

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây Dựng Website Bán Cây Cảnh Bonsai

Lê Văn Nghĩa

nghia.lv152651@sis.hust.edu.vn

**Ngành Công nghệ thông tin và truyền thông
Chuyên ngành Truyền Thông và Mạng Máy Tính**

Giảng viên hướng dẫn: PGS. TS. Tạ Hải Tùng

Chữ kí GVHD

Khoa:

Kỹ thuật máy tính

Trường:

Công nghệ thông tin và Truyền thông

HÀ NỘI, 02/2023

LỜI CẢM ƠN

Trước khi trình bày nội dung đồ án tốt nghiệp, em xin dành những dòng đầu tiên để gửi lời cảm ơn chân thành đến:

- Thầy Tạ Hải Tùng-Hiệu trưởng Trường Công Nghệ Thông Tin và Truyền Thông kiêm giảng viên chuyên ngành kỹ thuật máy tính đã trực tiếp hướng dẫn em, giúp đỡ và đưa ra những ý kiến đóng góp, để em có thể hoàn thành đồ án tốt nghiệp này.
- Các thầy,cô trong viện Công nghệ thông tin và truyền thông đã tận tình giảng dạy, truyền đạt kiến thức, kinh nghiệm quý báu của mình cho chúng em trong suốt thời gian học tập tại trường. Những lời giảng dạy đó làm nền tảng để chúng em tiếp tục hoàn thiện nâng cao năng lực nghề nghiệp của bản thân mình
- Các bạn bè, anh, chị, em trong trường Đại học Bách khoa Hà Nội cũng như tại các đơn vị em từng thực tập. Mọi người đã nhiệt tình chỉ bảo hướng dẫn, từng bước rèn luyện năng lực lập trình cho em, giúp cho tầm hiểu biết của em được mở rộng và có thể áp dụng được những điều mình đã học được tại trường đại học
- Do kiến thức, năng lực, kinh nghiệm của em khi thực hiện đồ án này còn hạn chế, thời gian thực hiện có hạn. Vì vậy đồ án còn nhiều thiếu sót, em rất mong nhận được những lời góp ý từ các thầy các cô và các bạn, để em có thể rút ra được kinh nghiệm và tránh lặp lại trong các dự án về sau.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Sau một thời gian nghiên cứu và phát triển đề tài "website bán cây cảnh bonsai" thì em nhận thấy: Thị trường cây cảnh càng lúc càng phát triển nhưng người dùng chưa thể tìm được mặt hàng ưng ý thông qua mạng phát triển hiện nay. Vì vậy em phát triển trang web bán cây cảnh bonsai để đáp ứng nhu cầu của phần lớn những người yêu thích cây cảnh nhưng chưa tìm được hướng tiếp cận, vì vậy website của em phát triển dựa trên những cây cảnh thường xuyên xuất hiện trong các dịp lễ như Đào, quất, mai, táo phục vụ nhu cầu của đại đa số mọi người, phù hợp với nhiều tầng lớp xã hội. Website em xây dựng bao gồm 1 số chức năng: Xem và tìm kiếm cây, mua hàng và cập nhật tình trạng, quản lý tài khoản, quản lý hồ sơ cây cảnh, cập nhật ảnh cho quản trị viên, cập nhật shipper, tổng hợp chi tiết sản phẩm, tổng hợp theo từng loại cây, công nợ khách hàng, thu nhập của quán trong từng thời điểm. Qua đồ án này em hiểu thêm về các công cụ lập trình em sử dụng cũng như các công nghệ làm web như vuejs và java spring boot. Em hi vọng với kết quả này, đề tài sẽ tiếp tục phát triển để phục vụ cộng đồng,

ABSTRACT

After a while of researching and developing the topic "website for selling bonsai", I noticed: The bonsai market is growing, but users have not been able to find satisfactory products through the current development network. So I developed a website to sell bonsai to meet the needs of the majority of bonsai lovers but have not found an approach, so my website is based on the bonsai trees that often appear in the garden. Holidays such as peach, kumquat, apricot, apple serve the needs of the majority of people, suitable for many social classes. The website I built includes a number of functions: View and search for trees, purchase and update status, manage accounts, manage bonsai records, update photos for administrators, update shipper, total Detailed product summary, summarizing each type of tree, customer debt, income of the shop from time to time. Through this project, I understand more about the programming tools I use as well as web technologies like vuejs and java springboot. I hope with this result, the topic will continue to develop to serve the community,

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	1
1.4 Bố cục đồ án	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	3
2.1 Khảo sát hiện trạng	3
2.2 Tổng quan chức năng	3
2.2.1 Biểu đồ use case tổng quát	3
2.2.2 Biểu đồ use case quản lý tài khoản.....	6
2.2.3 Biểu đồ usecase phân giải chức năng tìm kiếm.....	7
2.2.4 Tổng quan chức năng thêm cây của quản trị viên.....	9
2.2.5 Quy trình nghiệp vụ	11
2.3 Đặc tả chức năng	12
2.3.1 Đặc tả use case phân quyền.....	12
2.3.2 Đặc tả use case quản lý nhân viên giao hàng.....	13
2.4 Yêu cầu phi chức năng	13
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	14
3.1 Tìm hiểu và giới thiệu tổng quan về công nghệ.....	14
3.2 Mô hình phát triển của công nghệ	14
3.3 Công nghệ Font-end/Client.....	15
3.3.1 Giới thiệu ngôn ngữ JavaScripts	15
3.3.2 Framework VueJs.....	16

3.4 Công nghệ Backend/Server.....	16
3.4.1 Ngôn ngữ lập trình Java.....	16
3.4.2 FrameWork Java Spring boot	17
3.4.3 Hệ quản trị cơ sở dữ liệu	18
3.5 Công cụ sử dụng.....	18
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNG GIÁ	20
4.1 Thiết kế kiến trúc.....	20
4.1.1 Mô hình MVC	20
4.1.2 Thiết kế tổng quan.....	22
4.1.3 Thiết kế chi tiết gói	23
4.2 Thiết kế chi tiết.....	27
4.2.1 Thiết kế giao diện	27
4.2.2 Thiết kế lớp	29
4.2.3 Thiết kế cơ sở dữ liệu	36
4.3 Xây dựng ứng dụng.....	38
4.3.1 Thư viện và công cụ sử dụng.....	38
4.3.2 Kết quả đạt được	39
4.3.3 Minh họa các chức năng chính	39
4.4 Kiểm thử.....	41
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	44
5.1 Ưu và nhược điểm của mô hình MVC	44
5.1.1 Ưu điểm mô hình MVC	44
5.1.2 Nhược điểm mô hình MVC	44
5.2 Vì sao nên sử dụng mô hình MVC.....	45
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	47
6.1 Kết luận.....	47

6.2 Hướng phát triển.....	47
TÀI LIỆU THAM KHẢO.....	49

DANH MỤC HÌNH VẼ

Hình 2.1	Usecase tổng quát	4
Hình 2.2	Usecase quản lý tài khoản	6
Hình 2.3	Biểu đồ hoạt động lấy lại password	6
Hình 2.4	Usecase phân giả chức năng tìm kiếm	7
Hình 2.5	Biểu đồ hoạt động chức năng tìm kiếm	8
Hình 2.6	Usecase thêm cây của quản trị viên	9
Hình 2.7	Biểu đồ hoạt động thêm cây	9
Hình 2.8	Biểu đồ hoạt động quy trình nghiệp vụ đặt hàng	11
Hình 3.1	Mô hình Client-Server	14
Hình 4.1	Mô hình MVC	20
Hình 4.2	Quy trình xử lý của MVC	21
Hình 4.3	Biểu đồ phụ thuộc gói	22
Hình 4.4	Thiết kế chi tiết gói Admin	23
Hình 4.5	Thiết kế chi tiết gói User	24
Hình 4.6	Thiết kế chi tiết gói Shipper	25
Hình 4.7	Thiết kế chi tiết gói Bill	26
Hình 4.8	Thiết kế chi tiết gói Tree	27
Hình 4.9	Giao diện của phân hệ người dùng	28
Hình 4.10	Giao diện quản lý danh mục cây của quản trị viên	28
Hình 4.11	Giao diện quản lý hóa đơn của quản trị viên	29
Hình 4.12	Chi tiết lớp admin	29
Hình 4.13	Chi tiết lớp admincontroller	30
Hình 4.14	Chi tiết lớp adminservice	30
Hình 4.15	Chi tiết lớp Bill	31
Hình 4.16	Chi tiết lớp BillController	31
Hình 4.17	Chi tiết lớp Billdetail	32
Hình 4.18	Chi tiết lớp Billservice	32
Hình 4.19	Chi tiết lớp Tree	33
Hình 4.20	Chi tiết lớp TreeController	33
Hình 4.21	Chi tiết lớp TreeService	34
Hình 4.22	Chi tiết lớp User	34
Hình 4.23	Chi tiết lớp UserController	35
Hình 4.24	Chi tiết lớp UserService	35
Hình 4.25	Biểu đồ tuần tự thay đổi password với user	36

Hình 4.26	Sơ đồ thực thể liên kết	36
Hình 4.27	Chức năng tìm kiếm cho người dùng	39
Hình 4.28	Thêm vào giỏ hàng của người dùng	40
Hình 4.29	Đặt hàng cho người dùng	40
Hình 4.30	Admin tìm kiếm hóa đơn mong muốn	40
Hình 4.31	Thêm cây vào danh mục mong muốn của quản trị viên	41
Hình 4.32	Người dùng khởi tạo đơn hàng	41
Hình 4.33	Quản trị viên xem đơn hàng khởi tạo của ngày hôm nay	42
Hình 4.34	Xác nhận đơn và tạo người vận chuyển đơn	42
Hình 4.35	Xác nhận hoàn thành đơn và tổng kết	43
Hình 5.1	Nhược điểm mô hình MVC không hỗ trợ Preview như ASP.NET	45

DANH MỤC BẢNG BIỂU

Bảng 2.1	Mô tả Usecase tổng quát	5
Bảng 2.2	Mô tả chức năng lấy lại password	7
Bảng 2.3	Đặc tả usecase lọc tìm kiếm	8
Bảng 2.4	Mô tả một phiên chức năng đặt cây	12
Bảng 2.5	Đặc tả chức năng phân quyền	12
Bảng 2.6	Đặc tả chức năng quản lý nhân viên giao hàng	13
Bảng 4.1	Thiết kế bảng User	37
Bảng 4.2	Thiết kế bảng quản trị viên	37
Bảng 4.3	Thiết kế bảng tree	37
Bảng 4.4	Thiết kế bảng Bill	37
Bảng 4.5	Thiết kế bảng Nhân viên giao hàng	38
Bảng 4.6	Thiết kế bảng hóa đơn	38
Bảng 4.7	Bảng thư viện và các công nghệ sử dụng	38
Bảng 4.8	Bảng tổng hợp kết quả	39

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

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

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Xuất phát từ tình hình kinh tế ngày càng phát triển dẫn đến nhu cầu tiêu dùng và hưởng thụ cuộc sống của con người ngày càng gia tăng. Thuận theo nhu cầu cái đẹp trong cuộc sống dẫn đến nghề cây cảnh càng lúc càng phát triển, đặc biệt là cây cảnh bonsai. Người tiêu dùng có thể bỏ ra khoản tiền lớn để đầu tư cho thú chơi cây cảnh. Thế nhưng hiện nay rất ít trang web bán cây cảnh trên thị trường hoặc có nhưng chưa đáp ứng được nhu cầu tìm kiếm của người dân thông qua internet, đặc biệt là vào các dịp lễ đông người như dịp Tết. Nếu vấn đề này được giải quyết thì người tiêu dùng có thể tìm kiếm và tham khảo thị trường giá cây hiện nay trước khi ra ngoài mua hoặc có thể ngồi ở nhà mua và ship đến tận nơi mà không cần phải đi lại nhiều (điều này phù hợp với người lớn tuổi và người bận rộn), người bán cũng có thể quản lý được doanh thu và thống kê chi tiết mà không cần phải nhớ hay ghi lại. Ngoài cây cảnh bonsai website hoàn toàn có thể bán tất cả các loại cây cảnh khác tùy thuộc vào nhu cầu của người sử dụng cũng như người bán.

1.2 Mục tiêu và phạm vi đề tài

Qua tìm hiểu thị trường hiện nay thì nghề cây cảnh đang rất phát triển. Nhưng việc người tiêu dùng có thể tin tưởng để mua cây cảnh bonsai qua mạng là vô cùng ít. Các trang web cây cảnh hiện nay tồn tại rất nhiều nhưng tạm nham nhiều loại cây khác nhau và thiếu thẩm mỹ.

Xuất phát từ nhu cầu người sử dụng cũng như hạn chế tìm kiếm cây cảnh bonsai trên mạng toàn cầu hiện nay, em lập ra website này nhằm phục vụ nhu cầu tìm hiểu và khảo sát giá thành thị trường của các loại cây cảnh bonsai cho đại đa số người dân mà không cần phải ra ngoài thị trường-đặc biệt là dịp lễ đông người như Tết...Người dùng có thể mua và vận chuyển trực tiếp đến tận nhà mà không phải đi xa mà vẫn có thể xem chi tiết cây bằng hình ảnh trên website

1.3 Định hướng giải pháp

Từ phạm vi đề tài đã nêu ở mục trên em xin đề xuất hướng giải pháp như sau:

Xây dựng trang phân hệ bán hàng và phân hệ quản trị-kết nối dữ liệu với MySQL, cùng với đó là công nghệ mới nhất VueJs và Java SpringBoot

Với trang phân hệ bán hàng người dùng có thể xem và mua hàng. Đối với phân hệ quản trị, Admin có thể quản lý chi tiêu và thống kê báo cáo chi tiết. Cuối cùng hướng đến 1 trang website hoàn thiện phục vụ nhu cầu của người tiêu dùng và nhu cầu quản lý của người bán hay doanh nghiệp vừa và nhỏ

1.4 Bố cục đề án

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức như sau:

Chương 2 trình bày về khảo sát và phân tích yêu cầu của bài toán một cách chi tiết. Đưa ra các khảo sát hiện trạng và tổng quát cũng như đặc tả về chức năng

Trong Chương 3, em sẽ giới thiệu về công nghệ mình sử dụng đó là VueJs được viết bằng JavaSripts cho phần Font-end và phía Back-end là công nghệ Java-Springboot, những hạn chế cũng như ưu điểm của công nghệ đang sử dụng.

Chương 4 em sẽ đi vào chi tiết về thiết kế chi tiết, thiết kế giao diện người dùng cũng như thiết kế cơ sở dữ liệu để người dùng có thể hiểu rõ về cách thức hoạt động của website

Và Chương 5 em sẽ tổng kết và đánh giá cũng như hướng phát triển của website trong tương lai

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

Theo tình hình khảo sát hiện trạng thực tế hiện nay thì đa phần người tiêu dùng đều đến tận nơi để mua cây, xem trực tiếp tại cửa hàng uy tín hoặc người quen giới thiệu thay vì tìm kiếm trên mạng. Bên cạnh đó là hệ thống website bán cây cảnh hiện nay chưa phổ biến hoặc chưa tối ưu đến người dùng hoặc là người dùng chưa có hiểu biết về công nghệ dẫn đến số lượng người dùng website buôn bán trực tuyến còn hạn chế-đặc biệt là website cây cảnh. Dựa theo tình hình khảo sát trên em đã tạo ra website này với mong muốn kết nối cửa hàng với người tiêu dùng một cách chân thật nhất, cũng như người dùng có đánh giá 1 cách khách quan nhất khi mua hàng qua mạng. Ưu điểm của website là người dùng không cần đi lại nhiều tốn chi phí mà vẫn có thể xem chi tiết cũng như giá thành của từng loại cây mình mong muốn, tiếp đó là cửa hàng có thể thống kê chi tiết, thêm, sửa, xóa một cách dễ dàng, sẵn sàng tiếp cận người tiêu dùng. Ngoài những yêu điểm kể trên thì nhược điểm của trang web là chỉ xem chi tiết cây ở một góc độ nào đó phụ thuộc vào hình ảnh chủ cửa hàng đăng, tiếp theo cần có điện thoại hoặc thiết bị smartphone có kết nối internet để có thể truy cập

2.2 Tổng quan chức năng

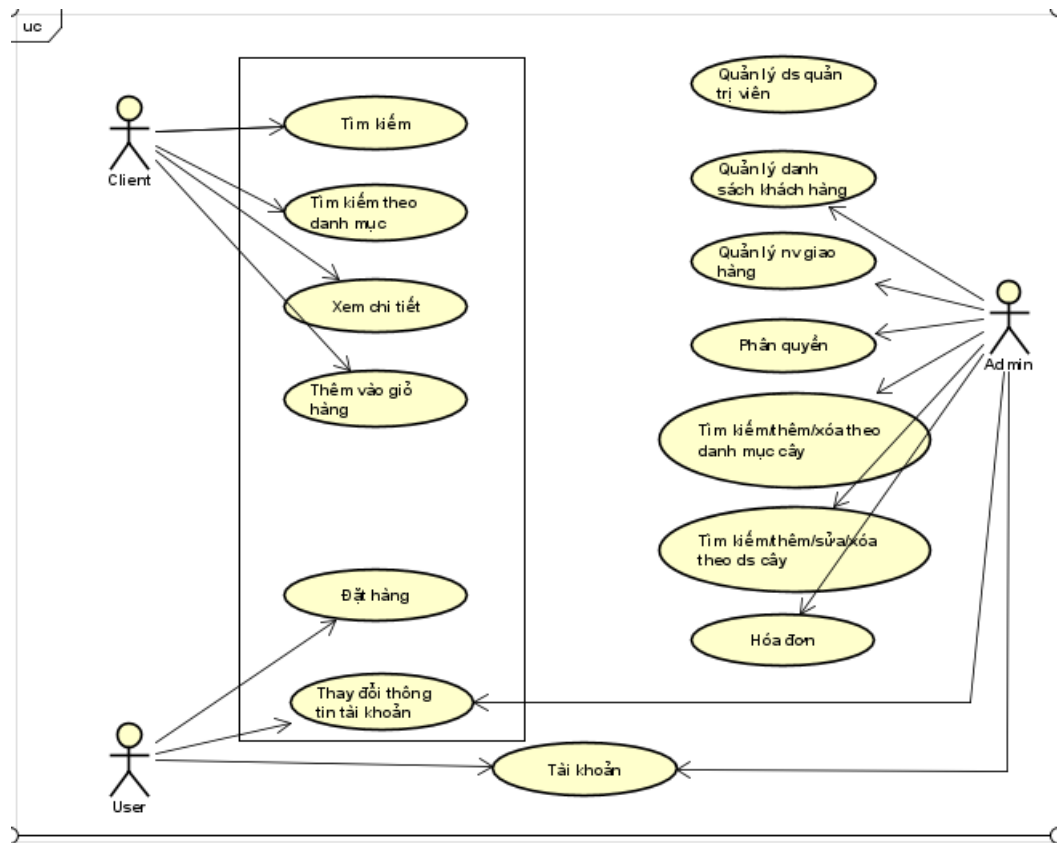
Tổng quát chức năng của phần mềm:

Đối với phân hệ bán hàng-cụ thể ở đây là người dùng: Có những chức năng cơ bản của 1 người dùng như tìm kiếm cây, tìm kiếm theo danh mục cây, tìm kiếm theo tên, xem chi tiết và mô tả của cây, xem giá thành cũng như đặt hàng đối với loại cây mong muốn, xem trạng thái mặt hàng đã mua cũng như giá thành phải thanh toán

Đối với phân hệ quản trị-cụ thể ở đây là Admin: Có những chức năng tổng quát như tìm cây theo mã, theo tên, thêm cây, thêm danh mục cây, thêm ảnh, quản trị viên, quản trị người dùng, quản lý người vận chuyển, quản lý danh sách cây, tổng kết, báo cáo và thống kê qua từng thời gian mong muốn

2.2.1 Biểu đồ use case tổng quát

Dưới đây là biểu đồ Usecase tổng quát!



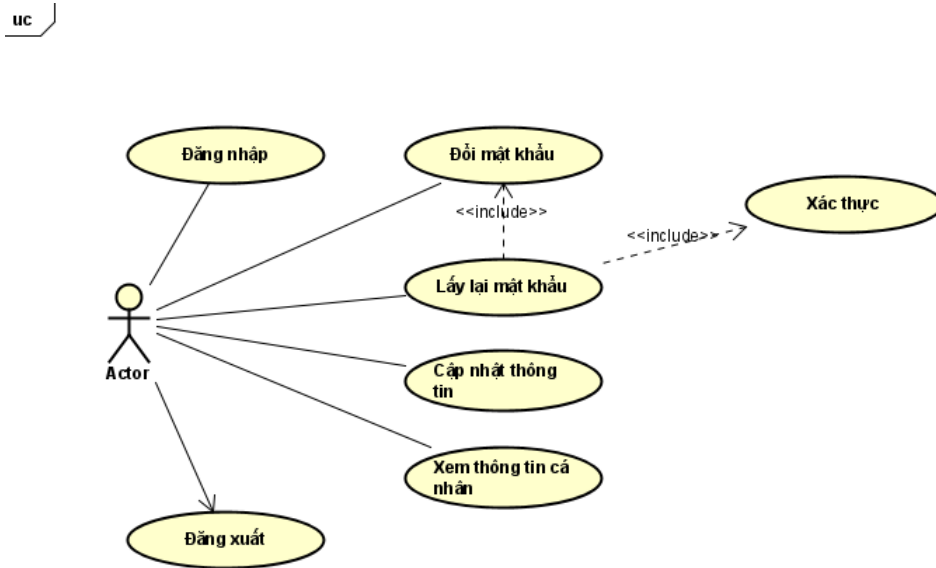
Hình 2.1: Usecase tổng quát

Từ Biểu đồ Usecase tổng quan em sẽ mô tả chi tiết chức năng của từng usecase

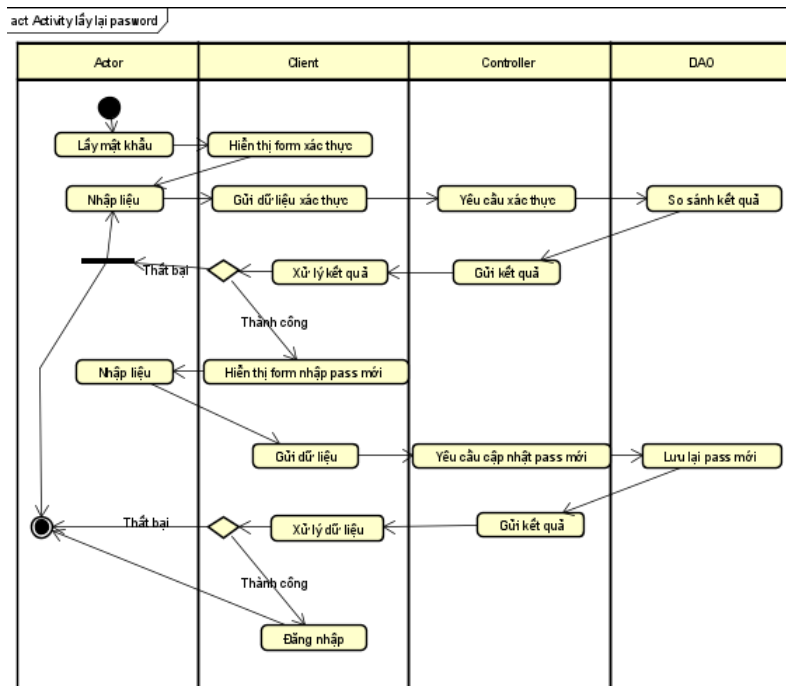
Bảng 2.1: Mô tả Usecase tổng quát

Chức năng	Các tác nhân	Mô tả ngắn gọn
Tài khoản	User, Admin	Là nhóm các chức năng liên quan tới tài khoản như đăng nhập đăng xuất, lấy lại mật khẩu, hồ sơ cá nhân
Thay đổi thông tin tài khoản	User, Admin	Là các thông tin liên quan đến thay đổi số dt, địa chỉ email
Đặt hàng	User, Guest	Là chức năng dùng để đặt hàng cây cảnh
Tìm kiếm	User, Guest	Là chức năng tìm kiếm cây theo tên mong muốn
Tìm kiếm theo danh mục	User, Guest	Là chức năng tìm kiếm theo danh mục cây
Xem chi tiết	User, Guest	Là chức năng xem chi tiết cây và mô tả của cây đó
Thêm vào giỏ hàng	User, Guest	Là chức năng thêm cây muốn mua vào giỏ hàng
Quản lý ds quản trị viên	Admin	Là chức năng quản lý danh sách các quản trị viên của hệ thống
Quản lý ds khách hàng	Admin	Là chức năng quản lý User
Quản lý nv giao hàng	Admin	Là chức năng quản lý người vận chuyển của cửa hàng
Phân quyền	Admin	Là chức năng phân quyền cho quản trị viên
Tìm kiếm/sửa/xóa danh mục cây	Admin	Là chức năng tìm kiếm/sửa/xóa theo danh mục cây
Tìm kiếm/sửa/xóa theo ds cây	Admin	Là chức năng tìm kiếm/sửa/xóa theo danh sách cây
Hóa đơn	Admin	Là chức năng kiểm tra mọi thông tin theo hóa đơn

2.2.2 Biểu đồ use case quản lý tài khoản



Hình 2.2: Usecase quản lý tài khoản

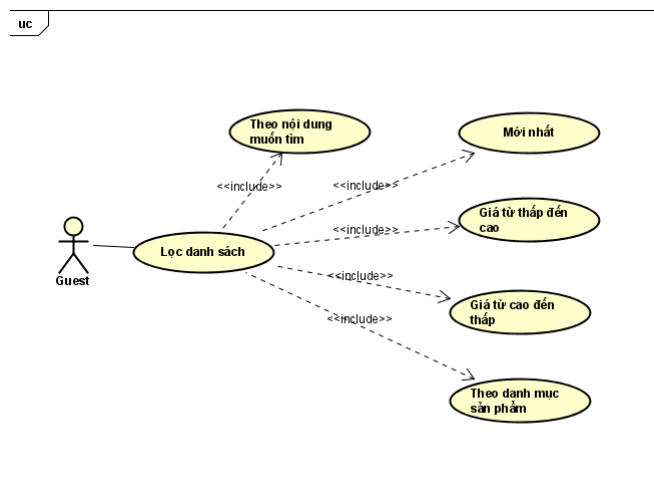


Hình 2.3: Biểu đồ hoạt động lấy lại password

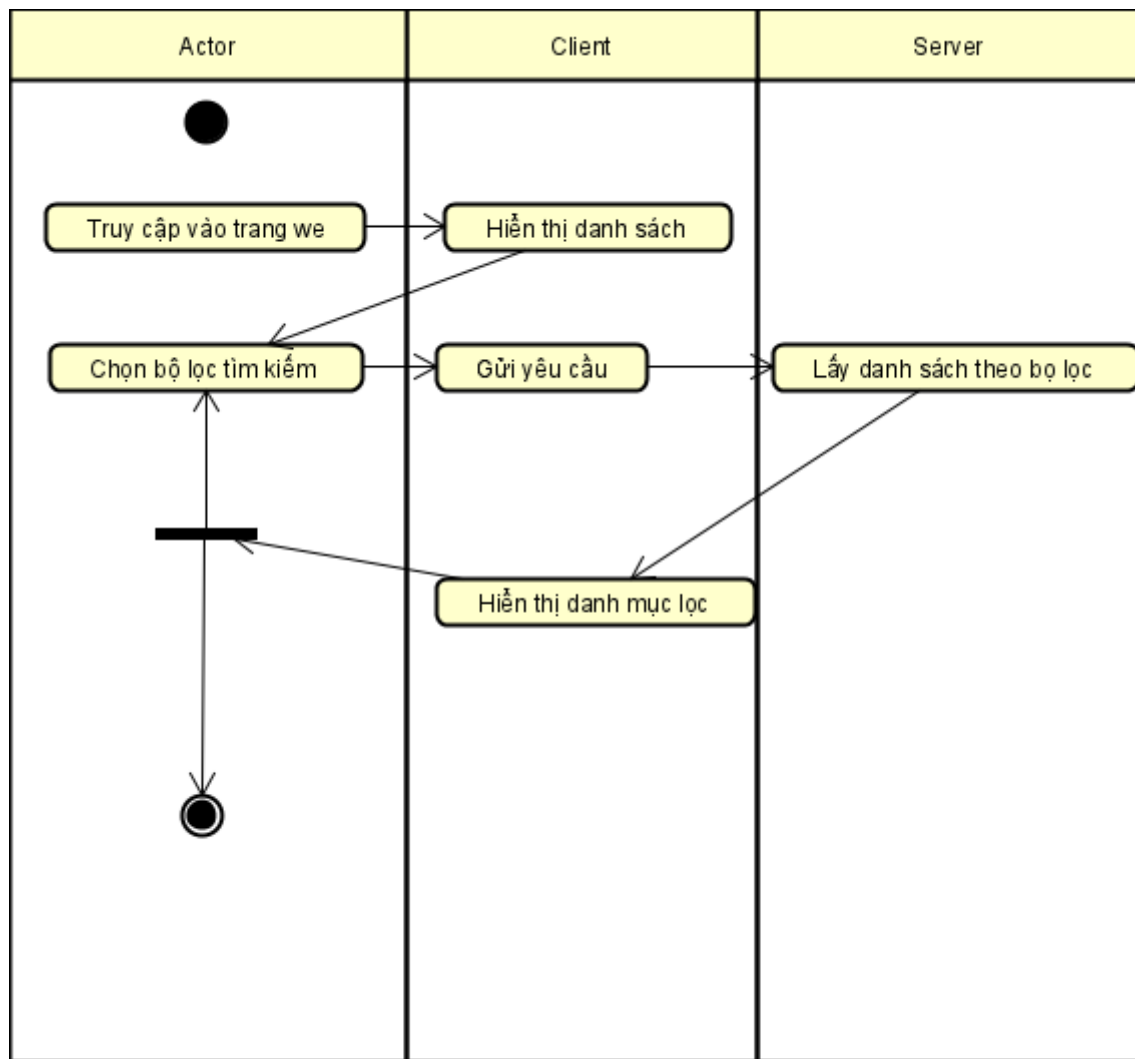
Bảng 2.2: Mô tả chức năng lấy lại password

Tên chức năng	Lấy lại mật khẩu
Tác nhân	User
Mô tả	Các tác nhân quên mật khẩu hoặc tài khoản cần lấy lại tài khoản để đăng nhập
Luồng sự kiện	1-Tác nhân chọn mục quên mật khẩu trên giao diện 2-Client hiện form nhập thông tin xác thực tài khoản 3-Tác nhân nhập và yêu cầu client gửi đi dữ liệu xác thực 4- Controller của server nhận yêu cầu xác thực và yêu cầu DAO xác minh 5-DAO trả lại kết quả xác minh cho controller để gửi kết quả cho Client 6-Client xử lý kết quả 6.1 Nếu xác minh sai quay về B2 6.2 Nếu đúng client hiện thì form nhập pass mới 7-Tác nhân nhập mật khẩu mới 8-Client xử lý kết quả nhận từ server
Ngoại lệ	Nếu bước 6.1 khi quay về B2 tác nhân có thể thoát luôn

2.2.3 Biểu đồ usecase phân giả chức năng tìm kiếm



Hình 2.4: Usecase phân giả chức năng tìm kiếm

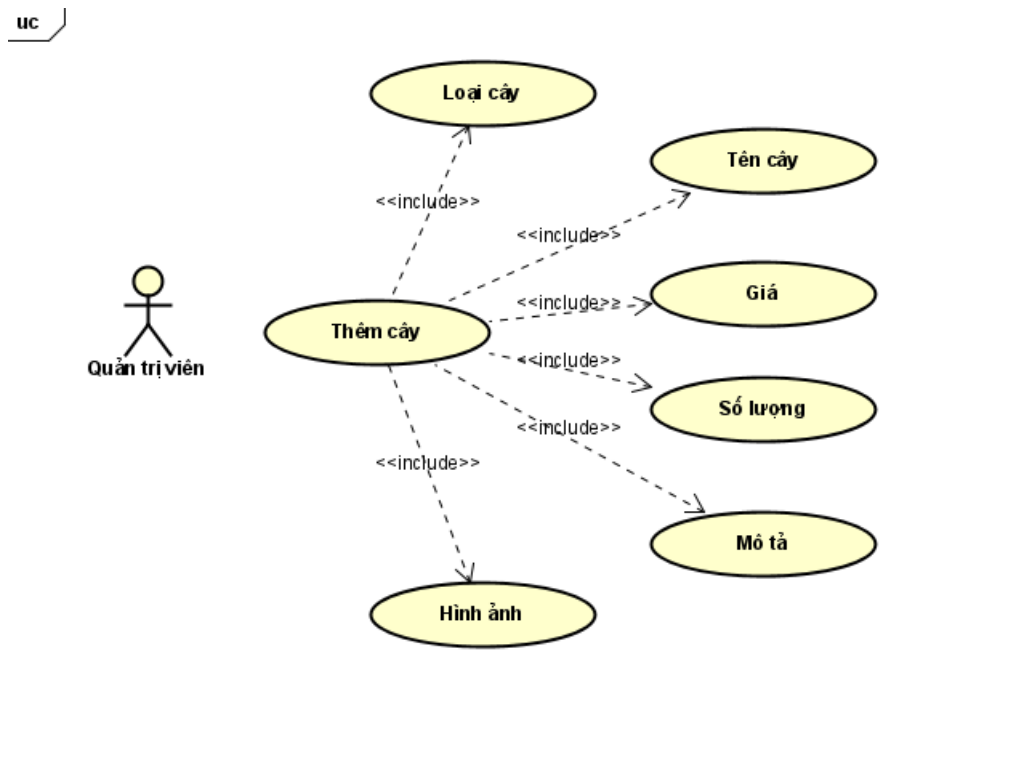


Hình 2.5: Biểu đồ hoạt động chức năng tìm kiếm

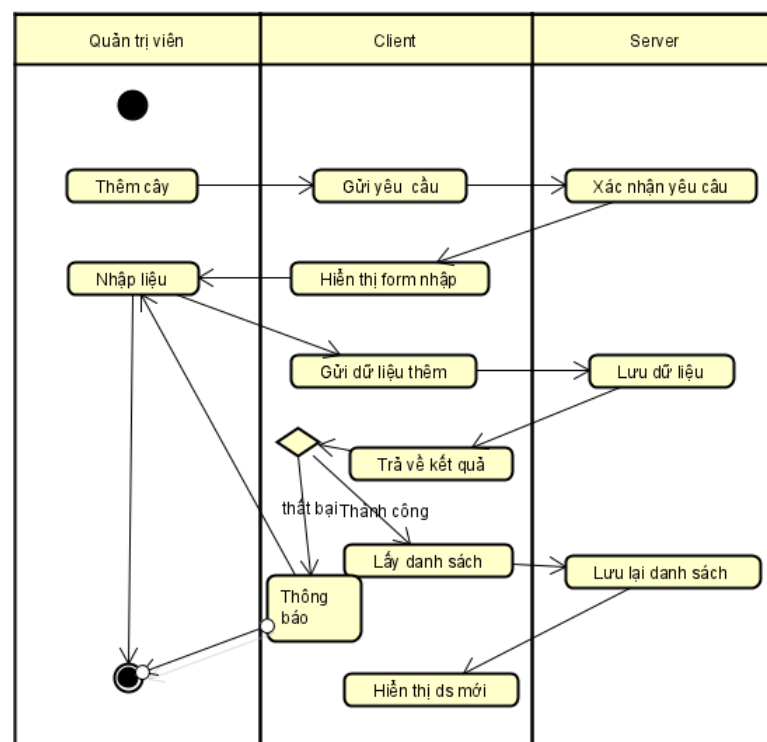
Bảng 2.3: Đặc tả usecase lọc tìm kiếm

Tên chức năng	Lọc tìm kiếm
Luồng xử lý	1-Truy cập vào trang web 2-Client hiển thị danh trang chủ website 3-Tác nhân chọn lọc tìm kiếm theo mong muốn 4-Client gửi yêu cầu xử lý 5- Server trả về kết quả cho Client 6-Client hiển thị danh sách đã lọc cho người dùng 7-Quay lại bước 3 hoặc thoát ra ngoài

2.2.4 Tổng quan chức năng thêm cây của quản trị viên



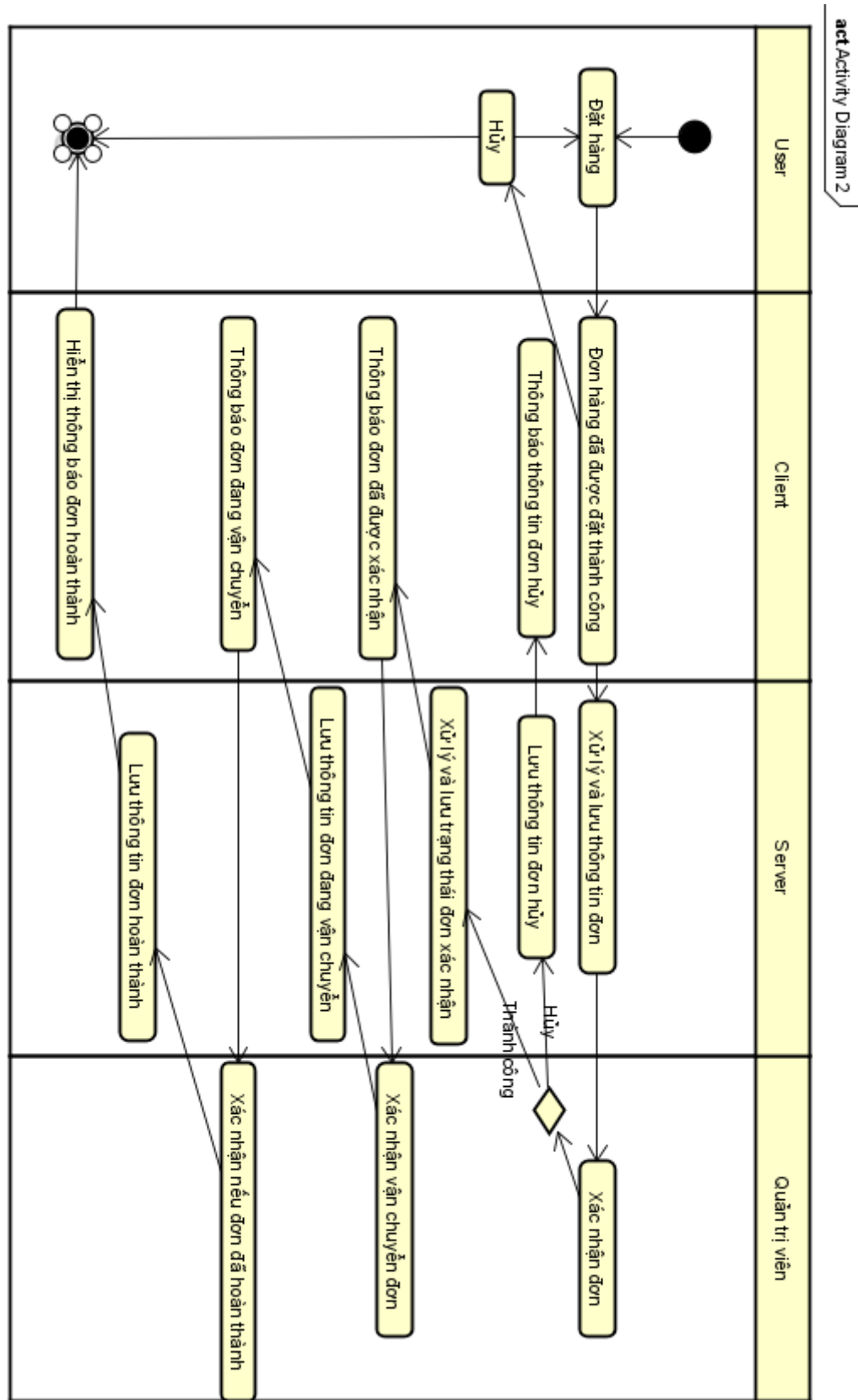
Hình 2.6: Usecase thêm cây của quản trị viên



Hình 2.7: Biểu đồ hoạt động thêm cây

Tên chức năng	Thêm cây
Tác nhân	Quản trị viên
Luồng xử lý	<p>1-Thêm cây</p> <p>2-Client thông báo và gửi về server</p> <p>3-Server nhận thông tin và trả về cho Client form nhập thêm cây</p> <p>4-Tác nhân nhập liệu thêm cây theo từng mục</p> <p>5-Gửi thông báo thêm cho Client</p> <p>6-Client gửi yêu cầu cho server</p> <p>7-Server trả kết quả cho Client</p> <p>7.1 Nếu thành công cây sẽ được lưu lại và thông báo cho Client</p> <p>7.2 Nếu thất bại sẽ có thông báo nhập lại ở B4</p> <p>8. Kết thúc</p>

2.2.5 Quy trình nghiệp vụ



Hình 2.8: Biểu đồ hoạt động quy trình nghiệp vụ đặt hàng

Bảng 2.4: Mô tả một phiên chức năng đặt cây

Tên chức năng	Đặt hàng
Tác nhận	User,Client,Server,Admin
Luồng xử lý	1-Người dùng đặt hàng 2-Client hiển thị thông báo đặt hàng thành công 3-Người dùng có thể hủy đơn trước khi quản trị viên xác nhận 4-Hệ thống lưu thông tin đơn hàng đã đặt 5-Quản trị viên xác nhận đơn hàng hoặc hủy đơn hàng đó 6-Hệ thống lưu thông tin đơn xác nhận và gửi thông báo cho Client 7-Client hiển thị thông báo đơn hàng đã được xác nhận 8-Quản trị viên xác nhận đơn hàng đã được vận chuyển 9-Hệ thống lưu thông tin đơn và trả về cho Client 10-Client thông báo tới User đơn đã được vận chuyển 11-Quản trị thay đổi đơn hàng thành hoàn thành sau khi đơn giao đến người dùng 12-Server lưu thông tin đơn hoàn thành trả về Client 13-Client thông báo cho User đơn đã hoàn thành 14-Kết thúc phiên đặt hàng

2.3 Đặc tả chức năng

Sau đây em xin đặc tả về các chức năng của một số usecase quan trọng

2.3.1 Đặc tả use case phân quyền

Bảng 2.5: Đặc tả chức năng phân quyền

Tên chức năng	Phân quyền
Tác nhận	Admin
Luồng xử lý	1-Quản trị viên chọn mục vai trò hệ thống và thêm 2-Điền tên đầy đủ và phần mô tả 3-Chọn hành động với các danh mục mong muốn gồm có(tạo, xóa, update, xem) 4-Lưu lại thông tin 5-Server xác nhận và thông với Quản trị viên 6-Vào phần Quản trị viên và chọn người mong muốn 7-Ất quyền mong muốn với quản trị viên 8-Kết thúc phiên

2.3.2 Đặc tả use case quản lý nhân viên giao hàng

Bảng 2.6: Đặc tả chức năng quản lý nhân viên giao hàng

Tên chức năng	Quản lý nhân viên giao hàng
Tác nhân	Admin
Luồng xử lý	1-Chọn nhân viên giao hàng 2-Thêm nhân viên giao hàng 3-Điền thông tin nhân viên theo form mẫu 4-Nhấn lưu và gửi yêu cầu tới server 5-Server nhận thông tin và thông báo kết quả 6-Xem lại thông tin nhân viên giao hàng 7-Kết thúc
Ngoại lệ	Có thể tìm kiếm /sửa/xóa và đặt trạng thái của nhân viên giao hàng

2.4 Yêu cầu phi chức năng

Để dễ dàng sử dụng hệ thống một cách hiệu quả thì nên dùng CSDL đơn giản để quản lý như MySQL và các công nghệ phổ biến hiện nay. Cùng với đó hệ thống hoạt động tốt trên nền tảng Brower: GG Chorme, CocCoc..chưa hoạt động hiệu quả trên các thiết bị di động thông minh

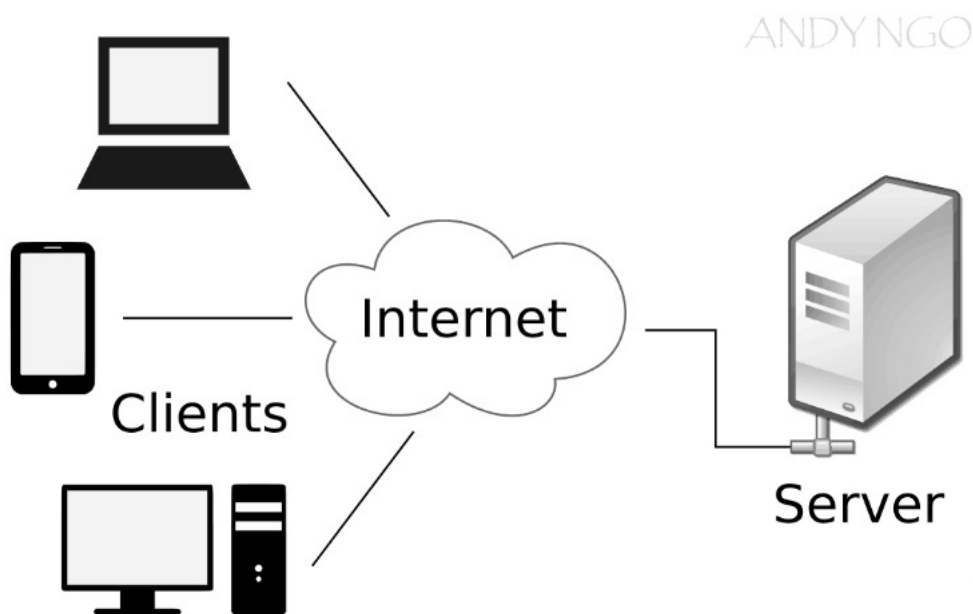
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 Tìm hiểu và giới thiệu tổng quan về công nghệ

Web là một nền tảng được khai thác rất nhiều để tạo nên các sản phẩm phục vụ các nhu cầu trong cuộc sống. Vì thế các công nghệ đi kèm với nó ngày càng được phát triển không ngừng. Rất nhiều ngôn ngữ, framework ra đời để làm cho việc lập trình web ngày càng đơn giản, hiệu quả. Giao diện được xây dựng trên web cũng ngày được cải thiện, đẹp hơn, dễ dùng và dễ thao tác hơn. Vì vậy em chọn web là nền tảng để phát triển sản phẩm. Để tạo dựng phát triển sản phẩm em lựa chọn: ngôn ngữ lập trình client javascript (js) với framework vuejs, ngôn ngữ lập trình server java với framework java spring boot, hệ quản trị cơ sở dữ liệu mysql.

3.2 Mô hình phát triển của công nghệ

Mô hình được lựa chọn để phát triển sản phẩm là mô hình client – server với kiến trúc đơn giản như hình.



Hình 3.1: Mô hình Client-Server

Hiện tại mô hình này vẫn đang được sử dụng với nhiều kiến trúc khác nhau và kiến trúc tốt nhất đối với mô hình này có lẽ là kiến trúc microservice. Nhưng với khả năng không cho phép nên em phát triển sản phẩm với mô hình client – server với kiến trúc nguyên khối (monolithic). Trong mô hình này máy chủ (server) là nơi lưu trữ tài nguyên, tạo ra các chương trình dịch vụ thực hiện các yêu cầu của client.

Một số những ưu điểm mà mô hình này mang lại là:

- Có khả năng chống quá tải mạng
- Đảm bảo toàn vẹn dữ liệu khi có sự cố xảy ra
- Dễ dàng mở rộng hệ thống
- Chỉ cần chung định dạng giao tiếp là có thể hoạt động không cần chung nền tảng
- Cho phép tích hợp các công nghệ kỹ thuật hiện đại
- Cho phép truy cập tài nguyên từ xa

Trong đồ án này máy chủ (server) và máy khách (client) sẽ giao tiếp với nhau theo các api.

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

Các API này dần dần được quy chuẩn để dễ dàng, thuận lợi khi sử dụng trong phát triển các sản phẩm không riêng các sản phẩm xây dựng trên nền tảng web. Và tiêu chuẩn đang dùng phổ biến nhất là RestFul API (REpresentational State Transfer). Và em lựa chọn java spring boot làm framework dùng để tạo ra các restful api phía server.

3.3 Công nghệ Font-end/Client

3.3.1 Giới thiệu ngôn ngữ JavaScripts

JavaScript là một ngôn ngữ lập trình website, được tích hợp và nhúng trong HTML giúp website sống động hơn. JavaScript cho phép kiểm soát các hành vi của trang web tốt hơn so với khi chỉ sử dụng mỗi HTML. Vậy ứng dụng thực tiễn của JavaScript là gì? Các slideshow, pop-up quảng cáo và tính năng autocompleate của Google là những ví dụ dễ thấy nhất cho bạn, chúng đều được viết bằng JavaScript JavaScript có thể được ứng dụng trong rất nhiều lĩnh vực:

- Lập trình website.
- Xây dựng ứng dụng cho website máy chủ.
- Ứng dụng di động, app, trò chơi. ...

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên

Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

3.3.2 Framework VueJs

Vue đã được tạo ra và phát triển bởi Evan You khi ông đang làm kỹ sư tại công ty Google. Lúc Evan đang sử dụng Angular cho một số dự án khác thì đã có phát biểu rằng: “Điều gì sẽ xảy ra như trích một phần mà mình thích về Angular rồi xây dựng thêm một cái gì đó nhẹ nhàng hơn mà không cần phải chú ý đến những khái niệm bổ sung”. Đây là phát biểu được xem là bước đệm giúp cho Vue đã và đang có mặt tại thời điểm hiện tại. Vue đã được phát hành vào đầu năm 2014 và dự án này đã được tăng tải lên HackerNew, Echo Js ngay trong ngày đầu tiên ra mắt. [5] Element là một trong những thư viện UI dành cho vuejs.

3.4 Công nghệ Backend/Server

3.4.1 Ngôn ngữ lập trình Java

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

Java (phiên âm Tiếng Việt: "Gia-va") là một ngôn ngữ lập trình hướng đối tượng, dựa trên lớp được thiết kế để có càng ít phụ thuộc thực thi càng tốt. Nó là ngôn ngữ lập trình có mục đích chung cho phép các nhà phát triển ứng dụng viết một lần, chạy ở mọi nơi (WORA), nghĩa là mã Java đã biên dịch có thể chạy trên tất cả các nền tảng hỗ trợ Java mà không cần biên dịch lại. Các ứng dụng Java thường được biên dịch thành bytecode có thể chạy trên bất kỳ máy ảo Java (JVM) nào bất kể kiến trúc máy tính bên dưới. Cú pháp của Java tương tự như C và C++, nhưng có ít cơ sở cấp thấp hơn các ngôn ngữ trên. Java runtime cung cấp các khả năng động (chẳng hạn như phản ánh và sửa đổi mã thời gian chạy) thường không có sẵn trong các ngôn ngữ biên dịch truyền thống. Tính đến năm 2019 Java là một trong những ngôn ngữ lập trình phổ biến nhất được sử dụng theo GitHub, đặc biệt cho các ứng dụng web máy khách-máy chủ, với 9 triệu nhà phát triển đã được báo cáo.

3.4.2 FrameWork Java Spring boot

Spring Boot là một dự án phát triển bởi JAV (ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.

Spring boot là một phần quan trọng của spring framework. Spring framework được ra đời để giúp các nhà phát triển có thể xây dựng hệ thống và chạy ứng dụng trên JVM một cách thuận tiện, đơn giản và nhanh chóng. Đây là một mã nguồn mở được phát triển và rất nhiều người sử dụng. Trên thực tế, Spring framework là tập hợp gồm rất nhiều các dự án nhỏ khác nhau như: Spring MVC (sử dụng để xây dựng các ứng dụng trên nền tảng web), Spring Data, Spring Boot.

SpringBoot là một framework tỏ ra những lợi thế vượt trội. Khi sử dụng Spring Boot, rất nhiều thứ được cải tiến hỗ trợ lập trình viên như:

- AutoConfig: tự động cấu hình
- Dựa trên các Annotation để tạo ra các bean thay vì XML
- Server Tomcat có thể được nhúng ngay trong file JAR build ra và có thể chạy ở bất kì đâu mà java chạy được.

Việc ứng dụng Spring Boot giúp các lập trình viên tập trung tốt hơn đến mục tiêu viết code của mình thay vì phải lo nghĩ nhiều đến các kỹ thuật code như với Node.js.

Bản thân Spring Boot chứa đựng nhiều các library bên trong, cấu trúc code cũng trở nên chuẩn mực hơn, bạn không cần phải quá bận tâm đến việc code làm sao cho thật tốt. Do đó, bạn có nhiều thời gian hơn để tập trung vào sự logic của sản phẩm.

Spring Boot là sự cải tiến tuyệt vời của Spring, cải thiện được rất nhiều khâu rườm rà đến từ Spring. Hơn thế, quá trình học hỏi và tiếp thu Spring Boot cũng trở nên dễ dàng hơn rất nhiều.

Một số ưu điểm nổi bật:

- Phát triển web một cách đơn giản và nhanh chóng
- Config an toàn
- Có thể hỗ trợ YAML (YAML Ain't Markup Language)
- Có tính quản trị cao
- Dễ dàng ứng dụng Spring và các sự kiện
- Có thể cấu hình ở bên ngoài và tạo ra các tệp thuộc tính

- Tính bảo mật cao
- Có ghi log

Một số đặc điểm nổi bật:

- SpringApplication: Đây là class được dùng để khởi chạy ứng dụng từ hàm main(). Vì thế, khi khởi chạy ứng dụng, các lập trình viên chỉ cần gọi method run() là được.
- Externalized Configuration: Spring Boot cho phép người dùng có thể sử dụng cấu hình (config) từ bên ngoài. Chính vì vậy mà mỗi ứng dụng được tạo ra có thể chạy được trên nhiều môi trường khác nhau.
- Profiles: Dùng để phân chia các loại cấu hình cho các môi trường khác nhau.
- Loggin: Sử dụng phục vụ cho toàn bộ chức năng log trong phạm vi nội bộ. Những logging này sẽ được quản lý một cách mặc định

3.4.3 Hệ quản trị cơ sở dữ liệu

Trong dự án lần này em chọn hệ quản trị CSDL là MySQL

Một hệ quản trị cơ sở dữ liệu (tiếng Anh: Database Management System, viết tắt DBMS) là một chương trình máy tính (một bộ các chương trình) được thiết kế để quản lý một cơ sở dữ liệu, một tập hợp dữ liệu lớn có cấu trúc, phục vụ cho các yêu cầu về dữ liệu của một số lượng lớn người sử dụng

Hệ quản trị cơ sở dữ liệu mysql là một hệ quản trị cơ sở dữ liệu mã nguồn mở. Mysql có một số ưu điểm sau:

- Khả năng mở rộng và linh hoạt
- Hiệu năng cao
- Tính sẵn sàng cao
- Hỗ trợ giao dịch mạnh mẽ
- Bảo vệ dữ liệu mạnh mẽ
- Phát triển ứng dụng toàn diện
- Quản lý dễ dàng
- Chi phí sở hữu thấp
- Là mã nguồn mở

3.5 Công cụ sử dụng

Trong quá trình nghiên cứu, xây dựng và phát triển, các công cụ được sử dụng:

- Công cụ quản lý mã nguồn: git
- Công cụ lập trình client: vscode
- Công cụ lập trình server: IntelliJ
- Công cụ vẽ biểu đồ: AstahUML
- Công cụ test api: postman
- Công cụ quản lý cơ sở dữ liệu: mysql workbench

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1 Thiết kế kiến trúc

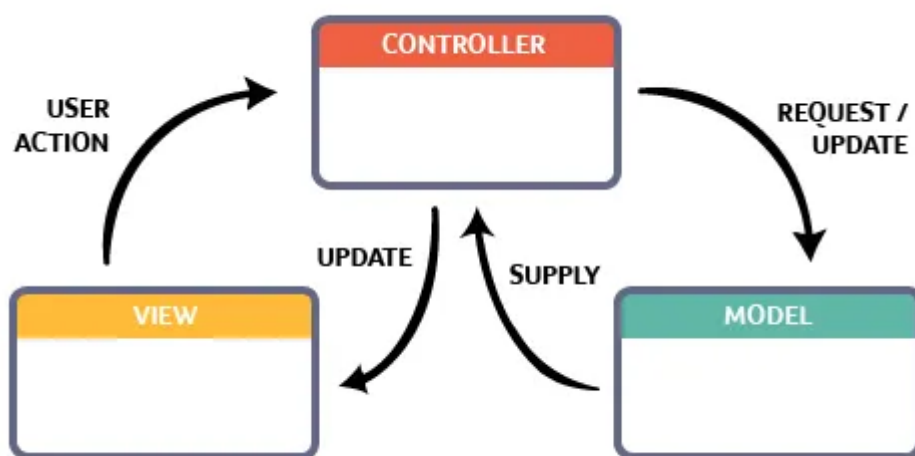
4.1.1 Mô hình MVC

MVC là viết tắt của cụm từ “Model-View-Controller“. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

Model (dữ liệu): Quản lý xử lý các dữ liệu.

View (giao diện): Nơi hiển thị dữ liệu cho người dùng.

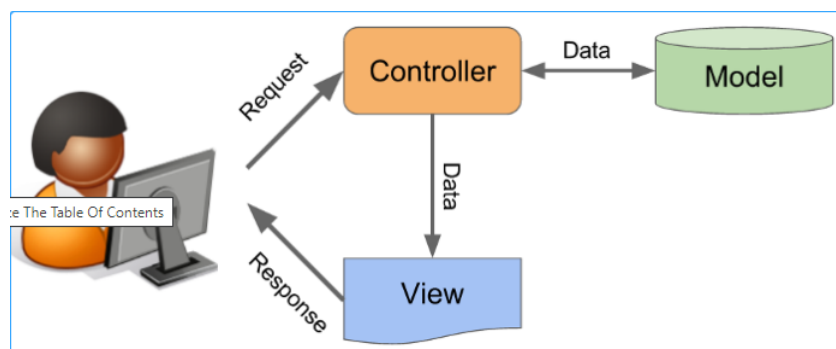
Controller (bộ điều khiển): Điều khiển sự tương tác của hai thành phần Model và View.



Hình 4.1: Mô hình MVC

Luồng xử lý trong MVC:

- Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
- Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
- Model chuẩn bị data và gửi lại cho Controller.
- Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.



Hình 4.2: Quy trình xử lý của MVC

Ở đây, View không giao tiếp trực tiếp với Model. Sự tương tác giữa View và Model sẽ chỉ được xử lý bởi Controller.

Trong kiến trúc thiết kế của "website cây cảnh bonsai" của em thì:

Model (dữ liệu):

Nhiệm vụ chính của thành phần này chỉ đơn giản là quản lý dữ liệu. Model sẽ chịu trách nhiệm quản lý dữ liệu từ cơ sở dữ liệu, API hay JSON. Trong ứng dụng, Model chứa các đối tượng (cây) với tất cả thông tin (dữ liệu) cần thiết cho ứng dụng.

View (Giao diện người dùng):

Nhiệm vụ của **View** là quyết định xem người dùng sẽ nhìn thấy gì trên màn hình của họ?

Trong Ứng dụng thì phần **Tree, TreeDetail, Cart, Bill, BillDetail, User** là những thành phần mà người dùng có thể xem. Các chức năng đều xác định những gì mỗi chế độ xem muốn khởi tạo và hiển thị. Chức năng này sẽ quyết định câu hỏi là người dùng sẽ nhìn gì và như thế nào trên màn hình.

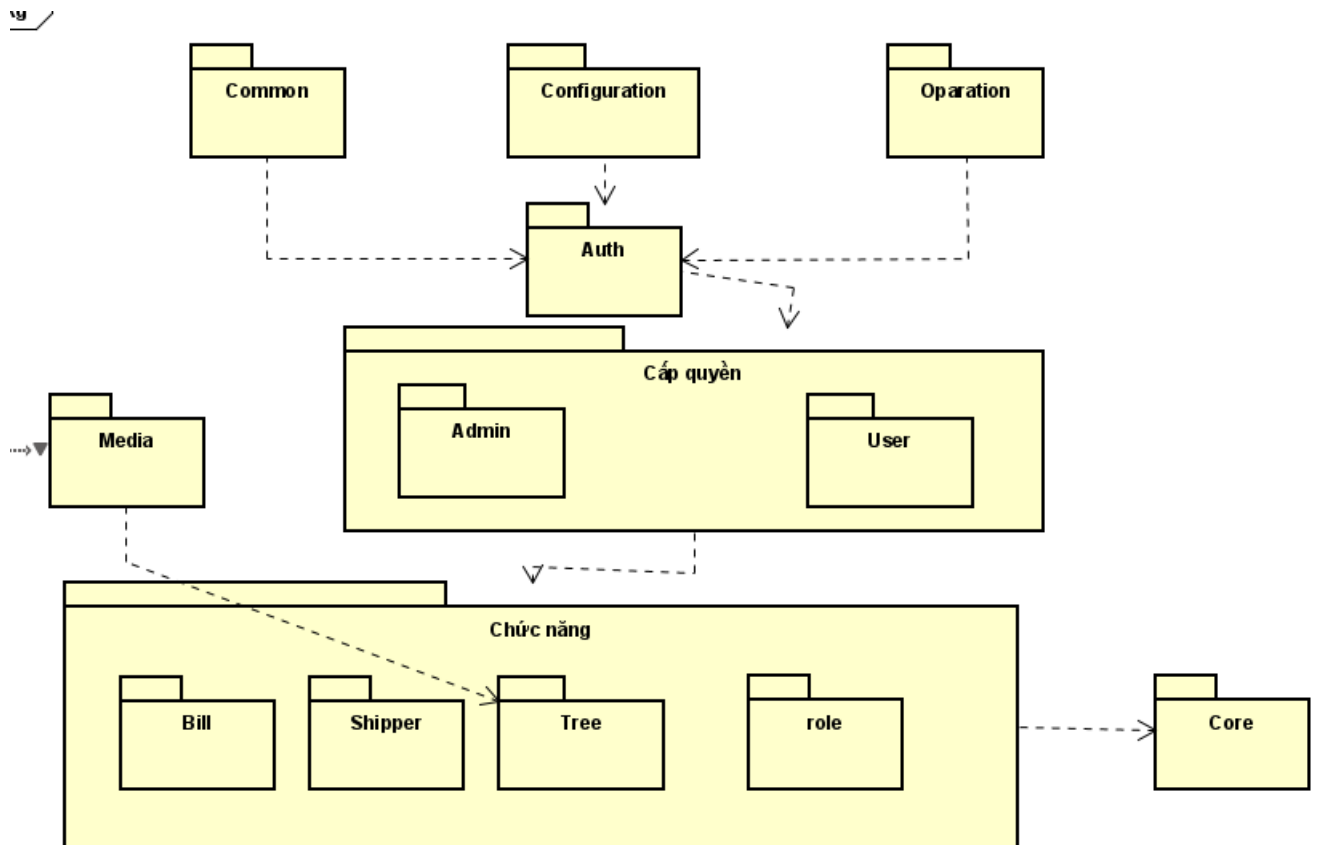
Controller:

Chức năng của controller là lấy, sửa đổi và cung cấp dữ liệu cho người dùng. Về cơ bản, controller là liên kết giữa View và Model.

Thông qua các hàm **search, add, update, delete, getter** được khởi tạo trong các cấu trúc phần mềm mà **Controller** lấy dữ liệu từ model và khởi tạo view.

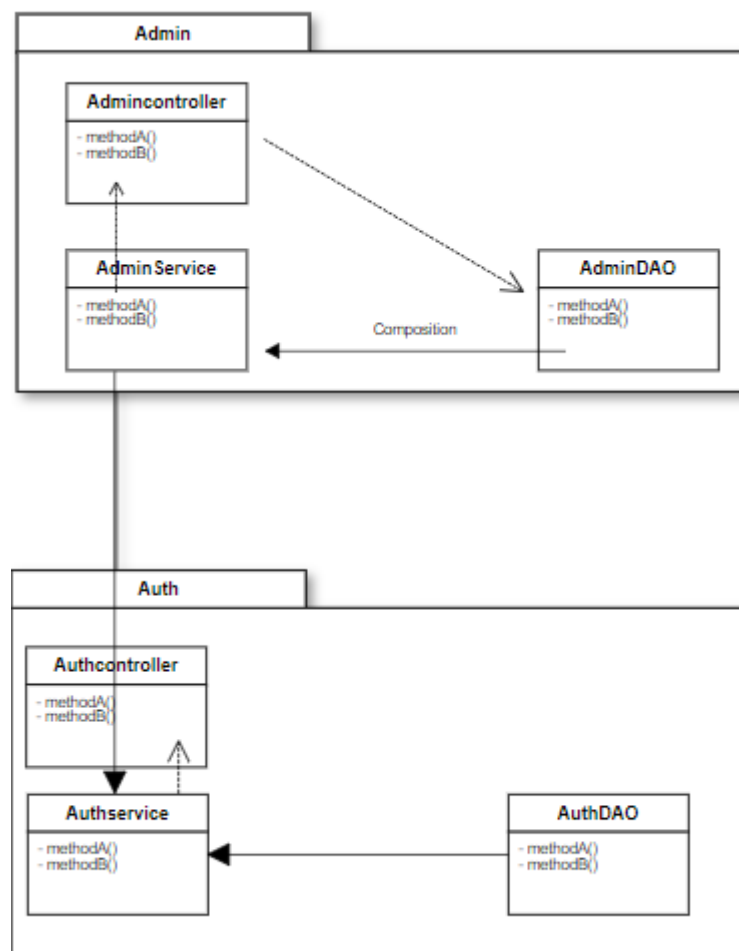
Phần ưu và nhược điểm của mô hình MVC em xin trình bày trong 5.1 và lý do vì sao chúng ta nên dùng MVC trong phần 5.2

4.1.2 Thiết kế tổng quan



Hình 4.3: Biểu đồ phụ thuộc gói

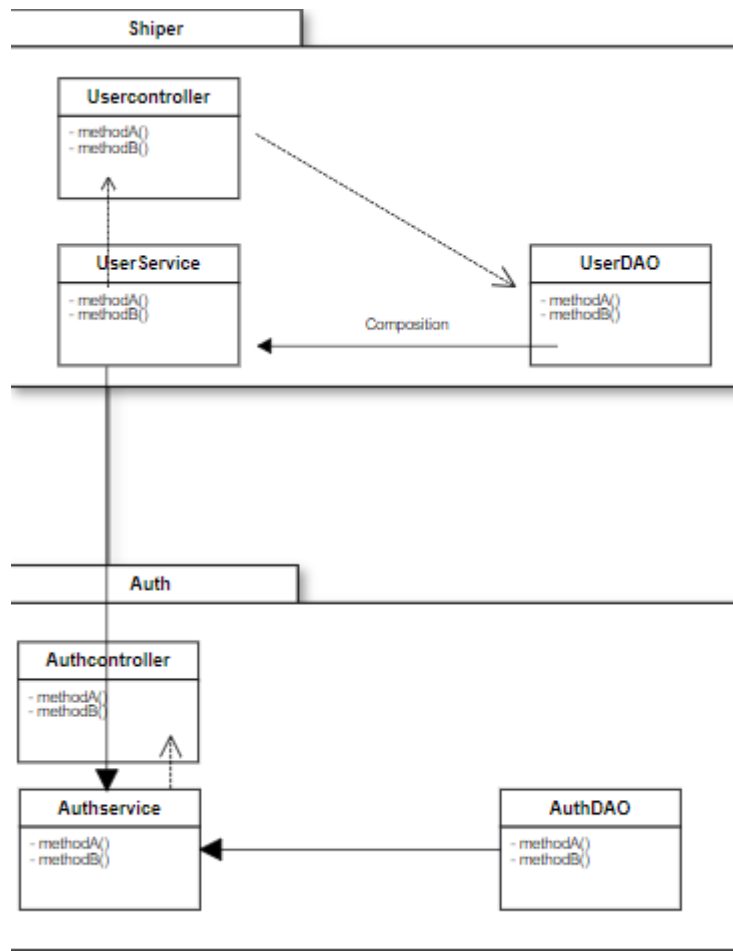
4.1.3 Thiết kế chi tiết gói



Hình 4.4: Thiết kế chi tiết gói Admin

Gói **Admin** gồm 3 class là: **Admincontroller**, **adminservice** , **adminDAO**

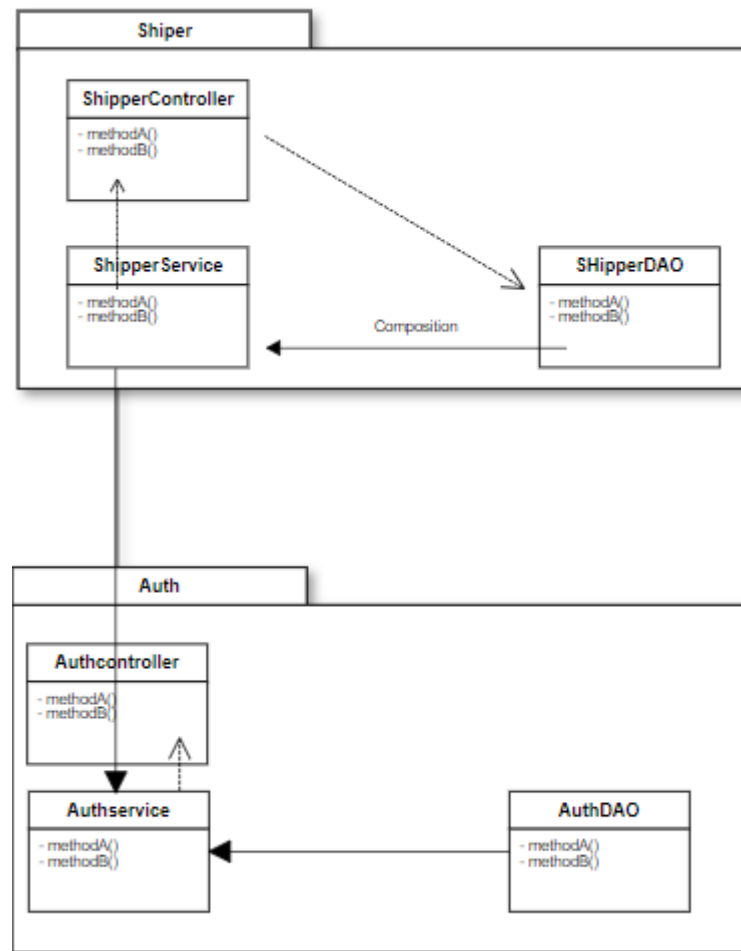
Để có thể hoạt động được, package **Admin** cần có sự kiểm soát của package **Auth** nơi mà lớp **AuthService** và **AuthDAO** hoạt động



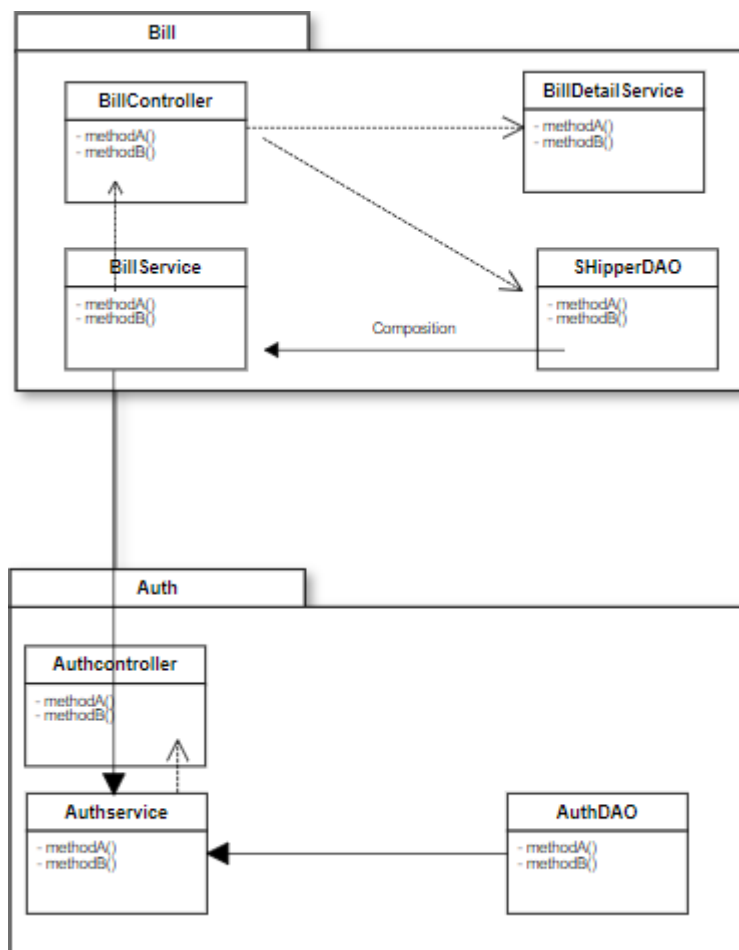
Hình 4.5: Thiết kế chi tiết gói User

Gói User gồm 3 class là: **Usercontroller**, **Userservice** , **UserDAO**

Để có thể hoạt động được, package **User** cần có sự kiểm soát của package **Auth** nơi mà lớp **AuthService** và **AuthDAO** hoạt động



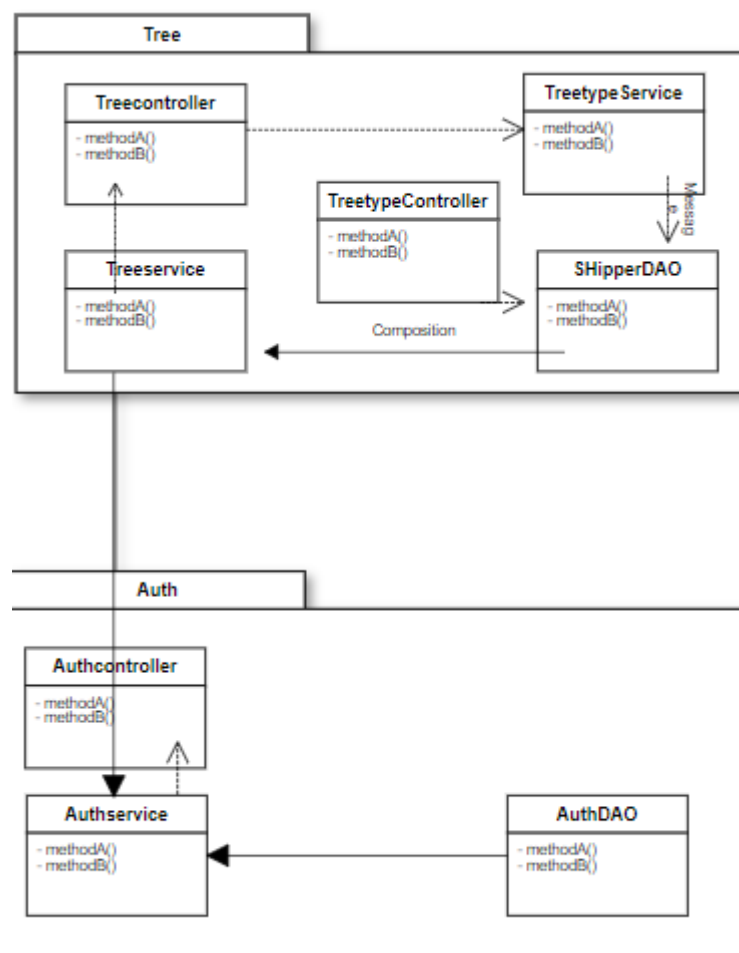
Hình 4.6: Thiết kế chi tiết gói Shipper



Hình 4.7: Thiết kế chi tiết gói Bill

Package **Bill** gồm 4 class: **Billcontroller**, **billservice**, **billdetailService** và **Bill-DAO**

Cũng như các gói ở trên thì package **Bill** cũng hoạt động dựa trên sự xác thực của package **Auth** và class **Authservice**



Hình 4.8: Thiết kế chi tiết gói Tree

Package **tree** gồm 5 class: **Treecontroller**, **Treeservice**, **TreedetailService**, **Treedetailcontroller** và **TreeDAO**

Cũng như package **Bill** thì package **Tree** cũng hoạt động dựa trên sự xác thực của package **Auth** và class **Authservice**

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Giao diện của em được thiết kế trên màn hình Asus X550LN với 15'6ích và độ phân giải màn hình là 1366x768 với tông màu hỗ trợ chủ đạo là màu xanh lá cây và màu trắng. Các chữ, nút tìm kiếm và tất cả đều được thiết kế chuẩn như các website khác và màu là màu đen trên nền trắng. Sau đây là một số giao diện website của người dùng và của phân hệ quản trị và một số chức năng quan trọng đi cùng

LOGO

CÂY CẢNH BONSAI

hotline
SDT:...

admin

Logout

Dashboard

Quản trị viên

Khách hàng

Nhân viên giao hàng

Vai trò hệ thống

Cây

Danh mục cây

Danh sách cây

Hóa đơn

Báo cáo

danh sách chức năng

Hóa đơn/

+thêm

-Xóa

Mã hóa đơn

tìm

Trạng thái

tất cả

Từ ngày

aa/bb/cccc

đến ngày

xx/yy/zzz

Người nhận

tìm

người gửi

tìm

Loại giao hàng

TÌM

Mã hóa đơn

Tên KH

NV giao hàng

Loại vận chuyển

Giá trị

Địa chỉ

Trạng thái

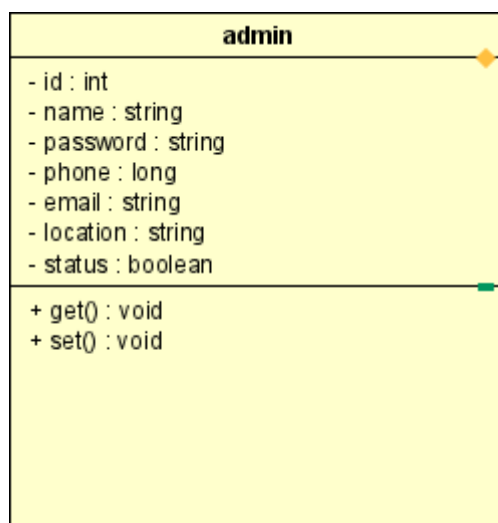
Ngày tạo

Danh sách các hóa đơn

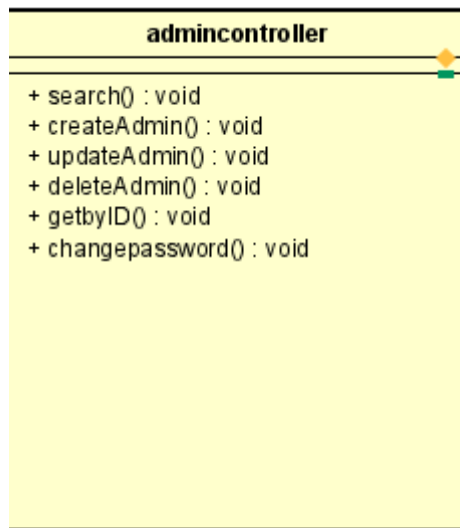
Hình 4.11: Giao diện quản lý hóa đơn của quản trị viên

4.2.2 Thiết kế lớp

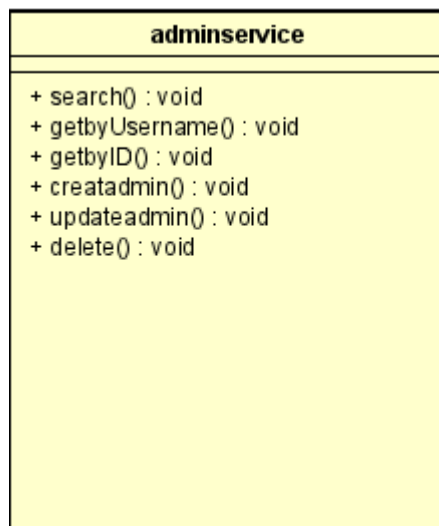
Đầu tiên là thiết kế lớp của package **Admin**



Hình 4.12: Chi tiết lớp admin

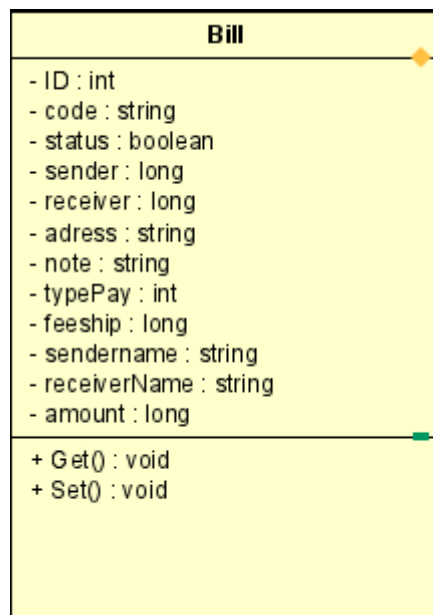


Hình 4.13: Chi tiết lớp admincontroller

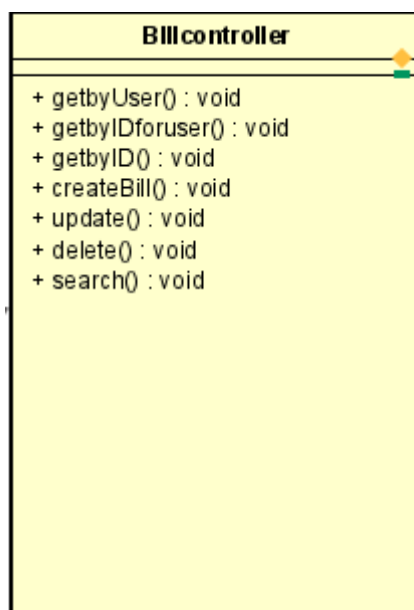


Hình 4.14: Chi tiết lớp adminservice

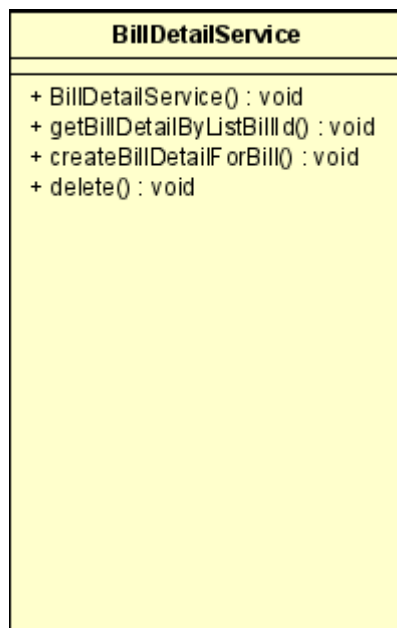
Tiếp theo đến lớp của package **Bill**



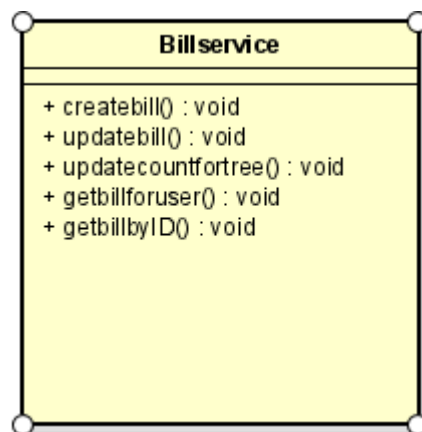
Hình 4.15: Chi tiết lớp Bill



Hình 4.16: Chi tiết lớp BillController



Hình 4.17: Chi tiết lớp Billdetail



Hình 4.18: Chi tiết lớp Billservice

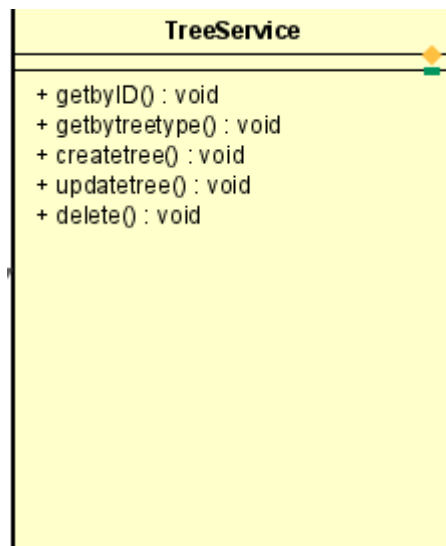
Tiếp đến là lớp của package **Tree**

Tree
<ul style="list-style-type: none">- ID : int- name : string- cost : long- discount : int- description : string- count : int- typeId : long- images : string
<ul style="list-style-type: none">+ get() : void+ set() : void

Hình 4.19: Chi tiết lớp Tree

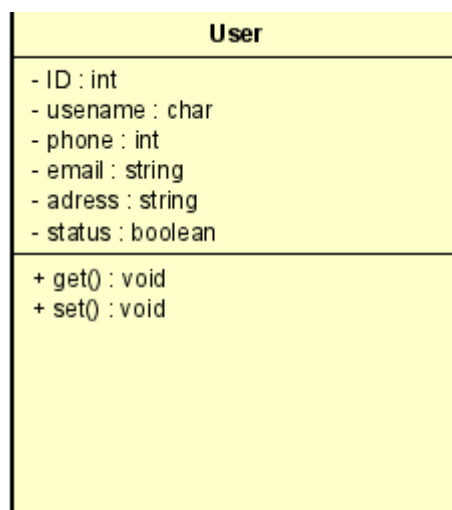
Treecontroller
<ul style="list-style-type: none">+ search() : void+ update() : void+ create() : void+ getByID() : void+ delete() : void

Hình 4.20: Chi tiết lớp TreeController

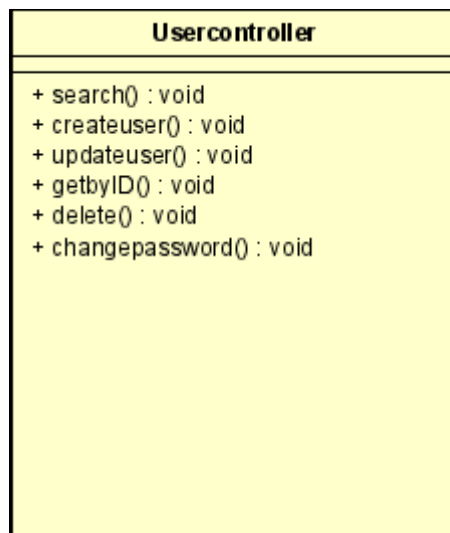


Hình 4.21: Chi tiết lớp **TreeService**

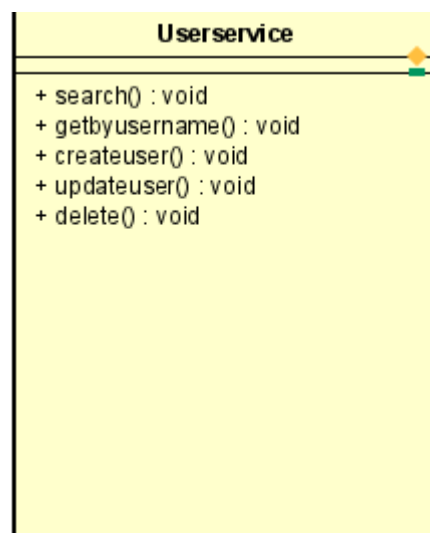
Và cuối cùng là thiết kế lớp của package **User**



Hình 4.22: Chi tiết lớp **User**

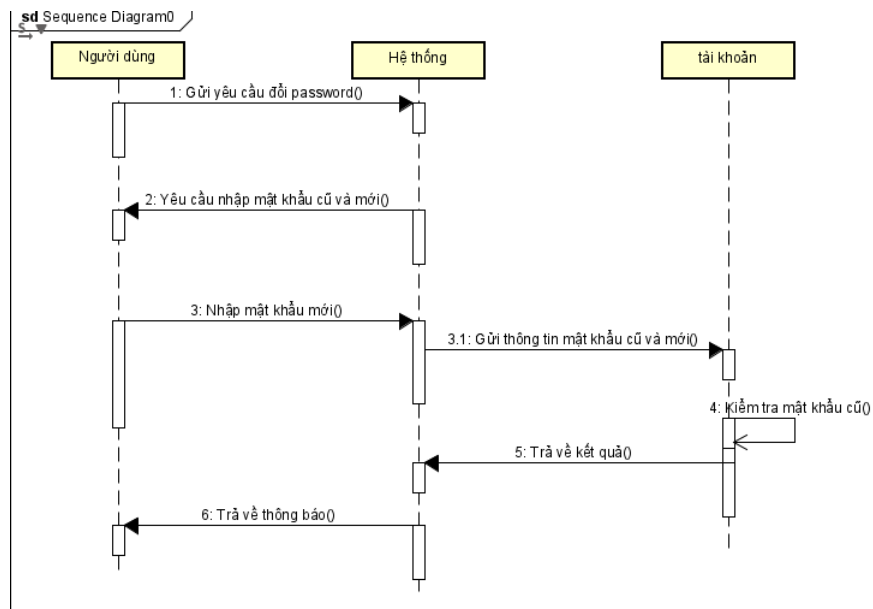


Hình 4.23: Chi tiết lớp UserController



Hình 4.24: Chi tiết lớp UserService

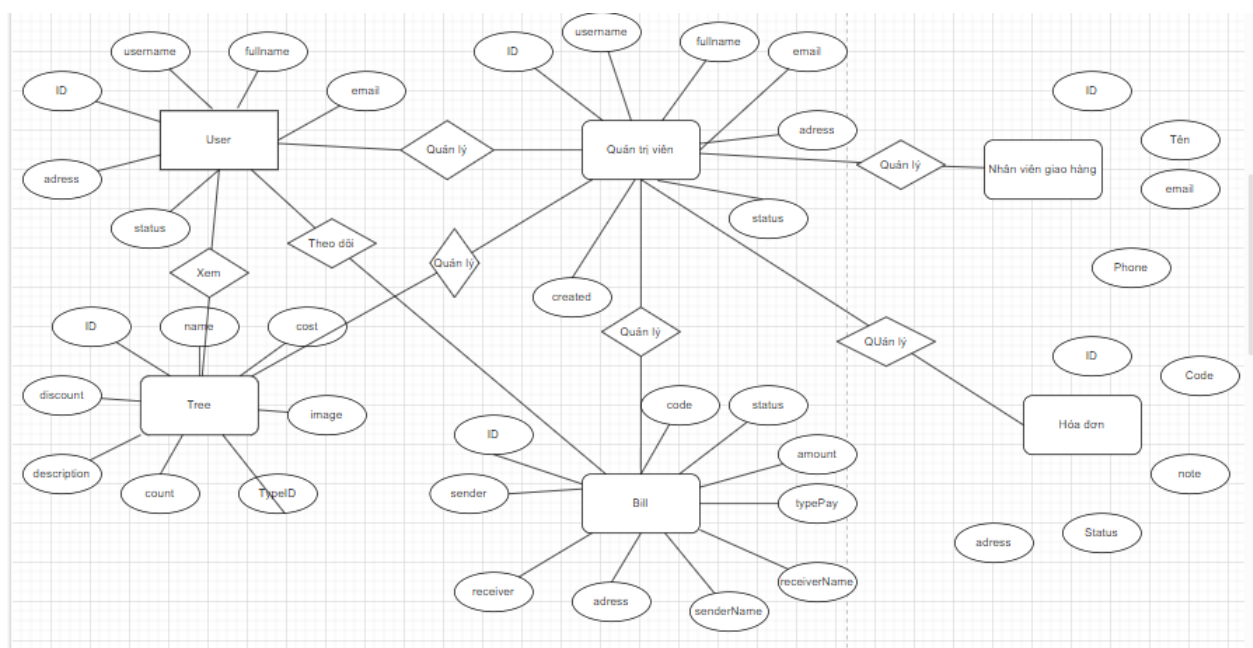
Biểu đồ tuần tự thay đổi password đối với người dùng (User)



Hình 4.25: Biểu đồ tuần tự thay đổi password với user

4.2.3 Thiết kế cơ sở dữ liệu

Sơ đồ thực thể liên kết



Hình 4.26: Sơ đồ thực thể liên kết

Bảng 4.1: Thiết kế bảng User

Tên	Kiểu dữ liệu	Khóa
UserID	int	Khóa chính
Username	String	
fullname	String	
email	String	
adress	String	
status	int	

Bảng 4.2: Thiết kế bảng quản trị viên

Tên	Kiểu dữ liệu	Khóa
Admin_ID	int	Khóa chính
Username	String	
fullname	String	
email	String	
adress	String	
status	int	
created	null	

Bảng 4.3: Thiết kế bảng tree

Tên	Kiểu dữ liệu	Loại khóa
Tree_ID	int	Khóa chính
name	string	
cost	int	
image	string	
type_ID	int	
discount	int	
description	string	

Bảng 4.4: Thiết kế bảng Bill

Tên	Kiểu dữ liệu	Loại khóa
Bill_ID	int	khóa chính
code	string	
status	int	
amount	long	
typePay	int	
sender	string	
receiver	string	
senderName	string	
ReceiverName	string	
adress	string	

Bảng 4.5: Thiết kế bảng Nhân viên giao hàng

Tên	Kiểu dữ liệu	Loại khóa
ID	int	Khóa chính
Name	string	
fullname	string	
phone	int	
adress	string	
email	string	

Bảng 4.6: Thiết kế bảng hóa đơn

Tên	Kiểu dữ liệu	Loại khóa
Hoadon_ID	int	Khóa chính
Code	string	
note	string	
status	int	
adress	string	

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 4.7: Bảng thư viện và các công nghệ sử dụng

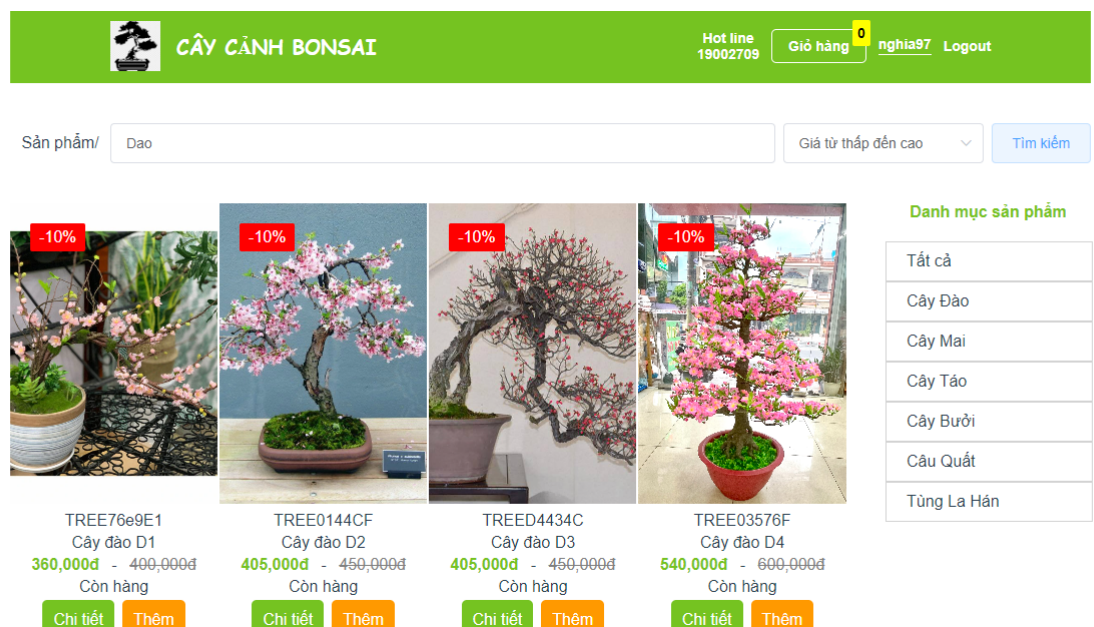
Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	IntelliJ	https://www.jetbrains.com/idea/download/#section=windows
JDK8	Java8	https://downloadvn.com/java-development-kit-8
Giao diện	Visual Code	https://code.visualstudio.com/download
API	Postman	https://www.postman.com/downloads/
Quản trị CSDL	MySQL workbench	https://dev.mysql.com/downloads/workbench/
Vẽ UML	Astah	https://astah.net/downloads/
Vẽ UML	drawio	https://app.diagrams.net/
Thiết kế giao diện	Paint	https://www.thegioididong.com/game-app/tai-paint-262084
Client	Nodejs	https://nodejs.org/en/download/
Tìm kiếm	GG chorme	
framework	Thư viện VueJs	https://vuejs.org/

4.3.2 Kết quả đạt được

Bảng 4.8: Bảng tổng hợp kết quả

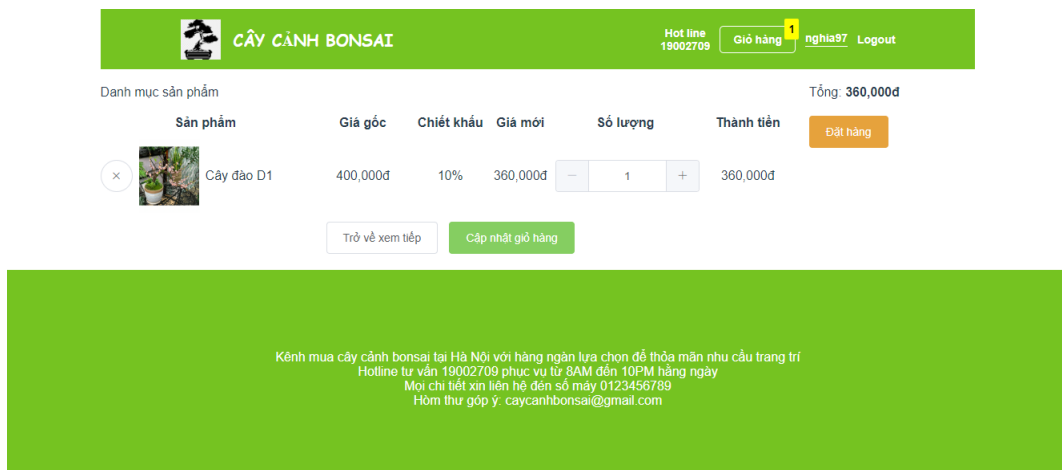
Sản phẩm	số lượng	mô tả
ĐATN	1	Gồm nhiều folder
Folder	7	Gồm:Admin,Backup,Baocao ,Bieudo,Client,Server,Picture
Client	10	Gồm:User,Header,Tree,Bill,Cart Login,Signup,Treedetail, homepage
Server	9	Gồm nhiều package:admin,auth,common,configure, core,media, shiper, tree,user,
Git	1	330MB dung lượng

4.3.3 Minh họa các chức năng chính



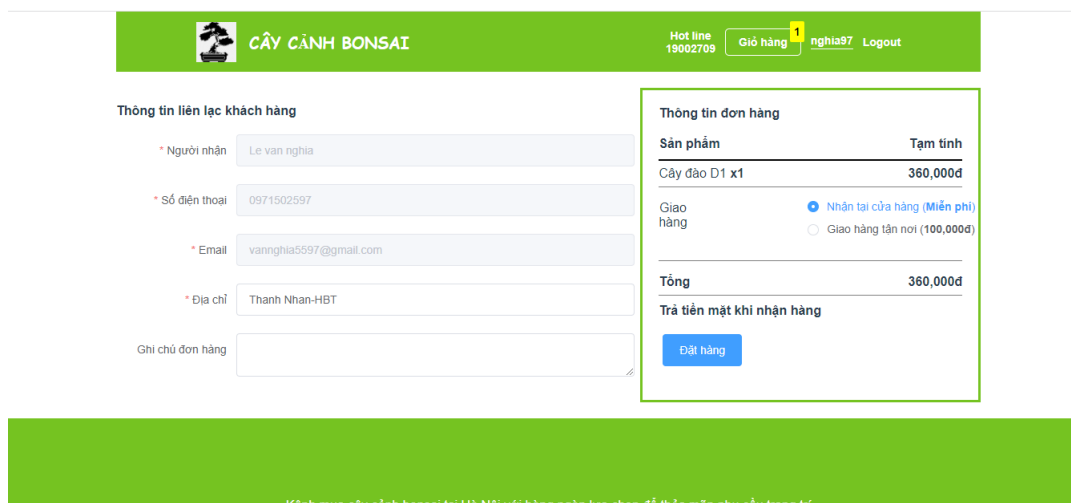
Hình 4.27: Chức năng tìm kiếm cho người dùng

Chức năng này giúp người dùng tìm kiếm theo mong muốn và sắp xếp chúng theo lọc



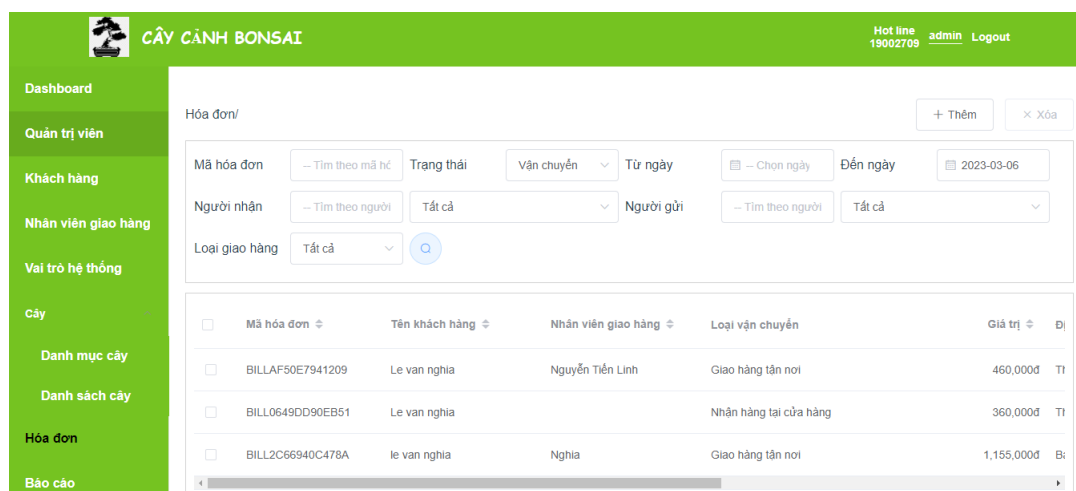
Hình 4.28: Thêm vào giỏ hàng của người dùng

Giúp người dùng thêm cây vào giỏ hàng chờ thành toán



Hình 4.29: Đặt hàng cho người dùng

Người dùng đặt hàng và chọn phương thức thanh toán và điền địa chỉ



Hình 4.30: Admin tìm kiếm hóa đơn mong muốn

Có nhiệm vụ tìm kiếm hóa đơn theo mong muốn

Cây Cảnh Bonsai

Dashboard
Quản trị viên
Khách hàng
Nhân viên giao hàng
Vai trò hệ thống
Cây
Danh mục cây
Danh sách cây
Hóa đơn
Báo cáo

Danh sách cây/

Mã cây	Tên cây
TREE5FAB5A	Cây Bưởi B1
TREE2F2F26	Cây Bưởi B2
TREE2C8744	Cây Bưởi B3
TREE95F73A	Cây Bưởi B4
TREE76e9E1	Cây đào D1
TREE0144CF	Cây đào D2

Loại cây: Cây Đào

Tên cây:

Giá: 0

Chiết khấu: 0

Số lượng: 0

Mô tả:

Hình ảnh: [Click to upload](#)

[Lưu](#)

Hình 4.31: Thêm cây vào danh mục mong muốn của quản trị viên

Quản trị viên thêm cây vào mục mình mong muốn kèm hình ảnh

4.4 Kiểm thử

Cây Cảnh Bonsai Hotline 19002799 [Giỏ hàng](#) [nghia97](#) [Logout](#)

Danh mục hóa đơn: BILLEFE7385E3546

Chi tiết đơn hàng	Tổng
Sản phẩm	
Cây đào D1 x1	360,000đ
Giao nhận hàng	Giao hàng tận nơi 100,000đ
Tổng cộng	460,000đ
Địa chỉ giao	Thanh Nhan-HBT
Lưu ý	Giao hàng cẩn thận

Đơn hàng của bạn đặt thành công

Mã đơn hàng: BILLEFE7385E3546
Ngày: 06/03/2023 02:21:44
Tổng cộng: 460,000đ
Phương thức thanh toán: Trả tiền mặt khi nhận hàng

Thông tin liên hệ khách hàng

Họ tên: Le van nghia
Số điện thoại: 0971502597
Email: vannghia5597@gmail.com
Địa chỉ: Thanh Nhan-HBT

[Hủy](#)

Hình 4.32: Người dùng khởi tạo đơn hàng

Sau khi người dùng đặt hàng xong thì sẽ có thông tin đơn hàng được khởi tạo. Quản trị viên vào xác nhận đơn hàng hoặc hủy đơn hàng

Hình 4.33: Quản trị viên xem đơn hàng khởi tạo của ngày hôm nay

Tiếp theo đó quản trị viên xác nhận đơn hàng và tạo người vận chuyển cho đơn hàng đó

Hình 4.34: Xác nhận đơn và tạo người vận chuyển đơn

Cuối cùng quản trị viên hoàn thành đơn nếu đã giao đơn xong cho User:

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNG GIÁ

Hóa đơn >

Mã hóa đơn

BILLEFE7385E3546

Trạng thái

Hoàn thành

Người nhận

Le van nghĩa

Loại giao hàng

Giao hàng tận nơi

Người gửi

Nguyễn Tiến Linh

* Địa chỉ

Thanh Nhan - HBT

Ghi chú

Giao hàng cẩn thận

Tổng

460,000đ

Danh sách cây

Mã cây	Tên cây	Giá gốc	Chiết khấu	Số lượng	Giá trị
TREE76e9E1	Cây đào D1	400,000đ	10	1	360,000đ

Lưu hóa đơn

Hình 4.35: Xác nhận hoàn thành đơn và tổng kết

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

5.1 Ưu và nhược điểm của mô hình MVC

5.1.1 Ưu điểm mô hình MVC

- Đầu tiên, nhắc tới ưu điểm mô hình MVC thì đó là băng thông (Bandwidth) nhẹ vì không sử dụng viewstate nên khá tiết kiệm băng thông. Việc giảm băng thông giúp website hoạt động ổn định hơn.
- Kiểm tra đơn giản và dễ dàng, kiểm tra lỗi phần mềm trước khi bàn giao lại cho người dùng.
- Một lợi thế chính của MVC là nó tách biệt các phần Model, Controller và View với nhau.
- Sử dụng mô hình MVC chức năng Controller có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau
- Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau.
- Có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau.
- Hỗ trợ TTD (test-driven development). Chúng ta có thể tạo một ứng dụng với unit test và viết các won test case.
- Phiên bản mới nhất của MVC hỗ trợ trợ thiết kế responsive website mặc định và các mẫu cho mobile. Chúng ta có thể tạo công cụ View của riêng mình với cú pháp đơn giản hơn nhiều so với công cụ truyền thống.

5.1.2 Nhược điểm mô hình MVC

Bên cạnh những ưu điểm MVC mang lại thì nó cũng có một số nhược điểm cần khắc phục.

MVC đa phần phù hợp với công ty chuyên về website hoặc các dự án lớn thì mô hình này phù hợp hơn so với với các dự án nhỏ, lẻ vì khá là cồng kềnh và mất thời gian.



Hình 5.1: Nhược điểm mô hình MVC không hỗ trợ Preview như ASP.NET

- Không thể Preview các trang như ASP.NET.
- Khó triển khai.

5.2 Vì sao nên sử dụng mô hình MVC

Quy trình phát triển nhanh hơn

MVC hỗ trợ phát việc phát triển nhanh chóng và song song. Nếu một **mô hình MVC** được dùng để phát triển bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên **View** và một developer khác có thể làm việc với **Controller** để tạo logic nghiệp vụ cho ứng dụng web đó.

Khả năng cung cấp nhiều chế độ view:

Trong **mô hình MVC**, bạn có thể tạo nhiều View cho chỉ một mô hình. Ngày nay, nhu cầu có thêm nhiều cách mới để truy cập ứng dụng và đang ngày càng tăng. Do đó, việc sử dụng MVC để phát triển chắc chắn là một giải pháp tuyệt vời.

Hơn nữa, với phương pháp này, việc nhân bản code rất hạn chế. Vì nó tách biệt dữ liệu và logic nghiệp vụ khỏi màn hình.

Các sửa đổi không ảnh hưởng đến toàn bộ mô hình:

Đối với bất kỳ ứng dụng web nào, người dùng có xu hướng thay đổi thường xuyên. Bạn có thể quan sát thông qua những thay đổi thường xuyên về màu sắc, font chữ, bố cục màn hình. Hay là thêm hỗ trợ thiết bị mới cho điện thoại hay máy tính bảng...

Việc thêm một kiểu view mới trong MVC rất đơn giản. Vì phần Model không phụ thuộc vào phần View. Do đó, bất kỳ thay đổi nào trong Model sẽ không ảnh

hưởng đến toàn bộ kiến trúc.

MVC Model trả về dữ liệu mà không cần định dạng:

MVC pattern có thể trả về dữ liệu mà không cần áp dụng bất kỳ định dạng nào. Do đó, các thành phần giống nhau có thể được sử dụng với bất kỳ giao diện nào.

Ví dụ: tất cả loại dữ liệu đều có thể được định dạng bằng HTML. Ngoài ra, nó cũng có thể được định dạng bằng Macromedia Flash hay Dream Viewer.

Hỗ trợ kỹ thuật Asynchronous:

Kiến trúc MVC có thể được tích hợp với cả JavaScript Framework. Có nghĩa là, các ứng dụng MVC có thể hoạt động ngay cả với các file PDF, trình duyệt riêng cho web hay các widget trên desktop.

Ngoài ra, MVC cũng hỗ trợ kỹ thuật Asynchronous, giúp các developer phát triển các ứng dụng có thể load rất nhanh.

Nền tảng MVC thân thiện với SEO:

Nền tảng MVC hỗ trợ phát triển các trang web thân thiện với SEO. Bằng nền tảng này, bạn có thể dễ dàng phát triển các URL thân thiện với SEO để tạo ra nhiều lượt truy cập hơn.

Những ngôn ngữ như JavaScript hay jQuery có thể được tích hợp với MVC. Từ đó phát triển nhiều ứng dụng web giàu tính năng, đặc biệt là với mô hình MVC trong Java.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Bắt tay vào làm đồ án tốt nghiệp với mức khởi đầu về các công nghệ lựa chọn gần như bằng không, các chức năng vụ trong đồ án còn chỉ là những ý tưởng mập mờ. Nhưng với sự hỗ trợ nhiệt tình từ thầy Tạ Hải Tùng, cùng với sức nghiên cứu, tìm hiểu, thử nghiệm thì em cũng đã hoàn thành được đồ án tốt nghiệp này. Do còn thiếu kinh nghiệm trong việc xây dựng và phát triển phần mềm, sản phẩm hoàn thành vẫn còn tồn tại nhiều thiếu sót, em mong muốn nhận được thêm những đóng góp từ các thầy các cô và các bạn, để khắc phục không chỉ đưa được sản phẩm vào ứng dụng thực tế mà còn rút kinh nghiệm cho cá dự án tiếp theo của bản thân mình. Sản phẩm được hoàn thành tuy không phải là sản phẩm hoàn hảo nhất, nhưng sản phẩm đáp ứng các chức năng mua bán cây cảnh cho người dùng, quản lý hóa đơn, xem chi tiết cây cảnh, xem giá thành, báo cáo hóa đơn cũng như tổng kết số lượng cây thông qua Quản trị viên. Qua sản phẩm lần này, em được làm quen và hiểu thêm về các công nghệ làm web đặc biệt là hai framework đang rất phổ biến là vuejs và spring boot. Trong tương lai em mong muốn được nghiên cứu kỹ hơn để đưa sản phẩm vào thực tế ứng dụng đóng góp cho xã hội. Lời cuối cùng em mong nhận được những hỗ trợ, những ý kiến đóng góp để hoàn thành đồ án được kết quả tốt hơn.

Em xin chân thành cảm ơn!

6.2 Hướng phát triển

Trong tương lai em sẽ cố gắng hoàn thiện phần chức năng còn thiếu và sẽ cố gắng làm ra một sản phẩm hoàn thiện nhất với đầy đủ tầm nhìn và xu hướng của xã hội.

Em sẽ cố gắng trau dồi kiến thức để học hỏi thêm những cách làm web xinh đẹp bắt mắt hơn cùng với đó là đẩy mạnh công nghệ ReactJS và VueJS em đang làm

MỘT SỐ LƯU Ý VỀ TÀI LIỆU THAM KHẢO

1 <**Web Programming with HTML5, CSS**> **John Dean** Thương hiệu: English books *Artificial intelligence*, vol. 63, no. 1-2, pp. 341–385, 1993

2<**Thực hành sử dụng VueJS cho thiết kế website:** >

Thạc Bình Cường - Vũ Thị Hậu Nhà Xuất Bản Giáo Dục Việt Nam. Nhà Xuất Bản Giáo Dục Việt Nam 2007.

3 **N. T. Hải** *Mạng máy tính và các hệ thống mở*. Nhà xuất bản giáo dục, 1999.

4<**High Performance MySQL Proven Strategies for Operating at Scale, 4th Edition:**> **Silvia Botros** Nhà xuất bản quốc tế năm 2021

5<**Đồ án tốt nghiệp mẫu**> ĐH bách khoa hà Nội 2022

6<**Tài liệu tham khảo từ Internet:**

<https://www.w3schools.com/javascrip>

<https://fullstack.edu.vn/> của tác giả **Sơn Đăng**

PHỤ LỤC