng hẹn hò đã có trên thị trường để xác định các tính năng phổ biến, điểm mạnh và hạn chế còn tồn đọng.

- Xây dựng giao diện thân thiện, đơn giản và dễ sử dụng cho người dùng.

- Phát triển các tính năng cốt lõi của ứng dụng liên quan tới hồ sơ người dùng, tìm kiếm, Gợi ý hồ sơ phù hợp.

- Nghiên cứu, tìm hiểu về phát triển ứng dụng di động trên nền tảng Android.

## **4. Phạm vi đề tài**

Ứng dụng di động được thiết kế để đáp ứng nhu cầu tìm kiếm các mối quan hệ trực tuyến của người dùng. Cung cấp giao diện thân thiện với người dùng và tương thích với các thiết bị Android phổ biến trên thị trường hiện nay.

## **5. Bố cục của đề tài**

Trong báo cáo đồ án này, cấu trúc của đề tài được chia thành 3 chương chính như sau:

* **Chương 1**: Tổng quan về đề tài
* **Chương 2:** Phân tích thiết kế hệ thống
* **Chương 3:** Kết quả và kiểm thử

# TỔNG QUAN VỀ ĐỀ TÀI

* 1. Khảo sát

Trong thời đại công nghệ hiện nay, ứng dụng hẹn hò đóng vai trò không thể thay thế trong việc kết nối mọi người và thay đổi phương thức hẹn hò truyền thống. Vì vậy, thị trường ứng dụng hẹn hò phát triển rất mạnh mẽ tại Việt Nam và trên toàn thế giới. Đây là nguồn tham khảo vô cùng chất lượng góp phần xây dựng nên ứng dụng hẹn hò Loner. Em cũng tham khảo thêm về quan điểm cá nhân của các bạn được coi là “hướng nội” để có cái nhìn khách quan từ đó phát triển ứng dụng đáp ứng được các nhu cầu thực tế đặt ra.

Hiện tại, với một số ứng dụng hẹn hò phổ biến trên thị trường như Tinder, Bumble có một số ưu, nhược điểm như sau:

* Ưu điểm:
* Thao tác tiện lợi và nhanh chóng: Quy trình đăng ký và tìm kiếm đối tượng thường đơn giản và nhanh chóng.
* Đa dạng tính năng hỗ trợ: Cung cấp nhiều tính năng như vuốt để thích/ bỏ qua, nhắn tin, xem thông tin cá nhân, tìm kiếm theo bộ lọc, ...
* Thuật toán gợi ý mạnh mẽ: Sử dụng thuật toán để gợi ý các đối tượng dựa trên vị trí, sở thích, và các thông tin khác.
* Khả năng tiếp cận lớn: Dễ dàng đề xuất, kết nối với một lượng lớn người dùng tiềm năng ở nhiều địa điểm khác nhau.
* Phù hợp với nhu cầu đa dạng của người dùng: Một số người dùng tìm kiếm các mối quan hệ nghiêm túc, trong khi những người khác chỉ muốn kết bạn hay hẹn hò ngắn hạn.
* Nhược điểm:
* Khó khăn trong việc tìm kiếm những người có cùng sở thích hoặc tính cách tương đồng: Các bộ lọc tìm kiếm có thể chưa đủ chi tiết để đáp ứng các nhu cầu tìm kiếm của người dùng có sở thích đặc biệt.
* Sự quá tập trung vào ngoại hình: Một số ứng dụng ưu tiên hiển thị hình ảnh và đánh giá dựa trên vẻ bề ngoài, điều này có thể không phù hợp với những người coi trọng tính cách và sở thích hơn.
* Áp lực phải gây ấn tượng nhanh chóng: Người dùng thường cảm thấy áp lực phải tạo ra một hồ sơ hấp dẫn mới đủ sức thu hút.
* Thiếu không gian cho việc kết nối chậm rãi và sâu sắc: Một số ứng dụng thường khuyến khích việc gặp gỡ trực tiếp nhanh chóng, có thể không phù hợp với những người muốn xây dựng quan hệ từ từ.
* Thiếu các tính năng hỗ trợ giao tiếp cho người hướng nội: Có thể thiếu các công cụ hoặc gợi ý để giúp người ngại giao tiếp bắt đầu và duy trì cuộc trò chuyện một cách thoải mái.
* Nguy cơ gặp phải những người không chân thành hoặc mục đích xấu: Hành vi lừa đảo hẹn hò gây ra những trải nghiệm tiêu cực.

Xác định xây dựng ứng dụng hẹn hò Loner với những yêu cầu sau:

- Xây dựng chức năng vuốt phải/ trái để thích/ bỏ qua.

- Xây dựng chức năng xem thông tin chi tiết người dùng.

- Xây dựng chức năng tìm kiếm người dùng theo phạm vi bán kính, theo thành phố.

- Xây dựng chức năng nhắn tin gửi tin nhắn văn bản và hình ảnh.

- Xây dựng chức năng quản lý thông tin cá nhân.

- Xây dựng chức năng thông báo gồm tin nhắn, 2 người “match” nhau.

- Xây dựng chức năng chat với AI gợi ý trò chuyện.

* 1. Giới thiệu về công nghệ và ngôn ngữ sử dụng
     1. Giới thiệu về công nghệ MAUI
        1. **Khái niệm.**

MAUI là một framework phát triển ứng dụng đa nền tảng được Microsoft giới thiệu là sự kế thừa từ Xamarin.Forms. MAUI cho phép lập trình viên viết mã một lần bằng C# và XAML để chạy trên Android, IOS, Windows và macOS. MAUI ra đời với mục tiêu đơn giản hóa việc phát triển ứng dụng trên nhiều hệ điều hành bằng một codebase thống nhất.

MAUI là một công nghệ còn khá mới nhưng hiệu năng được tối ưu, giao diện native trên mỗi nền tảng cùng với sự hỗ trợ của hệ sinh thái .NET khiến MAUI ngày càng trở nên phổ biến.

* + - 1. **Ưu và nhược điểm của công nghệ MAUI**
* Ưu điểm:
* Dễ dàng xây dựng các ứng dụng đa nền tảng với 1 mã nguồn duy nhất giúp tiết kiệm chi phí và thời gian phát triển.
* Tăng khả năng mở rộng và bảo trì thông qua mô hình MVVM.
* Dùng chung với các công nghệ hiện đại với .NET 8/9, LINQ, Dependency Injection, ...
* Sử dụng Handler thay vì renderer giúp giảm độ trễ và tăng hiệu suất UI.
* Nhược điểm:
* Yêu cầu kiến thức kỹ thuật cao và thời gian học tập.  
    
    
    
    
  + - 1. **Cấu trúc và thành phần của công nghệ MAUI**
* App.xaml / App.xaml.cs: Định nghĩa resource và thiết lập trang chính.
* Platforms: Chứa các mã cho từng nền tảng riêng biệt nếu cần.
* Resources: Nơi chứa các ảnh, font, styles được sử dụng chung.
* AppShell: Các cấu trúc điều hướng ứng dụng hiện đại.
* ContentPage: Trang giao diện cơ bản trong ứng dụng.
* ContentView: Các thành phần UI có thể tái sử dụng ở nhiều nơi.
* Controls (Điều khiển): Các thành phần UI cơ bản như: nút, nhãn, ô nhập, …
* Handler: Cơ chế mà MAUI kết nối các điều khiển với các điều khiển gốc cho từng nền tảng.
* MauiProgram.cs: Cấu hình ứng dụng, đăng ký dịch vụ, font, các thư viện được sử dụng.
  + 1. Giới thiệu về công nghệ ASP.NET Core Web API
       1. **Khái niệm**

ASP.NET Core Web API là một framework mã nguồn mở của Microsoft được sử dụng để xây dựng các dịch vụ web (API) dựa trên HTTP. Cho phép client như mobile gửi yêu cầu và nhận dữ liệu dưới dạng Json hoặc Xml thông qua các phương thức của chuẩn RESTful API.

ASP.NET Core Web API ra đời với mục cho phép backend hoạt động như một dịch vụ độc lập, nhẹ, linh hoat, hiệu năng cao, có thể phục vụ nhiều client như: web, mobile, desktop, ...

* + - 1. **Ưu và nhược điểm của ASP.NET Core Web API**
* Ưu điểm:
* Hiệu năng cao, sử dụng Kestrel – một web server cực nhanh, tối ưu hiệu suất.
* Bảo mật tốt: Hỗ trợ xác thực JWT, CORS, ASP.NET Identity, v.v.
* Dễ dàng kiểm tra các API với cấu hình Swagger UI.
* Kiến trúc hiện đại: Hỗ trợ tốt các mô hình kiến trúc như MVC, Clean Architecture, Microservices.
* Hỗ trợ đa nền tảng: Chạy được trên Windows, Linux, macOS, phù hợp với môi trường cloud hiện đại.
* Hỗ trợ mã nguồn mở: Mọi tài liệu đều được phát triển công khai với cộng đồng rất lớn.
* Nhược điểm:
* Dung lượng ứng dụng lớn.
* Tốn thời gian khởi động, đặc biệt với ứng dụng phức tạp.
* Cấu hình ban đầu phức tạp hơn.
  + - 1. **Cấu trúc và các thành phần của ASP.NET Core Web API**

Cấu trúc và các thành phần chính của Spring Boot bao gồm:

* Domain layer: chứa các ánh xạ database – entity, các Interface Repository.
* Application layer: Chứa các use cases – logic của các tác vụ cụ thể, định nghĩa DTOs – để truyền dữ liệu, service interface – để sử dụng trong các use cases.
* Infrastructure layer: Nơi thể hiện các interface định nghĩa trong Domain hoặc Application, là nơi kết nối tới database, ghi file. Bao gồm các Repositories – triển khai các Interface Repository, DataContext.
* Presentation Layer: Nơi cung cấp các entry point để người dùng truy cập tới. Bao gồm các Controller – xử lý HTTP và gọi tới phần xử lý logic ở Application layer, Programs.cs – cấu hình DI, middleware, xác thực, phân quyền, cấu hình Swagger, thiết lập cấu hình cho dự án, v.v.  
    
    
  + 1. Giới thiệu mô hình MVVM(Model – View – ViewModel)
       1. **Khái niệm**
* Mô hình MVVM (Model-View-ViewModel) là một mô hình phát triển phần mềm được sử dụng phổ biến trong lĩnh vực phát triển ứng dụng web, đặc biệt là trong các ứng dụng dựa trên các nền tảng như MAUI framework.
* Mô hình MVVM tách biệt các thành phần trong ứng dụng mobile, tạo ra một sự phân cấp rõ ràng giữa chúng, giúp giảm thiểu sự phức tạp của mã, cải thiện khả năng bảo trì và mở rộng ứng dụng.
* MVVM hỗ trợ ứng dụng động, cho phép người dùng tương tác với dữ liệu dễ dàng nhờ cơ chế liên kết dữ liệu.
* Cấu trúc mô hình MVVM.
* Model: đại diện cho dữ liệu và các thành phần liên quan đến dữ liệu, bao gồm các lớp, cấu trúc dữ liệu, cơ sở dữ liệu, v.v.
* View: đại diện cho giao diện người dùng (UI), hiển thị dữ liệu người dùng và quản lý sự kiện người dùng, liên kết dữ liệu đến ViewModel.
* ViewModel: xử lý logic của ứng dụng, giữ liên kết dữ liệu giữa Model và View, cập nhật dữ liệu vào View.
  + - 1. **Ưu và nhược điểm của MVVM**
* Ưu điểm:
* Tách biệt hoàn toàn giữa giao diện người dùng và logic ứng dụng, giúp tập trung vào xử lý logic hiệu quả hơn.
* Dễ để bảo trì và kiểm thử nhờ phân tách rõ ràng các thành phần.
* Giải quyết vấn đề xung đột trong quản lý dữ liệu qua data binding và ViewModel.
* Nhược điểm:
* Khó học khi mới bắt đầu với front-end development.
* Tạo ra quá nhiều lớp và feature, gây phức tạp trong quản lý các tầng giao diện, logic của ứng dụng và database.
  + - 1. **Mô hình MVVM đối với công nghệ MAUI framework**
* Trong MAUI Framework, MVVM được sử dụng để phân chia ứng dụng thành các phần riêng biệt và đơn giản hóa việc quản lý mã nguồn.
* MAUI cung cấp một số công cụ để giúp triển khai MVVM, bao gồm:
* Data Binding: Cho phép tự động cập nhật dữ liệu giữa ViewModel và View.
* Dependency Injection: Cho phép lấy các thành phần khác nhau của ứng dụng và sử dụng chúng để tạo ra ViewModel.
* Event Binding: Cho phép kết nối các sự kiện người dùng với các phương thức trong ViewModel.
  + 1. Giới thiệu về ngôn ngữ lập trình C#
       1. **Khái niệm**

C# là một ngôn ngữ lập trình đa nền tảng, hướng đối tượng, hiện đại được phát triển bởi Microsoft như một phần của nền tảng .NET. C# được ra giới thiệu lần đầu vào năm 2001 được phát triển bởi Anders Hejlsberg. C# chịu ảnh hưởng của nhiều ngôn ngữ lập trình khác như C++, Java kết hợp những điểm mạnh của chúng trong một cú pháp rõ ràng và nhất quán.

C# hoàn toàn là ngôn ngữ hướng đối tượng, hỗ trợ các khái niệm như lớp (class), đối tượng (object), kế thừa (inheritance), đa hình (polymorphism), đóng gói (encapsulation) và trừu tượng (abstraction). Điều này giúp tổ chức mã nguồn một cách logic và dễ quản lý.

C# được thiết kế để hoạt động liền mạch với nền tảng .NET, cho phép truy cập vào một thư viện lớp (class library) phong phú cung cấp các chức năng cho nhiều tác vụ khác nhau (ví dụ: I/O, mạng, xử lý dữ liệu, bảo mật).

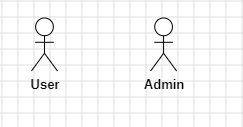
* + - 1. **Ưu và nhược điểm của ngôn ngữ lập trình C#**
* Ưu điểm:
* C# là một ngôn ngữ đa năng, có thể được sử dụng để phát triển nhiều loại ứng dụng khác nhau.
* C# được thiết kế dựa trên các nguyên tắc hướng đối tượng một cách chặt chẽ.
* Cơ chế thu gom rác tự động giúp các nhà phát triển không cần lo lắng về việc quản lý bộ nhớ thủ công.
* C# cung cấp cú pháp đơn giản để viết mã bất đồng bộ, giúp ứng dụng phản hồi tốt hơn và tận dụng hiệu quả tài nguyên hệ thống.
* Nhược điểm:
* So với một số ngôn ngữ như Go hoặc Rust, các ứng dụng .NET có thể có bấm thước runtime lớn hơn do bao gồm các thư viện .NET.
* Yêu cầu kỹ năng lập trình cao hơn.
* Mặc dù hiệu suất của C# thường rất tốt, trong một số trường hợp đòi hỏi hiệu suất cực cao C, C++ được ưu tiên hơn.
  + - 1. **Tầm quan trọng của ngôn ngữ lập trình C# đối với công nghệ ASP.NET Core Web API**
* C# là ngôn ngữ chủ đạo xây dựng toàn bộ framework ASP.NET Core, từ các thư viện cơ bản đến các thành phần phức tạp như MVC Core, Identity, Entity Framework Core, đều được viết bằng C#. Do đó, để phát triển ứng dụng ASP.NET Core Web API, bắt buộc phải sử dụng C#
* Ưu điểm khi sử dụng với ASP.NET Core Web API:
* **.**NET Runtime được tối ưu hóa: ASP.NET Core chạy trên .NET runtime hiệu suất cao.
* Hỗ trợ phát triển các ứng dụng đa nền tảng, ứng dụng có thể chạy trên bất cứ hệ điều hành nào và triển khai lên các môi trường khác nhau.
* Kiến trúc hiện đại và linh hoạt.
* Hỗ trợ các tính năng OOP như kế thừa, đa hành, tính trường tượng và giao diện giúp phát triển Angular hiệu quả và dễ dàng.
* Phát triển API mạnh mẽ giúp dễ dàng tích hợp với các ứng dụng và dịch vụ khác.
* Việc sử dụng C# trong ASP.NET Core Web API tăng tính ổn định, dễ bảo trì và nâng cao chất lượng ứng dụng.
  + 1. Hệ quản trị cơ sở dữ liệu quan hệ SQL Server
       1. **Tổng quan về SQL Server**
* SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft. Nó là một phần mềm mạnh mẽ được thiết kế để lưu trữ, quản lý và truy xuất dữ liệu một cách hiệu quả và an toàn.
* SQL Server hoạt động theo mô hình client-server. Server là nơi chứa cơ sở dữ liệu và các dịch vụ quản lý. Client là các ứng dụng gửi yêu cầu đến server để tương tác với dữ liệu.
* SQL Server được sử dụng rộng rãi trong các ứng dụng web, mobile, cửa hàng trực tuyến, hệ thống quản lý nội dung, và các ứng dụng doanh nghiệp khác. Nó có thể được triển khai trên các máy chủ riêng hoặc trên đám mây.
* SQL Server được đánh giá cao về bảo mật, độ ổn định và hiệu suất, là lựa chọn phổ biến cho nhà phát triển và doanh nghiệp.
  + - 1. **Các thành phần cơ bản của SQL Server**
* Server: Quản lý kết nối, xử lý truy vấn và lưu trữ dữ liệu.
* Databases: Chứa các bảng, chỉ số và ràng buộc, có thể tạo nhiều cơ sở dữ liệu trên một server
* Tables: Thành phần cơ bản nhất, chứa các hàng (rows) và cột (columns).
* Columns: Thuộc tính của bảng, có tên và kiểu dữ liệu.
* Rows: Bản ghi của bảng, gồm các giá trị cho các cột tương ứng.
* Indexes: Tăng tốc độ truy vấn dữ liệu.
* Views: Tạo “cửa sổ” ảo của dữ liệu từ các bảng khác nhau.
* Triggers: Chương trình tự động bấm hoạt khi có sự kiện xảy ra.
* Các công cụ quản lý cơ sở dữ liệu: SQL Server Management Studio (SSMS) là công cụ chính giúp quản trị cơ sở dữ liệu SQL Server.
* Các tính năng hỗ trợ: bảo mật, sao lưu và phục hồi dữ liệu, xử lý truy vấn tối ưu, đồng bộ hóa dữ liệu, v.v.
  + - 1. **Ưu và nhược điểm của SQL Server**
* Ưu điểm:
* Miễn phí và mã nguồn mở, có cộng đồng lớn hỗ trợ.
* Hiệu suất tốt, xử lý hàng ngàn truy vấn mỗi giây an toàn và bảo mật
* Linh hoạt và dễ sử dụng, hỗ trợ nhiều ngôn ngữ lập trình.
* Nhược điểm:
* Không hỗ trợ tính năng NoSQL.
* Giới hạn khi xử lý dữ liệu lớn.
* Chậm áp dụng các tính năng mới.
  + - 1. **Sự tương tác giữa SQL Server và ASP.NET Core Web API**
* Entity Framework Core (EF Core) là một ORM (Object-Relational Mapper) mã nguồn mở, là một phần của hệ sinh thái .NET Core, giúp tạo ra một lớp trung gian giữa tầng ứng dụng (bao gồm cả Controller và Service Layer trong ASP.NET Core Web API) và cơ sở dữ liệu SQL Server. Nó giúp chúng ta thao tác với cơ sở dữ liệu dễ dàng hơn, tự động hóa nhiều tác vụ liên quan đến ánh xạ đối tượng và giảm thiểu code lặp lại.
* EF Core cho phép các nhà phát triển .NET làm việc với cơ sở dữ liệu SQL Server bằng cách sử dụng các đối tượng C# (entities) thay vì viết trực tiếp các câu lệnh SQL. Nó quản lý việc ánh xạ giữa các lớp C# và các bảng trong cơ sở dữ liệu, cũng như việc tạo và thực thi các truy vấn.
* Để sử dụng EF Core với SQL Server trong dự án ASP.NET Core Web API, cần thêm các NuGet package cần thiết vào dự án. Quan trọng nhất là package Microsoft.EntityFrameworkCore.SqlServer để cung cấp provider (trình điều khiển) cho SQL Server. Bạn cũng có thể cần các package khác như Microsoft.EntityFrameworkCore.Tools cho các tác vụ migration (quản lý lược đồ cơ sở dữ liệu).
  1. Kết luận chương 1

Trong chương 1, em đã trình bày về việc khảo sát và các nghiên cứu về công nghệ sử dụng trong dự án bao gồm: khái niệm, ưu điểm, nhược điểm, các thư viện, cấu trúc và thành phần, tầm quan trọng của chúng.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

* 1. Xác định các tác nhân của hệ thống

Ứng dụng hẹn hò Loner có 1 tác nhân chính là người dùng.

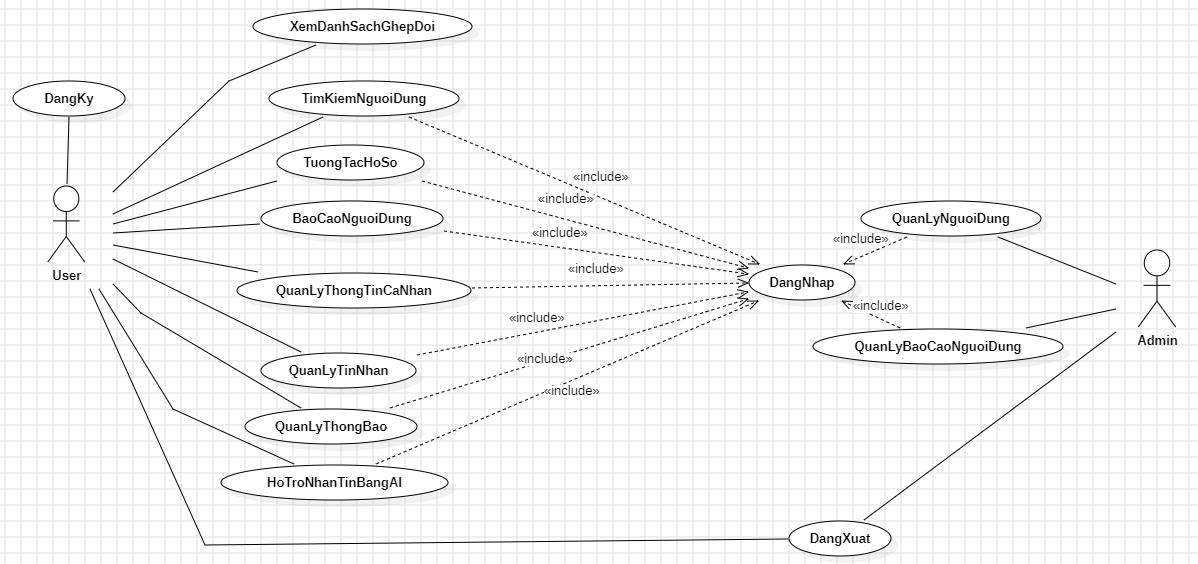


Hình 2‑1: Các tác nhân của hệ thống

Bảng 2‑1: Bảng mô tả các tác nhân của hệ thống

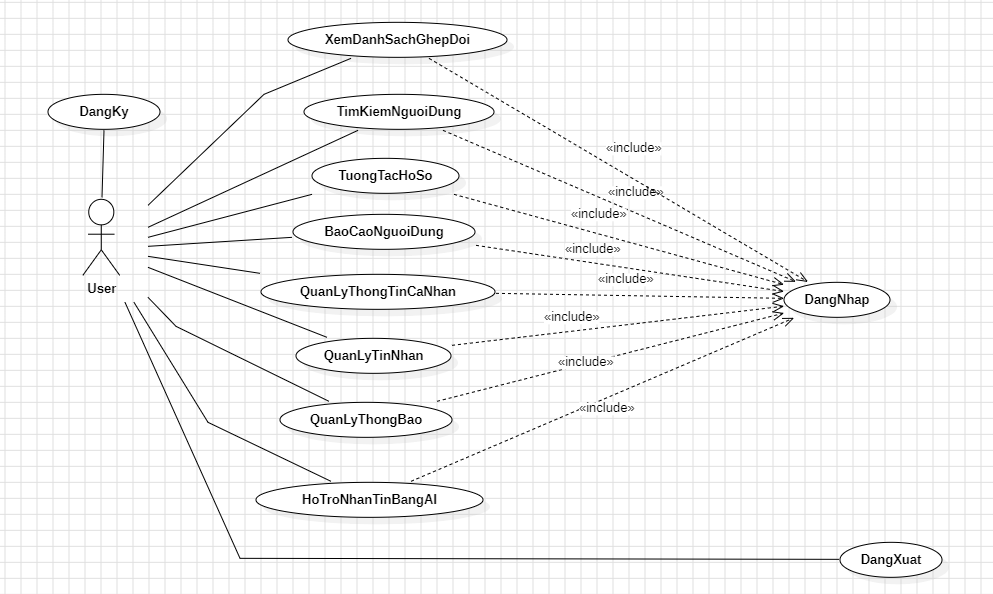
|  |  |  |
| --- | --- | --- |
| STT | Tác nhân | Mô tả |
| 1 | User (Người dùng) | Là người tác động vào hệ thống, sử dụng các chức năng của hệ thống như quẹt trái, phải, nhắn tin, … |
| 2 | Admin (Quản trị viên) | Là người có chức năng quản lý hệ thống, quản lý người dùng, quản lý báo cáo thống kê, … |

* 1. Xác định các chức năng của hệ thống
* Yêu cầu chức năng:
  + Chức năng xem danh sách đã ghép đôi.
  + Chức năng tìm kiếm người dùng.
  + Chức năng đăng nhập.
  + Chức năng đăng ký.
  + Chức năng tương tác hồ sơ.
  + Chức năng báo cáo người dùng.
  + Chức năng quản lý thông tin cá nhân.
  + Chức năng quản lý tin nhắn.
  + Chức năng quản lý thông báo.
  + Chức năng hỗ trợ nhắn tin bằng AI.
  + Chức năng quản lý người dùng.
  + Chức năng quản lý báo cáo người dùng.
* Yêu cầu phi chức năng:
  + Hệ thống phải được bảo mật khỏi sự truy cập trái phép
  + Hệ thống phải có khả năng xử lý số lượng người dùng truy cập nhiều đột xuất mà không ảnh hưởng tới hiệu suất.
  + Hệ thống có thể tăng hoặc giảm quy mô khi cần thiết.
  + Hệ thống có thể bảo trì và cập nhật dễ dàng.
  + Hệ thống phải dễ sử dụng và dễ hiểu.
  + Hệ thống phải tin cậy và đáp ứng được yêu cầu người dùng.
  1. Biểu đồ Use case
     1. Biểu đồ Use case chính

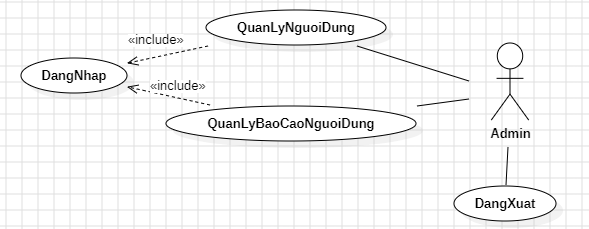


Hình 2‑2: Biểu đồ Use case chính

1. **Xem danh sách ghép đôi:** Cho phép người dùng xem những người đã được ghép đôi.
2. **Tìm kiếm người dùng:** Cho phép người dùng tìm kiếm đối tượng khác phù hợp với mình.
3. **Tương tác hồ sơ:** Cho phép người dùng quẹt để tương tác với người dùng khác.
4. **Báo cáo người dùng**: Cho phép người dùng báo cáo về hành vi không phù hợp của người dùng khác.
5. **Quản lý thông tin cá nhân:** Cho phép người dùng chỉnh sửa thông tin cá nhân.
6. **Quản lý tin nhắn:** Cho phép người dùng xem, gửi tin nhắn.
7. **Quản lý thông báo:** Cho phép người dùng xem, đánh dấu là đã đọc, xóa thông báo.
8. **Hỗ trợ nhắn tin bằng AI:** Cho phép người dùng hỏi và nhận gợi ý nhắn từ AI.
9. **Đăng nhập:** Cho phép người dùng, quản trị viên đăng nhập tài khoản vào hệ thống.
10. **Đăng ký:** Cho phép người dùng đăng ký tài khoản.
11. **Đăng xuất:** Cho phép người dùng, quản trị viên đăng xuất khỏi hệ thống.
12. **Quản lý người dùng:** Cho phép người quản trị quản quản lý tài khoản người dùng.
13. **Quản lý báo cáo người dùng:** Cho phép người quản trị quản lý các lượt báo cáo từ người dùng.
    * 1. Biểu đồ Use case thứ cấp

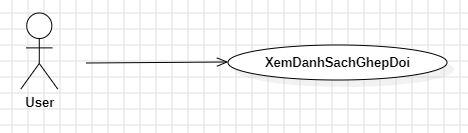


Hình 2‑3: Biểu đồ Use case thứ cấp phía người dùng



Hình 2‑4: Biểu đồ Use case thứ cấp phía người quản trị

* 1. Mô tả chi tiết các Use case
     1. Use case “Xem danh sách ghép đôi”
        1. Sơ đồ UC



Hình 2‑5: Use case “Xem danh sách ghép đôi”

* + - 1. Đặc tả UC

**Mô tả vắn tắt:** Use case này cho phép người dùng xem danh sách những người dùng đã được ghép đôi.

**Luồng sự kiện:**

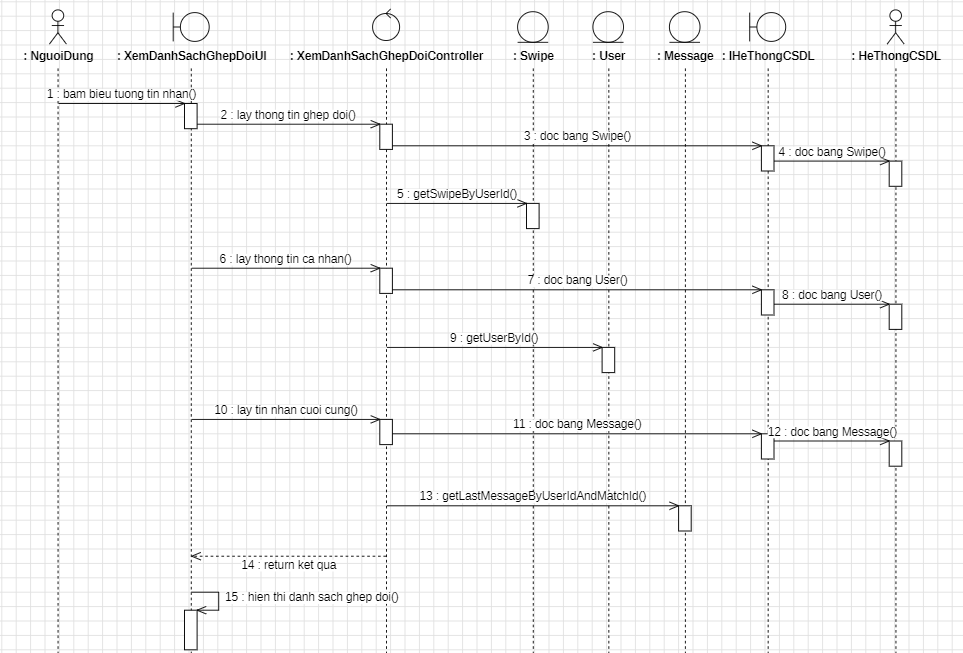
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng bấm vào biểu tượng tin nhắn trên thanh điều hướng của ứng dụng. Hệ thống sẽ lấy Id người dùng đã được ghép đôi từ bảng Swipe, thông tin người dùng bao gồm ảnh đại diện, tên từ bảng User và tin nhắn cuối cùng từ bảng Message.

* **Luồng rẽ nhánh:**

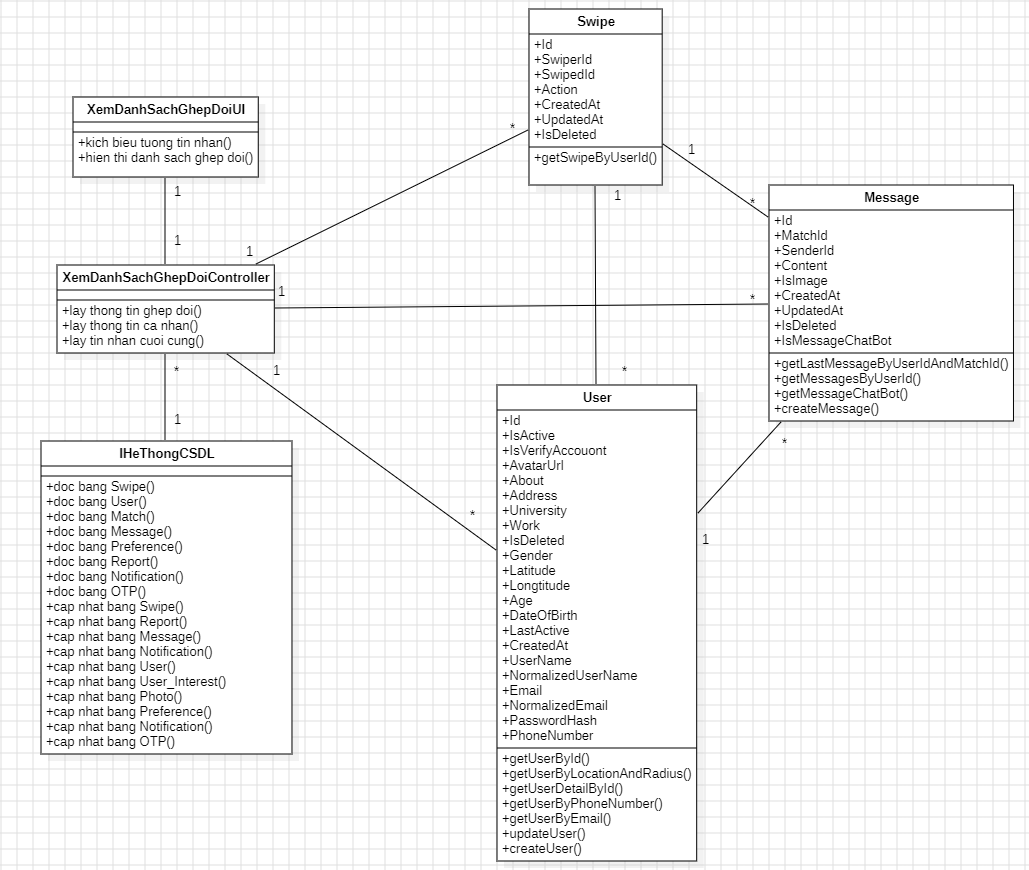
1. Tại bước 1, nếu chưa có người ghép đôi nào, hệ thống sẽ hiển thị “Bạn chưa có lượt ghép đôi nào” lên màn hình, use case kết thúc.
2. Tại bất kỳ bước nào trong luồng cơ bản, nếu không có kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo “Có lỗi xảy ra” và use case kết thúc.

* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** Người dùng cần đăng nhập để thực hiện use case.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Nhắn tin.
  + - 1. Biểu đồ trình tự



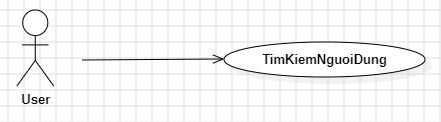
Hình 2‑6: Biểu đồ trình tự UC “Xem danh sách ghép đôi”

* + - 1. Biểu đồ lớp chi tiết



Hình 2‑7: Biểu đồ lớp chi tiết use case “Xem danh sách ghép đôi”

* + 1. Use case “***Tìm kiếm người dùng***”
       1. Biểu đồ UC



Hình 2‑8: Use case “*Tìm kiếm người dùng*”

* + - 1. Đặc tả UC

**Mô tả vắn tắt:** Cho phép người dùng tìm kiếm đối tượng khác phù hợp với mình.

**Luồng sự kiện**

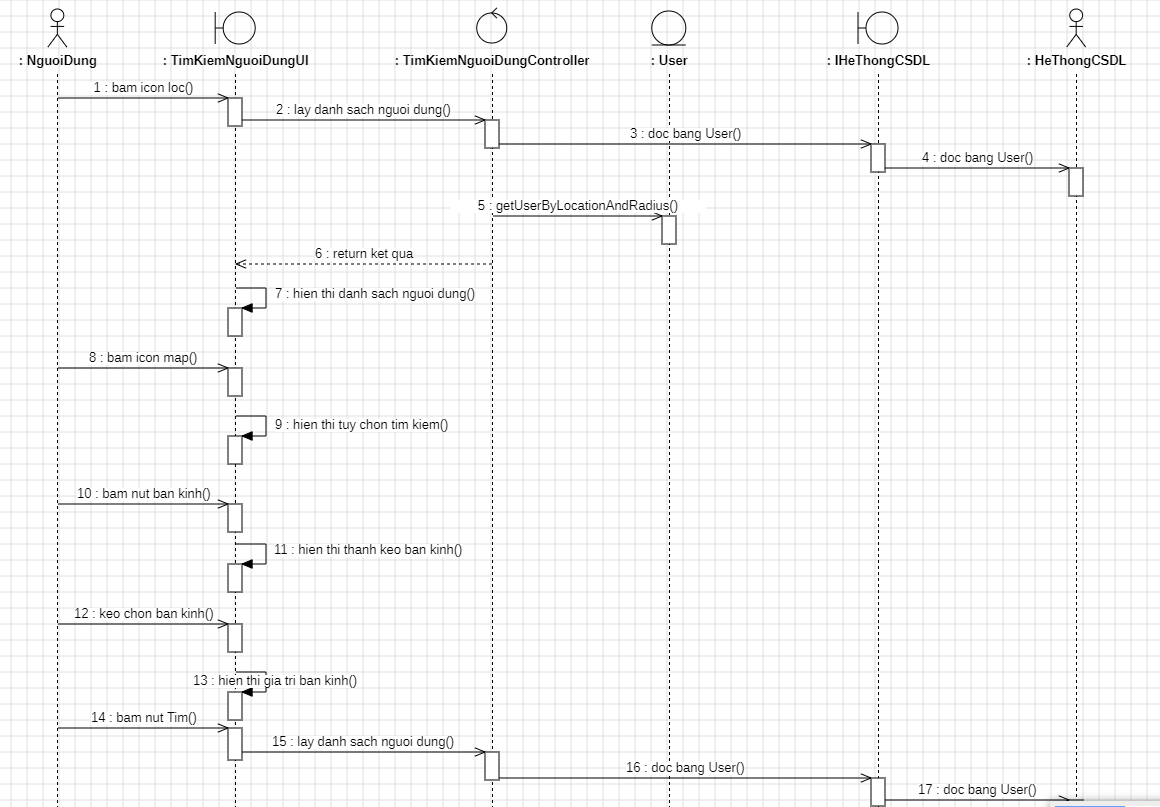
* **Luồng cơ bản:**

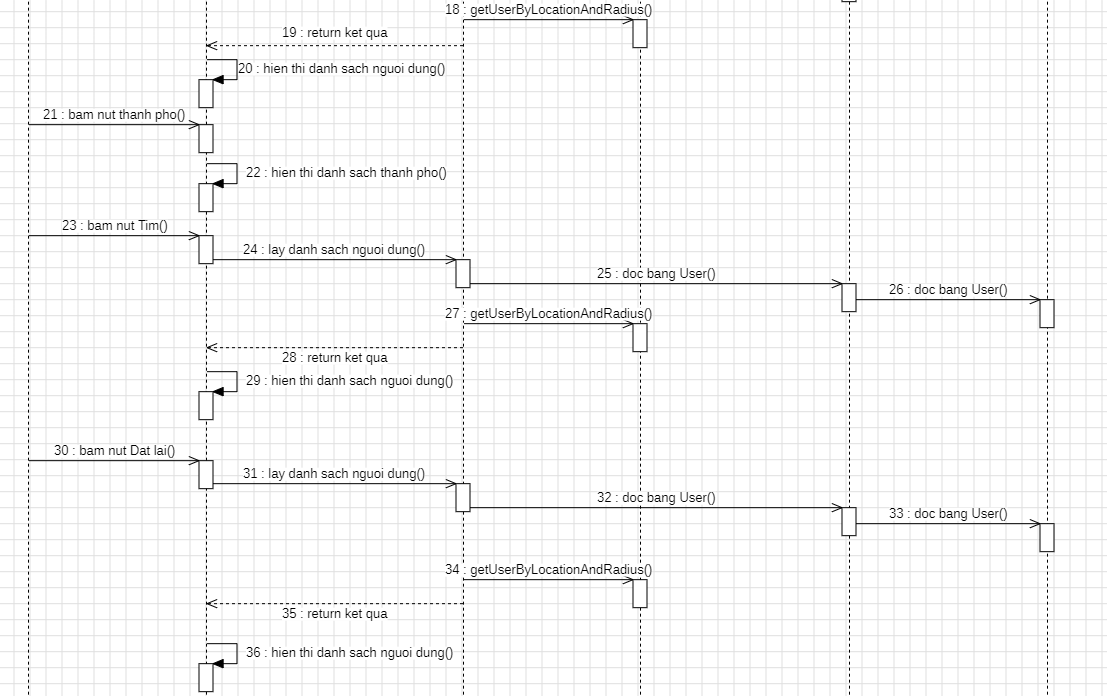
1. Use case này bắt đầu khi dùng bấm vào icon lọc trên thanh điều hướng. Hệ thống sẽ hiển thị danh sách người dùng nằm trong khoảng tìm kiếm mặc định phù hợp với địa chỉ hiện tại.
2. Người dùng bấm vào icon map trên thanh điều hướng. Hệ thống sẽ hiển thị các nút tìm kiếm tùy chọn.
3. Người dùng bấm nút bán kính, hệ thống hiển thị thanh kéo sau đó người dùng kéo để chọn bán kính mong muốn và bấm nút “Tìm”. Hệ thống sẽ lấy thông tin người gồm: ảnh đại diện, địa chỉ, tên và khoảng cách trong bảng User có khoảng cách nằm trong khoảng tìm kiếm và hiển thị lên màn hình.
4. Người dùng bấm nút thành phố, hệ thống hiển thị danh sách thành phố sau đó người dùng kéo để chọn bán kính mong muốn và bấm nút “Tìm”. Hệ thống sẽ lấy thông tin người gồm: ảnh đại diện, địa chỉ, tên và khoảng cách trong bảng User có khoảng cách nằm trong khoảng tìm kiếm của thành phố và hiển thị lên màn hình.
5. Người dùng bấm nút đặt lại. Hệ thống sẽ lấy thông tin người gồm: ảnh đại diện, địa chỉ, tên và khoảng cách trong bảng User có khoảng cách nằm trong khoảng tìm kiếm mặc định phù hớp với địa chỉ hiện tại và hiển thị lên màn hình.

* **Luồng rẽ nhánh**

1. Tại bất kỳ thời điểm nào trong luồng cơ bản, hệ thống không kết nối được với cơ sở dữ liệu thì sẽ hiển thị thống báo “Có lỗi xảy ra” và use case kết thúc.

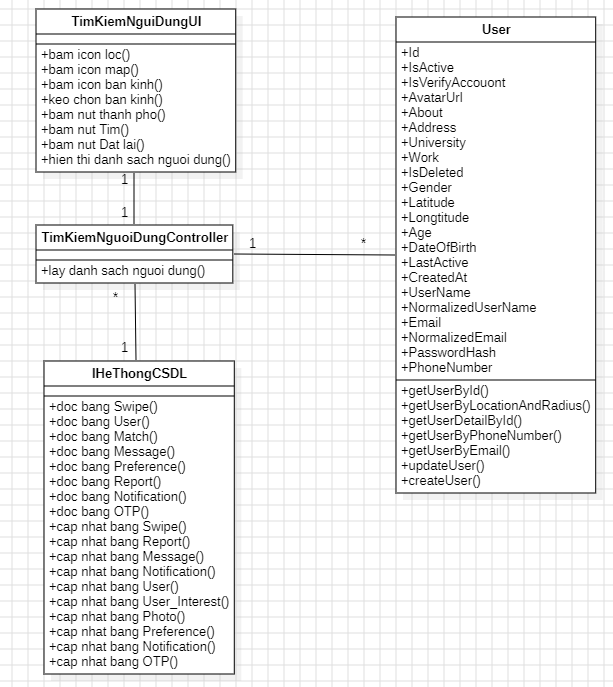
* **Các yêu cầu đặc biệt:** Hỗ trợ kết quả tìm kiếm nhanh và chính xác.
* **Tiền điều kiện:** Người dùng cần đăng nhập để thực hiện use case.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Xem hồ sơ.
  + - 1. Biểu đồ trình tự





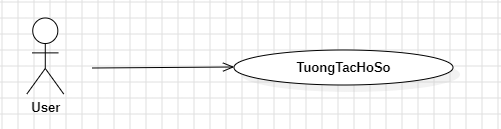
Hình 2‑9: Biểu đồ trình tự UC “Tìm kiếm người dùng”

* + - 1. Biểu đồ lớp chi tiết



Hình 2‑10: Biểu đồ lớp chi tiết use case “Tìm kiếm người dùng”

* + 1. Use case “Tương tác hồ sơ”
       1. Biểu đồ UC



Hình 2‑11: Use case “Tương tác hồ sơ”

* + - 1. Đặc tả UC

**Mô tả vắn tắt:** Use case này cho phép người dùng quẹt trái, phải, xuống dưới để tương tác với hồ sơ người dùng khác.

**Luồng sự kiện**

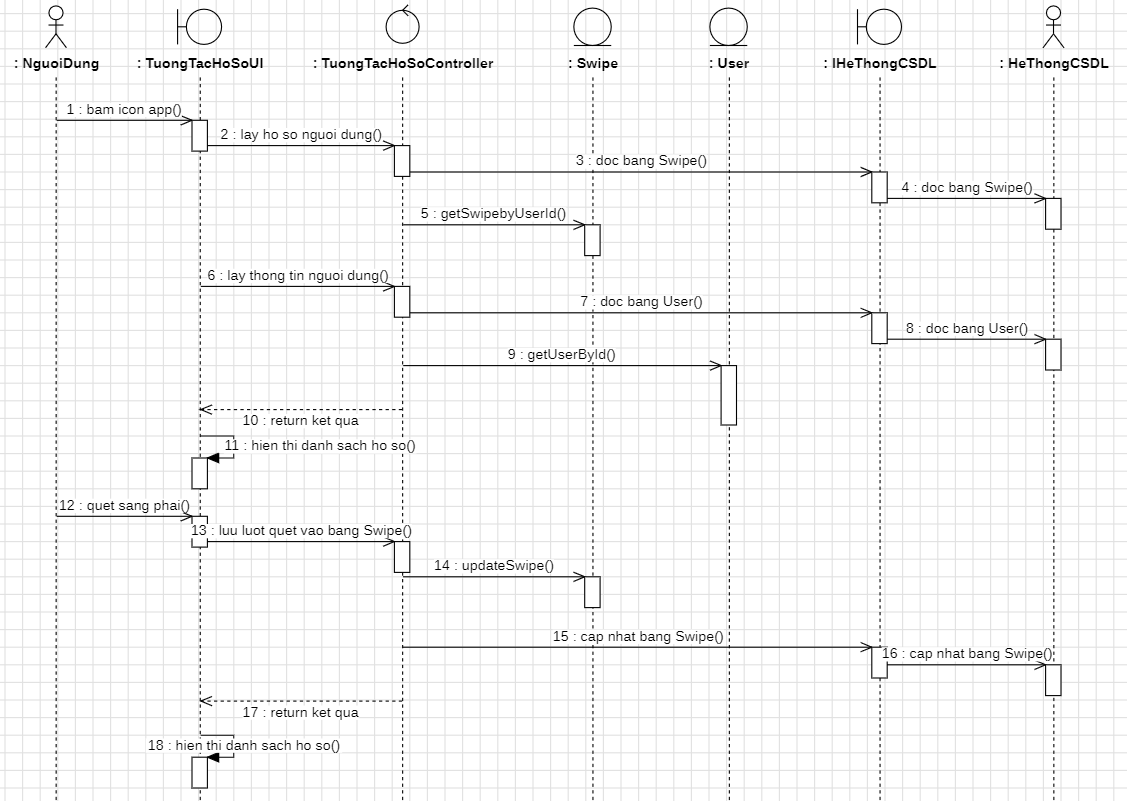
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng bấm vào icon app trên thanh tab bar, hệ thống sẽ ưu tiên chọn những hồ sơ nào mà người dùng chưa từng thấy hoặc đã từng bỏ qua từ bảng Swipe, sau đó sẽ lấy thông tin gồm: ảnh đại diện, tên, tuổi, sở thích từ bảng User trong cơ sở dữ liệu và hiển thị danh sách hồ sơ.
2. Người dùng quẹt hồ sơ hiện tại sang bên phải, Hệ thống sẽ lưu thông tin lượt quẹt với hành động thích vào bảng Swipe trong cơ sở dữ liệu, sau đó sẽ hiển thị hồ sơ tiếp theo.
3. Người dùng quẹt hồ sơ hiện tại sang bên trái, Hệ thống sẽ lưu thông tin lượt quẹt với hành động không thích vào bảng Swipe trong cơ sở dữ liệu, sau đó sẽ hiển thị hồ sơ tiếp theo.
4. Người dùng quẹt hồ sơ hiện tại xuống bên dưới, hệ thống sẽ hiển thị thông tin chi tiết của người dùng bao gồm: tên, tuổi, nghề nghiệp, trường học (nếu có), địa chỉ, giới tính, khoảng cách với người dùng hiện tại, giới thiệu bản thân, sở thích từ bảng User trong cơ sở dữ liệu.

* **Luồng rẽ nhánh:**

1. Tại bất kì bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị “Có lỗi xảy ra” và use case kết thúc.

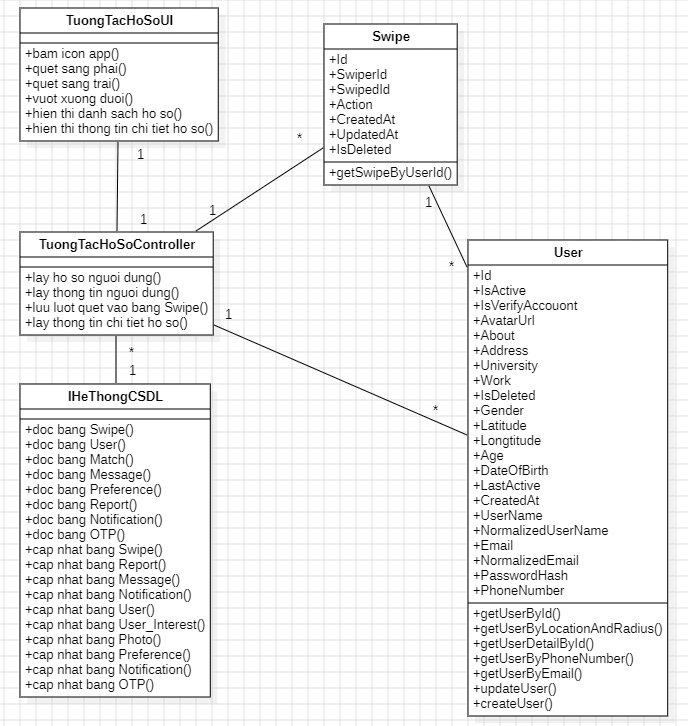
* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** Người dùng cần đăng nhập để thực hiện use case.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Xem ho so.
  + - 1. Biểu đồ trình tự





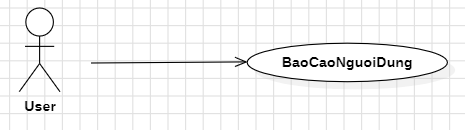
Hình 2‑12: Biểu đồ trình tự UC” Tương tác hồ sơ”

* + - 1. Biểu đồ lớp chi tiết



Hình 2-13: Biểu đồ lớp chi tiết use case “Tương tác hồ sơ”

* + 1. Use case “Báo cáo người dùng”
       1. Biểu đồ UC



Hình 2‑14: Use case “Báo cáo người dùng”

* + - 1. Đặc tả UC

**Mô tả vắn tắt:** Use case này cho phép người báo cáo các hành vi không phù hợp với người dùng khác.

**Luồng sự kiện:**

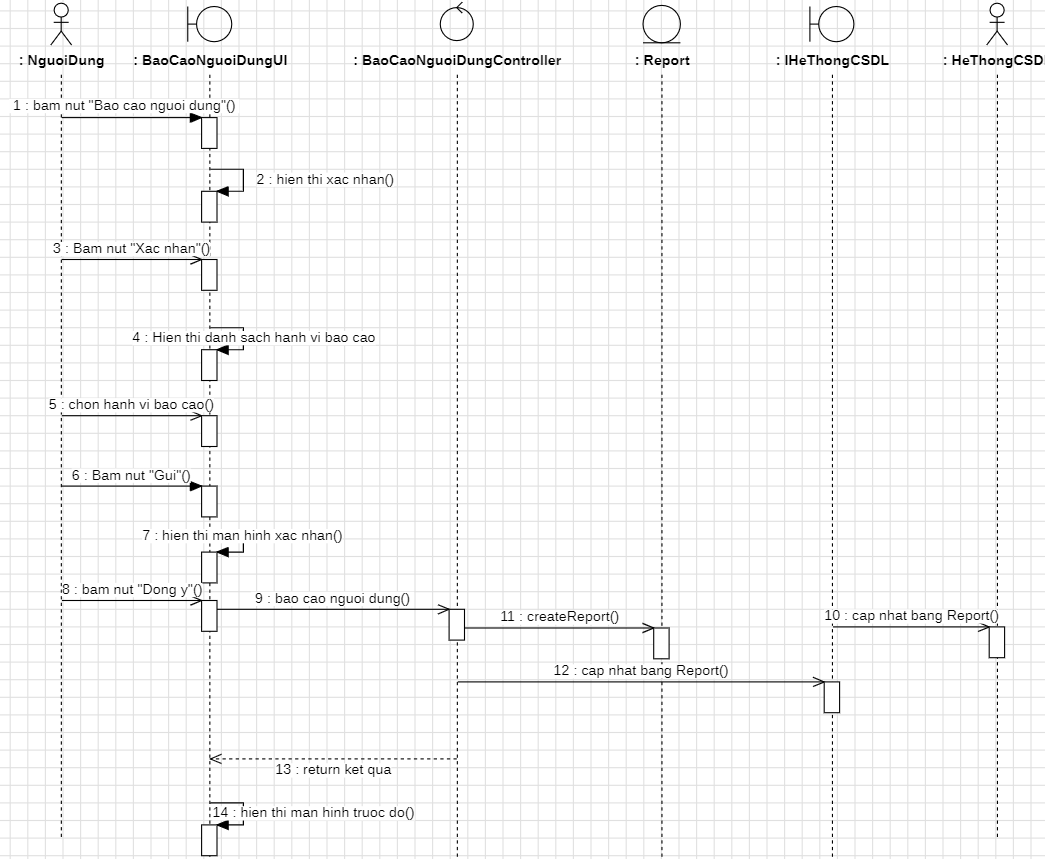
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng bấm nút báo cáo người dùng, Hệ thống màn hình yêu cầu xác nhận báo cáo. Người dùng bấm nút “Xác nhận”, hệ thống sẽ hiển thị màn hình danh sách chi tiết các nội dung, hành vi muốn báo cáo.
2. Người dùng chọn nội dung hành vi báo cáo. Sau đó bấm nút “Gửi”, hệ thống hiển thị màn hình xác nhận, người dùng bấm “Đồng ý”, hệ thống sẽ cập nhật bảng Report trong cơ sở dữ liệu và thoát khỏi màn hình báo cáo.

* **Luồng rẽ nhánh**

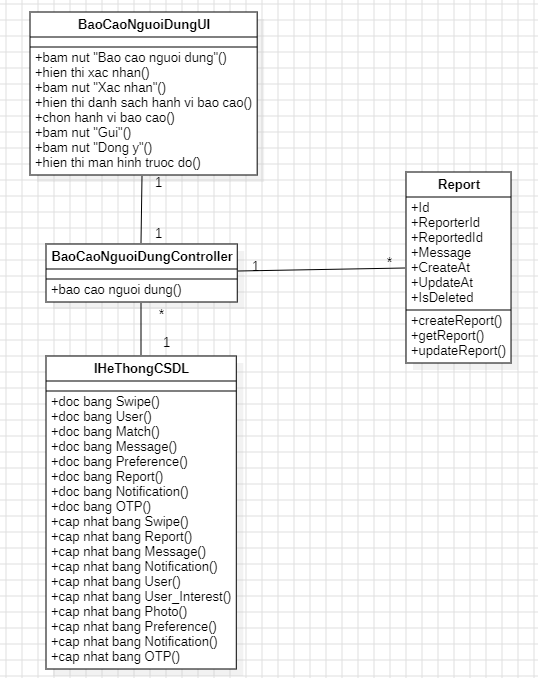
1. Tại mục 1 và 2 trong luồng cơ bản khi người bấm nút “Hủy bỏ”, hệ thống sẽ hủy bỏ quá trình báo cáo và quay trở lại màn hình trước đó. Use case kết thúc.
2. Tại bất cứ thời điểm nào trong luồng cơ bản, nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt:** Không có.
* **Tiền điều kiện:** Người dùng cần đăng nhập để thực hiện use case.
* **Hậu điều kiện:** Không có.
  + - 1. Biểu đồ trình tự



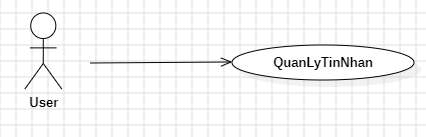
Hình 2‑15: Biểu đồ trình tự UC” Báo cáo người dùng”

* + - 1. Biểu đồ lớp chi tiết



Hình 2‑16: Biểu đồ lớp chi tiết use case “Báo cáo người dùng”

* + 1. Use case “Quản lý tin nhắn”
       1. Biểu đồ UC



Hình 2‑17: Use case “Quản lý tin nhắn”

* + - 1. Đặc tả UC

**Mô tả vắn tắt**: Use case này cho phép người dùng xem, gửi tin nhắn.

**Luồng sự kiện**

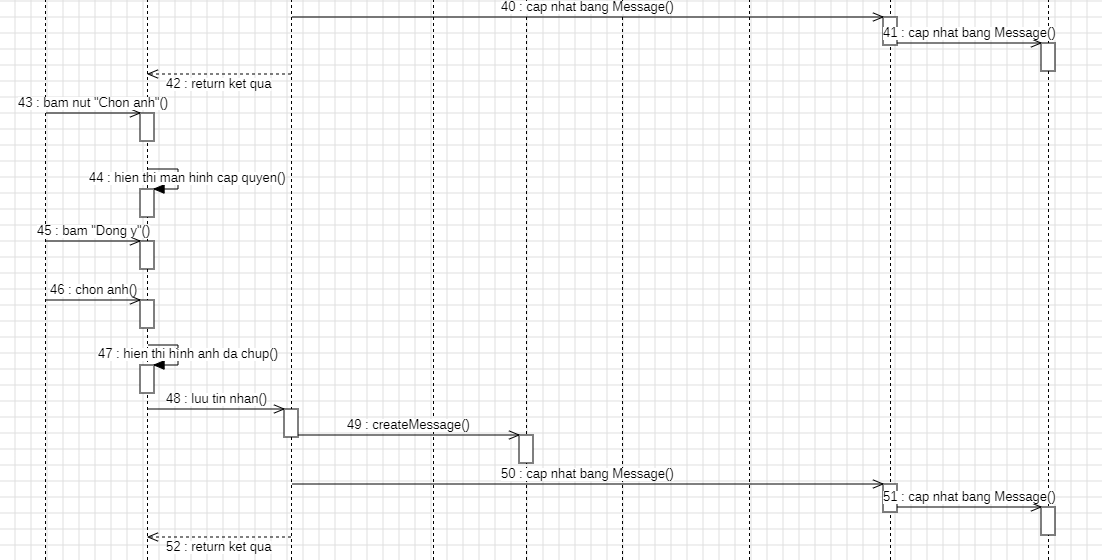
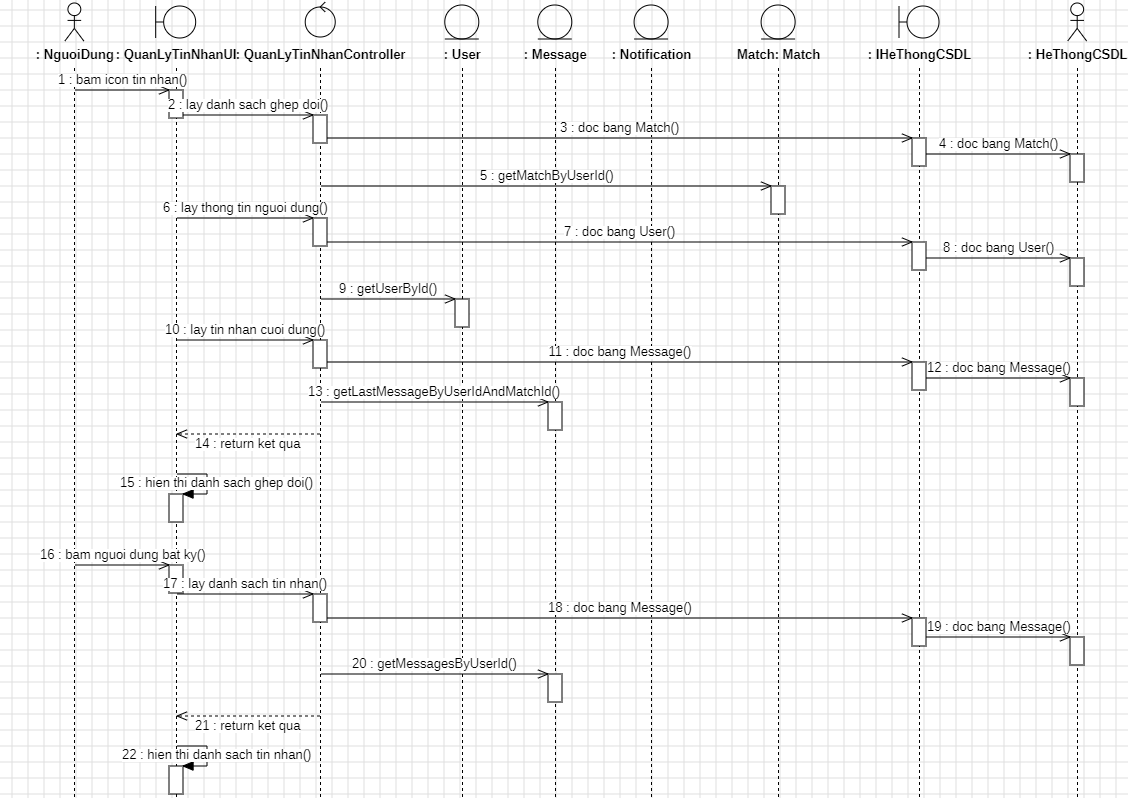
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng bấm vào icon tin nhắn trên thanh tab bar. Hệ thống hiển thị danh sách người dùng đã ghép đôi từ bảng Match và lấy thông tin bao gồm: ảnh đại diện, tên từ bảng User, tin nhắn cuối cùng từ bảng Message trong cơ sở dữ liệu.
2. Người dùng bấm vào 1 dòng bất kỳ, hệ thống hiển thị danh sách tin nhắn bao gồm tin nhắn văn bản, hình ảnh từ bảng Message.
3. Người dùng nhập tin nhắn văn bản vào ô nhập dữ liệu, sau đó bấm nút “Gửi”. Hệ thống sẽ hiển thị tin nhắn vừa gửi lên màn hình và lưu tin nhắn vào bảng Message
4. Người dùng bấm icon hình ảnh, hệ thống hiển thị màn hình chọn phương thức gửi ảnh.
5. Khi người dùng bấm “Chụp ảnh”, hệ thống hiển thị màn hình cấp quyền truy cập camera nếu chưa có quyền, người dùng bấm “Đồng ý” chuyển sang màn hình chụp ảnh. Sau khi chụp ảnh, hệ thống sẽ hiển thị hình ảnh lên màn hình và lưu tin nhắn hình ảnh vào bảng Message.
6. Khi người dùng bấm “Chọn ảnh”, hệ thống hiển thị màn hình cấp quyền đọc, ghi dữ liệu nếu chưa có quyền, người dùng bấm “Đồng ý” chuyển sang màn hình chọn ảnh. Sau khi chọn ảnh, hệ thống sẽ hiển thị hình ảnh lên màn hình và lưu tin nhắn hình ảnh vào bảng Message.

* **Luồng rẽ nhánh**

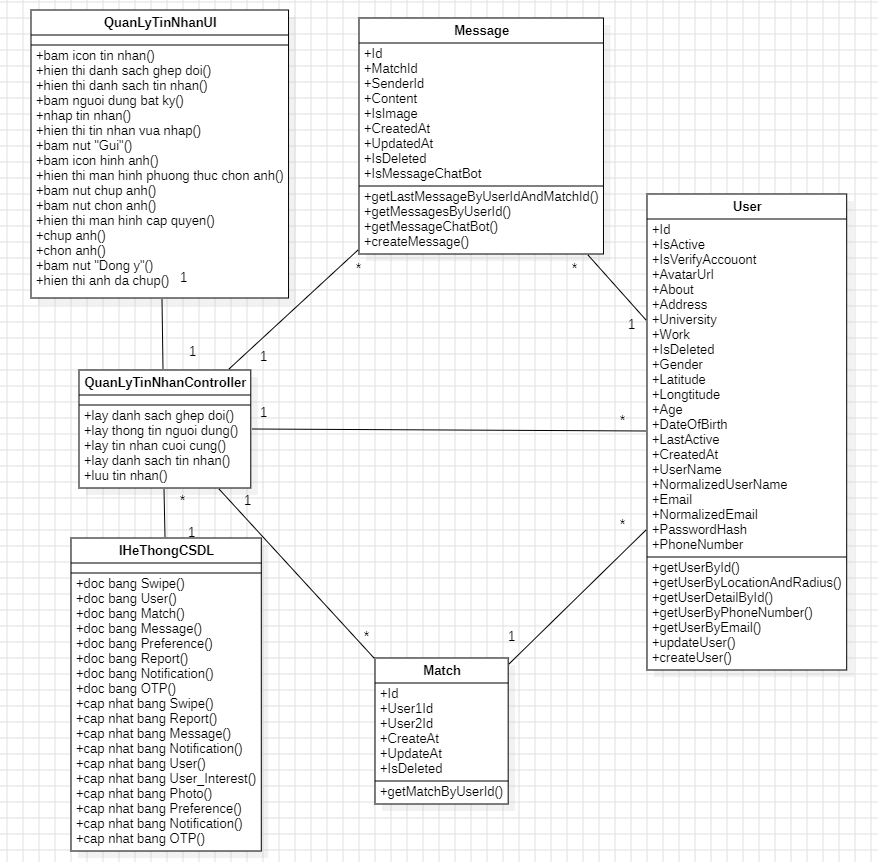
1. Tại bước 1 trong luồng cơ bản, nếu chưa có người ghép đôi nào, hệ thống sẽ hiển thị “Bạn chưa có lượt ghép đôi nào” lên màn hình và use case kết thúc.
2. Tại bước 2 trong luồng cơ bản, nếu chưa có tin nhắn nào, hệ thống sẽ hiển thị “Chưa có tin nhắn” và use case kết thúc.
3. Tại bước 5 và bước 6 trong luồng cơ bản, nếu người dùng không cấp quyền truy cập, hệ thống sẽ quay trở lại màn hình tin nhắn và use case kết thúc.
4. Tại bất cứ bước nào trong luồng cơ bản xảy ra lỗi, hệ thống sẽ hiển thị thông báo “Có lỗi xảy ra” và use case kết thúc.

* **Các yêu cầu đặc biệt:** Người dùng cần đăng nhập để thực hiện use case.
* **Tiền điều kiện:** Không có.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Báo cáo người dùng.
  + - 1. Biểu đồ trình tự



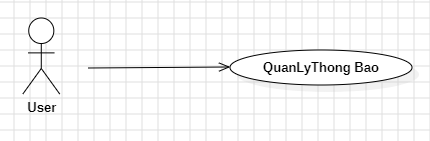
Hình 2‑18: Biểu đồ trình tự UC “Quản lý tin nhắn”

* + - 1. Biểu đồ lớp chi tiết



Hình 2‑19: Biểu đồ lớp chi tiết use case “Quản lý tin nhắn”

* + 1. Use case “Quản lý thông báo”
       1. Biểu đồ UC



Hình 2‑20: Use case “Quản lý thông báo”

* + - 1. Đặc tả UC

**Mô tả vắn tắt:** Use case này cho phép người dùng xem, đánh dấu là đã đọc và xóa thông báo.

**Luồng sự kiện**

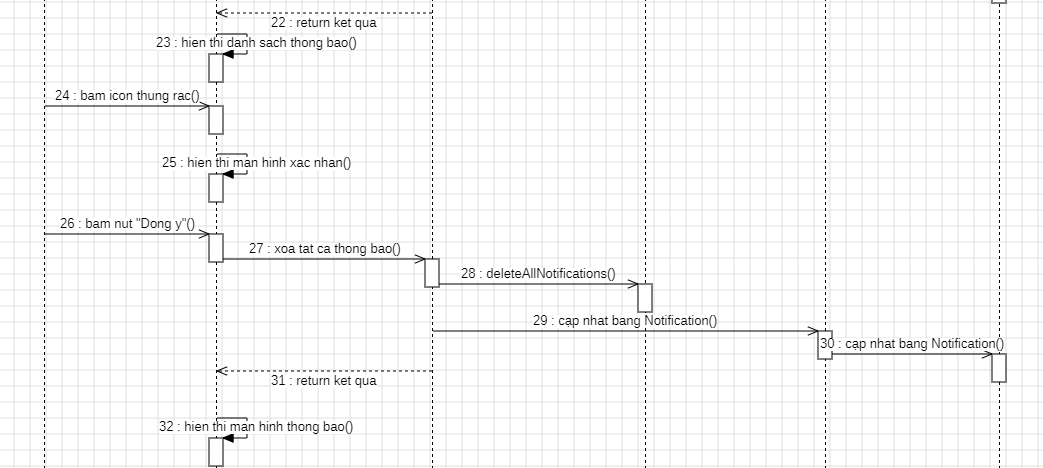
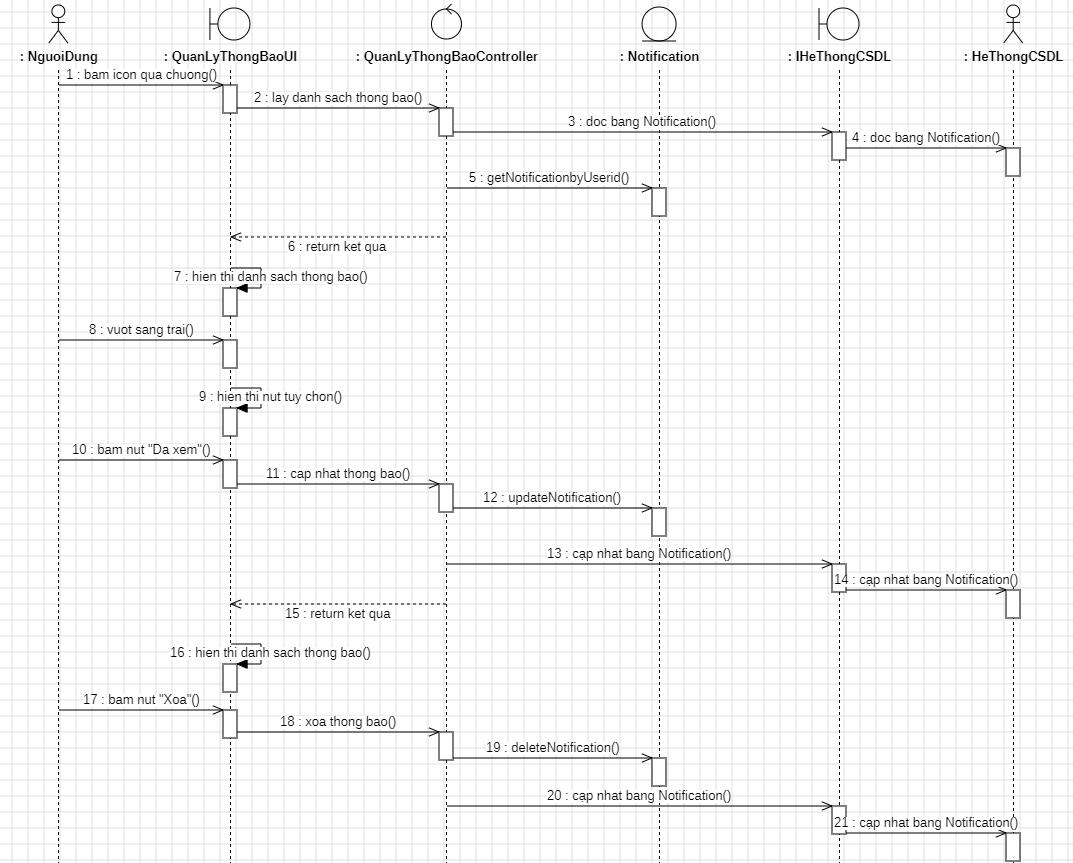
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng bấm vào icon quả chuông trên thanh tab bar. Hệ thống sẽ hiển thị danh sách thông báo bao gồm: hình ảnh người gửi, tiêu đề, nội dung từ bảng Notification trong cơ sở dữ liệu.
2. Người dùng vuốt thông báo sang bên trái, hiển thị 2 nút tùy chọn với thông báo đó.
3. Người dùng bấm nút “Đã xem”, hệ thống cập nhật bảng Notification và hiển thị lại danh sách thông báo với thông báo đã xem được xếp cuối cùng.
4. Người dùng bấm nút “Xóa”, hệ thống xóa thông báo và cập nhật bảng Notifications sau đó hiển thị lại danh sách thông báo với thông báo.
5. Người dùng bấm icon thùng rác trên thanh navigation bar, hệ thống hiển thị màn hình xác nhận muốn xóa tất cả thông báo hay không. Khi người dùng bấm nút “Đồng ý”, hệ thống xóa tất cả thông báo và cập nhật bảng Notification, sau đó quay trở lại màn hình thông báo.

**Luồng rẽ nhánh:**

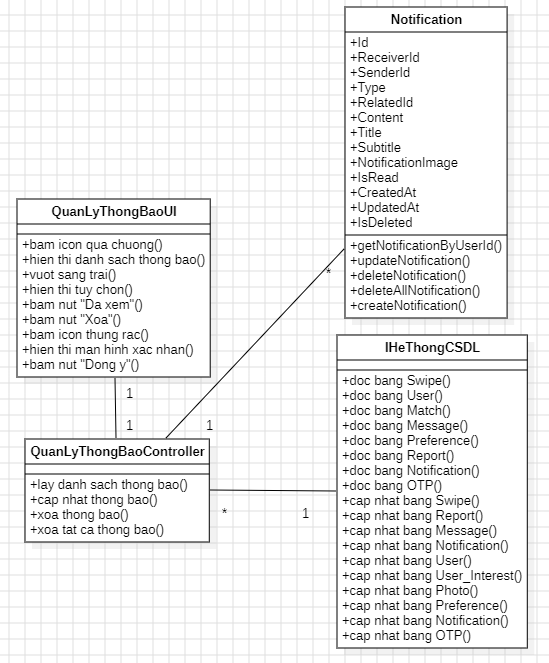
1. Tại bước 5 trong luồng cơ bản, nếu người dùng bấm “Hủy bỏ” thì hệ thống dừng quá trình xóa toàn bộ thông báo, quay trở lại màn hình danh sách thông báo và use case kết thúc.
2. Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo “Có lỗi xảy ra” và use case kết thúc.

* **Các yêu cầu đặc biệt:** Người dùng cần đăng nhập để thực hiện use case.
* **Tiền điều kiện:** Không có.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Quản lý tin nhắn.
  + - 1. Biểu đồ trình tự



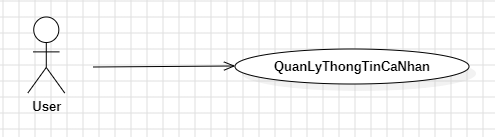
Hình 2‑21: Biểu đồ trình tự UC “Quản lý thông báo”

* + - 1. Biểu đồ lớp chi tiết



Hình 2‑22: Biểu đồ lớp chi tiết use case “Quản lý thông báo”

* + 1. Use case “Quản lý thông tin cá nhân”
       1. Biểu đồ UC



Hình 2‑23: Use case “Quản lý thông tin cá nhân”

* + - 1. Đặc tả UC

**Mô tả vắn tắt:** Use case này cho phép người dùng chỉnh sửa thông tin cá nhân.

**Luồng sự kiện**

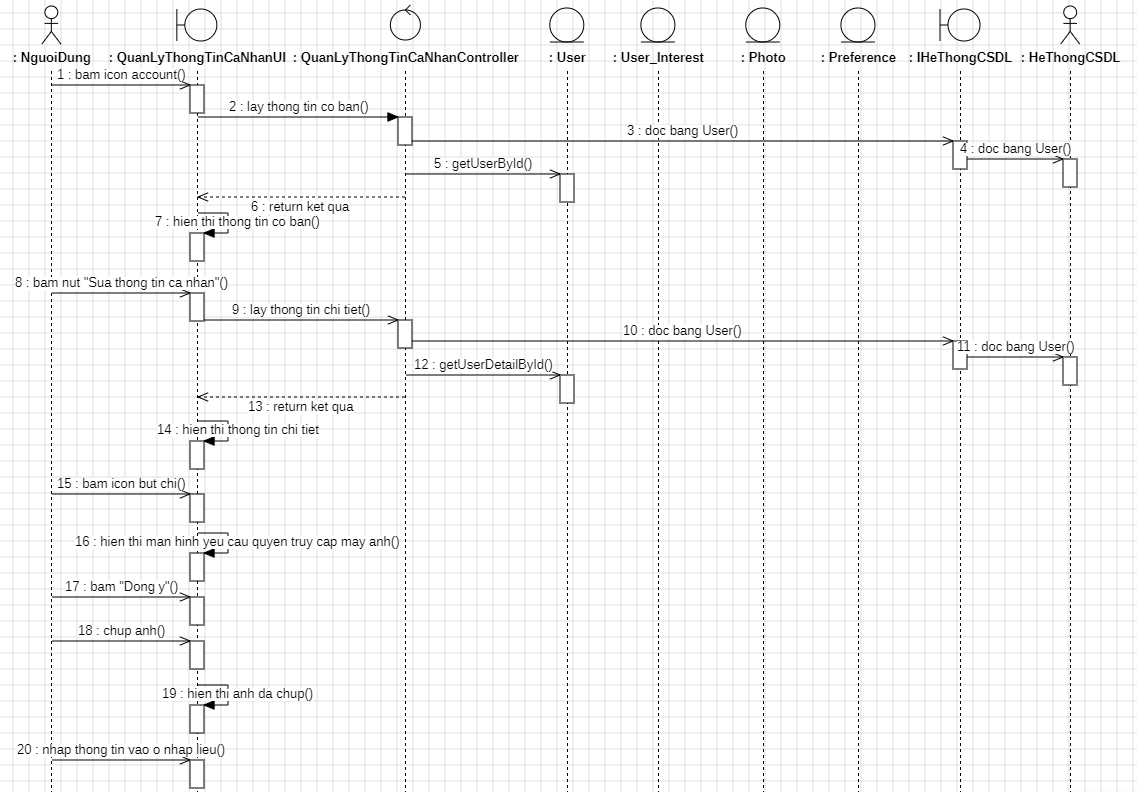
* **Luồng cơ bản:**

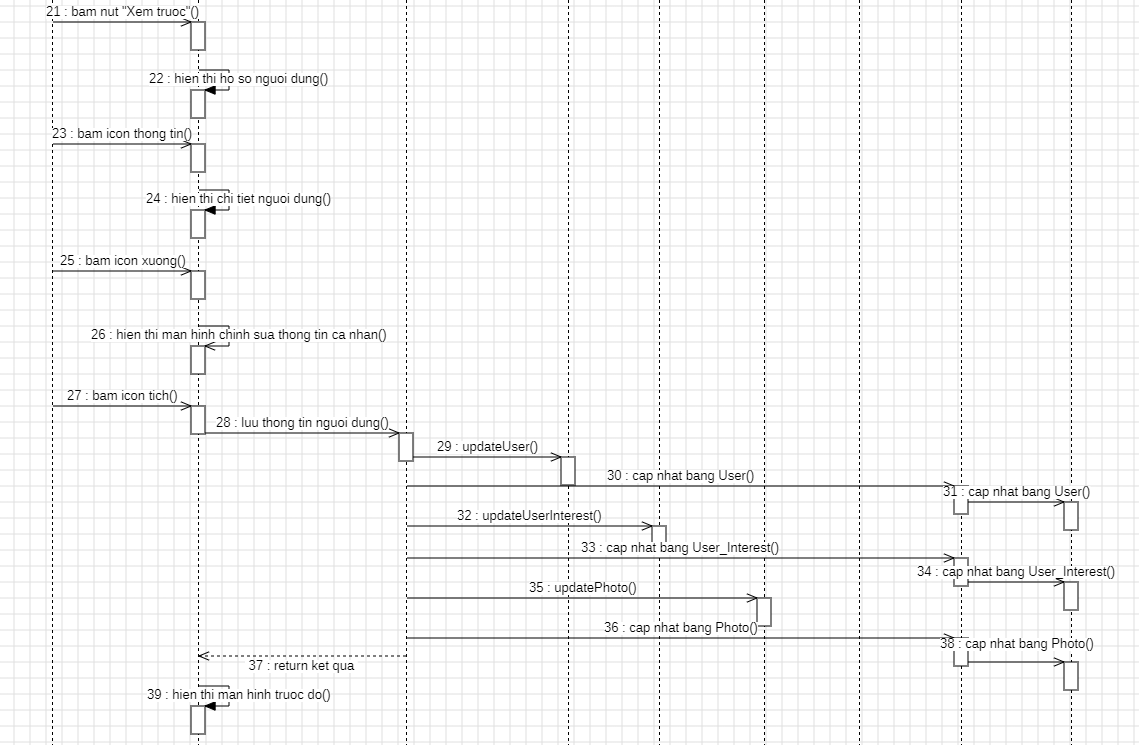
1. Use case này bắt đầu khi người dùng bấm vào icon account trên thanh tab bar, hệ thống hiển thị tên, tuổi từ bảng User trong cơ sở dữ liệu và các nút tùy chọn.
2. Khi người dùng bấm vào nút “Sửa thông tin cá nhân”, hệ thống sẽ hiện thị ảnh, giới thiệu, sở thích, giới tính, đại học, công việc từ bảng User trong cơ sở dữ liệu.
3. Khi người dùng bấm vào icon bút chì để chụp ảnh hoặc chỉnh sửa ảnh, hệ thống hiển thị màn hình yêu cầu quyền truy cập máy ảnh nếu chưa cấp, người dùng bấm “Đồng ý”, sau khi chụp ảnh hệ thống sẽ hiển thị hình ảnh. Người dùng nhập các thông tin còn lại vào ô nhập liệu, sau đó bấm nút “Xem trước”, hệ thống hiển thị hồ sơ của người dùng sẽ được hiển thị cho người dùng khác. Khi bấm vào icon thông tin, hệ thống sẽ hiển thị màn hình chi tiết hồ sơ của người dùng. Bấm icon mũi tên xuống để quay trở lại màn hình chỉnh sửa thông tin cá nhân.
4. Khi người dùng bấm icon tích, hệ thống sẽ cập nhật thông tin vào bảng User, User\_Interest, Photo trong cơ sở dữ liệu và quay trở lại màn hình trước đó.
5. Khi người dùng bấm icon cài đặt, hệ thống sẽ hiển thị thông tin bao gồm: số điện thoại, email, địa chỉ, giới tính hiển thị, độ tuổi hiển thị từ bảng User và Preference trong cơ sở dữ liệu. Người dùng nhập dữ liệu vào ô nhập liệu và bấm vào icon tích. Hệ thống sẽ cập nhật thông tin vào bảng User, Preference trong cơ sở dữ liệu và quay trở lại màn hình trước đó.

* **Luồng rẽ nhánh**

1. Tại bước 3 trong luồng cơ bản, nếu người dùng chọn “Hủy bỏ”, hệ thống sẽ không hiển thị màn hình chụp ảnh và use case kết thúc.
2. Tại bất kỳ thời điểm nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo “Có lỗi xảy ra” và use case kết thúc

* **Các yêu cầu đặc biệt:** Người dùng cần đăng nhập để thực hiện use case.
* **Tiền điều kiện:** Không có.
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Đăng xuất.
  + - 1. Biểu đồ trình tự

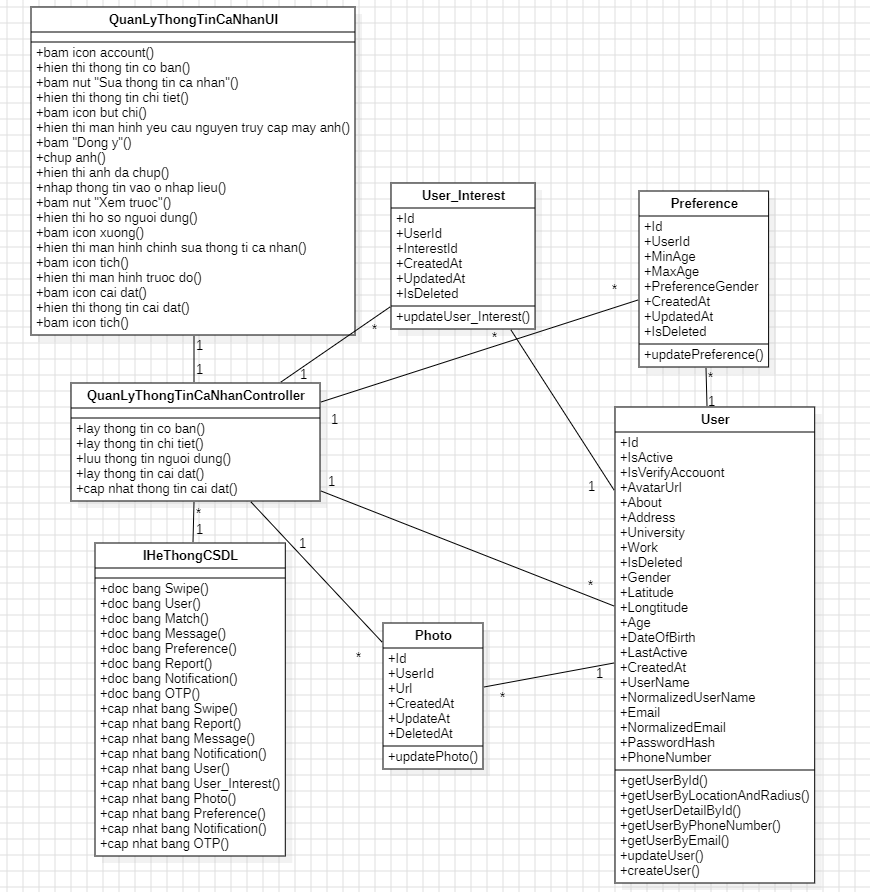






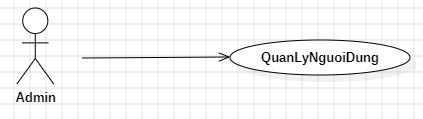
Hình 2‑24: Biểu đồ trình tự UC “Quản lý thông tin cá nhân”

* + - 1. Biểu đồ lớp chi tiết



Hình 2‑25: Biểu đồ lớp chi tiết use case “Quản lý thông tin cá nhân”

* + 1. Use case “Quản lý người dùng”
       1. Biểu đồ UC



Hình 2‑26: Use case “Quản lý người dùng”

* + - 1. Đặc tả UC

**Mô tả vắn tắt:** Use case này cho phép người quản trị quản lý danh sách tài khoản người dùng, xem thông tin người dùng, khóa và mở tài khoản người dùng.

**Luồng sự kiện**

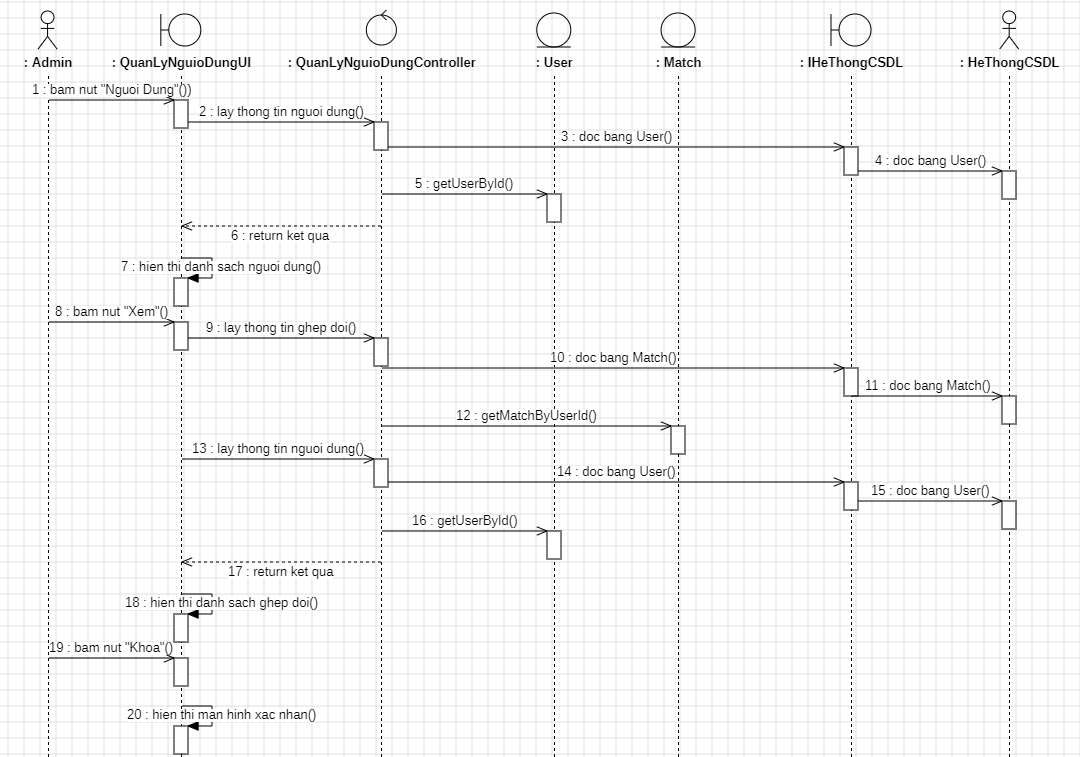
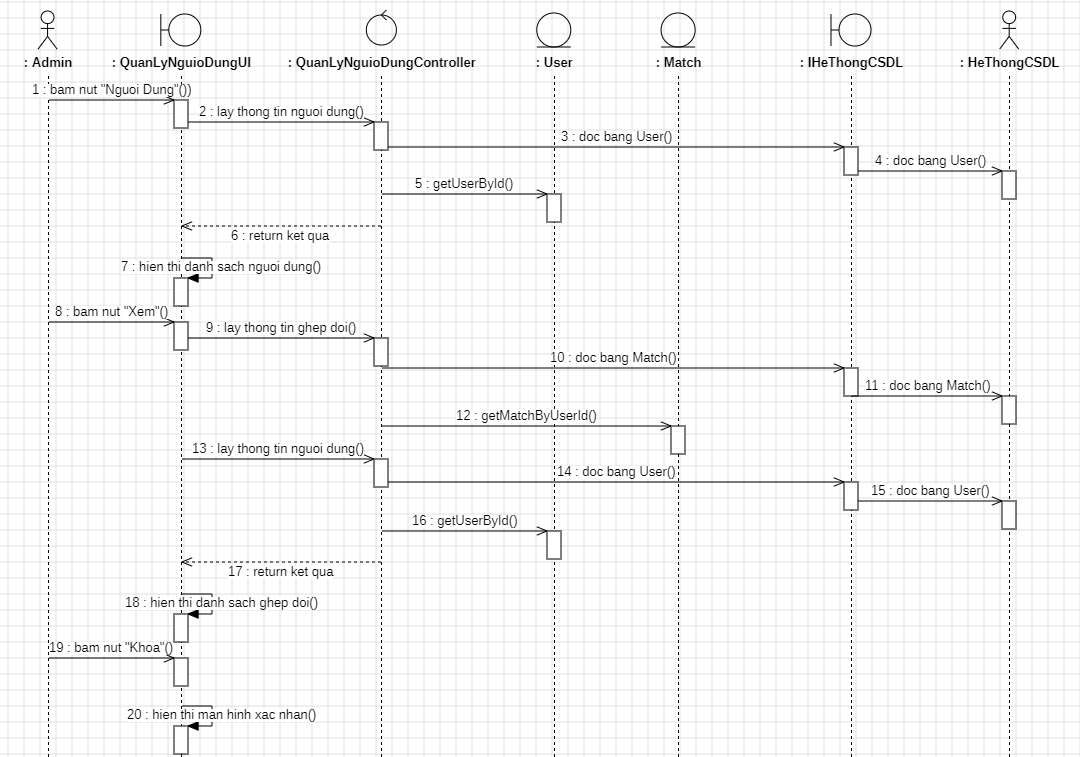
* **Luồng cơ bản:**

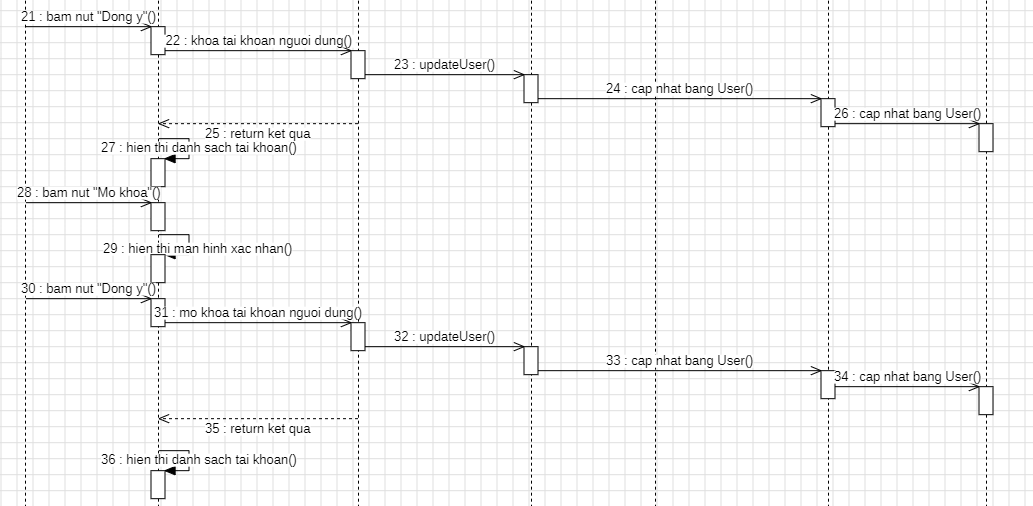
1. Use case này bắt đầu khi người quản trị bấm vào nút “Người dùng” trên menu quản trị. Hệ thống sẽ lấy thông tin về người dùng từ bảng User trong cơ sở dữ liệu và hiện thị lên màn mình.
2. Xem thông tin tài khoản:
3. Người quản trị bấm vào nút “Xem” trên cửa sổ danh sách tài khoản. Hệ thống lây tất cả thông tin của người dùng và những người dùng đã được ghép đôi từ bảng User, Match lên màn hình.
4. Khóa/ Mở khóa tài khoản người dùng:
5. Người quản trị bấm vào nút "Khóa" trên một dòng tài khoản mở khóa hoặc “Mở khóa” trên 1 dòng tài khoản bị khóa. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận Khóa/ Mở khóa.
6. Người quản trị bấm vào nút "Đồng ý". Hệ thống sẽ Khóa/Mở khóa tài khoản người dùng trong bảng User và hiển thị danh sách các tài khoản đã được cập nhật.

* **Luồng rẽ nhánh:**

1. Tại bước 3b trong luồng cơ bản, nếu người quản trị bấm vào nút "Hủy", hệ thống sẽ bỏ qua thao tác khóa và hiển thị danh sách các người dùng trong bảng User và use case kết thúc.
2. Tại bất kỳ thời điểm nào trong luồng cơ bản, hệ thống không kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo “Có lỗi xảy ra” và use case kết thúc.

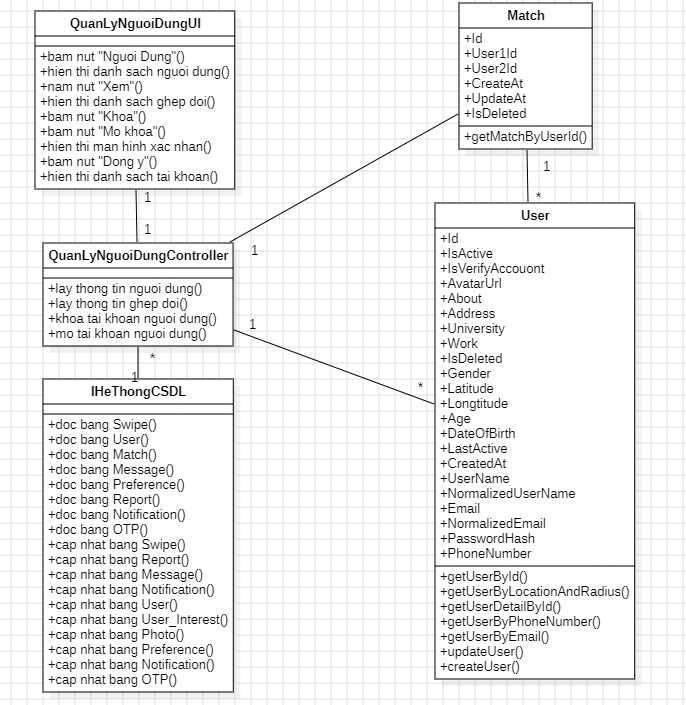
* **Các yêu cầu đặc biệt:** Use case này chỉ cho phép người quản trị thực hiện.
* **Tiền điều kiện:** Người quản trị cần đăng nhập để thực hiện.
* **Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về người dùng sẽ được cập nhật trong cơ sở dữ liệu.
  + - 1. Biểu đồ trình tự





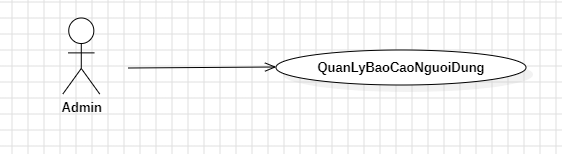
Hình 2‑27: Biểu đồ trình tự UC “Quản lý người dùng”

* + - 1. Biểu đồ lớp chi tiết



Hình 2‑28: Biểu đồ lớp chi tiết use case “Quản lý người dùng”

* + 1. Use case “Quản lý báo cáo người dùng”
       1. Biểu đồ UC



Hình 2-29: Use case “Quản lý báo cáo người dùng”

* + - 1. Đặc tả UC

**Mô tả vắn tắt:** Use case này cho phép người quản trị quản lý, xử lý báo cáo từ người dùng.

**Luồng sự kiện**

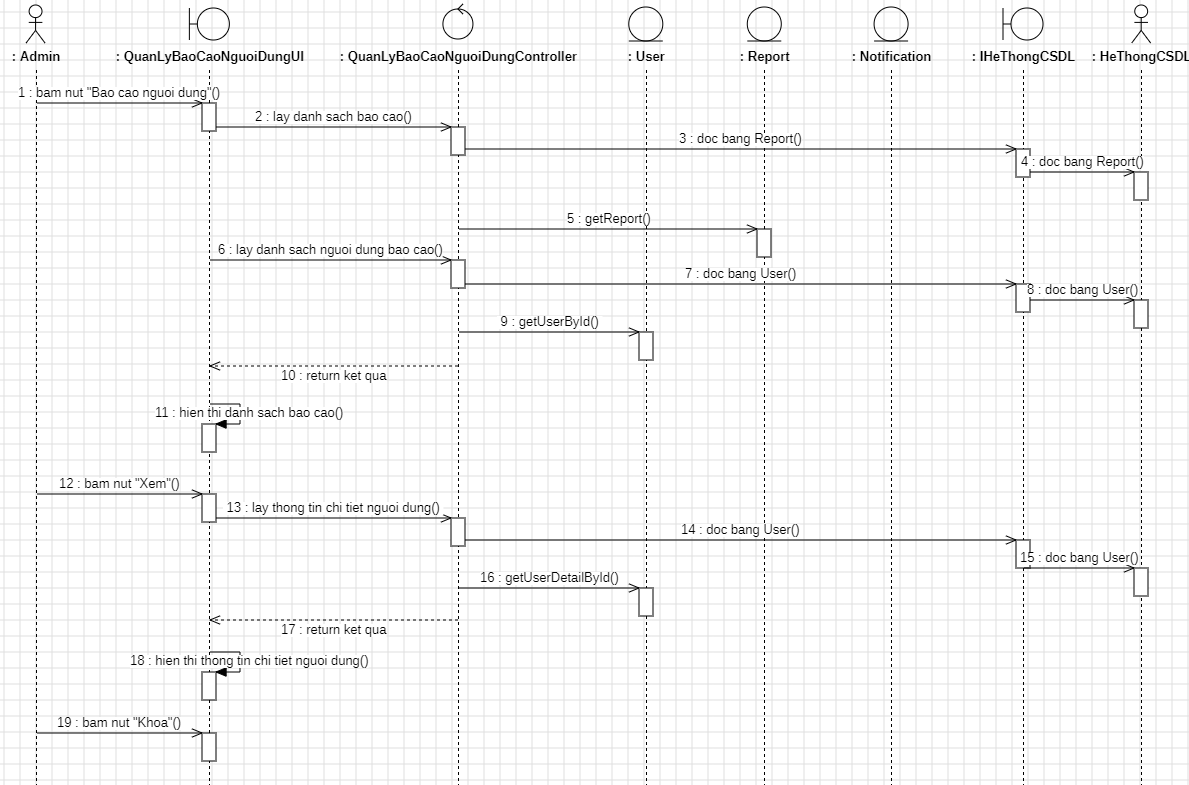
* **Luồng cơ bản:**

1. Use case này bắt đầu khi người quản trị bấm vào nút “Báo cáo người dùng” trên menu quản trị. Hệ thống hiển thị tất cả thông tin về các báo cáo của người dùng, thông tin cơ bản của người báo cáo và người được báo cáo từ bảng User, Report trong cơ sở dữ liệu
2. Người quản trị bấm vào nút “Xem” để xem thông tin chi tiết về người báo cáo/ người được báo cáo.
3. Người quản trị tùy thuộc vào số lượng và hành vi được báo cáo để Khóa/ Gửi thông báo cảnh báo tới người dùng.
   1. Người quản trị bấm vào nút “Khóa” trên 1 dòng tài khoản. Hệ thống hiển thị màn hình yêu cầu xác nhận Khóa.
   2. Người quản trị bấm vào nút "Đồng ý". Hệ thống sẽ Khóa tài khoản người dùng trong bảng User, cập nhật trạng thái báo cáo trong bảng Report và hiển thị danh sách các tài khoản đã được cập nhật.
   3. Người quản trị bấm vào nút cảnh báo, hệ thống thêm dữ liệu vào bảng Notification và gửi cảnh báo tới người dùng.

* **Luồng rẽ nhánh:**

1. Tại bước 3b trong luồng cơ bản, nếu người quản trị bấm vào nút "Hủy", hệ thống sẽ bỏ qua thao tác khóa và hiển thị danh sách các người dùng trong bảng User và use case kết thúc.
2. Tại bất kỳ thời điểm nào trong luồng cơ bản, hệ thống không kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo “Có lỗi xảy ra” và use case kết thúc.

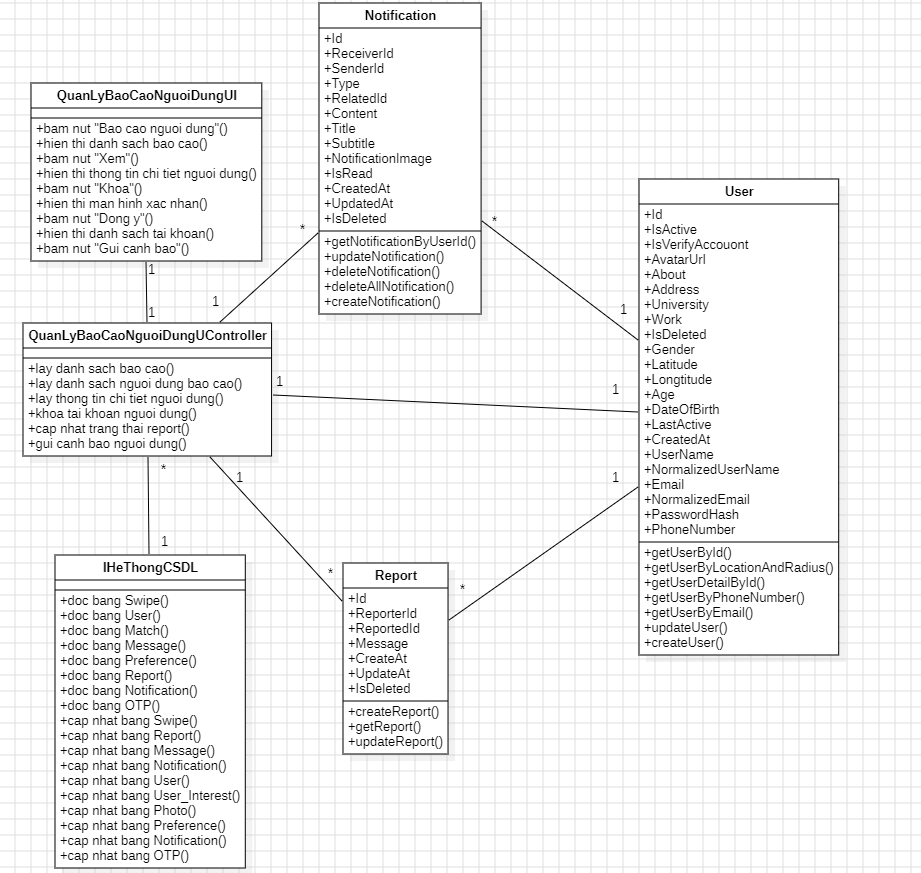
* **Các yêu cầu đặc biệt:** Use case này chỉ cho phép người quản trị thực hiện.
* **Tiền điều kiện:** Người quản trị cần đăng nhập để thực hiện.
* **Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về người dùng sẽ được cập nhật trong cơ sở dữ liệu.
* **Điểm mở rộng:** Không có,
  + - 1. Biểu đồ trình tự





Hình 2-30: Biểu đồ trình tự UC “Quản lý báo cáo người dùng”

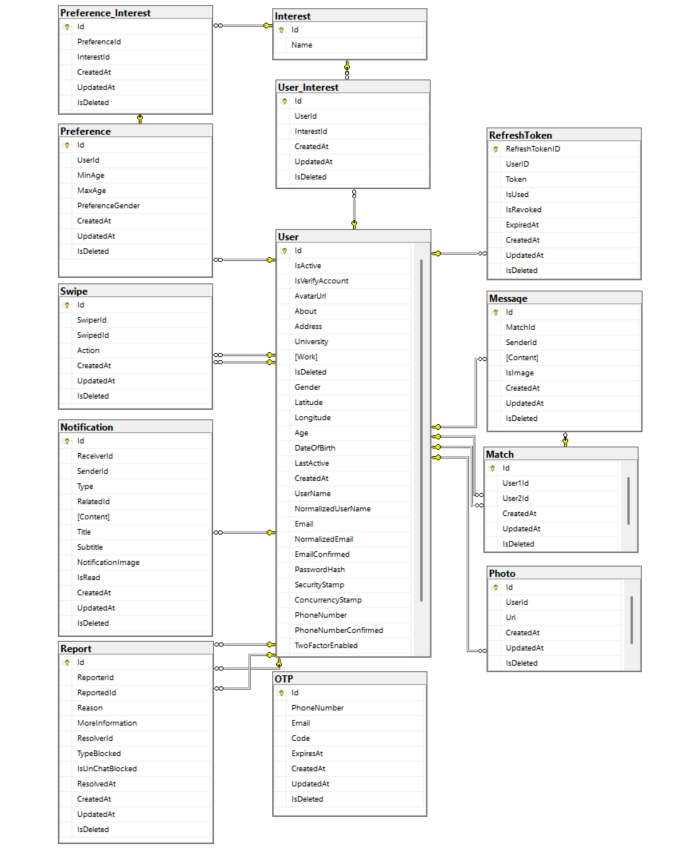
* + - 1. Biểu đồ lớp chi tiết



Hình 2‑31: Biểu đồ lớp chi tiết use case “Quản lý báo cáo người dùng”

* 1. Cơ sở dữ liệu
     1. Sơ đồ dữ liệu quan hệ

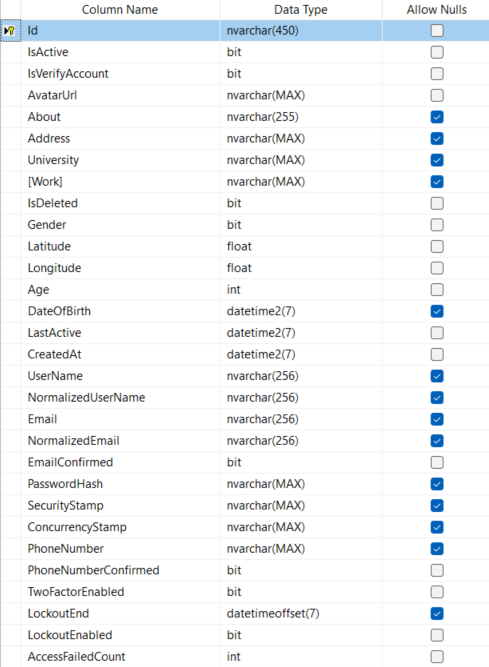
Bảng 2‑2: Bảng sơ đồ dữ liệu quan hệ



Dưới đây sẽ trình bày về các bảng trong CSDL quan hệ của hệ thống được xây dựng:

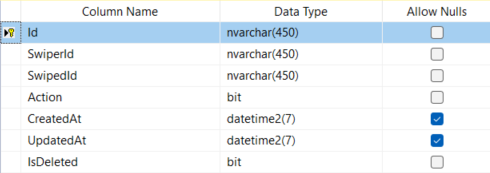
* **User**: Lưu trữ tài khoản người dùng để đăng nhập vào ứng dụng. Chi tiết về tài khoản bao gồm: tên người dùng, mật khẩu, email, số điện thoại, giới tính, công việc, địa chỉ, giới thiệu, tuổi, tên tài khoản, tuổi, ngày sinh, ảnh đại diện.
* **Swipe**: Lưu trữ lượt quẹt trái, phải của người dùng. Chi tiết bảng quẹt bao gồm: người quẹt, người được quẹt, trạng thái quẹt.
* **Match**: Lưu trữ thông tin các cặp đôi đã chấp nhận nhau. Chi tiết bảng matches bao gồm: người dùng thứ nhất, người dùng thứ 2.
* **Message**: Lưu trữ tin nhắn người dùng. Chi tiết bảng tin nhắn gồm: Người gửi, người nhận, nội dung, hình ảnh.
* **Photo**: Lưu trữ thông tin hình ảnh của người dùng. Chi tiết hình ảnh bao gồm: Người dùng, đường dẫn ảnh.
* **Notification**: Lưu trữ thông báo. Chi tiết bảng thông báo bao gồm: Người gửi, người nhận, loại thông báo, nội dung, tiêu đề, hình ảnh.
* **OTP**: Lưu trữ thông tin về mã xác thực. Chi tiết bảng mã xác thực bao gồm: bộ sưu tập, sản phẩm.
* **RefreshToken**: Lưu trữ thông tin về token xác thực. Chi tiết refreshToken bao gồm: người dùng, token, trạng thái sử dụng, thời gian hết hạn.
* **Interest**: Lưu trữ các loại sở thích mặc định. Chi tiết bảng sở thích bao gồm: Sở thích.
* **User\_Interest**: Lưu trữ các loại sở thích của người dùng. Chi tiết bảng sở thích bao gồm: người dùng, sở thích.
* **Preference**: Lưu trữ các tùy chọn khác của người dùng. Chi tiết bảng tùy chọn bao gồm: người dùng, tuổi tối thiểu, tuổi tối đa, giới tính hiển thị.
* **Preference\_Interest**: Lưu trữ kết hợp sở thích và tùy chọn của người dùng. Chi tiết bảng bao gồm: người dùng, sở thích, tùy chọn.
* **Report**: Lưu trữ báo cáo từ người dùng.
  + 1. Chi tiết các bảng cơ sở dữ liệu
       1. **Bảng User**

Bảng 2‑3: Bảng cơ sở dữ liệu User



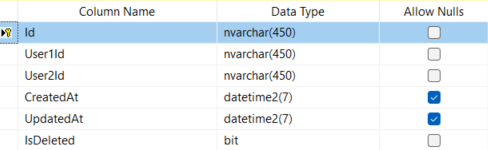
* + - 1. **Bảng Swipe**

Bảng 2‑4: Bảng cơ sở dữ liệu Swipe



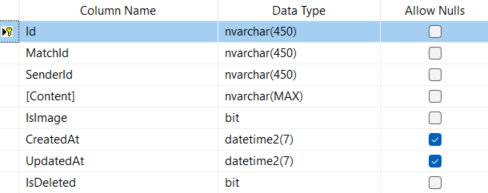
* + - 1. **Bảng Match**

Bảng 2‑5: Bảng cơ sở dữ liệu Match



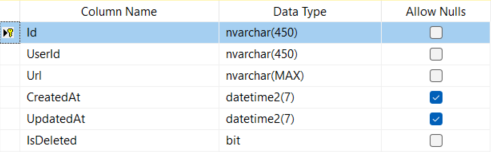
* + - 1. **Bảng Message**

Bảng 2‑6: Bảng cơ sở dữ liệu Message



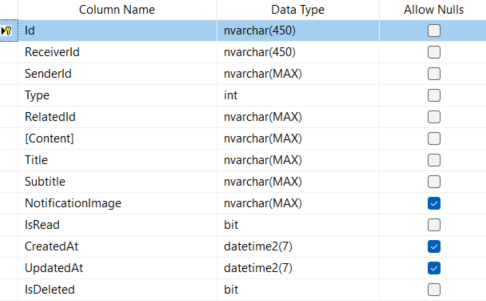
* + - 1. **Bảng Photo**

Bảng 2‑7: Bảng cơ sở dữ liệu Photo



* + - 1. **Bảng Notification**

Bảng 2‑8: Bảng cơ sở dữ liệu Notification



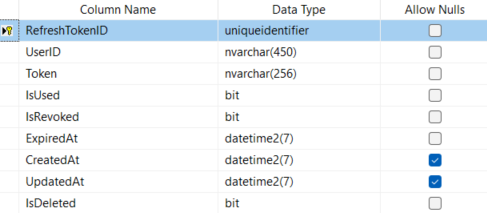
* + - 1. **Bảng OTP**

Bảng 2‑9: Bảng cơ sở dữ liệu OTP



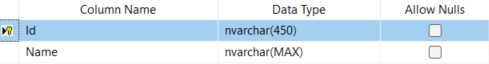
* + - 1. **Bảng RefreshToken**

Bảng 2‑10: Bảng cơ sở dữ liệu RefreshToken



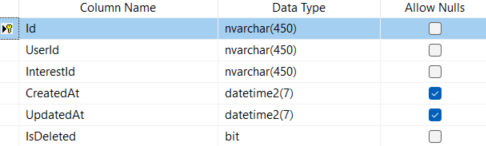
* + - 1. **Bảng Interest**

Bảng 2‑11: Bảng cơ sở dữ liệu Interest



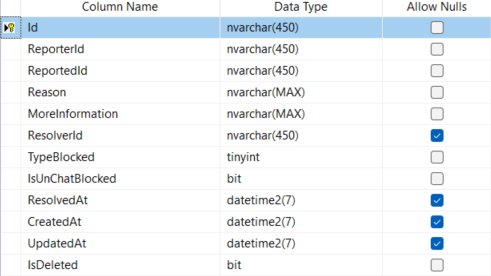
* + - 1. **Bảng User\_Interest**

Bảng 2‑12: Bảng cơ sở dữ liệu User\_Interest



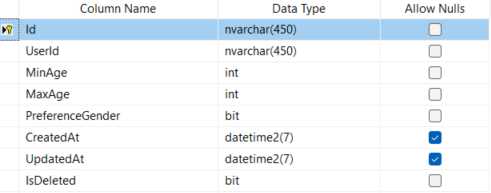
* + - 1. **Bảng Report**

Bảng 2‑13: Bảng cơ sở dữ liệu Report



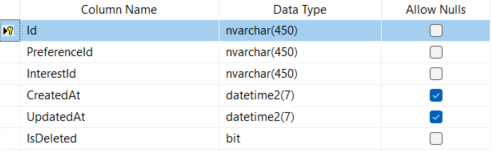
* + - 1. **Bảng Preference**

Bảng 2‑14: Bảng cơ sở dữ liệu Preference

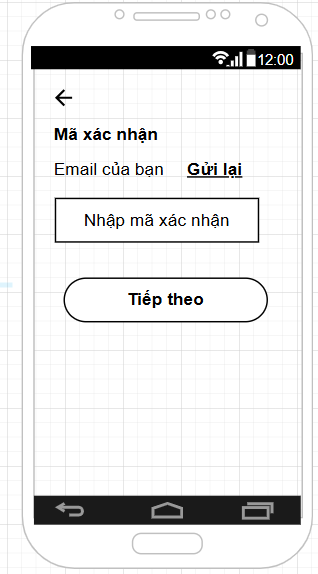
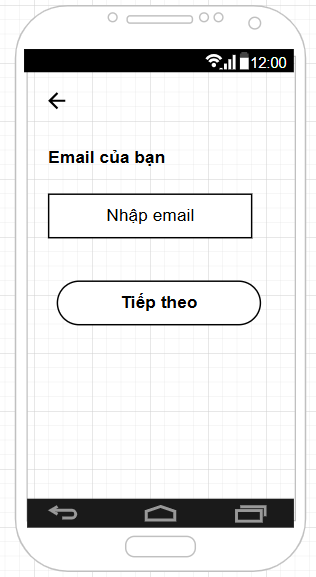
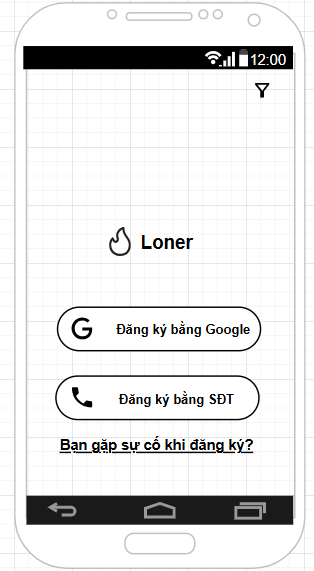


* + - 1. **Bảng Preference\_Interest**

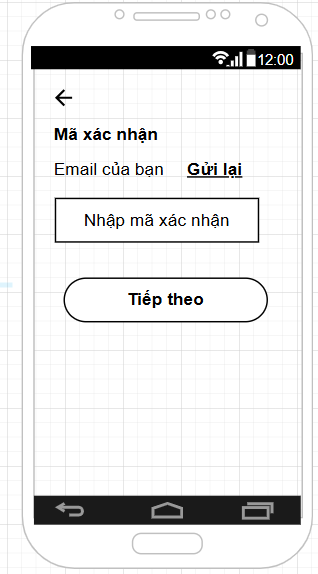
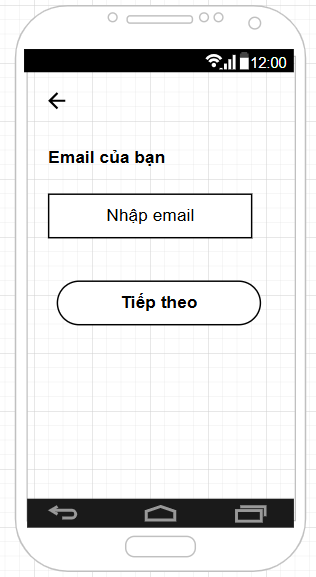
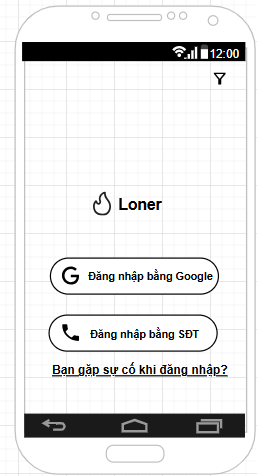
Bảng 2‑15: Bảng cơ sở dữ liệu Prefernce\_Interest



* 1. Thiết kế giao diện chung cho User
     1. Giao diện Đăng nhập, Đăng ký



Hình 2‑32: Thiết kế giao diện Đăng ký



Hình 2‑33: Thiết kế giao diện Đăng nhập

* + 1. Giao diện Trang chủ



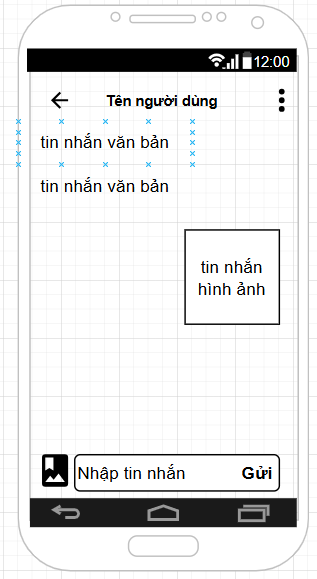
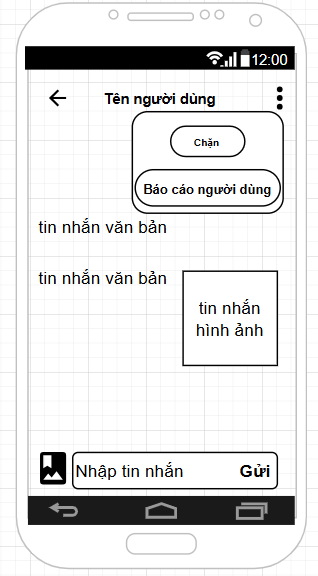
Hình 2‑34: Thiết kế giao diện Trang chủ

* + 1. Giao diện Xem hồ sơ người dùng

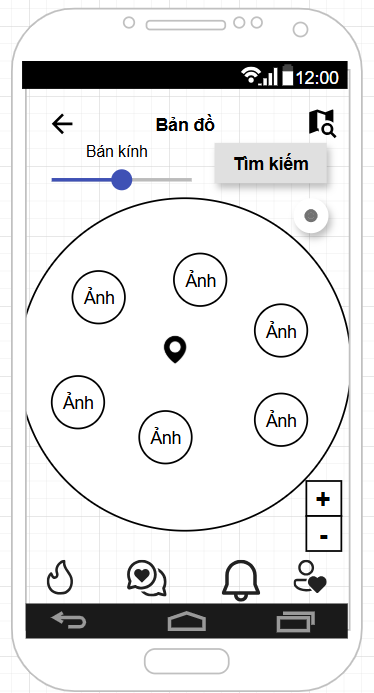
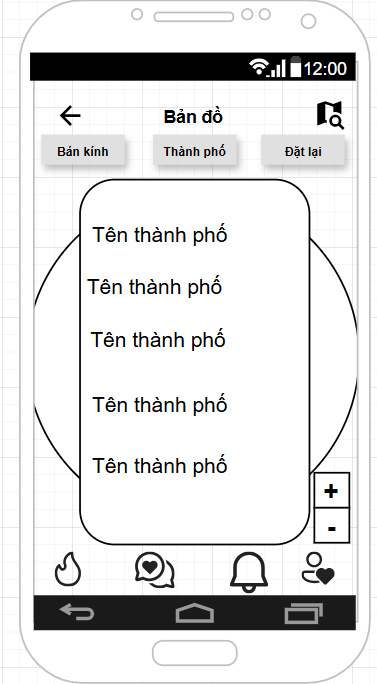
Hình 2‑35: Thiết kế giao diện Xem hồ sơ người dùng

* + 1. Giao diện Nhắn tin

Hình 2-36: Thiết kế giao diện Nhắn tin

* + 1. Giao diện Tìm kiếm người dùng

Hình 2‑37: Thiết kế giao diện Tìm kiếm người dùng

* 1. Kết luận chương 2

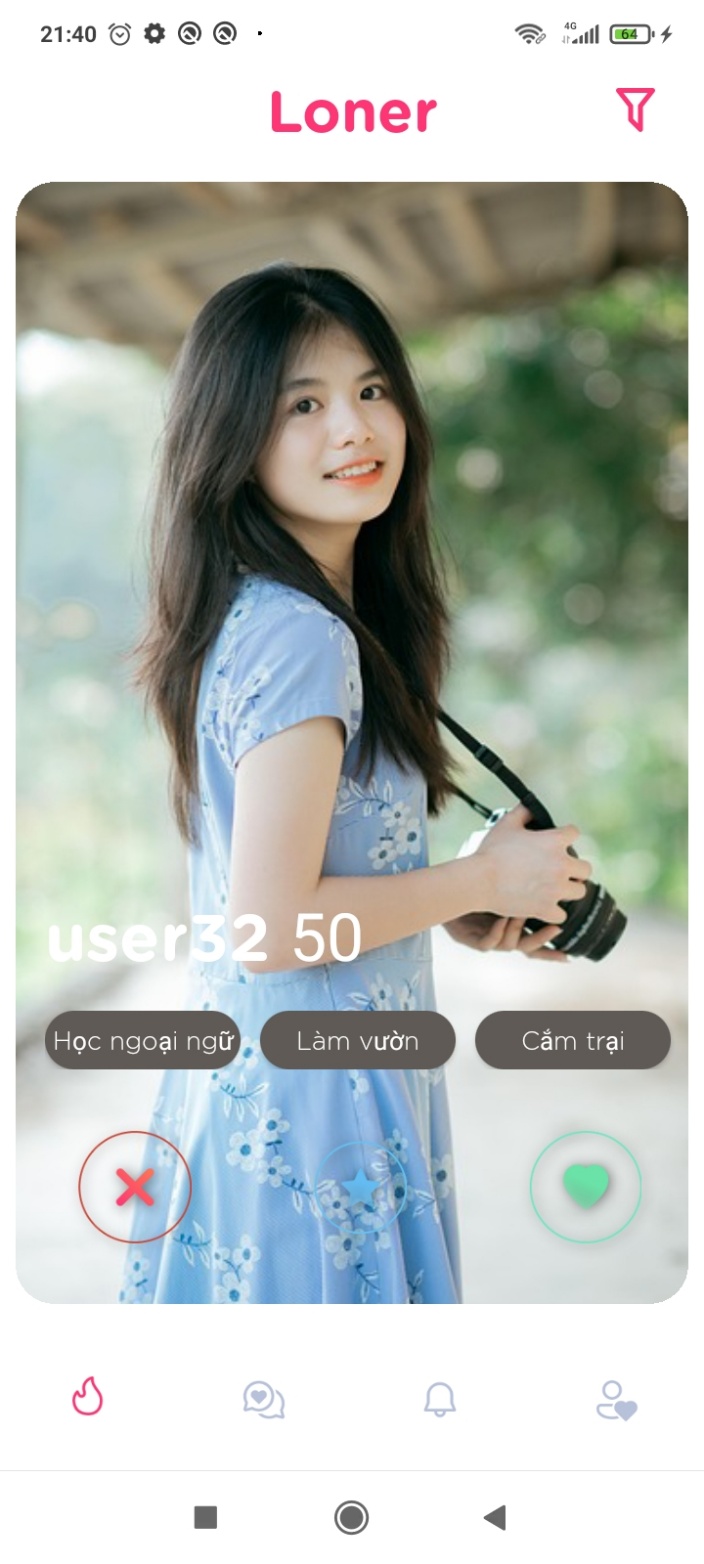
Trong chương 2, em đã trình bày các nghiên cứu, phân tích và thiết kế giao diện của em về ứng dụng hẹn hò Loner bao gồm: các tác nhân, biểu đồ use case tổng quát, đặc tả use case và mô tả cơ sở dữ liệu, thiết kế giao diện màn hình.

# KẾT QUẢ VÀ KIỂM THỬ

* 1. Kết quả sản phẩm
     1. Trang màn hình giao diện của người người dùng

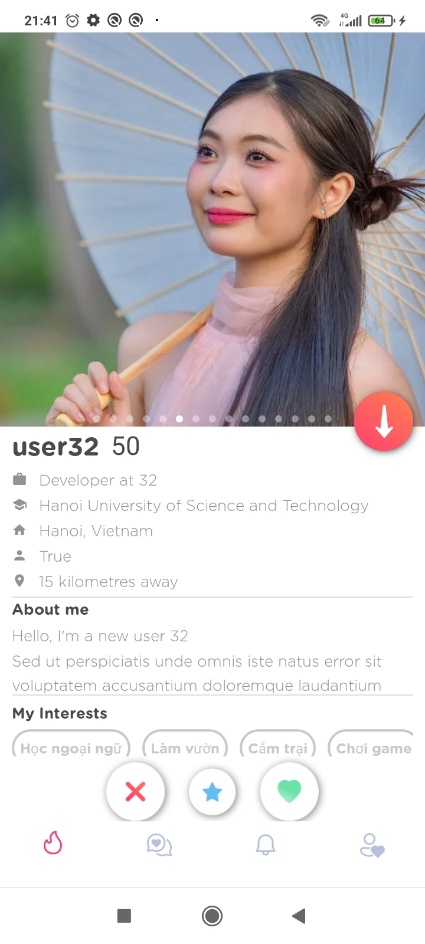
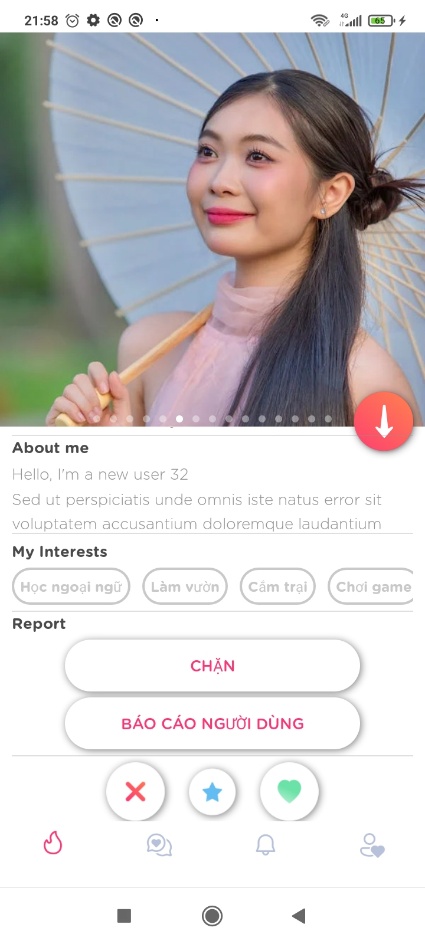
Kết quả xây dựng ứng dụng hẹn hò Loner trên nền tảng Android bằng MAUI và ASP.NET Core Web API được thể hiện thông qua giao diện của các chức năng cụ thể dưới đây.

* + - 1. Giao diện trang chủ (Swipe Page)



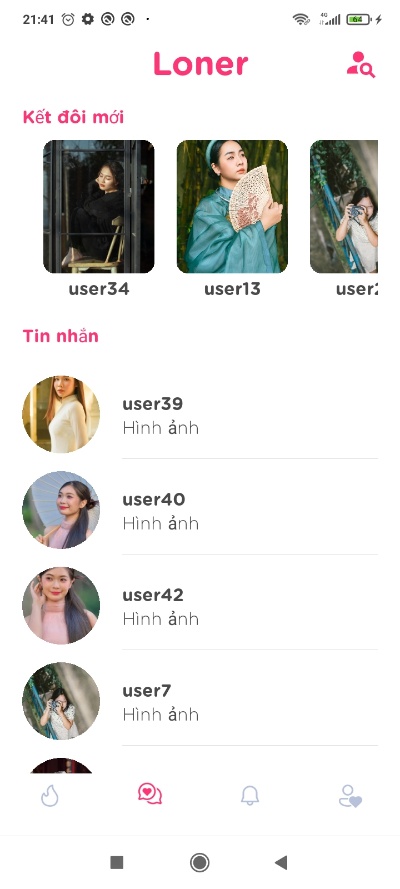
Hình 3‑1: Giao diện trang chủ ứng dụng (Swipe page)

* + - 1. Giao diện xem hồ sơ cá nhân

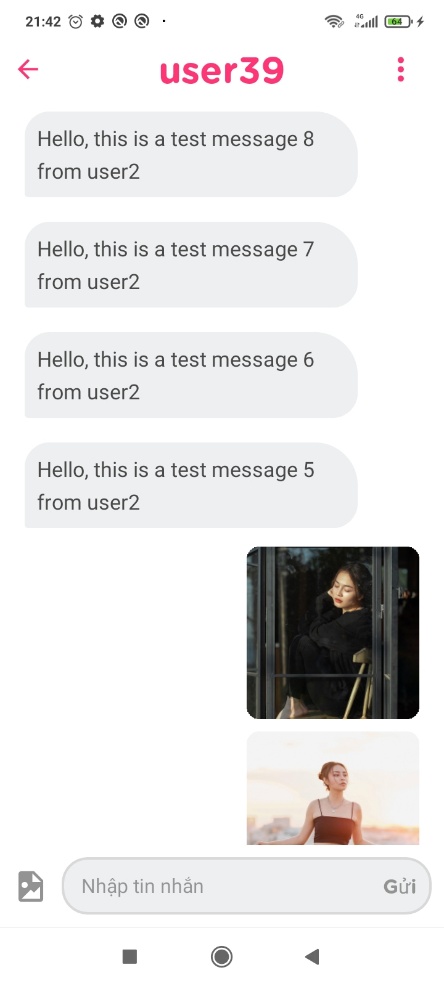
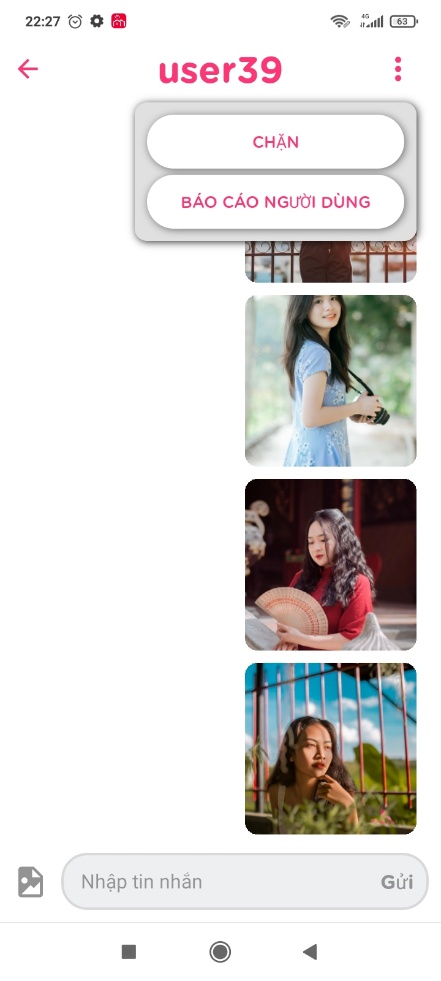
Hình 3‑2: Giao diện hiển thị hồ sơ cá nhân

* + - 1. Giao diện danh sách người dùng đã ghép đôi



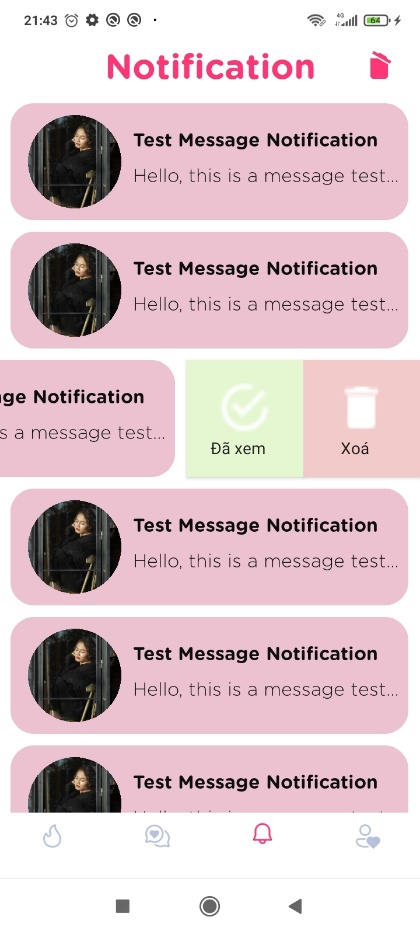
Hình 3‑3: Giao diện danh sách người dùng đã ghép đôi

* + - 1. Giao diện nhắn tin

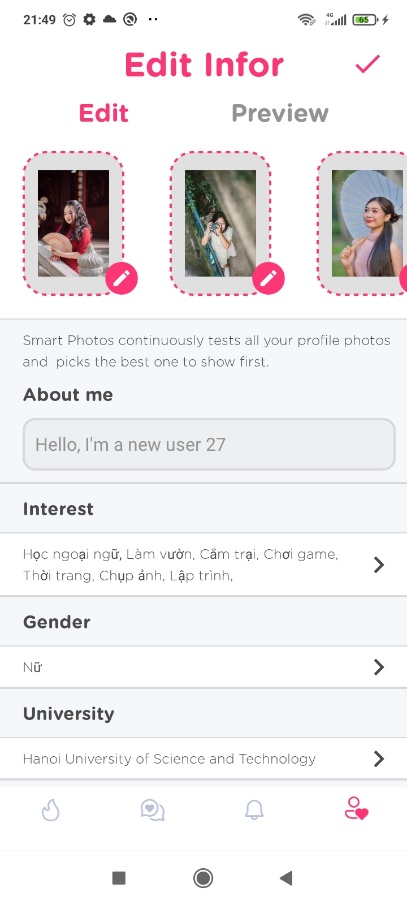
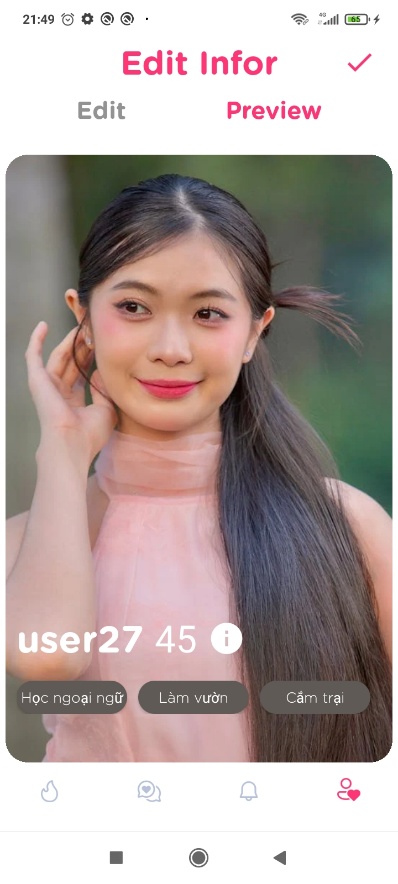
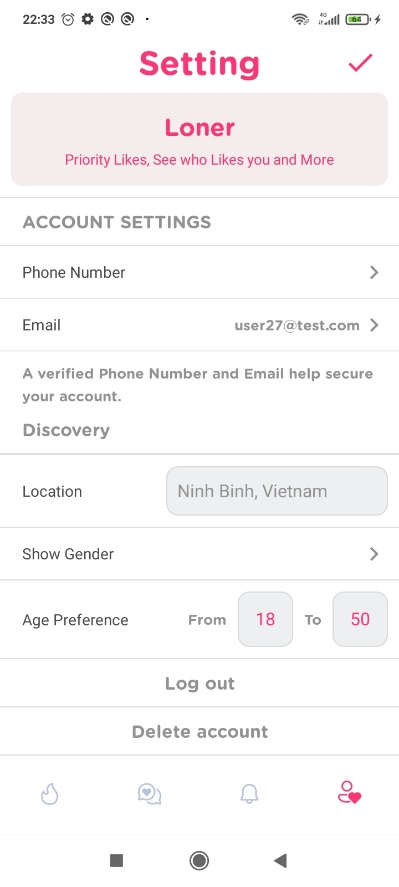
Hình 3-4: Giao diện nhắn tin

* + - 1. Giao diện thông báo

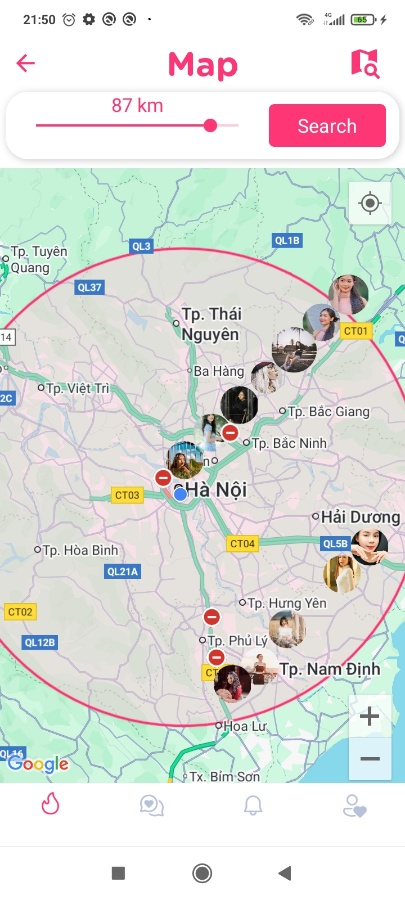
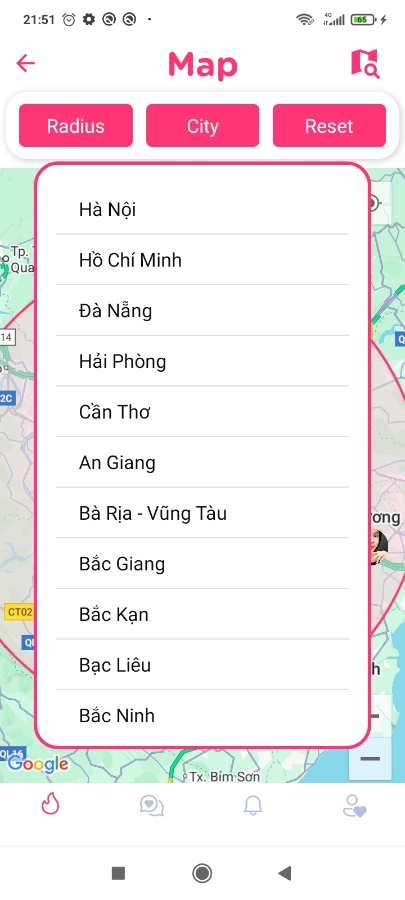
Hình 3‑5: Giao diện thông báo

* + - 1. Giao quản lý thông tin cá nhân

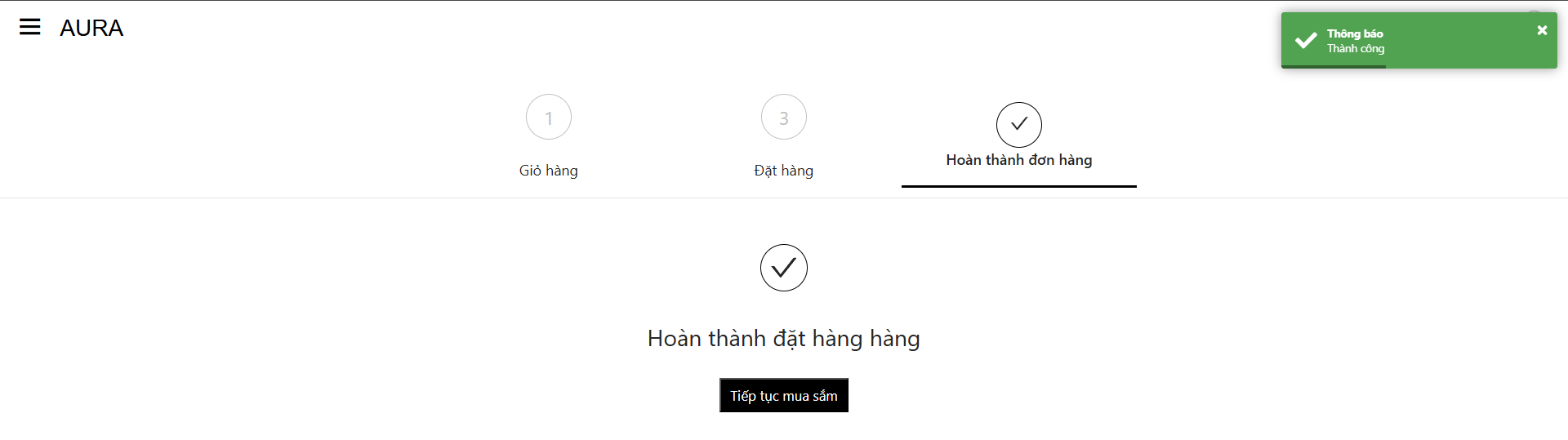
Hình 3‑6: Giao diện quản lý thông tin cá nhân

* + - 1. Giao diện Tìm kiếm người dùng

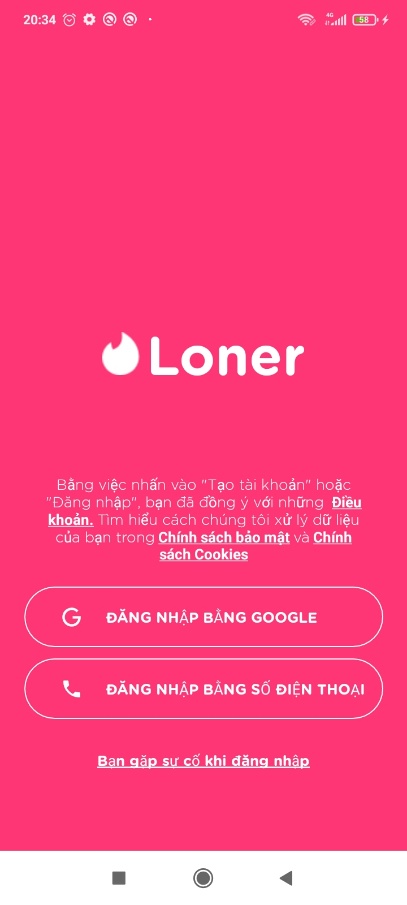
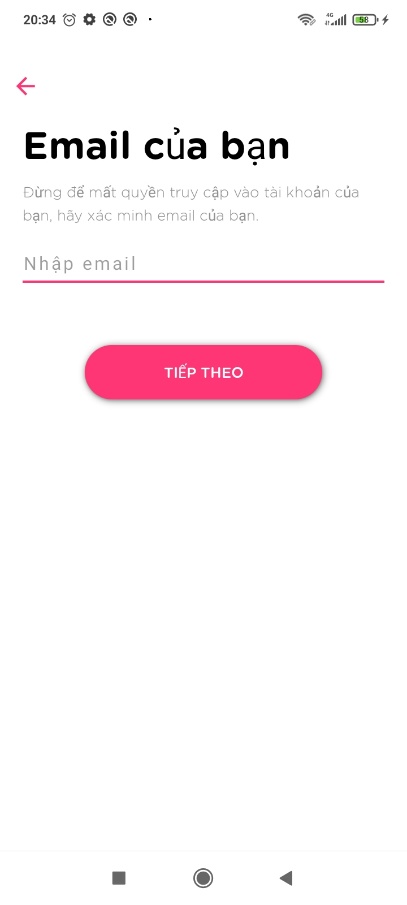
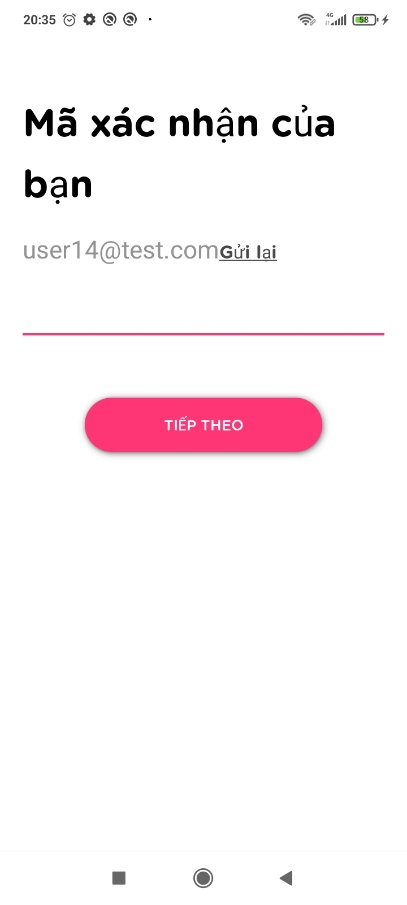
Hình 3‑7: Giao diện tìm kiếm người dùng

* + - 1. Giao diện nhắn tin chat bot



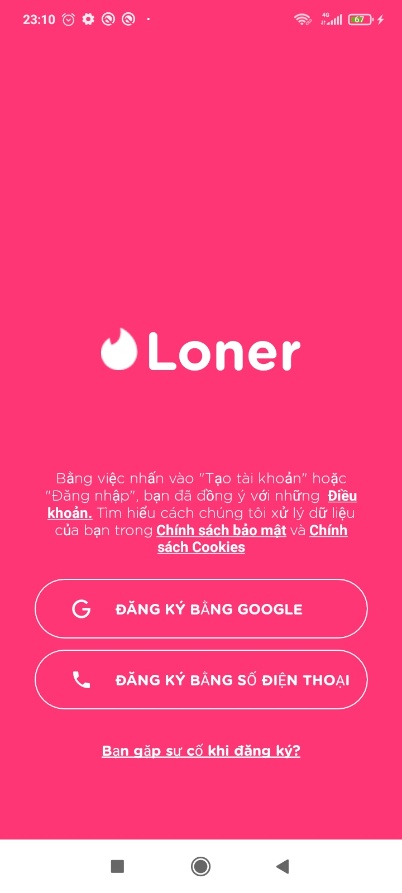
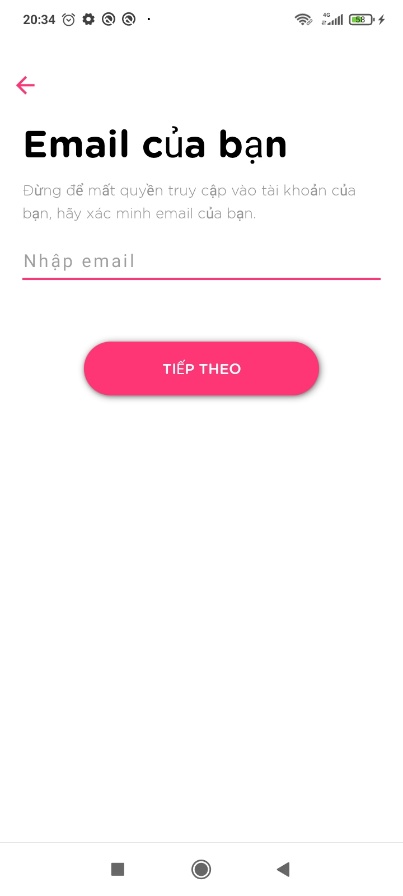
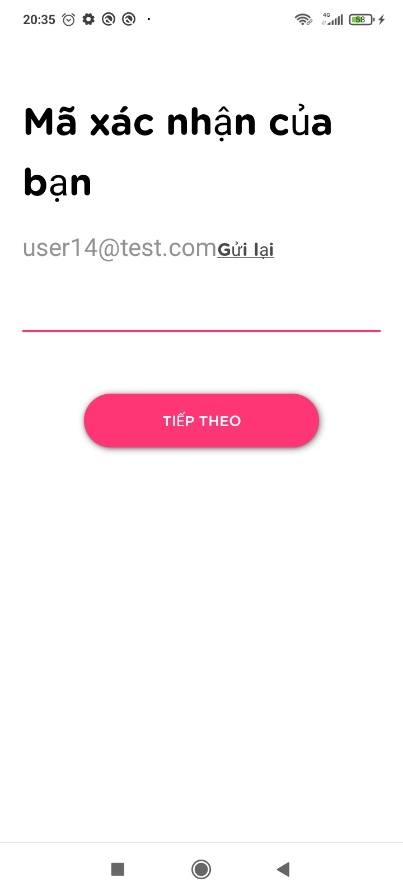
Hình 3‑8: Giao diện nhắn tin chat bot

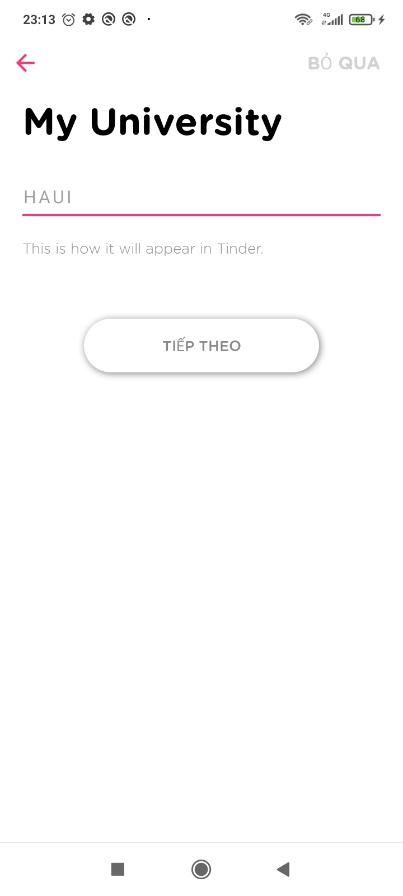
* + - 1. Giao diện đăng nhập

Hình 3‑9: Giao diện đăng nhập

* + - 1. Giao diện đăng ký

Hình 3‑10: Giao diện đăng ký

* + - 1. Giao diện báo cáo người dùng

Hình 3‑11: Giao diện báo cáo người dùng

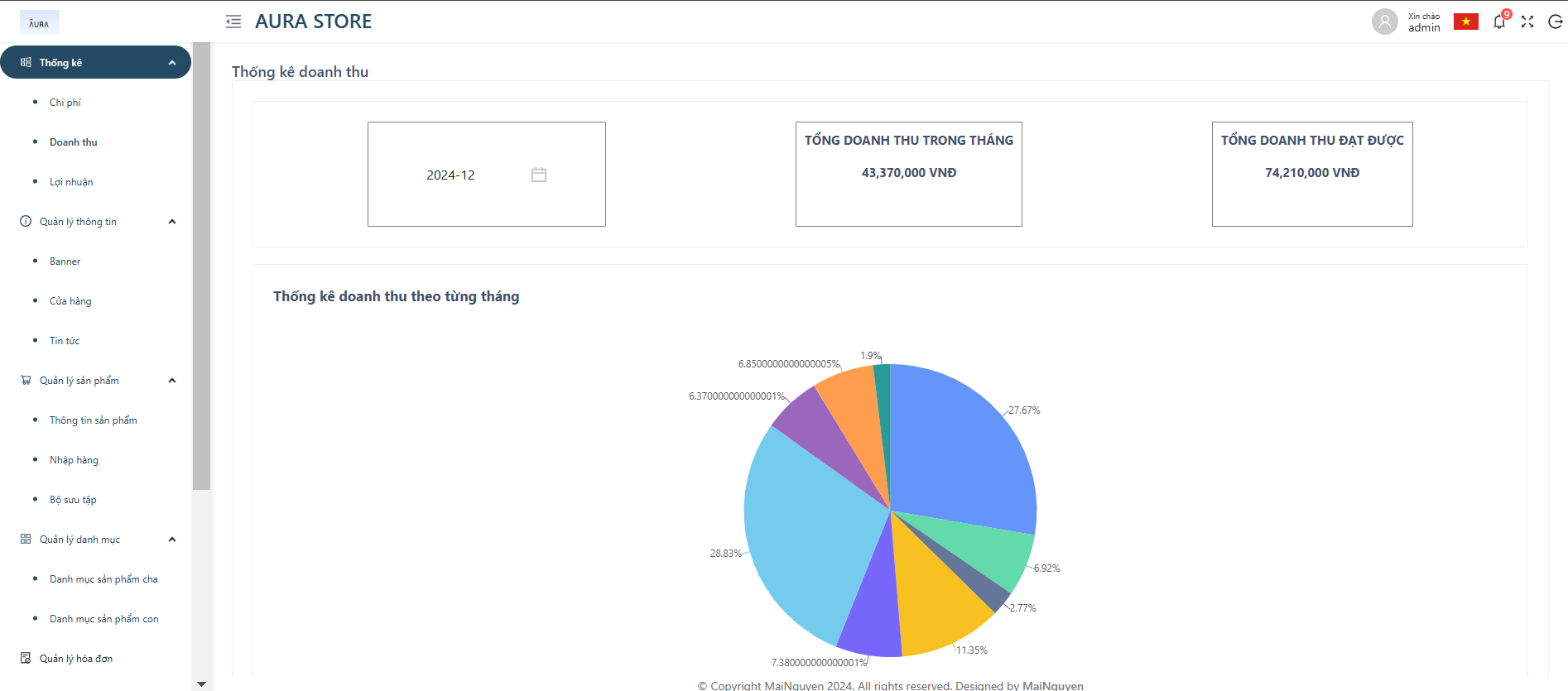
* + 1. Trang màn hình giao diện của người quản trị
       1. Giao diện quản lý người dùng

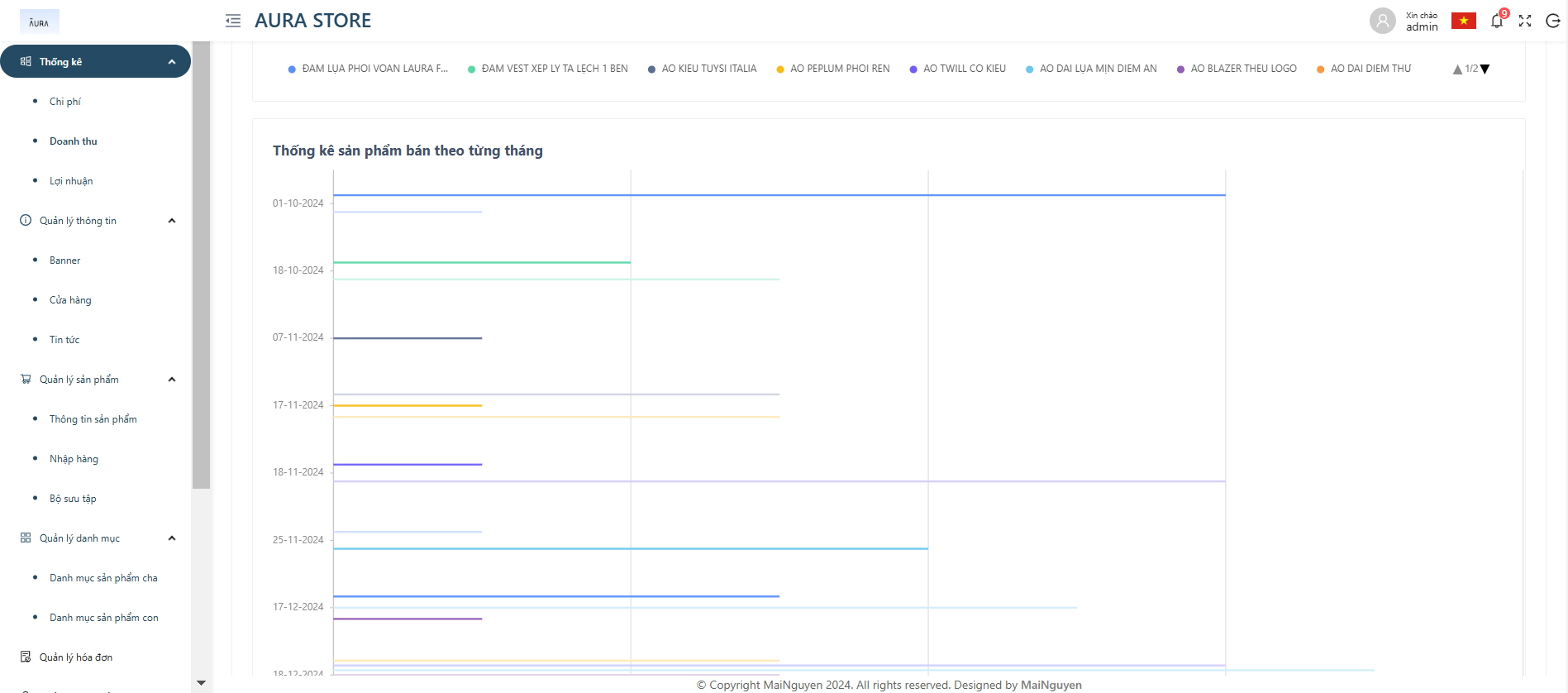


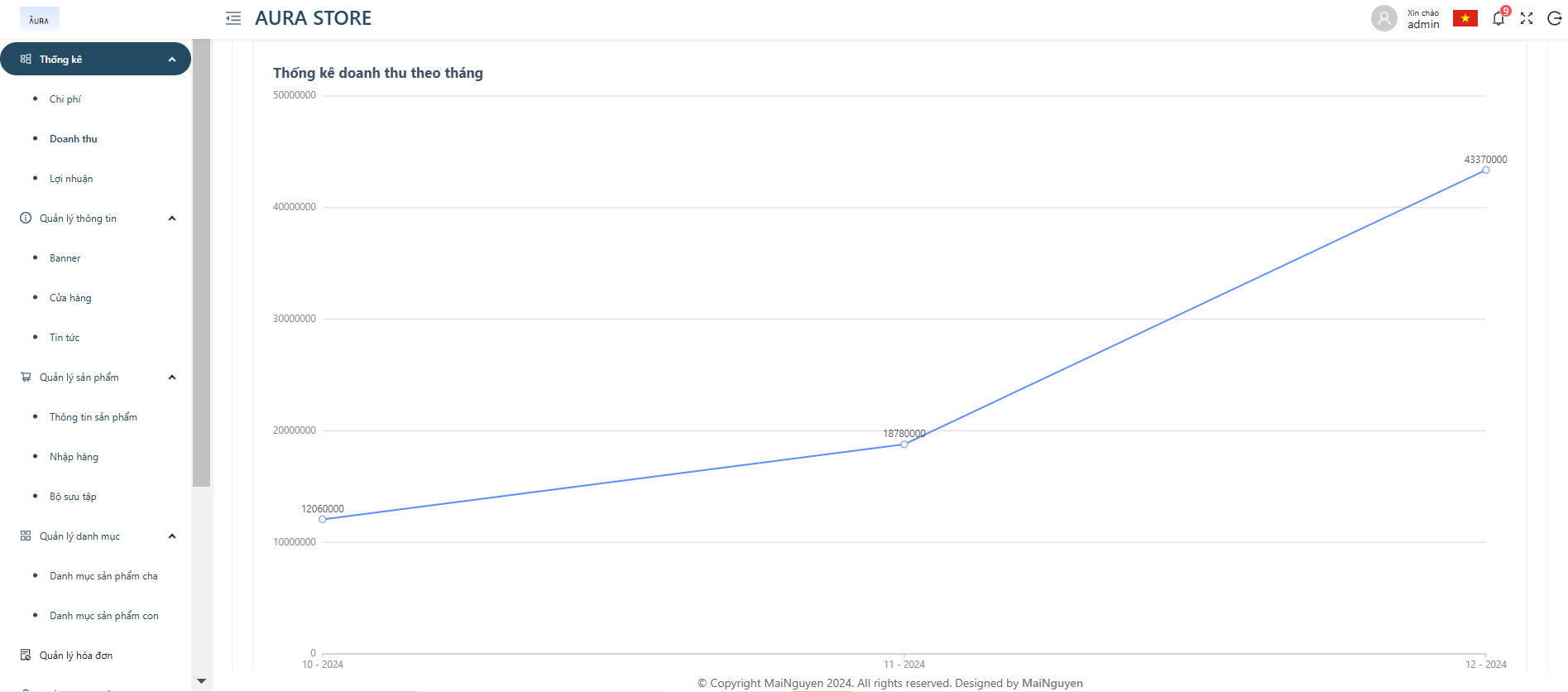


Hình 3-12: Giao diện quản lý người dùng

* + - 1. Giao diện quản lý báo cáo







Hình 3‑13: Giao diện quản lý báo cáo

* 1. Kiểm thử ứng dụng

3. 3. 1. Mục đích

* Đảm bảo chất lượng phần mềm: Xác định và khắc phục các lỗi trong quá trình phát triển để ứng dụng hoạt động ổn định, chính xác.
* Liệt kê những yêu cầu cho việc kiểm thử (Test Requirements).
* Xác minh tính năng**:** Đảm bảo rằng tất cả các tính năng được phát triển đều hoạt động đúng như yêu cầu đề ra.
* Ước lượng những yêu cầu về tài nguyên và chi phí cho việc kiểm thử.
* Những tài liệu được lập sau khi hoàn thành việc kiểm thử.
  + 1. Định nghĩa và từ viết tắt

Bảng 3-1: Bảng định nghĩa và viết tắt

|  |  |
| --- | --- |
| **Thuật ngữ** | **Định nghĩa** |
| URL | Cách gọi chung địa chỉ trang Web |
| Test Designer/ Tester | Người thiết kế kiểm thử/ Kiểm thử viên |

|  |  |
| --- | --- |
| **Từ viết tắt** | **Mô tả** |
| UC | Use case |
| TC | Test case |
| GUI | Giao diện người dùng |
| ST | System test |
| UAT | User acceptance test |
| TR | Test report |

* + 1. Lịch trình kiểm thử

Bảng 3-2: Bảng lịch trình kiểm thử

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mốc công việc** | **Sản phẩm** | **Thời gian** | **Bắt đầu** | **Kết thúc** |
| Lập kế hoạch kiểm thử | Test plan | 3h | 16/05/2025 | 18/05/2025 |
| Xem lại các tài liệu | Test plan | 3h | 16/05/2025 | 18/05/2025 |
| Thiết kế các testcase | Test case | 4h | 16/05/2025 | 18/05/2025 |
| Viết các testcase | Test case | 4h | 16/05/2025 | 18/05/2025 |
| Xem lại các testcase | Test case | 2h | 19/05/2025 | 20/05/2025 |
| Thực thi các testcase | Test case | 5h | 19/05/2025 | 20/05/2025 |
| Ghi nhận và đánh giá kết quả kiểm thử | Test report | 3h | 19/05/2025 | 20/05/2025 |

* + 1. Phạm vi kiểm thử

Các chức năng được kiểm thử:

- Đăng nhập: Kiểm tra chức năng đăng nhập vào hệ thống với tài khoản

vừa tạo.

- Xem danh sách đã ghép đôi: Kiểm tra chức năng xem danh sách người dùng đã được ghép đôi.

* + 1. Tài nguyên kiểm thử

1. Phần cứng

Điện thoại cá nhân có kết nối mạng Internet

Bảng 3-3: Bảng tài nguyên phần cứng

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU** | **RAM** | **ROM** | **Architecture** |
| MediaTek Helio G35, 2.3 GHz | 4 GB | 128 GB | ARM Cortex-A53 |

1. Phần mềm

Bảng 3-4: Bảng tài nguyên phần mềm

|  |  |  |
| --- | --- | --- |
| **Tên phần mềm** | **Phiên bản** | **Loại** |
| Android Studio | 19.0.1084.56 | Ứng dụng desktop |
| Microsoft Windows 11 | 11 | Hệ điều hành |

1. Phần mềm

Bảng 3-5: Bảng công cụ kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| **Hoạt động** | **Công cụ** | **Nhà cung cấp** | **Phiên bản** |
| Quản lý Test Case | Microsoft Office Excel | Microsoft | 2021 |
| Quản lý phiên bản | Microsoft Office Word | Microsoft | 2021 |
| Theo dõi lỗi | Microsoft Office Excel | Microsoft | 2021 |

1. Nhân sự

|  |  |
| --- | --- |
| Thành viên | Vai trò |
| Đinh Khắc Diên | Test Manager / Test Designer / Tester |

* + 1. Chiến lược kiểm thử

1. Các giai đoạn kiểm thử

Kiểm thử ở mức hệ thống và kiểm thử chấp nhận:

* Dùng kiểu kiểm thử tự động để kiểm thử từng chức năng.
* Việc kiểm thử chỉ bắt đầu khi đã hoàn thiện bộ test case để kiểm thử chức năng.
* Thiết kế test case theo phương pháp phân vùng tương đương.
* Chỉ thực hiện kiểm thử hồi quy, không thực hiện kiểm thử lại.

1. Loại kiểm thử

Bảng 3-6: Loại kiểm thử

|  |  |
| --- | --- |
| Mục đích kiểm tra | Đảm bảo các chức năng được kiểm tra hoạt động chính xác theo đặc tả yêu cầu |
| Kỹ thuật | Thực thi tất cả các trường hợp có thể có cho mỗi nhóm chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để xác định:   * Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng * Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng |
| Tiêu chuẩn dừng | Tất cả các test case đã được thiết kế đều được thực thi. |
| Chịu trách nhiệm kiểm thử | Test Designer / Tester |
| Cách kiểm thử | Kiểm thử bằng tay thủ công, tuần tự theo các bước được định nghĩa trong test case |
| Xử lý ngoại lệ | Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử. |

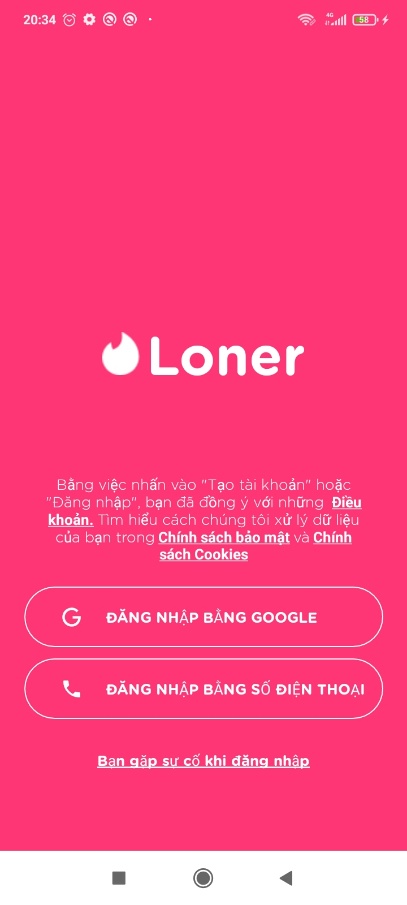
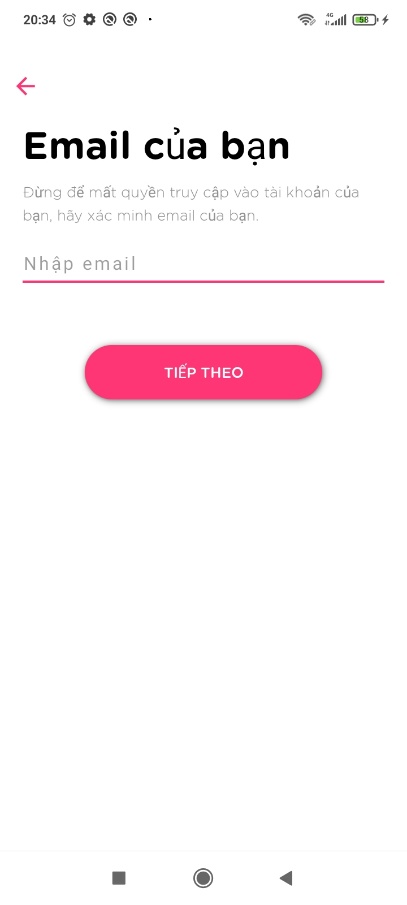
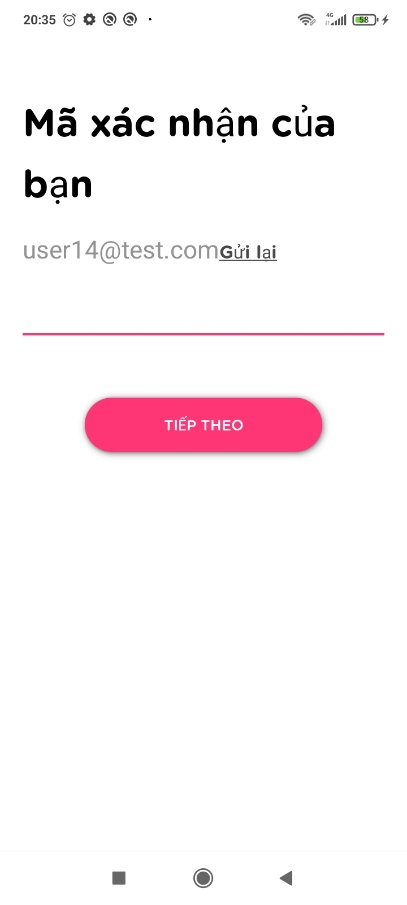
* + 1. Điều kiện chấp nhận kiểm thử
* Tỉ lệ test case đạt (passed): 100%
* Tỉ lệ test case không đạt (failed): 0%
* Hệ thống vẫn chạy ổn định trên các thiết bị Android khác nhau (Vivo, Redmi)

Bảng 3-7: Bảng phân loại lỗi

|  |  |
| --- | --- |
| **Mức độ nghiêm trọng** | **Đặc tả lỗi** |
| High | * Không thể đăng nhập * Không thể hiển thị danh sách ghép đôi |
| Medium | * Đăng nhập sai không báo lỗi, không trả về kết quả * Hiển thị thiếu thông tin danh sách ghép đôi |
| Low | * Đăng nhập sai báo lỗi, không trả về kết quả |

* + 1. Thực hiện kiểm thử

1. Kiểm thử chức năng đăng nhập
2. Giao diện chức năng:

Hình 3‑14: Giao diện đăng nhập

1. Mô tả giao diện và các điều kiện ràng buộc

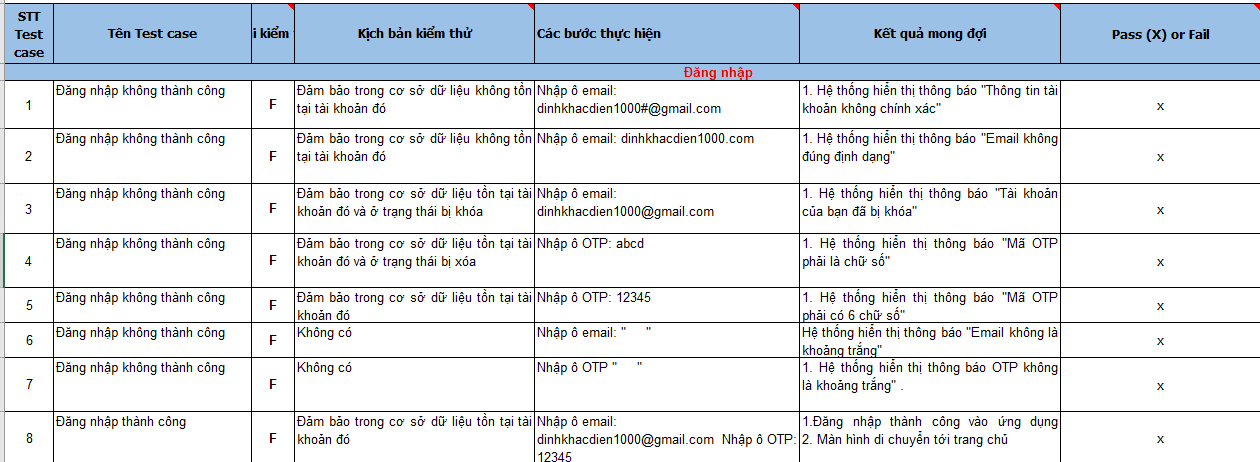
* Tiến hành nhập email và nhấn “Tiếp theo”.
* Tiến hành nhập mã OTP và nhấn “Tiếp theo”.
* Điều kiện ràng buộc: Không được để trống email hoặc mã OTP.

1. Sử dụng phương pháp phân hoạch tương đương thiết lập các ca kiểm thử:

|  |  |  |
| --- | --- | --- |
| Điều kiện đầu vào | Vùng hợp lệ | Vùng không hợp lệ |
| Email | Bất kỳ ký tự nào | Không có ký tự nào |
| OTP | Chỉ là số | Ký tự chữ cái |
| Nhập không đúng mã OTP gửi về |

1. Danh sách các test case

Bảng 3-8: Bảng test case Đăng nhập

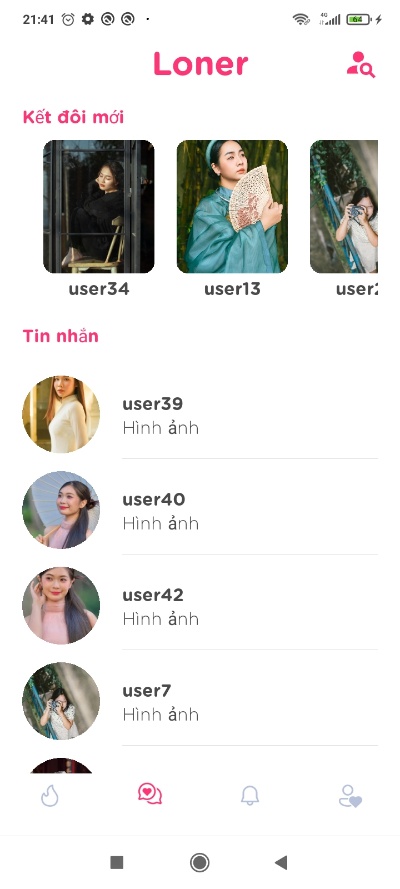
****

1. Thực hiện kiểm thử tự động bằng Appium

Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Appium và NUnit. Ở đây ta có kịch bản kiểm thử với các test case được test tự động.

2) Kiểm thử chức năng hiển thị danh sách ghép đôi

a) Giao diện chức năng



Hình 3‑15: Giao diện danh sách ghép đôi

b) Mô tả giao diện và các điều kiện ràng buộc

* Tiến hành đăng nhập vào tài khoản sau đó bấm vào icon tin nhắn trên thanh tab bar.
* Điều kiện ràng buộc: Phải đăng nhập tài khoản.

1. Sử dụng phương pháp phân hoạch tương đương thiết lập các ca kiêm thử

|  |  |  |
| --- | --- | --- |
| Điều kiện đầu vào | Vùng hợp lệ | Vùng không hợp lệ |
| Tài khoản người dùng | Người dùng có 1 hoặc nhiều ghép đôi | Người dùng không có ghép đôi nào |
| Kết nối Internet | Có kết nối Internet | Không có kết nối Internet |

1. Danh sách các test case

Bảng 3-9: Bảng test case Hiển thị danh sách ghép đôi



1. Thực hiện kiểm thử tự động bằng Appium

Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Appium và NUnit. Ở đây ta có kịch bản kiểm thử với các test case được test tự động.

* + 1. Kết quả kiểm thử

Quá trình kiểm thử không phát hiện lỗi trên ứng dụng ở các tính năng có khả năng bị ảnh hưởng sau khi thêm các tính năng mới, tỷ lệ pass 100%. Các test case đều chạy ổn định trên công cụ Selenium.

Dựa trên kết quả kiểm thử của các chức năng cần thiết và các chức năng mới trên ứng dụng, ta có một vài lời kết luận:

Đánh giá về tính ổn định: Ứng dụng có cho thấy tính ổn định tốt trong quá trình kiểm thử hồi quy. Không có các lỗi nghiêm trọng được phát hiện, như các lỗi giao diện người dùng cơ bản hay lỗi hoạt động của các tính năng chính.

Độ phủ kiểm thử: Tỷ lệ phủ của các ca kiểm thử đối với các chức năng cần thiết và các chức năng mới là đủ để đảm bảo chất lượng. Các kịch bản kiểm thử đã bao quát được các trường hợp thông thường cũng như trường hợp biên của từng tính năng.

Hiệu suất và thời gian phản hồi: Ứng dụng đáp ứng nhanh và không có vấn đề đáng kể về hiệu suất trong quá trình kiểm thử. Thời gian phản hồi của các tương tác người dùng là hợp lý và không gây ngạt ngào

* 1. Kết luận chương 3

Trong chương 3, em đã trình bày về kết quả đạt được sau khi hoàn thành dự án bao gồm: giao diện người dùng, giao diện người quản trị và phần kiểm thử hệ thống.

# KẾT LUẬN

1. **Kết quả đạt được**

* Qua quá trình thực hiện đồ án xây dựng ứng dụng hẹn hò Loner trên nền tảng Android, em đã phần nào hiểu được quy trình và nghiệp vụ, cách thức xây dựng một ứng dụng trên Android hoàn chỉnh. Đồng thời nó cũng giúp em có thêm được cho mình nhiều kinh nghiệm và kiến thức quý báu.

Có thể kể đến một số ưu điểm của ứng dụng:

* Ứng dụng hoạt động mượt mà với các chức năng cơ bản như quẹt, tìm kiếm người dùng, nhắn tin, thông báo, … trên nền tảng Android.
* Có giao diện thiết kế gọn gàng, dễ sử dụng với animation đẹp mắt phù hợp với nhiều độ tuổi người dùng.
* Người dùng có thể tìm kiếm được thông tin người dùng khác trong một phạm vi bán kính một cách dễ dàng.
* Ứng dụng có hiệu suất ổn định và mượt mà trên nhiều thiết bị Android khác nhau, mang lại trải nghiệm tốt hơn cho người dùng.

1. **Hạn chế**

* Chưa tối ưu được toàn bộ chức năng, một số lỗi vẫn có thể xảy ra.
* Chưa triển khai được lên Google Play.

1. **Hướng phát triển**

* Trong thời gian tới, em sẽ cố gắng khắc phục các hạn chế còn tồn đọng, những vấn đề chưa được giải quyết tốt, và bổ sung thêm một số chức năng mới cho hệ thống như: Hỗ trợ theo dõi vị trí theo thời gian thực, chức năng thanh toán để nâng cấp tài khoản sử dụng nhiều chức năng hơn, sử dụng các mô hình AI để gợi ý kết nối, ứng dụng giao thức MCP cho phép AI đọc tin nhắn người dùng để gợi ý thông minh hơn, …

# TÀI LIỆU THAM KHẢO

1. Đỗ Ngọc Sơn, Phan Văn Viên, Nguyễn Phương Nga - *Giáo trình Hệ quản trị cơ sở dữ liệu* - Trường Đại học Công nghiệp Hà Nội - NXB Khoa học và Kỹ thuật.
2. Nguyễn Trung Phú, Trần Phương Nhung, Đỗ Thị Minh Nguyệt - *Giáo trình Thiết kế Web* - Trường Đại học Công nghiệp hà Nội – NXB Thống kê.
3. Nguyễn Thị Thanh Huyền (2011) - *Giáo trình phân tích thiết kế hệ thống –* Trường Đại học Công nghiệp Hà Nội - NXB Giáo dục Việt Nam.
4. Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý - *Giáo trình Nhập môn công nghệ phần mềm* - NXB Đại học Công nghiệp Hà Nội, 2016.