**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI**

**🙞…🕮…🙜**

**BÁO CÁO KẾT QUẢ THỰC TẬP TỐT NGHIỆP**

**ĐỀ TÀI: WEB ĐẶT VÉ XE TRỰC TUYẾN**

Giảng viên hướng dẫn:

Nhóm sinh viên: MSSV:

Lớp sinh hoạt:

Ngành: **CÔNG NGHỆ THÔNG TIN**

Khoa:

*Đà Nẵng,2021*

**LỜI CẢM ƠN**

Đà Nẵng, ngày 30 tháng 04 năm 2022

***Sinh viên thực hiện***

***Huỳnh Văn Phú***

**LỜI CAM ĐOAN**

Đà Nẵng, ngày 30 tháng 4 năm 2022 Người cam đoan

Huỳnh Văn Phú

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

*Đà Nẵng, ngày 30 tháng 05 năm 2022*

**Cán bộ hướng dẫn**

**NHẬN XÉT CỦA HỘI ĐỒNG CHẤM THI**

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

..........................................................................................................................................

*Đà Nẵng, ngày 30 tháng 05 năm 2022*

**Cán bộ chấm thi**

MỤC LỤC

[MỤC LỤC 6](#_Toc102546179)

[LỜI MỞ ĐẦU 10](#_Toc102546180)

[**1.** **Lý Do Chọn Đề Tài** 11](#_Toc102546181)

[2. Mục đích nghiên cứu 11](#_Toc102546182)

[Nhiệm vụ nghiên cứu 12](#_Toc102546183)

[3. Đối tượng và phạm vi nghiên cứu 12](#_Toc102546184)

[*Phạm vi nghiên cứu* 12](#_Toc102546185)

[4. Phương pháp nghiên cứu 13](#_Toc102546186)

[*Phương pháp nghiên cứu lý luận* 13](#_Toc102546187)

[*Phương pháp nghiên cứu thực nghiệm* 13](#_Toc102546188)

[5. Cấu trúc khóa luận 13](#_Toc102546189)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 14](#_Toc102546190)

[**1.1.** **Khái niệm Frontend** 14](#_Toc102546191)

[**1.1.1.** **HTML** 14](#_Toc102546192)

[**1.1.2.** **CSS** 16](#_Toc102546193)

[**1.1.3.** **SASS/SCSS** 17](#_Toc102546194)

[**1.1.4.** **Tailwind css** 20](#_Toc102546195)

[**1.1.4.1.** **Tại sao nên dùng Tailwind CSS** 21](#_Toc102546196)

[**1.1.4.2.** **Cài đặt tailwind css** 22](#_Toc102546197)

[**1.1.5.** **Javascript** 26](#_Toc102546198)

[**1.1.5.1.** **Chương trình cơ bản của javascript** 27](#_Toc102546199)

[**1.1.6.** **Reactjs** 29](#_Toc102546200)

[Tại sao các JavaScript developer lại sử dụng ReactJS? 30](#_Toc102546201)

[**1.1.6.1.** **JSX** 31](#_Toc102546202)

[**1.1.6.2.** **Virtual DOM** 32](#_Toc102546203)

[**1.1.6.3.** **Ưu điểm khi dùng framework Reactjs** 34](#_Toc102546204)

[**1.2.** **Khái niệm về Backend** 35](#_Toc102546205)

[**1.2.1.** **Nodejs** 36](#_Toc102546206)

[**1.2.1.1.** **Cấu trúc Nodejs** 36](#_Toc102546207)

[**1.2.1.2.** **Nodejs framework** 37](#_Toc102546208)

[**1.2.1.3.** **Tại sao nên chọn nodejs** 39](#_Toc102546209)

[**1.2.2.** **Hệ quản trị cơ sở dữ liệu** 40](#_Toc102546210)

[**1.2.2.1.** **Giới thiệu về mysql** 40](#_Toc102546211)

[**1.2.2.2.** **Một số công dụng chính của MySQL** 42](#_Toc102546212)

[**1.3.** **Một số thư viện hỗ trợ** 46](#_Toc102546213)

[**1.3.1.** **Ant Design** 46](#_Toc102546214)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 49](#_Toc102546215)

[2.1. Phân tích hệ thống 49](#_Toc102546216)

[2.1.1. Giới thiệu hệ thống 49](#_Toc102546217)

[2.1.2. Xác định chức năng chính của hệ thống 50](#_Toc102546218)

[2.1.3. Tác nhân của hệ thống 50](#_Toc102546219)

[2.1.4. Xác định các Use case, các gói Use case và vẽ biểu đồ Use case tổng quát](#_Toc102546220)

[**2.2.** **Biểu đồ usecase** 52](#_Toc102546221)

[**2.2.1.** **Use case người dùng – khách hàng** 53](#_Toc102546222)

[**2.2.2.** **Use case người quản trị** 53](#_Toc102546223)

[**2.2.4.** **Use case đăng ký** 55](#_Toc102546224)

[**2.2.5.** **Use case thanh toán** 55](#_Toc102546225)

[**2.2.6.** **Use case đặt vé** 56](#_Toc102546226)

[**2.2.7.** **Use case quản lý vé** 57](#_Toc102546227)

[**2.2.8.** **Use case quản lý thông tin người dung** 57](#_Toc102546228)

[**2.2.9.** **Use case quản lý điểm trạm** 58](#_Toc102546229)

[**2.2.10.** **Use case quản lý lịch trình chuyến xe** 59](#_Toc102546230)

[**2.2.11.** **Use case quản lý xe** 60](#_Toc102546231)

[**2.2.12.** **Use case quản lý nhà xe** 61](#_Toc102546232)

[**2.2.13.** **Use case thống kê** 62](#_Toc102546233)

[**2.3.** **Biểu đồ tuần tự** 63](#_Toc102546234)

[***Biểu đồ tuần tự của khách hàng*** 63](#_Toc102546235)

[**2.3.1.** **Biểu đồ tuần tự đăng nhập** 63](#_Toc102546236)

[**2.3.2.** **Biểu đồ tuần tự đăng ký** 64](#_Toc102546237)

[**2.3.3.** **Biểu đồ tuần tự đặt vé** 64](#_Toc102546238)

[**2.3.4.** **Biểu đồ tuần tự hủy vé** 65](#_Toc102546239)

[**2.3.5.** **Biểu đồ tuần tự sửa thông tin cá nhân** 65](#_Toc102546240)

[**2.3.6.** **Biểu đồ tuần tự thanh toán** 66](#_Toc102546241)

[**2.3.7.** **Biểu đồ tuần tự đăng xuất** 67](#_Toc102546242)

[***Biểu đồ tuần tự cho quản trị viên*** 68](#_Toc102546243)

[**2.3.8.** **Biểu đồ tuần tự quản lý vé** 68](#_Toc102546244)

[**2.3.9.** **Biểu đồ tuần tự quản lý xe** 69](#_Toc102546245)

[**2.3.10.** **Biều đồ tuần tự quản lý thông tin cá nhân** 69](#_Toc102546246)

[**2.3.11.** **Biều đồ tuần tự quản lý lịch trình chuyến đi** 70](#_Toc102546247)

[**2.3.12.** **Biểu đồ tuần tự quản lý nhà xe** 71](#_Toc102546248)

[**2.3.13.** **Biểu đồ tuần tự quản lý trạm** 72](#_Toc102546249)

[**2.3.14.** **Biểu đồ tuần tự quản lý các điểm đón / trả** 73](#_Toc102546250)

[**2.4.** **Biểu đồ hoạt động** 74](#_Toc102546251)

[**2.4.1.** **Biểu đồ hoạt động đăng nhập** 74](#_Toc102546252)

[**2.4.2.** **Biểu đồ hoạt động đăng ký** 75](#_Toc102546253)

[**2.4.3.** **Biểu đồ hoạt động đặt vé** 76](#_Toc102546254)

[**2.4.4.** **Biều đồ hoạt động thanh toán** 77](#_Toc102546255)

[**2.4.5.** **Biều đồ hoạt động hủy vé** 78](#_Toc102546256)

[**2.4.6.** **Biểu đồ hoạt động quản lý vé** 79](#_Toc102546257)

[**2.4.7.** **Biểu đồ hoạt động thêm chuyến đi** 81](#_Toc102546258)

[**2.4.8.** **Biểu đồ hoạt động sửa chuyến đi** 81](#_Toc102546259)

[**2.4.9.** **Biểu đồ hoạt động xóa chuyến đi** 82](#_Toc102546260)

[**2.5.** **Thiết kế hệ thống** 83](#_Toc102546261)

[**2.5.1.** **Thiết kế cơ sở dữ liệu** 83](#_Toc102546262)

[**2.5.1.1.** **Sơ đồ tổng quan** 83](#_Toc102546263)

[**2.5.1.2.** **Bảng user ( người dùng )** 84](#_Toc102546264)

[**2.5.1.3.** **Bảng ticket (vé)** 85](#_Toc102546265)

[**2.5.1.4.** **Bảng vehicle ( xe)** 85](#_Toc102546266)

[**2.5.1.5.** **Bảng chuyến đi** 85](#_Toc102546267)

[**2.5.1.6.** **Bảng chuyến xe** 86](#_Toc102546268)

[**2.5.1.7.** **Bảng trạm ( station)** 86](#_Toc102546269)

[**2.5.1.8.** **Bảng nhà xe** 87](#_Toc102546270)

[**2.5.1.9.** **Bảng ghế (seat)** 87](#_Toc102546271)

[**2.5.1.10.** **Bảng điểm dừng , điểm trả (Point)** 87](#_Toc102546272)

[**2.5.2.** **Thiết kế module** 88](#_Toc102546273)

[**Module quản lý nhà xe**: Module này thực hiện các nhiệm vụ quản lý hãng xe. Chức năng cảu module này gồm có: 88](#_Toc102546274)

[Module quản lý xe: Module này thực hiện nhiệm vụ quản lý xe. Chức năng của module này gồm có: 88](#_Toc102546275)

[**Module quản lý loại ghế**: Module này thực hiện các nhiệm vụ quản lý loại ghế. Chức năng của module này gồm có: 88](#_Toc102546276)

[**Module quản lý tuyến xe**: Module này thực hiện các nhiệm vụ quản lý tuyến xe. Chức năng của module này gồm có: 88](#_Toc102546277)

[**Module quản lý điểm xuất phát và điểm đến**: Module này thực hiện các nhiệm vụ quản lý điểm xuất phát và điểm đến. Chức năng của module này gồm có: 88](#_Toc102546278)

[**Module thống kê báo cáo**: Module này thực hiện các nhiệm vụ thống kê báo cáo đơn đặt vé qua mạng của hành khách theo nhiều tiêu chí khác nữa. Chức năng của module này gồm có: 89](#_Toc102546279)

[**Module quản trị người dùng**: Module này thực hiện các nhiệm vụ quản trị người dùng. Chức năng của module này gồm có: 89](#_Toc102546280)

[**Module đặt vé**: Module này thực hiện các nhiệm vụ cho phép khách hàng tìm kiếm và thực hiện việc đặt vé. Chức năng của module này gồm có: 89](#_Toc102546281)

[CHƯƠNG 3 : CÀI ĐẶT VÀ XÂY DỰNG CHƯƠNG TRÌNH 90](#_Toc102546282)

[3.1. Ý tưởng giải quyết bài toán 90](#_Toc102546283)

[**3.2.** **Cách thức thực hiện** 91](#_Toc102546284)

[**3.3.** **Kết quả đat được** 96](#_Toc102546285)

[**3.3.1.** **Màn hình đăng nhập** 96](#_Toc102546286)

[**3.3.2.** **Màn hình đăng ký tài khoản** 96](#_Toc102546287)

[**3.3.3.** **Màn hình trang chủ** 97](#_Toc102546288)

[**3.3.4.** **Màn hình danh sách chuyến đi** 98](#_Toc102546289)

[**3.3.5.** **Màn hình thông tin cá nhân** 98](#_Toc102546290)

[**3.3.6.** **Màn hình thông tin vé người dùng** 99](#_Toc102546291)

[**3.3.7.** **Màn hình chọn ghế ngồi** 100](#_Toc102546292)

[**3.3.8.** **Màn hình thanh toán** 101](#_Toc102546293)

[**3.3.9.** **Màn hình quản trị** 102](#_Toc102546294)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 103](#_Toc102546295)

[Kết quả đạt đuợc: 104](#_Toc102546296)

[Hạn chế 105](#_Toc102546297)

[Tài liệu tham khảo 106](#_Toc102546298)

# LỜI MỞ ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của công nghệ thông tin và những ứng dụng của nó trong đời sống. Máy tính điện tử không còn là một thứ phương tiện lạ lẫm đối với mọi người mà nó dần trở thành một công cụ làm việc và giải trí thông dụng và hữu ích của chúng ta, không chỉ ở công sở mà còn ngay cả trong gia đình.

Với sự phát triển của công nghệ hiện nay đã bao trùm cả thế giới , việc sử dụng những phương tiện, công cụ, web, app đã xuất hiện rộng rãi.Trong nền kinh tế hiện nay, với xu thế toàn cầu hoá nền kinh tế thế giới, mọi mặt của đời sống xã hội ngày càng được nâng cao, đặc biệt là nhu cầu trao đổi hàng hoá của con người ngày càng tăng cả về số lượng và chất lượng.Hiện nay các công ty tin học hàng đầu thế giới không ngừng đầu tư và cải thiện các giải pháp cũng như các sản phẩm nhằm cho phép tiến hành thương mại hóa trên Internet. Thông qua các sản phẩm và công nghệ này, chúng ta dễ dàng nhận ra tầm quan trọng và tính tất yếu của thương mại điện tử. Với những thao tác đơn giản trên máy có nối mạng Internet bạn sẽ có tận tay những gì mình cần mà không phải mất nhiều thời gian. Bạn chỉ cần vào các trang dịch vụ thương mại điện tử, làm theo hướng dẫn và click vào những gì bạn cần. Các nhà dịch vụ sẽ mang đến tận nhà cho bạn. Để tiếp cận và góp phần đẩy mạnh sự phổ biến của thương mại điện tử ở Việt Nam. Chúng em đã tìm hiểu và lựa chọn đề tài Nghiên cứu khoa học “**Xây dựng web đặt vé xe trực tuyến”** nhằm đem lại nguồn lợi ích từ công nghệ thông tin vào đời sống

1. **Lý Do Chọn Đề Tài**

Trong đại dịch covid 19 nhiều người dân , người lao động , người ở các địa phương khác phải ở lại tại nơi để phòng chống lay lan dịch bệnh gây ra các tình trạng thiếu việc làm nhưng lại không thể về quê. Từ đó kéo theo sự di chuyển chỗ ở, chỗ làm việc của rất nhiều người. Mọi người có nhu cầu đi lại ngày càng nhiều. Mặt khác, do kinh tế phát triển nên nhu cầu đi thăm quan, thăm viếng người nhà ở xa tăng.

Trên thực tế nhu cầu đi lại của người dân tăng đột biến. Và với cách mua và bán vé xe truyền thống đã không đáp ứng được nhu cầu bức xúc đó. Thường diễn ra cảnh chen lấn xô đẩy để mua vé. Từ thực tế đó đã gây cho người dân rất nhiều bức xúc như chờ vài tiếng mà không mua được vé, đến lượt mua vé thì được thông báo hết vé. Còn đối với các công ty vận tải thì cũng gặp khó khăn trong việc tổ chức bán vé xe. Cảnh chen lấn xô đẩy đó đã tạo điều kiện cho bọn móc túi, cướp giật, bán vé chợ đen hoạt động. Càng làm cho tình hình thêm tồi tệ, người dân và doanh nghiệp càng thêm bức xúc.

Từ những bức xúc đó, nên tôi đã quyết định chọn đề tài khóa luận tốt nghiệp là xây dựng hệ thống website đặt vé xe khách chất lượng cao. Hệ thống sẽ giải quyết được những khó khăn trên. Khi mà công nghệ thông tin phát triển mạnh, mạng internet về tận từng hộ gia đình, người dân thường xuyên tiếp xúc với máy tính, mạng internet thì hệ thống ra đời là rất phù hợp với tình hình thực tiễn. Đặc biệt với những người bận rộn không có thời gian ra bến xe mua vé thì với những cái click chuột mà mua được vé xe thì điều này thật có ý nghĩa. Khi hệ thống đưa vào hoạt động không chỉ mạng lại sự tiện lợi cho người dân trong việc mua vé xe mà còn giúp các công ty vận tải phục vụ hành khách tốt hơn.

## Mục đích nghiên cứu

* 1. Phân tích và thiết kế một hệ thống bán và quản lý vé xe khách trực tuyến.
  2. Triển khai ứng dụng bằng một phần mềm ứng dụng cụ thể.
  3. Triển khai kiểm tra bằng thực nghiệm về việc xử lý các vấn đề trong công tác quản lý như: Số hành khách, quản lý số vé, hành trình của các tuyến xe...

### Nhiệm vụ nghiên cứu

* Làm sáng tỏ cở sở lý luận của đề tài
* Làm rõ hoạt động của hệ thống quản lý và bán vé xe online hiện nay.
* Khảo sát tình hình thực tế của hệ thống hiện tại

## Đối tượng và phạm vi nghiên cứu

*Đối tượng nghiên cứu*

Đối tượng nghiên cứu là việc quản lý, cách thức tổ chức hoạt động của hệ thống bán vé xe online.

### *Phạm vi nghiên cứu*

Hệ thống được nghiên cứu về các nhu cầu thực tế của khách hàng là tìm kiếm xe và mua được vé, giá cả của vé cũng như thời gian chuyến xe đi là nhanh nhất.

Ý nghĩa khoa học và thực tiễn

Chương trình được xây dựng đã áp dụng thành tựu của công nghệ thông tin vào công tác xây dựng và quản lý một hệ thống giao thông trực tuyến, góp phần vào công cuộc xây dựng một nền giao thông an toàn không chỉ tại địa bàn mà còn rộng ra khắp cả nước.

Chương trình được xây dựng giúp giảm thiểu hoạt động làm việc của nhà quản lý, hành khách có thể dễ dàng mua được vé mà không phải chen lấn xếp hàng, các nhà xe cũng dễ dàng kiểm soát vé cũng như thông tin của từng hành khách.

## Phương pháp nghiên cứu

### *Phương pháp nghiên cứu lý luận*

Nghiên cứu qua việc đọc sách, báo, tài liệu liên quan nhằm xây dựng cơ sở lý thuyết của đề tài và các biện pháp cần thiết để giải quyết các vấn đề của đề tài.

### *Phương pháp nghiên cứu thực nghiệm*

Thông qua quan sát thực tế, yêu cầu của cơ sở những lý luận được nghiên cứu, phần mềm sẽ được đưa vào chạy thử nghiệm thực tế để từ đó có những điều chỉnh cho phù hợp hơn.

## Cấu trúc khóa luận

Ngoài phần mở đầu và kết luận, cấu trúc khóa luận gồm 3 chương:

* **Chương 1. Cơ sở lý thuyết**
* **Chương 2. Phân tích và thiết kế hệ thống**
* **Chương 3. Xây dựng và cài đặt chương trình**

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

* 1. **Khái niệm Frontend**
* Front End (còn được biết đến như client-side) là tất cả những gì liên quan đến điều mà người dùng nhìn thấy mỗi khi truy cập vào một trang web, bao gồm phạm trù thiết kế và các ngôn ngữ như HTML hay CSS.
* Người dùng tương tác trực tiếp với nhiều khía cạnh thuộc front-end như: nhận biết vị trí đặt để của logo, màu sắc chủ đạo, tìm kiếm và đọc thông tin, sử dụng các button và tính năng trên web… Mục đích cuối cùng của Front End là nhằm mang lại một giao diện bắt mắt, giúp người dùng dễ dàng thao tác và sử dụng.
  + 1. **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, vâng vâng.
  + 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>
  + <p>Đây là cách bạn thêm đoạn văn trong HTML.</p>
  + <p>Bạn có thể thêm nhiều đoạn văn!</p>
  + 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.

**Ưu điểm:**

* + 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.

**Khuyết điểm:**

* + Được dùng chủ yếu cho web tĩnh. Đối với các tính năng động, bạn cần sử dụng JavaScript hoặc ngôn ngữ backend bên thứ 3 như PHP.
  + Nó có thể thực thi một số logic nhất định cho người dùng. Vì vậy, hầu hết các trang đều cần được tạo riêng biệt, kể cả khi nó sử dụng cùng các yếu tố, như là headers hay footers.
  + Một số trình duyệt chậm hỗ trợ tính năng mới.
  + Khó kiểm soát cảnh thực thi của trình duyệt (ví dụ, những trình duyệt cũ không render được tag mới)
    1. **CSS**
* CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…
* Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.
  + 1. **SASS/SCSS**
* SASS/SCSS là một chương trình tiền xử lý CSS (CSS preprocessor). Nó giúp bạn viết CSS theo cách của một ngôn ngữ lập trình, có cấu trúc rõ ràng, rành mạch, dễ phát triển và bảo trì code hơn. Ngoài ra nó có rất nhiều các thư viện hỗ trợ kèm theo giúp bạn viết code CSS một cách dễ dàng vào đơn giản hơn. Có rất nhiều loại CSS Preprocessor trong đó bao gồm SASS, Stylus hay LESS.
* ***SASS*** và ***SCSS*** về bản chất vấn đề là giống nhau, chỉ khác nhau ở cách viết
* ***Sass*** là chữ viết tắt của Syntactically Awesome Style Sheets, chương trình tiền xử lý bằng ngôn ngữ kịch bản (Preprocessor Scripting Language ), sẽ được biên dịch thành ***CSS***. Nghĩa là, mình sẽ làm style bằng SASS, rồi ***SASS*** sẽ render việc mình làm thành file ***CSS.***
* ***SASS*** bản thân có hai kiểu viết khác nhau, một kiểu như là HAML, Pug – Sử dụng indent (cách thụt đầu dòng) để phân tách các khối code , sử dụng xuống dòng để phân biệt rules , có phần mở rộng là .sass.
* Giả sử bạn chỉ muốn CSS cho thẻ ul với class menu, với CSS thuần bạn viết
* .navbar ul.menu {
* list-style: none;
* }
* Nếu bạn tiếp tục muốn CSS cho thẻ li trong thẻ ul (có class là menu) thì
* .navbar ul.menu li {
* padding: 5px;
* }
* Rồi bạn lại tiếp tục muốn CSS cho thẻ a trong thẻ li... thì cứ phải viết lặp đi lặp lại tên tag (hoặc class, hoặc id) cha của thẻ muốn css thì rất cực phải không nào ?
* Nested Rule của SASS sinh ra để giúp bạn làm điều trên một cách đơn giản hơn.
* Cú pháp viết như sau:
* .navbar {
* ul.menu {
* list-style: none;
* li {
* padding: 5px;
* a {
* text-decoration: none;
* }
* }
* }
* }
* Và sau khi được đoạn SASS trên được compile ra CSS thuần sẽ như sau:
* .navbar ul.menu {
* list-style: none;
* }
* .navbar ul.menu li {
* padding: 5px;
* }
* .navbar ul.menu li a {
* text-decoration: none;
* }
* Thực tế quy tắc xếp chồng được sử dụng rất nhiều khi vào 1 project có viết css bằng SASS
  + 1. **Tailwind css**

****

Là một utility-first CSS framework, nó cũng giống như Bootstrap, nó có những class built-in mà chúng ta có thể dùng. Tailwind CSS có nhiều các class bao gồm các thuộc tính CSS khác nhau và quan trọng, chúng ta có thể dễ dàng mở rộng tạo mới ra những class bằng chính những class của nó.

* + - 1. **Tại sao nên dùng Tailwind CSS**

Tailwind giống với Boostrap nhưng một điều tiện lợi khi dùng framework này là chúng ta có nhiều class mới hơn tiện lợi hơn Boostrap. Các bạn có thể tham khảo tại đây trong Tailwind. Và hơn nữa, việc có nhiều thêm những class nhưng với quy tắc đặt tên cực kỳ thân thiện với người dùng, người dùng cũng có thể nhìn vào class đó và có thể biết được class này nó đang style cái gì. Chúng ta cũng phải nói đến khả năng tùy biến và mở rộng cao, đem đến cho ta sự linh hoạt không giới hạn.

* + - 1. **Cài đặt tailwind css**
* Đầu tiên các bạn phải tạo một Nextjs project
* npm init -y
* npm install --save next react react-dom @zeit/next-css
* Tiếp theo chúng ta sẽ cài tailwind và một vài các package khác được sử dụng để complie CSS
* npm install tailwindcss autoprefixer postcss-loader --save-dev
* Và chúng ta cũng cần cài thêm next-css plugin
* npm install @zeit/next-css
* Khi chúng ta init nextjs project xong rồi, thì chúng ta sẽ tạo 1 file postcss.config.js để require 2 plugin tailwindcss và autoprefixer vào nhé
* module.exports = {
* plugins: [
* require('tailwindcss'),
* require('autoprefixer'),
* ],
* }
* Tiếp theo trong file next.config.js chúng ta sẽ config như sau
* import path from 'path'
* import fs from 'fs'
* import withPlugins from 'next-compose-plugins'
* import css from '@zeit/next-css'
* import less from '@zeit/next-less'
* import lessToJS from 'less-vars-to-js'
* import progressBar from 'next-progressbar'
* import nextConfig from './next.base.config'
* export default withPlugins([
* progressBar,
* css,
* [less, lessConfig],
* ], nextConfig)
* Trong file package.json các bạn cho đoạn script này vào như sau
* "scripts": {
* "dev": "next",
* "build": "next build",
* "start": "next start"
* }
* Trong thư mục assets ở trong project các bạn tạo styles/less/tailwind.less
* @tailwind base;
* @tailwind utilities;
* @tailwind components;
* Sau đó các bạn tạo thêm app.less với nội dung
* @import "./libs/tailwind.less";
  + 1. **Javascript**
  + JavaScript được tạo trong mười ngày bởi Brandan Eich, một nhân viên của Netscape, vào tháng 9 năm 1995. Được đặt tên đầu tiên là Mocha, tên của nó được đổi thành Mona rồi LiveScript trước khi thật sự trở thành JavaScript nổi tiếng như bây giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc quyền bởi Netscape và chỉ có các tính năng hạn chế, nhưng nó tiếp tục phát triển theo thời gian, nhờ một phần vào cộng đồng các lập trình viên đã liên tục làm việc với nó.
  + Trong năm 1996, JavaScript được chính thức đặt tên là ECMAScript. ECMAScript 2 phát hành năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999. Nó liên tục phát triển thành JavaScript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính bàn.

***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.
  + - 1. **Chương trình cơ bản của javascript**

Tất cả những đoạn mã Javascript đều phải đặt trong cặp thẻ mở <script> và thẻ đóng </script>, ví dụ:

1. <html>
2. <head>
3. <title></title>
4. <script language="javascript">
5. alert("Hello World!");
6. </script>
7. </head>
8. <body>
10. </body>
11. </html>

Bước 1: Bây giờ chúng ta thực hành nhé, bạn hãy tạo một file index.html và lưu tại bất kì vị trí nào, desktop hay ổ D, ổ C gì đó thì tùy bạn, miễn là phần mở rộng là .html, sau đó mở file đó bằng Notepad++ rồi gõ nội dung sau vào:

<html>

    <head>

        <title></title>

    </head>

    <body>

        <input type="button" value="Click Me"/>

    </body>

</html>

Bước 2: Viết mã Javascript khi click vào button có id="clickme" thì thông báo lên màn hình.

<html>

    <head>

        <title></title>

    </head>

    <body>

        <input type="button" id="clickme" value="Click Me"/>

        <script language="javascript">

        // Lấy element có id=clickme lưu vào biến button

        var button = document.getElementById('clickme');

        // Khi click vào element chứa trong button thì thực hiện một function,

        // bên trong function thông báo lên Hello World!

        button.addEventListener('click', function(){

            alert('Hello World!');

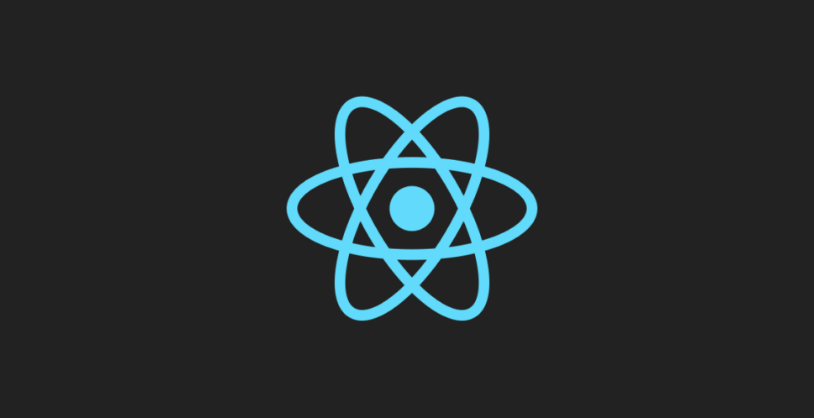
        });

        </script>

    </body>

</html

* + 1. **Reactjs**



ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

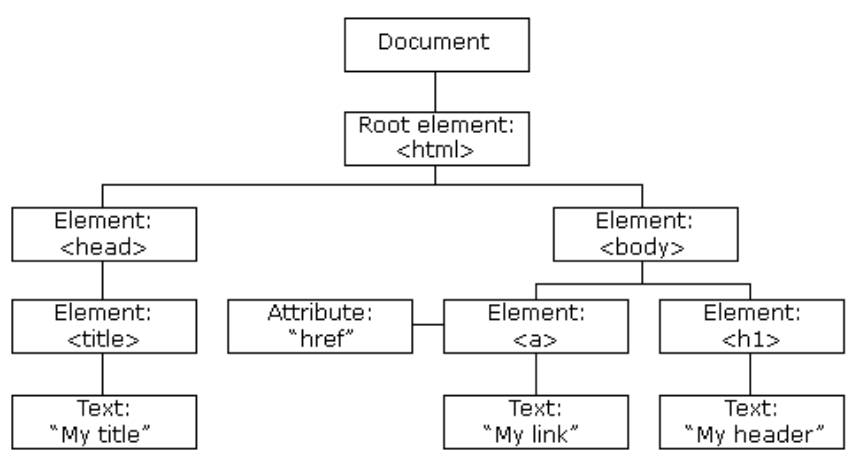
#### Tại sao các JavaScript developer lại sử dụng ReactJS?

* ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,… Giả sử bạn đang lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm ưng ý rồi và nhấn vào nút “Thêm vào giỏ hàng”, thì việc tiếp theo mà bạn phải làm đó là thêm sản phẩm được chọn vào giỏ hàng và hiển thị lại sản phẩm đó khi user vào xem => xử lý tương tác.
* Trước khi có ReactJS, lập trình viên thường gặp rất nhiều khó khăn trong việc sử dụng “vanilla JavaScript”(JavaScript thuần) và JQuery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển ứng dụng sẽ lâu hơn và xuất hiện nhiều bug, rủi ro hơn. Vì vậy vào năm 2011, Jordan Walke – một nhân viên của Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.
* Hơn nữa, để tăng tốc quá trình phát triển và giảm thiểu những rủi ro có thể xảy ra trong khi coding, React còn cung cấp cho chúng ta khả năng Reusable Code (tái sử dụng code) bằng cách đưa ra 2 khái niệm quan trọng bao gồm:

**JSX.**

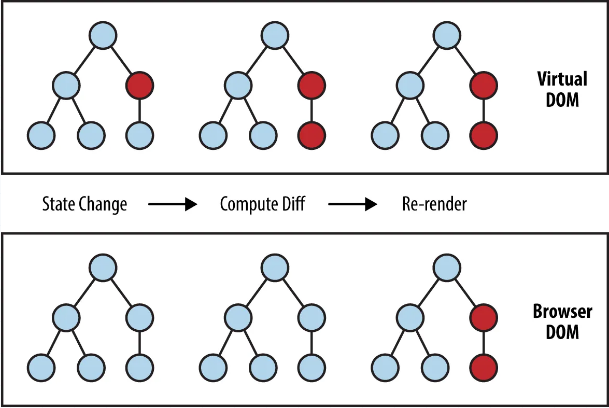
**Virtual DOM.**

* + - 1. **JSX**



* Trọng tâm chính của bất kỳ website cơ bản nào đó là những HTML documents. Trình duyệt Web đọc những document này để hiển thị nội dung của website trên máy tính, tablet, điện thoại của bạn. Trong suốt quá trình đó, trình duyệt sẽ tạo ra một thứ gọi là Document Object Model (DOM) – một tree đại diện cho cấu trúc website được hiển thị như thế nào. Lập trình viên có thể thêm bất kỳ dynamic content nào vào những dự án của họ bằng cách sử dụng ngôn ngữ JavaScript để thay đổi cây DOM.
* JSX (nói ngắn gọn là JavaScript extension) là một React extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản. Và kể từ lúc ReactJS browser hỗ trợ toàn bộ những trình duyệt Web hiện đại, bạn có thể tự tin sử dụng JSX trên bất kỳ trình duyệt nào mà bạn đang làm việc.
  + - 1. **Virtual DOM**

Khái niệm DOM ảo xuất hiện và hoạt động tốt hơn đáng kể so với DOM thực. DOM ảo chỉ là một đại diện ảo của DOM. Mỗi khi trạng thái ứng dụng của chúng tôi thay đổi, DOM ảo sẽ được cập nhật thay vì DOM thực.



Trong React, mỗi phần giao diện người dùng là một thành phần, và mỗi thành phần có một hoặc nhiều trạng thái. React tuân theo mô hình có thể quan sát được và lắng nghe các thay đổi trạng thái (observable pattern). Khi một trong các trạng thái của phần giao diện thay đổi, React cập nhật DOM ảo. Sau khi DOM ảo đã được cập nhật, React sẽ so sánh phiên bản DOM ảo hiện tại với phiên bản trước đó. Quá trình này được gọi là "diffing".

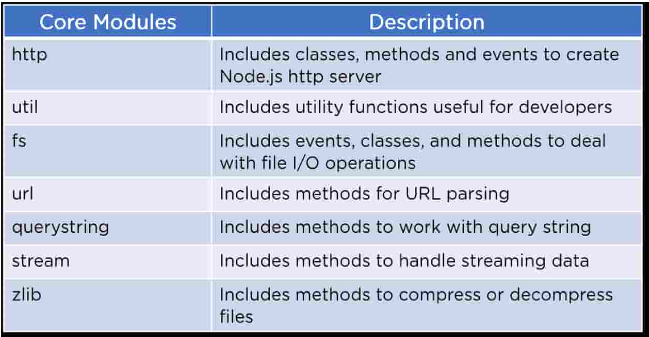
Khi React biết các đối tượng DOM ảo nào đã thay đổi, sẽ tiến hành cập nhật duy nhất đối tượng đó vào DOM thật. Điều này làm cho hiệu xuất nhanh hơn đáng kể khi so sánh với thao tác trực tiếp với DOM thực, làm cho React nổi bật như một thư viện JavaScript hiệu suất cao.

* + - 1. **Ưu điểm khi dùng framework Reactjs**
* Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật len DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí
* Khi đó ta có thể viết các methods, các funtions tác động lên component Hoge (thay đổi Model data), View sẽ lập tức được cập nhật cho dù ta không phải động chạm trực tiếp gì vào phần tử DOM trên View.
* Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể them vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
* Nó có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.
* Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dung sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dung vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
* Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
* Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.
  1. **Khái niệm về Backend**

BackEnd là tất cả những phần hỗ trợ hoạt động của website hoặc ứng dụng mà người dùng không thể nhìn thấy được. Có thể cho rằng BackEnd giống như bộ não của con người. Nó xử lý những yêu cầu, câu lệnh và lựa chọn thông tin chính xác để hiển thị lên màn hình.

BackEnd của bất kỳ website nào cũng được cấu thành từ 3 thành phần là máy chủ, ứng dụng và cơ sở dữ liệu. Nhờ có nó mà website hoạt động hiệu quả, cung cấp thông tin chính xác với tốc độ nhanh cho người dùng.

* + 1. **Nodejs**
  + Node.js là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.
  + Node.Js là môi trường máy chủ mã nguồn mở. (open source server enviroment)
  + Node.js hoàn toàn free.
  + Node.js chạy được trên nhiều nền tảng (Windows, Linux, Unix. Mac OS X, etc.)
  + Node.js sử dụng javascript để chạy trên môi trường server.
    - 1. **Cấu trúc Nodejs**
* Module giống như các thư viện JavaScript sử dụng trong ứng dụng Node.js application bao gồm một bộ các chức năng. Để đưa một module vào ứng dụng Node.js thì phải sử dụng hàm require() function with the với dấu ngoặc đơn chứa tên module.
* Node.js có rất nhiều module cung cấp các chức năng cơ bản cần thiết cho ứng dụng web. Ví dụ như các module trong bảng sau:



* + - 1. **Nodejs framework**

**Express**

Khi nói đến các framework Node.js phổ biến nhất, Express là lựa chọn đầu tiên của các nhà phát triển vì nó được 73% các nhà phát triển yêu thích và sử dụng.

Express là một trong những Node.Js Framework động cung cấp sự linh hoạt hoàn toàn và sự tự do cho các nhà phát triển trong việc tùy chỉnh giao diện của ứng dụng.

Vì nó nhẹ, do đó, nó cực kỳ nhanh và giúp định hướng các máy chủ và bộ định tuyến. Ngoài ra, khung này lý tưởng cho các ứng dụng web một trang, trang web hoặc các API HTTP công khai.

**Lợi ích của Express**

* + - Cung cấp hỗ trợ cho các plugin và tiện ích mở rộng khác nhau để tăng cường chức năng của nó
    - Tích hợp dữ liệu liền mạch
    - Một đường cong học tập dễ dàng cho các nhà phát triển đã quen thuộc với Node.Js
    - Cơ chế định tuyến dựa trên URL sử dụng HTTP
    - Mạnh mẽ, có thể mở rộng và hoàn toàn có thể tùy chỉnh
    - Đi kèm với MVC giúp tạo các ứng dụng tập trung

**SocketIO**

Socket.io là một full-stack framework có thể được sử dụng bởi nhiều ngôn ngữ lập trình. Nó cung cấp một API đơn giản và dễ hiểu, dễ dàng hoạt động hài hòa với hầu hết mọi dịch vụ. Với tốc độ phát triển vượt trội và độ tin cậy của nó, các nhà phát triển thực sự ấn tượng với framework này.

* Lợi ích của việc sử dụng Socket.io
* Cung cấp phân tích thời gian thực
* Phát trực tuyến nhị phân và quản lý ổ cắm dễ dàng
* Khả năng điều chỉnh định tuyến URL cho các ổ cắm web
* Tự động phát hiện và sửa lỗiSocket.io về cơ bản được sử dụng để phát triển các ứng dụng thời gian thực như API bảng điều khiển, Phiếu điểm, Chatbots, ứng dụng hội nghị truyền hình và hơn thế nữa, trong đó máy chủ được yêu cầu để đẩy dữ liệu mà không cần yêu cầu từ phía máy khách.
  + - 1. **Tại sao nên chọn nodejs**

**Khả năng mở rộng**

Các ứng dụng Node.js có thể dễ dàng mở rộng, theo chiều dọc và chiều ngang. Trong khi quy mô theo chiều dọc cho phép bạn thêm nhiều tài nguyên hơn vào các nút hiện tại, quy mô theo chiều ngang cho phép bạn thêm các nút mới nhanh hơn. Hơn nữa, các ứng dụng Node.js không yêu cầu một khối lớn trong toàn bộ quá trình phát triển, vì nó hoạt động cùng với một tập hợp các microservices và mô-đun. Nó đơn giản, dễ dàng và hoàn toàn phù hợp cho các công ty khởi nghiệp đang muốn phát triển.

**Hiệu suất cao**

Node.js chạy trên công cụ Google V8 JS tuân thủ mã Javascript trực tiếp vào mã máy. Điều này làm cho nó hiệu quả về tài nguyên cũng như tăng tốc độ của khung. Trên thực tế, bản thân Google đầu tư rất nhiều vào công cụ tìm kiếm của họ để cải thiện hiệu suất một cách nhất quán

**Hỗ trợ cộng đồng lớn**

Được thành lập vào năm 2015, Node.js Foundation là một nhóm các nhà sáng lập ấn tượng đến từ các doanh nghiệp hàng đầu như IBM, Microsoft, Fidelity và SAP. Một số lượng lớn các nhà phát triển phần mềm và nhiều người hâm mộ Node.js tích cực đang không ngừng đóng góp vào cộng đồng Node.js không ngừng phát triển và phát triển để hỗ trợ các nhà phát triển đồng nghiệp. Ngoài ra, Node.js được hỗ trợ trên Github, giúp việc phát triển và hỗ trợ framework trở nên cực kỳ dễ dàng và nhanh chóng.

**Tăng tốc độ phát triển**

Node.js cho phép bạn các nhà phát triển đẩy nhanh quá trình phát triển ứng dụng web bằng các công cụ nhẹ và đa dạng của nó. Sổ đăng ký npm cung cấp nhiều giải pháp, mô-đun và thư viện để tạo các ứng dụng web bằng cách thu được nhiều giá trị hơn và đầu tư với ít nỗ lực hơn. Node.js, cùng với các thư viện npm, giúp bạn giảm lỗi và giảm kích thước ứng dụng web của mình bằng các mẫu có thể sử dụng lại của nó. Sự tối ưu hóa này ảnh hưởng trực tiếp đến thời gian đưa ra thị trường.

**Tự do phát triển ứng dụng:**

Các nhà phát triển làm việc với Node.js có quyền tự do phát triển các ứng dụng đa nền tảng. Với các khuôn khổ như Electron và NW.js, bạn có thể tạo các ứng dụng web đa nền tảng với Node.js vì nó tương thích với máy tính để bàn, thiết bị di động và phát triển web cho Mac, Windows và Linux. Node.js hiện cũng được lựa chọn rộng rãi cho các giải pháp IoT. Vì vậy, nếu bạn đang tìm cách nâng cao hiệu suất và hiệu suất mã hóa của nhóm, thì Node.js là lựa chọn tốt nhất dành cho bạn.

* + 1. **Hệ quản trị cơ sở dữ liệu**
       1. **Giới thiệu về mysql**

##### **SQL là gì?**

SQL (Structured Query Languge hay ngôn ngữ truy vấn có cấu trúc) là một loại ngôn ngữ máy tính phổ biến để tạo, sửa, và lấy dữ liệu từ một hệ quản trị cơ sở dữ liệu quan hệ. Ngôn ngữ này phát triển vượt xa so với mục đích ban đầu là để phục vụ các hệ quản trị cơ sở dữ liệu đối tượng-quan hệ. Nó là một tiêu chuẩn ANSI/ISO.

Hệ quản trị cơ sở dữ liệu:

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

Ví dụ điển hình của hệ quản trị cơ sở dữ liệu bao gồm kế toán, nguồn nhân lực và hệ thống hỗ trợ khách hàng. Đầu tiên, hệ quản trị cơ sở dữ liệu chỉ có ở các công ty lớn với đầy đủ phần cứng cần thiết hỗ trợ cho một tập hợp dữ liệu lớn. Hệ quản trị cơ sở dữ liệu. Gần đây, nó đã trở thành một phần tiêu chuẩn của bất kỳ công ty nào.

Có rất nhiều hệ quản trị CSDL như (SQL Server của Microsoft, MySQL của Oracle, ... ), nhưng trong bài viết này, chúng ta cùng tìm hiểu MySQL

Giới thiệu về hệ quản trị dữ liệu MySQL:

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,..

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...

* + - 1. **Một số công dụng chính của MySQL**
* Kết nối dữ liệu trên sever dùng lệnh SQL để tương tác với cơ sở dữ liệu
* Kết hợp các trang Web với cơ sở dữ liệu bằng lệnh SQL. Tất cả các
* chức năng của SQL đều có thể được thực hiện bằng các công cụ khác của các phần mềm có sử dụng SQL.
* Thêm, cập nhật, xóa các bảng ghi trên toàn bảng theo những điều kiện khác nhau.
* Câu lệnh đơn giản, ngắn gọn, dễ nhớ, dễ sử dụng.
* Quản lý kho dữ liệu cho các giao dịch.
* Khả năng lưu trữ dữ liệu với các kiểu dữ liệu có miền giá trị rộng, bao gồm text, numeric, XML và các đối tượng lớn.
* Đáp trả các yêu cầu từ các ứng dụng client.
* Sử dụng Transact - SQL, XML hay các câu lệnh SQL Server khác để gởi các yêu cầu giữa ứng dụng client và SQL Server.
  + - 1. **Các câu lệnh cơ bản MYSQL**

• **SELECT**

Danh sách chọn trong câu lệnh SELECT được sử dụng để chỉ định các trường, các biểu thức cần hiển thị trong các cột của kết quả truy vấn. Các trường, các biểu thức được chỉ định ngay sau từ khoá SELECT và phân cách nhau bởi dấu phẩy. Sử dụng danh sách chọn trong câu lệnh SELECT bao gồm các trường hợp sau:

Khi cần hiển thị tất cả các trường trong các bảng, sử dụng ký tự \* trong danh sách chọn thay vì phải liệt kê danh sách tất cả các cột.

Trong trường hợp này, các cột được hiển thị trong kết quả truy vấn sẽ tuân theo thứ tự mà chúng đã được tạo ra khi bảng được định nghĩa.

Ví dụ: Câu lệnh

SELECT \* FROM LOP

Trong trường hợp cần chỉ định cụ thể các cột cần hiển thị trong kết quả truy vấn, ta chỉ định danh sách các tên cột trong danh sách chọn. Thứ tự của các cột trong kết quả truy vấn tuân theo thứ tự của các trường trong danh sách chọn.

Ví dụ: Câu lệnh

SELECT malop, tenlop, namnhaphoc FROM lop

Mệnh đề FROM chỉ ra tên 1 bảng hay những bảng có quan hệ cần truy vấn thông tin.

Mệnh đề WHERE để tạo nên điều kiện cần lọc mẩu tin theo tiêu chuẩn được định nghĩa. Thông thường WHERE dùng cột (trường) để so sánh với giá trị cột khác, hay biểu thức chứa cột (trường) bất kỳ có trong bảng (table)

Cấu trúc:

SELECT <danh sách các cột> FROM <danh sách các bảng> [WHERE <điều kiện ràng buộc>]

[GROUP BY <tên cột/biểu thức trong SELECT>] [HAVING <điều kiện bắt buộc của GROUP BY>] [ORDER BY <danh sách cột>]

• **UPDATE**

Phát biểu SQL dạng UPDATE dùng cập nhật lại dữ liệu đã tồn tại trong bảng. Khi UPDATE dùng cập nhật dữ liệu cho một mẩu tin chỉ định nào đó thường lệnh này sử dụng chung với mệnh đề WHERE.

Nếu cập nhật tất cả các mẩu tin trong bảng bạn có thể bỏ mệnh đề WHERE.

Cấu trúc như sau:

UPDATE <tên bảng SET <cột <giá trị , [<cột <giá trị [WHERE <condition>]

Nếu cập nhật giá trị là kết quả trả về từ phát biểu SELECT trên 1 hay nhiều bảng khác.Cấu trúc như sau:

UPDATE <tên bảng> SET <cột <select… FROM tên bảng WHERE… [WHERE <restrictive conditions>]

• **INSERT**

Khi cần thêm mẩu tin vào bảng (table) trong CSDL SQL, bạn có nhiều cách để thực hiện công việc này, nhưng để sử dụng các phát biểu SQL mang tính chuyên nghiệp bạn cần sử dụng phát biểu INSERT.

Khi thêm dữ liệu, cần chú ý kiểu dữ liệu giống hoặc tương ứng với kiểu dữ liệu đã khai báo của cột (column), nếu không phù hợp thì lỗi sẽ phát sinh.

Muốn INSERT vào CSDL thì ta có cấu trúc như sau: INSERT INTO <tên bảng> [<danh sách cột>]

• **DELETE**

Dùng để xóa mẩu tin trong bảng được chỉ ra bởi tên bảng và mệnh đề WHERE nhằm xác định mẩu tin cần xóa theo một điều kiện nào đó.

Cấu trúc như sau:

DELETE FROM <tên bảng> WHERE <conditions>

* 1. **Một số thư viện hỗ trợ**
     1. **Ant Design**



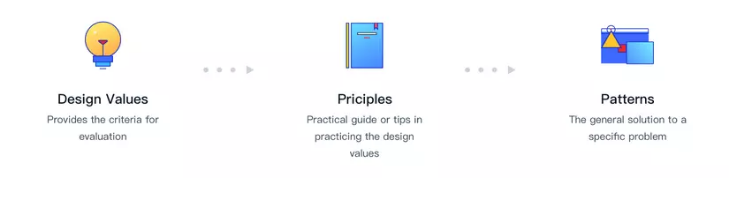
Ant.Design, thường được gọi là antd, là một khung thiết kế UI được xây dựng trên thư viện React.js để phát triển các ứng dụng web và React. Nó thuộc sở hữu của công ty Ant Design của Trung Quốc và được xây dựng theo nguyên tắc Thiết kế Vật liệu.

Antd là một khung hoàn chỉnh, giàu tính năng, không có nhiều phức tạp, cung cấp cho bạn tùy chọn sử dụng trực tiếp với dự án React của bạn. Do đó, nó cung cấp một tập hợp các thành phần React chất lượng cao mà có lẽ bạn chưa từng chú ý đến.

Ant.Design được viết bằng TypeScript và hoạt động tốt khi cộng tác với các công nghệ khác, vì nó cung cấp quy trình phát triển front-end npm, webpack và DVA.

**Giá trị trong thiết kế**

* Ant Design cung cấp một đánh giá thực tế về các thiết kế đẹp dành cho cả các nhà thiết kế của Ant Design và những người đang sử dụng nó. Đồng thời, nó xây dựng một nền tảng trên cùng các nguyên tắc và mẫu thiết kế để đưa ra hướng dẫn và giải pháp chung cho mục tiêu thiết kế được chỉ định.



* + - Sự tự nhiên trong thiết kế sẽ có ảnh hưởng rất lớn tới hành vi của người dùng, các nhà thiết kế nên lấy cảm hứng từ đó và áp dụng nó vào công việc thiết kế hàng ngày. Chúng tôi đã đang bắt đầu khám phá và sẽ theo đuổi sự tự nhiên trong thiết kế, đó là định hướng tương lai của chúng tôi.
    - Hệ thống thị giác đóng vai trò quan trọng nhất trong việc tiếp nhận và nhận thức của con người. Bằng cách tinh chỉnh các quy luật khách quan trong tự nhiên và áp dụng nó vào thiết kế giao diện, điều này sẽ tạo ra việc trải nghiệm sản phẩm tốt hơn cho người dùng. Ngoài ra, hệ thống thính giác hoặc hệ thống xúc giác có thể được thêm vào phần thiết kế trong tương lai để mang lại trải nghiệm sản phẩm thực tế hơn qua nhiều chiều. Đó có thể coi là ngôn ngữ lập trình trực quan.
    - Trong thiết kế sản phẩm, các phương pháp như phân tích hành vi, trí tuệ nhân tạo và cảm biến có thể được áp dụng để hỗ trợ người dùng đưa ra quyết định hiệu quả nhất, giảm thao tác người dùng không cần thiết, làm cho tương tác giữa người với máy tính trở nên tự nhiên hơn.

# CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Phân tích hệ thống

### Giới thiệu hệ thống

Hệ thống đặt vé xe online là một chương trình quản lý bán vé cho khách hàng , Website này sẽ cung cấp cho hành khách các thông tin như: Yêu cầu dịch vụ (du lịch, tham quan, dã ngoại,…) các địa điểm, các tuyến đường, số ghế cần đặt, loại xe, chi phí. Cung cấp cho nhà quản lý những chức năng để thay đổi thông tin, tiến hành giao dịch với khách hàng cũng như nắm bắt được nhu cầu của khách hàng và quá trình phát triển của công ty.

Người quản lý Website sẽ được kích hoạt quyền admin sau khi đăng ký các thông tin tài khoản. Admin sẽ quản lý toàn bộ từ nhân viên, các loại xe, vé xe, địa điểm, tuyến xe và các chi phí tới các công việc quản lý cũng như việc đặt vé quản lý hành khách. Các thành viên sẽ đăng ký làm quản trị viên của hệ thống dưới sự giám sát của người quản lý, để trở thành quản trị viên các thành viên cần phải cung cấp đầy đủ thông tin mà Website yêu cầu trong mục đăng ký thành viên như: Họ và tên, ngày tháng năm sinh, địa chỉ, số điện thoại, tên đăng nhập và mật khẩu.

Đối với khách hàng, hệ thống cho phép truy cập vào Website, khách hàng chỉ cần click vào mục đặt vé, sau đó tìm kiếm các thông tin về điểm bắt đầu, điểm kết thúc, dự kiến thời gian đi, lựa chọn chỗ ngồi,… Sau khi lựa chọn các thông tin chỉ cần ghi lại thông tin về họ tên, độ tuổi, tài khoản mail, số điện thoại liên lạcSau khi đặt được vé mà mình muốn, khách hàng lựa chọn hình thức giao dịch bằng việc chọn mục thanh toán, một form với các thông tin cần thiết cho việc thanh toán hiện ra. Website sẽ cung cấp nhiều lựa chọn cho khách hàng tiện lợi trong việc thanh toán, hành khách có thể thanh toán qua tài khoản ATM của các ngân hàng như:Momo , paypal…. Sau khi chọn hình thức thanh toán và điền đầy đủ thông tin trong hóa đơn đặt vé khác hàng chọn gửi hóa đơn. Khi hóa đơn đến hệ thống Website, người quản lý sẽ tiếp nhận hóa đơn để xác nhận thời gian, địa điểm để quản lý lượng hành khách còn lại.

### Xác định chức năng chính của hệ thống

Dựa vào phần giới thiệu về hệ thống ta chia chức năng của hệ thống thành các nhóm chức năng chính như sau:

* Nhóm chức năng thành viên đăng ký, đăng nhập.
* Nhóm chức năng xem thông tin cá nhân, thông tin về vé xe, tuyến đường, loại xe, thông tin hóa đơn.
* Nhóm chức năng quản lý, quản lý thông tin cá nhân, quản lý các thành viên của hệ thống, quản lý các loại vé xe, tuyến đường, loại xe, giá vé, hóa đơn.
* Nhóm chức năng đặt vé trực tuyến và xử lý hóa đơn.

### Tác nhân của hệ thống

Dựa vào mô tả bài toán trong phần giới thiệu về hệ thống ở trên, ta có thể xác định được các tác nhân chính của hệ thống như sau:

KHÁCH HÀNG: Người dùng chung, là nhóm người dùng có các quyền chung nhất như xem thông tin, tìm kiếm thông tin, đặt vé,…

QUẢN TRỊ VIÊN (do người quản lý tạo ra): Là nhóm có quyền thao tác trong phần quản trị theo một hoặc một nhóm chức năng do Admin cấp quyền, Gồm tất cả các quyền của các nhóm khác và có quyền tạo, hủy, cấp tài khoản truy cập hệ thống, cấp quyền quản lý cho các tài khoản quản lý hệ thống.

HỆ THỐNG TÍN DỤNG: Bao gồm hệ thống ngân hàng, hệ thống Website hay cổng thanh toán trực tuyến, tham gia vào quá trình thực hiện thanh toán các hóa đơn của khách hàng thành viên.

### Xác định các Use case, các gói Use case và vẽ biểu đồ Use case tổng quát

Xác định các Use case

**Tác nhân Khách hàng có Use case sau:**

**- Đăng Nhập , Đăng ký**

**- Xem thông tin**

**- Tìm kiếm thông tin**

**- Xem thông tin vé , cá nhân**

**- Hủy vé**

**- Sửa thông tin cá nhân**

**- Đặt vé xe**

**- Thanh toán**

**- Đăng xuất**

**Tác nhân Quản trị viên có các Use case sau:**

**- Quản lý các chức năng hệ thống do quản trị trị phân quyền**

**- Đăng xuất**

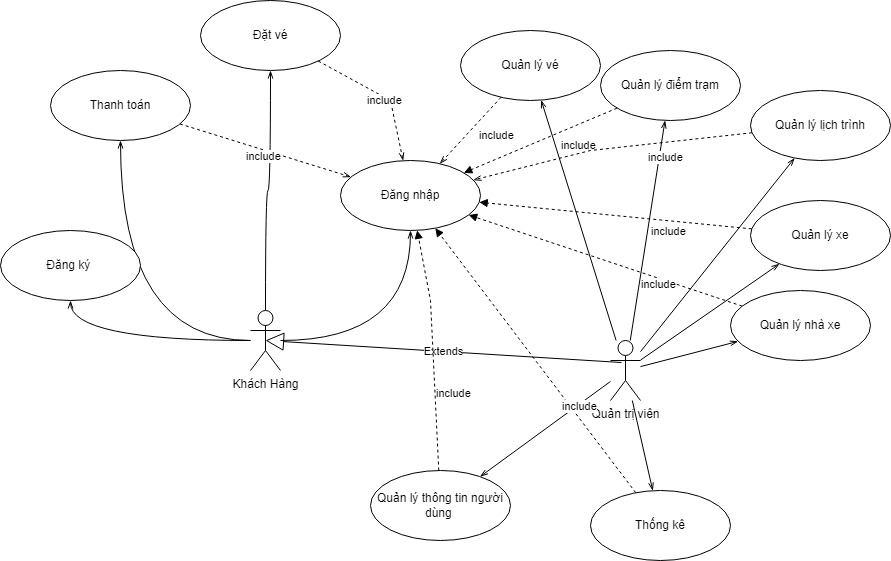
**- Đăng nhập**

**- Tiếp nhận và xử lý các phản hồi**

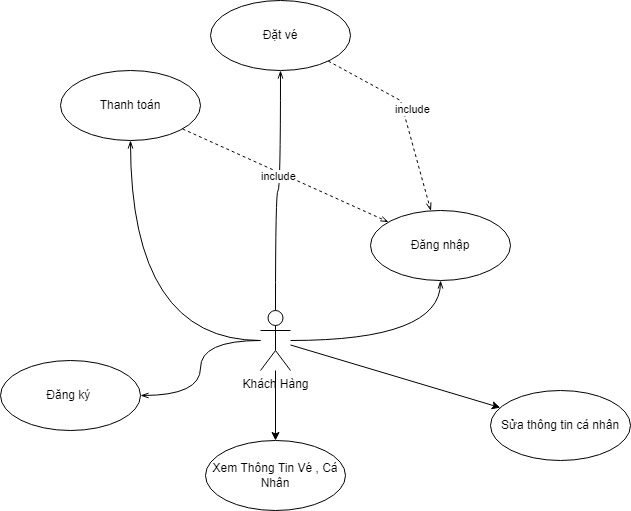
**- Đăng xuất**

* 1. **Biểu đồ usecase**

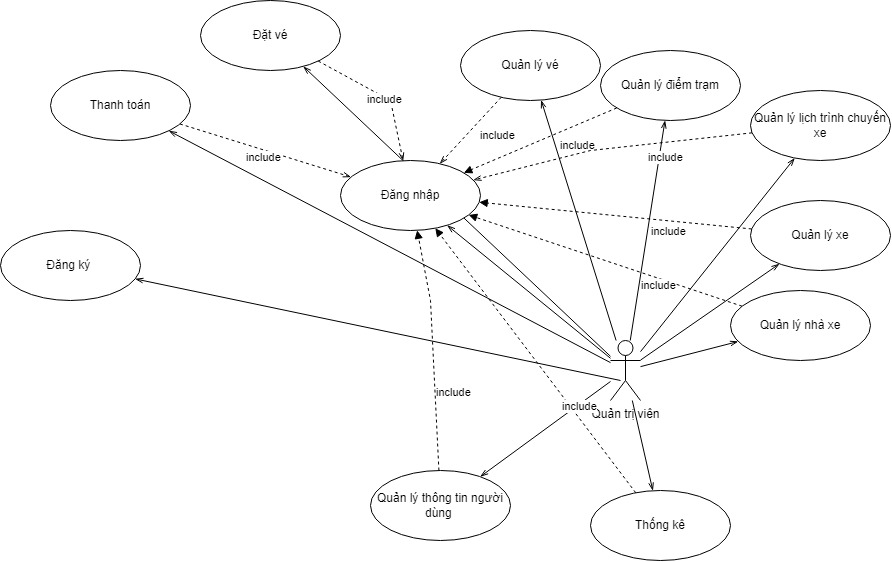
Từ việc phân tích các Use case của từng tác nhân, ta xây dựng thành các Use case như sau:

****

* + 1. **Use case người dùng – khách hàng**



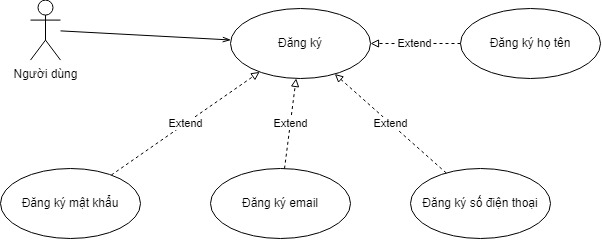
* + 1. **Use case người quản trị**



* + 1. **Use case đăng nhập**



* Mục đích: Người dùng . quản trị viên có thể đăng nhập vào hệ thống để sử dụng các chức năng của chương trình
* Tác nhân: Khách hàng,quản trị viên
* Điều kiện: Click vào nút đăng nhập để thực hiện.
  + 1. **Use case đăng ký**

****

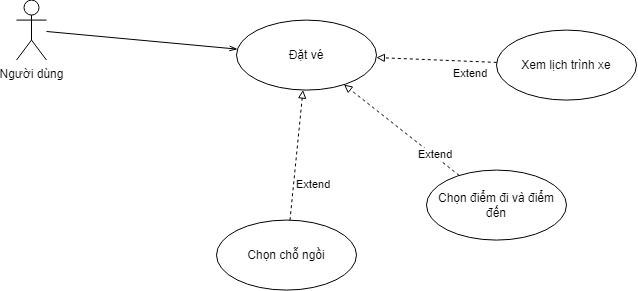
* Mục đích: Người dùng đăng kí buộc nhập phải đủ 5 form tên,số điện thoại,email,mật khẩu,xác nhận mật khẩu
* Tác nhân: Khách hàng , quản trị viên
* Điều kiện: Click vào nút đăng kí
  + 1. **Use case thanh toán**

****

* Tác nhân: Khách hàng
* Mô tả: Use case cho phép khách hàng thực hiện việc thanh toán.

Tiền điều kiện: Khách hàng đã đăng ký số ghế

* + 1. **Use case đặt vé**

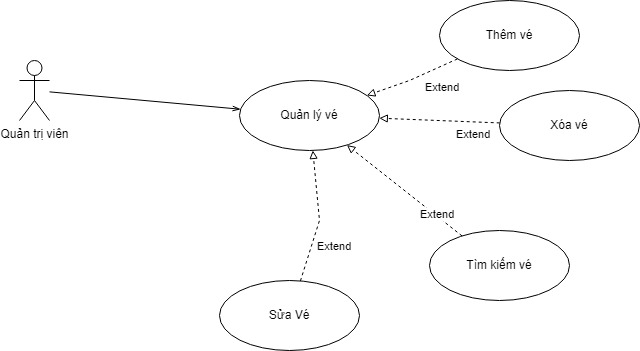


* + - Tác nhân: Khách hàng
    - Mô tả: Use case cho phép khách hàng chọn điểm đi về điểm đến
    - Tiền điều kiện: Địa điểm chọn phải hiển thị trên hệ thống. Luồng sự kiện chính:

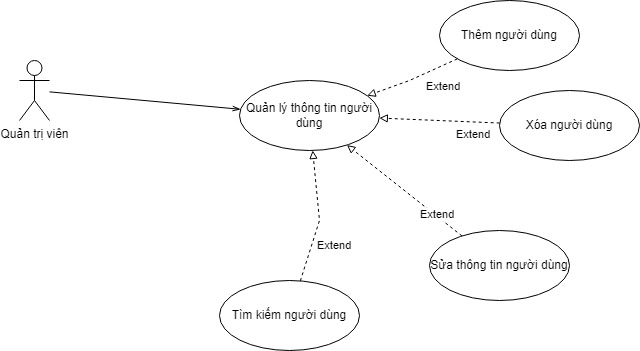
Khách hàng chọn vào mục thông tin địa điểm Hệ thống hiển thị các địa điểm đã có

Khách hàng xem thông tin về địa điểm, lựa chọn điểm đi và điểm đến.

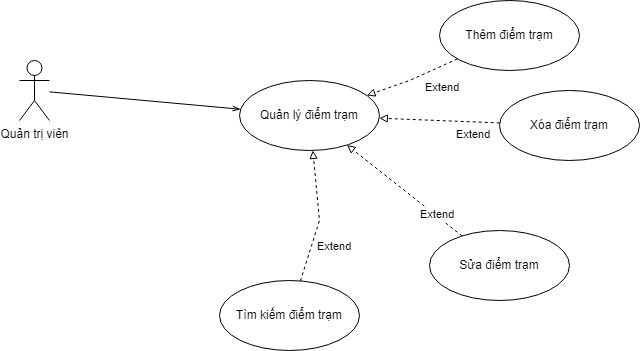
* + 1. **Use case quản lý vé**

****

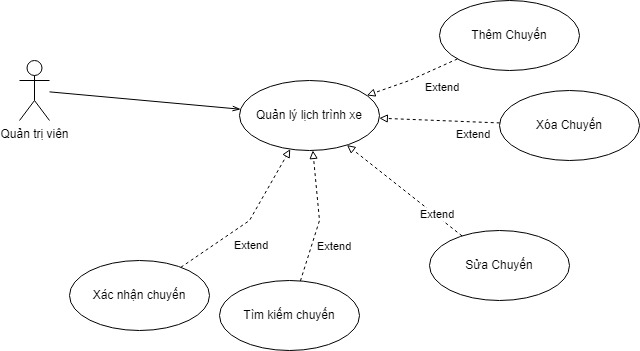
* Mục đích: Quản trị viên có thể thêm vé , sửa vé , xóa vé , tìm kiếm vé , xác nhận vé theo lịch trình của tất cả user
* Tác nhân: Quản trị viên
* Điều kiện: Click vào quản lý vé
  + 1. **Use case quản lý thông tin người dung**

****

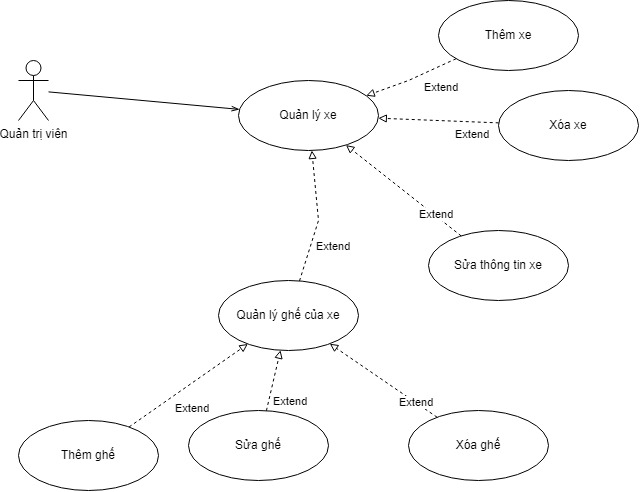
* Mục đích: Quản trị viên có thể thêm người dùng , sửa người dùng , xóa người dùng , tìm kiếm người dùng .
* Tác nhân: Quản trị viên
* Điều kiện: Click vào quản lý người dùng
  + 1. **Use case quản lý điểm trạm**

****

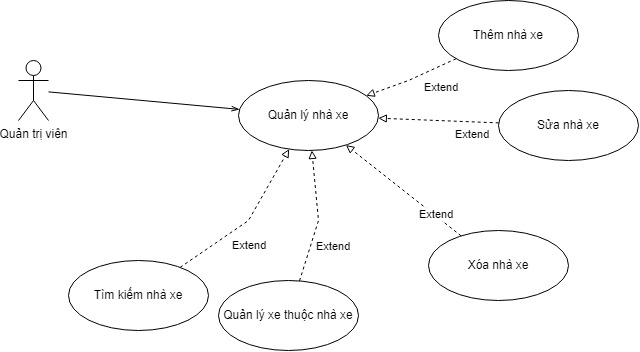
* Mục đích: Quản trị viên có thể thêm điểm trạm, sửa điểm trạm, xóa điểm trạm, tìm kiếm điểm trạm.
* Tác nhân: Quản trị viên
* Điều kiện: Click vào quản lý điểm trạm
  + 1. **Use case quản lý lịch trình chuyến xe**

****

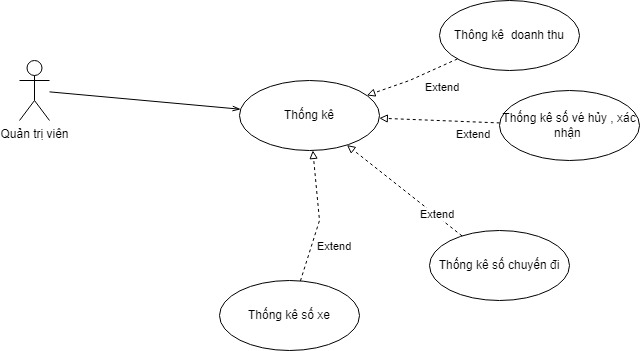
* Mục đích: Quản trị viên có thể thêm lịch trình, Sửa lịch trình, xóa lịch trình, tìm kiếm lịch trình , xác nhận lịch trình
* Tác nhân: Quản trị viên
* Điều kiện: Click vào quản lý lịch trình
  + 1. **Use case quản lý xe**

****

* Mục đích: Quản trị viên có thể thêm thông tin xe, Sửa thông tin xe, xóa thông tin xe, tìm kiếm xe , quản lý chỗ ngồi trên xe dễ dàng
* Tác nhân: Quản trị viên
* Điều kiện: Click vào quản lý xe
  + 1. **Use case quản lý nhà xe**



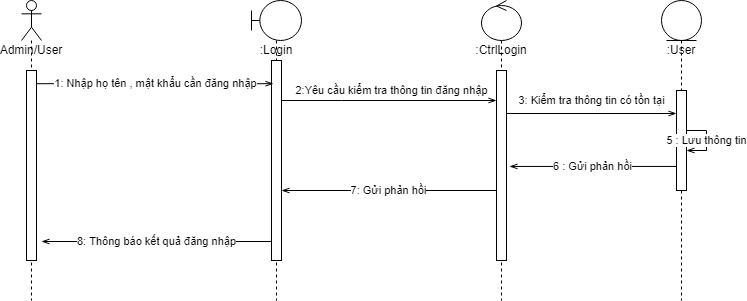
* Mục đích: Quản trị viên có thể thêm thông tin nhà xe, Sửa thông tin nhà xe, xóa thông tin nhà xe, tìm kiếm nhà xe , quản lý các xe thuộc nhà xe dễ dàng
* Tác nhân: Quản trị viên
* Điều kiện: Click vào quản lý nhà xe
  + 1. **Use case thống kê**

****

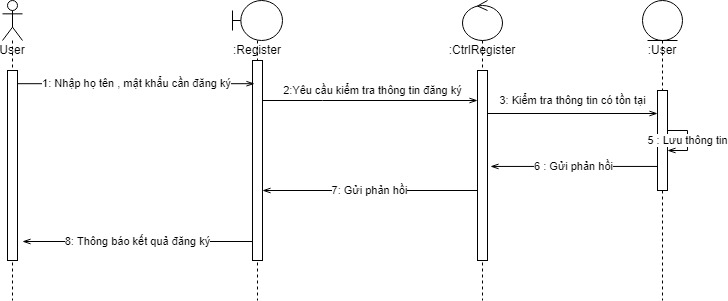
* Mục đích: Quản trị viên xem các thống kê doanh thu , vé , chuyến đi , xe
* Tác nhân: Quản trị viên
* Điều kiện: Click vào Thống kê
  1. **Biểu đồ tuần tự**

***Biểu đồ tuần tự của khách hàng***

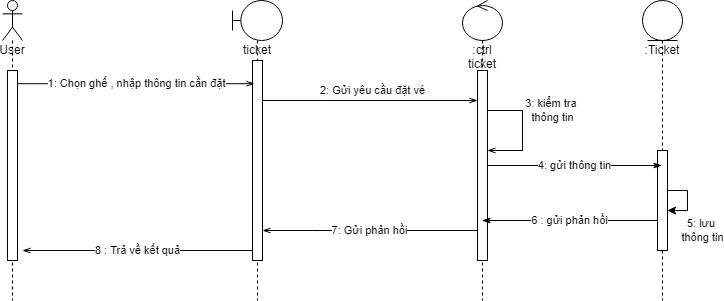
* + 1. **Biểu đồ tuần tự đăng nhập**



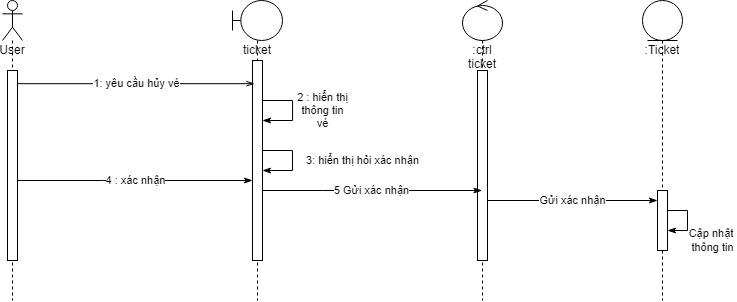
* + 1. **Biểu đồ tuần tự đăng ký**



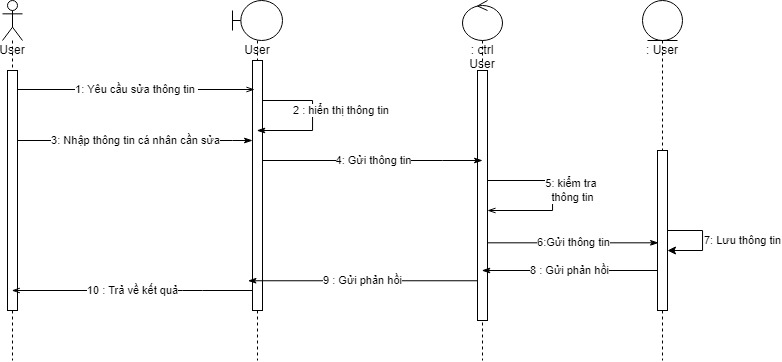
* + 1. **Biểu đồ tuần tự đặt vé**

****

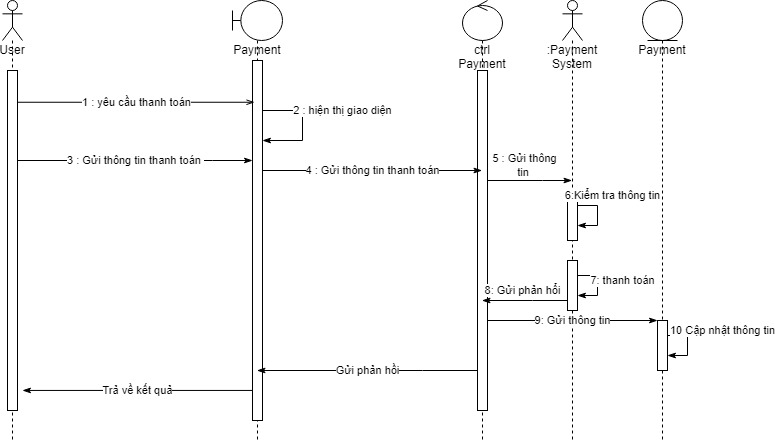
* + 1. **Biểu đồ tuần tự hủy vé**

****

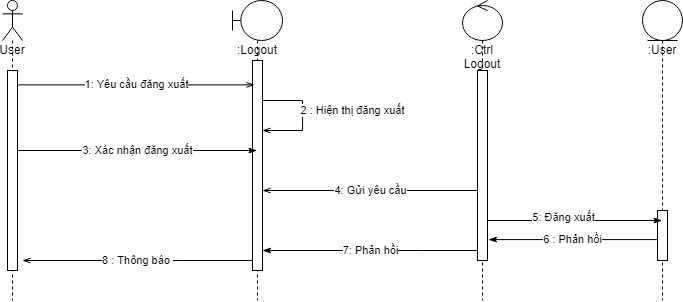
* + 1. **Biểu đồ tuần tự sửa thông tin cá nhân**

****

* + 1. **Biểu đồ tuần tự thanh toán**

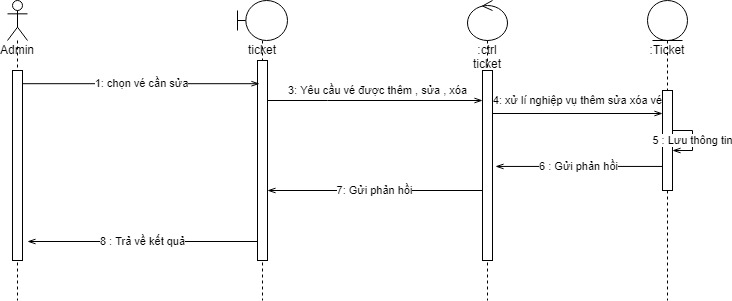
****

* + 1. **Biểu đồ tuần tự đăng xuất**

****

***Biểu đồ tuần tự cho quản trị viên***

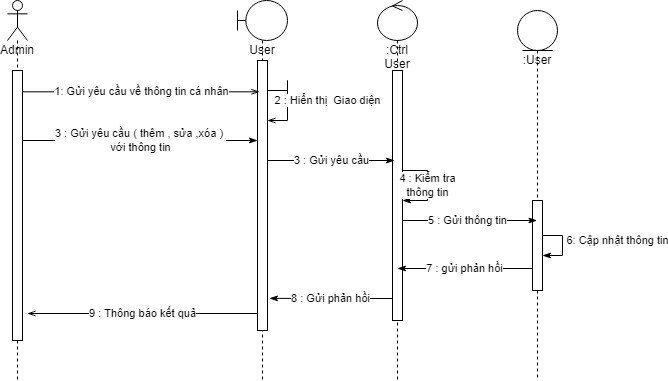
* + 1. **Biểu đồ tuần tự quản lý vé**

****

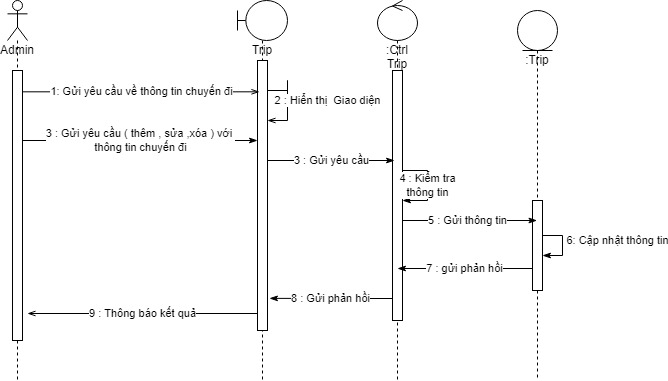
* + 1. **Biểu đồ tuần tự quản lý xe**

****

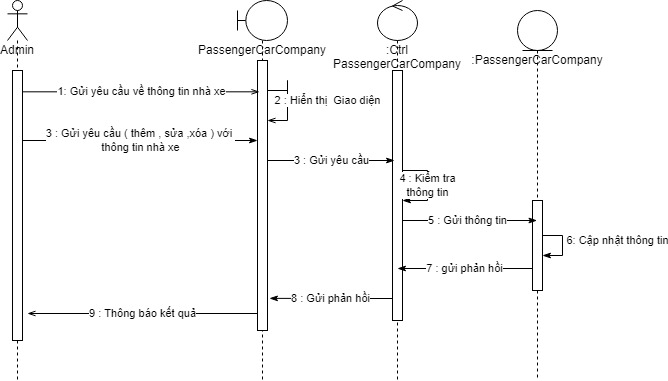
* + 1. **Biều đồ tuần tự quản lý thông tin cá nhân**

****

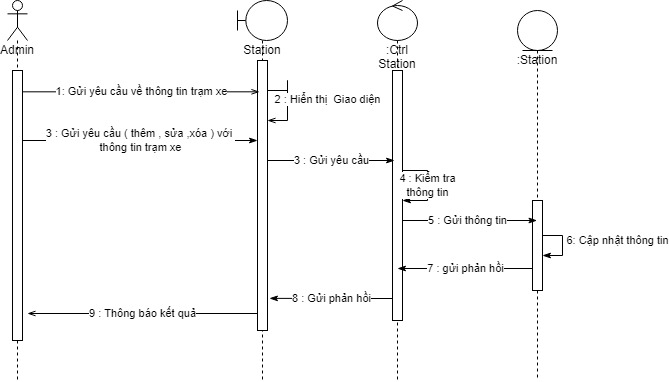
* + 1. **Biều đồ tuần tự quản lý lịch trình chuyến đi**

****

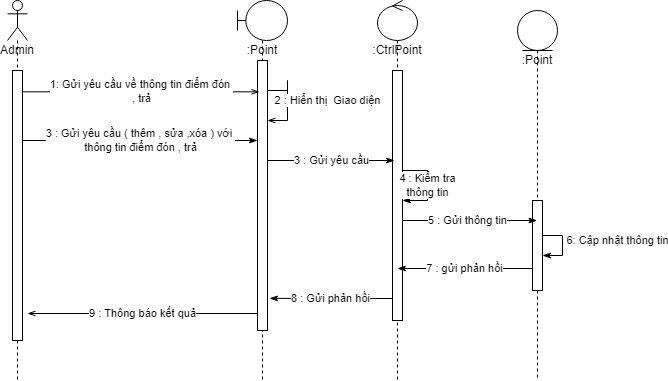
* + 1. **Biểu đồ tuần tự quản lý nhà xe**

****

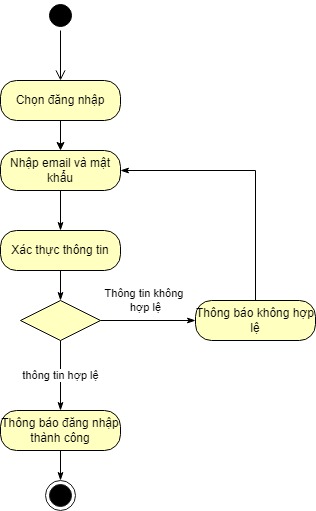
* + 1. **Biểu đồ tuần tự quản lý trạm**

****

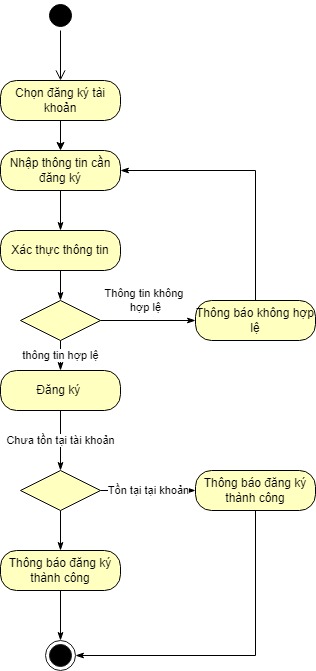
* + 1. **Biểu đồ tuần tự quản lý các điểm đón / trả**

****

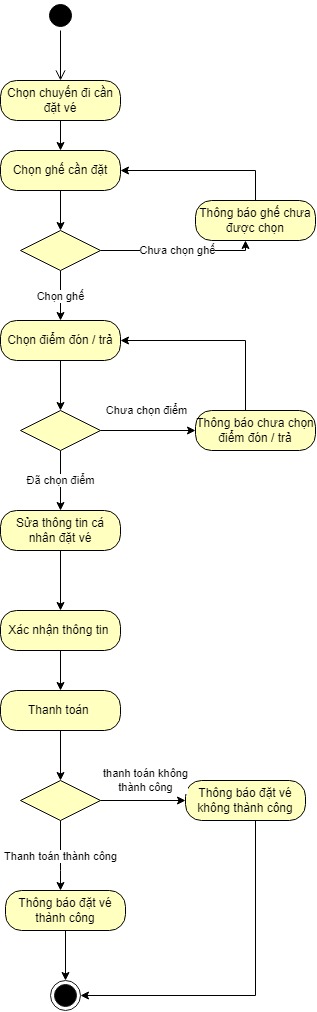
* 1. **Biểu đồ hoạt động**
     1. **Biểu đồ hoạt động đăng nhập**

****

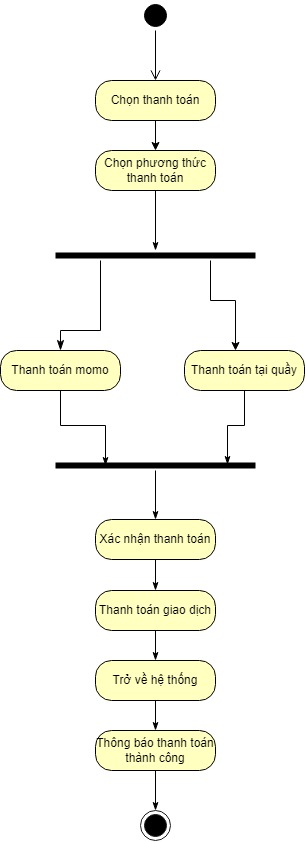
* + 1. **Biểu đồ hoạt động đăng ký**

****

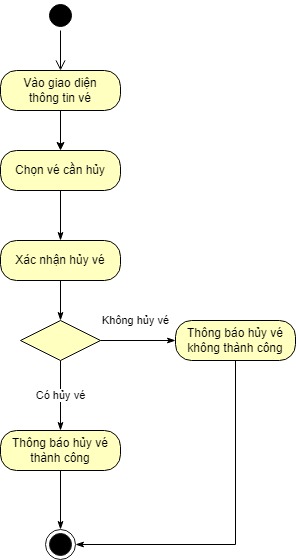
* + 1. **Biểu đồ hoạt động đặt vé**

****

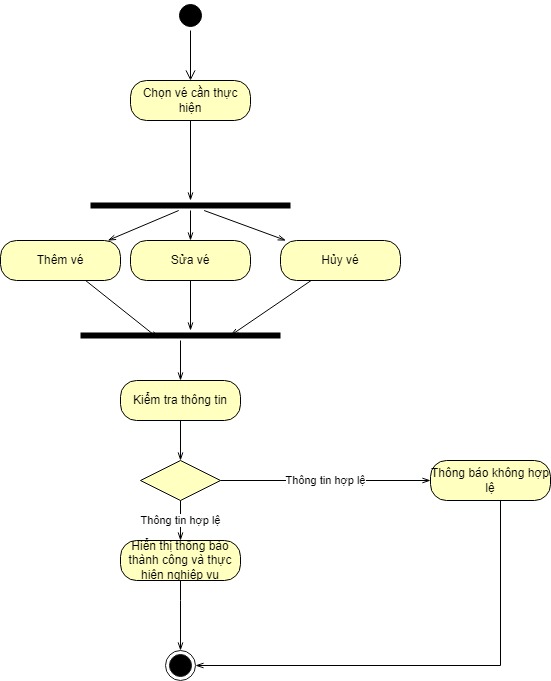
* + 1. **Biều đồ hoạt động thanh toán**

****

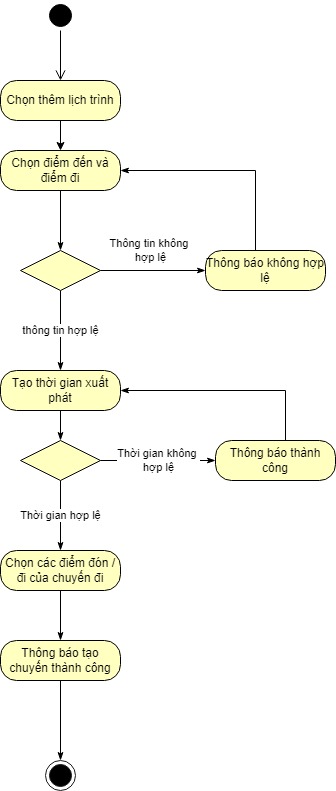
* + 1. **Biều đồ hoạt động hủy vé**

****

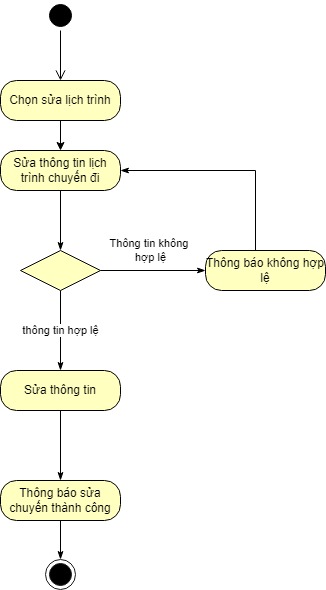
* + 1. **Biểu đồ hoạt động quản lý vé**

****

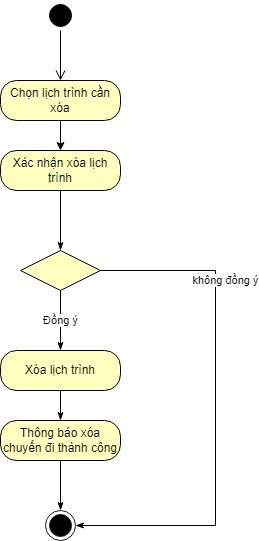
* + 1. **Biểu đồ hoạt động thêm chuyến đi**

****

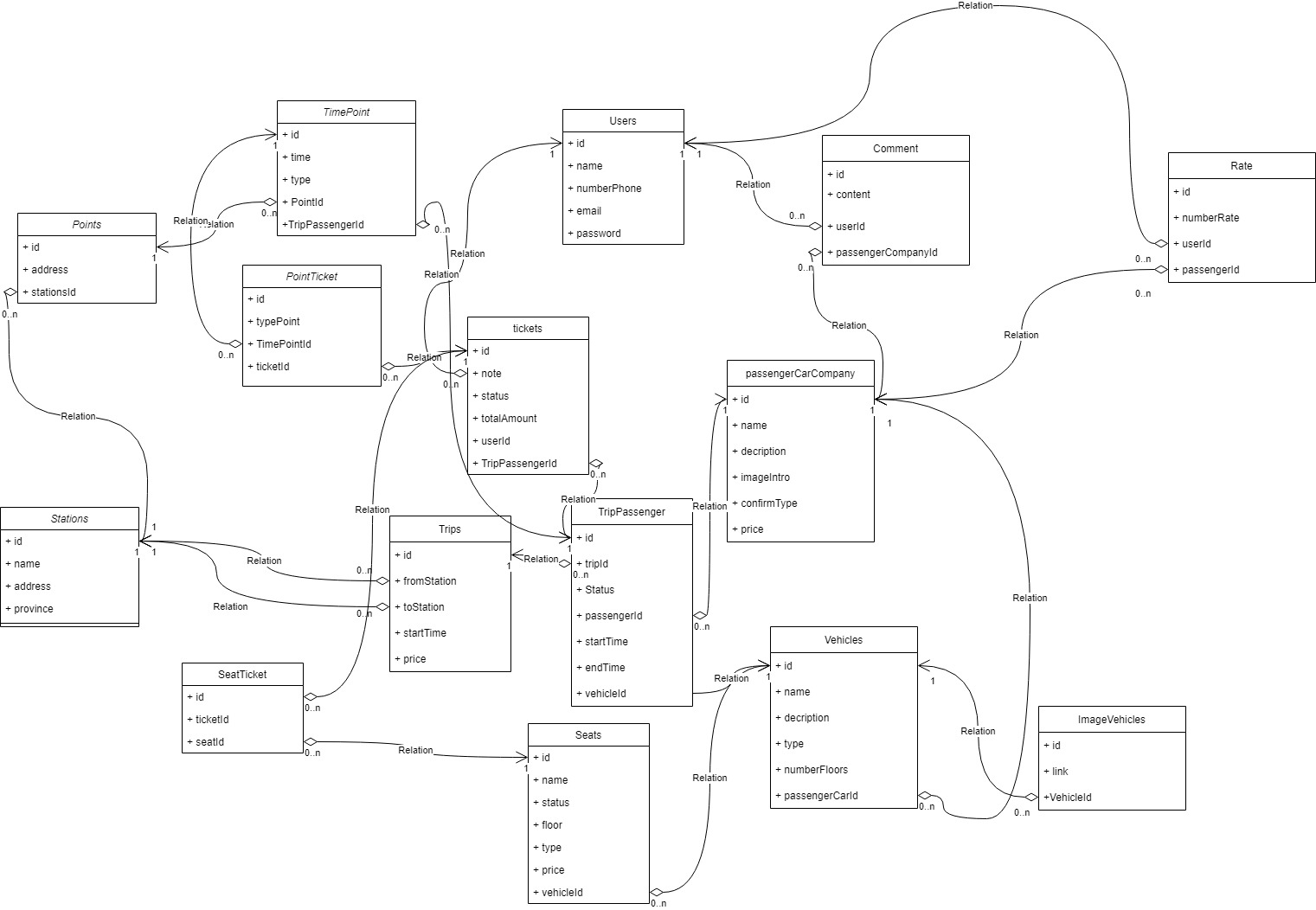
* + 1. **Biểu đồ hoạt động sửa chuyến đi**

****

* + 1. **Biểu đồ hoạt động xóa chuyến đi**

****

* 1. **Thiết kế hệ thống**
     1. **Thiết kế cơ sở dữ liệu**
        1. **Sơ đồ tổng quan**

****

* + - 1. **Bảng user ( người dùng )**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ghi chú** |
| Id | Mã khách hàng | Varchar(20) | Khóa chính |
| Name | Họ và tên khách hàng | Varchar(50) |  |
| NumberPhone | Điện thoại | Int |  |
| Email | Địa chỉ khách hàng | Varchar(50) |  |
| Password | Mật khẩu | Varhar(15) |  |

* + - 1. **Bảng ticket (vé)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ghi chú** |
| Id | Mã vé | Int | Khóa chính |
| note | Ngày tháng | Varchar(255) |  |
| status | Trạng thái | Varchar(255) |  |
| totalAmount | Tổng tiền | Float |  |
| userId | Mã khách hàng | Int | Khóa ngoài |
| TripPassengerId | Mã chuyến xe | Int | Khóa ngoài |

* + - 1. **Bảng vehicle ( xe)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ghi chú** |
| Id | Mã xe | Int | Khóa chính |
| name | Tên xe | Nvarchar(50) |  |
| type | Loại Xe | Nvarchar(50) |  |
| description | Mô tả | Varchar(255) |  |
| numberFloors | Số tầng xe | Int |  |
| PassengerCarId |  |  | Khóa ngoài |

* + - 1. **Bảng chuyến đi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ghi chú** |
| Id | Mã chuyến đi | Int | Khóa chính |
| name | Tên xe | Nvarchar(50) |  |
| status | Tình trạng | Nvarchar(50) |  |
| FromStation | Mã station đi | Int | Khóa ngoài |
| ToStation | Mã station đến | Int | Khóa ngoài |
| startTime | Thời gian khởi hành | DateTime |  |

* + - 1. **Bảng chuyến xe**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ghi chú** |
| Id | Mã chuyến xe | Int | Khóa chính |
| TripId | Mã Chuyến đi | Nvarchar(50) | Khóa ngoài |
| Status | Tình trạng | Nvarchar(50) |  |
| PassengerId | Mã Nhà Xe | Int | Khóa ngoài |
| + startTime | Thời gian xuất phát | DateTime |  |
| + endTime | Thời gian đến | DateTime |  |
| VehicleId | Mã Xe | int | Khóa ngoài |

* + - 1. **Bảng trạm ( station)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ghi chú** |
| Id | Mã station | Int | Khóa chính |
| name | Tên trạm | Nvarchar(50) |  |
| address | Địa chỉ | Nvarchar(50) |  |
| province | Tỉnh / thành phố | Nvarchar(50) |  |

* + - 1. **Bảng nhà xe**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ghi chú** |
| Id | Mã nhà xe | Int | Khóa chính |
| name | Tên trạm | Nvarchar(50) |  |
| description | Địa chỉ | Nvarchar(50) |  |
| imageIntro | Hình Ảnh đại diện | varchar(50) |  |
| ConfirmType | Loại xác nhận | varchar(50) |  |
| price | Giá trung bình | Float |  |

* + - 1. **Bảng ghế (seat)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ghi chú** |
| Id | Mã nhà xe | Int | Khóa chính |
| name | Tên trạm | Nvarchar(50) |  |
| description | Địa chỉ | Nvarchar(50) |  |
| imageIntro | Hình Ảnh đại diện | varchar(50) |  |
| ConfirmType | Loại xác nhận | varchar(50) |  |
| price | Giá trung bình | Float |  |

* + - 1. **Bảng điểm dừng , điểm trả (Point)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ghi chú** |
| Id | Mã nhà xe | Int | Khóa chính |
| address | Địa chỉ | Nvarchar(50) |  |
| stationId | Mã trạm | Int | Khóa ngoại |

* + 1. **Thiết kế module**

**Module quản lý nhà xe**: Module này thực hiện các nhiệm vụ quản lý hãng xe. Chức năng cảu module này gồm có:

Tạo mới hãng xe Chỉnh sửa hãng xe Xóa hãng xe

Module quản lý xe: Module này thực hiện nhiệm vụ quản lý xe. Chức năng của module này gồm có:

Tạo mới xe Chỉnh sửa xe Xóa xe

**Module quản lý loại ghế**: Module này thực hiện các nhiệm vụ quản lý loại ghế. Chức năng của module này gồm có:

Tạo mới loại ghế Chỉnh sửa loại ghế Xóa hãng loại ghế

**Module quản lý tuyến xe**: Module này thực hiện các nhiệm vụ quản lý tuyến xe. Chức năng của module này gồm có:

Tạo mới tuyến xe Chỉnh sửa tuyến xe Xóa tuyến xe

**Module quản lý điểm xuất phát và điểm đến**: Module này thực hiện các nhiệm vụ quản lý điểm xuất phát và điểm đến. Chức năng của module này gồm có:

Tạo mới điểm xuất phát hoặc điểm đến Chỉnh sửa điểm xuất phát hoặc điểm đến Xóa điểm xuất phát và điểm đến

**Module thống kê báo cáo**: Module này thực hiện các nhiệm vụ thống kê báo cáo đơn đặt vé qua mạng của hành khách theo nhiều tiêu chí khác nữa. Chức năng của module này gồm có:

Cho phép thống kê đơn đặt hàng trong một khoảng thời gian nhất định, hay theo trạng thái thanh toán của hợp đồng đặt vé.

Cho phép chọn trường hiển thị kết quả cần thống kê Đếm số hợp đồng thỏa mãn tiêu chí thống kê

**Module quản trị người dùng**: Module này thực hiện các nhiệm vụ quản trị người dùng. Chức năng của module này gồm có:

Tạo người dùng, xóa, và sửa thông tin người dùng Tao lập danh sách người dung sử dụng,Phân quyền các bộ phận chức năng và các nhân viên theo chức danh hợp lý.

**Module đặt vé**: Module này thực hiện các nhiệm vụ cho phép khách hàng tìm kiếm và thực hiện việc đặt vé. Chức năng của module này gồm có:

Tìm kiếm chuyến xe phù hợp ,Tìm kiếm thời gian phù hợp, Đặt vé

# CHƯƠNG 3 : CÀI ĐẶT VÀ XÂY DỰNG CHƯƠNG TRÌNH

## Ý tưởng giải quyết bài toán

Xây dựng một Website hỗ trợ khách hàng đặt vé xe trực tuyến và các tổ chức hoạt động về lĩnh vực vận tải (bán vé xe) có thể quản lý việc đặt vé dễ dàng thông qua Website này, thay cho quản lý việc quản lý thông thường. Hệ thống Website phải tạo cho khách hàng một cái nhìn tổng thể về “một chuyến đi” mà mình đã chọn. Trang website sẽ bao gồm các mục như: Trang chủ để khách hàng vào đặt vé, lựa chọn các thông tin về thời gian, địa điểm. Sau khi lựa chọn được các thông tin cần thiết khách hàng nhấn tìm kiếm vé để lựa chọn loại xe phù hợp và số ghế tương ứng. Cuối cùng khách hành sẽ điền các thông tin của mình để thực hiện thanh toán online qua cổng thông thanh toán điện tử momo để thực hiện việc thanh toán. Sau khi thanh toán thành công khách hàng có thể nhận được thông báo về số điện thoại hoặc qua địa chỉ email. Nếu hành khách muốn hủy vé hoặc thay đổi vé có thể gọi điện trực tiếp tới số hotline của hệ thống. Ngoài ra khách hàng còn có thể tìm hiểu một số thông tin mà hệ thống cung cấp như điểm du lịch, các thông tin xe, mọi thắc mắc của khách hàng sẽ được gửi qua mục ý kiến của khách hàng và

ý kiến đó sẽ được chuyển qua hòm thư điện tử của nhà quản lý.

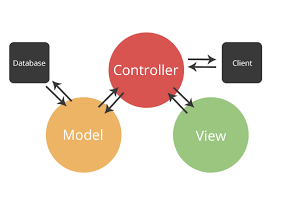
* 1. **Cách thức thực hiện**

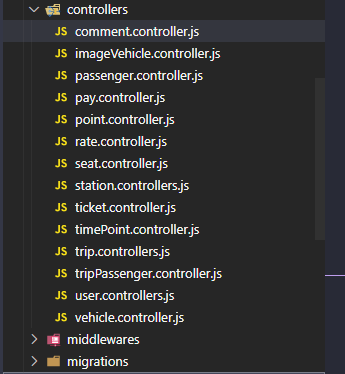
Tìm hiểu thực trạng hệ thống bán vé xe hiện nay để nắm bắt được cách quản lý vé và khách hàng của nhà xe.

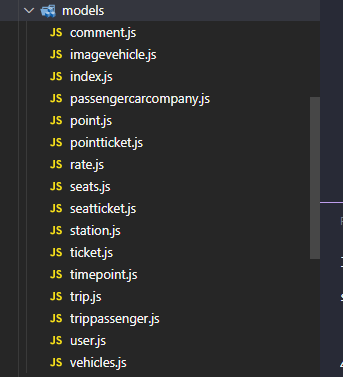
Tìm hiểu nhu cầu của hành khách cũng như những khó khăn mà họ gặp trong mỗi chuyến đi.

Tìm kiếm các tài liệu và công cụ để xây dựng một hệ thống bán vé trực tuyến khắc phục các nhược điểm của hệ thống bán vé thông thường.

Hệ thống sẽ được xây dựng dựa trên mô hình **MVC**:







**\* Cài đặt website**

Đặc điểm yêu cầu về kỹ thuật Khách hàng: Sử dụng Website Mục tiêu dự án: Kinh doanh

Để Website chạy được cần tạo một trang web sử dụng Reactjs build client và Server là Nodejs và MySQL Sever 2020. Người làm quản lý sẽ đăng nhập tên và mật khẩu để quản lý quản lý trang Web. Các trang web sẽ có phần đăng ký, người dùng nhập lần lượt các thông tin ở từng lĩnh vực nhất định. Các thông tin người dùng được lưu trữ trong cơ sở dữ liệu và cơ sở dữ liệu chứa thông tin của người dùng hiện tại được cập nhật.

Yêu cầu phần mềm:

- REACT Version 17.0.0

- Server build By Nodejs

- MySQL Server 2019

- Visual Studio Code

Một số thư viện hỗ trợ

"antd": "^4.18.7",

        "axios": "^0.26.0",

        "chart.js": "^3.7.1",

        "formik": "^2.2.9",

        "history": "^4.10.0",

        "lodash": "^4.17.21",

        "moment": "^2.29.1",

        "react": "^17.0.2",

        "react-chartkick": "^0.5.2",

        "react-dom": "^17.0.2",

        "react-redux": "^7.2.6",

        "react-router-dom": "^5.2.0",

        "react-scripts": "5.0.0",

        "react-slick": "^0.28.1",

        "redux": "^4.1.2",

        "redux-thunk": "^2.4.1",

        "sass": "^1.49.8",

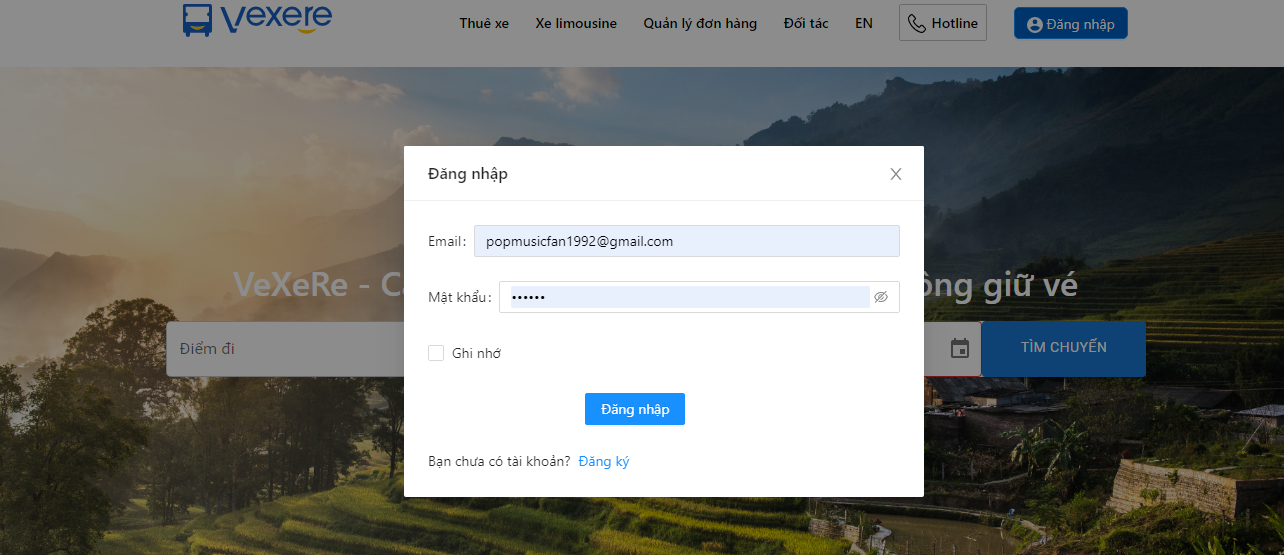
        "slick-carousel": "^1.8.1",

        "styled-components": "^5.3.3",

        "web-vitals": "^2.1.4",

        "yup": "^0.32.11"

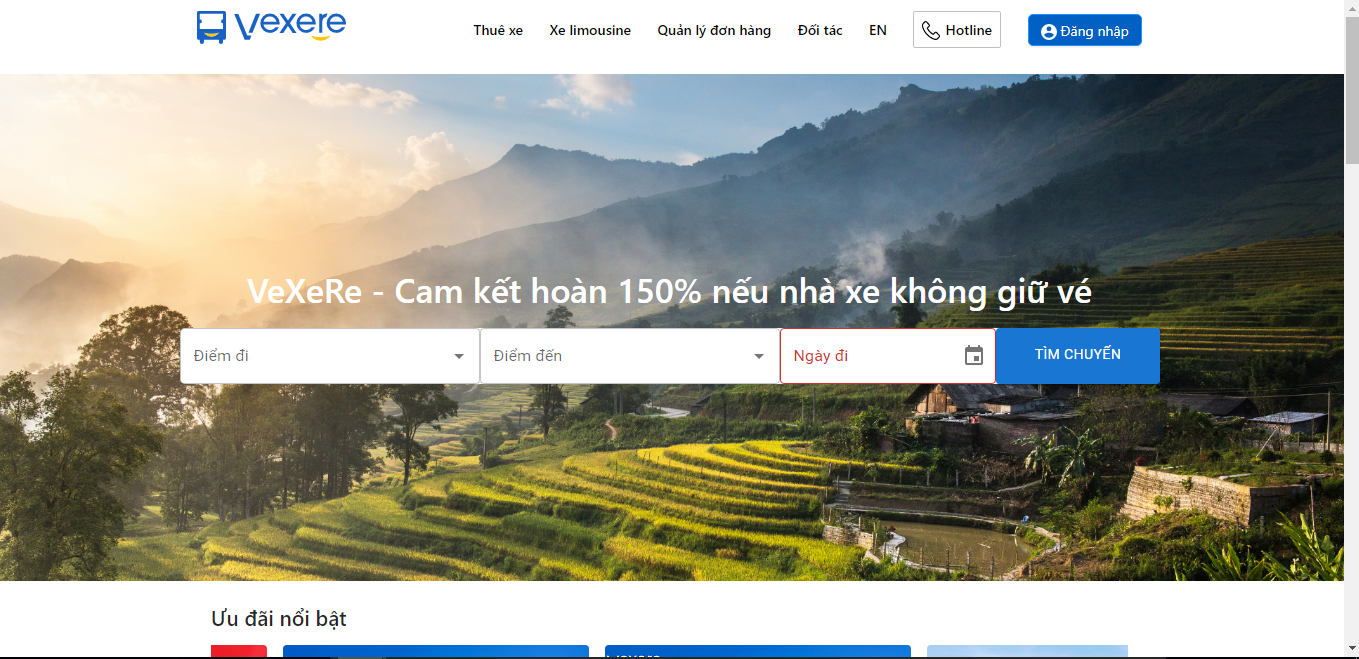
* 1. **Kết quả đat được**
     1. **Màn hình đăng nhập**



* + 1. **Màn hình đăng ký tài khoản**

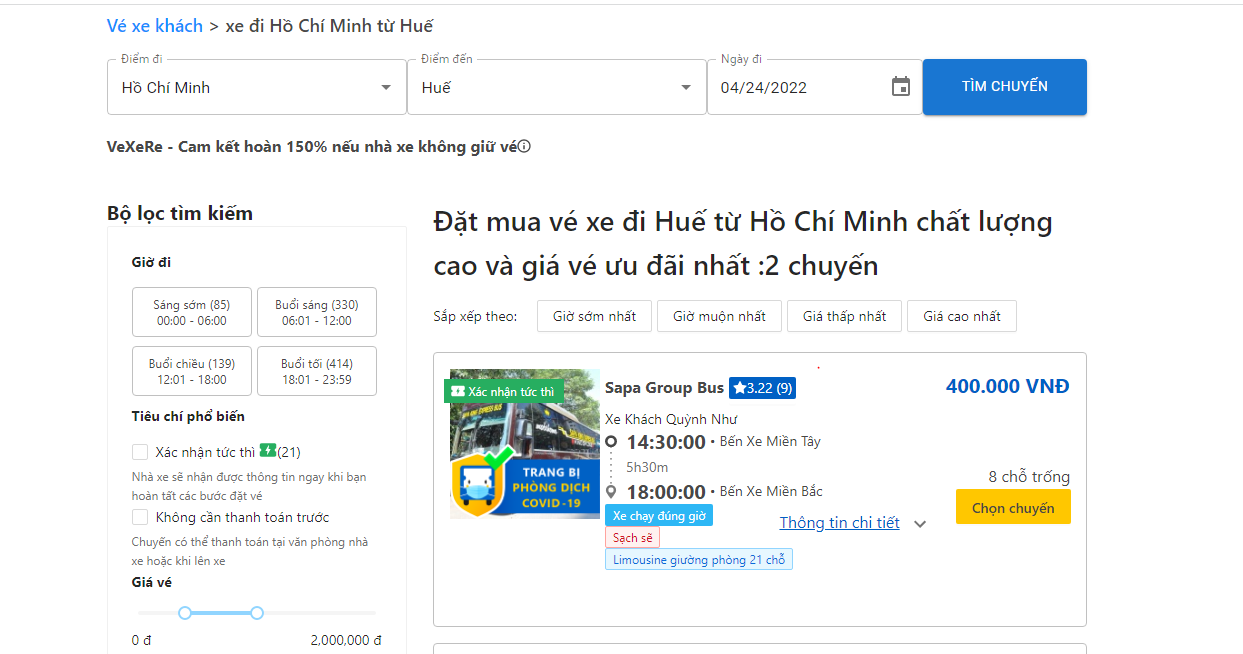


* + 1. **Màn hình trang chủ**



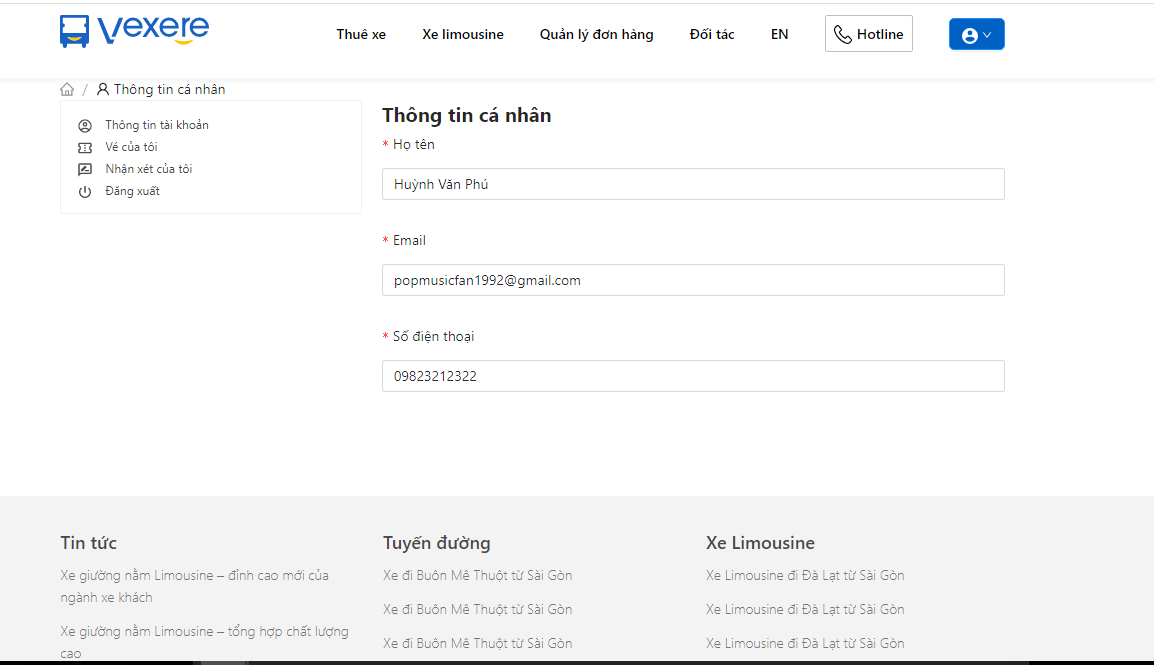
Hình ảnh giao diện trang chủ bao gồm thanh tìm kiếm chuyến đi điều khiển ở giữa carousel để khách hàng lựa chọn.

* + 1. **Màn hình danh sách chuyến đi**

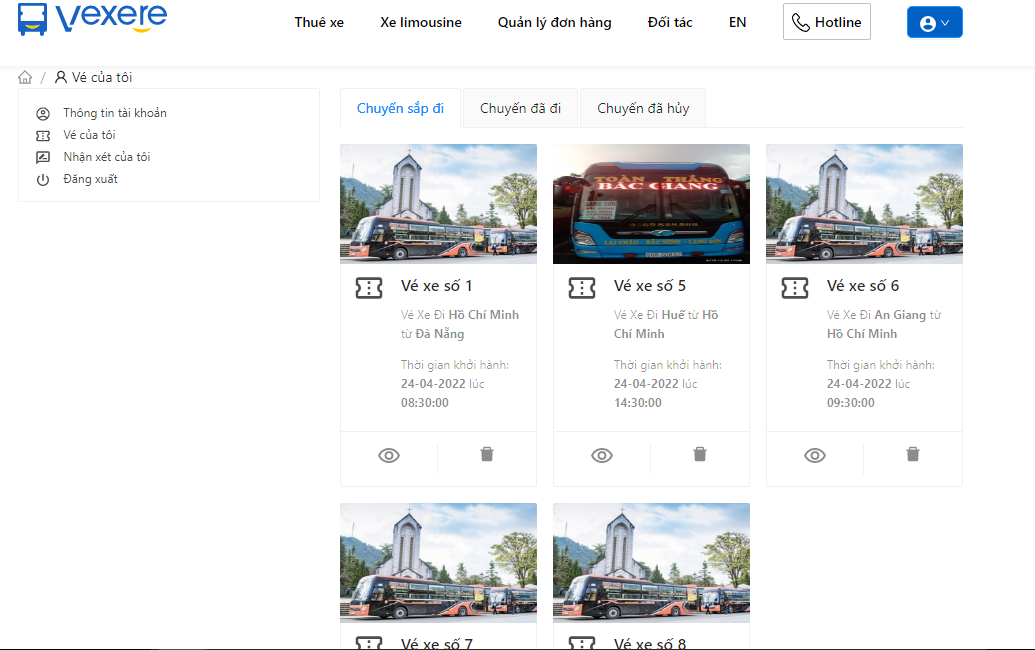


*Sau khi lựa chọn được tuyến đường khách hàng sẽ chọn các hãng xe mà mình muốn đi, tương ứng với các hãng xe khách nhau thì giá cả sẽ phù hợp với chất lượng của từng loại xe*

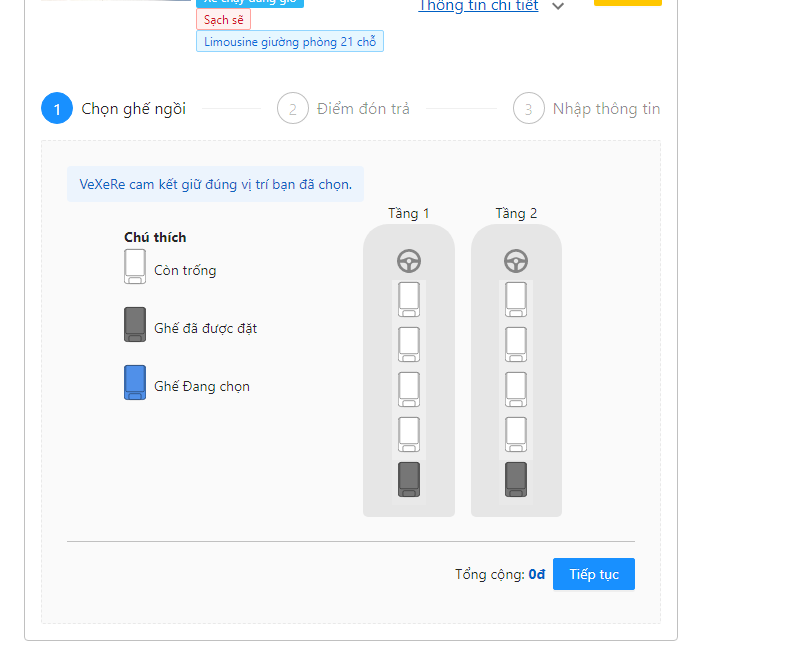
* + 1. **Màn hình thông tin cá nhân**



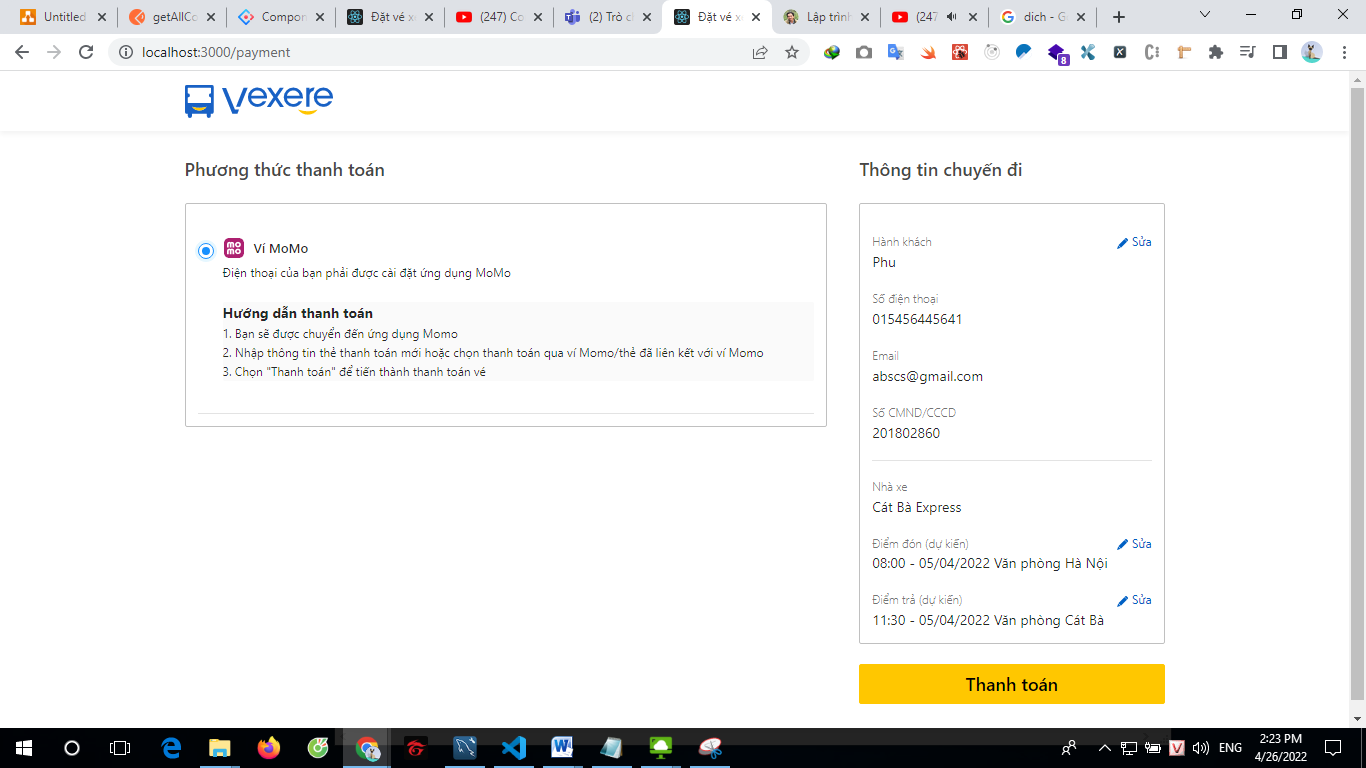
* + 1. **Màn hình thông tin vé người dùng**



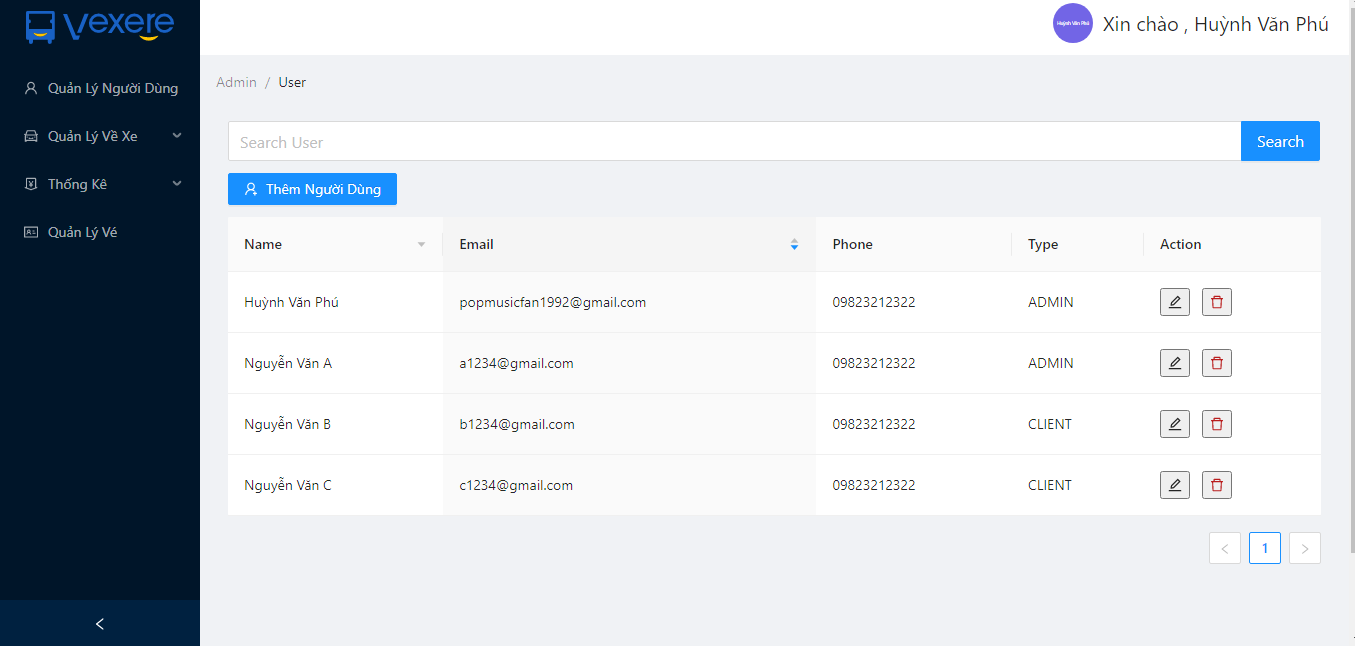
* + 1. **Màn hình chọn ghế ngồi**



* + 1. **Màn hình thanh toán**



* + 1. **Màn hình quản trị**



Giao diện trang quản trị hiển thị các chức năng của người quản lý Website. Danh mục các chức năng ở bên trên, khi chọn một đối tượng thì các điều khiển của chức năng đó sẽ hiển thị. Người dùng quản lý các thông tin về xe, tuyến xe, địa điểm…

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Với sự phát triển của xã hội hiện nay đặc biệt là với thời kì bùng nổ công nghệ thông tin, con người thường hướng tới những công việc nhanh chóng, tiết kiệm, hiệu quả và việc đi lại luôn là vấn đề được quan tâm nhất. Chính vì vậy mà khi mua vé bao giờ cũng xảy ra tình trạng chen lấn, xô đẩy tạo cơ hội cho những kẻ xấu. Nắm bắt các điểm này, em đã xây dựng một Website bán vé xe khách trực tuyến làm đề tài khóa luận của mình, hệ thống vừa giảm bớt công việc quản lý vé và hành khách của nhà quản lý lại vừa đem lại một chuyến đi vô cùng thoải mái đối với khách hàng. Tuy rằng Website chưa thật sự hoàn thiện xong nhưng cũng đã đem lại rất nhiều thuận lợi cho nhà quản lý và sự lựa chọn của khách hàng qua việc tìm kiếm vé và việc thanh toán tích hợp, giảm bớt rất nhiều công việc khi phải quản lý thủ công, mang lại một khoản doanh thu lớn cho công ty.

# Kết quả đạt đuợc:

+ Về công nghệ:

* Tìm hiểu và nắm bắt được các công cụ thiết kế Web.
* Biết được cách thiết kế Web động cũng như cách tổ chức cơ sở dữ liệu.
* Các dịch vụ trên Internet, đặc biệt là Web.

+ Về cài đặt chương trình:

* Giới thiệu các hãng xe đến với khách hàng.
* Cho phép tra cứu các thông tin về địa điểm, thời gian, số ghế của xe
* Cho phép khách hàng thực hiện việc đặt vé qua mạng.
* Tiếp nhận đơn đặt vé của khách hàng.
* Lập hóa đơn.
* Thống kê: Thống kê số vé đã được đăng ký

**+** Tính năng của chương trình:

* Thông tin về thời gian và địa điểm mang tính tham khảo nên có thể cập nhật chính xác
* Giao diện thân thiện với người dùng.
* Do điều kiện, nên chương trình chỉ mới chạy được trên nếu dữ liệu nhỏ.

# Hạn chế

Do kiến thức còn hạn hẹp và thời gian tìm hiểu thực tế có hạn nên hệ thống chưa hoàn thành đầy đủ tất cả các chức năng của Website, thiết kế còn chưa được đẹp mắt, chưa tạo được sự thu hút đặc biệt. Chưa tận dụng hết các điểm mạnh của ngôn ngữ được xây dựng để Website được chuyên nghiệp hơn **Hướng phát triển**

Website bán vé trực tuyến sau khi được hoàn thành chắc chắn sẽ là hướng bán vé chính của các công ty bán vé do các ưu điểm vượt trội của nó so với việc bán vé thông thường.

Phát triển tính năng đặt vé real-time

Phát triển thêm các chương trình giảm giá

Phát triển thêm các chương trình tin tức

Phát triển thành một hệ thống thương mại điện tử tích hợp thanh toán trực tuyến. Khi được phát triển trở thành một hệ thống thương mại điện tử thì Website không còn gói gọn cho một công ty bán vé trên một địa bàn mà nó sẽ mở rộng thêm các địa điểm và các tuyến lân cận.

# Tài liệu tham khảo

* Web đặt vé xe : <https://vexere.com/>
* <https://vi.reactjs.org/docs/getting-started.html> - Web reactjs
* https://sequelize.org/ - Tổ chức cơ sở dữ liệu