

ĐỒ ÁN 3 (PROJECT 3)

Phần mềm quản lý dữ liệu khách hàng hỗ trợ đội tư vấn tuyển sinh.

Phạm Thành Nam

nam.pt173266@sis.hust.edu.vn

Ngành Khoa học máy tính

Chuyên ngành Công nghệ phần mềm

Giảng viên hướng dẫn: ThS.Nguyễn Mạnh Tuấn

Bộ môn: Công nghệ phần mềm

Viện: Công nghệ thông tin – Truyền thông

HÀ NỘI, 12/2020

Lời cam kết

Họ và tên sinh viên : **Phạm Thành Nam**

Điện thoại liên lạc: **0348984167** Email: **nam.pt173266@sis.hust.edu.vn**

Lớp: **KHMT-03-K62**. Hệ đào tạo : **KSCQ**.

Tôi – *Phạm Thành Nam* – cam kết Đồ án 3 (Project 3) (PROJECT 3) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của ThS.Nguyễn Mạnh Tuấn. Các kết quả nêu trong PROJECT 3 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 PROJECT 3 – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ 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 30 tháng 12 năm 2020

Tác giả PROJECT 3

Lời cảm ơn

Lời cảm ơn của sinh viên đến Ths.Nguyễn Mạnh Tuấn đã tạo điều kiện cho em hoàn thành project 3 với một quyết tâm cao nhất.

Mục lục

Lời cam kết	2
Lời cảm ơn	3
Mục lục	4
Danh mục hình vẽ.....	7
Danh mục bảng.....	9
Chương 1 Giới thiệu đề tài	10
1.1 Đặt vấn đề	10
1.2 Mục tiêu và phạm vi đề tài.....	10
1.3 Định hướng giải pháp	12
1.4 Bố cục đồ án	13
Chương 2 Khảo sát và phân tích yêu cầu	15
2.1 Khảo sát hiện trạng	15
2.2 Tổng quan chức năng.....	16
2.2.1 Biểu đồ use case tổng quan	16
2.2.2 Các biểu đồ use case phân rã.....	18
2.2.3 Quy trình nghiệp vụ.....	25
2.3 Đặc tả chức năng.....	26
2.3.1 Đặc tả use case thêm khách hàng	26
2.3.2 Đặc tả use case thêm vai trò	28
2.3.3 Đặc tả use case thêm giao dịch.....	30

2.3.4 Đặc tả use case cấp quyền người dùng	32
2.4 Yêu cầu phi chức năng.....	33
2.4.1 Yêu cầu về chức năng kỹ thuật.....	33
2.4.2 Đáp ứng yêu cầu về bảo mật	33
2.4.3 Yêu cầu về giao diện	33
Chương 3 Công nghệ sử dụng.....	34
3.1.1 Hệ quản trị CSDL MySQL.....	34
3.1.2 Java Spring Boot	36
3.1.3 JSON Web Token (JWT)	37
3.1.4 Mã hóa SHA	39
Chương 4 Phát triển và triển khai ứng dụng	41
4.1 Thiết kế kiến trúc	41
4.1.1 Mô hình kiến trúc ba lớp	41
4.1.2 Thiết kế tổng quan.....	46
4.1.3 Thiết kế chi tiết gói.....	48
4.2 Thiết kế chi tiết	53
4.2.1 Thiết kế giao diện	53
4.2.2 Thiết kế lớp.....	62
4.2.3 Thiết kế cơ sở dữ liệu	65
4.3 Xây dựng ứng dụng	69
4.3.1 Thư viện và công cụ sử dụng	69
4.3.2 Kết quả đạt được.....	69
4.3.3 Minh hoạ các chức năng chính.....	70
Chương 5 Các giải pháp và đóng góp nổi bật.....	73
5.1 Xây dựng phần mềm hỗ trợ quản lý khách hàng	73

5.1.1 Bài toán đặt ra.....	73
5.1.2 Giải quyết bài toán	73
5.1.3 Kết quả đạt được.....	75
Chương 6 Kết luận và hướng phát triển.....	76
6.1 Kết luận.....	76
6.2 Hướng phát triển	76

Danh mục hình vẽ

Hình 2.1	Sơ đồ use case tổng quan	17
Hình 2.2	Sơ đồ phân rã use case quản lý khách hàng	18
Hình 2.3	Sơ đồ phân rã use case quản lý khóa học	19
Hình 2.4	Sơ đồ phân rã use case quản lý phiên chăm sóc.....	20
Hình 2.5	Sơ đồ phân rã use case quản lý giao dịch.....	21
Hình 2.6	Sơ đồ phân rã use case quản trị người dùng.....	22
Hình 2.7	Sơ đồ phân rã use case quản trị vai trò.....	23
Hình 2.8	Sơ đồ phân rã use case phân quyền người dùng	24
Hình 2.9	Sơ đồ nghiệp vụ cấp quyền cho người dùng	25
Hình 2.10	Sơ đồ nghiệp vụ tạo phiên chăm sóc khách hàng	26
Hình 3.1	Ví dụ về JWT	38
Hình 3.2	Mã hóa dữ liệu SHA-256.	39
Hình 4.1	Kiến trúc 3 lớp của hệ thống.	41
Hình 4.2	Mô hình client – server	43
Hình 4.3	Cơ chế hoạt động của RBAC.	44
Hình 4.4	Biểu đồ phụ thuộc gói	46
Hình 4.5	Thiết kế gói chi tiết thành phần quản lý khách hàng.....	48
Hình 4.6	Thiết kế gói chi tiết cho phân quyền và quản lý người dùng.	49
Hình 4.7	Thiết kế chi tiết gói cho phần quản lý phiên chăm sóc	50
Hình 4.8	Chi tiết gói thành phần quản lý giao dịch	51

Hình 4.9 Chi tiết gói thành phần quản lý giao dịch(android app).....	52
Hình 4.10 Sơ đồ phân cấp chức năng hệ thống.....	59
Hình 4.11 Ví dụ các mẫu thông báo cố định.....	60
Hình 4.12 Ví dụ về hộp thoại xác nhận.....	61
Hình 4.13 Thiết kế chi tiết một số lớp phân quyền và quản lý người dùng.....	62
Hình 4.14 Thiết kế chi tiết một số lớp cho thành phần quản lý khách hàng.....	63
Hình 4.15 Thiết kế chi tiết một số lớp cho thành phần quản lý khách hàng(android app)	64
Hình 4.16 Sơ đồ thực thể liên kết ER cho CSDL chung.....	65
Hình 4.17 Sơ đồ CSDL tổng quát	65
Hình 4.18 Giao diện quản lý quyền trên web số 1	70
Hình 4.19 Giao diện quản lý quyền trên web số 2	70
Hình 4.20 Giao diện di động quản lý nhân viên, khách hàng, khóa học.....	71
Hình 4.21 Giao diện di động quản lý phiên chăm sóc, giao dịch.	72

Danh mục bảng

Bảng 1 Quy tắc thiết kế thông tin với nội dung một chiều.....	53
Bảng 2 Quy tắc thiết kế thông tin với nội dung tương tác	55
Bảng 3 Quy tắc thiết kế form kiểm tra tính hợp lệ của dữ liệu đầu vào.....	57
Bảng 4 Thiết kế chi tiết bảng User	66
Bảng 5 Thiết kế chi tiết bảng Transaction.....	67
Bảng 6 Thiết kế chi tiết bảng History_care	68
Bảng 7 Danh sách thư viện và công cụ sử dụng.....	69

Chương 1 Giới thiệu đề tài

1.1 Đặt vấn đề

Trung tâm ABC là trung tâm tin học, đào tạo nhân lực CNTT chất lượng và cung cấp nguồn nhân lực cho các công ty lớn nhỏ về sản xuất phần mềm. Lúc đầu mới thành lập, nguồn nhân sự và học viên của công ty có hạn, nên việc lưu trữ và quản lý thông tin của khách hàng, nhân viên, sản phẩm,..... vẫn đáp ứng đầy đủ nhu cầu công việc.

Tuy nhiên, theo thời gian, công ty lớn mạnh và mở rộng quy mô đào tạo. Nên việc quản lý thông tin và dữ liệu qua sổ sách không thể đáp ứng được nhu cầu công việc. Nhân viên phụ trách sổ sách, hóa đơn, chăm sóc khách hàng,... đều rất mất nhiều thời gian trong việc tìm kiếm, chất lọc ghi chép và chăm sóc khách hàng.

Chính vì vậy, mục tiêu của đề tài là “xây dựng phần mềm hỗ trợ quản lý khách hàng cho đội tư vấn tuyển sinh” của trung tâm. Với một hệ thống thống nhất, dễ sử dụng cho nhân viên của trung tâm, góp phần thúc đẩy trung tâm phát triển tốt và mở rộng hơn nữa.

1.2 Mục tiêu và phạm vi đề tài

Trước khi bắt tay vào thực hiện đề tài, em đã thực hiện việc khảo sát thị trường tư vấn tuyển sinh và bán khóa học lập trình cho những người có nhu cầu học lập trình và mong muốn theo nghề nghiệp này ở Việt Nam. Hàng loạt các trung tâm, trường lớp đào tạo lập trình viên ra đời thu hút khá nhiều học viên tham dự mặc dù các khóa học này đều không hề ngắn (4 – 8 tháng để có thể bước chân vào nghề và từ 3 – 5 năm để trở thành chuyên gia lập trình đích thực).

Sự phát triển của ngành công nghệ thông tin kéo theo nhu cầu tuyển dụng việc làm tăng cao. Do đó, người học lập trình không ngừng tăng theo cấp số cộng, kéo theo sự phát triển hàng loạt các trung tâm đào tạo mọc lên như nấm. Tuy nhiên, một trung tâm tốt không chỉ là một trung tâm cung cấp khóa học cho học viên tốt, mà còn khâu quản lý dữ liệu và thông tin của từng thực thể cũng phải tốt không kém. Em đã có khảo sát về một vài hệ thống quản lý khách hàng, bán hàng và có sự đánh giá tổng quan về các vài hệ thống này như sau :

Đầu tiên là phần mềm KiotViet Quản lý của công ty Citigo Software JSC , là phần mềm quản lý khách hàng, thu chi,... mặc dù có đủ các chức năng giúp người dùng quản lý bán hàng, khách hàng tuy nhiên chức năng lưu lại lịch sử chăm sóc khách hàng chưa có , cũng như chức năng tìm kiếm khó khăn vì yêu cầu phải nhập chính xác tên mới ra.

Tiếp theo là phần mềm CenterOnline (CO) là một sản phẩm của Công ty cổ phần Đào tạo Quản lý Trực tuyến OMT (OMT) dành cho các trung tâm ngoại ngữ, ngoại khóa. Ưu điểm : phần mềm quản lý trung tâm ngoại ngữ, ngoại khóa CenterOnline là giải pháp quản lý toàn diện, đa chi nhánh, tính năng đa dạng tích hợp trong một hệ thống, tương tác nhanh chóng đến đúng đối tượng, đáp ứng nghiệp vụ, chức năng của các đối tượng toàn trung tâm: quản lý, nhân viên (kế toán, tuyển sinh, tư vấn, marketing...). Phần mềm quản lý trung tâm gồm 5 chức năng chính (5 Modules): CRM (quản trị quan hệ khách hàng), Đào tạo, Người dùng, Tài chính, Báo cáo. Tuy nhiên, nhược điểm của phần mềm này là chưa phổ thông, việc sử dụng khá phức tạp, người dùng phải học và hiểu rõ về luồng nghiệp vụ của phần mềm, không thích hợp cho một trung tâm vừa và nhỏ do có quá nhiều nghiệp vụ không cần thiết (Ví dụ :quản lý trả lương ,theo dõi con cho phụ huynh,v.v.....)

Nhận thấy những hạn chế này, em quyết định lựa chọn đề tài “xây phần mềm quản lý dữ liệu khách hàng cho đội tư vấn tuyển sinh” bao gồm một bộ quy tắc quản lý thống nhất, khắc phục những hạn chế trên. Phần mềm bao gồm những chức năng chính sau :

- Quản lý truy cập và phân quyền
- Quản lý khách hàng
- Quản lý chăm sóc khách hàng
- Quản lý nhân viên
- Quản lý giao dịch
- Quản lý khóa học

Mục tiêu cuối cùng của phần mềm là nhằm góp phần nào cải thiện những nhược điểm mà các ứng dụng đi trước mắc phải, đồng thời cũng tạo tiền đề cho sự phát triển của các trung tâm đào tạo nguồn nhân lực chất lượng cao theo nhu cầu của xã hội.

1.3 Định hướng giải pháp

Hai yếu tố quan trọng quyết định việc quản lý thành bại của phần mềm, đó là: quản lý giao dịch liên quan trực tiếp đến thu chi và quản lý chăm sóc khách hàng. Chính vì vậy để có thể giải quyết bài toán thì trước hết cần phải nghiên cứu và giải quyết được hai vấn đề này.

Đầu tiên là quản lý chăm sóc khách hàng, dựa vào việc khảo sát các trung tâm ở chương 1.2, em đề xuất giải pháp như sau:

Lưu trữ hồ sơ, giấy tờ, sơ yếu lý lịch khách hàng giúp tra cứu, thống kê nhanh chóng, dễ dàng. Quản lý thông tin cá nhân như họ tên, ngày sinh, giới tính; thông tin liên hệ như số điện thoại (chính, phụ,...), e-mail (chính/phụ), địa chỉ. Về chăm sóc khách hàng (tư vấn) bao gồm: nội dung tư vấn, lưu lịch sử tư vấn, đặt lịch hẹn tư vấn.

Tiếp theo là quản lý giao dịch, em đề xuất giải pháp như sau:

Quản lý giao dịch liên quan trực tiếp đến thu chi, tiền nong của nhiều khách hàng. Quản lý các thông tin như: lý lịch khách hàng, nhân viên giao dịch, khóa học đã đăng ký, tình trạng thanh toán,....

Sau khi giải quyết được những vấn đề trên, hệ thống đã giải quyết cơ bản được những vấn đề trong công tác tư vấn tuyển sinh của trung tâm .

1.4 Bố cục đồ án

Báo cáo đồ án này được chia làm sáu chương chính và nội dung của các chương được tóm lược như sau:

Chương 1 là chương giới thiệu đề tài, tại đây nêu ra các vấn đề tồn tại trong thực tế mà các công ty doanh nghiệp gặp phải, tìm hiểu các giải pháp và công nghệ có sẵn trên thị trường. Trên cơ sở đó tìm hiểu chỉ rõ các yếu tố đã được giải quyết các vấn đề còn tồn đọng, tại đồ án này đã giải quyết và triển khai được những vấn đề gì. Đồng thời so sánh hệ thống xây dựng với các hệ thống có sẵn trên thị trường.

Tại Chương 2 em chỉ rõ các đối tượng người dùng cũng như các chức năng tổng quan cần xây dựng của hệ thống. Phần phân tích nêu rõ những ca sử dụng dành cho từng đối tượng trong hệ thống, miêu tả rõ quy trình nghiệp vụ và chi tiết những chức năng quan trọng, những yêu cầu chức năng và phi chức năng của hệ thống.

Với những khảo sát và đánh giá đã thu thập được ở Chương 2, tại Chương 3 em sẽ đưa ra các cơ sở lý thuyết được áp dụng cho đề tài cũng như những công nghệ sử dụng trong việc xử lý và giải quyết các vấn đề đã nêu ra. Đầu tiên là việc lựa chọn CSDL MySQL bởi các lợi ích và tính ứng dụng của nó. Tiếp đó sử dụng ngôn ngữ Java với công nghệ framework Spring Boot để xây dựng và phát triển ứng dụng.

Ở Chương 4 của báo cáo đồ án này, sẽ trình bày về quá trình phát triển, kiểm thử và triển khai ứng dụng. Trong phần phát triển ứng dụng, nêu rõ kiến trúc sử dụng để xây dựng hệ thống, thiết kế lớp, thiết kế gói, thiết kế cơ sở dữ liệu thông số, thiết kế giao diện hệ thống và trình bày về thư viện và công cụ đã sử dụng cho việc phát triển hệ thống. Trong phần kiểm thử, em trình bày những trường hợp kiểm thử và kỹ thuật kiểm thử đã sử dụng. Cuối cùng là báo cáo về kế hoạch triển khai hệ thống trên môi trường thực tế.

Trong Chương 5 của báo cáo đề án này, em xin trình bày đóng góp và giải pháp nổi bật của mình khi thực hiện phát triển xong ứng dụng bao gồm các đóng góp trong việc tìm hiểu và xây dựng phần mềm quản lý khách hàng cho đội tư vấn tuyển sinh, hỗ trợ các cơ chế quản lý giúp đỡ người dùng trong quá trình sử dụng.

Cuối cùng, Chương 6 là kết luận về sản phẩm của em với những thiếu sót trong việc xây dựng hệ thống, những kinh nghiệm đã học hỏi được trong quá trình xây dựng hệ thống và hướng phát triển sau này để hoàn thiện tất cả chức năng mong muốn của hệ thống.

Chương 2 Khảo sát và phân tích yêu cầu

Ở chương này sẽ trình bày về các nhược điểm thực tế của trung tâm từ đó tham khảo và xây dựng phần mềm quản lý khách hàng hỗ trợ đội tư vấn tuyển sinh. Tiếp đó, trình bày tổng quan về các chức năng của hệ thống.

2.1 Khảo sát hiện trạng

Trung tâm ABC là trung tâm mới thành lập, chuyên đào tạo lập trình viên theo nhu cầu. Trung tâm kinh doanh chủ yếu các khóa học bao gồm : lập trình mobile, flutter, lập trình java web,.....Là một doanh nghiệp mới thành lập nên hoạt động chủ yếu vẫn mang tính thủ công, dựa vào kinh nghiệm của hệ thống nhân viên. Do đặc điểm lượng khách hàng, sản phẩm, giao dịch ngày càng lớn nên công tác quản lý mất rất nhiều thời gian, công sức mà độ chính xác lại không cao, điều này có khi gây thiệt hại cho trung tâm. Ngày nay, trung tâm đang từng bước hiện đại hóa hệ thống các máy tính và đưa hệ thống phần mềm vào sử dụng để thay thế công tác quản lý thủ công.

Hoạt động quản lý khách hàng thủ công nên bộc lộ những hạn chế :

- Tra cứu thông tin về khách hàng, sản phẩm, giao dịch , phiên chăm sóc từng khách hàng,... nhiều khi rất mất thời gian, thiếu chính xác.
- Việc lưu trữ các thông tin về khách hàng, sản phẩm, giao dịch , phiên chăm sóc từng khách hàng ... phải sử dụng nhiều giấy tờ, sổ sách nên rất cồng kềnh cho lưu trữ.
- Không đáp ứng được nhu cầu thông tin phục vụ hàng ngày, hàng giờ và mở rộng trung tâm trong tương lai.

Trước tình hình đó, vấn đề đặt ra cần phải có một hệ thống mới đáp ứng được các yêu cầu cơ bản sau :

Lưu trữ hồ sơ, giấy tờ, sơ yếu lý lịch khách hàng giúp tra cứu, thống kê nhanh chóng, dễ dàng. Quản lý thông tin cá nhân như họ tên, ngày sinh, giới tính; thông tin liên hệ như số điện thoại (chính, phụ,...), e-mail (chính/phụ), địa chỉ. Về chăm sóc khách hàng(tư vấn) bao gồm: nội dung tư vấn, lưu lịch sử tư vấn, đặt lịch hẹn tư vấn.

Quản lý giao dịch liên quan trực tiếp đến thu chi, tiền nong của nhiều khách hàng. Quản lý các thông tin như : thông tin khách hàng, thông tin nhân viên giao dịch, khóa học đã đăng ký, tình trạng thanh toán,....

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quan

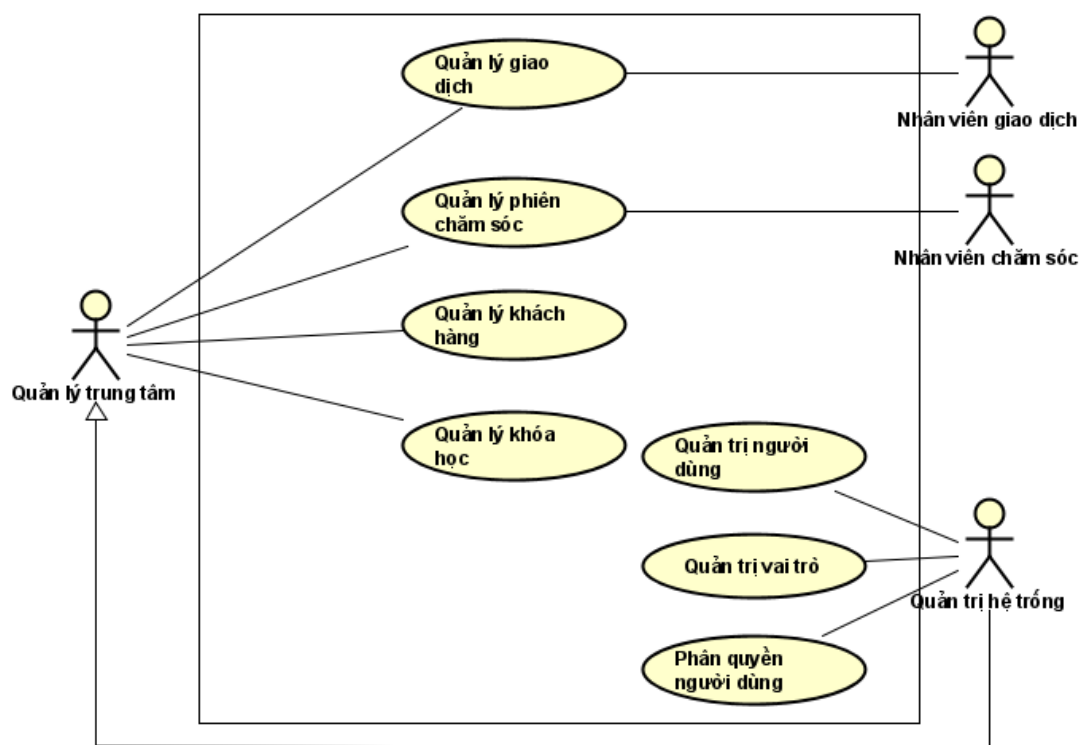
Khi chưa đăng nhập, người dùng cần yêu cầu quản trị hệ thống cấp tài khoản và vai trò để bắt đầu sử dụng các chức năng của phần mềm. Khi người dùng đăng nhập thành công, tùy vào vai trò đã được thêm thì phần mềm sẽ hiển thị giao diện chức năng tương ứng.

Sau khi đăng nhập , nếu người dùng có vai trò là nhân viên chăm sóc khách hàng thì có thể thực hiện quản lý khách hàng, quản lý nhiều phiên chăm sóc khách hàng của từng khách hàng.

Nếu người dùng có vai trò là nhân viên giao dịch thì có thể thực hiện quản lý khách hàng, quản lý các giao dịch đối với từng khách hàng.

Nếu người dùng có vai trò là quản lý trung tâm thì ngoài việc thừa hưởng các chức năng của nhân viên chăm sóc , nhân viên giao dịch thì còn có thể thực hiện quản lý khách hàng, quản lý khóa học

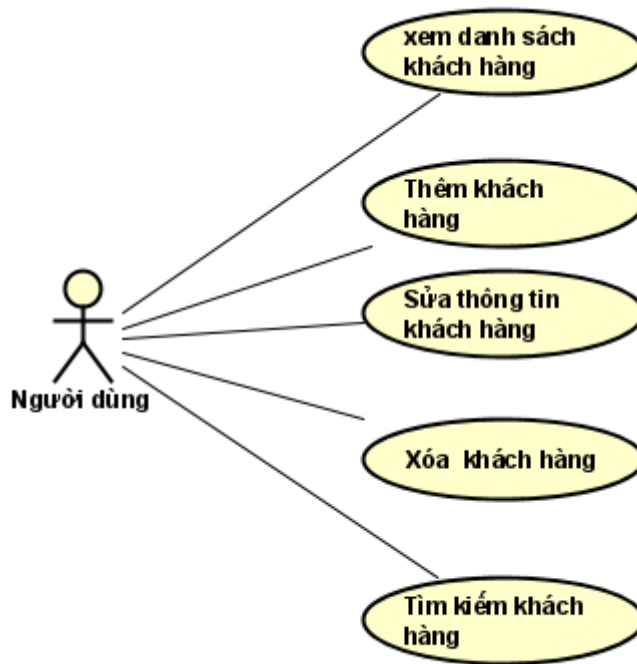
Vai trò quản trị hệ thống có thể thực hiện các chức năng của quản lý trung tâm, ngoài ra còn có thêm những chức năng : quản trị người dùng, quản trị vai trò người dùng, phân quyền người dùng hệ thống.



Hình 1 Sơ đồ use case tổng quan

2.2.2 Các biểu đồ use case phân rã

2.2.2.1 Phân rã use case Quản lý khách hàng



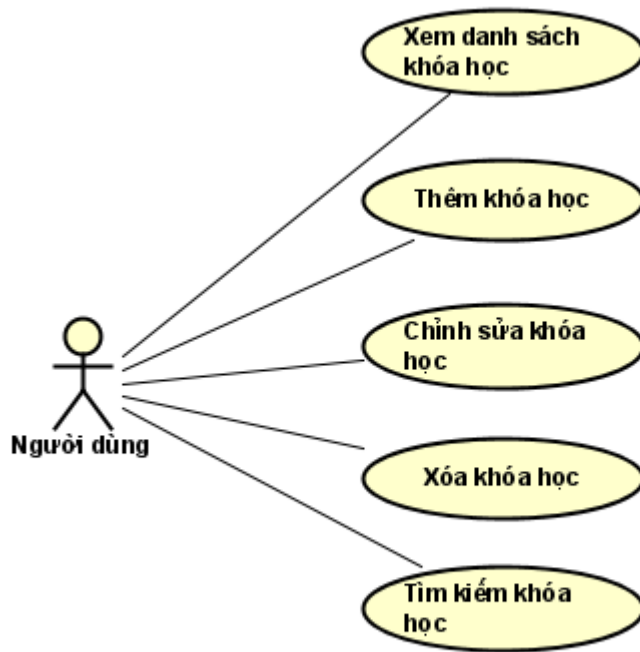
Hình 2 Sơ đồ phân rã use case quản lý khách hàng

Người dùng ở đây bao gồm:

- Quản lý trung tâm
- Quản trị hệ thống

Trong ca sử dụng này, người dùng có thể xem danh sách khách hàng, tìm kiếm thông tin khách hàng: theo tên, email, số điện thoại,.... Người dùng có thể tạo mới thông tin khách hàng, chỉnh sửa thông tin khách hàng và xóa thông tin khách hàng.

2.2.2.2 Phân rã use case Quản lý khóa học



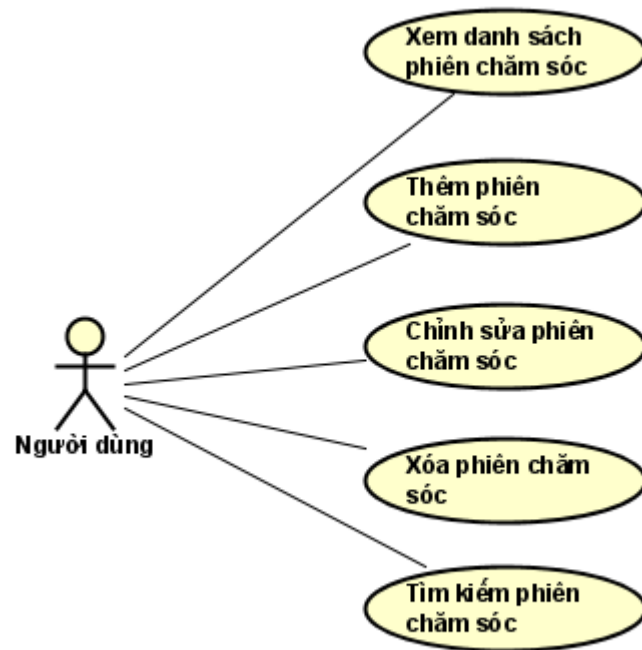
Hình 3 Sơ đồ phân rã use case quản lý khóa học

Người dùng ở đây bao gồm:

- Quản lý trung tâm
- Quản trị hệ thống

Trong ca sử dụng này, người dùng có thể xem danh sách khóa học, tìm kiếm thông tin khóa học: theo tên, giá, mô tả, Người dùng có thể tạo mới khóa học, chỉnh sửa khóa học và xóa khóa học.

2.2.2.3 Phân rã use case quản lý phiên chăm sóc khách hàng



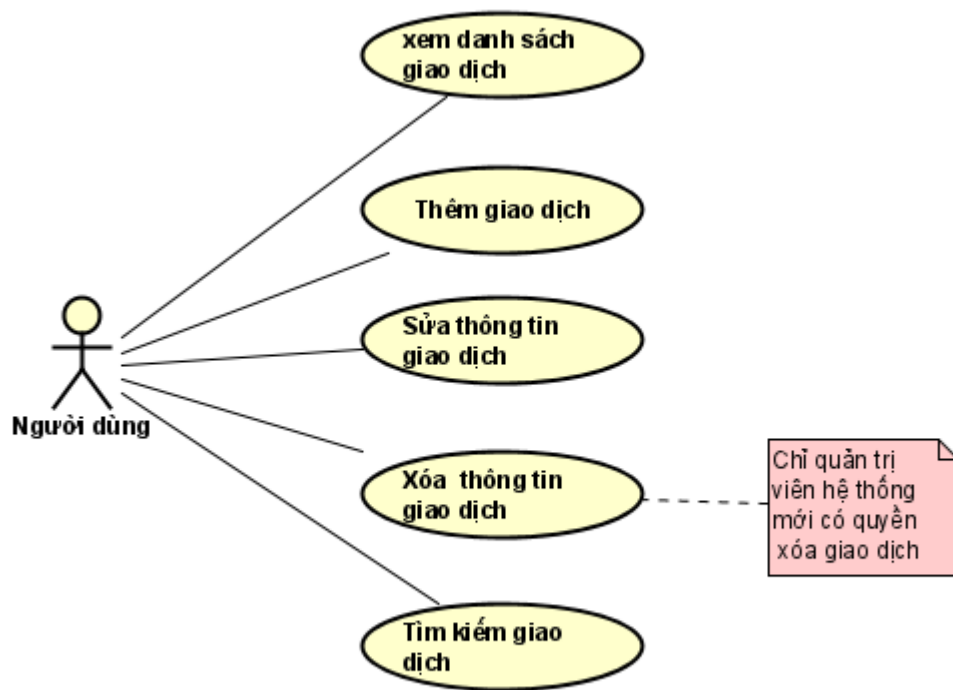
Hình 4 Sơ đồ phân rã use case quản lý phiên chăm sóc

Người dùng ở đây bao gồm:

- Quản lý trung tâm
- Quản trị hệ thống
- Nhân viên chăm sóc khách hàng

Trong ca sử dụng này, người dùng có thể xem danh sách phiên chăm sóc, tìm kiếm thông tin phiên chăm sóc: theo tên, khách hàng, ngày chăm sóc, Người dùng có thể tạo mới phiên chăm sóc, chỉnh sửa phiên chăm sóc và xóa phiên chăm sóc.

2.2.2.4 Phân rã use case quản lý giao dịch



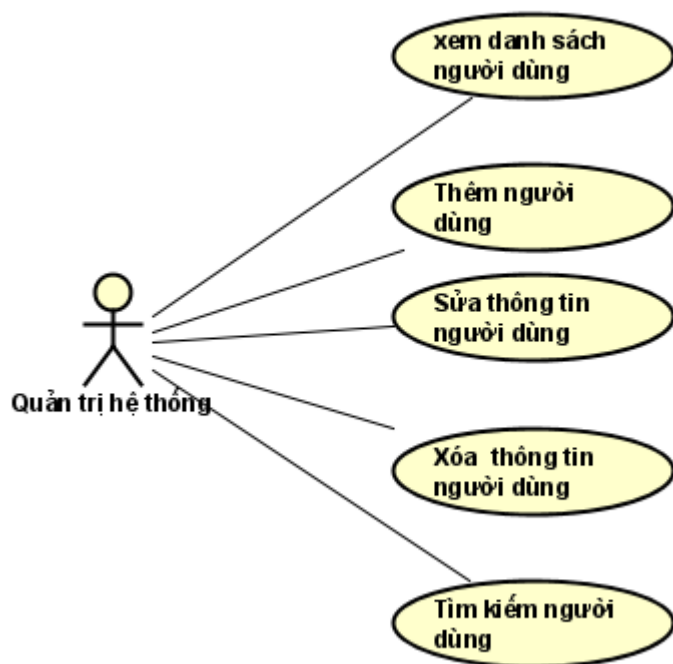
Hình 5 Sơ đồ phân rã use case quản lý giao dịch

Người dùng ở đây bao gồm:

- Quản lý trung tâm
- Quản trị hệ thống
- Nhân viên giao dịch

Trong ca sử dụng này, người dùng có thể xem danh sách giao dịch, tìm kiếm thông tin giao dịch: theo tên Khách hàng, tên Nhân viên giao dịch, ngày giao dịch,.... Người dùng có thể tạo mới giao dịch, chỉnh sửa giao dịch và xóa giao dịch.

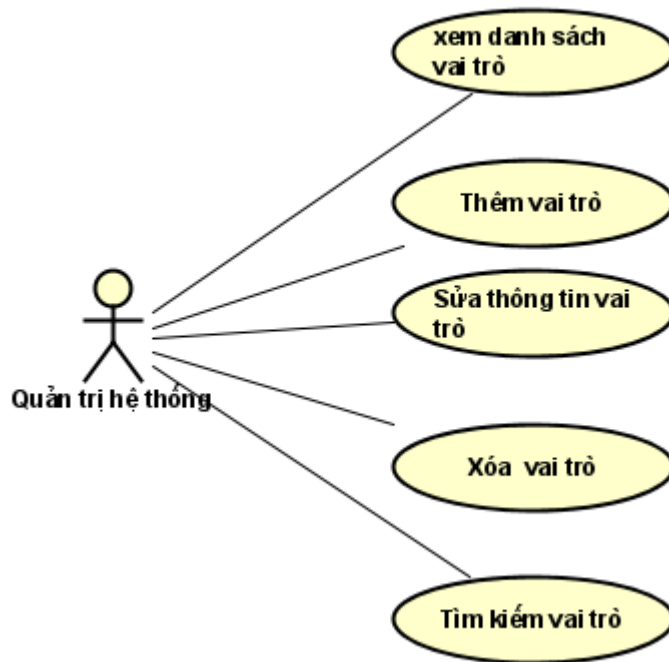
2.2.2.5 Quản trị người dùng



Hình 6 Sơ đồ phân rã use case quản trị người dùng

Trong ca sử dụng này, Quản trị hệ thống có thể xem danh sách người dùng hệ thống, tìm kiếm thông tin người dùng: theo tên, email, số điện thoại, Quản trị hệ thống có thể tạo mới người dùng, chỉnh sửa người dùng và xóa người dùng.

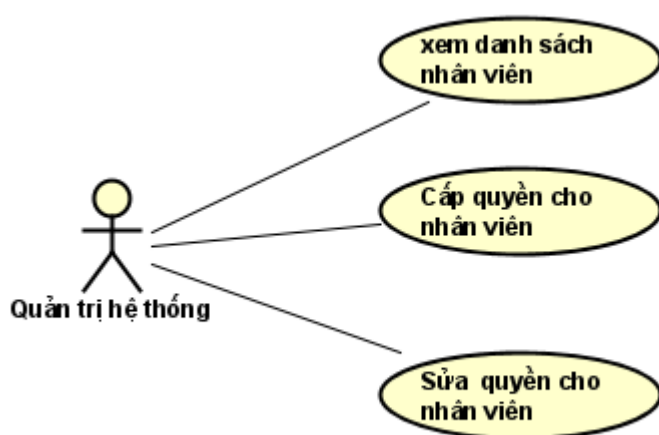
2.2.2.6 Phân rã use case quản trị vai trò



Hình 7 Sơ đồ phân rã use case quản trị vai trò

Vai trò ở đây hiểu là quyền được phép truy cập các trang nào (ví dụ : vào trang sản phẩm, khách hàng,...), được phép thực thi những hành động nào (ví dụ : thêm, sửa, xóa bản ghi), một vai trò có thể gồm một hoặc nhiều trang, một hoặc nhiều hành động. Trong ca sử dụng này, Quản trị hệ thống có thể xem danh sách vai trò, tìm kiếm thông tin vai trò theo tên. Quản trị hệ thống có thể tạo mới vai trò, chỉnh sửa vai trò và xóa vai trò.

2.2.2.7 Phân rã use case phân quyền người dùng

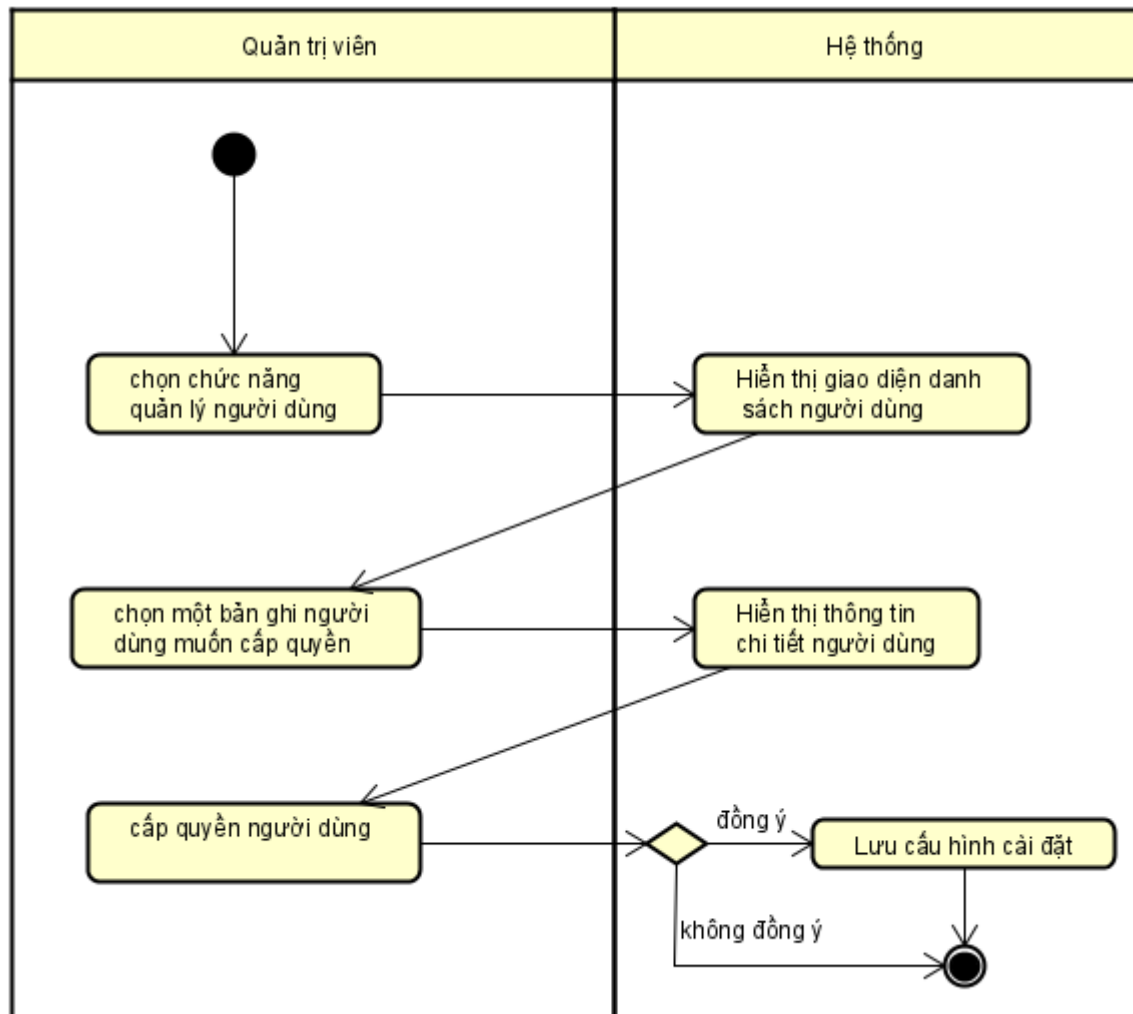


Hình 8 Sơ đồ phân rã use case phân quyền người dùng

Trong ca sử dụng này, Quản trị hệ thống có thể cấp quyền cho người dùng theo các vai trò đã tạo trong use case quản lý vai trò. Tùy vào quyền mà người dùng được cấp, các tài khoản người dùng khi đăng nhập thành công sẽ được vào giao diện tương ứng với quyền hạn của mình, được thực thi các hành động thêm sửa xóa theo quyền hạn của mình.

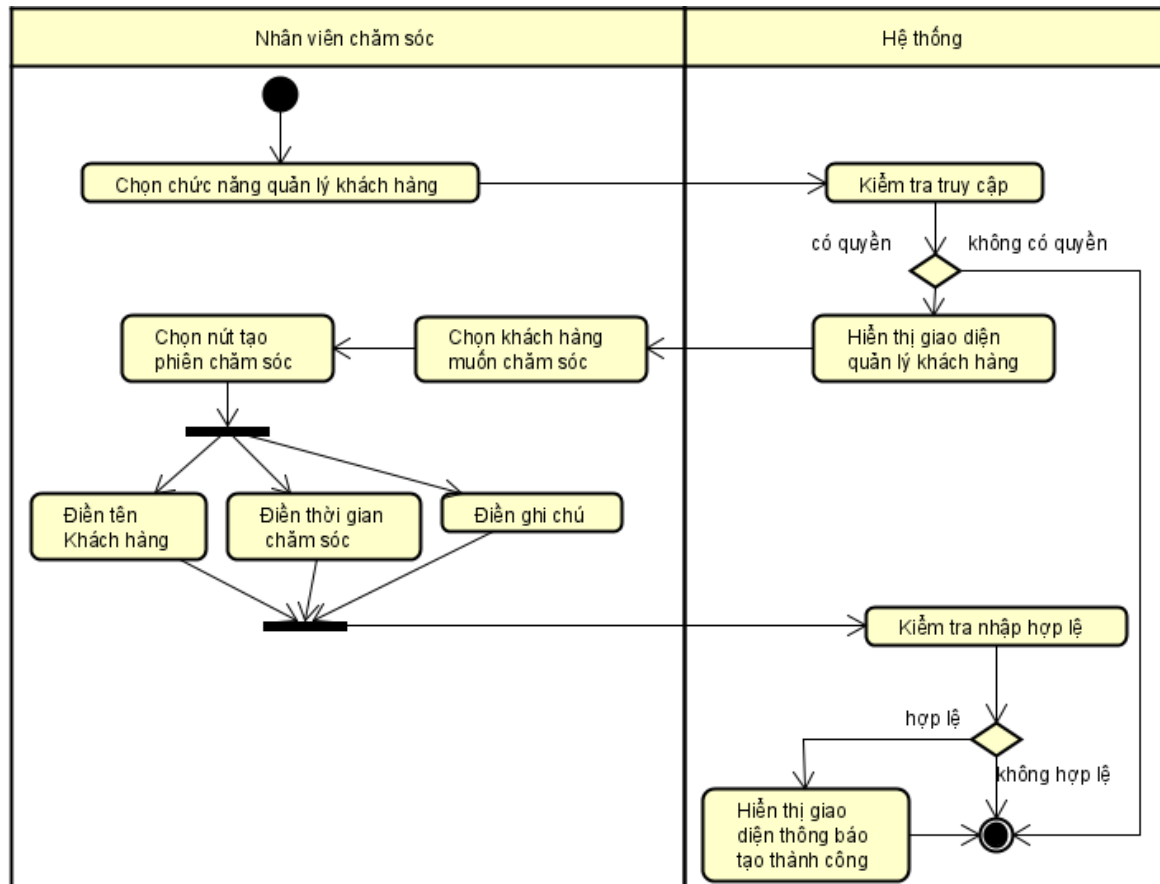
2.2.3 Quy trình nghiệp vụ

2.2.3.1 Nghiệp vụ cấp quyền cho người dùng



Hình 9 Sơ đồ nghiệp vụ cấp quyền cho người dùng

2.2.3.2 Nghiệp vụ tạo phiên chăm sóc khách hàng



Hình 10 Sơ đồ nghiệp vụ tạo phiên chăm sóc khách hàng

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case thêm khách hàng

Mã Use case	UC001	Tên Use case	Thêm khách hàng
Tác nhân	Quản lý trung tâm, Quản trị hệ thống		
Tiền điều kiện	Đăng nhập hệ thống với vai trò quản lý trung tâm hoặc quản trị hệ thống		

Luồng sự kiện chính (Thành công)	ST T	Thực hiện bởi	Hành động
	1.	Người dùng	Chọn thêm khách hàng
	2.	Hệ thống	Hiển thị hộp thoại khách hàng
	3.	Người dùng	Nhập thông tin khách hàng (*)
	4.	Hệ thống	Kiểm tra thông tin nhập vào
	5.	Hệ thống	Thông báo thêm mới thành công (**)
	6.	Hệ thống	Chuyển về giao diện xem danh sách khách hàng
Luồng sự kiện thay thế	ST T	Thực hiện bởi	Hành động
	5a.	Hệ thống	Thông báo lỗi
	6a.	Hệ thống	Hiển thị trạng thái lỗi trên ô nhập liệu
Hậu điều kiện	Không		

Dữ liệu đầu vào (*)

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Tên khách hàng	Nhập vào tên khách hàng	Có		Phạm Thành Nam,A Nam,...
2	Số điện thoại	Nhập số điện thoại khách hàng	Có	Dữ liệu số (đầu số 03 05 09), 10 số.	0348984167

3	Email	Nhập số email khách hàng	có	Dữ liệu chuỗi , có @gmail.com	Nampt10@gmail.com
4	Địa chỉ	Nhập địa chỉ khách hàng	không	Dữ liệu chuỗi	Hà Nội

Dữ liệu đầu ra (**)

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
1.	Thông báo kết quả	Thể hiện kết quả khi người dùng nhấn vào lưu thông tin	Chuỗi ký tự	Thêm khách hàng thành công.

2.3.2 Đặc tả use case thêm vai trò

Mã Use case	UC002	Tên Use case	Thêm vai trò
Tác nhân	Quản trị hệ thống		
Tiền điều kiện	Đăng nhập hệ thống với vai trò quản trị hệ thống		
Luồng sự kiện chính	ST T	Thực hiện bởi	Hành động
	1.	Quản trị hệ thống	Chọn thêm vai trò
	2.	Hệ thống	Hiển thị form điền thông tin
	3.	Quản trị hệ	Nhập thông tin vai trò (*)

(Thành công)		thống	
	4.	Hệ thống	Kiểm tra thông tin nhập vào
	5.	Hệ thống	Thông báo thêm mới thành công (**)
	6.	Hệ thống	Chuyển về giao diện xem danh sách vai trò
Luồng sự kiện thay thế	ST T	Thực hiện bởi	Hành động
	5a.	Hệ thống	Thông báo lỗi
	6a.	Hệ thống	Hiển thị trạng thái lỗi trên ô nhập liệu
Hậu điều kiện	Không		

Dữ liệu đầu vào (*)

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Tên vai trò	Nhập vào tên vai trò	Có		Quản lý trung tâm 01
2	Mô tả vai trò	Nhập vào mô tả vai trò	không	Dữ liệu chuỗi	Đây là quyền quản lý trung tâm

Dữ liệu đầu ra (**)

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
-----	----------------	-------	--------------------	-------

1.	Thông báo kết quả	Thể hiện kết quả khi người dùng nhấn vào lưu thông tin	Chuỗi ký tự	Thêm vai trò thành công.
----	-------------------	--	-------------	--------------------------

2.3.3 Đặc tả use case thêm giao dịch

Mã Use case	UC003	Tên Use case	Thêm giao dịch
Tác nhân	Nhân viên giao dịch		
Tiền điều kiện	Đăng nhập hệ thống với vai trò nhân viên giao dịch		
Luồng sự kiện chính (Thành công)	ST T	Thực hiện bởi	Hành động
	1.	nhân viên giao dịch	Chọn thêm giao dịch
	2.	Hệ thống	Hiển thị form điền thông tin
	3.	nhân viên giao dịch	Nhập thông tin giao dịch (*)
	4.	Hệ thống	Kiểm tra thông tin nhập vào
	5.	Hệ thống	Thông báo thêm mới thành công (**)
	6.	Hệ thống	Chuyển về giao diện xem danh sách giao dịch
Luồng sự kiện thay thế	ST T	Thực hiện bởi	Hành động
	5a.	Hệ thống	Thông báo lỗi
	6a.	Hệ thống	Hiển thị trạng thái lỗi trên ô nhập liệu
Hậu điều kiện	Không		

Dữ liệu đầu vào (*)

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Tên khách hàng	Chọn 1 khách hàng trong danh sách có sẵn	Có		Phạm Văn Minh
2	Thời gian giao dịch	Nhập vào thời gian giao dịch	có	Dữ liệu thời gian	01/12/2019 05:00 PM
3	Tên sản phẩm	Chọn 1 sản phẩm trong danh sách có sẵn	có		Java Web
4	Trạng thái	Chọn 1 trạng thái trong danh sách có sẵn	Có		Đã thanh toán

Dữ liệu đầu ra (**)

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
1.	Thông báo kết quả	Thể hiện kết quả khi người dùng nhấn vào lưu thông tin	Chuỗi ký tự	Thêm vai trò thành công.

2.3.4 Đặc tả use case cấp quyền người dùng

Mã Use case	UC004	Tên Use case	Cấp quyền người dùng
Tác nhân	Quản trị hệ thống		
Tiền điều kiện	Đăng nhập hệ thống với vai trò quản trị hệ thống		
Luồng sự kiện chính (Thành công)	ST T	Thực hiện bởi	Hành động
	1.	Quản trị hệ thống	Truy cập quản trị người dùng
	2.	Hệ thống	Hiển thị danh sách người dùng
	3.	Quản trị hệ thống	Chọn chỉnh sửa thông tin người dùng
	4.	Hệ thống	Hiển thị danh sách vai trò người dùng
	5.	Quản trị hệ thống	Chọn thêm vai trò cho người dùng
	6.	Hệ thống	Cập nhật lại danh sách vai trò người dùng
Luồng sự kiện thay thế	ST T	Thực hiện bởi	Hành động
	5a.	Quản trị hệ thống	Chọn chỉnh sửa vai trò muốn sửa cho người dùng
Hậu điều kiện	Không		

2.4 Yêu cầu phi chức năng

2.4.1 Yêu cầu về chức năng kỹ thuật

Phần mềm sử dụng hệ quản trị dữ liệu đáp ứng yêu cầu:

- Hiệu năng: nhiều client có thể truy cập cùng lúc.
- Bảo mật: hỗ trợ kiểm soát truy cập người dùng.

2.4.2 Đáp ứng yêu cầu về bảo mật

Đảm bảo an toàn được đặt ra trong quá trình lưu trữ, truyền tải dữ liệu giữa các thành phần trong môi trường internet.

Các tiêu chuẩn về bảo mật và toàn vẹn dữ liệu cần đáp ứng yêu cầu:

- ✓ Mã hóa các dữ liệu có yêu cầu bảo mật cao
- ✓ Sử dụng hỗ trợ an toàn bảo mật của các ứng dụng nền

2.4.3 Yêu cầu về giao diện

Giao diện sử dụng đảm bảo các tính chất như thân thiện với người dùng, tính dễ học và sử dụng. Các sự kiện khi tương tác với giao diện phải có tính phản hồi, khi phải chờ đợi xử lý tiến trình thì cần có thông báo cho người dùng.

Chương 3 Công nghệ sử dụng

3.1.1 Hệ quản trị CSDL MySQL

3.1.1.1 MySQL là gì?

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.

Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS...

Bên cạnh đó MySQL là một trong những ví dụ rất cơ bản về hệ quản trị CSDL quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL). MySQL được sử dụng cho việc hỗ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl.

3.1.1.2 Ưu điểm của hệ quản trị CSDL MySQL

Hệ quản trị CSDL MySQL có rất nhiều ưu điểm khác nhau, đầu tiên phải nói đến tốc độ xử lý. MySQL rất nhanh, những nhà phát triển cho rằng MySQL là cơ sở dữ liệu nhanh nhất mà người dùng có thể có. Tiếp đến là tính dễ sử dụng, MySQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn.

Tiếp theo là về giá thành, MySQL là miễn phí cho hầu hết các việc sử dụng trong một tổ chức. Hơn nữa, MySQL còn hỗ trợ ngôn ngữ truy vấn: MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Bạn cũng có thể truy cập MySQL bằng cách sử dụng các ứng dụng mà hỗ trợ ODBC (Open Database Connectivity - một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft).

Về Năng lực, MySQL cho phép nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Bạn có thể truy cập MySQL tương tác với sử dụng một vài giao diện để bạn có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web...

Về khả năng kết nối và bảo mật, MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó bạn có thể chia sẻ dữ liệu của bạn với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người mà không nên nhìn thấy dữ liệu của bạn thì không thể nhìn được.

Về tính linh động, MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với mọi phần cứng từ các máy PC ở nhà cho đến các máy server.

Về sự phân phối rộng, MySQL rất dễ dàng đạt được, chỉ cần sử dụng trình duyệt web của bạn. Nếu bạn không hiểu làm thế nào mà nó làm việc hay tò mò về thuật toán, bạn có thể lấy mã nguồn và tìm tòi nó. Nếu bạn không thích một vài cái, bạn có thể thay đổi nó.

Cuối cùng về khả năng hỗ trợ, người dùng có thể tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ trả lời các câu hỏi trên mailing list thường chỉ trong vài phút. Khi lỗi được phát hiện, các nhà phát triển sẽ đưa ra cách khắc phục trong vài ngày, thậm chí có khi trong vài giờ và cách khắc phục đó sẽ ngay lập tức có sẵn trên Internet.

Chính vì những ưu điểm vượt trội đó mà MySQL được rất nhiều người dùng ưu chuộng và sử dụng.

3.1.2 Java Spring Boot

3.1.2.1 Java Spring Boot là gì?

Java Spring Boot là chương trình chạy trên một Web hoặc ứng dụng máy chủ (Application Server) và hành động như một lớp trung gian giữa một yêu cầu đến từ một trình duyệt Web hoặc HTTP khách (Client) khác và cơ sở dữ liệu hoặc các ứng dụng trên máy chủ HTTP (HTTP Server). Java Spring Boot thường được sử dụng để:

- Nhận client request và lấy thông tin từ request: Đọc dữ liệu rõ ràng do khách hàng (trình duyệt) gửi
- Xử lý nghiệp vụ và phát sinh chuyên môn (bằng cách truy cập database): Quá trình xử lý dữ liệu và tạo ra các kết quả này có thể yêu cầu nói chuyện với một cơ sở dữ liệu, thực hiện một cuộc gọi RMI hoặc CORBA, gọi một dịch vụ Web, hoặc tính trực tiếp phản hồi.
- Tạo và gửi request đến client:
Không chỉ gửi dữ liệu rõ ràng (tức là tài liệu) tới khách hàng (trình duyệt) dưới nhiều định dạng như văn bản (HTML hoặc XML), nhị phân (hình ảnh GIF), Excel, mà còn gửi phản hồi HTTP ẩn cho khách hàng (trình duyệt). Điều này bao gồm nói với trình duyệt hoặc các trình khách khác loại tài liệu đang được trả về (ví dụ, HTML), thiết lập cookie và các tham số bộ nhớ đệm, và các tác vụ khác.

3.1.2.2 Ưu Điểm Của Spring Boot

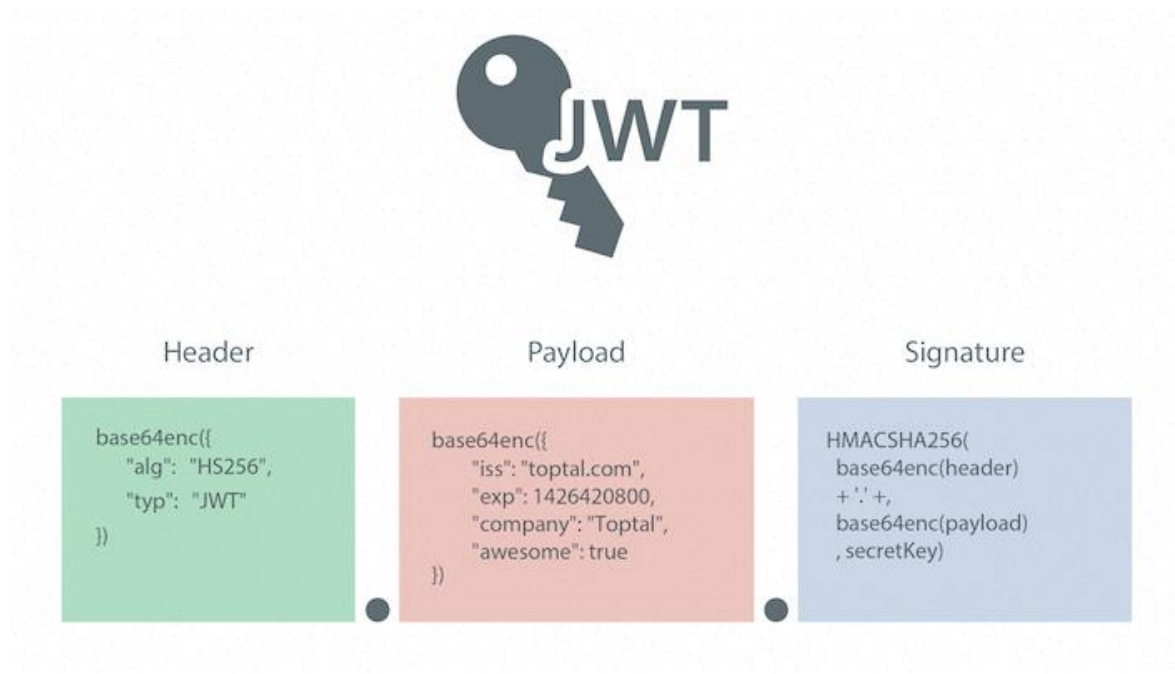
Spring Boot được sử dụng để thay thế cho những công nghệ Web động. Việc sử dụng Spring Boot mang lại những lợi thế:

- Dễ di chuyển. Spring Boot được viết bằng Java nên nó có tính di động cao, thực hiện được trên nhiều hệ điều hành, trên các Web Server khác nhau. Khái niệm “Viết một lần, chạy ở mọi nơi” cũng rất đúng với Spring Boot.
- Mạnh mẽ. Spring Boot hỗ trợ rất hiệu quả cho việc sử dụng các giao diện lỗi API như lập trình mạng, xử lý đa luồng, xử lý ảnh, nén dữ liệu, kết nối các CSDL, bảo mật, xử lý phân tán, v.v. Nó cũng thích hợp để trao đổi tin, truyền thông giữa Client và Server một cách bình thường.
- Hiệu quả. Spring Boot có tính hiệu quả cao. Một khi được tải về, nó sẽ được lưu lại trong bộ nhớ của máy chủ. Spring Boot duy trì các trạng thái của nó, do vậy những tài nguyên ngoại như việc kết nối với CSDL cũng sẽ được lưu giữ lại.
- An toàn. Bởi vì Spring Boot được viết bằng Java nên nó kế thừa được tính an toàn của Java. Cơ chế tự động dọn rác và việc không sử dụng con trỏ của Java giúp cho Spring Boot thoát khỏi nhiều công việc quản lý bộ nhớ. Đồng thời nó xử lý các lỗi rất an toàn theo cơ chế xử lý ngoại lệ của Java.
- Tính tích hợp. Các ứng dụng Spring Boot được tích hợp với các Server. Chúng cộng tác với các Server tốt hơn các chương trình CGI.

Chính vì vậy Java Spring Boot thường được dùng cho những dự án lớn, và chuyên nghiệp bởi những tính năng mà nó đem lại.

3.1.3 JSON Web Token (JWT)

JSON Web Token (JWT) là 1 tiêu chuẩn mở (RFC 7519) định nghĩa cách thức truyền tin an toàn giữa các thành viên bằng 1 đối tượng JSON. Thông tin này có thể được xác thực và đánh dấu tin cậy nhờ vào "chữ ký" của nó. Phần chữ ký của JWT sẽ được mã hóa lại bằng HMAC hoặc RSA.



Hình 3.1 Ví dụ về JWT

Đặc điểm nổi bật của JWT:

- Kích thước nhỏ: JWT có thể được truyền thông qua URL, hoặc qua giao thức POST, hay nhét vào bên trong phần HTTP Header. Kích thước nhỏ hơn ứng với công việc truyền tải sẽ nhanh hơn. Dưới đây là cách thức truyền token vào trong HTTP Header sử dụng Bearer Schema.
- Khép kín: Phần Payload (hiểu nôm na là khối hàng) chứa toàn bộ những thông tin mà chúng ta cần tới, ví dụ như thông tin của người dùng (thay vì phải truy vấn cơ sở dữ liệu nhiều lần)

JWT sẽ được sử dụng trong các trường hợp sau:

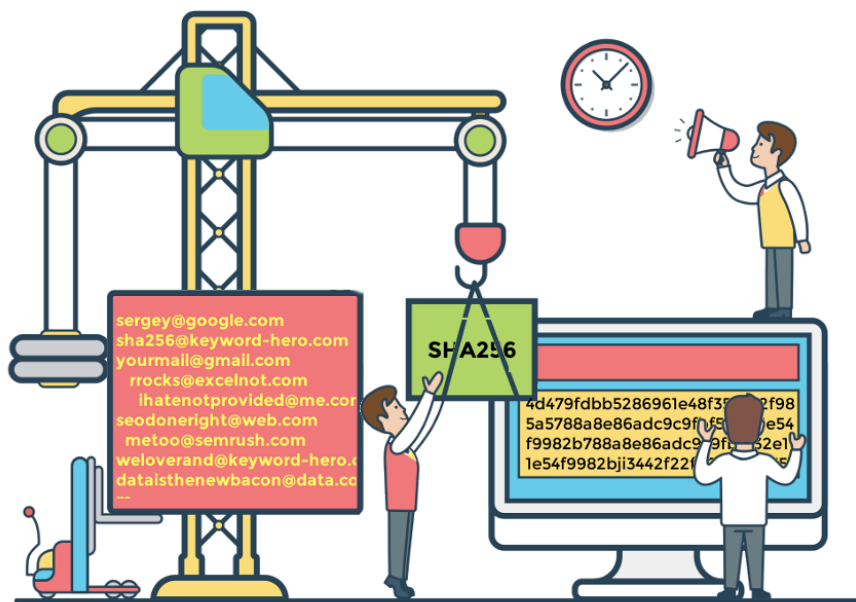
- Authentication: Đây là trường hợp phổ biến nhất thường sử dụng JWT. Khi người dùng đã đăng nhập vào hệ thống thì những request tiếp theo từ phía người dùng sẽ chứa thêm mã JWT. Điều này cho phép người dùng được cấp

quyền truy cập vào các url, service, và resource mà mã Token đó cho phép. Phương pháp này không bị ảnh hưởng bởi Cross-Origin Resource Sharing (CORS) do nó không sử dụng cookie.

- Trao đổi thông tin: JSON Web Token là 1 cách thức khá hay để truyền thông tin an toàn giữa các thành viên với nhau, nhờ vào phần signature của nó. Phía người nhận có thể biết được người gửi là ai thông qua phần signature. Và chữ ký được tạo ra bằng việc kết hợp cả phần header, payload lại nên thông qua đó ta có thể xác nhận được chữ ký có bị giả mạo hay không.

3.1.4 Mã hóa SHA

SHA (Secure Hash Algorithm) là một thuật toán băm dùng để chuyển một đoạn dữ liệu nhất định thành một đoạn dữ liệu có chiều dài không đổi với xác suất khác biệt cao. Trong hệ thống đã sử dụng chuẩn mã hóa dữ liệu SHA256.



Hình 3.2 Mã hóa dữ liệu SHA-256.

Đặc điểm:

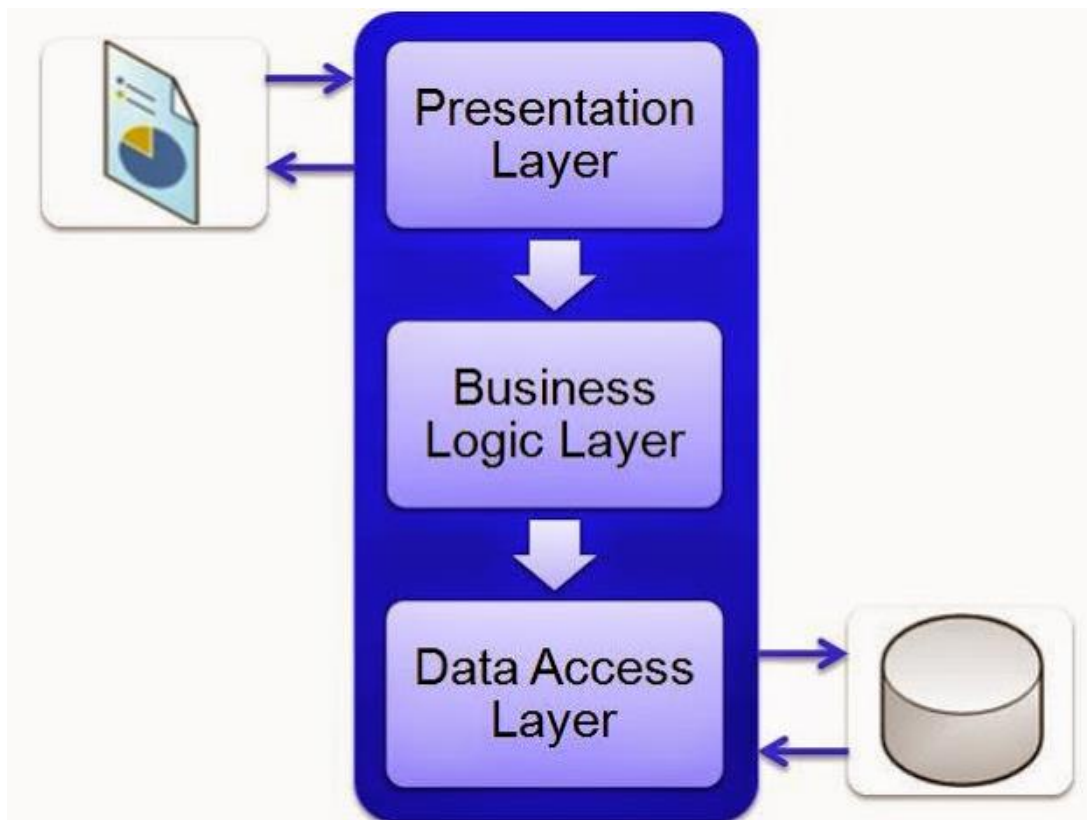
- SHA-256 tạo ra một mã băm có cố định 256-bit (32-byte) gần như duy nhất. Một chuỗi băm được tạo ra không thể được tính toán trở lại. Điều này làm cho nó phù hợp để xác nhận mật khẩu, thách thức xác thực hàm băm, chống giả mạo, chữ ký số.
- SHA-256 là một trong những hàm băm kế tiếp đời sau SHA-1 và là một trong những hàm băm mạnh nhất hiện tại.
- SHA-256 hỗ trợ sức mạnh xử lý thô.

Chương 4 Phát triển và triển khai ứng dụng

4.1 Thiết kế kiến trúc

4.1.1 Mô hình kiến trúc ba lớp

Phần mềm được xây dựng và được triển khai trên mạng Internet, dựa trên kiến trúc 3 lớp. Mô hình kiến trúc 3 lớp như sau:



Hình 4.1 Kiến trúc 3 lớp của hệ thống.

Trong mô hình trên:

a) Tầng trình diễn (Presentation Tier):

Là tầng tương ứng cho việc giao tiếp giữa người sử dụng với tầng bên dưới (Business Logic Tier) thông qua webservice hoặc thông qua các đối tượng tương ứng.

b) Tầng logic (Business Logic Tier):

Đây là tầng quan trọng nhất của kiến trúc 3 lớp. Mục đích chính của tầng này là để xử lý các nghiệp vụ logic, tăng hiệu năng của hệ thống, tăng cường bảo mật, tăng tính mềm dẻo hơn và tính sử dụng lại.

Tầng này chia làm:

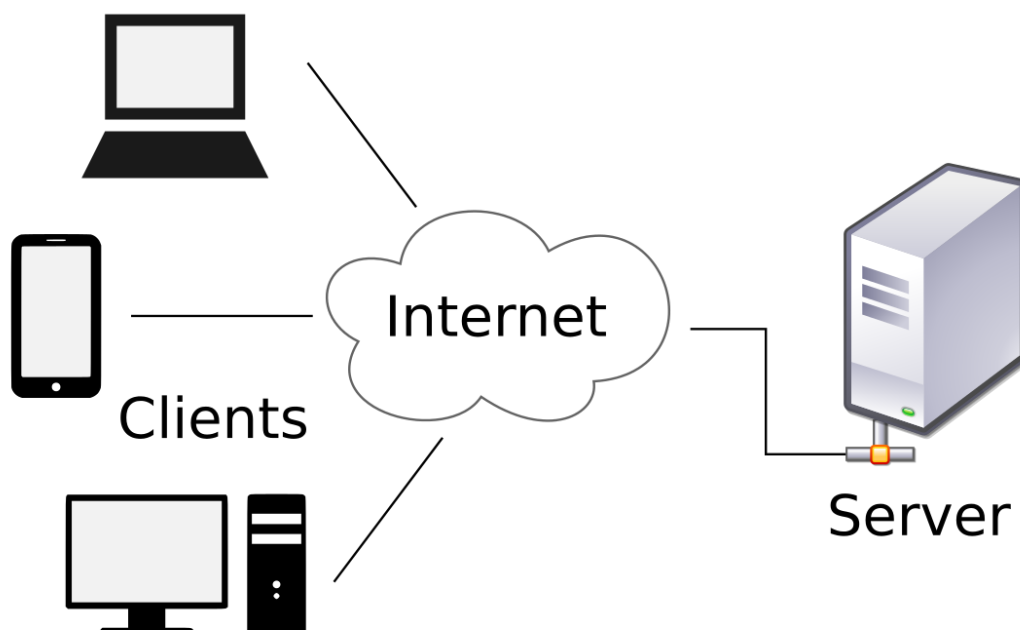
- Service Interface: Đây là cài đặt các dịch vụ phục vụ cho việc cài đặt và triển khai các component logic của phân hệ
- Service Component: Đây là nơi xử lý các nghiệp vụ về quản lý khách hàng, hóa đơn, chăm sóc ,v.v.

c) Tầng cơ sở dữ liệu (Data Tier):

Tầng này phục vụ cho việc lấy, lưu trữ và cập nhật thông tin. Thông thường ở tầng này là một hệ quản trị cơ sở dữ liệu: MS SQL Server, Oracle, MySQL ... hoặc có thể là các file lưu trữ bên ngoài (CSV, Excel...)

4.1.1.2 Mô hình client – server

Mô hình client-server là một mô hình nổi tiếng trong [mạng máy tính](#), được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có. Ý tưởng của mô hình này là máy con (đóng vai trò là máy khách) gửi một yêu cầu (request) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.



Hình 4.2 Mô hình client – server

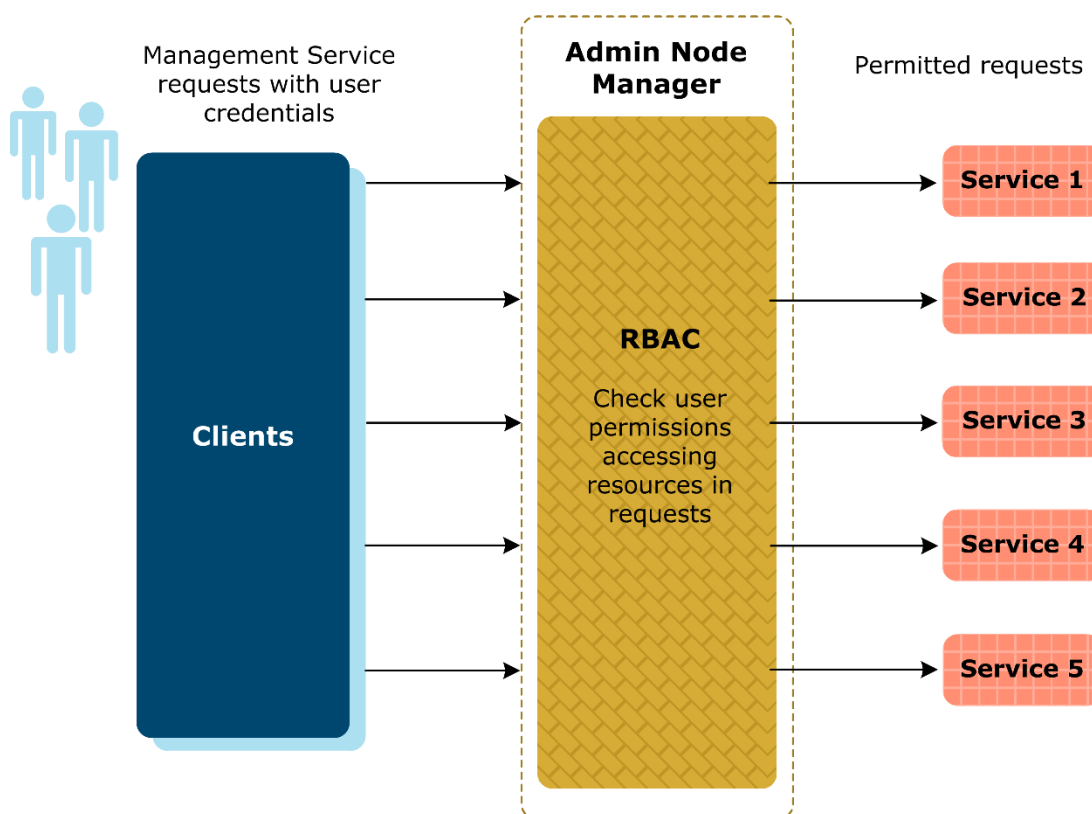
Client trong mô hình client/server là một máy trạm mà chỉ được sử dụng bởi 1 người dùng với để muốn thể hiện tính độc lập cho nó. Bản thân mỗi một client cũng đã được tích hợp nhiều chức năng trên hệ điều hành mà nó chạy, nhưng khi được nối vào một mạng internet theo mô hình client/server thì nó còn có thể sử dụng thêm các chức năng do hệ điều hành mạng cung cấp với nhiều dịch vụ khác nhau (cụ thể là các dịch vụ do các server trên mạng này cung cấp), ví dụ như nó có thể yêu cầu lấy dữ liệu từ một server hay gửi dữ liệu lên server đó.

Server còn được định nghĩa như là một máy tính nhiều người sử dụng (multiuser computer). Vì một server phải quản lý nhiều yêu cầu từ các client trên mạng cho nên nó hoạt động sẽ tốt hơn nếu hệ điều hành của nó là đa nhiệm với các tính năng hoạt động độc lập song song với nhau. Server cung cấp và điều khiển các tiến trình truy cập vào tài nguyên của hệ thống. Các ứng dụng chạy trên server phải được tách rời nhau để một lỗi của ứng dụng này không làm hỏng ứng dụng khác. Tính đa nhiệm đảm bảo một tiến trình không sử dụng toàn bộ tài nguyên hệ thống.

4.1.1.3 Mô hình phân quyền Role-Based Access Control (RBAC)

Trong an ninh đối với các hệ thống máy tính, điều khiển truy cập trên cơ sở vai trò (tiếng Anh: Role-Based Access Control - viết tắt là RBAC) là một trong số các phương pháp điều khiển và đảm bảo quyền sử dụng cho người dùng. Đây là một phương pháp có thể thay thế Điều khiển truy cập tùy quyền (discretionary access control - DAC) và Điều khiển truy cập bắt buộc (mandatory access control - MAC).

Trong nội bộ một tổ chức, các vai trò (roles) được kiến tạo để đảm nhận các chức năng công việc khác nhau. Mỗi vai trò được gắn liền với một số quyền hạn cho phép nó thao tác một số hoạt động cụ thể ('permissions'). Các thành viên trong lực lượng cán bộ công nhân viên (hoặc những người dùng trong hệ thống) được phân phối một vai trò riêng, và thông qua việc phân phối vai trò này mà họ tiếp thu được một số những quyền hạn cho phép họ thi hành những chức năng cụ thể trong hệ thống.



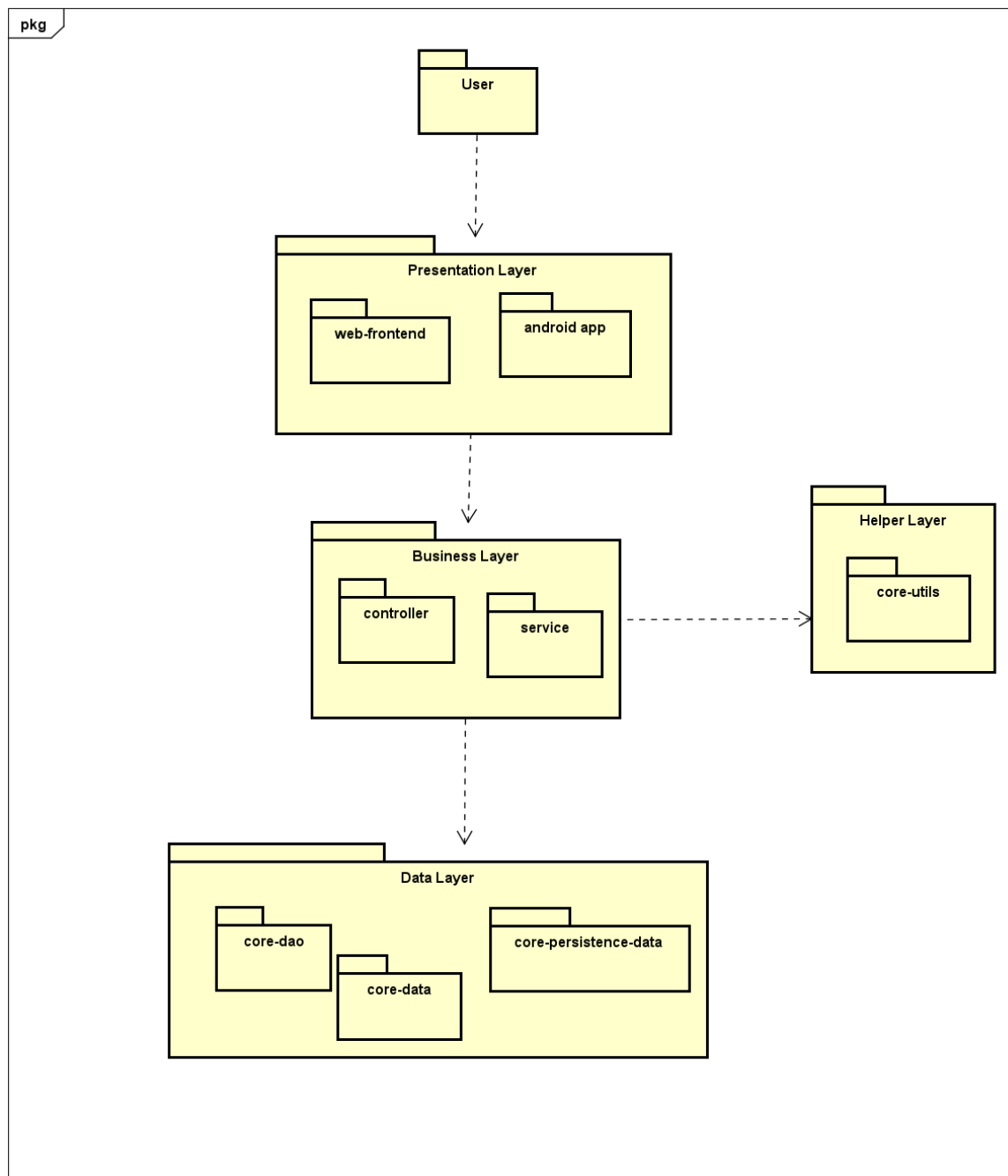
Hình 4.3 Cơ chế hoạt động của RBAC.

Mô hình này được tạo dựng từ 4 thành phần chính: Where, What, Who, và Glue. Trong đó:

- Where hoặc Scope đại diện cho các đối tượng, tác vụ từ role cụ thể được gán cho tới những đối tượng được hỗ trợ, chẳng hạn như 1 tài khoản người dùng cá nhân, 1 nhóm người dùng, hoặc toàn bộ mô hình, tổ chức.
- What hoặc Role tương ứng với những gì mà role có thể thực hiện được. Trên thực tế thì Exchange Server 2010 có tổng cộng tất cả 65 role dành cho người sử dụng.
- Who hoặc Role Group như chúng ta đã đề cập trước đó, chỉ là bộ tổ hợp nhiều role được sắp xếp với nhau. Có thể kết hợp với Scope để tạo thành đối tượng Role Assignment hoàn chỉnh.

Việc sử dụng các nhóm role chưa được định nghĩa - predefined của RBAC có thể đơn giản hóa quá trình phân quyền quản trị tới nhiều tài khoản Administrator khác nhau.

4.1.2 Thiết kế tổng quan



Hình 4.4 Biểu đồ phụ thuộc gói

Chi tiết hơn về các gói:

Tầng trình diễn (Presentation Layer)

- web-front-end: chứa các file thực hiện việc tạo giao diện web cho phần mềm
- android-app : chứa các file thực hiện tạo giao diện di động cho phần mềm

Tầng nghiệp vụ (Business Layer):

- controller: thực hiện kết nối với tầng trình diễn, thực hiện nghiệp vụ
- service: mô tả phương thức hỗ trợ cho từng nghiệp vụ.

Tầng dữ liệu (Data Layer):

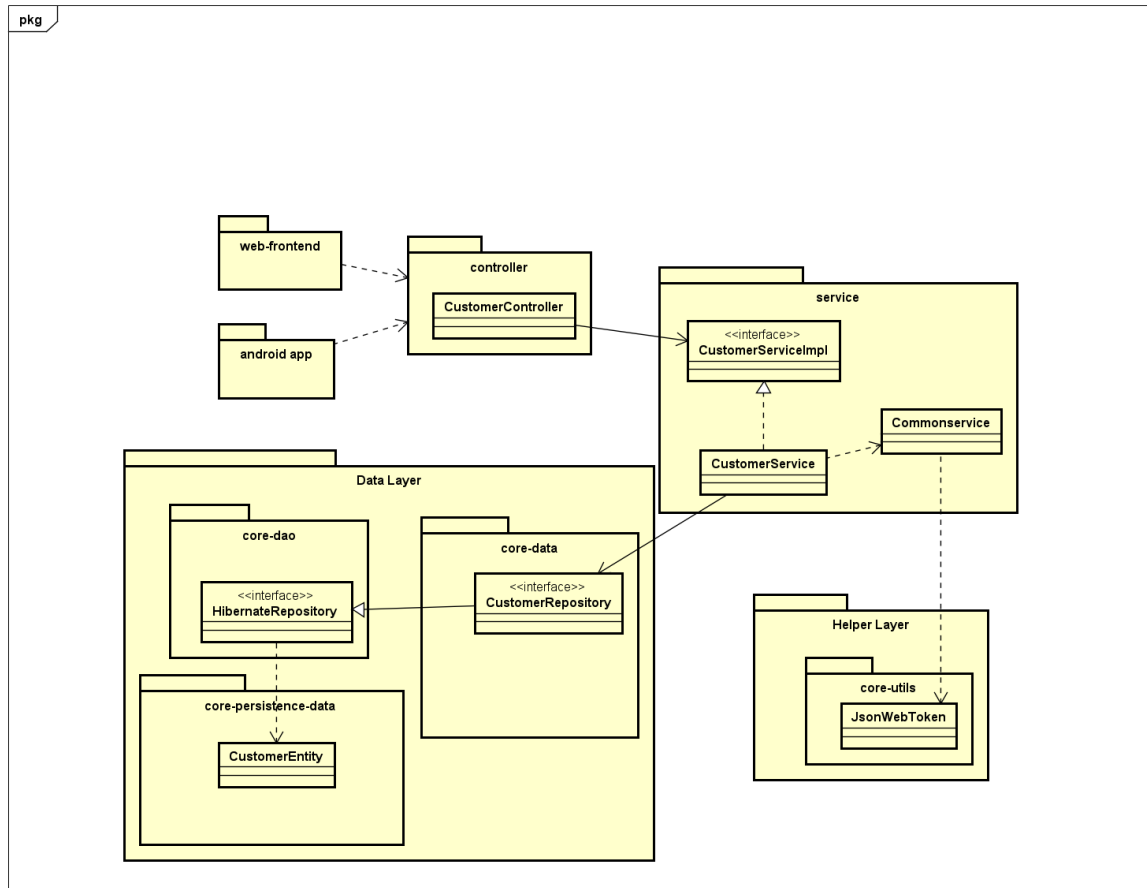
- core-dao: mô tả phương thức tương tác với cơ sở dữ liệu riêng cho từng entity
- core-data: chứa phương thức tương tác với cơ sở dữ liệu chung cho hệ thống
- core-persistence-data: chứa các entity ánh xạ tương ứng với từng bảng trong cơ sở dữ liệu.

Tầng hỗ trợ (Helper Layer):

- core-utils: chứa các phương thức dùng chung cho toàn bộ phần mềm.

4.1.3 Thiết kế chi tiết gói

4.1.3.1 Chi tiết gói cho thành phần quản lý khách hàng



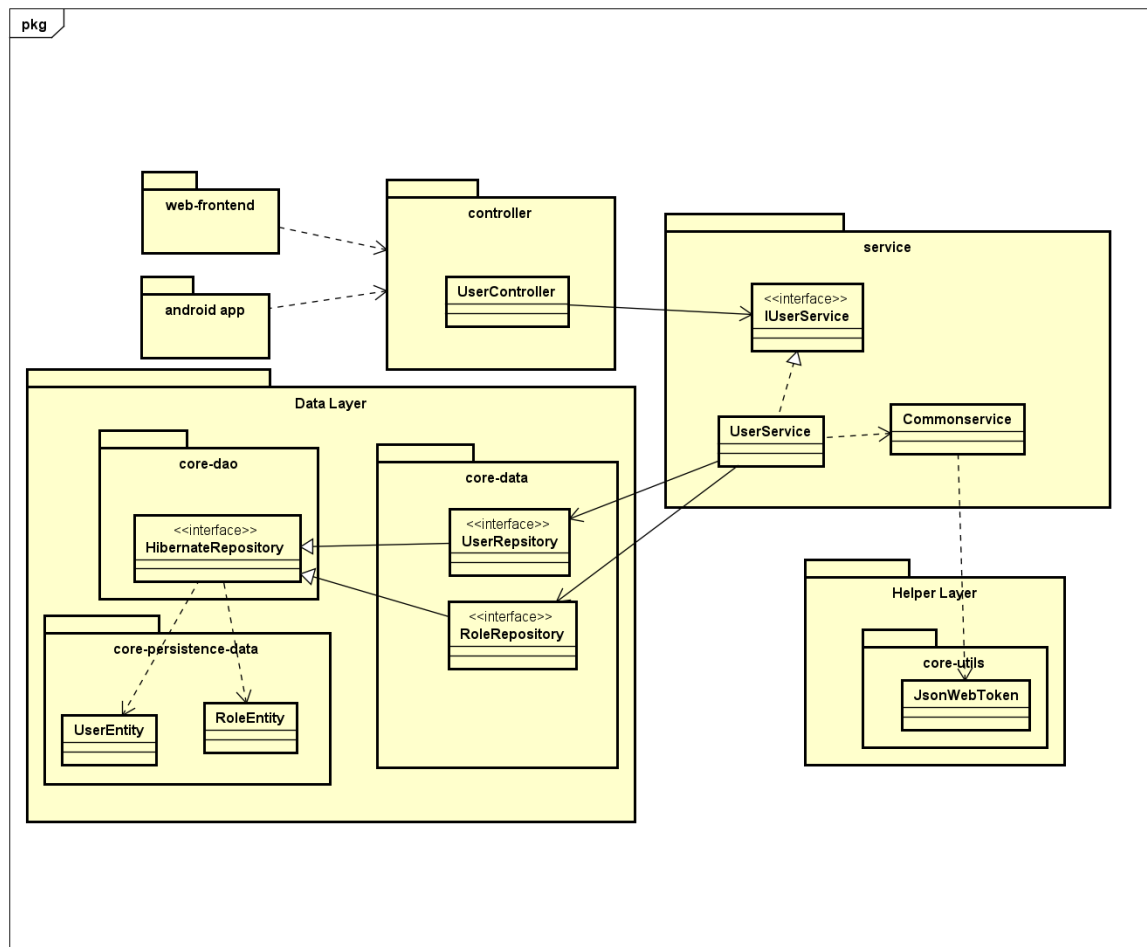
Hình 4.5 Thiết kế gói chi tiết thành phần quản lý khách hàng

Hình 4.5 Mô tả chi tiết các gói có liên quan đến các chức năng quản lý khách hàng, trong đó:

- Gói core-persistence-data: Chứa các lớp thực thể liên kết với các trường trong CSDL.
- Gói core-data: Chứa phương thức thực hiện các truy vấn csdl dùng chung cho cả phân hệ.
- Gói core-dao: Class HibernateRepository hỗ trợ khởi tạo kết nối tới CSDL

- Gói service: Thực hiện các yêu cầu nghiệp vụ từ controller
- controller: thực hiện kết nối với tầng trình diễn, thực hiện nghiệp vụ
- Gói web-front-end, android app: Chứa các thành phần giao diện cho chức năng quản lý nhân viên.
- Gói Helper: chứa các phương thức dùng chung cho toàn bộ phần mềm.

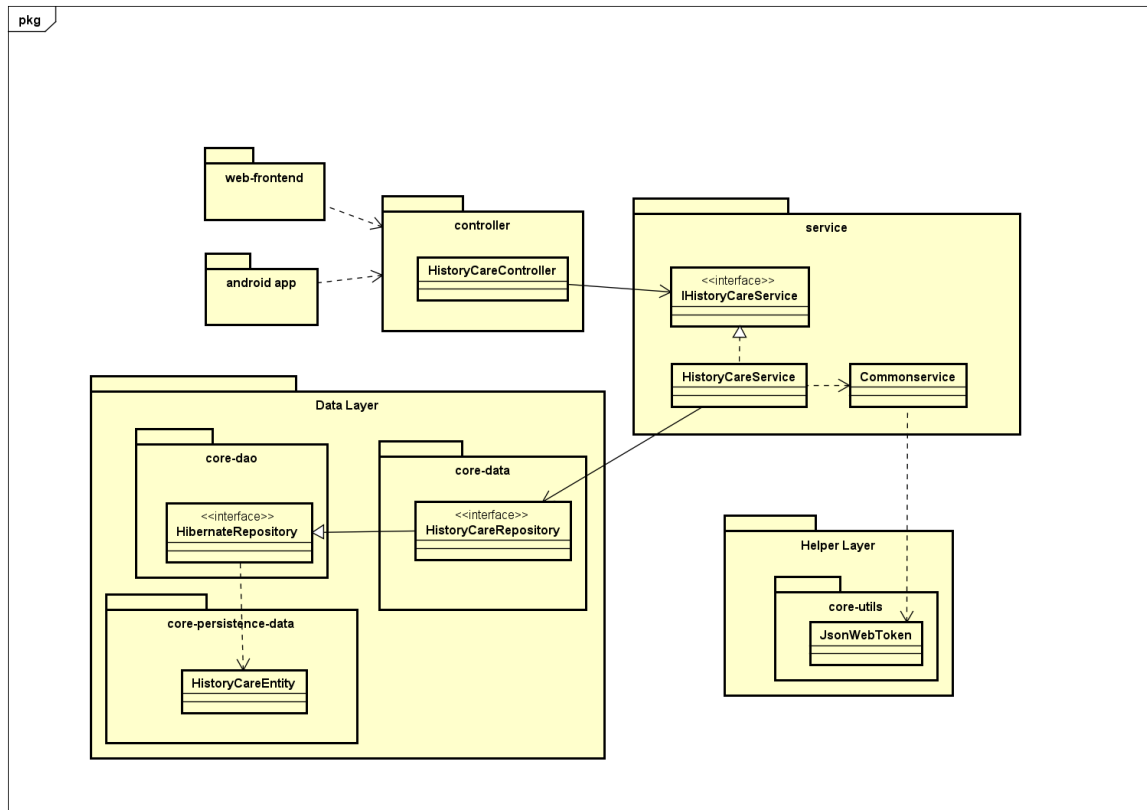
4.1.3.2 Chi tiết gói cho thành phần phân quyền và quản lý người dùng



Hình 4.6 Thiết kế gói chi tiết cho phân quyền và quản lý người dùng.

Hình 4.6 Mô tả chi tiết các gói có liên quan đến các chức năng phân quyền và quản lý người dùng, trong đó nhiệm vụ của các gói sẽ tương tự như đã đặc tả ở phần 4.1.3.1.

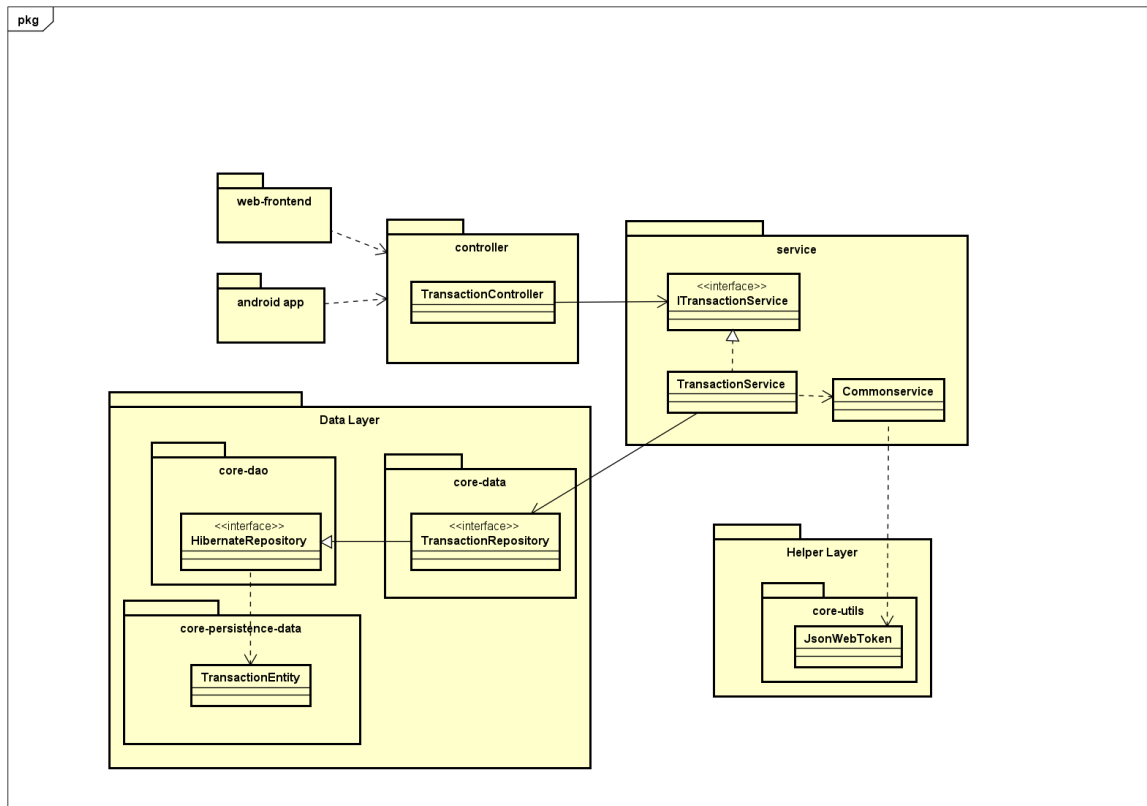
4.1.3.3 Chi tiết gói cho thành phần quản lý phiên chăm sóc



Hình 4.7 Thiết kế chi tiết gói cho phần quản lý phiên chăm sóc

Hình 4.7 Mô tả chi tiết các gói có liên quan đến các chức năng quản lý phiên chăm sóc, trong đó nhiệm vụ của các gói sẽ tương tự như đã đặc tả ở phần 4.1.3.1.

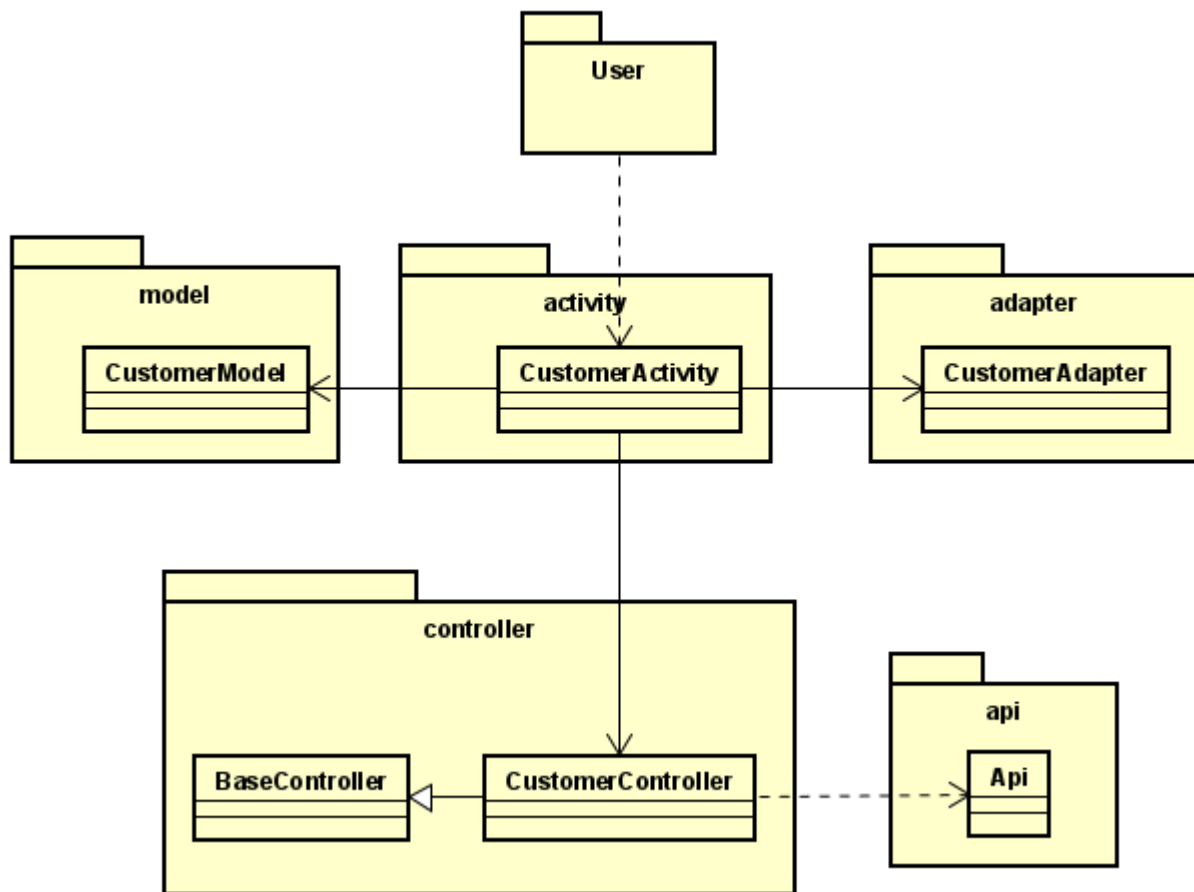
4.1.3.4 Chi tiết gói thành phần quản lý giao dịch



Hình 4.8 Chi tiết gói thành phần quản lý giao dịch

Hình 4.8 Mô tả chi tiết các gói có liên quan đến các chức năng quản lý giao dịch, trong đó nhiệm vụ của các gói sẽ tương tự như đã đặc tả ở phần 4.1.3.1.

4.1.3.5 Chi tiết gói thành phần quản lý khách hàng (android app)



Hình 9 Chi tiết gói thành phần quản lý giao dịch(android app)

Mô tả chi tiết các gói có liên quan đến các chức năng quản lý khách hàng trên app di động, trong đó:

- Gói activity, adapter: Chứa các lớp thực hiện kết nối giao diện.
- Gói model: Chứa các lớp thực thể
- Gói controller: Thực hiện kết nối với tầng giao diện, thực hiện nghiệp vụ
- Gói api : Thực hiện kết nối với backend web service để truy xuất dữ liệu từ cơ sở dữ liệu.

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

4.2.1.1 Nguyên tắc thiết kế chung

Dưới đây là một số tiêu chí thiết kế phần mềm được áp dụng trong suốt quá trình xây dựng phần mềm. Việc áp dụng các quy tắc này sẽ giúp cho phần mềm trở nên nhất quán, đẹp hơn và đặc biệt trở nên thân thiện với người dùng phần mềm. Các tiêu chuẩn này đề cao tính nhất quán, tăng tính phản hồi cho người dùng sử dụng các màu sắc trang nhã, dễ nhìn không lòe loẹt hoặc dùng các màu sắc quá nổi bật nhằm phù hợp cho phần lớn các nhóm người dùng khác nhau. Các quy tắc sẽ được mô tả cụ thể bên dưới:

a) Nội dung một chiều

Quy tắc về thiết kế thông tin đối với nội dung một chiều được trình bày ở bảng sau:

Bảng 1 Quy tắc thiết kế thông tin với nội dung một chiều

STT	Thông tin	Các nguyên tắc
1	Bảng	<ul style="list-style-type: none">• Hiện thị số dòng đang hiển thị.• Hiện thị số trang đang hiển thị.• Hiện thị tổng số trang và tổng số dòng hiện có

STT	Thông tin	Các nguyên tắc
		<ul style="list-style-type: none"> • Tiêu đề bảng cần làm nổi bật so với nội dung trong dòng dữ liệu. Quy tắc đảm bảo độ tương phản: <ul style="list-style-type: none"> ▪ Tiêu đề có màu khác với nội dung (Tiêu đề: #455a64; Nội dung: #000000) ▪ Cỡ chữ của tiêu đề lớn hơn nội dung (Tiêu đề: 600; Nội dung: 400) ▪ Trên tiêu đề có phần sắp xếp • Với những bảng quá dài cần có tiêu đề phụ phía dưới bảng. • Hiện thị giao diện tìm kiếm nhanh ở bảng
3	Lưới thông tin	<ul style="list-style-type: none"> • Lưới thông tin được chia phù hợp với kích thước màn hình hiển thị tương ứng. Số hàng tối đa trong một màn hình 5 hàng (Không tính hàng tiêu đề). Số cột tối đa trên một màn hình là 7 • Luôn có tiêu đề cho các phần nội dung. Tiêu đề phải được làm nổi bật: <ul style="list-style-type: none"> ▪ Cỡ chữ: 21px ▪ Font-weight: bold

STT	Thông tin	Các nguyên tắc
		<ul style="list-style-type: none"> Thông tin cần được phân chia theo chiều ngang phù hợp (vì những hạn chế về mở rộng chiều ngang). Nội dung chỉ phải có độ rộng tối đa bằng kích thước màn hình, đối với các màn hình có độ rộng khác nhau thì hỗ trợ cơ chế co giãn phù hợp đảm bảo bố trí giống nhau trên các màn hình khác nhau Với những nội dung khác biệt cần có những thẻ, đường biên, tiêu đề phân biệt rõ ràng. <ul style="list-style-type: none"> Độ rộng viền: 1px Kiểu viền: solid Màu viền: #dee2e6

b) Nội dung tương tác

Quy tắc về thiết kế thông tin đối với nội dung tương tác được trình bày ở bảng sau:

Bảng 2 Quy tắc thiết kế thông tin với nội dung tương tác

STT	Thông tin	Các nguyên tắc
1	Liên kết	<ul style="list-style-type: none"> Liên kết cần thay đổi (đổi màu, sáng lên) khi trỏ chuột được đưa qua

STT	Thông tin	Các nguyên tắc
2	Menu	<ul style="list-style-type: none"> Với giao diện đang hiển thị thì đề mục trong menu cần được làm nổi bật Các đề mục trong menu sẽ được làm nổi khi trỏ chuột được đưa qua Menu sẽ chỉ hiện icon trong chế độ thu gọn menu Trong chế độ thu gọn menu: nội dung đề mục sẽ hiển thị khi trỏ chuột đưa qua các icon.
3	Phím tương tác	<ul style="list-style-type: none"> Phím tương tác có thể hiển thị nội dung hoặc chỉ icon tùy giao diện. Phím tương tác sẽ được làm nổi khi trỏ chuột được đưa qua. Với những phím cần có nội dung mô tả: nội dung sẽ hiện ra thì trỏ chuột được đưa qua. Phím tương tác có hiệu ứng khi nhấn chọn.
4	Biểu mẫu nhập liệu	<ul style="list-style-type: none"> Ô đang nhập dữ liệu sẽ được đánh dấu nổi bật so với những ô khác Giao diện nhập thời gian phải tùy chỉnh để chọn thời gian dễ dàng (hạn chế nhập thời gian từ bàn phím).

STT	Thông tin	Các nguyên tắc
		<ul style="list-style-type: none"> • Với ô dữ liệu hạn chế nhập cần đánh dấu làm mờ so với những ô khác • Giao diện validate dữ liệu cần trả về ngay tại ô nhập dữ liệu • Luôn cung cấp giá trị mặc định. • Chỉ ra loại đơn vị cho dữ liệu nhập vào.

Đối với một số form dữ liệu cần kiểm tra tính hợp lệ của dữ liệu người dùng nhập vào: Nếu là các form cơ bản, phân hệ cung cấp hàm kiểm tra tính hợp lệ của dữ liệu do người dùng nhập vào viết bằng javascripts (ví dụ, kiểm tra một ô có rỗng hay không). Các form phức tạp hơn sẽ được kiểm tra bằng các hàm viết bằng Java (ví dụ, kiểm tra giá trị nhập vào có khớp với giá trị lưu trong cơ sở dữ liệu hay không).

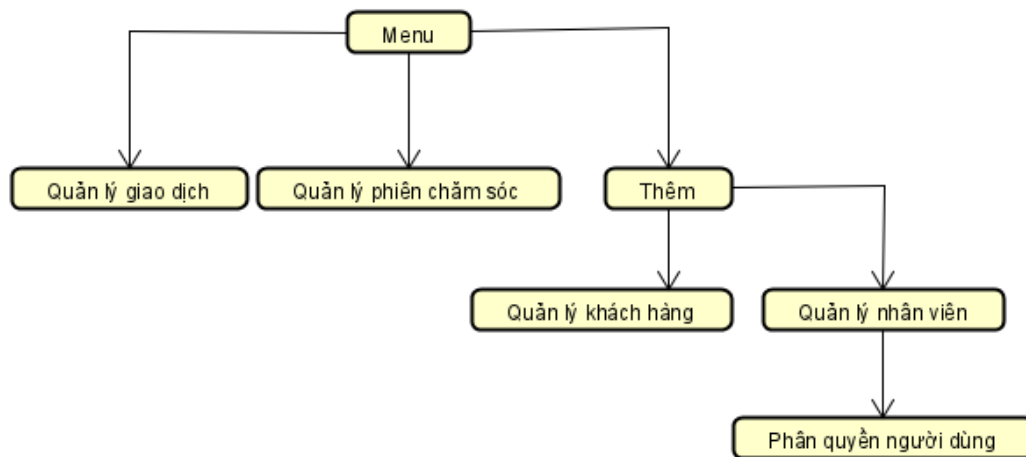
Bảng 3 Quy tắc thiết kế form kiểm tra tính hợp lệ của dữ liệu đầu vào

Đối tượng cần kiểm tra	Thao tác kiểm tra	Dung lỗi
Mọi ô nhập liệu văn bản	Người dùng cần điền đủ thông tin	Hiện thị thông báo yêu cầu điền lại
Địa chỉ email	Có chứa ký tự @, ít nhất 1 ký tự. @ không phải ký tự đầu tiên được nhập vào Có ít nhất 2 ký tự giữa @ và.	Hiện thị thông báo nhập sai định dạng email

4.2.1.2 Thiết kế màn hình

Đối với phần thiết kế màn hình này sẽ đưa ra sơ đồ phân cấp các chức năng đồng thời từ các giao diện thiết kế phải tuyệt đối tuân thủ theo các nguyên tắc thiết kế chung đã được trình bày chi tiết ở phần 4.2.1.1. Chi tiết thiết kế sẽ được trình bày cụ thể ở phần bên dưới:

a) Sơ đồ phân cấp chức năng



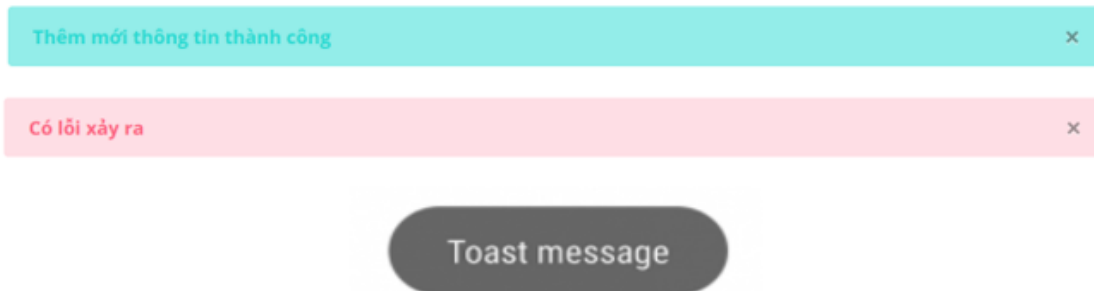
Hình 10 Sơ đồ phân cấp chức năng hệ thống

b) Thiết kế thông báo

Thông báo dạng cố định

Thông báo cố định sẽ được làm nổi bật khi hiển thị:

- Xuất hiện trên/dưới cùng của màn hình wireframe.
- Thông báo được đặt trong một thẻ màu nổi bật so với màu nền.



Hình 11 Ví dụ các mẫu thông báo cố định.

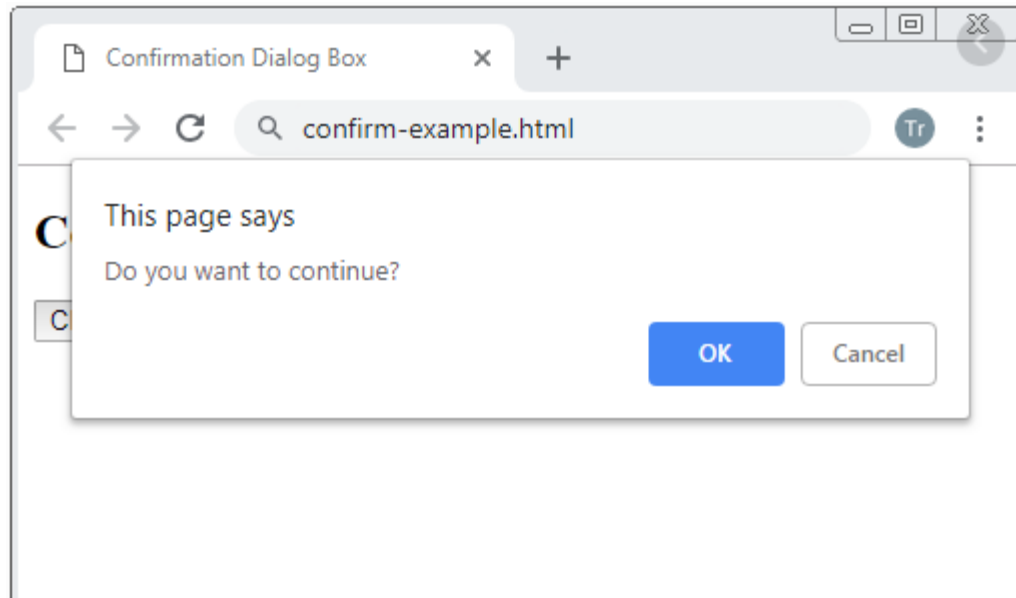
Thông báo cố định gồm 3 loại:

- Thông báo thành công: màu xanh
 - Màu nền, viền: #93ede9
 - Màu chữ: #26dad2
- Thông báo lỗi, xóa: màu đỏ
 - Màu nền, viền: #fedee5
 - Màu chữ: #fc6180

c) Hộp thoại

Hộp thoại sẽ hiển thị khi phần mềm yêu cầu người dùng xác nhận một thao tác quan trọng, khi hộp thoại hiển thị:

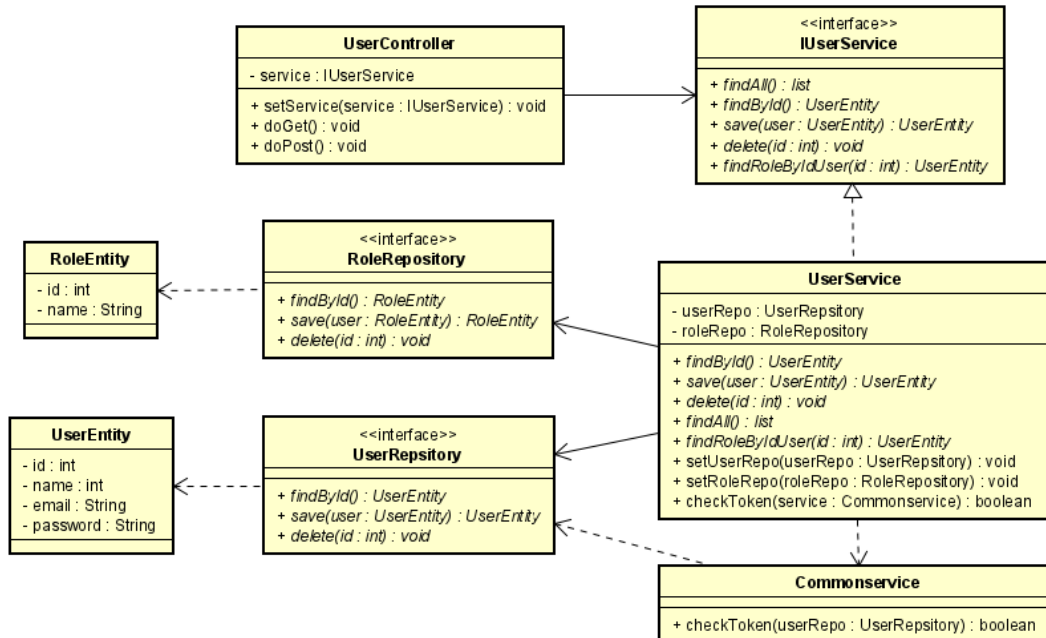
- Xuất hiện ngay giữa màn hình.
- Các thành phần khác của website sẽ được làm tối (không thể tương tác).
- Các button được thiết kế tương ứng với thao tác xác nhận trên hộp thoại.



Hình 12 Ví dụ về hộp thoại xác nhận.

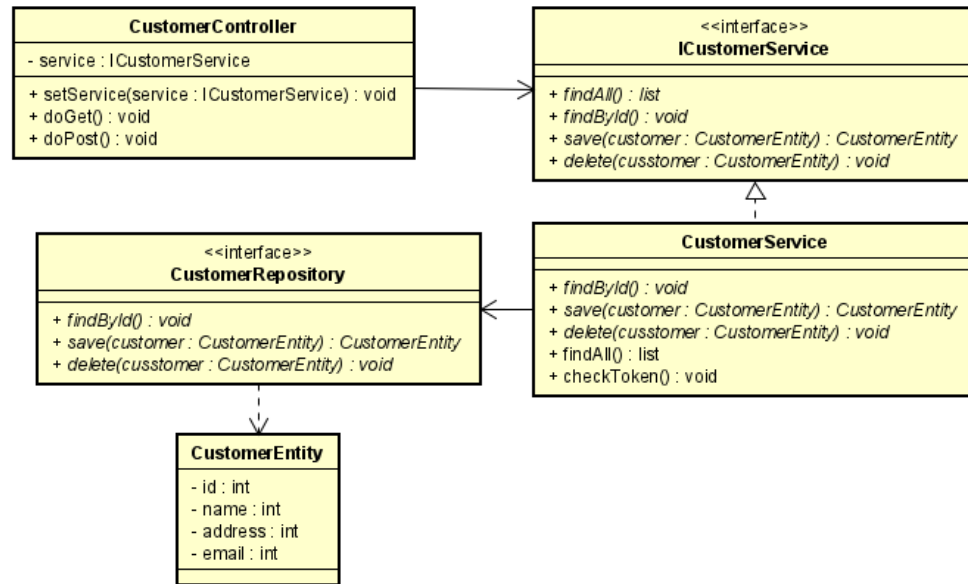
4.2.2 Thiết kế lớp

4.2.2.1 Thành phần phân quyền và quản lý người dùng



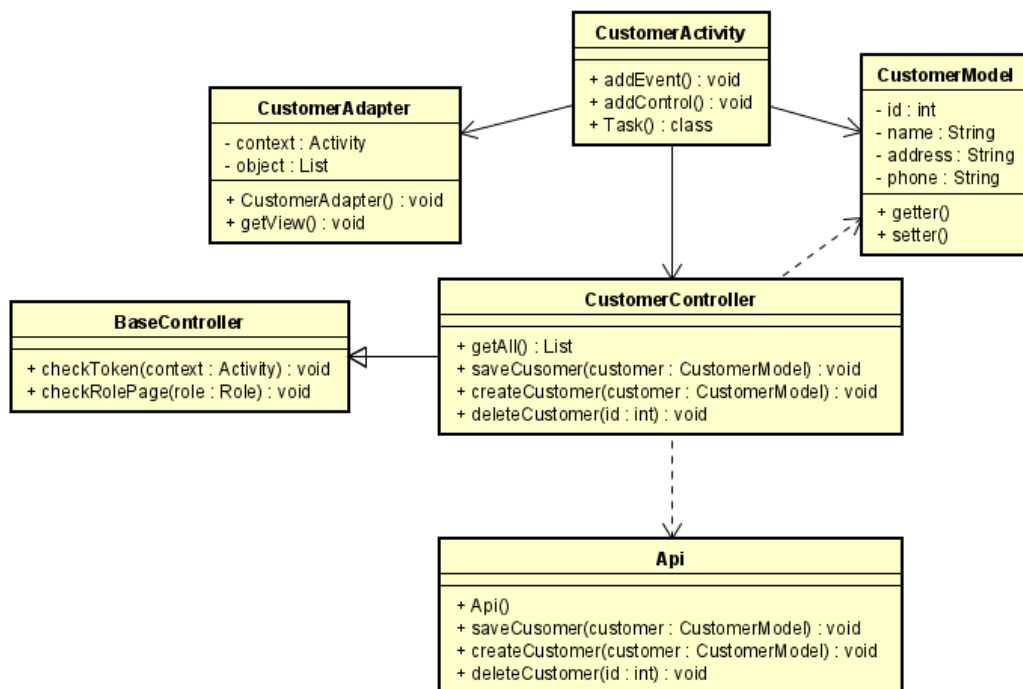
Hình 13 Thiết kế chi tiết một số lớp phân quyền và quản lý người dùng

4.2.2.2 Thành phần quản lý khách hàng



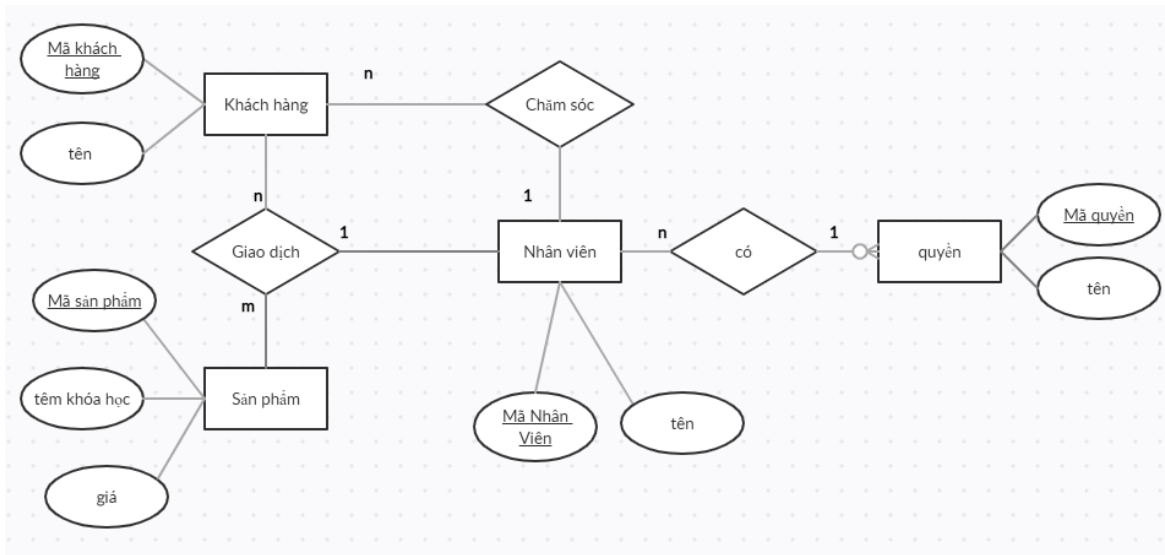
Hình 14 Thiết kế chi tiết một số lớp cho thành phần quản lý khách hàng

4.2.2.3 Thành phần quản lý khách hàng (android app)

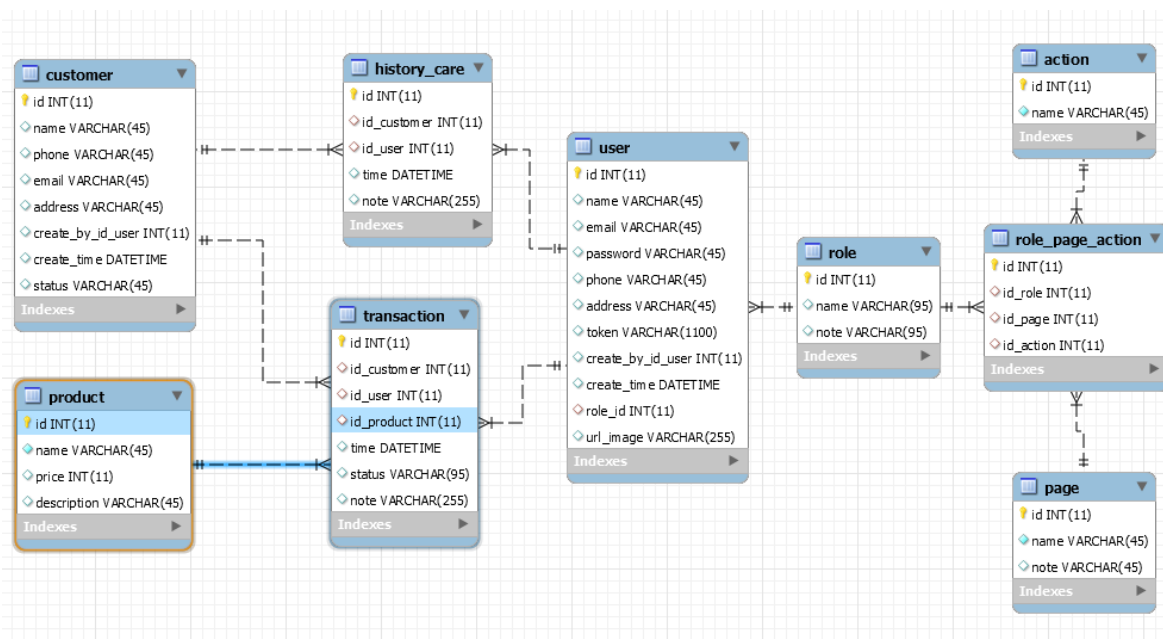


Hình 15 Thiết kế chi tiết một số lớp cho thành phần quản lý khách hàng(android app)

4.2.3 Thiết kế cơ sở dữ liệu



Hình 16 Sơ đồ thực thể liên kết ER cho CSDL chung



Hình 17 Sơ đồ CSDL tổng quát

Hình 4.14 và **Hình 4.15** mô tả tổng quan về kiến trúc của cả hệ thống tuy nhiên phần trình bày chi tiết chỉ xin phép trình bày các bảng chính nhất, bao gồm các bảng sau:

- User: Lưu trữ người dùng hệ thống
- Transaction: Lưu các giao dịch giữa user và customer
- History_care : Lưu lịch sử chăm sóc giữa user và customer

Bảng 4 Thiết kế chi tiết bảng User

Thuộc tính	Kiểu dữ liệu	Kích thước	Giải thích	Ràng buộc
id	int	–	Id lưu trữ	Khóa chính
name	Varchar	250	Tên user	Khóa ngoài bảng Session
phone	Varchar	250	Điện thoại người dùng	-
email	Text	-	Email người dùng	
password	Varchar	250	Mật khẩu đã được mã hóa	-
token	Varchar	1000	Mã truy cập	-
create_by_id_user	Varchar	10	Id người tạo	-

Thuộc tính	Kiểu dữ liệu	Kích thước	Giải thích	Ràng buộc
create_time	Datetime	-	Thời gian tạo	-
role_id	Varchar	10	Id quyền	Khóa ngoài
url_image	Varchar	250	Link ảnh	-

Bảng 5 Thiết kế chi tiết bảng Transaction

Thuộc tính	Kiểu dữ liệu	Kích thước	Giải thích	Ràng buộc
id	int	-	Id lưu trữ	Khóa chính
id_customer	int	-	Mã khách hàng	Khóa ngoài bảng customer
id_user	int	-	Mã người dùng	Khóa ngoài bảng user
id_product	int	-	Mã sản phẩm	Khóa ngoài bảng product
time	Varchar	250	Mật khẩu đã được mã hóa	-
status	Varchar	1000	Mã truy cập	-

Thuộc tính	Kiểu dữ liệu	Kích thước	Giải thích	Ràng buộc
create_by_id_user	int	-	mã người tạo	-
create_time	Datetime	-	Thời gian tạo	-
note	Varchar	250	Ghi chú	-

Bảng 6 Thiết kế chi tiết bảng History_care

Thuộc tính	Kiểu dữ liệu	Kích thước	Giải thích	Ràng buộc
id	int	-	Id lưu trữ	Khóa chính
id_customer	int	-	Mã khách hàng	Khóa ngoài bảng customer
id_user	int	-	Mã người dùng	Khóa ngoài bảng user
time	DateTime	-	Thời gian chăm sóc	-
note	Varchar	1000	Ghi chú	-
create_by_id_user	int	-	mã người tạo	-
create_time	Datetime	-	Thời gian tạo	-

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 7 Danh sách thư viện và công cụ sử dụng

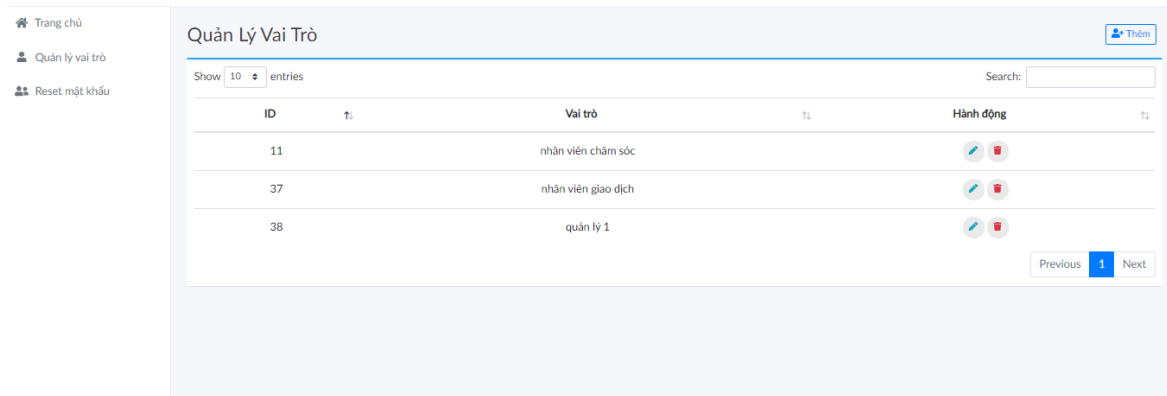
Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	IntelliJ IDEA 64 bit	https://www.jetbrains.com/idea/
Ngôn ngữ lập trình	Java 8	http://www.oracle.com
Quản trị CSDL	MySql 8.0	https://www.mysql.com/
Công cụ quản lý	Maven 3.6.2	https://maven.apache.org/
Framework	Spring boot 2.1.1	https://hibernate.org/
Thư viện	Gson 2.8.5	https://mvnrepository.com/

4.3.2 Kết quả đạt được

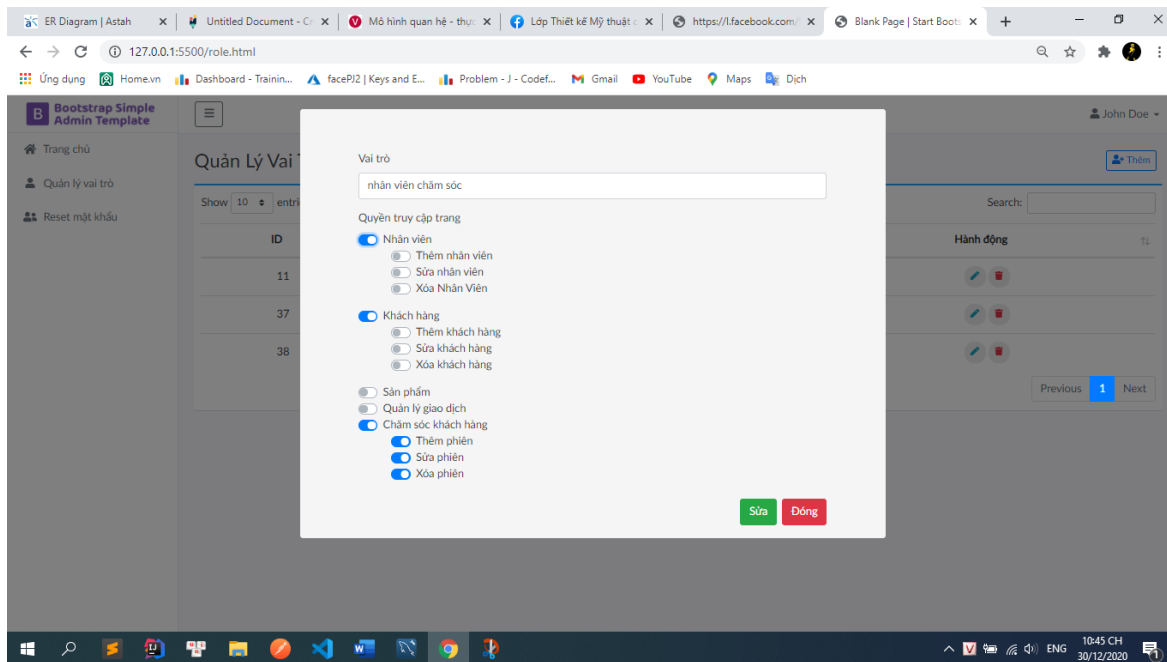
Sau thời gian thực hiện đồ án đã xây dựng thành công phần mềm. Phần mềm bao gồm 4 module chính sau:

- Thành phần phân quyền và quản lý truy cập người dùng
- Thành phần quản lý khách hàng
- Thành phần quản lý phiên chăm sóc
- Thành phần quản lý giao dịch

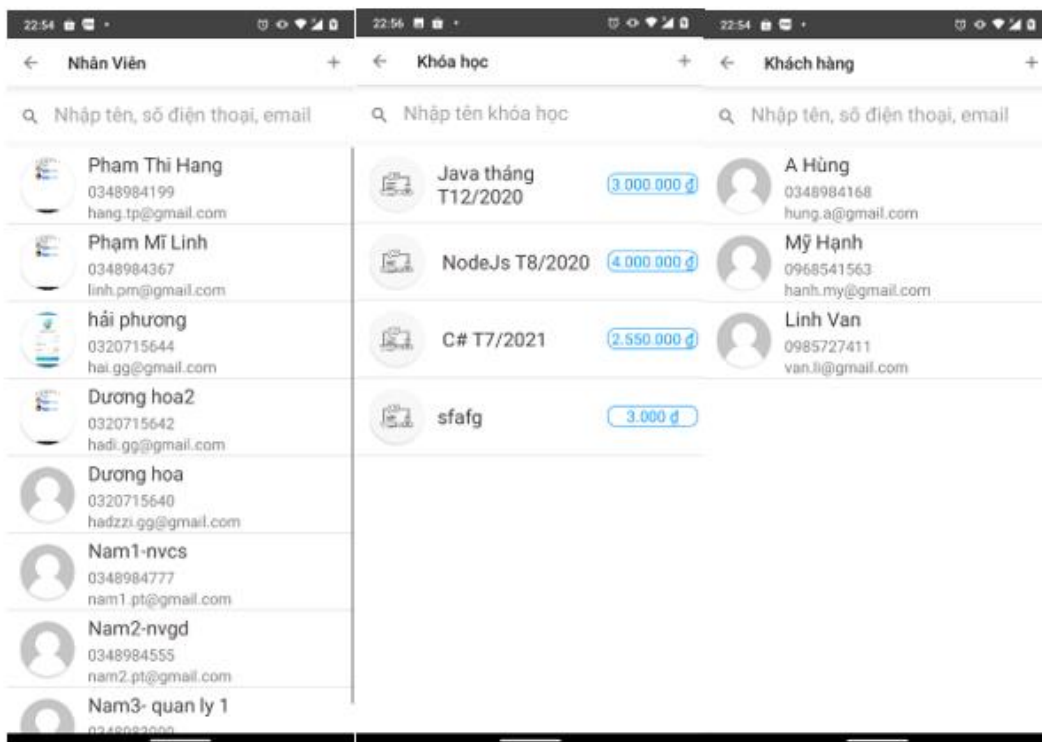
4.3.3 Minh hoạ các chức năng chính



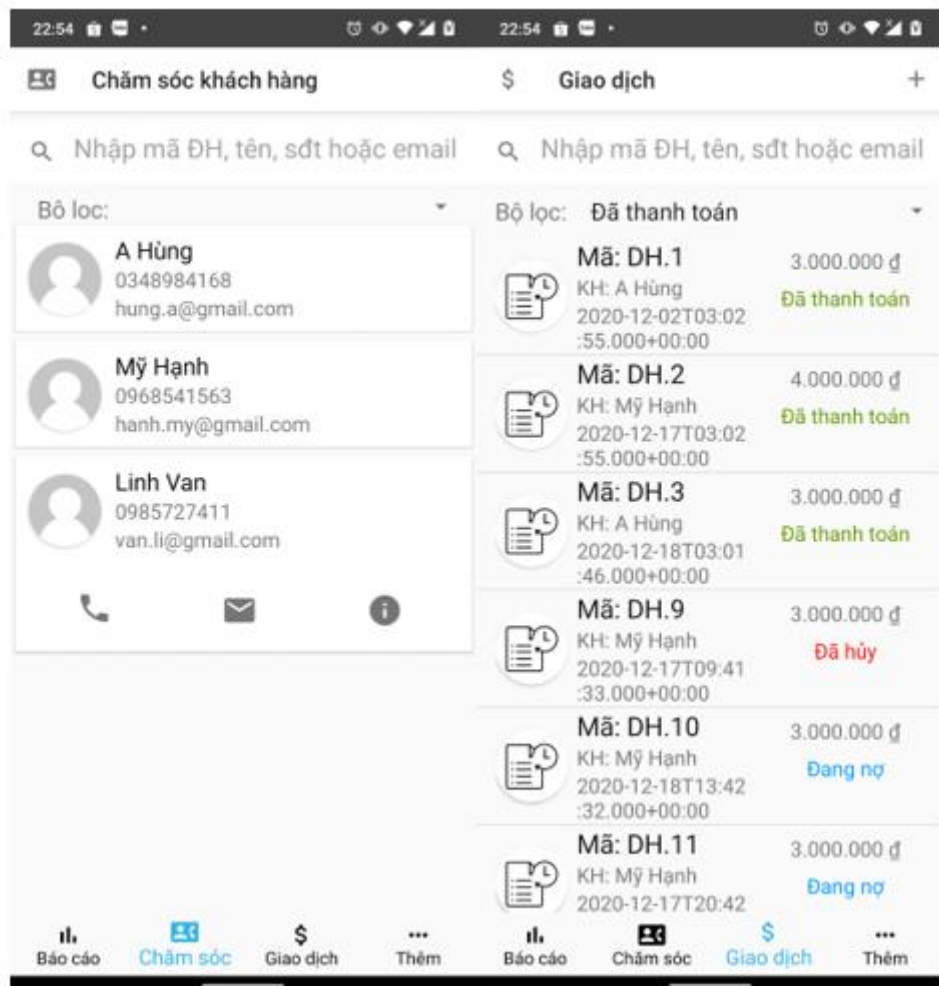
Hình 18 Giao diện quản lý quyền trên web số 1



Hình 19 Giao diện quản lý quyền trên web số 2



Hình 20 Giao diện di động quản lý nhân viên, khách hàng, khóa học



Hình 21 Giao diện di động quản lý phiên chăm sóc, giao dịch.

Chương 5 Các giải pháp và đóng góp nổi bật

Chương này sẽ trình bày những đóng góp nổi bật của cá nhân trong việc xây dựng phần mềm bao gồm xây dựng phần mềm hỗ trợ quản lý khách hàng.

5.1 Xây dựng phần mềm hỗ trợ quản lý khách hàng

5.1.1 Bài toán đặt ra

Xây dựng một công cụ thực hiện việc quản lý khách hàng gồm quản lý giao dịch và chăm sóc khách hàng. Chính vì vậy dẫn đến việc ra đời một phần mềm quản lý và hỗ trợ người dùng quản lý.

Bên cạnh việc hỗ trợ nghiệp vụ quản lý thì phần mềm cần phải tích hợp và chạy được trên nhiều nền tảng hệ điều hành khác nhau trên các máy tính, thiết bị di động để đảm bảo tính phổ biến. Bên cạnh đó không chỉ hỗ trợ quản lý khách hàng mà phần mềm cần hỗ trợ người dùng quản lý giao dịch và chăm sóc khách hàng.

Trong thời gian của Project 3 và sự nỗ lực của cá nhân thì phần mềm đã được ra đời và chi tiết sẽ trình bày ở bên dưới.

5.1.2 Giải quyết bài toán

Phần mềm được xây dựng là sự tổng hợp của 6 module chính bao gồm:

- Thành phần phân quyền và quản lý truy cập người dùng
- Thành phần quản lý khách hàng
- Thành phần quản lý phiên chăm sóc

- Thành phần quản lý giao dịch

Như vậy có thể thấy phần mềm là một hệ thống tổng thể không chỉ thực hiện công việc quản lý đơn thuần mà còn hỗ trợ người dùng quản lý truy cập và phân quyền. Ở phần này cá nhân xin trình bày những đóng góp và xây dựng của mình trong việc phân quyền, truy cập người dùng, quản lý khách hàng và quản lý phiên chăm sóc.

5.1.2.1 Phân quyền và quản lý người dùng

Trong nội bộ một tổ chức, các vai trò (roles) được kiến tạo để đảm nhận các chức năng công việc khác nhau. Mỗi vai trò được gắn liền với một số quyền hạn cho phép nó thao tác một số hoạt động cụ thể ('permissions'). Các thành viên trong lực lượng cán bộ công nhân viên (hoặc những người dùng trong hệ thống) được phân phối một vai trò riêng, và thông qua việc phân phối vai trò này mà họ tiếp thu được một số những quyền hạn cho phép họ thi hành những chức năng cụ thể trong hệ thống. Trên mô hình RBAC đã trình bày ở mục 4.1.1 thì phần mềm đã xây dựng cung cấp chi tiết một hệ thống phân quyền đến từng trang, từng hành động của người dùng, người quản trị hệ thống cao nhất sẽ thực hiện việc tạo và cấp quyền cho người dùng.

5.1.2.2 Quản lý khách hàng

Tiếp đó, đối với lượng dữ liệu khách hàng lớn, để đơn giản hóa quản lý, phần mềm đã xung cấp một giao diện ứng dụng di động để giải quyết việc này, người sử dụng hệ thống có được cấp quyền quản lý khách hàng này hay không, hành động(xem, thêm, sửa, xóa) tương ứng hay không sẽ tiến hành công việc quản lý dễ dàng.

5.1.2.3 Quản lý phiên chăm sóc

Cuối cùng, ngoài việc quản lý thông tin chăm sóc đơn thuần như quản lý khách hàng, khóa học, ... như thêm, sửa, xóa đã đề cập ở việc quản lý khách hàng, nhân viên có quyền chăm sóc sẽ thực hiện việc chăm sóc, nhằm tránh xung đột chăm sóc cùng một thời điểm, hệ thống sẽ kiểm tra trạng thái truy cập của nhân viên vào từng

khách hàng để có thể tránh việc xung đột chăm sóc khách hàng của nhiều nhân viên tại cùng một thời điểm.

5.1.3 Kết quả đạt được

Phần mềm quản lý khách hàng cho đội tư vấn quyền sinh cơ bản hoàn thành và có đầy đủ các tính năng so với các phần mềm trên thị trường và thậm chí còn vượt bậc ở một số mặt. Trên cơ sở đó đã tạo tiền đề cho việc mở rộng thị trường đào tạo tay nghề cao ở Việt Nam.

Chương 6 Kết luận và hướng phát triển

6.1 Kết luận

Sau một quá trình nghiên cứu làm việc và dưới sự hướng dẫn tận tình của ThS. Nguyễn Mạnh Tuấn thì đồ án đã cơ bản hoàn thành. Đồ án bao gồm phần chính đó là một phần mềm quản lý khách hàng cho đội tư vấn tuyển sinh.

Phần mềm hỗ trợ quản lý khách hàng. Phần mềm là sự kế thừa và phát triển của các phần mềm đã có trong thị trường. Phần mềm cung cấp cho người dùng một môi trường để thực hiện việc quản lý khách hàng. Bên cạnh việc chỉ hỗ trợ người dùng quản lý dữ liệu thì phần mềm này còn hỗ trợ người dùng ở mức sâu hơn là phân quyền chi tiết người dùng.

Tuy đã hoàn thiện xong nhưng phần mềm vẫn còn tồn tại một số vấn đề. Giao diện chưa thực sự bắt mắt, chưa thực thân thiện với người dùng, chưa đẹp.

6.2 Hướng phát triển

Có thể nói phần mềm hỗ trợ quản lý khách hàng tuy đã hoàn thiện xong vẫn còn tồn tại một số vấn đề tồn tại và cần khắc phục.

Đầu tiên là giao diện cần xây dựng đẹp và bắt mắt người dùng. Một số ô nhập liệu đặc biệt như thanh toán, thêm nhân viên cần hỗ trợ tự động.

Sau đó , ngoài quản lý khóa học , phần mềm có thể quản lý nhiều thể loại khác , như mỹ phẩm, đồ gia dụng, nội thất, Tùy thuộc từng đối tượng khách hàng. Việc mở rộng hoàn toàn không khó và có thể thực hiện trong tương lai