

ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC SÀI GÒN

VÕ QUANG ĐĂNG KHOA 3120560047  
VÕ VĂN HÙNG 3120560031

TRUY VẤN CÂY DƯỢC LIỆU VIỆT NAM  
DỰA TRÊN ĐỘ TƯƠNG ĐỒNG VỀ DƯỢC TÍNH

KHÓA LUẬN TỐT NGHIỆP

NGÀNH: KỸ THUẬT PHẦN MỀM  
TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC

TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2024

ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC SÀI GÒN

VÕ QUANG ĐĂNG KHOA 3120560047  
VÕ VĂN HÙNG 3120560031

TRUY VẤN CÂY DƯỢC LIỆU VIỆT NAM  
DỰA TRÊN ĐỘ TƯƠNG ĐỒNG VỀ DƯỢC TÍNH

KHÓA LUẬN TỐT NGHIỆP

NGÀNH: KỸ THUẬT PHẦN MỀM  
TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC

NGƯỜI HƯỚNG DẪN: THS. LÊ NHỊ LÂM THÚY

TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2024

## LỜI CAM ĐOAN

*Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi, các số liệu và kết quả nghiên cứu nêu trong khóa luận là trung thực, được các đồng tác giả cho phép sử dụng và chưa từng được công bố trong bất kì một công trình nào khác.*

Tác giả khóa luận

**Võ Quang Đăng Khoa**

**Võ Văn Hùng**

## LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành nhất tới các thầy cô trong khoa công nghệ thông tin, trường đại học Sài Gòn, vì đã tạo điều kiện và hỗ trợ tận tình trong suốt quá trình học tập và thực hiện đề tài này.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc tới cô ThS. Lê Nhị Lãm Thúy, người đã tận tâm hướng dẫn và theo sát góp ý, chỉnh sửa, chia sẻ kinh nghiệm quý báu và hỗ trợ nhóm hoàn thiện khóa luận một cách tốt nhất và chính chu nhất.

Vì kiến thức cũng như kinh nghiệm của chúng em còn nhiều hạn chế, khó tránh khỏi những thiếu sót trong cách trình bày, chúng em rất mong nhận được sự đóng góp ý kiến của quý thầy cô.

Kính chúc thầy cô sức khỏe và thành công trong sự nghiệp.

Thành phố Hồ Chí Minh, ngày ... tháng ... năm 2024

## MỤC LỤC

<b>LỜI CAM ĐOAN.....</b>	<b>i</b>
<b>LỜI CẢM ƠN.....</b>	<b>ii</b>
<b>MỤC LỤC.....</b>	<b>1</b>
<b>DANH MỤC VIẾT TẮT.....</b>	<b>4</b>
<b>DANH MỤC CÁC BẢNG.....</b>	<b>5</b>
<b>DANH MỤC CÁC HÌNH ẢNH.....</b>	<b>7</b>
<b>MỞ ĐẦU .....</b>	<b>10</b>

## CHƯƠNG 1 TỔNG QUAN

<b>1.1. Lý do chọn đề tài .....</b>	<b>11</b>
<b>1.2. Mục đích đề tài .....</b>	<b>12</b>
<b>1.3. Đối tượng và phạm vi nghiên cứu.....</b>	<b>12</b>
1.3.1. Đối tượng nghiên cứu .....	12
1.3.2. Phạm vi nghiên cứu .....	12
<b>1.4. Phương pháp nghiên cứu.....</b>	<b>13</b>
<b>1.5. Tổng quan tình hình nghiên cứu trong và ngoài nước .....</b>	<b>13</b>
<b>1.6. Công nghệ sử dụng.....</b>	<b>14</b>

## CHƯƠNG 2 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

<b>2.1. Phân tích yêu cầu.....</b>	<b>16</b>
2.1.1. Yêu cầu chức năng.....	16
2.1.2. Yêu cầu phi chức năng.....	16
<b>2.2. Mô hình hóa yêu cầu .....</b>	<b>17</b>
2.2.1. Usecase .....	17
2.2.2. Sơ đồ lớp .....	28
2.2.3. Sơ đồ tuần tự .....	31
<b>2.3. Thiết kế cơ sở dữ liệu .....</b>	<b>36</b>

## CHƯƠNG 3 CƠ SỞ LÝ THUYẾT

<b>3.1. Cơ sở dữ liệu vector .....</b>	<b>39</b>
3.1.1. Cơ sở dữ liệu vector là gì? .....	39

3.1.2.	Vector embedding và vai trò trong AI .....	41
3.1.3.	Ưu điểm của cơ sở dữ liệu vector .....	42
3.1.4.	Sự khác nhau giữa chỉ mục vector độc lập và cơ sở dữ liệu vector.....	43
3.1.5.	Cơ sở dữ liệu vector làm việc thế nào?.....	45
3.1.6.	Kiến trúc không máy chủ và lợi ích.....	46
3.1.7.	Thuật toán trong cơ sở dữ liệu vector .....	48
3.1.8.	Lọc siêu dữ liệu.....	52
<b>3.2.</b>	<b>Các giải thuật tìm kiếm tương đồng.....</b>	<b>54</b>
3.2.1.	Độ tương đồng với cosine.....	54
3.2.2.	Khoảng cách Euclid .....	55
3.2.3.	Tích vô hướng .....	55
3.2.4.	Khoảng cách Manhattan .....	56
3.2.5.	Thuật toán Hamming .....	57
<b>3.3.</b>	<b>Cơ sở dữ liệu Qdrant .....</b>	<b>58</b>
3.3.1.	Giới thiệu cơ sở dữ liệu Qdrant .....	58
3.3.2.	Ưu điểm và nhược điểm của cơ sở dữ liệu Qdrant .....	59
3.3.3.	Triển khai dữ liệu lên Qdrant.....	59

## CHƯƠNG 4

### XÂY DỰNG MÔ HÌNH XỬ LÝ

<b>4.1.</b>	<b>Kiến trúc hệ thống.....</b>	<b>64</b>
4.1.1.	Khung nhìn logic .....	64
4.1.2.	Khung nhìn vật lý .....	66
4.1.3.	Quy trình xử lý yêu cầu .....	66
4.1.4.	Ưu điểm của kiến trúc.....	67
<b>4.2.</b>	<b>Mô hình xử lý dữ liệu.....</b>	<b>68</b>
4.2.1.	Mô hình tổng quát.....	68
4.2.2.	Cấu trúc của dữ liệu cây thuốc.....	69
4.2.3.	Thiết kế lược đồ cơ sở dữ liệu .....	70
4.2.4.	Xây dựng cơ sở dữ liệu .....	72
4.2.5.	Xây dựng CSDL với MySQL .....	74
<b>4.3.</b>	<b>Mô hình vector.....</b>	<b>76</b>
4.3.1.	Mô hình tổng quát.....	76
4.3.2.	Tổng quát hóa các thành phần dược liệu .....	76
4.3.3.	Chuẩn hóa giá trị thành phần dược liệu .....	78
4.3.4.	Thuật toán tìm kiếm.....	79

4.3.5.	Đẩy dữ liệu vector lên cơ sở dữ liệu Qdrant.....	82
--------	--	----

## **CHƯƠNG 5**

### **XÂY DỰNG ỨNG DỤNG**

<b>5.1.</b>	<b>Giao diện .....</b>	<b>84</b>
5.1.1.	Trang chủ .....	84
5.1.2.	Trang tra cứu.....	90
5.1.3.	Trang xem chi tiết.....	92
5.1.4.	Trang giới thiệu.....	94
5.1.5.	Trang đăng nhập .....	95
5.1.6.	Trang thống kê .....	96
5.1.7.	Trang quản lý họ cây thuốc.....	98
5.1.8.	Trang quản lý cây thuốc.....	102
<b>5.2.</b>	<b>Kiểm thử.....</b>	<b>105</b>
5.2.1.	Môi trường kiểm thử.....	105
5.2.2.	Chức năng tìm kiếm.....	106
5.2.3.	Chức năng tìm kiếm nâng cao .....	107
5.2.4.	Chức năng xem chi tiết cây thuốc.....	108
5.2.5.	Chức năng đăng nhập.....	109
5.2.6.	Chức năng thống kê .....	110
5.2.7.	Chức năng quản lý họ cây thuốc.....	111
5.2.8.	Chức năng quản lý cây thuốc.....	112

## **CHƯƠNG 6**

### **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

<b>6.1.</b>	<b>Kết quả đạt được .....</b>	<b>113</b>
<b>6.2.</b>	<b>Hướng phát triển .....</b>	<b>114</b>
<b>TÀI LIỆU THAM KHẢO .....</b>		<b>116</b>

**DANH MỤC VIẾT TẮT**

TPHH	Thành phần hóa học
API	Application Programming Interface
REST	Representational State Transfer
CSDL	Cơ sở dữ liệu
FAISS	Facebook AI Similarity Search
CRUD	Create, Read, Update, Delete
VPS	Virtual Private Server
HTTP	Hypertext Transfer Protocol
JSON	Javascript Object Notation
gRPC	gRPC Remote Procedure Calls



## DANH MỤC CÁC BẢNG

Bảng 2.1: Đặc tả usecase tìm kiếm theo từ khóa .....	19
Bảng 2.2: Đặc tả usecase tìm kiếm theo 10 thành phần hóa học .....	20
Bảng 2.3: Đặc tả usecase xem chi tiết cây thuốc .....	21
Bảng 2.4: Đặc tả usecase đăng nhập .....	22
Bảng 2.5: Đặc tả usecase thống kê.....	23
Bảng 2.6: Đặc tả usecase xem thông tin họ cây thuốc.....	24
Bảng 2.7: Đặc tả usecase thêm họ cây thuốc .....	25
Bảng 2.8: Đặc tả usecase sửa họ cây thuốc.....	25
Bảng 2.9: Đặc tả usecase xem thông tin cây thuốc.....	26
Bảng 2.10: Đặc tả usecase thêm cây thuốc .....	27
Bảng 2.11: Đặc tả usecase sửa cây thuốc.....	27
Bảng 2.12: Bảng mô tả các thuộc tính của lớp Families.....	29
Bảng 2.13: Bảng mô tả các phương thức của lớp Families .....	29
Bảng 2.14: Bảng mô tả các thuộc tính của lớp Images.....	29
Bảng 2.15: Bảng mô tả các thuộc tính của lớp MedicinalPlants .....	29
Bảng 2.16: Bảng mô tả các phương thức của lớp MedicinalPlants .....	30
Bảng 2.17: Bảng mô tả thực thể cây thuốc .....	37
Bảng 2.18: Bảng mô tả thực thể họ cây thuốc .....	37
Bảng 2.19: Bảng mô tả thực thể hình ảnh.....	38
Bảng 3.1: Bảng so sánh sự khác nhau giữa chỉ mục vector độc lập và cơ sở dữ liệu vector.....	43
Bảng 4.1: Kết quả truy vấn .....	79
Bảng 4.2: Kết quả truy vấn .....	80
Bảng 5.1: Bảng danh sách biến cố của giao diện trang chủ (thanh header) .....	84
Bảng 5.2: Bảng mô tả giao diện trang chủ (thanh header).....	84
Bảng 5.3: Bảng danh sách biến cố của giao diện trang chủ (slider) .....	86
Bảng 5.4: Bảng mô tả giao diện trang chủ (slider) .....	86
Bảng 5.5: Bảng danh sách biến cố của giao diện trang chủ (phần dược liệu được tìm nhiều nhất).....	87
Bảng 5.6: Bảng mô tả giao diện trang chủ (phần dược liệu được tìm nhiều nhất) .....	87

Bảng 5.7: Bảng danh sách biến cố của giao diện trang chủ (phần dược quý) .....	88
Bảng 5.8: Bảng mô tả giao diện trang chủ (phần dược liệu quý) .....	88
Bảng 5.9: Bảng danh sách biến cố của giao diện trang chủ (phần “BẠN CÓ BIẾT”)... 89	
Bảng 5.10: Bảng mô tả giao diện trang chủ (phần “BẠN CÓ BIẾT”).....	89
Bảng 5.11: Bảng danh sách biến cố của giao diện trang tra cứu .....	90
Bảng 5.12: Bảng mô tả giao diện trang tra cứu .....	91
Bảng 5.13: Bảng danh sách biến cố của giao diện trang chi tiết cây thuốc .....	92
Bảng 5.14: Bảng mô tả giao diện trang chi tiết cây thuốc .....	93
Bảng 5.15: Bảng danh sách biến cố của giao diện trang giới thiệu .....	94
Bảng 5.16: Bảng mô tả giao diện trang giới thiệu .....	94
Bảng 5.17: Bảng danh sách biến cố của giao diện trang đăng nhập.....	95
Bảng 5.18: Bảng mô tả giao diện trang đăng nhập .....	96
Bảng 5.19: Bảng danh sách biến cố của giao diện trang thống kê .....	96
Bảng 5.20: Bảng mô tả giao diện trang thống kê.....	97
Bảng 5.21: Bảng danh sách biến cố của giao diện quản lý họ cây thuốc (phần xem họ cây thuốc) .....	98
Bảng 5.22: Bảng mô tả giao diện quản lý họ cây thuốc (phần xem họ cây thuốc).....	98
Bảng 5.23: Bảng danh sách biến cố của giao diện quản lý họ cây thuốc (phần thêm họ cây thuốc) .....	99
Bảng 5.24: Bảng mô tả giao diện quản lý họ cây thuốc (phần thêm họ cây thuốc) .....	100
Bảng 5.25: Bảng danh sách biến cố của giao diện quản lý họ cây thuốc (phần sửa họ cây thuốc).....	101
Bảng 5.26: Bảng mô tả giao diện quản lý họ cây thuốc (phần sửa họ cây thuốc).....	101
Bảng 5.27: Bảng danh sách biến cố của giao diện quản lý cây thuốc (phần xem các cây thuốc).....	102
Bảng 5.28: Bảng mô tả giao diện quản lý cây thuốc (phần xem các cây thuốc) .....	102
Bảng 5.29: Bảng danh sách biến cố của giao diện quản lý cây thuốc (phần thêm cây thuốc).....	103
Bảng 5.30: Bảng mô tả giao diện quản lý cây thuốc (phần thêm cây thuốc) .....	103
Bảng 5.31: Bảng danh sách biến cố của giao diện quản lý cây thuốc (phần sửa cây thuốc).....	105
Bảng 5.32: Bảng mô tả giao diện quản lý cây thuốc (phần sửa cây thuốc).....	105

Bảng 5.33: Bảng kết quả kiểm thử chức năng tìm kiếm.....	106
Bảng 5.34: Bảng kết quả kiểm thử chức năng tìm kiếm nâng cao .....	107
Bảng 5.35: Bảng kết quả kiểm thử chức năng xem chi tiết cây thuốc.....	108
Bảng 5.36: Bảng kết quả kiểm thử chức năng đăng nhập.....	109
Bảng 5.37: Bảng kết quả kiểm thử chức năng thống kê .....	110
Bảng 5.38: Bảng kết quả kiểm thử chức năng quản lý họ cây thuốc.....	111
Bảng 5.39: Bảng kết quả kiểm thử chức năng quản lý cây thuốc.....	112

## **DANH MỤC CÁC HÌNH ẢNH**

Hình 2.1: Usecase tổng quát ở vai trò quản trị viên.....	17
Hình 2.2: Usecase tổng quát ở vai trò người dùng .....	18
Hình 2.3: Usecase tìm kiếm cây thuốc theo từ khóa.....	18
Hình 2.4: Usecase tìm kiếm cây thuốc tương đồng theo 10 thành phần hóa học .....	19
Hình 2.5: Usecase xem chi tiết cây thuốc .....	20
Hình 2.6: Usecase đăng nhập.....	21
Hình 2.7: Usecase thống kê.....	23
Hình 2.8: Usecase quản lý họ cây thuốc .....	24
Hình 2.9: Usecase quản lý cây thuốc .....	26
Hình 2.10: Sơ đồ lớp .....	28
Hình 2.11: Sơ đồ tuần tự của chức năng tìm kiếm.....	31
Hình 2.12: Sơ đồ tuần tự chức năng tìm kiếm theo 10 TPHH.....	31
Hình 2.13: Sơ đồ tuần tự xem chi tiết cây thuốc.....	32
Hình 2.14: Sơ đồ tuần tự chức năng đăng nhập.....	32
Hình 2.15: Sơ đồ tuần tự xem thống kê .....	33
Hình 2.16: Sơ đồ tuần tự xem thông tin họ cây thuốc .....	33
Hình 2.17: Sơ đồ tuần tự thêm họ cây thuốc.....	34
Hình 2.18: Sơ đồ tuần tự sửa họ cây thuốc .....	34
Hình 2.19: Sơ đồ tuần tự xem thông tin cây thuốc .....	35
Hình 2.20: Sơ đồ tuần tự thêm cây thuốc.....	35
Hình 2.21: Sơ đồ tuần tự sửa cây thuốc .....	36
Hình 2.22: Sơ đồ thực thể kết hợp .....	36

Hình 3.1: Vai trò của cơ sở dữ liệu vector với AI .....	40
Hình 3.2: Cách hoạt động của một cơ sở dữ liệu vector.....	45
Hình 3.4: Lượng tử hóa tích.....	49
Hình 3.5: Băm vị trí nhạy cảm.....	50
Hình 3.6: Thuật toán thế giới nhỏ có điều hướng theo phân cấp (nguồn: <a href="https://www.pinecone.io/learn/vector-database/">https://www.pinecone.io/learn/vector-database/</a> ) .....	51
Hình 3.8: Hình minh họa cho thuật toán Manhattan (nguồn: <a href="https://weaviate.io/blog/distance-metrics-in-vector-search">https://weaviate.io/blog/distance-metrics-in-vector-search</a> ).....	56
Hình 3.9: Cơ sở dữ liệu Qdrant.....	58
Hình 4.1: Kiến trúc phần mềm.....	64
Hình 4.2: Mô hình tổng quát.....	68
Hình 4.3: Ảnh minh họa về thông tin của một cây thuốc .....	69
Hình 4.4: Lược đồ cơ sở dữ liệu .....	70
Hình 4.5: Mô hình quan hệ .....	71
Hình 4.6: Kết quả cài đặt cơ sở dữ liệu.....	71
Hình 4.7: Kết quả trích xuất dữ liệu họ cây thuốc .....	72
Hình 4.8: Kết quả trích xuất dữ liệu cây thuốc .....	73
Hình 4.9: Kết quả sau khi chuyển dữ liệu từ excel lên cơ sở dữ liệu của MySQL (phần họ cây thuốc).....	74
Hình 4.10: Kết quả sau khi chuyển dữ liệu từ excel lên cơ sở dữ liệu của MySQL (phần cây thuốc) .....	75
Hình 4.11: Mô hình vector hóa dữ liệu dược tính.....	76
Hình 4.12: Kết quả thống kê số lần xuất hiện của mỗi thành phần hóa học.....	77
Hình 4.13: Kết quả trích xuất dữ liệu 10 thành phần hóa học cho mỗi cây thuốc trong excel .....	78
Hình 4.14: Các thành phần hóa học có giá trị khoảng $[a, b]$ .....	78
Hình 4.15: Giá trị nhỏ nhất trong tất cả các giá trị của các thành phần hóa học .....	79
Hình 4.16: Kết quả với 04 độ đo.....	80
Hình 4.17: Kết quả với 04 độ đo.....	81
Hình 4.18: Kết quả sau khi chuyển dữ liệu vector từ excel lên cơ sở dữ liệu Qdrant ....	83
Hình 5.1: Giao diện thanh header .....	84
Hình 5.2: Giao diện slider của trang chủ .....	85

Hình 5.3: Giao diện hiển thị một số dược liệu được tìm kiếm nhiều nhất.....	87
Hình 5.4: Giao diện hiển thị một số dược liệu quý của trang chủ .....	88
Hình 5.5: Giao diện mục “BẠN CÓ BIẾT” của trang chủ.....	89
Hình 5.6: Giao diện trang tra cứu .....	90
Hình 5.7: Giao diện trang chi tiết cây thuốc .....	92
Hình 5.8: Giao diện trang giới thiệu .....	94
Hình 5.9: Giao diện trang đăng nhập .....	95
Hình 5.10: Giao diện trang thống kê.....	96
Hình 5.11: Giao diện xem các họ cây thuốc .....	98
Hình 5.12: Giao diện thêm họ cây thuốc mới.....	99
Hình 5.13: Giao diện sửa họ cây thuốc.....	100
Hình 5.14: Giao diện xem các cây thuốc .....	102
Hình 5.15: Giao diện thêm cây thuốc mới.....	103
Hình 5.16: Giao diện sửa cây thuốc .....	104

## MỞ ĐẦU

Trong bối cảnh hiện đại hóa và phát triển của y học trong kỷ nguyên số, việc bảo tồn và phát triển các tri thức y học cổ truyền là một trong những mục tiêu quan trọng góp phần nâng cao sức khỏe cộng đồng. Khi nhắc đến y học cổ truyền, thì cây thuốc là một thành phần không thể thiếu, chúng đã được sử dụng từ rất lâu đời trên thế giới và ở Việt Nam nhằm chữa trị và phòng ngừa bệnh. Đặc biệt ở các quốc gia Châu Á, nơi mà khí hậu và thổ nhưỡng giúp nhiều cây dược liệu phát triển tốt.

Đã có hàng ngàn cây thuốc với các dược tính phong phú, việc tra cứu thông tin theo thành phần hoạt chất của các cây thuốc sẽ gặp nhiều khó khăn do khối lượng dữ liệu rất lớn, đồng thời các giá trị về thành phần hoạt chất là các giá trị thực có sai số nên không thể tra cứu theo dạng chính xác, và chi phí tra cứu về mặt thời gian sẽ cao. Việc giải quyết vấn đề này đã trở nên dễ dàng hơn rất nhiều trong thời đại tin học hóa hiện nay, khi mà chúng ta có thể áp dụng các phương pháp tiên tiến trong lĩnh vực dữ liệu để số hóa các tài liệu giấy và thiết kế các giải pháp tìm kiếm hiệu quả.

Với đề tài “Truy vấn cây dược liệu Việt Nam dựa trên độ tương đồng về dược tính”, chúng tôi hướng đến việc xây dựng một ứng dụng giúp người dùng có thể tra cứu, tìm kiếm các cây thuốc có dược tính tương tự dựa trên các thông số về thành phần hóa học. Hệ thống được xây dựng dựa trên nền tảng cơ sở dữ liệu vector, một phương thức cho phép truy vấn dữ liệu lớn với thông tin truy vấn có rất nhiều thông tin (gồm rất nhiều thành phần) và các thông tin này có thể truy vấn được với giá trị gần chính xác. Ngoài ra nhằm tạo thuận lợi cho người dùng, nhóm xây dựng một website trên nền tảng Angular và .Net, giúp người dùng có được trải nghiệm tiện lợi và dễ dàng khi sử dụng.

Khoá luận không chỉ giúp chúng tôi trang bị những kiến thức mới về lập trình và quản lý cơ sở dữ liệu, mà còn góp phần nâng cao hiệu quả trong việc nghiên cứu và ứng dụng cây thuốc vào y học hiện đại của người Việt. Chúng tôi hy vọng khoá luận đáp ứng nhu cầu tìm hiểu thông tin của người dùng về dược liệu Việt Nam theo thành phần dược tính.

## CHƯƠNG 1. TỔNG QUAN

### 1.1. Lý do chọn đề tài

Việt Nam có hàng ngàn loại cây thuốc với các dược tính phong phú, việc tra cứu thông tin về thành phần hoạt chất của các cây thuốc có dược tính tương đồng trên các tài liệu truyền thống gặp không ít khó khăn do khối lượng dữ liệu quá lớn. Đồng thời, các giá trị về thành phần hoạt chất là giá trị thực với sai số nhất định, khiến việc tra cứu chính xác trở nên bất khả thi và đòi hỏi nhiều thời gian cũng như công sức. Bài toán xây dựng ứng dụng truy vấn cây thuốc dựa trên độ tương đồng về dược tính rất cần thiết hiện nay.

Cơ sở dữ liệu vector là một trong những phương thức hiện nay cho phép lưu trữ và xử lý dữ liệu dưới dạng vector, giúp cải thiện hiệu quả trong việc tìm kiếm, phân tích và truy vấn các thông tin phức tạp. Nghiên cứu và ứng dụng cơ sở dữ liệu vector trong việc xây dựng ứng dụng tra cứu các cây thuốc có độ tương đồng về dược tính mang lại những lợi ích vượt trội sau:

- Tối ưu hóa tốc độ truy vấn: cơ sở dữ liệu vector giúp tăng tốc quá trình tìm kiếm, đặc biệt là khi cần phân tích và so sánh các loại cây thuốc có dược tính tương đồng trong một tập dữ liệu lớn.
- Nâng cao độ chính xác: Việc biểu diễn dữ liệu dưới dạng vector cho phép hệ thống phân tích mối quan hệ phức tạp giữa các thuộc tính của cây thuốc, từ đó đưa ra kết quả tìm kiếm chính xác hơn.
- Khả năng xử lý dữ liệu không gian nhiều chiều: cơ sở dữ liệu vector phù hợp với việc xử lý dữ liệu có nhiều thuộc tính, chẳng hạn như độ tương đồng về dược tính dựa trên thành phần hóa học giữa các loại cây, giúp cải thiện khả năng phân tích và tra cứu.
- Tiết kiệm thời gian và tài nguyên: Dựa vào cơ chế xử lý thông tin theo hướng “thông minh” nên vấn đề truy vấn sẽ hiệu quả về chi phí, kết quả, ...

Trong bối cảnh các nghiên cứu y học cổ truyền ngày càng được tin học hóa, đề tài “Truy vấn cây dược liệu Việt Nam dựa trên độ tương đồng về dược tính” không chỉ mang ý nghĩa thực tiễn mà còn đóng góp quan trọng vào việc phát triển các giải pháp công nghệ hỗ trợ tra cứu và ứng dụng cây thuốc trong y học hiện đại.

## 1.2. Mục đích đề tài

- Mô hình hóa dữ liệu được tính thành dữ liệu vector.
- Xây dựng cơ sở dữ liệu vector tương ứng.
- Xây dựng mô hình đánh giá độ tương đồng hai vector phục vụ giải quyết bài toán truy vấn dữ liệu vector.
- Thiết kế và xây dựng website dựa trên công nghệ mới hiện nay như Angular và .Net để hiện thực trình bày kết quả truy vấn bằng giao diện web cho người dùng.

## 1.3. Đối tượng và phạm vi nghiên cứu

### 1.3.1. Đối tượng nghiên cứu

Đề tài tập trung vào việc nghiên cứu và ứng dụng cơ sở dữ liệu vector cùng các công nghệ liên quan để xây dựng hệ thống truy vấn cây thuốc dựa trên độ tương đồng về dược tính. Các đối tượng nghiên cứu cụ thể bao gồm:

- Phương pháp cơ sở dữ liệu vector hỗ trợ xây dựng hệ thống thông tin dạng vector của các cây thuốc.
- Công nghệ Angular và .Net: Angular là công nghệ phổ biến để xây dựng giao diện người dùng động, trong khi .Net cung cấp nền tảng mạnh mẽ cho việc phát triển phía máy chủ. Sự kết hợp này sẽ giúp xây dựng một hệ thống quản lý và truy vấn cây thuốc có hiệu suất cao, giao diện thân thiện và dễ sử dụng cho người dùng.
- Mô hình hóa dữ liệu được tính thành dữ liệu vector dựa trên tài liệu “3033 Cây Thuốc Đông Y (Tuệ Tĩnh)”, đây là một nguồn cung cấp dữ liệu về các cây thuốc cho ứng dụng với độ tin cậy cao.

### 1.3.2. Phạm vi nghiên cứu

Phạm vi nội dung: Đề tài tập trung vào việc nghiên cứu, ứng dụng cơ sở dữ liệu vector và các kỹ thuật liên quan hỗ trợ bên cạnh đó, để ứng dụng vào đề tài nhằm hỗ trợ cho nền y học cổ truyền hiện đại trong việc tra cứu độ tương đồng về dược tính của các cây thuốc.

Phạm vi dữ liệu: 3033 cây thuốc Việt Nam, dựa trên tài liệu y học “3033 Cây Thuốc Đông Y (Tuệ Tĩnh)”.



Phạm vi thời gian: Đề tài được thực hiện trong khoảng thời gian 15 tuần học kể từ 02/09/2024 đến 15/12/2024, bao gồm các giai đoạn: nghiên cứu lý thuyết, thu thập yêu cầu, phân tích thiết kế, chuẩn bị dữ liệu, phát triển hệ thống, kiểm thử và đánh giá kết quả.

Phạm vi kỹ thuật: Sử dụng cơ sở dữ liệu vector để lưu trữ và truy vấn dữ liệu được tính cây thuốc. Sử dụng Angular để phát triển giao diện người dùng, đảm bảo tính trực quan và dễ sử dụng. Áp dụng .Net làm nền tảng phát triển backend. Triển khai hệ thống trên nền tảng web, đảm bảo tương thích nhiều thiết bị như máy tính, máy tính bảng và thiết bị di động.

#### **1.4. Phương pháp nghiên cứu**

Phương pháp nghiên cứu lý thuyết: Nguyên cứu thông tin trong tài liệu “3033 Cây Thuốc Đông Y” của thầy Tuệ Tĩnh và các lý thuyết về cơ sở dữ liệu vector, công nghệ .Net để ứng dụng vào việc xây dựng một ứng dụng hỗ trợ truy vấn thông tin các cây thuốc có độ tương đồng về dược tính.

Phương pháp nghiên cứu thực nghiệm: Để xây dựng hệ thống tra cứu cây thuốc Việt Nam dựa trên độ tương đồng về dược tính. Đầu tiên, chúng tôi thực hiện thu thập dữ liệu từ các nguồn tài liệu y học cổ truyền về cây thuốc. Tiếp theo, áp dụng cơ sở dữ liệu vector và thiết kế các thuật toán cần thiết để ứng dụng vào việc phân tích và đánh giá độ tương đồng dược tính. Dữ liệu sau khi xử lý được lưu trữ trong cơ sở dữ liệu từ đó phát triển giao diện truy vấn cho phép người dùng nhập thông tin về cây thuốc để cho ra kết quả. Cuối cùng, website sẽ được kiểm thử với các trường hợp thực tế để đảm bảo tính chính xác và hiệu quả của hệ thống.

#### **1.5. Tổng quan tình hình nghiên cứu trong và ngoài nước**

Tại Việt Nam, các nghiên cứu ứng dụng công nghệ thông tin trong việc quản lý và tra cứu các dược liệu ngày càng được quan tâm. Tuy nhiên, phần lớn các nghiên cứu chủ yếu dừng lại ở việc triển khai cơ sở dữ liệu về cây thuốc và mô tả những đặc tính dược lý và cho phép tìm kiếm theo tên mà chưa cho phép truy vấn theo đặc tính dược liệu như: “duoclieuvietnam.com.vn”, “tracuuduoclieu.vn”, ....

Trên thế giới, nhiều quốc gia đã triển khai và phát triển các cơ sở dữ liệu về dược liệu truyền thống như Trung Quốc, Ấn Độ, Hàn Quốc,... nơi có nền y học cổ truyền phát triển mạnh mẽ. Các cơ sở dữ liệu như Traditional Chinese Medicine Integrative Database (TCMID) và Traditional Chinese Medicine Database and Information System (TCM-ID) đã ứng dụng các công nghệ tiên tiến để xây dựng hệ thống lưu trữ và truy xuất thông tin về cây thuốc, hỗ trợ quá trình nghiên cứu và phát triển thuốc mới. Các nghiên cứu quốc tế cũng cho thấy việc sử dụng cơ sở dữ liệu vector giúp tối ưu hóa khả năng phân tích và tìm kiếm, nhất là trong các bài toán cần đánh giá độ tương đồng về dược tính giữa các loại cây.

Vì vậy, đề tài “Truy vấn cây dược liệu Việt Nam dựa trên độ tương đồng về dược tính” của chúng tôi không chỉ kế thừa từ các nghiên cứu đã có mà còn áp dụng phương pháp cơ sở dữ liệu vector để tối ưu hóa hệ thống truy vấn, góp phần nâng cao hiệu quả và độ chính xác trong quá trình tra cứu, hỗ trợ phát triển y học hiện đại dựa trên nền tảng tri thức y học cổ truyền.

## 1.6. Công nghệ sử dụng

Hệ thống tra cứu dược liệu dựa trên độ tương đồng về dược tính sẽ được xây dựng bằng cách tích hợp các công nghệ tiên tiến để đảm bảo hiệu suất và tốc độ cao trong quá trình truy vấn dữ liệu:

- Front-end: Giao diện người dùng sẽ được phát triển bằng Angular 18, một thư viện javascript mạnh mẽ giúp xây dựng ứng dụng web đơn trang (SPA) với khả năng tương tác cao. Kết hợp với Bootstrap 5 để thiết kế giao diện trực quan, thân thiện và tương thích cho nhiều thiết bị từ máy tính đến di động, giúp người dùng dễ dàng tra cứu thông tin về các loại cây thuốc.
- Back-end: ASP.NET Core API sẽ được sử dụng để phát triển API cung cấp các dịch vụ cho hệ thống, đảm bảo khả năng xử lý nhanh chóng và độ bảo mật cao.
- Cơ sở dữ liệu: Sử dụng cơ sở dữ liệu vector Qdrant để lưu trữ các vector biểu diễn các giá trị thành phần hóa học của một cây thuốc và hỗ trợ trong việc truy vấn. Song song đó, chúng tôi còn sử dụng thêm một cơ sở dữ liệu quan hệ MySQL để lưu trữ các thông tin quan hệ về cây thuốc như họ cây thuốc, tên thuốc, hình ảnh,

công dụng, nơi sinh sống, bộ phận dùng,... sự kết hợp giữa hai loại cơ sở dữ liệu có thể giải quyết dễ dàng bài toán đang đặt ra.

## CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 2.1. Phân tích yêu cầu

#### 2.1.1. Yêu cầu chức năng

- *Tìm kiếm các cây thuốc tương đồng dựa trên n thành phần hóa học:* Cho phép người dùng nhập vào n thành phần hóa học và hệ thống sẽ thực hiện tìm các vector cây thuốc gần kề (tương đồng) trong cơ sở dữ liệu để trả về danh sách các cây thuốc tương ứng. Tuy nhiên vì cấu hình máy chủ của chúng tôi có giới hạn vì vậy chúng tôi chọn  $n = 10$ . Có thể tăng lên nếu cấu hình máy chủ đủ mạnh.
- *Xem thông tin chi tiết của cây thuốc:* Sau khi tìm kiếm người dùng có thể xem các thông tin chi tiết của từng cây thuốc (bao gồm: tên tiếng Việt, tên khoa học, hình ảnh, công dụng, thành phần hóa học,...) bằng cách nhấn vào hình của cây đó.
- *Gợi ý cây thuốc tương tự:* Khi người dùng tra cứu thông tin về một cây thuốc cụ thể, hệ thống sẽ cho phép người dùng xem các cây thuốc có vector gần kề dựa trên các thành phần hóa học tương đồng.
- *Các chức năng dành riêng cho người quản trị:*
  - *Đăng nhập:* Quản trị viên sẽ được cung cấp một tài khoản có vai trò là người quản trị để đăng nhập vào hệ thống quản trị.
  - *Thống kê:* Người quản trị có thể nhìn thấy các thống kê về số lượng các họ cây thuốc, cây thuốc, các cây thuốc được tìm kiếm nhiều nhất.
  - *Quản lý họ cây thuốc:* Người quản trị có thể thực hiện các thao tác như tìm kiếm, thêm, xóa, sửa họ cây thuốc.
  - *Quản lý cây thuốc:* Người quản trị có thể thực hiện các thao tác như tìm kiếm, thêm, xóa, sửa cây thuốc.
- *Giao diện người dùng thân thiện:* Hệ thống cần có giao diện thân thiện, dễ sử dụng, cho phép người dùng tra cứu thông tin một cách dễ dàng, đặc biệt là chức năng tìm kiếm tương đồng và kết quả tìm kiếm trực quan.

#### 2.1.2. Yêu cầu phi chức năng

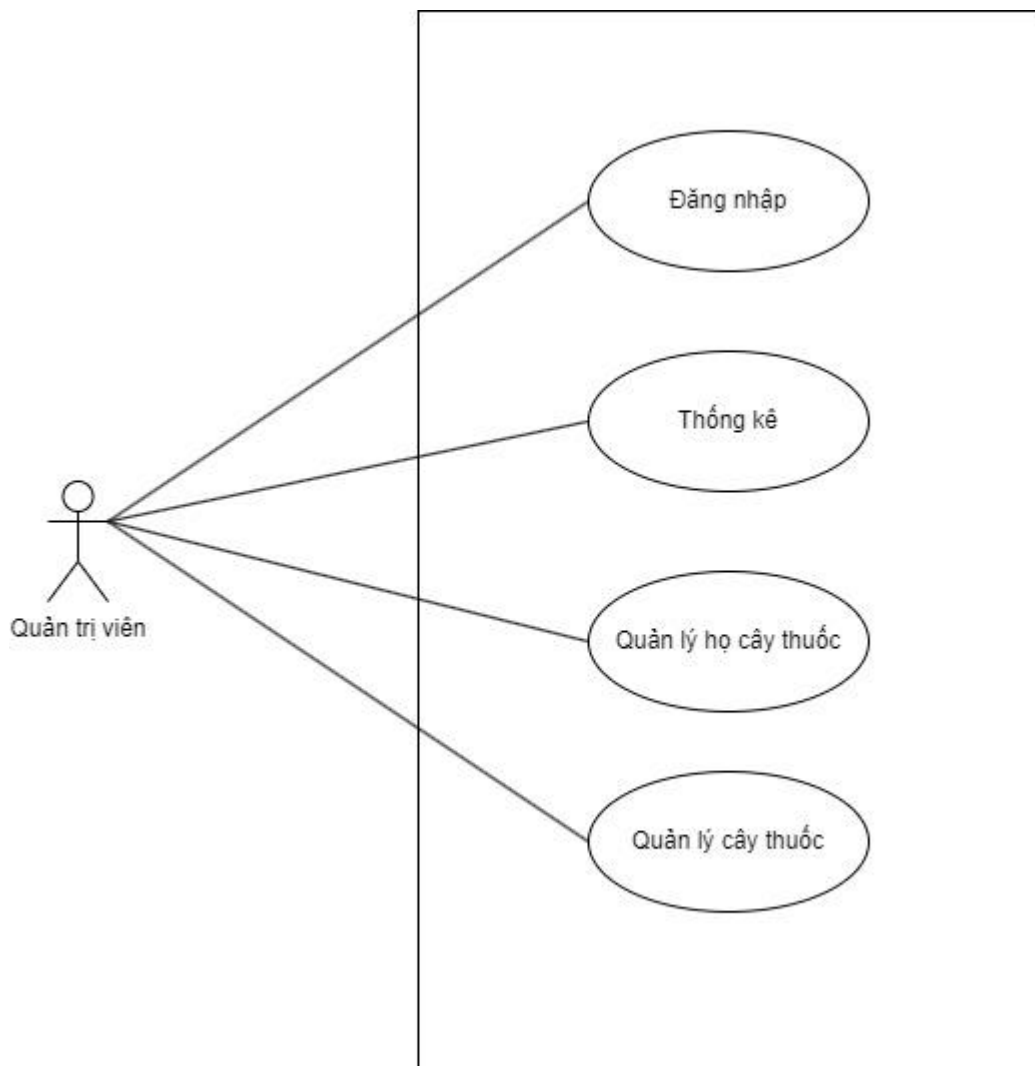
- *Tính chính xác:* Kết quả trả về từ việc tìm kiếm phải có độ chính xác cao, đảm bảo các cây thuốc được trả về là những lựa chọn phù hợp với yêu cầu của người dùng.

- *Hiệu suất*: Hệ thống phải đáp ứng được các yêu cầu về thời gian xử lý truy vấn nhanh chóng, đặc biệt khi tra cứu các cây thuốc tương đồng. Thời gian phản hồi cần phải dưới 1 giây cho một truy vấn.
- *Tính sẵn sàng*: Hệ thống phải đảm bảo khả năng hoạt động liên tục với thời gian downtime tối thiểu.
- *Khả năng mở rộng*: Hệ thống phải có khả năng xử lý lượng lớn dữ liệu và các vector hóa học mà không làm giảm hiệu suất, cho phép mở rộng từ vài trăm đến hàng triệu bản ghi.

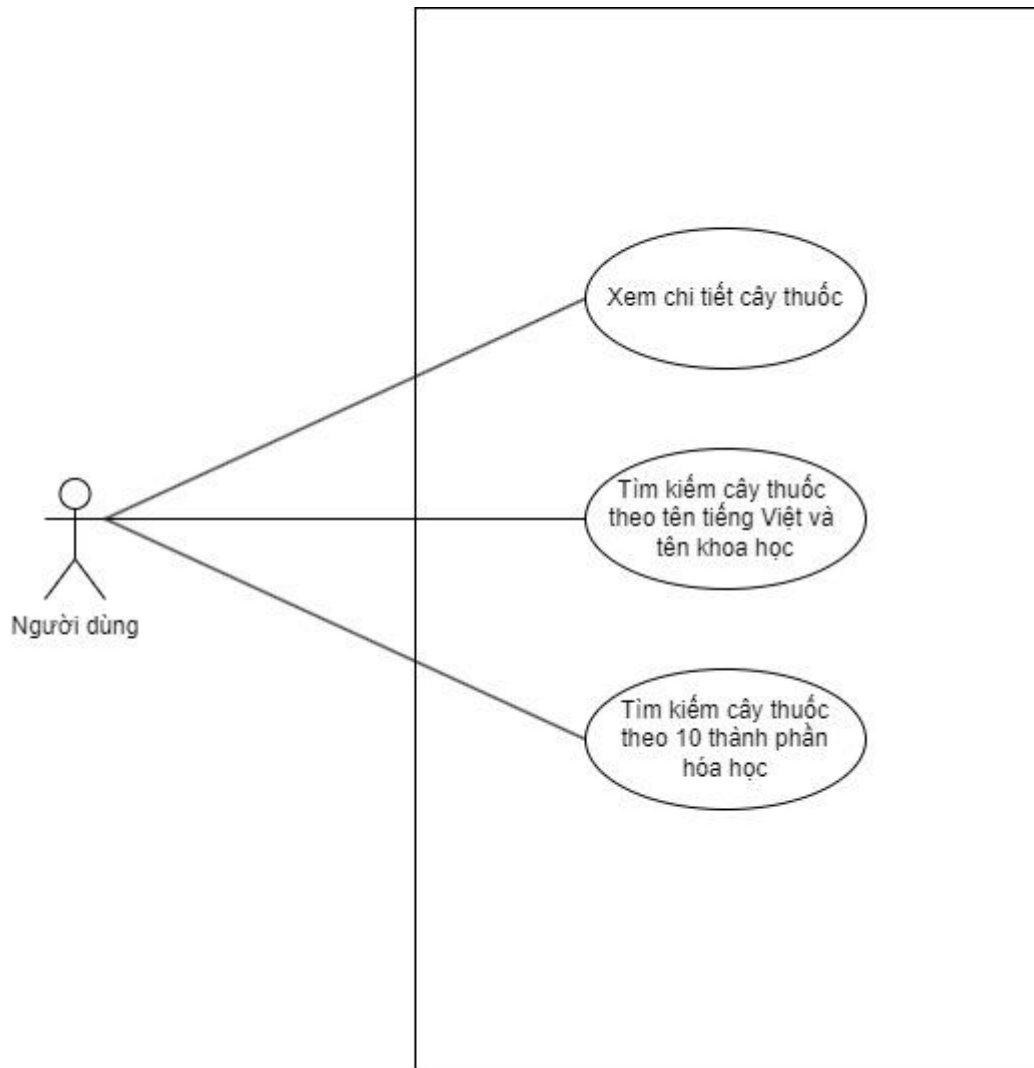
## 2.2. Mô hình hóa yêu cầu

### 2.2.1. Usecase

#### 2.2.1.1. Usecase tổng quát

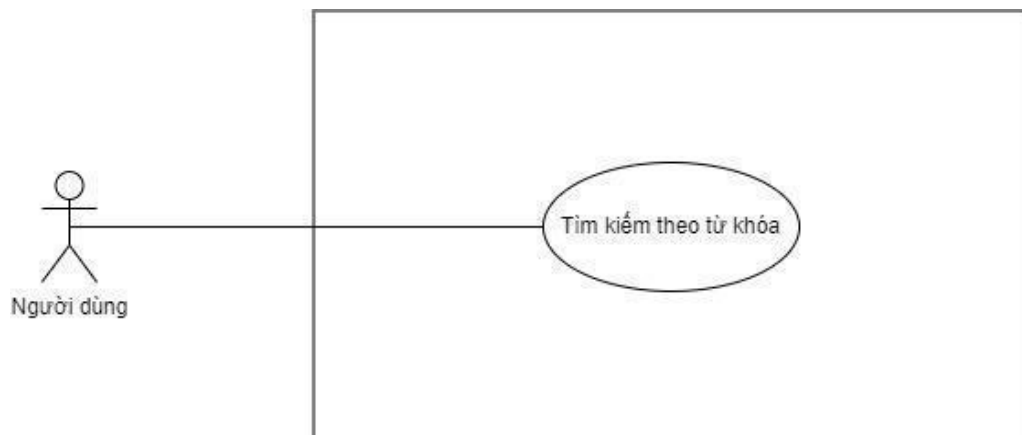


Hình 2.1: Usecase tổng quát ở vai trò quản trị viên



Hình 2.2: Usecase tổng quát ở vai trò người dùng

#### 2.2.1.2. Usecase tìm kiếm cây thuốc theo từ khóa

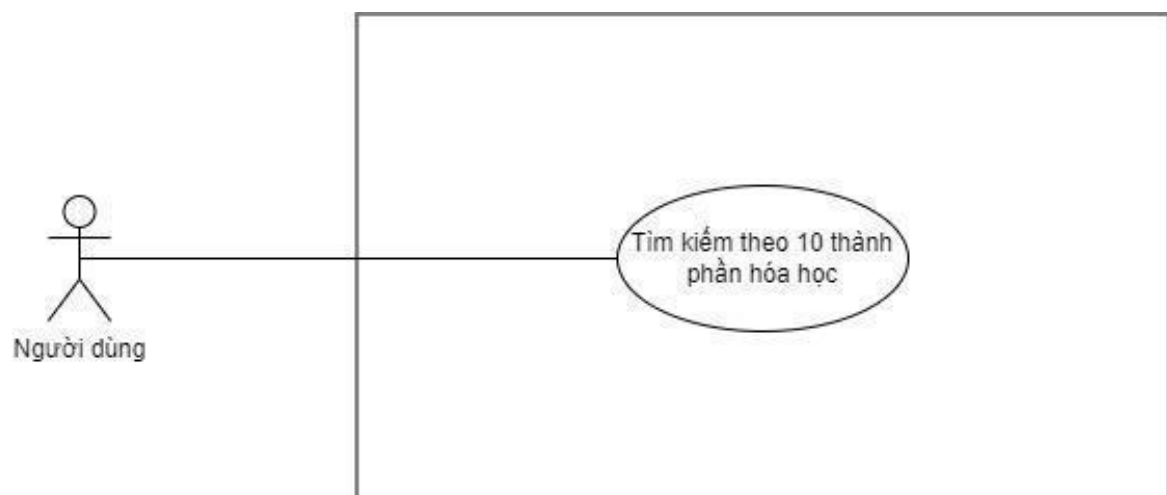


Hình 2.3: Usecase tìm kiếm cây thuốc theo từ khóa

Bảng 2.1: Đặc tả usecase tìm kiếm theo từ khóa

Đặc tả use case tìm kiếm theo từ khóa	
<b>ID</b>	UC.TK
<b>Tóm tắt</b>	Người dùng muốn tìm kiếm cây thuốc, thực hiện thao tác nhập từ khóa (tên Việt Nam, tên khoa học của cây thuốc) vào ô tìm kiếm sau đó ấn vào nút “Tra cứu”.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhập từ khóa vào ô tìm kiếm.</li> <li>2. Người dùng nhấn nút “Tra cứu”.</li> <li>3. Hệ thống lấy nội dung từ khóa.</li> <li>4. Hệ thống gửi yêu cầu đến máy chủ.</li> <li>5. Hệ thống nhận kết quả dạng JSON để hiển thị trên giao diện.</li> </ol>
<b>Dòng sự kiện phụ</b>	<ul style="list-style-type: none"> <li>• Người dùng không nhập từ khóa và nhấn nút “Tra cứu”. Hệ thống báo lỗi và yêu cầu người dùng phải nhập nội dung tìm kiếm trước khi nhấn nút “Tra cứu”.</li> </ul>
<b>Tiền điều kiện</b>	
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>• Hệ thống hiển thị các cây thuốc tìm kiếm được theo từ khóa có phân trang.</li> </ul>

### 2.2.1.3. Usecase tìm kiếm các cây thuốc tương đồng theo 10 thành phần hóa học

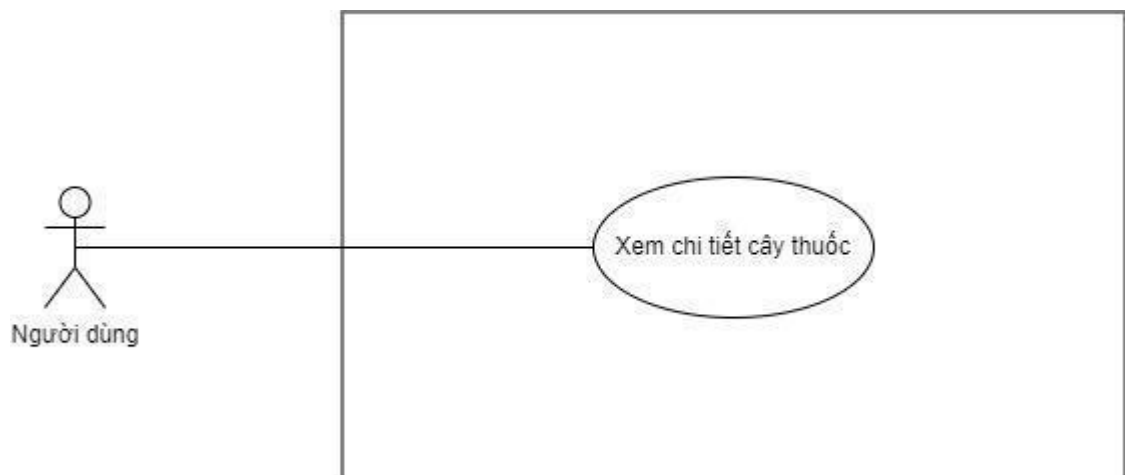


Hình 2.4: Usecase tìm kiếm cây thuốc tương đồng theo 10 thành phần hóa học

Bảng 2.2: Đặc tả usecase tìm kiếm theo 10 thành phần hóa học

Đặc tả use case tìm kiếm theo 10 thành phần hóa học	
<b>ID</b>	UC.TKNC
<b>Tóm tắt</b>	Người dùng muốn tìm kiếm cây thuốc, thực hiện thao tác nhập giá trị của 10 thành phần hóa học sau đó ấn vào nút “Tìm kiếm”.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhập giá trị của 10 thành phần hóa học.</li> <li>2. Người dùng bấm nút “Tìm kiếm”.</li> <li>3. Hệ thống lấy giá trị 10 thành phần hóa học và chuyển thành dạng vector 10 chiều.</li> <li>4. Hệ thống gửi yêu cầu đến máy chủ.</li> <li>5. Hệ thống nhận kết quả top 10 cây thuốc gần đúng với yêu cầu tìm kiếm ở dạng JSON để hiển thị trên giao diện.</li> </ol>
<b>Dòng sự kiện phụ</b>	
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>• Người dùng ở trang tra cứu dược liệu.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>• Hệ thống hiển thị 10 cây thuốc kết quả có phân trang.</li> </ul>

#### 2.2.1.4. Usecase xem chi tiết cây thuốc



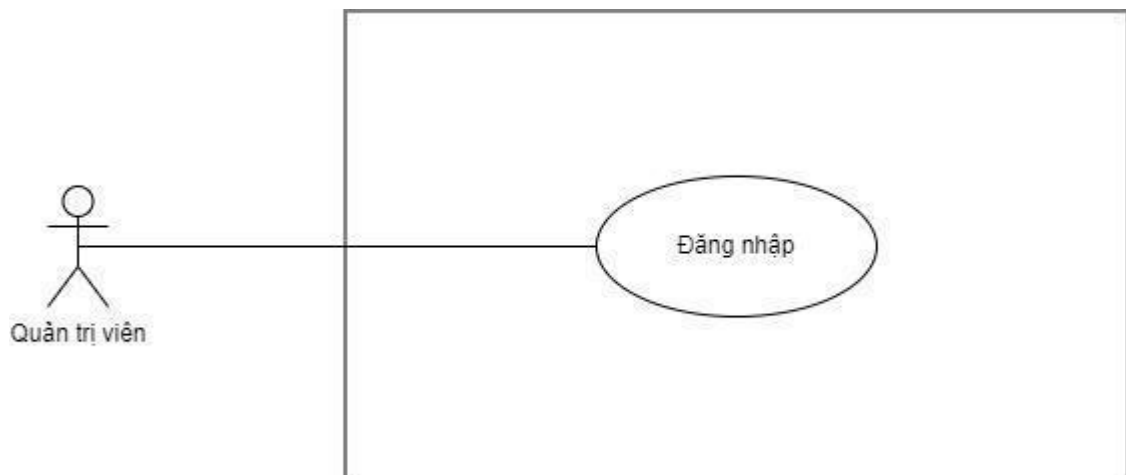
Hình 2.5: Usecase xem chi tiết cây thuốc



Bảng 2.3: Đặc tả usecase xem chi tiết cây thuốc

Đặc tả use case xem chi tiết cây thuốc	
<b>ID</b>	UC.XCTCT
<b>Tóm tắt</b>	Người dùng xem các thông tin chi tiết của một cây thuốc.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhấn vào cây thuốc cần xem thông tin chi tiết.</li> <li>2. Hệ thống gửi yêu cầu lấy thông tin chi tiết của cây thuốc đến máy chủ.</li> <li>3. Hệ thống nhận kết quả chi tiết của cây thuốc ở dạng JSON để hiển thị trên giao diện.</li> </ol>
<b>Dòng sự kiện phụ</b>	
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>• Người dùng ở trang chủ, trang tra cứu dược liệu hoặc trang giới thiệu.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>• Hệ thống hiển thị thông tin chi tiết cây thuốc lên giao diện để người dùng có thể thực hiện các thao tác tiếp theo.</li> </ul>

#### 2.2.1.5. Usecase đăng nhập

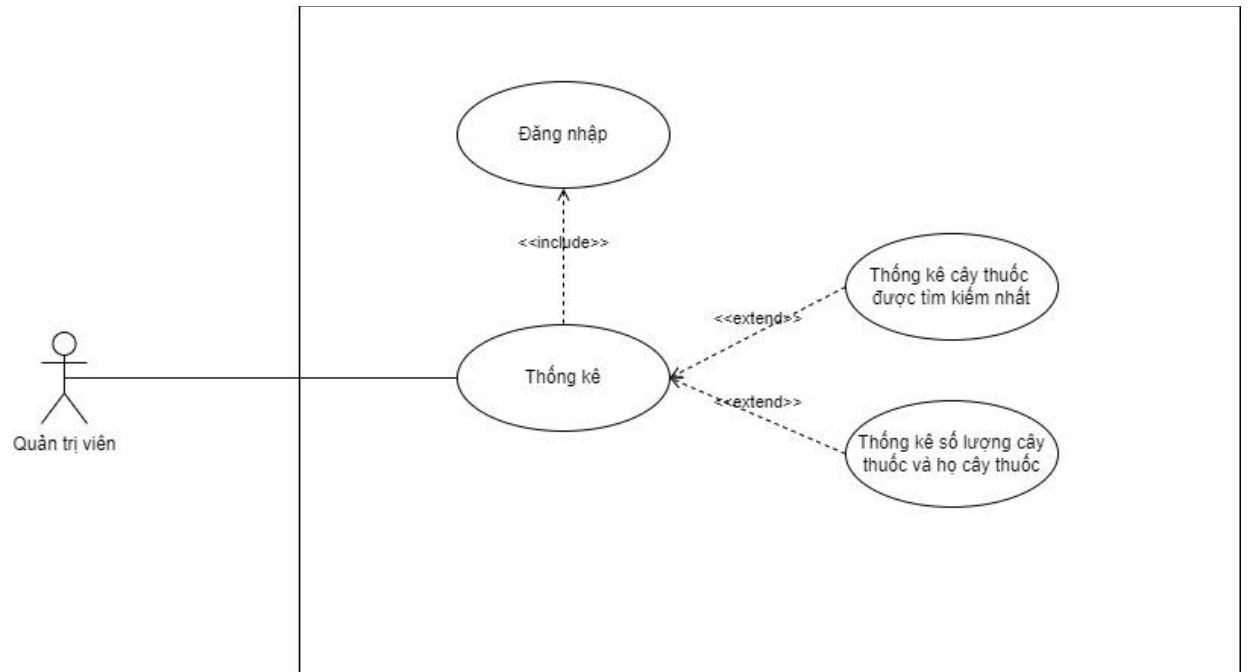


Hình 2.6: Usecase đăng nhập

Bảng 2.4: Đặc tả usecase đăng nhập

<b>Đặc tả use case đăng nhập</b>	
<b>ID</b>	UC.DN
<b>Tóm tắt</b>	Người dùng muốn vào trang quản trị thì phải đăng nhập.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhấn vào nút “Đi đến trang quản trị”.</li> <li>2. Hệ thống hiện ra giao diện đăng nhập.</li> <li>3. Người dùng nhập email và mật khẩu.</li> <li>4. Người dùng bấm đăng nhập.</li> <li>5. Hệ thống kiểm tra tài khoản và mật khẩu có hợp lệ không.</li> <li>6. Hệ thống chuyển hướng đến trang quản trị và cho phép người dùng tương tác với các chức năng quản trị.</li> </ol>
<b>Dòng sự kiện phụ</b>	<ol style="list-style-type: none"> <li>1. Người dùng không nhập đầy đủ các trường, hệ thống thông báo lỗi.</li> <li>2. Người dùng không nhập đúng tài khoản mật khẩu, hệ thống thông báo lỗi.</li> </ol>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>• Người dùng chưa đăng nhập.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>• Người dùng đăng nhập thành công vào hệ thống, hệ thống điều hướng đến trang quản trị.</li> </ul>

### 2.2.1.6. Usecase thống kê

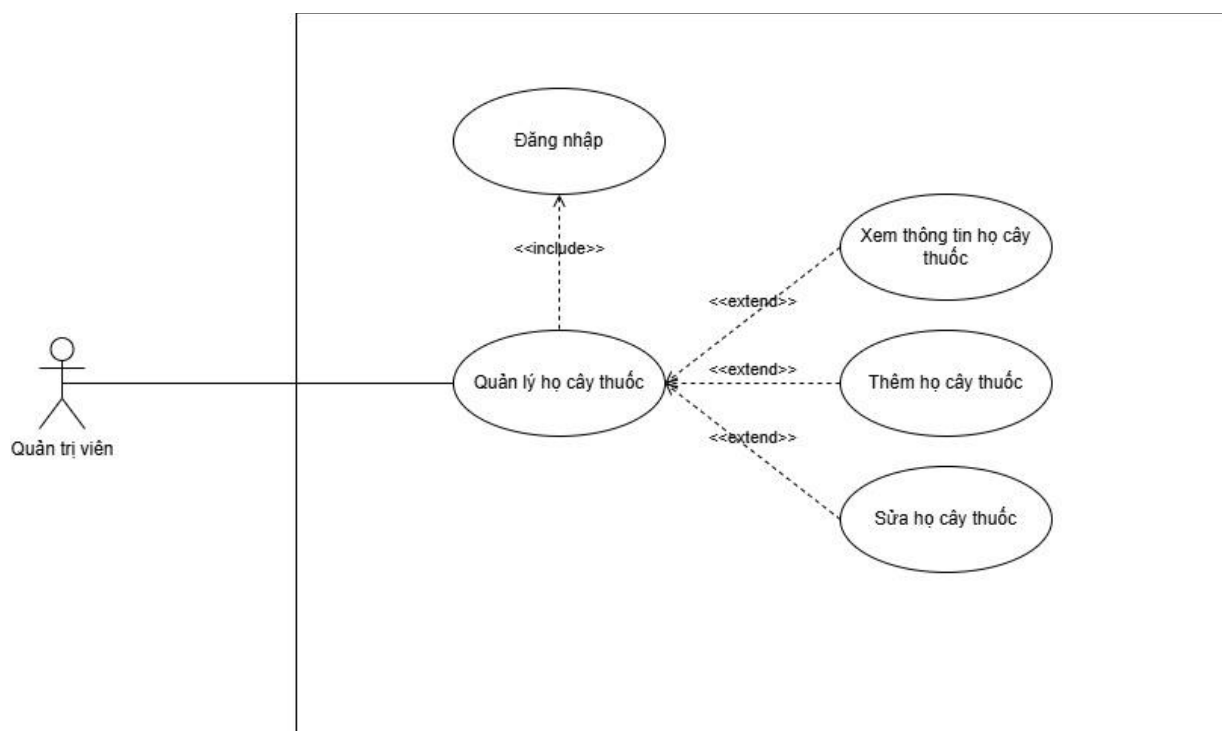


Hình 2.7: Usecase thống kê

Bảng 2.5: Đặc tả usecase thống kê

Đặc tả use case thống kê	
<b>ID</b>	UC.TK
<b>Tóm tắt</b>	Người dùng muốn xem các thống kê về số lượng họ cây thuốc, cây thuốc và các cây thuốc được tìm kiếm nhiều nhất.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhấn vào nút “Thống kê”.</li> <li>2. Hệ thống gửi yêu cầu đến máy chủ để lấy danh sách các cây thuốc, họ cây thuốc.</li> <li>3. Hệ thống trên front-end nhận kết quả trả về dưới dạng JSON và thực hiện các chức năng thống kê số lượng cây thuốc, họ cây thuốc và các cây thuốc được tìm kiếm nhiều nhất.</li> </ol>
<b>Dòng sự kiện phụ</b>	
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>• Người dùng đăng nhập vào hệ thống quản trị.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>• Hệ thống hiển thị các kết quả thống kê lên giao diện.</li> </ul>

### 2.2.1.7. Usecase quản lý họ cây thuốc



Hình 2.8: Usecase quản lý họ cây thuốc

Bảng 2.6: Đặc tả usecase xem thông tin họ cây thuốc

Đặc tả use case xem thông tin họ cây thuốc	
<b>ID</b>	UC.XTTHT
<b>Tóm tắt</b>	Người dùng muốn xem thông tin các họ cây thuốc hiện có trong hệ thống.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhấn vào mục “Quản lý họ cây thuốc”.</li> <li>2. Hệ thống gửi yêu cầu lấy danh sách các họ cây thuốc đến máy chủ.</li> <li>3. Hệ thống nhận kết danh sách các họ cây thuốc ở dạng JSON để hiển thị trên giao diện.</li> </ol>
<b>Dòng sự kiện phụ</b>	
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>• Người dùng đăng nhập vào hệ thống quản trị.</li> </ul>

<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>Hệ thống hiển thị danh sách các họ cây thuốc hiện có trong hệ thống lên giao diện để người dùng thực hiện các thao tác tiếp theo.</li> </ul>
----------------------	---

*Bảng 2.7: Đặc tả usecase thêm họ cây thuốc*

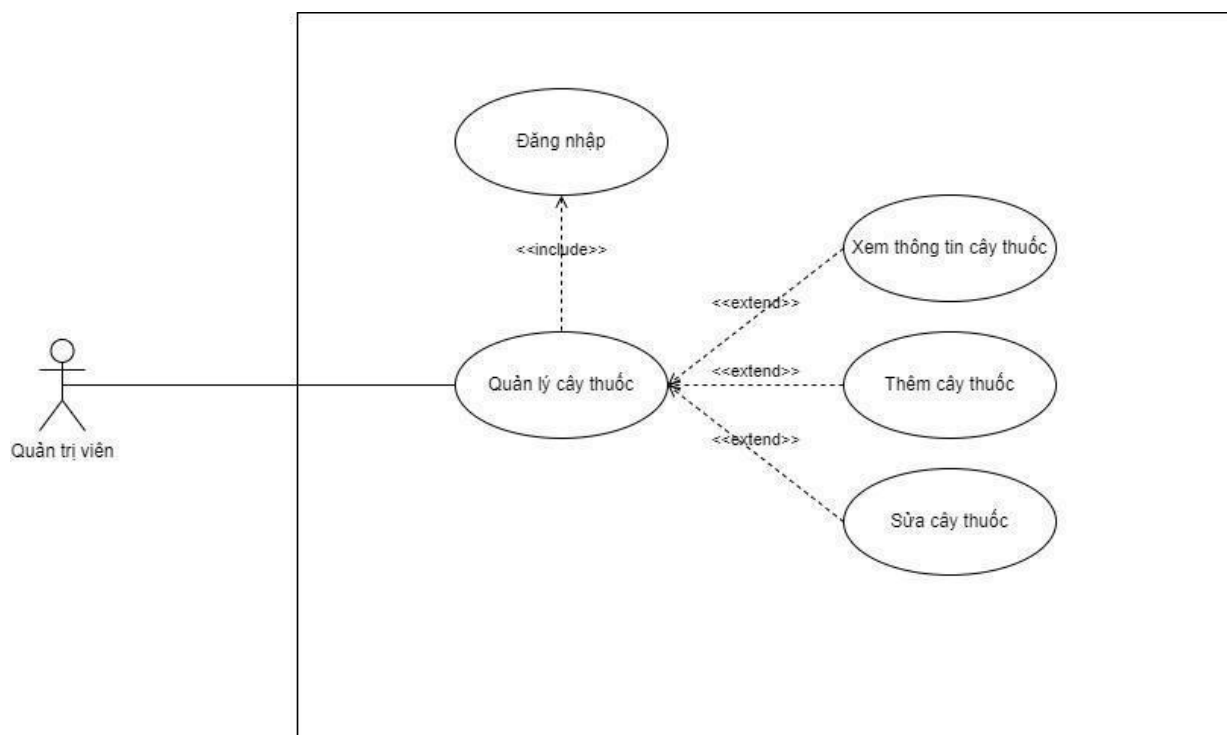
<b>Đặc tả use case thêm họ cây thuốc</b>	
<b>ID</b>	UC.THT
<b>Tóm tắt</b>	Người dùng muốn thêm một họ cây thuốc mới.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>Người dùng nhấn vào nút “Thêm họ cây thuốc”.</li> <li>Hệ thống hiển thị giao diện để người dùng nhập các thông tin tên khoa học, tên tiếng việt của họ cây thuốc.</li> <li>Người dùng nhập đầy đủ các thông tin theo yêu cầu và nhấn nút “Lưu”.</li> <li>Hệ thống gửi yêu cầu thêm họ cây thuốc mới đến máy chủ để tiến hành thêm họ cây thuốc mới vào hệ thống.</li> </ol>
<b>Dòng sự kiện phụ</b>	<ul style="list-style-type: none"> <li>Người dùng nhập không đầy đủ thông tin và nhấn “Lưu”, hệ thống báo lỗi.</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>Người dùng đăng nhập vào hệ thống quản trị.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>Hệ thống hiển thị danh sách các họ cây thuốc được cập nhật mới nhất (bao gồm cả họ cây thuốc vừa thêm) lên giao diện để người dùng thao tác.</li> </ul>

*Bảng 2.8: Đặc tả usecase sửa họ cây thuốc*

<b>Đặc tả use case sửa họ cây thuốc</b>	
<b>ID</b>	UC.SHT
<b>Tóm tắt</b>	Người dùng muốn sửa một họ cây thuốc.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>Người dùng tìm kiếm họ cây thuốc cần sửa.</li> <li>Người dùng nhấn vào biểu tượng hình cây viết trên dòng hiển thị họ cây thuốc.</li> <li>Hệ thống hiển thị giao diện để người dùng cập nhật các thông tin tên khoa học, tên tiếng việt của họ cây thuốc.</li> <li>Người dùng cập nhật đầy đủ các thông tin của họ cây thuốc và nhấn nút “Lưu”.</li> </ol>

	5. Hệ thống gửi yêu cầu sửa họ cây thuốc đến máy chủ để tiến hành sửa họ cây thuốc trong hệ thống.
<b>Dòng sự kiện phụ</b>	<ul style="list-style-type: none"> <li>Người dùng cập nhật không đầy đủ thông tin và nhấn “Lưu”, hệ thống báo lỗi.</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>Người dùng đăng nhập vào hệ thống quản trị.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>Hệ thống hiển thị danh sách các họ cây thuốc được cập nhật mới nhất (bao gồm cả họ cây thuốc vừa sửa) lên giao diện để người dùng thao tác.</li> </ul>

### 2.2.1.8. Usecase quản lý cây thuốc



Hình 2.9: Usecase quản lý cây thuốc

Bảng 2.9: Đặc tả usecase xem thông tin cây thuốc

Đặc tả use case xem thông tin cây thuốc	
<b>ID</b>	UC.XTTCT
<b>Tóm tắt</b>	Người dùng muốn xem thông tin các cây thuốc hiện có trong hệ thống.

<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhấn vào mục “Quản lý cây thuốc”.</li> <li>2. Hệ thống gửi yêu cầu lấy danh sách các cây thuốc đến server.</li> <li>3. Hệ thống nhận kết danh sách các cây thuốc ở dạng JSON để hiển thị trên giao diện.</li> </ol>
<b>Dòng sự kiện phụ</b>	
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>• Người dùng đăng nhập vào hệ thống quản trị.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>• Hệ thống hiển thị danh sách các cây thuốc hiện có trong hệ thống lên giao diện để người dùng thực hiện các thao tác tiếp theo.</li> </ul>

*Bảng 2.10: Đặc tả usecase thêm cây thuốc*

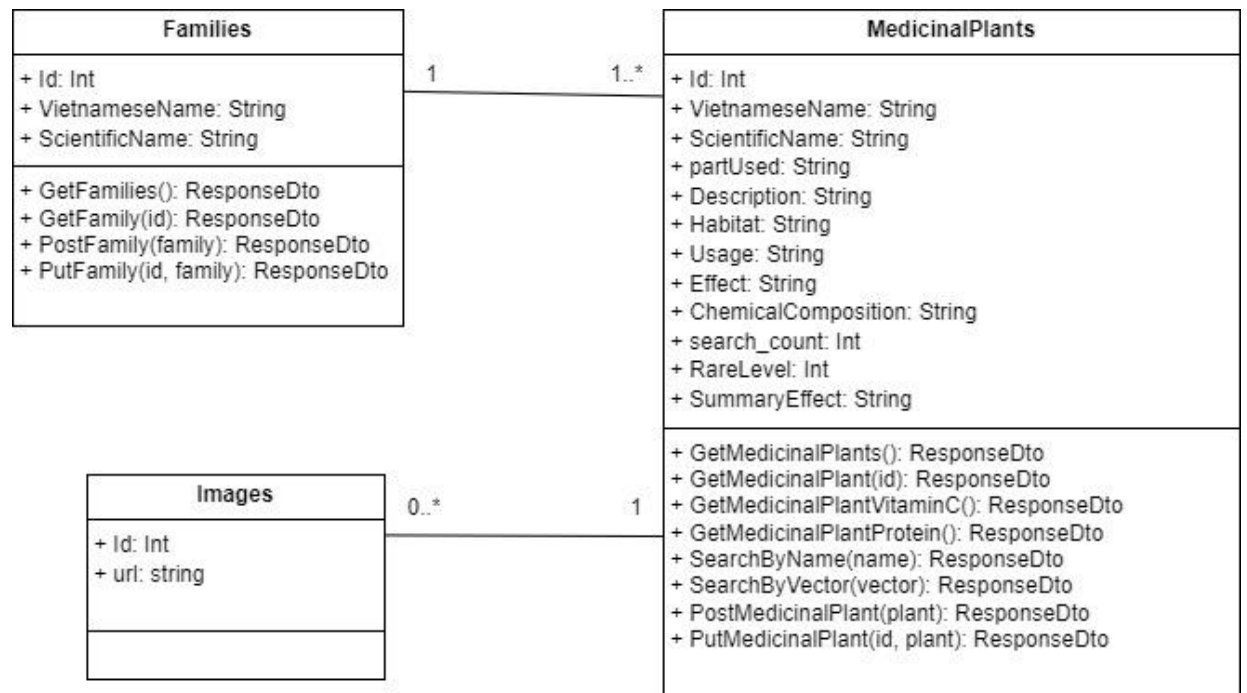
<b>Đặc tả use case thêm cây thuốc</b>	
<b>ID</b>	UC.TCT
<b>Tóm tắt</b>	Người dùng muốn thêm một cây thuốc mới.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng nhấn vào nút “Thêm cây thuốc”.</li> <li>2. Hệ thống hiển thị giao để người dùng nhập các thông tin của cây thuốc.</li> <li>3. Người dùng nhập đầy đủ các thông tin theo yêu cầu và nhấn nút “Lưu”.</li> <li>4. Hệ thống gửi yêu cầu thêm cây thuốc mới đến máy chủ để tiến hành thêm cây thuốc mới vào hệ thống, đồng thời chuyển các thành phần hóa học của cây thuốc thành vector và lưu vào cơ sở dữ liệu Qdrant.</li> </ol>
<b>Dòng sự kiện phụ</b>	<ul style="list-style-type: none"> <li>• Người dùng nhập không đầy đủ thông tin và nhấn “Lưu”, hệ thống báo lỗi.</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>• Người dùng đăng nhập vào hệ thống quản trị.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>• Hệ thống hiển thị danh sách các cây thuốc được cập nhật mới nhất (bao gồm cả cây thuốc vừa thêm) lên giao diện để người dùng thao tác.</li> </ul>

*Bảng 2.11: Đặc tả usecase sửa cây thuốc*

<b>Đặc tả use case sửa cây thuốc</b>
--------------------------------------

<b>ID</b>	UC.SCT
<b>Tóm tắt</b>	Người dùng muốn sửa một cây thuốc.
<b>Dòng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng tìm kiếm cây thuốc cần sửa.</li> <li>2. Người dùng nhấn vào biểu tượng hình cây viết trên dòng hiển thị cây thuốc.</li> <li>3. Hệ thống hiển thị giao diện để người dùng cập nhật các thông tin của cây thuốc.</li> <li>4. Người dùng cập nhật đầy đủ các thông tin của cây thuốc và nhấn nút “Lưu”.</li> <li>5. Hệ thống gửi yêu cầu sửa cây thuốc đến máy chủ để tiến hành sửa cây thuốc trong hệ thống.</li> </ol>
<b>Dòng sự kiện phụ</b>	<ul style="list-style-type: none"> <li>• Người dùng cập nhật không đầy đủ thông tin và nhấn “Lưu”, hệ thống báo lỗi.</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>• Người dùng đăng nhập vào hệ thống quản trị.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>• Hệ thống hiển thị danh sách các cây thuốc được cập nhật mới nhất (bao gồm cả cây thuốc vừa sửa) lên giao diện để người dùng thao tác.</li> </ul>

### 2.2.2. Sơ đồ lớp



Hình 2.10: Sơ đồ lớp

Lớp Families



Bảng 2.12: Bảng mô tả các thuộc tính của lớp Families

STT	Tên	Phạm vi	Kiểu dữ liệu	Ý nghĩa
1	Id	Public	Int	Mã họ cây thuốc.
2	VietnameseName	Public	String	Tên tiếng Việt họ cây thuốc.
3	ScientificName	Public	String	Tên khoa học họ cây thuốc.

Bảng 2.13: Bảng mô tả các phương thức của lớp Families

STT	Tên	Phạm vi	Mô tả	Đầu vào	Đầu ra
1	GetFamilies()	public	Lấy thông tin của tất cả cây thuốc.		Danh sách cây thuốc.
2	GetFamily()	public	Lấy thông tin của họ cây thuốc.	Mã họ cây thuốc.	Thông tin họ cây thuốc.
3	PostFamily()	public	Thêm họ cây thuốc mới vào CSDL.	Thông tin họ cây thuốc mới.	
4	PutFamily()	public	Sửa thông tin họ cây thuốc.	Thông tin họ cây thuốc cần sửa.	

### Lớp Images

Bảng 2.14: Bảng mô tả các thuộc tính của lớp Images

STT	Tên	Phạm vi	Kiểu dữ liệu	Ý nghĩa
1	Id	Public	Int	Mã hình ảnh.
2	url	Public	String	Đường dẫn hình ảnh cây thuốc.

### Lớp MedicinalPlants

Bảng 2.15: Bảng mô tả các thuộc tính của lớp MedicinalPlants

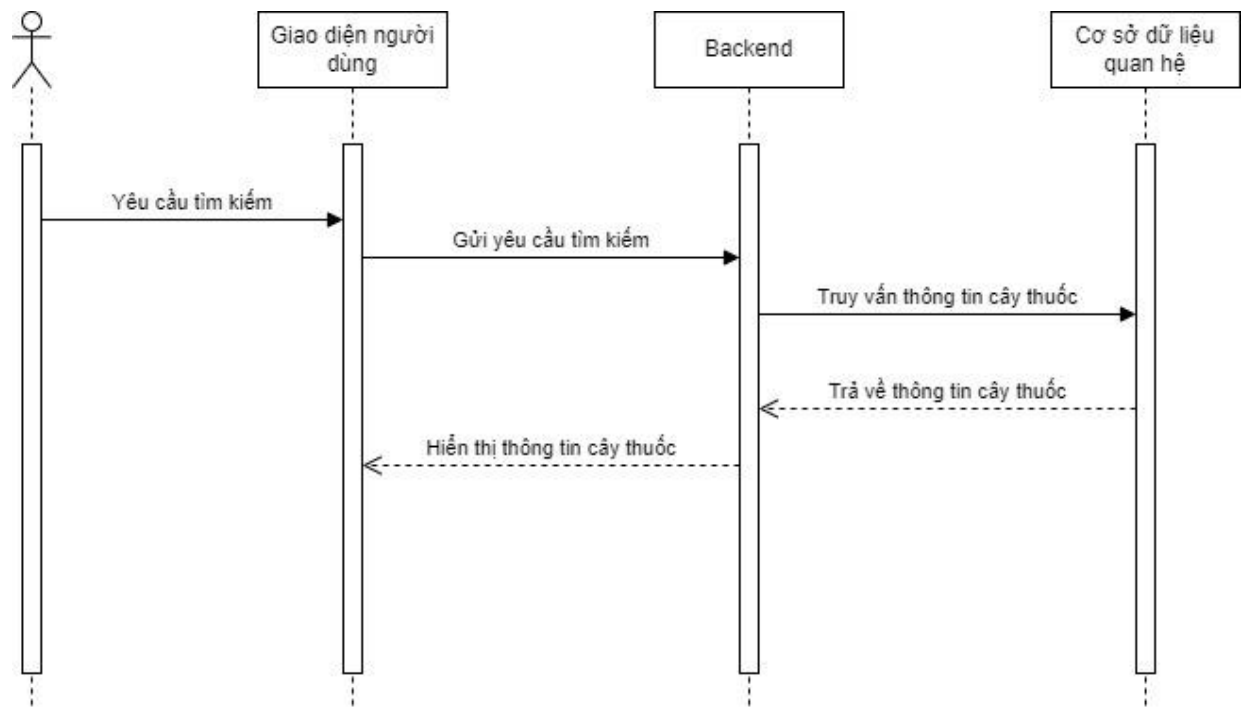
STT	Tên	Phạm vi	Kiểu dữ liệu	Ý nghĩa
1	Id	Public	Int	Mã cây thuốc.
2	VietnameseName	Public	String	Tên tiếng Việt của cây thuốc.
3	ScientificName	Public	String	Tên khoa học của cây thuốc.
4	partUsed	Public	String	Thành phần sử dụng.

5	Description	Public	String	Mô tả.
6	Habitat	Public	String	Nơi sinh sống.
7	Usage	Public	String	Cách dùng.
8	Effect	Public	String	Công dụng.
9	ChemicalComposition	Public	String	Thành phần hóa học.
10	search_count	Public	Int	Số lần được tìm kiếm.
11	RareLevel	Public	Int	Mức độ hiếm.
12	SummaryEffect	Public	String	Công dụng tóm tắt.

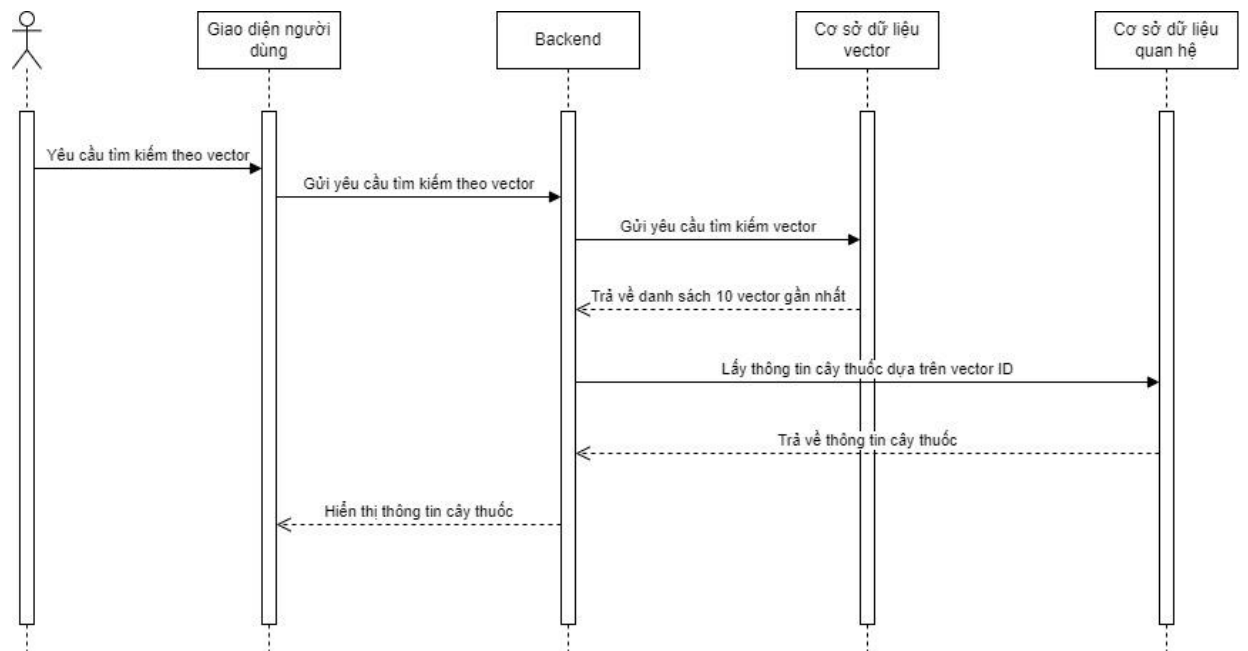
*Bảng 2.16: Bảng mô tả các phương thức của lớp MedicinalPlants*

STT	Tên	Phạm vi	Mô tả	Đầu vào	Đầu ra
1	GetMedicinalPlants()	public	Lấy thông tin của tất cả cây thuốc.		Danh sách cây thuốc.
2	GetMedicinalPlant()	public	Lấy thông tin của một cây thuốc.	Mã cây thuốc.	Thông tin cây thuốc.
3	GetMedicinalPlant VitaminC()	public	Lấy thông tin của top 4 cây thuốc giàu vitamin C.		Danh sách 4 cây thuốc giàu Vitamin C.
4	GetMedicinalPlant Protein()	public	Lấy thông tin của top 4 cây thuốc giàu protein.		Danh sách 4 cây thuốc giàu Protein.
5	SearchByName()	public	Tìm kiếm cây thuốc theo tên.	Từ khóa tìm kiếm.	Danh sách cây thuốc.
6	SearchByVector()	public	Tìm kiếm cây thuốc theo thành phần hóa học (vector).	Vector float 10 chiều.	Danh sách 10 cây thuốc.
7	PostMedicinalPlant()	public	Thêm cây thuốc mới vào CSDL.	Thông tin cây thuốc.	
8	PutMedicinalPlant()	public	Sửa thông tin cây thuốc.	Mã cây thuốc và thông tin cây thuốc.	

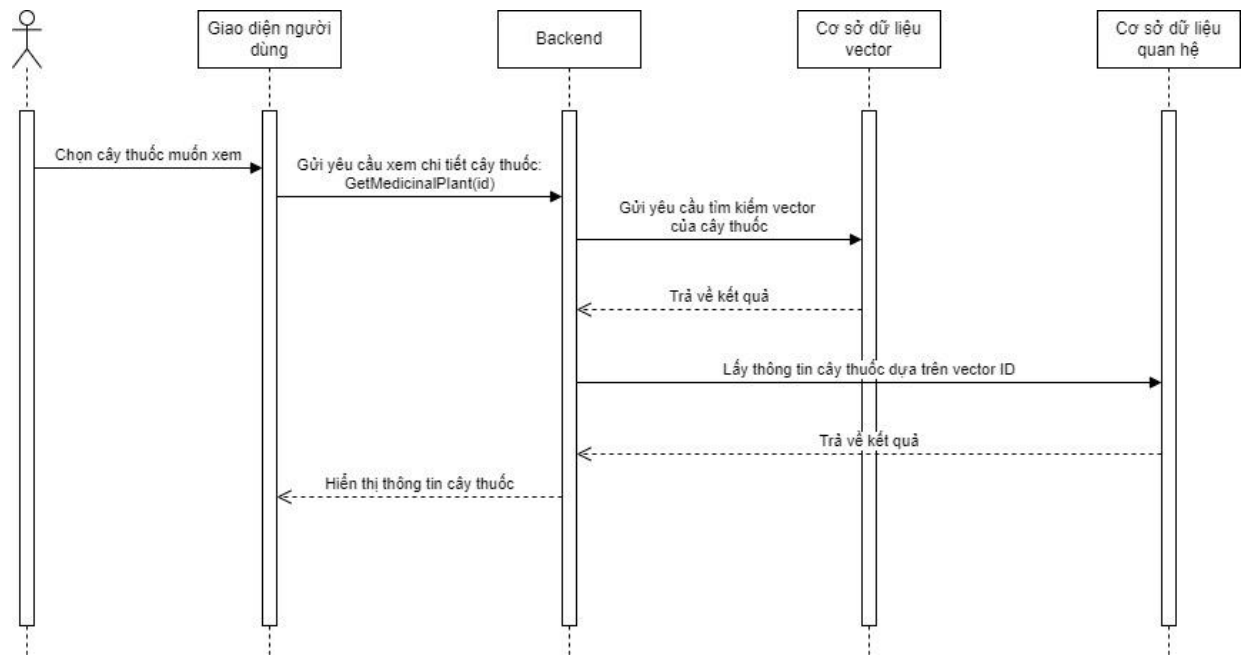
### 2.2.3. Sơ đồ tuần tự



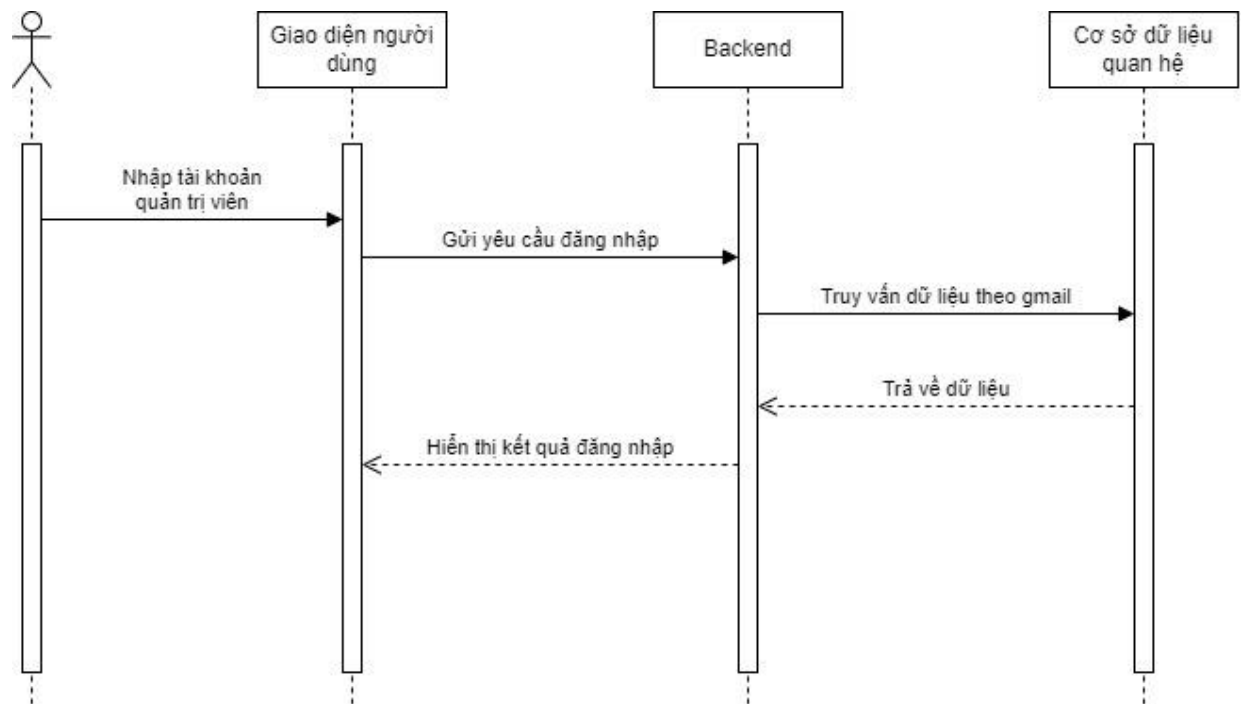
Hình 2.11: Sơ đồ tuần tự của chức năng tìm kiếm



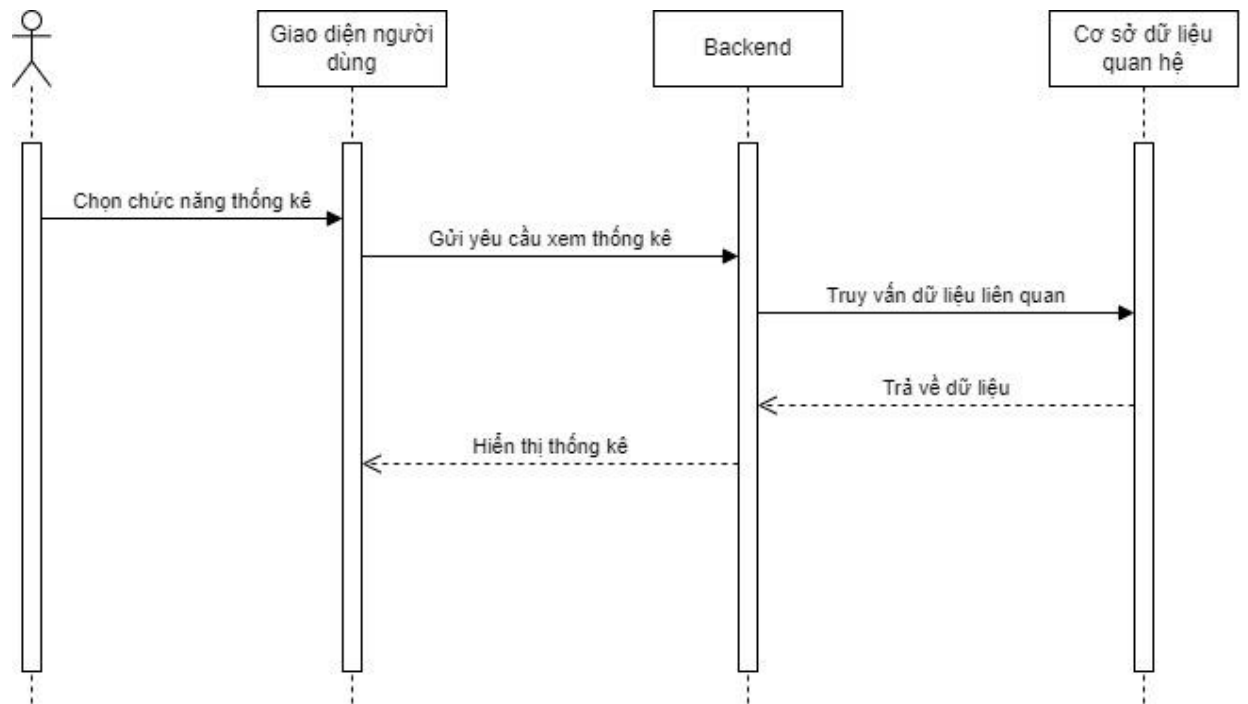
Hình 2.12: Sơ đồ tuần tự chức năng tìm kiếm theo 10 TPHH



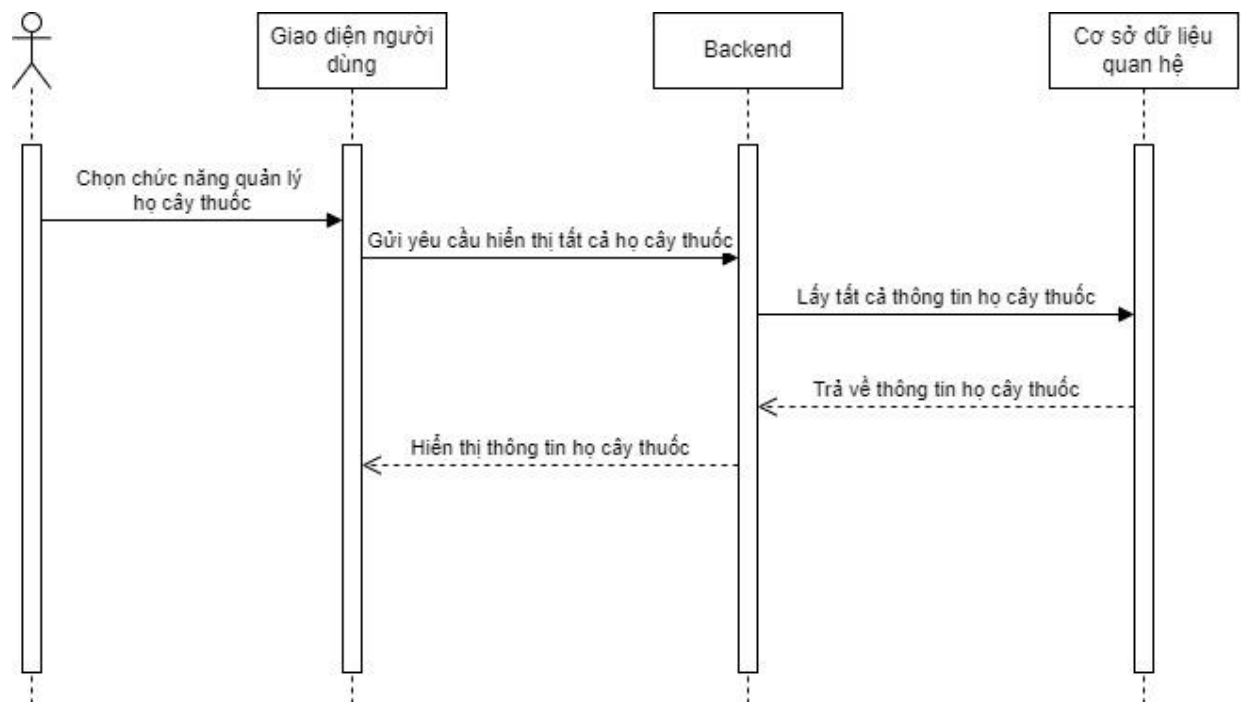
Hình 2.13: Sơ đồ tuần tự xem chi tiết cây thuốc



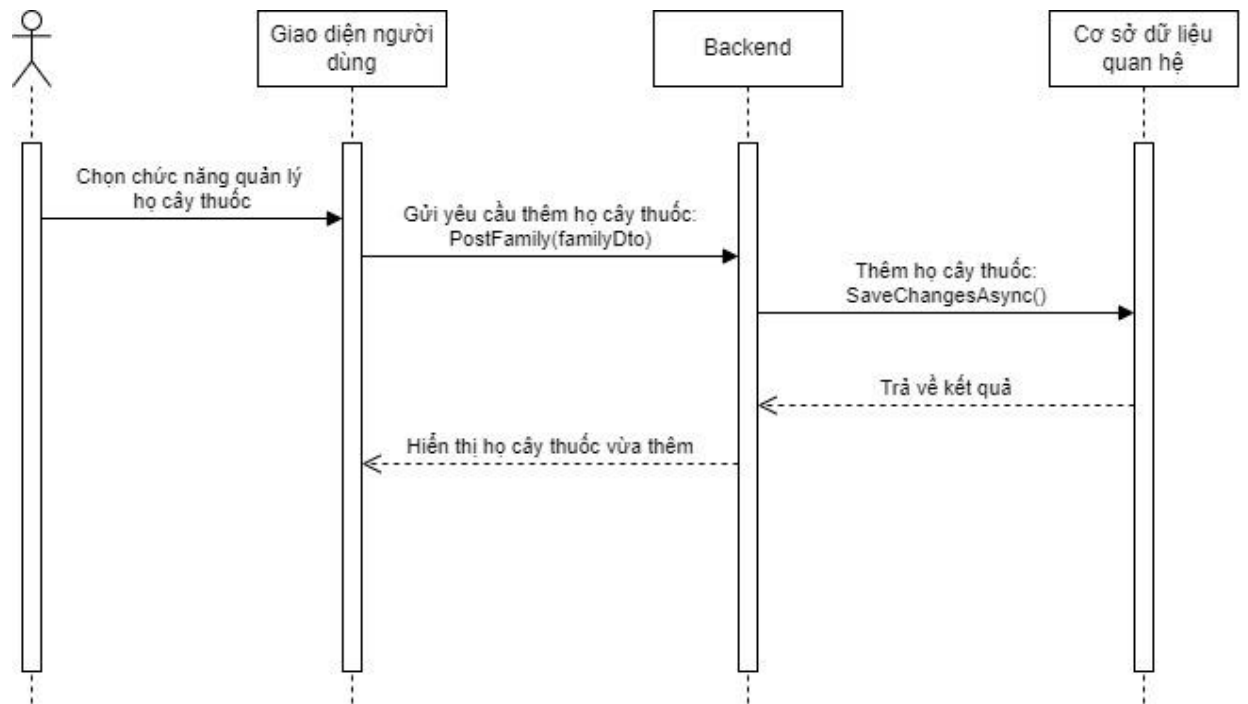
Hình 2.14: Sơ đồ tuần tự chức năng đăng nhập



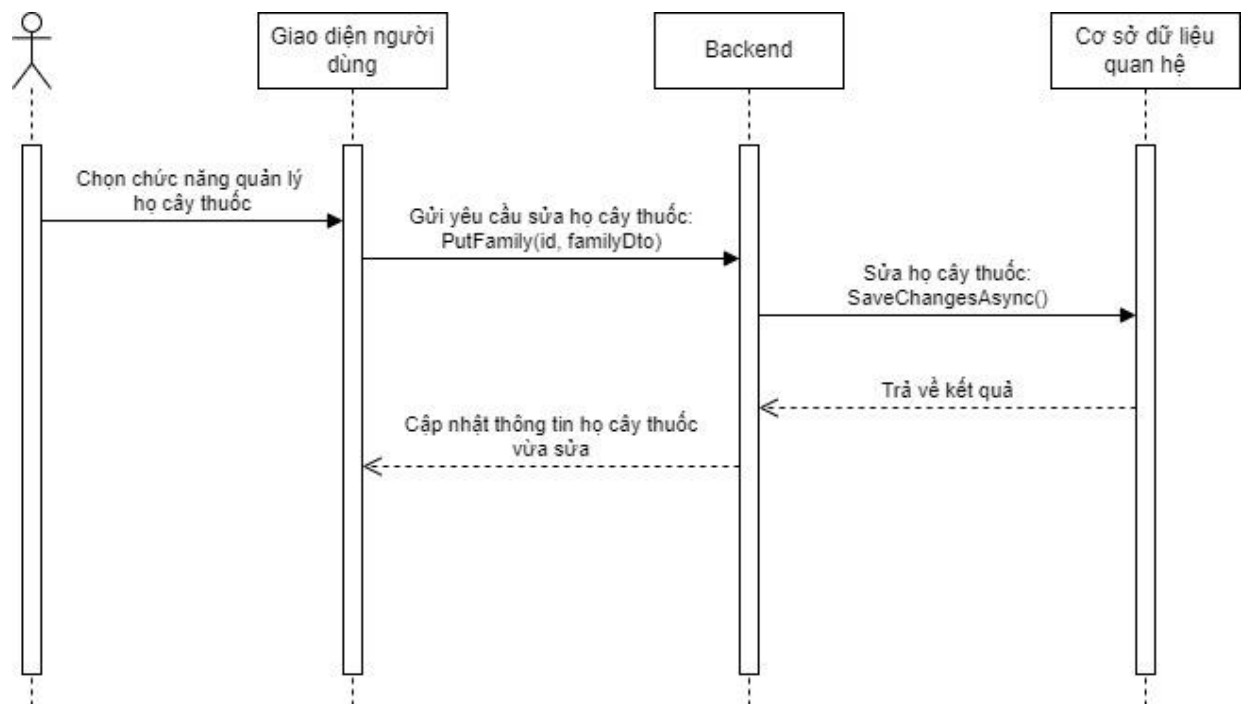
Hình 2.15: Sơ đồ tuần tự xem thống kê



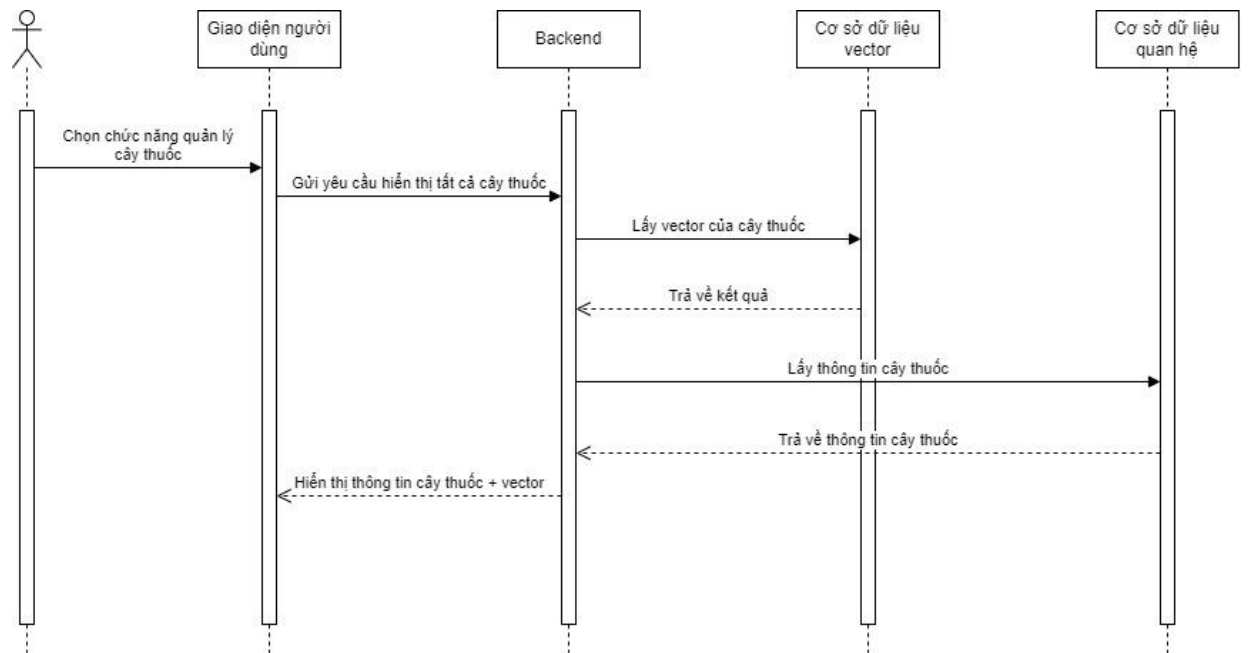
Hình 2.16: Sơ đồ tuần tự xem thông tin họ cây thuốc



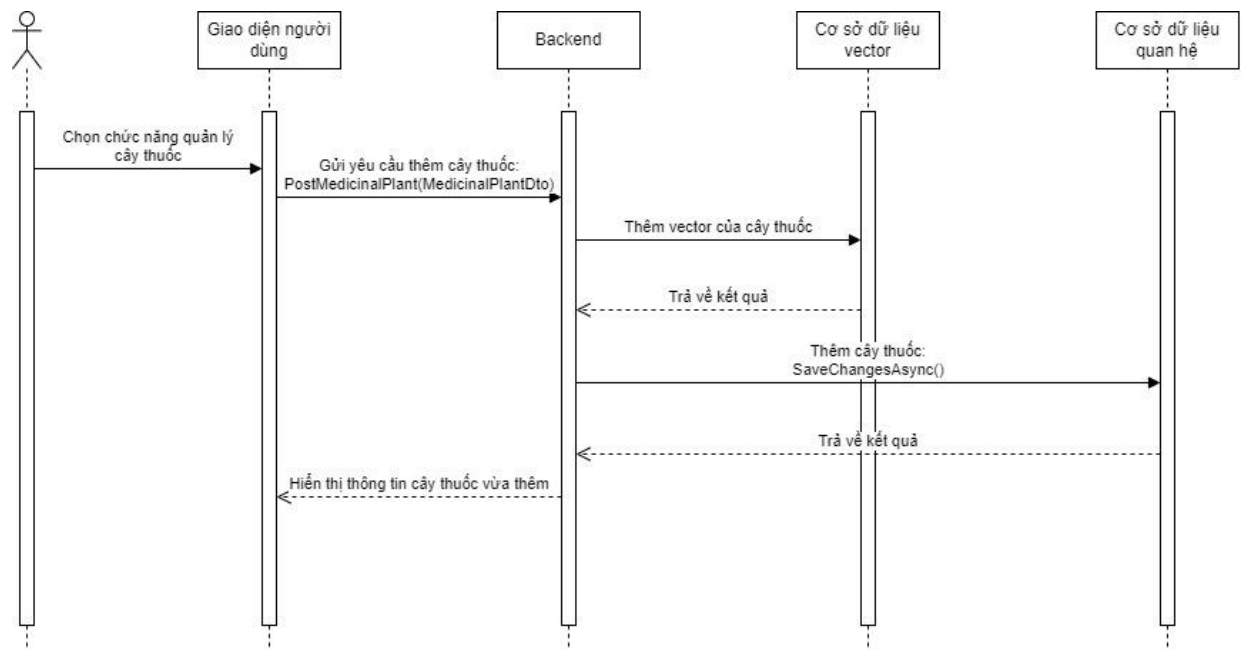
Hình 2.17: Sơ đồ tuần tự thêm họ cây thuốc



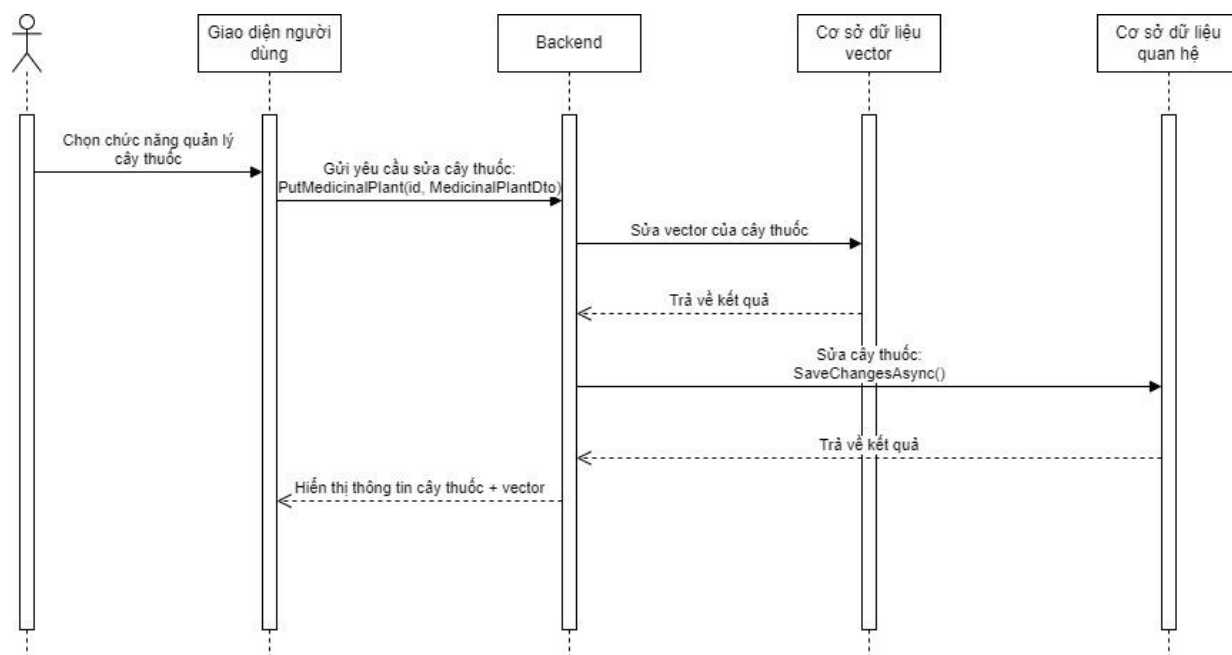
Hình 2.18: Sơ đồ tuần tự sửa họ cây thuốc



Hình 2.19: Sơ đồ tuần tự xem thông tin cây thuốc

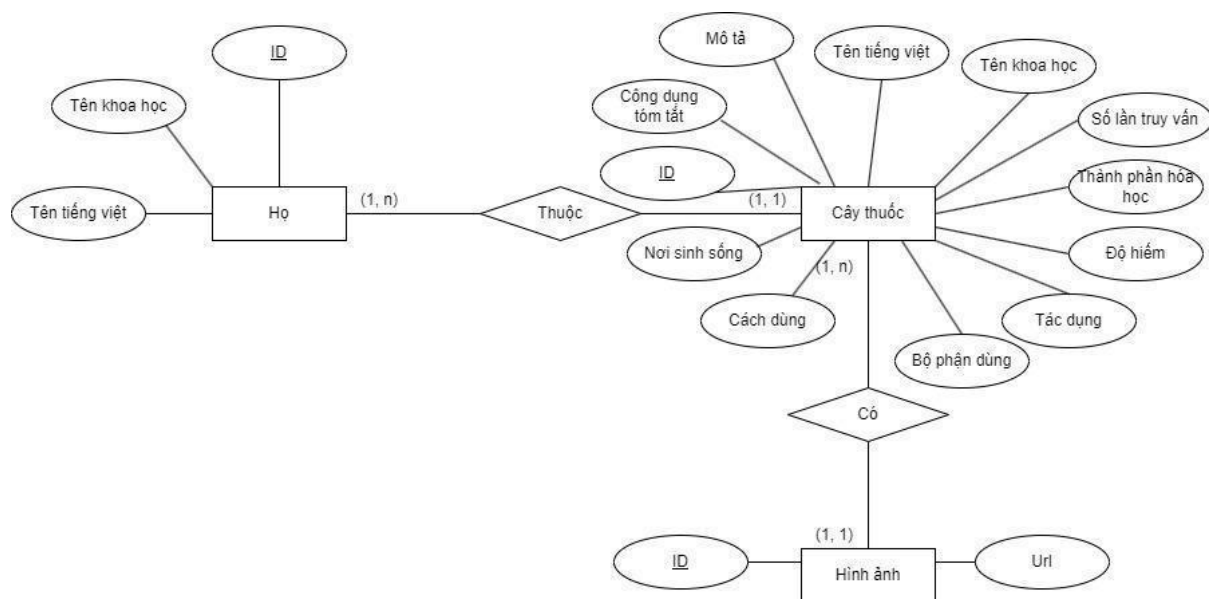


Hình 2.20: Sơ đồ tuần tự thêm cây thuốc



Hình 2.21: Sơ đồ tuần tự sửa cây thuốc

### 2.3. Thiết kế cơ sở dữ liệu



Hình 2.22: Sơ đồ thực thể kết hợp



Bảng 2.17: Bảng mô tả thực thể cây thuốc

<b>Mô tả:</b> Lưu thông tin cây thuốc trong hệ thống.			
Thuộc tính	Kiểu	Not null	Diễn giải
Id	Int	x	Mã cây thuốc.
VietnameseName	String	x	Tên tiếng Việt của cây thuốc.
ScientificName	String	x	Tên khoa học của cây thuốc.
partUsed	String	x	Thành phần sử dụng.
Description	String	x	Mô tả.
Habitat	String	x	Nơi sinh sống.
Usage	String	x	Cách dùng.
Effect	String	x	Công dụng.
ChemicalComposition	String	x	Thành phần hóa học.
search_count	Int	x	Số lần được tìm kiếm.
RareLevel	Int	x	Mức độ hiếm.
SummaryEffect	String	x	Công dụng tóm tắt.

Bảng 2.18: Bảng mô tả thực thể họ cây thuốc

<b>Mô tả:</b> Lưu thông tin họ cây thuốc trong hệ thống.			
Thuộc tính	Kiểu	Not null	Diễn giải
Id	Int	x	Mã họ cây thuốc.

VietnameseName	String	x	Tên tiếng Việt họ cây thuốc.
ScientificName	String	x	Tên khoa học họ cây thuốc.

*Bảng 2.19: Bảng mô tả thực thể hình ảnh*

<b>Mô tả:</b> Lưu thông tin đường dẫn hình ảnh cây thuốc trong hệ thống.			
Thuộc tính	Kiểu	Not null	Diễn giải
Id	Int	x	Mã hình ảnh.
url	String	x	Đường dẫn hình ảnh cây thuốc.

## CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

### 3.1. Cơ sở dữ liệu vector

#### 3.1.1. Cơ sở dữ liệu vector là gì?

Hiện nay, việc xử lý dữ liệu hiệu quả đang trở thành yếu tố thiết yếu trong các ứng dụng trí tuệ nhân tạo, đặc biệt là với các mô hình ngôn ngữ lớn, tìm kiếm ngữ nghĩa. Một yếu tố cốt lõi của những ứng dụng này là vector embedding – một dạng biểu diễn dữ liệu vector mang thông tin ngữ nghĩa, giúp AI không chỉ hiểu ngữ cảnh mà còn lưu trữ và truy xuất “trí nhớ dài hạn” khi xử lý các tác vụ phức tạp.

Các vector embedding được tạo ra bởi các mô hình AI, chẳng hạn như mô hình ngôn ngữ lớn. Mỗi embedding chứa nhiều thuộc tính, hay còn gọi là các đặc trưng, đại diện cho các chiều dữ liệu quan trọng. Những chiều này đóng vai trò quan trọng trong việc nhận diện các mối quan hệ, mẫu hình và cấu trúc ẩn trong dữ liệu. Tuy nhiên, sự phức tạp và quy mô ngày càng lớn của các embedding đặt ra những thách thức lớn cho việc lưu trữ, truy vấn và phân tích dữ liệu, đặc biệt khi sử dụng các cơ sở dữ liệu truyền thống.

Cơ sở dữ liệu truyền thống, vốn dựa trên các giá trị vô hướng, không được thiết kế để xử lý các embedding. Điều này dẫn đến hiệu suất kém, thiếu khả năng mở rộng và không đáp ứng được yêu cầu phân tích theo thời gian thực. Các chỉ mục vector độc lập có thể hỗ trợ tìm kiếm tương đồng, nhưng lại thiếu các chức năng quản lý dữ liệu toàn diện như thêm, xóa, sửa, đọc dữ liệu hoặc lọc siêu dữ liệu.

Để khắc phục những hạn chế trên, các cơ sở dữ liệu vector ra đời với khả năng chuyên biệt hóa cho việc lưu trữ và truy vấn các vector embedding. Các cơ sở dữ liệu này không chỉ hỗ trợ các chức năng cơ bản của cơ sở dữ liệu truyền thống mà còn tích hợp thêm các tính năng mạnh mẽ cho việc xử lý các embedding. Chúng cung cấp khả năng mở rộng ngang, lọc siêu dữ liệu, và hoạt động không cần máy chủ, giúp tối ưu hóa chi phí lưu trữ và xử lý.

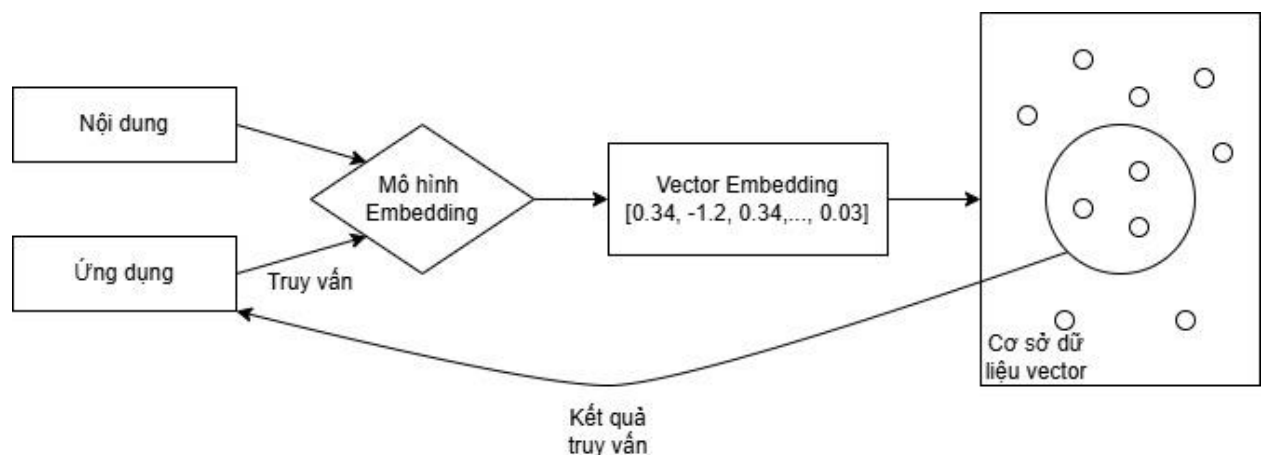
Hiện nay, các cơ sở dữ liệu vector thế hệ mới đang phát triển theo hướng sử dụng kiến trúc tinh vi hơn để cải thiện hiệu suất và quản lý chi phí. Một ví dụ điển hình là cơ sở dữ liệu vector không máy chủ, cho phép tách biệt chi phí lưu trữ và xử lý. Điều này

không chỉ giảm thiểu chi phí mà còn nâng cao khả năng mở rộng, hỗ trợ các ứng dụng AI tiên tiến.

Với sự hỗ trợ của cơ sở dữ liệu vector, các hệ thống AI có thể tận dụng dữ liệu hiệu quả hơn, từ truy xuất thông tin ngữ nghĩa đến duy trì trí nhớ dài hạn. Điều này mở ra nhiều cơ hội mới trong việc phát triển các ứng dụng AI hiện đại như tìm kiếm ngữ nghĩa, chatbot thông minh, hệ thống gợi ý, và phân tích dữ liệu phức tạp.

Cơ sở dữ liệu vector không chỉ là giải pháp lưu trữ, mà còn là công cụ then chốt giúp hiện thực hóa tiềm năng của AI trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận diện hình ảnh, và học sâu. Sự phát triển liên tục của phương pháp này sẽ định hình tương lai của các ứng dụng AI, giúp chúng trở nên mạnh mẽ hơn và hiệu quả hơn trong việc giải quyết các vấn đề thực tế.

Sơ đồ (Hình 3.1) giúp chúng ta hiểu rõ hơn về vai trò của cơ sở dữ liệu vector trong loại ứng dụng này.



Hình 3.1: Vai trò của cơ sở dữ liệu vector với AI

Ý nghĩa:

- Bước đầu, chúng ta sử dụng mô hình embedding để tạo những vector embedding cho nội dung mà chúng ta muốn lập chỉ mục.
- Vector embedding được thêm vào cơ sở dữ liệu vector, với một số tham chiếu tới nội dung gốc mà embedding được tạo từ đó.

- Khi ứng dụng đưa ra một query, chúng ta sử dụng cùng mô hình embedding để tạo các vector embedding cho việc truy vấn và sử dụng các vector embedding đó để truy vấn trong cơ sở dữ liệu để tìm kiếm các vector embedding tương tự.

### 3.1.2. Vector embedding và vai trò trong AI

Vector embedding là một phương pháp biểu diễn dữ liệu dưới dạng vector trong không gian nhiều chiều. Đây là một kỹ thuật quan trọng trong lĩnh vực trí tuệ nhân tạo, giúp chuyển đổi dữ liệu phức tạp như văn bản, hình ảnh, âm thanh, hoặc video thành các đại diện toán học dễ xử lý hơn. Mỗi vector embedding chứa thông tin ngữ nghĩa, giúp các hệ thống AI hiểu được mối liên kết, tương đồng và khác biệt giữa các dữ liệu.

Vector embedding được tạo ra bởi các mô hình AI tiên tiến, như các mô hình ngôn ngữ lớn hoặc các mô hình học sâu. Mỗi embedding là một vector trong không gian  $n$  chiều, trong đó mỗi chiều đại diện cho một đặc trưng hoặc thuộc tính của dữ liệu. Ví dụ:

- Trong xử lý ngôn ngữ tự nhiên, các từ như “cat” và “dog” sẽ có các vector embedding gần nhau hơn so với từ “car,” vì chúng có ý nghĩa ngữ cảnh tương tự.
- Trong xử lý hình ảnh, hai hình ảnh có nội dung tương đồng sẽ có vector embedding gần nhau hơn trong không gian vector.

Vai trò của vector embedding trong AI:

- Hiểu ngữ nghĩa và ngữ cảnh: Embedding cho phép các mô hình AI không chỉ xử lý dữ liệu dưới dạng toán học mà còn hiểu được ngữ nghĩa. Điều này rất quan trọng trong các ứng dụng như tìm kiếm ngữ nghĩa, xử lý văn bản, và nhận diện hình ảnh. Ví dụ: Trong tìm kiếm, vector embedding giúp hiểu truy vấn của người dùng và trả về các kết quả liên quan nhất, ngay cả khi truy vấn không khớp chính xác với dữ liệu.
- Tăng cường khả năng học máy: Embedding giúp các mô hình học máy học được mối quan hệ giữa các đối tượng, cải thiện khả năng phân loại và dự đoán. Bằng cách chuyển đổi dữ liệu thành các vector, AI có thể nhận diện các mẫu và mối liên hệ ẩn sâu trong dữ liệu.
- Hỗ trợ trí nhớ dài hạn: Trong AI, vector embedding được sử dụng để lưu trữ và duy trì trí nhớ dài hạn, giúp các hệ thống AI có thể “ghi nhớ” thông tin và sử dụng lại khi cần thiết. Điều này đặc biệt quan trọng trong các ứng dụng liên quan đến chatbot, trợ lý ảo, hoặc các hệ thống đề xuất.

- Cơ sở cho tìm kiếm tương đồng: Embedding là nền tảng cho các ứng dụng tìm kiếm dựa trên sự tương đồng, chẳng hạn tìm kiếm hình ảnh, video, hoặc tài liệu liên quan. AI sử dụng khoảng cách giữa các vector (như khoảng cách Cosine hoặc Euclid) để xác định mức độ tương đồng giữa các dữ liệu.

Mặc dù có nhiều ưu điểm, việc quản lý vector embedding không phải là nhiệm vụ dễ dàng. Các embedding thường có kích thước lớn và chứa nhiều chiều dữ liệu, đòi hỏi hệ thống lưu trữ và xử lý mạnh mẽ. Điều này đặt ra nhu cầu về các cơ sở dữ liệu vector chuyên biệt, được tối ưu hóa để lưu trữ, tìm kiếm và quản lý embedding hiệu quả.

### 3.1.3. Ưu điểm của cơ sở dữ liệu vector

Cơ sở dữ liệu vector là một giải pháp chuyên biệt, được thiết kế để lưu trữ, quản lý và truy xuất các vector embedding. Nhờ vào các tính năng tối ưu hóa, cơ sở dữ liệu vector mang lại nhiều ưu điểm vượt trội so với các hệ thống lưu trữ dữ liệu truyền thống, đặc biệt trong bối cảnh xử lý dữ liệu phức tạp và quy mô lớn của AI. Dưới đây là các ưu điểm chính:

- Tối ưu hóa cho dữ liệu vector:

- Thiết kế chuyên biệt để lưu trữ và xử lý các vector embedding.
- Tốc độ truy xuất và tìm kiếm vector tương đồng nhanh chóng.
- Hiệu suất cao khi xử lý dữ liệu lớn và đa chiều.

- Hỗ trợ tìm kiếm tương đồng hiệu quả:

- Sử dụng các thuật toán như “thuật toán tìm kiếm láng giềng gần đúng”.
- Tìm kiếm ngữ nghĩa, nhận diện hình ảnh tương tự.
- Gợi ý sản phẩm và thông tin dựa trên dữ liệu vector.

- Tích hợp linh hoạt với siêu dữ liệu:

- Kết hợp truy vấn vector với các tiêu chí siêu dữ liệu.
- Lọc dữ liệu dựa trên thuộc tính như ngày tạo, loại dữ liệu.

- Khả năng mở rộng ngang:

- Xử lý dữ liệu lớn trên nhiều máy chủ.
- Đáp ứng hiệu suất cao khi khối lượng dữ liệu tăng.

- Tăng tính sẵn sàng và độ tin cậy của hệ thống.
- Hoạt động không cần máy chủ:
  - Tự động quản lý tài nguyên, giảm chi phí vận hành.
  - Tách biệt chi phí lưu trữ và xử lý để tối ưu hóa ngân sách.
  - Linh hoạt trong việc đáp ứng khối lượng công việc thay đổi.
- Tích hợp dễ dàng với hệ thống AI:
  - Truy cập nhanh chóng các vector để đào tạo hoặc suy luận AI.
  - Cải thiện độ chính xác và hiệu quả của ứng dụng AI.
- Tối ưu tài nguyên và giảm chi phí:
  - Hiệu quả trong việc lưu trữ và truy xuất dữ liệu vector.
  - Giảm chi phí phần cứng và hạ tầng so với hệ thống truyền thống.
- Độ chính xác cao trong xử lý dữ liệu ngữ nghĩa:
  - Hỗ trợ tính toán khoảng cách (Cosine, Euclid,...) giữa các vector.
  - Đảm bảo độ chính xác cao trong các bài toán phân loại và tìm kiếm.
  - Phù hợp với chatbot, phân tích văn bản, hệ thống đề xuất, tìm kiếm tương đồng.

### 3.1.4. Sự khác nhau giữa chỉ mục vector độc lập và cơ sở dữ liệu vector

Trong lĩnh vực xử lý dữ liệu vector và tìm kiếm tương đồng, chỉ mục vector độc lập như FAISS và cơ sở dữ liệu vector đều được sử dụng để lưu trữ và truy vấn các vector embedding. Tuy nhiên, cơ sở dữ liệu vector mang lại nhiều ưu điểm vượt trội hơn so với chỉ mục vector độc lập nhờ tính năng quản lý dữ liệu toàn diện và tích hợp. Bảng (3.1) là sự khác biệt chi tiết giữa hai khái niệm này.

*Bảng 3.1: Bảng so sánh sự khác nhau giữa chỉ mục vector độc lập và cơ sở dữ liệu vector*

Điểm so sánh	Chỉ mục vector độc lập	Cơ sở dữ liệu vector
<b>Quản lý dữ liệu</b>	<ul style="list-style-type: none"> <li>- Cung cấp khả năng lưu trữ và truy vấn vector embedding.</li> <li>- Không hỗ trợ các thao tác dữ liệu cơ bản như thêm, xóa</li> </ul>	<ul style="list-style-type: none"> <li>- Hỗ trợ đầy đủ các thao tác CRUD.</li> <li>- Đơn giản hóa việc quản lý và bảo trì dữ liệu vector nhờ tích hợp lưu trữ sẵn có.</li> </ul>

	hoặc cập nhật một cách dễ dàng. - Yêu cầu tích hợp với các giải pháp lưu trữ bên ngoài, dẫn đến việc quản lý dữ liệu phức tạp hơn.	
<b>Lưu trữ siêu dữ liệu và lọc</b>	- Không hỗ trợ lưu trữ siêu dữ liệu gắn liền với vector. - Truy vấn chỉ dựa trên nội dung vector mà không có các bộ lọc nâng cao.	- Lưu trữ siêu dữ liệu liên quan đến mỗi vector embedding. - Cho phép truy vấn với các bộ lọc siêu dữ liệu, cung cấp các truy vấn chi tiết và linh hoạt hơn.
<b>Khả năng mở rộng</b>	- Cần triển khai và quản lý các giải pháp tùy chỉnh, chẳng hạn sử dụng Kubernetes để phân phối và xử lý dữ liệu lớn. - Thiếu tính năng mở rộng tích hợp.	- Được thiết kế để mở rộng theo nhu cầu người dùng và dung lượng dữ liệu. - Hỗ trợ xử lý phân tán và song song, dễ dàng thích nghi với khối lượng dữ liệu lớn. - Kiến trúc không máy chủ tối ưu chi phí khi mở rộng.
<b>Cập nhật dữ liệu theo thời gian thực</b>	- Mỗi lần cập nhật dữ liệu, cần thực hiện quá trình lập lại chỉ mục toàn bộ, tốn kém thời gian và tài nguyên. - Không hỗ trợ cập nhật dữ liệu thời gian thực.	- Hỗ trợ cập nhật dữ liệu thời gian thực mà không cần lập lại chỉ mục toàn bộ. - Cho phép tối ưu hóa hiệu suất thông qua các cải tiến như tái xây dựng chỉ mục mà vẫn duy trì độ mới của dữ liệu.
<b>Sao lưu dữ liệu</b>	- Không có cơ chế sao lưu tự động hoặc quản lý bộ sưu tập dữ liệu.	- Hỗ trợ sao lưu toàn bộ dữ liệu một cách tự động. - Cho phép người dùng chọn chỉ mục cụ thể để sao lưu dưới dạng collection, giúp dễ dàng phục hồi và tái sử dụng.
<b>Tích hợp hệ thống</b>	- Khả năng tích hợp bị hạn chế, yêu cầu cấu hình tùy chỉnh với các công cụ bên ngoài.	- Dễ dàng tích hợp với các công cụ ETL (như Spark), nền tảng phân tích (như Tableau), và công cụ trực quan hóa (như Grafana).
<b>Bảo mật và kiểm soát truy cập</b>	- Không tích hợp sẵn các tính năng bảo mật dữ liệu hoặc kiểm soát quyền truy cập.	- Cung cấp các tính năng bảo mật tích hợp, như kiểm soát truy cập và phân vùng dữ liệu qua namespace.
<b>Hiệu quả tổng thể</b>	- Tập trung vào truy vấn vector nhanh, nhưng hạn chế	- Cung cấp giải pháp toàn diện để lưu trữ, quản lý, và



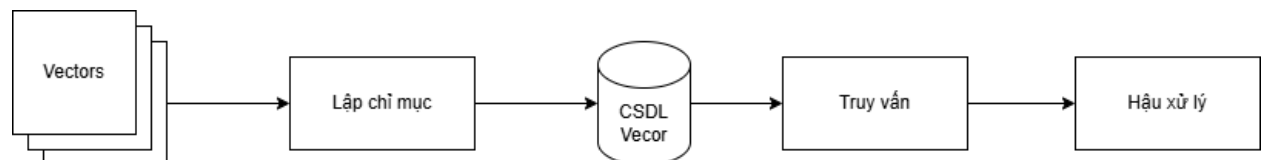
	về khả năng quản lý dữ liệu và tính năng bổ sung.	truy vấn vector embedding, đảm bảo hiệu suất cao, bảo mật, và tích hợp dễ dàng.
--	---	---

### 3.1.5. Cơ sở dữ liệu vector làm việc thế nào?

Cơ sở dữ liệu vector khác biệt hoàn toàn so với cơ sở dữ liệu truyền thống trong cách hoạt động và xử lý dữ liệu. Trong khi cơ sở dữ liệu truyền thống lưu trữ và truy vấn dữ liệu như chuỗi, số,... thì cơ sở dữ liệu vector được tối ưu hóa để xử lý và truy vấn các vector embedding. Những vector này đại diện cho dữ liệu phức tạp (văn bản, hình ảnh, âm thanh) trong không gian nhiều chiều.

Trong cơ sở dữ liệu truyền thống, chúng ta thường thực hiện truy vấn để tìm những hàng có giá trị chính xác khớp với câu truy vấn. Tuy nhiên, trong cơ sở dữ liệu vector, quá trình truy vấn dựa trên mức độ tương đồng, sử dụng các thước đo như *Cosine* hoặc *Euclid*,... Điều này cho phép cơ sở dữ liệu tìm các vector gần nhất với vector truy vấn thay vì tìm kết quả khớp chính xác.

Để đạt được điều này, cơ sở dữ liệu vector sử dụng một tập hợp các thuật toán tham gia vào quá trình tìm kiếm các láng giềng gần đúng.



Hình 3.2: Cách hoạt động của một cơ sở dữ liệu vector

Cách hoạt động của một cơ sở dữ liệu vector như (Hình 3.2) và có thể được cô đọng thành ba bước chính:

#### a. Lập chỉ mục

Đây là bước đầu tiên, nơi cơ sở dữ liệu sử dụng các thuật toán lập chỉ mục như:

- Lượng tử hóa tích: Nén vector để tăng tốc độ truy vấn.
- Băm vị trí nhảy cảm: Tạo các băm giúp tìm kiếm nhanh.
- Thuật toán thế giới nhỏ có điều hướng theo phân cấp: Cấu trúc đồ thị giúp điều hướng nhanh đến các vector gần nhất.

Mục tiêu là ánh xạ các vector vào một cấu trúc dữ liệu phù hợp để tăng tốc độ truy vấn trong tương lai.

### **b. Truy vấn**

Trong bước này, cơ sở dữ liệu so sánh vector truy vấn với các vector trong tập dữ liệu đã lập chỉ mục.

Áp dụng các thước đo tương đồng để xác định các vector gần nhất.

Đây là bước xác định các vector tiềm năng có độ tương đồng cao với vector truy vấn.

### **c. Xử lý hậu kỳ**

Sau khi xác định được các vector gần nhất, cơ sở dữ liệu có thể: Sắp xếp lại danh sách kết quả dựa trên một thước đo tương đồng khác. Sau đó trả về kết quả cuối cùng sau khi tinh chỉnh. Bước này giúp cải thiện độ chính xác của kết quả và tăng cường chất lượng tìm kiếm.

Ưu Điểm:

- **Tốc độ cao:** Cung cấp truy vấn nhanh chóng, ngay cả với tập dữ liệu lớn.
- **Độ chính xác tốt:** Đạt được kết quả gần đúng với độ chính xác cao trong hầu hết các trường hợp.
- **Tính linh hoạt:** Hỗ trợ nhiều loại dữ liệu khác nhau và các thước đo tương đồng.

Hạn Chế:

- **Thỏa hiệp giữa độ chính xác và tốc độ:** Khi tăng độ chính xác, thời gian truy vấn có thể chậm hơn.
- **Phức tạp trong cấu hình:** Việc chọn thuật toán lập chỉ mục và tối ưu hóa có thể yêu cầu kiến thức chuyên môn.

### **3.1.6. Kiến trúc không máy chủ và lợi ích**

Kiến trúc không máy chủ là một bước tiến trong phương pháp cơ sở dữ liệu vector, giúp giải quyết những hạn chế của các cơ sở dữ liệu vector thế hệ đầu tiên. Bằng cách tối ưu hóa chi phí, cải thiện tính linh hoạt và đáp ứng nhu cầu sử dụng ngày càng tăng của

các ứng dụng AI, cơ sở dữ liệu vector không máy chủ đã trở thành một thành phần quan trọng trong các hệ thống hiện đại.

### 3.1.6.1. Nhược điểm của các cơ sở dữ liệu vector thể hệ đầu tiên

Các cơ sở dữ liệu vector thể hệ đầu tiên có hiệu suất cao nhưng đi kèm với chi phí đắt đỏ và những hạn chế:

- Không tách biệt lưu trữ và xử lý: Hệ thống phải sử dụng tài nguyên xử lý mọi lúc, ngay cả khi không cần thiết, gây lãng phí chi phí.
- Quản lý đa người dùng: Việc đồng bộ và tối ưu hóa tài nguyên cho các namespace ít được truy vấn vẫn tiêu tốn tài nguyên đáng kể.
- Độ mới: Các cơ sở dữ liệu vector thường gặp vấn đề trong việc cập nhật dữ liệu mới theo thời gian thực, đặc biệt khi khối lượng dữ liệu lớn.

Kiến trúc không máy chủ ra đời để khắc phục những điểm yếu này, mang đến một thể hệ cơ sở dữ liệu vector hoàn toàn mới.

### 3.1.6.2. Lợi ích của kiến trúc không máy chủ

#### a. Tách biệt lưu trữ và xử lý

Cách hoạt động:

- Dữ liệu được chia thành các phân vùng bằng thuật toán phân vùng hình học, chẳng hạn như *Voronoi tessellation*.
- Phân vùng này cho phép chỉ tìm kiếm trong một phần nhỏ của chỉ mục vector, thay vì toàn bộ không gian tìm kiếm, tối ưu hóa tài nguyên.

Lợi ích:

- Giảm chi phí xử lý khi không sử dụng.
- Tăng tốc độ truy vấn bằng cách tập trung vào các phân vùng quan trọng.
- Cân bằng giữa chi phí xử lý và độ trễ.

#### b. Lớp độ mới

Cách hoạt động:

- Dữ liệu mới được lưu tạm thời trong một lớp đệm tạm thời gọi là lớp độ mới.
- Truy vấn sẽ được gửi đồng thời đến cả lớp này và chỉ mục phân vùng chính, đảm bảo dữ liệu luôn được cập nhật.

Lợi ích:

- Cung cấp kết quả nhanh chóng cho dữ liệu mới ngay sau khi chèn.

- Hạn chế sự chậm trễ trong quá trình xây dựng chỉ mục vector.

### c. Quản lý đa người dùng

Cách hoạt động:

- Sử dụng phân tích hành vi người dùng để nhóm các người dùng có nhu cầu tương tự trên cùng một phần cứng.
- Tách biệt hoàn toàn giữa các namespace của từng người dùng để đảm bảo hiệu suất.

Lợi ích:

- Tăng hiệu quả sử dụng tài nguyên khi phục vụ cả người dùng có tần suất truy vấn cao và thấp.
- Đảm bảo độ trễ thấp ngay cả với các yêu cầu phức tạp.

Kiến trúc không máy chủ là một bước đột phá, mang lại sự cân bằng giữa hiệu suất, chi phí và khả năng mở rộng. Với sự tách biệt giữa lưu trữ và xử lý, lớp độ mới để đảm bảo dữ liệu thời gian thực, và quản lý đa người dùng hiệu quả, cơ sở dữ liệu vector không máy chủ là nền tảng lý tưởng cho các ứng dụng AI hiện đại, đáp ứng nhu cầu ngày càng tăng của các trường hợp sử dụng AI phức tạp.

### 3.1.7. Thuật toán trong cơ sở dữ liệu vector

Cơ sở dữ liệu vector sử dụng nhiều thuật toán khác nhau để xây dựng chỉ mục và tối ưu hóa quy trình truy vấn vector embedding. Các thuật toán này tập trung vào việc tạo ra các cấu trúc dữ liệu có thể truy cập nhanh, đồng thời bảo tồn tính tương đồng của dữ liệu gốc. Chúng giúp giảm không gian tìm kiếm và tăng tốc độ xử lý, đồng thời mang lại khả năng xử lý hiệu quả ngay cả với tập dữ liệu lớn. Dưới đây là các thuật toán phổ biến được sử dụng trong cơ sở dữ liệu vector.

#### 3.1.7.1. Chiếu ngẫu nhiên

Khái niệm:

Chiếu ngẫu nhiên là phương pháp giảm chiều dữ liệu bằng cách chiếu các vector ban đầu vào một không gian có chiều thấp hơn.

Một ma trận ngẫu nhiên được tạo ra để biểu diễn không gian đích, và vector gốc được nhân với ma trận này để tạo ra các vector mới, có chiều thấp hơn nhưng vẫn giữ được các đặc tính tương đồng.

Quy trình:

- Tạo ma trận ngẫu nhiên: Kích thước của ma trận xác định số chiều của không gian đích.
- Dự phóng vector: Tính tích vô hướng giữa vector đầu vào và ma trận ngẫu nhiên để tạo ra vector mới.
- Truy vấn: Chiếu vector truy vấn vào cùng không gian đích bằng ma trận ngẫu nhiên và so sánh với các vector đã được lưu trữ để tìm các vector gần nhất.

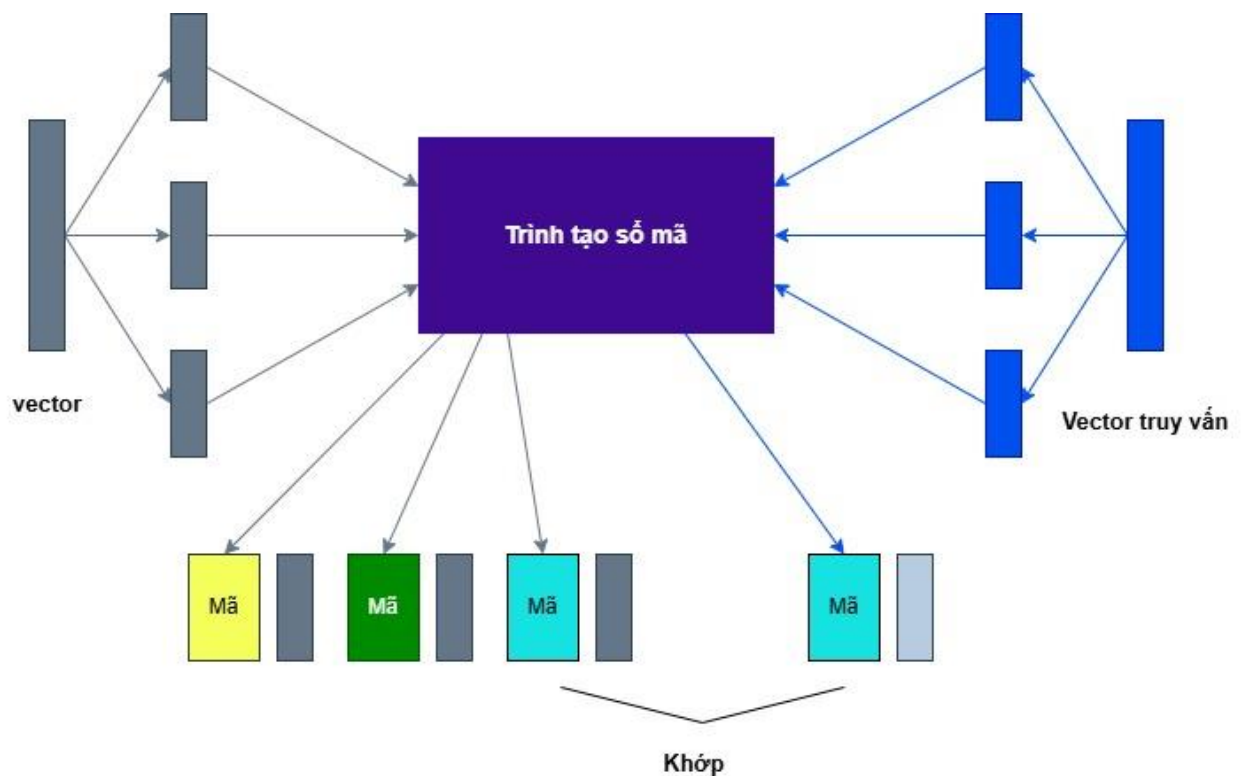
Ưu điểm:

- Giảm thời gian tìm kiếm nhờ giảm số chiều của không gian tìm kiếm.
- Dễ triển khai và sử dụng cho các tập dữ liệu lớn.

Nhược điểm:

- Kết quả chỉ là gần đúng và phụ thuộc vào chất lượng của ma trận ngẫu nhiên.
- Việc tạo ma trận ngẫu nhiên có thể tốn kém nếu dữ liệu quá lớn.

### 3.1.7.2. Lượng tử hóa tích



Hình 3.3: Lượng tử hóa tích

Khái niệm:

Lượng tử hóa tích (Hình 3.4) là một kỹ thuật chia nhỏ vector gốc thành các phần nhỏ và gán mã đại diện cho từng phần để đơn giản hóa việc lưu trữ và tìm kiếm.

Quy trình:

- Chia nhỏ: Vector được chia thành các đoạn.
- Huấn luyện: Áp dụng phân cụm k-means cho từng đoạn để tạo ra một “Trình tạo số mã” chứa các điểm trung tâm của các cụm.
- Mã hóa: Gán mã cho từng đoạn bằng cách tìm điểm gần nhất trong “Trình tạo số mã”.
- Truy vấn: Vector truy vấn được chia nhỏ, mã hóa bằng “Trình tạo số mã”, và so sánh với các mã đã lưu trữ để tìm vector gần nhất.

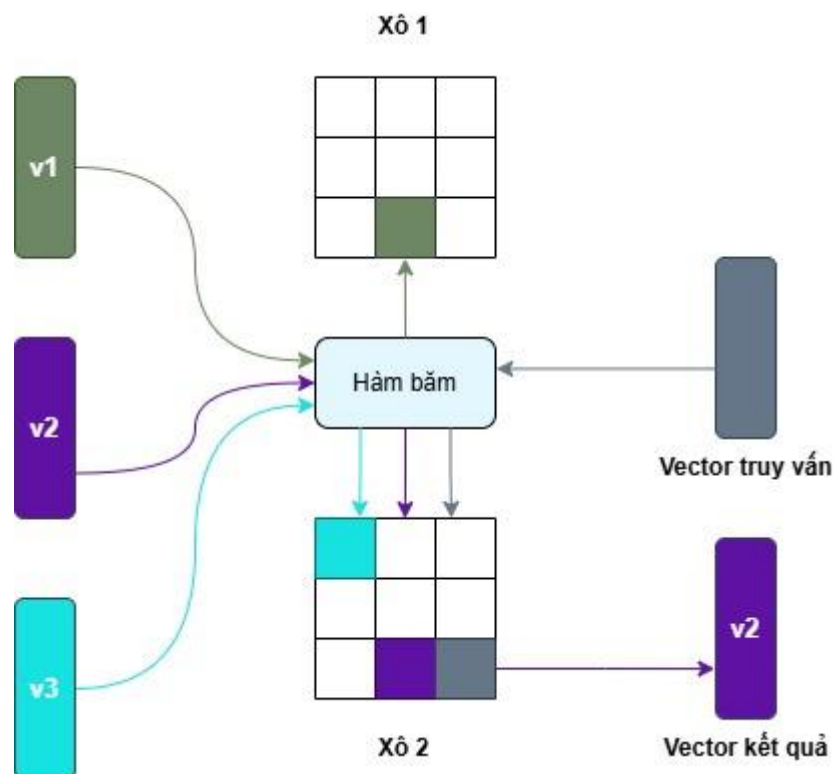
Ưu điểm:

- Tối ưu hóa không gian lưu trữ, giảm kích thước của vector.
- Tăng tốc độ truy vấn nhờ tìm kiếm trên mã thay vì toàn bộ vector gốc.

Nhược điểm:

- Độ chính xác của kết quả phụ thuộc vào số lượng mã đại diện trong “Trình tạo số mã”.
- Cân bằng giữa độ chính xác và chi phí tính toán cần được xem xét.

### 3.1.7.3. Băm vị trí nhảy cảm



Hình 3.4: Băm vị trí nhảy cảm

Khái niệm:

Bấm vị trí nhảy cảm (Hình 3.5) là một kỹ thuật bấm vector vào các “xô” dựa trên độ tương đồng cho vị trí nhảy cảm, kỹ thuật này cho phép tìm kiếm nhanh mà không cần quét toàn bộ tập dữ liệu.

Quy trình:

- **Bấm:** Vector được ánh xạ vào các bảng bấm bằng một tập các hàm bấm nhảy cảm với tương đồng.
- **Truy vấn:** Vector truy vấn được bấm và so sánh với các vector trong cùng “xô” để tìm các vector gần nhất.

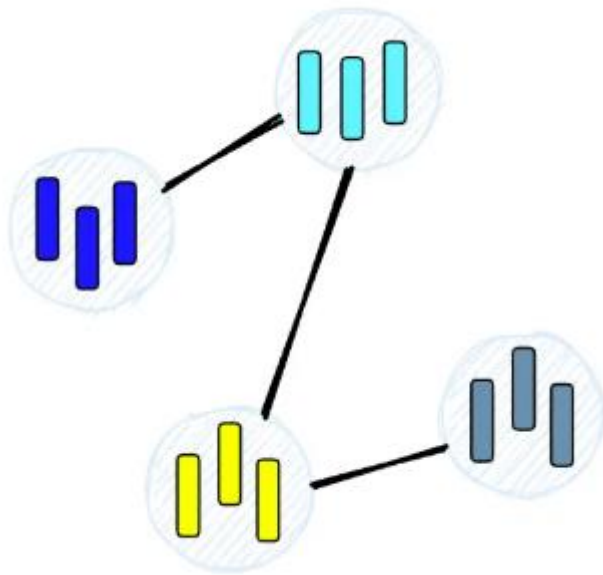
Ưu điểm:

- Tốc độ tìm kiếm nhanh nhờ giảm không gian tìm kiếm xuống còn một số “xô”.
- Dễ dàng triển khai trên các tập dữ liệu lớn.

Nhược điểm:

- Kết quả chỉ là gần đúng và phụ thuộc vào chất lượng của các hàm bấm.
- Sử dụng nhiều hàm bấm có thể tăng chi phí tính toán.

#### 3.1.7.4. Thuật toán thế giới nhỏ có điều hướng theo phân cấp



Hình 3.5: Thuật toán thế giới nhỏ có điều hướng theo phân cấp (nguồn: <https://www.pinecone.io/learn/vector-database/>)

Khái niệm:

Thuật toán thế giới nhỏ có thể điều hướng theo phân cấp (Hình 3.6) là một kỹ thuật tìm kiếm lân cận gần nhất dựa trên đồ thị được sử dụng trong nhiều cơ sở dữ liệu vector. Tìm kiếm lân cận gần nhất mà không có chỉ mục liên quan đến việc tính toán khoảng

cách từ truy vấn đến từng điểm trong cơ sở dữ liệu, đối với các tập dữ liệu lớn, điều này là quá sức về mặt tính toán. Đối với dữ liệu có nhiều chiều, các kỹ thuật tìm kiếm vector chính xác dựa trên cây như cây k-d và cây R không hoạt động đủ tốt do tính chất phức tạp của tính đa chiều. Để khắc phục điều này, các tìm kiếm lân cận gần nhất k đã được đề xuất, chẳng hạn như băm nhảy cảm và lượng tử hóa tích đánh đổi hiệu suất để lấy độ chính xác. Đồ thị của thuật toán thế giới nhỏ có điều hướng theo phân cấp cung cấp tìm kiếm lân cận gần nhất k có thể mở rộng theo logarit ngay cả trong dữ liệu có nhiều chiều.

Quy trình:

- Xây dựng đồ thị: Các vector được phân cụm hoặc gán ngẫu nhiên vào các node. Tạo cạnh giữa các node có vector tương đồng.
- Truy vấn: Bắt đầu từ một node gốc, thuật toán điều hướng qua đồ thị để đến node có vector gần nhất với vector truy vấn.

Ưu điểm:

- Cung cấp độ chính xác cao nhờ điều hướng thông minh.
- Tối ưu hóa tốc độ truy vấn cho tập dữ liệu lớn.

Nhược điểm:

- Chi phí xây dựng đồ thị ban đầu cao.
- Cần tài nguyên để duy trì cấu trúc đồ thị khi thêm dữ liệu mới.

### 3.1.8. Lọc siêu dữ liệu

Trong cơ sở dữ liệu vector, mỗi vector được lưu trữ thường đi kèm với siêu dữ liệu. Siêu dữ liệu cung cấp các thông tin bổ sung giúp cải thiện khả năng truy vấn và lọc kết quả. Ngoài việc tìm kiếm vector tương đồng, cơ sở dữ liệu vector còn có thể lọc kết quả dựa trên các truy vấn siêu dữ liệu. Tuy nhiên, quá trình lọc siêu dữ liệu không đơn giản và có thể làm chậm quá trình truy vấn nếu không được tối ưu hóa.

#### 3.1.8.1. Cách hoạt động của lọc siêu dữ liệu

Cơ sở dữ liệu vector duy trì 2 chỉ mục riêng biệt:

- Chỉ mục cho các vector, hỗ trợ tìm kiếm tương đồng.
- Chỉ mục cho siêu dữ liệu, hỗ trợ lọc dựa trên các thuộc tính bổ sung của vector.

Lọc siêu dữ liệu có thể được thực hiện trước hoặc sau khi thực hiện tìm kiếm vector. Mỗi phương pháp có ưu và nhược điểm riêng.



### 3.1.8.2. Các phương pháp lọc siêu dữ liệu

#### a. Lọc Trước

Cách thức: Lọc siêu dữ liệu được thực hiện trước khi tìm kiếm vector.

Hệ thống sẽ thu hẹp không gian tìm kiếm bằng cách chỉ chọn các vector thỏa mãn tiêu chí siêu dữ liệu.

Sau đó, quá trình tìm kiếm vector tương đồng sẽ chỉ áp dụng trên không gian đã lọc.

Ưu điểm:

- Giảm không gian tìm kiếm, tăng tốc độ tìm kiếm vector.
- Tiết kiệm tài nguyên xử lý bằng cách loại bỏ các vector không phù hợp ngay từ đầu.

Nhược điểm:

- Có nguy cơ bỏ sót các kết quả phù hợp nhưng không đáp ứng đầy đủ tiêu chí siêu dữ liệu.
- Khi tiêu chí siêu dữ liệu quá phức tạp, quá trình lọc có thể tiêu tốn nhiều tài nguyên, làm chậm hệ thống.

#### b. Lọc Sau

Cách thức: Lọc siêu dữ liệu được thực hiện sau khi tìm kiếm vector.

Toàn bộ không gian tìm kiếm vector được quét để tìm kết quả tương đồng.

Sau đó, các kết quả được kiểm tra lại và loại bỏ nếu không đáp ứng tiêu chí siêu dữ liệu.

Ưu điểm:

- Đảm bảo không bỏ sót bất kỳ kết quả tương đồng nào, kể cả khi không đáp ứng tiêu chí siêu dữ liệu.
- Linh hoạt hơn trong việc áp dụng các bộ lọc phức tạp.

Nhược điểm:

- Làm tăng thời gian xử lý do phải kiểm tra lại từng kết quả.
- Có thể trả về nhiều kết quả không liên quan trước khi áp dụng bộ lọc siêu dữ liệu.

### 3.1.8.3. Tối ưu hóa lọc siêu dữ liệu

Để cân bằng giữa hiệu suất tìm kiếm và độ chính xác của lọc siêu dữ liệu, cơ sở dữ liệu vector sử dụng nhiều kỹ thuật tối ưu hóa:

Phương pháp chỉ mục nâng cao:

- Sử dụng các chỉ mục siêu dữ liệu tiên tiến, chẳng hạn như B-trees hoặc các cấu trúc dữ liệu khác, để tăng tốc độ lọc siêu dữ liệu.
- Tạo mối liên kết giữa siêu dữ liệu và chỉ mục vector để giảm thời gian xử lý.

Xử lý song song

- Sử dụng nhiều luồng xử lý để thực hiện lọc siêu dữ liệu và tìm kiếm vector cùng lúc.
- Kỹ thuật này giúp giảm độ trễ và cải thiện hiệu suất tổng thể.

Tối ưu hóa tỷ lệ lọc

- Lựa chọn giữa lọc trước hay lọc sau tùy thuộc vào đặc thù của dữ liệu và yêu cầu truy vấn.
- Tùy chỉnh quá trình lọc để đạt được sự cân bằng giữa tốc độ và độ chính xác.

#### 3.1.8.4. Ứng dụng thực tế

Lọc siêu dữ liệu rất quan trọng trong nhiều ứng dụng:

- **Tìm kiếm ngữ nghĩa:** Lọc kết quả dựa trên các thuộc tính như ngày tạo, danh mục, hoặc từ khóa.
- **Hệ thống đề xuất:** Gợi ý sản phẩm phù hợp với sở thích và hành vi của người dùng.
- **Phân tích dữ liệu đa chiều:** Truy vấn dữ liệu dựa trên các thuộc tính cụ thể để tìm mối liên hệ hoặc xu hướng.

Lọc siêu dữ liệu là một tính năng quan trọng của cơ sở dữ liệu vector, giúp tăng cường tính chính xác và độ linh hoạt trong truy vấn. Dù áp dụng phương pháp lọc trước hay lọc sau, việc tối ưu hóa quá trình lọc là cần thiết để đảm bảo hiệu suất cao và kết quả phù hợp. Các kỹ thuật như chỉ mục nâng cao và xử lý song song là chìa khóa để đạt được sự cân bằng giữa tốc độ và độ chính xác trong các ứng dụng AI hiện đại.

### 3.2. Các giải thuật tìm kiếm tương đồng

#### 3.2.1. Độ tương đồng với cosine

Độ tương đồng với cosine là một phương pháp đo lường độ tương đồng giữa hai vector trong không gian nhiều chiều. Phương pháp này xác định độ tương đồng dựa trên góc giữa hai vector thay vì độ lớn của chúng. Do đó, độ tương đồng với cosine thường được sử dụng trong xử lý ngôn ngữ tự nhiên và các hệ thống truy vấn tài liệu để so sánh sự tương đồng giữa các tài liệu.

Công thức tính toán độ tương đồng với cosine của hai vector A và B được định nghĩa như công thức (1)

$$\text{Cos}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad (1)$$

- $A \cdot B$ : Tích vô hướng của hai vector A và B.
- $\|A\|$  và  $\|B\|$ : Độ lớn của hai vector.

Ý nghĩa của độ tương đồng với cosine :

- Giá trị cosine dao động từ 0 đến 1.
- Nếu giá trị càng gần 1, hai vector càng tương đồng.
- Nếu giá trị bằng 0, hai vector hoàn toàn không liên quan.

### 3.2.2. Khoảng cách Euclid

Khoảng cách Euclid tính toán khoảng cách giữa hai vector bằng cách lấy tổng của các giá trị vector bình phương. Khoảng cách có thể có bất kỳ giá trị nào từ 0 đến vô cùng. Nếu khoảng cách là 0, các vector là giống hệt nhau. Khoảng cách càng lớn, các vector càng cách xa nhau.

Công thức tính khoảng cách Euclid được định nghĩa như (2).

$$L2 = \sum_{i=1}^n (x_i - y_i)^2 \quad (2)$$

Trong đó:

- $x_i, y_i$  là các phần tử tại vị trí I của hai vector X và Y.
- n là số chiều của vector.

### 3.2.3. Tích vô hướng

Tích vô hướng là một phép toán giữa hai vector, trong đó các phần tử tương ứng của hai vector được nhân với nhau và cộng lại để tạo ra một giá trị số.

Tích vô hướng cho thấy độ tương đồng giữa hai vector. Cụ thể:

- Tích vô hướng có giá trị dương khi hai vector có cùng hướng
- Tích vô hướng có giá trị âm khi hai vector có hướng ngược nhau.
- Tích vô hướng bằng 0 khi hai vector vuông góc với nhau, không liên quan.

Công thức tính tích vô hướng của hai vector được định nghĩa như (3):

$$A \cdot B = \sum_{i=1}^n A_i \cdot B_i \quad (3)$$

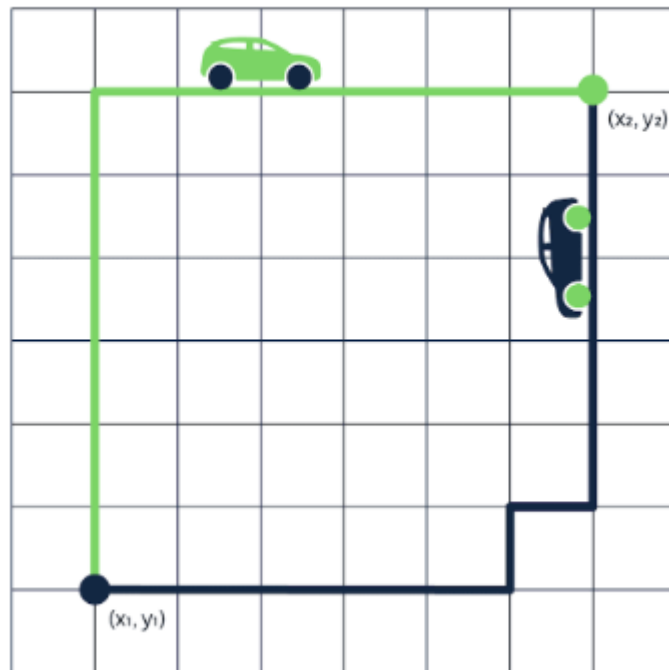
Trong đó:

- $A_i, B_i$  là các phần tử của vector  $A$  và  $B$  tại chỉ số  $i$ .
- $n$  là số chiều của các vector.

### 3.2.4. Khoảng cách Manhattan

Khoảng cách Manhattan tính toán khoảng cách giữa một cặp vector. Là một cách đo khoảng cách giữa hai điểm trong không gian dựa trên việc di chuyển theo trục ngang và trục dọc (giống như đi trên đường phố Manhattan).

Tên gọi này xuất phát từ cách bố trí dạng lưới giống như đường phố Manhattan (Hình 3.8). Thành phố được thiết kế với các tòa nhà ở mọi góc phố và đường một chiều. Nếu bạn muốn đi từ điểm A đến điểm B, đường ngắn nhất không phải là đường thẳng vì bạn không thể lái xe qua các tòa nhà. Tuyến đường nhanh nhất là tuyến đường có ít khúc cua và ngã rẽ hơn.



Hình 3.6: Hình minh họa cho thuật toán Manhattan (nguồn: <https://weaviate.io/blog/distance-metrics-in-vector-search>)

Công thức tính khoảng cách Manhattan giữa hai điểm  $A=[a_1, a_2, \dots, a_n]$ ,  $B = [b_1, b_2, \dots, b_n]$  trong không gian  $n$  chiều được định nghĩa như (4):

$$\text{Khoảng cách Manhattan} = \sum_{i=1}^n |a_i - b_i| \quad (4)$$

Trong đó:

- $a_i$  và  $b_i$  là các phần tử của vector  $A$  và  $B$  tại chỉ số  $i$ .
- $|a_i - b_i|$  là giá trị tuyệt đối của sự khác biệt giữa các phần tử tương ứng.

### 3.2.5. Thuật toán Hamming

#### 3.2.5.1. Khoảng cách Hamming

Khái niệm: Khoảng cách Hamming là một thước đo để so sánh hai vector (có thể là vector số hoặc nhị phân). Nó tính số vị trí cần thay đổi để biến vector này thành vector kia.

Ý nghĩa:

- Khoảng cách Hamming càng nhỏ thì hai vector càng giống nhau.
- Thường được áp dụng trong các bài toán phân loại, nhận dạng mẫu và so khớp dữ liệu.

#### 3.2.5.2. Truy xuất đoạn nhị phân

Khái niệm: Truy xuất đoạn nhị phân là một phương pháp chuyển vector thành chuỗi nhị phân (0 hoặc 1) để lưu trữ hiệu quả hơn trong hệ thống mà vẫn giữ được cấu trúc ngữ nghĩa của dữ liệu.

Cách tính Khoảng cách Hamming trong truy xuất đoạn nhị phân: So sánh từng bit (0 hoặc 1) giữa hai chuỗi nhị phân bằng phép toán XOR.

Lợi ích:

- Chuỗi nhị phân giúp tiết kiệm bộ nhớ và lưu trữ lượng dữ liệu lớn trong bộ nhớ máy tính.
- Phù hợp với các hệ thống yêu cầu tốc độ xử lý cao và truy xuất dữ liệu nhanh.

### 3.3. Cơ sở dữ liệu Qdrant

#### 3.3.1. Giới thiệu cơ sở dữ liệu Qdrant



*Hình 3.7: Cơ sở dữ liệu Qdrant*

Qdrant là một cơ sở dữ liệu vector nguồn mở, được thiết kế để lưu trữ và truy vấn các vector embedding. Đây là một giải pháp tối ưu cho các bài toán tìm kiếm tương đồng và phân tích dữ liệu dựa trên vector. Với kiến trúc hiện đại và dễ tích hợp, Qdrant phù hợp với các ứng dụng AI yêu cầu hiệu suất cao và quản lý dữ liệu phức tạp.

Các Đặc Điểm Chính:

- Tích hợp dễ dàng: Qdrant hỗ trợ REST API, giúp dễ dàng tích hợp vào nhiều nền tảng khác nhau.
- Quản lý siêu dữ liệu mạnh mẽ: Cho phép gắn siêu dữ liệu với các vector, hỗ trợ truy vấn linh hoạt và chi tiết.
- Hỗ trợ “thuật toán tìm kiếm láng giềng gần đúng”: Tối ưu hóa tốc độ và hiệu suất truy vấn vector.
- Khả năng mở rộng: Qdrant có thể mở rộng để xử lý dữ liệu lớn và hỗ trợ song song nhiều truy vấn.
- Nguồn mở: Dễ dàng tùy chỉnh và triển khai trong các môi trường khác nhau.

Ứng Dụng Chính:

- Tìm kiếm ngữ nghĩa trong văn bản, hình ảnh, hoặc âm thanh.
- Hệ thống gợi ý cá nhân hóa.
- Phân tích dữ liệu trong không gian vector.
- Tìm kiếm tương đồng.

### 3.3.2. Ưu điểm và nhược điểm của cơ sở dữ liệu Qdrant

#### 3.3.2.1. Ưu Điểm

- Hiệu suất cao: Qdrant tối ưu hóa cho truy vấn các láng giềng gần đúng, cung cấp kết quả nhanh chóng ngay cả với tập dữ liệu lớn.
- Quản lý siêu dữ liệu tốt: Cho phép lưu trữ và truy vấn dữ liệu dựa trên các bộ lọc siêu dữ liệu chi tiết.
- Tích hợp linh hoạt: Hỗ trợ REST API và gRPC, dễ dàng tích hợp với các ứng dụng hiện đại.
- Nguồn mở và cộng đồng hỗ trợ: Dễ dàng truy cập, tùy chỉnh và nhận sự hỗ trợ từ cộng đồng người dùng.
- Khả năng mở rộng: Hỗ trợ môi trường phân tán, giúp xử lý dữ liệu lớn một cách hiệu quả.

#### 3.3.2.2. Nhược Điểm

- Chi phí tài nguyên: Việc xử lý và lưu trữ dữ liệu vector lớn đòi hỏi phần cứng mạnh và bộ nhớ cao.
- Yêu cầu kỹ thuật cao: Đòi hỏi kiến thức về cơ sở dữ liệu vector để triển khai và tối ưu hóa hiệu quả.
- Hạn chế trong truy vấn phức tạp: Các truy vấn kết hợp nhiều lớp lọc siêu dữ liệu và vector có thể chậm trong một số trường hợp.

### 3.3.3. Triển khai dữ liệu lên Qdrant

#### 3.3.3.1. Đăng ký một VPS

a. Dùng thẻ ghi nợ quốc tế đăng ký một tài khoản.

b. Cách tạo máy ảo EC2.

Bước 1: Đăng nhập vào AWS Management Console

- Truy cập [AWS Management Console](#).
- Đăng nhập vào tài khoản AWS.

Bước 2: Truy cập vào dịch vụ EC2

- Tại màn hình chính của AWS Console, tìm và chọn EC2.

### Bước 3: Khởi tạo một Instance mới

- Trong giao diện EC2 Dashboard, chọn Launch Instance.
- Đặt Tên cho instance để dễ quản lý và nhận diện.

### Bước 4: Chọn Amazon Machine Image

- Chọn hệ điều hành muốn cài đặt cho instance. Ví dụ: Amazon Linux 2 (miễn phí trong gói Free Tier)
- Xác nhận loại AMI chúng ta chọn có nằm trong gói Free Tier nếu đang sử dụng AWS Free Tier.

### Bước 5: Chọn loại Instance

- Chọn loại instance phù hợp. Nếu đang trong gói Free Tier, hãy chọn t2.micro hoặc t3.micro.
- Nhấn Next hoặc Review and Launch nếu không cần điều chỉnh thông số nâng cao.

### Bước 6: Cấu hình các thông số của Instance

- Number of Instances: Để mặc định là 1.
- Network: Để mặc định.
- Các thông số khác có thể giữ nguyên trừ khi chúng ta muốn cài đặt mạng hoặc phân bổ CPU theo nhu cầu cụ thể.

### Bước 7: Cấu hình dung lượng lưu trữ

- Dung lượng mặc định cho Amazon Linux hoặc Ubuntu thường là 8 GB (có thể điều chỉnh tùy theo nhu cầu).
- Nếu trong gói Free Tier, có thể dùng tối đa 30 GB lưu trữ SSD miễn phí.

### Bước 8: Thiết lập Security Group

- Security Group là nhóm quy tắc tường lửa.
- Nhấn Create a new security group và thêm quy tắc cho SSH.
- Type: Chọn SSH (cho phép kết nối từ xa).
- Port: Đặt là 22 cho SSH.
- Source: Chọn My IP (nên dùng để bảo mật), hoặc Anywhere (0.0.0.0/0) nếu muốn truy cập từ mọi nơi.



- Để mở cổng cho ứng dụng khác, chọn Add Rule và thêm quy tắc cho từng ứng dụng, như HTTP (80) hoặc HTTPS (443).

#### Bước 9: Xem lại và khởi chạy Instance

- Xem lại tất cả cài đặt và nhấn Launch.
- AWS sẽ yêu cầu chọn hoặc tạo cặp khóa để truy cập instance qua SSH.
- Nếu chưa có, chọn Create a new key pair, đặt tên và tải file .ppk về máy.
- Lưu file này cẩn thận – chúng ta sẽ cần nó để truy cập instance qua SSH.

#### c. Cách kết nối điều khiển từ xa và cài đặt Qdrant trên EC2.

- Kết nối đến EC2 qua Putty

#### Bước 1: Chuyển đổi khóa sang định dạng Putty (.ppk).

- Tải PuTTY nếu chưa có.
- Mở PuTTY, chọn Load và tải tệp khóa .pem mà AWS cung cấp khi chúng ta tạo instance.
- Sau đó, nhấp Save private key để lưu thành định dạng .ppk (tệp này sẽ dùng để kết nối qua Putty).

#### Bước 2: Mở Putty và kết nối.

- Mở Putty, nhập Public Ipv4 address của EC2 (tìm thấy trong trang quản lý instance của AWS) vào mục Host Name.
- Chuyển đến Connection > SSH > Auth, nhấp Browse và chọn tệp .ppk mà chúng ta vừa lưu. (Lưu ý phải xóa tất cả quyền truy cập của file này chỉ để lại quyền administrator thôi, không thì sẽ thất bại).
- Quay lại tab Session, lưu cấu hình nếu muốn và nhấn Open để mở kết nối.
- Chúng ta sẽ thấy màn hình yêu cầu nhập “login as”, hãy nhập “ec2-user” và nhấn Enter.

#### 3.3.3.2. Triển khai Qdrant

- Cài đặt Docker

#### Bước 1: Để cài Docker trên Amazon Linux, chạy lệnh sau:

```
sudo yum update -y
```

```
sudo yum install -y docker
```

Bước 2: Sau khi cài đặt, khởi động Docker:

```
sudo service docker start
```

Bước 3: Thêm người dùng vào nhóm Docker để chạy Docker không cần sudo:

```
sudo usermod -aG docker ec2-user
```

Bước 4. Đăng xuất và đăng nhập lại để áp dụng thay đổi nhóm.

- Triển khai Qdrant

Bước 1: Kéo ảnh Qdrant từ Docker Hub, sử dụng docker image của qdrant để triển khai dễ dàng hơn là cấu hình thủ công.

```
Docker pull qdrant/qdrant
```

Bước 2: Chạy container Qdrant:

```
docker run -d -p 6333:6333 qdrant/qdrant
```

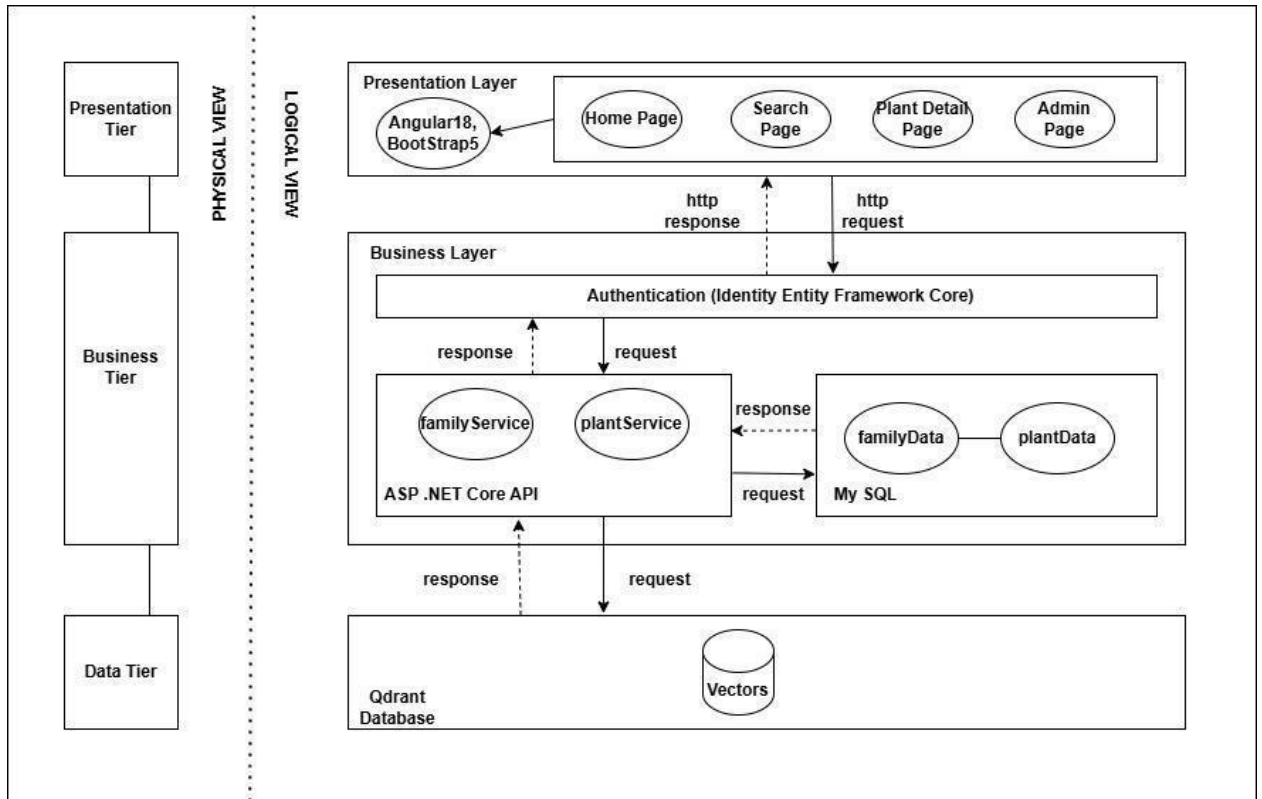
- Lệnh docker run -d -p 6333:6333 qdrant/qdrant có các ý nghĩa như sau:

- docker run: Đây là lệnh được sử dụng để tạo và chạy một container mới từ một image trong Docker.
- -d: Tùy chọn này có nghĩa là chạy container ở chế độ nền. Điều này có nghĩa là container sẽ chạy trong nền và chúng ta sẽ không thấy output của nó trong terminal. Nếu chúng ta không sử dụng tùy chọn này, terminal sẽ bị chiếm giữ bởi container cho đến khi nó dừng lại.
- -p 6333:6333: Tùy chọn này thiết lập ánh xạ cổng. Phần trước dấu hai chấm (6333) là cổng trên máy chủ EC2 mà chúng ta sẽ sử dụng để truy cập dịch vụ Qdrant. Phần sau dấu hai chấm (6333) là cổng mà container Qdrant đang lắng nghe bên trong. Nói cách khác, khi chúng ta truy cập `http://<your-ec2-public-ip>:6333`, yêu cầu sẽ được chuyển đến cổng 6333 bên trong container, nơi mà Qdrant đang hoạt động.

- qdrant/qdrant: Đây là tên của image Docker mà chúng ta muốn chạy. Trong trường hợp này, nó là image Qdrant từ Docker Hub. Nếu chúng ta chưa tải image này, Docker sẽ tự động tải về trước khi chạy.
- Cấu hình bảo mật cho cổng 6333
- Truy cập trang quản lý Security Groups trong AWS.
  - Tìm và chọn Security Group của EC2, thêm quy tắc Custom TCP Rule cho cổng 6333, với nguồn là địa chỉ IP cụ thể hoặc 0.0.0.0/0 nếu muốn cho phép truy cập từ mọi nơi.
- Kiểm tra Qdrant: Mở trình duyệt và truy cập <http://<your-ec2-public-ip>:6333/dashboard> để kiểm tra kết nối đến Qdrant.

## CHƯƠNG 4. XÂY DỰNG MÔ HÌNH XỬ LÝ

### 4.1. Kiến trúc hệ thống



Hình 4.1: Kiến trúc phần mềm

Hệ thống (Hình 4.1) gồm ba tầng chính: Tầng giao diện, tầng xử lý các tính toán logic, và tầng dữ liệu. Mỗi tầng có một vai trò cụ thể riêng, từ giao diện người dùng đến xử lý logic và quản lý dữ liệu, đảm bảo hiệu quả trong hoạt động của hệ thống. Dưới đây là phần mô tả chi tiết các thành phần của kiến trúc.

#### 4.1.1. Khung nhìn logic

Khung nhìn logic mô tả cách các thành phần trong hệ thống tương tác với nhau trên phương diện logic, bao gồm các thành phần trong từng tầng và mối quan hệ giữa chúng.

##### 4.1.1.1. Lớp giao diện

Lớp giao diện chịu trách nhiệm về giao diện người dùng, sử dụng công nghệ Angular 18 kết hợp với Bootstrap 5 để cung cấp giao diện trực quan và thân thiện người dùng. Nó bao gồm các thành phần sau:

- Home Page: Trang chủ của hệ thống, gồm các chức năng như hiển thị các cây thuốc được tìm kiếm nhiều nhất, hiển thị các cây thuốc quý và những thông tin bạn có thể chưa biết về một số cây thuốc.
- Search Page: Trang tìm kiếm, cho phép người dùng nhập các truy vấn để tìm kiếm thông tin liên quan đến cây thuốc, cũng như tìm kiếm các cây thuốc có độ tương đồng về dược tính.
- Plant Detail Page: Trang chi tiết, hiển thị thông tin cụ thể về một cây thuốc cụ thể khi người dùng nhấn chọn.
- Admin Page: Trang quản trị, cho phép người quản lý thực hiện các tác vụ như xem thống kê cũng như thêm, sửa, hoặc xóa dữ liệu về họ cây thuốc, cây thuốc.

Lớp này gửi các yêu cầu HTTP đến lớp xử lý logic và nhận phản hồi HTTP từ nó. Các dữ liệu được hiển thị trên giao diện đều xuất phát từ các lớp phía sau.

#### **4.1.1.2. Lớp xử lý logic**

Lớp này là trung tâm xử lý của hệ thống, đảm nhiệm việc thực hiện các logic nghiệp vụ phức tạp và giao tiếp giữa lớp giao diện và lớp dữ liệu. Các thành phần chính trong lớp này bao gồm:

- Hệ thống xác thực (sử dụng Identity Entity Framework Core)
  - Xử lý việc xác thực người dùng, đảm bảo rằng chỉ các tài khoản hợp lệ ví dụ như người quản trị mới có thể truy cập vào hệ thống để thực hiện các thao tác thêm, xóa, sửa dữ liệu.
  - Tích hợp với ASP.NET Core Identity để quản lý thông tin đăng nhập, tạo token, và kiểm soát quyền truy cập.
- ASP.NET Core API
  - Cung cấp API cho các thao tác giữa lớp giao diện và lớp dữ liệu.
  - Đóng vai trò trung gian, xử lý các yêu cầu từ lớp giao diện và gửi chúng đến lớp dữ liệu, sau đó xử lý, tính toán và trả về phản hồi.

familyService: Xử lý logic liên quan đến dữ liệu của họ cây thuốc.

plantService: Xử lý logic liên quan đến cây thuốc.

- Cơ sở dữ liệu MySQL

- Lưu trữ dữ liệu quan hệ như thông tin về họ cây thuốc và cây thuốc.

- Các API trong lớp logic truy cập dữ liệu trong cơ sở dữ liệu MySQL thông qua Entity Framework Core.

#### 4.1.1.3. Lớp dữ liệu

Lớp dữ liệu chịu trách nhiệm quản lý và lưu trữ dữ liệu về các vector thành phần hóa học của các cây thuốc. Sử dụng cơ sở dữ liệu Qdrant để lưu trữ vector embedding để hỗ trợ tìm kiếm tương đồng dựa trên vector.

Đây là phần cốt lõi cho chức năng tìm kiếm các cây thuốc có độ tương đồng về được tính. Hệ thống gửi các vector truy vấn đến Qdrant và nhận lại danh sách các vector gần nhất.

#### 4.1.2. Khung nhìn vật lý

Khung nhìn vật lý biểu diễn cách các thành phần được triển khai trên hạ tầng vật lý.

##### 4.1.2.1. Tầng giao diện

Tầng này chạy trên trình duyệt của người dùng và không yêu cầu cài đặt đặc biệt. Nó giao tiếp với các API ở tầng logic thông qua HTTP. Tất cả các logic giao diện đều được thực hiện trên máy khách.

##### 4.1.2.2. Tầng xử lý logic

Tầng xử lý logic được triển khai trên một máy chủ chạy ASP.NET Core API. Phần này gồm các thành phần sau:

- Hệ thống xác thực: Tích hợp Identity Framework Core để quản lý người dùng.
- Dịch vụ API: Được triển khai để xử lý các yêu cầu HTTP từ tầng giao diện.
- Gồm cả cơ sở dữ liệu MySQL để lưu trữ dữ liệu quan hệ.

##### 4.1.2.3. Tầng dữ liệu

Tầng dữ liệu là một máy ảo EC2 của AWS, để triển khai cơ sở dữ liệu vector Qdrant nhằm mục đích lưu trữ dữ liệu vector của các cây thuốc.

#### 4.1.3. Quy trình xử lý yêu cầu

Dưới đây là quy trình diễn hình khi người dùng tương tác với hệ thống:

- Người dùng gửi yêu cầu từ lớp giao diện:
  - Ví dụ: Truy vấn danh sách các cây thuốc theo tên.

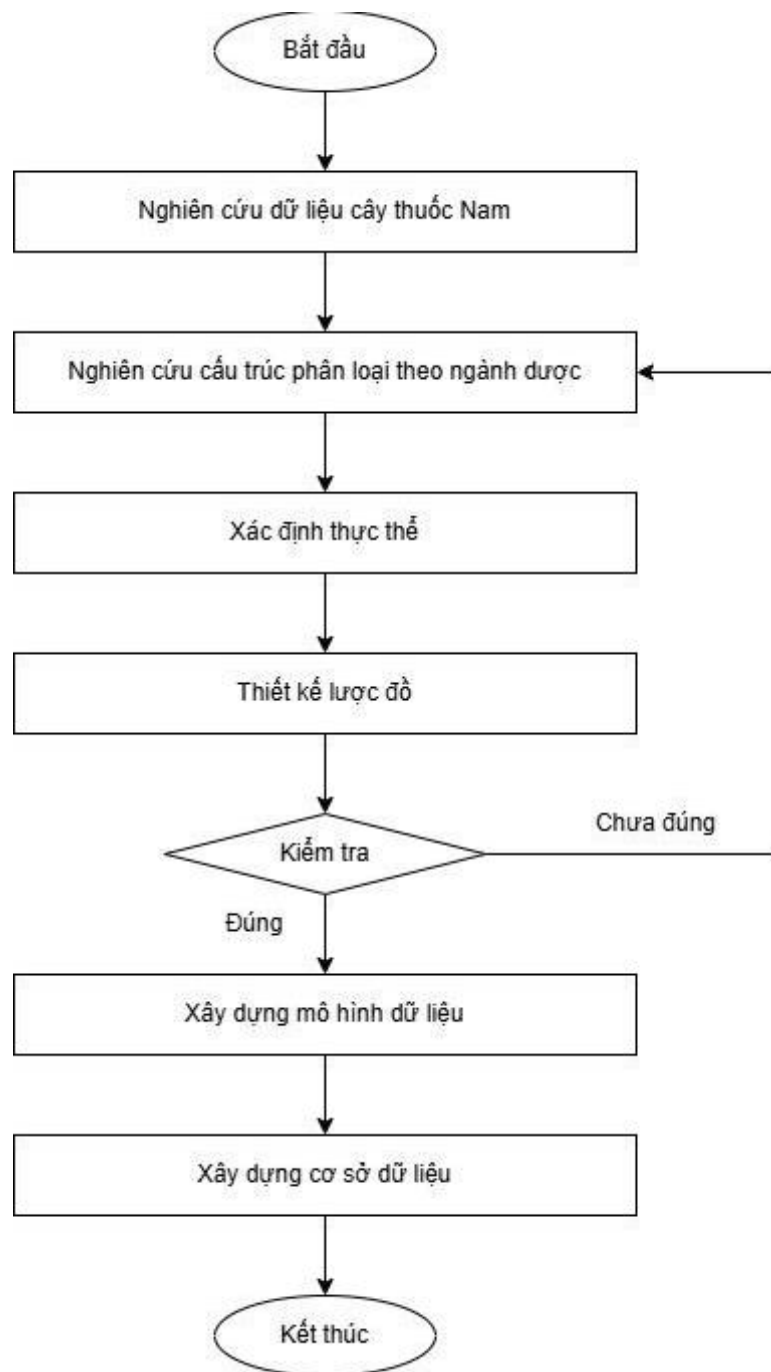
- Lớp giao diện gửi yêu cầu HTTP đến lớp logic.
- API xử lý yêu cầu trong lớp logic:
  - API nhận yêu cầu và kiểm tra xác thực người dùng thông qua hệ thống xác thực.
- Gửi yêu cầu đến:
  - MySQL: Nếu yêu cầu liên quan đến thông tin quan hệ như họ cây thuốc hoặc thông tin cây thuốc.
  - Qdrant: Nếu yêu cầu liên quan đến tìm kiếm vector embedding.
- MySQL xử lý dữ liệu quan hệ:
  - Truy vấn thông tin họ cây thuốc hoặc cây thuốc từ cơ sở dữ liệu quan hệ.
  - Gửi dữ liệu trở lại tầng logic thông qua familyService hoặc plantService.
- Qdrant xử lý vector embedding (nếu cần):
  - Với các truy vấn tìm kiếm các cây thuốc có độ tương đồng về dược tính, API gửi yêu cầu đến máy chủ chứa cơ sở dữ liệu vector.
  - Qdrant trả về danh sách vector tương đồng, và kết quả được chuyển về lớp giao diện.
- Kết quả trả về lớp giao diện:
  - Dữ liệu được trả về thông qua giao thức HTTP và hiển thị trên giao diện người dùng.

#### 4.1.4. Ưu điểm của kiến trúc

- Phân tầng rõ ràng: Tách biệt giữa các tầng giao diện, logic và dữ liệu, giúp dễ bảo trì và mở rộng.
- Hiệu suất cao: MySQL tích hợp trực tiếp vào tầng logic, giảm độ trễ; Còn Qdrant thì tối ưu hóa tìm kiếm vector.
- Tích hợp linh hoạt: Kết hợp công nghệ Angular, Bootstrap, ASP.NET Core API, và Qdrant giải quyết được bài toán đặt ra.
- Bảo mật cao: Tích hợp ASP.NET Core Identity.
- Tính thực tiễn cao: Dễ triển khai, phù hợp với nhiều bài toán từ tìm kiếm ngữ nghĩa đến quản lý dữ liệu.

## 4.2. Mô hình xử lý dữ liệu

### 4.2.1. Mô hình tổng quát



Hình 4.2: Mô hình tổng quát



#### 4.2.2. Cấu trúc của dữ liệu cây thuốc

##### A kê

1. Tên thường gọi: A kê (Akee)
2. Tên khoa học: *Blighia sapida* Koen thuộc họ Bồ Hòn – Sapindaceae.
3. Mô tả: Cây gỗ cao 12-13m, thẳng, vỏ tán xoè rộng và các nhánh cứng. Lá to mang 8 lá chét gần như mọc đối, hình bầu dục thuôn dài, dài đến 13cm, rộng 5cm, không lông, màu ôliu tươi, lúc khô gan rất lồi ở mặt dưới. Chùm hoa 1-2 ở nách lá có lông mịn, cuống hoa dài 1,5cm; lá dài 5; cánh hoa 5 màu trắng, hẹp, có một vảy to ở gốc; nhị 8; bầu 3 ô có vòi nhụy ngắn. Quả nang dài từ 7-10cm hình tam giác, màu vàng hay đỏ tươi, mở thành 3 mảnh: hạt 3, tròn, bóng loáng, mang áo hạt to ở gốc màu trắng, bao đến 1/2 hạt. Loài cây mọc ở Trung Phi, được nhập trồng ở Đồng Nai làm cây cảnh



4. Bộ phận dùng: Áo hạt, lá, vỏ và quả - Arillus, Folium et Fructus *Blighiae*.

5. Thành phần hoá học: Áo hạt chứa 69,2% nước, 5% protid, 20% lipid, 4,6% glucid, 1,2% tro, 40% Ca, 16 mg% P, 2,7 mg% Fe. Cuống noãn giữa hạt và áo hạt chứa 2 chất độc mà một chất là hypoglucin A. Quả chứa những nhân tố làm giảm glycoza huyết thông thường (amino acid cyclopropanoid, hypoglucin A và hypoglucin B) đều là chất độc có thể dùng thay insulin. Tro quả chứa nhiều Kalium.

6. Tính vị, tác dụng: Áo hạt cứng có dầu và có mùi vị của hạt dẻ, dùng ăn được. Ở Châu Âu, người ta cho đó là một thức ăn làm ngon miệng, khi đem nấu chín trong bơ hoặc với cá biển. Tác dụng trị lý và sốt, chữa cảm lạnh và chảy mũi, giảm đau, chống độc, chống nôn, cũng là chất độc và kích thích v.v...

7. Cách dùng: Áo hạt được dùng là thuốc trị lý và sốt. Lá (và vỏ) sắc nước uống lợi tiêu hoá, cũng dùng chữa cảm lạnh và chảy mũi.

Nhiều bộ phận khác của cây được xem như làm giảm đau, chống độc, chống nôn, cũng là chất độc và kích thích. Được dùng làm thuốc trị viêm kết mạc, phù thũng, đau nửa đầu, đau mắt, viêm tinh hoàn, loét, ghẻ cóc và sốt vàng da.

4

Hình 4.3: Ảnh minh họa về thông tin của một cây thuốc

Dựa vào phân loại và dược tính của ngành dược, cây thuốc được xác định các thực thể:

- Họ cây thuốc:

- ID
- Tên khoa học
- Tên tiếng việt

- Cây thuốc:

- ID
- Họ cây thuốc
- Tên khoa học
- Tên tiếng việt
- Mô tả

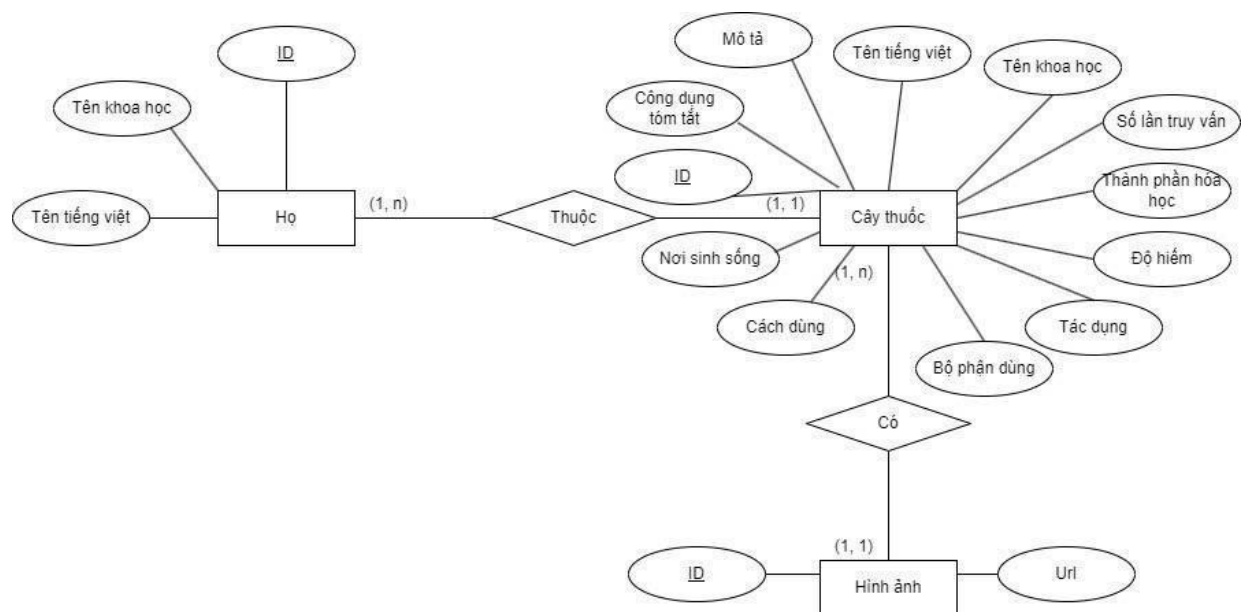
- Nơi sinh sống
- Cách dùng
- Tác dụng
- Bộ phận dùng
- Thành phần hóa học

- Hình ảnh cây thuốc:

- ID
- Đường dẫn hình ảnh

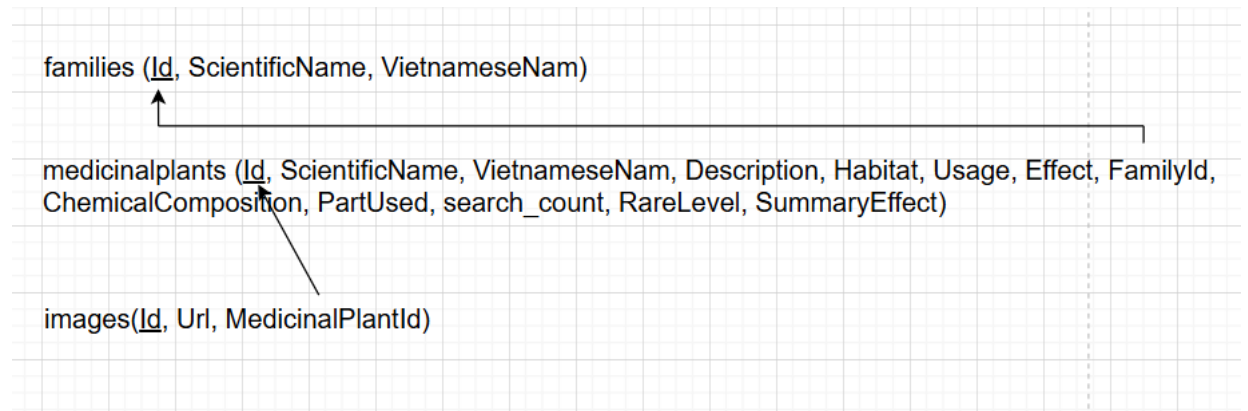
#### 4.2.3. Thiết kế lược đồ cơ sở dữ liệu

Sau khi đã xác định được các thực thể và các thuộc tính lưu trữ cần thiết, tiến hành thiết kế lược đồ cơ sở dữ liệu như hình 4.4.



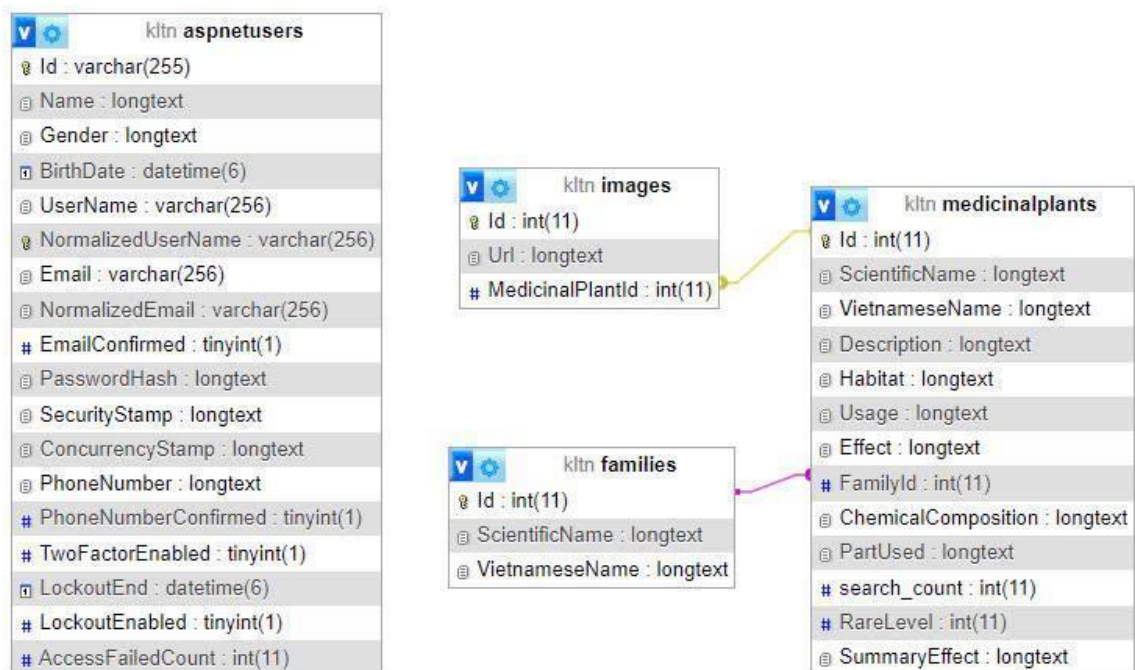
Hình 4.4: Lược đồ cơ sở dữ liệu

(Các mô tả đã được trình bày trong phần 2.3)



Hình 4.5: Mô hình quan hệ

Cơ sở dữ liệu quan hệ sau khi triển khai trong MySQL, (Hình 4.5).



Hình 4.6: Kết quả cài đặt cơ sở dữ liệu

#### 4.2.4. Xây dựng cơ sở dữ liệu

##### 4.2.4.1. Theo họ cây thuốc

STT	ID	Tên họ	Tên khoa học
1	01	Cúc	Asteraceae
2	02	Ô rô	Acanthaceae
3	03	Lưỡi răn	Ophioglossaceae
4	04	Hoa hồng	Rosaceae
5	05	Ấu	Trapaceae
6	06	Bồ Hòn	Sapindaceae
7	07	Thuốc phiện	Papaveraceae
8	08	Thầu dầu	Euphorbiaceae
9	09	Rau dền	Amaranthaceae
10	10	Bấc	Juncaceae
11	11	Vòi voi	Boraginaceae
12	12	Hoa môi	Lamiaceae
13	13	Cam	Rutaceae
14	14	Thanh thất	Sinaroubaceae
15	15	Bách bộ	Stemonaceae
16	16	Trôm	Sterculiaceae
17	17	Lan	Orchidaceae
18	18	Hoa tán	Apiaceae
19	19	Đậu	Fabaceae
20	20	Cẩm chướng	Caryophyllaceae
21	21	Sim	Myrtaceae
22	22	Cỏ roi ngựa	Verbenaceae
23	23	Gừng	Zingiberaceae
24	24	Hoa loa kèn	Liliaceae
25	25	Xoan	Meliaceae
26	26	Long đởm	Gentianaceae
27	27	Nho	Vitaceae

Hình 4.7: Kết quả trích xuất dữ liệu họ cây thuốc

#### 4.2.4.2. Theo cây thuốc

STT	ID	Tên cây thuốc	Tên khoa học	thuộc họ	Mô tả	Bộ phận dùng	TPHH	Tác dụng	Nơi sinh sống	Cách dùng
1	0001	A kê	Blighia sapida Koe	Bồ Hòn – Sap	Cây gỗ cao 12	Áo hạt lá vỏ và	Áo hạt chứa đ	Áo hạt cứng	Loài cây mọc ở Tru	Áo hạt được d
2	0002	A phiện (Thuốc phiện)								
3	0003	Actisô	Cynara scolymus L	Cúc - Asterac	Cây thân thảo	Toàn cây (lá, th	Cụm hoa chứ	Bông Actisô c	Cây mọc ở vùng Ca	Sử dụng Actisô
4	0004	Anh đào	Prunus cerasoides	Hoà hồng – R	Cây gỗ nhỏ c	Quả - Fructus P	Nhân hạt chứ	Quả có vỏ qu	Loài của núi cao Hy	
5	0005	Ắc ó								
6	0006	Ấm địa quyết								
7	0007	Ấu				Quả – Fructus T	Trong nhân h	Củ ấu có vị n	Cây của vùng cổ nh	Người ta trồng
8	0008	Ấu dai								
9	0009	Ấu gai								
10	0010	Ấu nước	Trapa natans L. và	Ấu – Trapace	Cây thủy sinh	Quả - Fructus T	Quả chứa pro	cứng tương t	Cây mọc ở ao hồ đ	
11	0011	Ba bét hoa nhiều	Mallotus floribundus	Thầu dầu - E	Cây nhỏ nh	Lá - Folium Mall	Hạt chứa dầu	Lá dùng nấu u	Cây mọc ở các đ	Lá dùng nấu u
12	0012	Ba bóng								
13	0013	Ba chạc	Euodia lepta (Spre	Cam - Rutace	Cây nhỏ cao	Lá và rễ - Folio	Rễ chứa alcal	Vị đắng mùi th	Rất phổ biến khắp r	
14	0014	Ba chạc Poilane								
15	0015	Ba chẽ	Dendrobium trian	họ Đậu - Fab	Cây nhỏ cao	Lá - Folium Den	Lá chứa tanin	Cây có tác dụ	Cây của vùng Ấn Đ	Nhân dân thườ
16	0016	Ba chìa								
17	0017	Bà chuột								
18	0018	Bà cốt tiêu								
19	0019	Bà dột	Eupatorium triplin	Cúc - Asterac	Cây thảo mọc	Toàn thân chứa	Cây chứa tinh	Nhân dân nh	Cây có nguồn gốc N	Thường dùng h
20	0020	Bà đậu	Croton tiglium L.	Thầu dầu - E	Cây gỗ nhỏ c	Hạt - Fructus C	Hạt chứa kho	Hạt có vị cay	Cây của vùng Ấn Đ	Thường dùng h
21	0021	Bà đậu tây	Hura crepitans L.	Thầu dầu - E	Cây gỗ có kíc	Vỏ cây và hạt -	Cây chứa m	Nhựa mủ của	Gốc ở châu Mỹ nhi	Ta thường dùng
22	0022	Bà gạc	Rauvolfia verticillat	Trúc đào - Ap	Cây nhỏ, cao	Rễ - Radix Rau	Trong rễ và lá	Rễ Bà gạc c	Cây mọc hoang ở H	Hiện nay ta chế
23	0023	Bà gạc Ấn Độ	Rauvolfia serpentin	Trúc đào - Ap	Cây thảo cao	Rễ - Radix Rau	Hàm lượng al	Vỏ rễ có vị đ	Loài có gặp mọc h	Reserpin thườ
24	0024	Bà gạc Châu đốc	Rauvolfia chauli	Trúc đào - Ap	Cây gỗ to, đ	Vỏ thân và vỏ r	Người ta đã p	Reserpin có t	Cây đặc hữu của m	Vỏ rễ sắc nước
25	0025	Bà gạc châu Phi	Rauvolfia vomitoria	Trúc đào - Ap	Cây nhỏ cao	Rễ - Radix Rau	Chủ yếu là al	Cũng như các	Chỉ mới gặp ở huy	




Hình 4.8: Kết quả trích xuất dữ liệu cây thuốc

Tiến hành nghiên cứu và trích xuất ra các dữ liệu cần thiết của các cây thuốc có thể sử dụng, sau đó lưu vào tệp excel. Kết quả như (Hình 4.8).

#### 4.2.5. Xây dựng CSDL với MySQL

				<b>Id</b>	<b>ScientificName</b>	<b>VietnameseName</b>
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				1	Sapindaceae	Bồ Hòn
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				2	Asteraceae	Cúc
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				3	Trapaceae	Ấu
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				4	Euphorbiaceae	Thầu dầu
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				5	Dipterocarpaceae	Dầu
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				6	Apocynaceae	Trúc đào
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				7	Lamiaceae	Hoa môi
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				8	Stemonaceae	Bách bộ
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				9	Myrtaceae	Sim
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				10	Zingiberaceae	Gừng
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				11	Liliaceae	Hoa loa kèn
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				12	Liliaceae	Loa kèn
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				13	Arecaceae	Cau
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				14	Combretaceae	Bàng
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				15	Cucurbitaceae	Bầu bí
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				16	Rutaceae	Cam
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				17	Araceae	Ráy
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				18	Pontederiaceae	Bèo lục bình
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				19	Pontederiaceae	Lục bình
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				20	Lauraceae	Long não
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				21	Capparaceae	Màn màn
<input type="checkbox"/>		Sửa		Chép		Xóa bỏ
				22	Solanaceae	Cà

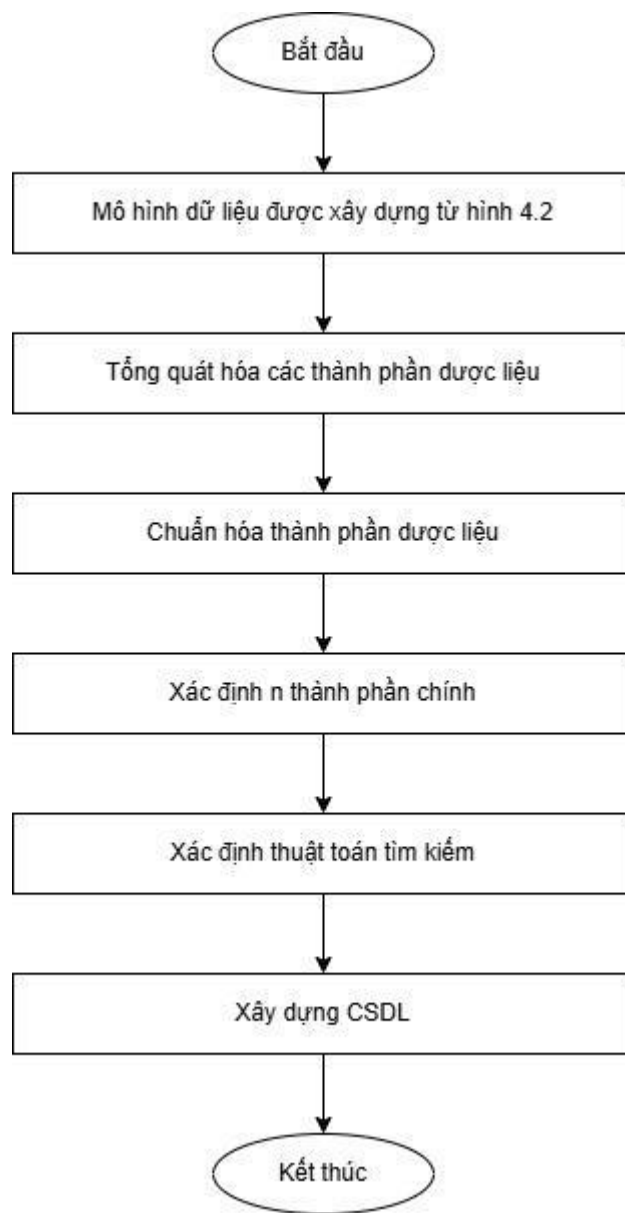
Hình 4.9: Kết quả sau khi chuyển dữ liệu từ excel lên cơ sở dữ liệu của MySQL (phần họ cây thuốc)

			Id	ScientificName	VietnameseName	Description	Habitat	Usage	Effect	FamilyId	ChemicalComposition	PartUsed	search_count	RareLe
<input type="checkbox"/>				1	Blighia sapida Koen	A kê	Cây gỗ cao 12-13m thẳng cso tán xoè rộng và các nh...	Loài cây mọc ở Trung Phi	Áo hạt được dùng là thuốc trị ly và sốt. Lá (và vỏ... Bông	Áo hạt cứng có dầu và có mùi vị của hạt dẻ dùng ăn...	1	Áo hạt chứa 69,2% nước, 5% protid 20% lipid 4,6%5 ...	Áo hạt là vỏ và quả - Arillus, Folium et Fructus B...	16
<input type="checkbox"/>				3	Cynara scolymus L.	Actisô	Cây thân thảo cao 1-1,2 m. Thân cây có lông mềm, có...	Cây mọc ở vùng Carthage và các vùng Địa trung hải...	Sử dụng Actisô dưới nhiều dạng, có thể dùng tươi h... chín, tàng...	Actisô có tính bổ dưỡng khi đã nấu chín, tàng...	2	Cum hoa chứa 3-3,15% protid; 0,1-0,3% lipid; 11-15...	Toàn cây (lá, thân, rễ, cum hoa) – Herba Cynarae S...	13
<input type="checkbox"/>				40	Trapa natans L.	Đào mướp	Cây thủy sinh nổi khả to. Lá...	Cây mọc ở ao hồ đầm ruộng từ...	cũng tương tự như các loài Đ...	Quả chứa protid 4,97%	Quả - Fructus	2		2

Hình 4.10: Kết quả sau khi chuyển dữ liệu từ excel lên cơ sở dữ liệu của MySQL (phần cây thuốc)

### 4.3. Mô hình vector

#### 4.3.1. Mô hình tổng quát



Hình 4.11: Mô hình vector hóa dữ liệu dược tính.

#### 4.3.2. Tổng quát hóa các thành phần dược liệu

Mỗi loại cây thuốc sẽ có những thành phần giống và khác nhau về số lượng loại và mỗi loại cũng có số lượng gram khác nhau. Khóa luận sẽ xem xét toàn bộ dữ liệu đã được mô hình ở phần trước để có được số lượng thành phần tối đa và số lượng gram ít nhất, nhiều nhất cho từng loại dược tính. Do điều kiện năng lực xử lý thiết bị của chúng em, nên chúng em quyết định sẽ dùng 10 thành phần dược tính chung nhất mà nhiều cây thuốc nhất có.



Bước 1: Thống kê số lần xuất hiện của mỗi thành phần hóa học trong sách.

STT	ID	Tên thành phần hóa học	Đơn vị tính	Number of occurrences	Note	% frequency
1	1	Tinh dầu	%	135		2,9278
4	4	Protid	%	131		2,8410
2	2	Vitamin C	gama, mg%	105		2,2772
3	3	Lipid	%	95		2,0603
5	5	Tanin	%	87		1,8868
6	6	Nước	%	73		1,5832
7	7	Glucid	%	73		1,5832
8	8	Alcaloid	%	66		1,4314
9	9	Tro	%	63		1,3663
10	10	Dầu	%	59		1,2795
11	11	Saponin		45		0,9759
12	12	Caroten	mg%, %	44		0,9542
14	14	Nhựa	%	39		0,8458
15	15	Vitamin A	gama	38		0,8241
16	16	Fe	mg%	36		0,7807
17	17	Tinh bột	%	34		0,7374
18	18	Cellulose	%	33		0,7157
19	19	Vitamin B2	gama, mg%	31		0,6723
20	20	Phosphor	%, mg%	30	khoảng	0,6506
21	21	Calcium	mg%	30	khoảng	0,6506
22	22	Vitamin B1	gama,mg%	27		0,5856
23	23	Glucosid		27		0,5856
24	24	glucose	%	27		0,5856
25	25	Kalium	mg%	25		0,5422
26	26	Ca	%, mg%	24		0,5205
27	27	Dầu béo	%	24		0,5205

Hình 4.12: Kết quả thống kê số lần xuất hiện của mỗi thành phần hóa học

Đọc tài liệu và trích xuất số lần xuất hiện của mỗi thành phần hóa học, sau đó suy ra tỉ lệ xuất hiện của thành phần đó, kết quả như (Hình 4.12).

Bước 2: Chọn ra top 10 thành phần hóa học dựa trên kết quả thống kê.

Dựa trên bảng thống kê, tiến hành chọn ra 10 thành phần hóa học xuất hiện nhiều nhất, nhưng phải đảm bảo điều kiện là số lần xuất hiện đó phải chứa giá trị hóa học. Vì có một số thành phần xuất hiện nhiều lần nhưng chỉ liệt kê tên mà không đi kèm với giá trị.

Bước 3: Trích xuất dữ liệu 10 thành phần hóa học cho mỗi cây thuốc trong excel.

STT	ID	Tên cây thuốc	Tinh dầu	Vitamin C	Lipid	Protid	Tanin	Glucid	Alcaloid	Tinh bột	Caroten	Cellulose	Not Valid
1	0001	A kê	0	0	20	5	0	4.6	0	0	0	0	
2	0003	Actisô	0	10	0.1-0.3	3-3.15	0	11-15.5	0	0	0	0	
4	0007	Ấu	0	0	10.3	0	0	0	0	49	0	0	
5	0010	Ấu nước	0	?	0.67	4.97	0	46.6	0	0	0	0	
10	0020	Ba đậu	0	0	0	18	0	0	0	0	0	0	
11	0021	Ba đậu tây	0	0	0	25.63	0	0	0	0	0	0	
12	0022	Ba gác	0	0	0	0	0	0	0.9-2.12	0	0	0	
13	0023	Ba gác Ấn Độ	0	0	0	0	0	0	0.8-1	6-7	0	0	
14	0024	Ba gác Châu đốc	0	0	0	0	0	0	2.39	0	0	0	
15	0025	Ba gác châu Phi	0	0	0	0	0	0	1-1.5	0	0	0	
16	0026	Ba gác Cuba	0	0	0	0	0	0	2.78	0	0	0	
17	0028	Ba gác lá nhỏ	0	0	0	0	0	0	2.5-2.8	0	0	0	
18	0029	Ba gác lá to	0	0	0	0	0	0	2.59	0	0	0	
21	0034	Bạc hà	0.5-1.5	0	0	0	0	0	0	0	0	0	
22	0035	Bạc hà cay	1-3	0	0	0	0	0	0	0	0	0	
25	0042	Bách bộ	0	0	0.83	9	0	2.3	0	0	0	0	
28	0048	Bạch đàn chanh	0.5-2	0	0	0	0	0	0	0	0	0	
30	0051	Bạch đàn lá liễu	0.65	0	0	0	0	0	0	0	0	0	
34	0056	Bạch đậu khấu	2.4	0	0	0	0	0	0	0	0	0	
36	0060	Bách hợp	0	?	0.1	4	0	30	0	0	0	0	
41	0069	Bạch truật	1.4	0	0	0	0	0	0	0	0	0	
46	0085	Bàng	0	0	1.1	2.6	0	0	0	0	0	7.6	

Hình 4.13: Kết quả trích xuất dữ liệu 10 thành phần hóa học cho mỗi cây thuốc trong excel

Sau khi đã chọn được 10 thành phần hóa học để thiết lập vector 10 chiều, thì tiến hành chuẩn bị dữ liệu cho vector như (Hình 4.13).

Bước 4: Tiền xử lý dữ liệu vector lần nữa.

- Xử lý phân đơn vị của các thành phần hóa học. Có 2 thành phần VitaminC và Caroten mang đơn vị là mg% trong đa số các cây thuốc, và một số ít mang đơn vị là %, 8 thành phần còn lại có đơn vị là %. Cách xử lý là đổi toàn bộ giá trị % của VitaminC và Caroten sang mg% hết bằng cách nhân giá trị hiện tại với 1000, để đảm bảo độ chính xác của dữ liệu.

- Các thành phần hóa học có giá trị trong khoảng  $[a, b]$  thì lấy giá trị trung bình.

Lipid	Protid
20	5
0.1-0.3	3-3.15

Hình 4.14: Các thành phần hóa học có giá trị khoảng  $[a, b]$

### 4.3.3. Chuẩn hóa giá trị thành phần được liệu

Có những dược liệu có giá trị gram rất nhỏ, nhưng rất quan trọng trong các loại thuốc khác nhau. Mà các thành phần này có giá trị gram quá nhỏ so với các thành phần khác nên khi xây dựng vector truy vấn sẽ dẫn đến có khả năng các thành phần này sẽ không đóng góp vào kết quả truy vấn (các thuật toán xem như là 0).

Xác định giá trị nhỏ nhất trong tất cả các giá trị của các thành phần hóa học của các cây thuốc thì kết quả như hình 4.15. Để tránh khả năng các thành phần có giá trị quá nhỏ bị các thuật toán xem như là 0, chúng tôi sử dụng phương pháp nhân các giá trị vector cho  $10^k$ . Phương pháp này nhằm đồng nhất hóa dữ liệu một cách nhanh và đơn giản cho bài toán khóa luận thay vì gán trọng số cho từng thành phần riêng biệt.

STT	ID	Tên cây thuốc	Tinh dầu	Vitamin C	Lipid	Protid	Tanin	Glucid	Alcaloid	Tinh bột	Caroten	Cellulose
1002	1002	Gai ma vương	0	0	3.5	0	0	0	0.001	0	0	0

Hình 4.15: Giá trị nhỏ nhất trong tất cả các giá trị của các thành phần hóa học

Giá trị nhỏ nhất tìm được  $\min = 0.001$  là giá trị của thành phần Alcaloid của cây Gai ma vương. Suy ra, nhân toàn bộ các giá trị vector cho hệ số  $10^3$ .

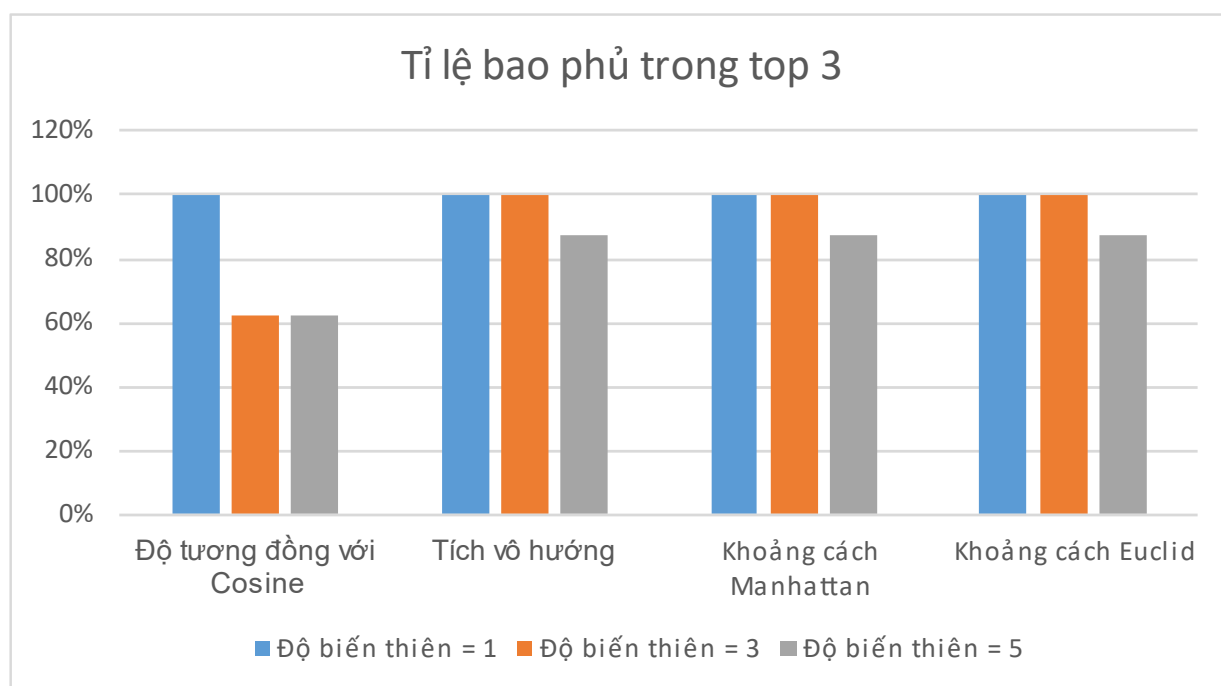
#### 4.3.4. Thuật toán tìm kiếm

##### 4.3.4.1. Tìm kiếm với các thành phần trước khi chuẩn hóa

Khóa luận xem xét các thành phần dược liệu có mức độ quan trọng như nhau để xem xét truy vấn, thì kết quả với độ phủ cho giá trị  $k = 3$  với phương pháp k-top như bảng 4.1 và hình 4.16. Với độ biến thiên lần lượt là 1, 3, 5.

Bảng 4.1: Kết quả truy vấn

Độ biến thiên	Độ tương đồng với Cosine	Tích vô hướng	Khoảng cách Manhattan	Khoảng cách Euclid
1	100%	100%	100%	100%
3	63%	100%	100%	100%
5	63%	88%	88%	88%



Hình 4.16: Kết quả với 04 độ đo

Nhận xét:

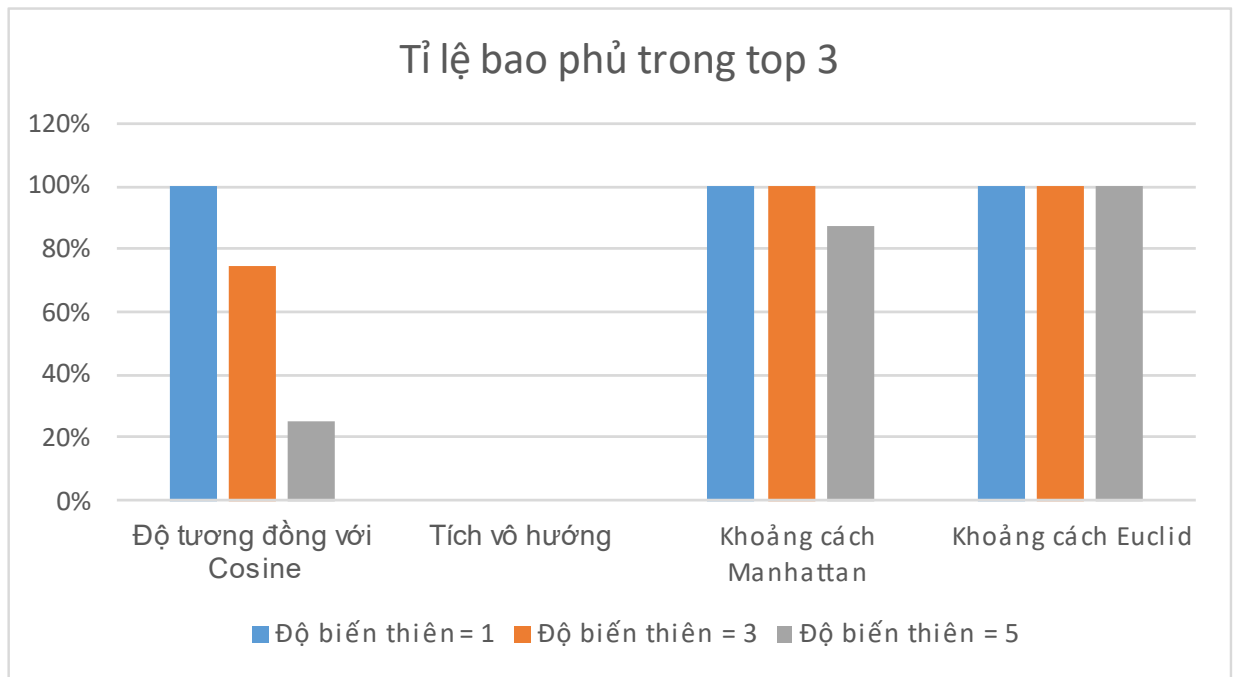
- Độ chính xác đạt mức cao khi độ biến thiên bằng 1
- Khi độ biến thiên tăng (3 và 5) đã có sự khác biệt rõ rệt, tỷ lệ bao phủ của thuật toán cosine giảm đáng kể còn các thuật toán còn lại gần như là như nhau.

#### 4.3.4.2. Tìm kiếm với các thành phần sau khi chuẩn hóa

Khóa luận xem xét các thành phần được liệu có mức độ quan trọng khác nhau để xem xét truy vấn, thì kết quả với độ phủ cho giá trị  $k = 3$  với phương pháp k-top như bảng 4.2 và hình 4.17. Với độ biến thiên lần lượt là 1, 3, 5.

Bảng 4.2: Kết quả truy vấn

Độ biến thiên	Độ tương đồng với Cosine	Tích vô hướng	Khoảng cách Manhattan	Khoảng cách Euclid
1	100%	0%	100%	100%
3	75%	0%	100%	100%
5	25%	0%	88%	100%



*Hình 4.17: Kết quả với 04 độ đo*

Nhận xét:

- Độ chính xác đạt mức cao khi độ biến thiên bằng 1
- Khi độ biến thiên tăng (3 và 5) đã có sự khác biệt rõ rệt, tỷ lệ bao phủ của thuật toán độ tương đồng với Cosine giảm đáng kể, thuật toán tích vô hướng không còn bao phủ kết quả mục tiêu. Còn thuật toán khoảng cách Manhattan và khoảng cách Euclid gần như tương đương nhau tuy nhiên khoảng cách Euclid nhỉnh hơn.

#### **4.3.4.3. Đánh giá và lựa chọn**

##### **Tích vô hướng**

- Hoạt động tốt khi các giá trị đầu vào thấp và chưa nhân hệ số.
- Khuyến cáo nên loại bỏ thuật toán tích vô hướng trong tình huống các giá trị vector quá cao.

##### **Độ tương đồng với Cosine**

- Phù hợp khi dữ liệu ít nhiễu (ít biến thiên so với kết quả mục tiêu), ngược lại khi dữ liệu nhiễu nhiều sẽ dẫn đến giảm độ chính xác đáng kể cần xem xét đánh giá lại.

- Khi dữ liệu được nhân hệ số độ bao phủ đã tăng một chút tuy nhiên vẫn còn thấp so với các thuật toán còn lại.

### **Khoảng cách Manhattan**

- Hiệu suất cao và đồng đều trong các trường hợp. Phù hợp cho dữ liệu bị điều chỉnh như nhân hệ số hoặc biến thiên lớn.

### **Khoảng cách Euclid**

- Hiệu suất cao nhất trong vài trường hợp tìm kiếm so với khoảng cách Manhattan và đồng đều trong các trường hợp. Phù hợp cho dữ liệu bị điều chỉnh như nhân hệ số hoặc biến thiên lớn.

**Kết luận:** Dựa trên phân tích của chúng tôi, thuật toán khoảng cách Euclid sẽ là lựa chọn tối ưu cho các trường hợp tìm kiếm của chúng tôi. Ngoài ra, khi điều chỉnh dữ liệu bằng cách nhân hệ số sẽ cho ra kết quả tốt hơn.

#### **4.3.5. Đẩy dữ liệu vector lên cơ sở dữ liệu Qdrant**

Sau khi đã hoàn thành việc trích xuất dữ liệu các thành phần hóa học của các cây thuốc có thể sử dụng vào excel, chúng tôi tiến hành viết mã để chuyển dữ liệu từ excel lên cơ sở dữ liệu của Qdrant để lưu trữ:

POINTS
INFO
SEARCH QUALITY
SNAPSHOTS
VISUALIZE
GRAPH

Find similar by ID or filter by payload key-value pair. Example: name: John Doe, age: 25, id: c0847827-d005-4e45-b328-887f72373d2d , id: 1234567890

Point 1

Payload:

Tên cây thuốc
A kê

Vectors:

Vectors:

Default vector

Length:
10

OPEN GRAPH
FIND SIMILAR

Point 3

Payload:

Tên cây thuốc
Actisô

Vectors:

Vectors:

Default vector

Length:
10

OPEN GRAPH
FIND SIMILAR

Point 7

Payload:

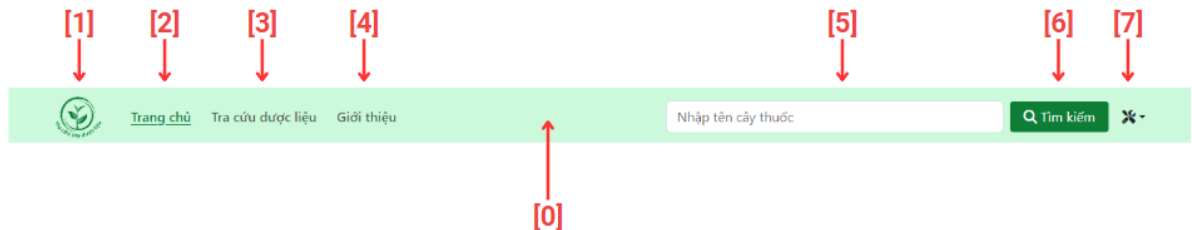
Tên cây thuốc
Ấu

Hình 4.18: Kết quả sau khi chuyển dữ liệu vector từ excel lên cơ sở dữ liệu Qdrant

## CHƯƠNG 5. XÂY DỰNG ỨNG DỤNG

### 5.1. Giao diện

#### 5.1.1. Trang chủ



Hình 5.1: Giao diện thanh header

Danh sách biến cố:

Bảng 5.1: Bảng danh sách biến cố của giao diện trang chủ (thanh header)

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khi truy cập trang chủ.	Hiển thị giao diện trang chủ.	
1	Khi nhấn logo.	Hiển thị giao diện trang chủ.	
2	Khi nhấn “Trang chủ”	Hiển thị giao diện trang chủ.	
3	Khi nhấn “Tra cứu dược liệu”	Hiển thị giao diện tra cứu dược liệu.	
4	Khi nhấn “Giới thiệu”	Hiển thị giao diện trang giới thiệu.	
6	Khi nhấn “Giới thiệu”	Hiển thị giao diện trang tra cứu dược liệu và cây thuốc theo từ khóa đã nhập.	
7	Khi nhấn biểu tượng cờ lê.	Hiển thị giao diện trang đăng nhập.	

Mô tả giao diện

Bảng 5.2: Bảng mô tả giao diện trang chủ (thanh header)

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	home_page	div	Khung chứa các thành phần của	N/A	N/A	



			giao diện trang chủ.			
1	logo	img	Biểu tượng của website.	N/A	N/A	
2	home_page_link	a	Nút điều hướng sang giao diện trang chủ.	N/A	N/A	
3	search_page_link	a	Nút điều hướng sang giao diện trang tra cứu.	N/A	N/A	
4	introduce_page_link	a	Nút điều hướng sang giao diện trang giới thiệu.	N/A	N/A	
5	search_input	input	Ô nhập từ khóa tìm kiếm cây thuốc.	N/A	N/A	
6	search_btn	button	Nút tìm kiếm.	N/A	N/A	
7	login_page_btn	button	Nút điều hướng sang giao diện đăng nhập.	N/A	N/A	



Hình 5.2: Giao diện slider của trang chủ

Danh sách biến cố:

*Bảng 5.3: Bảng danh sách biến cố của giao diện trang chủ (slider)*

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khi nhấn vào hình ảnh.	Hiển thị giao diện chi tiết về cây thuốc.	
1	Khi nhấn biểu tượng qua trái.	Hiển thị ảnh cây thuốc ở bên trái.	
3	Khi nhấn biểu tượng qua phải.	Hiển thị ảnh cây thuốc ở bên phải.	

Mô tả giao diện

*Bảng 5.4: Bảng mô tả giao diện trang chủ (slider)*

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	banner	div	Hình cây cây thuốc phổ biến.	N/A	N/A	
1	previous_btn	button	Quay về hình ảnh trước.	N/A	N/A	
2	info_banner	div	Hiển thị thông tin cây thuốc	N/A	N/A	
3	next_btn	button	Quay về hình ảnh trước.	N/A	N/A	



Hình 5.3: Giao diện hiển thị một số dược liệu được tìm kiếm nhiều nhất

Danh sách biến cố:

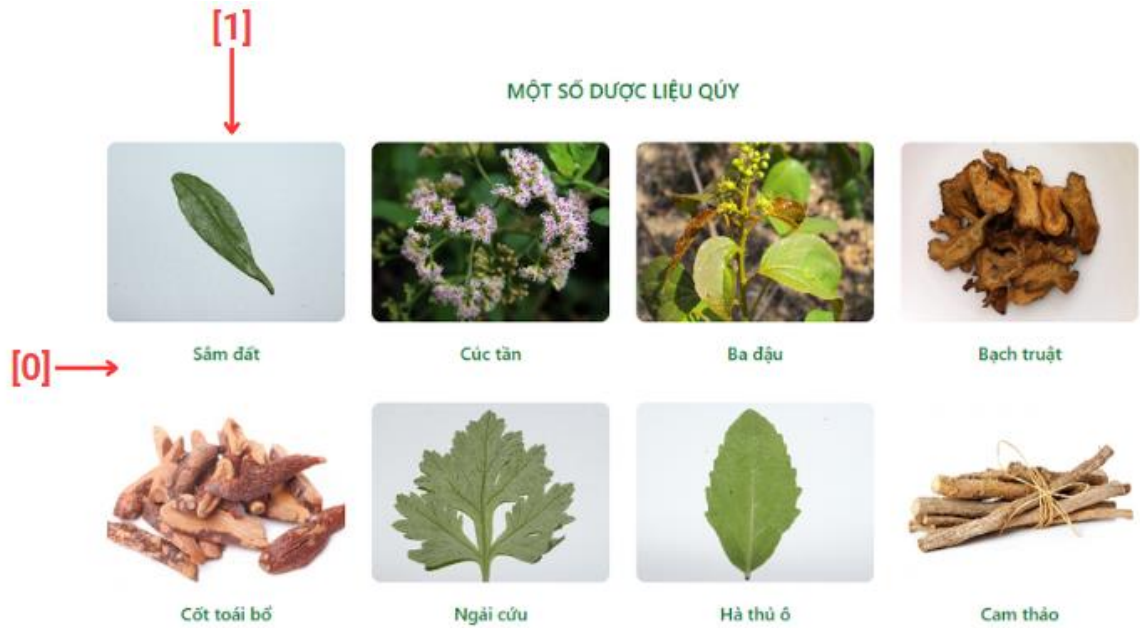
Bảng 5.5: Bảng danh sách biến cố của giao diện trang chủ (phần dược liệu được tìm nhiều nhất)

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
1	Khi nhấn vào hình ảnh cây thuốc.	Hiển thị giao diện chi tiết về cây thuốc.	

Mô tả giao diện

Bảng 5.6: Bảng mô tả giao diện trang chủ (phần dược liệu được tìm nhiều nhất)

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	section0	div	Hiển thị cây thuốc được tìm kiếm nhiều nhất và theo công dụng.	N/A	N/A	
1	image	img	Hình ảnh cây thuốc được tìm kiếm nhiều.	N/A	N/A	



Hình 5.4: Giao diện hiển thị một số dược liệu quý của trang chủ

Danh sách biến cố:

Bảng 5.7: Bảng danh sách biến cố của giao diện trang chủ (phần dược liệu quý)

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
1	Khi nhấn vào hình ảnh cây thuốc.	Hiển thị giao diện chi tiết về cây thuốc.	

Mô tả giao diện

Bảng 5.8: Bảng mô tả giao diện trang chủ (phần dược liệu quý)

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	section2	div	Hiển thị 8 cây thuốc quý.	N/A	N/A	
1	image	img	Hình ảnh cây thuốc quý.	N/A	N/A	

<p><b>[2]</b></p> 	<p><b>BẠN CÓ BIẾT</b></p> <p><b>Quả Ổi:</b> Vitamin C trong ổi cao hơn gấp 4 lần lượng vitamin C có trong cam. Chúng giúp cơ thể khỏe mạnh, tăng cường khả năng hệ miễn dịch, giúp tránh được các mầm mống gây bệnh nhiễm trùng như cảm lạnh, cảm cúm, ho và ngăn ngừa các trường hợp bị nhiễm siêu vi. Có ít chất béo trong ổi nên rất thích hợp cho người giảm cân. Đặc biệt có chất lycopene trong trái ổi ruột đỏ (lời đào), không chỉ có tác dụng chống những bệnh liên quan tới béo phì mà còn những bệnh khác như giảm bớt nguy cơ bệnh tim mạch và bệnh tiểu đường cấp 1.</p> <p><b>Công dụng:</b> Quả ổi có vị ngọt, chua, tính bình; có tác dụng lợi tiểu, tiêu thũng, giải độc bại rung. <a href="#">Xem chi tiết</a></p>	
<p><b>[0]</b></p> 	<p><b>Tía Tó:</b> Dịch chiết từ hạt nảy mầm của cây tía tô đã được chứng minh là có tác dụng chống đái tháo đường trên mô hình chuột thí nghiệm bị đái tháo đường type 2 thông qua các cơ chế: làm giảm lượng đường huyết bất kỳ, giảm nồng độ triglycerid và nồng độ cholesterol toàn phần trong huyết thanh; làm tăng sự không dung nạp đường và tăng nhạy cảm với insulin; kích hoạt protein kinase hoạt hóa bởi AMP (AMPK) và thông qua đó, ức chế sản tạo đường ở gan. Acid chlorogenic và acid rosmarinic ức chế enzyme Aldose reductase, làm giảm nguy cơ xảy ra các biến chứng đái tháo đường.</p> <p><b>Công dụng:</b> Tía tô có tác dụng phát tán phong hàn, lý khí khoan hồng; giải uất, hoá đờm, an thai, giải độc của cua cá. Cảnh không có tác dụng phát biểu, chỉ có tác dụng lý khí. <a href="#">Xem chi tiết</a></p>	<p><b>[1]</b></p>
	<p><b>Bí Đỏ:</b> Bí đỏ chứa nhiều chất axit glutamine, chất cần thiết cho hoạt động não bộ. Đây là một chất có vai trò quan trọng trong việc trợ giúp phản ứng chuyển hóa ở các tế bào thần kinh và não. Ngoài ra nó còn có tác dụng bồi bổ dây thần kinh. Phụ nữ mang thai ăn bí đỏ sẽ rất tốt cho sự phát triển các tế bào não, tăng cường sức sống, sức khỏe của thai nhi. Chúng còn giúp tăng cường phòng ngừa, điều trị chứng tăng huyết áp, phù và các biến chứng khác khi mang thai.</p> <p><b>Công dụng:</b> Quả có vị ngọt, tính ấm, có tác dụng bổ dưỡng, làm dịu, giải nhiệt, giải khát, trị ho, nhuận tràng, lợi tiểu. Hạt có vị ngọt, có đặc có tác dụng tẩy giun sán do tác dụng của các lipid, không kích thích và không độc nó cũng có tính làm dịu và giải nhiệt. <a href="#">Xem chi tiết</a></p>	

Hình 5.5: Giao diện mục “BẠN CÓ BIẾT” của trang chủ

Danh sách biến cố:

*Bảng 5.9: Bảng danh sách biến cố của giao diện trang chủ (phần “BAN CÓ BIẾT”)*

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
1	Khi nhấn vào nút “xem chi tiết”.	Hiện thị giao diện chi tiết về cây thuốc.	

## Mô tả giao diện

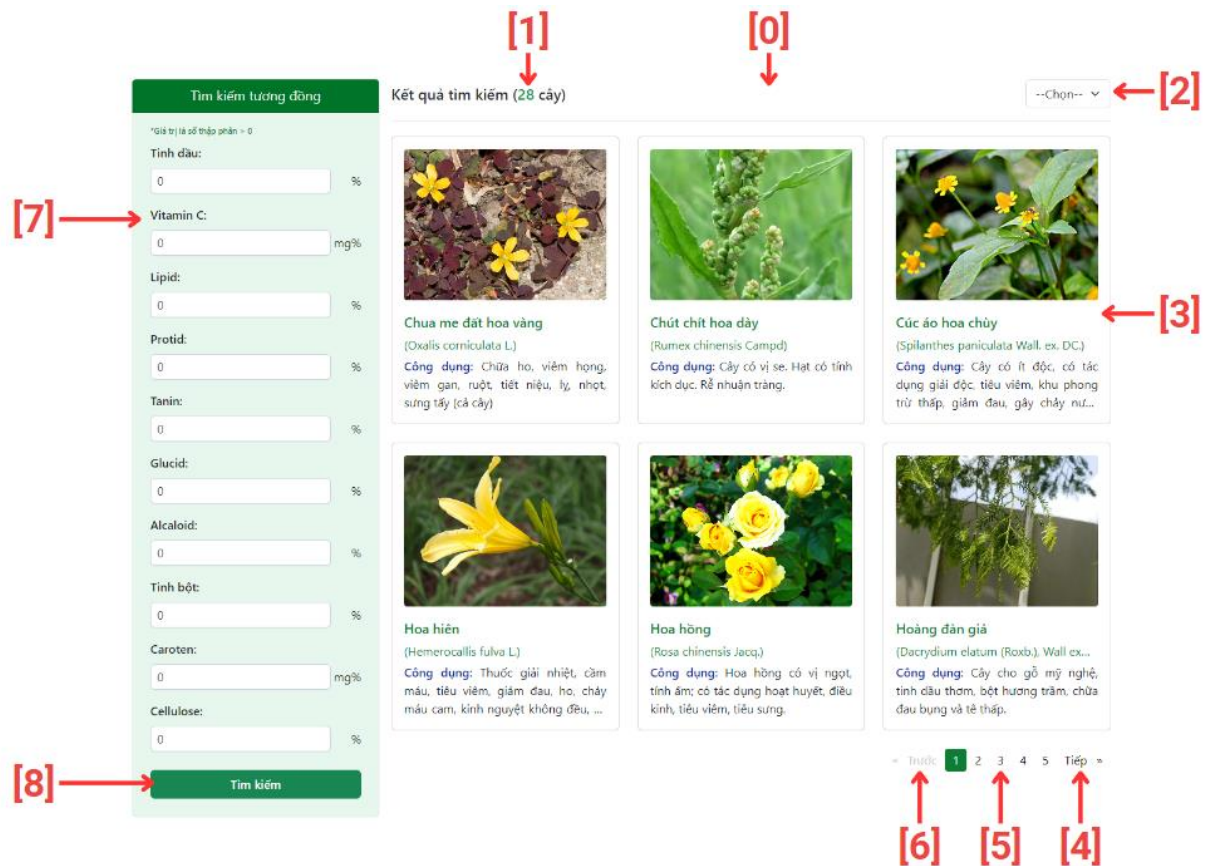
*Bảng 5.10: Bảng mô tả giao diện trang chủ (phần “BAN CÓ BIẾT”)*

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	section3	div	Hiển thị thông tin và hình ảnh cây thuốc phổ thông.	N/A	N/A	



1	detail_btn	button	Xem thông tin chi tiết cây thuốc.	N/A	N/A	
2	info	div	Hiển thị thông tin cây thuốc.	N/A	N/A	

### 5.1.2. Trang tra cứu



Hình 5.6: Giao diện trang tra cứu

Danh sách biến cố:

Bảng 5.11: Bảng danh sách biến cố của giao diện trang tra cứu

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khi truy cập trang tra cứu được liệu.	Hiển thị giao diện trang tra cứu được liệu.	
2	Khi nhấn “Chọn”	Hiển thị các lựa chọn sắp xếp cây thuốc.	
3	Khi nhấn vào.	Hiển thị giao diện chi tiết cây thuốc.	

4	Khi nhấn nút “Tiếp”.	Hiển thị trang xem cây thuốc tiếp theo.	
5	Khi nhấn chọn số.	Hiển thị trang xem cây thuốc theo số đã chọn.	
6	Khi nhấn nút “Trước”	Hiển thị trang xem cây thuốc trước đó.	
8	Khi nhấn nút “Tìm kiếm”	Tìm kiếm cây thuốc tương đồng và hiển thị.	

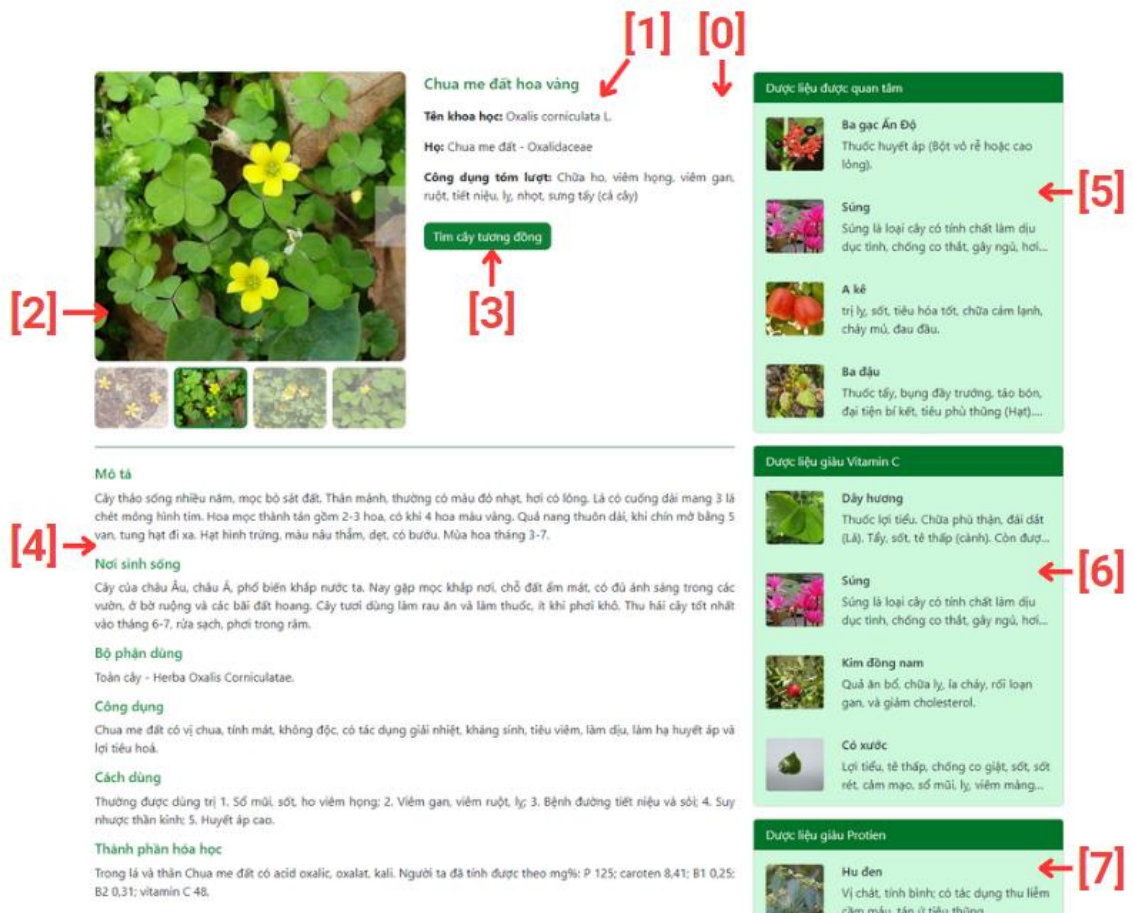
Mô tả giao diện

*Bảng 5.12: Bảng mô tả giao diện trang tra cứu*

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	search_page	div	Khung chứa các thành phần của giao diện trang tra cứu được liệt.	N/A	N/A	
1	search_result	div	Hiển thị số cây thuốc đã tìm được.	N/A	N/A	
2	sort	select	Danh sách các tiêu chí sắp xếp cây thuốc.	1	sort	
3	plant_card	a	Hiển thị thông tin ngắn cây thuốc.	N/A	N/A	
4	next_page_btn	button	Nút điều hướng qua trang tiếp theo.	N/A	N/A	
5	page_number	button	Nút điều hướng qua trang chỉ định.	N/A	N/A	
6	previous_page_btn	button	Nút điều hướng qua trang trước đó.	N/A	N/A	

7	vector_search	div	Hiển thị biểu mẫu tìm kiếm theo 10 TPHH.	N/A	N/A	
8	vector_search_btn	button	Nút tìm kiếm theo 10 TPHH.	N/A	N/A	

### 5.1.3. Trang xem chi tiết



Hình 5.7: Giao diện trang chi tiết cây thuốc

Danh sách biến cố:

Bảng 5.13: Bảng danh sách biến cố của giao diện trang chi tiết cây thuốc

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khi nhấn vào hình ảnh cây thuốc.	Hiển thị giao diện trang xem chi tiết cây thuốc.	
3	Khi nhấn nút “Tìm cây tương đồng”.	Hiển thị giao diện tra cứu được liệu và các cây tương đồng.	

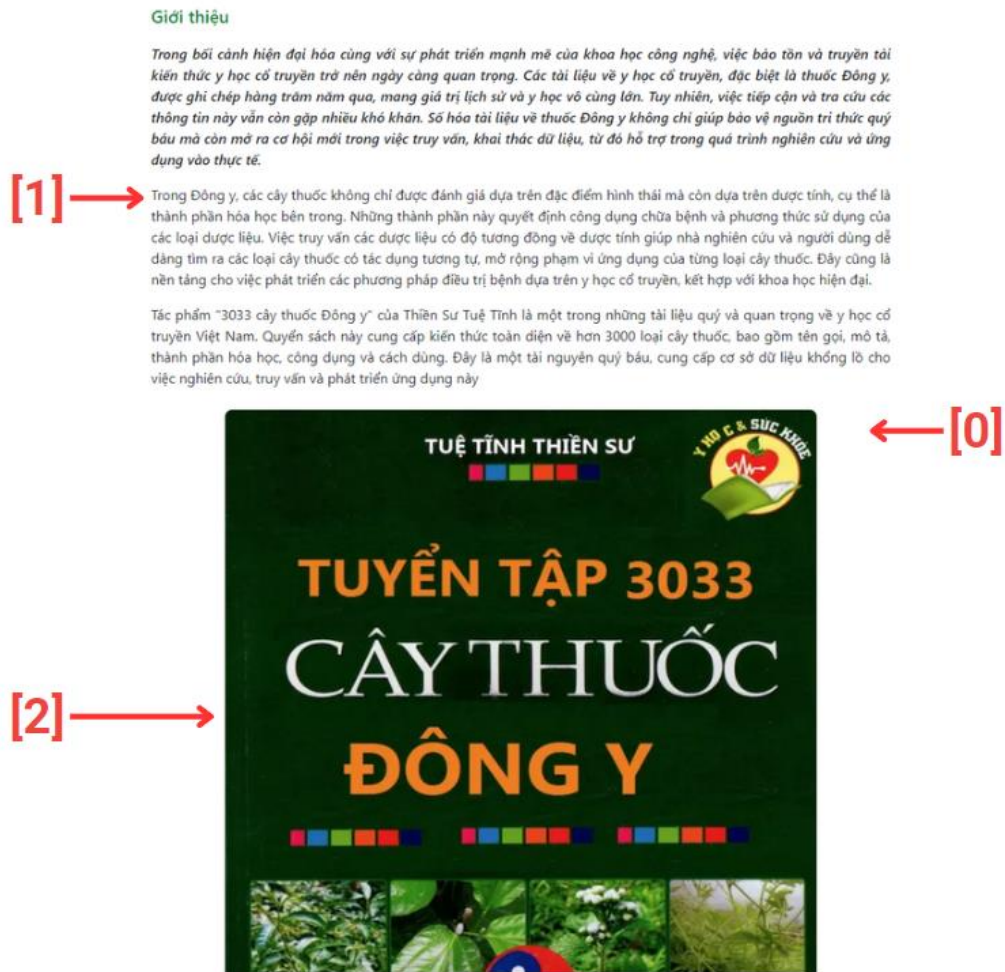


## Mô tả giao diện

Bảng 5.14: Bảng mô tả giao diện trang chi tiết cây thuốc

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	detail_page	div	Khung chứa các thành phần của giao diện trang xem chi tiết.	N/A	N/A	
1	info1	div	Thông tin chung của cây thuốc.	N/A	N/A	
2	plant_img	img	Hình ảnh cây thuốc.	N/A	N/A	
3	familiar_search_btn	button	Nút tìm cây thuốc tương đồng.	N/A	N/A	
4	infor2	div	Thông tin chi tiết cây thuốc.	N/A	N/A	
5	sec1	div	Khung chứa các cây thuốc được quan tâm.	N/A	N/A	
6	sec2	div	Khung chứa các cây thuốc giàu Vitamin C.	N/A	N/A	
7	sec3	div	Khung chứa các cây thuốc giàu Protein.	N/A	N/A	

### 5.1.4. Trang giới thiệu



Hình 5.8: Giao diện trang giới thiệu

Danh sách biến cố:

Bảng 5.15: Bảng danh sách biến cố của giao diện trang giới thiệu

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khi truy cập trang giới thiệu.	Hiển thị giao diện trang giới thiệu.	

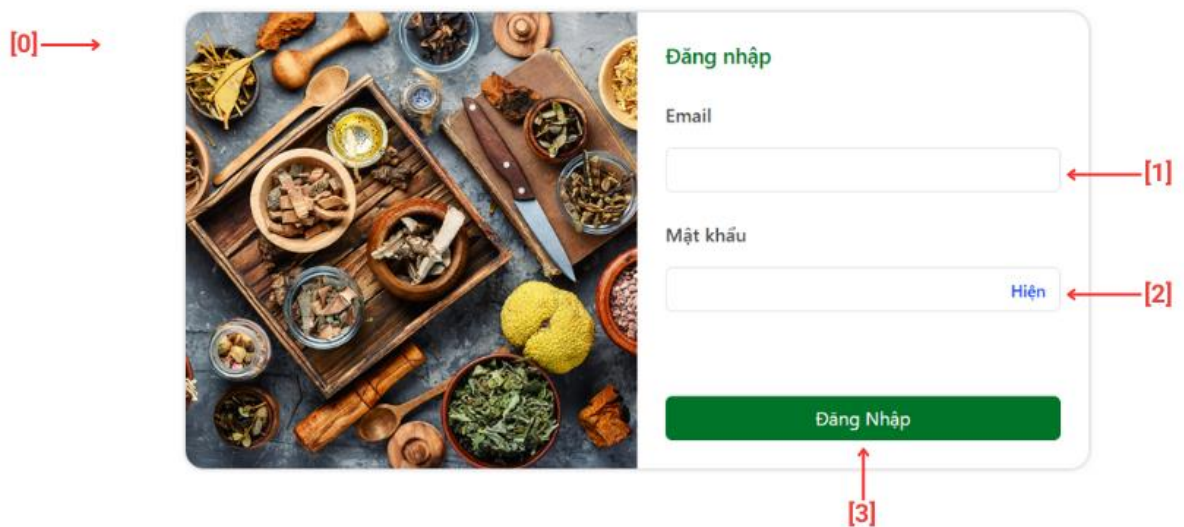
Mô tả giao diện

Bảng 5.16: Bảng mô tả giao diện trang giới thiệu

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	introduce_page	div	Khung chứa các thành	N/A	N/A	

			phần của giao diện trang giới thiệu.			
1	book_info	div	Thông tin sách và lời dẫn.	N/A	N/A	
2	book_img	img	Hình ảnh bìa sách.	N/A	N/A	

### 5.1.5. Trang đăng nhập



Hình 5.9: Giao diện trang đăng nhập

Danh sách biến cố:

Bảng 5.17: Bảng danh sách biến cố của giao diện trang đăng nhập

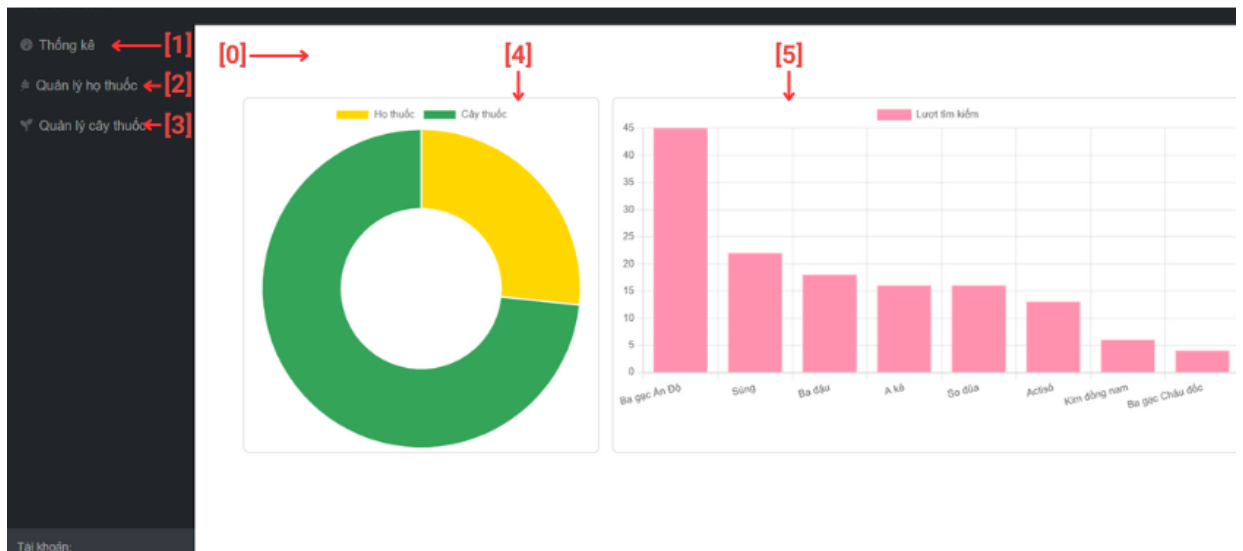
STT	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khi truy cập trang đăng nhập.	Hiển thị giao diện đăng nhập.	
3	Khi nhấn nút “Đăng nhập”.	Kiểm tra email và mật khẩu: - Nếu hợp lệ, điều hướng đến trang quản trị. - Nếu không hợp lệ, thông báo lỗi.	

Mô tả giao diện

Bảng 5.18: Bảng mô tả giao diện trang đăng nhập

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	sign_in	div	Khung chứa các thành phần của giao diện đăng nhập.	N/A	N/A	
1	input_email	input	Ô nhập email.	N/A	N/A	
2	input_password	input	Ô nhập mật khẩu.	N/A	N/A	
3	btn_sign_in	button	Nút đăng nhập.	N/A	N/A	

### 5.1.6. Trang thống kê



Hình 5.10: Giao diện trang thống kê

Danh sách biến cố:

Bảng 5.19: Bảng danh sách biến cố của giao diện trang thống kê

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khi truy cập trang thống kê.	Hiển thị giao diện thống kê.	
1	Khi nhấn nút “Thống kê” trên sidebar.	Hiển thị giao diện thống kê.	
2	Khi nhấn nút “Quản lý họ cây thuốc” trên sidebar.	Hiển thị giao diện quản lý họ cây thuốc.	

3	Khi nhấn nút “Quản lý cây thuốc” trên sidebar.	Hiển thị giao diện quản lý cây thuốc.	
---	--	---------------------------------------	--

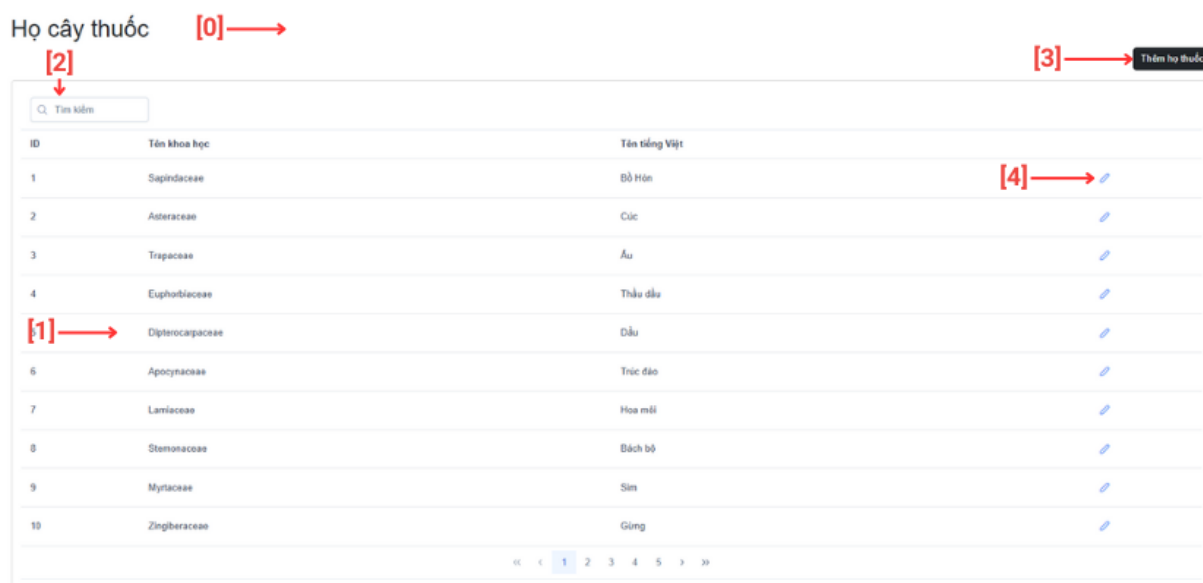
Mô tả giao diện

*Bảng 5.20: Bảng mô tả giao diện trang thống kê*

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	statistics	div	Khung chứa các thành phần của giao diện thống kê.	N/A	N/A	
1	nav_item_statistics	a	Link điều hướng tới giao diện thống kê.	N/A	N/A	
2	nav_item_family	a	Link điều hướng tới giao diện quản lý họ cây thuốc.	N/A	N/A	
3	nav_item_plant	a	Link điều hướng tới giao diện quản lý cây thuốc.	N/A	N/A	
4	stat_num_object	div	Khung chứa biểu đồ thống kê số lượng cây thuốc và họ cây thuốc.	N/A	N/A	
5	stat_most_searched_plants	div	Khung chứa biểu đồ thống kê các cây thuốc được tìm kiếm nhiều nhất.	N/A	N/A	

### 5.1.7. Trang quản lý họ cây thuốc

#### 5.1.7.1. Giao diện xem các họ cây thuốc



Hình 5.11: Giao diện xem các họ cây thuốc

Danh sách biến cố:

Bảng 5.21: Bảng danh sách biến cố của giao diện quản lý họ cây thuốc (phần xem họ cây thuốc)

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khi truy cập trang quản lý họ cây thuốc.	Hiển thị giao diện quản lý họ cây thuốc.	
3	Khi nhấn nút “Thêm họ cây thuốc”.	Mở modal thêm họ cây thuốc.	
4	Khi nhấn biểu tượng hình cây viết (Sửa họ cây thuốc).	Hiển thị giao diện sửa họ cây thuốc.	

Mô tả giao diện

Bảng 5.22: Bảng mô tả giao diện quản lý họ cây thuốc (phần xem họ cây thuốc)

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	family	div	Khung chứa các thành phần của giao diện quản lý họ cây thuốc.	N/A	N/A	

1	families	table	Bảng chứa các họ cây thuốc.	N/A	N/A	
2	search	input	Ô tìm kiếm họ cây thuốc.	N/A	N/A	
3	btn_insert	button	Nút mở modal thêm họ cây thuốc.	N/A	N/A	
4	btn_edit	button	Nút mở giao diện sửa họ cây thuốc.	N/A	N/A	

### 5.1.7.2. Giao diện thêm họ cây thuốc



Hình 5.12: Giao diện thêm họ cây thuốc mới

Danh sách biến cố:

Bảng 5.23: Bảng danh sách biến cố của giao diện quản lý họ cây thuốc (phần thêm họ cây thuốc)

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
3	Khi nhấn nút “Hủy”.	Đóng modal thêm họ cây thuốc.	
4	Khi nhấn nút “Lưu”.	Thêm họ cây thuốc mới.	

Mô tả giao diện

Bảng 5.24: Bảng mô tả giao diện quản lý họ cây thuốc (phần thêm họ cây thuốc)

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	input_id	input	Ô hiển thị id họ cây thuốc.	N/A	N/A	
1	input_vietnamese_name	input	Ô nhập tên việt nam của họ cây thuốc.	N/A	N/A	
2	input_science_name	input	Ô nhập tên khoa học của họ cây thuốc.	N/A	N/A	
3	btn_cancel	button	Nút đóng modal thêm họ cây thuốc.	N/A	N/A	
4	btn_save	button	Nút thêm họ cây thuốc.	N/A	N/A	

### 5.1.7.3. Giao diện sửa họ cây thuốc

Tên khoa học	Tên tiếng Việt
<input type="text" value="Sapindaceae"/> <span style="color: red;">←[0]</span>	<input type="text" value="Bồ Hòn"/> <span style="color: red;">←[1]</span>
	<div style="display: flex; justify-content: space-around;"> <span style="color: green;">✓</span> <span style="color: red;">✗</span> </div> <div style="display: flex; justify-content: space-around;"> <span style="color: red;">↑[2]</span> <span style="color: red;">↑[3]</span> </div>

Hình 5.13: Giao diện sửa họ cây thuốc

Danh sách biến cố:



*Bảng 5.25: Bảng danh sách biến cố của giao diện quản lý họ cây thuốc (phần sửa họ cây thuốc)*

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
2	Khi nhấn nút “Lưu”.	Sửa họ cây thuốc.	
3	Khi nhấn nút “Hủy”.	Đóng giao diện sửa họ cây thuốc.	

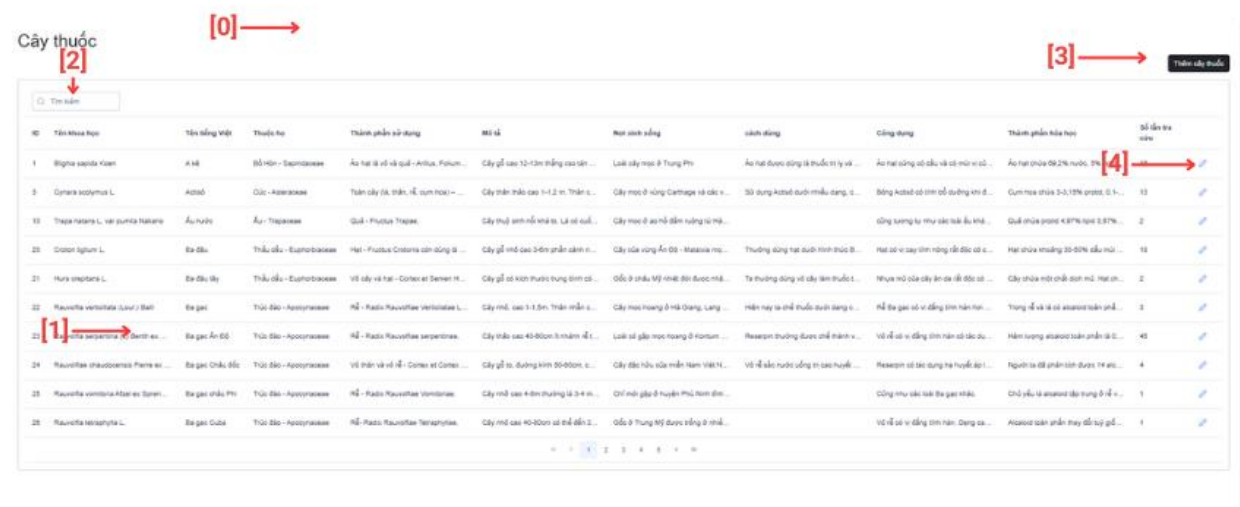
#### Mô tả giao diện

*Bảng 5.26: Bảng mô tả giao diện quản lý họ cây thuốc (phần sửa họ cây thuốc)*

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	input_vietnamese_name	input	Ô nhập tên việt nam của họ cây thuốc.	N/A	N/A	
1	input_science_name	input	Ô nhập tên khoa học của họ cây thuốc.	N/A	N/A	
2	btn_save	button	Nút sửa họ cây thuốc.	N/A	N/A	
3	btn_cancel	button	Nút đóng giao diện sửa họ cây thuốc.	N/A	N/A	

### 5.1.8. Trang quản lý cây thuốc

#### 5.1.8.1. Giao diện xem các cây thuốc



Hình 5.14: Giao diện xem các cây thuốc

Danh sách biến cố:

*Bảng 5.27: Bảng danh sách biến cố của giao diện quản lý cây thuốc (phần xem các cây thuốc)*

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
0	Khi truy cập trang quản lý cây thuốc.	Hiện thị giao diện quản lý cây thuốc.	
3	Khi nhấn nút “Thêm cây thuốc”.	Mở modal thêm cây thuốc.	
4	Khi nhấn biểu tượng hình cây viết (Sửa cây thuốc).	Mở modal sửa cây thuốc.	

## Mô tả giao diện

*Bảng 5.28: Bảng mô tả giao diện quản lý cây thuốc (phần xem các cây thuốc)*

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	plant	div	Khung chứa các thành phần của giao diện quản lý cây thuốc.	N/A	N/A	
1	plants	table	Bảng chứa các cây thuốc.	N/A	N/A	

2	search	input	Ô tìm kiếm cây thuốc.	N/A	N/A	
3	btn_insert	button	Nút mở modal thêm cây thuốc.	N/A	N/A	
4	btn_edit	button	Nút mở modal sửa cây thuốc.	N/A	N/A	

### 5.1.8.2. Giao diện thêm cây thuốc

Hình 5.15: Giao diện thêm cây thuốc mới

Danh sách biến cố:

Bảng 5.29: Bảng danh sách biến cố của giao diện quản lý cây thuốc (phần thêm cây thuốc)

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
13	Khi nhấn nút “Hủy”.	Đóng modal thêm cây thuốc.	
14	Khi nhấn nút “Lưu”.	Thêm cây thuốc mới.	

Mô tả giao diện

Bảng 5.30: Bảng mô tả giao diện quản lý cây thuốc (phần thêm cây thuốc)

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	i_vietnamese_name	input	Ô nhập tên tiếng việt.	N/A	N/A	

1	i_science_name	input	Ô nhập tên khoa học.	N/A	N/A	
2	i_part_used	input	Ô nhập bộ phận dùng.	N/A	N/A	
3	s_rare_level	select	Chọn mức độ hiếm.	N/A	N/A	
4	s_family	select	Chọn họ.	N/A	N/A	
5	t_desc	text-area	Ô nhập mô tả.	N/A	N/A	
6	t_brief_effect	text-area	Ô nhập công dụng tóm tắt.	N/A	N/A	
7	t_habitat	text-area	Ô nhập nơi sinh sống.	N/A	N/A	
8	t_chemistries	text-area	Ô nhập thành phần hóa học.	N/A	N/A	
9	t_usage	text-area	Ô nhập cách dùng.	N/A	N/A	
10	t_effect	text-area	Ô nhập công dụng.	N/A	N/A	
11	btn_choose_pic	button	Button chọn ảnh cây thuốc.	N/A	N/A	
12	vector	div	Phần nhập 10 giá trị hóa học để cấu thành vector 10 chiều.	N/A	N/A	
13	btn_cancel	button	Nút đóng modal thêm cây thuốc.	N/A	N/A	
14	btn_save	button	Nút thêm cây thuốc.	N/A	N/A	

### 5.1.8.3. Giao diện sửa cây thuốc

Hình 5.16: Giao diện sửa cây thuốc

Danh sách biến cố:

Bảng 5.31: Bảng danh sách biến cố của giao diện quản lý cây thuốc (phần sửa cây thuốc)

STT	Điều kiện kích hoạt	Xử lý	Ghi chú
13	Khi nhấn nút “Hủy”.	Đóng modal sửa cây thuốc.	
14	Khi nhấn nút “Lưu”.	Sửa cây thuốc.	

Mô tả giao diện

Bảng 5.32: Bảng mô tả giao diện quản lý cây thuốc (phần sửa cây thuốc)

STT	Tên	Kiểu	Ý nghĩa	Miền giá trị	Giá trị mặc định	Ghi chú
0	i_vietnamese_name	input	Ô sửa tên tiếng việt.	N/A	N/A	
1	i_science_name	input	Ô sửa tên khoa học.	N/A	N/A	
2	i_part_used	input	Ô sửa bộ phận dùng.	N/A	N/A	
3	s_rare_level	select	Sửa mức độ hiếm.	N/A	N/A	
4	s_family	select	Sửa họ.	N/A	N/A	
5	t_desc	text-area	Ô sửa mô tả.	N/A	N/A	
6	t_brief_effect	text-area	Ô sửa công dụng tóm tắt.	N/A	N/A	
7	t_habitat	text-area	Ô sửa nơi sinh sống.	N/A	N/A	
8	t_chemistries	text-area	Ô sửa thành phần hóa học.	N/A	N/A	
9	t_usage	text-area	Ô sửa cách dùng.	N/A	N/A	
10	t_effect	text-area	Ô sửa công dụng.	N/A	N/A	
11	btn_choose_pic	button	Button chọn ảnh cây thuốc.	N/A	N/A	
12	vector	div	Phần sửa 10 giá trị hóa học để cấu thành vector 10 chiều.	N/A	N/A	
13	btn_cancel	button	Nút đóng modal sửa cây thuốc.	N/A	N/A	
14	btn_save	button	Nút sửa cây thuốc.	N/A	N/A	

## 5.2. Kiểm thử

### 5.2.1. Môi trường kiểm thử

Cấu hình máy thử nghiệm:

- CPU: i7-1065G7.
- RAM: 8GB.

- Lưu trữ: SSD 512GB.

### 5.2.2. Chức năng tìm kiếm

Bảng 5.33: Bảng kết quả kiểm thử chức năng tìm kiếm

ID Trường hợp Kiểm thử	Mô tả trường hợp Kiểm thử	Các bước Kiểm thử	Dữ liệu kiểm thử	Kết quả mong đợi	Kết quả thực tế	Thành công/T hất bại
TC_TK_001	Người dùng nhập từ khóa tên cây thuốc vào ô tìm kiếm, hệ thống xử lý và trả về các cây thuốc có tên chứa từ khóa.	1. Truy cập trang tra cứu dược liệu, trang chủ hoặc trang giới thiệu. 2. Nhập từ khóa vào ô tìm kiếm trên thanh header. 4. Bấm nút “Tra cứu”	Từ khóa: “a kê”	Hệ thống trả về các cây thuốc có chứa từ khóa “a kê” và hiển thị trên giao diện. Tốc độ tải dưới 3s	Hệ thống trả về các cây thuốc có chứa từ khóa “a kê” và hiển thị trên giao diện. Tốc độ tải dưới 3s	Thành công
TC_TK_002	Người dùng nhập từ khóa tên cây thuốc vào ô tìm kiếm, hệ thống xử lý và trả về các cây thuốc có tên chứa từ khóa.	1. Truy cập trang tra cứu dược liệu, trang chủ hoặc trang giới thiệu. 2. Nhập từ khóa vào ô tìm kiếm trên thanh header. 4. Bấm nút “Tra cứu”	Từ khóa: “cải trời”	Hệ thống trả về các cây thuốc có chứa từ khóa “cải trời” và hiển thị trên giao diện. Tốc độ tải dưới 3s	Hệ thống trả về các cây thuốc có chứa từ khóa “cải trời” và hiển thị trên giao diện. Tốc độ tải dưới 3s	Thành công
TC_TK_003	Người dùng nhập từ khóa tên cây thuốc vào ô tìm kiếm, hệ thống xử lý và trả về các	1. Truy cập trang tra cứu dược liệu, trang chủ hoặc trang giới thiệu.	Từ khóa: “mâm xôi”	Hệ thống trả về các cây thuốc có chứa từ khóa	Hệ thống trả về các cây thuốc có chứa từ khóa	Thành công

	cây thuốc có tên chứa từ khóa.	2. Nhập từ khóa vào ô tìm kiếm trên thanh header. 4. Bấm nút “Tra cứu”		“mâm xôi” và hiển thị trên giao diện. Tốc độ tải dưới 3s	“mâm xôi” và hiển thị trên giao diện. Tốc độ tải dưới 3s	
--	--------------------------------	---	--	--	--	--

### 5.2.3. Chức năng tìm kiếm nâng cao

Bảng 5.34: Bảng kết quả kiểm thử chức năng tìm kiếm nâng cao

ID Trường hợp Kiểm thử	Mô tả trường hợp Kiểm thử	Các bước Kiểm thử	Dữ liệu kiểm thử	Kết quả mong đợi	Kết quả thực tế	Thành công/T hất bại
TC_TK NC _001	Người dùng nhập 10 thành phần hóa học để tìm kiếm cây tương đồng, hệ thống xử lý và trả về 10 cây tương đồng.	1. Truy cập trang tra cứu. 2. Nhập 10 thành phần hóa học. 4. Bấm nút “Tìm kiếm”	Mảng thành phần: [0, 0, 20, 5, 0, 4.6, 0, 0, 0, 0] (cây A kê).	Hệ thống trả về 10 cây thuốc có độ tương đồng về dược tính với cây “A kê” và hiển thị trên giao diện. Tốc độ tải dưới 3s	Hệ thống trả về 10 cây thuốc có độ tương đồng về dược tính với cây “A kê” và hiển thị trên giao diện. Tốc độ tải dưới 3s	Thành công
TC_TK NC _002	Người dùng nhập 10 thành phần hóa học để tìm kiếm cây tương đồng, hệ thống xử lý và trả về 10 cây tương đồng.	1. Truy cập trang tra cứu. 2. Nhập 10 thành phần hóa học. 4. Bấm nút “Tìm kiếm”	Mảng thành phần: [0, 0, 0.5, 0.22, 0, 0, 0, 0, 0] (cây Mía lau).	Hệ thống trả về 10 cây thuốc có độ tương đồng về dược tính với	Hệ thống trả về 10 cây thuốc có độ tương đồng về dược tính với	Thành công

				cây “Mía lau” và hiển thị trên giao diện. Tốc độ tải dưới 3s	cây “Mía lau” và hiển thị trên giao diện. Tốc độ tải dưới 3s	
TC_TK NC _003	Người dùng nhập 10 thành phần hóa học để tìm kiếm cây tương đồng, hệ thống xử lý và trả về 10 cây tương đồng.	1. Truy cập trang tra cứu. 2. Nhập 10 thành phần hóa học. 4. Bấm nút “Tìm kiếm”	Mảng thành phần: [0, 0, 2.78, 7.55, 3, 0, 0, 10.21, 0, 14.24] (cây Mắm đen).	Hệ thống trả về 10 cây thuốc có độ tương đồng về được tính với cây “Mắm đen” và hiển thị trên giao diện. Tốc độ tải dưới 3s	Hệ thống trả về 10 cây thuốc có độ tương đồng về được tính với cây “Mắm đen” và hiển thị trên giao diện. Tốc độ tải dưới 3s	Thành công

#### 5.2.4. Chức năng xem chi tiết cây thuốc

Bảng 5.35: Bảng kết quả kiểm thử chức năng xem chi tiết cây thuốc

ID Trường hợp Kiểm thử	Mô tả trường hợp Kiểm thử	Các bước Kiểm thử	Dữ liệu kiểm thử	Kết quả mong đợi	Kết quả thực tế	Thành công/T hất bại
TC_XC TCT _001	Người dùng nhấn vào cây thuốc muốn xem chi tiết, hệ thống hiển thị trang chi tiết với các	1. Truy cập trang tra cứu, giới thiệu hoặc trang chủ. 2. Nhấn vào cây thuốc muốn xem.	Cây thuốc: “A kê”	Hệ thống hiển thị trang chi tiết cây “A kê” với đầy đủ các	Hệ thống hiển thị trang chi tiết cây “A kê” với đầy đủ các	Thành công



	đầy đủ các thông tin của cây thuốc.			thông tin của cây. Tốc độ tải dưới 3s	thông tin của cây. Tốc độ tải dưới 3s	
TC_XC TCT _002	Người dùng nhấn vào cây thuốc muốn xem chi tiết, hệ thống hiển thị trang chi tiết với các đầy đủ các thông tin của cây thuốc.	1. Truy cập trang tra cứu, giới thiệu hoặc trang chủ. 2. Nhấn vào cây thuốc muốn xem.	Cây thuốc: “Mía lau”	Hệ thống hiển thị trang chi tiết cây “Mía lau” với đầy đủ các thông tin của cây. Tốc độ tải dưới 3s	Hệ thống hiển thị trang chi tiết cây “Mía lau” với đầy đủ các thông tin của cây. Tốc độ tải dưới 3s	Thành công

### 5.2.5. Chức năng đăng nhập

Bảng 5.36: Bảng kết quả kiểm thử chức năng đăng nhập

ID Trường hợp Kiểm thử	Mô tả trường hợp Kiểm thử	Các bước Kiểm thử	Dữ liệu kiểm thử	Kết quả mong đợi	Kết quả thực tế	Thành công/T hất bại
TC_DN _001	Xác minh người dùng đăng nhập thành công với tên đăng nhập và mật khẩu hợp lệ.	1. Truy cập trang đăng nhập. 2. Khung đăng nhập xuất hiện 3. Nhập vào email và mật khẩu 4. Bấm nút “Đăng nhập”	Tên đăng nhập: “vovanhung2864@gmail.com” Mật khẩu: “Admin123*”	Đăng nhập thành công, người dùng ở giao diện quản trị. Tốc độ tải dưới 3s	Đăng nhập thành công, người dùng ở giao diện quản trị. Tốc độ tải dưới 3s	Thành công

TC_DN_002	Xác minh người dùng đăng nhập thành công với tên đăng nhập và mật khẩu hợp lệ.	1. Truy cập trang đăng nhập. 2. Khung đăng nhập xuất hiện 3. Nhập vào email và mật khẩu 4. Bấm nút “Đăng nhập”	Tên đăng nhập: “dangkhoha013@gmail.com” Mật khẩu: “Admin123*”	Đăng nhập thành công, người dùng ở giao diện quản trị. Tốc độ tải dưới 3s	Đăng nhập thành công, người dùng ở giao diện quản trị. Tốc độ tải dưới 3s	Thành công
-----------	--	---	--	---	---	------------

### 5.2.6. Chức năng thống kê

Bảng 5.37: Bảng kết quả kiểm thử chức năng thống kê

ID Trường hợp Kiểm thử	Mô tả trường hợp Kiểm thử	Các bước Kiểm thử	Dữ liệu kiểm thử	Kết quả mong đợi	Kết quả thực tế	Thành công/T hất bại
TC_XTK_001	Quản trị viên nhấn vào thống kê trong sidebar.	1. Vào trang admin hoặc sau khi đăng nhập thành công. 2. Nhấn vào trang thống kê.	Tên đăng nhập: “dangkhoha013@gmail.com” Mật khẩu: “Admin123*”	Hệ thống hiển thị trang thống kê với đầy đủ các biểu đồ.	Hệ thống hiển thị trang thống kê với đầy đủ các biểu đồ.	Thành công
TC_XTK_002	Người dùng vào giao diện thống kê thông qua url.	1. Nhập url vào trang thống kê.		Hệ thống hiển thị trang đăng nhập yêu cầu phải đăng nhập.	Hệ thống hiển thị trang đăng nhập yêu cầu phải đăng nhập.	Thành công

### 5.2.7. Chức năng quản lý họ cây thuốc

Bảng 5.38: Bảng kết quả kiểm thử chức năng quản lý họ cây thuốc

ID Trường hợp Kiểm thử	Mô tả trường hợp Kiểm thử	Các bước Kiểm thử	Dữ liệu kiểm thử	Kết quả mong đợi	Kết quả thực tế	Thành công/T hất bại
TC_QL HT _001	Tìm kiếm thông tin họ cây thuốc.	1. Truy cập trang quản lý họ cây thuốc. 2. Nhập vào ô tìm kiếm từ khóa muốn tìm.	Từ khóa “Bồ Hòn”	Hệ thống hiển thị thông tin họ cây thuốc “Bồ Hòn” Tốc độ tải dưới 1s	Hệ thống hiển thị thông tin họ cây thuốc “Bồ Hòn” Tốc độ tải dưới 1s	Thành công
TC_QL HT _002	Thêm họ cây thuốc.	1. Truy cập trang quản lý họ cây thuốc. 2. Nhấn thêm họ cây thuốc. 3. Nhập thông tin họ cây thuốc cần thêm.	Tên khoa học: “Fabaceae” Tên tiếng Việt: “Đậu”	Hệ thống thêm thông tin họ cây thuốc vào CSDL. Tốc độ tải dưới 1s	Hệ thống thêm thông tin họ cây thuốc vào CSDL. Tốc độ tải dưới 1s	Thành công
TC_QL HT _003	Sửa họ cây thuốc.	1. Truy cập trang quản lý họ cây thuốc. 2. Nhấn sửa họ cây thuốc “Đậu”. 3. Nhập thông tin họ cây thuốc mới.	Tên khoa học: “Fabaceae2” Tên tiếng Việt: “Đậu xanh”	Hệ thống sửa thông tin họ cây thuốc vào CSDL. Tốc độ tải dưới 1s	Hệ thống sửa thông tin họ cây thuốc vào CSDL. Tốc độ tải dưới 1s	Thành công

### 5.2.8. Chức năng quản lý cây thuốc

Bảng 5.39: Bảng kết quả kiểm thử chức năng quản lý cây thuốc

ID Trường hợp Kiểm thử	Mô tả trường hợp Kiểm thử	Các bước Kiểm thử	Dữ liệu kiểm thử	Kết quả mong đợi	Kết quả thực tế	Thành công/T hất bại
TC_QL CT _001	Tìm kiếm thông tin cây thuốc.	1. Truy cập trang quản lý cây thuốc. 2. Nhập vào ô tìm kiếm từ khóa muốn tìm.	Từ khóa “A kê”	Hệ thống hiển thị đầy đủ thông tin cây “A kê”.Tốc độ tải dưới 1s	Hệ thống hiển thị đầy đủ thông tin cây “A kê”.Tốc độ tải dưới 1s	Thành công
TC_QL CT _002	Thêm cây thuốc.	1. Truy cập trang quản lý họ cây thuốc. 2. Nhấn thêm họ cây thuốc. 3. Nhập thông tin cây thuốc cần thêm.	Họ cây thuốc: “Cúc” Các trường text còn lại: “đậu” Hình: 4 hình ngẫu nhiên. 10 TPHH: Nhập giá trị ngẫu nhiên >0	Hệ thống thêm thông tin cây thuốc vào CSDL. Tốc độ tải dưới 3s	Hệ thống thêm thông tin cây thuốc vào CSDL. Tốc độ tải dưới 3s	Thành công
TC_QL CT _003	Sửa cây thuốc.	1. Truy cập trang quản lý họ cây thuốc. 2. Nhấn sửa họ cây thuốc “Đậu”. 3. Nhập thông tin cây thuốc mới.	Họ cây thuốc: “Ấu” Các trường text còn lại: “đậu xanh” Hình: 4 hình ngẫu nhiên. 10 TPHH: Nhập giá trị ngẫu nhiên >0	Hệ thống sửa thông tin cây thuốc vào CSDL. Tốc độ dưới 3s	Hệ thống sửa thông tin cây thuốc vào CSDL. Tốc độ dưới 3s	Thành công

## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1. Kết quả đạt được

Sau quá trình thực hiện khóa luận tốt nghiệp này, chúng tôi đã đạt được một số kết quả như nhau:

- Đầu tiên, chúng tôi đã tiến hành số hóa được các thông tin cây thuốc từ dữ liệu trên giấy sang dữ liệu số để dễ dàng lưu trữ, tra cứu và xử lý. Với các dữ liệu quan hệ của cây thuốc thì được lưu trong MySQL, trong khi dữ liệu về các vector thành phần hóa học thì được lưu trong cơ sở dữ liệu Qdrant.
- Thứ hai, chúng tôi đã xây dựng thành công một hệ thống thông tin để tra cứu cây thuốc áp dụng cơ sở dữ liệu vector để phục vụ cho mục đích tìm kiếm các cây có độ tương đồng về dược tính.
- Thứ ba, để người dùng có thể dễ dàng thao tác với hệ thống thông tin trên, chúng tôi còn xây dựng cho người dùng một website để thực hiện các thao tác như tra cứu thông tin, tìm kiếm cây tương đồng về dược tính. Với front-end, chúng tôi áp dụng Angular18 và Bootstrap5. Ở back-end, chúng tôi sử dụng ASP .NET Core API để triển khai.
- Thứ tư, trong quá trình thực hiện khóa luận, chúng tôi đã có cái nhìn chuyên sâu hơn về cách sử dụng các công nghệ như Angular, .NET và cơ sở dữ liệu vector.
- Thứ năm, tôi đã lập tài liệu phân tích và thiết kế chi tiết hệ thống gồm các bản vẽ: lược đồ cơ sở dữ liệu, sơ đồ lớp, sơ đồ tuần tự, kèm theo đó là bảng đặc tả chi tiết cho lược đồ cơ sở dữ liệu, sơ đồ lớp dựa trên các yêu cầu chức năng. Các tài liệu này mô tả 1 cách chi tiết hệ thống sẽ được xây dựng dựa trên những gì và các chức năng thực hiện ra sao, đảm bảo tính đúng của hệ thống. Tôi cũng đã lập tài liệu thiết kế giao diện và bảng đặc tả biến cố, các thành phần giao diện một cách chi tiết, đảm bảo giao diện không bị khuyết bất cứ chức năng nào.
- Cuối cùng, chúng tôi đã hoàn thành được các chức năng cần có của một hệ thống tra cứu các cây thuốc như: tìm kiếm các cây thuốc theo từ khóa, tìm kiếm các cây thuốc có độ tương đồng về dược tính, xem thông tin của một cây thuốc bất kì. Ngoài ra, để có thể quản trị website, chúng tôi còn xây dựng riêng một trang quản trị cho người quản trị với các chức năng như: đăng nhập, thống kê (số lượng họ

cây thuốc, cây thuốc, các cây thuốc được tìm kiếm nhiều nhất), quản lý họ cây thuốc, quản lý cây thuốc với các chức năng như (tìm kiếm, thêm, sửa).

Chúng tôi hy vọng hệ thống tra cứu cây thuốc được xây dựng sẽ đóng góp một phần nhỏ trong việc bảo tồn và phát huy giá trị của y học cổ truyền, đồng thời hỗ trợ các nhà nghiên cứu trong việc tìm kiếm và phân tích thông tin về dược liệu một cách hiệu quả hơn.

Qua quá trình thực hiện khóa luận đã giúp chúng tôi rèn luyện và nâng cao các kỹ năng chuyên môn như sử dụng công nghệ Angular, .Net và cơ sở dữ liệu vector, cũng như kỹ năng phân tích, thiết kế và triển khai hệ thống thông tin thực tế. Những kỹ năng sẽ giúp ích chúng tôi rất nhiều trên con đường sự nghiệp.

## 6.2. Hướng phát triển

Phần mềm đáp ứng được nhu cầu cơ bản trong việc tìm kiếm dược liệu tương đồng về 10 thành phần hóa học. Tuy nhiên, vẫn tồn tại một số hạn chế để phát triển phổ biến rộng rãi hơn, để khắc phục chúng tôi đề xuất các hướng phát triển trong tương lai gồm:

- Tìm kiếm theo ngữ nghĩa áp dụng các mô hình học máy (Machine Learning): Áp dụng các mô hình để phân tích ngữ nghĩa về công dụng/dược tính của cây thuốc giúp việc tìm kiếm sẽ tiện dụng hơn đối với người dùng ít có hiểu biết về cây thuốc. Khi cần tra cứu dược liệu về “trị ho” thì kết quả sẽ cung cấp thông tin liên quan đến tác dụng này.
- Gợi ý cây thuốc theo triệu chứng: người dùng có thể mô tả các triệu chứng hoặc vấn đề của mình (ví dụ: đau đầu, nhức vai, sưng phù, mệt mỏi,...). Dựa trên các triệu chứng mà người dùng gặp phải mà để gợi ý cây thuốc, giúp người dùng chưa xác định được bệnh cụ thể có thể dễ dàng tìm kiếm giải pháp trị bệnh từ cây thuốc.
- Tìm kiếm bằng hình ảnh: áp dụng các công nghệ thị giác máy tính để tìm kiếm thông tin cây thuốc bằng cách phân tích hình ảnh cây thuốc. Việc tìm kiếm bằng hình ảnh giúp người dùng không chuyên về y học có thể tiếp cận thông tin cây thuốc nhanh chóng và hiệu quả hơn.
- Cung cấp thông tin chi tiết về liều lượng và sự kết hợp giữa các cây thuốc: hệ thống cung cấp thông tin liều lượng theo hình thức như sắc uống, ngâm, xông. Ngoài ra, phần mềm cần chỉ ra liều lượng theo độ tuổi, tình trạng sức khỏe. Không chỉ vậy

phần mềm cũng cần đưa ra các gợi ý kết hợp với các cây thuốc khác và cảnh báo khi sự kết hợp có tác dụng phụ.

## TÀI LIỆU THAM KHẢO

[1]	R. Schwaber-Cohen, "What is a Vector Database & How Does it Work? Use Cases + Examples," 3 5 2023. [Online]. Available: <a href="https://www.pinecone.io/learn/vector-database/">https://www.pinecone.io/learn/vector-database/</a> .
[2]	E. Cardenas, "Distance Metrics in Vector Search," 15 8 2023. [Online]. Available: <a href="https://weaviate.io/blog/distance-metrics-in-vector-search">https://weaviate.io/blog/distance-metrics-in-vector-search</a> .
[3]	"Vector database management systems: Fundamental concepts, use-cases, and current challenges," [Online]. Available: <a href="https://arxiv.org/pdf/2309.11322">https://arxiv.org/pdf/2309.11322</a> .
[4]	D. Dascalescu, "Vector Embeddings Explained," 16 1 2023. [Online]. Available: <a href="https://weaviate.io/blog/vector-embeddings-explained">https://weaviate.io/blog/vector-embeddings-explained</a> .