**\*\*Lời nói đầu\*\* ..................................................................................................................... [...]**

**\*\*CHƯƠNG 1: GIỚI THIỆU\*\* .......................................................................................... [...]**

**1.1. Tổng quan về bài toán tìm kiếm sản phẩm bằng hình ảnh ............................. [...]**

**1.2. Bối cảnh và sự cần thiết của hệ thống ......................................................... [...]**

**1.3. Cách tiếp cận, giải quyết và đối tượng, phạm vi nghiên cứu ......................... [...]**

**1.3.1. Cách tiếp cận và giải pháp công nghệ ................................................... [...]**

**1.3.2. Đối tượng và phạm vi của hệ thống ..................................................... [...]**

**1.3.3. Mục tiêu và nhiệm vụ của đề án .......................................................... [...]**

**1.3.4. Bố cục của báo cáo ............................................................................... [...]**

**1.3.5. Mô tả bài toán ...................................................................................... [...]**

**\*\*CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ NỀN TẢNG CÔNG NGHỆ\*\* ...................... [...]**

**2.1. Frontend .............................................................................................................. [...]**

**2.1.1. ReactJS và các thư viện liên quan (Bootstrap, MUI...) .................. [...]**

**2.1.2. Quản lý trạng thái và Routing ............................................................ [...]**

**2.2. Backend ............................................................................................................. [...]**

**2.2.1. FastAPI và Python .............................................................................. [...]**

**2.2.2. Xác thực và Phân quyền (JWT) ....................................................... [...]**

**2.3. Xử lý và Tìm kiếm Hình ảnh ....................................................................... [...]**

**2.3.1. Trích xuất đặc trưng ảnh (EfficientNet/TensorFlow) ....................... [...]**

**2.3.2. Băm ảnh (Image Hashing) ............................................................... [...]**

**2.3.3. Tìm kiếm Vector tương đồng (Faiss) ............................................... [...]**

**2.4. Cơ sở dữ liệu .................................................................................................. [...]**

**2.4.1. MongoDB ........................................................................................... [...]**

**2.5. Dịch vụ bên ngoài ........................................................................................ [...]**

**2.5.1. Cloudinary (Lưu trữ ảnh) ................................................................ [...]**

**2.5.2. Nhanh.vn (Tích hợp API) ................................................................ [...]**

**\*\*CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG\*\* .............................................. [...]**

**3.1. Phân tích chức năng hệ thống (Use Cases) ................................................... [...]**

**3.1.1. Sơ đồ UseCase tổng quát .................................................................... [...]**

**3.1.2. Sơ đồ UseCase Đăng nhập / Đăng ký ................................................ [...]**

**3.1.3. Sơ đồ UseCase Tìm kiếm ảnh ........................................................... [...]**

**3.1.4. Sơ đồ UseCase Quản lý sản phẩm (CRUD) ..................................... [...]**

**3.1.5. Sơ đồ UseCase Quản lý người dùng (Admin) ................................... [...]**

**3.1.6. Sơ đồ UseCase Quản lý cấu hình (Admin) ....................................... [...]**

**3.1.7. Sơ đồ UseCase Tích hợp Nhanh.vn (Ví dụ: Tạo đơn hàng) .............. [...]**

**3.2. Phân tích cấu trúc hệ thống ........................................................................ [...]**

**\*(Có thể là Biểu đồ lớp hoặc mô tả các thành phần chính)\***

**3.3. Phân tích hành vi hệ thống (Sequence Diagrams) ....................................... [...]**

**3.3.1. Luồng Đăng nhập .............................................................................. [...]**

**3.3.2. Luồng Tìm kiếm ảnh ......................................................................... [...]**

**3.3.3. Luồng Thêm sản phẩm (bao gồm xử lý ảnh) ................................... [...]**

**3.3.4. Luồng Cập nhật/Xóa sản phẩm ....................................................... [...]**

**3.3.5. Luồng Tương tác với Nhanh.vn API ............................................... [...]**

**3.4. Thiết kế kiến trúc hệ thống ........................................................................ [...]**

**3.4.1. Biểu đồ gói (Package Diagram) ........................................................ [...]**

**3.4.2. Biểu đồ thành phần (Component Diagram) ...................................... [...]**

**3.4.3. Biểu đồ triển khai (Deployment Diagram) ........................................ [...]**

**3.5. Thiết kế giao diện hệ thống ........................................................................ [...]**

**3.5.1. Giao diện Đăng nhập / Đăng ký ...................................................... [...]**

**3.5.2. Giao diện Tìm kiếm ảnh và Kết quả ................................................. [...]**

**3.5.3. Giao diện Quản lý Sản phẩm ........................................................... [...]**

**3.5.4. Giao diện Quản lý Người dùng ........................................................ [...]**

**3.5.5. Giao diện Quản lý Cấu hình ............................................................ [...]**

**3.5.6. Giao diện Hồ sơ người dùng / Đổi mật khẩu ................................... [...]**

**3.5.7. Giao diện Tạo đơn hàng (Tích hợp Nhanh.vn) ............................... [...]**

**3.6. Thiết kế Cơ sở dữ liệu ............................................................................... [...]**

**3.6.1. Sơ đồ quan hệ thực thể (ERD - nếu cần) ....................................... [...]**

**3.6.2. Thiết kế Schema cho các Collection (Users, Companies, Products, Images, AppConfigs) [...]**

**\*\*CHƯƠNG 4: TRIỂN KHAI THỬ NGHIỆM VÀ ĐÁNH GIÁ\*\* ............................... [...]**

**4.1. Môi trường triển khai ............................................................................... [...]**

**\*(Server, Database, Cloud Services...)\***

**4.2. Quy trình triển khai .................................................................................. [...]**

**4.3. Kết quả thử nghiệm và Đánh giá .............................................................. [...]**

**4.3.1. Đánh giá chức năng .......................................................................... [...]**

**4.3.2. Đánh giá hiệu năng (Tốc độ tìm kiếm, xử lý ảnh...) ....................... [...]**

**4.3.3. Đánh giá độ chính xác của tìm kiếm ảnh ........................................ [...]**

**4.3.4. So sánh với mục tiêu đề ra ............................................................... [...]**

**4.3.5. Ưu điểm và Hạn chế của hệ thống ................................................... [...]**

**4.4. Các mục tiêu phát triển trong tương lai ...................................................... [...]**

**\*\*Kết luận\*\* ........................................................................................................................ [...]**

**\*\*(Optional) Phụ lục\*\* ...................................................................................................... [...]**

**\*(Ví dụ: Tài liệu API, Cấu hình chi tiết...)\***

**\*\*Tài liệu tham khảo\*\* .................................................................................................... [...]**

**Lời nói đầu**

Trong bối cảnh thương mại điện tử và bán hàng đa kênh phát triển mạnh mẽ, hình ảnh sản phẩm đóng vai trò ngày càng quan trọng trong việc thu hút khách hàng, quảng bá thương hiệu và quản lý hàng hóa hiệu quả. Tuy nhiên, việc tìm kiếm và quản lý sản phẩm dựa trên hình ảnh, đặc biệt khi số lượng mặt hàng lớn, thường gặp nhiều khó khăn với các phương pháp truyền thống như tìm kiếm theo tên hay mã sản phẩm. Nhu cầu về một công cụ tìm kiếm thông minh, có khả năng nhận diện và so sánh sản phẩm trực tiếp từ hình ảnh trở nên cấp thiết hơn bao giờ hết.

Các phương pháp tìm kiếm hiện tại thường tỏ ra kém hiệu quả khi người dùng chỉ có hình ảnh trực quan hoặc muốn tìm các sản phẩm tương tự về kiểu dáng, màu sắc. Việc quản lý kho hàng, đối chiếu sản phẩm, và phát hiện trùng lặp dựa trên hình ảnh cũng là một thách thức không nhỏ đối với nhiều doanh nghiệp. Thêm vào đó, việc tích hợp khả năng tìm kiếm bằng hình ảnh vào các quy trình bán hàng và quản lý kho hiện có để tối ưu hóa hoạt động là một nhu cầu thực tế nhưng chưa có nhiều giải pháp đáp ứng tốt.

Chính vì vậy, đề án **"Xây dựng hệ thống tìm kiếm sản phẩm bằng hình ảnh"** (Search Images API) được thực hiện nhằm giải quyết những thách thức trên. Dự án tập trung vào việc phát triển một hệ thống ứng dụng công nghệ trí tuệ nhân tạo (AI) và học máy (Machine Learning), cụ thể là các kỹ thuật xử lý và nhận dạng hình ảnh, để cho phép người dùng tìm kiếm sản phẩm một cách nhanh chóng và chính xác chỉ thông qua việc tải lên hoặc cung cấp đường dẫn hình ảnh. Hệ thống được thiết kế để phục vụ cho nhiều công ty, với khả năng quản lý dữ liệu sản phẩm và người dùng riêng biệt.

Đề án này không chỉ hướng tới việc cung cấp một công cụ tìm kiếm hình ảnh mạnh mẽ mà còn nhằm mục tiêu hỗ trợ các doanh nghiệp quản lý sản phẩm trực quan và hiệu quả hơn, giảm thiểu sai sót và trùng lặp thông qua khả năng nhận diện hình ảnh. Hơn nữa, hệ thống được xây dựng với định hướng tích hợp vào các quy trình nghiệp vụ sẵn có, kết nối với các nền tảng quản lý bán hàng như Nhanh.vn, từ đó nâng cao trải nghiệm cho nhân viên và tối ưu hóa hoạt động kinh doanh trong kỷ nguyên số. Hy vọng rằng, dự án sẽ tạo ra một giải pháp công nghệ tiên tiến, đáp ứng hiệu quả nhu cầu thực tiễn của các doanh nghiệp hiện nay.

**CHƯƠNG 1: GIỚI THIỆU**

**1.1. Tổng quan về bài toán tìm kiếm sản phẩm bằng hình ảnh**

Trong kỷ nguyên số hóa, khối lượng hình ảnh sản phẩm được tạo ra và lưu trữ trên các nền tảng trực tuyến, hệ thống quản lý kho, và các kênh bán hàng ngày càng tăng theo cấp số nhân. Hình ảnh không chỉ là phương tiện trực quan để giới thiệu sản phẩm mà còn chứa đựng nhiều thông tin quan trọng về đặc điểm, kiểu dáng, màu sắc, và chất liệu. Tuy nhiên, việc khai thác hiệu quả nguồn dữ liệu hình ảnh khổng lồ này đặt ra một thách thức lớn, đặc biệt là trong việc tìm kiếm và định danh sản phẩm.Các phương pháp tìm kiếm sản phẩm truyền thống chủ yếu dựa trên văn bản (text-based search), như tìm kiếm theo tên sản phẩm, mã SKU, mô tả, hoặc từ khóa. Mặc dù hữu ích, các phương pháp này bộc lộ nhiều hạn chế:

* **Khi người dùng không biết tên hoặc thông tin mô tả chính xác:** Người dùng có thể nhìn thấy một sản phẩm trong thực tế hoặc qua một hình ảnh nhưng không biết tên gọi hay mã hiệu của nó.
* **Mô tả sản phẩm không đầy đủ hoặc không nhất quán:** Dữ liệu văn bản đi kèm sản phẩm có thể thiếu sót, sai lệch hoặc không đồng bộ giữa các kênh bán hàng.
* **Nhu cầu tìm kiếm sản phẩm tương tự về mặt hình ảnh:** Người dùng hoặc nhân viên quản lý thường có nhu cầu tìm các sản phẩm có kiểu dáng, màu sắc, hoặc hoa văn tương đồng với một sản phẩm mẫu, điều mà tìm kiếm văn bản khó đáp ứng hiệu quả.
* **Quản lý trùng lặp và đối soát kho hàng:** Việc xác định các sản phẩm trùng lặp hoặc đối soát hàng hóa trong kho dựa trên hình ảnh là một nhu cầu thực tế nhưng khó thực hiện thủ công.

Bài toán **tìm kiếm sản phẩm bằng hình ảnh** (image-based product search hoặc visual search) ra đời nhằm giải quyết những hạn chế trên. Thay vì dựa vào từ khóa văn bản, hệ thống sẽ phân tích nội dung trực quan của một hình ảnh đầu vào (ảnh truy vấn) và tìm kiếm trong cơ sở dữ liệu các hình ảnh sản phẩm có đặc điểm thị giác tương đồng nhất.Để giải quyết bài toán này, các kỹ thuật tiên tiến trong lĩnh vực Trí tuệ Nhân tạo (AI) và Học máy (Machine Learning), đặc biệt là Thị giác Máy tính (Computer Vision) và Học sâu (Deep Learning), được ứng dụng để:

* **Trích xuất đặc trưng hình ảnh:** "Hiểu" được nội dung cốt lõi của hình ảnh và biểu diễn nó dưới dạng các vector số (feature vectors) mà máy tính có thể so sánh được.
* **Xây dựng chỉ mục tìm kiếm hiệu quả:** Tổ chức các vector đặc trưng của hàng triệu sản phẩm sao cho việc tìm kiếm các vector tương đồng diễn ra nhanh chóng.
* **Đo lường độ tương đồng:** Tính toán mức độ giống nhau giữa hình ảnh truy vấn và các hình ảnh trong cơ sở dữ liệu dựa trên các đặc trưng đã trích xuất.

Tìm kiếm sản phẩm bằng hình ảnh không chỉ nâng cao trải nghiệm mua sắm cho khách hàng mà còn mang lại lợi ích thiết thực cho doanh nghiệp trong việc quản lý sản phẩm, tối ưu hóa kho hàng, và phân tích thị trường. Đây là một lĩnh vực nghiên cứu và ứng dụng đang phát triển mạnh mẽ, hứa hẹn mang lại nhiều giá trị trong bối cảnh thương mại điện tử và quản lý dữ liệu ngày càng phức tạp.

**1.2. Bối** **cảnh và sự cần thiết của hệ thống**

Sự tăng trưởng mạnh mẽ của thương mại điện tử và bán hàng đa kênh đã tạo ra một lượng dữ liệu hình ảnh sản phẩm khổng lồ. Trong bối cảnh này, các doanh nghiệp, đặc biệt trong ngành bán lẻ và thời trang, đối mặt với những thách thức vận hành đáng kể khi chỉ dựa vào dữ liệu văn bản để quản lý kho hàng và tương tác với khách hàng. Việc kiểm kê, đối soát sản phẩm, và đáp ứng nhu cầu khám phá sản phẩm dựa trên cảm hứng thị giác trở nên kém hiệu quả, tốn kém thời gian và nguồn lực.Đồng thời, những đột phá trong công nghệ Trí tuệ Nhân tạo và Thị giác Máy tính đã cung cấp khả năng phân tích và "hiểu" nội dung hình ảnh một cách tự động và chính xác. Điều này mở ra cơ hội để giải quyết các bài toán quản lý và tìm kiếm mà trước đây rất khó thực hiện ở quy mô lớn.Vì vậy, việc xây dựng một **Hệ thống Tìm kiếm Sản phẩm bằng Hình ảnh** trở nên **cần thiết** để:

* **Tối ưu hóa quy trình nghiệp vụ:** Cải thiện hiệu quả quản lý kho, giảm thiểu sai sót trong đối soát và phát hiện trùng lặp sản phẩm bằng cách tận dụng thông tin trực quan.
* **Nâng cao khả năng khám phá và bán hàng:** Đáp ứng nhu cầu tìm kiếm sản phẩm dựa trên cảm hứng hình ảnh của khách hàng và hỗ trợ nhân viên tư vấn hiệu quả hơn.
* **Khai thác giá trị dữ liệu hình ảnh:** Biến kho hình ảnh khổng lồ thành một tài sản có thể truy vấn và khai thác, thay vì chỉ là dữ liệu lưu trữ tĩnh.

Hệ thống này không chỉ là một công cụ tìm kiếm nâng cao mà còn là một giải pháp cần thiết giúp doanh nghiệp thích ứng với môi trường kinh doanh hiện đại, tối ưu hóa hoạt động và nâng cao năng lực cạnh tranh bằng cách khai thác hiệu quả sức mạnh của dữ liệu hình ảnh.

**1.3. Cách tiếp cận, giải quyết và đối tượng, phạm vi nghiên cứu**

**1.3.1. Cách tiếp cận và giải pháp công nghệ**

Để giải quyết bài toán tìm kiếm sản phẩm bằng hình ảnh và các yêu cầu quản lý liên quan, đề án áp dụng cách tiếp cận xây dựng hệ thống theo kiến trúc Client-Server, ứng dụng các công nghệ hiện đại và phù hợp:

* **Kiến trúc:**
* **Backend:** Sử dụng Python với framework FastAPI để xây dựng các API mạnh mẽ, hiệu năng cao, hỗ trợ xử lý bất đồng bộ tốt, phù hợp cho các tác vụ xử lý ảnh và tìm kiếm.
* **Frontend:** Sử dụng thư viện ReactJS (JavaScript) để xây dựng giao diện người dùng tương tác, linh hoạt và dễ dàng bảo trì.
* **Cơ sở dữ liệu:** Lựa chọn MongoDB, một hệ quản trị cơ sở dữ liệu NoSQL hướng tài liệu, linh hoạt trong việc lưu trữ cấu trúc dữ liệu đa dạng của sản phẩm và đặc trưng ảnh.
* **Giải pháp công nghệ cốt lõi:**
* **Xử lý ảnh và Trích xuất đặc trưng:** Sử dụng thư viện TensorFlow/Keras với mô hình học sâu tiên tiến (như EfficientNet) để trích xuất các vector đặc trưng (feature vectors) đại diện cho nội dung thị giác của hình ảnh sản phẩm.
* **Tìm kiếm tương đồng:**
* Sử dụng thư viện Faiss để xây dựng chỉ mục (index) từ các vector đặc trưng, cho phép tìm kiếm các sản phẩm có hình ảnh tương đồng nhất một cách nhanh chóng và hiệu quả, ngay cả với số lượng lớn sản phẩm.
* Kết hợp với kỹ thuật băm ảnh (Image Hashing) bằng thư viện imagehash để lọc sơ bộ hoặc tăng cường độ chính xác.
* **Xác thực & Phân quyền:** Áp dụng cơ chế JSON Web Tokens (JWT) để bảo mật API và quản lý quyền truy cập của người dùng (Admin, User).
* **Dịch vụ bên ngoài:**
* **Lưu trữ ảnh:** Tích hợp với dịch vụ Cloudinary để lưu trữ, quản lý và tối ưu hóa việc phân phối hình ảnh sản phẩm.
* **Tích hợp nghiệp vụ:** Kết nối với API của Nhanh.vn để đồng bộ hóa dữ liệu, lấy thông tin cần thiết (địa chỉ, nhân viên) và hỗ trợ tạo đơn hàng, liên kết hệ thống tìm kiếm với quy trình bán hàng thực tế.

Cách tiếp cận này cho phép xây dựng một hệ thống vừa mạnh mẽ về khả năng tìm kiếm thông minh, vừa linh hoạt trong quản lý dữ liệu và có khả năng tích hợp với các nền tảng khác.

**1.3.2. Đối tượng và phạm** **vi của hệ thống**

**Đối tượng sử dụng:**

* **Công ty/Doanh nghiệp:** Các tổ chức cần một giải pháp để quản lý và tìm kiếm danh mục sản phẩm của mình thông qua hình ảnh. Hệ thống được thiết kế để hỗ trợ nhiều công ty độc lập (multi-tenant).
* **Người dùng cuối:** Bao gồm hai vai trò chính trong mỗi công ty:
  + - Admin: Quản trị viên của công ty, có quyền quản lý toàn bộ dữ liệu của công ty đó (sản phẩm, hình ảnh, người dùng thuộc công ty, cấu hình API Nhanh.vn).
    - User: Nhân viên thông thường, có quyền xem, tìm kiếm, thêm/sửa/xóa sản phẩm (theo phân quyền) và sử dụng các tính năng tích hợp (ví dụ: tạo đơn hàng qua Nhanh.vn)

**Phạm vi của hệ thống:**

* **Chức năng cốt lõi:**
  + - Tìm kiếm sản phẩm dựa trên hình ảnh tải lên hoặc URL.
      * Quản lý vòng đời sản phẩm (Thêm, Xem, Sửa, Xóa – CRUD).
      * Quản lý hình ảnh liên kết với sản phẩm.
      * Quản lý người dùng và phân quyền trong phạm vi công ty.
      * Quản lý thông tin công ty và cấu hình API tích hợp.
    - **Tích hợp:**
      * Lưu trữ và truy xuất ảnh từ Cloudinary.
      * Tương tác hai chiều với API Nhanh.vn (lấy dữ liệu danh mục, địa chỉ, nhân viên; đẩy thông tin đơn hàng).
    - **Giới hạn:**
      * Hệ thống tập trung vào tìm kiếm dựa trên sự tương đồng thị giác, không bao gồm phân tích ngữ nghĩa sâu hoặc các thuộc tính phức tạp khác nếu không được nhập liệu.
      * Không bao gồm các tính năng phân tích kinh doanh, báo cáo chi tiết (trừ các chức năng được hỗ trợ qua API Nhanh.vn).
      * Không trực tiếp xử lý các quy trình thanh toán.

**1.3.3. Mục tiêu và nhiệm vụ của đề án  
Mục tiêu chính:**

* Xây dựng thành công một hệ thống ứng dụng web cho phép tìm kiếm sản phẩm bằng hình ảnh một cách nhanh chóng, chính xác và hiệu quả.
* Cung cấp giao diện quản lý dữ liệu (sản phẩm, người dùng, cấu hình) trực quan và thân thiện cho các đối tượng người dùng khác nhau.
* Thiết kế hệ thống có khả năng hỗ trợ nhiều công ty độc lập trên cùng một nền tảng.
* Tích hợp liền mạch với các dịch vụ bên thứ ba là Cloudinary (lưu trữ ảnh) và Nhanh.vn (nghiệp vụ bán hàng).
* Đảm bảo hệ thống hoạt động ổn định, an toàn và có khả năng mở rộng trong tương lai.

**Nhiệm vụ cụ thể:**

* Khảo sát, phân tích yêu cầu chi tiết và thiết kế tổng thể hệ thống (kiến trúc, cơ sở dữ liệu, giao diện).
* Phát triển module Backend (API): Xây dựng các endpoint cho việc xác thực, quản lý CRUD dữ liệu, xử lý ảnh (trích xuất đặc trưng, hashing), thực hiện tìm kiếm (Faiss), và tích hợp API Nhanh.vn.
* Phát triển module Frontend: Xây dựng giao diện người dùng cho các chức năng đăng nhập, tìm kiếm, hiển thị kết quả, quản lý dữ liệu; thực hiện gọi API từ Backend.
* Thiết lập và cấu hình cơ sở dữ liệu MongoDB.
* Cấu hình và kiểm thử việc tích hợp với Cloudinary và Nhanh.vn.
* Thực hiện kiểm thử các chức năng (unit test, integration test) và kiểm thử người dùng (UAT).
* Viết tài liệu kỹ thuật và báo cáo tổng kết đề án.

**1.3.5. Mô tả bài toán**

Bài toán mà hệ thống cần giải quyết có thể được mô tả qua các yếu tố đầu vào, xử lý và đầu ra như sau:

* **Đầu vào (Input):**
* Dữ liệu đăng nhập của người dùng (email/username, password).
* Hình ảnh truy vấn: Dưới dạng tệp tải lên từ máy người dùng hoặc đường dẫn URL.
* Dữ liệu sản phẩm: Thông tin chi tiết khi thêm/sửa sản phẩm (tên, mã, mô tả, giá, hình ảnh...).
* Dữ liệu người dùng mới (khi Admin tạo tài khoản).
* Thông tin cấu hình API Nhanh.vn (appId, businessId, accessToken...).
* Các yêu cầu thao tác CRUD từ người dùng qua giao diện.
* Yêu cầu tương tác với API Nhanh.vn (ví dụ: thông tin tạo đơn hàng).
* **Xử lý (Processing):**
* Xác thực thông tin đăng nhập và phân quyền truy cập chức năng.
* Với ảnh sản phẩm mới: Tải lên Cloudinary, trích xuất vector đặc trưng, tính toán image hash, lưu thông tin và các giá trị này vào MongoDB.
* Với ảnh truy vấn: Trích xuất vector đặc trưng và/hoặc tính image hash.
* Sử dụng chỉ mục Faiss và/hoặc so sánh hash để tìm kiếm các vector/hash tương đồng nhất trong cơ sở dữ liệu (trong phạm vi công ty của người dùng).
* Truy xuất thông tin chi tiết của các sản phẩm tương ứng từ MongoDB.
* Thực hiện các thao tác tạo, đọc, cập nhật, xóa dữ liệu (Companies, Users, Products, Images, AppConfigs) trong MongoDB theo yêu cầu.
* Gọi các API của Nhanh.vn để lấy hoặc gửi dữ liệu theo yêu cầu (lấy danh sách tỉnh thành, tạo đơn hàng...).
* **Đầu ra (Output):**
* Hiển thị danh sách các sản phẩm có hình ảnh tương đồng nhất với ảnh truy vấn cho người dùng.
* Phản hồi trạng thái (thành công/thất bại) cho các thao tác quản lý dữ liệu.
* Hiển thị thông tin người dùng, sản phẩm, cấu hình theo yêu cầu.
* Hiển thị dữ liệu trả về từ API Nhanh.vn (ví dụ: danh sách địa chỉ, trạng thái tạo đơn hàng).
* Giao diện người dùng được cập nhật để phản ánh kết quả xử lý.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ NỀN TẢNG CÔNG NGHỆ**

Chương này trình bày về các cơ sở lý thuyết và các công nghệ, nền tảng, thư viện chính được lựa chọn và sử dụng để xây dựng hệ thống Tìm kiếm Sản phẩm bằng Hình ảnh. Việc lựa chọn công nghệ được cân nhắc dựa trên tính phù hợp với bài toán, hiệu năng, khả năng mở rộng và sự phổ biến trong cộng đồng phát triển.

**2.1. Frontend**

Giao diện người dùng (Frontend) của hệ thống được xây dựng dưới dạng một ứng dụng web đơn trang (Single Page Application - SPA), cung cấp trải nghiệm tương tác, mượt mà cho người dùng cuối (Admin và User).

**2.1.1. ReactJS và các thư viện liên quan**

* **ReactJS:** Là thư viện JavaScript phổ biến do Facebook phát triển, được chọn làm nền tảng chính cho Frontend. ReactJS sử dụng kiến trúc dựa trên thành phần (Component-Based Architecture), cho phép chia nhỏ giao diện thành các phần độc lập, dễ quản lý, tái sử dụng và bảo trì (App.jsx, các tệp trong components/). Cơ chế Virtual DOM của React giúp tối ưu hóa việc cập nhật giao diện, mang lại hiệu năng cao.
* **Thư viện UI:** Hệ thống có thể sử dụng các thư viện UI như Material UI (MUI) hoặc Bootstrap (hoặc CSS tùy chỉnh trong styles/ và App.css, index.css) để xây dựng các thành phần giao diện (nút bấm, form nhập liệu, bảng hiển thị dữ liệu, modal...) một cách nhanh chóng, nhất quán và đảm bảo tính thẩm mỹ cũng như khả năng đáp ứng (responsive) trên nhiều kích thước màn hình.
* **Axios:** Thư viện HTTP client dựa trên Promise được sử dụng để thực hiện các yêu cầu API từ Frontend đến Backend một cách dễ dàng và quản lý được (services/api.service.js). Nó hỗ trợ nhiều tính năng như interceptors, xử lý lỗi, hủy yêu cầu.

**2.1.2. Quản lý trạng thái và Routing**

* **Routing (Định tuyến):** Thư viện “react-router-dom” được sử dụng để quản lý việc điều hướng giữa các trang (views) khác nhau trong ứng dụng SPA mà không cần tải lại toàn bộ trang web. Nó cho phép ánh xạ các URL tới các thành phần giao diện tương ứng (ví dụ: /login, /search, /admin/products), tạo ra trải nghiệm người dùng liền mạch (Cấu hình trong App.jsx).
* **Quản lý trạng thái (State Management):** Tùy thuộc vào độ phức tạp, hệ thống có thể sử dụng:
* **React Context API & Hooks:** Cung cấp cơ chế chia sẻ trạng thái giữa các thành phần mà không cần truyền props qua nhiều cấp (prop drilling), phù hợp cho các trạng thái toàn cục như thông tin người dùng đăng nhập, cấu hình công ty.
* **Component State (useState, useReducer):** Dùng cho các trạng thái cục bộ bên trong từng thành phần.
* **Client-side Caching:** Hệ thống có thể triển khai cơ chế cache phía client (services/cache.service.js) để lưu trữ tạm thời các dữ liệu ít thay đổi (ví dụ: danh sách tỉnh thành, cấu hình), giúp giảm số lượng yêu cầu API và tăng tốc độ phản hồi cho người dùng.

**2.2. Backend**

Backend đóng vai trò xử lý logic nghiệp vụ, quản lý dữ liệu, cung cấp API cho Frontend và tích hợp với các dịch vụ bên ngoài.

**2.2.1. FastAPI và Python**

* **Python:** Ngôn ngữ lập trình chính được chọn cho Backend nhờ hệ sinh thái thư viện phong phú, đặc biệt mạnh mẽ trong lĩnh vực Khoa học dữ liệu, AI và Machine Learning, rất phù hợp cho bài toán xử lý và tìm kiếm ảnh.
* **FastAPI:** Là một web framework hiện đại, hiệu năng cao dựa trên Python 3.7+ với các ưu điểm nổi bật:
* **Tốc độ:** Hiệu năng tương đương NodeJS và Go nhờ xây dựng trên Starlette (ASGI) và Pydantic.
* **Phát triển nhanh:** Cú pháp ngắn gọn, giảm thiểu mã lặp lại.
* **Ít lỗi:** Sử dụng type hints của Python và Pydantic để kiểm tra kiểu dữ liệu tự động, giảm lỗi trong quá trình chạy.
* **Tài liệu tự động:** Tự động sinh tài liệu API tương tác (Swagger UI, ReDoc) dựa trên mã nguồn, giúp việc kiểm thử và tích hợp dễ dàng (main.py).
* **Hỗ trợ Async:** Cho phép xử lý đồng thời nhiều yêu cầu, phù hợp với các tác vụ I/O bound như gọi API bên ngoài hoặc truy vấn cơ sở dữ liệu.
* **Pydantic:** Thư viện được FastAPI sử dụng để định nghĩa cấu trúc dữ liệu đầu vào/ra của API (models/), tự động phân tích (parse), kiểm tra tính hợp lệ (validate) và tuần tự hóa (serialize) dữ liệu.

**2.2.2. Xác thực và Phân quyền (JWT)**

* **JSON Web Tokens (JWT):** Là tiêu chuẩn mở (RFC 7519) để tạo ra các token truy cập dựa trên định dạng JSON. Hệ thống sử dụng JWT cho việc xác thực:

1. Người dùng đăng nhập thành công.
2. Backend tạo ra một JWT chứa thông tin định danh người dùng (ID, vai trò, công ty ID) và ký bằng một khóa bí mật.
3. Token được gửi về Frontend và lưu trữ (thường trong localStorage hoặc sessionStorage).
4. Với mỗi yêu cầu cần xác thực sau đó, Frontend gửi kèm JWT trong Header Authorization.
5. Middleware (middleware/auth\_middleware.py) trên Backend sẽ giải mã, xác thực chữ ký và kiểm tra tính hợp lệ (ví dụ: hết hạn) của token trước khi cho phép truy cập vào tài nguyên được bảo vệ.

* **Password Hashing:** Thư viện như “passlib” được sử dụng để băm mật khẩu người dùng trước khi lưu vào cơ sở dữ liệu, đảm bảo an toàn ngay cả khi cơ sở dữ liệu bị lộ.
* **Phân quyền:** Dựa trên thông tin vai trò (role) được lưu trong JWT, Backend có thể kiểm tra và giới hạn quyền truy cập vào các API hoặc chức năng cụ thể (ví dụ: chỉ Admin mới được quản lý người dùng khác).

**2.3. Xử lý và Tìm kiếm Hình ảnh**

Đây là phần cốt lõi của hệ thống, ứng dụng các kỹ thuật AI/ML để thực hiện tìm kiếm dựa trên nội dung hình ảnh.

**2.3.1. Trích xuất đặc trưng ảnh (EfficientNet/TensorFlow)**

* **Deep Learning & CNN:** Sử dụng các mô hình Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) đã được huấn luyện trước (pre-trained) trên các tập dữ liệu lớn như ImageNet. Các mô hình này có khả năng học được các đặc trưng thị giác phức tạp từ hình ảnh.
* **EfficientNet (hoặc mô hình tương tự):** Là một họ các mô hình CNN hiện đại, đạt được độ chính xác cao với số lượng tham số và chi phí tính toán tối ưu hơn nhiều mô hình khác. Hệ thống sử dụng một mô hình như EfficientNet thông qua thư viện **TensorFlow/Keras** (utils/image\_processing.py) để:
* Tiền xử lý ảnh đầu vào (thay đổi kích thước, chuẩn hóa pixel).
* Đưa ảnh qua mô hình CNN (loại bỏ lớp phân loại cuối cùng).
* Lấy đầu ra từ một lớp trung gian (thường là lớp global pooling) dưới dạng một **vector đặc trưng (feature vector)** - một mảng số thực có kích thước cố định (ví dụ: 1280 chiều) đại diện cho nội dung ngữ nghĩa của ảnh.

**2.3.2. Băm ảnh (Image Hashing)**

* **Perceptual Hashing:** Sử dụng các thuật toán băm nhận thức (ví dụ: pHash, aHash, dHash) từ thư viện **imagehash** (utils/image\_processing.py) để tạo ra một chuỗi hash ngắn gọn (ví dụ: 16 ký tự hex - image\_hash trong sample\_database.json) từ nội dung hình ảnh.
* **Đặc điểm:** Các ảnh tương tự về mặt thị giác (ngay cả khi có thay đổi nhỏ về kích thước, định dạng, hoặc nén) sẽ tạo ra các giá trị hash gần giống nhau (dựa trên khoảng cách Hamming).
* **Ứng dụng:** Có thể dùng để:
* Lọc nhanh các ứng viên tiềm năng trước khi so sánh vector đặc trưng (tốn kém hơn).
* Phát hiện các ảnh gần như trùng lặp hoàn toàn.

**2.3.3. Tìm kiếm Vector tương đồng (Faiss)**

* **Bài toán Nearest Neighbor Search:** Khi có một vector đặc trưng của ảnh truy vấn, cần tìm kiếm trong hàng ngàn/triệu vector đặc trưng của các sản phẩm trong cơ sở dữ liệu những vector "gần nhất" (tương đồng nhất).
* **Faiss (Facebook AI Similarity Search):** Là thư viện mã nguồn mở do Facebook phát triển, cung cấp các thuật toán hiệu quả và tối ưu hóa cao cho việc tìm kiếm tương đồng trên các tập vector lớn, đặc biệt là vector mật độ cao (dense vectors) sinh ra từ các mô hình Deep Learning (utils/image\_search.py).
* **Quy trình các bước:**

1. **Xây dựng Index:** Các vector đặc trưng của tất cả sản phẩm được nạp vào Faiss để xây dựng một cấu trúc dữ liệu chỉ mục (ví dụ: IndexFlatL2, IndexIVFFlat). Quá trình này thường thực hiện offline hoặc định kỳ.
2. **Tìm kiếm:** Khi có vector truy vấn, Faiss sử dụng chỉ mục để nhanh chóng tìm ra K vector gần nhất (K-Nearest Neighbors - KNN) dựa trên một độ đo khoảng cách (thường là Euclidean L2 hoặc Cosine Similarity).
3. **Kết quả:** Faiss trả về ID của các ảnh tương đồng nhất và khoảng cách tương ứng. Hệ thống sau đó dựa vào ID này để lấy thông tin sản phẩm chi tiết từ MongoDB.

**2.4. Cơ sở dữ liệu**

**2.4.1. MongoDB**

* **NoSQL Document Database:** MongoDB được chọn làm cơ sở dữ liệu chính vì tính linh hoạt trong cấu trúc dữ liệu (schema-flexible), phù hợp để lưu trữ thông tin sản phẩm vốn có thể đa dạng, cùng với các trường dữ liệu đặc thù như mảng vector đặc trưng (features), image hash, và các cấu hình (app\_configs).
* **Cấu trúc dữ liệu:** Dữ liệu được tổ chức thành các **Collections** (tương đương bảng trong SQL) chứa các **Documents** (tương đương hàng, định dạng BSON - tương tự JSON). Các collection chính bao gồm: companies, users, products, images, app\_configs (như trong sample\_database.json).
* **Khả năng mở rộng:** MongoDB hỗ trợ mở rộng theo chiều ngang (horizontal scaling) tốt, phù hợp khi dữ liệu tăng lên.
* **Driver/ODM:** Backend sử dụng driver như Motor (cho lập trình bất đồng bộ) hoặc Object-Document Mapper (ODM) như Beanie để tương tác với MongoDB một cách thuận tiện và an toàn từ code Python (utils/database.py).

**2.5. Dịch vụ bên ngoài**

Hệ thống tích hợp với các dịch vụ của bên thứ ba để mở rộng chức năng và tối ưu hóa hoạt động.

**2.5.1. Cloudinary (Lưu trữ ảnh)**

* **Cloud-based Image Management:** Thay vì lưu trữ file ảnh trực tiếp trên server backend, hệ thống sử dụng Cloudinary, một nền tảng quản lý tài sản số trên đám mây.
* **Lợi ích:**
* **Lưu trữ và Phân phối:** Cung cấp CDN (Content Delivery Network) giúp tải ảnh nhanh chóng từ vị trí gần người dùng nhất.
* **Tối ưu hóa và Biến đổi:** Tự động tối ưu hóa kích thước, định dạng ảnh; cho phép thực hiện các thao tác biến đổi ảnh (resize, crop, thêm watermark...) thông qua URL.
* **Giảm tải cho Backend:** Backend chỉ cần lưu trữ URL của ảnh trên Cloudinary (image\_url trong sample\_database.json), không cần xử lý việc lưu file.
* **Tích hợp:** Frontend thường tải ảnh trực tiếp lên Cloudinary (services/cloudinary.service.js) và gửi URL kết quả về cho Backend lưu trữ.

**2.5.2. Nhanh.vn (Tích hợp API)**

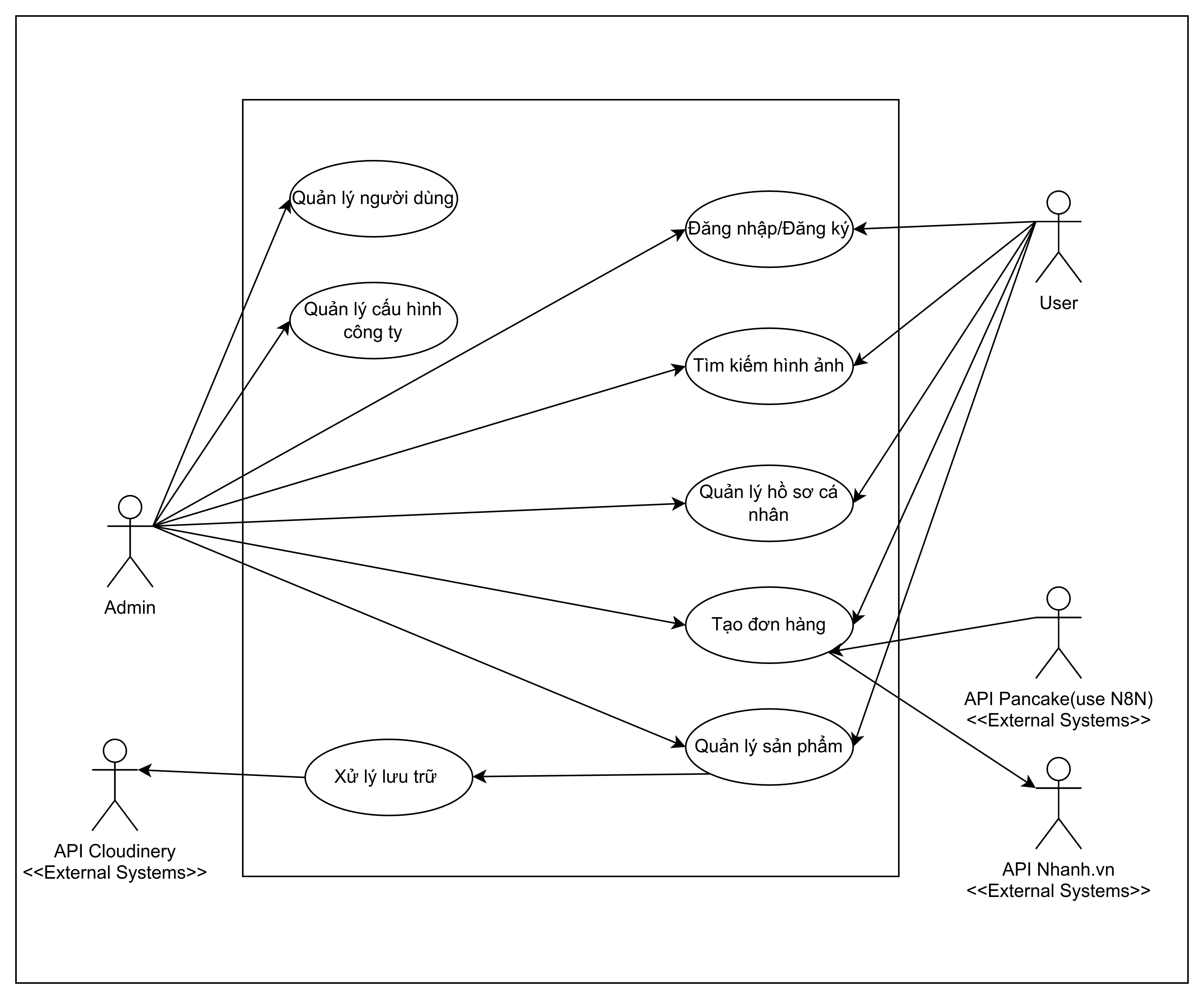
* **Nền tảng quản lý bán hàng:** Nhanh.vn là một nền tảng phổ biến cung cấp giải pháp quản lý bán hàng đa kênh tại Việt Nam.
* **Tích hợp API:** Hệ thống tích hợp với API của Nhanh.vn (routers/nhanhvn.py) để:
* **Lấy dữ liệu:** Đồng bộ hoặc truy vấn thông tin cần thiết từ Nhanh.vn như danh sách tỉnh/thành, quận/huyện, phường/xã; danh sách nhân viên bán hàng; thông tin sản phẩm (nếu cần đối chiếu).
* **Gửi dữ liệu:** Đẩy thông tin đơn hàng mới (được tạo từ giao diện hoặc có thể từ nguồn khác) lên Nhanh.vn để xử lý tiếp trong quy trình bán hàng chung của doanh nghiệp.
* **Lợi ích:** Kết nối hệ thống tìm kiếm ảnh với quy trình vận hành kinh doanh thực tế, tạo luồng làm việc liền mạch cho người dùng. Cấu hình kết nối (appId, businessId, accessToken) được lưu trữ an toàn cho từng công ty (app\_configs).

**CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

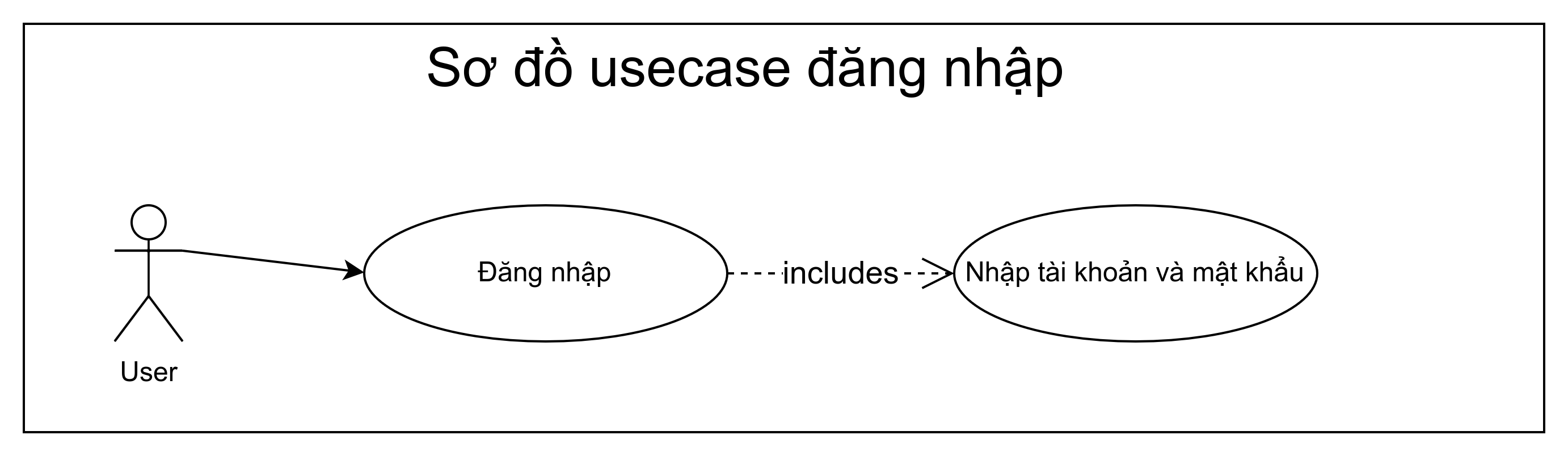
**3.1. Phân tích chức năng hệ thống (Use Cases)**

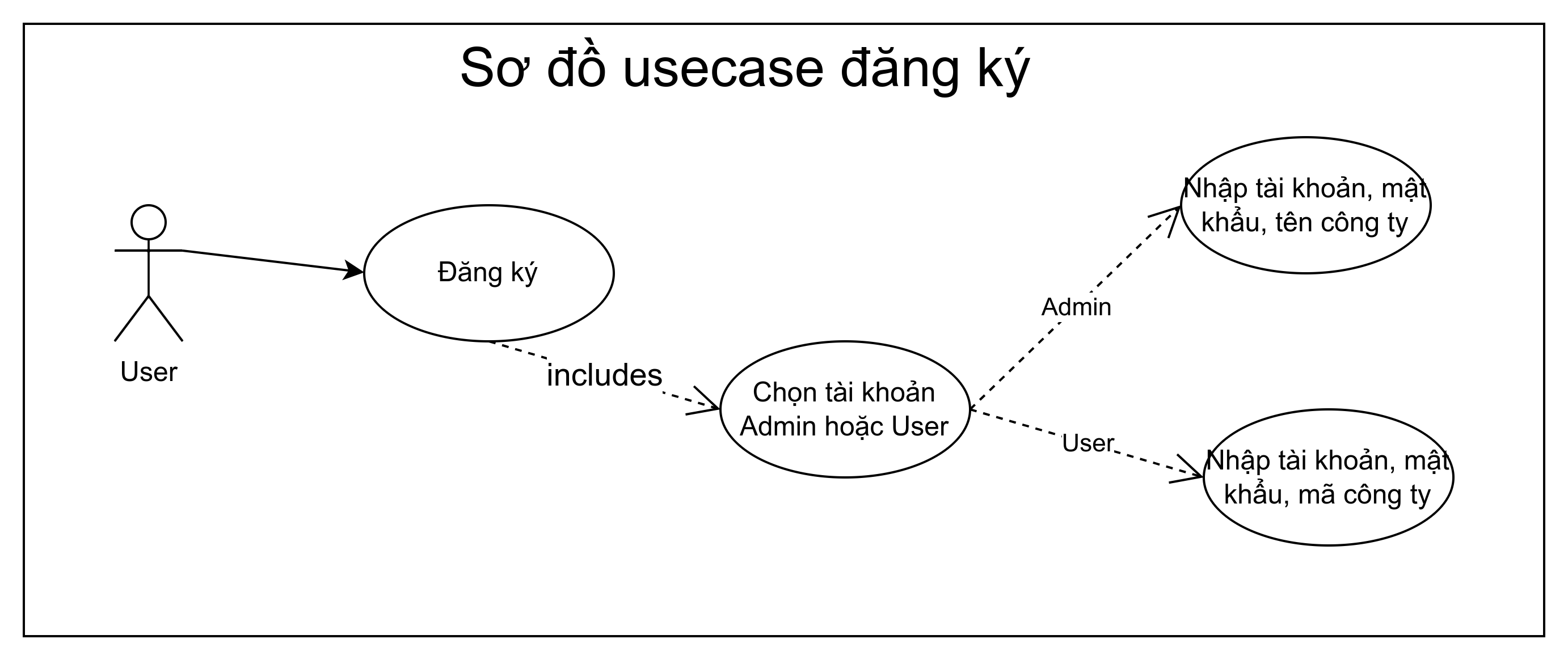
**3.1.1. Phân tích chức năng bằng biểu đồ usecase phân rã**

**3.1.1.1. Sơ đồ UseCase tổng quát**

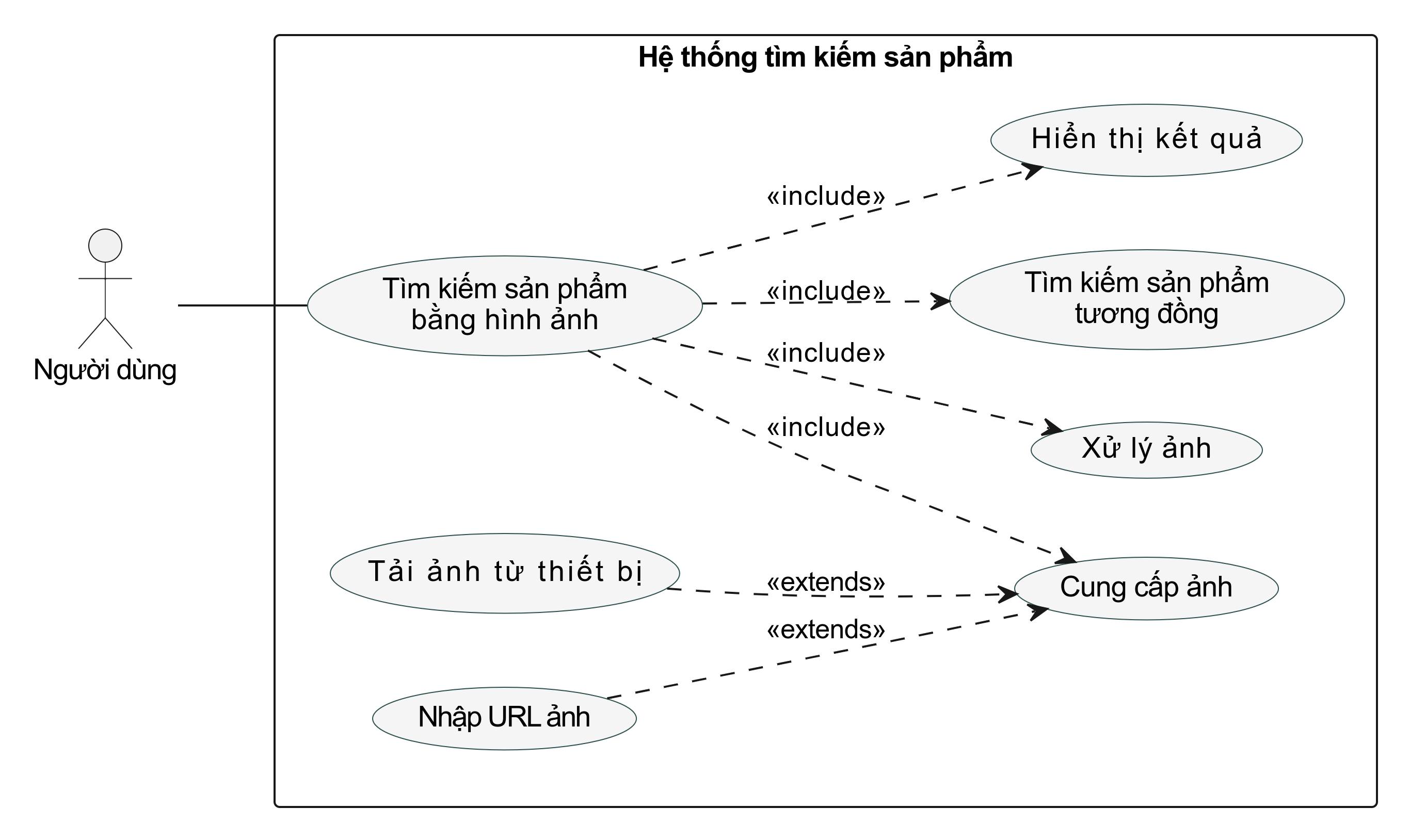
*Hình 3.1: Sơ đồ usecase tổng quát*

**3.1.1.2. Sơ đồ UseCase phân rã chức năng Đăng nhập / Đăng ký**

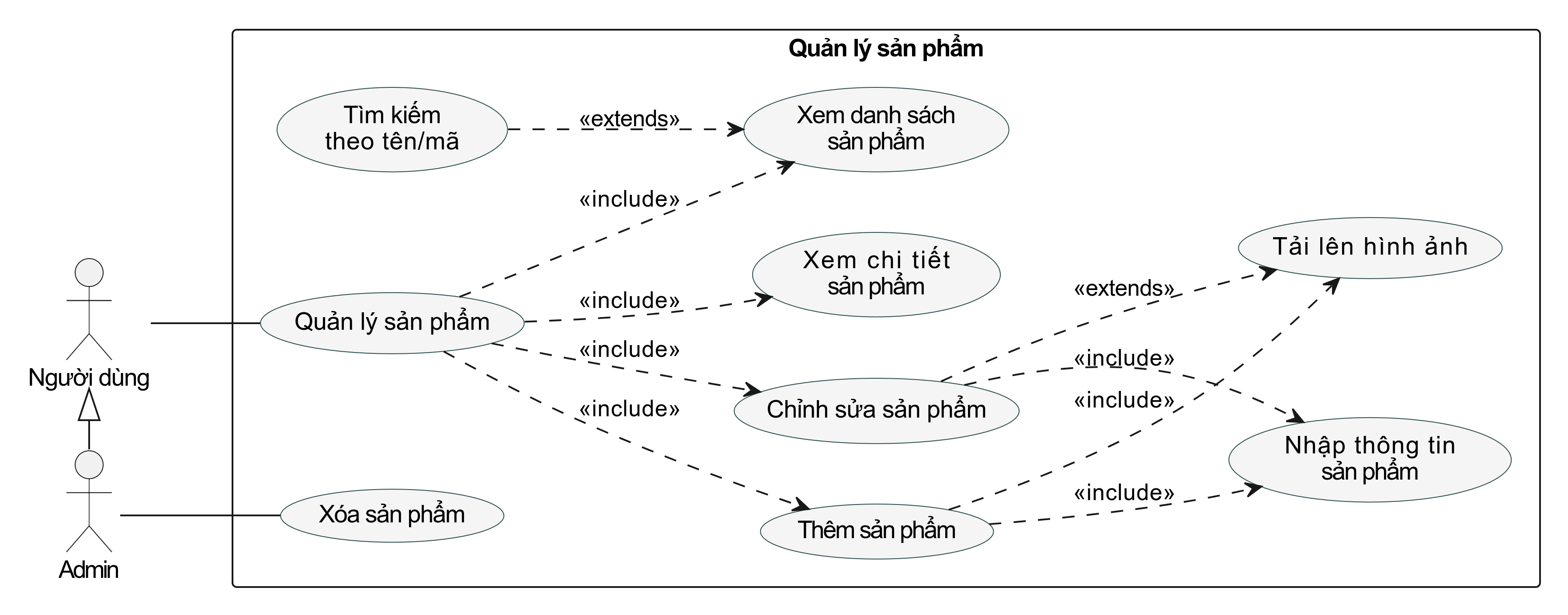
*Hình 3.2: Sơ đồ usecase phân rã chức năng đăng nhập*

*Hình 3.3: Sơ đồ usecase phân rã chức năng đăng ký*

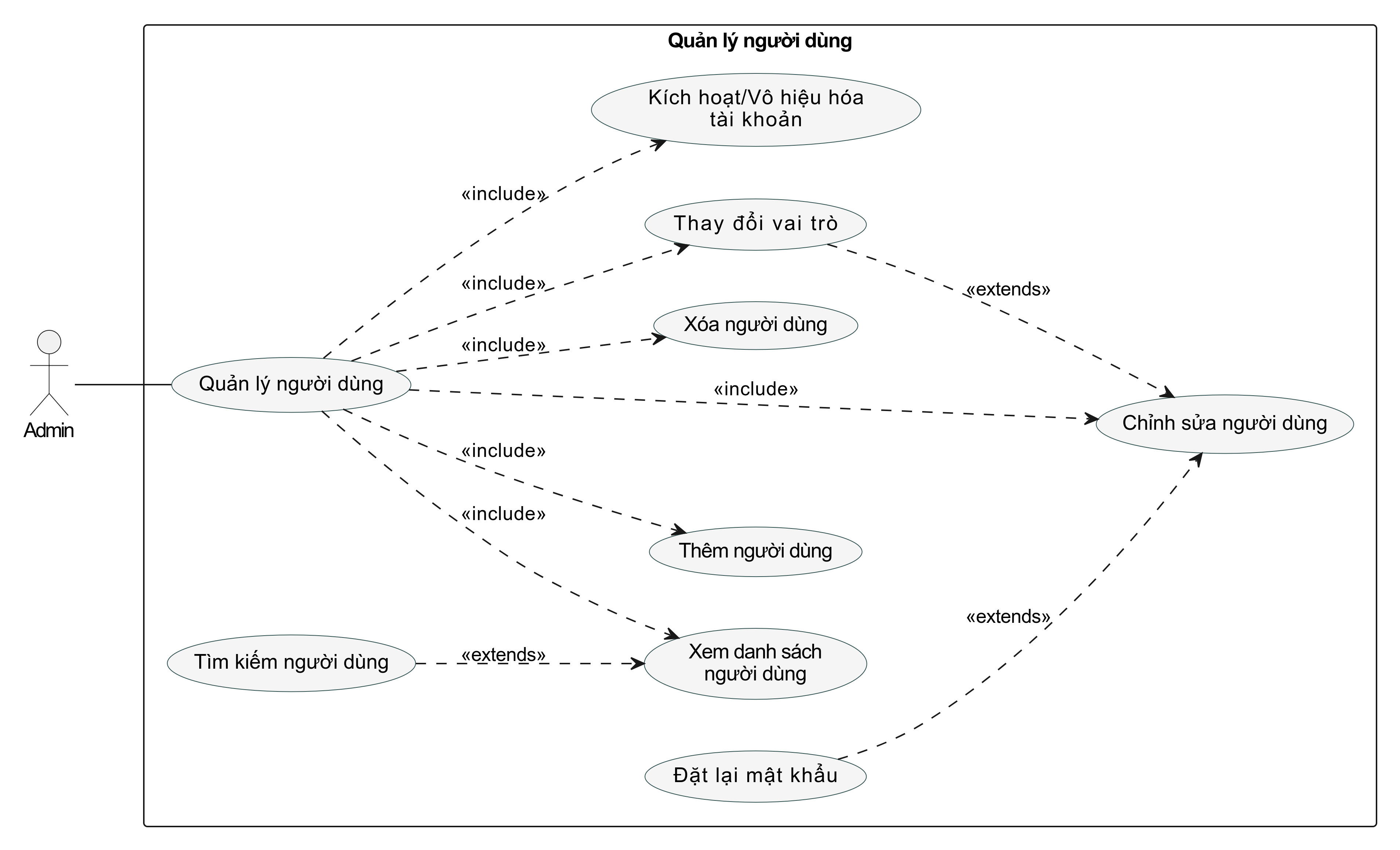
**3.1.1.3. Sơ đồ UseCase phân rã chức năng Tìm kiếm ảnh**

*Hình 3.4: Sơ đồ usecase phân rã chức năng tìm kiếm sản phẩm*

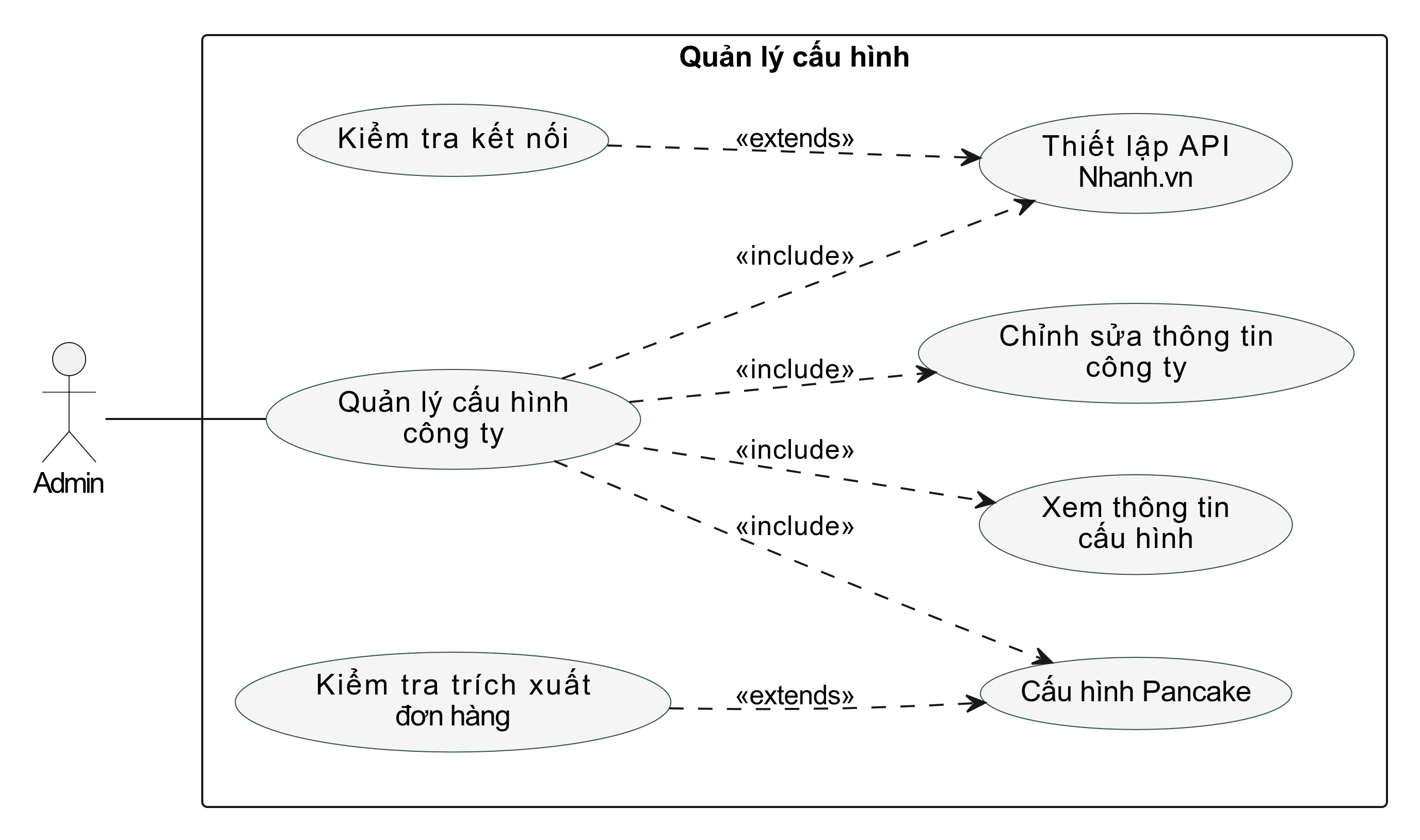
**3.1.1.4. Sơ đồ UseCase *phân rã chức năng* Quản lý sản phẩm**

*Hình 3.5: Sơ đồ usecase phân rã chức năng quản lý sản phẩm*

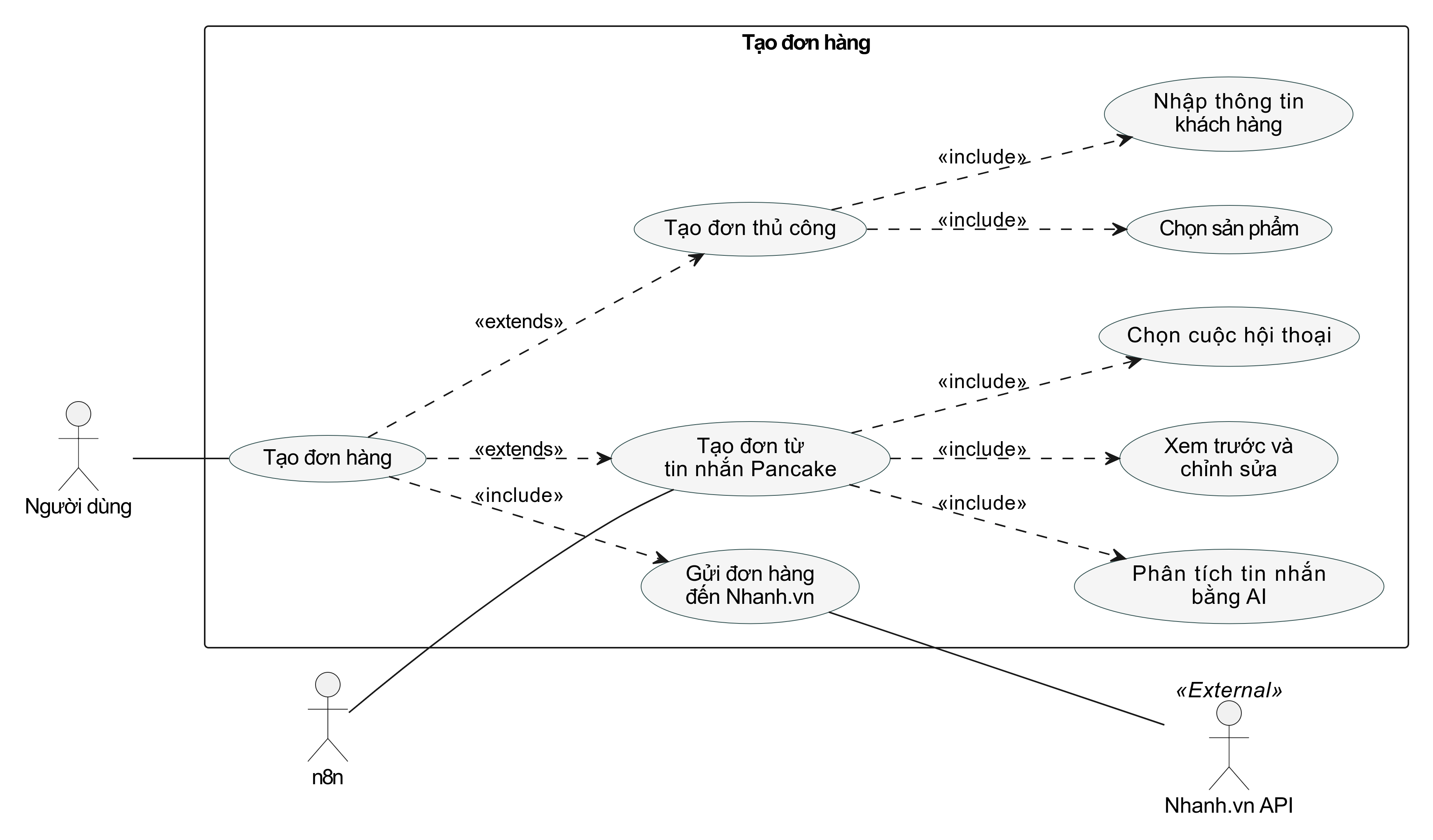
**3.1.1.5. Sơ đồ UseCase *phân rã chức năng* Quản lý người dùng (Admin)**

*Hình 3.6: Sơ đồ usecase phân rã chức năng quản lý người dùng*

**3.1.1.6. Sơ đồ UseCase *phân rã chức năng* Quản lý cấu hình (Admin)**

*Hình 3.7: Sơ đồ usecase phân rã chức năng quản lý cấu hình*

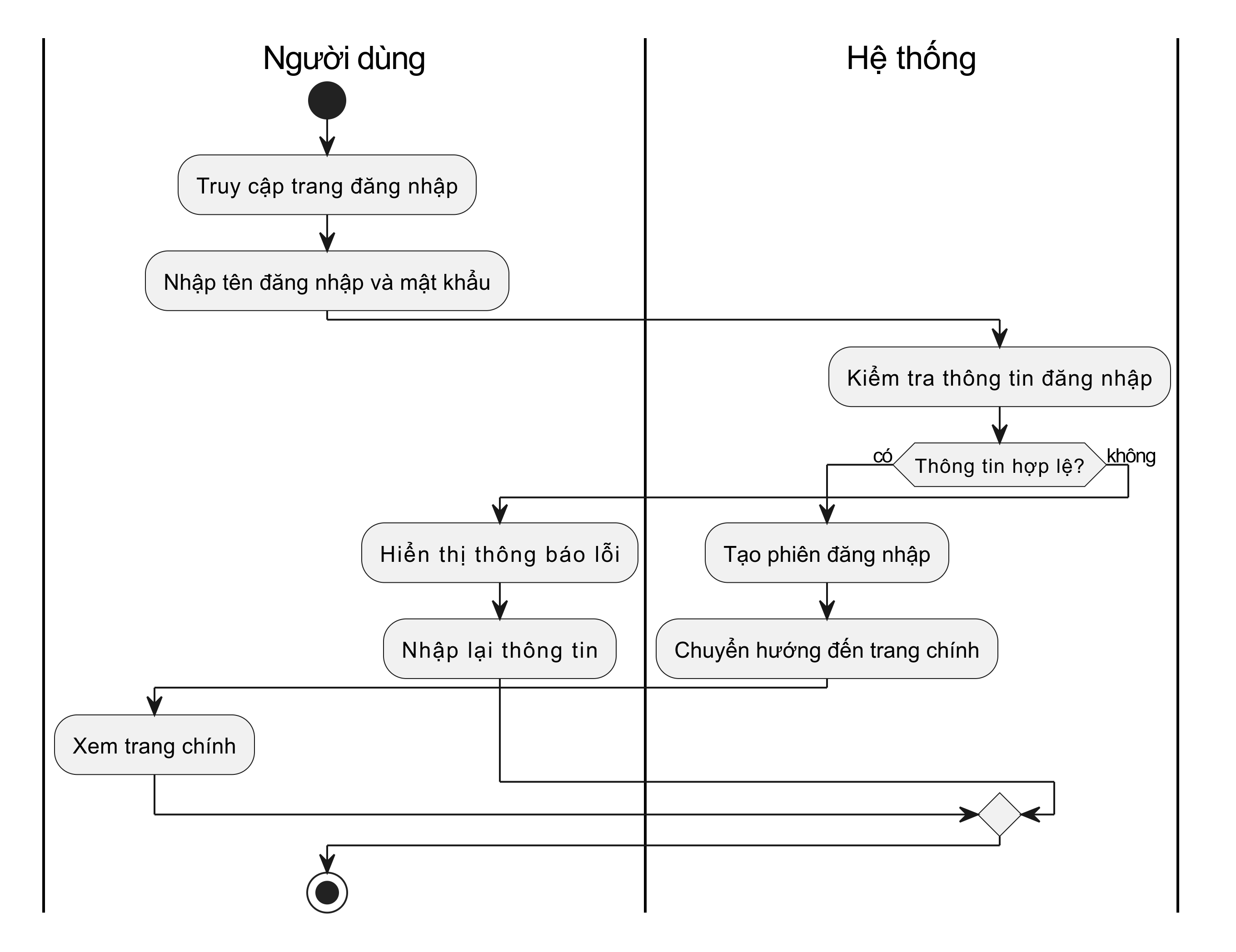
**3.1.1.7. Sơ đồ UseCase *phân rã chức năng* Tạo đơn hàng**

*Hình 3.8: Sơ đồ usecase phân rã chức năng tạo đơn hàng*

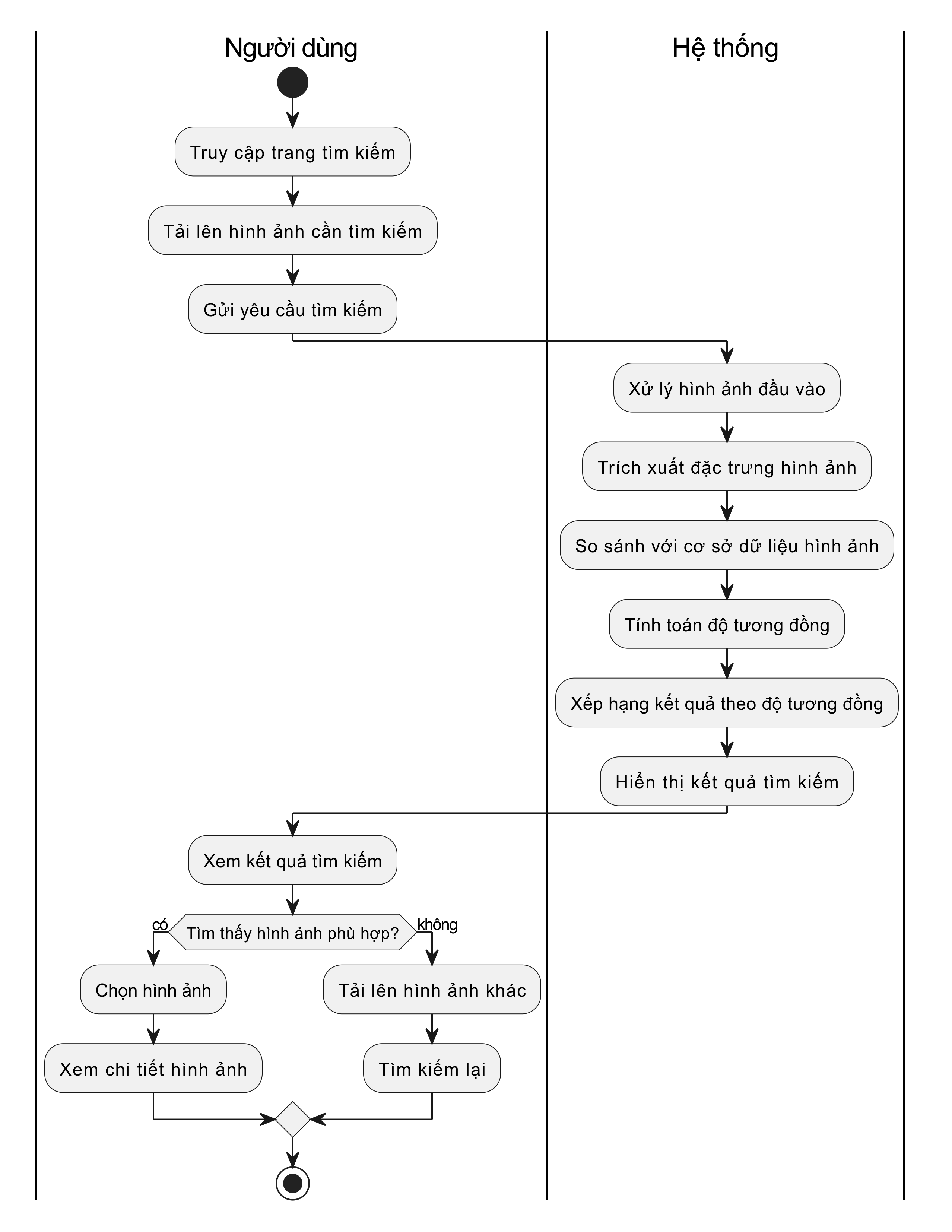
**3.1.2. Phân tích chức năng bằng sơ đồ hoạt động**

**3.1.2.1. Sơ đồ hoạt động đăng ký/đăng nhập**

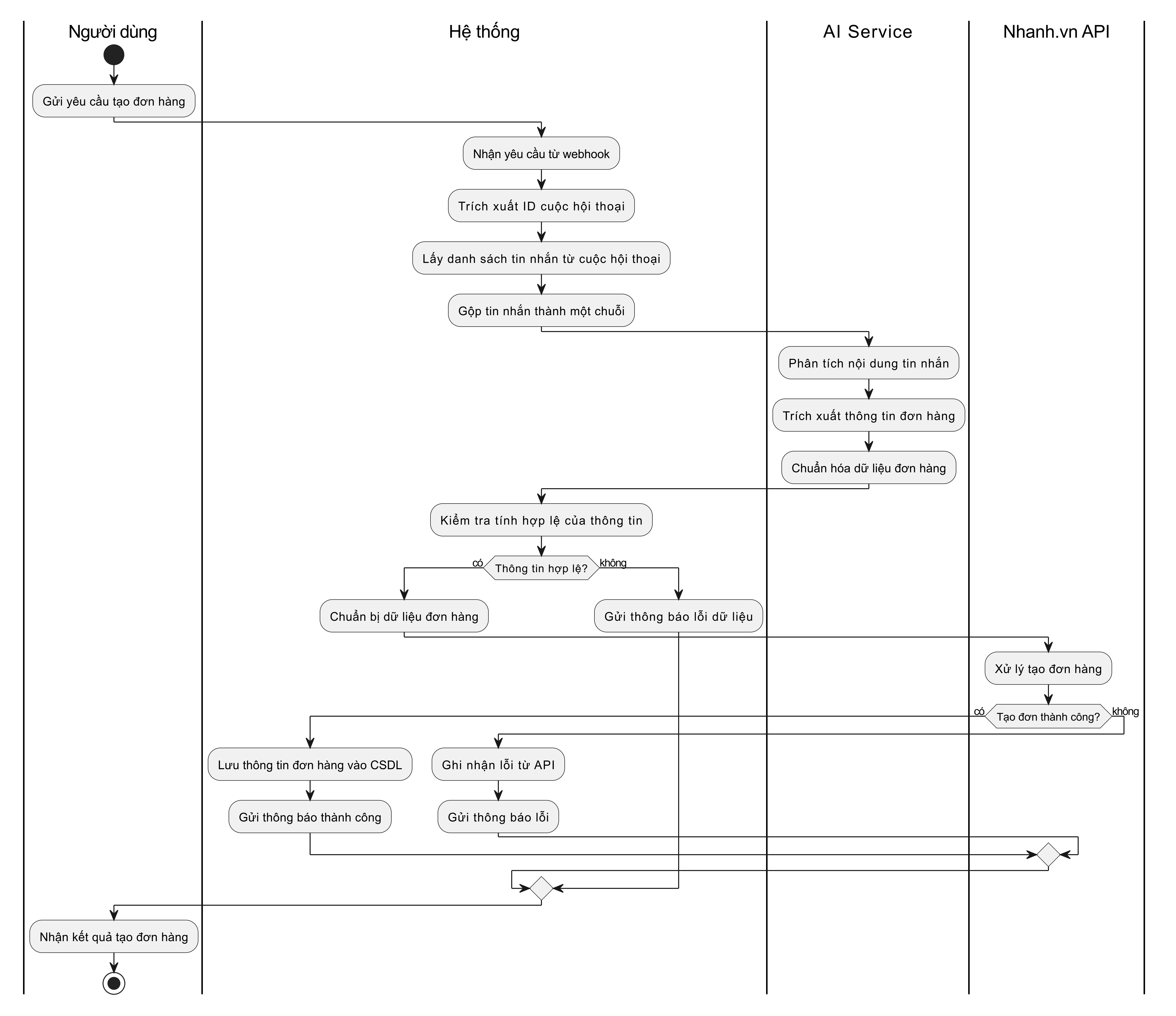
*Hình 3.9: Sơ đồ hoạt động đăng ký*

*Hình 3.10: Sơ đồ hoạt động đăng nhập*

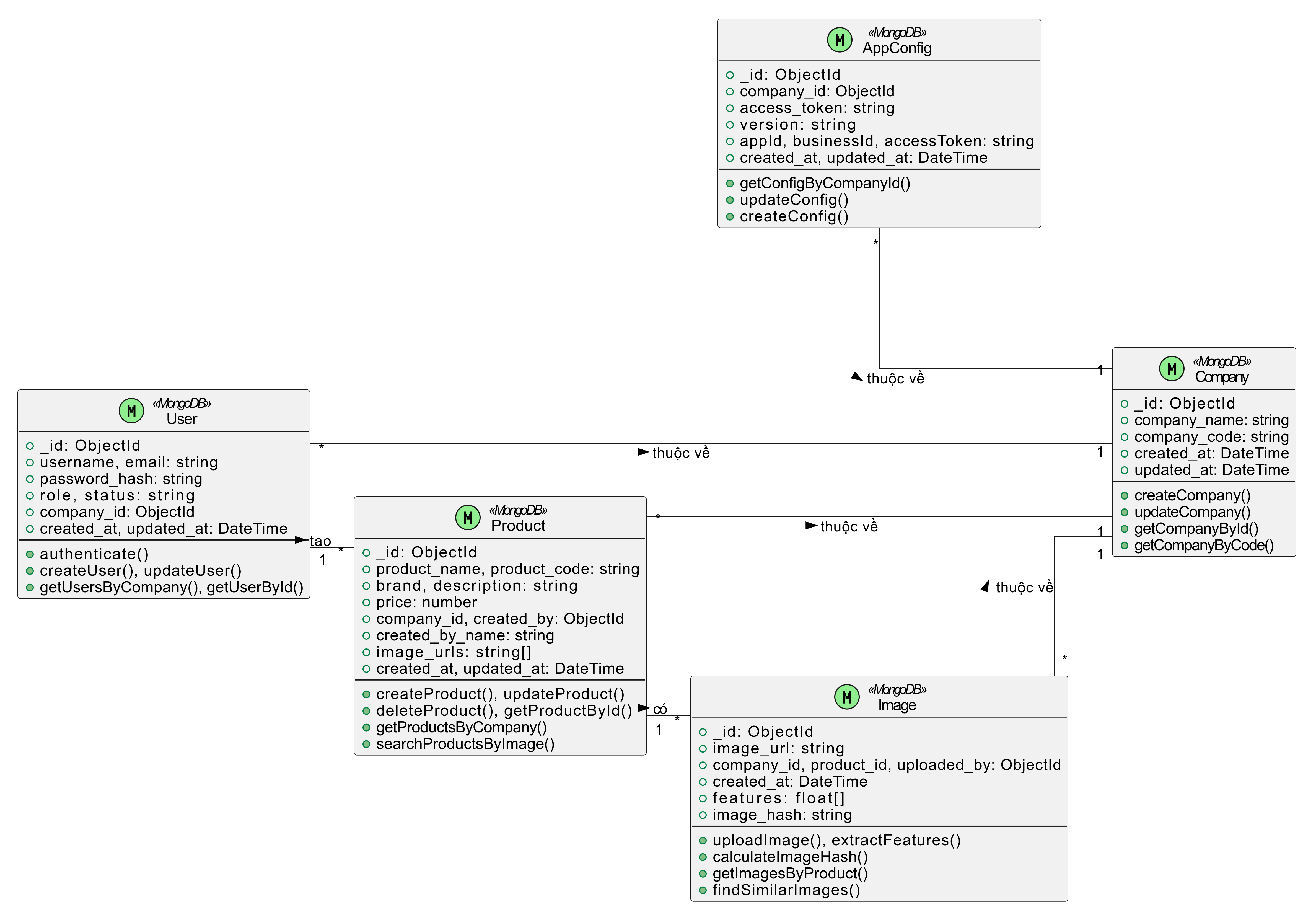
***3.1.2.2. Sơ đồ hoạt động tìm kiếm hình ảnh***

*Hình 3.11: Sơ đồ hoạt động tìm kiếm hình ảnh*

***3.1.2.2. Sơ đồ hoạt động tạo đơn hàng***

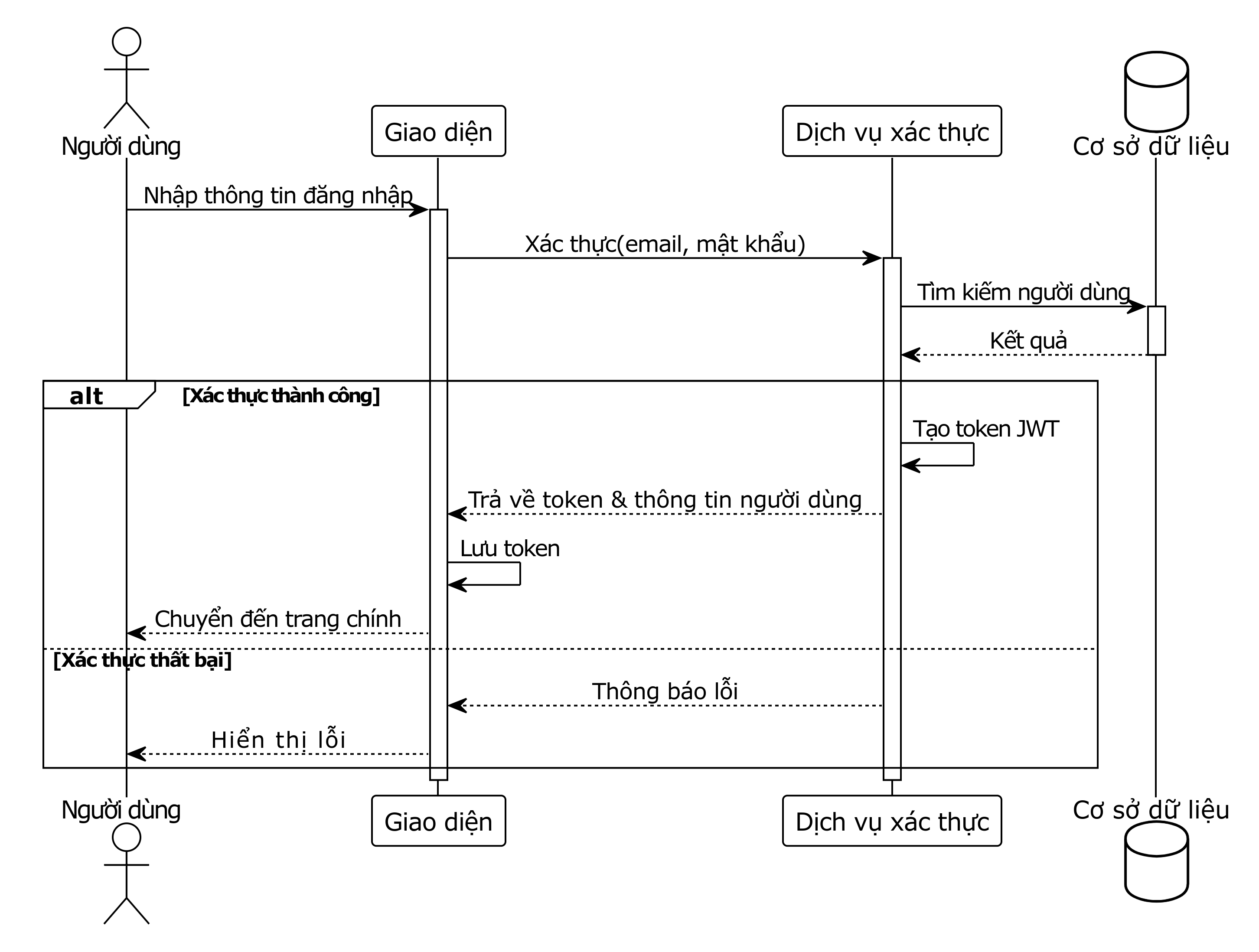
*Hình 3.12: Sơ đồ hoạt động tạo đơn hàng*

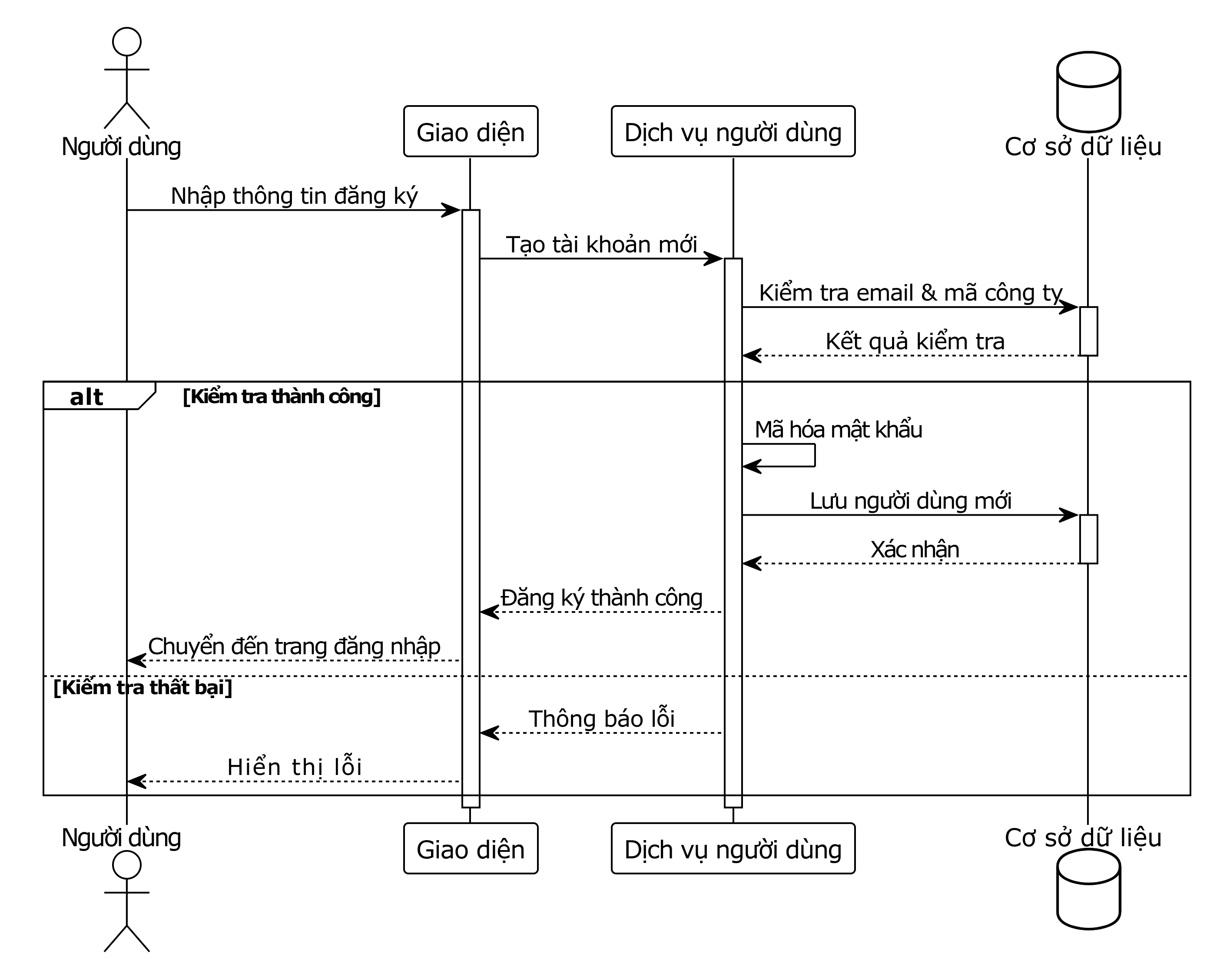
***3.1.3. Phân tích cấu trúc***

*Hình 3.13: Sơ đồ lớp lĩnh vực*

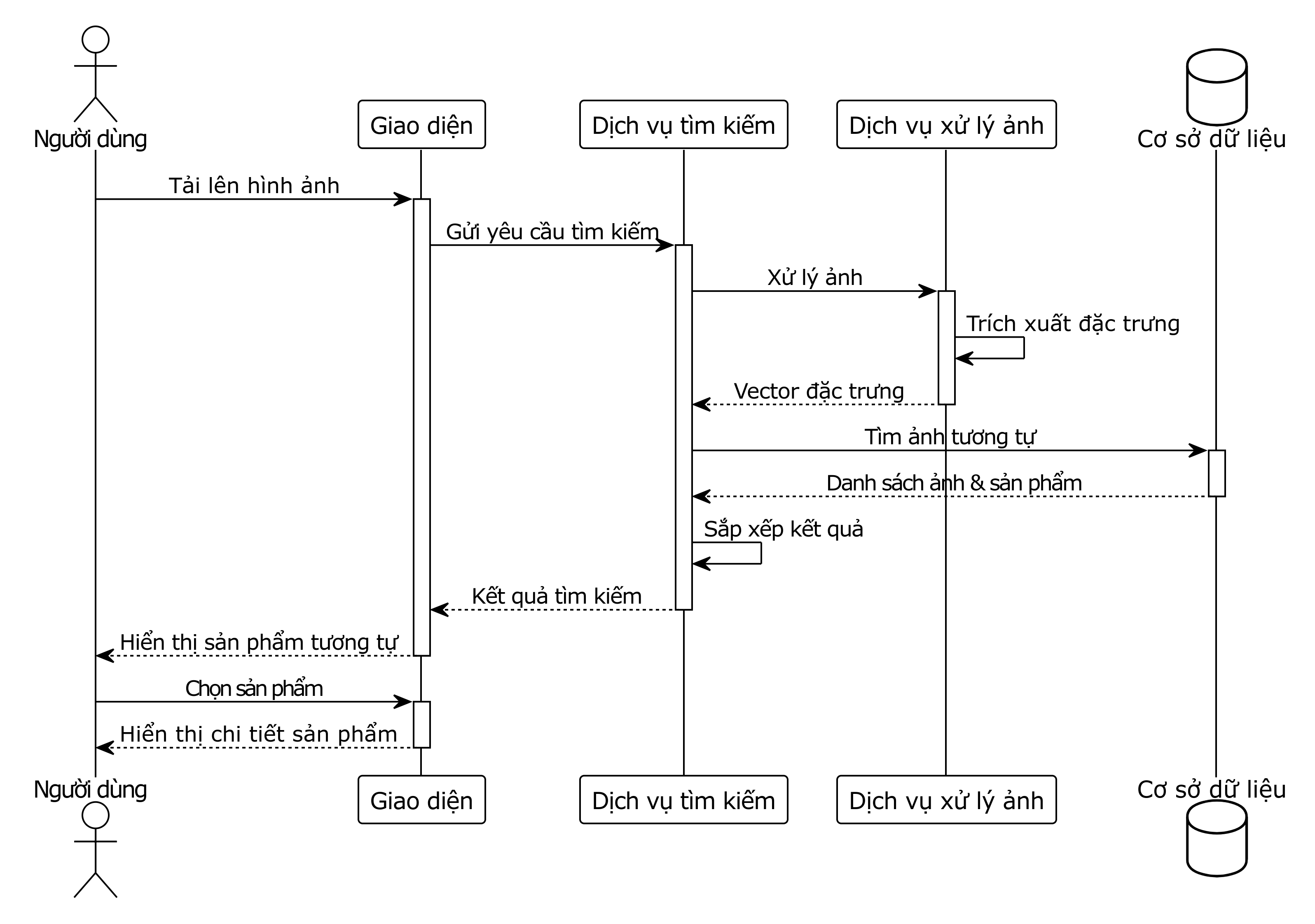
***3.1.3. Phân tích hành vi***

***3.1.3.1. Sơ đồ hành vi đăng ký/đăng nhập***

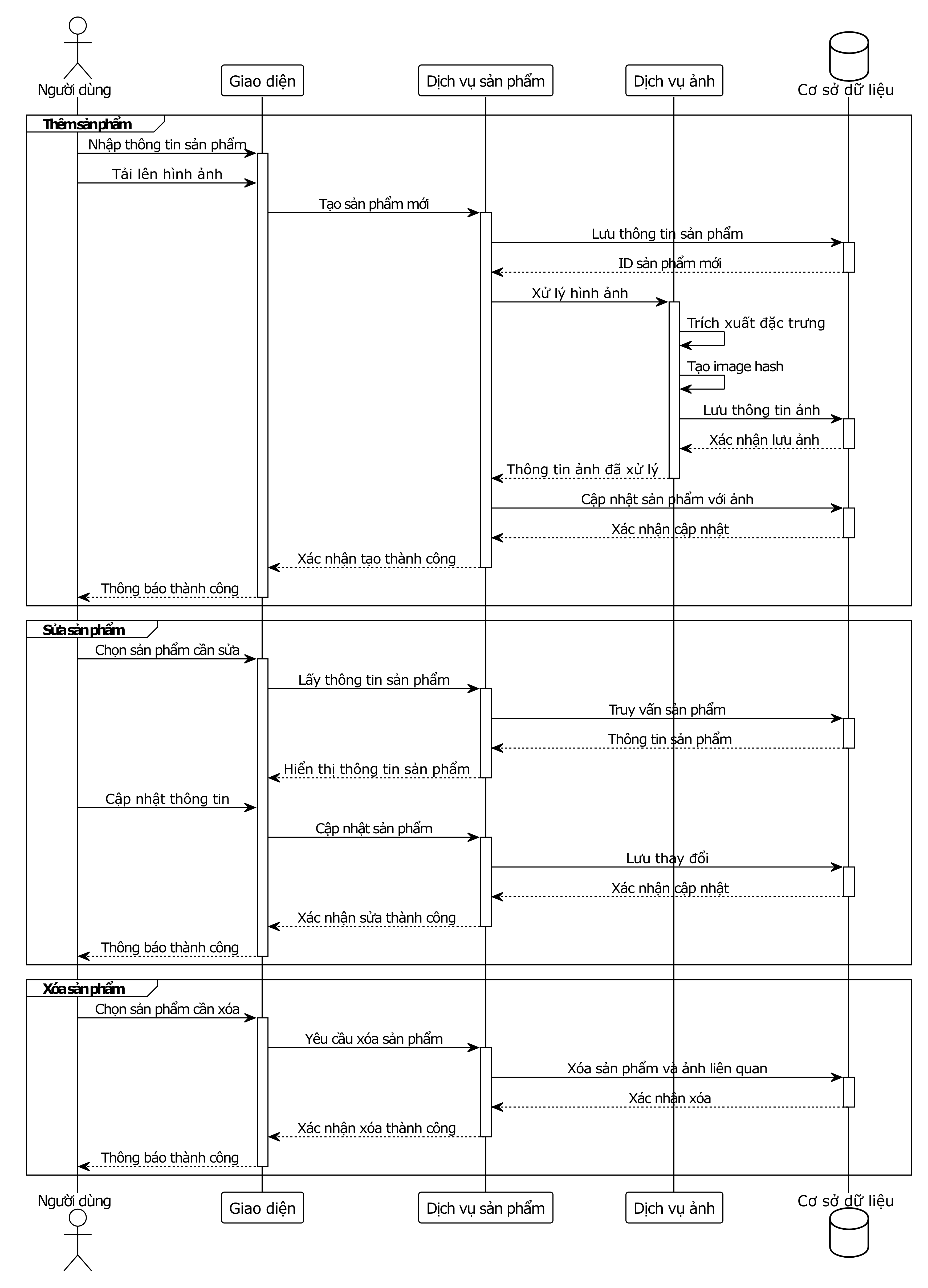
*Hình 3.14: Sơ đồ hành vi đăng nhập*

*Hình 3.14: Sơ đồ hành vi đăng ký*

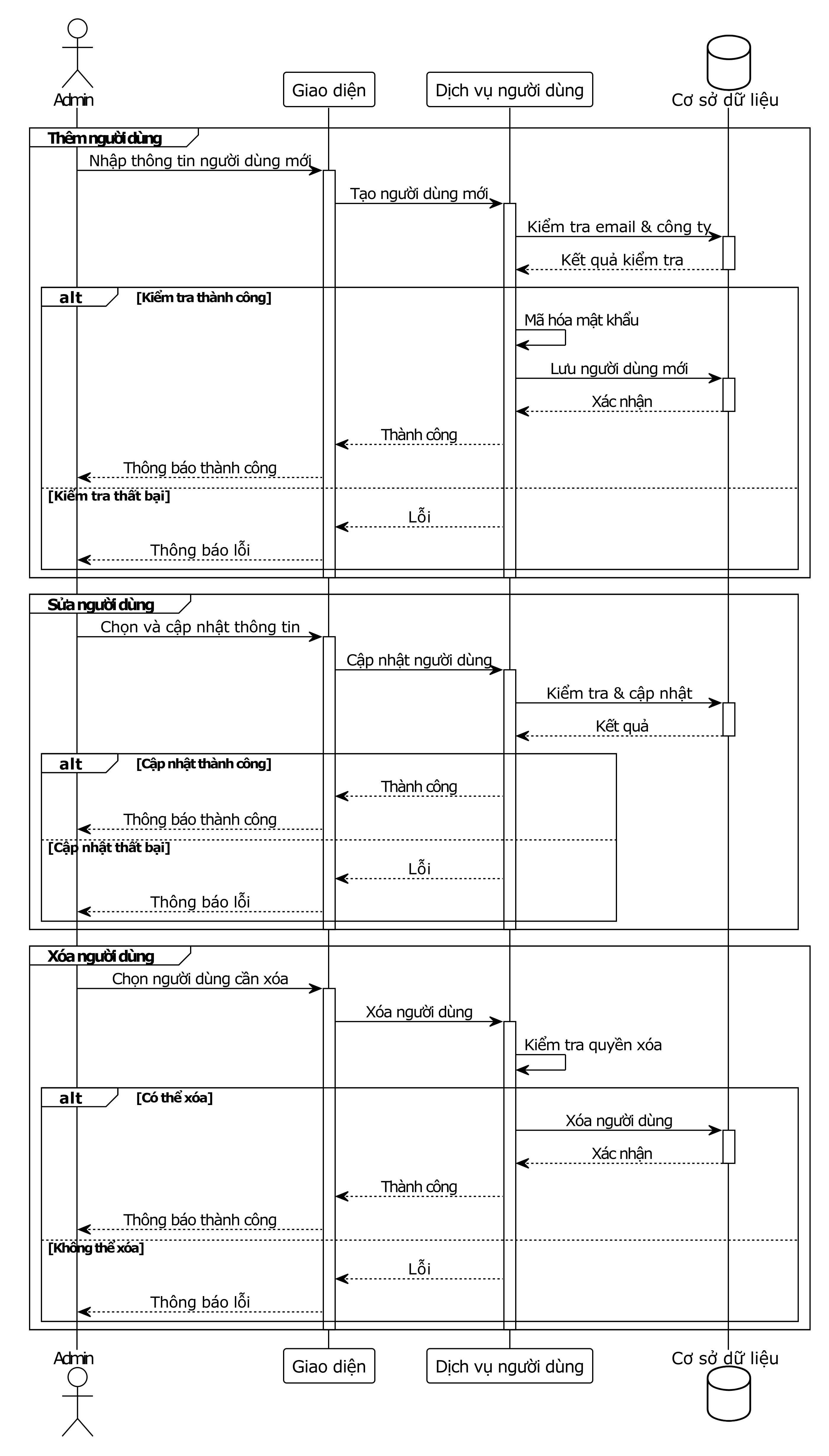
**3.1.3.2. Sơ đồ hành vi tìm kiếm hình ảnh**

*Hình 3.15: Sơ đồ hành vi tìm kiếm hình ảnh*

**3.1.3.4. Sơ đồ hành vi thêm/sửa/xóa sản phẩm**

*Hình 3.16: Sơ đồ hành vi thêm/sửa/xóa sản phẩm*

**3.1.3.5. Sơ đồ hành vi thêm/sửa/xóa người dùng**

******

*Hình 3.16: Sơ đồ hành vi thêm/sửa/xóa người dùng*

**3.1.3.4. Sơ đồ hành vi tạo đơn hàng**

*Hình 3.16: Sơ đồ hành vi tạo đơn hàng*

**3.5. Thiết kế giao diện hệ thống**

**3.5.1. Giao diện Đăng nhập / Đăng ký**

**3.5.2. Giao diện Tìm kiếm ảnh và Kết quả**

**3.5.3. Giao diện Quản lý Sản phẩm**

**3.5.4. Giao diện Quản lý Người dùng**

**3.5.5. Giao diện Quản lý Cấu hình**

**3.5.6. Giao diện Hồ sơ người dùng / Đổi mật khẩu**

**3.5.7. Giao diện Tạo đơn hàng**