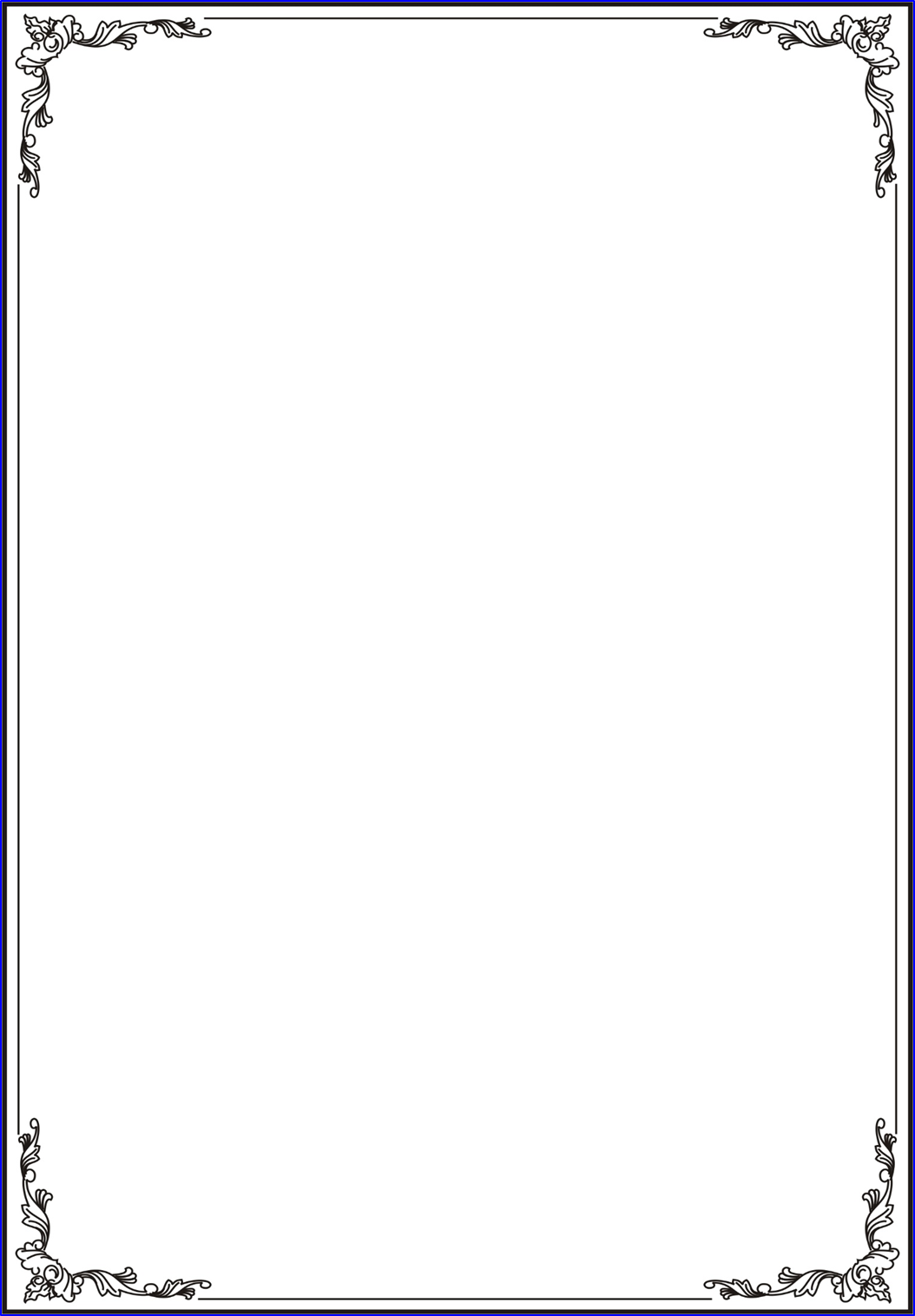
****

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**-- -- oOo-- --**



**LUẬN VĂN NGÀNH**

**HỆ THỐNG THÔNG TIN**

**ĐỀ TÀI**

**QUẢN LÝ HỢP TÁC XÃ CHO THUÊ XE Ô TÔ**

Giảng viên hướng dẫn: Sinh viên thực hiện:

TS. Trương Quốc Định Nguyễn Việt Thảo Nguyên

MSSV: B1400833

Khóa: 40

Tháng 5/2019

***Cần Thơ, 12/2018***

Cần Thơ, ngày 16/ 05/ 2018

# LỜI CẢM ƠN

Trước tiên em xin chân thành cảm ơn Thầy Trương Quốc Định – trưởng Bộ môn Hệ thống thông tin, Thầy đã hướng dẫn và giúp đỡ em tận tình trong suốt quá trình thực hiện đề tài luận văn tốt nghiệp của mình.

Em xin gởi lời cảm ơn đến toàn thể quý Thầy, Cô trong khoa CNTT&TT đã truyền đạt cho em những kiến thức quý báu cả chuyên môn lẫn kỹ năng, làm nền tảng để em thực hiện đề tài tốt này.

Ngoài ra, em cũng xin gởi lời cảm ơn đến các anh chị đồng nghiệp công ty TMA Solutions đã tạo điều kiện thuận lợi để em hoàn thành luận văn này.

Trong quá trình học tập cũng như làm luận văn, mặc dù đã cố gắng hết sức nhưng thời gian và khả năng của bản thân em có hạn nên vẫn khó tránh khỏi sai sót. Vì vậy rất mong nhận được những ý kiến đóng góp quý báu của Thầy và các bạn!

Cuối cùng, em xin chúc Thầy, Cô luôn dồi dào sức khỏe!

Xin chân thành cảm ơn!

Cần Thơ, ngày tháng … năm 2019

Sinh viên thực hiện

Nguyễn Việt Thảo Nguyên

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

Cần Thơ, ngày … tháng … năm 2019

Giảng viên hướng dẫn

# NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

………………………………………………………………………………………......

Cần Thơ, ngày … tháng … năm 2019

Giảng viên phản biện

MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc529093413)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 3](#_Toc529093414)

[NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN 4](#_Toc529093415)

[CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 7](#_Toc529093416)

[Tóm tắt 8](#_Toc529093417)

[Abstract 9](#_Toc529093418)

[PHẦN 1. TỔNG QUAN 10](#_Toc529093419)

[I. MÔ TẢ BÀI TOÁN. 10](#_Toc529093420)

[1. Bài toán. 10](#_Toc529093421)

[2. Vấn đề đặt ra. 11](#_Toc529093422)

[II. CHI TIẾT BÀI TOÁN. 12](#_Toc529093423)

[III. MỤC TIÊU CẦN ĐẠT ĐƯỢC. 13](#_Toc529093424)

[IV. HƯỚNG GIẢI QUYẾT. 13](#_Toc529093425)

[V. KẾ HOẠCH THỰC HIỆN. 14](#_Toc529093426)

[PHẦN 2. LÝ THUYẾT. 14](#_Toc529093427)

[I. CÁC KIẾN THỨC ĐƯỢC ÁP DỤNG TRONG ĐỀ TÀI. 14](#_Toc529093428)

[II. KẾT QUẢ VẬN DỤNG LÝ THUYẾT. 15](#_Toc529093429)

[1. Ruby on Rails (RoR) 15](#_Toc529093430)

[2. Mô hình MVC 16](#_Toc529093431)

[3. PostgreSQL 17](#_Toc529093432)

[4. 19](#_Toc529093433)

[PHẦN 3. KẾT QUẢ ỨNG DỤNG. 20](#_Toc529093434)

[I. PHÂN TÍCH. 20](#_Toc529093435)

[1. Mô hình CDM 20](#_Toc529093436)

[2. Mô hình PDM 20](#_Toc529093437)

[3. Mô hình Usecase 20](#_Toc529093438)

[II. DEMO 20](#_Toc529093439)

[PHẦN 4. KẾT QUẢ ỨNG DỤNG. 20](#_Toc529093440)

[I. KẾT QUẢ ĐẠT ĐƯỢC. 20](#_Toc529093441)

[1. Giải quyết được các vấn đề sau: 20](#_Toc529093442)

[2. Thu hoạch về kinh nghiệm chuyên môn: 20](#_Toc529093443)

[II. ƯU ĐIỂM, HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN. 20](#_Toc529093444)

# CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ/THUẬT NGỮ** | **Ý NGHĨA** |
| 1 | MVC | Model – View – Controller. Một mô hình phát triển ứng dụng |
| 2 | DB | Database – Cơ sở dữ liệu |
| 3 | CSDL | Cơ sở dữ liệu |
|  |  |  |
|  |  |  |
|  |  |  |
|  | HTX | Hợp tác xã |

# Tóm tắt

Khi mệt mỏi hay muốn tận hưởng cuộc sống, chúng ta thường tìm cho mình những vùng đất mới để đổi gió hoặc khám phá. Phương tiện vận chuyển cùng là một mối quan tâm để chuyến đi chủ động và thoải mái hơn. Và nếu như bạn có thể tìm cho mình một phương tiện di chuyển thuận lợi, nhanh chóng, thì bạn đã nắm trong tay chiếc chìa khóa dẫn đến những cung đường mới. Tuy nhiên, quy trình thuê xe tại các công ty cho thuê hiện nay rất mất thời gian do cần thực hiện nhiều giấy tờ, cùng với đó là bạn không thể tự tra cứu tại nhà hay thuê xe từ xa mà bạn cần đến liên hệ trực tiếp tại công ty. Do đó, việc phát triển một hệ thống có thể quản lý hợp tác xã cho thuê xe và cho phép khách thuê xe trực tiếp là rất cần thiết.

Hệ thống có 3 nhóm người dung chính: quản trị viên (Administrator), khách hàng thuê xe (Customer), thành viên hợp tác xã (Member). Hệ thống cho phép quản trị viên đăng nhập vào hệ thống, thực hiện các tạo mới, cập nhật trên hệ thống trong một khoảng thời gian xác định nào đó. Khách thuê xe có thể thuê xe trực tiếp ngay trên giao diện web của hệ thống theo đúng yêu cầu của mình. Đặc biệt thành viên hợp tác xã có thể đăng bài cho thuê xe và nhận lợi nhuận dựa trên số lần thuê xe của khách. Bên cạnh đó, hệ thống còn được tối ưu hóa khi sử dụng hai loại ngôn ngữ (Vietnamese - English), giúp thuận tiện và tạo sự thoải mái khi sử dụng.

# Abstract

When tired or want to enjoy life, we often find for ourselves the new land to change the air or just simply to explore. The transportation is also a concern to make the trip more active and more comfortable. And if you can find a convenient way to move quickly, then you have the key to the new roads. However, the process of renting a car at a rental company today is very time-consuming due to the need to do many papers, along with that you can not search the information by yourself at home or hire a remote vehicle that you need to contact directly at the company. Therefore, developing a system that can manage a car rental cooperative and allow direct car rental is essential.

The system has 3 primary user groups: Administrator, Customer, Member. The system administrator can login on the system, execute the new generate, update on system in a specified time interval that. The customer can rent a car directly on the UI of our system. Especially, the member has a register for the vehicle and got the pass of the client. Besides that, the system is optimized when using two language type (Vietnamese - English), help convenient and creating the comfortable when used.

**HỆ THỐNG QUẢN LÝ HỢP TÁC XÃ**

**CHO THUÊ XE Ô TÔ**

# PHẦN 1. TỔNG QUAN

## MÔ TẢ BÀI TOÁN.

### Bài toán.

Công ty TNHH Thương mại và dịch vụ Canva là mô hình hợp tác xã chuyên cho thuê các loại xe ô tô từ 4 đến 45 chỗ.

Khách đến công ty thuê xe sẽ liên hệ trực tiếp với bộ phận Điều hành. Bộ phận này sẽ tìm trong BẢNG BÁO GIÁ xem loại xe ô tô khách yêu cầu có không?

* Trường hợp không có loại xe như khách hàng yêu cầu thì thông báo cho khách hàng biết.
* Trường hợp có loại xe đúng như khách hàng yêu cầu. Bộ phận Điều hành sẽ tiến hành tra cứu trong SỔ XE, SỔ THUÊ XE theo thời gian khách hàng yêu cầu.
* Nếu không còn xe thì thông báo cho khách biết thời gian khác sớm nhất có thể có loại xe khách yêu cầu. Nếu khách hàng có nhu cầu đặt trước xe thì lưu thông tin lại vào SỔ THUÊ XE và công ty sẽ chủ động liên hệ khi có xe.
* Nếu công ty vẫn còn xe phù hợp nhu cầu của khách hàng thì tiến hành làm các thủ tục cần thiết, hợp đồng dựa trên yêu cầu của khách hàng. Bộ phận Điều hành sẽ thêm xe vào BẢNG ĐIỀU XE.

Thủ tục để thuê xe bao gồm: chứng minh nhân dân người ký hợp đồng và đặt trước 80% tiền thuê xe. 20% còn lại sẽ được khách hàng thanh toán sau khi hoàn tất việc trả xe. Bộ phận kế toán sẽ viết hóa đơn thanh toán 80% kèm hợp đồng.

Khách hàng sẽ nhận xe theo đúng thời gian và địa điểm đã thỏa thuận trên hợp đồng. Tài xế của công ty sẽ kiểm tra lại hợp đồng thuê xe cũng như hóa đơn thanh toán của khách hàng khi khách tiến hành nhận xe. Nếu không có gì sai sót, tài xế và khách hàng sẽ tiến hành thực hiện theo hợp đồng đã ký kết trước đó.

Nếu trong thời gian từ lúc ký hợp đồng đến khi nhận xe mà khách hàng muốn hủy hợp đồng thì sẽ phải chịu 10% tổng số tiền thuê như đã ghi trên hợp đồng thuê xe. Bộ phận kế toán sẽ tiến hành thanh toán khoản tiền còn lại và hủy hợp đồng.

Khi đến hạn hợp đồng và trả xe, khách hàng phải hoàn tất các thủ tục và chi phí cho phía công ty. Bộ phận kế toán sẽ thu 20% số tiền còn lại và lập hóa đơn gởi cho khách hàng và kết thúc hợp đồng.

Trong trường hợp khách hàng không tiến hành trả xe đúng thời gian ghi trên hợp đồng thì sẽ bị tính thêm tiền theo bảng giá mà công ty quy định riêng. Nếu việc quá thời gian do phía công ty gây nên (xe bị hư hỏng, tài xế của HTX gây tai nạn) thì phía khách hàng không phải chịu thêm bất kỳ khoản phí nào.

Những ô tô hỏng sẽ được chuyển đến bộ phận sửa chữa. Sau khi hoàn tất quá trình bảo dưỡng, xe sẽ được đưa về bãi và cho khách thuê. Trong quá trình đó, bộ phận sửa chữa cập nhật lại SỔ SỬA CHỮA.

Các bộ phận có liên quan trong công ty đều phải viết BÁO CÁO về tình trạng xe cho thuê, xe đã trả,.. thống kê xe nào được thuê nhiều nhất. Để báo cáo lại cho ban Giám đốc khi có yêu cầu.

### Vấn đề đặt ra.

Trên thực tế, nguồn nhân lực của công ty rất hạn chế. Việc quản lý cho thuê xe ô tô thì lại rất phức tạp, không chỉ đơn thuần là quản lý thông tin, số lượng xe mà còn quản lý các thông tin liên quan như: xe có thể cho thuê, xe đã cho thuê, xe đang trong quá trình bảo dưỡng, tài xế của hợp tác xã, … Ngoài ra còn các nghiệp vụ kế toán, thống kê, báo cáo theo ngày, tháng để dễ dàng cập nhật giá xe.

Theo thực tế cho thấy, việc quản lý 1 hợp tác xã cho thuê xe ô tô như trên mang nhiều bất cập:

* Lưu trữ bằng giấy dễ hư hỏng, mất mát và khó khôi phục.
* Làm thủ công tốn sức và tốn thời gian khá nhiều.
* Việc tìm kiếm các thông tin cần thiết trên giấy tờ sẽ gây mất thời gian và thiếu tính chính xác.
* Quá trình lập và xử lý các loại báo cáo chậm, dễ sai sót.

Giải pháp được đưa ra ở đây là xây dựng một hệ thống “Quản lý hợp tác xã cho thuê xe ô tô” dựa trên nền tảng công nghệ giúp cho việc quản lý, tìm kiếm, thống kê, báo cáo đáp ứng được yêu cầu nghiệp vụ, được xây dựng trên một giao diện thân thiện, dễ sử dụng và nhanh chóng. Ngoài ra còn hỗ trợ trang web đơn giản cho khách hàng tiến hành trực tiếp thuê xe, tìm hiểu các thông tin về hợp tác xã/ công ty và cho phép chủ xe đăng ký trở thành thành viên của hợp tác xã và cho thuê xe của họ.

## CHI TIẾT BÀI TOÁN.

Dịch vụ cho thuê xe ô tô là một dịch vụ liên quan đến nhu cầu về phương tiện di chuyển, đi lại rất cần thiết hiện nay. Công ty sẽ hợp tác cùng các thành viên nhằm thành lập một hợp tác xã để có nguồn xe cho thuê nhiều hơn nhưng vẫn đảm bảo chất lượng cùng giá cả phải chăng, mang lợi lợi ích cho đôi bên. Các xe cho thuê từ các thành viên khác trong công ty đều được lập hợp đồng phân chia lợi nhuận.

Mỗi xe cho thuê đều có thông tin bao gồm mã ID, model, năm sản xuất, biển số xe, chủ sở hữu xe

Khi thành viên trong hợp tác xã có nhu cầu hợp tác cung cấp xe cho thuê thì công ty sẽ tiến hành lập hợp đồng hợp tác góp xe, nhằm tính chiết khấu phần trăm cho thành viên. Mỗi hợp đồng đều mang một số riêng, sẽ bao gồm ngày lập hợp đồng, thông tin của công ty và thông tin của người góp xe cho thuê, thông tin của xe cho thuê và phần trăm lợi nhuận.

Khi khách hàng có nhu cầu thuê xe, phía Công ty sẽ tiến hành lập hợp đồng cho thuê xe. Hợp đồng này bao gồm số, thông tin công ty, thông tin khách hàng, thông tin của xe được thuê, hình thức thuê, chi phí.

Có 3 hình thức thuê xe: dài hạn, ngắn hạn và tự lái. Với hai hình thức dài và ngắn hạn thì có thể kèm theo tài xế hoặc không, tùy theo nhu cầu phía khách hàng. Do đó phía công ty phải quản lý phần thông tin của tài xế bao gồm ID tài xế, họ tên tài xế, số CMND, số điện thoại, địa chỉ liên lạc và đặc biệt là số bằng lái xe cũng như hạng bằng nhằm đảm bảo tính an toàn và chuyên nghiệp.

## MỤC TIÊU CẦN ĐẠT ĐƯỢC.

Mục tiêu đề tài là nghiên cứu và xây dựng hệ thống “Quản lý dịch vụ cho thuê xe” cần đạt các tiêu chí sau:

* Xây dựng website cho phép khách hàng trực tiếp thuê xe, cho phép chủ xe đăng ký thành viên và đăng xe cho thuê.
* Xây dựng giao diện cho Admin thực hiện các chức năng cơ bản như lập hợp đồng thuê xe, hợp đồng góp xe, quản lý xe, quản lý tài xế, quản lý bảo dưỡng xe cùng các công tác thống kê, báo cáo.

## HƯỚNG GIẢI QUYẾT.

Về phạm vi và đối tượng nghiên cứu:

* Liên hệ Giáo viên hướng dẫn để tham khảo ý kiến.
* Liên hệ trực tiếp nhân viên của Công ty Liên doanh Vận chuyển Quốc Tế Hải Vân tại TP.HCM để nắm rõ quy trình nghiệp vụ.
* Nghiên cứu trên Internet các trang web hỗ trợ cho thuê xe.

Về lý thuyết:

* Nắm vững kiến thức về phân tích và thiết kế CSDL.
* Quy trình phát triển phần mềm.
* Kiến thức về Ruby on Rails.
* Tìm hiểu về framework hỗ trợ thiết kế giao diện …

Về kỹ thuật:

* Dùng Power Designer 16.5 để thiết kế CDM, PDM.
* Dùng StarUML để thiết kế sơ đồ Usecase.
* Hệ quản trị cơ sở dữ liệu ?
* Ngôn ngữ xây dựng hệ thống:
* Giao diện cho khách hàng:
* Giao diện cho Admin:
* Dữ liệu:

## KẾ HOẠCH THỰC HIỆN.

|  |  |
| --- | --- |
| **Thời gian** | **Công việc** |
| Tuần 1-4 | Chọn đề tài |
| Tuần 5 | Tìm hiểu, nghiên cứu đề tài |
| Tuần 6-7 | Viết đặc tả và thiết kế sơ đồ usecase và mô hình CDM |
| Tuần 8-10 | Hiệu chỉnh đặc tả và sơ đồ, mô hình |
| Tuần 11-14 | Chuyển CDM qua PDM Viết báo cáo |
| Tuần 15-18 | Nghiên cứu và Lập trình |
| Tuần 19 | Lập trình và Viết báo cáo |
| Tuần 20 | Hiệu chỉnh và hoàn thành báo cáo |

# PHẦN 2. LÝ THUYẾT.

## CÁC KIẾN THỨC ĐƯỢC ÁP DỤNG TRONG ĐỀ TÀI.

Để giải quyết những vấn đề mà bài toán đặt ra, có những bước tiếp cận kiến thức như sau:

* Sử dụng framework Ruby on Rails.
* Dùng hệ quản trị cơ sở dữ liệu PostgreSQL 10.? để tạo và quản lý cơ sở dữ liệu.
* Sử dụng Heroku host để deploy.
* Sử dụng phần mềm Power Designer 16.5 làm công cụ để phân tích và thiết kế các mô hình dữ liệu CDM, PDM.
* Sử dụng phần mềm StarUML để thiết kế sơ đồ UseCase.

## KẾT QUẢ VẬN DỤNG LÝ THUYẾT.

Sơ lược về các công nghệ đã được sử dụng trong hệ thống:

### Ruby on Rails (RoR)

* Ruby là gì?

Ruby là một ngôn ngữ lập trình hướng đối tượng do một giáo sư người Nhật có tên là Yukihiro Matsumoto, tên thường gọi là "Matz" sáng lập ra. Ruby được giới thiệu lần đầu vào năm 1995.

Ngôn ngữ Ruby được viết ra với mong muốn tạo một ngôn ngữ lập trình thân thiện với lập trình viên. Ruby được lấy cảm hứng từ các ngôn ngữ lập trình khác như Perl, Smalltalk, Effiel và Lisp.

* Ruby on Rails là gì?

Ruby on Rails hay gọi tắt là Rails là một Web Framework của ngôn ngữ Ruby được ra đời bởi David Heinemeier Hansson, Rails được xây dựng nên để hỗ trợ các lập trình viên việc phát triển các phần mềm web một cách nhanh nhất có thể.

Rails framework được tạo ra như là một câu trả lời đối với các web frameworks khác như J2EE, .NET. Để quá trình phát triển phần mềm diễn ra nhanh hơn, Rails sử dụng các qui ước triệt để và đảm nhận xử lý rất nhiều task khiến người lập trình viên không phải bận tâm về nó nữa như : mail management, object-database mappers, file structures, code generation..., đây chính là hai đặc điểm nổi bật nhất của Rails, điều này không chỉ giúp các lập trình viên viết code ít hơn, phát triển ứng dụng nhanh hơn mà còn làm ứng dụng dễ hiểu và dễ bảo trì hơn.

Giống như đa số các Web framework khác Rails sử dụng mẫu kiến trúc Model - View - Controller (MVC) để tăng cường khả năng bảo trì và phát triển của úng dụng.

Ruby on Rails cho phép phát triển ứng dụng Web gồm 2 phần cơ bản:

* Phần ngôn ngữ Ruby: “Ruby là một ngôn lập trình mã nguồn mở, linh hoạt, với một sự nổi bật về sự đơn giản dễ dùng và hữu ích. Nó có cú pháp “tao nhã” và tự nhiên dễ đọc và dễ dàng để viết”.
* Phần Framework Rails bao gồm nhiều thư viện liên kết.

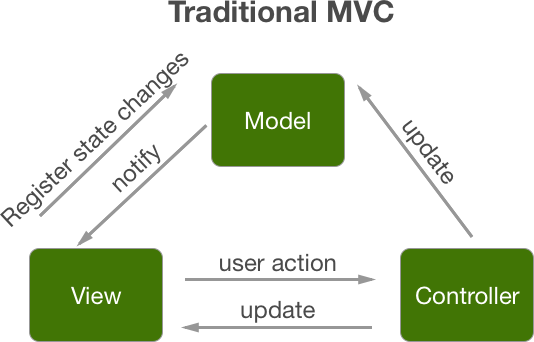
Một số trang web nổi tiếng sử dụng RoR:

* https://github.com/ : Hệ thống quản lý source phân tán, được nhiều công ty lớn sử dụng, và có hơn 1,5 triệu người đang sử dụng trang web này như một nơi lưu trữ các dự án của họ
* http://twitter.com/: là trang mạng xã hội lớn thứ hai sau facebook, rất được sử dụng phổ biến tại Mĩ và các nước châu Âu
* Lịch sử hình thành Ruby on Rails

Ruby được tạo ra bởi Yukihiro “Matz” Matsumoto từ 24/2/1993 và đưa ra bản chính thức vào năm 1995.

Rails ra mắt công chúng lần đầu tiên vào năm 2004, Rails thoạt đầu được dùng như là nền tảng cho một công cụ quản lý dự án Basecamp được tạo ra bởi nhà phát triển web David Heinemeier Hansson, một nhân viên của công ty phát triển web 37signals (Mỹ). Ban đầu họ xây dựng Rails không phải với mục đích là xây dựng ra một framework riêng, chủ tâm là dùng nó để xây dựng các ứng dụng khác của 37signals. Sau đó Heinemeier Hansson thấy tiềm năng của nó giúp cho anh ấy làm các công việc dễ dàng hơn bằng cách rút ra các tính năng phổ biến như trừu tượng cơ sở dữ liệu và khuôn mẫu (template) bên trong, và sau đó nó trở thành phiên bản đầu tiên được tung ra của Ruby on Rails.

### Mô hình MVC



**Hình 1. Mô hình MVC**

MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia một ứng dụng thành ba phần tương tác được với nhau để tách biệt giữa cách thức mà thông tin được xử lý nội hàm và phần thông tin được trình bày và tiếp nhận từ phía người dùng.

RoR sử dụng các mẫu kiến trúc Model – View – Controller(MVC) để tăng cường khả năng bảo trì và phát triển của ứng dụng. MVC cho phép chúng ta chia ứng dụng thành các tầng sử lý logic, nghiệp vụ và giao diện người dùng một cách rõ ràng, điều này cũng gúp cho việc kiểm thử và tái sử dụng code được dễ ràng hơn.

#### Model

Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý,…

Tầng Model xử lý các nghiệp vụ của ứng dụng và trực tiếp thao tác với dữ liệu. Trong RoR, tầng model thường được sử dụng để tương tác với các thành phần tương ứng với chúng trong cơ sở dữ liệu và validate dữ liệu.

#### View

Tầng view hiển thị giao diện người dùng, trong RoR các view là các file HTML được nhúng cùng với các đoạn mã Ruby. Các đoạn mã Ruby được nhúng trong HTML file là khá đơn giản, thường chỉ gồm các vòng lặp và các lệnh điều kiện rẽ nhánh, nó được sử dụng để hiện thị dữ liệu lên form của view.

#### Controller

Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng. Controller tương tác với Model và View. Các request đến từ trình duyệt sẽ được xử lý bởi Controller, sau đó Controller có thể sẽ tương tác với Model để lấy dữ liệu sau đó trả về cho View để hiển thị thông tin.

#### Ưu và nhược điểm của MVC

Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì.

Đối với dự án nhỏ việc áp dụng mô hình MC gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.

### PostgreSQL

#### PostgreSQL là gì?

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ và đối tượng dựa trên POSTGRES, bản 4.2, được khoa điện toán của đại học California tại Berkeley phát triển. POSTGRES mở đường cho nhiều khái niệm quan trọng mà các hệ quản trị dữ liệu thương mại rất lâu sau mới có.

PostgreSQL chạy được trên tất cả các hệ điều hành như Linux, UNIX, Mac OS X, Solaris và Windows. PostgreSQL cần rất ít lực để duy trì vì nó chạy rất ổn dịnh. Nó cũng không quy định những hạn chế trong việc sử dụng mã nguồn của phần mềm, vì thế nó có thể được dùng, sửa đổi và phổ biến bởi bất cứ ai cho bất cứ mục đích gì. Nếu phát triển ứng dụng bằng PostgreSQL, chi phí sẽ thấp hơn hẳn các hệ thống CSDL khác.

#### Các tính chất tiêu biểu:

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ và đối tượng. Là một chương trình mã nguồn mở xây dựng trên mã nguồn ban đầu của đại học Berkeley. Nó theo chuẩn SQL99 và có nhiều đặc điểm hiện đại:

* Kiểu dữ liệu người dùng định nghĩa
* Bảng kế thừa
* Khóa và khóa ngoại
* Các khung nhìn (view), rule, truy vấn phức hợp (subquery)
* Giao dịch bên trong vòng trong
* Việc kiểm tra truy cấp đồng thời đa phiên bản (multiverson concurrency control)
* Sao chép không đồng bộ
* Đánh chỉ mục partial , expression, GiST và GIN cùng nhiều chuẩn khác

Hơn nữa, PostgreSQL có thể dùng trong nhiều trường hợp khác, chẳng hạn như tạo ra các khả năng mới như:

* Kiểu dữ liệu
* Hàm
* Toán tử
* Hàm tập hợp
* Phương pháp liệt kê
* Ngôn ngữ theo thủ tục
* Truy vấn xử lý song song (parallel query)
* Sao chép dữ liệu dạng luồng (Streaming replication)

#### PostgreSQL – Điều gì làm nên sự khác biệt?

Là hệ quản trị CSDL mạnh, cung cấp các tính năng phức tạp như cung cấp truy vấn đồng thời đa phiên bản MVCC (có trước Oracle); khôi phục dữ liệu tại từng thời điểm (recovery), quản lý dung lượng bảng (tablespaces), sao chép không đồng bộ, giao dịch lồng nhau (savepoints), sao lưu trực tuyến hoặc nội bộ, truy vấn phức tạp và tối ưu hóa.

Có nhiều ngôn ngữ hộ trợ C++/C/Java/Perl, PHP, Ruby, ODBC..nên có thể khai thác nhiều thư viện giao tiếp với hàng trăm chức năng từ chức năng cơ bản như chuỗi số đến thuật toán phức tạp như mã hóa, và đặc biệt tương thích với Oracle. Trigger và các thủ tục có thể được viết thêm bằng C thêm vào CSDL như là một thư viện, cho phép linh hoạt mở rộng khả năng của mình.

PostgreSQL không quy định những hạn chế trong việc sử dụng mã nguồn của phần mềm. Cho phép các nhà phát triển có thể sáng tạo các kiểu dữ liệu tiên tiến của riêng mình từ các dữ liệu gốc ban đầu như địa chỉ vật lý, địa chỉ mạng..Cũng có thể phát triển plugin riêng như thêm 1 hàm tối ưu hóa thay cho 1 phần của hệ thống. Bởi vậy PostgreSQL có thể được dùng, sửa đổi và phổ biến bởi bất kỳ ai cho bất kỳ mục đích nào.

PostgreSQL cũng là hệ quản trị cơ sở dữ liệu hỗ trợ mạnh trong việc lưu trữ dữ liệu không gian. PostgreSQL kết hợp với module Postgischo phép người dùng lưu trữ các lớp dữ liệu không gian. Khi sử dụng PostgreSQL, Postgis kết hợp với các phần mềm GIS hỗ trợ hiển thị, truy vấn, thống kê hoặc xử lý dữ liệu không gian.

#### Độ phổ biến của PostgreSQL

Chính vì những ưu điểm và sự linh hoạt mà PostgreSQL đã được sự hài lòng của các chuyên gia về công nghệ và người dùng qua nhiều giải thưởng như Linux New Media cho hệ diều hành tốt nhất và năm lần chiến thắng giả do Linux Journal Editors bình chọn về DBMS tốt nhất. Nhiều công ty lớn như Apple, Fujitsu, Red Hat, Cisco, Juniter Network.. đều đang xây dựng sản phẩm dịch vụ trên nền PostgreSQL.

### Heroku

#### Heroku là gì?

Heroku là một nền tảng đám mây giống như một dịch vụ (PaaS) hỗ trợ một số ngôn ngữ lập trình. Heroku, một trong những nền tảng đám mây đầu tiên, đã được phát triển từ tháng 6 năm 2007. Ban đầu nó chỉ hỗ trợ ngôn ngữ lập trình Ruby, nhưng hiện nay đã hỗ trợ mở rộng thêm Java, Node.js, Scala, Clojure, Python, PHP và Go. Vì lý do này, Heroku được cho là nền tảng đa ngôn ngữ vì nó cho phép nhà phát triển xây dựng, chạy và mở rộng các ứng dụng theo cách tương tự trên tất cả các ngôn ngữ. Heroku đã được Salesforce.com mua lại vào năm 2010 với giá 212 triệu đô la.

#### Lịch sử hình thành Heroku

Heroku ban đầu được phát triển bởi James Lindenbaum, Adam Wiggins, và Orion Henry để hỗ trợ các dự án tương thích với nền tảng lập trình Ruby được gọi là Rack. Việc phát triển mẫu thử nghiệm mất khoảng sáu tháng. Sau đó, Heroku phải đối mặt với những hạn chế vì thiếu khách hàng thị trường thích hợp vì nhiều nhà phát triển ứng dụng đã sử dụng các công cụ và môi trường của riêng họ.

Vào tháng 10 năm 2009, Byron Sebastian gia nhập Heroku với tư cách là CEO. Vào ngày 8 tháng 12 năm 2010, Salesforce.com đã mua lại Heroku như một công ty con của Salesforce.com. Vào ngày 12 tháng 7 năm 2011, Yukihiro "Matz" Matsumoto, nhà thiết kế chính của ngôn ngữ lập trình Ruby, đã gia nhập công ty với tư cách là Kiến trúc sư trưởng, Ruby. Cùng tháng đó, Heroku đã thêm hỗ trợ cho Node.js và Clojure. Vào ngày 15 tháng 9 năm 2011, Heroku và Facebook đã giới thiệu “Heroku for Facebook”. Hiện tại Heroku hỗ trợ cơ sở dữ liệu Redis bên cạnh PostgreSQL chuẩn.

#### Kiến trúc Heroku

Các ứng dụng chạy trên Heroku thường có một tên miền duy nhất (thường là "applicationname.herokuapp.com") được sử dụng để định tuyến các yêu cầu HTTP tới dyno chính xác. Mỗi vùng chứa ứng dụng hoặc dynos, được trải đều trên một "lưới dyno" bao gồm một số máy chủ. Máy chủ Git của Heroku xử lý đẩy kho ứng dụng từ người dùng được phép.

Tất cả các dịch vụ Heroku được lưu trữ trên nền tảng điện toán đám mây EC2 của Amazon.

Công việc ở Heroku có thể được tóm tắt thành hai loại chính:

* Triển khai (Deploy)
* Nội dung chính của sự phát triển là source code, các phụ thuộc liên quan nếu chúng tồn tại và một Procfile cho command.
* Ứng dụng này được gửi tới Heroku bằng một trong các cách sau: Git, GitHub, Dropbox hoặc thông qua API.
* Có các gói có ứng dụng cùng với tất cả các phụ thuộc, và language runtime, và tạo ra các slugs. Chúng được gọi là build-pack và là phương tiện cho quá trình biên dịch slug.
* Một slug là một sự kết hợp / bundle của source code, các phụ thuộc được xây dựng, runtime và kết quả được biên dịch / tạo ra của hệ thống xây dựng đã sẵn sàng để thực thi.
* Tiếp theo là các Config vars chứa dữ liệu cấu hình tùy biến có thể được thay đổi độc lập với source code.
* Tiện ích bổ sung là các dịch vụ đám mây của bên thứ ba, chuyên biệt, giá trị gia tăng có thể dễ dàng được đính kèm vào một ứng dụng, mở rộng chức năng của nó.
* Bản phát hành là sự kết hợp của một slug(ứng dụng), config varsvà các tiện ích bổ sung (add-ons).
* Heroku duy trì một log know được gọi là append-only ledger của bản phát hành mà nhà phát triển tạo ra.

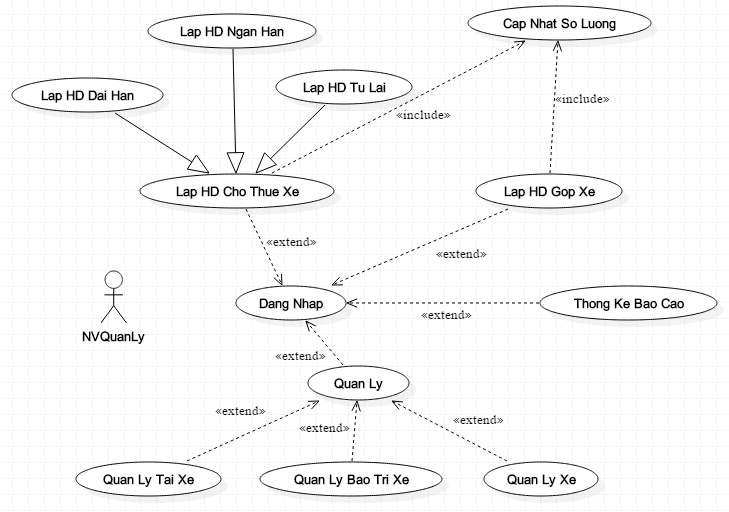
# PHẦN 3. KẾT QUẢ ỨNG DỤNG.

## PHÂN TÍCH.

### Mô hình CDM

### Mô hình PDM

### Mô hình Usecase



### Thiết kế CSDL logic

#### Chuyển đổi mô hình khái niệm dữ liệu sang mô hình đạt chuẩn 3NF.

1. NHÃN HIỆU (Mã nhãn hiệu, Code Nhãn Hiệu, Tên nhãn hiệu)
2. MODEL XE (Mã model, Tên Model, Số chỗ, Giá, Đơn vị tiền tệ)
3. HỢP ĐỒNG THUÊ XE (Số hợp đồng, Ngày bắt đầu, Ngày kết thúc, Thuê tài xế, Giá, Đơn vị tiền tệ)
4. XE (Mã xe, Biển số xe, Trạng thái, Số km đầu, Số km cuối, Trạng thái thuê, Giá, Đơn vị tiền tệ, Hình ảnh)
5. BẢNG GIÁ (Mã giá, Loại hình dịch vụ, Loại ngày thuê, Giá, Đơn vị tiền tệ)
6. KHÁCH HÀNG (Mã khách hàng, Tên khách hàng, Giới tính, Ngày sinh, Địa chỉ, Số điện thoại, Số chứng minh nhân dân)
7. TÀI XẾ (Mã tài xế, Tên tài xế, Giới tính, Ngày sinh, Địa chỉ, Số điện thoại, Số chứng minh nhân dân, Số bằng lái xe, Hạng bằng)
8. HỢP ĐỒNG ĐẦU TƯ (Số hợp đồng, Ngày bắt đầu, Ngày kết thúc, Hình thức đầu tư, Chiết khấu)
9. THÀNH VIÊN (Mã thành viên, Tên thành viên, Giới tính, Ngày sinh, Địa chỉ, Số điện thoại, Số chứng minh nhân dân)
10. HÀNH KHÁCH (Mã hành khách, Tên khách hàng, Giới tính, Ngày sinh, Địa chỉ, Số điện thoại)

#### Mô hình E-R

### Thiết kế CSDL vật lý

#### Table CAR\_BRANDS

|  |  |  |  |
| --- | --- | --- | --- |
| CAR\_BRANDS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | code | varchar(5) |  |
| 3 | name | varchar(30) |  |

#### Table CAR\_MODELS

|  |  |  |  |
| --- | --- | --- | --- |
| CAR\_MODELS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | name | varchar(20) |  |
| 3 | quantity\_of\_seat | integer |  |
| 4 | price | numeric(12,2) |  |
| 5 | unit\_price | varchar |  |

#### Table CAR\_RENTAL\_CONTRACTS

|  |  |  |  |
| --- | --- | --- | --- |
| CAR\_RENTAL\_CONTRACTS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | start\_date | date |  |
| 3 | end\_date | date |  |
| 4 | rent\_driver | boolean |  |
| 5 | price | numeric(15,2) |  |
| 6 | unit\_price | varchar |  |

#### Table CARS

|  |  |  |  |
| --- | --- | --- | --- |
| CARS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | car\_number\_plate | varchar(15) |  |
| 3 | status | varchar |  |
| 4 | before\_mileage | numeric(12,2) |  |
| 5 | after\_mileage | numeric(12,2) |  |
| 6 | price | varchar(20) |  |
| 7 | unit\_price | varchar |  |
| 8 | rent\_status | varchar |  |
| 9 | image | string |  |

#### Table COSTS

|  |  |  |  |
| --- | --- | --- | --- |
| COSTS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | service\_type | varchar |  |
| 3 | date\_type | varchar |  |
| 4 | cost | numeric(15,2) |  |
| 5 | unit\_price | varchar |  |

#### Table CUSTOMERS

|  |  |  |  |
| --- | --- | --- | --- |
| CUSTOMERS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | name | varchar |  |
| 3 | gender | varchar(10) |  |
| 4 | dob | date |  |
| 5 | address | varchar |  |
| 6 | tel | varchar |  |
| 7 | identity\_no | varchar |  |

#### Table DRIVERS

|  |  |  |  |
| --- | --- | --- | --- |
| DRIVERS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | name | varchar(30) |  |
| 3 | gender | varchar(10) |  |
| 4 | dob | date |  |
| 5 | address | varchar |  |
| 6 | tel | varchar |  |
| 7 | identity\_no | varchar |  |
| 8 | license\_no | varchar |  |
| 9 | class | varchar(2) | Hạng bằng |

#### Table INVESTMENT\_CONTRACTS

|  |  |  |  |
| --- | --- | --- | --- |
| INVESTMENT\_CONTRACTS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | start\_date | date |  |
| 3 | end\_date | date |  |
| 4 | type | varchar |  |
| 5 | discount | double precision |  |

#### Table MEMBERS

|  |  |  |  |
| --- | --- | --- | --- |
| MEMBERS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | name | varchar(30) |  |
| 3 | gender | varchar(10) |  |
| 4 | dob | date |  |
| 5 | address | varchar |  |
| 6 | tel | varchar |  |
| 7 | identity\_no | varchar |  |
| 8 | string | varchar |  |

#### Table PASSENGERS

|  |  |  |  |
| --- | --- | --- | --- |
| PASSENGERS | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| 1 | id | bigint |  |
| 2 | name | varchar |  |
| 3 | gender | varchar(10) |  |
| 4 | dob | date |  |
| 5 | tel | varchar |  |

Bảng Users?

## DEMO

# PHẦN 4. KẾT QUẢ ỨNG DỤNG.

## KẾT QUẢ ĐẠT ĐƯỢC.

### Giải quyết được các vấn đề sau:

* Xây dựng được hệ thống nhằm mục đích giúp cho người quản lý giải quyết các công việc được thuận lợi và dễ dàng hơn là quản lý trên sổ sách, giấy tờ, giảm chi phí, nhân công và thời gian. Giúp cho khách hàng đến đăng ký thuê xe và làm thủ tục được nhanh chóng hơn.
* Hỗ trợ khách hàng đăng ký thuê xe, chủ xe đăng ký góp xe trực tuyến thông qua giao diện web thân thiện mà không cần tốn thời gian đến trực tiếp HTX.
* Việc phân tích, thiết kế hệ thống đã tương đối đầy đủ, bám sát với nghiệp vụ hoạt động của một công ty chuyên cho thuê xe ô tô.
* Việc cập nhật dữ liệu là tương đối tốt.
* Chương trình hiện tại hỗ trợ cho việc mở rộng các chức năng sau này.

### Thu hoạch về kinh nghiệm chuyên môn:

* Tăng khả năng phân tích, thiết kế và cài đặt một hệ thống web tương đối hoàn chỉnh.
* Nâng cao tư duy lập trình, khả năng tự học và nghiên cứu công nghệ mới.
* Hiểu rõ hơn về Ruby cũng như framework Rails.
* …

## ƯU ĐIỂM, HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN.

### Ưu điểm:

### Hạn chế:

* Chưa hỗ trợ người dùng nhập dữ liệu từ file Excel, CSV, …
* Một số chức năng hiện có vẫn chưa tối ưu.
* Tính chuyên nghiệp chưa cao, chưa giải quyết trọn vẹn các vấn đề nảy sinh trong quá trình quản lý
* Tính thẩm mỹ chưa cao.

### Hướng phát triển:

* Hỗ trợ nhập dữ liệu từ file Excel, CSV, …
* Hỗ trợ xuất thống kê, báo cáo ra file PDF, Excel.
* Tự động gởi mail đến cho khách hàng khi đăng ký thuê xe thành công, khi có thay đổi về suất thuê và mail cảm ơn khi kết thúc hợp đồng thuê xe.
* Thiết kế hoàn thiện chức năng, giúp hệ thống chạy nhanh hơn.
* Chỉnh sửa giao diện thân thiện hơn với người dùng.
* Phát triển thêm tính năng đánh giá cho khách thuê xe.
* Phát triển thêm cho hệ thống chức năng thêm các mã giảm giá, chương trình khuyến mãi nhằm thu hút lượng lớn khách hàng.
* Phát triển tính năng blog trải nghiệm, du lịch giúp trang nội dung thêm đa dạng.

# TÀI LIỆU THAM KHẢO

1. Tài liệu sử dụng Ruby on Rails: <https://rubyonrails.org/>
2. Tài liệu sử dụng PostgreSQL: <https://www.postgresql.org/>
3. Tài liệu Heroku: <https://www.heroku.com/>