

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN**



MÔN: ĐỒ ÁN TỐT NGHIỆP

Đề tài:

**Phát triển ứng dụng sử dụng giải pháp GIS
phát triển điểm bán**

Giảng viên hướng dẫn: TS. **Hoàng Văn Thông**

Sinh viên thực hiện: **Nguyễn Đức Duy** - 171202884

Lớp: Công nghệ thông tin 4-K58

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN**



MÔN: ĐỒ ÁN TỐT NGHIỆP

Đề tài:

**Phát triển ứng dụng sử dụng giải pháp GIS
phát triển điểm bán**

Giảng viên hướng dẫn: TS. **Hoàng Văn Thông**

Sinh viên thực hiện: **Nguyễn Đức Duy** - 171202884

Lớp: Công nghệ thông tin 4-K58

LỜI CẢM ƠN

Để hoàn thành đồ án tốt nghiệp này, em xin chân thành gửi lời cảm ơn đến TS.Hoàng Văn Thông, giảng viên trực tiếp phụ trách hướng dẫn đề tài đồ án tốt nghiệp. Sự hướng dẫn tận tình của thầy đã giúp em có thêm nhiều kiến thức, khắc phục được những lỗi em thường mắc phải để đồ án được hoàn thiện tốt hơn. Qua những hướng dẫn tận tâm của thầy đã giúp chúng em hiểu sâu hơn về những kiến thức đã học, và trau dồi thêm những kiến thức mới bô ích.

Em xin chân thành gửi lời cảm ơn quý thầy cô trong bộ môn cũng như quý thầy cô trong trường đã tận tình giúp đỡ và trang bị cho em những kiến thức cơ bản, cần thiết để em có thể hoàn thành tốt đồ án tốt nghiệp này.

Em cũng xin gửi lời cảm ơn đến các anh chị trong Ekgis đã tạo điều kiện cho em có môi trường học tập và nghiên cứu để hoàn thành đồ án trong thời gian qua.

Mặc dù có nhiều cố gắng để thực hiện đồ án tốt nghiệp một cách hoàn chỉnh nhất. Tuy nhiên do năng lực còn nhiều hạn chế nên không thể tránh khỏi những thiếu sót nhất định. Em rất mong nhận được sự đóng góp của quý thầy cô để đồ án tốt nghiệp của em được hoàn chỉnh hơn.

Em xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Ứng dụng và xu thế phát triển của **GIS** nhìn từ tương lai và sự phát triển của công nghệ này vào các ứng dụng trong đời sống xã hội, khoa học ngày càng rộng với mức độ quan tâm ngày càng lớn.

Cho đến nay **GIS** và các ứng dụng của nó đang phát triển nhanh chóng: các dịch vụ gắn với địa điểm, kết hợp GIS&GPS hiển thị vị trí nhà hàng, khách sạn, tìm đường được cài vào thiết bị cầm tay (PDA, laptop, mobilephone); các ứng dụng trên bản đồ web (Google Map, Map Quest, Yahoo!, Maps...), truy cập tự do đến kho dữ liệu địa lý, đặc biệt là không ảnh, cho phép tạo lập các ứng dụng của riêng mình; nghiên cứu sự biến động các quá trình trên trái đất qua năm tháng (sự mở rộng bờ cõi trong lịch sử, biến động đường bờ biển, độ che phủ rừng), tác động của biến đổi khí hậu và mực nước biển dâng.

Nhận thức được tầm quan trọng và tính hữu ích của **GIS**, **GIS** tại Việt Nam đã và đang được ưu tiên đầu tư phát triển và nâng cấp với sự quan tâm lớn lao của các hệ thống thông tin như: Hệ thống thông tin địa lý Tp.Hồ Chí Minh; Hệ thống thông tin hiện trạng công nghệ và môi trường của tỉnh Đồng Nai – DONAGIS; Hệ thống thông tin địa lý phục vụ quản lý của tỉnh Bến Tre – BETEGIS; Quảng Nam – QANAGIS; Phục vụ quản lý nông nghiệp của thành phố Đà Nẵng – DANAGIS; Hệ thống thông tin địa lý phục vụ phát triển kinh tế tỉnh Bình Dương – BIDOGIS...

Thấy được những ích lợi và tính thiết thực của **GIS**, em quyết định chọn đề tài là “**Phát triển ứng dụng sử dụng giải pháp GIS phát triển điểm bán**” để hiểu rõ hơn và phát triển **GIS**. Ngôn ngữ sử dụng để xây dựng module là TypeScript, hệ quản trị cơ sở dữ liệu SQL Server.

MỤC LỤC

LỜI CẢM ƠN	3
LỜI NÓI ĐẦU	4
DANH MỤC TỪ VIẾT TẮT	7
DANH MỤC BẢNG BIỂU	8
DANH MỤC HÌNH ẢNH	9
CHƯƠNG 1. GIỚI THIỆU CHUNG	11
1.1. Mục đích	11
1.2. Phạm vi	11
1.3. Cơ sở lý thuyết	11
1.3.1. Tổng quan về GIS	11
1.3.2. Angular	15
1.3.3. Framework ASP.Net Zero	17
1.4. Tổng quát về ngôn ngữ mô hình hóa UML	22
1.4.1. Các thành phần ngôn ngữ UML	22
1.4.2. Biểu đồ	22
1.4.3. Biểu đồ Use case (Use Case Diagram)	22
1.4.4. Biểu đồ lớp (Class Diagram)	22
1.4.5. Biểu đồ trình tự (Sequence Diagram)	23
1.4.6. Tác nhân (Actor)	23
1.4.7. Ca sử dụng (Use case)	23
1.4.8. Lớp (Class)	23
1.4.9. Thành phần	24
1.4.10. Các liên kết	24
1.4.11. Các công nghệ khác	25
CHƯƠNG 2. KHẢO SÁT VÀ THIẾT KẾ HỆ THỐNG	26
2.1. Khảo sát	26
2.2. Kết luận	27
2.3. Bài toán đặt ra	27
2.3.1. Bài toán đặt ra	27
2.3.2. Mô hình	30
2.3.3. Quy trình nghiệp vụ	30

2.3.4. Mô hình use case.....	31
2.3.5. Danh sách các tác nhân.....	33
2.3.6. Danh sách use case và mô tả.....	34
2.3.7. Đặc tả các yêu cầu chức năng.....	35
2.4. Mô hình quan hệ thực thể.....	46
2.4.1. Mô tả các bảng.....	46
2.4.2. Sơ đồ phân rã chức năng	52
2.4.3. Sơ đồ thư thế quan hệ	53
2.4.4. Biểu đồ lớp	53
CHƯƠNG 3. CÀI ĐẶT VÀ THỬ NGHIỆM WEBSITE	54
3.1. Môi trường cài đặt.....	54
3.2. Giao diện và chức năng.....	54
3.2.1. Trang đăng nhập quản lý dịch vụ.....	54
3.2.2. Trang quản lý dịch vụ	54
3.2.3. Trang đăng nhập	55
3.2.4. Quản lý ứng dụng.....	56
3.2.5. Giao diện chính website	57
3.2.6. Thiết lập cài đặt	58
3.2.7. Tìm vị trí để thêm điểm	59
3.2.8. Nhập điểm bằng file Excel hoặc file CSV	60
3.2.9. Thêm điểm bằng cách chấm trên bản đồ	62
3.2.10. Đánh giá các điểm đã thêm.....	62
3.2.11. Xác định Site mới	65
3.2.12. Cập nhật điểm bán	67
3.2.13. Một số chức năng khác	68
CHƯƠNG 4. KẾT LUẬN	70
4.1. Kết quả đạt được	70
4.2. Hạn chế.....	70
4.3. Hướng phát triển	70
DANH MỤC TÀI LIỆU THAM KHẢO	71

DANH MỤC TỪ VIẾT TẮT

STT	Thuật ngữ, từ viết tắt	Giải thích	Ghi chú
1	GIS	Geographic Information System	
2	UML	Unified Modeling Language	
3	CSDL	Cơ sở dữ liệu	
4	Tenant	Tổ chức	
5	UC	Use case	
6	Site selection	Lựa chọn điểm bán	
7	API	Application Programming Interface	
8	HTML	Hypertext Markup Language	
9	CSS	Cascading Style Sheets	
10	JS	Javascript	
11	HTTP	Hypertext Transfer Protocol	
12	DOM	Document Object Model	
13	ABP	ASP.NET Boilerplate.	
14	UI	User Interface	
15	UX	User Experience	

DANH MỤC BẢNG BIỂU

Bảng 2.1 Dang sách use case và mô tả	34
Bảng 2.2 Đặc tả use case đăng nhập	35
Bảng 2.3 Đặc tả use case đăng ký	36
Bảng 2.4 Đặc tả use case đăng xuất	37
Bảng 2.5 Đặc tả use case thêm điểm bán	38
Bảng 2.6 Đặc tả use case cập nhật danh sách điểm bán	39
Bảng 2.7 Đặc tả use case cập nhật chi tiết điểm bán	39
Bảng 2.8 Đặc tả use case cập nhật cập nhập file	40
Bảng 2.9 Đặc tả use case xem file	41
Bảng 2.10 Đặc tả use case tải file	41
Bảng 2.11 Đặc tả use case tải file	42
Bảng 2.12 Đặc tả use case tải file	43
Bảng 2.13 Đặc tả use case tải file	44
Bảng 2.15 Đặc tả use case tải file	45
Bảng 2.16 Bảng thống kê(SiteSelectionAnalyst)	46
Bảng 2.17 Bảng nhóm(Group)	46
Bảng 2.18 Bảng điểm bán(SiteSelectionLocation)	47
Bảng 2.19 Bảng cài đặt mặc định (SiteSelectionSetting)	47
Bảng 2.20 Bảng vị trí điểm (Poi)	48
Bảng 2.21 Bảng quản lý người dùng (AbpUser)	49
Bảng 2.22 Bảng quản lý Tenant (AbpTenant)	50
Bảng 2.23 Bảng danh sách quyền (ApbRoles)	50
Bảng 2.14 Bảng quan hệ người dùng phòng ban (AbpUserOrganizationUnits)	51
Bảng 2.15 Bảng quan hệ người dùng quyền (AbpUserRoles)	51

DANH MỤC HÌNH ẢNH

Hình 1.1. Kiến trúc ASP.NET Zero [2].	18
Hình 1.2. Mô hình kiến trúc ASP.NET Boilerplate [12].	19
Hình 1.3 Mô hình hoạt động của Unit of work.....	21
Hình 2.1. Các phần tử liên kết.	24
Hình 2.2 Mô hình quy trình lựa chọn điểm bán.....	29
Hình 2.3 Mô hình quy trình đánh giá điểm bán.....	30
Hình 2.4 Mô hình phát triển ứng dụng điểm bán.....	30
Hình 2.5 Biểu đồ phân làn quy trình trọn điểm bán	31
Hình 2.6 Biểu đồ phân làn quy trình đánh giá điểm bán	31
Hình 2.7 Sơ đồ use case tổng quát.....	32
Hình 2.8 Sơ đồ use case người dùng	32
Hình 2.9 Sơ đồ use case quản lý tenant	33
Hình 2.10 Sơ đồ quản lý phòng ban	33
Hình 2.11 Tác nhân hệ thống.....	34
Hình 2.12 Biểu đồ trình tự use case đăng nhập	36
Hình 2.13 Biểu đồ trình tự use case đăng ký	37
Hình 2.14 Biểu đồ trình tự use case thêm điểm bán	39
Hình 2.15 Biểu đồ trình tự use case tìm kiếm điểm bán.....	43
Hình 2.16 Sơ đồ phân rã chức năng.....	52
Hình 2.17 Sơ đồ thực thể quan hệ.....	53
Hình 2.18 Biểu đồ lớp.....	53
Hình 3.1 Trang đăng nhập quản lý dịch vụ	54
Hình 3.2 Danh sách dịch vụ.....	55
Hình 3.3 Trai nghiệm dịch vụ	55
Hình 3.4 Giao diện đăng nhập	56
Hình 3.5 Giao diện quản lý ứng dụng lưới	56
Hình 3.6. Form Tạo thư mục.	57
Hình 3.7. Form Tạo ứng dụng Shortlist.....	57
Hình 3.8 Giao diện chính	58

Hình 3.9 Giao diện cài đặt	58
Hình 3.10 Giao diện từng chức năng cài đặt	59
Hình 3.11 Tự động nhập	59
Hình 3.12 Sau khi thêm điểm bằng tìm kiếm	60
Hình 3.13 Nhập file Excel/CSV	60
Hình 3.14 Chọn Sheet.....	61
Hình 3.15 Nhập thông tin hiển thị	61
Hình 3.16 Phân nhóm điểm bán.....	62
Hình 3.17 Thêm nhóm bằng chấm trên bản đồ	62
Hình 3.18 Cửa sổ thiết lập đánh giá.....	63
Hình 3.19 Kết quả đánh giá điểm bán	63
Hình 3.20 Bản đồ nhiệt	64
Hình 3.21 Bản đồ phủ	64
Hình 3.22 Bản đồ theo độ phủ	64
Hình 3.23 Xác định điểm bán	65
Hình 3.24 Xác định vùng bao tại khu vực cầu giấy.....	66
Hình 3.25 Vẽ vùng bao	66
Hình 3.26 Đánh giá điểm bán	67
Hình 3.27 Thông tin điểm bán	67
Hình 3.28 Cập nhật file theo từng điểm bán.....	68
Hình 3.29 Tạo nhóm	68
Hình 3.30 Sắp xếp điểm bán	69
Hình 3.31 Định vị điểm bán.....	69

CHƯƠNG 1. GIỚI THIỆU CHUNG

1.1. Mục đích

- Cung cấp giải pháp từ khảo sát, xây dựng cơ sở dữ liệu, thành lập bản đồ các tỉnh thành trên cả nước
- Cung cấp thông tin hỗ trợ ra quyết định mua bán

1.2. Phạm vi

Hệ thống dành cho doanh nghiệp có nhu cầu tạo mới, đánh giá khu vực có tiềm năng sử dụng để nhằm mục đích cho phát triển kinh doanh.

1.3. Cơ sở lý thuyết

1.3.1. Tổng quan về GIS

GIS (Geographic Information System), là một tập hợp có tổ chức, bao gồm hệ thống phần cứng, phần mềm máy tính, dữ liệu địa lý và con người, được thiết kế nhằm mục đích nắm bắt, lưu trữ, cập nhật, điều khiển, phân tích, và hiển thị tất cả các dạng thông tin liên quan đến vị trí địa lý.

Các thành phần của GIS:

- **Phần cứng:** bao gồm máy tính và các thiết bị ngoại vi.
- **Phần mềm:** là bộ não của hệ thống, phần mềm GIS rất đa dạng và có thể chia làm 3 nhóm (nhóm phần mềm quản đồ họa, nhóm phần mềm quản trị bản đồ và nhóm phần mềm quản trị, phân tích không gian). Các thành phần chính trong phần mềm GIS là:
 - + Công cụ nhập và thao tác trên các thông tin địa lý.
 - + Hệ quản trị cơ sở dữ liệu
 - + Công cụ hỗ trợ hỏi đáp, phân tích và hiển thị địa lý.
 - + Giao diện đồ họa người - máy để truy cập các công cụ dễ dàng.
- **Dữ liệu:** bao gồm dữ liệu không gian (dữ liệu bản đồ) và dữ liệu thuộc tính (dữ liệu phi không gian). Dữ liệu không gian miêu tả vị trí địa lý của đối tượng trên bề mặt Trái đất. Dữ liệu thuộc tính miêu tả các thông tin liên quan đến đối tượng, các thông tin này có thể được định lượng hay định tính.
- **Phương pháp:** một phần quan trọng để đảm bảo sự hoạt động liên tục và có hiệu quả của hệ thống phục vụ cho mục đích của người sử dụng.

- **Con người:** Trong GIS, thành phần con người là thành phần quan trọng nhất bởi con người tham gia vào mọi hoạt động của hệ thống GIS (từ việc xây dựng cơ sở dữ liệu, việc tìm kiếm, phân tích dữ liệu ...). Có 2 nhóm người quan trọng là người sử dụng và người quản lý GIS.

GIS lưu giữ thông tin về thế giới thực dưới dạng tập hợp các lớp chuyên đề có thể liên kết với nhau nhờ các đặc điểm địa lý. Điều này đơn giản nhưng vô cùng quan trọng và là một công cụ đa năng đã được chứng minh là rất có giá trị trong việc giải quyết nhiều vấn đề thực tế, từ thiết lập tuyến đường phân phối của các chuyến xe, đến lập báo cáo chi tiết cho các ứng dụng quy hoạch, hay mô phỏng sự lưu thông khí quyển toàn cầu.

Tham khảo địa lý: Các thông tin địa lý hoặc chứa những tham khảo địa lý hiện (chẳng hạn như kinh độ, vĩ độ hoặc tọa độ lối quốc gia), hoặc chứa những tham khảo địa lý ẩn (như địa chỉ, mã bưu điện, tên vùng điều tra dân số, bộ định danh các khu vực rừng hoặc tên đường). Mã hóa địa lý là quá trình tự động thường được dùng để tạo ra các tham khảo địa lý hiện (vị trí bội) từ các tham khảo địa lý ẩn (là những mô tả, như địa chỉ). Các tham khảo địa lý cho phép định vị đối tượng (như khu vực rừng hay địa điểm thương mại) và sự kiện (như động đất) trên bề mặt quả đất phục vụ mục đích phân tích.

Mô hình Vector và Raster: Hệ thống thông tin địa lý làm việc với hai dạng mô hình dữ liệu địa lý khác nhau về cơ bản - mô hình vector và mô hình raster. Trong mô hình vector, thông tin về điểm, đường và vùng được mã hóa và lưu dưới dạng tập hợp các tọa độ x,y. Vị trí của đối tượng điểm, như lỗ khoan, có thể được biểu diễn bởi một tọa độ đơn x,y. Đối tượng dạng đường, như đường giao thông, sông suối, có thể được lưu dưới dạng tập hợp các tọa độ điểm. Đối tượng dạng vùng, như khu vực buôn bán hay vùng lưu vực sông, được lưu như một vòng khép kín của các điểm tọa độ.

Mô hình vector rất hữu ích đối với việc mô tả các đối tượng riêng biệt, nhưng kém hiệu quả hơn trong miêu tả các đối tượng có sự chuyển đổi liên tục như kiểu đất hoặc chi phí ước tính cho các bệnh viện. Mô hình raster được phát triển cho mô phỏng các đối tượng liên tục như vậy. Một ảnh raster là một tập hợp các ô lưới. Cả mô hình vector và raster đều được dùng để lưu dữ liệu địa lý với những ưu điểm, nhược điểm riêng. Các hệ GIS hiện đại có khả năng quản lý cả hai mô hình này.

GIS có 5 chức năng chủ yếu:

- **Thu thập dữ liệu:** là công việc khó khăn và nặng nề nhất trong quá trình xây dựng một ứng dụng GIS. Các dữ liệu được thu thập từ nhiều nguồn khác nhau như dữ liệu đo đạc từ thực địa, dữ liệu từ các loại bản đồ, dữ liệu thống kê...
- **Thao tác dữ liệu:** vì các dữ liệu được thu thập từ nhiều nguồn có định dạng khác nhau và có những trường hợp các dạng dữ liệu đòi hỏi được chuyển đổi và thao tác theo một số cách để tương thích với hệ thống. Ví dụ: các thông tin địa lý có giá trị biểu diễn khác nhau tại các tỷ lệ khác nhau (lớp dân cư trên bản đồ địa chính được thể hiện chi tiết hơn trong bản đồ địa hình). Trước khi các thông tin này được tích hợp với nhau thì chúng phải được chuyển về cùng một tỷ lệ (cùng mức độ chi tiết hoặc mức độ chính xác). Đây có thể chỉ là sự chuyển đổi tạm thời cho mục đích hiển thị hoặc cố định cho yêu cầu phân tích.
- **Quản lý dữ liệu:** là một chức năng quan trọng của tất cả các hệ thông tin địa lý. Hệ thống thông tin địa lý phải có khả năng điều khiển các dạng khác nhau của dữ liệu đồng thời quản lý hiệu quả một khối lượng lớn dữ liệu với một trật tự rõ ràng. Một yếu tố quan trọng của GIS là khả năng liên kết hệ thống giữa việc tự động hóa bản đồ và quản lý cơ sở dữ liệu (sự liên kết giữa dữ liệu không gian và thuộc tính của đối tượng). Các dữ liệu thông tin mô tả cho một đối tượng bất kỳ có thể liên hệ một cách hệ thống với vị trí không gian của chúng. Sự liên kết đó là một ưu thế nổi bật của việc vận hành GIS.
- **Hỏi đáp và phân tích dữ liệu:** Khi đã xây dựng được một hệ thống cơ sở dữ liệu GIS thì người dùng có thể hỏi các câu hỏi đơn giản như:
 - + Thông tin về thửa đất: Ai là chủ sở hữu của mảnh đất?
 - + Tìm đường đi ngắn nhất giữa hai vị trí A và B?
 - + Thống kê số lượng cây trồng trên tuyến phố?
 - + Hay xác định mật độ diện tích trồng cây xanh trong khu vực đô thị?...
- **Hiển thị dữ liệu:** GIS cho phép hiển thị dữ liệu tốt nhất dưới dạng bản đồ hoặc biểu đồ. Ngoài ra còn có thể xuất dữ liệu thuộc tính ra các bảng excel, tạo các bản báo cáo thống kê, hay tạo mô hình 3D, và nhiều dữ liệu khác.

GIS có các phần mềm thông dụng như: MapInfo, ArcGis, Mapping Ofice, GS – Map, MicroStation Geographics, ArcInfo, ArcView, ...

Ứng dụng của GIS:

- **Môi trường:** Ở mức đơn giản nhất là có thể dùng hệ thông tin địa lý GIS để đánh giá môi trường dựa vào vị trí và thuộc tính. Ứng dụng cao cấp hơn là chúng ta có thể sử

dụng GIS để mô hình hóa các tiến trình xói mòn đất cũng như sự ô nhiễm môi trường dựa vào khả năng phân tích của GIS.

- **Khí tượng thủy văn:** Hệ thông tin địa lý GIS có thể nhanh chóng đáp ứng phục vụ cho các công tác dự báo thiên tai lũ lụt cũng như các công tác dự báo vị trí của bão và các dòng chảy...
- **Nông nghiệp:** GIS có thể phục vụ cho các công tác quản lý sử dụng đất, nghiên cứu về đất trồng, có thể kiểm tra được nguồn nước.
- **Dịch vụ tài chính:** GIS được sử dụng trong lĩnh vực dịch vụ tài chính tương tự như là một ứng dụng đơn lẻ. Nó đã từng được áp dụng cho việc xác định vị trí những chi nhánh mới của Ngân hàng. Hiện nay việc **sử dụng GIS** đang tăng lên trong lĩnh vực này, nó là một công cụ đánh giá rủi ro và mục đích bảo hiểm, xác định với độ chính xác cao hơn những khu vực có độ rủi ro lớn nhất hay thấp nhất. Lĩnh vực này đòi hỏi những dữ liệu cơ sở khác nhau như là hình thức vi phạm luật pháp, địa chất học, thời tiết và giá trị tài sản
- **Y tế:** GIS có thể dẫn đường nó có thể đưa ra được lộ trình giữa xe cấp cứu và bệnh nhân cần cấp cứu qua đó giúp xe cấp cứu có thể nhanh nhất đến với vị trí của bệnh nhân làm tăng cơ hội sống sót của người bệnh, ngoài ra nó còn được dùng trong nghiên cứu các dịch bệnh nó có thể phân tích nguyên nhân bùng phát và lan truyền của bệnh dịch.
- **Giao thông:** Hệ thông tin địa lý GIS có thể được ứng dụng trong định vị trong vận tải hàng hóa, cũng như việc xác định lộ trình đường đi ngắn nhất, cũng như việc quy hoạch giao thông.
- **Thị trường bán lẻ:** Phần lớn siêu thị vùng ngoại ô được xác định vị trí với sự trợ giúp của GIS. **GIS** thường lưu trữ những dữ liệu về kinh tế-xã hội của khách hàng trong một vùng nào đó. Một vùng thích hợp cho việc xây dựng một siêu thị có thể được tính toán bởi thời gian đi đến siêu thị, và mô hình hóa ảnh hưởng của những siêu thị cạnh tranh. GIS cũng được dùng cho việc quản lý tài sản và tìm đường phân phối hàng ngắn nhất.
- Lợi ích và những mặt hạn chế của việc sử dụng GIS:
- **Lợi ích của việc sử dụng GIS:**
 - + Là cách tiết kiệm chi phí và thời gian trong việc lưu trữ số liệu.
 - + Có thể thu nhập số liệu với số lượng lớn.
 - + Số liệu lưu trữ có thể được cập nhật một cách dễ dàng.
 - + Chất lượng số liệu được quản lý, xử lý và hiệu chỉnh tốt.
 - + Dễ dàng truy cập, phân tích số liệu từ nhiều nguồn khác nhau.

- **Đặc biệt trong nông nghiệp, GIS có các ưu điểm:**

- + Là một công cụ khá mạnh trong việc lưu trữ và diễn đạt số liệu đặc biệt là các bản đồ.
- + Có thể cho ra những kết quả dưới dạng khác nhau như bản đồ, các bản thống kê.
- + Hỗ trợ các nhà khoa học đặc biệt về lĩnh vực nghiên cứu hệ thống canh tác, đánh giá đất đai, khả năng thích nghi của các kiểu sử dụng đất, quản lý và xử lý các bản đồ giao thửa trong quản lý đất đai.

- **Hạn chế:**

- + Chí phí và những vấn đề kỹ thuật đòi hỏi trong việc chuẩn bị lại các số liệu thô hiện có.
- + Đòi hỏi nhiều kiến thức của các kỹ thuật cơ bản về máy tính, yêu cầu lớn về nguồn tài chính BAN đầu.
- + Chi phí của việc mua sắm, lắp đặt thiết vị và phần mềm GIS khá cao.
- + Trong một số lĩnh vực ứng dụng, hiệu quả tài chính thu lại thấp.

1.3.2. Angular

- *Khái niệm cơ bản Angular.*

Angular là một javascript framework do google phát triển để xây dựng các Single Page Application (SPA) bằng JavaScript , HTML và TypeScript . Angular cung cấp các tính năng tích hợp cho animation , http service và có các tính năng như auto-complete , navigation , toolbar , menus ,.... Code được viết bằng TypeScript , biên dịch thành JavaScript và hiển thị tương tự trong trình duyệt.

Kiến trúc của ứng dụng Angular dựa trên khái niệm cơ bản nhất. Những khối cơ bản được xây dựng là NgModules, nó cung cấp một môi trường cho components. Một ứng dụng Angular được xác định bởi một bộ NgModules. Một ứng dụng luôn luôn có ít nhất một module gốc cho phép bootstrap và thường có nhiều module tính năng. Component định nghĩa giao diện, nó tập hợp các yếu tố màn hình mà Angular có thể chọn và sửa đổi theo logic và dữ liệu chương trình. Component sử dụng dịch vụ, nó cung cấp hàm cụ thể. Các nhà cung cấp dịch vụ có thể được đưa vào các thành phần dưới dạng phụ thuộc, làm cho mã được module hóa để có thể sử dụng lại và hiệu quả.

- *Modules.*

Mỗi ứng dụng Angular đều có một root module, thông thường nó được gọi là AppModule, nó cung cấp cơ chế khởi chạy ứng dụng. Một ứng dụng thường có nhiều modules làm những việc khác nhau.

Tổ chức mã thành các modules có chức năng riêng biệt giúp quản lý phát triển các ứng dụng phức tạp và thiết kế để có thể sử dụng lại. Ngoài ra kỹ thuật này cho phép tải các module theo yêu cầu để giảm thiểu việc tải các modules không cần thiết.

- *Components.*

Angular Component là một class nó được hỗ trợ bởi @Component decorator. Component điều khiển một phần của UI. Ứng dụng Task List ở trên có 3 component. TaskComponent hiển thị danh sách task, còn TaskAddComponent giúp chúng ta tạo mới task. RootComponent là component cha của các component này chỉ hiển thị tên của ứng dụng. Component được tạo ra trong ứng dụng mới nhìn sẽ như sau:

Component có 4 phần như sau:

- + Import statement
- + Class
- + Template
- + Metadata

Lệnh import các thành phần liên quan được yêu cầu bởi Component này. Class chứa các logic của ứng dụng. Nó được bổ sung thông tin bởi @Component decorator.

- *Templates, directives, và data binding.*

Một template HTML với đánh dấu Angular có thể sửa đổi các phần tử HTML trước khi chúng được hiển thị. Template directive cung cấp logic chương trình cũng như việc bind dữ liệu lên các DOM. Có hai loại bind dữ liệu:

- + Event binding: các sự kiện DOM được bind ra từ view ra hàm của component.
- + Property binding: Dữ liệu bind ra từ component sang thuộc tính HTML trong view.

Directives trong Angular là một lớp js, được khai báo là @directive . Có 3 directives trong Angular:

- + Components directives: Không có nghĩa gì khi gọi component là directive cũng được, vì rõ ràng là component cho phép định nghĩa selector và gọi ra như một thẻ html tag (<component-name></component-name>)

- + Structural directives: Là directive cấu trúc, dùng để vẽ html, hiển thị data lên giao diện html, và thay đổi cấu trúc DOM bằng việc thêm bớt các phần tử trong DOM. Các structural directive thường có dấu '*' ở trước của directive. Ví dụ ***ngFor**, ***ngIf**
- + Attribute directives: Thay đổi giao diện, tương tác của các đối tượng hoặc thay đổi directive khác hoặc thêm các thuộc tính động cho element html. ví dụ ***ngStyle**
- Services và dependency injection.

Sevice là một danh mục rộng bao gồm bất kỳ value, function, hoặc tính năng mà ứng dụng cần.

DI được nối vào Angular framework và sử dụng bất cứ đâu nhằm cung cấp những component mới với service hoặc bất cứ thứ gì mà nó cần. Component sử dụng service; điều đó có nghĩa bạn có thể tiêm một service vào một component, cấp quyền truy cập thành phần cho lớp service đó. Để định nghĩa một class như một service trong Angular, sử dụng [**@Injectable\(\)**](#) decorator để cung cấp metadata điều đó cho phép Angular tiêm vào nó vào một component như một dependence Sử dụng [**@Injectable\(\)**](#) decorator để chỉ ra rằng một component hoặc một class khác (giống như service khác, một pipe hoặc một NgModule) có dependence

- + Injector là cơ chế chính. Angular tạo một injector cho toàn ứng dụng và bổ sung injector nếu cần. Vậy nên bạn sẽ không phải tạo Injector
- + Một Injector tạo nhiều dependence, và bảo trì một container của dependence instance điều đó giúp nó tái sử dụng nếu có thể.
- + Provider là một object nó sẽ nói cho injector biết làm thế nào để có hoặc tạo một dependence. Đối với bất kỳ dependence nào bạn cần cho ứng dụng, bạn cần đăng ký một provider với ứng dụng injector , vậy Injector có thể sử dụng như cung cấp và tạo new instances.

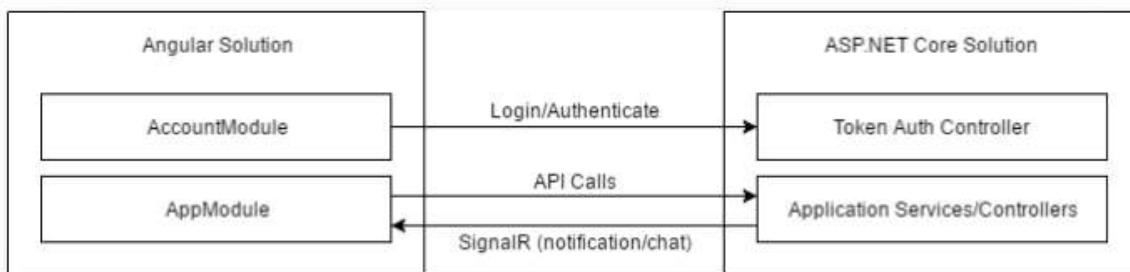
Routing.

RoutingModule là một module được đặt tại [**@angular/router**](#) cung cấp cho ứng dụng Angluar của chúng ta khả năng điều hướng và hiển thị nội dung phù hợp với địa chỉ URL. Với các ứng dụng web thông thường, việc điều hướng theo URL thường sẽ do phía server đảm nhiệm.

1.3.3. Framework ASP.Net Zero

ASP.NET Zero là một điểm bắt đầu cho ứng dụng web mới với một UI hiện đại và kiến trúc mạnh mẽ. Nó triển khai những yêu cầu cơ bản của một ứng dụng thương mại. Chúng ta nhận được mã nguồn mở để phát triển ứng dụng thương mại vì thế chúng ta tiết kiệm được thời gian để tập trung xử lý nghiệp vụ của riêng mình.

Kiến trúc của ASP.NET Zero được thể hiện ở hình dưới:



Hình 1.1. Kiến trúc ASP.NET Zero [2].

(Nguồn:<https://docs.aspnetzero.com/en/aspnet-core-angular/latest/Overview-Angular>)

Kiến trúc trên bao gồm giải pháp Angular cho người dùng và phía máy chủ có giải pháp ASP.NET Core, hai giải pháp này có thể được triển khai khác công trình trên cùng một máy chủ hoặc là trên máy chủ khác nhau. Khi được triển khai, đây thực sự là một ứng dụng HTML + JS + CSS đơn giản có thể được phục vụ trên mọi hệ điều hành và mọi máy chủ web.

ASP.NET Zero đã ra phiên bản 3.x và Angular 9.x dựa vào Single-Page Application (SPA) giải pháp cho ứng dụng trang web (với .NET Framework 4x). ASP.NET Zero cung cấp một hệ thống ủy quyền hoàn chỉnh với trang quản lý vai trò và người dùng, trang web cung cấp chức năng đăng nhập, đăng ký, đặt lại mật khẩu và xác thực email. Hệ thống hỗ trợ đa ngôn ngữ bao gồm các ngôn ngữ như Anh, Trung Quốc, Đức, Pháp, Ý, Nga, Bồ Đào Nha. Hệ thống có thể thêm, xóa và dịch ngôn ngữ trên UI.

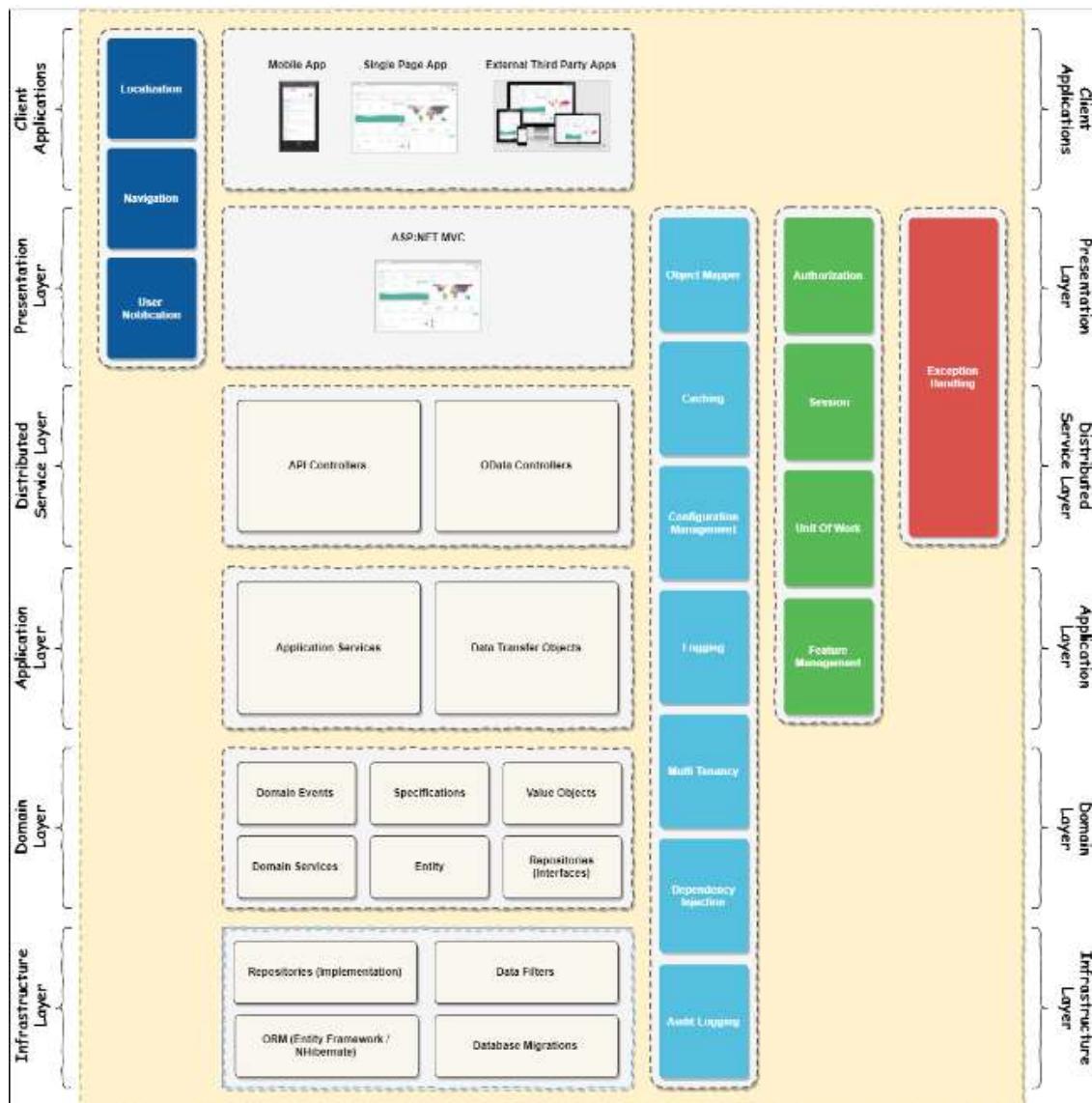
Hệ thống được xây dựng trên các công cụ như .NET Framework 4.6.x / .NET Core 2.x, Entity Framework Core-First và Migration, Angular, Bootstrap, AutoMapper, Metronic, PrimeNG.

- Framework ASP.NET Boilerplate.

ASP.NET Boilerplate (ABP) là một mã nguồn mở và là framework ứng dụng tốt. Nó không chỉ là một framework, nó còn cung cấp mô hình kiến trúc mạnh mẽ dựa trên Domain Driven Design.

Mô hình kiến trúc ứng dụng ASP.NET Boilerplate.

Mô hình bên dưới đề xuất và triển khai cho ứng dụng ASP.NET Boilerplate. ASP.NET Boilerplate không chỉ triển khai mô hình đơn giản bởi cung cấp lớp cơ bản và dịch vụ.



Hình 1.2. Mô hình kiến trúc ASP.NET Boilerplate [12].

(Nguồn: <https://github.com/aspnetboilerplate/aspnetboilerplate>)

Tầng Client Application: Đây là những client từ xa mà sử dụng ứng dụng như một dịch vụ thông qua HTTP APIs (API Controllers, OData Controllers). Một client từ xa có thể có SPA (Single Page App), một ứng dụng điện thoại, hoặc một bên thứ ba.

Tầng Presentation: ASP.NET Core (Model-View-Controller) có thể được xem xét tới tầng presentation. Nó có thể là một tầng physical (sử dụng ứng dụng thông qua HTTP APIs) hoặc một tầng logical (trực tiếp thêm vào và sử dụng dịch vụ). Trong mỗi trường hợp nó có thể bao gồm Localization, Navigation, Object Mapping, Caching, Configuration Management, Audit Logging. Nó cũng giao tiếp với Authorization, Session, Features (cho ứng dụng đa người dùng) và Exception Handling.

Tầng Distributed Service: tầng này sử dụng chức năng application/domain thông qua APIs từ xa như REST, OData, GraphQL,... Chúng không chứa logic nghiệp vụ nhưng chỉ chuyển HTTP requests tới tương tác domain. Tầng này thường bao gồm Authorization, Caching, Audit Logging, Object Mapping, Exception Handling, Session.

Tầng Application Layer: tầng này bao gồm Application Services mà nó sử dụng tầng domain và domain object (Domain Services, Entities,...) tới hiệu suất truy cập chức năng ứng dụng. Nó sử dụng Data Transfer Object tới lấy dữ liệu và trả dữ liệu cho tầng presentation hoặc distributed service. Nó có thể thỏa thuận với Authorization, Caching, Audit Logging, Object Mapping và Session.

Tầng Domain: Nó là tầng chính và nó triển khai domain logic của chúng ta. Nó bao gồm Entities, Value Object và Domain Services tới trình bày nghiệp vụ. Nó có thể bao gồm Specification và trigger Domain Events. Nó định nghĩa Interfaces Repository tới đọc entities từ nguồn dữ liệu.

Tầng Infrastructure: tầng này cho các tầng khác làm việc. Nó triển khai Interface Repository tới làm việc với một cơ sở dữ liệu thực.

- Những lợi ích của ASP.NET Boilerplate.

Dependency Injection: ASP.NET Boilerplate sử dụng và cung cấp một cơ sở hạ tầng DI thông thường. Vì lớp này nó là một dịch vụ ứng dụng, nó thông thường được đăng ký vào container DI là tạm thời.

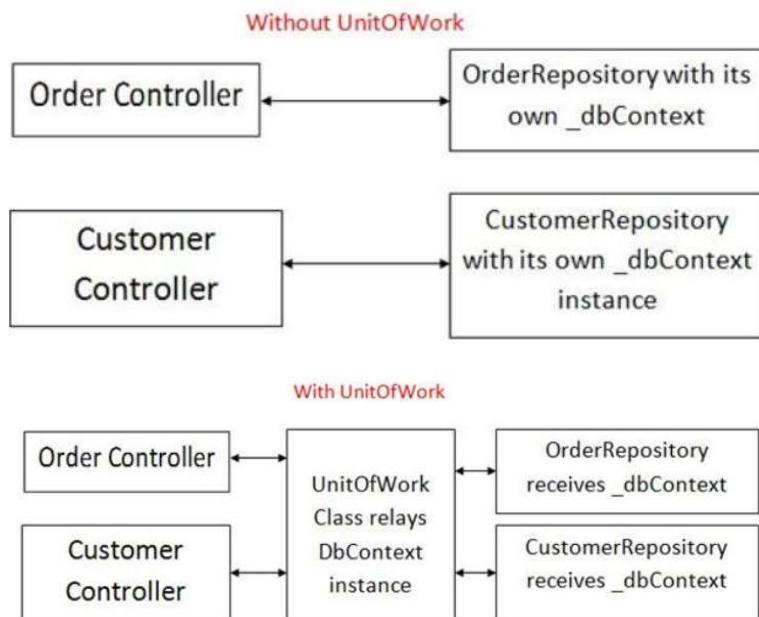
Repository: ABP có thể tạo một repository mặc định cho mỗi entity. Repository mặc định có hỗ trợ nhiều hàm. Chúng ta có thể mở rộng repository mặc định để phù hợp với những thứ chúng ta cần.

Authorization: ABP có thể kiểm tra khai báo quyền. Nó ngăn chặn truy cập tới chức năng cập nhật nếu người dùng hiện tại không có quyền cập nhật hoặc không đăng nhập vào hệ thống. ABP không chỉ sử dụng thuộc tính khai báo nhưng nó cũng có thêm vào nếu bạn có quyền.

Validation: ABP tự động kiểm tra nếu đầu vào là null. Nó cũng xác nhận toàn bộ thuộc tính của đầu vào dựa trên khai báo dữ liệu chuẩn và có thể thay đổi luật xác nhận. Nếu một yêu cầu là không hợp lệ nó cho ra ngoài thông báo lỗi và xử lý nó tới người dùng.

Audit Logging: User, browser, IP address, gọi dịch vụ, hàm, đối vào, thời gian gọi, thời gian xử lý và một số thông tin khác là tự động lưu cho mỗi yêu cầu dựa trên quy ước và thiết lập.

Unit Of Work: ABP cung cấp quản lý kết nối và giao dịch là một trong những khái niệm quan trọng trong ứng dụng có sử dụng kết nối cơ sở dữ liệu. Chúng ta cần biết tới khi nào mở một kết nối, khi nào bắt đầu một giao dịch, và kết một kết nối như thế nào. Việc quản lý như vậy đều sử dụng unit of work.



Hình 1.3 Mô hình hoạt động của Unit of work

Exception Handling: Chúng ta thường không phải xử lý thủ công ngoại lệ trong ABP trên ứng dụng web. Toàn bộ ngoại lệ được tự động xử lý bởi mặc định. Nếu một ngoại lệ xảy ra ABP tự động ghi lại vào nhật ký và trả về một kết quả thích hợp tới người dùng.

Logging: Chúng ta có thể viết nhật ký khi sử dụng đối tượng Logger được định nghĩa bên trong lớp cơ sở. Log4Net được sử dụng mặc định nhưng nó có thể thay đổi và có thể thiết lập.

Auto Mapping: Trong ABP có hàm MapTo của lớp IObjectMapper cho phép ánh xạ đầu vào dựa theo thuộc tính cho các thuộc tính thực thể. Chúng ta có thể dễ dàng ánh xạ các thuộc tính từ đối tượng này sang đối tượng khác dựa theo các quy ước đặt tên.

1.4. Tổng quát về ngôn ngữ mô hình hóa UML

1.4.1. Các thành phần ngôn ngữ UML

Ngôn ngữ UML bao gồm một loạt các phần tử đồ họa (graphic element) có thể được kết hợp với nhau để tạo ra các biểu đồ. Bởi đây là một ngôn ngữ, nên UML cũng có các nguyên tắc để kết hợp các phần tử đó. Một số những thành phần chủ yếu của ngôn ngữ UML:

Hướng nhìn (view), biểu đồ (diagram), phần tử mô hình hóa (model element): Các khái niệm được sử dụng trong các biểu đồ được gọi là các phần tử mô hình, thể hiện các khái niệm hướng đối tượng quen thuộc. Ví dụ: Lớp, đối tượng, thông điệp cũng như các quan hệ giữa các khái niệm này, bao gồm cả liên kết, phụ thuộc, khái quát hóa. Một phần tử mô hình thường được sử dụng trong nhiều biểu đồ khác nhau, nhưng nó luôn luôn có chỉ một ý nghĩa và một kí hiệu.

Cơ chế chung: Cơ chế chung cung cấp thêm những lời nhận xét bổ sung, các thông tin cũng như các quy tắc ngữ pháp chung về một phần tử mô hình; Chúng còn cung cấp thêm các cơ chế để có thể mở rộng ngôn ngữ UML cho phù hợp với một phương pháp xác định (một quy trình, một tổ chức hoặc một người dùng).

1.4.2. Biểu đồ

Biểu đồ là các hình vẽ miêu tả nội dung trong một hướng nhìn. UML có tất cả 9 loại biểu đồ khác nhau được sử dụng trong những sự kết hợp khác nhau để cung cấp tất cả các hướng nhìn của một hệ thống.

1.4.3. Biểu đồ Use case (Use Case Diagram)

Một biểu đồ Use case chỉ ra một số lượng các tác nhân ngoại cảnh và mối liên kết của chúng đối với Use case mà hệ thống cung cấp. Một Use case là một lời miêu tả của một chức năng mà hệ thống cung cấp.

1.4.4. Biểu đồ lớp (Class Diagram)

Một biểu đồ lớp chỉ ra cấu trúc tĩnh của các lớp trong hệ thống. Các lớp là đại diện cho các đối tượng được xử lý trong hệ thống. Các lớp có thể quan hệ với nhau trong nhiều dạng thức: liên kết (associated - được nối kết với nhau), phụ thuộc (dependent - một lớp này phụ thuộc vào lớp khác), chuyên biệt hóa (specialized - một lớp này là một kết quả chuyên biệt hóa của lớp khác), hay đóng gói (packaged - hợp với nhau thành một đơn vị). Tất cả các mối quan hệ đó đều được thể hiện trong biểu đồ lớp, đi kèm với cấu trúc bên trong của các lớp theo khái niệm thuộc tính (attribute) và thủ tục (operation). Biểu đồ được coi là biểu đồ tĩnh theo phương diện cấu trúc được miêu tả ở đây có hiệu lực tại bất kỳ thời điểm nào trong toàn bộ vòng đời hệ thống.

1.4.5. Biểu đồ trình tự (Sequence Diagram)

Một biểu đồ trình tự chỉ ra một cộng tác động giữa một loạt các đối tượng. Khía cạnh quan trọng của biểu đồ này là chỉ ra trình tự các thông điệp (message) được gửi giữa các đối tượng. Nó cũng chỉ ra trình tự tương tác giữa các đối tượng, điều sẽ xảy ra tại một thời điểm cụ thể nào đó trong trình tự thực thi của hệ thống.

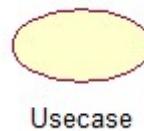
1.4.6. Tác nhân (Actor)

Một người/nhóm người hoặc một thiết bị hoặc hệ thống tác động hoặc thao tác đến hệ thống.



1.4.7. Ca sử dụng (Use case)

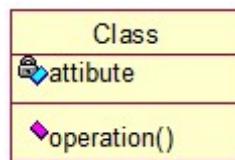
Ca sử dụng mô tả một tập hợp các hành động mà hệ thống thực hiện để cho ra một kết quả có thể quan sát được đối với một tác nhân.



1.4.8. Lớp (Class)

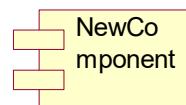
Một lớp mô tả một tập hợp các đối tượng có chung các thuộc tính, các phương thức, các mối quan hệ và ngữ nghĩa. Lớp được biểu diễn bằng một hình chữ nhật, bên trong có

tên, các thuộc tính và các phương thức. Ngoài ra khi muốn vẽ đơn giản thì có thể bỏ trống hoặc bỏ qua các thuộc tính hoặc các phương thức.



1.4.9. Thành phần

Thành phần là một bộ phận vật lý có thể thay thế được của một hệ thống, phù hợp với những điều kiện cụ thể và cung cấp các phương thức thực hiện một tập các giao diện.



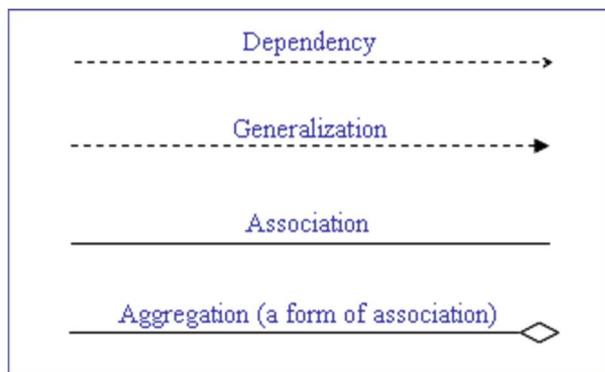
1.4.10. Các liên kết

Nối kết (Association): Nối các phần tử và các thực thể nối (link).

Khái quát hóa (Generalization): Còn được gọi là tính thừa kế, có ý nghĩa rằng một phần tử này có thể là một sự chuyên biệt hóa của một phần tử khác.

Sự phụ thuộc (Dependency): Chỉ ra rằng một phần tử này phụ thuộc trong một phương thức nào đó vào một phần tử khác.

Kết tập (Aggregation): Một dạng của nối kết, trong đó một phần tử này chứa các phần tử khác.



Hình 2.1. Các phần tử liên kết.

1.4.11.Các công nghệ khác

- Metronic : là một mẫu vô cùng tiện ích, ngay cả mua một giấy phép mở rộng thì nó cũng tiết kiệm chi phí vô cùng lớn. Code và CSS được tổ chức tốt và có tính năng vô cùng phong phú mà không bị cồng kềnh. Nó nhanh chóng được tích hợp vào sản phẩm phức tạp và đã có sẵn UX cho nhiều trường hợp sử dụng. Khi muốn mở rộng chức năng hệ thống này làm cho cực kỳ dễ dàng.
- PrimeNG: là một tập hợp các thành phần UI phong phú dành cho Angular. Tất cả các thành phần đều là mở nguồn mở và miễn phí sử dụng theo giấy phép MIT. PrimeNG được phát triển bởi PrimeTek Informatics, một nhà cung cấp có nhiều năm kinh nghiệm trong việc phát triển các giải pháp UI nguồn mở. Bộ tiện ích đã hoàn chỉnh hơn 80 thành phần được đóng gói thành các component để sử dụng dễ dàng cho tất cả yêu cầu UI.
- Bootstrap: là một framework mã nguồn mở miễn phí, dựa trên HTML, CSS và Javascript, nó được tạo ra để xây dựng các giao diện Website tương thích với tất cả các thiết bị có kích thước màn hình khác nhau. Bootstrap có ưu điểm dễ sử dụng, responsive và tương thích với các trình duyệt. Tuy nhiên với trình duyệt IE, Bootstrap 4 chỉ hỗ trợ từ IE10 trở lên.
- Mapbox: dễ dàng thêm một bản đồ số vào bất kì một trang web. Nó có thể hiển thị map tiles, dữ liệu dạng vector và có thể tải được từ bất kì nguồn dữ liệu nào. Mapbox đã được phát triển để tiếp tục sử dụng thông tin địa lý của tất cả các loại. Nó hoàn toàn miễn phí, mã nguồn mở javascript được phát hành theo giấy phép hai điều khoản.
- Turf.js: là một thư viện JavaScript mã nguồn mở cho phép bạn thực hiện các thao tác không gian trong trình duyệt. Turf giúp bạn phân tích, tổng hợp và chuyển đổi dữ liệu để trực quan hóa dữ liệu theo những cách mới và trả lời các câu hỏi nâng cao..

CHƯƠNG 2. KHẢO SÁT VÀ THIẾT KẾ HỆ THỐNG

2.1. Khảo sát

Với nhịp sống công nghiệp hóa - hiện đại hóa, con người ngày một bận rộn, vấn đề tiết kiệm thời gian được quan tâm hàng đầu do đó đồ ăn nhanh ngày càng được ưa chuộng.

Lăm bắt được thực trạng đó, Five Star, 1 thương hiệu con đến từ đất nước Thái Lan. Thuộc tập đoàn Charoen Porkphand Foods (C.P.F). Hàng này luôn luôn phát triển các loại sản phẩm tốt chất lượng cũng như khẩu vị hợp với người châu Á đặc biệt là gà rán.

- Hiện đang có nhà máy tại Lô CN - B3, KCN Phú Nghĩa, xã Phú Nghĩa, huyện Chương Mỹ, Hà Nội. Lô 13, Đường 19A-N19 KCN Biên Hòa II, TP.Biên Hòa
- Đơn vị hoạt động trong lĩnh vực chăn nuôi, sản xuất thực phẩm theo chuỗi “Thức ăn chăn nuôi – Trang trại chăn nuôi – Sản xuất và chế biến thực phẩm”
- Hiện tại có khoảng 750 điểm bán gà rán FiveStar
- Dự kiến mở vài nghìn điểm bán
- Phân khúc thị trường định vị trung bình 30K/1 suất ăn
- Mô hình hợp tác



Mô hình mở điểm bán

- FiveStar cung cấp 1 đơn vị cung cấp giải pháp từ khảo sát, xây dựng cơ sở dữ liệu, thành lập bản đồ các tỉnh thành trên cả nước.
- Có thể tạo bản đồ đánh giá điểm bán theo nhiều tiêu chí
 - + Bản đồ doanh thu theo thời gian (ngày, tuần, tháng, quý, năm, theo khoảng bất kỳ,...)
 - + Bản đồ doanh thu theo cơ cấu sản phẩm
 - + Bản đồ theo số lượt mua hàng
 - + Bản đồ danh sách cửa hàng đang hoạt động
- Từng tiêu chí có thể chấm điểm để đánh giá, trên cơ sở tổng điểm để so sánh các vị trí tìm ra vị trí tối ưu.
- Hỗ trợ tìm kiếm lựa chọn điểm bán phù hợp theo yêu cầu.
- Các tiêu chí có thể mở rộng tùy theo nhu cầu thông tin FiveStar quan tâm

2.2. Kết luận

- Xây dựng giải pháp hỗ trợ điểm bán (Site Selection)
- Cung cấp thông tin hỗ trợ ra quyết định mở điểm bán
- Thống kê doanh thu và số lượng khách mua hàng ...

2.3. Bài toán đặt ra

2.3.1. Bài toán đặt ra

- Bài toán mở điểm bán (Site):
- + Tiêu chí đang áp dụng để mở điểm bán mới:
 - Nơi dân cư đông, tập trung nhiều khu ăn uống
 - Có trường học cấp 1,2,3. Trường Cao Đẳng, Đại Học, văn phòng, chung cư.
 - Gần bệnh viện, bến xe, chợ, siêu thị
 - Ngã 3, ngã 4, cây xăng lớn.
 - Những nơi đã có thương hiệu KFC, Lotteria, Jolibee
 - Bài toán: Dựa vào những tiêu chí mở điểm bán trên, và thông tin những điểm bán đã và đang hoạt động... để phân tích thị trường, khu vực và đưa ra gợi ý quyết định mở hay không.
- Giải pháp

CSDL:

- CSDL Giao thông, hành chính
- CSDL POI

- CSDL vị trí Site dự kiến
- CSDL vị trí Site hiện có

Ứng dụng:

- Quản lý bản đồ Site (bản đồ điểm bán)
 - Tìm kiếm Site (lựa chọn điểm bán)
 - Khảo sát Site
 - Đánh giá Site (đánh giá điểm bán hiện thời)
- Phương án xây dựng

FiveStar: Điều tra bổ sung thông tin Site

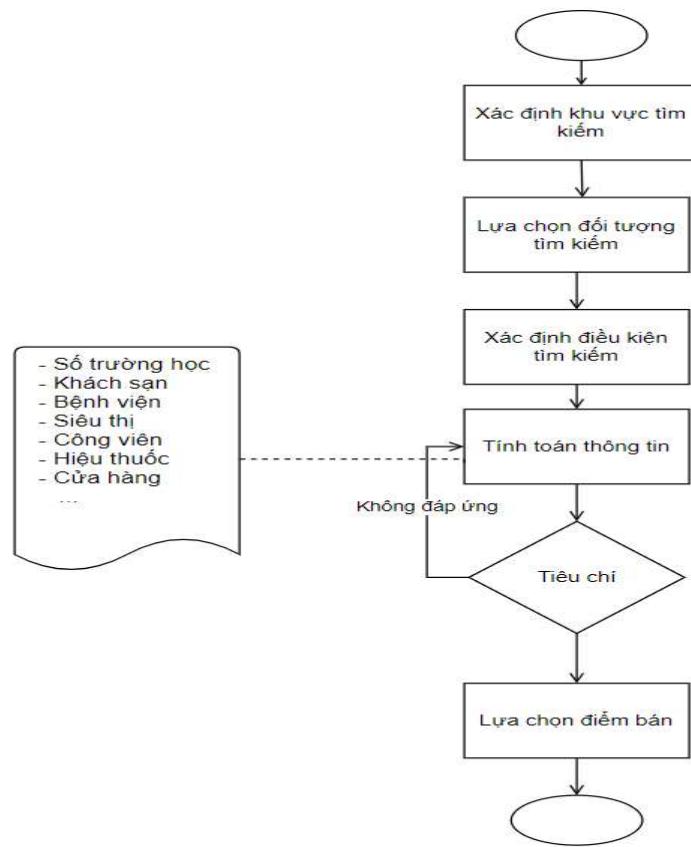
eKGIS cung cấp

- Thiết kế khung mô hình CSDL hệ thống và nghiệp vụ
- Khởi tạo dữ liệu ban đầu từ nguồn sẵn có của eKGIS và
- Thiết kế khung mô hình CSDL hệ thống và nghiệp vụ
- Khởi tạo dữ liệu ban đầu từ nguồn sẵn có của eKGIS và OpenStreetmap
- CSDL địa phận hành chính Việt Nam (đô thị, khu công nghiệp)
- CSDL Mạng lưới giao thông
- CSDL POI
- CSDL vị trí Site dự kiến theo các tiêu chí sơ bộ
- Nơi dân cư đông, tập trung nhiều khu ăn uống
- Có trường học cấp 1,2,3. Trường Cao Đẳng, Đại Học, văn phòng, chung cư.
- Gần bệnh viện, bến xe, chợ, siêu thị
- Ngã 3, ngã 4, cây xăng lớn.
- Những nơi đã có thương hiệu KFC, Lotteria, Jolibee

- **Quy trình :**

Quy trình lựa chọn điểm bán:

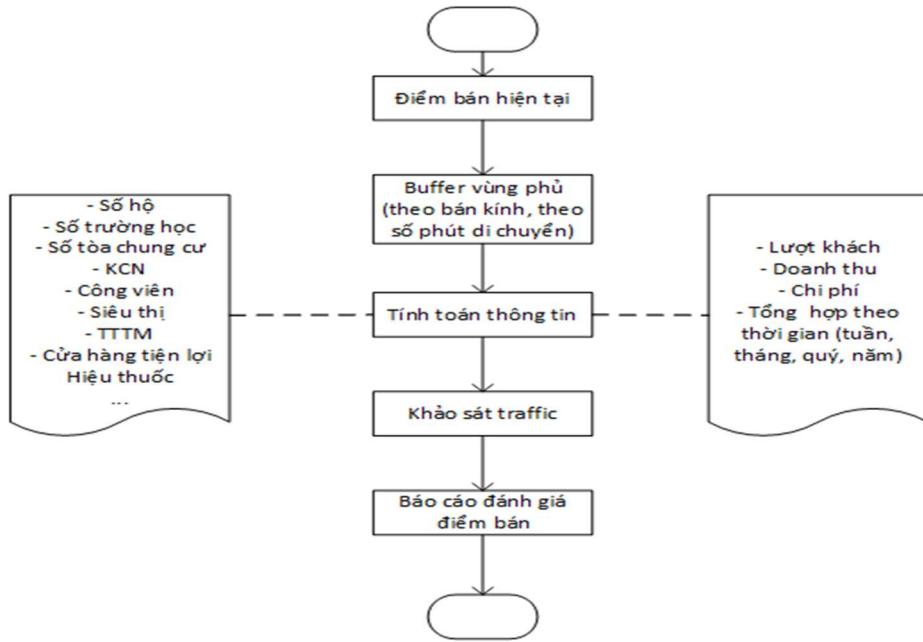
- Phần mềm sẽ hỗ trợ lựa chọn và tính toán các thông tin ban đầu để gợi ý khu vực (site) mở điểm bán
- Cán bộ phụ trách thị trường thực hiện điều tra khảo sát thực địa bằng ứng dụng phần mềm thu thập
- Phân tích các thông tin hiện có và thông tin bổ sung so sánh với tiêu chí để quyết định lựa chọn mở điểm bán



Hình 2.2 Mô hình quy trình lựa chọn điểm bán

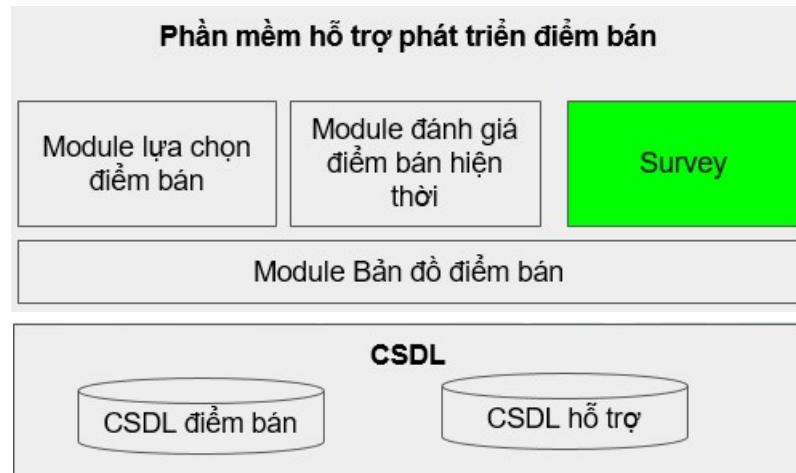
Đánh giá điểm bán hiện có

- Phần mềm dựa trên bản đồ điểm bán hiện có tự động tính toán các thông tin về tiêu chí mở điểm bán, kết quả bán hàng kết hợp thông tin khảo sát thực tế tại cửa hàng để tự động tạo báo cáo đánh giá điểm bán hiện trạng.



Hình 2.3 Mô hình quy trình đánh giá điểm bán

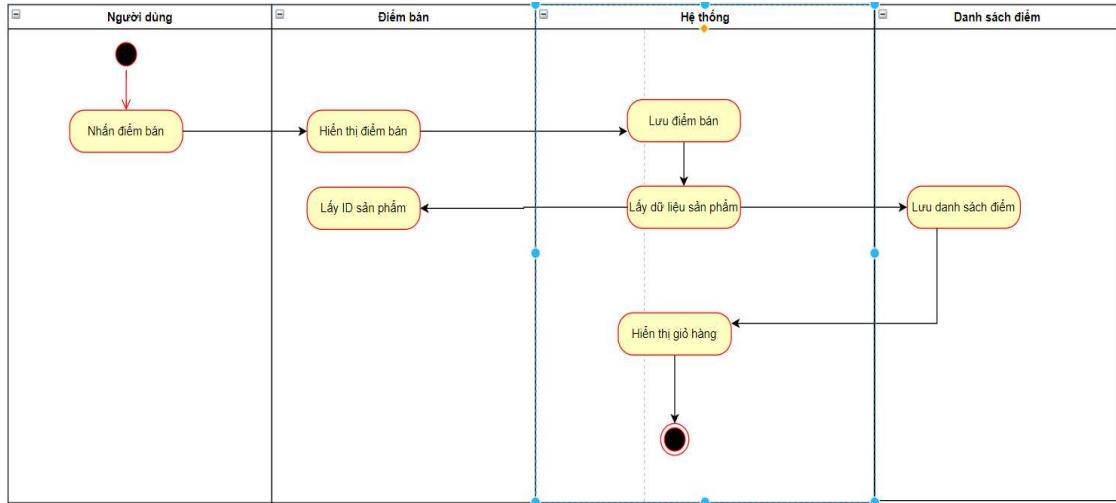
2.3.2. Mô hình



Hình 2.4 Mô hình phát triển ứng dụng điểm bán

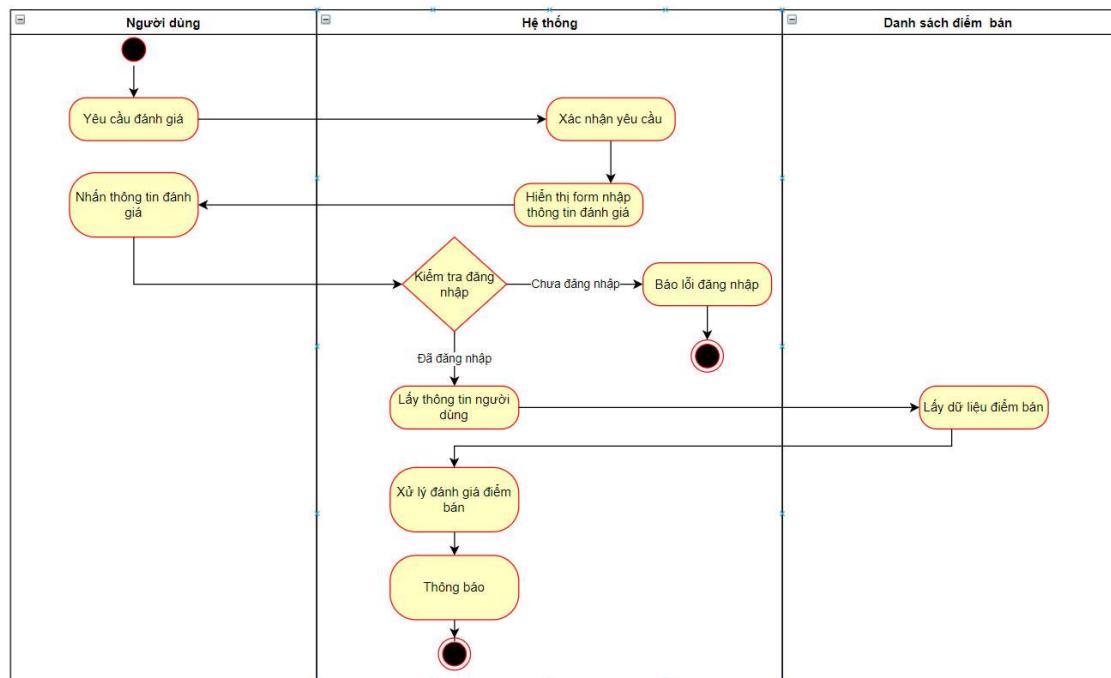
2.3.3. Quy trình nghiệp vụ

a, Quy trình trọn điểm bán



Hình 2.5 Biểu đồ phân lèn quy trình trọn điểm bán

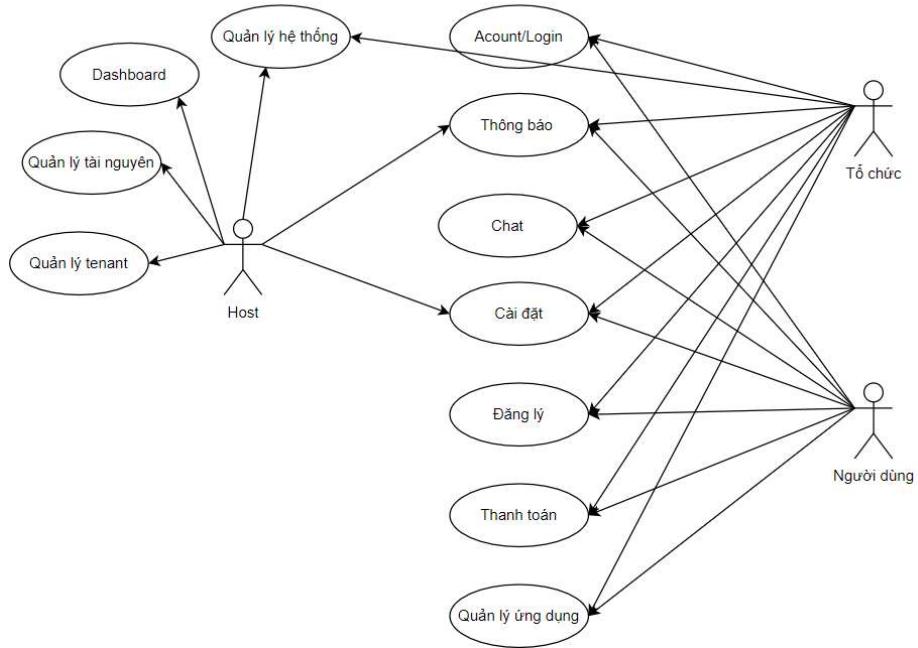
b, Quy trình đánh giá điểm bán



Hình 2.6 Biểu đồ phân lèn quy trình đánh giá điểm bán

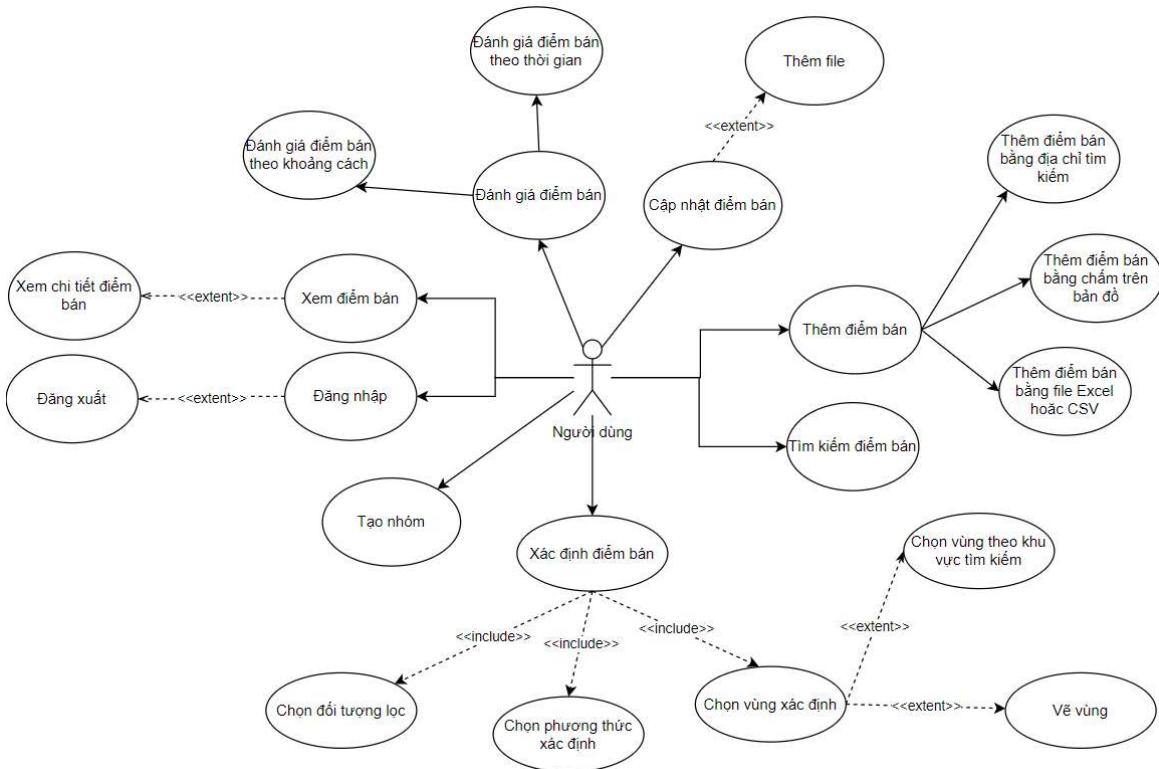
2.3.4. Mô hình use case

a, Sơ đồ use case tổng quát



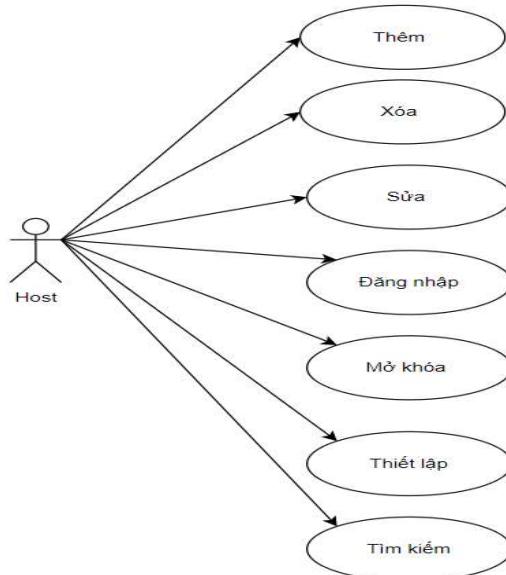
Hình 2.7 Sơ đồ use case tổng quát

b, Sơ đồ use case người dùng



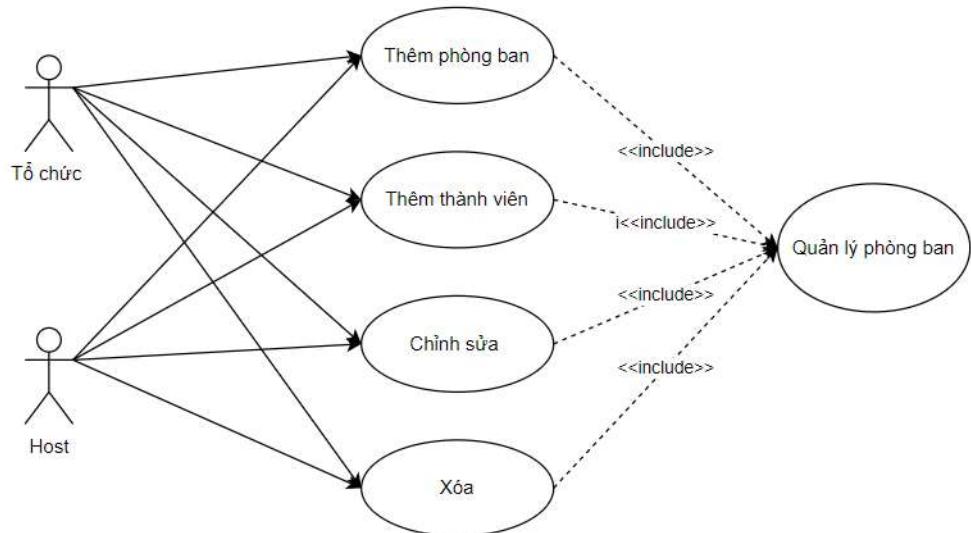
Hình 2.8 Sơ đồ use case người dùng

c, Sơ đồ use case quản lý tenant (Thuê bao)



Hình 2.9 Sơ đồ use case quản lý tenant

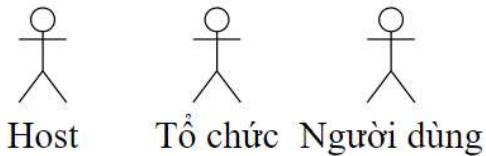
d, Sơ đồ use case quản lý phòng ban (Org Unit)



Hình 2.10 Sơ đồ quản lý phòng ban

2.3.5. Danh sách các tác nhân

Các tác nhân gồm có : host, tổ chức, người dùng



Hình 2.11 Tác nhân hệ thống

2.3.6. Danh sách use case và mô tả

Bảng 2.1 Danh sách use case và mô tả

ID	Tên Use case	Mô tả ngắn gọn Use case	Chức năng	Ghi chú
<i>UC_001</i>	<i>Quản lý người dùng</i>		<ul style="list-style-type: none"> -Đăng ký -Đăng nhập -Quên mật khẩu -Đăng xuất 	
<i>UC_002</i>	<i>Quản lý điểm bán</i>	<i>Cho cập nhật thông tin điểm bán</i>	<ul style="list-style-type: none"> -Thêm điểm bán -Cập nhật danh sách điểm bán -Cập nhật chi tiết thông tin điểm bán - Cập nhập tệp tin -Xem tệp tin -Tải tệp tin 	
<i>UC_003</i>	<i>Hỗ trợ người dùng</i>		<ul style="list-style-type: none"> -Tìm kiếm điểm bán 	
<i>UC_004</i>	<i>Quản lý đánh giá điểm bán</i>		<ul style="list-style-type: none"> -Phân vùng và thống kê đánh giá 	
<i>UC_005</i>	<i>Báo cáo</i>		<ul style="list-style-type: none"> - In thông tin đánh giá điểm bán 	

			<i>- Xuất điểm bán sang file Excel.</i>	
<i>UC_006</i>	<i>Quản lý kiểu mẫu</i>	<i>Thiết lập style cho điểm bán</i>	<i>- Thiết lập kiểu mẫu mặc định - Thiết lập kiểu mẫu từng điểm bán</i>	
<i>UC_007</i>	<i>Quản lý địa điểm</i>	<i>Các điểm được thu thập dữ liệu từ bên ngoài qua khảo sát hoặc nguồn dữ liệu nào đó ...</i>	<i>- Cập nhật địa điểm</i>	

Trong đó:

UC: Quy cách đánh số Use case

001, 002...: là số thứ tự của use case

2.3.7. Đặc tả các yêu cầu chức năng

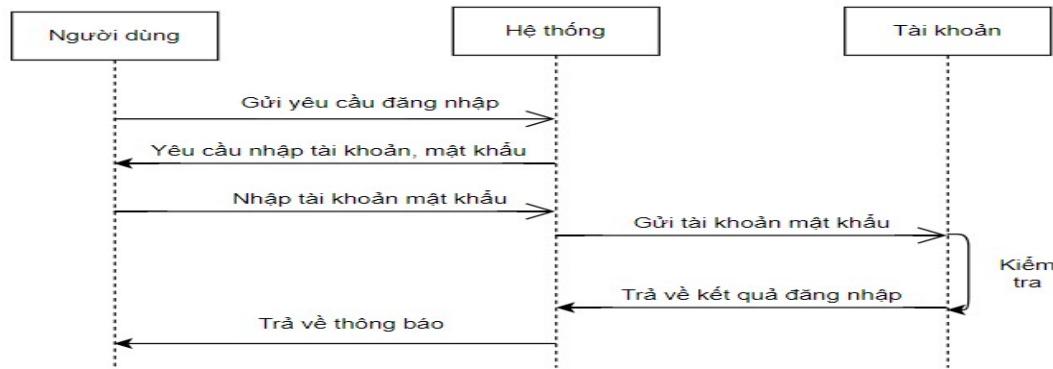
- Đặc tả use đăng nhập

Bảng 2.2 Đặc tả use case đăng nhập

Use case: UC_001_Đăng nhập	
Mục đích:	Đăng nhập vào hệ thống để thao tác với web
Mô tả:	Người dùng nhập tài khoản mật khẩu vào hệ thống để hệ thống kiểm tra
Tác nhân:	Host , Quản trị tổ chức, người sử dụng
Điều kiện trước:	Người dùng nhập tài khoản mật khẩu
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị trang “đăng nhập” bao gồm các text tên đăng nhập, mật khẩu và lựa chọn thuê bao. 2. Tác nhân chọn thuê bao và nhập thông tin theo yêu cầu. 3. Tác nhân nhấn nút “Đăng nhập”. 4. Hệ thống kiểm tra thông tin đăng nhập.

	5. Use case kết thúc.
Luồng sự kiện phụ	<ol style="list-style-type: none"> Nếu tại dòng sự kiện chính số 3 tác nhân “Hủy bỏ”. Nếu tạo dòng sự kiện chính số 4 hệ thống kiểm tra thông tin đăng nhập không hợp lệ, hệ thống thông báo lỗi. Usecase kết thúc.
Điều kiện sau:	Người đăng nhập đã có tài khoản trong hệ thống

+ Biểu đồ trình tự



Hình 2.12 Biểu đồ trình tự use case đăng nhập

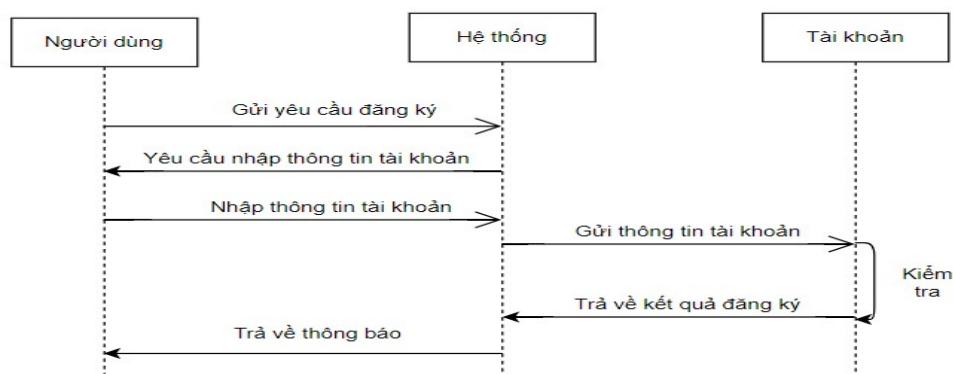
- ĐẶC TẢ USE CASE ĐĂNG KÝ

Bảng 2.3 Đặc tả use case đăng ký

Use case: UC_001_Đăng ký	
Mục đích:	Tạo tài khoản mới để đăng nhập vào web
Mô tả:	Người dùng nhấn đăng ký, nhập thông tin của người dùng theo yêu cầu của hệ thống
Tác nhân:	Người dùng
Điều kiện trước:	Nhập đầy đủ thông tin
Luồng sự kiện chính	<ol style="list-style-type: none"> Hệ thống hiển thị trang “đăng ký” bao gồm các text tên nhập họ tên, địa chỉ email, tên truy cập, mật khẩu, nhập lại mật khẩu, chứng minh không phải robot

	<ol style="list-style-type: none"> 2. Tác nhân nhấn nút “Đăng ký”. 3. Hệ thống kiểm tra thông tin đăng ký. 4. Use case kết thúc.
Luồng sự kiện phụ	Nếu tài khoản đã tồn tại hoặc thông tin chưa đúng yêu cầu , thông báo tới người dùng để họ chỉnh sửa lại
Điều kiện sau:	Tạo được tài khoản đăng nhập để vào trang

+ Biểu đồ trình tự



Hình 2.13 Biểu đồ trình tự use case đăng ký

- Đặc tả use case đăng xuất

Bảng 2.4 Đặc tả use case đăng xuất

Use case: UC_001_Đăng xuất	
Mục đích:	Thoát tài khoản ra khỏi trang web hoặc đổi tài khoản khác
Mô tả:	Khi người sử dụng muốn thoát thông tin trong hệ thống
Tác nhân:	Người dùng, Host, quản trị
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Tác nhân chọn “Đăng xuất”. 2. Hệ thống đăng xuất khỏi hệ thống và chuyển sang trang đăng nhập

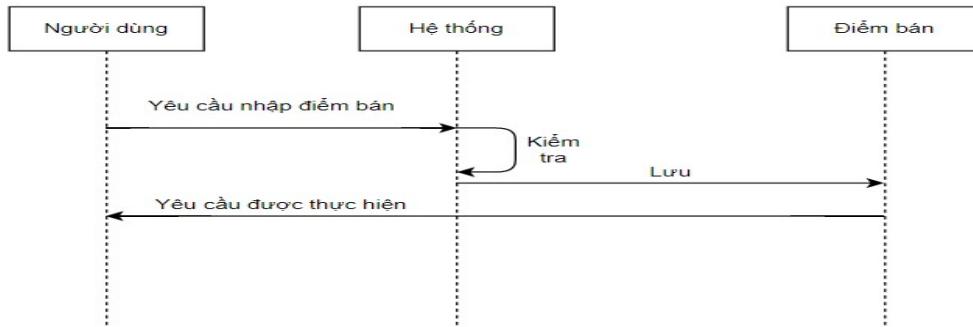
	3. Use case kết thúc.
Luồng sự kiện phụ	Nếu tại dòng sự kiện chính số 1, tác nhân “Hủy bỏ”. Use case kết thúc.
Điều kiện sau:	. Không có

- Đặc tả use case thêm điểm bán

Bảng 2.5 Đặc tả use case thêm điểm bán

Use case: UC_002 _Thêm điểm bán	
Mục đích:	Tạo điểm bán hiển thị trên bản đồ để sau này sử dụng đánh giá
Mô tả:	Khi người dùng muốn thêm điểm bán vào hệ thống
Tác nhân:	Người dùng , quản trị
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Tác nhân chọn nhập điểm bán theo 3 phương thức : <ul style="list-style-type: none"> - Tìm kiếm vị trí - Chấm trên bản đồ - Nhập file 2. Hệ thống cập nhật các điểm bán 3. Người dùng kết thúc sự kiện nhập 4. Use case kết thúc
Luồng sự kiện phụ	
Điều kiện sau:	. Không có

- + Biểu đồ đồ trình tự



Hình 2.14 Biểu đồ trình tự use case thêm điểm bán

- Đặc tả use case cập nhật danh sách điểm bán

Bảng 2.6 Đặc tả use case cập nhật danh sách điểm bán

Use case:UC_002_Cập nhật danh sách điểm bán	
Mục đích:	Sắp xếp lại thứ tự hiển thị của điểm bán
Mô tả:	Khi người dùng muốn thay đổi vị trí cửa các điểm bán với nhau giúp người dùng dễ dàng sử dụng
Tác nhân:	Người dùng , quản trị
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Tác nhân chọn vào điểm bán trên danh sách điểm bán, 2. Kéo điểm bán vào vị trí cần thay đổi 3. Vị trí kéo hợp lệ use case kết thúc
Luồng sự kiện phụ	Vị trí kéo không hợp lệ, use case kết thúc
Điều kiện sau:	. Không có

- Đặc tả use case cập nhật chi tiết điểm bán

Bảng 2.7 Đặc tả use case cập nhật chi tiết điểm bán

Use case:UC_002_Cập nhật chi tiết điểm bán
--

Mục đích:	Thêm, sửa, xóa thông tin điểm bán
Mô tả:	Khi có bất kì thông tin điểm bán nào thay đổi, tác nhân có thể vào chỉnh sửa lại thông tin điểm bán
Tác nhân:	Người dùng , quản trị
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hiển thị chi tiết điểm bán 2. Thêm mới điểm bán 3. Chỉnh sửa điểm bán 4. Xóa điểm bán 5. Di chuyển điểm bán 6. Hệ thống lưu thông tin vào CSDL. 7. Use case kết thúc.
Luồng sự kiện phụ	Nếu tại dòng sự kiện chính số 2-4, tác nhân “Hủy bỏ”. Use case kết thúc.
Điều kiện sau:	. Không có

- Đặc tả use case cập nhật file

Bảng 2.8 Đặc tả use case cập nhật cập nhật file

Use case:UC_002_Cập nhật file	
Mục đích:	Thêm các tệp tin liên quan tới điểm bán
Mô tả:	Các tệp tin liên quan sẽ được đính kèm với điểm bán nếu có
Tác nhân:	Người dùng , quản trị
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống, các định dạng file thêm phải được cho phép bởi hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hiển thị chi tiết điểm bán

	2. Thêm file 3. Xóa file 4. Hệ thống lưu thông tin vào CSDL. 5. Use case kết thúc.
Luồng sự kiện phụ	Nếu tại dòng sự kiện chính số 2-3, tác nhân “Hủy bỏ”. Use case kết thúc.
Điều kiện sau:	. File có kích thước không quá 200mb

- Đặc tả use case xem file

Bảng 2.9 Đặc tả use case xem file

Use case:UC_002_Xem file	
Mục đích:	Một vài tệp được hỗ trợ có thể xem trực tiếp trên web khi thêm
Mô tả:	
Tác nhân:	Người dùng , quản trị
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	1. Hiển thị chi tiết điểm bán 2. Xem file 3. Kiểm tra file đã load xong chưa 4. Mở file trên tab mới 5. Use case kết thúc.
Luồng sự kiện phụ	Nếu file chưa load xong, thông báo tới người dùng
Điều kiện sau:	Không có

- Đặc tả use case tải file

Bảng 2.10 Đặc tả use case tải file

Use case:UC_002_ Tải file

Mục đích:	Người dùng có thể tải bất kỳ file nào được đính kèm về máy
Mô tả:	Mọi file đính kèm đều được public cho phép người dùng tải công khai về máy
Tác nhân:	Người dùng , quản trị
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hiển thị chi tiết điểm bán 2. Tải file 3. Use case kết thúc.
Luồng sự kiện phụ	
Điều kiện sau:	Không có

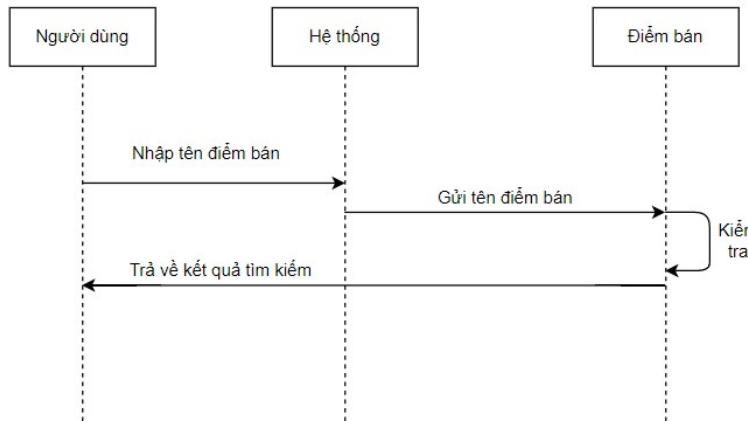
- Đặc tả use case tìm kiếm điểm bán

Bảng 2.11 Đặc tả use case tải file

Use case:UC_003_ Tìm kiếm điểm bán	
Mục đích:	Giúp người dùng thuận tiện trong việc thao tác với các điểm bán một cách dễ dàng nhất
Mô tả:	Tìm các điểm bán cần sử dụng
Tác nhân:	Người dùng , quản trị
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hiển thị cây danh sách điểm bán 2. Nhập ký tự vào ô tìm kiếm 3. Hiển thị điểm bán theo ký tự tác nhân nhập 4. Chọn điểm bán 5. Use case kết thúc.

Luồng sự kiện phụ	
Điều kiện sau:	Không có

+ Biểu đồ trình tự



Hình 2.15 Biểu đồ trình tự use case tìm kiếm điểm bán

- Đặc tả use case phân vùng và thống kê đánh giá

Bảng 2.12 Đặc tả use case tải file

Use case: UC_004_Phân vùng đánh giá	
Mục đích:	Điểm đánh giá được phân theo các vùng theo tùy chọn đánh giá của tác nhân đánh giá, đánh giá sẽ thống kê các loại đối tượng theo các vùng
Mô tả:	Vùng đánh giá được chia thành các khung và hiển thị theo các màu sắc khác nhau. Được hiển thị theo phạm vi đánh giá, hệ thống sẽ thống kê số liệu theo các vùng đánh giá.
Tác nhân:	Người dùng
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> Hiển thị danh sách điểm bán Đánh giá Thiết lập tùy chọn đánh giá

	4. Nhấn thực hiện 5. Hệ thống thực hiện đánh giá 6. Hiển thị đánh giá 7. Use case kết thúc.
Luồng sự kiện phụ :	Nếu tại dòng sự kiện chính số 2-4, tác nhân “Hủy bỏ”. Use case kết thúc.
Điều kiện sau:	Không có

- ĐẶC TẢ USE CASE THIẾT LẬP KIỂU MẪU MẶC ĐỊNH

Bảng 2.13 Đặc tả use case tải file

Use case:UC_006_ Thiết lập kiểu mẫu mặc định	
Mục đích:	Thay đổi kiểu mẫu điểm bán khi thêm mới các điểm bán, thay đổi khung bản đồ và vị trí hiển thị của nội dung hiển thị
Mô tả:	Khi người dùng thêm điểm bán, các điểm bán được hiển thị theo các kiểu mẫu đã thiết lập
Tác nhân:	Người dùng
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	1. Chọn “Tạo kiểu mẫu mặc định” 2. Thay đổi tùy chọn 3. Use case kết thúc.
Luồng sự kiện phụ	
Điều kiện sau:	Không có

- ĐẶC TẢ USE CASE THIẾT LẬP KIỂU MẪU CHO TỪNG ĐIỂM BÁN MẶC ĐỊNH

- **Bảng 2.14** Đặc tả use case tải file

Use case:UC_006_ Thiết lập kiểu mẫu cho từng điểm bán	
---	--

Mục đích:	Thay đổi kiểu hiển thị cho từng điểm bán trên bản đồ sao cho phù hợp với ngữ cảnh đề ra
Mô tả:	
Tác nhân:	Người dùng
Điều kiện trước:	Người dùng đang đăng nhập trong hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Chon điểm bán cần thay đổi 2. Chọn kiểu mẫu 3. Thay đổi tùy chọn 4. Use case kết thúc.
Luồng sự kiện phụ	
Điều kiện sau:	Không có

- Đặc tả use case cập nhật địa điểm

Bảng 2.15 Đặc tả use case tải file

Use case:UC_007_Cập nhập địa điểm	
Mục đích:	Thêm các địa điểm thu thập được vào trong cơ sở dữ liệu
Mô tả:	Các thông tin địa điểm sẽ được theo vào cơ sở dữ liệu thông qua khảo sát thực tế, các nguồn tài liệu ...
Tác nhân:	Tổ chức
Điều kiện trước:	
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Thao tác với CSDL
Luồng sự kiện phụ	
Điều kiện sau:	Không có

2.4. Mô hình quan hệ thực thể

2.4.1. Mô tả các bảng

Bảng 2.16 Bảng thống kê(SiteSelectionAnalysist)

STT	Tên thuộc tính	Kiểu thuộc tính	Mô tả thuộc tính
1	<u>Id</u>	Int	Mã điểm
2	PoiStats	Nvarchar	Thống kê
3	CreationTime	Datetime	Thời gian tạo
4	CreatorUserId	Bigint	Id người tạo
5	LastModificationTime	Datetime	Thời gian sửa cuối cùng
6	LastModifierUserId	Bigint	Id người sửa cuối cùng
7	LocationId	Nvarchar(MAX)	Mã liên kết
8	TenantId	Int	Thời gian sửa

Bảng 2.17 Bảng nhóm(Group)

STT	Tên thuộc tính	Kiểu thuộc tính	Mô tả thuộc tính
1	<u>Id</u>	Int	Mã điểm
2	AppId	Nvarchar	Thống kê
3	CreationTime	Datetime	Thời gian tạo
4	CreatorUserId	Bigint	Id người tạo
5	DeleteUserId	Bigint	Người xóa
6	DeletionTime	Datetime	Thời gian xóa
7	GroupId	Nvarchar	Mã liên kết
8	GroupName	Nvarchar	Tên nhóm
9	IsDeleted	Bit	Đã xóa

10	LastModificationTime	Datetime	Thời gian sửa cuối cùng
11	LastModifierUserId	Bigint	Id người sửa cuối cùng
12	Sorter	Nvarchar(MAX)	Mã liên kết
13	TenantId	Int	Thời gian sửa

Bảng 2.18 Bảng điểm bán(SiteSelectionLocation)

STT	Tên thuộc tính	Kiểu thuộc tính	Mô tả thuộc tính
1	<u>Id</u>	Int	Mã điểm
2	Name	Nvarchar	Tên điểm
3	Address	Nvarchar	Địa chỉ
4	Description	Nvarchar	Miêu tả
5	Lon	Float	Kinh độ
6	Lat	Float	Vĩ độ
7	GroupId	Nvarchar	Mã nhóm
8	Files	Nvarchar	Tệp tin
9	AppId	Nvarchar	Mã ứng dụng
10	LastModificationTime	Datetime	Thời gian sửa cuối cùng
11	LastModifierUserId	Bigint	Id người sửa cuối cùng
12	Sorter	Int	Sắp xếp
13	TenantId	Int	Mã tổ chức
14	CreationTime	Datetime	Thời gian tạo
15	CreatorUserId	Bigint	Id người tạo
16	LocationId	Nvarchar	Mã liên kết
17	Style	Nvarchar	Kiểu hiển thị

Bảng 2.19 Bảng cài đặt mặc định (SiteSelectionSetting)

TT	Tên cột	Kiểu dữ liệu	Mô tả
1	Id	Int	Id
2	CreationTime	datetime2(7)	Ngày tạo
3	CreatorUserId	Bigint	Mã người tạo
4	DeleterUserId	Bigint	Mã người xóa
5	LastModificationTime	datetime2(7)	Ngày sửa đổi cuối
6	LastModifierUserId	Bigint	Mã người sửa đổi cuối
7	Color	nvarchar(MAX)	Màu
8	Size	Float	Kích thước
9	Border	Float	Viền
10	Name	nvarchar(MAX)	Tên
11	Value	nvarchar(MAX)	Giá trị
12	Type	nvarchar(MAX)	Kiểu
13	PositionPopup	nvarchar(MAX)	Vị trí hiển thị cửa sổ
14	MapName	nvarchar(MAX)	Tên map
15	AppId	nvarchar(MAX)	Id ứng dụng
16	TenantId	Int	Mã tổ chức

Bảng 2.20 Bảng vị trí điểm (Poi)

STT	Tên thuộc tính	Kiểu thuộc tính	Mô tả thuộc tính
1	<u>MaDoiTuong</u>	Bigint	Mã điểm
2	MaLienKet	Nvarchar	Tên điểm
3	TenDoiTuong	Nvarchar	Địa chỉ
4	NhanBanDo	Nvarchar	Miêu tả
5	xDaiDien	Float	Kinh độ

6	yDaiDien	Float	Vĩ độ
7	Geom	Geometry	Mã nhóm
8	TrangThai	Int	Tệp tin
9	Osm_id	Nvarchar	Mã ứng dụng
10	Code	Int	
11	Fclass	Nvarchar	Thời gian sửa cuối cùng
12	Name	Nvarchar	Id người sửa cuối cùng
13			

Bảng 2.21 Bảng quản lý người dùng (AbpUser)

TT	Tên cột	Kiểu dữ liệu	Mô tả
1	Id	bigint	Mã người dùng
2	CreationTime	datetime2(7)	Ngày tạo
3	CreatorUserId	bigint	Mã người tạo
4	DeleterUserId	bigint	Mã người xóa
5	EmailAddress	varchar(256)	Địa chỉ email
6	EmailConfirmationCode	varchar(328)	Mã xác nhận email
7	IsActive	Bit	Đang hoạt động
8	IsEmailConfirmed	Bit	Email có xác nhận không
9	IsLockoutEnabled	Bit	Đã khóa
10	Name	nvarchar(64)	Tên
11	Password	nvarchar(128)	Mật khẩu
12	ProfilePictureId	uniqueidentifier	Ảnh đại diện
13	Surname	nvarchar(64)	Họ
14	TenantId	Int	Mã tổ chức

15	UserName	nvarchar(256)	Tên đăng nhập
----	----------	---------------	---------------

Bảng 2.22 Bảng quản lý Tenant (AbpTenant)

TT	Tên cột	Kiểu dữ liệu	Mô tả
1	Id	Int	Mã tổ chức
2	CreationTime	datetime2(7)	Ngày tạo
3	CreatorUserId	Bigint	Mã người tạo
4	DeleterUserId	Bigint	Mã người xóa
5	LastModificationTime	datetime2(7)	Ngày sửa đổi cuối
6	LastModifierUserId	Bigint	Mã người sửa đổi cuối
7	IsActive	Bit	Đang hoạt động
8	Name	nvarchar(64)	Tên
9	TenancyName	nvarchar(64)	Tên thuê bao
10	LogoId	uniqueidentifier	Mã logo
11	LogoFileType	nvarchar(64)	Loại tệp logo
12	EditionId	Int	Mã gói dịch vụ

Bảng 2.23 Bảng danh sách quyền (ApbRoles)

TT	Tên cột	Kiểu dữ liệu	Mô tả
1	Id	Int	Mã quyền
2	CreationTime	datetime2(7)	Ngày tạo
3	CreatorUserId	bigint	Mã người tạo
4	DeleterUserId	bigint	Mã người xóa
5	DisplayName	Nvarchar(64)	Tên hiển thị
6	IsDefault	Bit	Mặc định
7	IsStatic	Bit	

8	Name	Nvarchar(64)	Tên
9	Normalized Name	Nvarchar(64)	
10	TenantId	Int	Mã tổ chức

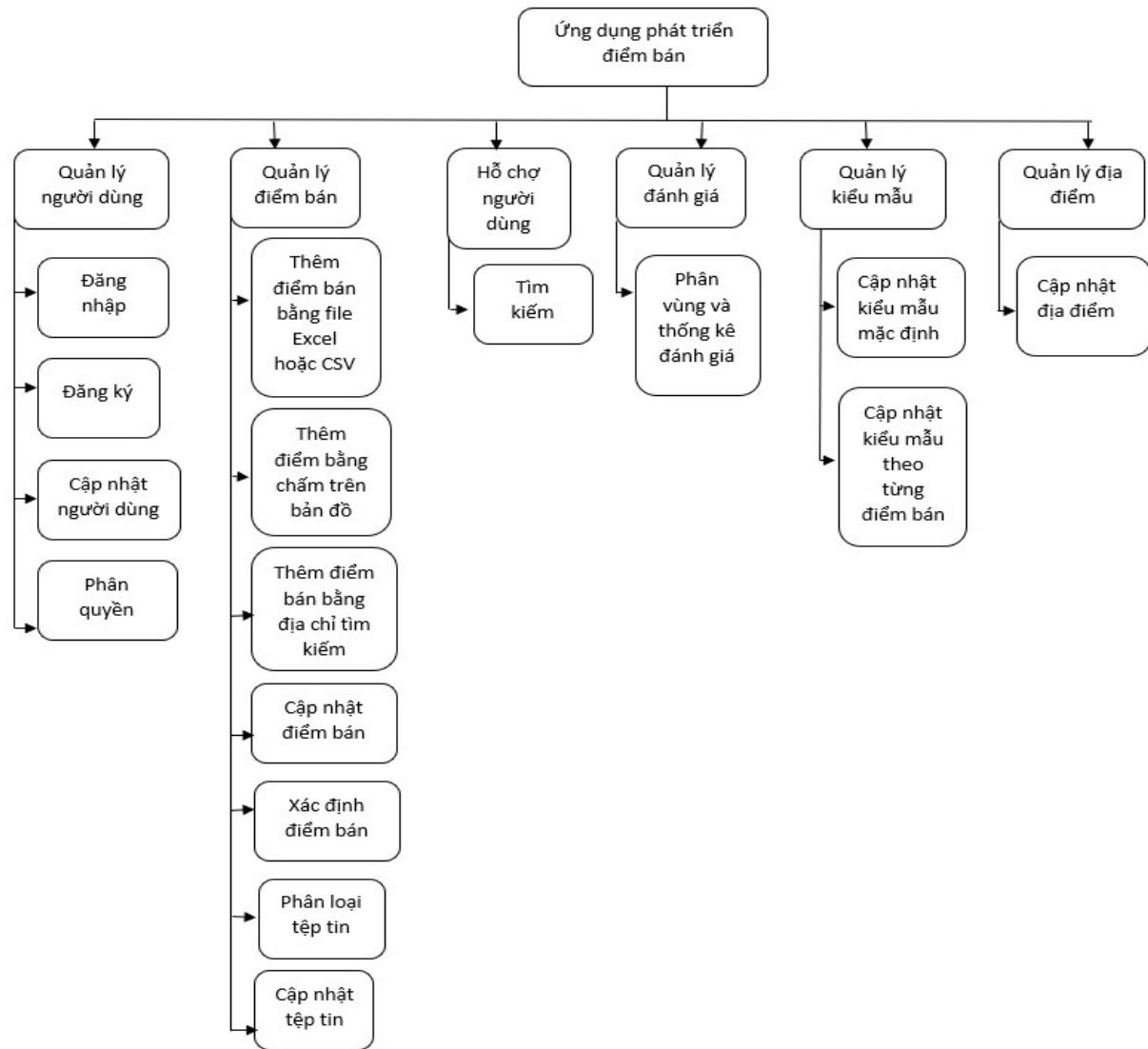
Bảng 2.14 Bảng quan hệ người dùng phòng ban (AbpUserOrganizationUnits)

TT	Tên cột	Kiểu dữ liệu	Mô tả
1	Id	Int	Mã quyền
2	CreationTime	datetime2(7)	Ngày tạo
3	CreatorUserId	bigint	Mã người tạo
4	OrganizationId	bigint	Mã phòng ban
5	TenantId	Nvarchar(64)	Mã tổ chức
6	UserId	Bit	Mã người dùng
7	IsDeleted	Bit	Xóa

Bảng 2.15 Bảng quan hệ người dùng quyền (AbpUserRoles)

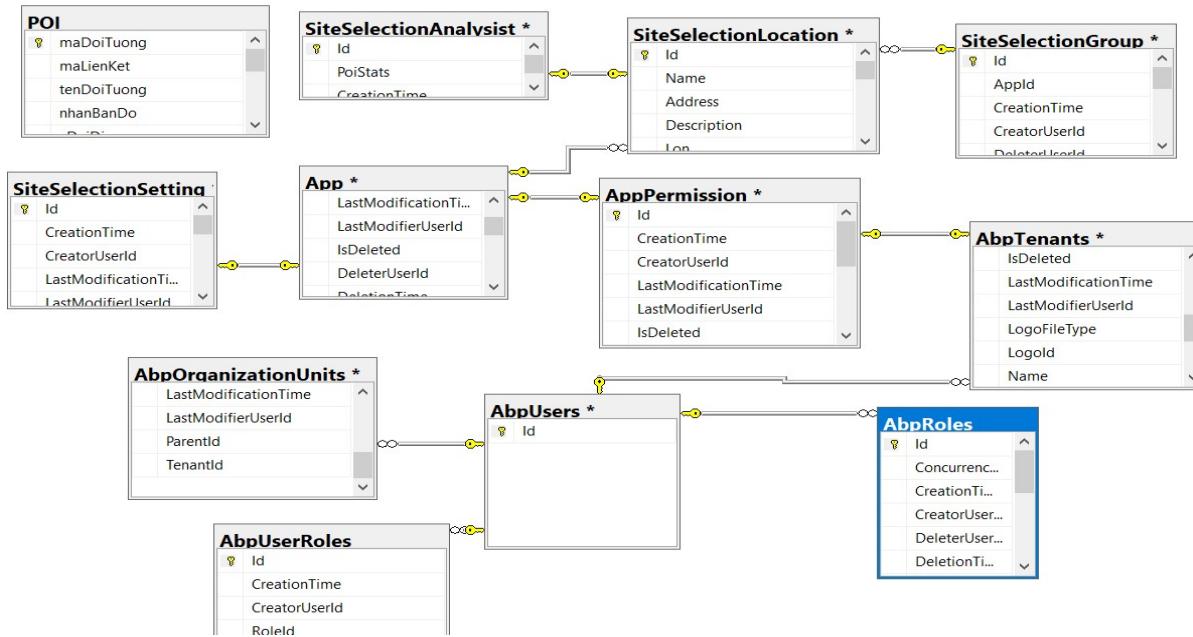
TT	Tên cột	Kiểu dữ liệu	Mô tả
1	Id	Int	Mã quyền
2	CreationTime	datetime2(7)	Ngày tạo
3	CreatorUserId	bigint	Mã người tạo
4	RoleId	bigint	Mã quyền
5	TenantId	Int	Mã tổ chức
6	UserId	Int	Mã người dùng

2.4.2. Sơ đồ phân rã chức năng



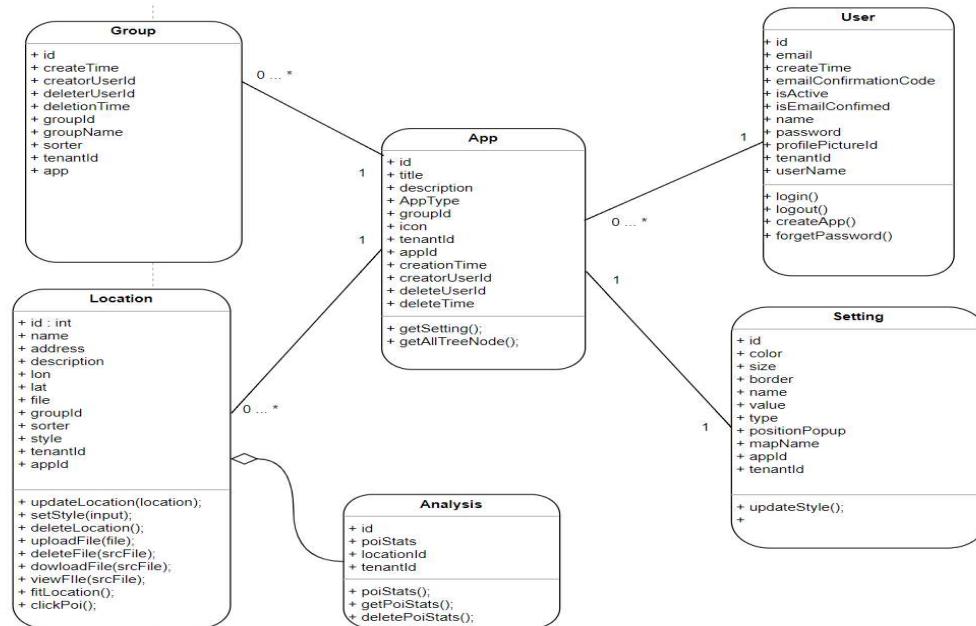
Hình 2.16 Sơ đồ phân rã chức năng

2.4.3. Sơ đồ thực thể quan hệ



Hình 2.17 Sơ đồ thực thể quan hệ

2.4.4. Biểu đồ lớp



Hình 2.12 Biểu đồ lớp

CHƯƠNG 3. CÀI ĐẶT VÀ THỬ NGHIỆM WEBSITE

3.1. Môi trường cài đặt

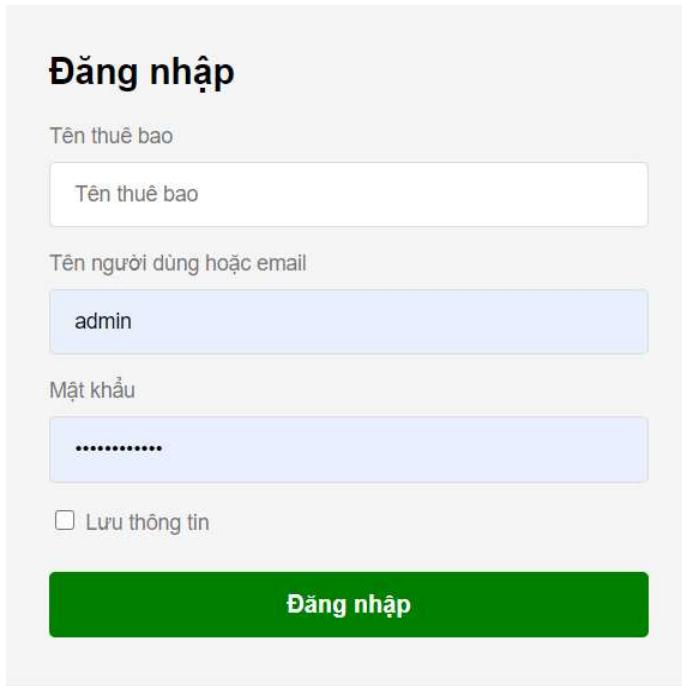
Website được phát triển trên nền tảng ASP.NET Zero, ngôn ngữ lập trình TypeScript, Framework AngularJS 1 và hệ quản trị cơ sở dữ liệu MySQL.

3.2. Giao diện và chức năng

3.2.1. Trang đăng nhập quản lý dịch vụ

Giao diện quản lý đăng nhập dịch vụ , chỉ người được cho phép (được cấp quyền) mới có thể truy cập và test các dịch vụ của hệ thống website

Yêu cầu người dùng nhập tên thuê bao , tổ chức và tài khoản người dùng



Đăng nhập

Tên thuê bao

Tên thuê bao

Tên người dùng hoặc email

admin

Mật khẩu

.....

Lưu thông tin

Đăng nhập

Hình 3.1 Trang đăng nhập quản lý dịch vụ

3.2.2. Trang quản lý dịch vụ

Người dùng có thể dễ dàng tìm kiếm các dịch vụ của hệ thống

Hình 3.2 Danh sách dịch vụ

Có thể test dịch vụ trực tiếp bằng cách truyền các tham số vào dịch vụ . dịch vụ sẽ trả ra kết quả tương ứng với tham số vừa truyền vào

```

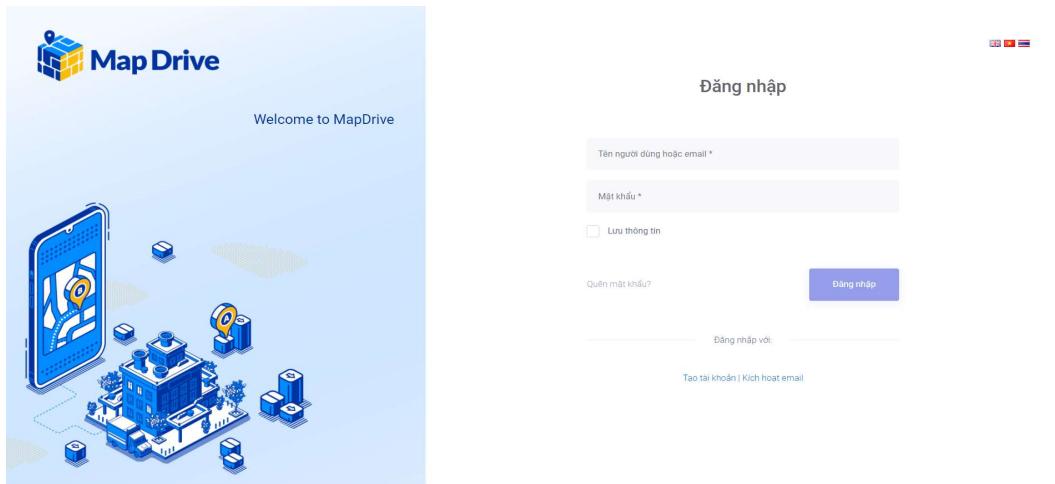
{
  "result": [
    {
      "id": 34,
      "appId": "1057d0b13b05",
      "name": "1",
      "parent": "3eac4c2538fb",
      "nameEn": "Nguyen Thanh Truc Giang"
    }
  ]
}

```

Hình 3.3 Trải nghiệm dịch vụ

3.2.3. Trang đăng nhập

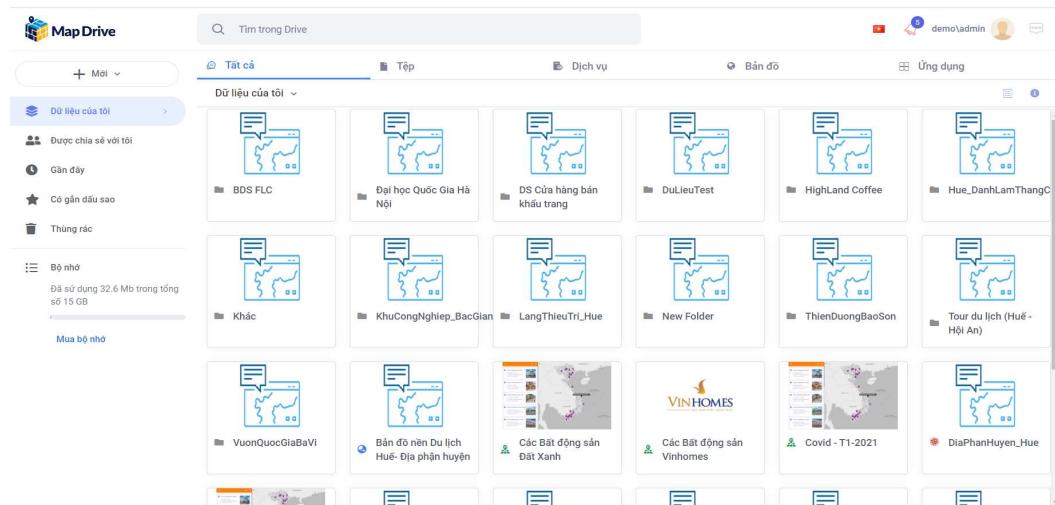
Giao diện đăng nhập của người dùng thuộc tổ chức *fivestar*. Trên giao diện, người dùng nếu chưa có tài khoản thì lựa chọn “Tạo tài khoản” để thêm tài khoản vào tổ chức *fivestar*. Ngoài ra cũng có đầy đủ các chức năng như “Quên mật khẩu”, “Kích hoạt Email”, “Lưu thông tin”. (xem Hình 7.4)



Hình 3.4 Giao diện đăng nhập

3.2.4. Quản lý ứng dụng

Sau khi đăng nhập thành công, người dùng được chuyển tới trang *Quản lý ứng dụng*. Như Hình 3.3 và Hình 3.4, mọi thao tác thêm, sửa, xóa, tìm kiếm, chia sẻ, mở, sửa ứng dụng đều có ở trên giao diện. Ở sidebar bên trái, người dùng được thêm, lọc ứng dụng theo các mục như Hình 3.3, 3.4. Tương tự như với thanh tab-bar ở bên phải.



Hình 3.5 Giao diện quản lý ứng dụng lưới

Để thêm mới thư mục, người dùng nhấp chuột nút Mới và lựa chọn “Thêm thư mục”. Form thêm thư mục như Hình 3.6.

Hình 3.6. Form Tạo thư mục.

Để thêm ứng dụng Shortlist, người dùng nhấp chuột nút Mới và lựa chọn “Thêm bản đồ” -> “Shortlist”. Form thêm Shortlist như Hình 3.7. Người dùng chọn mẫu Shortlist rồi điền tên ứng dụng để tạo ứng dụng Shortlist.

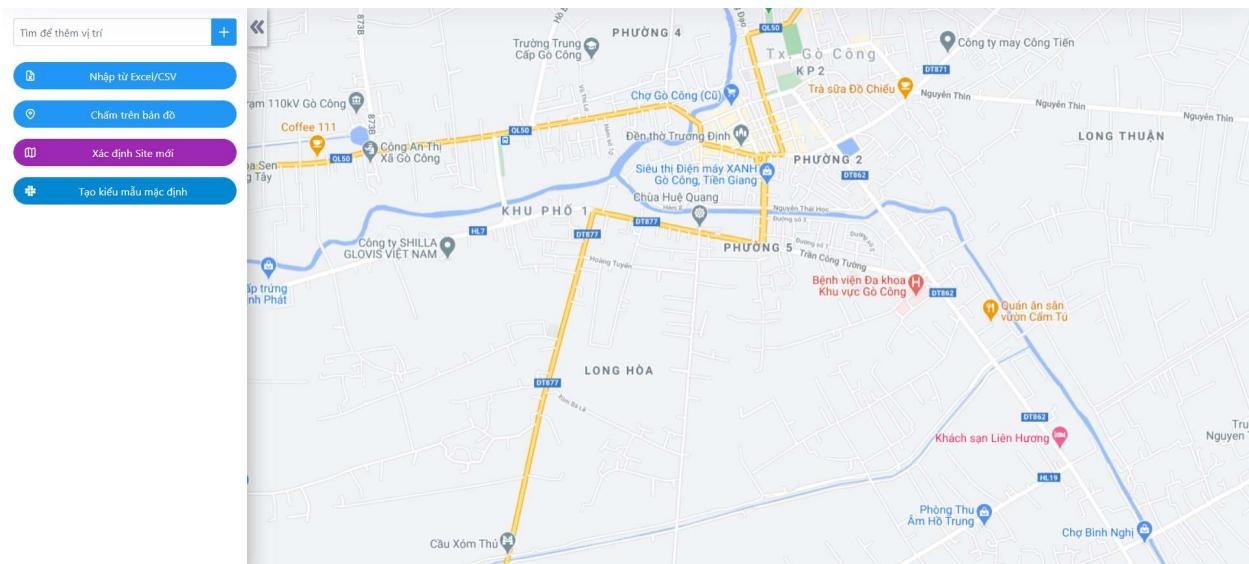
Hình 3.7. Form Tạo ứng dụng Shortlist.

3.2.5. Giao diện chính website

Màn hình giao diện chính gồm có logo của tổ chức ở phía trên bên trái màn hình, bên phải là thông tin người dùng, phía dưới là giao diện chức năng và bản đồ để hiện thị dữ liệu tương ứng.

Giao diện cung cấp cho người dùng 4 chức năng chính :

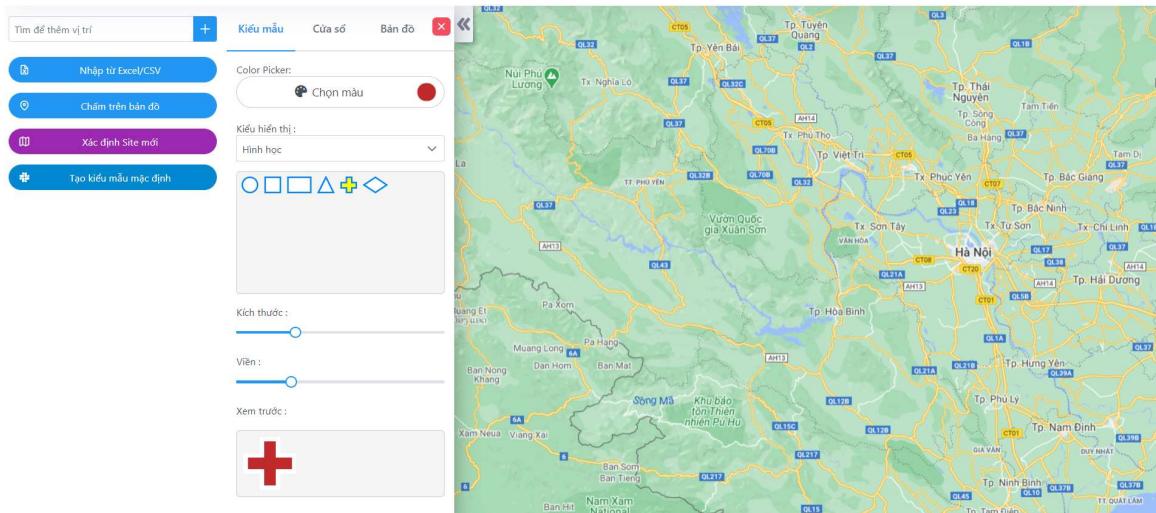
- Tìm vị trí để thêm điểm
- Thêm điểm bằng file Excel hoặc CSV
- Chấm trên bản đồ để thêm điểm
- Tạo nhóm



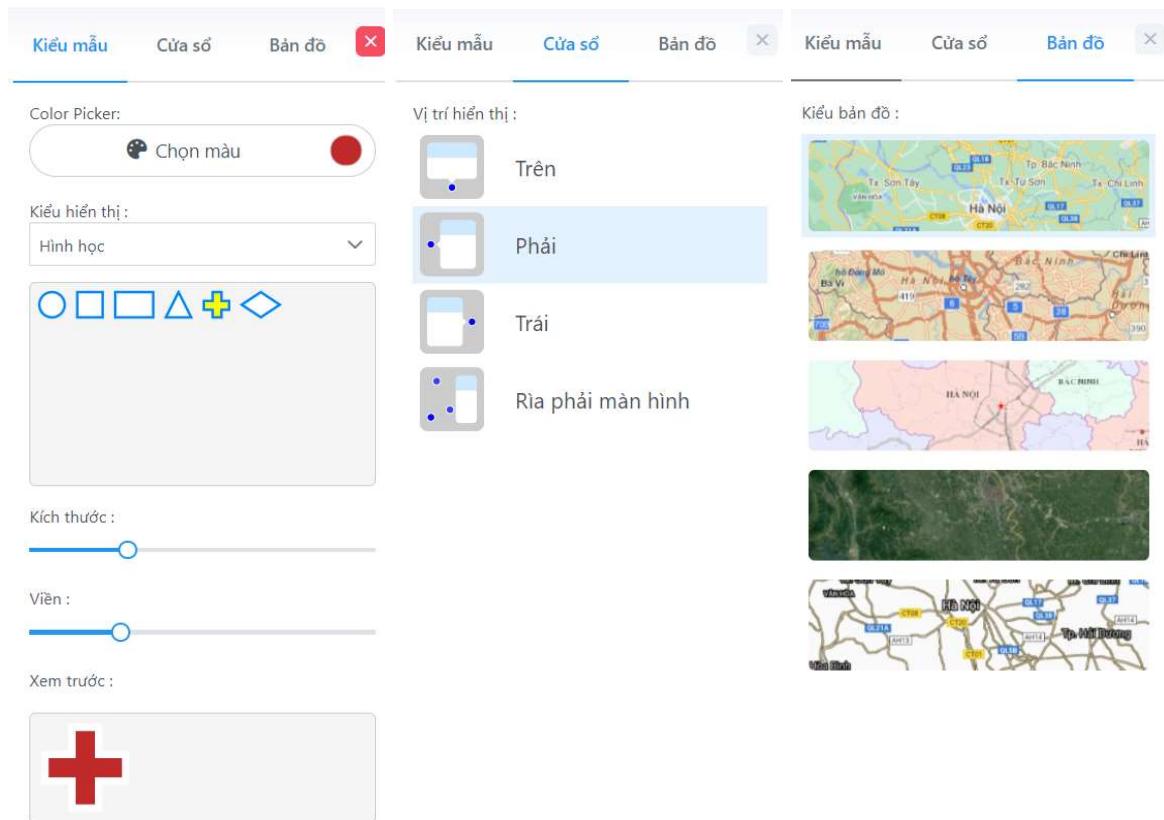
Hình 3.8 Giao diện chính

3.2.6. Thiết lập cài đặt

Ứng dụng có 3 giao diện thiết lập cài đặt mặc định: Kiểu mẫu, cửa sổ và bản đồ



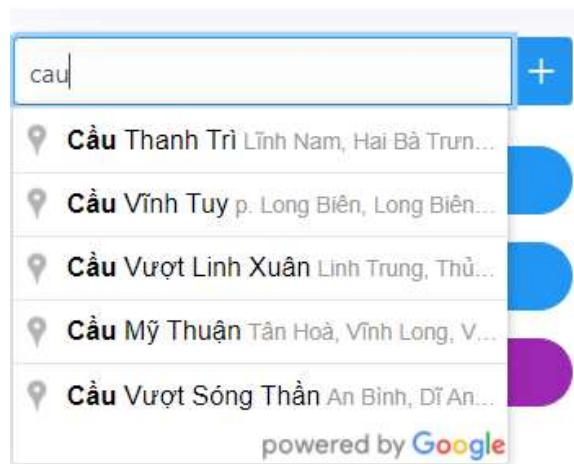
Hình 3.9 Giao diện cài đặt



Hình 3.10 Giao diện tùng chúc năng cài đặt

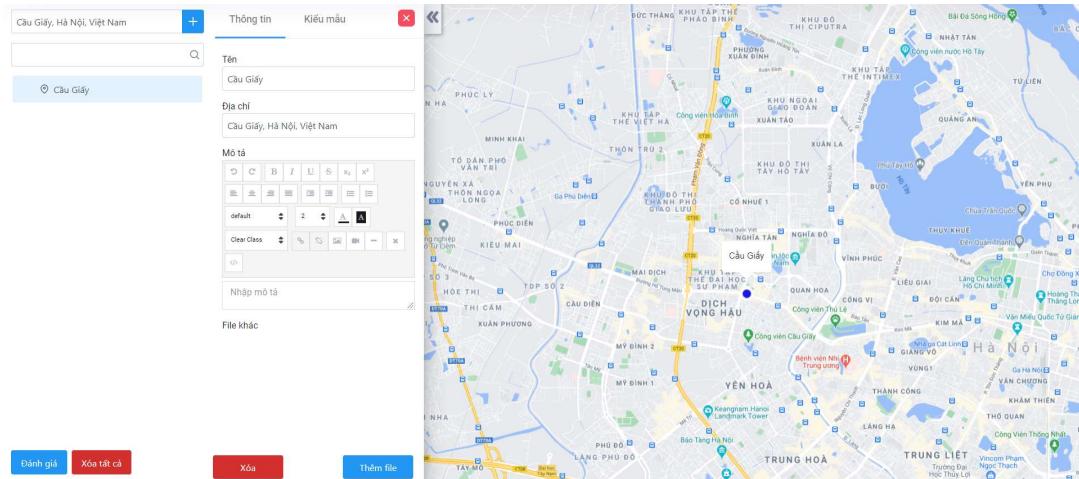
3.2.7. Tìm vị trí để thêm điểm

Người dùng nhập vị trí muốn thêm, hệ thống sẽ gợi ý các vị trí giúp người dùng dễ dàng lựa chọn.



Hình 3.11 Tự động nhập

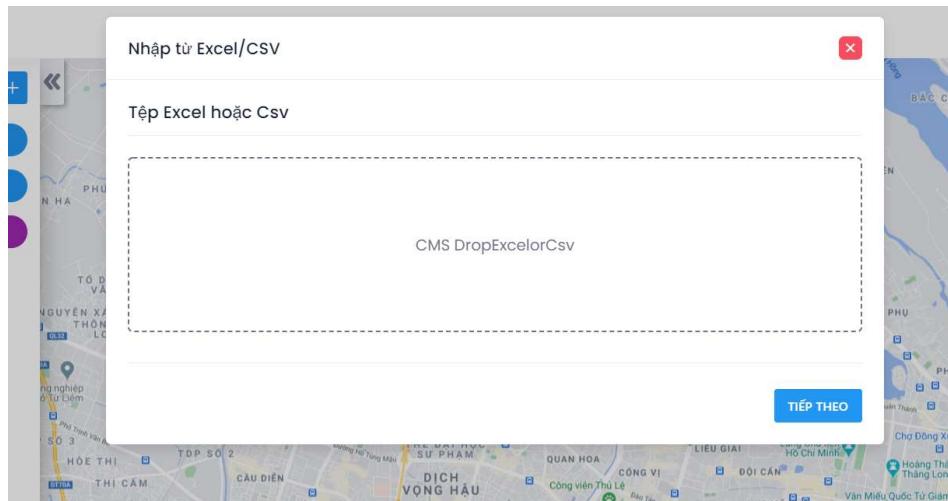
Sau khi chọn địa điểm, hệ thống sẽ tự động thêm file vào bản đồ , bản đồ tự động di chuyển chính giữa đến điểm mới tạo



Hình 3.12 Sau khi thêm điểm bằng tìm kiếm

3.2.8. Nhập điểm bằng file Excel hoặc file CSV

Khi chọn nhập điểm bằng file excel hoặc file csv, hệ thống hiện lên form yêu cầu người dùng nhập theo từng bước.



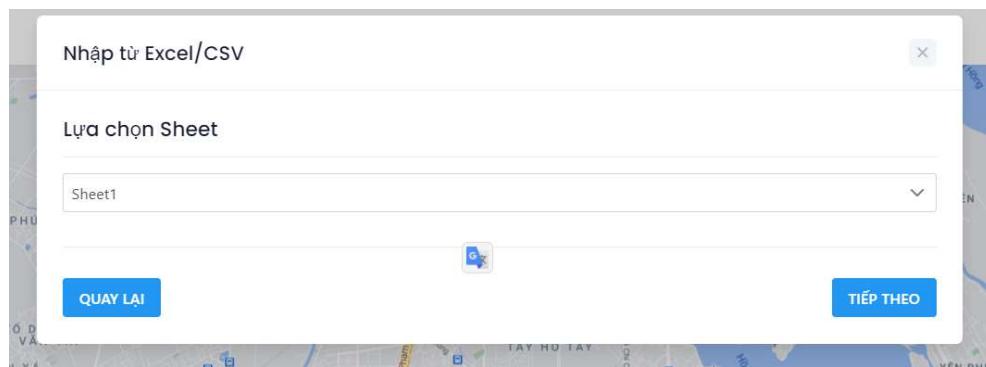
Hình 3.13 Nhập file Excel/CSV

Bước 1 : Chọn file:

Người dùng có thể nhập file bằng cách nhấn hoặc kéo thả vào vùng nhập file.

Khi người dùng chọn file, hệ thống tự động nhảy sang bước tiếp theo cho người dùng thực hiện thao tác một cách nhanh chóng hơn

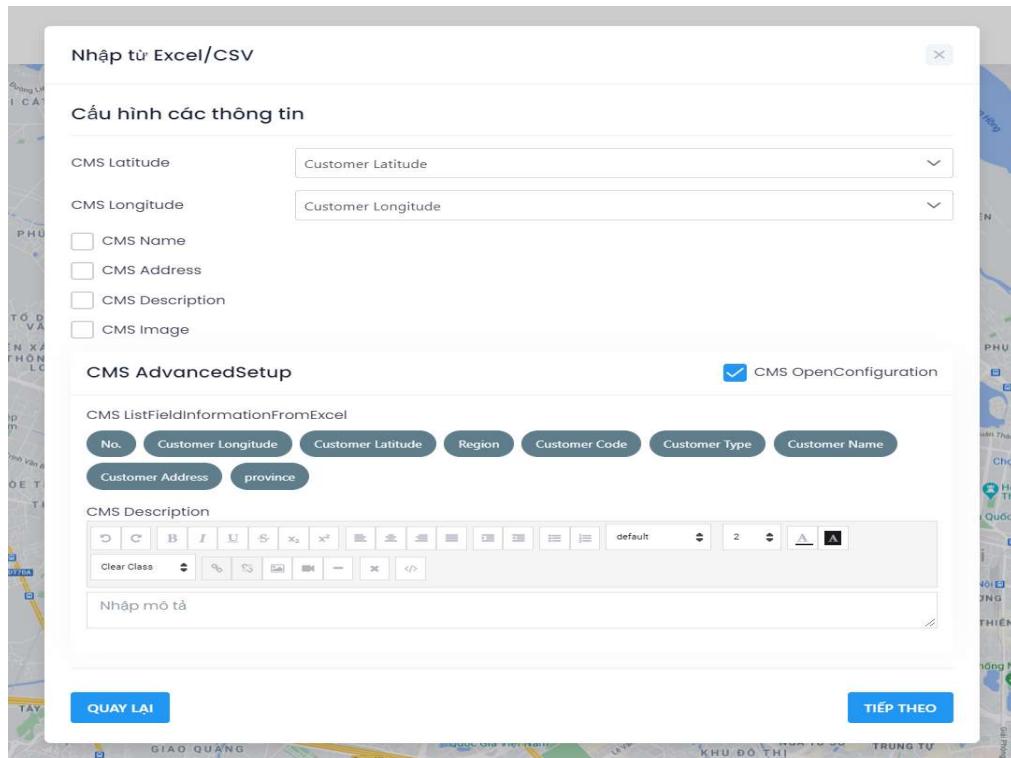
Bước 2 :Chọn Sheet



Hình 3.14 Chọn Sheet

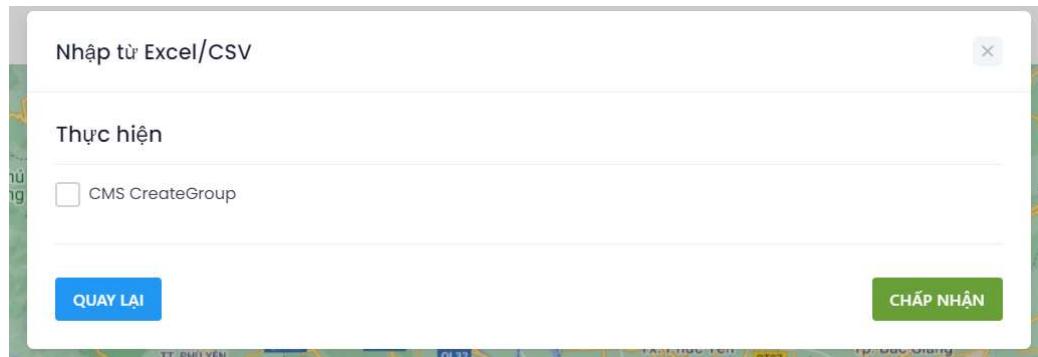
Bước 3 : Cấu hình thông tin hiển thị cho các điểm

Người dùng sẽ chọn các trường cần quan tâm để hiển thị thông tin cho các điểm



Hình 3.15 Nhập thông tin hiển thị

Bước 4 : Chọn nhóm

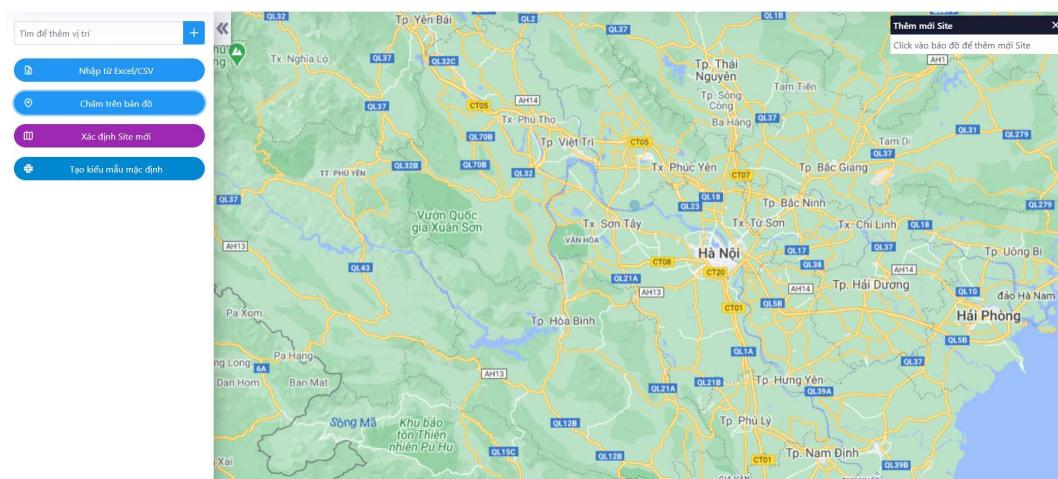


Hình 3.16 Phân nhóm điểm bán

Sau khi hoàn thành , hệ thống sẽ tự động thêm danh sách điểm lên bản đồ

3.2.9. Thêm điểm bằng cách chấm trên bản đồ

Khi kích hoạt chức năng thêm điểm bằng chấm trên bản đồ, con trỏ sẽ tạo 1 chấm mờ để hiển thị, giúp người dùng xác định vị trí dễ dàng hơn



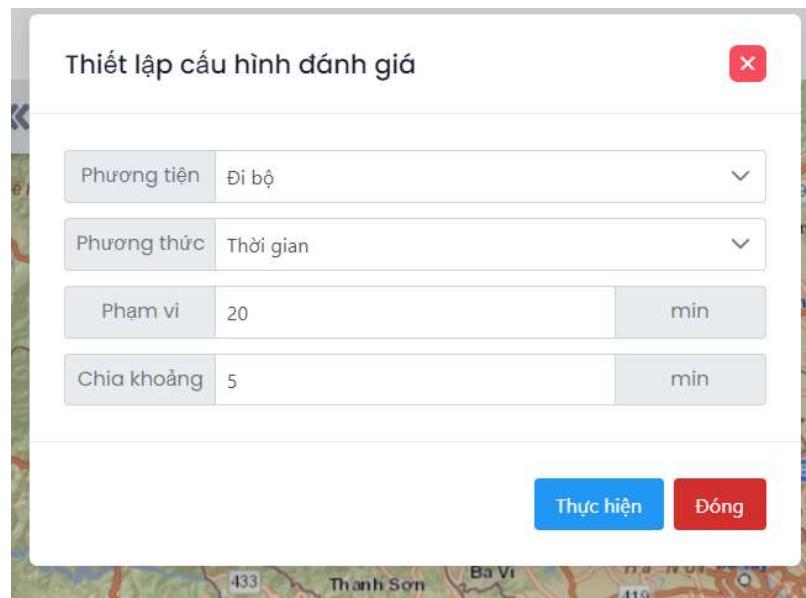
Hình 3.17 Thêm nhóm bằng chấm trên bản đồ

Khi chuột được nhấn, ứng dụng sẽ thêm điểm lên giao diện và lưu lại vào cơ sở dữ liệu

3.2.10. Đánh giá các điểm đã thêm

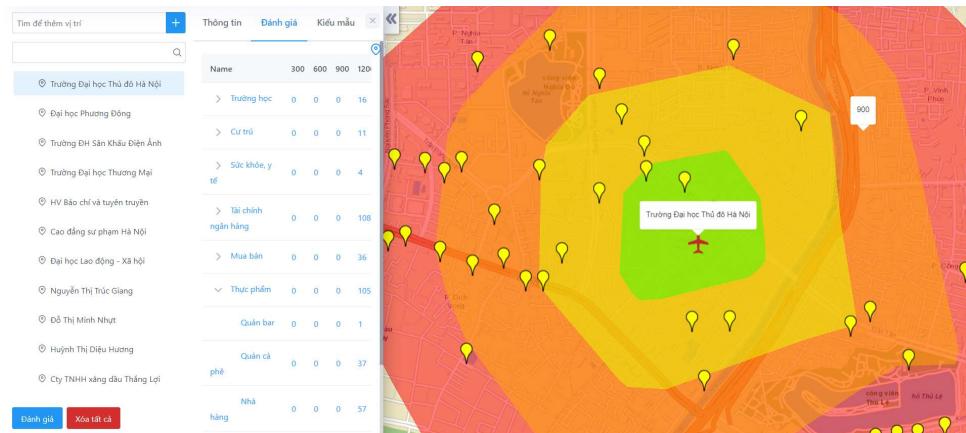
Khi việc thêm các điểm đã hoàn thành, người dùng nhấn vào nút đánh giá để bắt đầu tiến hành đánh giá

Hệ thống hiển thị cửa sổ tùy chọn cầu hình đánh giá, người dùng sẽ phải chọn 4 trường : phương tiện , phương thức, phạm vi, chia khoảng



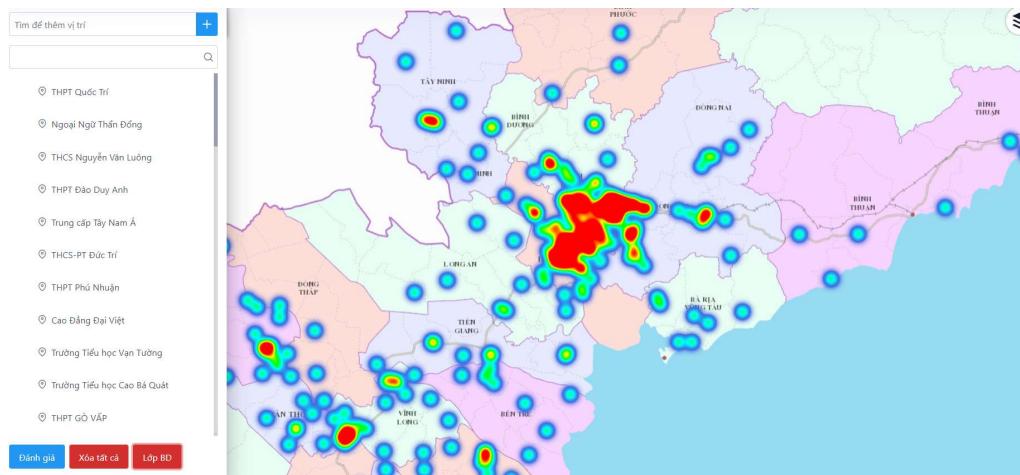
Hình 3.18 Cửa sổ thiết lập đánh giá

Khi người dùng nhấn thực hiện hệ thống sẽ phân tích và chia kết quả như hình **Hình 6.16**

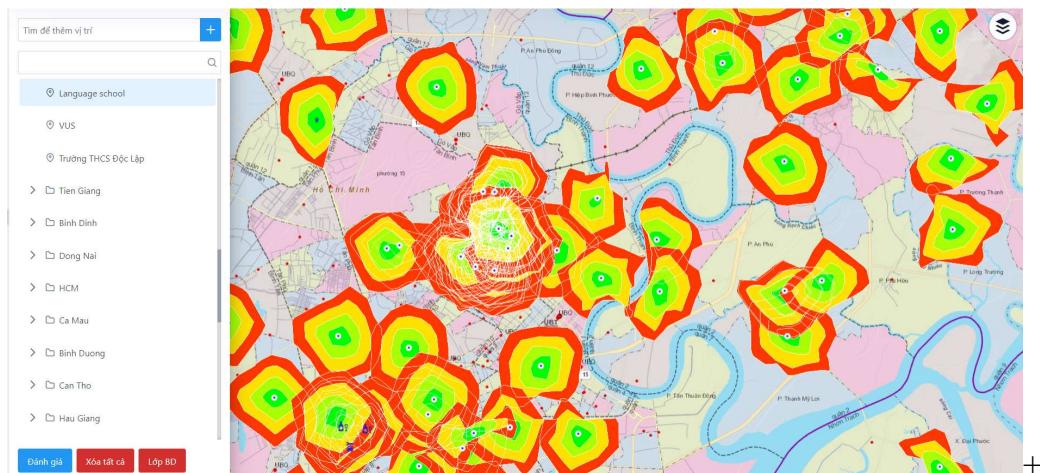


Hình 3.19 Kết quả đánh giá điểm bán

Ngoài ra người dùng có thể xem đánh giá của tất cả các điểm bán theo lớp bán đồ như bản đồ phủ, bản đồ nhiệt, bản đồ theo bán kính



Hình 3.20 Bản đồ nhiệt



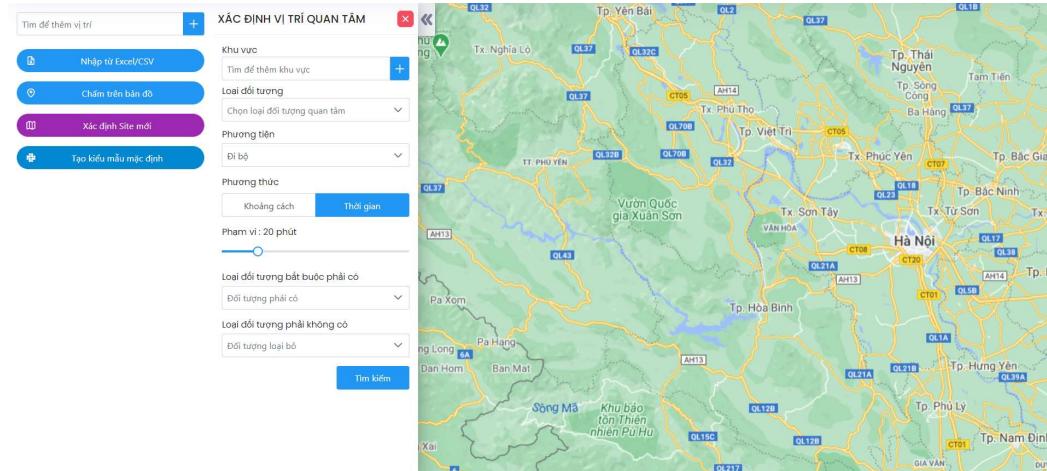
Hình 3.21 Bản đồ phủ



Hình 3.22 Bản đồ theo độ phủ

3.2.11.Xác định Site mới

Đây là giao diện mà hệ thống sẽ giải quyết quy trình thứ 1 – lựa chọn điểm bán, đã được nêu ở 3.1



Hình 3.23 Xác định điểm bán

Các thông tin cần nhập bắt buộc, hệ thống sẽ yêu cầu người dùng nhập vùng quan tâm, loại đối tượng quan tâm. Với các giá trị này hệ thống sẽ chả ra cho người dùng các điểm thuộc loại đối tượng trong khu vực đã trộn

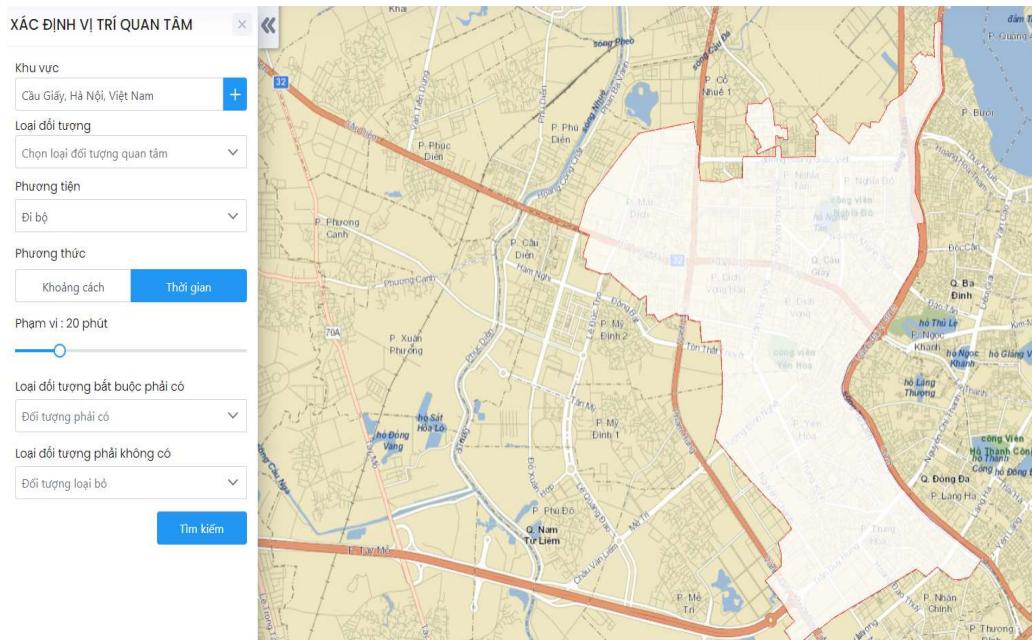
Ngoài ra người dùng có thể nhập thêm điều kiện lọc như phương tiện, phương thức, phạm vi, loại đối tượng bắt buộc có, loại đối tượng bắt buộc phải không có

Ví dụ : Trong khu vực cầu giấy , tìm các trường đại học mà trong khoảng 20 phút đi bộ bắt buộc phải có quán ăn và không có quán KFC

Với các giá trị vừa nhập trên, hệ thống sẽ đánh giá chi tiết thông kê các loại đối tượng theo từng điểm

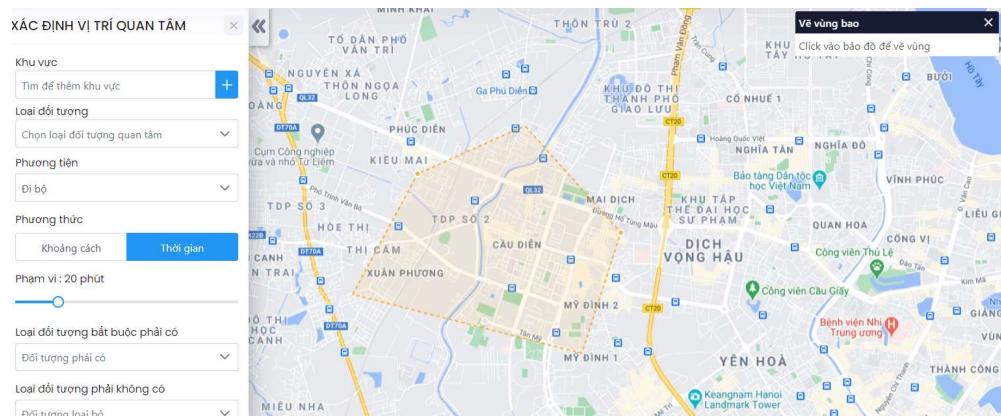
Để lựa chọn khu vực quan tâm, ta có 2 cách để chọn đó là nhập tên địa chỉ khu vực hoặc sử dụng chức năng vẽ vùng

- Với chức năng nhập tên khu vực, người dùng chỉ cần nhập tên, hệ thống sẽ tự động xác định vùng bao cụ thể xem **hình 3.24**



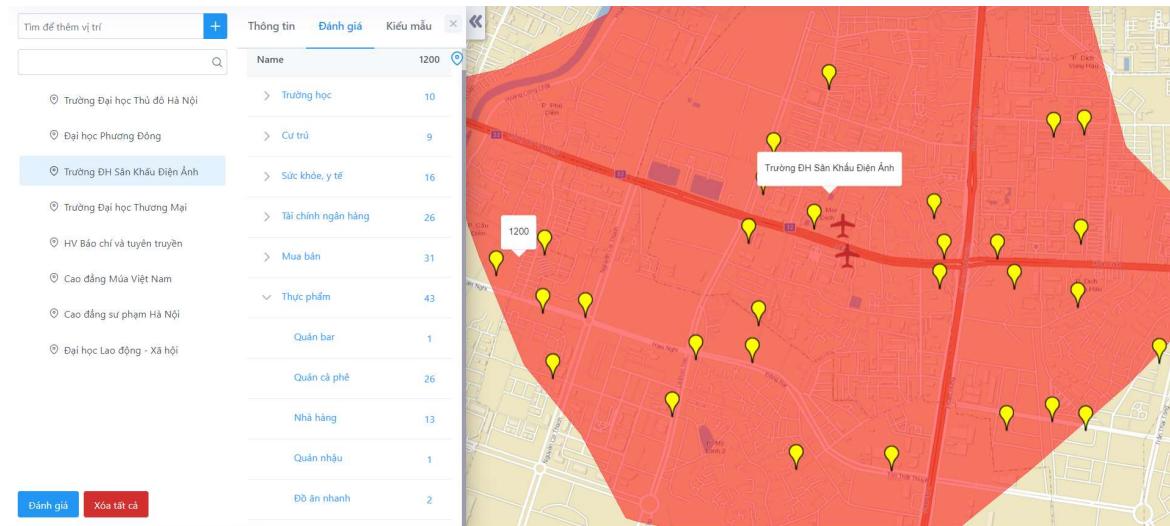
Hình 3.24 Xác định vùng bao tại khu vực cầu giấy

- Với chức năng vẽ, người dùng ấn vào nút có hình dấu “+” bên phải ô tìm kiếm khu vực và nhấp trên bản đồ để bắt đầu vẽ



Hình 3.25 Vẽ vùng bao

Khi nhập các tùy chọn và nhấp tìm kiếm hệ thống sẽ trả ra danh sách các điểm đã được đánh giá, người dùng có thể xem thông tin đánh giá bằng cách nhấp vào điểm xác định rồi chọn tap đánh giá



Hình 3.26 Đánh giá điểm bán

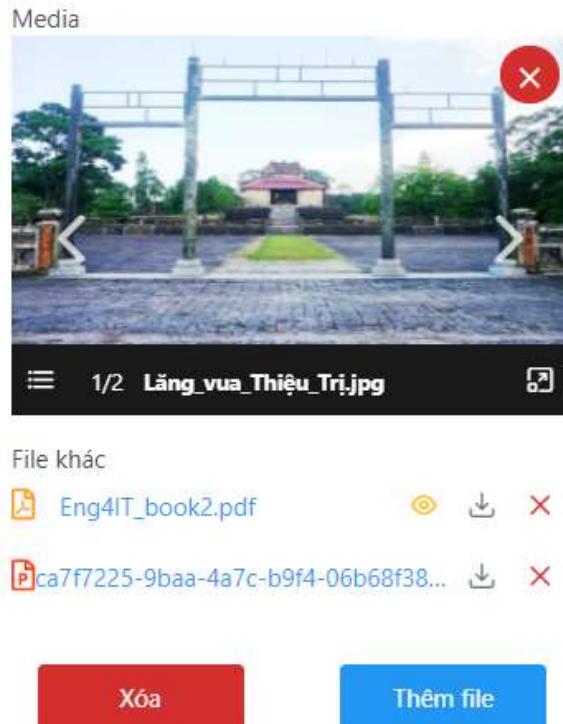
3.2.12. Cập nhật điểm bán

- Thông tin điểm bán
 - Người dùng có thể cập nhật trực thông tin hoặc xóa điểm bán ngày trên giao diện web

This figure shows a detailed view of a real estate listing form. At the top, there is a search bar 'Tim để thêm vị trí' and a button '+'. Below it is a section for 'Thông tin' (Information) and 'Kiểu mẫu' (Type). The 'Thông tin' section contains fields for 'Tên' (Name) with the value '32a Ngõ 7 Phố Phùng Chí Kiên, Nghĩa Đô' and 'Địa chỉ' (Address) with the same value. There is also a 'Mô tả' (Description) section with a rich text editor and a text input field containing 'ipi pi-map-marker'. At the bottom, there are three buttons: 'Đánh giá' (Review), 'Xóa tất cả' (Delete all), 'Xóa' (Delete), and 'Thêm file' (Add file).

Hình 3.27 Thông tin điểm bán

- Ngoài ra có chức năng thêm file với dung lượng không quá 200mb với các định dạng text , pdf, power point, word, các loại ảnh, video, excel ...
- Một vài định dạng file có thể xem trực tiếp trên site như hình ảnh, video, và pdf, các file web được hỗ trợ sẽ có hình con mắt  , nhấp vào đó để xem ngay.



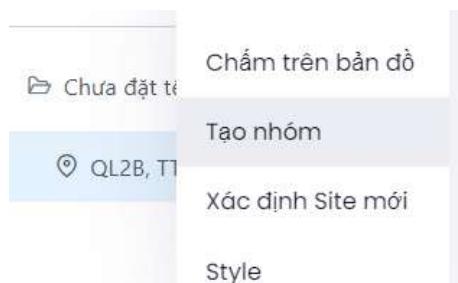
Hình 3.28 Cập nhật file theo từng điểm bán

- Tương tự với thêm web còn cung cấp cho người dùng chức năng tải file và xóa file

3.2.13. Một số chức năng khác

- Tạo nhóm

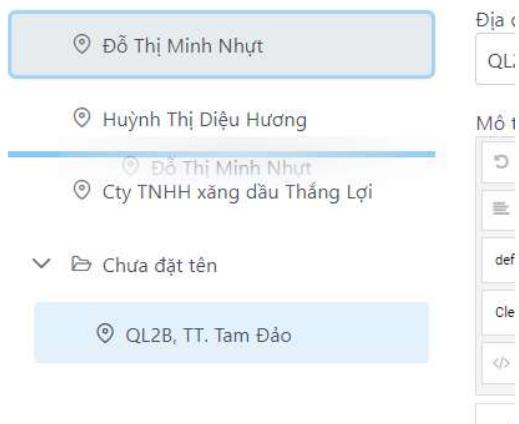
Giúp người dùng dễ dàng quản lý điểm bán hơn.



Hình 3.29 Tạo nhóm

- Kéo thả điểm

Sắp xếp lại vị trí hiển thị của điểm bán .

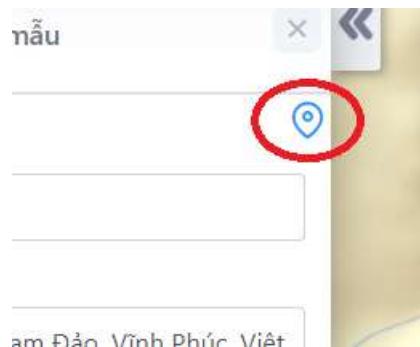


Hình 3.30 Sắp xếp điểm bán

- Di chuyển điểm bán vào giữa màn hình

Thuận tiện cho việc thao tác, chỉnh sửa, đánh giá điểm bán.

Khi người sử dụng kích hoạt, hệ thống sẽ di chuyển điểm bán vào chính giữa bản đồ đồng thời hiển thị cửa sổ nhãn tên điểm bán đó



Hình 3.31 Định vị điểm bán

CHƯƠNG 4. KẾT LUẬN

4.1. Kết quả đạt được

- Tìm hiểu ngôn ngữ và công cụ phát triển trang web
- Ứng dụng đã hoàn thành các yêu cầu được đặt ra.
- Trang web có giao diện thân thiện, dễ sử dụng.

4.2. Hạn chế

- Kinh nghiệm lập trình còn yếu kém
- Có rất nhiều kiến thức và ứng dụng mới mất nhiều thời gian để tìm hiểu

4.3. Hướng phát triển

- Hiển thị đầy đủ thông tin của các điểm
- Báo cáo
- Nâng cao bảo mật cơ sở dữ liệu
- Tiếp tục phát triển web cả về giao diện và các tính năng sao cho trải nghiệm người dùng tốt nhất
- Phát triển ứng dụng trên các nền tảng khác
- Thêm chức năng chia sẻ (Chia sẻ ứng dụng, chia sẻ đối tượng)

DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. <https://docs.mapbox.com/>
- [2]. <https://keenthemes.com/metronic/>
- [3]. <https://aspnetzero.com/>
- [4]. <https://aspnetboilerplate.com/>
- [5]. <https://www.primefaces.org/>
- [6]. <http://turfjs.org/>
- [7]. <https://docs.microsoft.com/en-us/sql/t-sql/spatial-geometry/ogc-methods-on-geometry-instances?view=sql-server-ver15>