**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**KHOÁ LUẬN TỐT NGHIỆP**

**ỨNG DỤNG CHO THUÊ XE TẬP LÁI TRÊN**

**WEB APP**

**SVTH:**

**TRẦN THẾ DUY – 17093161**

**NGUYỄN VĂN VỸ – 17100751**

**GVHD:**

**TS.NGUYỄN TRỌNG TIẾN**

**Thành phố Hồ Chí Minh, năm 2021**

**INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY**

**FACULTY OF INFORMATION TECHNOLOGY**

****

**TRAN THE DUY**

**NGUYEN VAN VY**

**CAR RENTAL APPLICATION ON**

**WEB APP**

Major: Software Engineering

**Supervior: Dr. Nguyen Trong Tien**

Ho Chi Minh City, 2021

**CAR RENTAL APPLICATION ON**

**WEB APP**

**Abstract:** Build a website where customers can reserve a form to rent a practice car, and then send the customers’s information to the server in real-time. The server will notify the website to help the staff receive the request to confirm the request from the customer

**Reason for writing:** With the development of information technology today, the life and needs of society are increasingly improved. Customers only need to book in advance through the website and the staff will confirm the customer's request, avoiding waiting when they need to rent a practice car.

**Problem:** Build a website for customers to make reservations, view vehicle information, and articles. Build a website dedicated to the company to be able to manage employees, order management, vehicle management, vehicle catalog management, article management, article card management for managers, admin only more statistics; Receiving orders for employees, especially real-time processing of pre-orders. Use Spring Server-Side Events to synchronize reservations, store data using the Mysql database system on AWS RDS. The system is located on AWS EC2 to speed up access from the website.

**Method**: Back-End uses Spring Boot RESTful API, to be able to create APIs to communicate with the website, in addition, it also uses modules like security to increase security, audit JPA to be able to track end-user changes system data, ... The website uses HTML, CSS, and JavaScript to create the user interface, in addition, it also uses libraries such as Bootstrap, jQuery to increase the friendliness of the interface, AJAX provides feedback and better information processing.

**Result**: Successfully built a website for both guests and employees in the company, the functions are almost complete.

**Conclusion**: Successfully built a mobile application website, the functions are almost complete. However, there are still some limitations in terms of performance and code optimization. In addition, the report is not highly complete

**LỜI CẢM ƠN**

Lời đầu tiên nhóm chúng em xin gửi lời cảm ơn chân thành nhất đến Quý Thầy Cô Trường Đại học Công nghiệp Thành phố Hồ Chí Minh và đặc biệt là Quý Thầy Cô Khoa Công nghệ thông tin trường Đại học Công nghiệp Thành phố Hồ Chí Minh đã trang bị cho chúng em đầy đủ các kiến thức tổng quan và chuyên ngành để có thể vận dụng vào đề tài khóa luận này.

Nhằm củng cố và vận dụng những kiến thức đã tích lũy được trong 4 năm học, nhóm chúng em quyết định chọn đề tài “Ứng dụng cho thuê xe tập lái trên web app” để làm đề tài khóa luận tốt nghiệp. Bên cạnh đó, chúng em cũng xin gửi lời cảm ơn sâu sắc nhất đến TS. Nguyễn Trọng Tiến, Thầy đã dành thời gian, tâm huyết và tận tâm hướng dẫn để bài khóa luận được hoàn thiện và đạt kết quả tốt nhất.

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

Cuối cùng, chúng em xin kính chúc quý Thầy Cô dồi dào sức khỏe và thành công trong sự nghiệp trồng người cao quý.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn !

Thành phố Hồ Chí Minh, ngày 20 tháng 05 năm 2021

**Sinh viên thực hiện**

**Trần Thế Duy, Nguyễn Văn Vỹ**

# NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN

TP. HCM, ngày tháng năm 2021

**Giảng viên hướng dẫn**

**TS. Nguyễn Trọng Tiến**

# NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN

TP. HCM, ngày tháng năm 2021

**Giảng viên phản biện**

MỤC LỤC

[**LỜI CẢM ƠN 1**](#_Toc74244097)

[**NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN 2**](#_Toc74244098)

[**NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN 3**](#_Toc74244099)

[**MỤC LỤC 4**](#_Toc74244100)

[**MỤC LỤC HÌNH ẢNH 16**](#_Toc74244101)

[**MỤC LỤC BẢNG BIỂU 21**](#_Toc74244102)

[**CHƯƠNG 1. GIỚI THIỆU 23**](#_Toc74244103)

[**1.1 Tổng quan 23**](#_Toc74244104)

[**1.2 Mục tiêu đề tài 23**](#_Toc74244105)

[**1.3 Phạm vi đề tài 24**](#_Toc74244106)

[**1.4 Mô tả yêu cầu chức năng 26**](#_Toc74244107)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 27**](#_Toc74244108)

[**2.1 Web Application [1] 27**](#_Toc74244109)

[**2.1.1 Giới thiệu về Web Application 27**](#_Toc74244110)

[**2.1.2 Đặc điểm của web application 28**](#_Toc74244111)

[**2.2 Ngôn ngữ lập trình HTML [2] 29**](#_Toc74244112)

[**2.2.1 Tổng quan về ngôn ngữ HTML 29**](#_Toc74244113)

[**2.2.2 HTML hoạt động như thế nào 30**](#_Toc74244114)

[**2.2.3 Các ưu điểm giúp HTML được sử dụng rộng rãi 30**](#_Toc74244115)

[**2.3 Ngôn ngữ lập trình CSS [3] 30**](#_Toc74244116)

[**2.3.1 Tổng quan về CSS 31**](#_Toc74244117)

[**2.3.2 CSS hoạt động như thế nào 32**](#_Toc74244118)

[**2.3.3 Ưu điểm của CSS 33**](#_Toc74244119)

[**2.4 Ngôn ngữ lập trình JavaScript [4] 35**](#_Toc74244120)

[**2.4.1 Tổng quan về JavaScript 35**](#_Toc74244121)

[**2.4.2 Cách JavaScript hoạt động 36**](#_Toc74244122)

[**2.4.3 Ưu điểm của JavaScript 36**](#_Toc74244123)

[**2.5 Ngôn ngữ lập tình Java [5] 37**](#_Toc74244124)

[**2.5.1 Java là gì? 37**](#_Toc74244125)

[**2.5.2 Các tính năng của Java 38**](#_Toc74244126)

[**2.5.3 Java được dùng để làm gì? 39**](#_Toc74244127)

[**2.6 Spring Boot trong Spring Framework 40**](#_Toc74244128)

[**2.6.1 Spring là gì? [6] 40**](#_Toc74244129)

[**2.6.2 Kiến trúc và các module của Spring Framework 40**](#_Toc74244130)

[**2.6.3 Spring Boot là gì? [7] 41**](#_Toc74244131)

[**2.6.4 Ưu điểm của Spring Boot 42**](#_Toc74244132)

[**2.7 RESTful API [8] 42**](#_Toc74244133)

[**2.7.1 RESTful API là gì? 43**](#_Toc74244134)

[**2.7.2 Cách thức hoạt động của RESTful API 43**](#_Toc74244135)

[**2.7.3 Ưu điểm của RESTful API 44**](#_Toc74244136)

[**2.8 Hệ quản trị cơ sở dữ liệu MySQL [9] 44**](#_Toc74244137)

[**2.8.1 MySQL là gì? 45**](#_Toc74244138)

[**2.8.2 Cách thức hoạt động của MySQL 45**](#_Toc74244139)

[**2.8.3 Vì sao MySQL lại phổ biến 46**](#_Toc74244140)

[**2.9 AWS là gì và các dịch vụ sử dụng trong đề tài 47**](#_Toc74244141)

[**2.9.1 AWS là gì? [12] 47**](#_Toc74244142)

[**2.9.2 Các ưu điểm của AWS 47**](#_Toc74244143)

[**2.9.3 Amazon RDS là gì? [13] 49**](#_Toc74244144)

[**2.9.4 Các tính năng của Amazon RDS [13] 50**](#_Toc74244145)

[**2.9.5 Amazon EC2 là gì? [14] 52**](#_Toc74244146)

[**2.9.6 Các đặc điểm của Amazon EC2 [10] 53**](#_Toc74244147)

[**2.10 API jQuery Ajax [11] 54**](#_Toc74244148)

[**2.10.1 Ajax là gì? 54**](#_Toc74244149)

[**2.10.2 jQuery Ajax 55**](#_Toc74244150)

[**CHƯƠNG 3. PHÂN TÍCH VÀ ĐẶC TẢ CÁC YÊU CẦU CHỨC NĂNG 57**](#_Toc74244151)

[**3.1 Use case tổng quát 57**](#_Toc74244152)

[**3.2 Use case chi tiết của từng tác nhân 57**](#_Toc74244153)

[**3.3 Danh sách tác nhân và mô tả 59**](#_Toc74244154)

[**3.4 Danh sách các tình huống hoạt động (Use cases) 60**](#_Toc74244155)

[**3.5 Danh sách Mock up 61**](#_Toc74244156)

[**3.5.1 Mock up đăng nhập 62**](#_Toc74244157)

[**3.5.2 Mock up thêm nhân viên 62**](#_Toc74244158)

[**3.5.3 Mock up ngưng hoạt động nhân viên 63**](#_Toc74244159)

[**3.5.4 Mock up mở lại nhân viên 64**](#_Toc74244160)

[**3.5.5 Mock up xoá thẻ bài viết 64**](#_Toc74244161)

[**3.5.6 Mock up ngưng hoạt động bài viết 64**](#_Toc74244162)

[**3.5.7 Mock up mở lại bài viết 64**](#_Toc74244163)

[**3.5.8 Mock up ngưng hoạt động danh mục 64**](#_Toc74244164)

[**3.5.9 Mock up mở lại danh mục 64**](#_Toc74244165)

[**3.5.10 Mock up ngưng hoạt động xe 64**](#_Toc74244166)

[**3.5.11 Mock up mở lại xe 64**](#_Toc74244167)

[**3.5.12 Mock up chỉnh sửa thẻ bài viết 64**](#_Toc74244168)

[**3.5.13 Mock up thêm thẻ bài viết 64**](#_Toc74244169)

[**3.5.14 Mock up thêm bài viêt 64**](#_Toc74244170)

[**3.5.15 Mock up chỉnh sửa bài viết 64**](#_Toc74244171)

[**3.5.16 Mock up chỉnh sửa danh mục xe 64**](#_Toc74244172)

[**3.5.17 Mock up thêm danh mục xe 64**](#_Toc74244173)

[**3.5.18 Mock up thêm xe 64**](#_Toc74244174)

[**3.5.19 Mock up chỉnh sửa xe 65**](#_Toc74244175)

[**3.5.20 Mock up thanh toán 66**](#_Toc74244176)

[**3.5.21 Mock up thuê xe 66**](#_Toc74244177)

[**3.5.22 Mock up xác nhận đặt trước 66**](#_Toc74244178)

[**3.5.23 Mock up huỷ đặt trước 66**](#_Toc74244179)

[**3.5.24 Mock up đăng ký đặt trước 66**](#_Toc74244180)

[**3.5.25 Mock up xem chi tiết hoá đơn 66**](#_Toc74244181)

[**3.5.26 Mock up xem bài viết 66**](#_Toc74244182)

[**3.5.27 Mock up xem thông tin xe 66**](#_Toc74244183)

[**3.5.28 Mock up tìm kiếm hoá đơn 66**](#_Toc74244184)

[**3.5.29 Mock up thống kê 66**](#_Toc74244185)

[**3.5.30 Mock up cập nhật thông tin người dùng 67**](#_Toc74244186)

[**3.6 Tình huống hoạt động 67**](#_Toc74244187)

[**3.6.1 Đặc tả UC01\_Đăng nhập 67**](#_Toc74244188)

[**3.6.1.1 Mô tả 67**](#_Toc74244189)

[**3.6.1.2 Activity diagram 69**](#_Toc74244190)

[**3.6.1.3 Sequence diagram 70**](#_Toc74244191)

[**3.6.2 Đặc tả UC02\_Thêm nhân viên 70**](#_Toc74244192)

[**3.6.2.1 Mô tả 70**](#_Toc74244193)

[**3.6.2.2 Activity diagram 72**](#_Toc74244194)

[**3.6.2.3 Sequence diagram 73**](#_Toc74244195)

[**3.6.3 Đặc tả UC03\_Ngưng hoạt động nhân viên 73**](#_Toc74244196)

[**3.6.3.1 Mô tả 73**](#_Toc74244197)

[**3.6.3.2 Activity diagram 75**](#_Toc74244198)

[**3.6.3.3 Sequence diagram 76**](#_Toc74244199)

[**3.6.4 Đặc tả UC04\_Mở lại nhân viên 76**](#_Toc74244200)

[**3.6.4.1 Mô tả 76**](#_Toc74244201)

[**3.6.4.2 Activity diagram 78**](#_Toc74244202)

[**3.6.4.3 Sequence diagram 79**](#_Toc74244203)

[**3.6.5 Đặc tả UC05\_Xoá thẻ bài viết 79**](#_Toc74244204)

[**3.6.5.1 Mô tả 79**](#_Toc74244205)

[**3.6.5.2 Activity diagram 81**](#_Toc74244206)

[**3.6.5.3 Sequence diagram 82**](#_Toc74244207)

[**3.6.6 Đặc tả UC06\_Ngưng hoạt động bài viết 82**](#_Toc74244208)

[**3.6.6.1 Mô tả 82**](#_Toc74244209)

[**3.6.6.2 Activity diagram 84**](#_Toc74244210)

[**3.6.6.3 Sequence diagram 85**](#_Toc74244211)

[**3.6.7 Đặc tả UC07\_Mở lại bài viết 85**](#_Toc74244212)

[**3.6.7.1 Mô tả 85**](#_Toc74244213)

[**3.6.7.2 Activity diagram 87**](#_Toc74244214)

[**3.6.7.3 Sequence diagram 88**](#_Toc74244215)

[**3.6.8 Đặc tả UC08\_Ngưng hoạt động danh mục 88**](#_Toc74244216)

[**3.6.8.1 Mô tả 88**](#_Toc74244217)

[**3.6.8.2 Activity diagram 90**](#_Toc74244218)

[**3.6.8.3 Sequence diagram 91**](#_Toc74244219)

[**3.6.9 Đặc tả UC09\_Mở lại danh mục 91**](#_Toc74244220)

[**3.6.9.1 Mô tả 91**](#_Toc74244221)

[**3.6.9.2 Activity diagram 93**](#_Toc74244222)

[**3.6.9.3 Sequence diagram 94**](#_Toc74244223)

[**3.6.10 Đặc tả UC10\_Ngưng hoạt động xe 94**](#_Toc74244224)

[**3.6.10.1 Mô tả 94**](#_Toc74244225)

[**3.6.10.2 Activity diagram 96**](#_Toc74244226)

[**3.6.10.3 Sequence diagram 97**](#_Toc74244227)

[**3.6.11 Đặc tả UC11\_Mở lại xe 97**](#_Toc74244228)

[**3.6.11.1 Mô tả 97**](#_Toc74244229)

[**3.6.11.2 Activity diagram 99**](#_Toc74244230)

[**3.6.11.3 Sequence diagram 100**](#_Toc74244231)

[**3.6.12 Đặc tả UC12\_Chỉnh sửa thẻ bài viết 100**](#_Toc74244232)

[**3.6.12.1 Mô tả 100**](#_Toc74244233)

[**3.6.12.2 Activity diagram 102**](#_Toc74244234)

[**3.6.12.3 Sequence diagram 103**](#_Toc74244235)

[**3.6.13 Đặc tả UC13\_Thêm thẻ bài viết 103**](#_Toc74244236)

[**3.6.13.1 Mô tả 103**](#_Toc74244237)

[**3.6.13.2 Activity diagram 105**](#_Toc74244238)

[**3.6.13.3 Sequence diagram 106**](#_Toc74244239)

[**3.6.14 Đặc tả UC14\_Thêm bài viết 106**](#_Toc74244240)

[**3.6.14.1 Mô tả 106**](#_Toc74244241)

[**3.6.14.2 Activity diagram 108**](#_Toc74244242)

[**3.6.14.3 Sequence diagram 109**](#_Toc74244243)

[**3.6.15 Đặc tả UC15\_Chỉnh sửa bài viết 109**](#_Toc74244244)

[**3.6.15.1 Mô tả 109**](#_Toc74244245)

[**3.6.15.2 Activity diagram 111**](#_Toc74244246)

[**3.6.15.3 Sequence diagram 112**](#_Toc74244247)

[**3.6.16 Đặc tả UC16\_Chỉnh sửa danh mục xe 113**](#_Toc74244248)

[**3.6.16.1 Mô tả 113**](#_Toc74244249)

[**3.6.16.2 Activity diagram 114**](#_Toc74244250)

[**3.6.16.3 Sequence diagram 115**](#_Toc74244251)

[**3.6.17 Đặc tả UC17\_Thêm danh mục xe 115**](#_Toc74244252)

[**3.6.17.1 Mô tả 115**](#_Toc74244253)

[**3.6.17.2 Activity diagram 117**](#_Toc74244254)

[**3.6.17.3 Sequence diagram 118**](#_Toc74244255)

[**3.6.18 Đặc tả UC18\_Thêm xe 118**](#_Toc74244256)

[**3.6.18.1 Mô tả 118**](#_Toc74244257)

[**3.6.18.2 Activity diagram 120**](#_Toc74244258)

[**3.6.18.3 Sequence diagram 121**](#_Toc74244259)

[**3.6.19 Đặc tả UC19\_Chỉnh sửa xe 121**](#_Toc74244260)

[**3.6.19.1 Mô tả 121**](#_Toc74244261)

[**3.6.19.2 Activity diagram 123**](#_Toc74244262)

[**3.6.19.3 Sequence diagram 124**](#_Toc74244263)

[**3.6.20 Đặc tả UC20\_Thanh toán 124**](#_Toc74244264)

[**3.6.20.1 Mô tả 124**](#_Toc74244265)

[**3.6.20.2 Activity diagram 126**](#_Toc74244266)

[**3.6.20.3 Sequence diagram 127**](#_Toc74244267)

[**3.6.21 Đặc tả UC21\_Thuê xe 127**](#_Toc74244268)

[**3.6.21.1 Mô tả 127**](#_Toc74244269)

[**3.6.21.2 Activity diagram 129**](#_Toc74244270)

[**3.6.21.3 Sequence diagram 130**](#_Toc74244271)

[**3.6.22 Đặc tả UC22\_Xác nhận đặt trước 130**](#_Toc74244272)

[**3.6.22.1 Mô tả 130**](#_Toc74244273)

[**3.6.22.2 Activity diagram 131**](#_Toc74244274)

[**3.6.22.3 Sequence diagram 132**](#_Toc74244275)

[**3.6.23 Đặc tả UC23\_Huỷ đặt trước 132**](#_Toc74244276)

[**3.6.23.1 Mô tả 132**](#_Toc74244277)

[**3.6.23.2 Activity diagram 134**](#_Toc74244278)

[**3.6.23.3 Sequence diagram 135**](#_Toc74244279)

[**3.6.24 Đặc tả UC24\_Đăng ký đặt trước 135**](#_Toc74244280)

[**3.6.24.1 Mô tả 135**](#_Toc74244281)

[**3.6.24.2 Activity diagram 136**](#_Toc74244282)

[**3.6.24.3 Sequence diagram 137**](#_Toc74244283)

[**3.6.25 Đặc tả UC25\_Xem chi tiết hoá đơn 137**](#_Toc74244284)

[**3.6.25.1 Mô tả 137**](#_Toc74244285)

[**3.6.25.2 Activity diagram 138**](#_Toc74244286)

[**3.6.25.3 Sequence diagram 139**](#_Toc74244287)

[**3.6.26 Đặc tả UC26\_Xem bài viết 139**](#_Toc74244288)

[**3.6.26.1 Mô tả 139**](#_Toc74244289)

[**3.6.26.2 Activity diagram 140**](#_Toc74244290)

[**3.6.26.3 Sequence diagram 141**](#_Toc74244291)

[**3.6.27 Đặc tả UC27\_Xem thông tin xe 141**](#_Toc74244292)

[**3.6.27.1 Mô tả 141**](#_Toc74244293)

[**3.6.27.2 Activity diagram 143**](#_Toc74244294)

[**3.6.27.3 Sequence diagram 144**](#_Toc74244295)

[**3.6.28 Đặc tả UC28\_Tìm kiếm hoá đơn 144**](#_Toc74244296)

[**3.6.28.1 Mô tả 144**](#_Toc74244297)

[**3.6.28.2 Activity diagram 146**](#_Toc74244298)

[**3.6.28.3 Sequence diagram 147**](#_Toc74244299)

[**3.6.29 Đặc tả UC29\_Thống kê 147**](#_Toc74244300)

[**3.6.29.1 Mô tả 147**](#_Toc74244301)

[**3.6.29.2 Activity diagram 148**](#_Toc74244302)

[**3.6.29.3 Sequence diagram 149**](#_Toc74244303)

[**3.6.30 Đặc tả UC30\_Cập nhật thông tin 149**](#_Toc74244304)

[**3.6.30.1 Mô tả 149**](#_Toc74244305)

[**3.6.30.2 Activity diagram 151**](#_Toc74244306)

[**3.6.30.3 Sequence diagram 152**](#_Toc74244307)

[**CHƯƠNG 4. THIẾT KẾ VÀ HIỆN THỰC 153**](#_Toc74244308)

[**4.1 Class diagram 153**](#_Toc74244309)

[**4.2 Sitemap 153**](#_Toc74244310)

[**4.3 Database diagram 154**](#_Toc74244311)

[**4.4 Hiện thực 155**](#_Toc74244312)

[**CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 156**](#_Toc74244313)

[**5.1 Kết quả đạt được 156**](#_Toc74244314)

[**5.2 Hạn chế của đồ án 157**](#_Toc74244315)

[**5.3 Hướng phát triển 157**](#_Toc74244316)

[**TÀI LIỆU THAM KHẢO 158**](#_Toc74244317)

**MỤC LỤC HÌNH ẢNH**

[Hình 1.1 Danh sách chức năng của hệ thống 24](file:///F:\Rent-car-project\Tai_Lieu\TranTheDuy_NguyenVanVy_BaoCaoKLTN_DeTai14.docx#_Toc74243976)

[Hình 2.1 Web Application 26](#_Toc74243977)

[Hình 2.2 Ví dụ về HTML 27](#_Toc74243978)

[Hình 2.3 Cấu trúc của CSS 30](#_Toc74243979)

[Hình 2.4 Ví dụ về CSS 31](#_Toc74243980)

[Hình 2.5 Ví dụ về JavaScript 33](#_Toc74243981)

[Hình 2.6 Java 36](#_Toc74243982)

[Hình 2.7 Kiến trúc và module của Spring Framework 39](#_Toc74243983)

[Hình 2.8 Spring Boot 40](#_Toc74243984)

[Hình 2.9 RESTful API Design 41](#_Toc74243985)

[Hình 2.10 Mô hình hoạt động của RESTful API 41](#_Toc74243986)

[Hình 2.11 RESTful API truyền tải dữ liệu 42](#_Toc74243987)

[Hình 2.12 Cách thức MySQL hoạt động 43](#_Toc74243988)

[Hình 2.13 Bảng xếp hạng các hệ quản trị cơ sở dữ liệu 2018 44](#_Toc74243989)

[Hình 2.14 Amazon Web Services 47](#_Toc74243990)

[Hình 2.15 Amazon RDS 48](#_Toc74243991)

[Hình 2.16 Amazon EC2 51](#_Toc74243992)

[Hình 3.1 Use case tổng quát của Web App 55](#_Toc74243993)

[Hình 3.2 Use case chi tiết của khách hàng 56](#_Toc74243994)

[Hình 3.3 Use case chi tiết của nhân viên 56](#_Toc74243995)

[Hình 3.4 Use case chi tiết của quản lý 57](#_Toc74243996)

[Hình 3.5 Use case chi tiết của admin 57](#_Toc74243997)

[Hình 3.6 Mock up đăng nhập 60](#_Toc74243998)

[Hình 3.7 Mock up thêm nhân viên 60](#_Toc74243999)

[Hình 3.8 Mock up thêm xe 62](#_Toc74244000)

[Hình 3.9 Mock up chỉnh sửa xe 63](#_Toc74244001)

[Hình 3.10 Mock up thống kê 64](#_Toc74244002)

[Hình 3.11 Activity đăng nhập 67](#_Toc74244003)

[Hình 3.12 Sequence đăng nhập 68](#_Toc74244004)

[Hình 3.13 Activity thêm nhân viên 70](#_Toc74244005)

[Hình 3.14 Sequence thêm nhân viên 71](#_Toc74244006)

[Hình 3.15 Activity ngưng hoạt động nhân viên 73](#_Toc74244007)

[Hình 3.16 Sequence ngưng hoạt động nhân viên 74](#_Toc74244008)

[Hình 3.17 Activity mở lại nhân viên 76](#_Toc74244009)

[Hình 3.18 Sequence mở lại nhân viên 77](#_Toc74244010)

[Hình 3.19 Activity xoá thẻ bài viết 79](#_Toc74244011)

[Hình 3.20 Sequence xoá thẻ bài viết 80](#_Toc74244012)

[Hình 3.21 Activity ngưng hoạt động bài viết 82](#_Toc74244013)

[Hình 3.22 Sequence ngưng hoạt động bài viết 83](#_Toc74244014)

[Hình 3.23 Activity mở lại bài viết 85](#_Toc74244015)

[Hình 3.24 Sequence mở lại bài viết 86](#_Toc74244016)

[Hình 3.25 Activity ngưng hoạt động danh mục 88](#_Toc74244017)

[Hình 3.26 Sequence ngưng hoạt động danh mục 89](#_Toc74244018)

[Hình 3.27 Activity mở lại danh mục 91](#_Toc74244019)

[Hình 3.28 Sequence mở lại danh mục 92](#_Toc74244020)

[Hình 3.29 Activity ngưng hoạt động xe 94](#_Toc74244021)

[Hình 3.30 Sequence ngưng hoạt động xe 95](#_Toc74244022)

[Hình 3.31 Activity mở lại xe 97](#_Toc74244023)

[Hình 3.32 Sequence mở lại xe 98](#_Toc74244024)

[Hình 3.33 Activity chỉnh sửa thẻ bài viết 100](#_Toc74244025)

[Hình 3.34 Sequence chỉnh sửa thẻ bài viết 101](#_Toc74244026)

[Hình 3.35 Activity thêm thẻ bài viết 103](#_Toc74244027)

[Hình 3.36 Sequence thêm thẻ bài viết 104](#_Toc74244028)

[Hình 3.37 Activity thêm thẻ bài viết 106](#_Toc74244029)

[Hình 3.38 Sequence thêm thẻ bài viết 107](#_Toc74244030)

[Hình 3.39 Activity chỉnh sửa bài viết 109](#_Toc74244031)

[Hình 3.40 Sequence chỉnh sửa bài viết 110](#_Toc74244032)

[Hình 3.41 Activity chỉnh sửa danh mục xe 112](#_Toc74244033)

[Hình 3.42 Sequence chỉnh sửa danh mục xe 113](#_Toc74244034)

[Hình 3.43 Activity thêm danh mục xe 115](#_Toc74244035)

[Hình 3.44 Sequence thêm danh mục xe 116](#_Toc74244036)

[Hình 3.45 Activity thêm xe 118](#_Toc74244037)

[Hình 3.46 Sequence thêm xe 119](#_Toc74244038)

[Hình 3.47 Activity chỉnh sửa xe 121](#_Toc74244039)

[Hình 3.48 Sequence chỉnh sửa xe 122](#_Toc74244040)

[Hình 3.49 Activity thanh toán 124](#_Toc74244041)

[Hình 3.50 Sequence thanh toán 125](#_Toc74244042)

[Hình 3.51 Activity thuê xe 127](#_Toc74244043)

[Hình 3.52 Sequence thuê xe 128](#_Toc74244044)

[Hình 3.53 Activity xác nhận đặt trước 129](#_Toc74244045)

[Hình 3.54 Sequence xác nhận đặt trước 130](#_Toc74244046)

[Hình 3.55 Activity huỷ đặt trước 132](#_Toc74244047)

[Hình 3.56 Sequence huỷ đặt trước 133](#_Toc74244048)

[Hình 3.57 Activity đăng ký đặt trước 134](#_Toc74244049)

[Hình 3.58 Sequence đăng ký đặt trước 135](#_Toc74244050)

[Hình 3.59 Activity xem chi tiết hoá đơn 136](#_Toc74244051)

[Hình 3.60 Sequence xem chi tiết hoá đơn 137](#_Toc74244052)

[Hình 3.61 Activity xem bài viết 138](#_Toc74244053)

[Hình 3.62 Sequence đăng nhập 139](#_Toc74244054)

[Hình 3.63 Activity xem thông tin xe 141](#_Toc74244055)

[Hình 3.64 Sequence xem thông tin xe 142](#_Toc74244056)

[Hình 3.65 Activity tìm kiếm hoá đơn 144](#_Toc74244057)

[Hình 3.66 Sequence tìm kiếm hoá đơn 145](#_Toc74244058)

[Hình 3.67 Activity thống kê 146](#_Toc74244059)

[Hình 3.68 Sequence tìm kiếm hoá đơn 147](#_Toc74244060)

[Hình 3.67 Activity cập nhật thông tin 149](#_Toc74244061)

[Hình 3.68 Sequence tìm kiếm hoá đơn 150](#_Toc74244062)

[Hình 4.1 Class diagram 151](#_Toc74244063)

[Hình 4.2 Database diagram 153](#_Toc74244064)

**MỤC LỤC BẢNG BIỂU**

[Bảng 3.1 Danh sách tác nhân và mô tả 60](#_Toc74244065)

[Bảng 3.2 Danh sách các tình huống hoạt động (Use cases) 61](#_Toc74244066)

[Bảng 3.3 Mô tả Use case đăng nhập 68](#_Toc74244067)

[Bảng 3.4 Mô tả Use case thêm nhân viên 71](#_Toc74244068)

[Bảng 3.5 Mô tả Use case ngưng hoạt động nhân viên 74](#_Toc74244069)

[Bảng 3.6 Mô tả Use case đăng nhập 77](#_Toc74244070)

[Bảng 3.7 Mô tả Use case đăng nhập 80](#_Toc74244071)

[Bảng 3.8 Mô tả Use case ngưng hoạt động bài viết 83](#_Toc74244072)

[Bảng 3.9 Mô tả Use case mở lại bài viết 86](#_Toc74244073)

[Bảng 3.10 Mô tả Use case đăng nhập 89](#_Toc74244074)

[Bảng 3.11 Mô tả Use case mở lại danh mục 92](#_Toc74244075)

[Bảng 3.12 Mô tả Use case ngưng hoạt động xe 95](#_Toc74244076)

[Bảng 3.13 Mô tả Use case mở lại xe 98](#_Toc74244077)

[Bảng 3.14 Mô tả Use case chỉnh sửa thẻ bài viết 101](#_Toc74244078)

[Bảng 3.15 Mô tả Use case thêm thẻ bài viết 104](#_Toc74244079)

[Bảng 3.16 Mô tả Use case thêm bài viết 107](#_Toc74244080)

[Bảng 3.17 Mô tả Use case chỉnh sửa bài viết 110](#_Toc74244081)

[Bảng 3.18 Mô tả Use case chỉnh sửa danh mục xe 114](#_Toc74244082)

[Bảng 3.19 Mô tả Use case thêm danh mục xe 116](#_Toc74244083)

[Bảng 3.20 Mô tả Use case thêm xe 119](#_Toc74244084)

[Bảng 3.21 Mô tả Use case chỉnh sửa xe 122](#_Toc74244085)

[Bảng 3.22 Mô tả Use case thanh toán 125](#_Toc74244086)

[Bảng 3.23 Mô tả Use case thuê xe 128](#_Toc74244087)

[Bảng 3.24 Mô tả Use case xác nhận đặt trước 131](#_Toc74244088)

[Bảng 3.25 Mô tả Use case huỷ đặt trước 133](#_Toc74244089)

[Bảng 3.26 Mô tả Use case đăng ký đặt trước 136](#_Toc74244090)

[Bảng 3.27 Mô tả Use case xem chi tiết hoá đơn 138](#_Toc74244091)

[Bảng 3.28 Mô tả Use case xem bài viết 140](#_Toc74244092)

[Bảng 3.29 Mô tả Use case xem thông tin xe 142](#_Toc74244093)

[Bảng 3.30 Mô tả Use case tìm kiếm hoá đơn 145](#_Toc74244094)

[Bảng 3.30 Mô tả Use case thống kê 148](#_Toc74244095)

[Bảng 3.30 Mô tả Use case cập nhật thông tin 150](#_Toc74244096)

# GIỚI THIỆU

## Tổng quan

**Tên đề tài:** Ứng dụng cho thuê xe tập lái trên web app.

Xã hội ngày nay càng phát triển, nhu cầu ứng dụng CNTT vào đời sống thực tiễn ngày càng đa dạng và phong phú hơn. Trong thời đại số, mọi công việc trong cuộc sống đều có sự hỗ trợ của công nghệ nhằm tăng hiệu quả của công việc và tăng năng suất mà trước đây cần rất nhiều nguồn nhân lực mới có thể hoàn thành. Hơn thế nữa, “Cách mạng Công nghiệp 4.0” đang diễn ra tại nhiều nước phát triển và một trong số đó, là Việt Nam cũng đang nỗ lực để đắt kịp cơ hội quý báu này để thay đổi bộ mặt nền kinh tế đất nước. Có thể nói, ứng dụng công nghệ thông tin vào doanh nghiệp là một điều không thể thiếu đối với một doanh nghiệp muốn phát triển nhanh chóng.

Hiện nay, để có thể giảm thiểu chi phí, cải thiện hệ thống vận hành, tăng tỷ lệ dịch vụ tiếp xúc khách hàng mọi lúc mọi nơi. Việc có một trang web để giúp cho khách hàng tiếp cận tới dịch vụ thuê xe tập lái một cách tốt hơn, là một điểm rất tiện lợi bảo mật ,và đỡ mất thời gian hơn trong việc chờ đợi khi tới nơi đăng ký. Ngoài ra, khi khách hàng đặt trước trên biểu mẫu, dữ liệu sẽ được thông báo đến nhân viên. Giúp tiết kiệm thời gian cho khách hàng, tăng sự hài lòng của dịch vụ cho khách hàng. Ngoài ra trang web sẽ hỗ trợ nhân viên trong công việc, giúp tăng sự đổi mới cho nhân viên, giảm bớt thời gian như cập nhật dịch vụ, quản lý trở nên dễ dàng hơn.

## Mục tiêu đề tài

Nhằm giải quyết vấn đề đặt ra ở phần tổng quan đề tài, mục tiêu của nhóm là xây dựng một trang web giúp khách hàng có thể đặt trước cho việc thuê xe tập lái thông qua trình duyệt web trên mọi hệ điều hành bằng ngôn ngữ HTML, CSS, và JavaScript để thực hiện các chức năng của khách hàng như đăng ký đặt trước, xem danh sách xe, các bài viết, … Ngoài ra trang web còn hỗ trợ nhân viên các chức năng như thanh toán, nhận yêu cầu đặt trước từ phía khách hàng, quản lý hệ thống dữ liệu. Để có thể đồng bộ dữ liệu từ phía khách hàng, nhóm sử dụng kỹ thuật Server Sent Events hay một tên khác là Event Source cho phép kết nối thời gian thực giữa máy chủ và trình duyệt. Ở đây là sử dụng Server Sent Events API của Spring Boot framework cung cấp để hiện thực chương trình.

Trang web ở phía khách hàng sẽ thực hiện công việc như một giao diện giúp người dùng có thể xem các thông tin về xe có thể thuê, đăng ký đặt trước, và ngoài ra còn cung cấp các bài viết hữu ích về thông tin cách lái xe, thông tin về giao thông, mẹo, … Giúp khách hàng có thêm nhiều hiểu biết hơn và cũng như biết thêm các loại xe, giá cả hiện có của dịch vụ thuê xe tập lái. Khách hàng khi đặt trước chỉ cần điền biểu mẫu và gửi đi, nhân viên sẽ nhận yêu cầu và gọi tới khách hàng để xác nhận yêu cầu mà không cần tốn thời gian đợi hoặc đến tận nơi để đặt lịch.

Tại quầy tiếp nhận sẽ có một thiết bị chạy trang web của công ty truy cập đến máy chủ và sẽ tiếp nhận yêu cầu từ phía khách hàng gửi về. Ở website này sẽ yêu cầu quyền truy cập cấp nhân viên để đảm bảo tính bảo mật của hệ thống, cũng như quyền sử dụng trang web của người nhân viên, quản lý và admin. Ngoài ra, về phía quản lý website hỗ trợ quản lý xe, danh mục xe, bài viết và thẻ bài viết; về phía admin thì có thêm quản lý nhân viên. Website hỗ trợ thay đổi thông tin về email cũng như mật khẩu sau khi đăng nhập thành công.

Hướng phát triển có thể kể đến như nâng cấp dịch vụ đặt trước, cho phép khách hàng chọn xe mong muốn và giờ lấy xe; tích hợp thanh toán bằng ví điện tử, quét QR Code sau khi thuê xe tại quầy để tăng tính chuyên nghiệp và giúp khách hàng thanh toán mà không cần tiền mặt; nâng cấp trang web về mặt giao diện và các dịch vụ lớn hơn như cho phép người dùng đăng ký khoá học lái xe.

## Phạm vi đề tài

Xây dựng trang web về phía khách hàng cho phép đăng ký đặt trước, xem thông tin xe, danh sách loại xe, các bài viết, sử dụng ngôn ngữ HTML, CSS và JavaScript cùng trình soạn thảo Ckeditor để có thể viết bài. Ngoài ra còn sử dụng thêm các thư viện bên ngoài để giúp giao diện trở nên thân thiện với người dùng hơn như Boostrap, Jquery.

Xây dựng trang web về phía công ty bằng HTML, CSS và JavaScript cho phép

nhân viên tiếp nhận thông tin khách hàng đăng ký, đặt xe để thuê tập lái và thanh toán; quản lý sẽ quản lý thông tin danh mục xe, xe, bài viết, thẻ bài viết; admin sẽ quản lý thêm nhân viên và mọi chức năng trước đó.

Đề tài sử dụng Spring Boot Framework để làm Back-End, Mysql để lưu trữ dữ liệu. Sử dụng các dịch vụ của AWS gồm RDS để lưu trữ dữ liệu, EC2 để chạy Back-End, và S3 để lưu trữ các hình ảnh của trang web.

## **Mô tả yêu cầu chức năng**

Hình 1.1 Danh sách chức năng của hệ thống

# CƠ SỞ LÝ THUYẾT

## Web Application [1]

Trong ngành công nghệ thông tin, web application hay còn gọi là ứng dụng web là một trình ứng dụng có khả năng tiếp cận qua web thông qua mạng Internet hay intranet. Ứng dụng web phổ biến vì sự có mặt vào bất cứ nơi đâu trong một chương trình.

Khả năng cập nhật và bảo trì ứng dụng web nhanh chóng vì không cần phải phân phối và cài đặt phần mềm trên hàng ngàn máy tính khác nhau, đó chính là lý do làm cho web application trở nên phổ biến. Phần mềm web app được ứng dụng trong mail, bán hàng trực tuyến, diễn đàn hội thảo, blog, đấu giá trực tuyến, phần mềm ứng dụng quản lý các chức năng và nguồn nhân lực, hệ quản trị nội dung và một số ứng dụng khác.

### Giới thiệu về Web Application

Ứng dụng web chính là một phần mềm ứng dụng nền tảng web để chạy các phần mềm theo mong muốn và nhu cầu của người sử dụng. Thông qua ứng dụng web người dùng có thể thực hiện được một số công việc như sau: chia sẻ hình ảnh, mua sắm, phần mềm tính tiền thuận tiện cho người bán…vì web app có tính tương tác cao hơn website rất nhiều.

Đối với một số người dùng không rành về công nghệ thông tin, nên họ cứ nghĩ rằng những thứ online vào được bằng trình duyệt web thì đều được hiểu là website cả. Chính vì thế, người dùng thường yêu cầu: website bán hàng, website quản lý siêu thị,… nhưng thực chất thì chúng đều là web application.



Hình 2.1 Web Application

### Đặc điểm của web application

Do tính đặc thù nên ứng dụng web có nhiều đặc điểm khác biệt so với các website thông thường, cụ thể như sau:

* Tính tương tác cao và có nhiều chức năng như: upload file, đăng thông tin, xuất báo cáo,…
* Web app được tạo ra bởi HTML và code ở back end: PHP, C#, java,…
* Ứng dụng web được dùng để thực hiện một công việc hay một chức năng của một ứng dụng cụ thể.
* Web app yêu cầu tính quản trị lớn và độ khó cao.
* Phải xây dựng cơ sở dữ liệu đặc thù và phân tích hệ thống.
* Có thể tích hợp trình soạn thảo văn bản

Từ những đặc điểm nổi bật của ứng dụng web ở trên, chính vì thế khi thiết kế và lập trình ứng dụng web đòi hỏi người lập trình phải có trình độ chuyên môn kỹ thuật cao. Đặc biệt, công đoạn cực kỳ quan trọng và không thể thiếu trong quy trình sản xuất ứng dụng web đó chính là phân tích hệ thống. Đối với một website thông thường thì lập trình viên chỉ mất khoảng thời gian từ 2 đến 3 ngày, hoặc thậm chí chỉ là vài giờ để tạo ra. Tuy nhiên, đối với web app thì phải trải qua một thời gian dài để xây dựng mô hình cơ sở dữ liệu và phân tích hệ thống.

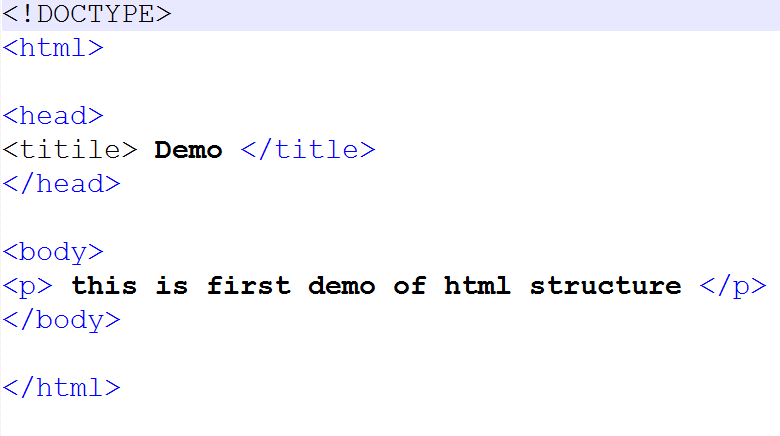
## Ngôn ngữ lập trình HTML [2]

### Tổng quan về ngôn ngữ HTML

HTML là chữ viết tắt của Hypertext Markup Language. Nó giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes, v.v…

HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được. Nó chỉ giống như Microsoft Word, dùng để bố cục và định dạng trang web.

Khi làm việc với HTML, chúng ta sẽ sử dụng cấu trúc code đơn giản (tags và attributes) để đánh dấu lên trang web. Ví dụ, chúng ta có thể tạo một đoạn văn bằng cách đặt văn bản vào trong cặp tag mở và đóng văn bản <p> và </p>.



Hình 2.2 Ví dụ về HTML

Tổng quan, HTML là ngôn ngữ markup, nó rất trực tiếp dễ hiểu, dễ học, và tất cả mọi người mới đều có thể bắt đầu học nó để xây dựng website.

### HTML hoạt động như thế nào

HTML documents là files kết thúc với đuôi .html hay .htm. Bạn có thể xem chúng bằng cách sử dụng bất kỳ trình duyệt web nào (như Google Chrome, Safari, hay Mozilla Firefox). Trình duyệt đọc các files HTML này và xuất bản nội dung lên internet sao cho người đọc có thể xem được nó.

Thông thường, trung bình một web chứa nhiều trang web HTML, ví dụ như: trang chủ, trang about, trang liên hệ, tất cả đều cần các trang HTML riêng.

Mỗi trang HTML chứa một bộ các tag (cũng được gọi là elements), bạn có thể xem như là việc xây dựng từng khối của một trang web. Nó tạo thành cấu trúc cây thư mục bao gồm section, paragraph, heading, và những khối nội dung khác.

Hầu hết các HTML elements đều có tag mở và tag đóng với cấu trúc như <tag></tag>.

### Các ưu điểm giúp HTML được sử dụng rộng rãi

* Ngôn ngữ được sử dụng rộng lớn này có rất nhiều nguồn tài nguyên hỗ trợ và cộng đồng sử dụng cực lớn.
* Sử dụng mượt mà trên hầu hết mọi trình duyệt.
* Có quá trình học đơn giản và trực tiếp.
* Mã nguồn mở và hoàn toàn miễn phí.
* Markup gọn gàng và đồng nhất.
* Chuẩn chính của web được vận hành bởi World Wide Web Consortium (W3C).
* Dễ dàng tích hợp với các ngôn ngữ backend như PHP và Node.js.

## Ngôn ngữ lập trình CSS [3]

### Tổng quan về CSS

CSS là ngôn ngữ tạo phong cách cho trang web – Cascading Style Sheet language. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là HTML. Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người viết web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ.

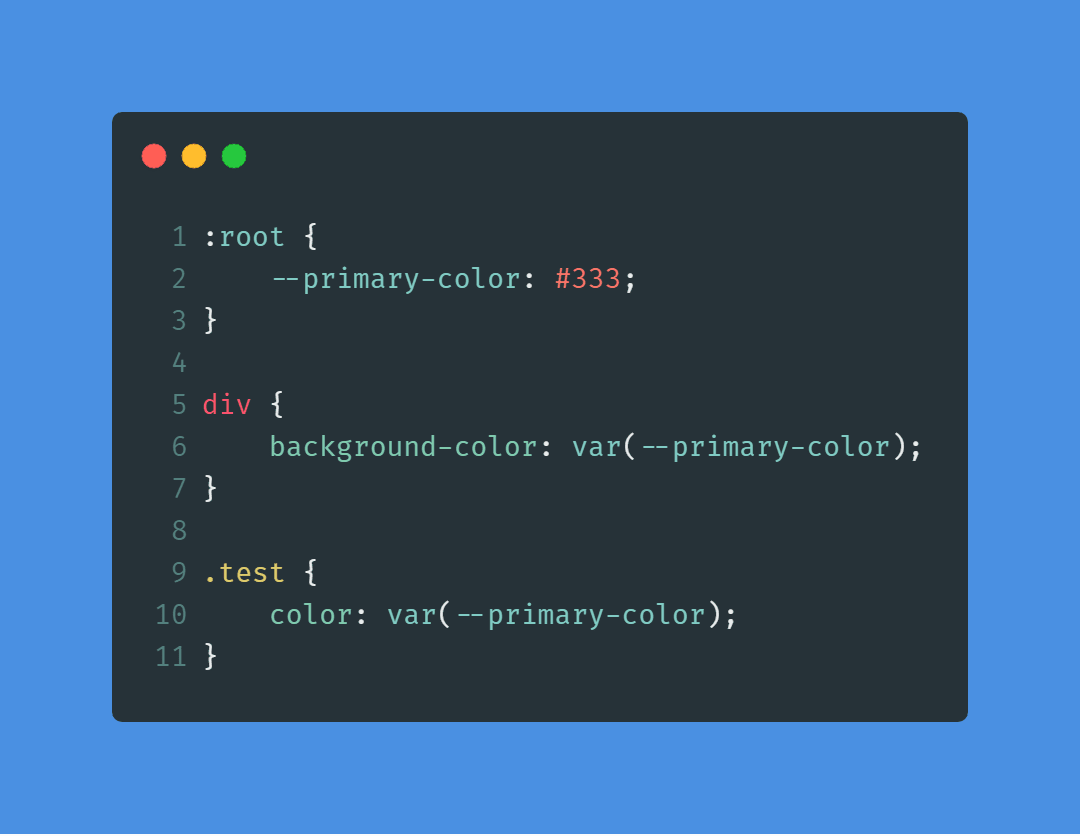
CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì một lý do đơn giản. HTML không được thiết kế để gắn tag để giúp định dạng trang web. Bạn chỉ có thể dùng nó để “đánh dấu” lên site.

Những tag như <font> được ra mắt trong HTML phiên bản 3.2, nó gây rất

nhiều rắc rối cho lập trình viên. Vì website có nhiều font khác nhau, màu nền và phong cách khác nhau. Để viết lại code cho trang web là cả một quá trình dài, cực nhọc. Vì vậy, CSS được tạo bởi W3C là để giải quyết vấn đề này.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

CSS về lý thuyết không có cũng được, nhưng khi đó website sẽ không chỉ là một trang chứa văn bản mà không có gì khác.



Hình 2.3 Cấu trúc của CSS

### CSS hoạt động như thế nào

CSS sử dụng cấu trúc tiếng Anh đơn giản để tạo ra một bộ các quy tắc bạn có thể tận dụng. Như đã nói ở trên, HTML không được dùng để tạo phong cách cho các yếu tố, nó chỉ đánh dấu từng phần để biết được yếu tố đó là gì thôi. Ví dụ: <p>Đây là văn bản.</p>.

Còn làm thế nào để tạo ra phong cách cho văn bản đó? Syntax của CSS rất đơn giản. Nó có phần block chọn và block khai báo. Bạn chọn một yếu tố và khai báo làm gì với nó. Rất đơn giản phải không?

Tuy nhiên, cũng có nhiều quy tắc cần ghi nhớ.

Selector sẽ trỏ về yếu tố HTML bạn cần muốn tạo phong cách. Block khai báo sẽ bao gồm một hay nhiều khai báo cách nhau bởi dấu chấm phẩy,

Mỗi khai báo bao gồm một tên CSS và giá trị, cách nhấu bởi dấu 2 chấm. Khai báo CSS luôn kết thúc bằng dấu chấm phẩn, và block khai báo được đặt trong dấu ngoặc nhọn.

Hãy xem qua ví dụ:



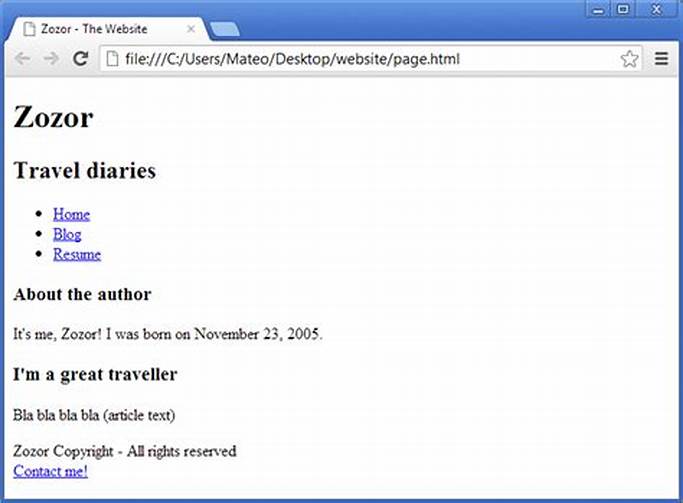
Hình 2.4 Ví dụ về CSS

Tất cả các thành phần trong element <p> sẽ dùng màu xanh và in đậm.

### Ưu điểm của CSS

Sự khác biệt giữa site có CSS và không có CSS rất dễ nhận biết.

Chắc bạn đã thấy rồi, khi website không load được đầy đủ và chỉ có nền trắng và chữ thì chỉ có màu đen và xanh. Như sau:



Hình 2.4. Trang Web khi không có CSS

Điều này có nghĩa là thành phần CSS của bạn không tải lên được hay nó không tồn tại.

Đó một website không có CSS, vậy bạn có muốn website của mình trông như thế không? Hẵn là không rồi phải không.

Trước khi sử dụng CSS, tất cả những phong cách của CSS cần được đính kèm vào trong HTML markup. Có nghĩa là bạn cần tách ra để xác định các thành phần như background, font colors, canh hàng, vâng vâng.

CSS giúp bạn định kiểu mọi thứ trên một file khác, bạn có thể tạo phong cách trước rồi sau đó tích hợp file CSS lên trên cùng của file HTML. Việc này giúp HTML markup rõ ràng và dễ quản lý hơn nhiều.

Tóm lại,với CSS bạn không cần lặp lại các mô tả cho từng thanh phần. Nó tiết kiệm thời gian, làm code ngắn lại để bạn có thể kiểm soát lỗi dễ dàng hơn.

CSS giúp bạn có nhiều styles trên một trang web HTML, vì vậy, khả năng điều chỉnh trang gần như vô hạn?

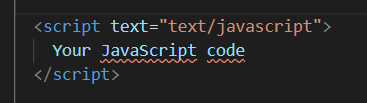
## Ngôn ngữ lập trình JavaScript [4]

### Tổng quan về JavaScript

JavaScript là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua. Nó cũng là một trong số 3 ngôn ngữ chính của lập trình web.

JavaScript liên tục phát triển kể từ đó, có lục đạt đến 92% website đang sử dụng JavaScript vào năm 2016. Chỉ trong 20 năm, nó từ một ngôn ngữ lập trình riêng trở thành công cụ quan trọng nhất trên bộ công cụ của các chuyên viên lập trình web. Nếu bạn đang dùng internet, vậy chắc chắn bạn đã từng sử dụng JavaScript rồi.

Cách chèn JavaScript đơn giản nhất là ví dụ dưới đây



Hình 2.5 Ví dụ về JavaScript

JavaScript là ngôn ngữ lập trình mang đến sự sinh động của website. Nó khác với HTML (thường chuyên cho nội dung) và CSS (thường chuyên dùng cho phong cách), và khác hẵn với PHP (chạy trên server chứ không chạy dưới máy client).

Bạn cần biết gì:

* JavaScript là ngôn ngữ dễ học;
* Nó được phát triển bởi Netscape, và đang được dùng trên 92% webstie;
* JS có thể được gắn vào một element của trang web hoặc sự kiện của trang web như cú click chuột;
* Hoạt động trên đa trình duyệt và đa thiết bị;
* Nhanh và nhẹ hơn các ngôn ngữ lập trình khác;
* Có thể ít an toàn hơn vì độ phổ biến của nó;
* Bạn có thể thêm JavaScript trực tiếp vào HTML hoặc bạn có thể lưu nó trên files riêng biệt và gọi lên khi cần.

### Cách JavaScript hoạt động

JavaScript thường được nhúng trực tiếp vào một trang web hoặc được tham chiếu qua file .js riêng. Nó là ngôn ngữ phía client, tức là script được tải về máy của khách truy cập và được xử lý tại đó thay vì phía server là xử lý trên server rồi mới đưa kết quả tới khách truy cập.

Hãy lưu ý là các trình duyệt web phổ biến cũng hỗ trợ việc người dùng có muốn tắt JavaScript hay không. Đó là lý do bạn nên biết trang web sẽ hoạt động như thế nào torng trường hợp không có JavaScript.

### Ưu điểm của JavaScript

JavaScript có rất nhiều ưu điểm khiến nó vượt trội hơn so với các đối thủ, đặc biệt trong các trường hợp thực tế. Sau đây chỉ là một số lợi ích của JavaScript:

* Bạn không cần một compiler vì web browser có thể biên dịch nó bằng HTML;
* Nó dễ học hơn các ngôn ngữ lập trình khác;
* Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn;
* Nó có thể được gắn trên một số element của trang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới;
* JS hoạt động trên nhiều trình duyệt, nền tảng, vâng vâng;
* Bạn có thể sử dụng JavaScript để kiểm tra input và giảm thiểu việc kiểm tra thủ công khi truy xuất qua database;
* Nó giúp website tương tác tốt hơn với khách truy cập;
* Nó nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác.

## Ngôn ngữ lập tình Java [5]

### Java là gì?

Java là một một ngôn ngữ lập trình hiện đại, bậc cao, hướng đối tượng, bảo mật và mạnh mẽ. và là một Platform.

Platform: Bất cứ môi trường phần cứng hoặc phần mềm nào mà trong đó có một chương trình chạy, thì được hiểu như là một Platform. Với môi trường runtime riêng cho mình (JRE) và API, Java được gọi là Platform.

Ngôn ngữ lập trình Java ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995. Phiên bản mới nhất của Java Standard Edition là Java SE 8. Với sự tiến bộ của Java và sự phổ biến rộng rãi của nó, nhiều cấu hình đã được xây dựng để phù hợp với nhiều loại nền tảng khác nhau. Ví dụ: J2EE cho các ứng dụng doanh nghiệp, J2ME cho các ứng dụng di động.

Các phiên bản J2 mới đã được đổi tên thành Java SE, Java EE và Java ME. Phương châm của java là "Write Once, Run Anywhere" - viết một lần chạy nhiều nơi, nghĩa là bạn chỉ cần viết một lần trên window chẳng hạn, sau đó vẫn chương trình đó bạn có thể chạy trên Linux, Android, các thiết bị J2ME...



Hình 2.6 Java

### Các tính năng của Java

Ngôn ngữ lập trình java có các tính năng sau:

* Hướng đối tượng - Trong Java, mọi thứ đều là một Object. Java có thể dễ dàng mở rộng và bảo trì vì nó được xây dựng dựa trên mô hình Object.
* Nền tảng độc lập - Không giống nhiều ngôn ngữ lập trình khác bao gồm cả C và C ++, khi Java được biên dịch, nó không được biên dịch thành ngôn ngữ máy nền tảng cụ thể, thay vào mã byte - nền tảng độc lập. Mã byte này được thông dịch bởi máy ảo (JVM) trên nền tảng nào đó mà nó đang chạy.
* Đơn giản - Java được thiết kế để dễ học. Nếu bạn hiểu khái niệm cơ bản về OOP Java, sẽ rất dễ để trở thành master về java.
* Bảo mật - Với tính năng an toàn của Java, nó cho phép phát triển các hệ thống không có virut, giả mạo. Các kỹ thuật xác thực dựa trên mã hoá khóa công khai.
* Kiến trúc - trung lập - Trình biên dịch Java tạo ra định dạng tệp đối tượng kiến trúc trung lập, làm cho mã biên dịch được thực thi trên nhiều bộ vi xử lý, với sự hiện diện của hệ điều hành Java.
* Portable - Là kiến trúc tập trung và không có khía cạnh thực hiện phụ thuộc của đặc tả này làm cho Java khả chuyển. Trình biên dịch trong Java được viết bằng ANSI C, đó là một tập con POSIX.
* Mạnh mẽ - Java làm nỗ lực để loại trừ các tình huống dễ bị lỗi bằng cách kiểm tra lỗi tại thời gian biên dịch và kiểm tra lỗi tại runtime.
* Đa luồng - Với tính năng đa luồng của Java có thể viết các chương trình có thể thực hiện nhiều tác vụ đồng thời. Tính năng thiết kế này cho phép các nhà phát triển xây dựng các ứng dụng tương tác có thể chạy trơn tru hơn.
* Thông dịch - Mã byte Java được dịch trực tiếp tới các máy tính gốc và không được lưu trữ ở bất cứ đâu.
* Hiệu năng cao - Với việc sử dụng trình biên dịch Just-In-Time, Java cho phép thực hiện hiệu năng cao.
* Phân tán - Java được thiết kế cho môi trường phân tán của Internet.
* Năng động - Java là năng động hơn C hoặc C++ vì nó được thiết kế để thích nghi với môi trường đang phát triển. Các chương trình Java có thể mang một lượng lớn thông tin tại runtime mà có thể được sử dụng để xác minh và giải quyết các truy cập vào các đối tượng tại runtime.

### Java được dùng để làm gì?

Java rất phổ biến và đã thống trị lĩnh vực này từ đầu những năm 2000 đến nay 2020.

Theo tập đoàn SUN, hiện nay có khoảng 3 tỷ thiết bị đang chạy java.

Java đã được sử dụng trong các lĩnh vực khác nhau. Ví dụ:

* Desktop App như acrobat reader, media player, antivirus, ...
* Web App như irctc.co.in, javatpoint.com, ...
* Enterprise App như các ứng dụng về xử lý nghiệp vụ ngân hàng, ...
* Thiết bị Mobile như các ứng dụng Android.
* Hệ thống nhúng
* Smart Card
* Robot
* Game App

## Spring Boot trong Spring Framework

### Spring là gì? [6]

Spring là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…

Spring nhẹ và trong suốt (nhẹ: kích thước nhỏ, version cơ bản chỉ khoảng 2MB; trong suốt: hoạt động một cách trong suốt với lập trình viên)

Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lơn.

Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: Dependency Injection và Aspect Oriented Programming.

Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object)

### Kiến trúc và các module của Spring Framework

Spring được chia làm nhiều module khác nhau, tùy theo mục đích phát triển ứng dụng mà ta dùng 1 trong các module đó.

Dưới đây là kiến trúc tổng thể của Spring Framework.



Hình 2.7 Kiến trúc và module của Spring Framework

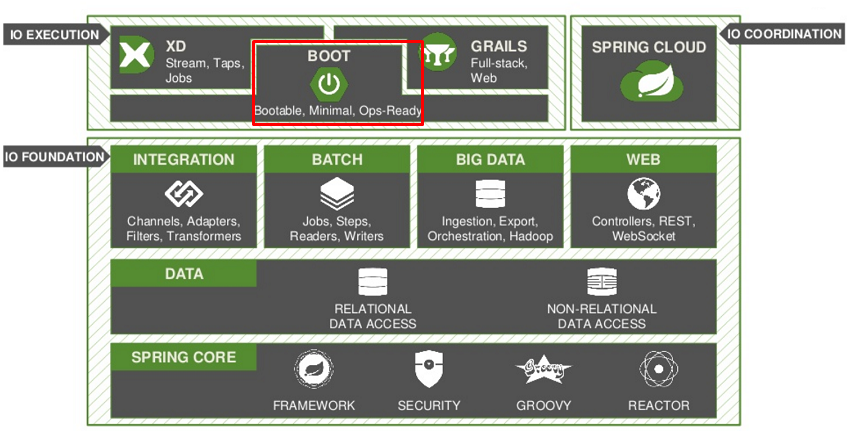
### Spring Boot là gì? [7]

Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) – Phát triển ứng dụng nhanh.

Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring.

Spring Boot không yêu cầu cấu hình XML

Nó là một chuẩn cho cấu hình thiết kế phần mềm, tăng cao năng suất cho developer.



Hình 2.8 Spring Boot

### Ưu điểm của Spring Boot

Có các tính năng của Spring Framework.

Tạo ứng dụng độc lập, có thể chạy bằng java -jar (cho cả java web)

Nhúng trực tiếp các ứng dụng server (Tomcat, Jetty…) do đó không cần phải triển khai file WAR

Cấu hình ít, tự động cậu hình bất kì khi nào có thể (Giảm thời gian viết code, tăng năng suất)

Không yêu cầu XML config…

Cung cấp nhiều plugin

Chuẩn cho Microservices (Cloud support; giảm việc setup, config; các thư viện hỗ trợ…)

## RESTful API [8]

### RESTful API là gì?

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 2.9 RESTful API Design

### Cách thức hoạt động của RESTful API



Hình 2.10 Mô hình hoạt động của RESTful API

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.

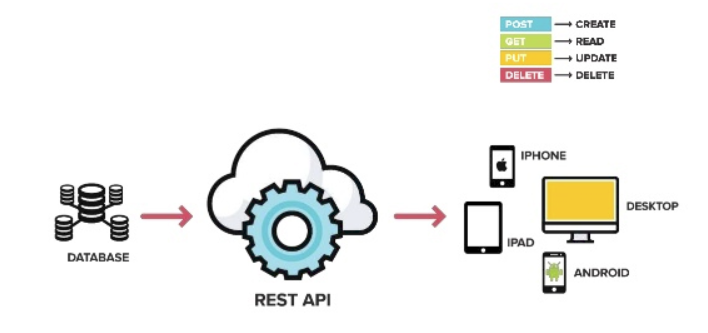
Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

Ngoài ra RESTful API còn hỗ trợ xác thực đối tượng và trả về kết quả của giao thức HTTP

### Ưu điểm của RESTful API

Một số ưu điểm chính khi sử dụng RESTFUL API là:

* Giúp cho ứng dụng rõ ràng hơn
* REST URL đại diện cho resource chứ không phải hành động
* Dữ liệu được trả về với nhiều định dạng khác nhau như: xml, html, json….
* Code đơn giản và ngắn gọn
* REST chú trọng vào tài nguyên của hệ thống



Hình 2.11 RESTful API truyền tải dữ liệu

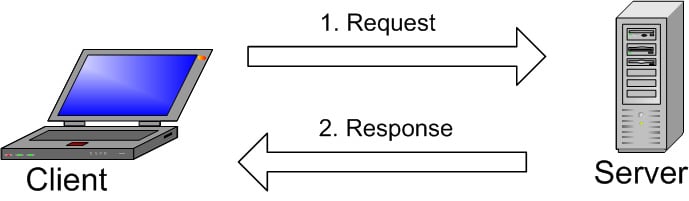
## Hệ quản trị cơ sở dữ liệu MySQL [9]

### MySQL là gì?

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

MySQL là một trong số các phần mềm RDBMS. RDBMS và MySQL thường được cho là một vì độ phổ biến quá lớn của MySQL. Các ứng dụng web lớn nhất như Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Kể cả khi ban đầu nó chỉ được dùng rất hạn chế nhưng giờ nó đã tương thích với nhiều hạ tầng máy tính quan trọng như Linux, macOS, Microsoft Windows, và Ubuntu.

### Cách thức hoạt động của MySQL



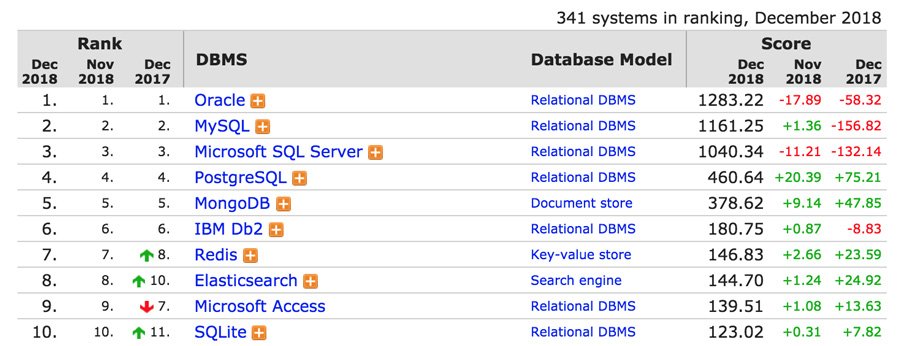
Hình 2.12 Cách thức MySQL hoạt động

Hình ảnh trên giải thích cấu trúc cơ bản về việc giao tiếp giữa client-server model. Một máy client sẽ liên lạc với máy server trong một mạng nhất định. Mỗi client có thể gửi một request từ giao diện người dùng (Graphical user interface – GUI) trên màn hình, và server sẽ trả về kết quả như mong muốn. Miễn là cả hai hiểu nhau. Cách vận hành chính trong môi trường MySQL cũng như vậy:

* MySQL tạo ra bảng để lưu trữ dữ liệu, định nghĩa sự liên quan giữa các bảng đó.
* Client sẽ gửi yêu cầu SQL bằng một lệnh đặc biệt trên MySQL.
* Ứng dụng trên server sẽ phản hồi thông tin và trả về kết quả trên máy client.

Có vậy thôi đó. Từ máy client, việc chọn GUI MySQL khá quan trọng. GUI càng nhẹ chừng nào, thì các thao tác quản lý data sẽ càng dễ dàng và nhanh chừng đó. MySQL GUI phổ biến nhất MySQL WorkBench, SequelPro, DBVisualizer, và Navicat DB Admin Tool. Một vài trong số chúng miễn phí, một vài bản thương mại, một vài bản chỉ chạy được trên macOS, và một vài ứng dụng chạy được hết trên các hệ điều hành phổ biến. Clients nên chọn GUI tùy vào nhu cầu của họ.

### Vì sao MySQL lại phổ biến



Hình 2.13 Bảng xếp hạng các hệ quản trị cơ sở dữ liệu 2018

Linh hoạt và dễ dùng: Bạn có thể sửa source code để đáp ứng nhu cầu của bạn mà không phải thanh toán têm bất kỳ chi phí nào. Quá trình cài đặt cũng rất đơn giản và thường không quá 30 phút.

Hiệu năng cao: Nhiều server clusters sử dụng MySQL. Bất kể bạn lưu trữ dữ liệu lớn của các trang thương mại điện tử hoặc những hoạt động kinh doanh nặng nề liên quan đến công nghệ thông tin, MySQL cũng có thể đáp ứng được với tốc độ cao, mượt mà.

Tiêu chuẩn trong ngành: Ngành công nghệ và dữ liệu đã sử dụng MySQL nhiều năm, vì vậy nó là một kỹ năng căn bản một chuyên gia lập trình. Người dùng MySQL cũng có thể triển khai dự án nhanh và thuê các chuyên gia dữ liệu với mức phí nếu họ cần.

An toàn: An toàn dữ liệu luôn là vấn đề quan trọng nhất khi chọn phần mềm RDBMS. Với hệ thống phân quyền truy cập và quản lý tài khoản, MySQL đặt tiêu chuẩn bảo mật rất cao. Mã hóa thông tin đăng nhập và chứng thực từ host đều khả dụng.

## AWS là gì và các dịch vụ sử dụng trong đề tài

### AWS là gì? [12]

Amazon Web Services (AWS) là nền tảng đám mây toàn diện và được sử dụng rộng rãi nhất, cung cấp trên 200 dịch vụ đầy đủ tính năng từ các trung tâm dữ liệu trên toàn thế giới. Hàng triệu khách hàng—bao gồm các công ty khởi nghiệp tăng trưởng nhanh nhất, các tập đoàn lớn nhất cũng như các cơ quan hàng đầu của chính phủ—đều tin tưởng vào AWS để giảm chi phí, trở nên linh hoạt hơn và đổi mới nhanh hơn.

### Các ưu điểm của AWS

**Đầy đủ chức năng nhất**

* AWS có nhiều dịch vụ hơn và nhiều tính năng hơn trong các dịch vụ đó, hơn bất kỳ nhà cung cấp đám mây nào khác–từ các công nghệ cơ sở hạ tầng như máy tính, ổ lưu trữ và cơ sở dữ liệu–đến các công nghệ mới nổi, như machine learning và trí tuệ nhân tạo, kho dữ liệu và phân tích, và Internet of Things. Điều này giúp bạn chuyển các ứng dụng hiện có của bạn lên đám mây nhanh hơn, dễ dàng hơn và hiệu quả hơn về chi phí cũng như xây dựng gần như mọi thứ bạn có thể hình dung.
* AWS cũng có chức năng sâu nhất trong các dịch vụ đó. Ví dụ, AWS cung cấp nhiều loại cơ sở dữ liệu nhất được xây dựng có mục đích cho các loại ứng dụng khác nhau để bạn có thể chọn công cụ phù hợp cho công việc để có chi phí và hiệu suất tốt nhất.

**Cộng đồng khách hàng và đối tác lớn nhất**

* AWS có cộng đồng lớn và linh hoạt nhất với hàng triệu khách hàng hoạt động và hàng chục ngàn đối tác trên toàn cầu. Khách hàng trên hầu hết các ngành công nghiệp thuộc mọi quy mô, bao gồm công ty khởi nghiệp, doanh nghiệp và tổ chức công, đang chạy mọi trường hợp sử dụng có thể diễn ra trên AWS. Mạng lưới đối tác AWS (APN) bao gồm hàng nghìn đơn vị tích hợp hệ thống chuyên về dịch vụ AWS và hàng chục nghìn nhà cung cấp phần mềm độc lập (ISV) điều chỉnh công nghệ của mình để hoạt động trên AWS.

**Bảo mật nhất**

* AWS được thiết kế để trở thành môi trường điện toán đám mây bảo mật và linh hoạt nhất hiện nay. Cơ sở hạ tầng cốt lõi của chúng tôi được xây dựng để đáp ứng các yêu cầu bảo mật cho quân đội, ngân hàng toàn cầu và các tổ chức khác với mức độ nhạy cảm cao. Điều này được hỗ trợ bởi một bộ công cụ bảo mật trên đám mây chuyên sâu, với 230 tính năng cũng như dịch vụ bảo mật, tuân thủ và quản trị. AWS hỗ trợ 90 tiêu chuẩn bảo mật và chứng nhận tuân thủ. Tất cả 117 dịch vụ AWS lưu trữ dữ liệu khách hàng đều cung cấp khả năng mã hóa các dữ liệu đó.

**Tốc độ đổi mới nhanh nhất**

* Với AWS, bạn có thể tận dụng các công nghệ mới nhất để thử nghiệm và đổi mới nhanh hơn. Chúng tôi liên tục đẩy nhanh tốc độ đổi mới để phát minh ra các công nghệ hoàn toàn mới mà bạn có thể sử dụng để chuyển đổi doanh nghiệp của mình. Ví dụ, vào năm 2014, AWS đã tiên phong trong không gian điện toán không có máy chủ với việc ra mắt AWS Lambda, cho phép các nhà phát triển chạy mã của họ mà không cần cung cấp hoặc quản lý máy chủ. Và AWS đã xây dựng Amazon SageMaker, một dịch vụ machine learning được quản lý hoàn toàn, trao quyền cho các nhà phát triển và nhà khoa học mỗi ngày sử dụng machine learning–mà không cần bất kỳ kinh nghiệm nào trước đó.

**Chuyên môn vận hành đã được chứng minh là tốt nhất**

* AWS có kinh nghiệm, sự trưởng thành, độ tin cậy, bảo mật và hiệu suất chưa từng có mà bạn có thể tin tưởng cho các ứng dụng quan trọng nhất của mình. Trong hơn 15 năm, AWS đã cung cấp dịch vụ đám mây cho hàng triệu khách hàng trên toàn thế giới, phục vụ nhiều trường hợp sử dụng đa dạng. AWS có nhiều kinh nghiệm vận hành nhất, trên quy mô lớn hơn so với bất kỳ nhà cung cấp đám mây nào.

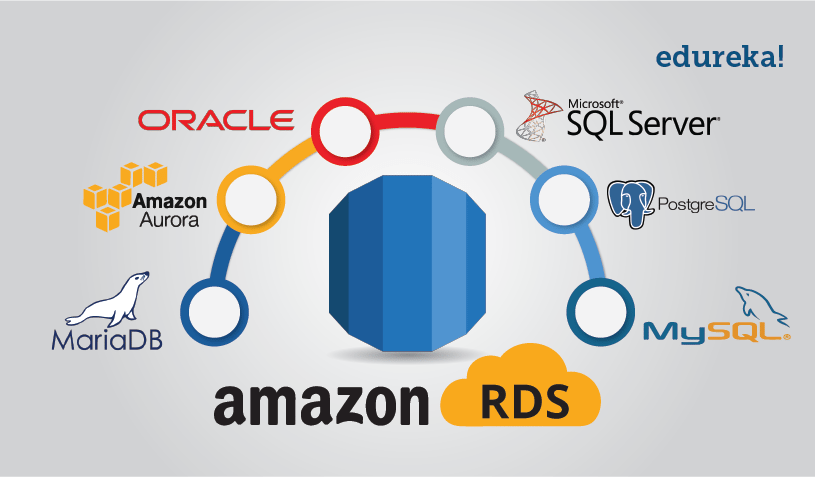


Hình 2.14 Amazon Web Services

### Amazon RDS là gì? [13]

Amazon Relational Database Service (Amazon RDS) là một dịch vụ được quản lý giúp bạn dễ dàng thiết lập, vận hành và thay đổi quy mô cơ sở dữ liệu quan hệ trên đám mây. Dịch vụ này cung cấp tính năng hiệu quả về chi phí và có thể thay đổi quy mô, trong khi vẫn quản lý được các tác vụ quản trị cơ sở dữ liệu tiêu tốn nhiều thời gian, giúp bạn thoải mái tập trung vào các ứng dụng và công việc của bạn.

Amazon RDS cho phép bạn truy cập các tính năng của một cơ sở dữ liệu quen thuộc như MySQL, MariaDB, Oracle, SQL Server hay PostgreSQL. Điều này có nghĩa là mã, ứng dụng và công cụ bạn đang sử dụng ngày nay cho cơ sở dữ liệu hiện tại của bạn sẽ hoạt động trơn tru với Amazon RDS. Amazon RDS có thể tự động sao lưu dự phòng cơ sở dữ liệu và cập nhật phần mềm cơ sở dữ liệu của bạn lên phiên bản mới nhất. Bạn hưởng lợi từ tính linh hoạt, đó là khả năng dễ dàng thay đổi quy mô tài nguyên điện toán hoặc dung lượng lưu trữ được liên kết với phiên bản cơ sở dữ liệu quan hệ của bạn. Ngoài ra, Amazon RDS còn tạo điều kiện thuận lợi để sử dụng bản sao để tăng cường độ khả dụng cơ sở dữ liệu, cải thiện độ bền dữ liệu hoặc thay đổi quy mô ra khỏi giới hạn công suất của một phiên bản cơ sở dữ liệu đơn lẻ cho khối lượng công việc cơ sở dữ liệu đọc nhiều. Đối với toàn bộ Amazon Web Services, không cần phải đầu tư ban đầu và bạn chỉ phải trả tiền cho tài nguyên bạn sử dụng.



Hình 2.15 Amazon RDS

### Các tính năng của Amazon RDS [13]

Giảm nhẹ gánh nặng quản trị

* Dễ sử dụng: Phiên bản cơ sở dữ liệu Amazon RDS được cấu hình sẵn với các thông số và cài đặt phù hợp với công cụ và lớp do bạn lựa chọn.
* Tự động vá lỗi phần mềm: Amazon RDS sẽ đảm bảo rằng phần mềm cơ sở dữ liệu quan hệ hiện đang chạy đợt triển khai của bạn được cập nhật theo bản vá lỗi mới nhất. Bạn có thể lựa chọn kiểm soát thêm khi và nếu phiên bản cơ sở dữ liệu của bạn được vá lỗi.
* Khuyến cáo thực hành tốt nhất: Amazon RDS cung cấp hướng dẫn thực hành tốt nhất bằng cách phân tích cấu hình và thông số sử dụng từ các phiên bản cơ sở dữ liệu của bạn.

Hiệu năng

* Bộ lưu trữ mục đích chung (SSD): Bộ lưu trữ mục đích chung của Amazon RDS là một tùy chọn lưu trữ trên ổ SSD, đảm bảo mức hiệu năng cơ sở ổn định bằng 3 IOPS trên mỗi GB được cung cấp và cho phép tăng vượt mức cơ sở lên tới 3.000 IOPS.
* Dung lượng lưu trữ (SSD) IOPS dự phòng: Lưu trữ IOPS dự phòng của Amazon RDS là một tùy chọn lưu trữ trên ổ SSD, được thiết kế để mang lại hiệu năng I/O nhanh, có thể dự báo được và ổn định.

Khả năng thay đổi quy mô

* Thay đổi quy mô điện toán bằng một nút nhấn
* Dễ dàng thay đổi quy mô lưu trữ
* Bản sao chỉ có quyền đọc

Độ khả dụng và độ bền cao

* Sao lưu tự động
* Bản kết xuất nhanh cơ sở dữ liệu
* Triển khai Nhiều vùng sẵn sàng
* Tự động thay thế máy chủ lưu trữ

Bảo mật

* Mã hóa khi đang lưu trữ và khi đang truyền
* Cách ly mạng
* Quyền ở cấp độ tài nguyên

Dễ quản lý

* Giám sát và thông số: Amazon RDS cung cấp số liệu Amazon CloudWatch dành cho phiên bản cơ sở dữ liệu của bạn và không tính thêm phí. Bạn có thể sử dụng Bảng điều khiển quản lý RDS để xem các số liệu vận hành chính, bao gồm mức sử dụng công suất điện toán/bộ nhớ/lưu trữ, hoạt động I/O và kết nối phiên bản.
* Thông báo sự kiện: Amazon RDS có thể thông báo cho bạn biết qua email hoặc tin nhắn văn bản SMS về các sự kiện cơ sở dữ liệu thông qua Amazon SNS.
* Quản lý cấu hình: Amazon RDS tích hợp với AWS Config để hỗ trợ tuân thủ và tăng cường bảo mật bằng cách ghi lại và kiểm tra các thay đổi về cấu hình phiên bản CSDL của bạn, bao gồm nhóm thông số, nhóm mạng con, bản kết xuất nhanh, nhóm bảo mật và đăng ký nhận thông báo sự kiện.

Hiệu quả chi phí

* Phiên bản dự trữ
* Dừng và bắt đầu sử dụng: Amazon RDS cho phép bạn dễ dàng dừng và tiếp tục sử dụng phiên bản cơ sở dữ liệu trong tối đa 7 ngày mỗi lần

### Amazon EC2 là gì? [14]

Amazon Elastic Compute Cloud (Amazon EC2) là một cơ sở hạ tầng điện toán đám mây được cung cấp bởi Amazon Web Services (AWS) giúp cung cấp tài nguyên máy tính ảo hoá theo yêu cầu.

Amazon EC2 cung cấp các ứng dụng máy tính ảo hoá có thể mở rộng về khả năng xử lý cùng các thành phần phần cứng ảo như bộ nhớ máy tính (ram), vi xử lý, linh hoạt trong việc lựa chọn các phân vùng lưu trữ dữ liệu ở các nền tảng khác nhau và sự an toàn trong quản lý dịch vụ bởi kiến trúc ảo hoá đám mây mạnh mẽ của AWS.

Amazon EC2 sẽ cung cấp một hoặc máy chủ ảo có thể kết hợp với nhau để dễ dàng triển khai ứng dụng nhanh nhất và đảm bảo tính sẵn sàng cao nhất. Thậm chí về mặt thanh toán bạn dễ dàng biết được các mức chi phí cần thanh toán dựa trên thông tin tài nguyên bạn sử dụng.



Hình 2.16 Amazon EC2

### Các đặc điểm của Amazon EC2 [10]

Scaling:

* Scaling Up/Down: Tăng/Giảm capacity(RAM, CPU,...) của Instance.
* Scaling In/Out: Tăng/Giảm số lượng Instance.

Security:

* Có thể thiết lập rank IP Private dành riêng cho EC2.
* Sử dụng Security Group và Network ACLS để control inbound/outbound.
* Có thể thiết lập IPsec VPN giữa Data Center và AWS Clound.
* Delicated Instance -> Tạo EC2 trên 1 hardware physical dành riêng cho 1 khách hàng duy nhất.

Cost:

* On-Demand Instance: Tính theo giờ, đáp ứng nhu cầu dùng trong thời gian ngắn. Dùng bao nhiêu, trả bấy nhiêu.
* Reserved Instance: Cho phép trả trước 1 lượng Server cho 1 hoặc 3 năm. Chi phí chỉ bằng 75% so với On-Demand. Nêú không có nhu cầu sử dụng trong thời gian dài, bạn có thể sale tại Reserved Instance Marketplace

## API jQuery Ajax [11]

### Ajax là gì?

Ajax = Asynchoronous JavaScript và XML

Nói ngắn gọn , Ajax là một bộ công cụ cho phép load dữ liệu từ server mà không yêu cầu tải lại trang.Nó sử dụng chức năng sẵn có XMLHttpRequest(XHR) của trình duyệt để thực hiện một yêu cầu đến server và xử lý dữ liệu server trả về.

Lấy một ví dụ như sau : khi một người dùng viết một nhận xét trên bài viết đăng trên trang Facebook. Sau khi người dùng gửi nhận xét thành công trang Facebook mà người đó đang truy cập cần phải được cập nhật để hiển thị nhận xét vừa mới được tạo ra này. Nếu load lại toàn bộ trang mà người dùng đang truy cập thì sẽ không hiệu quả do tất cả những gì chúng ta muốn là hiển thị nhận xét mới được tạo ra, Ajax được tạo ra để giải quyết vấn đề này, thay vì tải lại toàn bộ trang trình duyệt sẽ chỉ l những phần được thay đổi để tiết kiệm thời gian chờ đợi một lượng thông tin lớn về từ server .

Một số ứng dụng sử dụng Ajax như : Gmail , Google Maps , Youtube , Facebook …

### jQuery Ajax

Jquery cung cấp một số phương thức để thực hiện các chức năng ajax. Chúng ta có thể yêu cầu các text, HTML, XML và JSON từ server sử dụng cả giao thức HTTP GET và HTTP POST, chúng ta cũng có thể lấy dữ liệu từ bên ngoài trực tiếp vào trong phần tử được chọn.

**Phương thức jquery load ()**

* Phương thức load() lấy dữ liệu từ server và trả dữ liệu cho phần tử được chọn.

$(selector).load(URL,data,callback);

* Cú pháp:
  + URL: mà bạn muốn lấy dữ liệu.
  + Data: cặp key/value gửi đi cùng với yêu cầu.
  + Callback: tên của hàm sẽ được thực thi sau khi phương thức load hoàn thành.

**Phương thức Post trong JQuery Ajax**

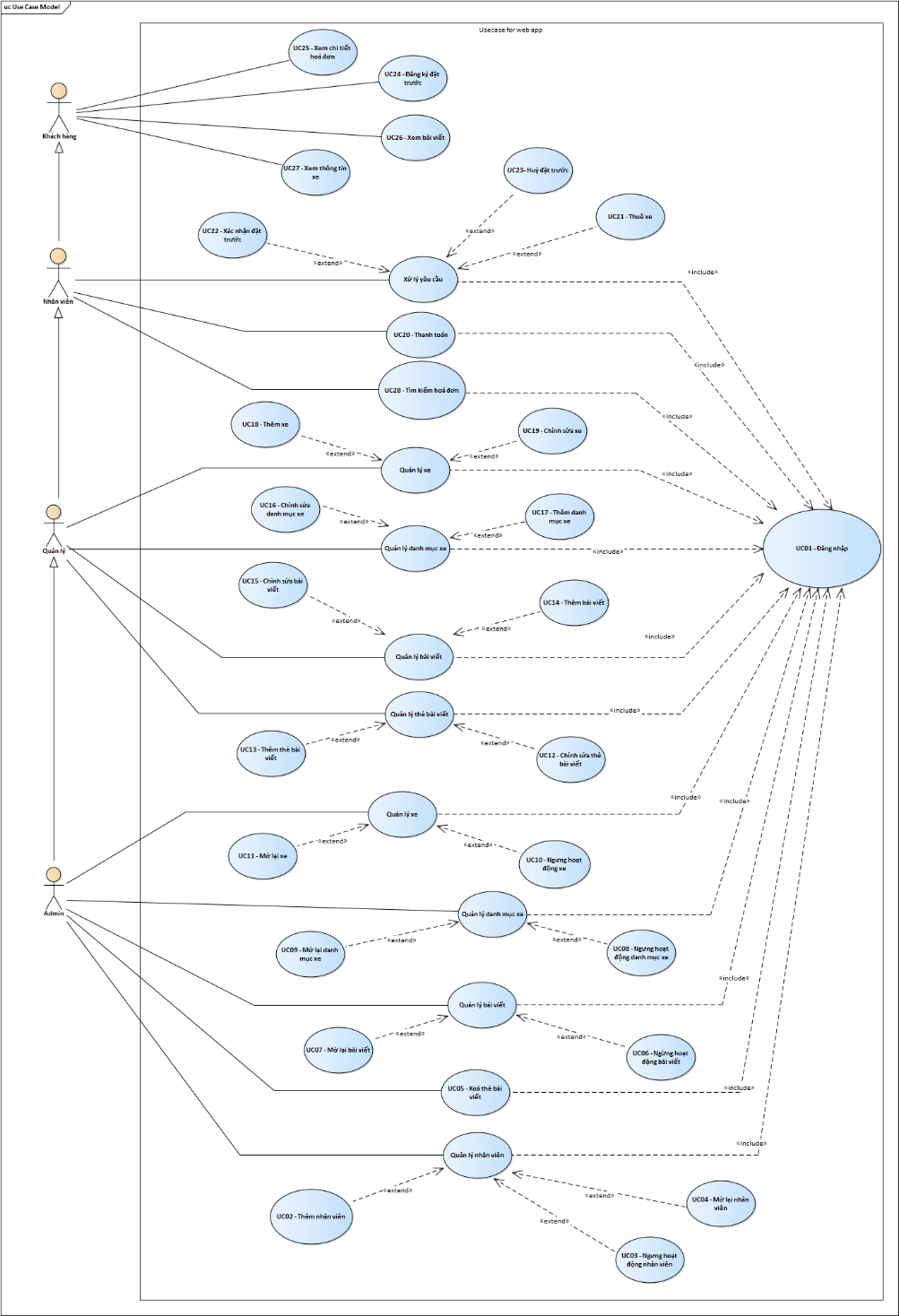
* Có tác dụng lấy dữ liệu từ server bằng phương thức HTTP POST REQUEST

$(selector).post(URL,data,function(data,status,xhr),dataType)

* Cú pháp:
  + url: required , đường dẫn đến file cần lấy thông tin
  + data: không bắt buộc ,là một đối tượng object gồm các key : value sẽ gửi lên server
  + function(data, status , xhr): là function sẽ xử lý khi thực hiện thành công với các parameters
    - data : bao gồm các dữ liệu trả về từ request
    - status : gồm trạng thái request (“success” , “notmodified” , “error” , “timeout” , or “parsererror”)
    - xhr : gồm các đối tượng XMLHttpRequest
  + dataType: là dạng dữ liệu trả về. (text, json, script, xml,html,jsonp )

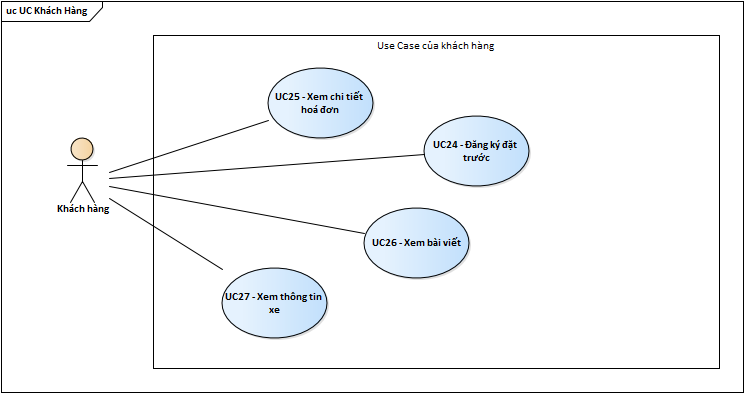
# PHÂN TÍCH VÀ ĐẶC TẢ CÁC YÊU CẦU CHỨC NĂNG

## Use case tổng quát

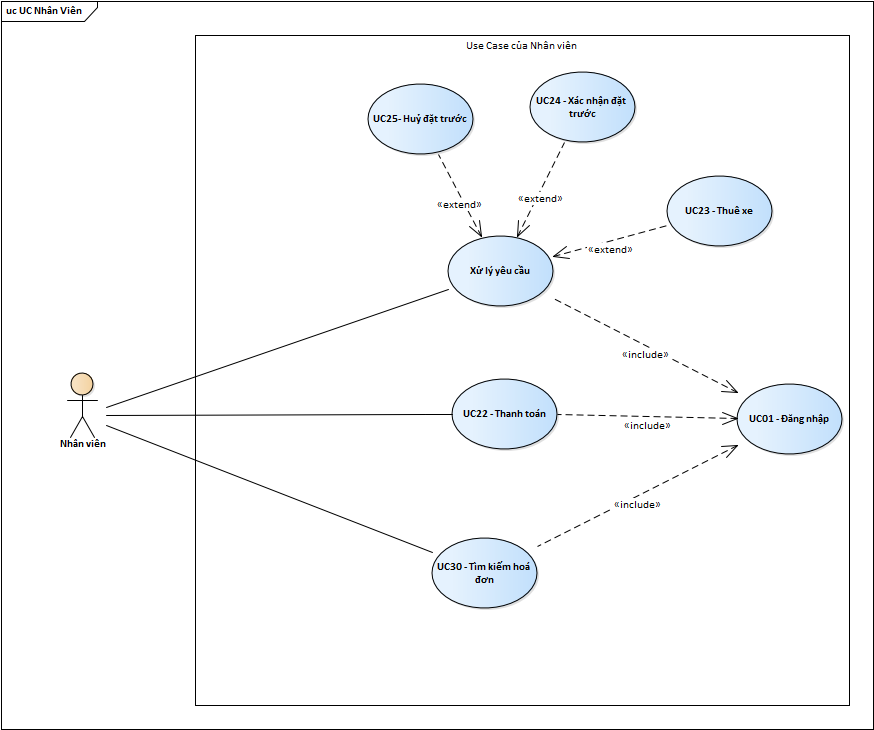


Hình 3.1 Use case tổng quát của Web App

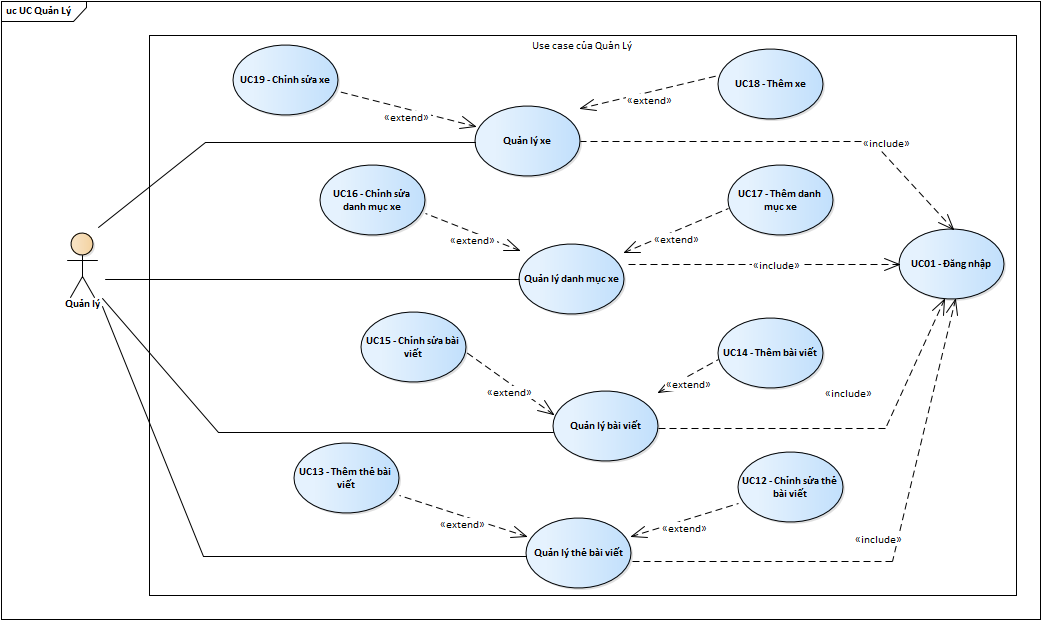
## Use case chi tiết của từng tác nhân



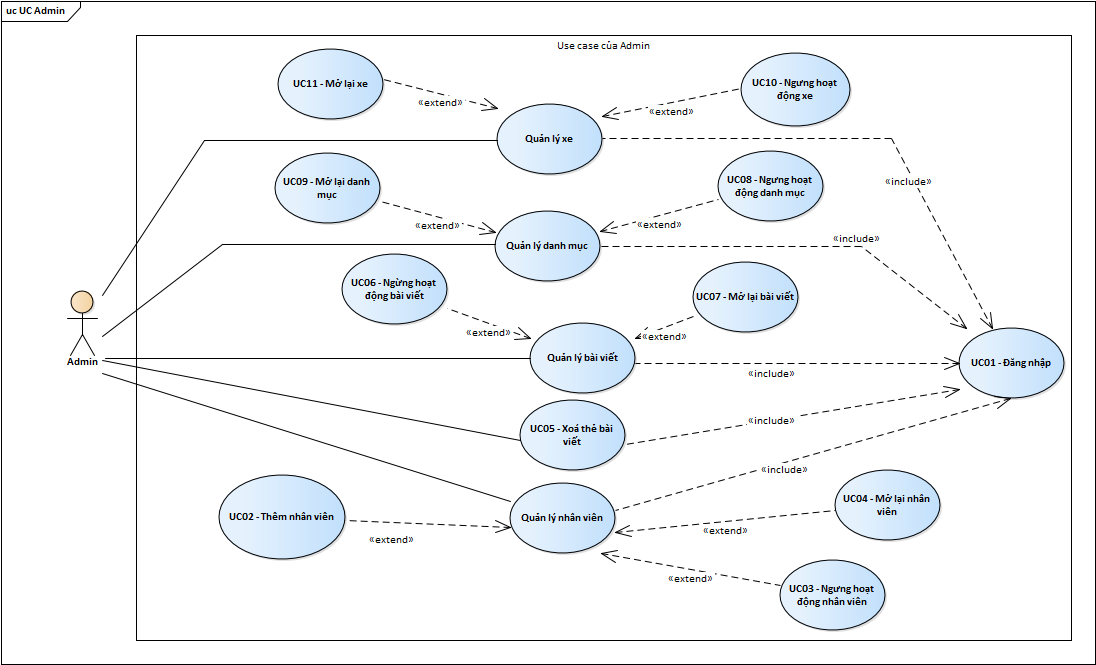
Hình 3.2 Use case chi tiết của khách hàng



Hình 3.3 Use case chi tiết của nhân viên



Hình 3.4 Use case chi tiết của quản lý



Hình 3.5 Use case chi tiết của admin

## Danh sách tác nhân và mô tả

|  |  |
| --- | --- |
| **Tác nhân** | **Mô tả tác nhân** |
| Khách hàng | Khách hàng có thể xem danh sách xe, đăng ký đặt trước, xem chi tiết hoá đơn, xem bài viết. |
| Nhân viên | Nhân viên có thể xử lý yêu cầu đặt trước, thanh toán, và tìm kiếm các hoá đơn đang trong tình trạng xử lý |
| Quản lý | Quản lý có thể quản lý xe, quản lý danh mục xe, quản lý bài viết, quản lý thẻ bài viết, và cũng như có mọi quyền của nhân viên |
| Admin | Admin có thể quản lý toàn bộ chức năng của quản lý và nhân viên, kèm theo là quản lý nhân viên |

Bảng 3.1 Danh sách tác nhân và mô tả

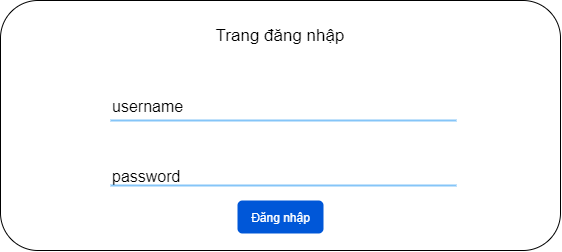
## Danh sách các tình huống hoạt động (Use cases)

|  |  |
| --- | --- |
| ID | Tên Use case |
| UC01 | Đăng nhập |
| UC02 | Thêm nhân viên |
| UC03 | Ngưng hoạt động nhân viên |
| UC04 | Mở lại nhân viên |
| UC05 | Xoá thẻ bài viết |
| UC06 | Ngưng hoạt động bài viết |
| UC07 | Mở lại bài viết |
| UC08 | Ngưng hoạt động danh mục xe |
| UC09 | Mở lại danh mục xe |
| UC10 | Ngưng hoạt động xe |
| UC11 | Mở lại xe |
| UC12 | Chỉnh sửa thẻ bài viết |
| UC13 | Thêm thẻ bài viết |
| UC14 | Thêm bài viết |
| UC15 | Chỉnh sửa bài viết |
| UC16 | Chỉnh sửa danh mục xe |
| UC17 | Thêm danh mục xe |
| UC18 | Thêm xe |
| UC19 | Chỉnh sửa xe |
| UC20 | Thanh toán |
| UC21 | Thuê xe |
| UC22 | Xác nhận đặt trước |
| UC23 | Huỷ đặt trước |
| UC24 | Đăng ký đặt trước |
| UC25 | Xem chi tiết hoá đơn |
| UC26 | Xem bài viết |
| UC27 | Xem danh sách xe |
| UC28 | Tìm kiếm hoá đơn |

Bảng 3.2 Danh sách các tình huống hoạt động (Use cases)

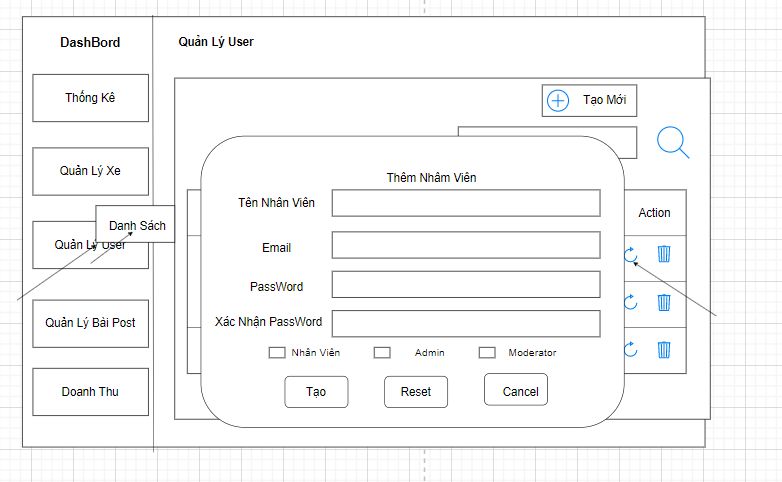
## Danh sách Mock up

### Mock up đăng nhập



Hình 3.6 Mock up đăng nhập

### Mock up thêm nhân viên



Hình 3.7 Mock up thêm nhân viên

### Mock up ngưng hoạt động nhân viên

### Mock up mở lại nhân viên

### Mock up xoá thẻ bài viết

### Mock up ngưng hoạt động bài viết

### Mock up mở lại bài viết

### Mock up ngưng hoạt động danh mục

### Mock up mở lại danh mục

### Mock up ngưng hoạt động xe

### Mock up mở lại xe

### Mock up chỉnh sửa thẻ bài viết

### Mock up thêm thẻ bài viết

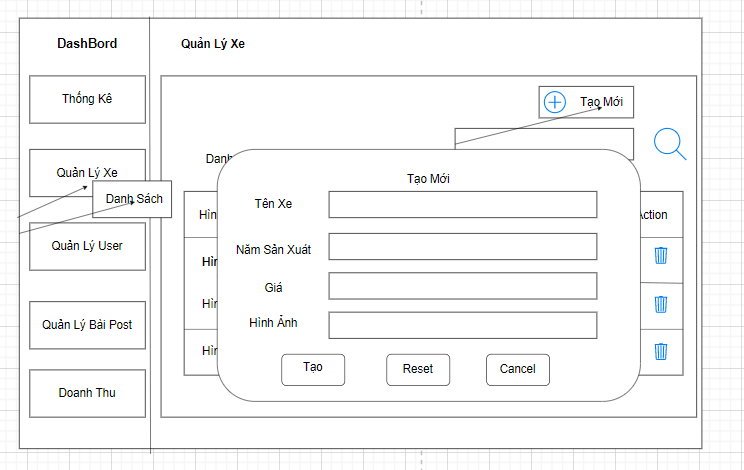
### Mock up thêm bài viêt

### Mock up chỉnh sửa bài viết

### Mock up chỉnh sửa danh mục xe

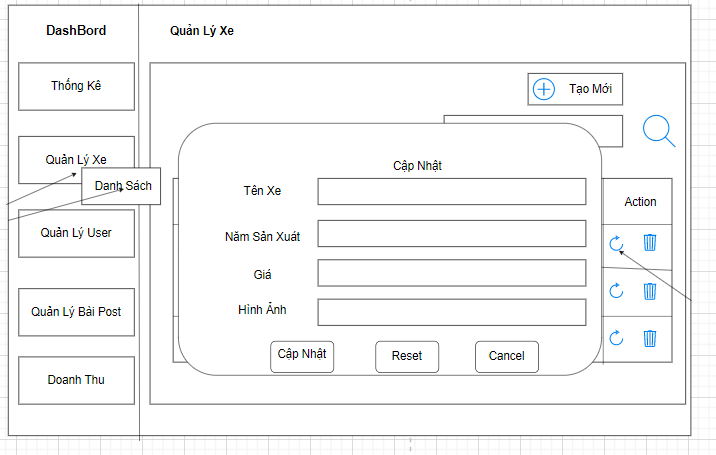
### Mock up thêm danh mục xe

### Mock up thêm xe



Hình 3.8 Mock up thêm xe

### Mock up chỉnh sửa xe



Hình 3.9 Mock up chỉnh sửa xe

### Mock up thanh toán

### Mock up thuê xe

### Mock up xác nhận đặt trước

### Mock up huỷ đặt trước

### Mock up đăng ký đặt trước

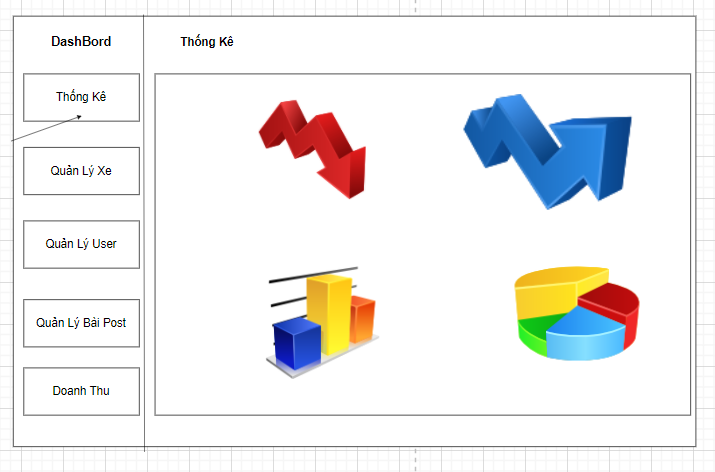
### Mock up xem chi tiết hoá đơn

### Mock up xem bài viết

### Mock up xem thông tin xe

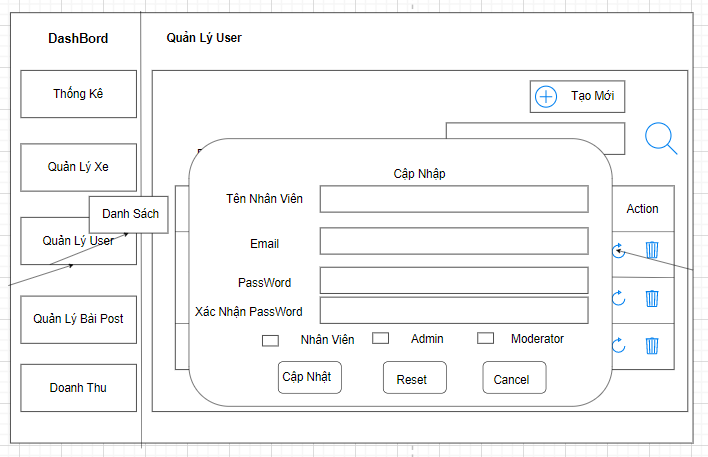
### Mock up tìm kiếm hoá đơn

### Mock up thống kê



Hình 3.10 Mock up thống kê

### Mock up cập nhật thông tin người dùng



## Tình huống hoạt động

### Đặc tả UC01\_Đăng nhập

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Đăng nhập | | |
| **Mô tả** | Cho phép người dùng đăng nhập các chức năng của hệ thống dựa theo phân quyền của người dùng | |
| **Tác nhân** | Nhân viên, quản lý, admin | |
| **Điều kiện trước** | Người dùng truy cập vào trang đăng nhập | |
| **Điều kiện sau** | Người dùng đăng nhập thành công | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** |  | 1. Hiển thị form đăng nhập. |
| 2. Điền thông tin đăng nhập (tài khoản, mật khẩu). |  |
| 3. Nhấn nút đăng nhập. |  |
|  | 4. Kiểm tra thông tin đăng nhập. |
|  | 5. Chuyển tới giao diện theo chức vụ của người dùng. |
| **Luồng sự kiện phụ (Alternative flows)** |  | 5.1. Thông tin không hợp lệ, quay lại bước 1 |

Bảng 3.3 Mô tả Use case đăng nhập

#### Activity diagram



Hình 3.11 Activity đăng nhập

#### Sequence diagram



Hình 3.12 Sequence đăng nhập

### Đặc tả UC02\_Thêm nhân viên

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Thêm nhân viên | | |
| **Mô tả** | Cho phép admin thêm một nhân viên mới | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện thêm nhân viên | |
| **Điều kiện sau** | Thêm thông tin nhân viên mới vào cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhập thông tin cần thiết để tạo tài khoản cho nhân viên mới(tên tài khoản, mật khẩu, chức vụ) |  |
| 2. Nhấn Xác nhận |  |
|  | 3. Kiểm tra thông tin nhân viên |
|  | 4. Thêm thông tin nhân viên vào cơ sở dữ liệu |
|  | 5. Thông báo thêm thành công lên giao diện |
| **Luồng sự kiện phụ (Alternative flows)** |  | 4.1. Thông báo thêm thất bại lên giao diện, kết thúc luồng |

Bảng 3.4 Mô tả Use case thêm nhân viên

#### Activity diagram



Hình 3.13 Activity thêm nhân viên

#### Sequence diagram



Hình 3.14 Sequence thêm nhân viên

### Đặc tả UC03\_Ngưng hoạt động nhân viên

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Ngưng hoạt động nhân viên | | |
| **Mô tả** | Cho phép admin ngưng hoạt động tài khoản của nhân viên | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý nhân viên, có danh sách nhân viên và có tài khoản đang hoạt động | |
| **Điều kiện sau** | Lưu thông tin nhân viên ngưng hoạt động vào cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn vào nút chuyển đổi tình trạng nhân viên sang “Disable” |  |
|  | 2. Xác nhận thay đổi tình trạng nhân viên |
| 3. Nhấn nút xác nhận |  |
|  | 4. Kiểm tra dữ liệu nhân viên |
|  | 5. Thay đổi thông tin nhân viên trên cơ sở dữ liệu |
|  | 6. Thông báo thay đổi thành công lên giao diện |
|  | 7. Cập nhật danh sách nhân viên |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Nhấn hút huỷ, kết thúc luồng |  |
|  | 5.1. Thông báo thay đổi thất bại, kết thúc luồng |

Bảng 3.5 Mô tả Use case ngưng hoạt động nhân viên

#### Activity diagram



Hình 3.15 Activity ngưng hoạt động nhân viên

#### Sequence diagram



Hình 3.16 Sequence ngưng hoạt động nhân viên

### Đặc tả UC04\_Mở lại nhân viên

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Mở lại nhân viên | | |
| **Mô tả** | Cho phép admin mở lại tài khoản nhân viên đã bị ngưng hoạt động | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý nhân viên, có danh sách nhân viên và có tài khoản ngưng hoạt động | |
| **Điều kiện sau** | Lưu thông tin nhân viên mở lại vào cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn vào nút chuyển đổi tình trạng sang “Enable” |  |
|  | 2. Xác nhận thay đổi tình trạng nhân viên |
| 3. Nhấn nút xác nhận |  |
|  | 4. Kiểm tra dữ liệu nhân viên |
|  | 5. Thay đổi thông tin nhân viên trên cơ sở dữ liệu |
|  | 6. Thông báo thay đổi thành công lên giao diện |
|  | 7. Cập nhật danh sách nhân viên |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Nhấn hút huỷ, kết thúc luồng |  |
|  | 5.1. Thông báo thay đổi thất bại, kết thúc luồng |

Bảng 3.6 Mô tả Use case đăng nhập

#### Activity diagram



Hình 3.17 Activity mở lại nhân viên

#### Sequence diagram



Hình 3.18 Sequence mở lại nhân viên

### Đặc tả UC05\_Xoá thẻ bài viết

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Xoá thẻ bài viết | | |
| **Mô tả** | Cho phép admin xoá thẻ của bài viết | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý thẻ, có danh sách thẻ | |
| **Điều kiện sau** | Xoá thẻ khỏi cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn vào nút xoá thẻ |  |
|  | 2. Xác nhận xoá thẻ |
| 3. Nhấn nút xác nhận |  |
|  | 4. Kiểm tra dữ liệu thẻ |
|  | 5. Xoá thẻ trên cơ sở dữ liệu |
|  | 6. Thông báo xoá thành công lên giao diện |
|  | 7. Cập nhật danh sách thẻ |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Nhấn nút huỷ, kết thúc luồng |  |
|  | 5.1. Thông báo xoá không thành công lên giao diện, kết thúc luồng |

Bảng 3.7 Mô tả Use case đăng nhập

#### Activity diagram



Hình 3.19 Activity xoá thẻ bài viết

#### Sequence diagram



Hình 3.20 Sequence xoá thẻ bài viết

### Đặc tả UC06\_Ngưng hoạt động bài viết

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Ngưng hoạt động bài viết | | |
| **Mô tả** | Cho phép admin ngưng hoạt động bài viết, không cho bài viết xuất hiện trên giao diện người dùng | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý bài viết, có danh sách bài viết và có bài viết chưa ngưng hoạt động | |
| **Điều kiện sau** | Lưu thông tin bài viết ngưng hoạt động vào cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn vào nút chuyển đổi tình trạng bài viết sang “Disable” |  |
|  | 2. Xác nhận thay đổi tình trạng bài viết |
| 3. Nhấn nút xác nhận |  |
|  | 4. Kiểm tra dữ liệu bài viết |
|  | 5. Thay đổi thông tin bài viết trên cơ sở dữ liệu |
|  | 6. Thông báo thay đổi thành công lên giao diện |
|  | 7. Cập nhật danh sách bài viết |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Nhấn hút huỷ, kết thúc luồng |  |
|  | 5.1. Thông báo thay đổi thất bại, kết thúc luồng |

Bảng 3.8 Mô tả Use case ngưng hoạt động bài viết

#### Activity diagram



Hình 3.21 Activity ngưng hoạt động bài viết

#### Sequence diagram



Hình 3.22 Sequence ngưng hoạt động bài viết

### Đặc tả UC07\_Mở lại bài viết

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Mở lại bài viết | | |
| **Mô tả** | Cho phép admin mở lại bài viết đã bị ngưng hoạt động | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý bài viết, có danh sách bài viết và có bài biết ngưng hoạt động | |
| **Điều kiện sau** | Lưu thông tin bài viết mở lại vào cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn vào nút chuyển đổi tình trạng bài viết sang “Enable” |  |
|  | 2. Xác nhận thay đổi tình trạng bài viết |
| 3. Nhấn nút xác nhận |  |
|  | 4. Kiểm tra dữ liệu bài viết |
|  | 5. Thay đổi thông tin bài viết trên cơ sở dữ liệu |
|  | 6. Thông báo thay đổi thành công lên giao diện |
|  | 7. Cập nhật danh sách bài viết |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Nhấn hút huỷ, kết thúc luồng |  |
|  | 5.1. Thông báo thay đổi thất bại, kết thúc luồng |

Bảng 3.9 Mô tả Use case mở lại bài viết

#### Activity diagram



Hình 3.23 Activity mở lại bài viết

#### Sequence diagram



Hình 3.24 Sequence mở lại bài viết

### Đặc tả UC08\_Ngưng hoạt động danh mục

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Ngưng hoạt động danh mục | | |
| **Mô tả** | Cho phép admin ngưng hoạt động danh mục, không cho danh mục xuất hiện trên giao diện người dùng | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý danh mục, có danh sách danh mục và có danh mục chưa ngưng hoạt động | |
| **Điều kiện sau** | Lưu thông tin danh mục ngưng hoạt động vào cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn vào nút chuyển đổi tình trạng danh mục sang “Disable” |  |
|  | 2. Xác nhận thay đổi tình trạng danh mục |
| 3. Nhấn nút xác nhận |  |
|  | 4. Kiểm tra dữ liệu danh mục |
|  | 5. Thay đổi thông tin dạnh mục trên cơ sở dữ liệu |
|  | 6. Thông báo thay đổi thành công lên giao diện |
|  | 7. Cập nhật danh sách danh mục |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Nhấn hút huỷ, kết thúc luồng |  |
|  | 5.1. Thông báo thay đổi thất bại, kết thúc luồng |

Bảng 3.10 Mô tả Use case đăng nhập

#### Activity diagram



Hình 3.25 Activity ngưng hoạt động danh mục

#### Sequence diagram



Hình 3.26 Sequence ngưng hoạt động danh mục

### Đặc tả UC09\_Mở lại danh mục

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Mở lại danh mục | | |
| **Mô tả** | Cho phép admin mở lại danh mục đã bị ngưng hoạt động | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý danh mục, có danh sách danh mục và có danh mục ngưng hoạt động | |
| **Điều kiện sau** | Lưu thông tin danh mục mở lại vào cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn vào nút chuyển đổi tình trạng danh mục sang “Enable” |  |
|  | 2. Xác nhận thay đổi tình trạng danh mục |
| 3. Nhấn nút xác nhận |  |
|  | 4. Kiểm tra dữ liệu danh mục |
|  | 5. Thay đổi thông tin danh mục trên cơ sở dữ liệu |
|  | 6. Thông báo thay đổi thành công lên giao diện |
|  | 7. Cập nhật danh sách danh mục |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Nhấn hút huỷ, kết thúc luồng |  |
|  | 5.1. Thông báo thay đổi thất bại, kết thúc luồng |

Bảng 3.11 Mô tả Use case mở lại danh mục

#### Activity diagram



Hình 3.27 Activity mở lại danh mục

#### Sequence diagram



Hình 3.28 Sequence mở lại danh mục

### Đặc tả UC10\_Ngưng hoạt động xe

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Ngưng hoạt động danh mục | | |
| **Mô tả** | Cho phép admin ngưng hoạt động xe, không cho xe xuất hiện trên giao diện người dùng | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý xe, có danh sách xe và có xe chưa ngưng hoạt động | |
| **Điều kiện sau** | Lưu thông tin xe ngưng hoạt động vào cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn vào nút chuyển đổi tình trạng xe sang “Disable” |  |
|  | 2. Xác nhận thay đổi tình trạng xe |
| 3. Nhấn nút xác nhận |  |
|  | 4. Kiểm tra dữ liệu xe |
|  | 5. Thay đổi thông tin xe trên cơ sở dữ liệu |
|  | 6. Thông báo thay đổi thành công lên giao diện |
|  | 7. Cập nhật danh sách xe |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Nhấn hút huỷ, kết thúc luồng |  |
|  | 5.1. Thông báo thay đổi thất bại, kết thúc luồng |

Bảng 3.12 Mô tả Use case ngưng hoạt động xe

#### Activity diagram



Hình 3.29 Activity ngưng hoạt động xe

#### Sequence diagram



Hình 3.30 Sequence ngưng hoạt động xe

### Đặc tả UC11\_Mở lại xe

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Mở lại xe | | |
| **Mô tả** | Cho phép admin mở lại xe đã bị ngưng hoạt động | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý xe, có danh sách xe và có xe ngưng hoạt động | |
| **Điều kiện sau** | Lưu thông tin xe mở lại vào cơ sở dữ liệu  Thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn vào nút chuyển đổi tình trạng xe sang “Enable” |  |
|  | 2. Xác nhận thay đổi tình trạng xe |
| 3. Nhấn nút xác nhận |  |
|  | 4. Kiểm tra dữ liệu xe |
|  | 5. Thay đổi thông tin xe trên cơ sở dữ liệu |
|  | 6. Thông báo thay đổi thành công lên giao diện |
|  | 7. Cập nhật danh sách xe |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Nhấn hút huỷ, kết thúc luồng |  |
|  | 5.1. Thông báo thay đổi thất bại, kết thúc luồng |

Bảng 3.13 Mô tả Use case mở lại xe

#### Activity diagram



Hình 3.31 Activity mở lại xe

#### Sequence diagram



Hình 3.32 Sequence mở lại xe

### Đặc tả UC12\_Chỉnh sửa thẻ bài viết

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Chỉnh sửa thẻ bài viết | | |
| **Mô tả** | Người dùng chỉnh sửa thông tin của thẻ bài viết | |
| **Tác nhân** | Quản lý, admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý thẻ bài viết và có danh sách thẻ bài viết | |
| **Điều kiện sau** | Cập nhật thông tin thẻ bài viết dưới cơ sở dữ liệu và thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn nút Chỉnh sửa thẻ bài viết |  |
|  | 2. Hiện lên biểu mẫu thông tin thẻ bài viết để chỉnh sửa |
| 3. Nhập thông tin cần chỉnh sửa |  |
| 4. Nhấn Lưu |  |
|  | 5. Kiểm tra dữ liệu chỉnh sửa |
|  | 6. Cập nhật thông tin chỉnh sửa lên cơ sở dữ liệu |
|  | 7. Thông báo chỉnh sửa thành công |
|  | 8. Cập nhật lại danh sách thẻ bài viết |
| **Luồng sự kiện phụ (Alternative flows)** |  | 6.1. Thông báo dữ liệu không hợp lệ, lưu không thành công, kết thúc luồng |

Bảng 3.14 Mô tả Use case chỉnh sửa thẻ bài viết

#### Activity diagram



Hình 3.33 Activity chỉnh sửa thẻ bài viết

#### Sequence diagram



Hình 3.34 Sequence chỉnh sửa thẻ bài viết

### Đặc tả UC13\_Thêm thẻ bài viết

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Thêm thẻ bài viết | | |
| **Mô tả** | Người dùng có thể thêm một thẻ bài viết mới | |
| **Tác nhân** | Quản lý, admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý thẻ bài viết | |
| **Điều kiện sau** | Thêm thông tin thẻ bài viết dưới cơ sở dữ liệu và thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn nút Thêm thẻ bài viết |  |
|  | 2. Hiển thị biểu mẫu nhập thông tin thẻ bài viết |
| 3. Nhập thông tin thẻ bài viết |  |
| 4. Nhấn Lưu |  |
|  | 5. Kiểm tra dữ liệu thêm mới |
|  | 6. Thêm dữ liệu mới lên cơ sở dữ liệu |
|  | 7. Thông báo thêm thành công |
|  | 8. Cập nhật lại danh sách thẻ bài viết |
| **Luồng sự kiện phụ (Alternative flows)** |  | 6.1. Thông báo dữ liệu không hợp lệ, lưu không thành công, kết thúc luồng |

Bảng 3.15 Mô tả Use case thêm thẻ bài viết

#### Activity diagram



Hình 3.35 Activity thêm thẻ bài viết

#### Sequence diagram



Hình 3.36 Sequence thêm thẻ bài viết

### Đặc tả UC14\_Thêm bài viết

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Thêm bài viết | | |
| **Mô tả** | Người dùng có thể thêm bài viết mới | |
| **Tác nhân** | Quản lý, admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý bài viết | |
| **Điều kiện sau** | Thêm thông tin bài viết dưới cơ sở dữ liệu và thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn nút Thêm bài viết |  |
|  | 2. Hiển thị biểu mẫu nhập thông tin bài viết |
| 3. Nhập thông tin bài viết |  |
| 4. Nhấn Lưu |  |
|  | 5. Kiểm tra dữ liệu thêm mới |
|  | 6. Thêm dữ liệu mới lên cơ sở dữ liệu |
|  | 7. Thông báo thêm thành công |
|  | 8. Cập nhật lại danh sách bài viết |
| **Luồng sự kiện phụ (Alternative flows)** |  | 6.1. Thông báo dữ liệu không hợp lệ, lưu không thành công, kết thúc luồng |

Bảng 3.16 Mô tả Use case thêm bài viết

#### Activity diagram



Hình 3.37 Activity thêm thẻ bài viết

#### Sequence diagram



Hình 3.38 Sequence thêm thẻ bài viết

### Đặc tả UC15\_Chỉnh sửa bài viết

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Chỉnh sửa bài viết | | |
| **Mô tả** | Người dùng chỉnh sửa thông tin của bài viết | |
| **Tác nhân** | Quản lý, admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý bài viết và có danh sách bài viết | |
| **Điều kiện sau** | Cập nhật thông tin bài viết dưới cơ sở dữ liệu và thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn nút Chỉnh sửa bài viết |  |
|  | 2. Hiện lên biểu mẫu thông tin bài viết để chỉnh sửa |
| 3. Nhập thông tin cần chỉnh sửa |  |
| 4. Nhấn Lưu |  |
|  | 5. Kiểm tra dữ liệu chỉnh sửa |
|  | 6. Cập nhật thông tin chỉnh sửa lên cơ sở dữ liệu |
|  | 7. Thông báo chỉnh sửa thành công |
|  | 8. Cập nhật lại danh sách bài viết |
| **Luồng sự kiện phụ (Alternative flows)** |  | 6.1. Thông báo dữ liệu không hợp lệ, lưu không thành công, kết thúc luồng |

Bảng 3.17 Mô tả Use case chỉnh sửa bài viết

#### Activity diagram



Hình 3.39 Activity chỉnh sửa bài viết

#### Sequence diagram



Hình 3.40 Sequence chỉnh sửa bài viết

### Đặc tả UC16\_Chỉnh sửa danh mục xe

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Chỉnh sửa danh mục xe | | |
| **Mô tả** | Người dùng chỉnh sửa thông tin của danh mục xe | |
| **Tác nhân** | Quản lý, admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý danh mục xe và có danh sách danh mục xe | |
| **Điều kiện sau** | Cập nhật thông tin danh mục xe dưới cơ sở dữ liệu và thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn nút Chỉnh sửa danh mục |  |
|  | 2. Hiện lên biểu mẫu thông tin danh mục để chỉnh sửa |
| 3. Nhập thông tin cần chỉnh sửa |  |
| 4. Nhấn Lưu |  |
|  | 5. Kiểm tra dữ liệu chỉnh sửa |
|  | 6. Cập nhật thông tin chỉnh sửa lên cơ sở dữ liệu |
|  | 7. Thông báo chỉnh sửa thành công |
|  | 8. Cập nhật lại danh sách danh mục xe |
| **Luồng sự kiện phụ (Alternative flows)** |  | 6.1. Thông báo dữ liệu không hợp lệ, lưu không thành công, kết thúc luồng |

Bảng 3.18 Mô tả Use case chỉnh sửa danh mục xe

#### Activity diagram



Hình 3.41 Activity chỉnh sửa danh mục xe

#### Sequence diagram



Hình 3.42 Sequence chỉnh sửa danh mục xe

### Đặc tả UC17\_Thêm danh mục xe

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Thêm danh mục xe | | |
| **Mô tả** | Người dùng có thể thêm danh mục xe mới | |
| **Tác nhân** | Quản lý, admin | |
| **Điều kiện trước** | Đã đăng nhập thành công  Đang ở giao diện quản lý danh mục | |
| **Điều kiện sau** | Thêm thông tin danh mục dưới cơ sở dữ liệu và thông báo lên giao diện người dùng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhấn nút Thêm danh mục |  |
|  | 2. Hiển thị biểu mẫu nhập thông tin danh mục |
| 3. Nhập thông tin danh mục |  |
| 4. Nhấn Lưu |  |
|  | 5. Kiểm tra dữ liệu thêm mới |
|  | 6. Thêm dữ liệu mới lên cơ sở dữ liệu |
|  | 7. Thông báo thêm thành công |
|  | 8. Cập nhật lại danh sách danh mục |
| **Luồng sự kiện phụ (Alternative flows)** |  | 6.1. Thông báo dữ liệu không hợp lệ, lưu không thành công, kết thúc luồng |

Bảng 3.19 Mô tả Use case thêm danh mục xe

#### Activity diagram



Hình 3.43 Activity thêm danh mục xe

#### Sequence diagram



Hình 3.44 Sequence thêm danh mục xe

### Đặc tả UC18\_Thêm xe

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Thêm xe | | |
| **Mô tả** | Quản lý, Admin thực hiện nhập các thông tin xe gồm: tên xe, giá cả, mô tả, biển số. | |
| **Tác nhân** | Quản Lý, Admin | |
| **Điều kiện trước** | Đang mở website, ở giao diện danh sách xe. | |
| **Điều kiện sau** | Thêm thông tin sản phẩm dưới cơ sở dữ liệu và thông báo lên giao diện người dùng. | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhập thông tin xe |  |
| 2. Ấn thêm. |  |
|  | 3. Kiểm tra thông tin |
|  | 4. Thêm thông tin xe vào dữ liệu |
|  | 5. Hiển thị kết quả. |
| **Luồng sự kiện phụ (Alternative flows)** |  | 4.1. Thông báo thêm không thành công |

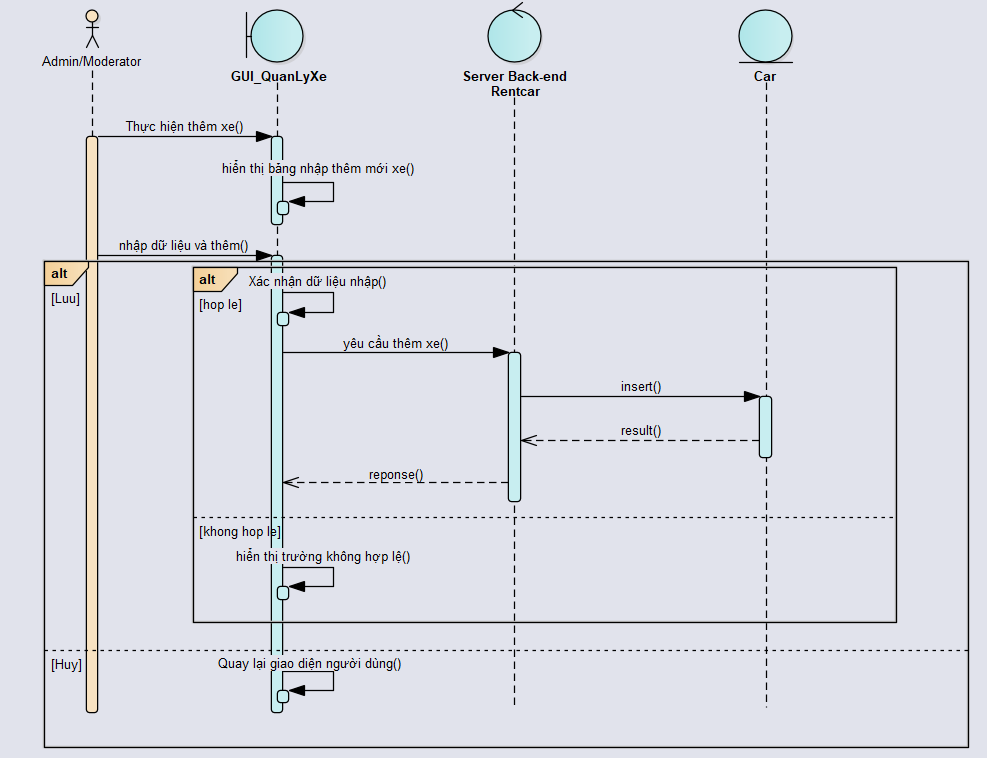
Bảng 3.20 Mô tả Use case thêm xe

#### Activity diagram



Hình 3.45 Activity thêm xe

#### Sequence diagram



Hình 3.46 Sequence thêm xe

### Đặc tả UC19\_Chỉnh sửa xe

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Chỉnh sửa xe | | |
| **Mô tả** | Quản lý, Admin chỉnh sửa thông tin (tên, giá, mô tả, tình trạng,số lượng xe còn ) sản phẩm. | |
| **Tác nhân** | Quản lý, Admin | |
| **Điều kiện trước** | Đang mở website, ở giao diện danh sách xe. | |
| **Điều kiện sau** | Cập nhật thông tin sản phẩm dưới cơ sở dữ liệu và thông báo lên giao diện người dùng. | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Chọn vào “Nút” chỉnh sửa xe. |  |
|  | 2. Hiện lên biểu mẫu thông tin chỉnh sửa sản phẩm. |
| 3. Nhập các thông tin cần chỉnh sửa theo nhu cầu. |  |
|  | 4. Kiểm tra dữ liệu hợp lệ. |
|  | 5. Hiển thị kết quả. |
| **Luồng sự kiện phụ (Alternative flows)** |  | 4.1. Hệ thống kiểm tra dữ liệu không hợp lệ.Quay lại bước 3 |

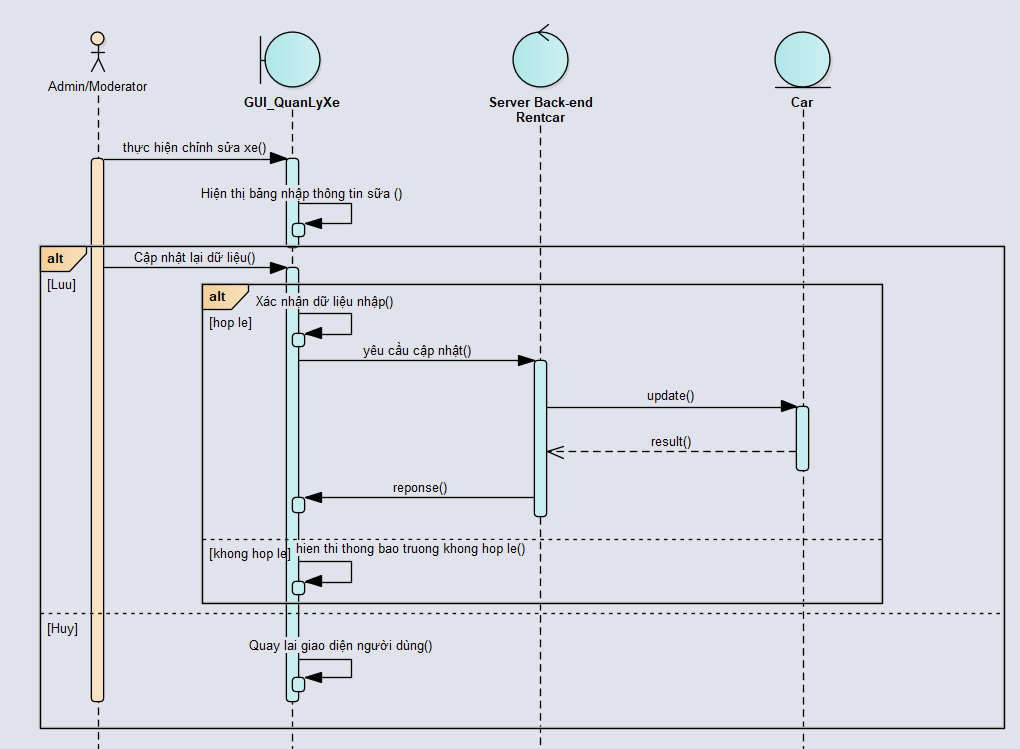
Bảng 3.21 Mô tả Use case chỉnh sửa xe

#### Activity diagram



Hình 3.47 Activity chỉnh sửa xe

#### Sequence diagram



Hình 3.48 Sequence chỉnh sửa xe

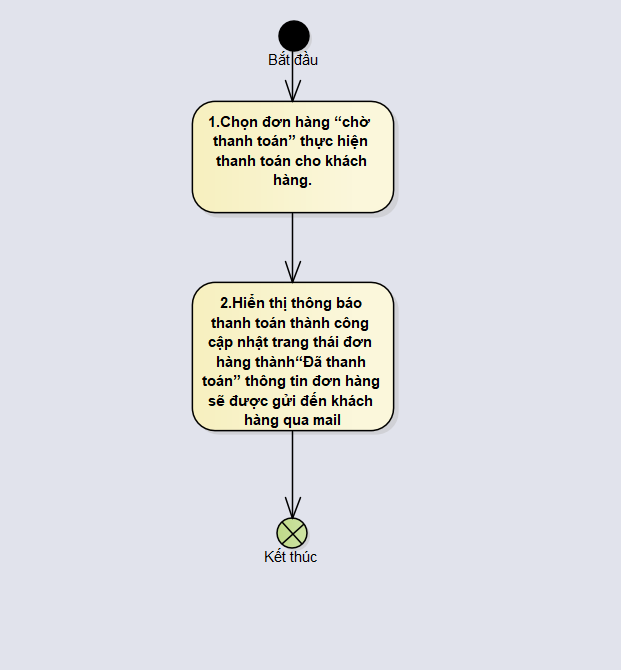
### Đặc tả UC20\_Thanh toán

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** thanh toán | | |
| **Mô tả** | Nhân viên chọn vào đơn hàng khách chờ thanh toán và thực hiện thanh toán đơn hàng thuê của khách hàng. | |
| **Tác nhân** | Nhân Viên | |
| **Điều kiện trước** | Đang mở website, ở giao diện danh sách đơn hàng. | |
| **Điều kiện sau** | Đơn hàng thanh toán thành công và chuyển trạng thái đơn hoàn thành | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Chọn đơn hàng “chờ thanh toán” thực hiện thanh toán cho khách hàng. |  |
|  | 2. Hiển thị thông báo thanh toán thành công cập nhật trang thái đơn hàng thành“Đã thanh toán” thông tin đơn hàng sẽ được gửi đến khách hàng qua mail |
| **Luồng sự kiện phụ (Alternative flows)** |  |  |

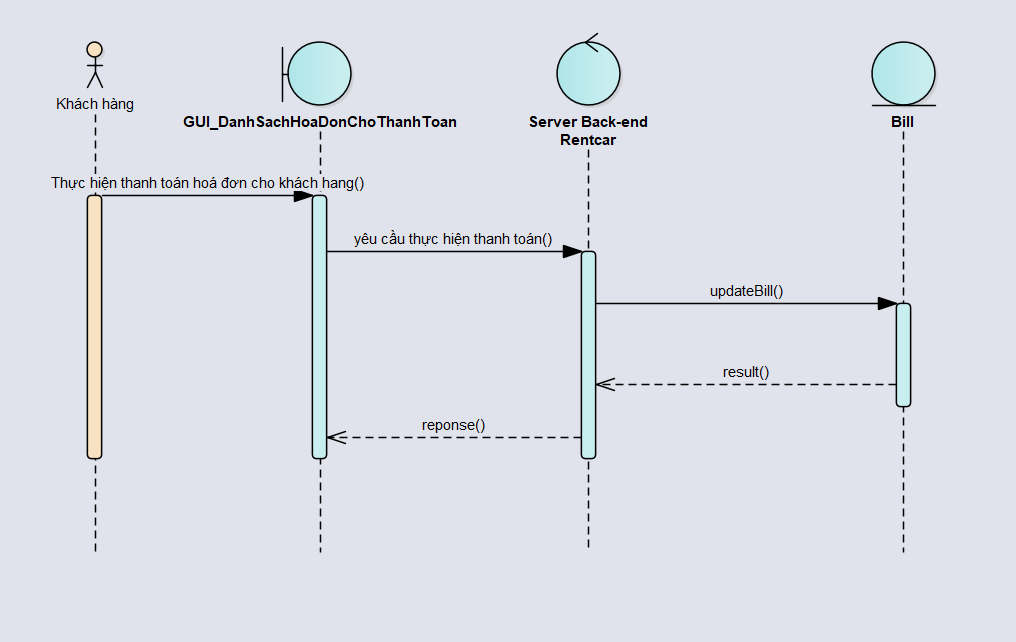
Bảng 3.22 Mô tả Use case thanh toán

#### Activity diagram



Hình 3.49 Activity thanh toán

#### Sequence diagram



Hình 3.50 Sequence thanh toán

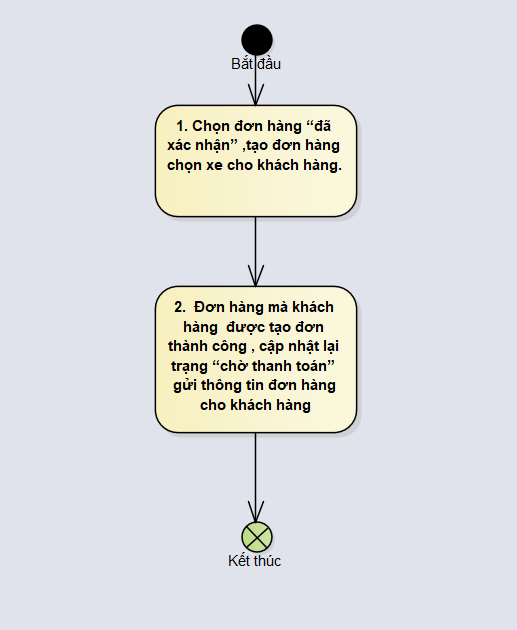
### Đặc tả UC21\_Thuê xe

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Thuê xe | | |
| **Mô tả** | Nhân viên chọn vào đơn hàng khách đã xác nhận và thực hiện xác nhận tạo đơn hàng thuê cho khách hàng. | |
| **Tác nhân** | Nhân Viên | |
| **Điều kiện trước** | Đang mở website, ở giao diện danh sách đơn hàng. | |
| **Điều kiện sau** | Đơn hàng đặt trước được tạo đơn hàng thành công cập nhật lại trạng thái đơn hàng thành “chờ thành thanh toán” , lưu thông tin danh sách | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Chọn đơn hàng “đã xác nhận” ,tạo đơn hàng chọn xe cho khách hàng. |  |
|  | 2. Đơn hàng mà khách hàng được tạo đơn thành công , cập nhật lại trạng “chờ thanh toán” gửi thông tin đơn hàng cho khách hàng |
| **Luồng sự kiện phụ (Alternative flows)** |  |  |

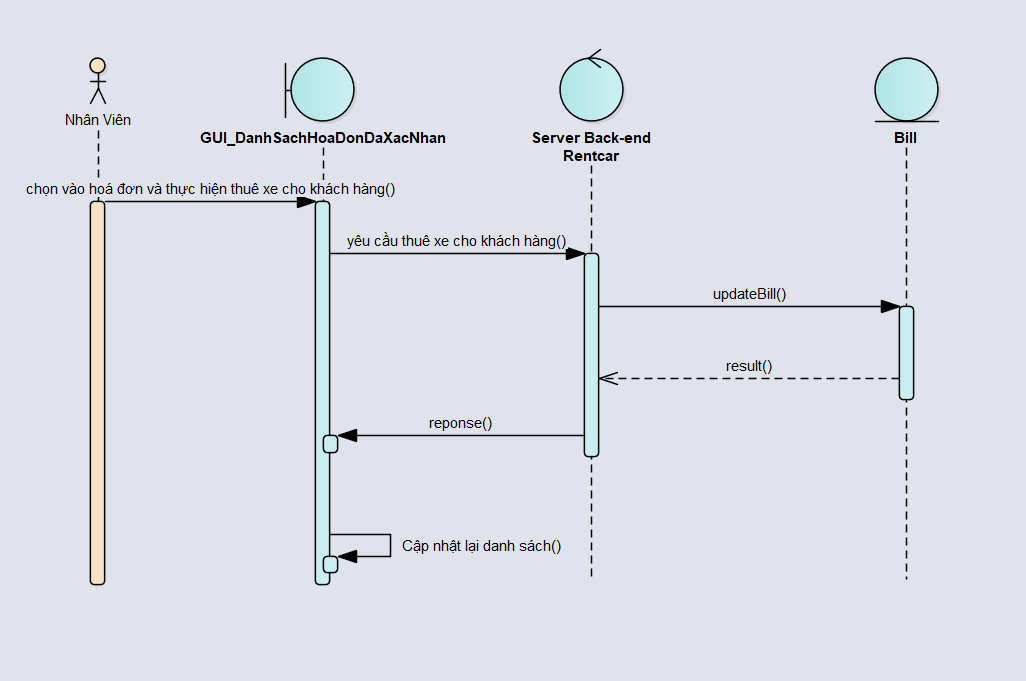
Bảng 3.23 Mô tả Use case thuê xe

#### Activity diagram



Hình 3.51 Activity thuê xe

#### Sequence diagram



Hình 3.52 Sequence thuê xe

### Đặc tả UC22\_Xác nhận đặt trước

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Xác nhận đặt trước | | |
| **Mô tả** | Nhân viên chọn vào đơn hàng khách đã đồng ý và thực hiện xác nhận đơn hàng của khách. | |
| **Tác nhân** | Nhân Viên | |
| **Điều kiện trước** | Đang mở website, ở giao diện danh sách đơn hàng. | |
| **Điều kiện sau** | Đơn hàng đặt trước được xác nhận thành công cập nhật lại trạng thái đơn hàng thành “đã xác nhận” , lưu danh sách đơn đã xác nhận vào danh sách. | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Chọn đơn hàng đặt trước của khách hàng , chọn vào phần xác nhận đơn hàng. |  |
|  | 2. Đơn hàng mà khách hàng đồng ý, cập nhật lại trạng thái “đã xác nhận” lưu đơn hàng của khách trong danh sách và cập lại danh sách đơn hàng. |
| **Luồng sự kiện phụ (Alternative flows)** |  |  |

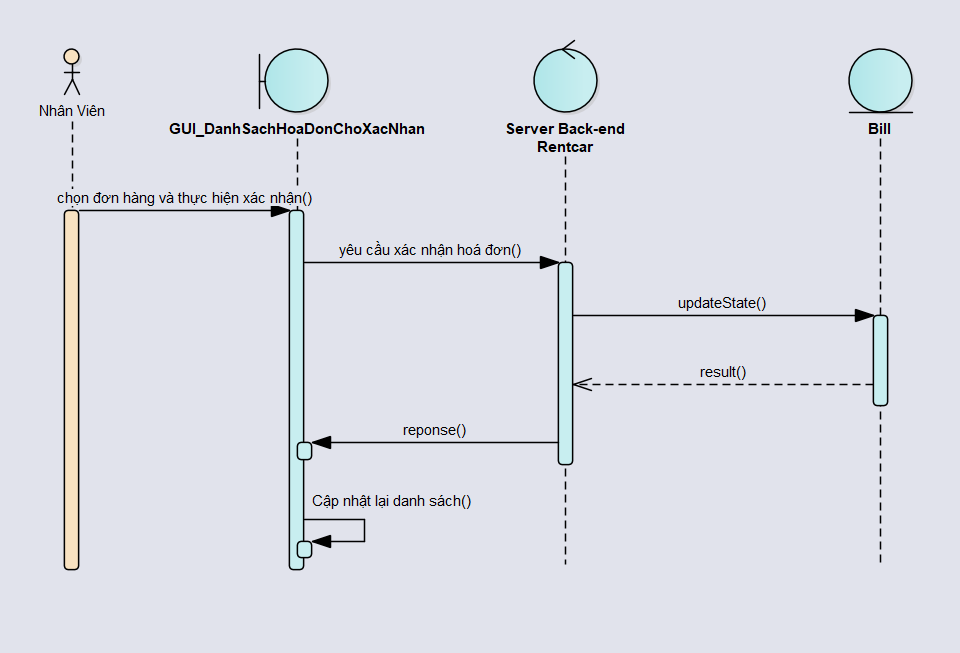
Bảng 3.24 Mô tả Use case xác nhận đặt trước

#### Activity diagram



Hình 3.53 Activity xác nhận đặt trước

#### Sequence diagram



Hình 3.54 Sequence xác nhận đặt trước

### Đặc tả UC23\_Huỷ đặt trước

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Huỷ đặt trước | | |
| **Mô tả** | Nhân viên chọn vào đơn hàng khách đã không đồng ý và thực hiện huỷ đơn hàng của khách | |
| **Tác nhân** | Nhân Viên | |
| **Điều kiện trước** | Đang mở website, ở giao diện danh sách đơn hàng. | |
| **Điều kiện sau** | Đơn hàng đặt trước xoá thành công ra khỏi danh sách và cập nhật lại danh sách trên website. | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Chọn đơn hàng đặt trước của khách hàng , chọn vào phần xoá đơn hàng. |  |
|  | 2. Hiển thị thống báo xác nhận xoá hoá đơn đặt trước của khách hàng. |
| 3. Chọn xoá |  |
|  | 4. xoá thành công ra khỏi danh sách và cập nhật lại danh sách. |
| **Luồng sự kiện phụ (Alternative flows)** | 3.1. Chọn huỷ, kết thúc luồng |  |

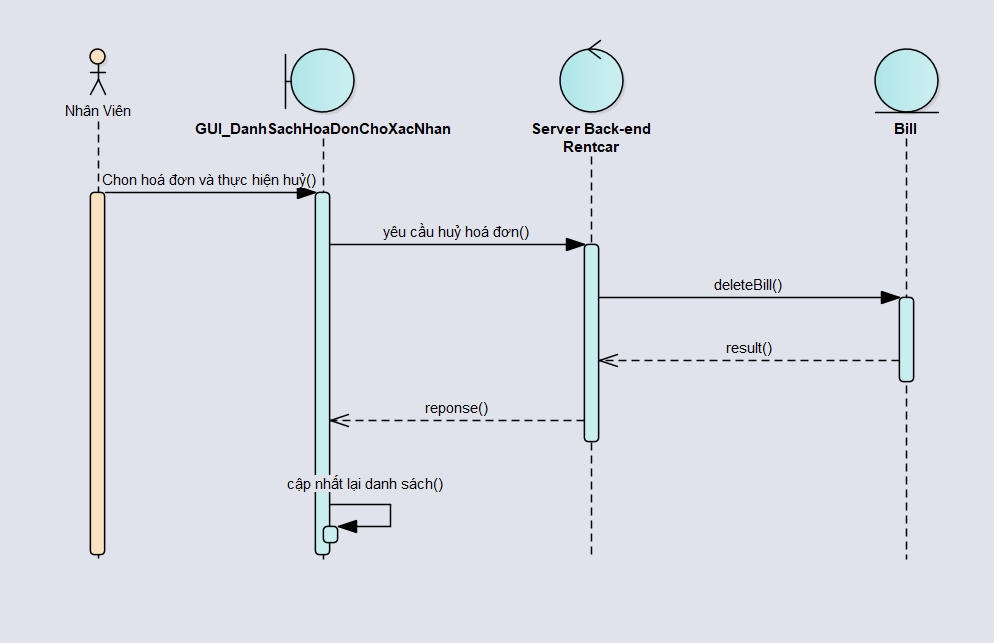
Bảng 3.25 Mô tả Use case huỷ đặt trước

#### Activity diagram



Hình 3.55 Activity huỷ đặt trước

#### Sequence diagram



Hình 3.56 Sequence huỷ đặt trước

### Đặc tả UC24\_Đăng ký đặt trước

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** đăng ký đặt trước | | |
| **Mô tả** | Người dùng nhập thông tin cá nhân của mình trên giao diện website để được gọi tư vấn thuê xe. | |
| **Tác nhân** | Khách hàng | |
| **Điều kiện trước** | Đang mở website, ở giao diện chính hoặc giao diện thuê xe | |
| **Điều kiện sau** | Hiển thị thông báo thành công và gửi email bao gồm thông tin đăng ký về cho khách hàng | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhập thông tin cá nhân |  |
| 2. Nhấn đăng ký |  |
|  | 3. Hiển thị thông báo thành công cho khách hàng, gửi email đăng ký thành công cho khách hàng |
| **Luồng sự kiện phụ (Alternative flows)** |  | 3.1. Hiển thị thông báo không thành công |

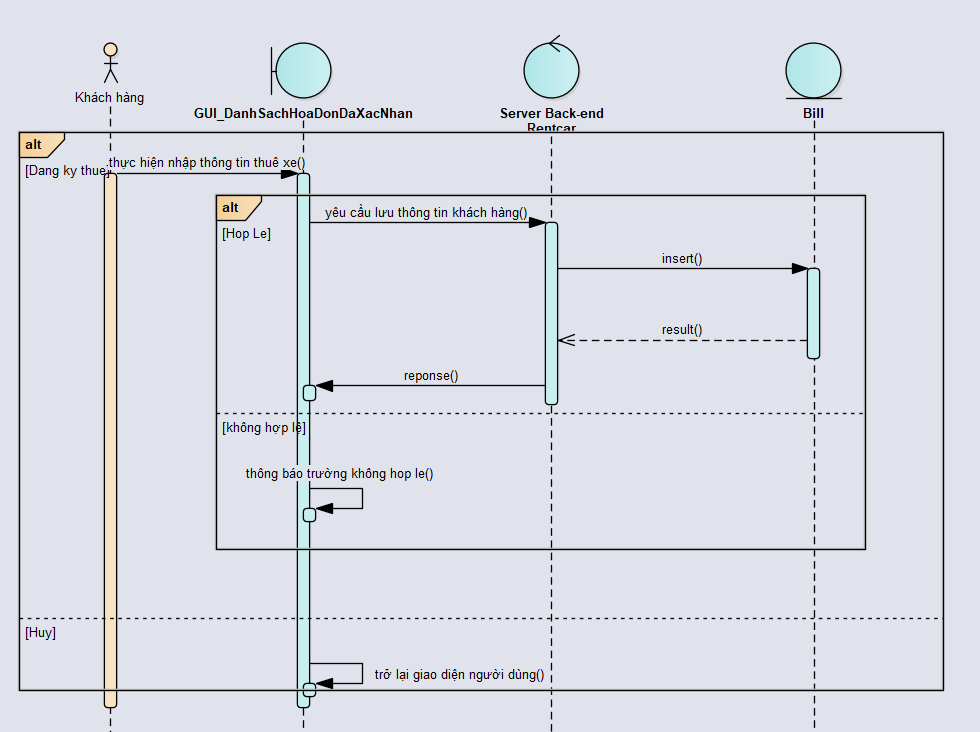
Bảng 3.26 Mô tả Use case đăng ký đặt trước

#### Activity diagram



Hình 3.57 Activity đăng ký đặt trước

#### Sequence diagram



Hình 3.58 Sequence đăng ký đặt trước

### Đặc tả UC25\_Xem chi tiết hoá đơn

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** xem chi tiết hoá đơn | | |
| Mô tả | Người dùng vào mail để xem chi tiết hoá đơn sau khi thanh tonas | |
| Tác nhân | Khách hàng | |
| Điều kiện trước | Đã thanh toán hoá đơn | |
| Điều kiện sau | Xem được chi tiết hoá đơn | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Mở ứng dụng hoặc trang web để xem mail |  |
| 2. Chọn mail có hoá đơn người dùng muốn xem |  |
|  | 3. Hiển thị chi tiết hoá đơn của khách hàng muốn xem |
| **Luồng sự kiện phụ (Alternative flows)** |  |  |

Bảng 3.27 Mô tả Use case xem chi tiết hoá đơn

#### Activity diagram



Hình 3.59 Activity xem chi tiết hoá đơn

#### Sequence diagram



Hình 3.60 Sequence xem chi tiết hoá đơn

### Đặc tả UC26\_Xem bài viết

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Xem bài viết | | |
| **Mô tả** | Người dùng chọn một bài viết trên giao diện website để xem thông tin về bài viết đó. | |
| **Tác nhân** | Người dùng | |
| **Điều kiện trước** | Đang mở website, ở giao diện chính hoặc giao diện danh sách bài viết | |
| **Điều kiện sau** | Hiển thị nội dung bài viết lên giao diện | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Click chọn bài viết muốn xem thông tin. |  |
|  | 2. Hiển thị nội dung chi tiết bài viết mà tác nhân đã chọn bao gồm hình ảnh, nội dung tiêu đề |
| **Luồng sự kiện phụ (Alternative flows)** |  | 2.1. Hiện thị không tìm thấy trang |

Bảng 3.28 Mô tả Use case xem bài viết

#### Activity diagram



Hình 3.61 Activity xem bài viết

#### Sequence diagram



Hình 3.62 Sequence đăng nhập

### Đặc tả UC27\_Xem thông tin xe

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Xem thông tin xe | | |
| **Mô tả** | Người dùng chọn một xe trên giao diện website để xem thông tin về xe đó bao gồm tên xe, giá cả, mô tả. | |
| **Tác nhân** | Khách hàng | |
| **Điều kiện trước** | Đang mở website, ở giao diện chính hoặc giao diện danh mục xe | |
| **Điều kiện sau** | Hiển thị đúng nội dung xe được chọn bao gồm hình ảnh, tên xe, giá tiền, mô tả | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Click chọn xe muốn xem thông tin. |  |
|  | 2. Hiển thị nội dung chi tiết xe mà tác nhân đã click chọn bao gồm hình ảnh, tên xe, giá tiền, mô tả, x chưa được thuê |
| **Luồng sự kiện phụ (Alternative flows)** |  | 2.1. Hệ thống hiển thị trang không tìm thấy |

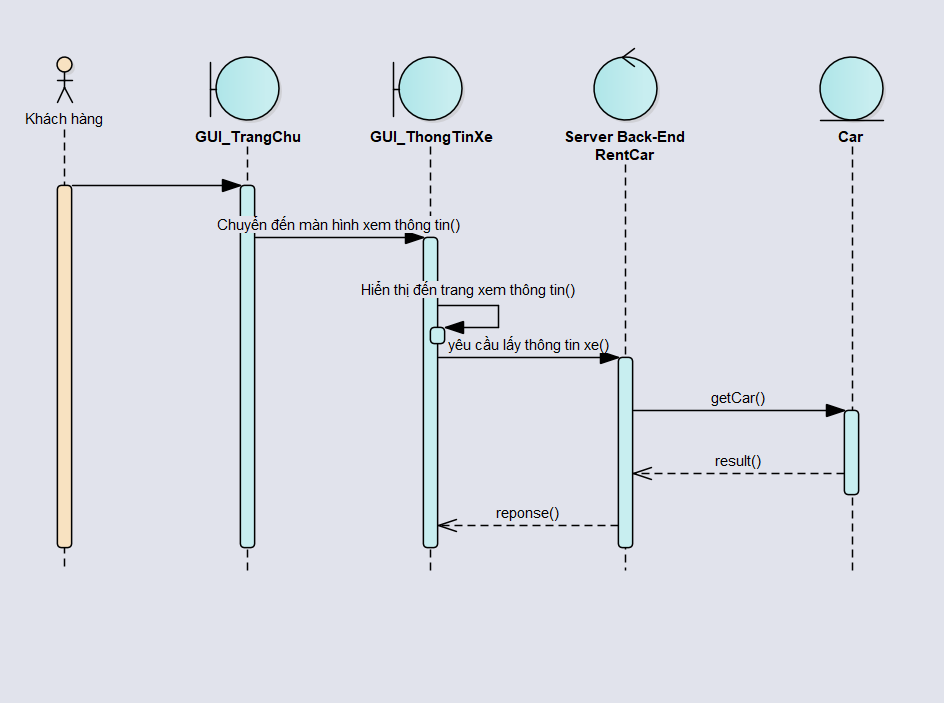
Bảng 3.29 Mô tả Use case xem thông tin xe

#### Activity diagram



Hình 3.63 Activity xem thông tin xe

#### Sequence diagram



Hình 3.64 Sequence xem thông tin xe

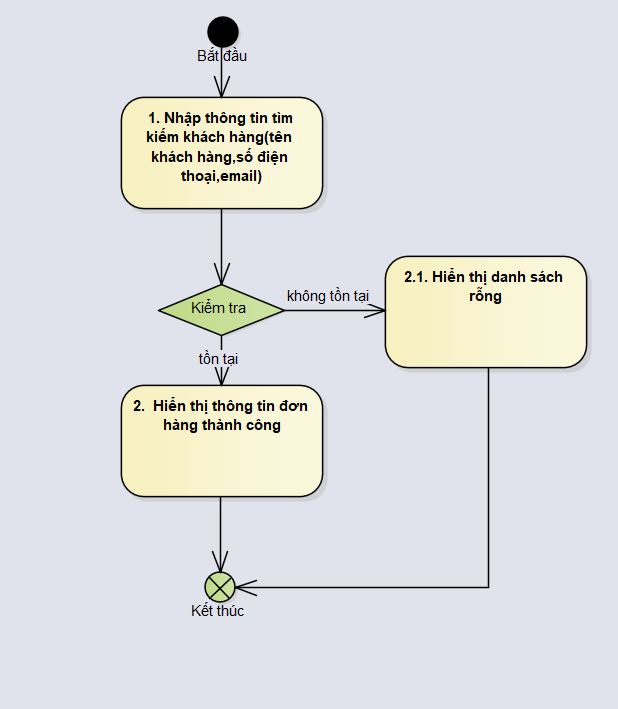
### Đặc tả UC28\_Tìm kiếm hoá đơn

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Tìm kiếm hoá đơn | | |
| **Mô tả** | Tìm kiếm thông tin nhân viên dựa trên email, số điện thoại, họ tên. | |
| **Tác nhân** | Nhân Viên | |
| **Điều kiện trước** | Đang mở website, ở giao diện danh sách đơn hàng. | |
| **Điều kiện sau** | Hiển thị thông tin tìm kiếm thành công. | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Nhập thông tin tìm kiếm khách hàng(tên khách hàng, số điện thoại, email) |  |
|  |  | 2. Hiển thị thông tin đơn hàng thành công |
| **Luồng sự kiện phụ (Alternative flows)** |  | 2.1. Hiển thị thông tin rỗng |

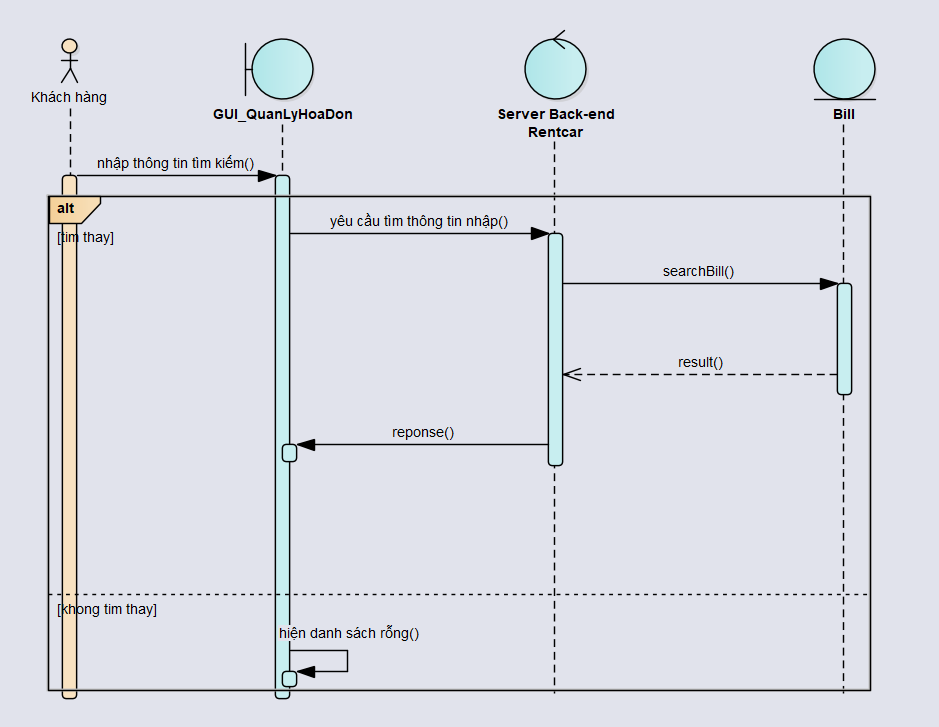
Bảng 3.30 Mô tả Use case tìm kiếm hoá đơn

#### Activity diagram



Hình 3.65 Activity tìm kiếm hoá đơn

#### Sequence diagram



Hình 3.66 Sequence tìm kiếm hoá đơn

### Đặc tả UC29\_Thống kê

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** thống kê | | |
| **Mô tả** | Người dùng chọn mốc thời gian để xem doanh thu của công ty | |
| **Tác nhân** | Admin | |
| **Điều kiện trước** | Đăng nhập thành công  Đang ở giao diện thống kê | |
| **Điều kiện sau** | Hiển thị biểu đồ thống kê | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Chọn mốc thời gian người dùng muốn xem |  |
| 2. Chọn xác nhận |  |
|  | 3. Hiển thị biểu đồ dựa theo số liệu |
| **Luồng sự kiện phụ (Alternative flows)** |  |  |

Bảng 3.30 Mô tả Use case thống kê

#### Activity diagram



Hình 3.67 Activity thống kê

#### Sequence diagram



Hình 3.68 Sequence tìm kiếm hoá đơn

### Đặc tả UC30\_Cập nhật thông tin

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** cập nhật thông tin | | |
| **Mô tả** | Người dùng cập nhật thông tin email, mật khẩu | |
| **Tác nhân** | Nhân viên, quản lý, admin | |
| **Điều kiện trước** | Đã đăng nhập  Đang ở giao diện thông tin cá nhân | |
| **Điều kiện sau** | Cập nhật dữ liệu lên cơ sở dữ liệu và hiển thị lên giao diện | |
|  | Tác nhân | Hệ thống |
| **Luồng sự kiện chính (Basic flows)** | 1. Người dùng nhập thông tin cần sửa |  |
| 2.. Chọn lưu |  |
|  | 3. Hệ thống lưu lên cơ sở dữ liệu |
|  | 4. Thông báo thay đổi thành công |
| **Luồng sự kiện phụ (Alternative flows)** |  | 3.1. Thông báo thay đổi thất bại |

Bảng 3.30 Mô tả Use case cập nhật thông tin

#### Activity diagram



Hình 3.67 Activity cập nhật thông tin

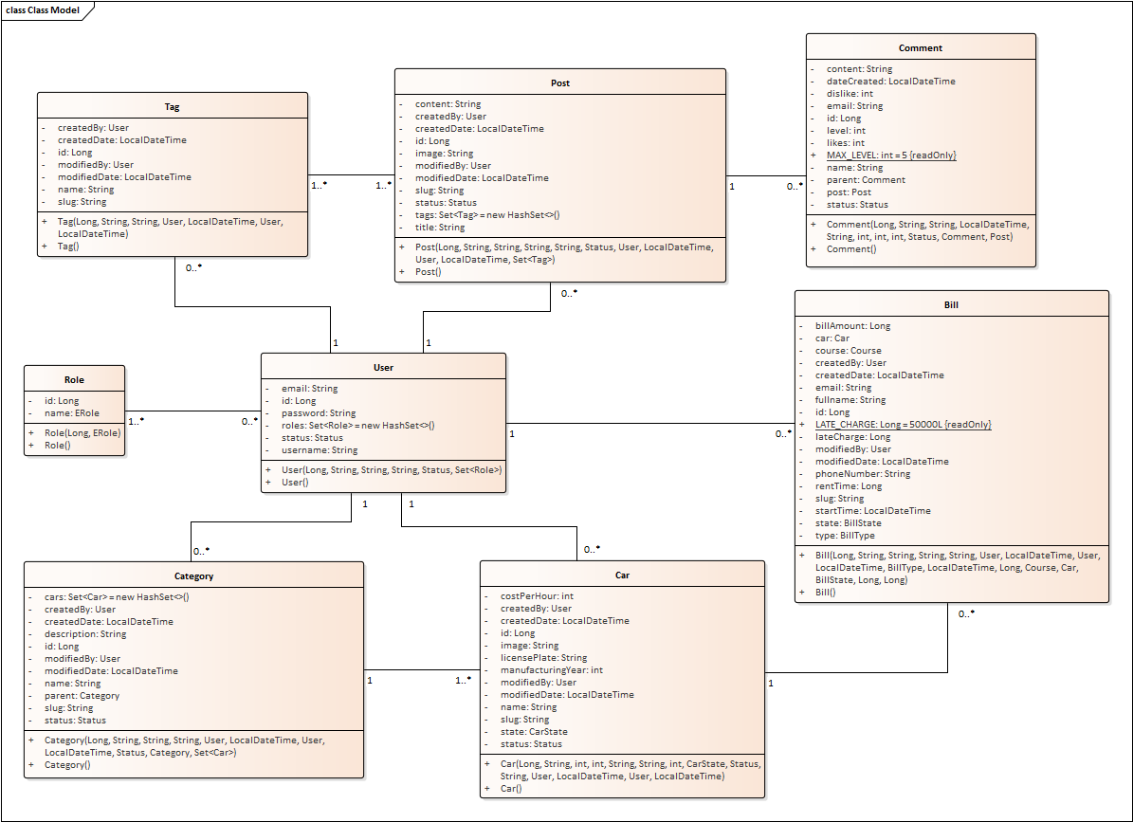
#### Sequence diagram



Hình 3.68 Sequence tìm kiếm hoá đơn

# THIẾT KẾ VÀ HIỆN THỰC

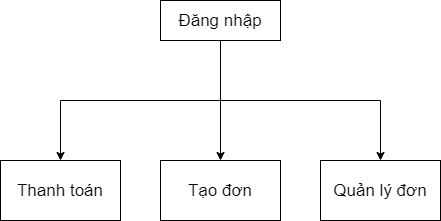
## Class diagram



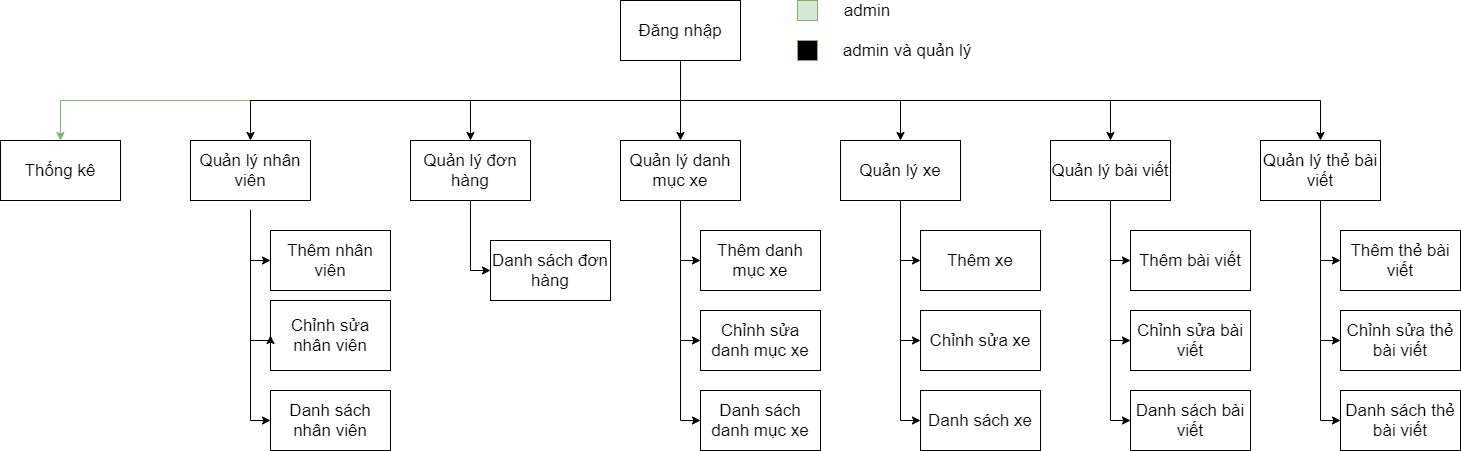
Hình 4.1 Class diagram

## Sitemap

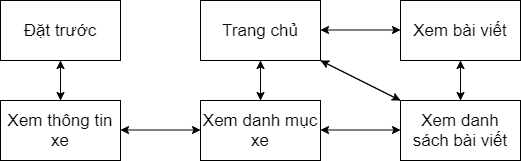
* Luồng màn hình Website nhân viên



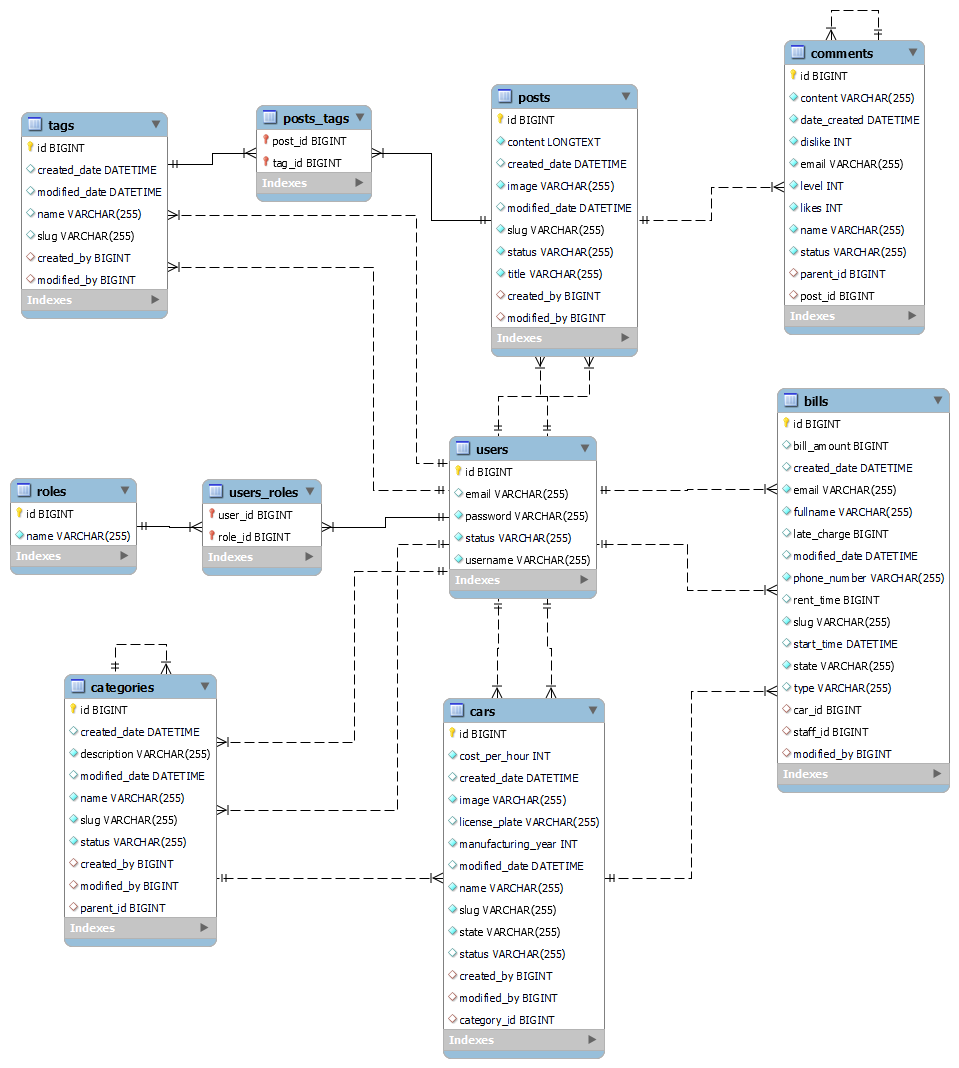
* Luồng màn hình Website quản lý và admin



* Luồng màn hình Website khách hàng



## Database diagram



Hình 4.2 Database diagram

## Hiện thực

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

Tìm hiểu thêm các công nghệ mới như:

* Spring framework (Spring Server-Sent Events, Spring security, Spring data jpa, Spring auditing)
* jQuery Ajax,
* AWS EC2, RDS,
* RESTful API

để áp dụng và xây dựng đề tài.

Sử dụng các kỹ năng lập trình và kiến thức về HTML, CSS và JavaScript, … để xây dựng GUI cho Website. Ngoài ra còn dùng các thư viện hỗ trợ góp phần xây dựng giao diện thân thiện hơn là Bootstrap, jQuery; sử dụng Ajax để có thể giao tiếp tới Back-End.

Sử dụng tốt kiến thức mới là Spring Boot RESTful API để áp dụng vào xây dựng một hệ thống Back-End ổn để giúp cho các chức năng trên trang web có sự ổn định và tính bảo mật. Ngoài ra còn hỗ trợ lưu giữ thông tin của người có tác động tới database nhằm mục đích truy vết khi có vấn đề xảy ra. Tất cả điều này đem tới trải nghiệm tuyệt vời và phong cách chuyên nghiệp của hệ thống ở cả phía giao diện và Back-End.

Website đã thực hiện thành công việc đăng nhập để phân quyền người dùng, nhằm việc bảo vệ sự an toàn của hệ thống tranh sự can thiệp từ người ngoài. Khi đăng nhập thành công thì website sẽ giao tiếp với Back-End thông qua một chuỗi token mà chỉ có Back-End mới có chữ ký để xác thực được chuỗi token đó.

Về phía Back-End thì đã được xây dựng một hệ thống dựa trên Spring Boot giúp cho việc đảm bảo tính ổn định của hệ thống, ngoài ra còn có thể trả về các thông tin lỗi về cho website. Đặc biệt là sử dụng Server-Side Events để có thể giúp hệ thống thông báo tức thì khi có người đăng ký dịch vụ. Giúp hệ thống đồng bộ dữ liệu từ website, Back-End và database

## Hạn chế của đồ án

* Chưa tối ưu hoá code, server thuê từ AWS chưa tốt nên phản hồi còn chậm, hiệu suất của hệ thống chưa cao
* Giao diện Website chưa tối ưu hơn bằng các framework hiện có như Angular 2, ReactJs, hay VueJs.
* Còn nhiều tính năng có thể mở rộng nhưng do thời gian không cho phép cải thiện chương trình

## Hướng phát triển

* Hoàn thiện giao diện website bằng framework như ReactJs, để có thể nhanh chóng phát triển tiếp lên thiết bị di động.
* Tối ưu hoá code hơn để đạt hiệu suất cao và khả năng xử lý nhanh chóng.
* Chia nhỏ hệ thống thành microservice để giúp cho việc quản lý database được chia nhỏ thành nhiều phần, bên cạnh đó sử dụng database no sql sẽ giúp cho việc tìm kiếm và lấu thông tin lên giao diện một cách nhanh hơn đối với các dữ liệu lớn.
* Phát triển thêm các dịch vụ như, cho thuê xe với nhu cầu khác ngoài tập lái. Mở thêm dịch vụ cho thuê xe máy, học lái xe A1, A2, B1, B2, …

# TÀI LIỆU THAM KHẢO

Tài liệu tiếng việt

1. Phần mềm Web application là gì?, [http://www.dzr-web.com/phan-mem-web-application-la-gi/.](http://www.dzr-web.com/phan-mem-web-application-la-gi/)
2. HTML là gì?, <https://www.hostinger.vn/huong-dan/html-la-gi>.
3. CSS là gì?, <https://www.hostinger.vn/huong-dan/css-la-gi>.
4. JavaScript là gì?, <https://www.hostinger.vn/huong-dan/javascript-la-gi>.
5. Java là gì?, <https://viettuts.vn/java>.
6. Spring là gì?, <https://viblo.asia/p/tong-quan-ve-spring-framework-YWOZryEyKQ0>.
7. Spring Boot là gì?, <https://stackjava.com/spring/spring-boot-la-gi-gioi-thieu-spring-boot-framework.html>.
8. RESTful API là gì?, <https://topdev.vn/blog/restful-api-la-gi/#restful-api-la-gi>.
9. MySQL là gì?, <https://www.hostinger.vn/huong-dan/mysql-la-gi/>.
10. Các đặc điểm của Amazon EC2, <https://viblo.asia/p/tim-hieu-ve-amazon-ec2-maGK7jRe5j2>.
11. Tìm hiểu tổng quan về jQuery Ajax, <https://viblo.asia/p/tim-hieu-tong-quan-ve-jquery-ajax-PwRGgmmrkEd>.

Tài liệu tiếng anh

1. AWS là gì?, <https://aws.amazon.com/vi/what-is-aws/>.
2. Amazon RDS là gì? Các tính năng của Amazon RDS, <https://aws.amazon.com/vi/rds/faqs/>.
3. Amazon EC2 là gì, <https://aws.amazon.com/vi/rds/features/>.