

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

**Phân bổ nguồn lực y tế phục vụ quá trình điều trị
dịch bệnh**

NGUYỄN HOÀNG LONG

long.nh184139@sis.hust.edu.vn

Ngành Công nghệ thông tin và truyền thông

Giảng viên hướng dẫn: PGS. TS. Trần Đình Khang

Chữ ký GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 08/2023

LỜI CAM KẾT

Họ và tên sinh viên: Nguyễn Hoàng Long

Điện thoại liên lạc: 0394635899

Email: long.nh184139@sis.hust.edu.vn

Lớp: Việt Nhật IS – K63

Hệ đào tạo: Đại học chính quy

Tôi – *Nguyễn Hoàng Long* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *PGS.TS. Trần Đình Khang*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày tháng năm

Tác giả ĐATN

Họ và tên sinh viên

LỜI CẢM ƠN

Để đồ án này được đạt kết quả như hiện nay em đã nhận được rất nhiều sự hỗ trợ và hướng dẫn của thầy Trần Đình Khang. Em xin chân thành cảm ơn thầy đã hướng dẫn, chỉ bảo tận tình để em hoàn thành được đồ án của mình.

Bên cạnh đó, em cũng xin gửi lời cảm ơn các thầy cô giáo và nhà trường đã tạo điều kiện tốt nhất để em có được môi trường học tập tốt nhất.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Với sự gia tăng nhanh chóng của các loại dịch bệnh truyền nhiễm hiện nay cả về độc lực và tốc độ lây nhiễm, mà tiêu biểu là covid-19, nền y tế trong nước và cả quốc tế đang phải đối đầu với những thách thức và rủi ro tiềm ẩn mà những dịch bệnh này có thể đem lại. Một trong những thách thức đó là công tác vận chuyển, hỗ trợ các trang thiết bị, VTYT một cách kịp thời tới những khu vực có tình hình lây nhiễm nghiêm trọng hoặc có khả năng phát sinh một ổ dịch mới. Từ đó có thể chủ động ngăn chặn, giảm thiểu tối đa những tác động tiêu cực mà những dịch bệnh này đem lại. Ngoài ra đồ án cũng sẽ xem xét tới năng lực vận chuyển, phân bổ các nguồn lực này, để có thể đưa ra những phương án vận chuyển tốt nhất về chi phí.

Hiểu được điều đó, đồ án lần này sẽ từng bước xây dựng nên một hệ thống có khả năng đưa ra những phương án hỗ trợ VTYT giữa các tỉnh thành một cách tối ưu, giúp cho VTYT được phân bổ một cách hợp lý và giảm chi phí vận chuyển.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	3
1.4 Bố cục đồ án	3
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	5
2.1 Khảo sát hiện trạng	5
2.2 Tổng quan chức năng	7
2.2.1 Biểu đồ use case tổng quát	7
2.2.2 Biểu đồ use case phân rã Quản lý người dùng	8
2.2.3 Biểu đồ use case phân rã Xem thông tin tình hình dịch bệnh	9
2.2.4 Biểu đồ use case phân rã Xem thông tin VTYT	9
2.2.5 Biểu đồ use case phân rã Xem tuyến đường hỗ trợ VTYT tối ưu.....	10
2.2.6 Biểu đồ use case phân rã Thêm dữ liệu.....	10
2.2.7 Biểu đồ use case phân rã Đánh giá tình hình dịch bệnh.....	11
2.2.8 Biểu đồ use case phân rã Đánh giá khả năng hỗ trợ VTYT.....	12
2.3 Đặc tả chức năng	12
2.3.1 Đặc tả use case Thêm dữ liệu	12
2.3.2 Đặc tả use case Đánh giá tình hình dịch bệnh.....	13
2.3.3 Đặc tả use case Đánh giá khả năng hỗ trợ VTYT	14
2.3.4 Đặc tả use case Xem tuyến đường hỗ trợ VTYT tối ưu	15
2.4 Yêu cầu phi chức năng	16

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG..... 18

3.1 Cơ sở lý thuyết	18
3.1.1 Thuật toán phân cụm mờ bán giám sát với nhiều tham số mờ hóa ...	18
3.1.2 Giải thuật Dijkstra.....	21
3.2 Nền tảng công nghệ	22
3.2.1 Hệ điều hành	22
3.2.2 Hệ quản trị cơ sở dữ liệu	23
3.2.3 Môi trường lập trình	23
3.2.4 Framework và API hỗ trợ.....	24

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ 29

4.1 Thiết kế kiến trúc.....	29
4.1.1 Lựa chọn kiến trúc phần mềm	29
4.1.2 Thiết kế tổng quan.....	31
4.1.3 Thiết kế chi tiết gói	32
4.2 Thiết kế chi tiết.....	34
4.2.1 Thiết kế giao diện	34
4.2.2 Thiết kế lớp	39
4.2.3 Thiết kế cơ sở dữ liệu	49
4.2.4 Xây dựng bảng đặc trưng phục vụ quá trình phân cụm.....	53
4.3 Xây dựng ứng dụng.....	59
4.3.1 Thư viện và công cụ sử dụng.....	59
4.3.2 Kết quả đạt được	59
4.3.3 Minh họa các chức năng chính	60
4.4 Kiểm thử.....	65
4.4.1 Kiểm thử chức năng Xem thông tin tình hình dịch bệnh.....	65
4.4.2 Kiểm thử chức năng Đánh giá tình hình dịch bệnh	66

4.4.3 Kiểm thử chức năng Đánh giá khả năng cung ứng VTYT	67
4.5 Triển khai ứng dụng	67
4.5.1 Mô hình triển khai.....	67
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	69
5.1 Kết luận	69
5.1.1 So sánh kết quả nghiên cứu với các sản phẩm tương tự	69
5.1.2 Đánh giá tổng quát	70
5.2 Hướng phát triển.....	72
TÀI LIỆU THAM KHẢO.....	75

DANH MỤC HÌNH VẼ

Hình 2.1	Tra cứu tình hình dịch bệnh trên website của Bộ Y Tế	5
Hình 2.2	Biểu đồ use case tổng quát	7
Hình 2.3	Biểu đồ use case phân ra Quản lý người dùng	8
Hình 2.4	Biểu đồ use case phân rã Xem thông tin tình hình dịch bệnh .	9
Hình 2.5	Biểu đồ use case phân rã Xem thông tin VTYT	9
Hình 2.6	Biểu đồ use case phân rã Xem tuyến đường hỗ trợ VTYT tối ưu	10
Hình 2.7	Biểu đồ use case phân rã Thêm dữ liệu	10
Hình 2.8	Biểu đồ use case phân rã Đánh giá tình hình dịch bệnh . . .	11
Hình 2.9	Biểu đồ use case phân rã Đánh giá khả năng hỗ trợ VTYT .	12
Hình 3.1	Minh họa thuật toán phân cụm	18
Hình 3.2	Non-blocking I/O trong Nodejs	24
Hình 3.3	API Traffic-Aware Directions của nextbillon.ai	27
Hình 4.1	Thành phần của mô hình MVC	30
Hình 4.2	Biểu đồ gói UML	31
Hình 4.3	Biểu đồ chi tiết gói cho chức năng Phân tích tình hình dịch bệnh	32
Hình 4.4	Biểu đồ chi tiết gói cho chức năng Phân tích khả năng hỗ trợ VTYT	33
Hình 4.5	Biểu đồ chi tiết gói cho chức năng Xem đề xuất phân bổ VTYT	33
Hình 4.6	Minh họa Layout tổng quát của trang web	35
Hình 4.7	Minh họa màn hình chức năng Thêm dữ liệu tĩnh	36
Hình 4.8	Minh họa màn hình chức năng Thêm dữ liệu động	36
Hình 4.9	Minh họa màn hình chức năng Xem số liệu dịch bệnh	37
Hình 4.10	Minh họa màn hình chức năng Xem số liệu VTYT	37
Hình 4.11	Minh họa màn hình chức năng Đánh giá khả năng hỗ trợ VTYT	38
Hình 4.12	Thiết kế chi tiết lớp trong chức năng Phân tích tình hình dịch bệnh	39
Hình 4.13	Giao tiếp giữa các lớp trong chức năng Phân tích tình hình dịch bệnh	40
Hình 4.14	Thiết kế chi tiết lớp trong chức năng Phân tích khả năng hỗ trợ VTYT	43
Hình 4.15	Giao tiếp giữa các lớp trong chức năng Phân tích khả năng hỗ trợ VTYT	44
Hình 4.16	Thiết kế chi tiết lớp trong chức năng Xem đề xuất phân bổ VTYT	47

Hình 4.17 Giao tiếp giữa các lớp trong chức năng Xem để xuất phân bổ	
VTYT	48
Hình 4.18 Biểu đồ thực thể liên kết	50
Hình 4.19 Màn hình chức năng Xem tình hình dịch bệnh	61
Hình 4.20 Màn hình chức năng Xem khả năng cung ứng VTYT	61
Hình 4.21 Màn hình chức năng Tra cứu tuyến đường cung ứng VTYT . .	62
Hình 4.22 Màn hình chức năng Quản lý tài khoản	62
Hình 4.23 Màn hình Thêm trọng số cho các thuộc tính của bảng đặc trưng tình hình dịch bệnh	63
Hình 4.24 Màn hình phân cụm tính thành theo cấp độ dịch	63
Hình 4.25 Màn hình Thêm trọng số cho các thuộc tính của bảng đặc trưng khả năng phân bổ VTYT	64
Hình 4.26 Màn hình phân cụm tính thành theo khả năng phân bổ VTYT	64
Hình 4.27 Màn hình chức năng Thêm dữ liệu	65
Hình 4.28 Quy trình triển khai server lên Heroku	68

DANH MỤC BẢNG BIỂU

Bảng 2.1	Đặc tả use case Thêm dữ liệu	13
Bảng 2.2	Đặc tả use case Đánh giá tình hình dịch bệnh	14
Bảng 2.3	Đặc tả use case Đánh giá khả năng cung ứng VTYT	15
Bảng 2.4	Đặc tả use case Xem tuyên đường hỗ trợ VTYT tối ưu	16
Bảng 4.1	Thiết kế lớp Tình hình dịch bệnh từng thành phố (Element) . .	41
Bảng 4.2	Thiết kế Lớp Tình hình dịch bệnh tất cả các tỉnh thành (Situ- ation)	41
Bảng 4.3	Thiết kế Lớp Dữ liệu phân cụm (SSmcFcm)	42
Bảng 4.4	Thiết kế lớp Tình hình VTYT từng thành phố (SupplyElement)	45
Bảng 4.5	Thiết kế lớp Dữ liệu tiền xử lý tìm đường đi (RoutePreData) .	49
Bảng 4.6	Thiết kế lớp Dijkstra	49
Bảng 4.7	Mô tả thực thể Tỉnh thành phố trong CSDL	51
Bảng 4.8	Mô tả thực thể dịch bệnh	51
Bảng 4.9	Mô tả thực thể Vật tư y tế	51
Bảng 4.10	Mô tả thực thể Người dùng	51
Bảng 4.11	Mô tả liên kết Thông kê số ca nhiễm	52
Bảng 4.12	Mô tả liên kết Thông kê số ca hồi phục	52
Bảng 4.13	Mô tả liên kết Thông kê số ca tử vong	52
Bảng 4.14	Mô tả khoảng cách	52
Bảng 4.15	Mô tả liên kết Sở hữu VTYT	53
Bảng 4.16	Bảng đặc trưng tình hình dịch bệnh	54
Bảng 4.17	Bảng đặc trưng khả năng hỗ trợ VTYT	57
Bảng 4.18	Danh sách thư viện và công cụ sử dụng	59
Bảng 4.19	Thông tin thống kê về ứng dụng	60
Bảng 4.20	Bảng Kiểm thử chức năng Xem thông tin tình hình dịch bệnh	65
Bảng 4.21	Bảng Kiểm thử chức năng Đánh giá tình hình dịch bệnh . .	66
Bảng 4.22	Bảng Kiểm thử chức năng Đánh giá khả năng cung ứng VTYT	67

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
EUD	Phát triển ứng dụng người dùng cuối(End-User Development)
GWT	Công cụ lập trình Javascript bằng Java của Google (Google Web Toolkit)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
IaaS	Dịch vụ hạ tầng

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Những năm gần đây tình hình dịch bệnh truyền nhiễm trên thế giới diễn biến khó lường, khó dự báo. Trong bối cảnh toàn cầu hóa ngày càng phát triển, các đại dịch nói chung đang tăng dần tần suất xuất hiện. Tình trạng ô nhiễm môi trường, biến đổi khí hậu, thiên tai, lụt bão, đô thị hóa và di dân gia tăng là điều kiện thuận lợi cho các dịch bệnh truyền nhiễm xuất hiện, lây lan và bùng phát. Năm 2022, thế giới tiếp tục ghi nhận các trường hợp mắc, tử vong do các bệnh lây truyền từ động vật sang người, các bệnh truyền nhiễm nguy hiểm, mới nổi tại nhiều quốc gia, mới nhất là sự lây lan của bệnh Marburg tại khu vực châu Phi; một số bệnh lưu hành, bệnh có vắc xin dự phòng cũng gia tăng số mắc ở nhiều nơi. Và đáng lưu tâm nhất trong những năm gần đây chính là sự xuất hiện của một biến chủng mới của virut gây suy hô hấp cấp tính SARS, được đặt tên là Covid-19.

Dịch bệnh Covid-19 đã làm thay đổi hoàn toàn cuộc sống của con người trên toàn thế giới. Điều đó được chứng minh qua số lượng ca nhiễm và tỷ lệ tử vong cao trong những tháng đầu tiên của đại dịch. Trước đó, chúng ta đã chứng kiến sự lan rộng nhanh chóng của virus này từ Trung Quốc sang các quốc gia khác trên toàn thế giới.

Các quốc gia đã bắt đầu đưa ra các biện pháp như giãn cách xã hội, đóng cửa các hoạt động không cần thiết, hạn chế đi lại và giảm thiểu số lượng người tập trung. Tuy nhiên, những biện pháp này vẫn chưa đủ để ngăn chặn sự lây lan của virus.

Tại Việt Nam, chính phủ đã đưa ra các biện pháp giãn cách xã hội và yêu cầu người dân đeo khẩu trang, đồng thời khuyến khích tập trung vào việc xét nghiệm và truy vết những người tiếp xúc gần với những người nhiễm bệnh. Sự quyết tâm của chính phủ và sự chung tay của toàn xã hội đã giúp Việt Nam kiểm soát được dịch bệnh và giảm thiểu tỷ lệ tử vong. Từ đó từng bước kiểm soát được bệnh dịch và đưa người dân trở lại trạng thái bình thường mới. Đến thời điểm hiện tại, kinh tế-xã hội Việt Nam đã gần như không còn bị ảnh hưởng bởi Covid-19.

Vậy tại sao lại lựa chọn nghiên cứu đề tài này? Nó có còn thiết thực trong tình hình hiện nay hay không? Dịch bệnh Covid-19 vẫn đang còn diễn biến phức tạp trên toàn thế giới. Các biến thể mới của virus cũng đang xuất hiện, đẩy các quốc gia phải đổi mới với nhiều thách thức mới trong việc kiểm soát dịch bệnh. Với sự phát triển của vaccine Covid-19, hy vọng của toàn thế giới là dịch bệnh sẽ được kiểm soát và đưa vào quản lý trong tương lai gần. Tuy nhiên, chúng ta vẫn cần phải giữ

cảnh giác và đồng lòng để đối phó với dịch bệnh này, và hi vọng rằng mọi người trên toàn thế giới sẽ tiếp tục đóng góp cho nỗ lực chung của chúng ta để vượt qua đại dịch này.

Dù Covid-19 đã kết thúc nhưng nó để lại nhiều bài học "đắt giá" không riêng gì cho ngành y tế mà cho tất cả các lĩnh vực khác. Vậy nên, để có thể hạn chế việc xuất hiện và bùng phát các dịch khác trong tương lai thì chúng ta cần áp dụng những bài học kinh nghiệm mà Covid-19 đã để lại.

Dịch bệnh qua đi không những gây ảnh hưởng xấu tới sức khỏe của con người mà còn ảnh hưởng tới kinh tế, xã hội, chính trị... của cả thế giới. Những hệ lụy mà dịch bệnh để lại có thể kéo dài và rất khó khắc phục vậy nên việc rút kinh nghiệm từ các đợt dịch đã từng xảy trước đây để phòng, chống và ứng phó kịp thời với những đợt dịch tiếp theo là vô cùng cần thiết.

Tại Việt Nam những năm 2021-2022, dịch bệnh Covid-19 đã có những diễn biến rất phức tạp, gây ra nhiều khó khăn trong công tác phòng chống dịch của cơ quan chức năng. Một trong những khó khăn đó là sự thiếu hụt về trang thiết bị, VTYT tại các điểm nóng về dịch bệnh (tiêu biểu như thành phố Hồ Chí Minh thời điểm tháng 5/2021, Hải Dương thời điểm tháng 11/2021,...). Việc thiếu thốn trang thiết bị VTYT phục vụ cho phòng chống dịch đã khiến cho bước đầu ngăn chặn dịch tại các điểm nóng về Covid-19 gặp rất nhiều thách thức.

Sau 2 năm, dịch bệnh dần được kiểm soát, mọi người cũng đã quên dần sự tồn tại của dịch bệnh. Tuy nhiên trong tương lai, Covid-19 hoặc 1 loại bệnh truyền nhiễm khác vẫn có thể bùng phát tại Việt Nam. Bài học về sự chủ quan trong công tác phòng chống dịch vẫn còn nguyên giá trị của nó. Sự chủ động và tích cực trong quá trình phòng chống dịch chính là chìa khóa quan trọng cho việc đảm bảo khả năng ngăn ngừa cũng như kiểm soát các loại dịch bệnh có thể lan rộng và gây ra những tổn hại nặng nề cho kinh tế - xã hội như đã xảy ra trong đại dịch Covid-19.

Vì vậy em đã quyết định lựa chọn đề tài “Phân bổ nguồn lực y tế phục vụ quá trình điều trị dịch bệnh”. Đồng thời em cũng xét đến vấn đề tối ưu chi phí trong việc phân bổ, cụ thể ở đây là tìm những tỉnh thành gần nhất có thể hỗ trợ VTYT cho khu vực đang cần.

1.2 Mục tiêu và phạm vi đề tài

Hiện nay có rất nhiều ứng dụng và website cung cấp cho người dùng dịch vụ kiểm tra về tình hình dịch bệnh truyền nhiễm. Có thể kể đến các ứng dụng/website sau: Ứng dụng “Vietnam Health Declaration” của Bộ Y tế, website của Bộ Y tế, ứng dụng Bluezone, website của Báo Tuổi Trẻ, website của Bộ Thông tin và Truyền

thông.

Có rất nhiều hệ thống cung cấp cho người dùng thông tin về tình hình dịch bệnh. Tuy nhiên có thể nhận thấy rằng những hệ thống này chỉ mang tới cho người dùng những dữ liệu mang tính chất thống kê như số lượng người mắc bệnh hay tử vong. Người dùng sẽ rất khó có thể có được một cái nhìn trực quan về mức độ lây nhiễm của dịch bệnh.

Khi mở rộng phạm vi người dùng là các nhân viên cơ sở y tế cấp thành phố, rất khó có thể tìm ra những hệ thống đánh giá khả năng tự cung ứng của những địa phương đó trước tình hình dịch bệnh hiện tại, xa hơn nữa là hỗ trợ/nhận hỗ trợ VTYT từ các tỉnh thành khác.

Dựa trên những phân tích và đánh giá ở trên, đồ án lần này sẽ hướng tới giải quyết những hạn chế mà các ứng dụng y tế hiện nay đang có. Đối với người dùng bình thường, đó là việc xây dựng hệ thống đánh giá trực quan về tình hình dịch bệnh; còn đối với các cơ sở y tế, cần xây dựng hệ thống đánh giá khả năng hỗ trợ/tự cung ứng của các địa phương, cùng với đó là xem xét các đề xuất vận chuyển VTYT hỗ trợ được tối ưu.

Phạm vi nghiên cứu của đề tài này là thu thập dữ liệu về tình hình dịch bệnh và VTYT. Nó sẽ là những dữ liệu cần thiết cho việc tính toán và đánh giá sau này. Sau đó là sử dụng những dữ liệu nhận được phục vụ cho bài toán vận chuyển.

1.3 Định hướng giải pháp

Khi đã xác định được mục tiêu cần giải quyết và phạm vi đề tài, chúng ta cần phải hoàn thành những việc sau:

i. Thu thập các dữ liệu cần thiết bằng các API được cung cấp bởi bên thứ 3. Sau đó xây dựng lên một hệ thống phân tích và trả lại cho phía người dùng các thông tin có ích. Đây chính là mục tiêu chính trong đồ án lần này.

ii. Xây dựng giao diện người dùng: giúp người dùng có thể tương tác với hệ thống để tìm kiếm những thông tin hữu ích, giao diện ứng dụng cũng yêu cầu phân quyền người dùng để cung cấp các dịch vụ tương ứng.

1.4 Bố cục đồ án

Dưới đây là khái quát về những việc cần làm trong các chương sau:

Chương 2: Đi sâu vào phân tích chi tiết các yêu cầu của đề tài, đưa ra được mô tả tổng quan các chức năng hệ thống sẽ có và đặc tả những chức năng đó.

Chương 3: Giới thiệu về các công nghệ, nền tảng sử dụng trong đồ án.

Chương 4: Bắt tay vào việc thiết kế các mô hình, thuật toán, giao diện người

dùng, kiểm thử và triển khai.

Chương 5: Trình bày những nội dung đóng góp tâm đắc nhất trong đồ án.

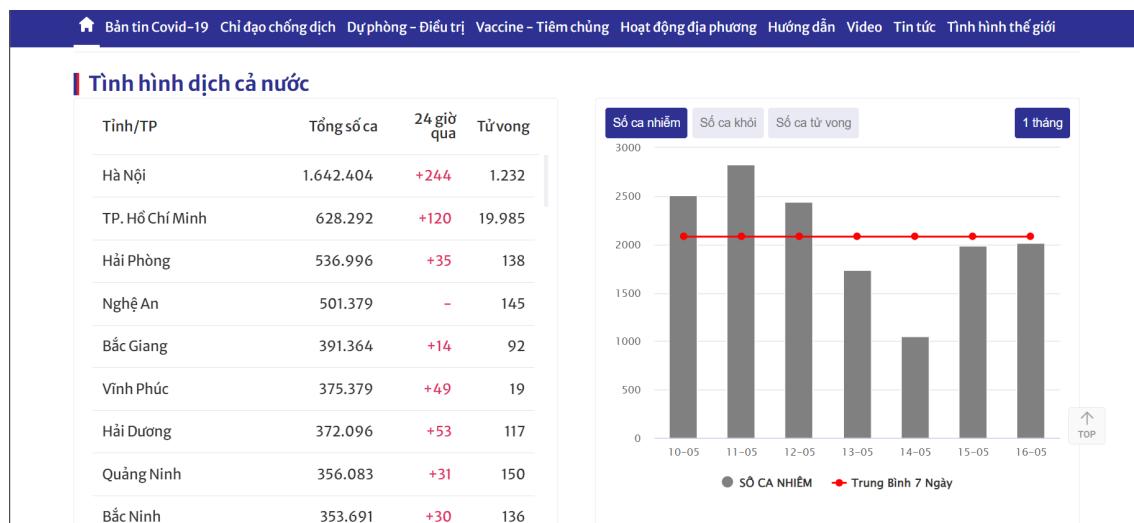
Chương 6: So sánh kết quả nghiên cứu với các sản phẩm nghiên cứu tương tự , đưa ra những hướng phát triển cho hệ thống trong tương lai.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

Cùng với sự phát triển của internet là tốc độ tăng chóng mặt số lượng người sử dụng internet. Trong thời điểm đại dịch Covid-19 diễn ra, nhu cầu về các ứng dụng hoặc website cho phép quan sát tình hình dịch bệnh rất lớn. Năm bắt được tâm lý này, rất nhiều ứng dụng/website đã được phát triển và phát hành tại Việt Nam. Có thể kể đến như:

- Ứng dụng “Vietnam Health Declaration” của Bộ Y tế: Ứng dụng này giúp người dân tự khai báo tình trạng sức khỏe và tiếp cận thông tin cập nhật về dịch bệnh tại Việt Nam.
- Website của Bộ Y tế, Ứng dụng Bluezone, Website của Báo Tuổi Trẻ, Website của Bộ Thông tin và Truyền thông: Trang web này cập nhật thông tin về tình hình dịch bệnh tại Việt Nam, bao gồm số lượng ca mắc mới, tử vong và phục hồi.
- Ứng dụng COVID-19 Tracker: Ứng dụng này cung cấp thông tin về tình hình dịch bệnh tại các tỉnh thành của Việt Nam.



Hình 2.1: Tra cứu tình hình dịch bệnh trên website của Bộ Y Tế

Điểm chung của các hệ thống này là chỉ đáp ứng được ở mức tối thiểu nhu cầu theo dõi tình hình dịch bệnh diễn ra trên khắp cả nước. Các hệ thống trên sẽ đưa ra các dữ liệu mang tính chất thống kê như số ca mắc mới, số ca hồi phục, số ca tử vong,... Khi người dùng đọc những thông tin đấy, họ cũng khó có thể tổng quát hóa lượng thông tin mình vừa đọc để có thể có được một bức tranh tổng quát về tình hình dịch bệnh đang diễn ra.

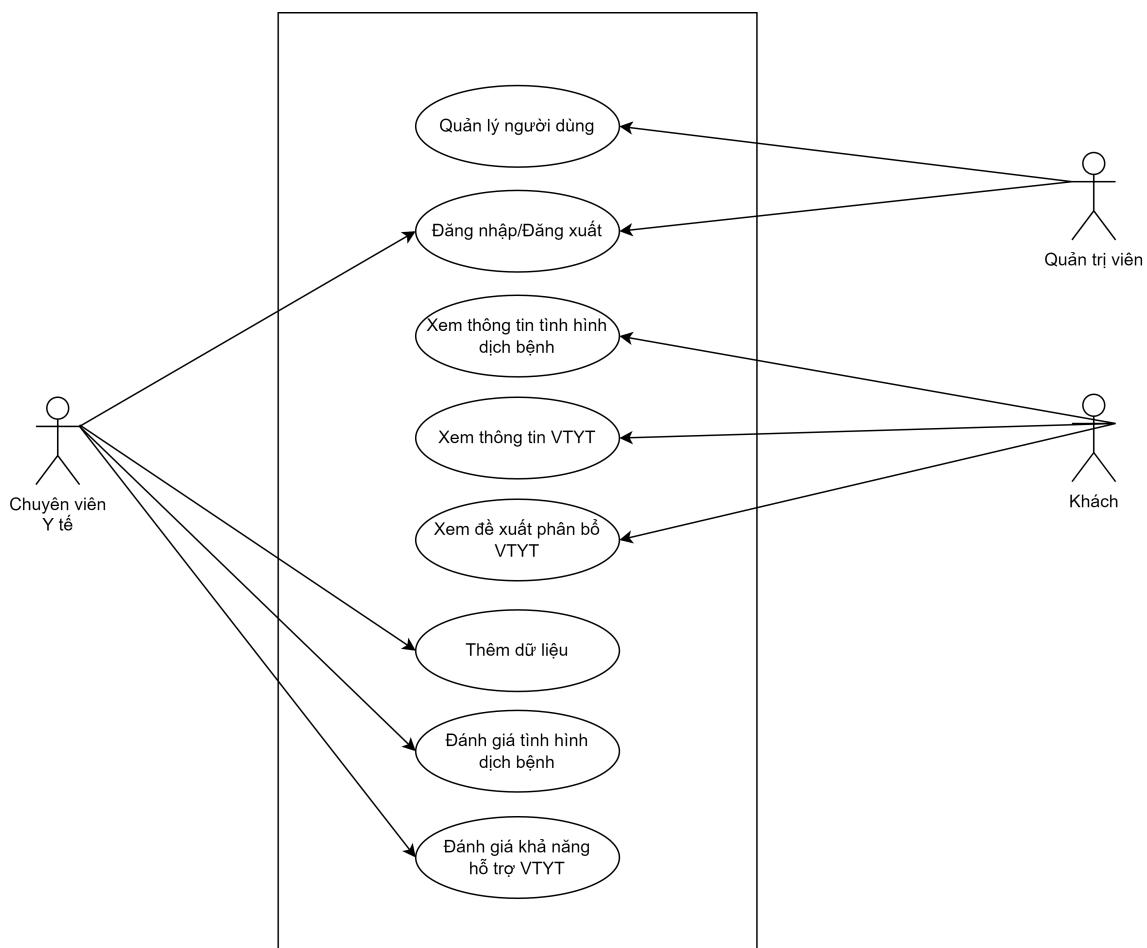
Đó là với trường hợp người dùng là những người không có chuyên môn về y tế, còn với những người làm việc tại các cơ sở y tế thì sao? Họ không chỉ quan tâm về tình hình dịch bệnh mà còn phải chú ý tới tình hình sử dụng trang thiết bị vật tư y tế phục vụ cho quá trình điều trị bệnh. Vì vậy những thông tin về khả năng tự cung ứng hay có khả năng hỗ trợ VTYT của các tỉnh thành cũng là thông tin rất cần thiết. Nhưng đến thời điểm hiện tại tôi vẫn chưa tìm thấy thông tin nào về việc các cơ sở y tế của Việt Nam có sử dụng những hệ thống phục vụ việc khai thác thông tin về khả năng hỗ trợ VTYT của các tỉnh thành.

Những vấn đề thiết thực và có nhu cầu lớn như vậy lại chưa có một ứng dụng nào hiện nay có thể giải quyết triệt để được. Điều đó thật đáng tiếc vì nếu có thể giải quyết những vấn đề đó thì chúng ta chắc chắn sẽ tăng sự chủ động trong công tác phòng chống các loại bệnh truyền nhiễm.

Vì vậy mà đề tài này sẽ nghiên cứu và xây dựng một hệ thống có thể giải quyết các vấn đề: đánh giá tình hình dịch bệnh và khả năng hỗ trợ VTYT của các tỉnh thành. Mở rộng hơn là bài toán vận chuyển VTYT giữa các địa phương với nhau với lộ trình tối ưu (về quãng đường).

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát



Hình 2.2: Biểu đồ use case tổng quát

Hệ thống bao gồm 3 tác nhân chính:

Quản trị viên: Người chịu trách nhiệm quản lý những người dùng khác, có nhiệm vụ thêm/sửa/xóa tài khoản người dùng.

Khách: Những người không đăng nhập vào hệ thống, truy cập vào hệ thống với mục đích tra cứu thông tin.

Chuyên viên y tế: Những nhân viên y tế có thẩm quyền cập nhật và chỉnh sửa dữ liệu y tế trên hệ thống.

Các use case chính được dùng trong hệ thống bao gồm:

Đăng nhập/ Đăng xuất: Chức năng cơ bản nhất, giúp xác định quyền hạn của người dùng.

Quản lý người dùng: Cho phép thêm, sửa hoặc xóa các tài khoản khác, chỉ được thực hiện bởi quản trị viên.

Xem thông tin tình hình dịch bệnh: Xem số liệu chi tiết về tình hình dịch bệnh và xem đánh giá cấp độ dịch bệnh tại các tỉnh thành.

Xem thông tin VTYT: Xem số liệu về số lượng VTYT từng loại mà mỗi tỉnh thành đang có, xem đánh giá về khả năng tự cung ứng/hỗ trợ VTYT của các tỉnh thành.

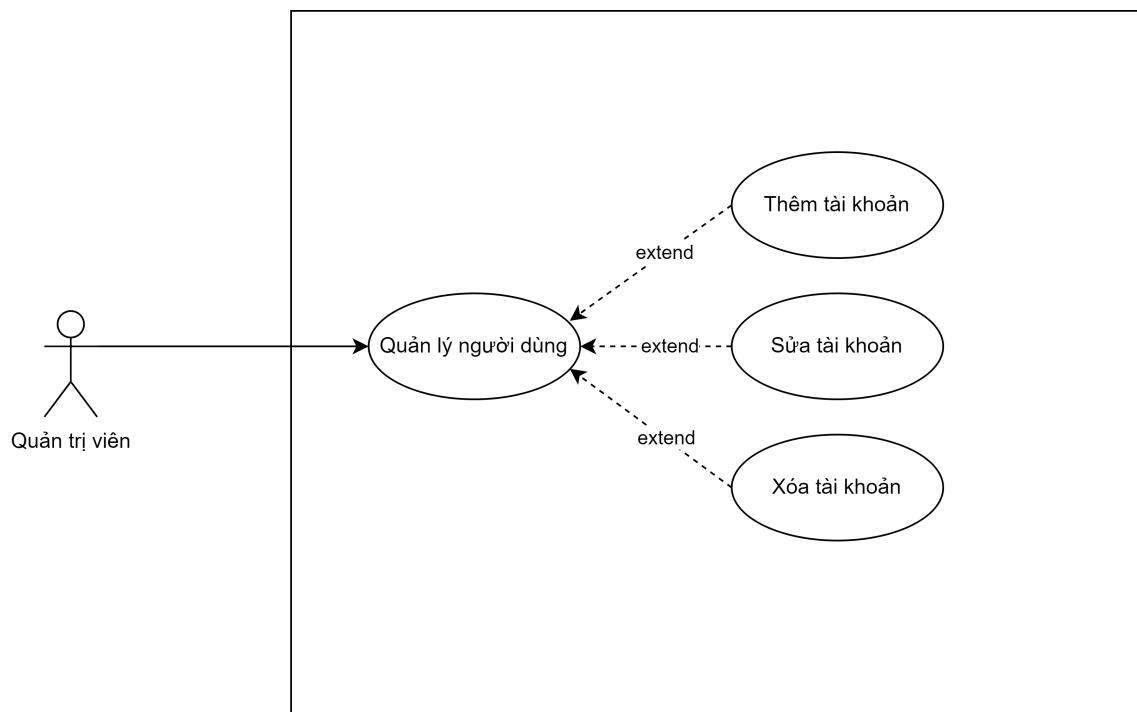
Xem tuyến đường hỗ trợ VTYT tối ưu: Xem các tuyến đường phân bổ VTYT được đề xuất. Các tuyến đường được đề xuất là tối ưu về mặt khoảng cách.

Thêm dữ liệu: Thêm dữ liệu vào hệ thống.

Đánh giá tình hình dịch bệnh: Đánh giá cấp độ dịch bệnh tại các tỉnh thành.

Đánh giá tình hình cung ứng VTYT: Đánh giá khả năng tự cung ứng/ hỗ trợ VTYT tới các tỉnh thành khác.

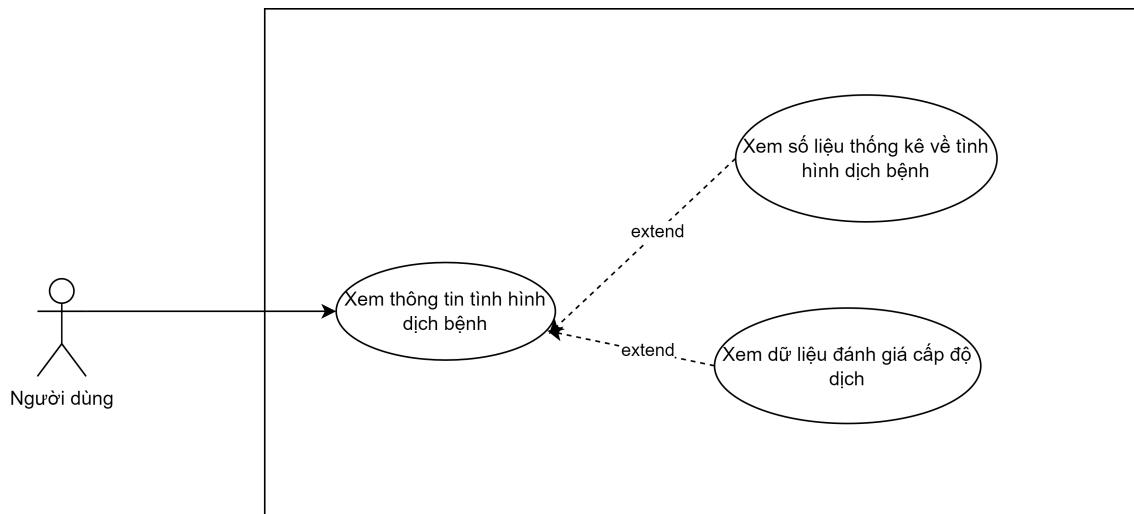
2.2.2 Biểu đồ use case phân rã Quản lý người dùng



Hình 2.3: Biểu đồ use case phân ra Quản lý người dùng

Trong use case Quản lý người dùng có các use case Thêm tài khoản, Sửa tài khoản, Xóa tài khoản. Trước khi thực hiện các use case đó thì cần kiểm tra quyền của tài khoản cần quản lý đó.

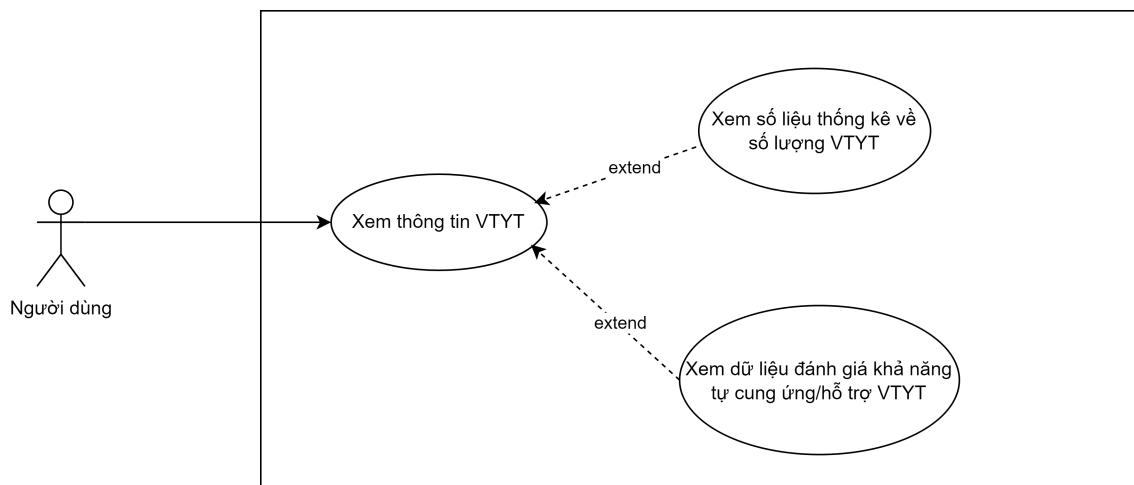
2.2.3 Biểu đồ use case phân rã Xem thông tin tình hình dịch bệnh



Hình 2.4: Biểu đồ use case phân rã Xem thông tin tình hình dịch bệnh

Khi sử dụng chức năng Xem thông tin tình hình dịch bệnh, người dùng có thể tra cứu được số liệu các ca bệnh và thông tin cấp độ dịch tương ứng với các lựa chọn của người dùng.

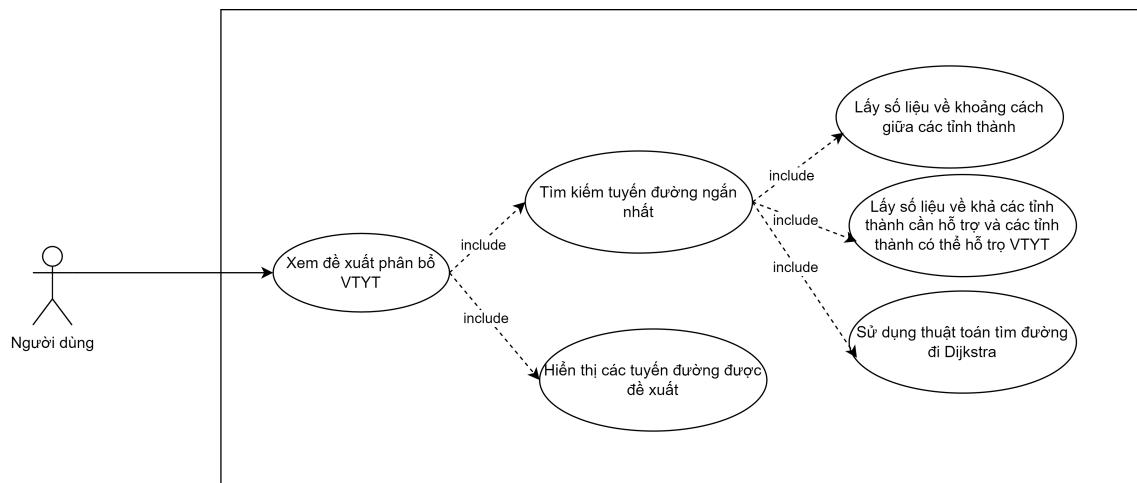
2.2.4 Biểu đồ use case phân rã Xem thông tin VTYT



Hình 2.5: Biểu đồ use case phân rã Xem thông tin VTYT

Tương tự như chức năng Xem thông tin tình hình dịch bệnh, người dùng có thể tra cứu thông tin về số lượng VTYT mỗi tỉnh thành đang sở hữu tương ứng với từng thành phố và từng loại bệnh dịch.

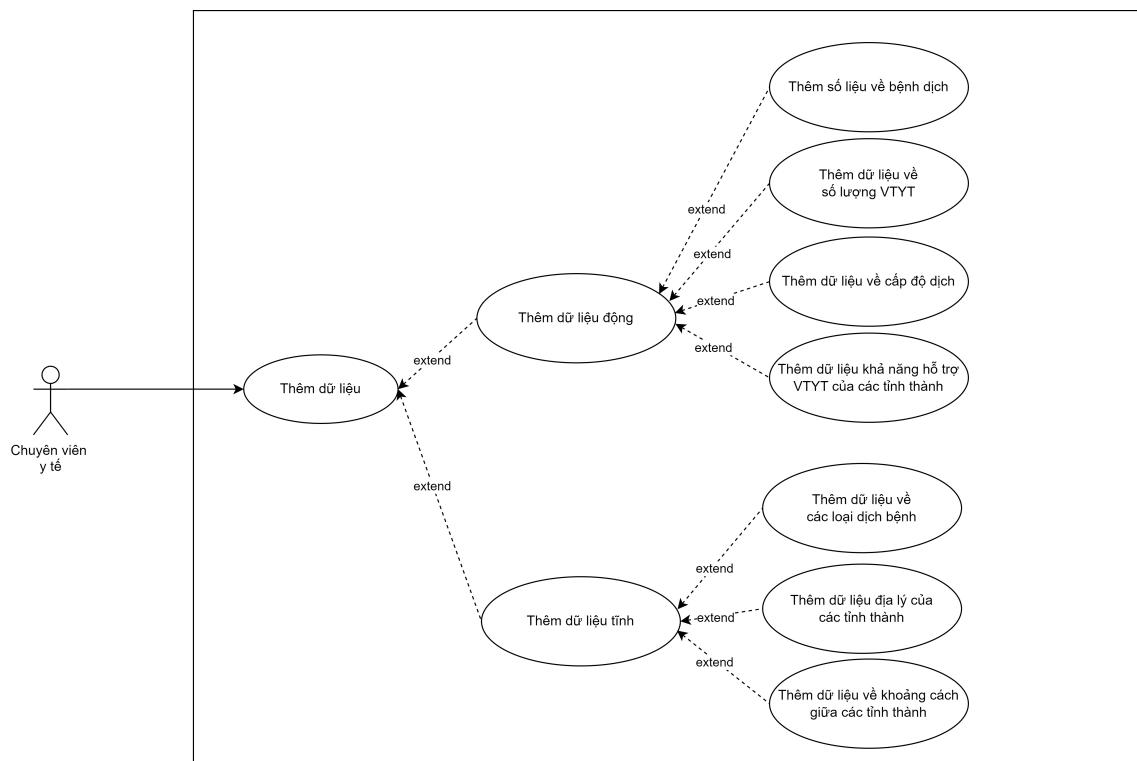
2.2.5 Biểu đồ use case phân rã Xem tuyến đường hỗ trợ VTYT tối ưu



Hình 2.6: Biểu đồ use case phân rã Xem tuyến đường hỗ trợ VTYT tối ưu

Trước khi xem các tuyến đường phân bổ được đề xuất, người dùng phải chọn tỉnh thành phố cần hỗ trợ VTYT. Hệ thống sẽ tiến hành phân tích và đưa ra được tuyến đường cung ứng VTYT tối ưu, cần lấy được dữ liệu về khoảng cách giữa các tỉnh thành và số liệu đánh giá tình hình cung ứng VTYT để tiến hành phân tích. Sau đó người dùng có thể xem được các phương án phân bổ VTYT từ những nơi có khả năng hỗ trợ VTYT cho các tỉnh thành khác.

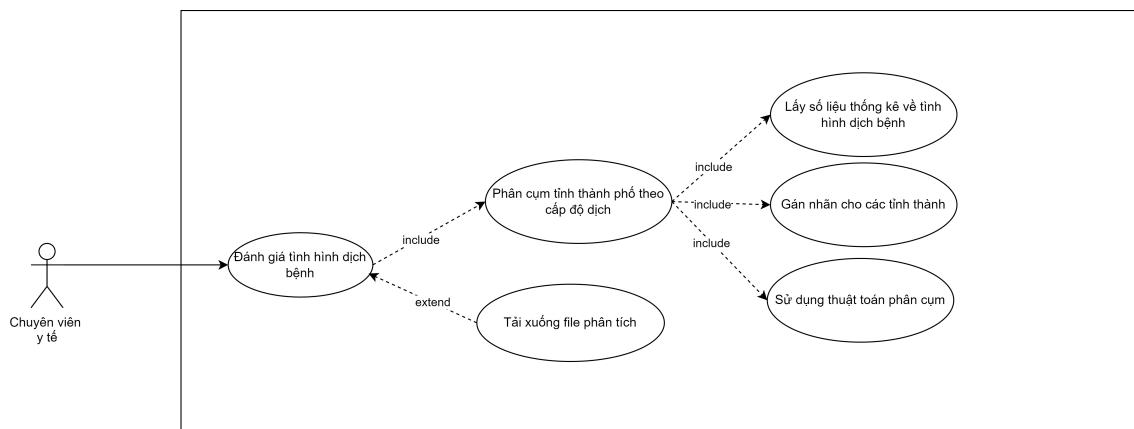
2.2.6 Biểu đồ use case phân rã Thêm dữ liệu



Hình 2.7: Biểu đồ use case phân rã Thêm dữ liệu

Dữ liệu được đưa vào trong CSDL có thể chia làm 2 loại: dữ liệu tĩnh và dữ liệu động. Dữ liệu tĩnh là những dữ liệu không cần cập nhật hàng ngày xuyên (bao gồm dữ liệu về thông tin địa lý của từng tỉnh thành, khoảng cách giữa các tỉnh thành, các loại bệnh dịch). Dữ liệu động là những dữ liệu yêu cầu cập nhật hàng ngày (bao gồm dữ liệu về các ca bệnh, số lượng VTYT của từng tỉnh thành, cấp độ dịch, khả năng hỗ trợ VTYT)

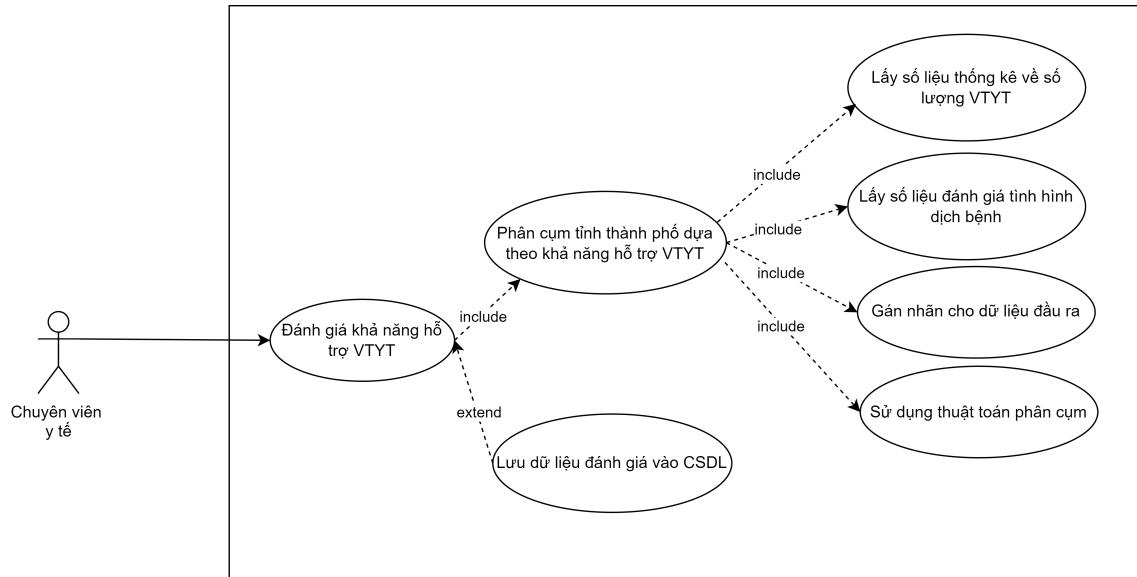
2.2.7 Biểu đồ use case phân rã Đánh giá tình hình dịch bệnh



Hình 2.8: Biểu đồ use case phân rã Đánh giá tình hình dịch bệnh

Trong usecase đánh giá tình hình dịch bệnh, trước tiên cần có được số liệu về tình hình dịch bệnh (số ca nhiễm, số ca hồi phục, số ca tử vong), sau đó tác nhân sẽ gán nhãn cho một số tỉnh thành. Nhãn chính là cấp độ dịch của tỉnh thành đó. Khi đã có được một số nhãn, tác nhân có thể sử dụng tính năng phân cụm để phân loại nhanh các tỉnh thành theo cấp độ dịch. Tác nhân có thể lựa chọn tải về file kết quả đánh giá.

2.2.8 Biểu đồ use case phân rã Đánh giá khả năng hỗ trợ VTYT



Hình 2.9: Biểu đồ use case phân rã Đánh giá khả năng hỗ trợ VTYT

Trong usecase Đánh giá khả năng cung ứng VTYT, chuyên viên y tế có thể phân cụm tỉnh thành phố dựa trên khả năng cung ứng VTYT, trước tiên là lấy về số liệu VTYT, số liệu về cấp độ dịch. Sau đó chuyên viên y tế gán nhãn cho một số tỉnh thành về khả năng hỗ trợ VTYT. Khi đã được một số nhãn, chuyên viên y tế có thể tiến hành phân cụm các tỉnh thành chưa có nhãn bằng thuật toán phân cụm. Cuối cùng họ có thể lựa chọn tải xuống kết quả đánh giá.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Thêm dữ liệu

Tên use case	Thêm dữ liệu
Luồng sự kiện chính	1. Chuyên viên y tế lựa chọn chức năng Thêm dữ liệu trên giao diện người dùng. 2. Chuyên viên y tế import file data. 3. Hệ thống xác định file data có phần mở rộng là cvs hay không. 4. Hệ thống đọc dữ liệu trong file data và chuyển thành script sql. 5. Hệ thống thêm dữ liệu mới vào CSDL. 6. Hệ thống thông báo thành công. 7. Hệ thống ghi lại hoạt động thêm dữ liệu vào trong Activity Log.
Luồng sự kiện phát sinh	3a. Hệ thống thông báo lỗi phần mở rộng của file data. 5a. Hệ thống thông báo lỗi do dữ liệu cần thêm vào xảy ra xung đột với dữ liệu gốc trong CSDL.
Tiền điều kiện	Chuyên viên y tế đăng nhập thành công.
Hậu điều kiện	Dữ liệu được thêm vào CSDL.

Bảng 2.1: Đặc tả use case Thêm dữ liệu

2.3.2 Đặc tả use case Đánh giá tình hình dịch bệnh

Tên use case	Đánh giá tình hình dịch bệnh
Luồng sự kiện chính	1. Chuyên viên y tế lựa chọn chức năng Đánh giá tình hình dịch bệnh trên giao diện người dùng.
	2. Chuyên viên y tế lựa chọn loại dịch bệnh và thời gian cần đánh giá.
	3. Hệ thống lấy dữ liệu thống kê tình hình dịch bệnh.
	4. Chuyên viên y tế gán nhãn cấp độ dịch cho một số tỉnh thành.
	5. Chuyên viên y tế nhấn nút "phân cụm"
	6. Hệ thống sử dụng thuật toán phân cụm để tiến hành đánh giá tình hình dịch bệnh tại các tỉnh thành.
	7. Hệ thống thông báo đánh giá thành công.
	8. Hệ thống hiển thị kết quả đánh giá trên màn hình.
	9. Chuyên viên y tế lựa chọn tải xuống file kết quả đánh giá
	10. Hệ thống ghi lại hoạt động vào trong Activity Log.
Luồng sự kiện phát sinh	
Tiền điều kiện	Chuyên viên y tế đăng nhập thành công.
Hậu điều kiện	Hoạt động được ghi lại trong Activity Log.

Bảng 2.2: Đặc tả use case Đánh giá tình hình dịch bệnh

2.3.3 Đặc tả use case Đánh giá khả năng hỗ trợ VTYT

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Tên use case	Đánh giá khả năng hỗ trợ VTYT
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Chuyên viên y tế lựa chọn chức năng Đánh giá khả năng hỗ trợ VTYT trên giao diện người dùng. 2. Chuyên viên y tế lựa chọn loại dịch bệnh cần đánh giá. 3. Hệ thống lấy số liệu về VTYT. 4. Hệ thống lấy số liệu đánh giá về tình hình dịch bệnh 5. Chuyên viên y tế gán nhãn cho một số tỉnh thành về khả năng cung ứng VTYT. 6. Chuyên viên y tế nhấn nút "phân cụm" 7. Hệ thống sử dụng thuật toán phân cụm để tiến hành đánh giá khả năng hỗ trợ VTYT tại các tỉnh thành. 8. Hệ thống thông báo đánh giá thành công. 9. Hệ thống hiển thị kết quả đánh giá trên màn hình. 10. Chuyên viên y tế lựa chọn tải xuống file kết quả đánh giá 11. Hệ thống ghi lại hoạt động vào trong Activity Log.
Luồng sự kiện phát sinh	
Tiền điều kiện	Chuyên viên y tế đăng nhập thành công.
Hậu điều kiện	Hoạt động được ghi lại trong Activity Log.

Bảng 2.3: Đặc tả use case Đánh giá khả năng cung ứng VTYT

2.3.4 Đặc tả use case Xem tuyến đường hỗ trợ VTYT tối ưu

Tên use case	Xem tuyến đường hỗ trợ VTYT tối ưu
Luồng sự kiện chính	1. Người dùng lựa chọn chức năng Xem tuyến đường hỗ trợ VTYT tối ưu.
	2. Người dùng lựa chọn loại dịch bệnh cần đánh giá.
	3. Hệ thống lấy số liệu về khả năng hỗ trợ VTYT.
	4. Hệ thống lấy dữ liệu về khoảng cách giữa các tỉnh thành.
	5. Hệ thống sử dụng thuật toán tìm đường đi để phân tích các tuyến đường vận chuyển tối ưu.
	6. Hệ thống thông báo phân tích thành công.
	7. Hệ thống hiển thị kết quả đề xuất trên màn hình.
Luồng sự kiện phát sinh	Không có
Tiền điều kiện	Tác nhân thực hiện là người dùng thông thường (chưa đăng nhập)
Hậu điều kiện	Không có

Bảng 2.4: Đặc tả use case Xem tuyến đường hỗ trợ VTYT tối ưu

2.4 Yêu cầu phi chức năng

Hiệu năng (Performance): Website cần có hiệu năng tốt để đảm bảo tải trang nhanh chóng và phản hồi nhanh. Cần xem xét tối ưu hóa thời gian tải trang tối đa, tải lượng người dùng đồng thời, và khả năng mở rộng để đáp ứng nhu cầu tăng trưởng trong tương lai.

Độ tin cậy (Reliability): Cần phải đảm bảo hoạt động liên tục và ổn định, tránh các lỗi hoặc sự cố gây mất dữ liệu hoặc gián đoạn truy cập. Nên xem xét đến các vấn đề sao lưu dữ liệu định kỳ, khôi phục dữ liệu sau sự cố, và kiểm tra lỗi tự động.

Tính dễ sử dụng (Usability): Website cần có giao diện người dùng thân thiện và dễ sử dụng. Các yêu cầu tính dễ sử dụng có thể bao gồm giao diện trực quan, bố cục rõ ràng, hướng dẫn sử dụng, tích hợp công cụ tìm kiếm và hỗ trợ đa ngôn ngữ.

Tính dễ bảo trì (Maintainability): Website cần được thiết kế và triển khai sao cho dễ dàng bảo trì và nâng cấp trong tương lai. Có thể liên quan đến việc sử dụng mã nguồn rõ ràng, tuân thủ các tiêu chuẩn lập trình, sử dụng các công cụ quản lý phiên bản, và có tài liệu hướng dẫn cập nhật.

Yêu cầu kỹ thuật: Website có thể có các yêu cầu kỹ thuật cụ thể như ngôn ngữ lập trình, framework, hệ quản trị cơ sở dữ liệu (CSDL), giao thức mạng, tích hợp với các dịch vụ khác như thanh toán trực tuyến, hỗ trợ đa nền tảng, và tích hợp API.

Bảo mật (Security): Website cần được bảo vệ chống lại các mối đe dọa bảo mật như tấn công hacker, lừa đảo, và mất thông tin cá nhân. Cần xem xét đến việc sử dụng các biện pháp bảo mật như mã hóa dữ liệu, xác thực người dùng, kiểm tra lỗi và cập nhật bảo mật thường xuyên.

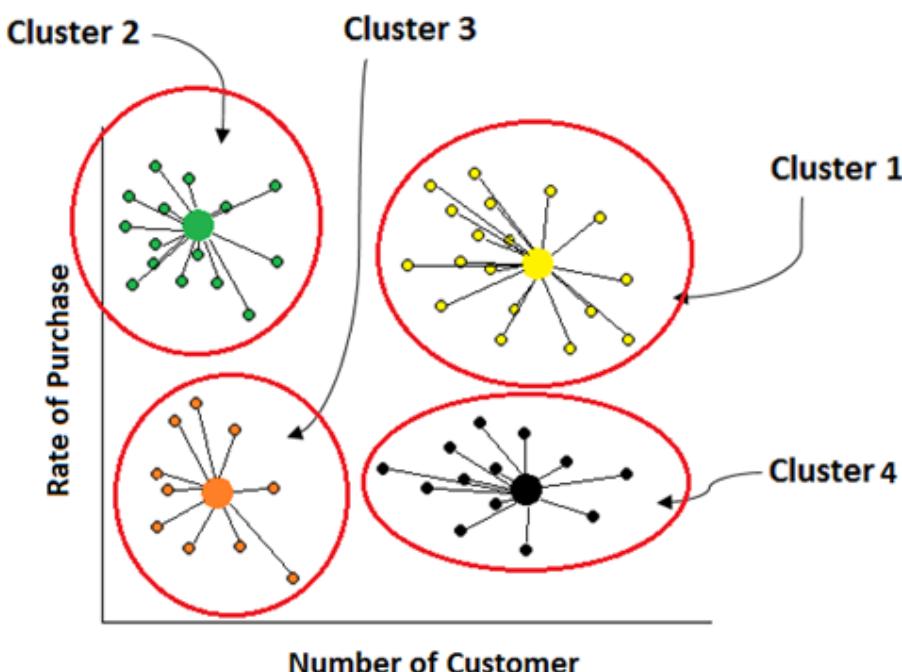
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 Cơ sở lý thuyết

3.1.1 Thuật toán phân cụm mờ bán giám sát với nhiều tham số mờ hóa

Khái niệm phân cụm

Phân cụm dữ liệu là phương pháp phân chia các phần tử của tập dữ liệu thành các cụm sao cho các phần tử dữ liệu trong cùng một cụm có các thuộc tính giống nhau và các phần tử dữ liệu trong các cụm khác nhau có các thuộc tính khác nhau. Phân cụm dữ liệu là bước tiền xử lý quan trọng để tạo ra tri thức ban đầu, hỗ trợ cho việc ra quyết định cho các bước xử lý tiếp theo.



Hình 3.1: Minh họa thuật toán phân cụm

Trong các phương pháp phân cụm hiện nay, thuật toán phân cụm Fuzzy C-mean (FCM) cho kết quả tương đối tốt, tận dụng được tính linh hoạt của logic mờ. Trong FCM, một phần tử dữ liệu có thể linh hoạt chọn một cụm để thuộc về, được đặc trưng bởi cấp thành viên của phần tử trong cụm. Cụ thể, cấp độ thành viên U_{ik} của phần tử i thuộc cụm k có giá trị trong khoảng từ 0 đến 1 và giá trị U_{ik} càng lớn thì phần tử i thuộc cụm k càng có nhiều khả năng.

Thuật toán phân cụm mờ bán giám sát với nhiều tham số mờ hóa

Thuật toán phân cụm mờ bán giám sát với nhiều tham số mờ hóa(SSMC-FCM)[1]

là một thuật toán quan trọng trong đồ án tốt nghiệp này. Được phát triển từ thuật toán phân cụm Fuzzy C-means (FCM), SSMC-FCM mở rộng tính năng của FCM bằng cách đưa thêm các tham số mờ hóa vào quá trình phân cụm, tạo ra một cách tiếp cận mạnh mẽ hơn trong việc phân cụm dữ liệu.

Thuật toán SSMC-FCM kết hợp giữa việc sử dụng các giá trị mờ (fuzzy) và các thông số điều chỉnh được đặt bởi người dùng để tạo ra các phân cụm mờ, linh hoạt và tùy chỉnh cho dữ liệu. Với sự hiện diện của các tham số mờ hóa, SSMC-FCM cho phép mô hình hóa các thông tin không chắc chắn hoặc mờ mà xuất hiện trong dữ liệu.

Cơ chế hoạt động của SSMC-FCM bao gồm các bước chính như sau:

Bước 1. Khởi tạo:

- Thiết lập số lượng phân cụm cần tạo và gán các giá trị ban đầu cho các trung tâm cụm.
- Thiết lập các thông số mờ hóa như hệ số mờ và hệ số học tập.

Bước 2. Tính toán mức độ thuộc và trung tâm cụm:

- Dựa trên các giá trị trung tâm cụm và thông số mờ hóa, tính toán mức độ thuộc của mỗi điểm dữ liệu đến từng cụm.
- Sử dụng mức độ thuộc và thông số mờ hóa, tính toán lại giá trị trung tâm cụm.

Bước 3. Cập nhật thông số mờ hóa:

- Dựa trên mức độ thuộc, tính toán lại các thông số mờ hóa để tăng cường tính linh hoạt và tùy chỉnh của phân cụm.

Bước 4. Lặp lại bước 2 và 3 cho đến khi đạt được sự hội tụ hoặc đạt đến số lượng lặp tối đa được định trước.

Bước 5. Kết quả:

- Sau khi thuật toán hội tụ, thu được các phân cụm mờ với các giá trị trung tâm và mức độ thuộc.
- Các phân cụm này cung cấp thông tin về sự tương đồng và tương phản giữa các điểm dữ liệu trong tập dữ liệu.

Chi tiết thuật toán

$m_{ij} = M$ với mọi $1 \leq j \leq C$, cho những điểm không giám sát i $m_{ik} = M_0$, và

$m_{ij} = M$ với mọi $j! = k$, cho những điểm giám sát i thuộc cụm k Hàm mục tiêu:

$$J(U, V) = N \sum_{i=1}^N \sum_{k=1}^C U_{ik} m_{ik} D_{ik}^2$$

với $X = \{X_1, X_2, \dots, X_N\}$ là tập dữ liệu, N là số phần tử trong tập dữ liệu X , C là số cụm, U_{ik} là hệ số thành viên của phần tử X_i đối với cụm k với tâm cụm là V_k , $0 \leq U_{ik} \leq 1$, $1 \leq i \leq N$, $1 \leq k \leq C$, $\sum_{j=1}^C U_{ik} = 1$, $m_{ik} > 1$ là hệ số mờ của X_i trong cụm k, và $D_{ik}^2 = \|X_i - V_k\|^2$ là khoảng cách giữa 2 vector X_i và V_k . $Y \subset \{1, 2, \dots, N\} \times \{1, 2, \dots, C\}$ là tập các phần tử được giám sát, ta có: $m_{ij} = M, \forall k : (i, k) \in /YM0, (i, j) \in YM, \exists k : (i, k) \in Y \wedge j6 = k$

Đầu vào: - Tập dữ liệu $X = \{X_1, X_2, \dots, X_N\}$ - Các hệ số mờ gồm $M > 1$ và $M' > M$ - Tập các dữ liệu được giám sát $Y \subset \{1, 2, \dots, N\} \times \{1, 2, \dots, C\}$. Đầu ra: Kết quả phân cụm tập dữ liệu X thành C cụm - Bước 1: Khởi tạo giá trị cho các tâm cụm V , cho $I = 0$ và $\varepsilon > 0$. - Bước 2: Tại vòng lặp thứ I, cập nhật ma trận hệ số U Đối với các phần tử không giám sát, sử dụng công thức bên dưới để tính U_{ik}

$$U_{ik} = \frac{1}{\sum_{j=1}^C \left(\frac{D_{ik}}{D_{ij}} \right)^{\frac{2}{M-1}}} = \left(\sum_{j=1}^C \left(\frac{D_{ik}}{D_{ij}} \right)^{\frac{2}{M-1}} \right)^{-1}$$

Đối với các phần tử được giám sát, thực hiện các tính toán sau: Tính d_{ij} theo công thức:

$$d_{ij} = \frac{D_{ij}}{d_{\min}}, \forall j = 1, \dots, C$$

trong đó: - D_{ij} là khoảng cách từ điểm đang xét tới các tâm cụm xung quanh - $d_{\min} = \min \{D_{ij}\} (j = 1, \dots, C)$ Tính μ_{ij}

$$\mu_{ij} = \left(\frac{1}{Md_{ij}^2} \right)^{\frac{1}{M'-1}}$$

Tính μ_{ik}

$$\frac{\mu_{ik}}{\left(\mu_{ik} + \sum_{j=1, j \neq k}^C \mu_{ij} \right)^{\frac{M'-M}{M'-1}}} = \left(\frac{1}{M'd_{ik}^2} \right)^{\frac{1}{M'-1}}$$

Sau đó tính U_{ij} theo công thức:

$$U_{ij} = \frac{\mu_{ij}}{\sum_{l=1}^C \mu_{il}}$$

Bước 3: Cập nhật giá trị các tâm cụm V cho vòng lặp thứ $(I + 1)$, dựa theo công thức số (10), với m_{ik} được tính từ công thức số (8).

$$V_k = \frac{\sum_{i=1}^N U_{ik}^{m_{ik}} X_i}{\sum_{i=1}^N U_{ik} m_{ik}}$$

Trong đó, m_{ik} được tính theo công thức:

Bước 4: Nếu $\|V(I) - V(I + 1)\| < \varepsilon$, tiếp tục thực hiện bước 5 . Nếu không, gán $I := I + 1$ và quay trở lại thực hiện bước 2 . Bước 5: Kết thúc.

3.1.2 Giải thuật Dijkstra

Trong đồ án tốt nghiệp này, một trong những giải thuật quan trọng được áp dụng là giải thuật Dijkstra[2] để tối ưu hóa các tuyến đường vận chuyển vật tư y tế (VTYT) giữa các tỉnh thành.

Giải thuật Dijkstra là một thuật toán tìm kiếm đường đi ngắn nhất trong đồ thị có trọng số. Được đặt theo tên của nhà toán học Edsger W. Dijkstra, giải thuật này đã được áp dụng rộng rãi trong nhiều lĩnh vực, từ quản lý mạng đến giao thông vận tải.

Với mục tiêu tối ưu hóa quãng đường vận chuyển VTYT giữa các tỉnh thành, giải thuật Dijkstra sẽ xác định đường đi ngắn nhất từ một đỉnh xuất phát đến tất cả các đỉnh còn lại trong đồ thị. Thuật toán hoạt động dựa trên nguyên tắc "mở rộng" từng bước để tìm kiếm đường đi ngắn nhất cho mỗi đỉnh, cho đến khi tất cả các đỉnh đã được duyệt.

Dưới đây là một mô tả chi tiết về cách giải thuật Dijkstra hoạt động trong việc tối ưu hóa tuyến đường vận chuyển VTYT giữa các tỉnh thành.

Bước 1. Khởi tạo:

- Đặt đỉnh xuất phát là đỉnh gốc, gán khoảng cách từ đỉnh gốc tới chính nó bằng 0.
- Gán tất cả các khoảng cách từ đỉnh gốc tới các đỉnh còn lại là vô cùng hoặc một giá trị lớn.

Bước 2. Duyệt qua các đỉnh:

- Với mỗi đỉnh chưa được duyệt, tìm đỉnh có khoảng cách nhỏ nhất từ đỉnh gốc tới nó. Đỉnh này sẽ được chọn để tiếp tục xét và cập nhật khoảng cách cho các đỉnh kề nó.
- Duyệt qua tất cả các đỉnh kề của đỉnh đang xét.

- Tính toán khoảng cách mới từ đỉnh gốc tới các đỉnh kề thông qua đỉnh đang xét. Nếu khoảng cách mới này nhỏ hơn khoảng cách hiện tại, cập nhật khoảng cách và đỉnh trước đó của đỉnh kề.

Bước 3. Lặp lại bước 2 cho đến khi tất cả các đỉnh đã được duyệt hoặc tìm thấy đường đi ngắn nhất tới đích.

- Tiếp tục duyệt qua các đỉnh chưa được xét và lặp lại bước 2.
- Khi tất cả các đỉnh đã được duyệt hoặc đã tìm thấy đường đi ngắn nhất tới đích, thuật toán kết thúc.

Bước 4. Xây dựng đường đi ngắn nhất:

- Dựa trên thông tin về khoảng cách và đỉnh trước đó, xây dựng đường đi ngắn nhất từ đỉnh gốc tới mỗi đỉnh trong đồ thị.
- Đường đi này có thể được lưu trữ dưới dạng một danh sách các đỉnh theo thứ tự từ đỉnh gốc tới đích.

Giải thuật Dijkstra sử dụng cơ chế "mở rộng" từng bước để duyệt qua các đỉnh và cập nhật khoảng cách tới các đỉnh kề. Nhờ vào việc lựa chọn đỉnh có khoảng cách nhỏ nhất, giải thuật Dijkstra đảm bảo tìm ra đường đi ngắn nhất từ đỉnh xuất phát tới các đỉnh còn lại. Trong đồ án tốt nghiệp, giải thuật Dijkstra được áp dụng để tìm tuyến đường vận chuyển VTYT tối ưu giữa các tỉnh thành, giúp cải thiện hiệu quả và tiết kiệm thời gian trong việc phân phối và vận chuyển vật tư y tế giữa các địa điểm khác nhau.

3.2 Nền tảng công nghệ

3.2.1 Hệ điều hành

Hệ thống sẽ được xây dựng trên hệ điều hành windows vì những lý do sau đây:

- Hỗ trợ nhiều ngôn ngữ lập trình: Windows hỗ trợ nhiều ngôn ngữ lập trình phổ biến như C++, Java, Python, Ruby và nhiều ngôn ngữ khác. Công cụ phát triển tích hợp (IDE) phong phú: Windows có nhiều công cụ phát triển tích hợp (IDE) phong phú và mạnh mẽ để lập trình viên lựa chọn, bao gồm Visual Studio, Eclipse, NetBeans, IntelliJ IDEA, và PyCharm. Những công cụ này giúp lập trình viên dễ dàng quản lý, xây dựng, và kiểm thử ứng dụng của mình. Trong đề tài này, em sẽ sử dụng Visual Studio Code (VSC) để thực hiện phát triển hệ thống.
- Tích hợp thư viện phong phú: Windows tích hợp nhiều thư viện lập trình phổ biến và phong phú để giúp lập trình viên xây dựng ứng dụng một cách nhanh chóng và dễ dàng hơn. Các thư viện này bao gồm .NET Framework, DirectX,

Windows API, và nhiều thư viện khác.

- Sử dụng rộng rãi và cộng đồng phát triển lớn: Windows là hệ điều hành phổ biến nhất trên thế giới, do đó có một cộng đồng phát triển lớn và đa dạng. Nhờ đó, lập trình viên có thể dễ dàng tìm kiếm các tài liệu, thư viện và giải pháp để giải quyết các vấn đề trong quá trình lập trình.

3.2.2 Hệ quản trị cơ sở dữ liệu

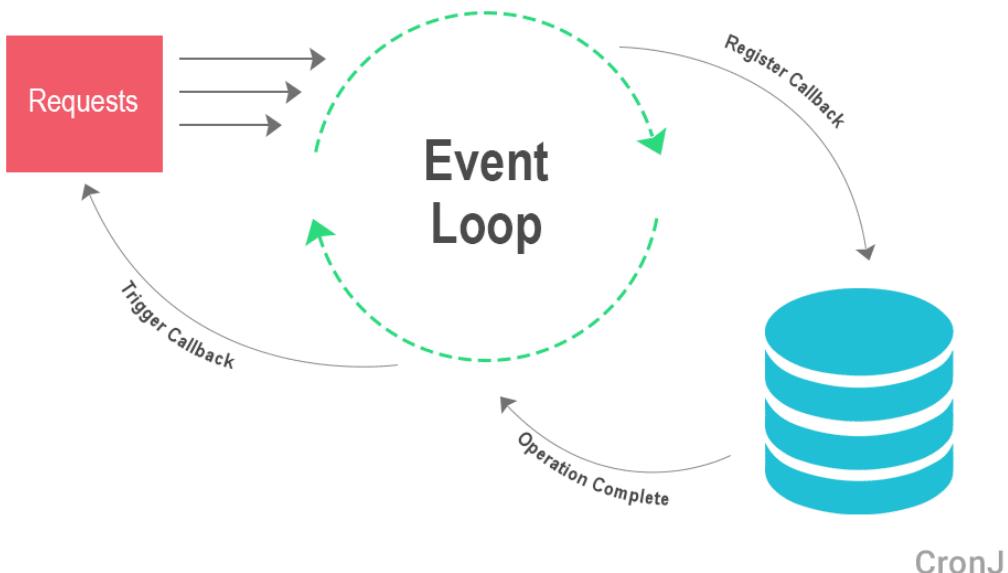
PostgreSQL là một hệ quản trị cơ sở dữ liệu mạnh mẽ được lựa chọn để quản lý cơ sở dữ liệu của hệ thống. Lý do chọn PostgreSQL là vì nó có những ưu điểm nổi trội sau:

- Độ tin cậy cao: PostgreSQL được thiết kế với mục tiêu đảm bảo tính toàn vẹn và độ tin cậy cao của dữ liệu. Hệ thống sẽ có khả năng duyệt qua các lỗi, phục hồi dữ liệu và giữ cho hệ thống ổn định ngay cả trong các tình huống khẩn cấp.
- Hiệu suất tốt: PostgreSQL có khả năng xử lý lượng lớn dữ liệu và thực thi các truy vấn phức tạp một cách hiệu quả. Cơ sở dữ liệu này hỗ trợ các chức năng tối ưu hóa truy vấn, bao gồm chỉ mục hóa, tối ưu hóa câu lệnh SQL và thông kê, giúp tăng cường hiệu suất và tối ưu hóa thời gian truy vấn.
- Hỗ trợ đa nền tảng: PostgreSQL có khả năng chạy trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS. Điều này giúp cho việc triển khai và sử dụng cơ sở dữ liệu trên các nền tảng khác nhau trở nên linh hoạt và thuận tiện.
- Hỗ trợ đa ngôn ngữ: PostgreSQL hỗ trợ nhiều ngôn ngữ lập trình, bao gồm C/C++, Java, Python, Ruby và Perl. Điều này cho phép lập trình viên có thể sử dụng ngôn ngữ lập trình ưa thích của họ để truy cập và làm việc với cơ sở dữ liệu một cách thuận tiện.
- Mã nguồn mở: PostgreSQL là một phần mềm mã nguồn mở và miễn phí, đồng nghĩa với việc lập trình viên có thể tùy chỉnh và sử dụng PostgreSQL một cách tự do. Đồng thời, cộng đồng người dùng đông đảo của PostgreSQL đóng góp và phát triển các tính năng mới, giúp cải thiện và mở rộng khả năng của cơ sở dữ liệu này.

3.2.3 Môi trường lập trình

Bởi vì sẽ phải giải quyết các bài toán phân cụm và bài toán tìm đường đi ngắn nhất với số lượng lớn dữ liệu đầu vào, chúng ta cần phải có một nền tảng có hiệu suất xử lý tốt, có thể xử lý hàng loạt những phép tính phức tạp trong thời gian ngắn, đồng thời có thể thực hiện các xử lý bất đồng bộ để tăng tốc độ tính toán. Từ những

yêu cầu đó, em đã lựa chọn Nodejs cho hệ thống cần xây dựng. Ngoài những ưu điểm trên, Nodejs còn có hỗ trợ non-blocking I/O giúp ứng dụng xử lý các yêu cầu đồng thời mà không ảnh hưởng đến hiệu suất.



Hình 3.2: Non-blocking I/O trong Nodejs

3.2.4 Framework và API hỗ trợ

a, Express.js

Express.js[3] là một framework mã nguồn mở được xây dựng trên ngôn ngữ JavaScript và chạy trên môi trường Node.js. Nó được sử dụng để xây dựng các ứng dụng web phía server, đặc biệt là các ứng dụng web RESTful và các ứng dụng web đơn giản, nhanh chóng và hiệu quả. Express.js giúp đơn giản hóa việc xây dựng server và xử lý các yêu cầu HTTP của ứng dụng.

Một số điểm nổi bật và đặc điểm của Express.js bao gồm:

- Minimalist và Unopinionated: Express.js được thiết kế nhằm tối giản hóa việc phát triển ứng dụng web, không áp đặt quá nhiều quy tắc và cấu trúc. Điều này cho phép lập trình viên tự do chọn cách tổ chức mã và sử dụng các plugin hoặc middleware theo ý muốn.
- Middleware: Express.js sử dụng các middleware để mở rộng chức năng của ứng dụng. Middleware là các hàm được thực hiện liên tục khi một yêu cầu HTTP được gửi đến server, cho phép thêm các tính năng, kiểm tra dữ liệu và thực hiện các hành động trước khi xử lý yêu cầu chính.

- Routing: Express.js cung cấp cơ chế routing linh hoạt cho phép định nghĩa các đường dẫn URL và các xử lý tương ứng cho mỗi đường dẫn. Điều này giúp quản lý mã nguồn một cách dễ dàng và giảm thiểu việc lặp lại mã.
- Công cụ xây dựng API RESTful: Express.js rất phù hợp để xây dựng các API RESTful với việc hỗ trợ các phương thức HTTP như GET, POST, PUT, DELETE, vv. Việc này giúp dễ dàng tạo các dịch vụ API cho các ứng dụng di động và web khác nhau.
- Xử lý đa luồng: Express.js hỗ trợ xử lý đa luồng thông qua các hàm middleware và cơ chế bất đồng bộ của Node.js. Điều này cho phép ứng dụng xử lý nhiều yêu cầu cùng một lúc mà không làm giảm hiệu suất.

Tóm lại, Express.js là một framework mạnh mẽ và linh hoạt để xây dựng các ứng dụng web phía server. Điểm mạnh của nó nằm ở tính đơn giản, sự linh hoạt, khả năng mở rộng thông qua middleware và hỗ trợ xây dựng các API RESTful. Điều này đã khiến Express.js trở thành một trong những lựa chọn hàng đầu cho việc xây dựng các ứng dụng web bằng Node.js.

b, React

React.js[4] là một thư viện mã nguồn mở được phát triển bởi Facebook. Nó được sử dụng để xây dựng giao diện người dùng (UI) cho các ứng dụng web phía client. React.js đơn giản, hiệu quả và mạnh mẽ, và nhanh chóng trở thành một công nghệ phổ biến trong cộng đồng lập trình web.

Một số điểm nổi bật và đặc điểm của React.js bao gồm:

- Components: React.js dựa trên khái niệm "components" (thành phần), cho phép bạn chia giao diện người dùng thành các phần nhỏ và độc lập có thể tái sử dụng. Điều này giúp quản lý mã và tạo giao diện phức tạp một cách dễ dàng và hiệu quả.
- Virtual DOM: React.js sử dụng một khái niệm gọi là "Virtual DOM" (cây DOM ảo) để cải thiện hiệu suất. Thay vì thao tác trực tiếp với DOM (Document Object Model), React sẽ tạo một bản sao ảo của DOM và so sánh với DOM thật để chỉ cập nhật các phần cần thiết, giúp tối ưu hóa hiệu năng của ứng dụng.
- One-way Data Binding: React.js triển khai one-way data binding (ràng buộc dữ liệu một chiều) giữa dữ liệu và giao diện người dùng. Điều này đảm bảo rằng bất kỳ thay đổi nào trong dữ liệu sẽ tự động cập nhật và hiển thị trên giao diện mà không cần phải làm lại việc cập nhật thủ công.
- JSX (JavaScript XML): React.js cho phép viết mã HTML bên trong mã JavaScript

thông qua cú pháp JSX. JSX giúp mã trở nên dễ đọc, dễ viết và dễ hiểu, đồng thời cũng cung cấp các tính năng của JavaScript như biến thức, hàm và lệnh điều kiện.

- Cộng đồng phong phú: React.js có một cộng đồng rộng lớn và tích cực, với rất nhiều tài liệu, tài nguyên và các thư viện mở rộng hữu ích. Điều này giúp hỗ trợ và nâng cao khả năng phát triển ứng dụng React.
- React Native: Ngoài việc phát triển các ứng dụng web, React.js còn được sử dụng trong React Native để xây dựng ứng dụng di động đa nền tảng (cross-platform). Chia sẻ mã giữa ứng dụng web và ứng dụng di động giúp giảm thiểu thời gian và công sức phát triển.

Tóm lại, React.js là một thư viện JavaScript mạnh mẽ và hiệu quả để xây dựng giao diện người dùng tương tác và phức tạp. Sự hỗ trợ của nó với các components, Virtual DOM, JSX và cộng đồng lớn đã khiến nó trở thành một trong những công nghệ hàng đầu cho việc phát triển các ứng dụng web và di động hiện đại.

c, Axios

Axios[5] là một thư viện HTTP client mã nguồn mở được sử dụng phổ biến trong các ứng dụng web phía client để gửi các yêu cầu HTTP đến máy chủ và xử lý các phản hồi từ máy chủ. Nó được viết bằng JavaScript và hỗ trợ cả trình duyệt và môi trường Node.js, cho phép tương tác với các API web và dữ liệu từ các nguồn khác nhau.

Một số điểm mạnh của Axios là:

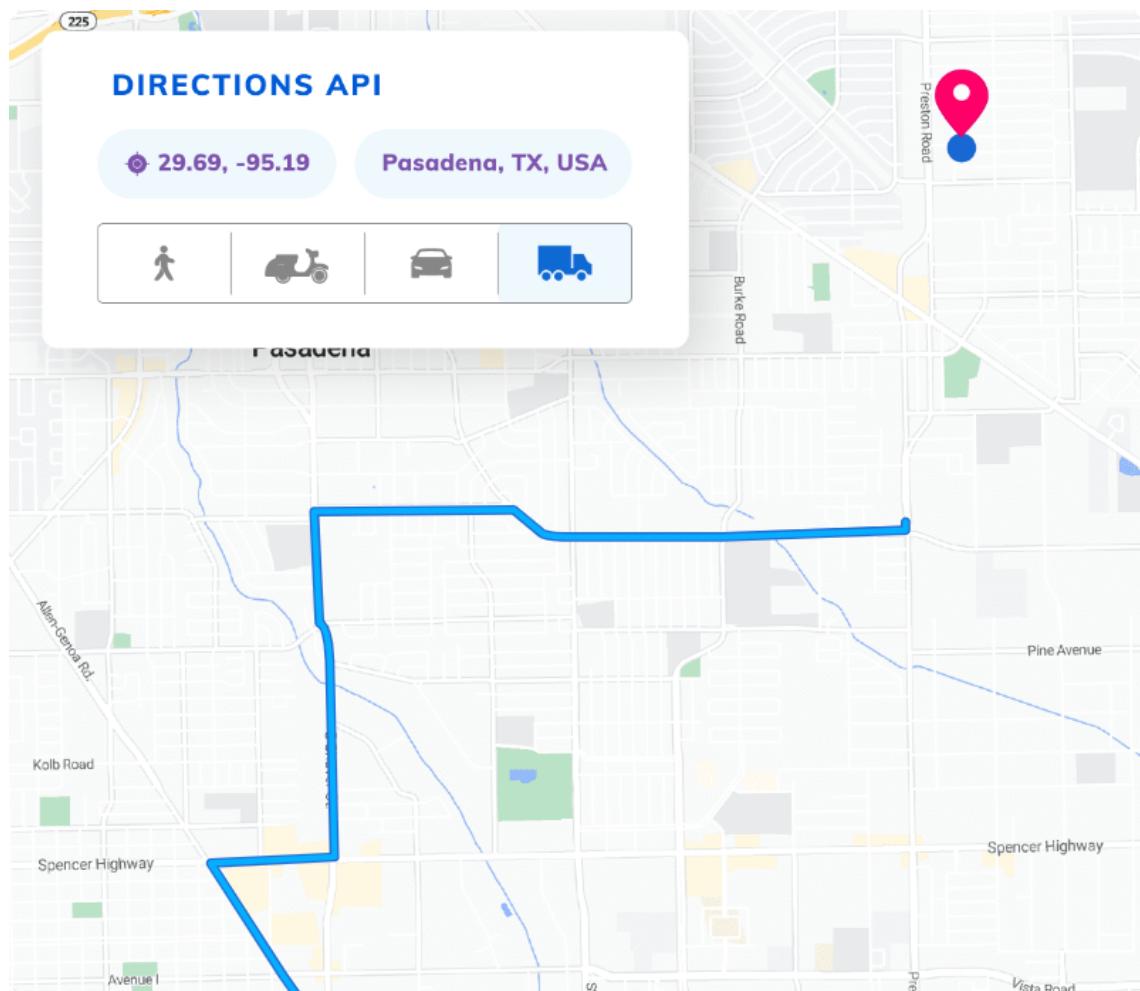
- Dễ sử dụng: Axios cung cấp một cách tiếp cận dễ dàng để thực hiện các yêu cầu HTTP thông qua các phương thức đơn giản như GET, POST, PUT, DELETE, vv. Điều này giúp giảm thiểu việc phải viết mã phức tạp để giao tiếp với API.
- Hỗ trợ promise: Axios sử dụng các promise để xử lý các yêu cầu và phản hồi. Điều này cho phép sử dụng các hàm then() và catch() để xử lý dữ liệu trả về hoặc xử lý lỗi một cách dễ dàng và sáng sủa hơn.
- Hỗ trợ chuyển tiếp và hủy bỏ yêu cầu: Axios cho phép bạn thực hiện chuyển tiếp yêu cầu và hủy bỏ yêu cầu một cách dễ dàng bằng cách sử dụng các token hủy bỏ. Điều này giúp giảm thiểu tình trạng yêu cầu trùng lặp và quản lý tốt hơn các yêu cầu đang chờ.
- Xử lý lỗi dễ dàng: Axios hỗ trợ xử lý lỗi một cách linh hoạt thông qua việc cài đặt các hàm xử lý lỗi (error handlers). Điều này giúp giảm thiểu các lỗi không mong muốn và làm cho việc gỡ lỗi dễ dàng hơn.

- Hỗ trợ giao diện gọi API đồng bộ và bất đồng bộ: Axios cho phép bạn thực hiện các yêu cầu đồng bộ hoặc bất đồng bộ, giúp tăng cường hiệu suất và tùy chỉnh cách ứng dụng của bạn tương tác với API.

Tóm lại, Axios là một công cụ mạnh mẽ và linh hoạt để giao tiếp với các API web. Sự dễ sử dụng, hỗ trợ promise và các tính năng linh hoạt khác khiến nó trở thành một lựa chọn phổ biến cho việc thực hiện các yêu cầu HTTP trong các ứng dụng web.

d, Traffic-Aware Directions API

Khi xây dựng một dự án liên quan đến định vị và định tuyến, việc sử dụng một API hỗ trợ bản đồ là một yếu tố quan trọng. Trên thị trường hiện nay, có nhiều API định tuyến khác nhau, nhưng trong đồ án này, tôi sẽ sử dụng Traffic-Aware Directions API của nextbillon.ai. Đây là một API định tuyến thông minh và hiệu quả, mang lại nhiều lợi ích cho dự án của mình.



Hình 3.3: API Traffic-Aware Directions của nextbillon.ai

API này được phát triển dựa trên các dữ liệu thời gian thực và thu thập thông tin giao thông trực tiếp từ các nguồn địa phương và toàn cầu. Do đó, API cung cấp

thông tin định tuyến chính xác và chính xác hơn so với các API khác chỉ dựa trên dữ liệu địa lý cũ.

Traffic-Aware Directions API được phát triển trên nền tảng công nghệ đám mây, cho phép tích hợp dễ dàng vào các ứng dụng khác nhau và mở rộng theo quy mô cần thiết. API cũng cung cấp tính năng đáp ứng tải cao, cho phép xử lý hàng ngàn yêu cầu định tuyến đồng thời mà không làm giảm hiệu suất.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1 Thiết kế kiến trúc

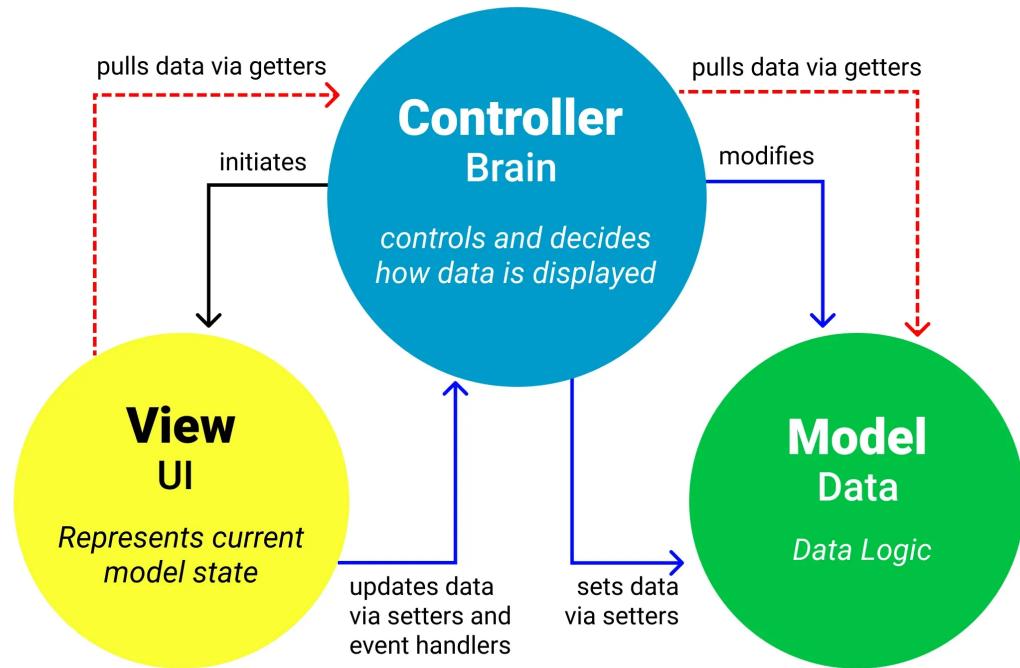
4.1.1 Lựa chọn kiến trúc phần mềm

Tổng quan mô hình MVC

MVC (Model-View-Controller) là một mô hình thiết kế phần mềm được sử dụng để tạo lập giao diện người dùng trên máy tính. Nó chia ứng dụng thành ba phần chính, mỗi phần có một nhiệm vụ riêng và độc lập với các phần khác:

- Model: Đây là bộ phận lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa hai thành phần khác là View và Controller. Model biểu diễn các đối tượng dữ liệu trong ứng dụng và quản lý các thao tác liên quan đến dữ liệu, chẳng hạn như lấy, cập nhật và xóa. Nó không biết gì về giao diện người dùng và thường thông báo cho Controller khi có sự thay đổi trong dữ liệu.
- View: Đây là phần giao diện dành cho người sử dụng. View chịu trách nhiệm hiển thị dữ liệu từ Model theo cách thích hợp cho người dùng. Nó có nhiệm vụ tạo ra các thành phần giao diện, như cửa sổ, nút, hoặc văn bản, để người dùng có thể tương tác với ứng dụng. View không nắm bắt kỳ thông tin về dữ liệu hay logic xử lý, chỉ đơn thuần là hiển thị thông tin mà nó nhận được từ Model.
- Controller: Là bộ phận có nhiệm vụ xử lý các yêu cầu của người dùng thông qua View. Nó làm trung gian giữa Model và View, nhận lệnh từ View và thực hiện các hành động cần thiết trên Model tương ứng. Nó có thể cập nhật dữ liệu trong Model hoặc yêu cầu View hiển thị lại thông tin mới nhất. Controller đảm bảo rằng các thành phần trong ứng dụng hoạt động chính xác và đồng bộ.

MVC Architecture Pattern

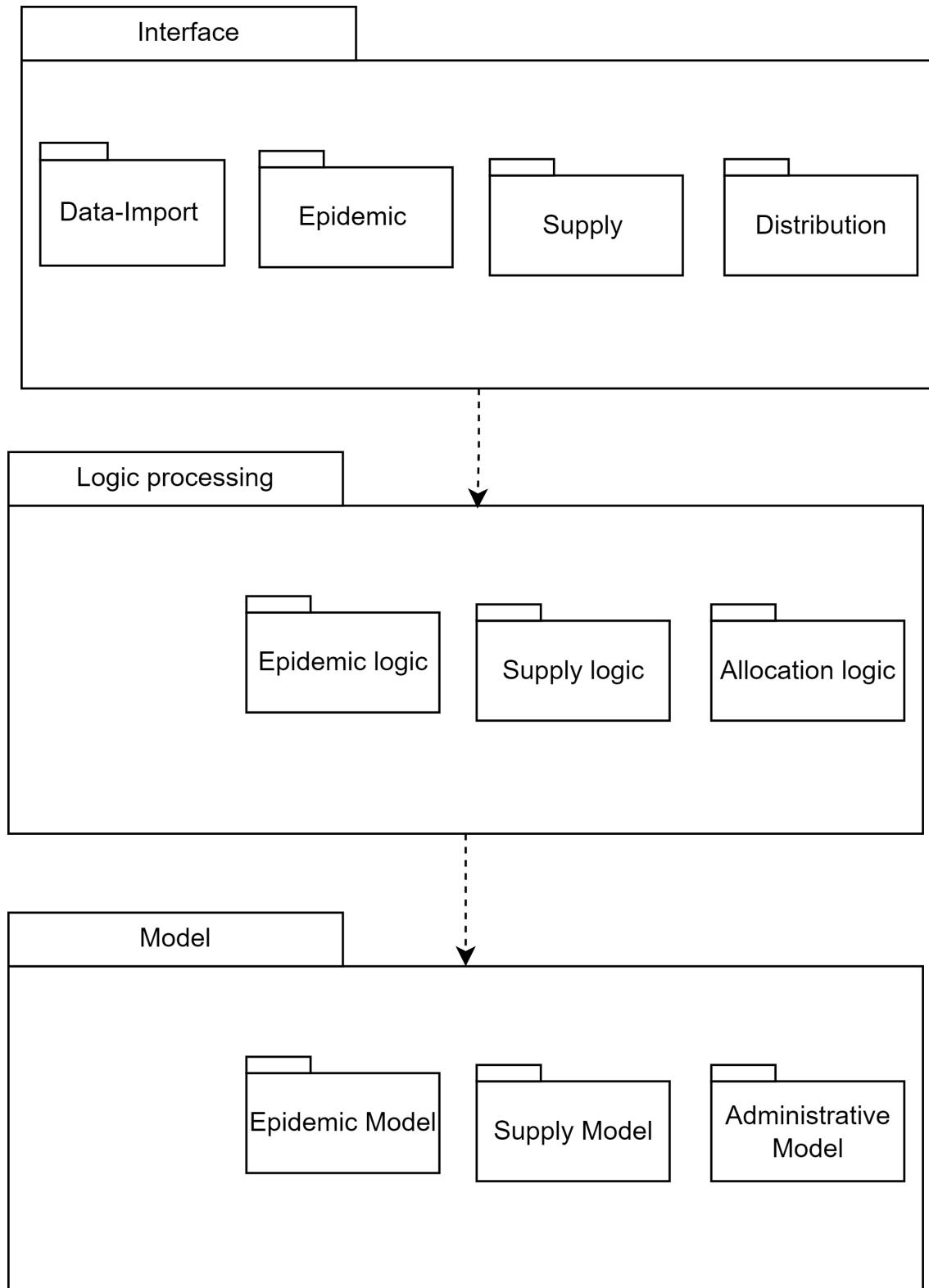


Hình 4.1: Thành phần của mô hình MVC

Ưu điểm của mô hình MVC

- Kiểm tra dễ dàng: Các thành phần độc lập giúp người lập trình dễ kiểm soát và khắc phục các vấn đề, lỗi phát sinh trước khi hoàn thiện sản phẩm đến người dùng.
- Chức năng control: Khi kết hợp với các loại ngôn ngữ lập trình thông dụng như CSS<HTML, Javascript thì mô hình MVC là sự hỗ trợ đóng vai trò tối ưu bộ control trên nền tảng ngôn ngữ lập trình.
- View và size: MVC giúp tối ưu diện tích băng thông khi sử dụng tránh trường hợp khi nhiều yêu cầu được thực hiện cùng lúc sẽ tạo ra nhiều tệp với dung lượng lớn ảnh hưởng trực tiếp đến đường truyền mạng.
- Tính kết hợp: Người lập trình có thể kết hợp mô hình MVC trên nhiều nền tảng website/ ứng dụng khác nhau giúp tiện lợi hơn khi viết code và giảm tải dung lượng.
- Kết cấu khá đơn giản: Phù hợp cho nhiều đối tượng sử dụng khi có nhu cầu lập trình website hoặc các loại ứng dụng.

4.1.2 Thiết kế tổng quan



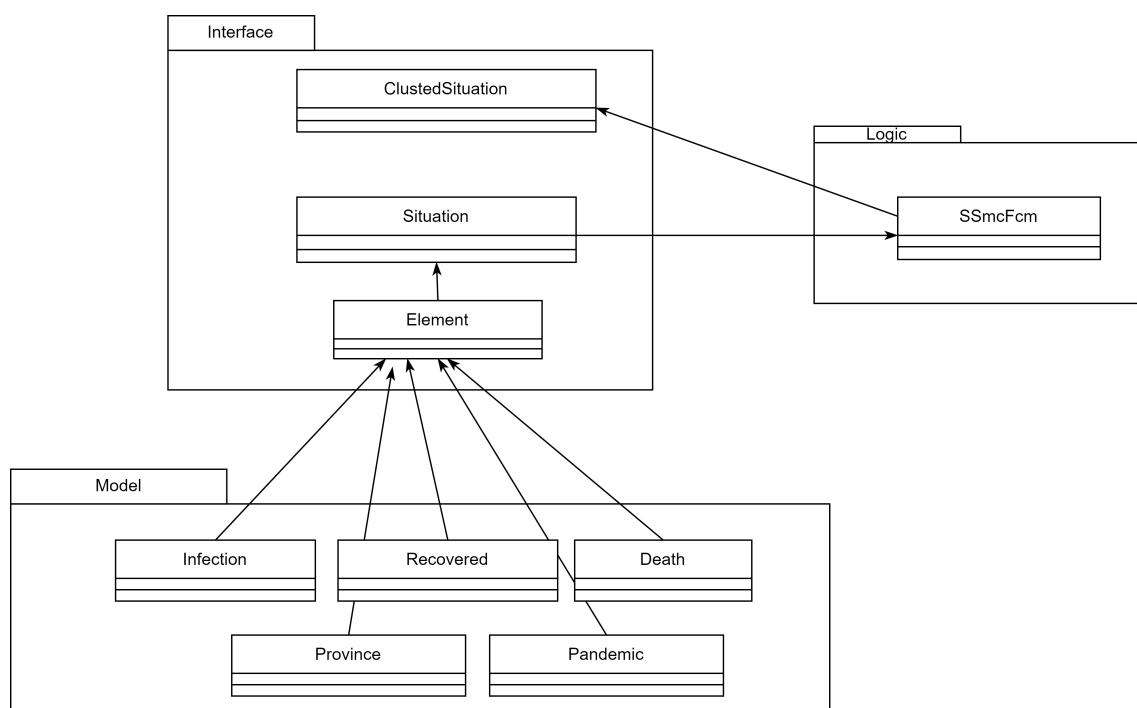
Hình 4.2: Biểu đồ gói UML

Biểu đồ gói UML trên gồm 3 gói tổng quát: Interface (chứa các class đảm nhiệm việc giao tiếp với giao diện người dùng), Logic processing (chứa các gói đảm nhiệm xử lý logic), Model (chứa các gói model nhỏ hơn).

Khi người dùng thao tác với UI, các thành phần của gói Interface sẽ tiếp nhận xử lý, những thao tác đó sẽ được gửi về Logic processing để tiếp nhận. Model sẽ cung cấp những data tới Logic processing để thực hiện tính toán. Kết quả thay đổi sẽ được trả về Interface để hiển thị lại cho người dùng.

4.1.3 Thiết kế chi tiết gói

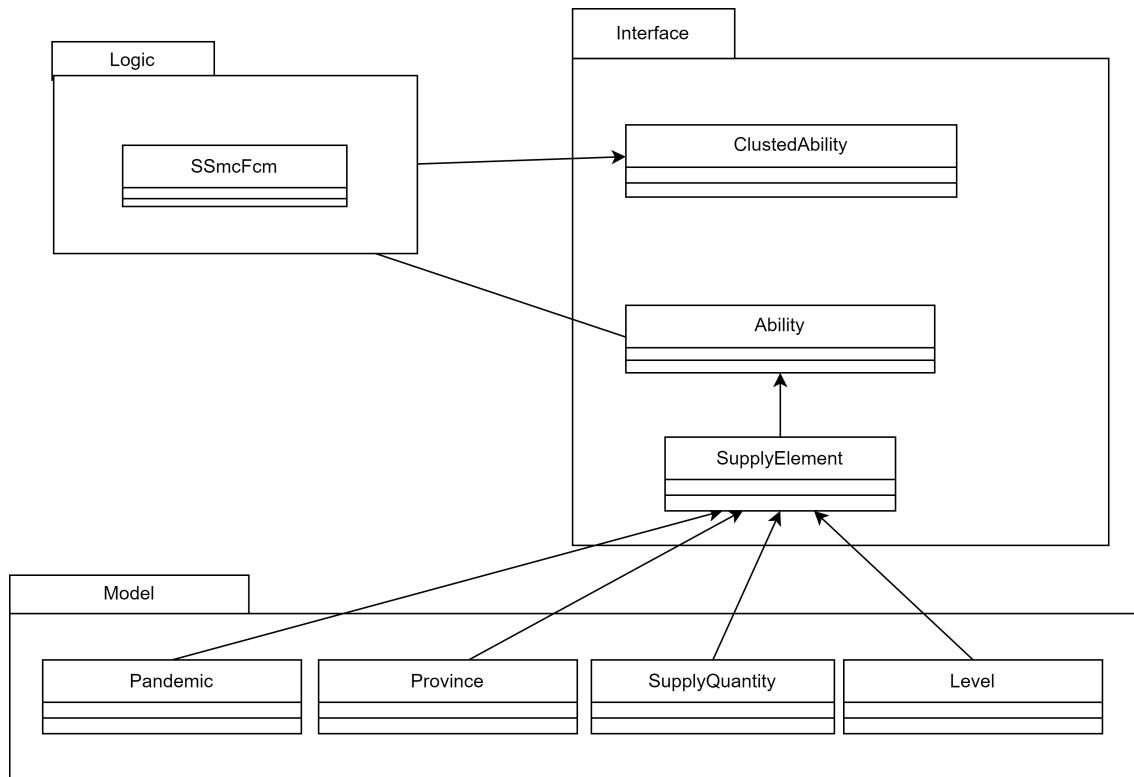
Biểu đồ chi tiết gói cho chức năng Phân tích tình hình dịch bệnh: Chi tiết các lớp cho chức năng này được trình bày trong phần a



Hình 4.3: Biểu đồ chi tiết gói cho chức năng Phân tích tình hình dịch bệnh

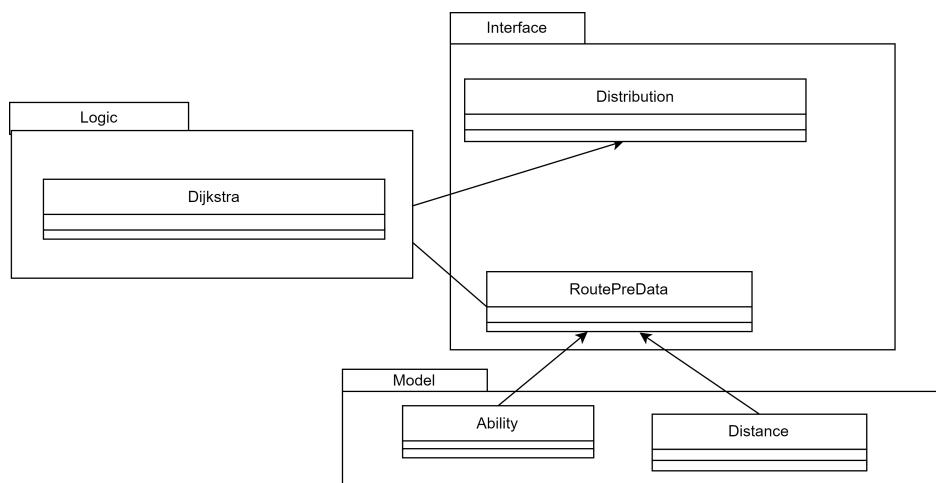
Biểu đồ chi tiết gói cho chức năng Phân tích khả năng hỗ trợ VTYT: Chi tiết các lớp cho chức năng này được trình bày trong phần b

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ



Hình 4.4: Biểu đồ chi tiết gói cho chức năng Phân tích khả năng hỗ trợ VTYT

Biểu đồ chi tiết gói cho chức năng Xem đề xuất phân bổ VTYT: Chi tiết các lớp cho chức năng này được trình bày trong phần c



Hình 4.5: Biểu đồ chi tiết gói cho chức năng Xem đề xuất phân bổ VTYT

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Đặc tả thông tin về màn hình mà ứng dụng của ứng dụng hướng tới như sau:

- **Độ phân giải màn hình:** 1920 x 1080 pixels.
- **Kích thước màn hình:** 19-24 inch.
- **Số lượng màu sắc hỗ trợ:** 16 triệu màu.
- **Tỷ lệ khung hình:** 16:9.

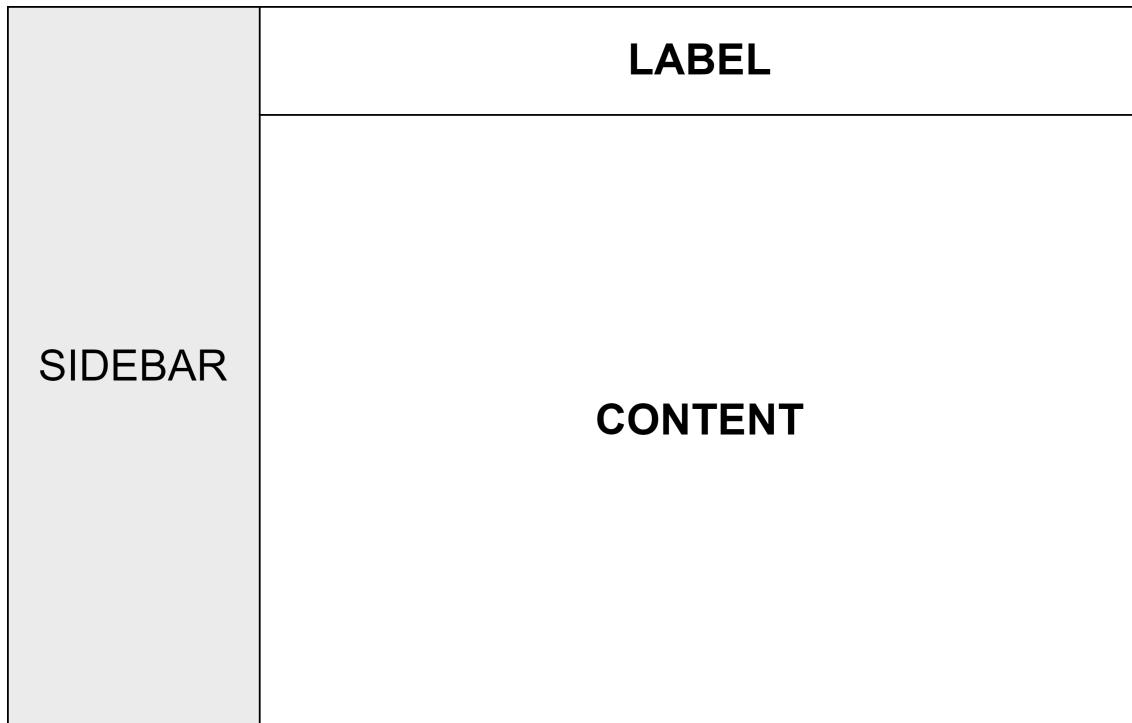
Các thông nhất/chuẩn hóa được đưa ra khi thiết kế giao diện ứng dụng như sau:

- **Thiết kế nút:** Sử dụng nút bấm có hình vuông và các nút có kích thước tương đồng nhau, tạo sự nhất quán trong cả giao diện.
- **Thiết kế điều khiển:** Đặt điều khiển và các tùy chọn quan trọng ở vị trí dễ nhìn và dễ tiếp cận, giúp người dùng dễ dàng tương tác với ứng dụng.
- **Vị trí hiển thị thông điệp phản hồi:** Hiển thị thông điệp phản hồi ngay trên màn hình chính, gần nút hoặc điều khiển liên quan để người dùng nhận biết kết quả hành động của họ.
- **Phối màu:** Sử dụng một bảng màu thống nhất và hài hòa, như sử dụng các màu chủ đạo là xanh dương và trắng để tạo cảm giác sáng sủa và dễ nhìn.

Với các thông nhất và chuẩn hóa này, mục đích của ứng dụng là sẽ mang lại trải nghiệm người dùng tốt và dễ sử dụng, giúp người dùng trải nghiệm sản phẩm một cách dễ dàng.

Một số hình ảnh minh họa thiết kế giao diện:

Minh họa Layout tổng quát của trang web:

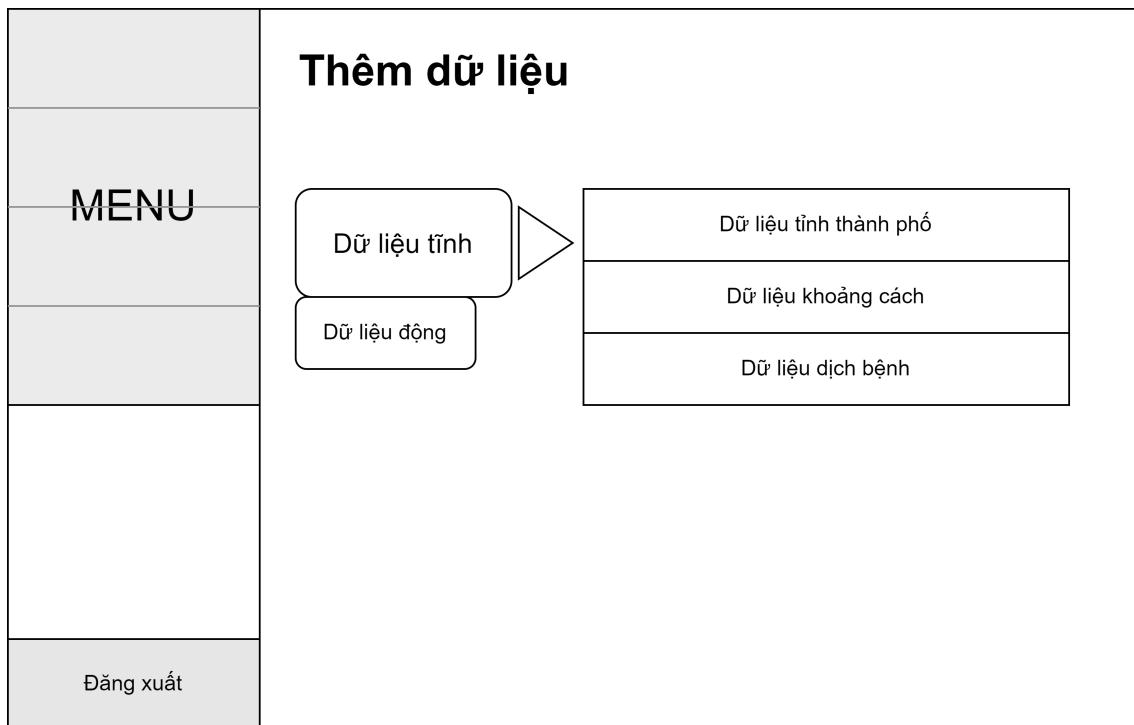


Hình 4.6: Minh họa Layout tổng quát của trang web

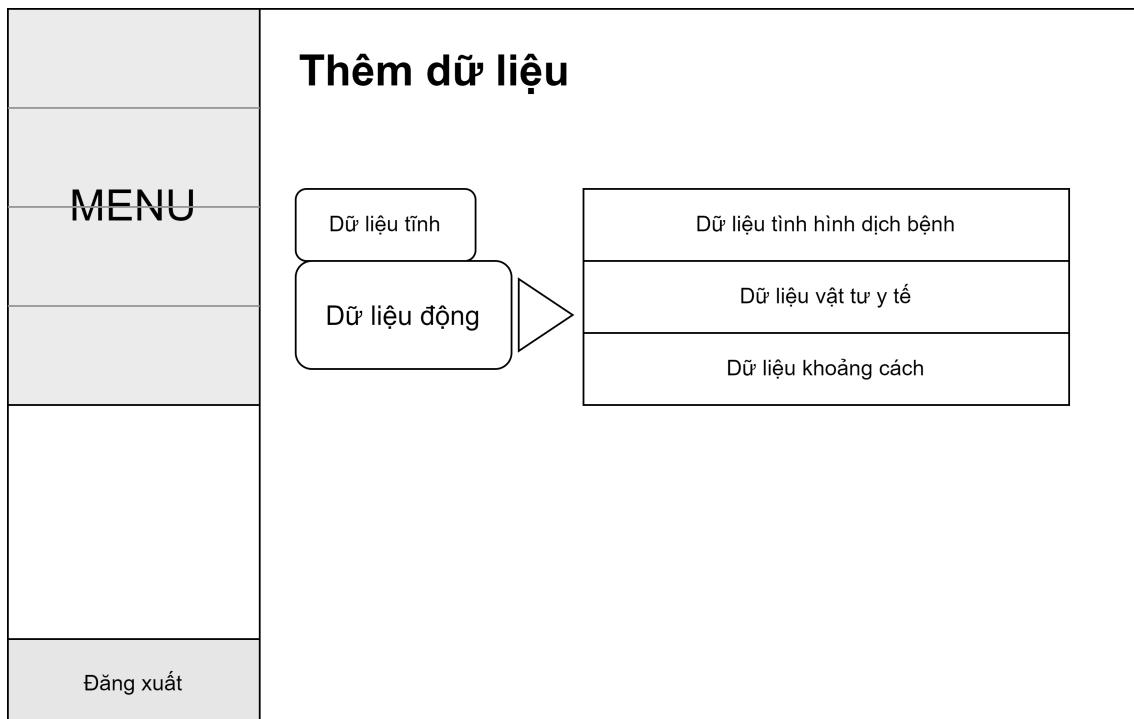
Tổng quan thiết kế trang web sẽ gồm 3 phần: Sidebar nằm ở bên trái là nơi chứa Menu giúp chuyển giữa các chức năng của ứng dụng. Ngoài ra đây cũng là nơi hiển thị trạng thái đăng nhập của người dùng. Bên phải màn hình là Label (Tên của chức năng) và Content (Nội dung hiển thị của chức năng đó).

Có thể thấy thiết kế tổng quan của trang web rất đơn giản. Điều này giúp cho những người mới tiếp cận sử dụng ứng dụng sẽ nhanh chóng thích nghi được với sản phẩm này. Việc không khiến cho layout của trang web phức tạp giúp cho sản phẩm trở nên thân thiện hơn rất nhiều cho người dùng cuối.

Màn hình chức năng Thêm dữ liệu:



Hình 4.7: Minh họa màn hình chức năng Thêm dữ liệu tĩnh



Hình 4.8: Minh họa màn hình chức năng Thêm dữ liệu động

Màn hình chức năng Xem số liệu dịch bệnh:

<p>MENU</p> <p>Đăng nhập</p>	<p>Xem tình hình dịch bệnh</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Loại dịch bệnh ▼ </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Tỉnh thành phố ▼ </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Thời gian ▼ </div> <div style="border: 1px solid black; width: 100%; height: 150px; margin-top: 10px;"></div> <p style="text-align: center;">Dữ liệu tình hình dịch bệnh</p>
--	--

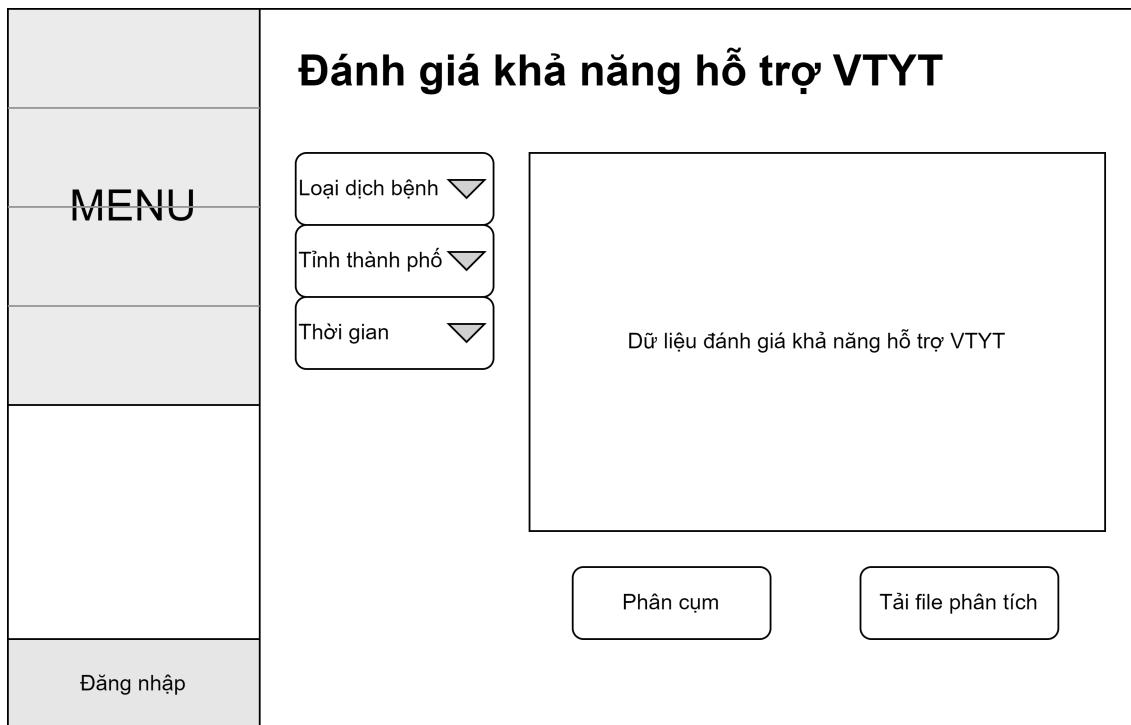
Hình 4.9: Minh họa màn hình chức năng Xem số liệu dịch bệnh

Màn hình chức năng Xem số liệu VTYT:

<p>MENU</p> <p>Đăng nhập</p>	<p>Xem số liệu VTYT</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Loại dịch bệnh ▼ </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Tỉnh thành phố ▼ </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Thời gian ▼ </div> <div style="border: 1px solid black; width: 100%; height: 150px; margin-top: 10px;"></div> <p style="text-align: center;">Dữ liệu VTYT</p>
--	--

Hình 4.10: Minh họa màn hình chức năng Xem số liệu VTYT

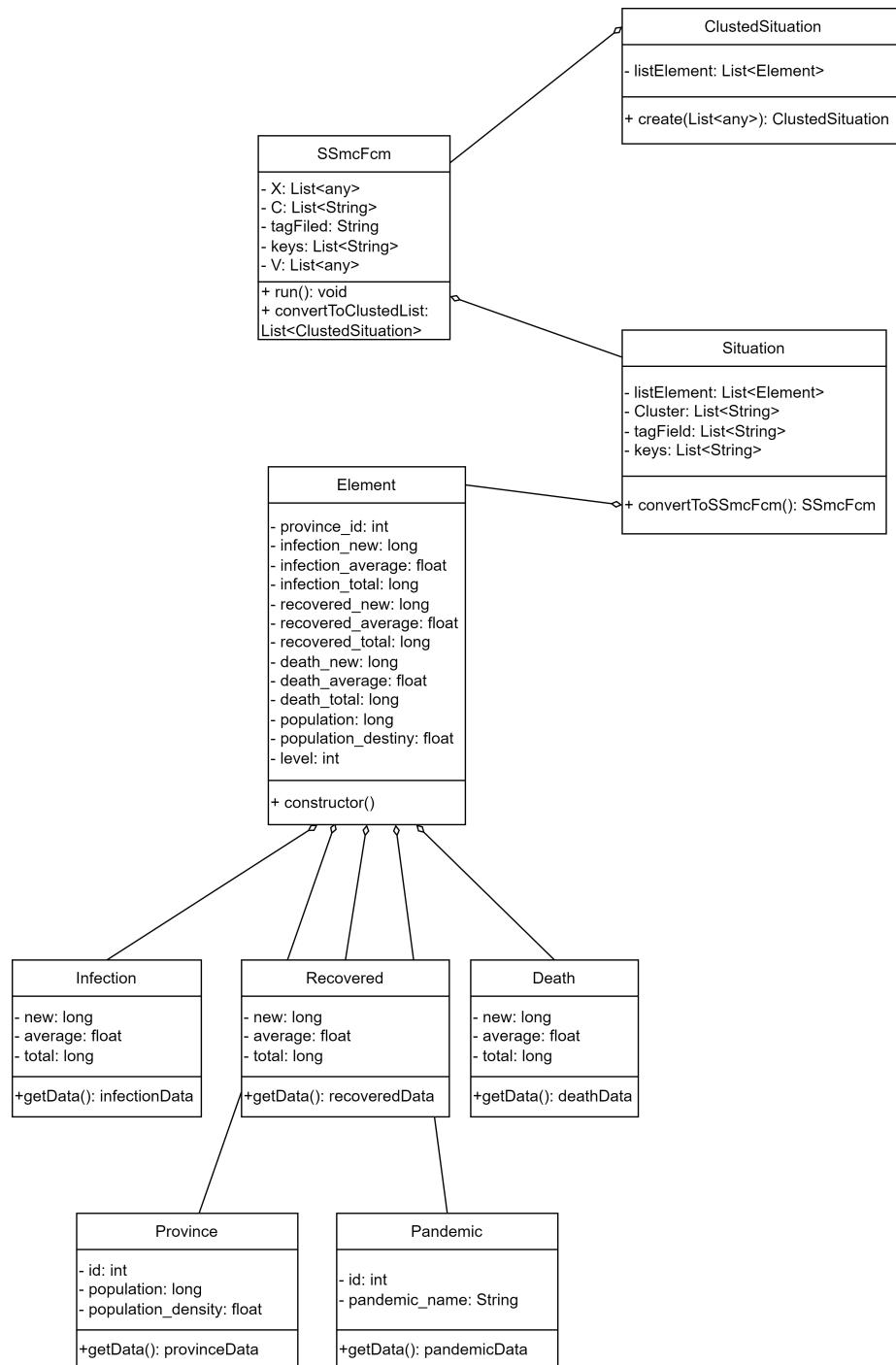
Màn hình chức năng Đánh giá khả năng hỗ trợ VTYT:



Hình 4.11: Minh họa màn hình chức năng Đánh giá khả năng hỗ trợ VTYT

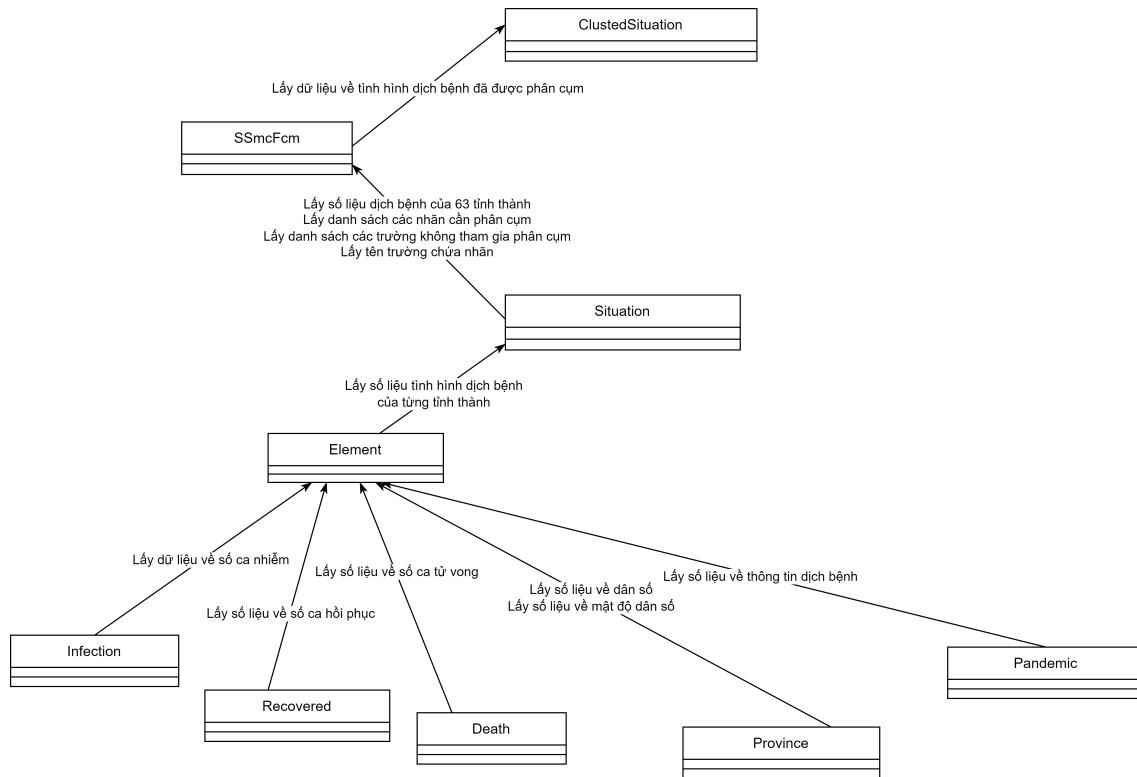
4.2.2 Thiết kế lớp

a, Thiết kế chi tiết lớp trong chức năng Phân tích tình hình dịch bệnh



Hình 4.12: Thiết kế chi tiết lớp trong chức năng Phân tích tình hình dịch bệnh

Giao tiếp giữa các lớp trong chức năng Phân tích tình hình dịch bệnh:



Hình 4.13: Giao tiếp giữa các lớp trong chức năng Phân tích tình hình dịch bệnh

Chi tiết các lớp nổi bật:

Lớp Tình hình dịch bệnh từng thành phố (Element):

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Tên lớp	Tình hình dịch bệnh từng thành phố (Element)
Thuộc tính	<ul style="list-style-type: none"> province_id: Id tỉnh thành phố infection_new: Số ca nhiễm mới infection_average: Số ca nhiễm trung bình infection_total: Tổng số ca nhiễm recovered_new: Số ca hồi phục mới recovered_average: Số ca hồi phục trung bình recovered_total: Tổng số ca hồi phục death_new: Số ca tử vong mới death_average: Số ca tử vong trung bình death_total: Tổng số ca tử vong population: Dâ số population_destiny: Mật độ dân số level: Cấp độ dịch
Phương thức	constructor(): Hàm khởi tạo

Bảng 4.1: Thiết kế lớp Tình hình dịch bệnh từng thành phố (Element)

Lớp Tình hình dịch bệnh tất cả các tỉnh thành (Situation):

Tên lớp	Tình hình dịch bệnh tất cả các tỉnh thành (Situation)
Thuộc tính	<ul style="list-style-type: none"> listElement: Danh sách tình hình dịch bệnh các tỉnh thành Cluster: Danh sách các nhãn tagField: Thuộc tính cần phân cụm keys: Các thuộc tính không tham gia quá trình phân cụm
Phương thức	convertToSSmcFcm(): Chuyển dữ liệu sang kiểu SSmcFcm

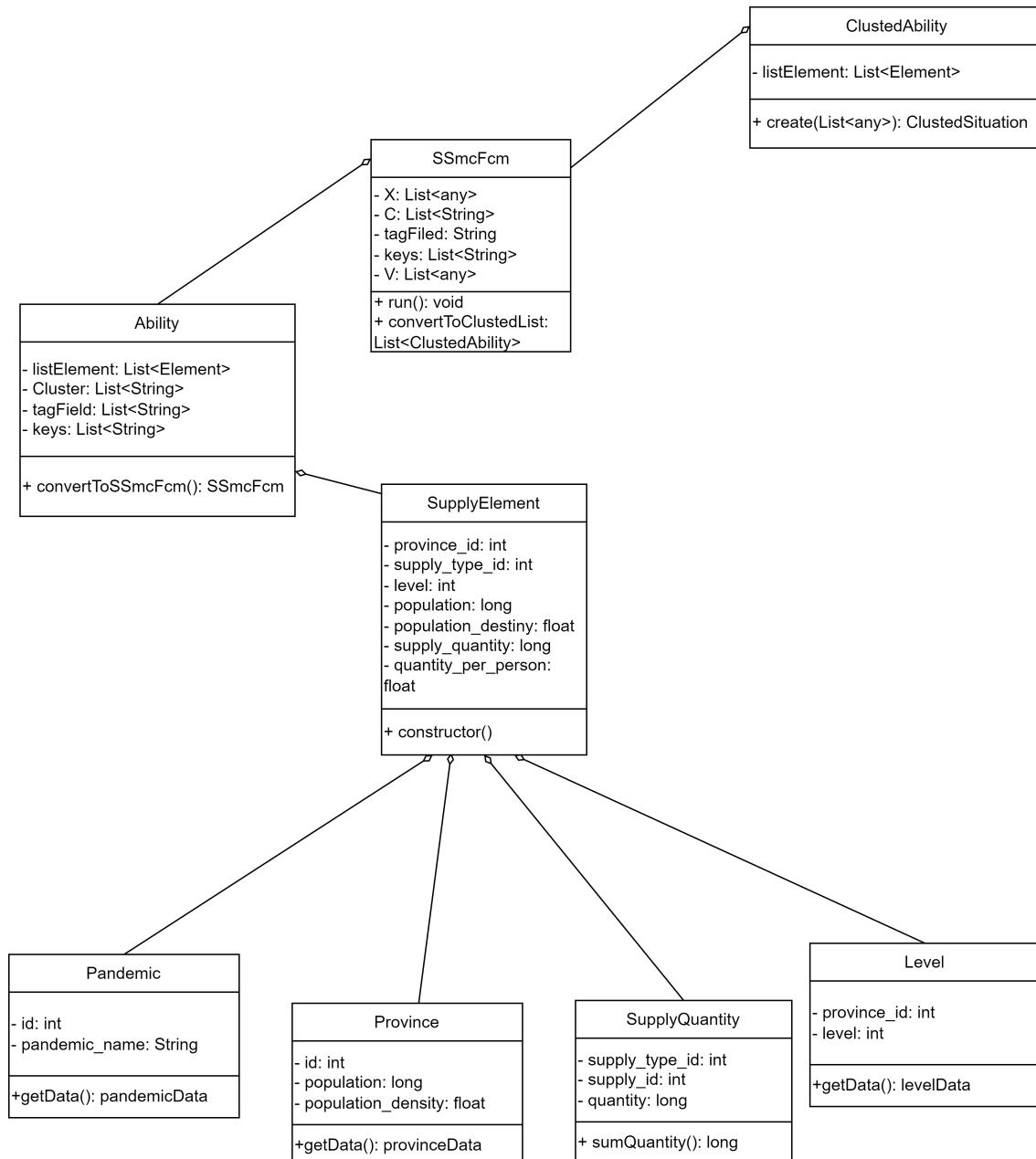
Bảng 4.2: Thiết kế Lớp Tình hình dịch bệnh tất cả các tỉnh thành (Situation)

Lớp Dữ liệu phân cụm (SSmcFcm):

Tên lớp	Dữ liệu phân cụm (SSmcFcm)
Thuộc tính	<p>X: mảng chứa dữ liệu đầu vào</p> <p>C: mảng chứa danh sách các nhãn cần phân cụm</p> <p>tagFiled: trường dữ liệu chứa nhãn</p> <p>keys: mảng chứa tên các trường dữ liệu không tham gia vào quá trình phân cụm</p> <p>V: mảng chứa các tâm phân cụm mới</p>
Phương thức	<p>run(): Khởi chạy quá trình phân cụm</p> <p>convertToClustedList: Chuyển dữ liệu phân cụm sang kiểu ClustedSituation</p>

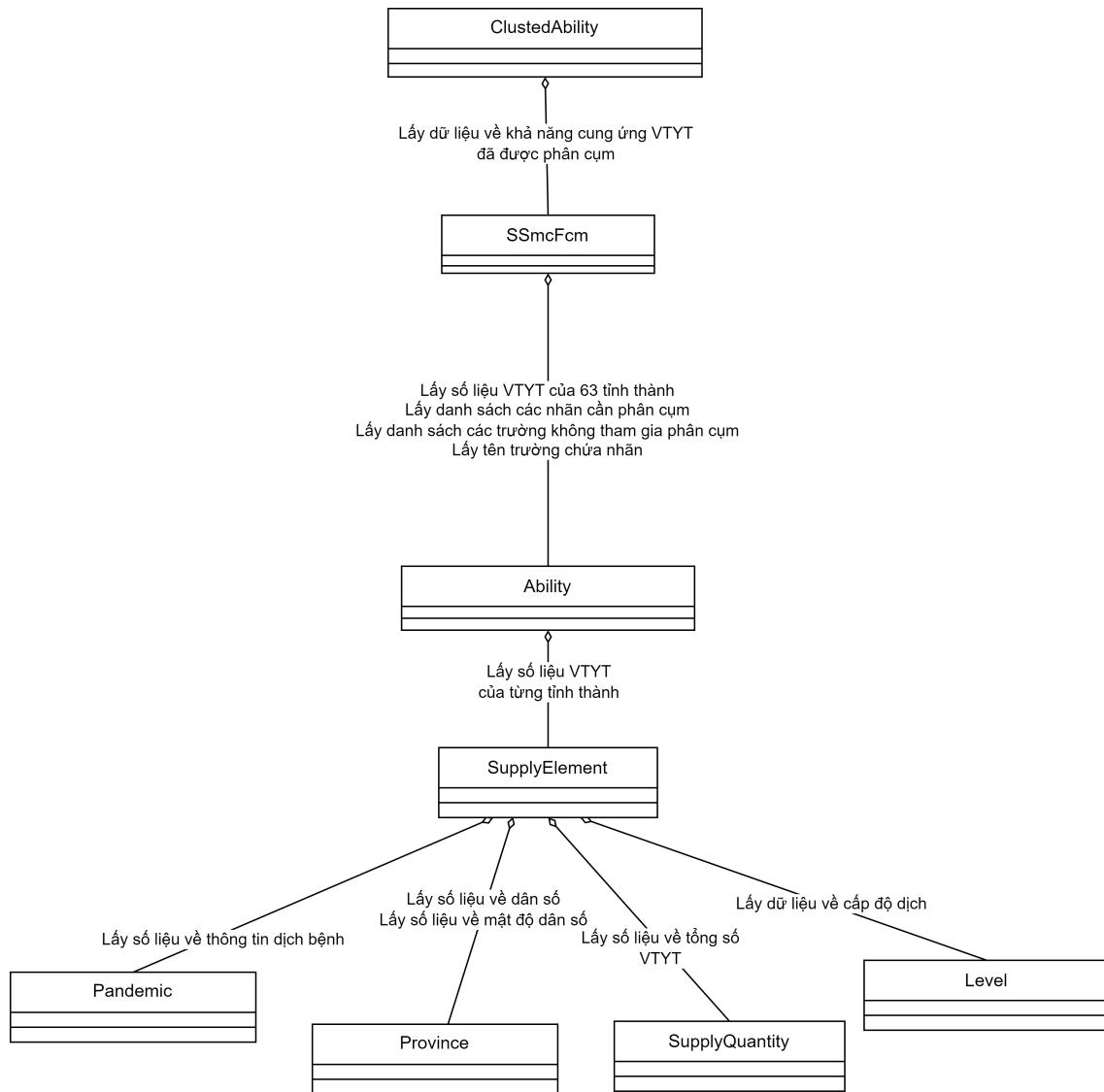
Bảng 4.3: Thiết kế Lớp Dữ liệu phân cụm (SSmcFcm)

b, Thiết kế chi tiết lớp trong chức năng Phân tích khả năng hỗ trợ VTYT



Hình 4.14: Thiết kế chi tiết lớp trong chức năng Phân tích khả năng hỗ trợ VTYT

Giao tiếp giữa các lớp trong chức năng Phân tích khả năng hỗ trợ VTYT:



Hình 4.15: Giao tiếp giữa các lớp trong chức năng Phân tích khả năng hỗ trợ VTYT

Chi tiết các lớp nổi bật:

Lớp Tỉnh hình VTYT từng thành phố (SupplyElement):

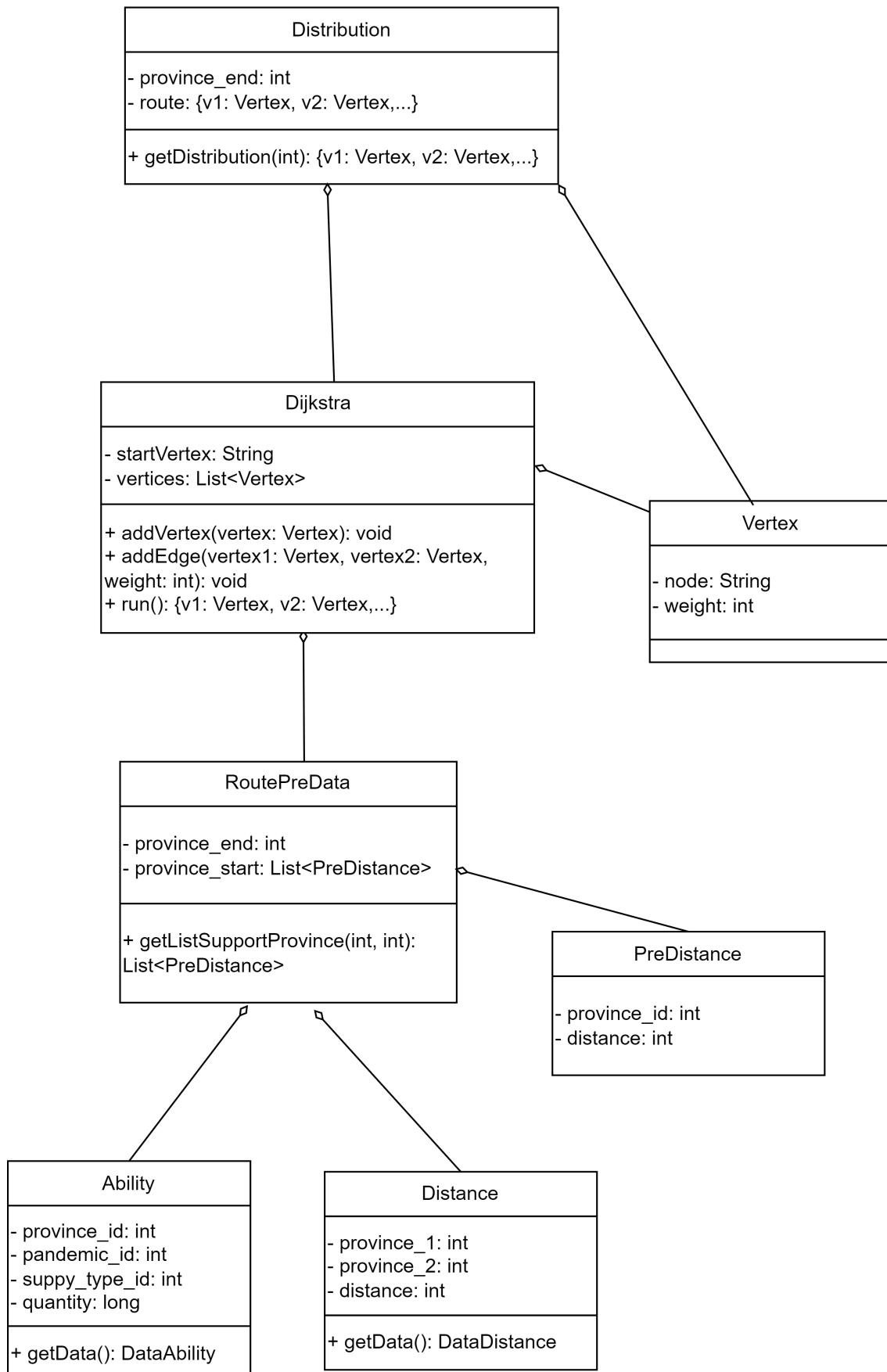
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Tên lớp	Tình hình VTYT từng thành phố (SupplyElement)
Thuộc tính	province_id: Id tỉnh thành phố supply_type_id: Id loại VTYT level: Cấp độ dịch population: Dân số population_destiny: Mật độ dân số supply_quantity: Số lượng VTYT quantity_per_person: Số lượng VTYT/người
Phương thức	constructor(): Hàm khởi tạo

Bảng 4.4: Thiết kế lớp Tình hình VTYT từng thành phố (SupplyElement)

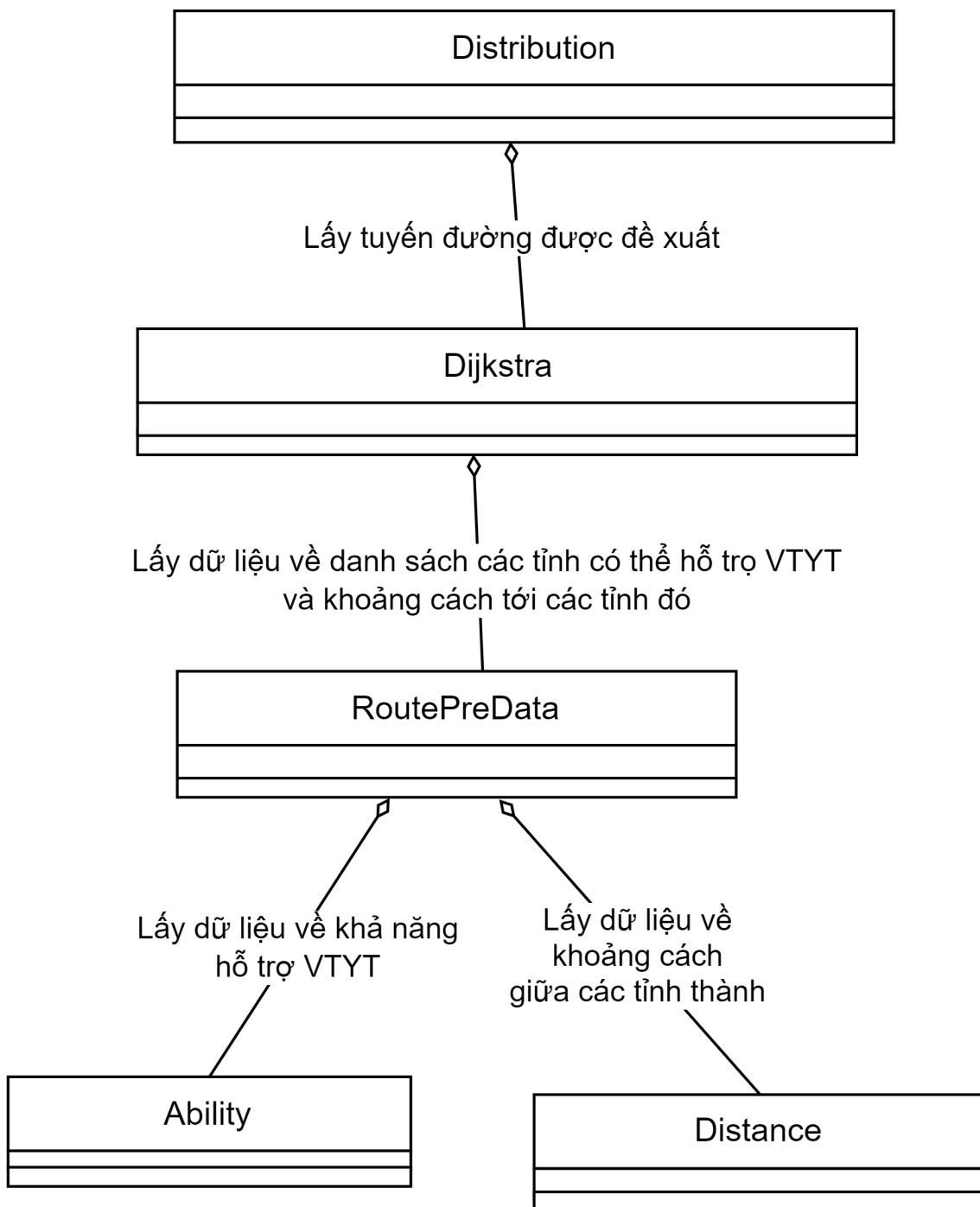
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

c, Thiết kế chi tiết lớp trong chức năng Xem đề xuất phân bổ VTYT



47 **Hình 4.16:** Thiết kế chi tiết lớp trong chức năng Xem đề xuất phân bổ VTYT

Giao tiếp giữa các lớp trong chức năng Xem đề xuất phân bổ VTYT:



Hình 4.17: Giao tiếp giữa các lớp trong chức năng Xem đề xuất phân bổ VTYT

Chi tiết các lớp nổi bật:

Lớp Dữ liệu tiền xử lý tìm đường đi (RoutePreData):

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Tên lớp	Dữ liệu tiền xử lý tìm đường đi (RoutePreData)
Thuộc tính	<p>province_end: Id của tỉnh thành cần hỗ trợ</p> <p>province_start: Danh sách các tỉnh thành có thể hỗ trợ VTYT và khoảng cách từ nơi đó đến tỉnh thành province_end</p>
Phương thức	<p>getListSupportProvince(int, int): input là id tỉnh thành cần hỗ trợ VTYT và id loại VTYT, output là danh sách các tỉnh thành có thể hỗ trợ VTYT và khoảng cách tới tỉnh thành đó.</p>

Bảng 4.5: Thiết kế lớp Dữ liệu tiền xử lý tìm đường đi (RoutePreData)

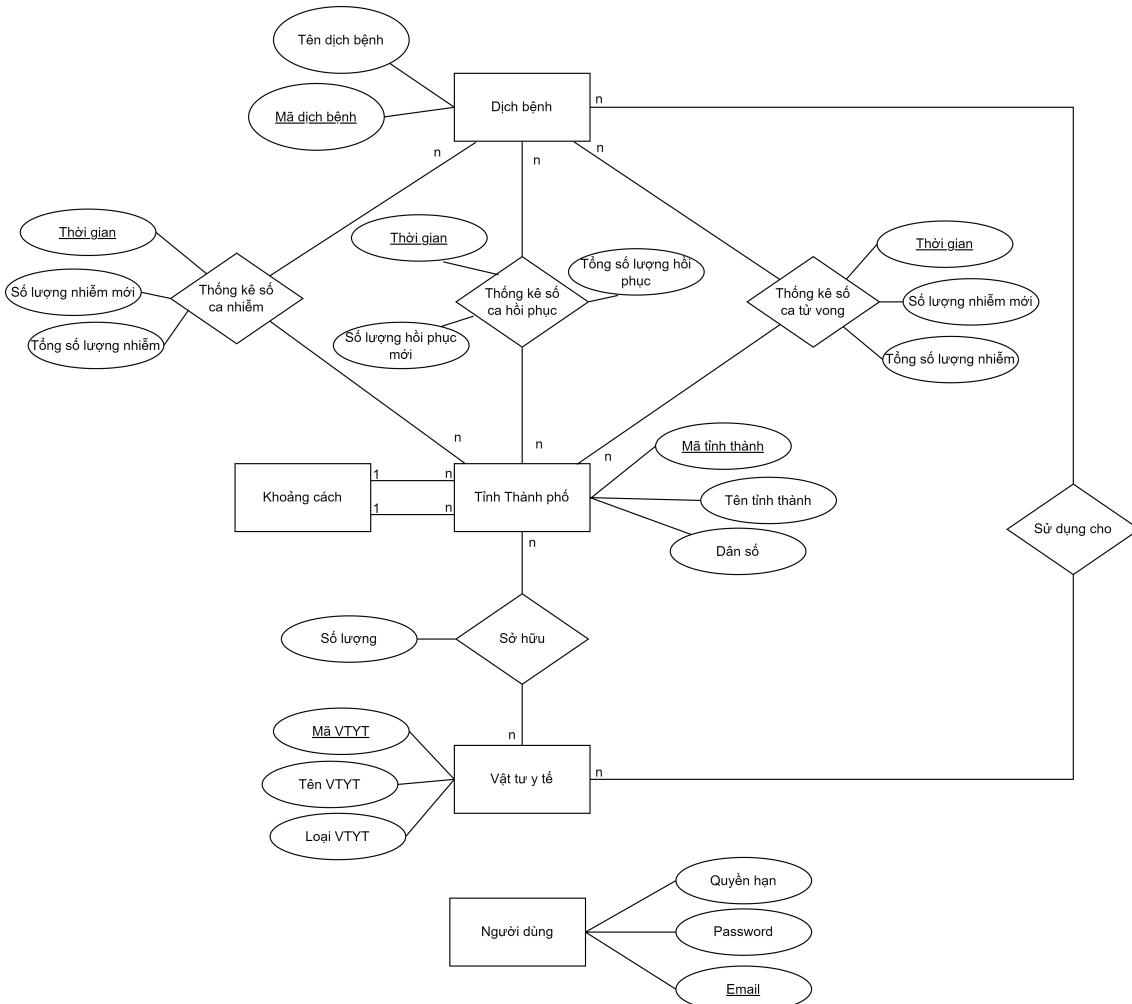
Lớp Dijkstra:

Tên lớp	Dijkstra
Thuộc tính	<p>startVertex: Điểm bắt đầu của đồ thị vô hướng</p> <p>vertices: Danh sách khoảng cách giữa 2 điểm trong đồ thị</p>
Phương thức	<p>addVertex(vertex: Vertex): Thêm các cạnh của đồ thị</p> <p>addEdge(vertex1: Vertex, vertex2: Vertex, weight: int): Thêm các đỉnh của đồ thị</p> <p>run(): Khởi chạy thuật toán, xuất ra kết quả là quãng đường ngắn nhất</p>

Bảng 4.6: Thiết kế lớp Dijkstra

4.2.3 Thiết kế cơ sở dữ liệu

Vẽ sơ đồ thực thể liên kết (E-R diagram):



Hình 4.18: Biểu đồ thực thể liên kết

Trong sơ đồ trên có 5 thực thể gồm tỉnh thành phố, khoảng cách, dịch bệnh, vật tư y tế và người dùng. Giữa Tỉnh thành và Dịch bệnh có các quan hệ Thông kê số ca nhiễm, số ca hồi phục và số ca tử vong. Mỗi tỉnh thành có thể có nhiều loại bệnh dịch đang lây lan và mỗi loại dịch bệnh lại có thể đang lây lan tại nhiều tỉnh thành.

Giữa 2 thực thể Tỉnh thành và Vật tư y tế có mối quan hệ Sở hữu. Mối quan hệ này được biểu thị bằng số lượng VTYT mà tỉnh thành đó đang có. Mỗi tỉnh thành có thể có nhiều loại VTYT và mỗi loại VTYT có thể có ở nhiều tỉnh thành.

Giữa 2 thực thể Vật tư y tế và Dịch bệnh có mối quan hệ Sử dụng, biểu thị rằng đối với loại dịch bệnh nào thì sẽ sử dụng những loại VTYT tương ứng.

Thực thể Người dùng biểu thị thông tin và quyền hạn của các tài khoản tham gia hệ thống, không có quan hệ gì với các thực thể khác.

Từ biểu đồ thực thể liên kết trên, ta sẽ thiết kế cơ sở dữ liệu theo hệ quản trị CSDL SQL bao gồm các bảng sau:

Bảng mô tả thực thể Tỉnh thành phố:

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Thuộc tính	Kiểu dữ liệu	Chú thích
<i>id</i>	int	Mã tỉnh thành
name	varchar	Tên tỉnh thành
population	bigint	Dân số
population_destiny	float	Mật độ dân số

Bảng 4.7: Mô tả thực thể Tỉnh thành phố trong CSDL

Bảng mô tả thực thể Dịch bệnh:

Thuộc tính	Kiểu dữ liệu	Chú thích
<i>id</i>	varchar	Mã dịch bệnh
name	varchar	Tên dịch bệnh

Bảng 4.8: Mô tả thực thể dịch bệnh

Bảng mô tả thực thể Vật tư y tế:

Thuộc tính	Kiểu dữ liệu	Chú thích
<i>id</i>	varchar	Mã VTYT
name	varchar	Tên VTYT
type	varchar	Loại VTYT

Bảng 4.9: Mô tả thực thể Vật tư y tế

Bảng mô tả thực thể Người dùng:

Thuộc tính	Kiểu dữ liệu	Chú thích
<i>email</i>	varchar	Email
username	varchar	Username
password	varchar	Password
role	int	Quyền hạn

Bảng 4.10: Mô tả thực thể Người dùng

Bảng mô tả liên kết Thông kê số ca nhiễm:

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Thuộc tính	Kiểu dữ liệu	Chú thích
<i>province_id</i>	int	Mã tỉnh thành
<i>epicdemic_id</i>	varchar	Mã dịch bệnh
<i>date</i>	date	Thời gian
<i>new</i>	bigint	Số ca nhiễm mới
<i>total</i>	bigint	Tổng số ca nhiễm

Bảng 4.11: Mô tả liên kết Thông kê số ca nhiễm

Bảng mô tả liên kết Thông kê số ca hồi phục:

Thuộc tính	Kiểu dữ liệu	Chú thích
<i>province_id</i>	int	Mã tỉnh thành
<i>epicdemic_id</i>	varchar	Mã dịch bệnh
<i>date</i>	date	Thời gian
<i>new</i>	bigint	Số ca hồi phục mới
<i>total</i>	bigint	Tổng số ca hồi phục

Bảng 4.12: Mô tả liên kết Thông kê số ca hồi phục

Bảng mô tả liên kết Thông kê số ca tử vong:

Thuộc tính	Kiểu dữ liệu	Chú thích
<i>province_id</i>	int	Mã tỉnh thành
<i>epicdemic_id</i>	varchar	Mã dịch bệnh
<i>date</i>	date	Thời gian
<i>new</i>	bigint	Số ca tử vong mới
<i>total</i>	bigint	Tổng số ca tử vong

Bảng 4.13: Mô tả liên kết Thông kê số ca tử vong

Bảng mô tả khoảng cách:

Thuộc tính	Kiểu dữ liệu	Chú thích
<i>province_id_1</i>	int	Mã tỉnh thành 1
<i>province_id_2</i>	int	Mã tỉnh thành 2
<i>distance</i>	bigint	Khoảng cách

Bảng 4.14: Mô tả khoảng cách

Bảng mô tả liên kết Sở hữu VTYT:

Thuộc tính	Kiểu dữ liệu	Chú thích
<i>province_id</i>	int	Mã tỉnh thành 1
<i>supply_id</i>	int	Mã VTYT
<i>quantity</i>	bigint	Số lượng

Bảng 4.15: Mô tả liên kết Sở hữu VTYT

4.2.4 Xây dựng bảng đặc trưng phục vụ quá trình phân cụm

a, Bảng đặc trưng tình hình dịch bệnh

Thực hiện tiền xử lý: Trước khi thực hiện quá trình phân cụm, chuyên viên y tế cần tiến hành gán nhãn cho một số tỉnh thành dựa trên một công thức đánh giá cấp độ dịch bệnh.

Công thức đánh giá cấp độ dịch bệnh:

$$k = \frac{\alpha r\bar{r}r^*}{(e\bar{e}e^*)(d\bar{d}d^*)(P\bar{P})}$$

Trong đó:

- k: Hệ số đánh giá cấp độ dịch
- α : Hệ số dịch bệnh (do chuyên viên y tế quyết định, tùy vào biến động dữ liệu)
- $r\bar{r}r^*$: Lần lượt là số ca hồi phục mới, số ca hồi phục trung bình 7 ngày, tổng số ca hồi phục
- $e\bar{e}e^*$: Lần lượt là số ca nhiễm mới, số ca nhiễm trung bình 7 ngày, tổng số ca nhiễm
- $d\bar{d}d^*$: Lần lượt là số ca tử vong mới, số ca tử vong trung bình 7 ngày, tổng số ca tử vong
- $P\bar{P}$: Lần lượt là tổng dân số và mật độ dân số.

Sau khi tính toán giá trị đánh giá, chuyên viên y tế sẽ sử dụng các ngưỡng để phân loại cấp độ dịch bệnh cho từng tỉnh thành. Nếu giá trị đánh giá nhỏ hơn 0.1, tức là tại tỉnh thành đó có tỷ lệ hồi phục cao hơn và số ca nhiễm, số ca tử vong và mật độ dân số thấp hơn, tỉnh thành đó sẽ được coi là cấp độ 1 (An toàn). Nếu giá trị đánh giá nằm trong khoảng từ 0.1 đến 0.15, tức là tỷ lệ hồi phục cao hơn nhưng số ca nhiễm, số ca tử vong và mật độ dân số có thể cao hơn, tỉnh thành đó sẽ được coi là cấp độ 2 (Chú ý). Cuối cùng, nếu giá trị đánh giá lớn hơn 0.15, tức là tỷ lệ hồi phục thấp hơn và số ca nhiễm, số ca tử vong và mật độ dân số cao hơn, tỉnh thành đó sẽ được coi là cấp độ 3 (Nguy hiểm).

Qua công việc tiền xử lý này, chuyên viên y tế sẽ có cái nhìn tổng quan về mức độ dịch bệnh tại từng tỉnh thành và có thể chuẩn bị dữ liệu cho quá trình phân cụm bán giám sát sau đó. Bằng việc áp dụng thuật toán phân cụm, các tỉnh thành có tình hình dịch bệnh tương đồng sẽ được gom vào cùng một nhóm, giúp chuyên gia y tế dễ dàng đưa ra các quyết định phù hợp để ứng phó với dịch bệnh một cách hiệu quả.

Tạo bảng đặc trưng: Bảng đặc trưng được sử dụng trong việc phân cụm các tỉnh thành dựa trên cấp độ dịch cung cấp một tập hợp các thuộc tính quan trọng để mô tả tình hình dịch bệnh và các yếu tố liên quan tới mỗi tỉnh thành.

Các thuộc tính trong bảng đặc trưng bao gồm:

Thuộc tính	Chú thích
province_id	Tỉnh/Thành phố
infection_new	Số lượng ca nhiễm mới
infection_average	Số lượng ca nhiễm trung bình 7 ngày
infection_total	Tổng số ca nhiễm
recovered_new	Số lượng ca hồi phục mới
recovered_average	Số lượng ca hồi phục trung bình 7 ngày
recovered_total	Tổng số ca hồi phục
death_new	Số lượng ca tử vong mới
death_average	Số lượng ca tử vong trung bình 7 ngày
death_total	Tổng số ca tử vong
population	Dân số
population_destiny	Mật độ dân cư
level	Cấp độ dịch

Bảng 4.16: Bảng đặc trưng tình hình dịch bệnh

- Số lượng ca nhiễm mới (infection_new): Đây là thuộc tính đo lường số lượng ca nhiễm mới trong một khoảng thời gian nhất định. Nó cho biết tốc độ lây lan của dịch bệnh trong mỗi tỉnh thành.
- Số lượng ca nhiễm trung bình trong 7 ngày (infection_average): Thuộc tính này tính trung bình số ca nhiễm trong 7 ngày gần nhất. Nó cung cấp một cái nhìn tổng quan về xu hướng dịch bệnh trong thời gian gần đây.
- Tổng số ca nhiễm (infection_total): Đây là thuộc tính đếm tổng số ca nhiễm tính đến thời điểm hiện tại. Nó cho biết quy mô tổng thể của dịch bệnh trong

mỗi tỉnh thành.

- Số lượng ca hồi phục mới (recovered_new): Thuộc tính này đo lường số lượng ca hồi phục mới trong một khoảng thời gian nhất định. Nó cho biết khả năng phục hồi và quy mô của quá trình chữa trị.
- Số lượng ca hồi phục trung bình trong 7 ngày (recovered_average): Thuộc tính này tính trung bình số ca hồi phục trong 7 ngày gần nhất. Nó cung cấp thông tin về xu hướng phục hồi của dịch bệnh.
- Tổng số ca hồi phục (recovered_total): Đây là thuộc tính đếm tổng số ca hồi phục tính đến thời điểm hiện tại. Nó cho biết tổng quan về số lượng người đã khỏi bệnh.
- Số lượng ca tử vong mới (death_new): Thuộc tính này đo lường số lượng ca tử vong mới trong một khoảng thời gian nhất định. Nó cho biết tác động của dịch bệnh đến mức độ tử vong.
- Số lượng ca tử vong trung bình trong 7 ngày (death_average): Thuộc tính này tính trung bình số ca tử vong trong 7 ngày gần nhất. Nó cung cấp thông tin về xu hướng tử vong của dịch bệnh.
- Tổng số ca tử vong (death_total): Đây là thuộc tính đếm tổng số ca tử vong tính đến thời điểm hiện tại. Nó cho biết quy mô tổng thể của tác động của dịch bệnh đến số lượng người tử vong.
- Dân số của mỗi tỉnh thành (population): Thuộc tính này đo lường dân số của mỗi tỉnh thành. Nó cho phép xem xét tác động của quy mô dân số đến quá trình lây lan và kiểm soát dịch bệnh.
- Mật độ dân cư của tỉnh thành (population_density): Thuộc tính này đo lường mật độ dân cư của mỗi tỉnh thành, tức là số lượng dân số trên một đơn vị diện tích. Nó cung cấp thông tin về mức độ tập trung của dân số và có thể ảnh hưởng đến tốc độ lây lan của dịch bệnh.
- Đầu ra của bảng đặc trưng là phân loại cấp độ dịch của từng tỉnh thành. Giá trị phân loại được gán cho mỗi tỉnh thành từ 1 đến 3, đại diện cho các cấp độ dịch khác nhau.
- Phân loại cấp độ dịch được xác định dựa trên các thuộc tính trong bảng đặc trưng và quyết định của thuật toán phân cụm. Thuật toán sẽ phân tích các mẫu và sự tương đồng giữa các tỉnh thành dựa trên các thuộc tính đã cho. Kết quả của quá trình này là việc gán mỗi tỉnh thành vào một trong ba cấp độ dịch:
 - Cấp độ dịch thấp: Đại diện cho các tỉnh thành có tình hình dịch bệnh

tương đối ổn định, ít ca nhiễm mới, số ca hồi phục cao, và số ca tử vong thấp.

- Cấp độ dịch trung bình: Đại diện cho các tỉnh thành có tình hình dịch bệnh có mức độ trung bình, với một số ca nhiễm mới, ca hồi phục và ca tử vong đáng kể.
- Cấp độ dịch cao: Đại diện cho các tỉnh thành có tình hình dịch bệnh nghiêm trọng, có số lượng ca nhiễm mới, ca hồi phục và ca tử vong cao.

Tạo bảng đặc trưng: Các thuộc tính trong bảng đặc trưng này cung cấp thông tin đa dạng và phản ánh tình hình dịch bệnh trong từng tỉnh thành. Bằng cách sử dụng các phương pháp phân cụm, chúng ta có thể xác định và nhận biết các mẫu tương đồng giữa các tỉnh thành và phân loại chúng vào các cấp độ dịch khác nhau. Điều này giúp cho việc đưa ra quyết định và triển khai các biện pháp kiểm soát và ứng phó dịch bệnh hiệu quả hơn trong mỗi khu vực.

b, Bảng đặc trưng khả năng hỗ trợ VTYT

Thực hiện tiền xử lý: Trước khi thực hiện quá trình phân cụm, chuyên viên y tế cần thực hiện công việc gán nhãn cho từng tỉnh thành dựa trên một công thức đánh giá khả năng hỗ trợ VTYT.

Công thức đánh giá khả năng hỗ trợ VTYT:

$$k = \frac{\beta * \bar{q}}{\bar{P} * l}$$

Trong đó:

- k : Hệ số đánh giá khả năng cung ứng VTYT
- β : Hệ số tùy chọn (do chuyên viên y tế quyết định, tùy vào biến động các số liệu)
- \bar{q} : số lượng VTYT trên mỗi người dân của tỉnh thành đó
- \bar{P} : mật độ dân cư của tỉnh thành
- l : cấp độ dịch bệnh tại tỉnh thành

Khi đã tính toán được giá trị đánh giá, chuyên viên y tế sẽ sử dụng các ngưỡng để phân loại khả năng hỗ trợ VTYT của từng tỉnh thành. Nếu giá trị đánh giá nằm trong khoảng từ 0.1 đến 0.15, tức là tỉnh thành đó có tỷ lệ VTYT trên người dân tương đối cao, mật độ dân cư thích hợp và cấp độ dịch bệnh ở mức chấp nhận được, tỉnh thành đó sẽ được coi là có khả năng tự cung ứng VTYT ($0.1 < \text{Giá trị đánh giá} < 0.15$). Nếu giá trị đánh giá nhỏ hơn 0.1, tức là tỉnh thành đó cần hỗ trợ VTYT từ các tỉnh thành khác do tỷ lệ VTYT trên người dân còn thấp hơn. Nếu giá trị đánh

giá lớn hơn 0.15, tức là tỉnh thành đó có thể hỗ trợ VTYT tới các tỉnh thành khác do tỷ lệ VTYT trên người dân cao hơn, giúp cân bằng nguồn cung cấp VTYT giữa các khu vực.

Công việc tiền xử lý này giúp chuyên viên y tế có cái nhìn tổng quan về khả năng hỗ trợ VTYT của mỗi tỉnh thành và chuẩn bị dữ liệu cho quá trình phân cụm bán giám sát sau đó. Bằng việc áp dụng thuật toán phân cụm, các tỉnh thành có khả năng hỗ trợ VTYT tương đồng sẽ được nhóm lại với nhau, giúp tối ưu hóa việc phân chia và phân bổ VTYT một cách hiệu quả và bền vững. Nhờ đó, các cơ quan y tế có thể có được cái nhìn tổng quan về tình hình VTYT trong quá trình điều trị dịch bệnh và đưa ra các quyết định phù hợp để tăng cường khả năng hỗ trợ và đáp ứng nhu cầu y tế trong thời điểm khẩn cấp.

Bảng đặc trưng được tạo ra để phân cụm các tỉnh thành dựa trên khả năng hỗ trợ Vật tư y tế (VTYT). Bảng này chứa các thuộc tính quan trọng để đánh giá và phân loại các tỉnh thành theo khả năng hỗ trợ VTYT.

Các thuộc tính trong bảng đặc trưng bao gồm:

Thuộc tính	Chú thích
province_id	Tỉnh/Thành phố
level	Cấp độ dịch
population	Dân số
population_destiny	Mật độ dân cư
supply_quantity	Số lượng VTYT
quantity_per_person	Số lượng VTYT/người
ability	Khả năng hỗ trợ VTYT

Bảng 4.17: Bảng đặc trưng khả năng hỗ trợ VTYT

- Số lượng VTYT (supply_quantity): Đây là thuộc tính đo lường số lượng VTYT có sẵn trong mỗi tỉnh thành. VTYT có thể bao gồm các loại vật tư y tế như khẩu trang, găng tay, bộ đồ bảo hộ, thuốc, và nhiều vật tư khác. Những loại VTYT được đưa vào bảng sẽ được quyết định bởi loại dịch bệnh đang quan sát. Số lượng VTYT cho biết khả năng cung cấp các tài nguyên y tế quan trọng để đối phó với dịch bệnh.
- Dân số của mỗi tỉnh thành (population): Đây là thuộc tính đo lường dân số của mỗi tỉnh thành. Nó cung cấp thông tin về quy mô dân số trong việc đánh giá nhu cầu và khả năng cung cấp VTYT. Tỉnh thành với dân số lớn có thể đòi hỏi một lượng VTYT lớn hơn để đáp ứng nhu cầu y tế.

- Mật độ dân cư của tỉnh thành (population_density): Đây là thuộc tính đo lường mật độ dân cư của mỗi tỉnh thành, tức là số lượng dân số trên một đơn vị diện tích. Mật độ dân cư có thể ảnh hưởng đến khả năng phân phối và tiếp cận VTYT, vì nó thể hiện mức độ tập trung dân số trong một khu vực. Tỉnh thành với mật độ dân cư cao có thể đặt ra một thách thức trong việc phân phối và sử dụng hiệu quả các tài nguyên y tế.
- Cấp độ dịch (level): Đây là thuộc tính đánh giá tình hình dịch bệnh của mỗi tỉnh thành phố. Thuộc tính này có thể được tính thông qua chức năng Phân tích tình hình dịch bệnh. Cấp độ dịch có ảnh hưởng rất lớn đến khả năng hỗ trợ VTYT của tỉnh thành đó. Nếu 1 tỉnh thành có nhiều VTYT nhưng cấp độ dịch lại đang là cấp 3 (nguy hiểm) thì sẽ khó có thể coi là thuộc nhóm có thể hỗ trợ những tỉnh thành khác. Thuộc tính này có ảnh hưởng rất lớn đến kết quả phân cụm.
- Khả năng hỗ trợ VTYT (ability): Đây là thuộc tính đầu ra, phân loại tỉnh thành dựa trên khả năng hỗ trợ VTYT. Có ba giá trị phân loại: "Cần được hỗ trợ", "Có thể đi hỗ trợ vùng khác", và "Chỉ đủ cung ứng cho chính nơi đó". Cấp độ dịch cho biết mức độ cần thiết và khả năng hỗ trợ VTYT của từng tỉnh thành. Việc phân loại này giúp định hình chiến lược và phân phối tài nguyên y tế đúng đắn đến từng tỉnh thành.

Bằng cách sử dụng các thuộc tính trong bảng đặc trưng, chúng ta có thể phân cụm các tỉnh thành vào các nhóm dựa trên khả năng hỗ trợ VTYT. Việc này giúp chúng ta đánh giá và xác định các tỉnh thành cần được ưu tiên hỗ trợ, các tỉnh thành có thể cung cấp hỗ trợ cho vùng khác và các tỉnh thành chỉ đủ cung ứng cho chính nơi đó. Kết quả này giúp quản lý và phân phối tài nguyên y tế một cách hiệu quả và định hướng hỗ trợ đúng đắn cho từng tỉnh thành trong tình hình dịch bệnh.

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Mục đích	Công cụ	URL
Ngôn ngữ lập trình	Javascript	
Môi trường lập trình	NodeJS 18.16.0 LTS	https://nodejs.org/en
Khung ứng dụng web back end	Express.js	https://expressjs.com/
API hỗ trợ liên kết tới CSDL	node-postgres	https://node-postgres.com/
API hỗ trợ code server	nodemon	https://www.npmjs.com/package/nodemon
xây dựng giao diện người dùng	ReactJS	https://react.dev/
Hỗ trợ CSS	bootstrap 5	https://getbootstrap.com/docs/5.0/getting-started
API hỗ trợ bản đồ	Traffic-Aware Directions API	https://nextbillions.ai/
hệ quản trị cơ sở dữ liệu	PostgreSQL 15	https://www.postgresql.org/
API hỗ trợ gửi request từ client lên server	Axios	https://axios-http.com/vi/docs/intro

Bảng 4.18: Danh sách thư viện và công cụ sử dụng

4.3.2 Kết quả đạt được

Kết quả đạt được:

Trong đồ án tốt nghiệp này, tôi đã phát triển một ứng dụng Quản lý Dịch bệnh và Vật tư y tế nhằm cung cấp thông tin về tình hình dịch bệnh tại các tỉnh thành, khả năng hỗ trợ vật tư y tế giữa các đơn vị y tế, cũng như tối ưu hóa tuyến đường vận chuyển vật tư y tế giữa các tỉnh thành. Ứng dụng này hỗ trợ ba vai trò khác nhau: khách, chuyên viên y tế và quản trị viên.

• Giao diện khách:

Chức năng tra cứu tình hình dịch bệnh tại các tỉnh thành: Khách hàng có thể tra cứu thông tin về số ca nhiễm, số ca tử vong, số ca hồi phục, và các thông tin liên quan khác về dịch bệnh tại các tỉnh thành.

Chức năng xem khả năng hỗ trợ vật tư y tế giữa các tỉnh thành: Khách hàng có thể xem thông tin về các loại vật tư y tế, số lượng có sẵn tại mỗi tỉnh thành và khả

năng cung ứng vật tư y tế giữa các tỉnh thành.

- **Giao diện chuyên viên y tế:**

Chức năng đánh giá tình hình dịch bệnh tại các tỉnh thành: Chuyên viên y tế, sau khi đăng nhập vào ứng dụng, có thể sử dụng thuật toán phân cụm mờ bán giám sát để đánh giá tình hình dịch bệnh tại các tỉnh thành. Thông qua việc phân tích dữ liệu, ứng dụng sẽ đưa ra đánh giá về mức độ nguy hiểm của dịch bệnh tại từng địa phương.

Chức năng đánh giá khả năng hỗ trợ vật tư y tế giữa các tỉnh thành: Chuyên viên y tế có thể sử dụng thuật toán phân cụm mờ bán giám sát để đánh giá khả năng hỗ trợ vật tư y tế giữa các tỉnh thành. Đánh giá này dựa trên số lượng và loại vật tư y tế có sẵn, nhu cầu vật tư y tế của từng địa phương và khả năng cung ứng vật tư y tế từ các tỉnh thành khác.

Chức năng xem tuyến đường vận chuyển vật tư y tế: Chuyên viên y tế có thể sử dụng giải thuật Dijkstra để xác định tuyến đường vận chuyển vật tư y tế giữa các tỉnh thành một cách tối ưu về quãng đường. Thông qua việc tối ưu hóa tuyến đường, ứng dụng giúp giảm thời gian và chi phí vận chuyển vật tư y tế giữa các đơn vị y tế.

- **Giao diện quản trị viên:**

Chức năng quản lý tài khoản: Quản trị viên có quyền thêm, sửa, xóa các tài khoản khác trong hệ thống. Điều này giúp đảm bảo an toàn và quản lý quyền truy cập đúng đắn với các người dùng trong ứng dụng.

Thông tin thống kê về ứng dụng:

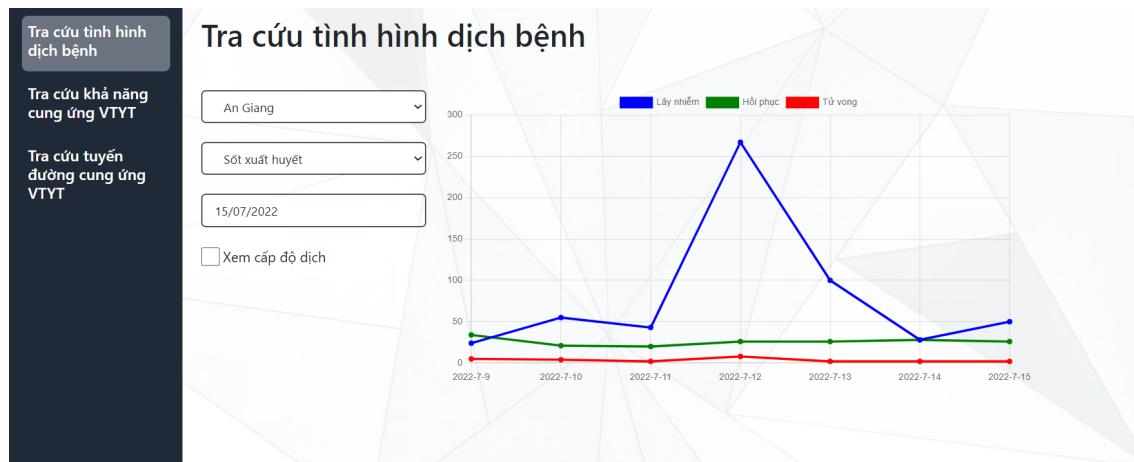
Thông số	Số liệu
Số dòng code	20734 dòng
Số file	114 file
Số folder	47 folder
Dung lượng mã nguồn	9.60 MB
Dung lượng sản phẩm đóng gói	2.148 MB

Bảng 4.19: Thông tin thống kê về ứng dụng

4.3.3 Minh họa các chức năng chính

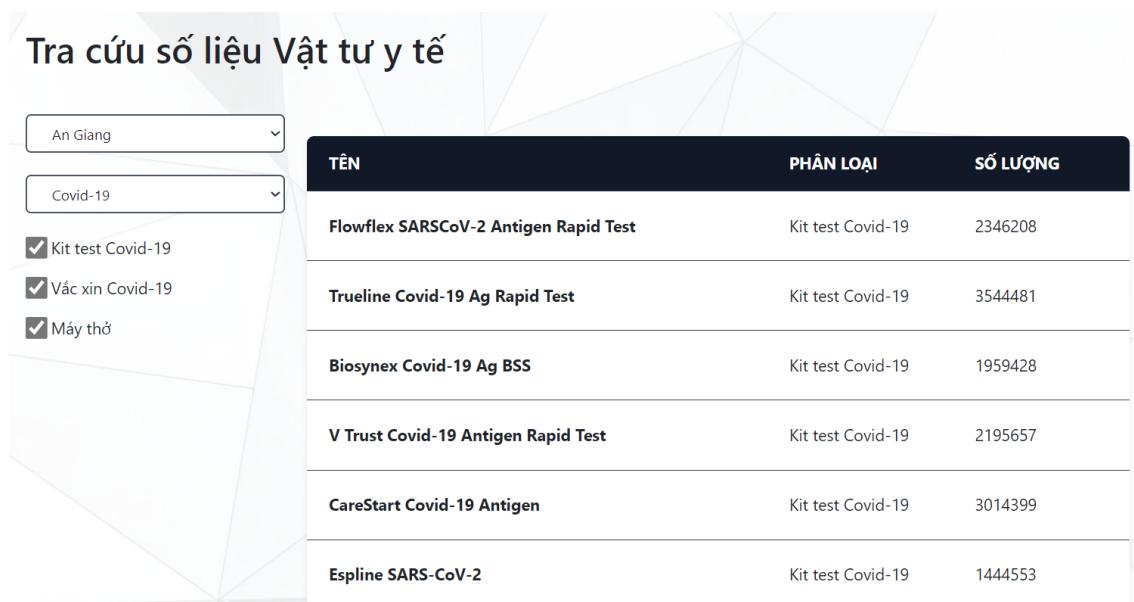
Màn hình chức năng Xem tình hình dịch bệnh:

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ



Hình 4.19: Màn hình chức năng Xem tình hình dịch bệnh

Màn hình chức năng Xem khả năng cung ứng VTYT:



Hình 4.20: Màn hình chức năng Xem khả năng cung ứng VTYT

Màn hình chức năng Tra cứu tuyến đường cung ứng VTYT:

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Tỉnh thành hỗ trợ	Tỉnh thành phố	Quãng đường (km)	Số lượng Vật tư hỗ trợ
Bến Tre		82	1974710
Đồng Nai		31	1180623
Hà Tĩnh		76	17562166
Hưng Yên		88	11906328

Hình 4.21: Màn hình chức năng Tra cứu tuyến đường cung ứng VTYT

Màn hình chức năng Quản lý tài khoản:

Danh sách tài khoản
dattebayo@gmail.com
doihua@gmail.com

Hình 4.22: Màn hình chức năng Quản lý tài khoản

Màn hình chức năng phân tích tình hình dịch bệnh:

- Màn hình Thêm trọng số cho các thuộc tính của bảng đặc trưng tình hình dịch bệnh

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Số lượng ca nhiễm mới	0.1	Số lượng ca nhiễm trung bình 7 ngày	0.1	Tổng số ca nhiễm	0.1
Số lượng ca hồi phục mới	0.1	Số lượng ca hồi phục trung bình 7 ngày	0.1	Tổng số ca hồi phục	0.1
Số lượng ca tử vong mới	0.1	Số lượng ca tử vong trung bình 7 ngày	0.1	Tổng số ca tử vong	0.1
Dân số	0.1	Mật độ dân cư	0.1		

Xác Nhận
Thoát

Hình 4.23: Màn hình Thêm trọng số cho các thuộc tính của bảng đặc trưng tình hình dịch bệnh

- Màn hình phân cụm tĩnh thành theo cấp độ dịch:

The screenshot displays a dashboard with a heatmap overlay on a table of COVID-19 metrics. The table has columns for Thành phố, Số lượng ca nhiễm mới (0.1), Số lượng ca nhiễm trung bình 7 ngày (0.1), Tổng số ca nhiễm (0.1), Số lượng ca hồi phục mới (0.1), Số lượng ca hồi phục trung bình 7 ngày (0.1), Tổng số ca hồi phục (0.1), Số lượng ca tử vong mới (0.1), Số lượng ca tử vong trung bình 7 ngày (0.1), Tổng số ca tử vong (0.1), Dân số (0.1), Mật độ dân cư (0.1), and Cấp độ dịch. The heatmap uses color coding where darker shades represent higher values. Two rows are highlighted: Bà Rịa - Vũng Tàu and Bình Dương. A dropdown menu next to the Cấp độ dịch column for both rows shows 'Cấp 1'.

Thành phố	Số lượng ca nhiễm mới (0.1)	Số lượng ca nhiễm trung bình 7 ngày (0.1)	Tổng số ca nhiễm (0.1)	Số lượng ca hồi phục mới (0.1)	Số lượng ca hồi phục trung bình 7 ngày (0.1)	Tổng số ca hồi phục (0.1)	Số lượng ca tử vong mới (0.1)	Số lượng ca tử vong trung bình 7 ngày (0.1)	Tổng số ca tử vong (0.1)	Dân số (0.1)	Mật độ dân cư (0.1)	Cấp độ dịch
Bà Rịa - Vũng Tàu	49	31.71	244977	73	24	11576	2	3.71	50505	1181302	596.38	Cấp 1
Bình Dương	41	85.57	403930	46	38.86	55327	3	7.71	9331	2678220	993.88	Cấp 1

Hình 4.24: Màn hình phân cụm tĩnh thành theo cấp độ dịch

Màn hình chức năng phân tích khả năng phân bổ VTYT:

- Màn hình Thêm trọng số cho các thuộc tính của bảng đặc trưng khả năng phân bổ VTYT:

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Số lượng Kit test Covid-19 /người

0.1

Dân số

0.1

Mật độ dân cư

0.1

Cấp độ dịch

0.1

Xác Nhận
Thoát

Hình 4.25: Màn hình Thêm trọng số cho các thuộc tính của bảng đặc trưng khả năng phân bổ VTYT

- Màn hình phân cụm tỉnh thành theo khả năng phân bổ VTYT:

Xem trọng số
Download dữ liệu

Xem gợi ý đánh giá

Thành phố	Số lượng Kit test Covid-19 /người (0.1)	Dân số (0.1)	Mật độ dân cư (0.1)	Cấp độ dịch (0.1)	Khả năng hỗ trợ VTYT
Thành phố Hồ Chí Minh	18.43	9411805	4566.62	3	Cần hỗ trợ
Bắc Ninh	19.68	1450518	1763.12	1	Tự cung ứng
Bình Dương	16.91	2678220	993.88	1	Tự cung ứng
Cần Thơ	16.21	1244736	864.88	1	Tự cung ứng
Đà Nẵng	21.25	1191381	927.22	1	Tự cung ứng

Hình 4.26: Màn hình phân cụm tỉnh thành theo khả năng phân bổ VTYT

Màn hình chức năng Thêm dữ liệu:

Thêm dữ liệu vào hệ thống



Hình 4.27: Màn hình chức năng Thêm dữ liệu

4.4 Kiểm thử

4.4.1 Kiểm thử chức năng Xem thông tin tình hình dịch bệnh

Chức năng	Test case	Kết quả
Xem số liệu dịch bệnh	Chọn loại dịch bệnh Covid-19 và hiển thị số liệu dịch bệnh	Đạt
	Chọn thời gian 2021-07-15, màn hình không hiển thị tình hình dịch bệnh	Đạt
	Chọn Tỉnh Hà Nội và thời gian 2022-07-18, hover con trỏ chuột vào ngày 17 trên biểu đồ, số liệu dịch bệnh của ngày đó sẽ hiện ra	Đạt
Xem cấp độ dịch	Chọn loại dịch bệnh Covid-19, màn hình hiển thị số liệu cấp độ dịch	Đạt
	Chọn tỉnh Hải Dương và thời gian 2021-05-01, màn hình không hiển thị cấp độ dịch	Đạt

Bảng 4.20: Bảng Kiểm thử chức năng Xem thông tin tình hình dịch bệnh

4.4.2 Kiểm thử chức năng Đánh giá tình hình dịch bệnh

Chức năng	Test case	Kết quả
Khởi động màn hình	Chọn loại dịch bệnh Sốt xuất huyết, không có thông tin về dịch bệnh và không thể click các nút "Chọn trọng số" hay "Phân cụm"	Đạt
	Chọn loại dịch bệnh sốt xuất huyết, dữ liệu về dịch bệnh được hiển thị	Đạt
Chuẩn bị trọng số phân cụm	Click nút "Chọn trọng số", dialog chọn trọng số hiện ra	Đạt
	Trong input đầu tiên nhập số 1.2	Đạt
	Trong input đầu tiên nhập dãy "abc", không nhập được dãy đó vào ô input	Đạt
Chuẩn bị dữ liệu tiền phân cụm	Chọn hiển thị trọng số, trọng số phân cụm tại các thuộc tính của bảng được hiện lên	Đạt
	Chọn hiển thị gợi ý, gợi ý gán nhãn được hiện lên	Đạt
	Tại cột Cấp độ dịch, chọn cấp độ dịch cho tỉnh An Giang là Cấp 1	Đạt
Phân cụm dữ liệu	Click vào nút "Phân cụm", các tỉnh thành sẽ được phân vào 3 cụm Cấp độ dịch và được sắp xếp từ Cấp 1 đến Cấp 3	Đạt
	Có thể thay đổi lại Cấp độ dịch của các tỉnh thành	Đạt
	Click vào nút Tải xuống file phân tích, file excel chứa bảng dữ liệu được phân tích sẽ được tải xuống	Đạt

Bảng 4.21: Bảng Kiểm thử chức năng Đánh giá tình hình dịch bệnh

4.4.3 Kiểm thử chức năng Đánh giá khả năng cung ứng VTYT

Chức năng	Test case	Kết quả
Khởi động màn hình	Chọn loại dịch bệnh Sốt xuất huyết, không có thông tin về số lượng VTYT và không thể click các nút "Chọn trọng số" hay "Phân cụm"	Đạt
	Chọn loại dịch bệnh sốt xuất huyết, dữ liệu về VTYT được hiển thị	Đạt
Chuẩn bị trọng số phân cụm	Click nút "Chọn trọng số", dialog chọn trọng số hiện ra	Đạt
	Trong input đầu tiên nhập số 0.3	Đạt
	Trong input đầu tiên nhập dãy "abc", không nhập được dãy đó vào ô input	Đạt
Chuẩn bị dữ liệu tiền phân cụm	Chọn hiển thị trọng số, trọng số phân cụm tại các thuộc tính của bảng được hiện lên	Đạt
	Chọn hiển thị gợi ý, gợi ý gán nhãn được hiện lên	Đạt
	Tại cột Khả năng cung ứng VTYT, chọn cho tỉnh An Giang là Có thể hỗ trợ	Đạt
Phân cụm dữ liệu	Click vào nút "Phân cụm", các tỉnh thành sẽ được phân vào 3 cụm Khả năng hỗ trợ là Cần được hỗ trợ, Tự cung ứng, Có thể hỗ trợ	Đạt
	Có thể thay đổi lại Khả năng hỗ trợ VTYT của các tỉnh thành	Đạt
	Click vào nút Tải xuống file phân tích, file excel chứa bảng dữ liệu được phân tích sẽ được tải xuống	Đạt

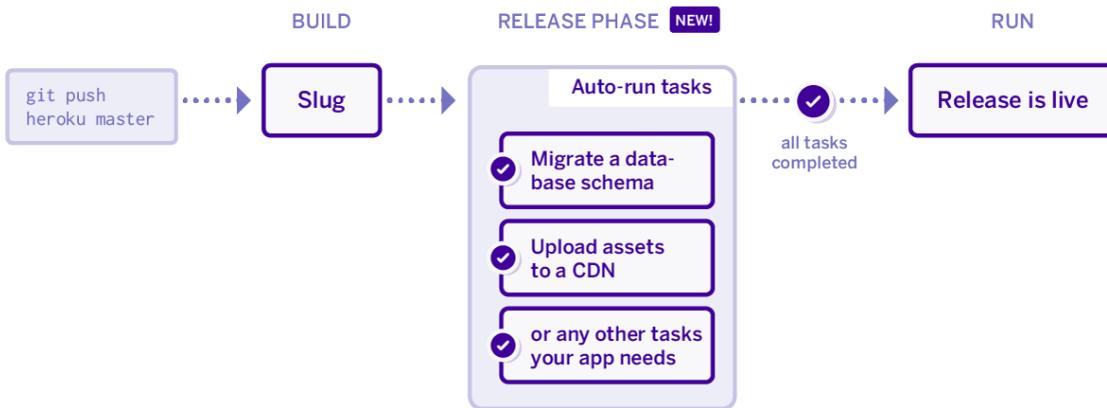
Bảng 4.22: Bảng Kiểm thử chức năng Đánh giá khả năng cung ứng VTYT

4.5 Triển khai ứng dụng

4.5.1 Mô hình triển khai

Ứng dụng bao gồm hai phần chính: server chạy trên Node.js Express và client chạy trên React.

Triển khai Server trên Heroku:



Hình 4.28: Quy trình triển khai server lên Heroku

Phần Server của ứng dụng được triển khai trên môi trường có khả năng chịu tải tốt và đủ tài nguyên để xử lý yêu cầu từ nhiều người dùng cùng một lúc. Chúng tôi quyết định sử dụng dịch vụ đám mây Heroku để triển khai Server vì Heroku hỗ trợ triển khai ứng dụng Node.js Express một cách dễ dàng và hiệu quả.

Khi triển khai Server trên Heroku, các gói phụ thuộc đã được cài đặt và cấu hình ứng dụng sao cho hoạt động ổn định và đáp ứng yêu cầu một cách linh hoạt. Heroku cung cấp tích hợp dễ dàng với Node.js Express và hỗ trợ tự động mở rộng (autoscaling), giúp đảm bảo ứng dụng có khả năng xử lý đồng thời nhiều yêu cầu từ người dùng một cách hiệu quả.

Triển khai Client trên Netlify:

Phần Client của ứng dụng được triển khai trong môi trường có khả năng lưu trữ tĩnh hiệu quả. Chúng tôi đã quyết định sử dụng dịch vụ Netlify để lưu trữ các tệp tĩnh của Client vì Netlify cung cấp môi trường lưu trữ tĩnh hiệu quả, giúp cải thiện hiệu suất của ứng dụng.

Với phần Client chạy trên React, trang web tĩnh đã tối ưu hóa, nhẹ nhàng và nhanh chóng. Netlify hỗ trợ lưu trữ các tệp tĩnh như HTML, CSS và JavaScript, giúp giảm thời gian tải trang và cải thiện trải nghiệm người dùng. Dịch vụ này cung cấp khả năng lưu trữ tĩnh hiệu quả, giúp dự án tiết kiệm chi phí và đảm bảo sự đáng tin cậy trong việc phục vụ tệp tĩnh cho người dùng.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

5.1.1 So sánh kết quả nghiên cứu với các sản phẩm tương tự

Sản phẩm hoàn thành là một ứng dụng web vô cùng hữu ích và đáng chú ý trong việc cung cấp thông tin về tình hình dịch bệnh và Vật tư y tế (VTYT) tại Việt Nam. Đối với người dùng thông thường, ứng dụng sẽ mang đến một loạt các chức năng hữu ích giúp họ nắm bắt thông tin quan trọng về dịch bệnh một cách nhanh chóng và hiệu quả.

Một trong những ưu điểm nổi bật của sản phẩm là khả năng cung cấp số liệu chi tiết và chính xác về dịch bệnh tại từng tỉnh thành phố trong cả nước. Thông tin này bao gồm số ca nhiễm, số ca tử vong, số ca hồi phục và cấp độ dịch bệnh hiện tại. Điều này giúp người dùng thông thường có cái nhìn tổng quan và chi tiết về tình hình dịch bệnh ở từng khu vực, từ đó họ có thể đưa ra quyết định thông minh về việc đi lại và giao tiếp với người khác.

Tiếp theo, ứng dụng cung cấp thông tin về VTYT tại mỗi tỉnh thành phố, bao gồm số lượng VTYT các loại mà mỗi địa phương sở hữu. Thông tin này đem đến cái nhìn rõ ràng về khả năng cung ứng VTYT của từng tỉnh thành, từ đó giúp người dùng thông thường đánh giá xem liệu các tỉnh có thể hỗ trợ VTYT cho các tỉnh khác hay không. Điều này đặc biệt quan trọng trong việc đảm bảo việc phân phối VTYT được thực hiện hiệu quả và đủ sức đáp ứng nhu cầu điều trị của các khu vực khác nhau.

Một chức năng hấp dẫn khác là khả năng đề xuất các tuyến đường vận chuyển VTYT tối ưu về mặt quãng đường vận chuyển. Ứng dụng sẽ tính toán và đưa ra các đề xuất về tuyến đường ngắn nhất và tiết kiệm nhất để vận chuyển VTYT giữa các tỉnh thành, giúp giảm thiểu thời gian và chi phí vận chuyển. Điều này đem lại lợi ích rất lớn trong việc tối ưu hóa việc phân phối VTYT và đảm bảo sự hỗ trợ hiệu quả giữa các khu vực đang cần.

Đối với người dùng là chuyên viên y tế, sản phẩm mang đến những công cụ hữu ích và mạnh mẽ để hỗ trợ công việc của họ. Chuyên viên y tế có thể lấy dữ liệu về tình hình dịch bệnh từ ứng dụng, từ đó thực hiện phân cụm các tỉnh thành phố dựa trên tình hình dịch bệnh thực tế. Điều này giúp họ hiểu rõ hơn về sự lan truyền của dịch bệnh và tập trung nguồn lực vào các khu vực đang có tình hình dịch bệnh phức tạp. Chính xác và hiểu biết rõ ràng về tình hình dịch bệnh cũng giúp chuyên viên y tế đưa ra các quyết định chính xác về việc điều trị và phòng ngừa dịch bệnh.

Ngoài ra, chuyên viên y tế cũng có khả năng lấy dữ liệu về VTYT từ ứng dụng. Họ có thể phân cụm các tỉnh thành phố dựa trên khả năng cung ứng VTYT, từ đó đưa ra các kế hoạch phân bổ nguồn lực y tế một cách chính xác và hiệu quả. Chuyên viên y tế cũng có thể thêm dữ liệu tĩnh và dữ liệu động vào cơ sở dữ liệu của hệ thống. Dữ liệu tĩnh là những loại dữ liệu không cần cập nhật thường xuyên như thông tin địa lý của các thành phố và thông tin về những loại dịch bệnh. Trong khi đó, dữ liệu động là các thông tin cần phải cập nhật hàng ngày như dữ liệu về các ca nhiễm bệnh, các ca hồi phục.

Tuy nhiên, sản phẩm cũng có một số hạn chế cần cân nhắc. Một trong số đó là việc không phải là nguồn tin chính thống từ chính phủ hoặc tổ chức y tế. Điều này có thể ảnh hưởng đến tính chính xác và đáng tin cậy của thông tin được cung cấp cho người dùng. Trong khi các trang web và ứng dụng chính thống như Trang tin tức của Bộ Y tế Việt Nam, Ứng dụng "Vietnam Health Declaration" và ứng dụng BlueZone có nguồn tin chính thống và đáng tin cậy từ chính phủ, sản phẩm của tôi lại cần đảm bảo rằng dữ liệu được cập nhật đúng đắn và đáng tin cậy.

Thêm vào đó, để đảm bảo tính chính xác và đáng tin cậy của thông tin, sản phẩm cần phải phụ thuộc vào dữ liệu cập nhật hàng ngày về dịch bệnh và VTYT. Điều này đòi hỏi sự duy trì và quản lý đáng tin cậy của cơ sở dữ liệu, cùng với các biện pháp bảo mật và bảo vệ dữ liệu để đảm bảo rằng thông tin không bị biến tướng hoặc lợi dụng một cách sai trái.

Ngoài ra, sản phẩm cần phải có giao diện người dùng thân thiện, dễ sử dụng và tương thích với nhiều thiết bị và trình duyệt khác nhau. Điều này giúp tăng tính tiện ích và hấp dẫn cho người dùng, đồng thời thu hút nhiều người sử dụng hơn.

Tóm lại, sản phẩm này có nhiều ưu điểm hấp dẫn như tích hợp đa chức năng, phân cụm thông minh và tích hợp dữ liệu động và tĩnh. Tuy nhiên, để cạnh tranh với các ứng dụng và trang web chính thống, sản phẩm cần đảm bảo tính chính xác và đáng tin cậy của thông tin cũng như sự phụ thuộc vào dữ liệu cập nhật hàng ngày. Đồng thời, cần đảm bảo giao diện người dùng thân thiện và tương thích để thu hút người dùng và nâng cao trải nghiệm của họ.

5.1.2 Đánh giá tổng quát

Trong suốt quá trình thực hiện Đề án Tốt nghiệp (ĐATN), tôi đã đối diện với nhiều thách thức và khó khăn, nhưng cũng đã đạt được nhiều thành công và đóng góp quan trọng cho dự án. Dưới đây là một tổng hợp chi tiết về những gì tôi đã làm được, chưa làm được, các đóng góp nổi bật và bài học kinh nghiệm mà tôi đã rút ra trong quá trình làm ĐATN.

Những gì tôi đã làm được:

Xác định đề tài và mục tiêu: Đầu tiên, tôi đã nghiên cứu và xác định đề tài "Phân bổ nguồn lực y tế phục vụ quá trình điều trị dịch bệnh". Tôi đã xác định mục tiêu cụ thể cho dự án, bao gồm việc phân tích dữ liệu về dịch bệnh và Vật tư y tế, thiết kế hệ thống và hiện thực ứng dụng.

Nghiên cứu lý thuyết và công nghệ: Tôi đã tiến hành nghiên cứu lý thuyết về các phương pháp phân cụm, thuật toán phân cụm bán giám sát và tối ưu chi phí. Tôi cũng tìm hiểu về các công nghệ để hiện thực phần server và client của ứng dụng.

Thiết kế và hiện thực hệ thống: Tôi đã thiết kế kiến trúc hệ thống và lựa chọn công nghệ phù hợp để hiện thực phần server bằng Node.js và Express, cũng như phần client bằng React. Tôi đã xây dựng các API để xử lý yêu cầu từ phía client và lưu trữ dữ liệu vào cơ sở dữ liệu.

Phân tích dữ liệu và đưa ra quyết định: Tôi đã thu thập và phân tích dữ liệu về dịch bệnh và Vật tư y tế từ các nguồn tin đáng tin cậy. Dựa trên các số liệu và thông tin này, tôi đã thực hiện phân cụm các tỉnh thành phố dựa trên tình hình dịch bệnh thực tế và phân cụm các tỉnh thành phố dựa trên khả năng cung ứng VTYT.

Tối ưu chi phí và đề xuất tuyến đường vận chuyển: Tôi đã áp dụng thuật toán tối ưu chi phí để đề xuất các tuyến đường vận chuyển VTYT tối ưu về mặt quãng đường. Việc này giúp tối thiểu hóa thời gian và chi phí vận chuyển và đảm bảo việc hỗ trợ VTYT được thực hiện hiệu quả.

Những gì tôi chưa làm được:

Dữ liệu thực tế: Một trong những thách thức lớn nhất mà tôi gặp phải là việc thu thập dữ liệu thực tế từ các nguồn tin chính thống và đáng tin cậy. Do giới hạn thời gian và tài nguyên, tôi chỉ có thể sử dụng dữ liệu mô phỏng để thực hiện phân cụm và tối ưu chi phí. Điều này có thể ảnh hưởng đến tính chính xác của kết quả đạt được.

Tối ưu hóa: Do tính phức tạp của bài toán và giới hạn thời gian, tôi chưa thể đạt được mức tối ưu hoàn hảo trong việc phân bổ nguồn lực y tế và đề xuất tuyến đường vận chuyển VTYT. Việc tối ưu hóa đòi hỏi sự nghiên cứu và thử nghiệm nhiều phương pháp khác nhau, điều này cần thời gian và kiến thức chuyên sâu hơn.

Các đóng góp nổi bật:

Phân cụm bán giám sát: Đóng góp quan trọng nhất của tôi là áp dụng thuật toán phân cụm bán giám sát để phân loại các tỉnh thành phố dựa trên tình hình dịch bệnh và khả năng cung ứng VTYT. Phương pháp này giúp tối ưu hóa quá trình phân bổ

nguồn lực y tế và đảm bảo việc hỗ trợ VTYT được thực hiện một cách hiệu quả.

Tối ưu chi phí và đề xuất tuyến đường vận chuyển: Tôi đã áp dụng thuật toán tối ưu chi phí để đề xuất các tuyến đường vận chuyển VTYT tối ưu về mặt quãng đường. Việc này giúp giảm thiểu thời gian và chi phí vận chuyển, đồng thời đảm bảo việc hỗ trợ VTYT đến các tỉnh thành phố khác được thực hiện một cách hiệu quả và nhanh chóng.

Bài học kinh nghiệm:

Xác định đề tài chính xác và khả thi: Việc xác định đề tài chính xác và khả thi là yếu tố quan trọng để đảm bảo quá trình thực hiện diễn ra suôn sẻ. Tôi nhận thức rõ rằng việc nghiên cứu kỹ lưỡng và xác định mục tiêu cụ thể cho dự án là điều cần thiết để đạt được kết quả tốt.

Hạn chế dữ liệu mô phỏng: Sử dụng dữ liệu mô phỏng có thể giới hạn tính chính xác và hiệu quả của kết quả đạt được. Tôi nhận thức rõ ràng việc thu thập dữ liệu thực tế từ các nguồn tin đáng tin cậy là rất quan trọng để đảm bảo tính chính xác và đáng tin cậy của sản phẩm.

Tối ưu hóa và nghiên cứu tiếp: Tối ưu hóa là một quá trình phức tạp và đòi hỏi thời gian và kiến thức chuyên sâu. Tôi nhận thức rõ ràng việc nghiên cứu và thử nghiệm nhiều phương pháp khác nhau là cần thiết để đạt được mức tối ưu hoàn hảo.

5.2 Hướng phát triển

Trong phần này, tôi sẽ trình bày định hướng công việc trong tương lai để hoàn thiện sản phẩm của mình, cũng như các nghiên cứu tiềm năng để nâng cao hiệu quả và cải thiện chức năng/nhiệm vụ đã thực hiện trong Đồ án Tốt nghiệp (ĐATN).

Để hoàn thiện các chức năng/nhiệm vụ đã làm, tôi nhận thấy có một số công việc cần thiết:

Mở rộng nguồn dữ liệu thực tế: Để tăng tính chính xác và đáng tin cậy của sản phẩm, tôi cần thu thập dữ liệu thực tế từ các nguồn tin chính thống và đáng tin cậy. Điều này bao gồm dữ liệu về dịch bệnh, thông tin VTYT, số liệu về các ca nhiễm bệnh và các ca hồi phục, giúp đảm bảo mô hình và kết quả phân cụm được thể hiện trên nền dữ liệu thực tế.

Tối ưu hóa mô hình và thuật toán: Mặc dù đã sử dụng giải thuật tham lam để giải quyết bài toán vận chuyển VTYT tối ưu, tôi sẽ tiếp tục nghiên cứu và thử nghiệm các thuật toán khác nhau để đạt được mức tối ưu cao hơn. Tối ưu hóa mô hình và thuật toán giúp tăng tính hiệu quả và chính xác của việc phân bổ nguồn lực y tế và đề xuất tuyến đường vận chuyển VTYT.

Tăng cường tính tương tác và giao diện người dùng: Để tăng sự hấp dẫn và tiện ích cho người dùng, tôi sẽ cải thiện giao diện người dùng của ứng dụng. Tạo ra một giao diện thân thiện, dễ sử dụng và có tính tương tác cao giúp người dùng dễ dàng tiếp cận thông tin về dịch bệnh và VTYT một cách trực quan và dễ hiểu.

Hỗ trợ đa nền tảng: Hiện tại, sản phẩm của tôi được triển khai trên môi trường đám mây Heroku và Netlify. Tuy nhiên, để đáp ứng nhu cầu ngày càng cao của người dùng, tôi sẽ nghiên cứu và triển khai sản phẩm trên nhiều nền tảng khác nhau, bao gồm ứng dụng di động (Android và iOS) và các nền tảng máy tính khác.

Sau khi hoàn thiện các chức năng/nhiệm vụ đã thực hiện, tôi nhìn thấy có một số hướng đi mới có thể nâng cao hiệu quả và cải thiện sản phẩm:

Tối ưu hóa giải thuật và mô hình: Hiện tại, tôi đã sử dụng giải thuật tham lam để giải quyết bài toán vận chuyển VTYT tối ưu về mặt quãng đường. Tuy nhiên, trong tương lai, tôi sẽ nghiên cứu và áp dụng các giải thuật tối ưu hóa khác nhau như tối ưu hóa tiến hóa, tối ưu hóa đồ thị, hoặc các phương pháp tối ưu hóa dựa trên học máy. Điều này giúp cải thiện hiệu suất của việc phân bổ VTYT và đề xuất các tuyến đường vận chuyển chính xác hơn.

Xây dựng mô hình dự đoán: Để nâng cao tính ứng dụng và đáng tin cậy của sản phẩm, tôi sẽ phát triển một mô hình dự đoán dựa trên dữ liệu lịch sử về dịch bệnh và VTYT. Mô hình dự đoán này sẽ sử dụng các thuật toán học máy để dự đoán tình hình dịch bệnh trong tương lai và nhu cầu VTYT của từng tỉnh thành. Kết quả dự đoán giúp chuyên viên y tế và nhà quản lý y tế đưa ra quyết định và kế hoạch hỗ trợ VTYT một cách chính xác và hiệu quả hơn.

Tích hợp hệ thống định vị GPS: Để tăng cường tính chính xác và hiệu quả của việc vận chuyển VTYT, tôi sẽ tích hợp hệ thống định vị GPS vào ứng dụng. Điều này cho phép theo dõi thời gian thực các tuyến đường vận chuyển và điều chỉnh khi cần thiết để tối ưu hóa quá trình vận chuyển. Người dùng và chuyên viên y tế có thể xem thông tin vận chuyển một cách trực quan và chính xác thông qua hệ thống định vị GPS.

Cập nhật dữ liệu động thường xuyên: Để đảm bảo tính chính xác và đáng tin cậy của dữ liệu, tôi sẽ thực hiện cập nhật dữ liệu động như số liệu về dịch bệnh và VTYT hàng ngày. Việc cập nhật thông tin thường xuyên giúp cung cấp dữ liệu thời gian thực và đáng tin cậy cho người dùng và chuyên viên y tế, từ đó đưa ra các quyết định và kế hoạch hỗ trợ VTYT một cách hiệu quả và nhanh chóng.

Mở rộng tính năng cho người dùng thông thường: Để tăng tính hấp dẫn và tiện ích cho người dùng thông thường, tôi sẽ mở rộng tính năng cho phép họ xem thông

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

tin chi tiết về dịch bệnh, cấp độ dịch và VTYT. Tôi cũng sẽ tăng cường tính tương tác và giao diện người dùng, tạo ra một giao diện thân thiện, dễ sử dụng và có tính tương tác cao.

Tóm lại, trong quá trình thực hiện ĐATN, tôi đã làm được rất nhiều công việc quan trọng để phân tích và giải quyết bài toán phân bổ nguồn lực y tế và vận chuyển VTYT hiệu quả. Tuy nhiên, còn nhiều việc phải làm và cải thiện để đưa sản phẩm của mình lên một tầm cao mới. Những định hướng công việc và cải tiến trong tương lai sẽ giúp tôi đạt được mục tiêu tối ưu hóa sản phẩm và đem lại lợi ích tối đa cho người dùng và ngành y tế. Tôi hy vọng rằng sản phẩm của mình có thể đóng góp ý nghĩa trong việc đối phó và quản lý dịch bệnh, đồng thời cải thiện hiệu quả trong việc phân bổ và vận chuyển VTYT.

TÀI LIỆU THAM KHẢO

- [1] M.-K. T. Tran Dinh Khang and M. Fowler, *A Novel Semi-Supervised Fuzzy C-Means Clustering Algorithm using Multiple Fuzzification Coefficients*. Hanoi University of Science and Technology, 2021.
- [2] J. Punia, *Dijkstra's algorithm*. [Online]. Available: <https://www.scaler.com/topics/data-structures/dijkstra-algorithm/> (visited on 07/20/2023).
- [3] *Express docs*. [Online]. Available: <https://expressjs.com/en/guide/routing.html> (visited on 05/01/2023).
- [4] *React docs*. [Online]. Available: <https://legacy.reactjs.org/docs/getting-started.html> (visited on 04/23/2023).
- [5] *Axios docs*. [Online]. Available: <https://axios-http.com/docs/intro> (visited on 05/21/2023).