

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

*

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
KHOA CÔNG NGHỆ THÔNG TIN

**HỆ THỐNG QUẢN LÝ GỬI XE CHO CÁC
TOÀ NHÀ**

Sinh viên thực hiện : **Đinh Đức Thắng**

Lớp CNTT 02 - K63

Giáo viên hướng dẫn: **ThS. Lê Đức Trung**

HÀ NỘI 8 - 2022

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Đinh Đức Thắng.

Điện thoại liên lạc: 0869693858 Email: thangdinh2611@gmail.com

Lớp: CNTT-TT 3 - K63 Hệ đào tạo: Đại học chính quy.

Đồ án tốt nghiệp được thực hiện tại: Trường Công nghệ thông tin và truyền thông
- Đại học bách khoa Hà Nội.

Thời gian làm DATN: Từ ngày 11/04/2022 đến 29/07/2022.

2. Mục đích nội dung của DATN

Đưa ra giải pháp cho doanh nghiệp, cụ thể là các giải pháp quản lý hệ thống giữ xe cho các tòa nhà, khu đô thị mới

3. Các nhiệm vụ cụ thể của DATN

- Phân tích, thiết kế hệ thống của hệ thống quản lý bãi gửi xe
- Xây dựng mô hình web để mô phỏng lại tài liệu phân tích

4. Lời cam đoan của sinh viên:

Tôi - *Đinh Đức Thắng* - cam kết DATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Lê Đức Trung*. Các kết quả nêu trong DATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày tháng năm 2022

Tác giả DATN

Đinh Đức Thắng

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của DATN và cho phép bảo vệ:

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

Hà Nội, ngày tháng năm 2022

Giáo viên hướng dẫn

ThS. Lê Đức Trung

LỜI CẢM ƠN

Trước khi trình bày nội dung chính của đề án, em xin gửi tới thầy, cô của trường Công nghệ thông tin và truyền thông - Đại học Bách khoa Hà Nội lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, em có thể tiếp thu những kiến thức quan trọng, để em có thể hoàn thành đề án tốt nghiệp và những công việc của em sau này.

Em xin gửi lời cảm ơn chân thành nhất tới giảng viên – ThS. Lê Đức Trung đã hướng dẫn và giúp đỡ em rất nhiều không chỉ trong quá trình thực hiện đề án mà còn ở một số môn học trong khoảng thời gian ngồi trên giảng đường. Cảm ơn thầy đã giúp đỡ em.

Hà Nội, ngày 05 tháng 08 năm 2022

Sinh viên

Đinh Đức Thắng

Mục lục

Chương 1. Giới thiệu đề tài	8
1.1. Đặt vấn đề	8
1.2. Mục tiêu và phạm vi của đề tài	8
1.2.1. Giới thiệu bài toán	8
1.2.2. Phạm vi	8
1.3. Hướng giải quyết	9
1.4. Bố cục đồ án	9
Chương 2. Khảo sát và phân tích yêu cầu	10
2.1. Khảo sát hiện trạng	10
2.2. Tổng quan chức năng	11
2.2.1. Biểu đồ usecase tổng quan	11
2.2.2. Biểu đồ usecase phân rã	12
2.2.2.1. Biểu đồ use case phân rã quản trị viên	12
2.2.2.2. Biểu đồ use case phân rã nhân viên	13
2.2.3. Quy trình nghiệp vụ	13
2.2.3.1. Quy trình phiên sử dụng của nhân viên	14
2.2.3.2. Quy trình phiên sử dụng của quản trị viên	15
2.2.3.3. Quy trình soát vé của nhân viên	16
2.2.3.4. Quy trình gia tạo vé tháng của nhân viên	18
2.2.3.5. Quy trình gia hạn vé tháng của nhân viên	19
2.2.3.6. Quy trình tạo tài khoản nhân viên của quản trị viên	20
2.2.3.7. Quy trình kiểm tra thống kê doanh thu	21
2.2.3.8. Quy trình quản lý số lượng xe	22
2.2.3.9. Quy trình tính thời gian làm việc cho nhân viên	23
2.3. Đặc tả chức năng	23
2.3.1. Đặc tả chức năng của Nhân viên	23
2.3.1.1. Đặc tả usecase "Soát vé đầu vào"	24
2.3.1.2. Đặc tả usecase "Soát vé đầu ra"	24
2.3.1.3. Đặc tả các usecase "Quản lý vé tháng"	25
2.3.2. Đặc tả chức năng của Quản trị viên	26
2.3.2.1. Đặc tả các Usecase "Quản lý nhân viên"	26
2.3.2.2. Đặc tả các Usecase "Quản lý doanh thu"	29

2.4.Yêu cầu phi chức năng.....	30
2.4.1. Yêu cầu về hiệu năng và cách sử dụng.....	30
2.4.2. Yêu cầu về tính dễ dùng	31
2.4.3. Yêu cầu về bảo mật.....	31
Chương 3. Công nghệ sử dụng.....	32
3.1.Ngôn ngữ lập trình - JavaScript	32
3.2.Công nghệ BackEnd - NodeJS	32
3.3.Công nghệ FrontEnd - VueJS	33
3.4.Cơ sở dữ liệu sử dụng - MongoDB	34
3.5.Các công nghệ khác	35
3.5.1. Ngôn ngữ đánh dấu siêu văn bản HTML	35
3.5.2. Framework Bootstrap	35
Chương 4. Phát triển và triển khai ứng dụng	36
4.1.Thiết kế kiến trúc	36
4.1.1. Lựa chọn kiến trúc phần mềm.....	36
4.1.2. Thiết kế tổng quan.....	37
4.1.3. Sơ đồ triển khai hệ thống	38
4.2.Thiết kế chi tiết	39
4.2.1. Thiết kế giao diện.....	39
4.2.1.1. Sơ đồ dịch chuyển màn hình.....	39
4.2.1.2. Giao diện màn hình chính của nhân viên	40
4.2.1.3. Giao diện màn hình soát vé vào của nhân viên.....	40
4.2.1.4. Giao diện màn hình soát vé ra của nhân viên	41
4.2.1.5. Giao diện màn hình tạo vé tháng của nhân viên	41
4.2.1.6. Giao diện màn hình chính của quản trị viên	42
4.2.1.7. Giao diện màn hình quản lý tài khoản nhân viên	42
4.2.1.8. Giao diện màn hình tạo tài khoản nhân viên	43
4.2.1.9. Giao diện màn hình quản lý lương, thời gian làm việc	44
4.2.1.10. Giao diện màn hình thống kê doanh thu.....	44
4.2.2. Thiết kế module - API.....	44
4.2.2.1. Thiết kế module "Đăng nhập"	45
4.2.2.2. Thiết kế module "Soát vé" của nhân viên.....	46
4.2.2.3. Thiết kế module "Quản lý vé tháng" của nhân viên.....	47
4.2.2.4. Thiết kế module "Đăng xuất"	49
4.2.2.5. Thiết kế module "Quản lý tài khoản nhân viên" của quản trị viên	50
4.2.2.6. Thiết kế module "Quản lý lương, thời gian làm việc của nhân viên"	51
4.2.2.7. Thiết kế module "Thống kê doanh thu và số lượng xe"	52
4.2.3. Thiết kế cơ sở dữ liệu	54
4.2.3.1. Sơ đồ liên kết thực thể	54
4.2.3.2. Chi tiết các bảng trong cơ sở dữ liệu	55

4.3.Xây dựng ứng dụng	56
4.3.1. Công cụ sử dụng	56
4.3.2. Kết quả đạt được	56
4.3.3. Kiểm thử	57
4.3.3.1. Kiểm thử tính năng "Hiển thị danh sách các xe trong khu vực quản lý" ...	57
4.3.3.2. Kiểm thử tính năng "Soát vé đầu vào"	57
4.3.3.3. Kiểm thử tính năng "Soát vé đầu ra"	57
4.3.3.4. Kiểm thử tính năng "Tạo vé tháng"	57
4.3.3.5. Kiểm thử tính năng "Gia hạn vé tháng"	58
4.3.3.6. Kiểm thử tính năng "Huỷ vé tháng"	58
4.3.3.7. Kiểm thử tính năng "Tạo tài khoản nhân viên"	58
4.3.3.8. Kiểm thử tính năng "Thống kê doanh thu"	58
4.3.3.9. Kiểm thử tính năng "Quản lý lương, thời gian làm việc của nhân viên"	59
4.3.3.10. Kiểm thử tính năng "Thống kê số lượng xe, vé tháng và vé ngày đã soát"	59
Chương 5. Đánh giá và đưa ra chương trình chạy thử	60
5.1.Các bước để khởi động chương trình	60
5.2.Đánh giá hiệu suất và đưa ra những điều cần khắc phục trong quá trình phát triển	61
Chương 6. Kết luận và hướng phát triển	62
6.1.Kết luận	62
6.2.Hướng phát triển	62

LỜI MỞ ĐẦU

Trong thời kỳ hiện đại hoá của đất nước, với tốc độ phát triển hiện tại, việc xuất hiện rất nhanh các khu đô thị cũng như là các chung cư, trung tâm mua sắm là điều tất yếu. Do đó để đáp ứng nhu cầu cũng như là chuyển đổi hệ thống số, em đã nghĩ đến ý tưởng phát triển hệ thống quản lý bãi gửi xe: quản lý vé xe theo ngày và quản lý vé xe tháng.

Bài toán quản lý các vé xe dựa trên các thủ cứng có sẵn và giúp đảm bảo tính an toàn và bảo mật với bãi xe và khách hàng của khu đô thị, chung cư, Với các công nghệ xây dựng website phổ biến dễ dàng nâng cấp cũng như là bảo trì mỗi khi xảy ra sự cố.

Trong phạm vi của đề án, em cũng đã cố gắng để tìm tòi, tham khảo để đưa ra các tính năng mới nhằm hỗ trợ và tăng tốc độ sử dụng của sản phẩm. Tuy nhiên, với trình độ hiện tại, em mong rằng mình có thể bổ sung thêm kiến thức và kỹ năng để phát triển sản phẩm trong tương lai.

Chương 1

Giới thiệu đề tài

1.1. Đặt vấn đề

Với sự phát triển gần đây của cơ sở hạ tầng nhà ở của Việt Nam. Các chủ đầu tư đều có một mong muốn phát triển hệ thống quản lý số trong thời đại công nghệ 4.0. Trong đó ngoài việc quản lý chung của toà nhà thì việc xây dựng một phần mềm quản lý và xử lý thao tác thu giữ xe cũng rất quan trọng, ảnh hưởng chung đến nhu cầu và đáp ứng hài lòng của người dân. Việc xây dựng hệ thống quản lý bãi xe cũng yêu cầu không được sai sót và phải đảm bảo tính dễ dùng để có thể triển khai việc đào tạo nhân viên quản lý trong các toà nhà hoặc khu đô thị, đồng thời tích dễ dàng tích hợp trí tuệ nhân tạo trong việc phát triển sản phẩm sau này

1.2. Mục tiêu và phạm vi của đề tài

1.2.1. Giới thiệu bài toán

- Triển khai hệ thống với vai trò nhân viên của khu vực quản lý xe, đáp ứng các chức năng cơ bản nhất của một khu vực giữ xe. Hiệu suất của hệ thống đạt mức độ cao đồng thời có khả năng tích hợp IOT và trí tuệ nhân tạo nhận diện trong tương lai.
- Phát triển hệ thống với vai trò quản lý của khu vực quản lý, tích hợp các tính năng quản lý về doanh thu. Có thể trở thành một phần của hệ thống quản lý khu đô thị, tòa nhà.

1.2.2. Phạm vi

- Trong thực tế, bất kỳ phần mềm nào cũng cần có các tính năng quản lý người dùng, nhóm người dùng, và cần phân quyền sử dụng các chức năng trong hệ thống một cách linh động.
- Mục đích của phần mềm nhằm tạo ra phân hệ quản lý nhân viên cũng như việc vận hành khu vực gửi xe. Nhân viên sẽ đăng ký với quản trị viên của khu vực để có thể

được cấp tài khoản hệ thống gửi xe. Quản trị viên sẽ tạo tài khoản và mật khẩu cho nhân viên để có thể sử dụng hệ thống.

- Khi nhân viên có yêu cầu hoặc mong muốn lấy lại mật khẩu cần có sự trao đổi với quản trị viên nhằm đảm bảo tính an toàn khi sử dụng hệ thống.
- Quản trị viên có quyền vô hiệu hóa, xóa tài khoản và chấp thuận yêu cầu đổi mật khẩu của nhân viên. Ngoài ra, quản trị viên còn có các tính năng riêng biệt thích hợp với việc quản lý hệ thống còn các tính năng của nhân viên sẽ tập trung vào việc vận hành hệ thống.

1.3. Hướng giải quyết

Trong đồ án này, em đã xây dựng một phiên bản hệ thống phần mềm quản lý xe trên web với các công nghệ của NodeJS - một môi trường runtime chạy JavaScript với đầy đủ các tính năng cơ bản và tiện lợi khi kết hợp cùng cơ sở dữ liệu MongoDB. Chi tiết các phương pháp sẽ được trình bày ở các chương sau.

1.4. Bố cục đồ án

Đồ án được trình bày theo bố cục sau:

- **Chương 1 - Mở đầu:** giới thiệu đề tài, đặt vấn đề, mục tiêu phạm vi đề tài, định hướng giải pháp.
- **Chương 2 - Khảo sát và phân tích yêu cầu:** Chương này chỉ ra các khả năng cũng như là các quyền và hành động mà các tác nhân có thể thực hiện dựa trên quyền đã đăng ký. Ngoài ra, chương còn tạo ra nền móng cho cơ sở dữ liệu được xây dựng ở chương 4.
- **Chương 3 - Công nghệ sử dụng trong xây dựng và phát triển hệ thống:** Giới thiệu các công nghệ được sử dụng trong hệ thống và lý do tại sao chúng được chọn. Phân tích điểm mạnh phù hợp với dự án phần mềm này.
- **Chương 4 - Hệ thống kiến trúc của ứng dụng:** Trình bày mô hình kiến trúc của ứng dụng, thiết kế giao diện, đưa ra trình tự các bước xử lý trong hệ thống khi tác nhân bên ngoài thao tác và phần thiết kế cơ sở dữ liệu
- **Chương 5 - Đánh giá và đưa ra chương trình chạy thử:** Trình bày về hệ thống mẫu được phát triển dựa theo tài liệu phân tích thiết kế hệ thống, đánh giá bản chạy thử
- **Chương 6 - Kết luận và hướng phát triển:** Nhận xét tổng quan về đề tài, khó khăn, thiếu sót cũng như đưa ra các hướng phát triển trong tương lai

Chương 2

Khảo sát và phân tích yêu cầu

2.1. Khảo sát hiện trạng

Với tình hình hiện tại của các đô thị lớn - đặc trưng ở đây là Hà Nội. Em cũng đã tìm hiểu và đưa ra các đánh giá với tình hình hiện tại. Khi lượng người đổ dồn về thành phố khiến việc làm dẫn đến tình trạng các khu đô thị, các tòa chung cư ngày càng gia tăng với tốc độ cao.

Theo như bài viết trên báo EMagazine: "Khu vực đường Nguyễn Tuân (quận Thanh Xuân) trở thành điểm nóng về phát triển cao ốc chung cư tại Hà Nội. Theo tính toán, dọc trục đường Nguyễn Tuân dài 720 m nhưng có tới khoảng 6.000 căn hộ chung cư đã và đang mở bán. Nhiều dự án lớn được xây dựng dọc trục đường này là Imperia Garden (1.632 căn hộ), TNR Goldseason (1.500 căn hộ), 90 Nguyễn Tuân (820 căn hộ), Việt Đức Complex (700 căn hộ), Thống Nhất Complex (552 căn hộ), The Legend (460 căn hộ)...".

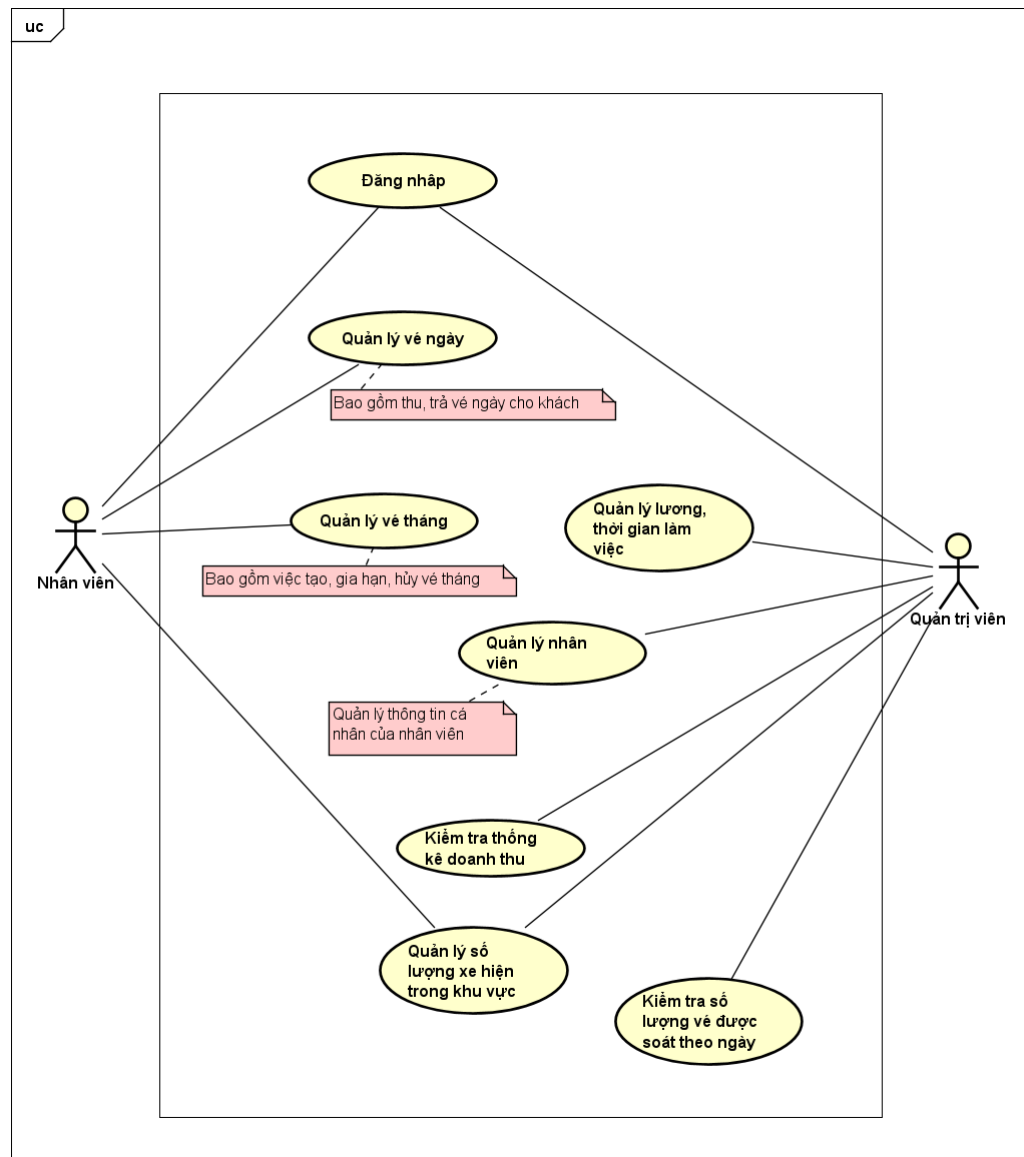
Còn trên báo tạp chí thương gia online: "Năm 2010, dân số đạt Hà Nội đạt khoảng 6,6 triệu người, toàn thị trường có hơn 20.000 căn nhà phố, chung cư chủ yếu là các căn hộ cũ, ước tính khoảng 73.000 căn. Đến năm 2019, dân số Hà Nội đạt 8,1 triệu người, khoảng 46.000 căn nhà phố và hơn 300.000 căn hộ chung cư. Theo nhận định, số lượng căn hộ chung cư tăng trưởng mạnh nhất trong các loại hình, tăng hơn 4 lần sau 10 năm."

Ta có thể thấy rằng việc các khu chung cư xuất hiện nhanh đồng nghĩa với việc quản lý thu giữ các loại phương tiện giao thông càng trở nên cấp thiết. Do đó, em nghĩ ra ý tưởng tạo ra hệ thống quản lý bãi gửi xe nhằm nâng cao chất lượng dịch vụ hạ tầng của khu đô thị, tòa nhà cũng như là giảm thiểu việc hao tốn thời gian với các hệ thống quản lý xe lỗi thời

2.2. Tổng quan chức năng

2.2.1. Biểu đồ usecase tổng quan

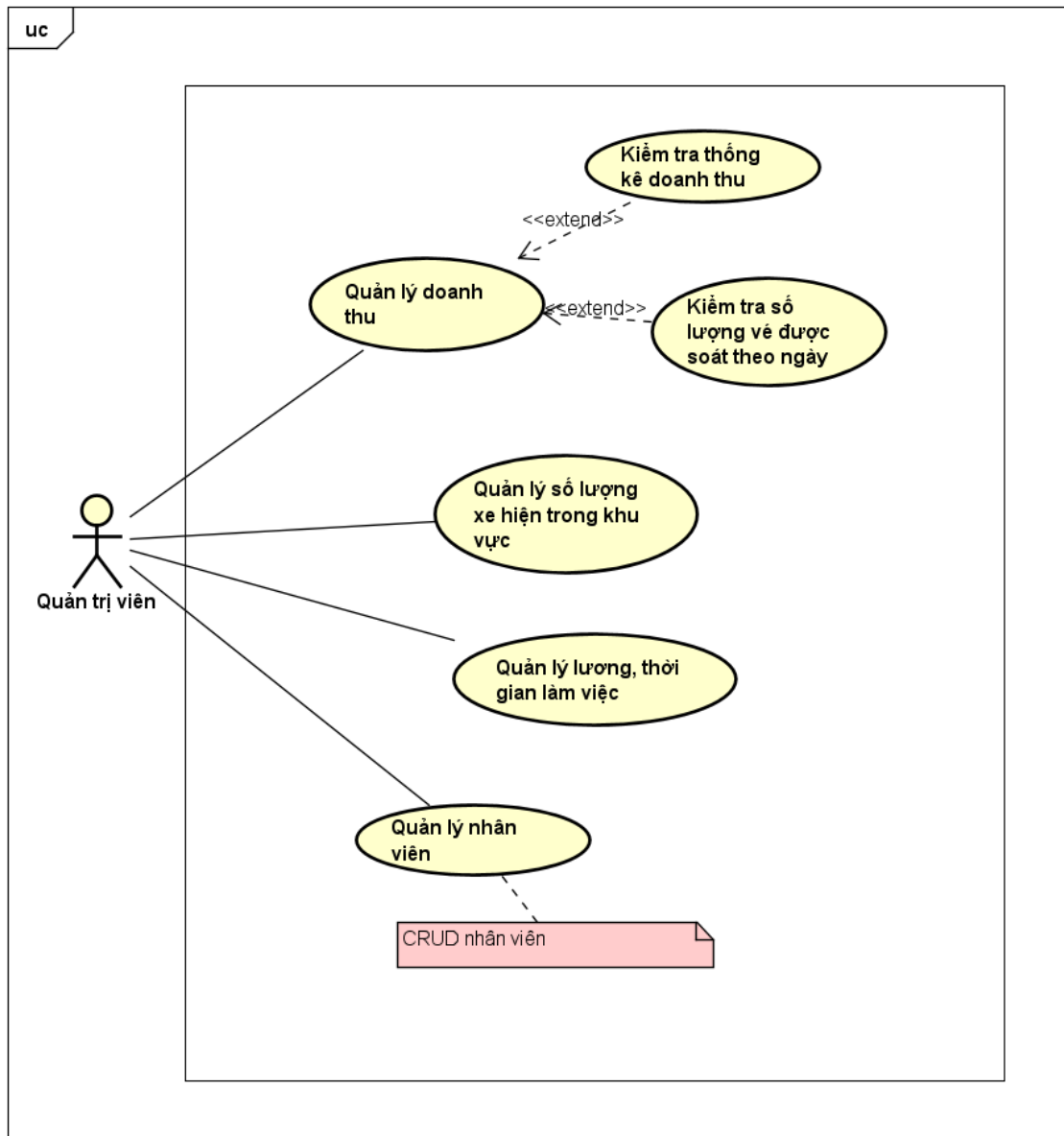
Các tác nhân tham gia: Nhân viên và Quản trị viên. Trong đó, Quản trị viên sẽ có vai trò quản lý các Nhân viên của mình và hệ thống.



Hình 2.2.1. Biểu đồ usecase tổng quan

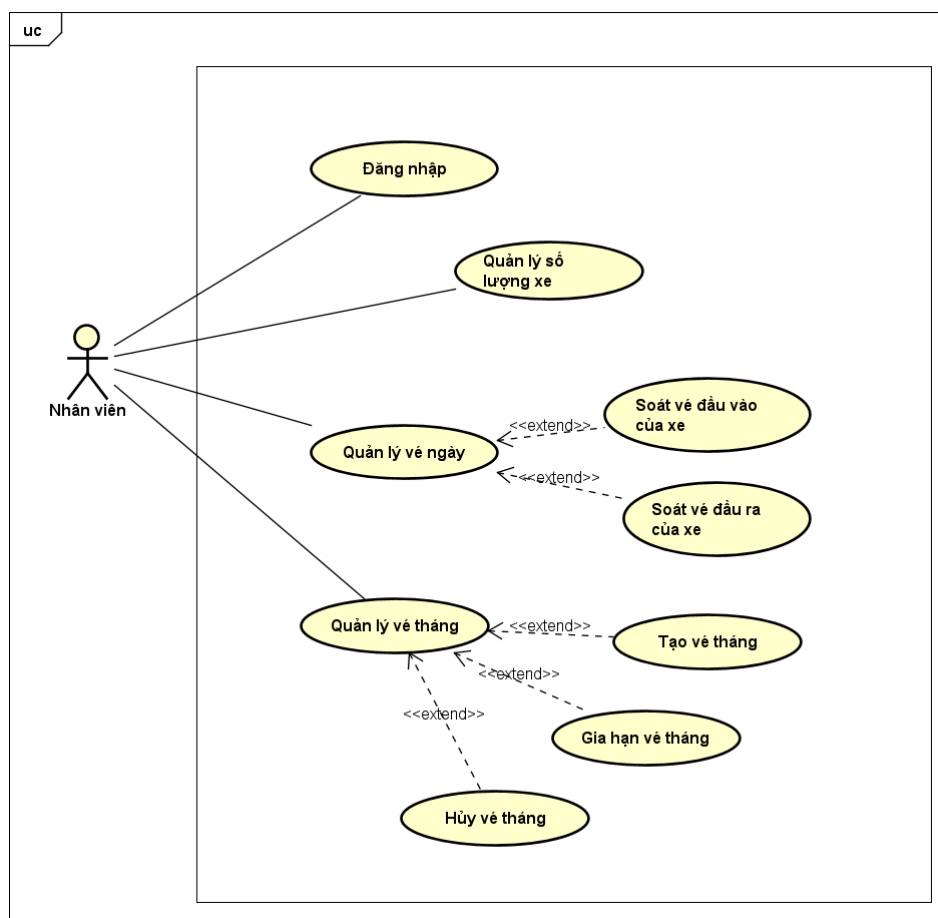
2.2.2. Biểu đồ usecase phân rã

2.2.2.1. Biểu đồ use case phân rã quản trị viên



Hình 2.2.2.1. Biểu đồ usecase phân rã quản trị viên

2.2.2.2. Biểu đồ use case phân rã nhân viên

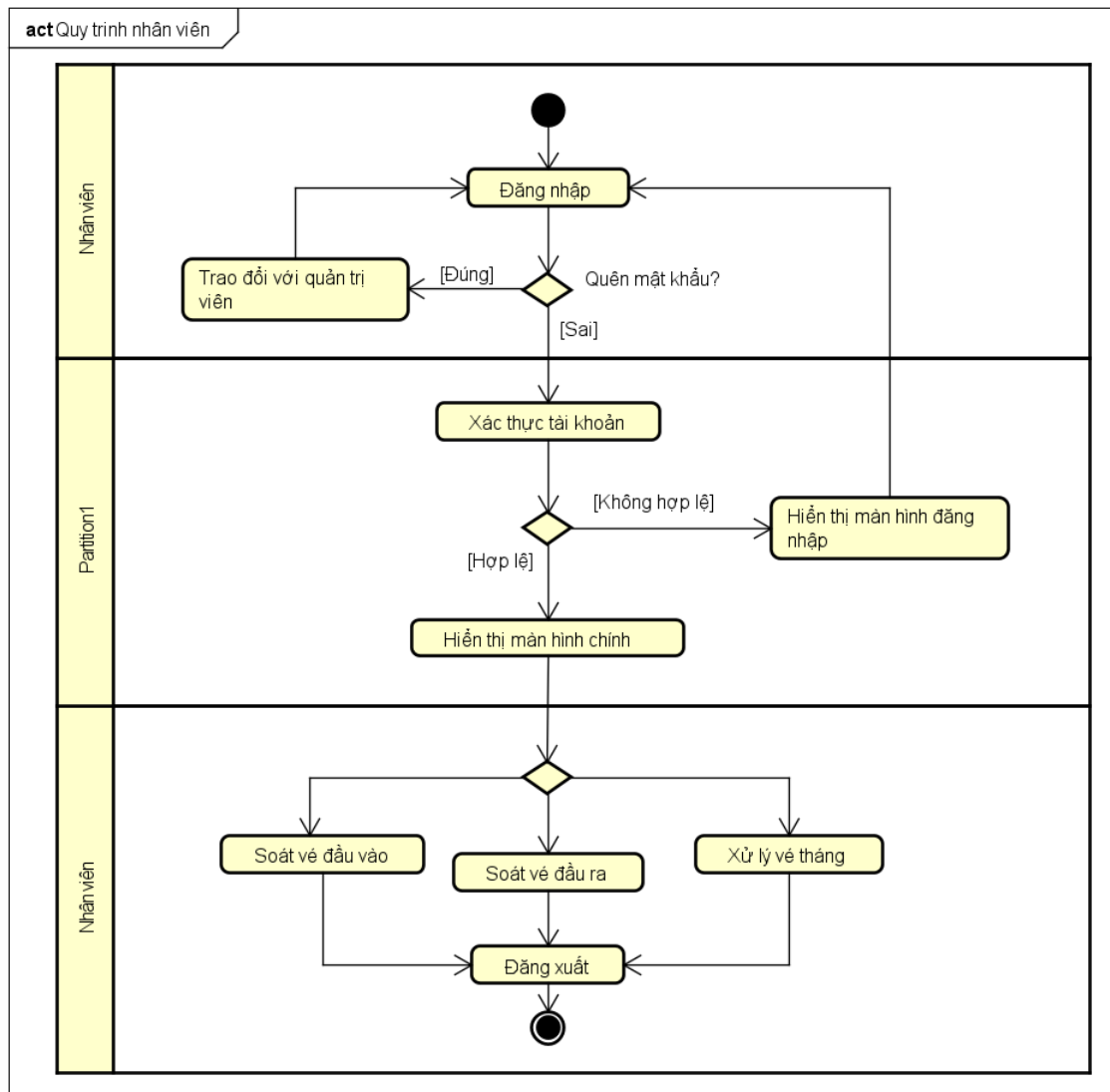


Hình 2.2.2.2. Biểu đồ usecase phân rã nhân viên

2.2.3. Quy trình nghiệp vụ

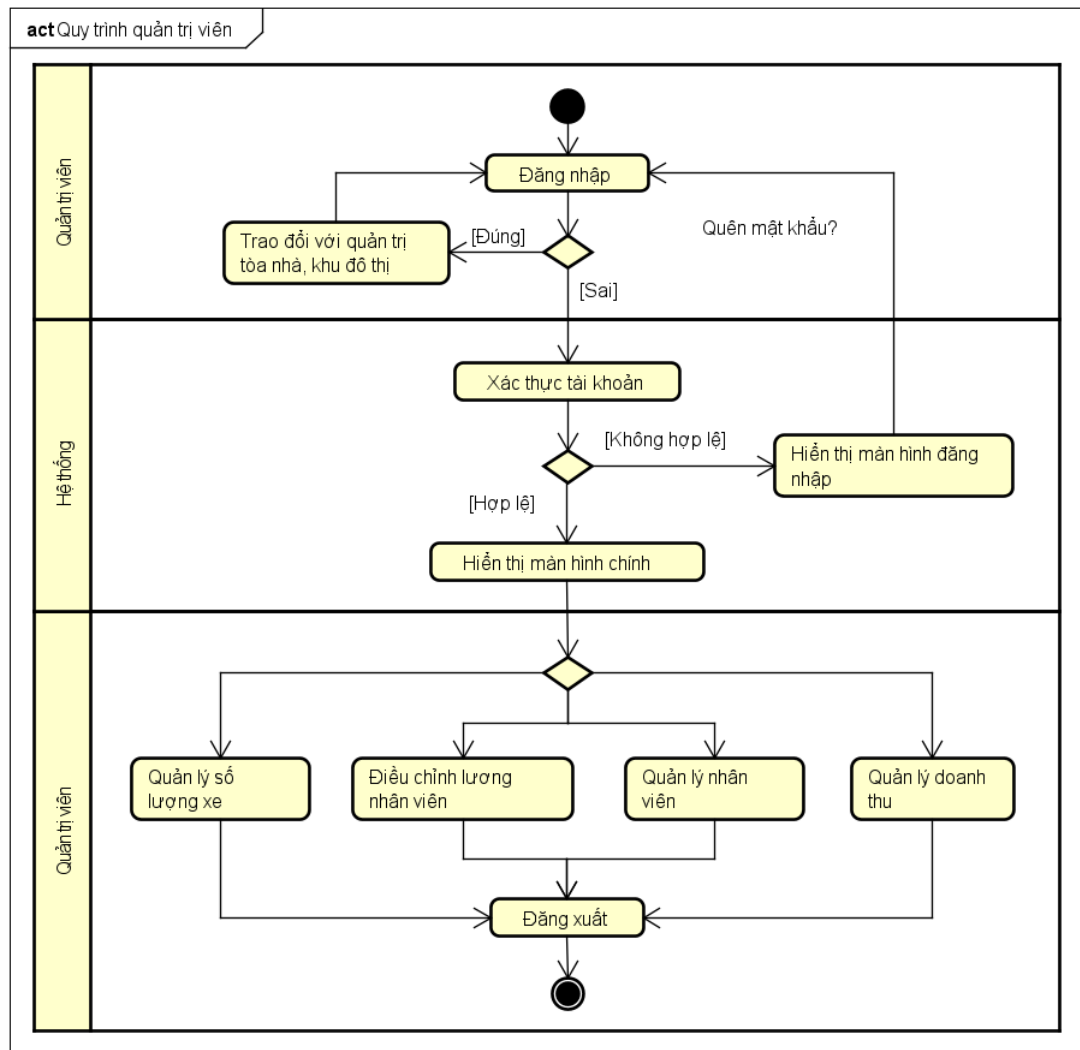
Trong phần này, em sẽ đưa ra quy trình phiên sử dụng của các tác nhân và một số luồng nghiệp vụ đặc trưng: quy trình soát vé của nhân viên, quy trình gia hạn vé tháng. Chi tiết về hành động trong các quy trình này được mô hình hoá trong các mục con của từng quy trình.

2.2.3.1. Quy trình phiên sử dụng của nhân viên



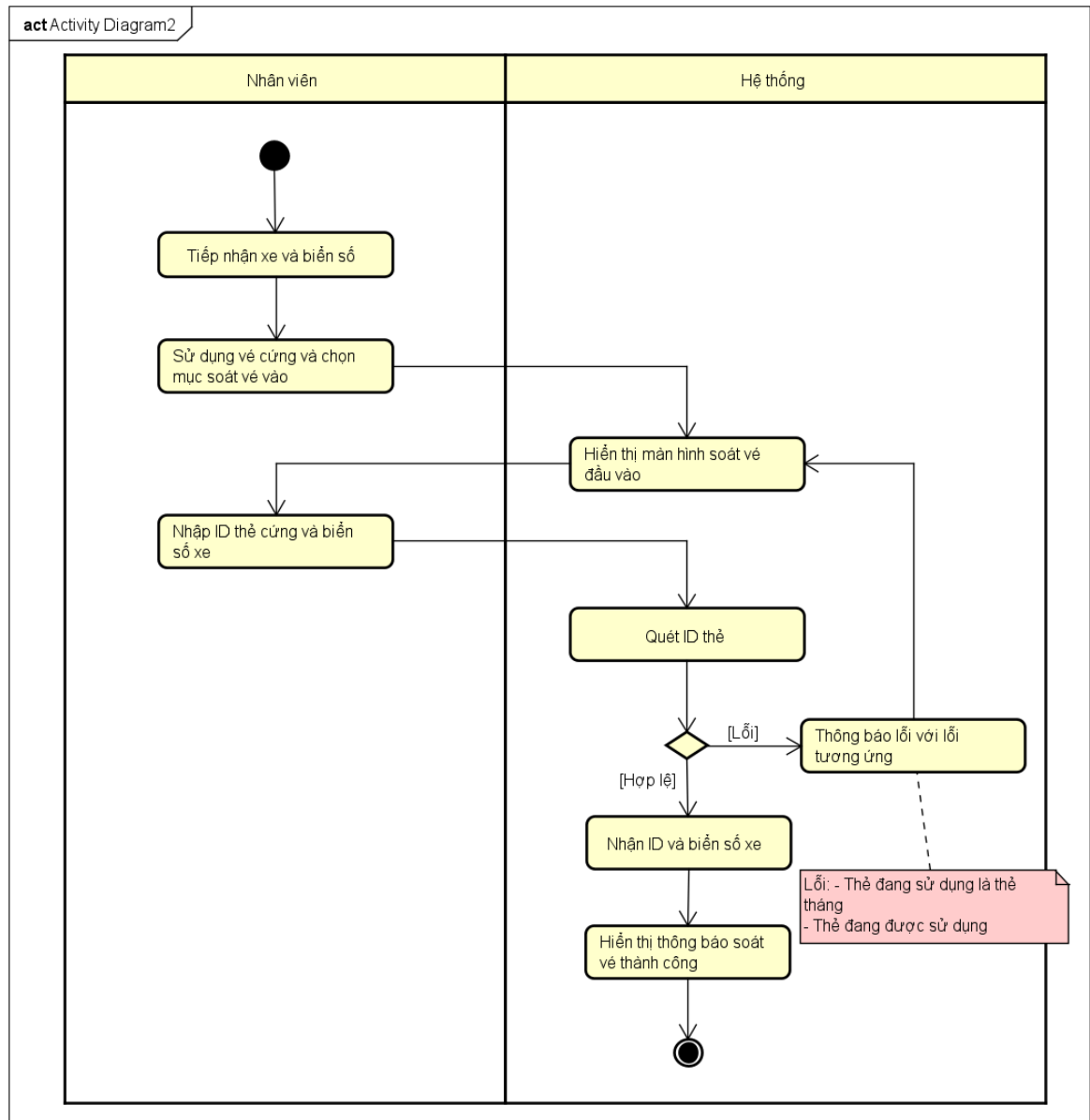
Hình 2.2.3.1 Quy trình phiên sử dụng của nhân viên

2.2.3.2. Quy trình phiên sử dụng của quản trị viên

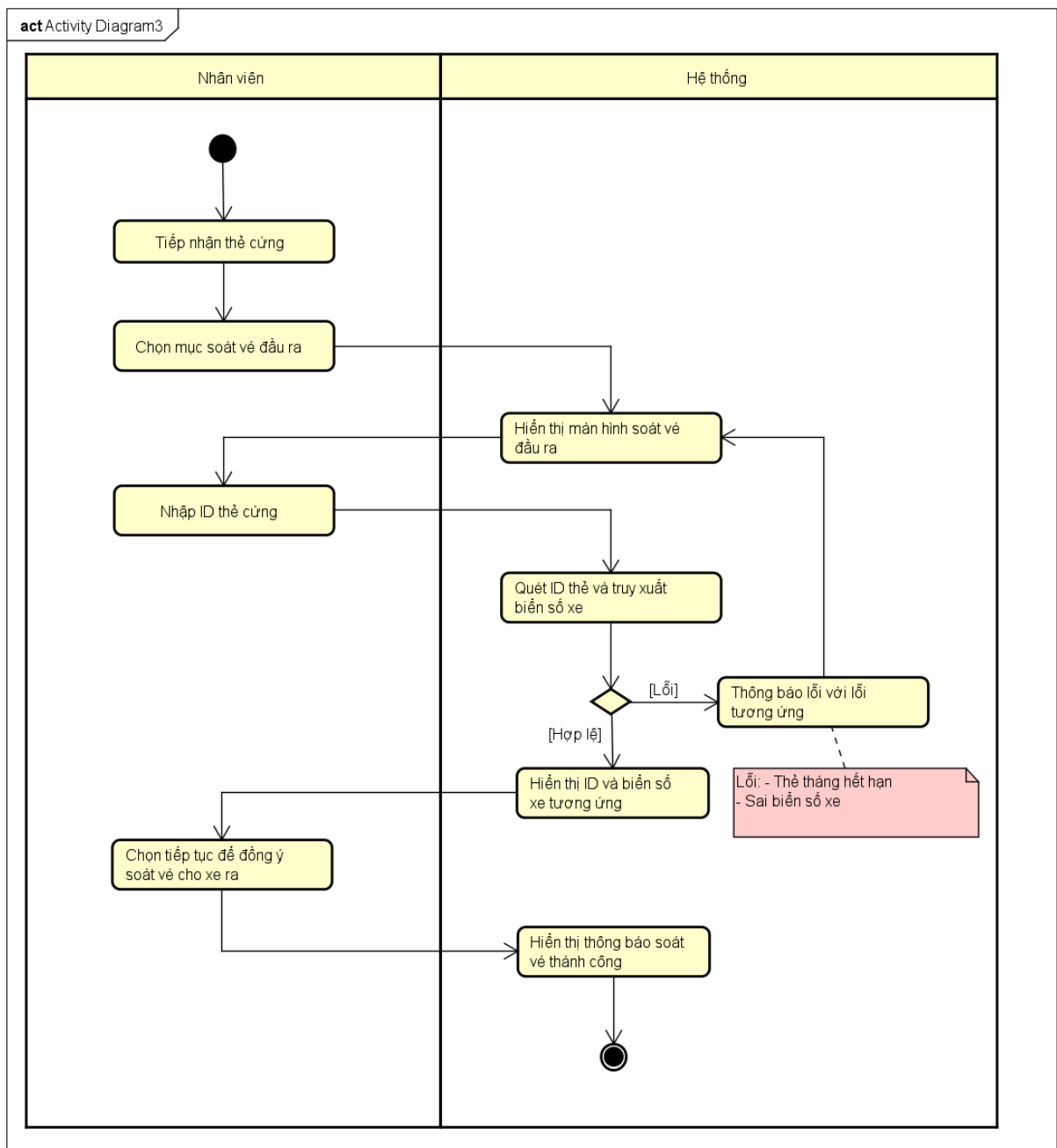


Hình 2.2.3.2 Quy trình phiên sử dụng của quản trị viên

2.2.3.3. Quy trình soát vé của nhân viên

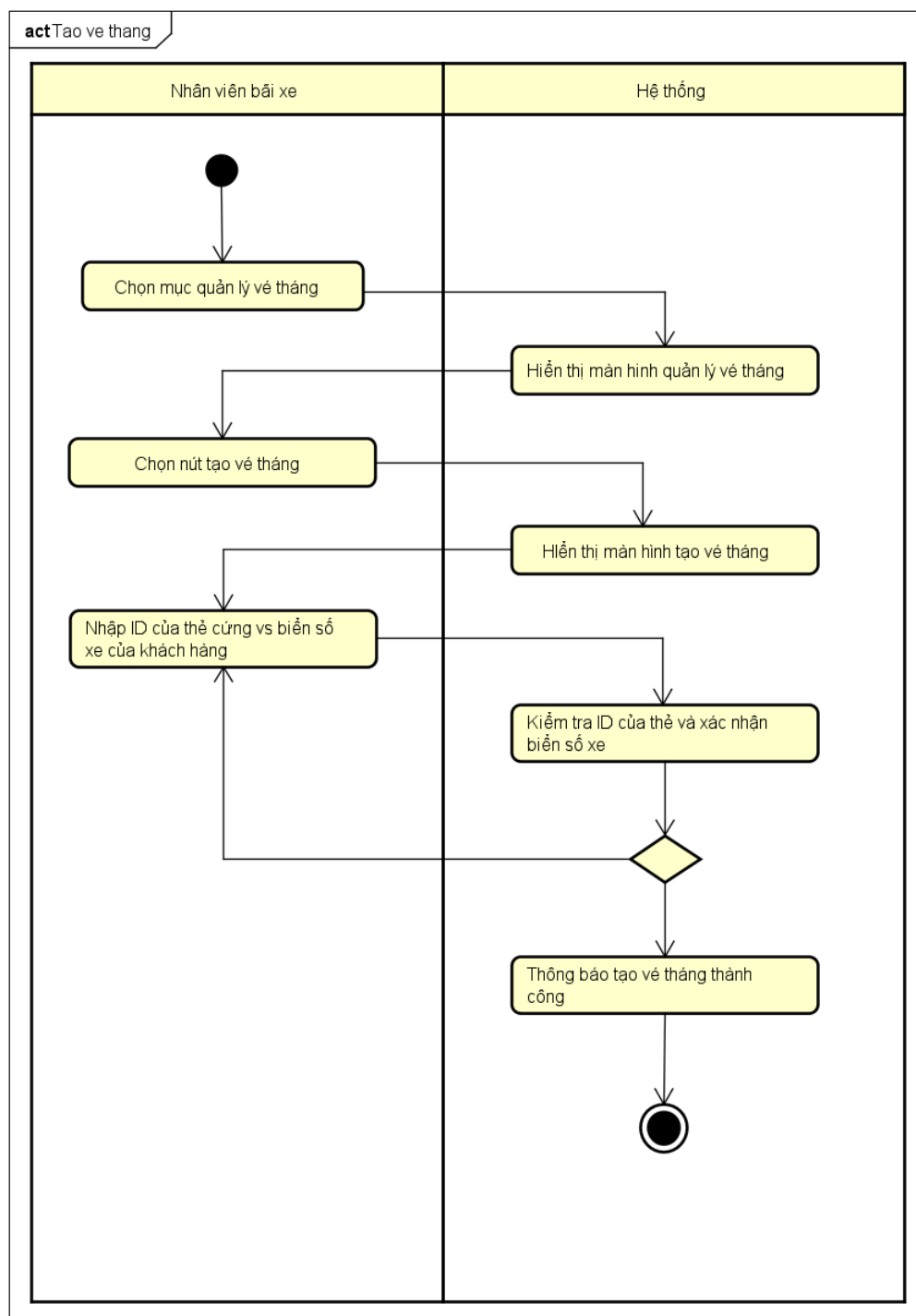


Hình 2.2.3.3a. Quy trình soát vé vào của nhân viên



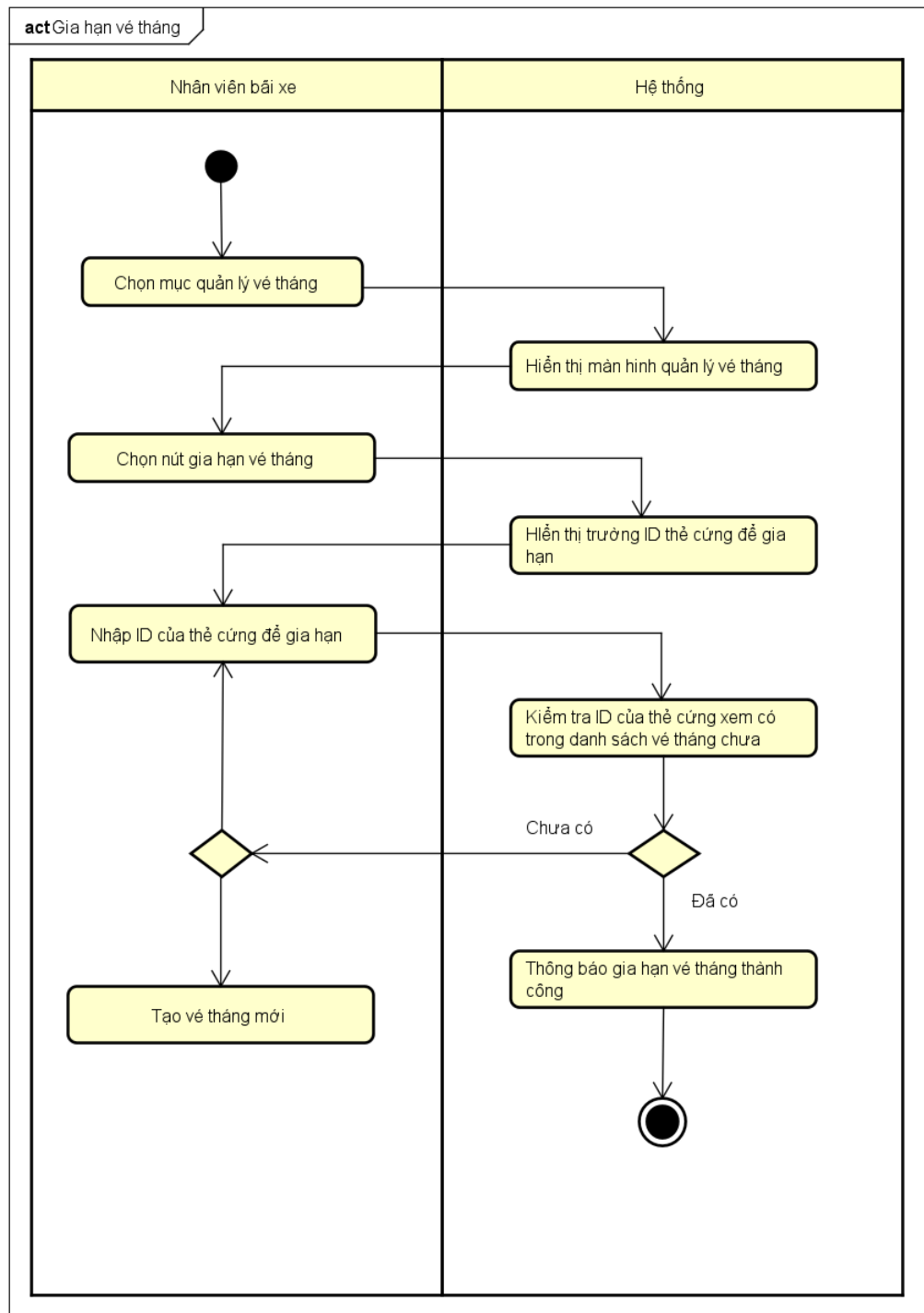
Hình 2.2.3.3b. Quy trình soát vé ra của nhân viên

2.2.3.4. Quy trình gia tạo vé tháng của nhân viên



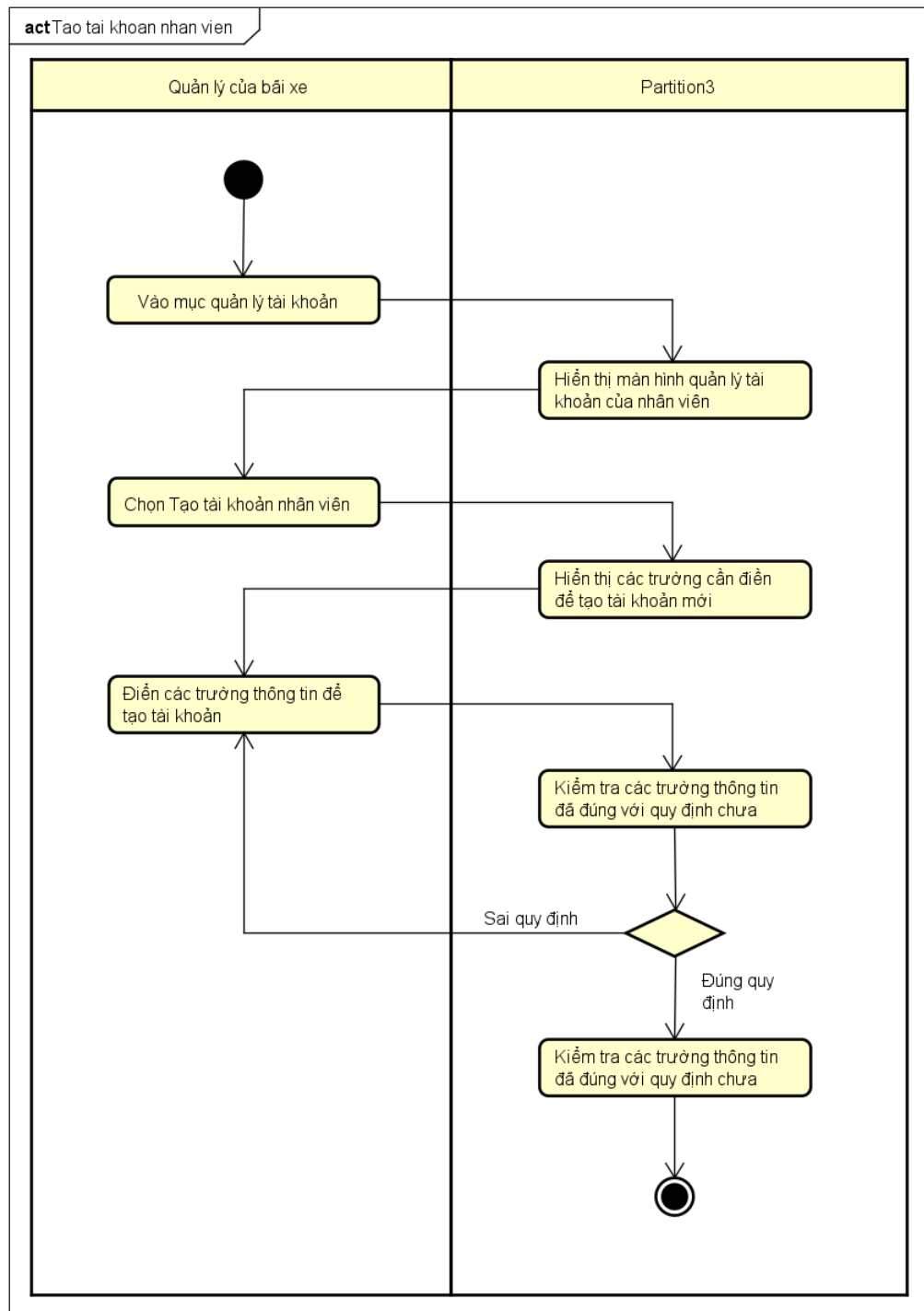
Hình 2.2.3.4. Quy trình tạo vé tháng của nhân viên

2.2.3.5. Quy trình gia hạn vé tháng của nhân viên



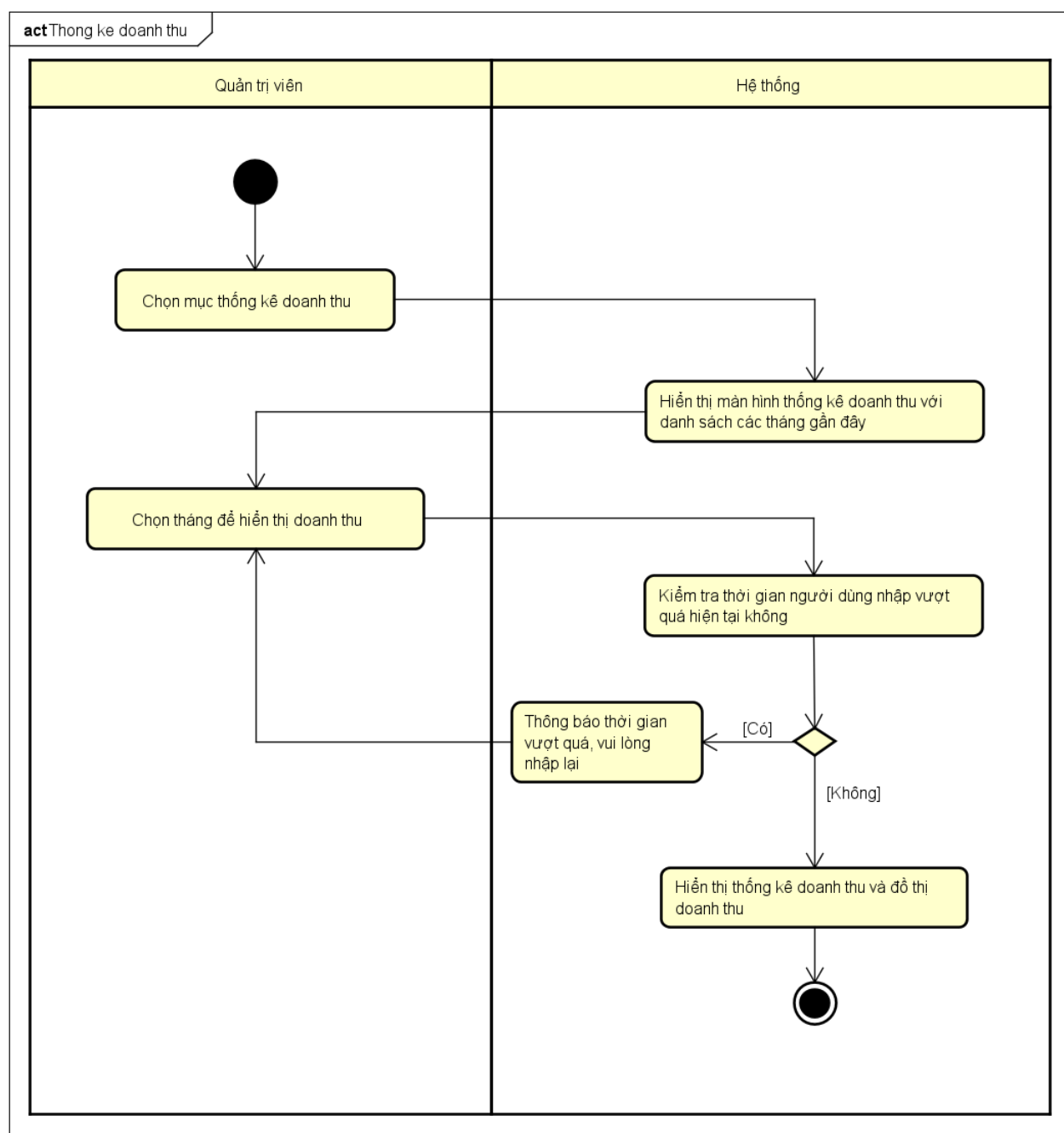
Hình 2.2.3.5. Quy trình gia hạn vé tháng của nhân viên

2.2.3.6. Quy trình tạo tài khoản nhân viên của quản trị viên



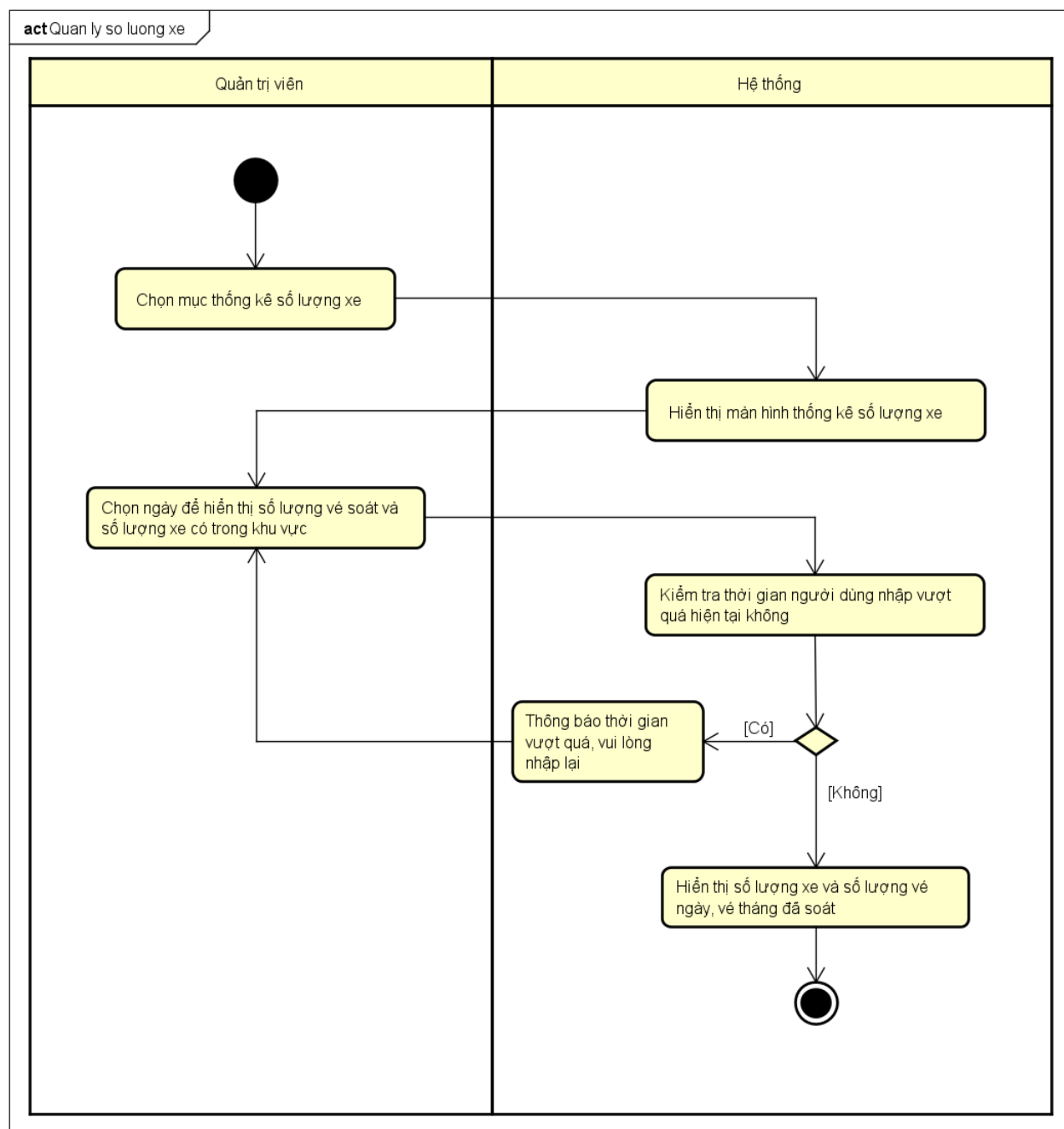
Hình 2.2.3.6. Quy trình tạo tài khoản nhân viên

2.2.3.7. Quy trình kiểm tra thống kê doanh thu



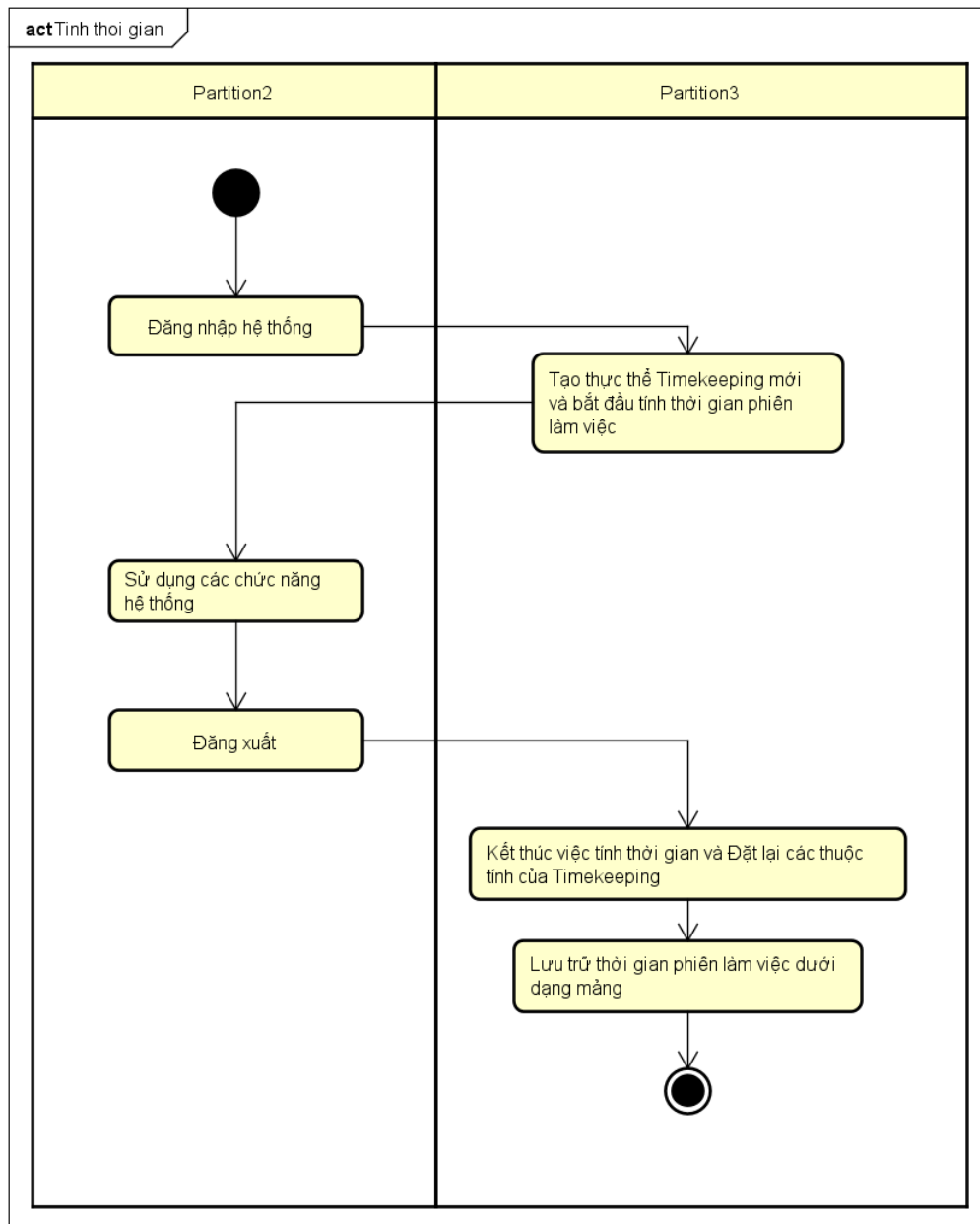
Hình 2.2.3.7. Quy trình kiểm tra, thống kê doanh thu của nhân viên

2.2.3.8. Quy trình quản lý số lượng xe



Hình 2.2.3.8. Quy trình quản lý số lượng xe

2.2.3.9. Quy trình tính thời gian làm việc cho nhân viên



Hình 2.2.3.9. Quy trình tính thời gian làm việc cho nhân viên

2.3. Đặc tả chức năng

Chi tiết về các use case được đưa ra trong phần 2 được đặc tả trong các phần dưới đây. Các usecase được đặc tả dưới đây là các tính năng đặc trưng có trong hệ thống giúp người đọc nắm bắt được khái quát rõ ràng nhất về phần mềm hệ thống.

2.3.1. Đặc tả chức năng của Nhân viên

Nhân viên có các chức năng đặc trưng là kiểm soát vé ngày và vé tháng. Các nhân viên khi muốn đăng ký hoặc chỉnh sửa tài khoản cần trao đổi với quản trị viên để được cấp tài khoản.

2.3.1.1. Đặc tả usecase "Soát vé đầu vào"

Tên use case	Soát vé đầu vào	
Tác nhân	Nhân viên	
Mô tả use case	Nhân viên sử dụng thẻ cứng để soát vé vào cho khách gửi xe	
Tiền điều kiện	Nhân viên đã đăng nhập	
Luồng sự kiện chính	Tác nhân	Hành động
	Nhân viên	Sử dụng thẻ cứng và tiếp nhận biển số xe
	Nhân viên	Chọn mục soát vé đầu vào
	Hệ thống	Hiển thị màn hình soát vé đầu vào
	Nhân viên	Nhập ID thẻ cứng và biển số xe
	Hệ thống	Quét ID của thẻ và nhận biển số xe
	Hệ thống	Thông báo soát vé vào thành công
	Nhân viên	Đưa thẻ cứng cho khách gửi xe
Luồng sự kiện thay thế	Tác nhân	Hành động
	Hệ thống	Thông báo lỗi khi quét ID thẻ cứng
	Nhân viên	Đổi thẻ cứng khác và nhập lại ID
Hậu điều kiện	Thông tin xe phải ở trên danh sách xe đang gửi	

2.3.1.2. Đặc tả usecase "Soát vé đầu ra"

Tên use case	Soát vé đầu ra	
Tác nhân	Nhân viên	
Mô tả use case	Nhân viên tiếp nhận thẻ cứng của khách hàng để soát vé ra cho khách gửi xe	
Tiền điều kiện	Nhân viên đã đăng nhập	
Luồng sự kiện chính	Tác nhân	Hành động
	Nhân viên	Nhận thẻ cứng từ khách gửi xe
	Nhân viên	Chọn mục soát vé đầu ra
	Hệ thống	Hiển thị màn hình soát vé đầu ra
	Nhân viên	Nhập ID thẻ cứng
	Hệ thống	Quét ID của thẻ và truy xuất biển số xe
	Hệ thống	Thông báo soát vé ra thành công
	Nhân viên	Thông báo khách có thể đưa xe ra khỏi khu vực gửi
Luồng sự kiện thay thế	Tác nhân	Hành động
	Hệ thống	Thông báo lỗi khi quét ID thẻ cứng
	Nhân viên	Yêu cầu khách hàng xem lại thẻ và thông tin để nhập lại ID thẻ
Hậu điều kiện	Danh sách các xe trong khu vực không còn xe đã được soát vé ra	

2.3.1.3. Đặc tả các usecase "Quản lý vé tháng"

Use case Quản lý vé tháng bao gồm các use case nhỏ và đặc trưng là Tạo và Gia hạn vé tháng, Hủy vé tháng tương tự gia hạn vé tháng.

Tên use case	Quản lý vé tháng	
Tác nhân	Nhân viên	
Mô tả use case	Nhân viên tiếp nhận thẻ cứng của khách hàng để tạo vé tháng, gia hạn hoặc hủy vé tháng	
Tiền điều kiện	Nhân viên đã đăng nhập	
Luồng sự kiện chính	Tạo vé tháng:	
	Tác nhân	Hành động
	Nhân viên	Nhận thẻ cứng từ khách gửi xe
	Nhân viên	Chọn mục quản lý vé tháng
	Hệ thống	Hiển thị màn hình quản lý vé tháng với danh sách vé tháng
	Nhân viên	Chọn tạo vé tháng
	Hệ thống	Hiển thị màn hình các trường thông tin cần điền để tạo vé tháng
	Nhân viên	Điền các thông tin cần thiết (*) để tạo vé tháng
	Hệ thống	Tạo vé tháng, thông báo tạo thành công và hiển thị lên danh sách
	Nhân viên	Trả lại thẻ cứng cho khách hàng
	Gia hạn vé tháng:	
	Tác nhân	Hành động
	Nhân viên	Nhận thẻ cứng từ khách gửi xe
	Nhân viên	Chọn mục quản lý vé tháng
	Hệ thống	Hiển thị màn hình quản lý vé tháng với danh sách vé tháng
	Nhân viên	Chọn nút Gia hạn vé tháng
	Hệ thống	Hiển thị màn hình với trường ID của vé tháng cần gia hạn
	Nhân viên	Nhập ID của thẻ cứng tương ứng với vé tháng
	Hệ thống	Thông báo gia hạn thành công
	Nhân viên	Trả lại thẻ cứng cho khách gửi xe
Luồng sự kiện thay thế	Không	
Hậu điều kiện	Danh sách vé tháng thay đổi tương ứng với các usecase	

(*) Các trường thông tin để tạo vé tháng

STT	Tên trường	Mô tả	Bắt buộc?	Ví dụ
1	ID thẻ	ID của thẻ cứng	Có	001, 002, 003
2	Biển số xe	Biển số xe của khách hàng	Có	29M1-30021
3	Loại hình xe	Xe của khách hàng là xe máy hoặc xe đạp	Có	Xe máy

2.3.2. Đặc tả chức năng của Quản trị viên

Quản trị viên có vai trò giám sát và quản lý khu vực gửi xe và nhân viên của bãi xe. Do đó, các chức năng đặc trưng của quản trị viên là "Quản lý nhân viên" và "Quản lý doanh thu".

2.3.2.1. Đặc tả các Usecase "Quản lý nhân viên"

Phần quản lý nhân viên bao gồm 2 phần là "Quản lý hồ sơ/tài khoản của nhân viên" và "Quản lý lương, thời gian làm việc của nhân viên"

- Usecase "Quản lý tài khoản nhân viên"

Tên use case	Quản lý tài khoản nhân viên	
Tác nhân	Quản trị viên	
Mô tả use case	Quản trị viên có thể thêm tài khoản nhân viên, thay đổi thông tin và xóa tài khoản	
Tiền điều kiện	Quản trị viên đã đăng nhập	
Luồng sự kiện chính	Tạo tài khoản nhân viên:	
	Tác nhân	Hành động
	Quản trị viên	Tiếp nhận thông tin của nhân viên
	Quản trị viên	Chọn mục tạo tài khoản nhân viên
	Hệ thống	Hiển thị màn hình danh sách tài khoản của các nhân viên
	Quản trị viên	Chọn nút tạo tài khoản
	Hệ thống	Hiển thị màn hình tạo tài khoản với các trường cần điền (*)
	Quản trị viên	Điền các trường thông tin cần thiết để tạo tài khoản sau đó chọn nút Tạo
	Hệ thống	Tiếp nhận thông tin tài khoản sau đó thông báo tạo thành công

Luồng sự kiện chính	Chỉnh sửa tài khoản nhân viên:	
	Tác nhân	Hành động
	Quản trị viên	Chọn mục quản lý nhân viên
	Hệ thống	Hiển thị màn hình danh sách tài khoản nhân viên
	Quản trị viên	Chọn ký tự chỉnh sửa tài khoản ở phía bên phải danh sách
	Hệ thống	Hiển thị thông tin của tài khoản được chọn
	Quản trị viên	Chỉnh sửa thông tin tài khoản theo ý muốn sau đó nhấn tiếp tục
	Hệ thống	Thông báo chỉnh sửa tài khoản thành công và hiển thị danh sách tài khoản
	Xóa tài khoản nhân viên:	
	Tác nhân	Hành động
	Quản trị viên	Chọn mục quản lý nhân viên
	Hệ thống	Hiển thị màn hình danh sách tài khoản nhân viên
	Quản trị viên	Chọn ký tự xóa tài khoản ở phía bên phải danh sách
	Hệ thống	Hiển thị thông báo xác nhận xóa cho quản trị viên
	Quản trị viên	Chọn nút "Có"
	Hệ thống	Xóa tài khoản của nhân viên và thông báo xóa thành công
Luồng sự kiện thay thế	Không	
Hậu điều kiện	Các thay đổi về thông tin tài khoản phải được hiển thị lên danh sách tài khoản sau mỗi usecase	

(*) Các trường thông tin của mục tạo tài khoản nhân viên:

STT	Tên trường	Mô tả	Bắt buộc?	Ví dụ
1	Tên nhân viên	Tên của nhân viên (Chỉ được dùng các chữ cái)	Có	Chien, Thang
2	Username	Tên đăng nhập của nhân viên (Phải dài hơn 5 ký tự)	Có	ddthang, ncchien
3	Mật khẩu	Mật khẩu của tài khoản nhân viên (Phải dài hơn 5 ký tự)	Có	123@abc, 20002111@
4	SDT	Số điện thoại của nhân viên (Chỉ bao gồm chữ số, độ dài chính xác 10 ký tự)	Có	0123456798
5	Địa chỉ	Địa chỉ của nhân viên	Không	So 6, duong Hoang Hoa Tham, Ba Dinh, Ha Noi

- Usecase "Quản lý lương, thời gian làm việc của nhân viên":

Tên use case	Quản lý lương, thời gian làm việc của nhân viên	
Tác nhân	Quản trị viên	
Mô tả use case	Quản lý có thể thay đổi mức lương và thống kê mức lương sẽ trả cho nhân viên	
Tiền điều kiện	Quản trị viên đã đăng nhập	
Luồng sự kiện chính	Tác nhân	Hành động
	Quản trị viên	Chọn mục quản lý lương nhân viên
	Hệ thống	Hiển thị màn hình quản lý lương
	Quản trị viên	Chọn tháng để xem mức lương của nhân viên tương ứng
	Hệ thống	Hiển thị màn hình bảng lương của các nhân viên
Luồng sự kiện thay thế	Tác nhân	Hành động
	Quản trị viên	Chọn nút thay đổi mức lương
	Hệ thống	Hiển thị trường điền mức lương mới cho nhân viên
	Quản trị viên	Nhập mức lương trả cho nhân viên tương ứng
	Hệ thống	Lưu mức lương và bắt đầu tính công thức mới vào tháng sau
Hậu điều kiện	Danh sách lương nhân viên phải thay đổi tương ứng sau các hành động	

2.3.2.2. Đặc tả các Usecase "Quản lý doanh thu"

Phần quản lý doanh thu, quản trị viên có thể xem doanh thu theo tháng và xem số lượng vé tháng, vé ngày theo ngày

- Usecase "Xem thống kê doanh thu"

Tên use case	Xem thống kê doanh thu	
Tác nhân	Quản trị viên	
Mô tả use case	Quản trị viên xem thống kê doanh thu theo ngày hoặc tháng của khu vực mình quản lý	
Tiền điều kiện	Quản trị viên đã đăng nhập	
Luồng sự kiện chính	Tác nhân	Hành động
	Quản trị viên	Chọn mục thống kê doanh thu
	Hệ thống	Hiển thị màn hình thống kê doanh thu với trường chọn ngày tháng
	Quản trị viên	Chọn hình thức hiển thị: theo ngày hoặc theo tháng và chọn ngày hoặc tháng tương ứng
	Hệ thống	Hiển thị màn hình thống kê doanh thu theo ngày hoặc tháng mà quản trị viên chọn
Luồng sự kiện thay thế	Tác nhân	Hành động
	Quản trị viên	Chọn ngày tháng vượt quá thời điểm hiện tại
	Hệ thống	Hiển thị thông báo chưa đến ngày hoặc tháng mà người dùng chọn
	Quản trị viên	Chọn lại ngày tháng phù hợp với thời điểm hiện tại
Hậu điều kiện	Không	

- Usecase "Xem số lượng xe và lượng vé tháng, vé ngày đã được soát"

Tên use case	Xem số lượng xe và số lượng vé bán được	
Tác nhân	Quản trị viên	
Mô tả use case	Quản trị viên xem thống kê số lượng xe và vé bán được	
Tiền điều kiện	Quản trị viên đã đăng nhập	
Luồng sự kiện chính	Tác nhân	Hành động
	Quản trị viên	Chọn mục thống kê số lượng xe
	Hệ thống	Hiển thị màn hình với trường chọn ngày tháng
	Quản trị viên	Chọn hình thức hiển thị: theo ngày hoặc theo tháng và chọn ngày hoặc tháng tương ứng
	Quản trị viên	Chọn ngày hoặc tháng tương ứng với hình thức hiển thị đã chọn
	Hệ thống	Hiển thị màn hình thống kê số lượng xe và vé theo ngày hoặc tháng mà quản trị viên chọn
Luồng sự kiện thay thế	Tác nhân	Hành động
	Quản trị viên	Chọn ngày tháng vượt quá thời điểm hiện tại
	Hệ thống	Hiển thị thông báo chưa đến ngày hoặc tháng mà người dùng chọn
	Quản trị viên	Chọn lại ngày tháng phù hợp với thời điểm hiện tại
Hậu điều kiện	Không	

2.4. Yêu cầu phi chức năng

2.4.1. Yêu cầu về hiệu năng và cách sử dụng

- Trong các chuỗi sự kiện của các use case, tất cả các bước có thao tác với CSDL, nếu có lỗi trong quá trình kết nối hoặc thao tác, cần có thông báo lỗi tương ứng để tác nhân biết là lỗi liên quan đến CSDL chứ không liên quan tới lỗi của người dùng
- Các use case do Quản trị viên và Nhân viên thì cần đăng nhập với vai trò tương ứng
- Các thao tác trong ứng dụng cần được trả về với tốc độ cao tránh việc làm trì hoãn các hoạt động bên ngoài của khách khi soát vé hoặc làm vé tháng

2.4.2. Yêu cầu về tính dễ dùng

Các chức năng được thiết kế dễ hiểu và dễ thao tác. Các trường thông tin cần điền rõ ràng và có hướng dẫn cụ thể các luồng quy trình cho người vận hành hệ thống

2.4.3. Yêu cầu về bảo mật

Hệ thống không cung cấp các mã OTP hay code để nhân viên thay đổi mật khẩu, thay vào đó nhân viên cần trao đổi với cấp trên về yêu cầu thay đổi mật khẩu. Hệ thống không xây dựng để có thể liên kết với internet nên tránh được nguy cơ bị tấn công mạng

Chương 3

Công nghệ sử dụng

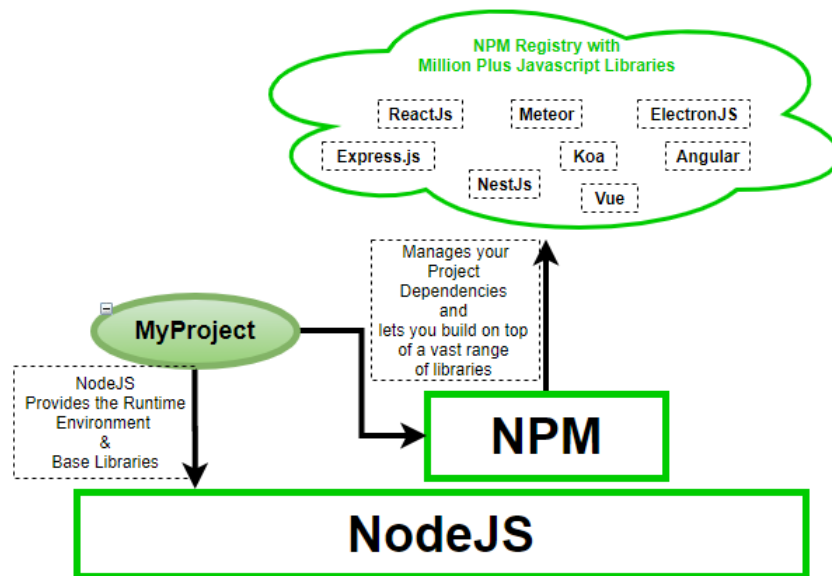
Ở chương này, em tập trung vào việc giới thiệu các công nghệ và cách thức ứng dụng công nghệ trong các tính năng có trong hệ thống. Sau một thời gian học hỏi và phân vân giữa lựa chọn công nghệ, em xin phép đưa ra các ưu điểm của từng thành phần phù hợp với mô hình và kiến trúc của chương trình

3.1. Ngôn ngữ lập trình - JavaScript

JavaScript là ngôn ngữ lập trình website phổ biến hiện nay, nó được tích hợp và nhúng vào HTML giúp website trở nên sống động hơn. JavaScript đóng vai trò như là một phần của trang web, thực thi cho phép Client-side script từ phía người dùng cũng như phía máy chủ (Nodejs) tạo ra các trang web động. Ngoài ra, JavaScript còn được biết đến nhờ khả năng xử lý bất đồng bộ `async/await` giúp tăng tốc độ xử lý.

3.2. Công nghệ BackEnd - NodeJS

NodeJS - được phát triển bởi Ryan Daln - là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng. Với thư viện có sẵn `npm` - Node package manager, việc cài đặt và sử dụng các package trở nên dễ dàng vào nhanh chóng cho việc cài đặt và triển khai sau này.



Hình 3.2. Hệ thống package NPM của nodeJS

NodeJS nhận và xử lý nhiều kết nối chỉ với một single-thread. Nhờ đó, hệ thống sẽ sử dụng ít lượng RAM nhất và giúp quá trình xử lý NodeJS nhanh hơn rất nhiều. Do đó, lượng phần cứng cần thiết để cài đặt hệ thống không cần quá cao cấp, đồng thời, tốc độ hệ thống cũng được thực hiện trong thời gian ngắn nhờ cơ chế bất đồng bộ, tránh gây ra việc hệ thống sụt giảm hiệu năng. Ngoài ra, NodeJS cũng xây dựng các Proxy phân vùng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.

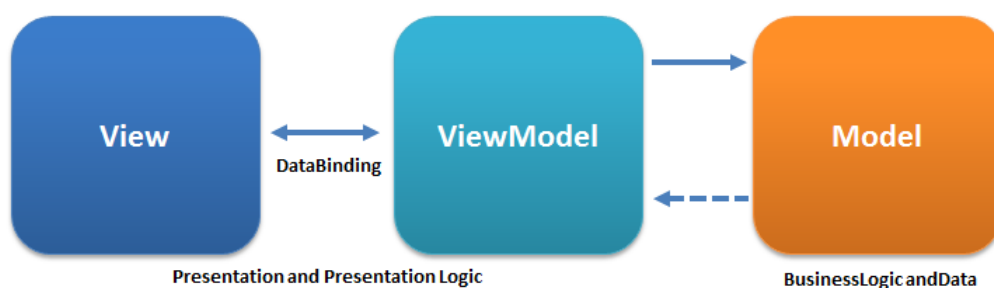
Các tính năng trong hệ thống quản lý gửi xe cũng mang khuôn dạng của các RESTful API - và javascript gần như là ngôn ngữ xử lý JSON dễ dàng nhất. Điều này cũng dẫn đến việc đồng nhất ngôn ngữ lập trình javascript trong cùng một dự án.

3.3. Công nghệ FrontEnd - VueJS

VueJS là một framework Javascript được tạo bởi Evan You với mục đích xây dựng giao diện người dùng cũng như xây dựng Single Page Application thân thiện với người dùng. Chúng xây dựng từ các thư viện, cách triển khai component, các chức năng đặc trưng của nó như SFC (Single File Component).

Vue tập trung xây dựng thư viện lõi vào lớp "View", do đó, việc triển khai rất nhẹ nhàng và không tốn nhiều tài nguyên, phù hợp với dự án nhỏ và linh hoạt. VueJS cũng là một framework dễ học và cung cấp tài liệu liên quan khá chi tiết và đầy đủ - thích hợp với việc xây dựng từ một bản phân tích thiết kế hệ thống đã có sẵn.

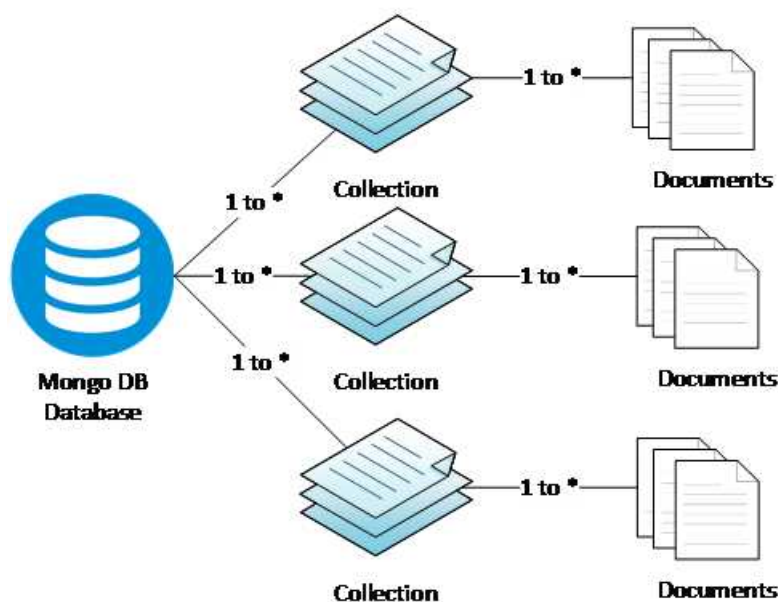
VueJS hoạt động theo mô hình MVVM, tạo ra các layer/ViewModel có thể tái sử dụng và ứng dụng cho nhiều View khác nhau, rút ngắn thời gian phát triển hệ thống



Hình 3.3. Mô hình MVVM

3.4. Cơ sở dữ liệu sử dụng - MongoDB

MongoDB là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON (Binary JSON). MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một model/collection/table sẽ có các kích cỡ và các bản ghi khác nhau. Các dữ liệu độc lập được lưu trữ trong các bản ghi kiểu JSON nên việc truy vấn dữ liệu sẽ được xử lý rất nhanh. Ngoài ra, MongoDB cũng có các ad hoc query - hỗ trợ search bằng field, search thông thường, regular expression searches, và range queries giúp tăng tốc độ truy vấn.



Hình 3.4. Cấu trúc lưu trữ, truy xuất của Mongo DB

Một trong những đặc điểm quan trọng của MongoDB chính là Replication: "nhân bản" dữ liệu đang tồn tại thích hợp với nhu cầu sử dụng dữ liệu lớn nhưng đòi hỏi toàn vẹn. Do đó, luôn có một bản sao đề phòng được MongoDB đưa ra để phòng sự cố.

Không những thế, mô hình dữ liệu tài liệu của MongoDB ánh xạ tự nhiên đến các đối tượng trong mã ứng dụng, hỗ trợ việc tìm hiểu và sử dụng. Tài liệu cung cấp khả năng

đại diện cho các mối quan hệ phân cấp trong việc lưu trữ các mảng và các cấu trúc phức tạp khác. MongoDB cũng phép truy vấn ngẫu nhiên, tập hợp theo thời gian thực và lập chỉ mục phục vụ việc truy cập và phân tích dữ liệu của bất kỳ cấu trúc nào.

3.5. Các công nghệ khác

3.5.1. Ngôn ngữ đánh dấu siêu văn bản HTML

HTML (viết tắt của từ HyperText Markup Language) là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Nó có thể được trợ giúp bởi các công nghệ như CSS và các ngôn ngữ kịch bản giống như JavaScript. HTML có thể nhúng các chương trình được viết bằng scripting như JavaScript, điều này ảnh hưởng đến hành vi và nội dung của các trang web. Việc bao gồm CSS xác định giao diện và bố cục của nội dung.

3.5.2. Framework Bootstrap

Bootstrap là một Framework bao gồm 3 thành phần cơ bản là: HTML, CSS và JavaScript. Sử dụng Bootstrap giúp việc thiết kế web tiết kiệm nhiều thời gian và đơn giản hơn. Bootstrap là một bộ sưu tập hoàn toàn miễn phí bao gồm các mã nguồn mở và các công cụ dựng front-end website với đầy đủ các thành phần. Bootstrap sẽ quy định sẵn các thuộc tính về kích thích, màu sắc và các chiều dài, chiều rộng của các vùng website. . . . Thông qua đó, designer có thể dễ dàng sáng tạo ra các website theo mong muốn nhưng vẫn đảm bảo tính thẩm mỹ.

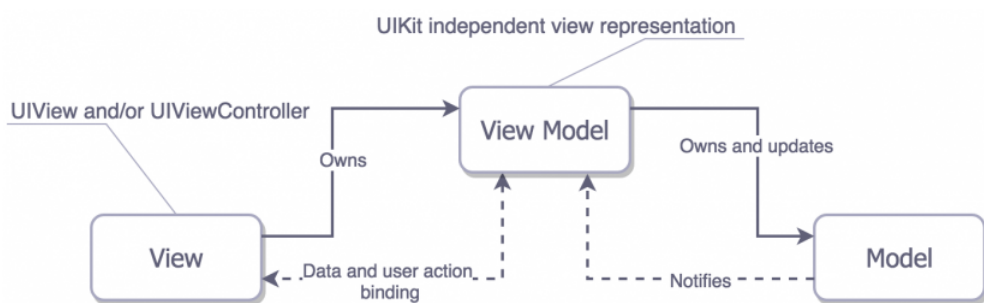
Chương 4

Phát triển và triển khai ứng dụng

4.1. Thiết kế kiến trúc

4.1.1. Lựa chọn kiến trúc phần mềm

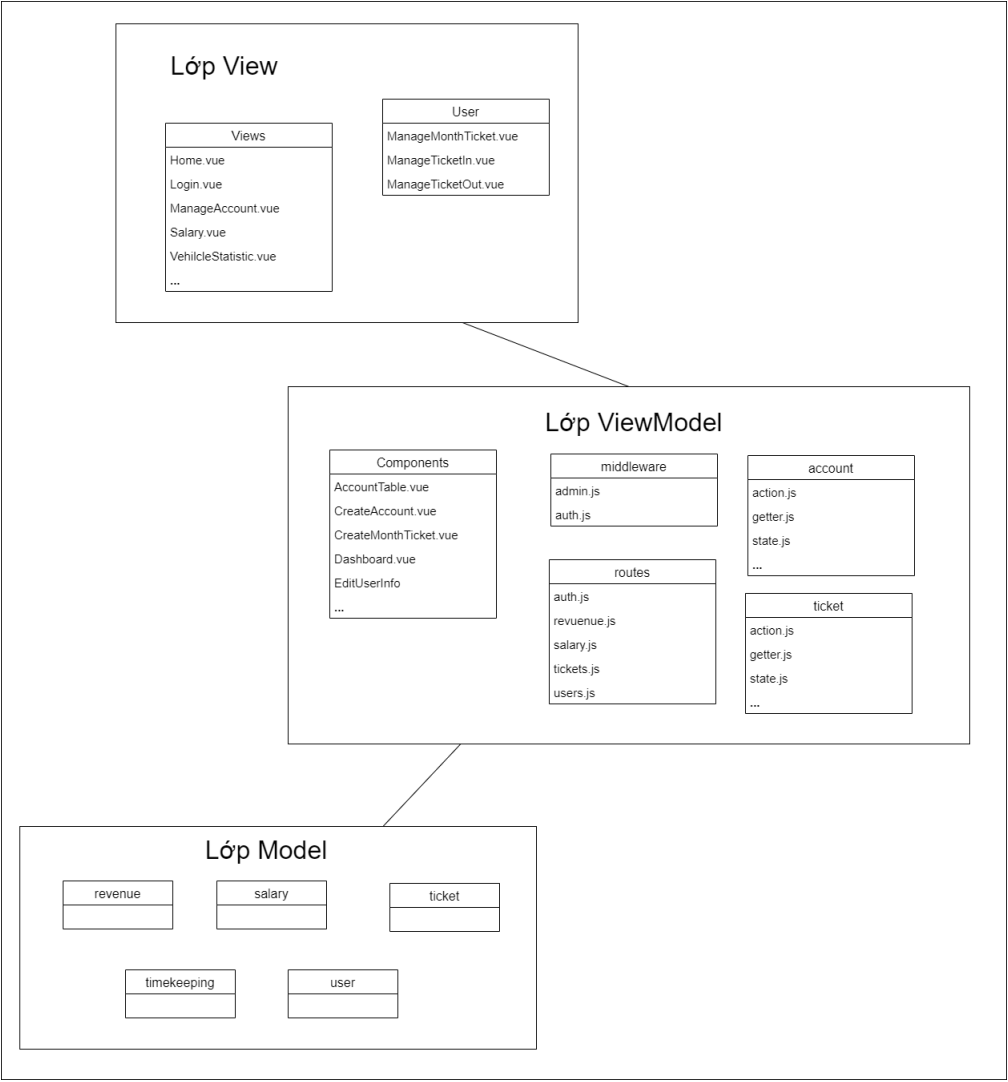
Kiến trúc phần mềm được em lựa chọn là mô hình MVVM (Model - View - ViewModel) tích hợp với công nghệ của VueJS. Ưu điểm của mô hình này chính là việc các ViewModel có thể tái sử dụng được cho nhiều View. Theo như hình vẽ dưới đây các tầng sở hữu (owns) sẽ không được biết đến bởi các tầng phía dưới nó, hỗ trợ cho Event/Data Stream.



Kiến trúc MVVM và các thành phần

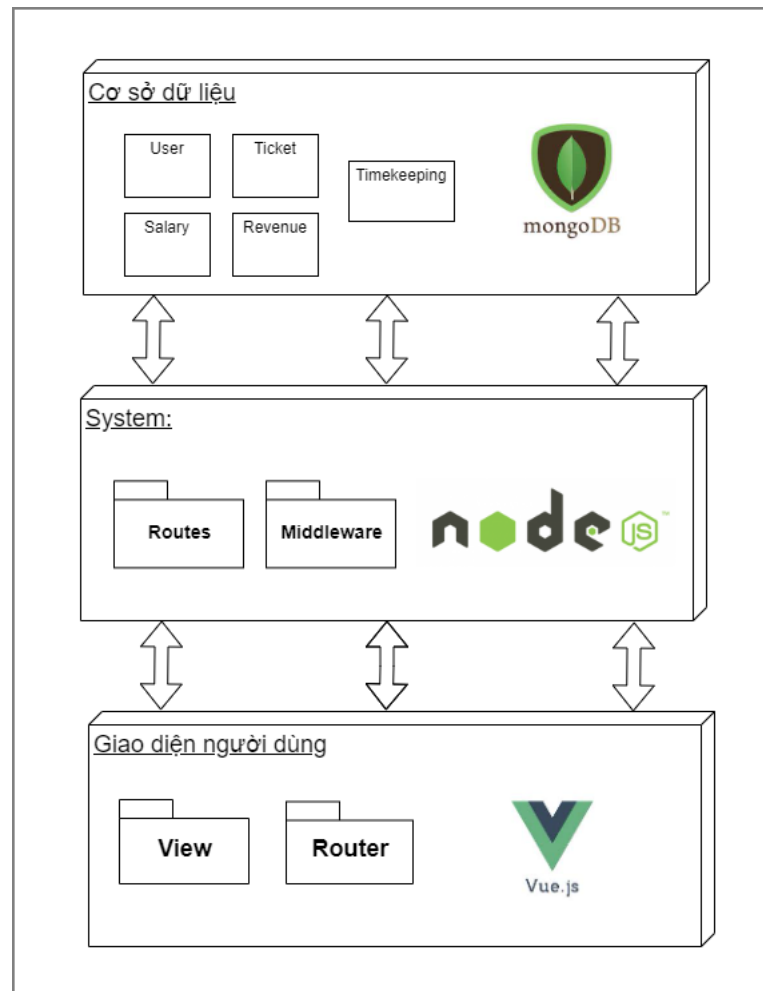
Việc sử dụng MVVM, cho phép hệ thống test có thể được diễn ra mà không cần đến View, hỗ trợ việc kiểm thử trong quá trình xây dựng. Ngoài ra với một bản phân tích thiết kế hệ thống có sẵn, quá trình tạo nên giao diện trở nên dễ dàng nhờ khả năng tái sử dụng. Mô hình MVVM giúp phát triển ứng dụng quy mô nhỏ nhanh, dễ bảo trì và nâng cấp.

4.1.2. Thiết kế tổng quan



Hình 4.1.2. Sơ đồ tổng quan hệ thống

4.1.3. Sơ đồ triển khai hệ thống

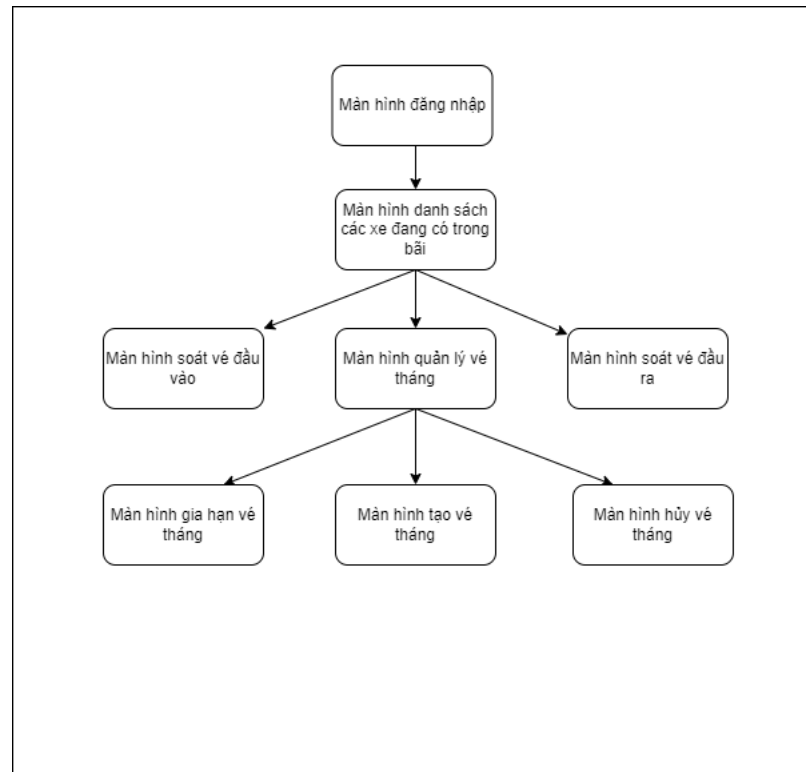


Hình 4.1.3. Sơ đồ triển khai hệ thống

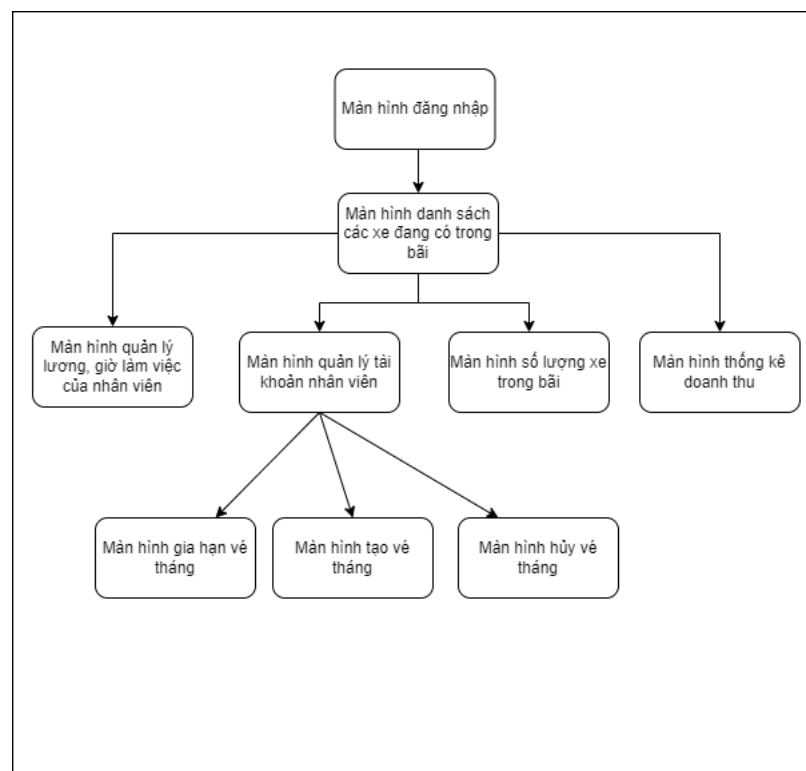
4.2. Thiết kế chi tiết

4.2.1. Thiết kế giao diện

4.2.1.1. Sơ đồ dịch chuyển màn hình

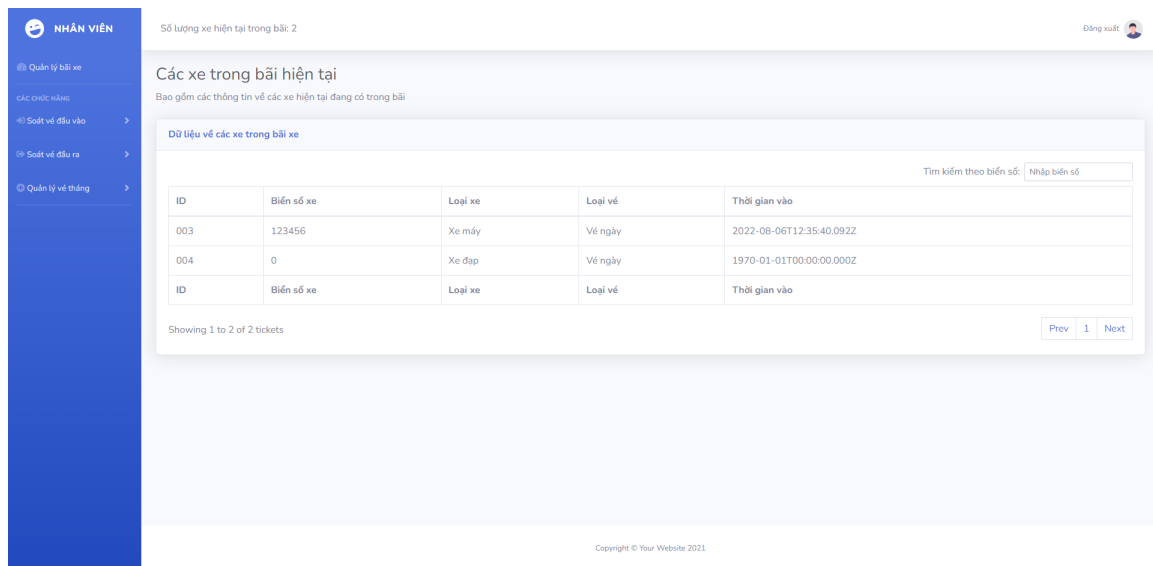


Hình 4.2.1.1a. Sơ đồ dịch chuyển màn hình của nhân viên



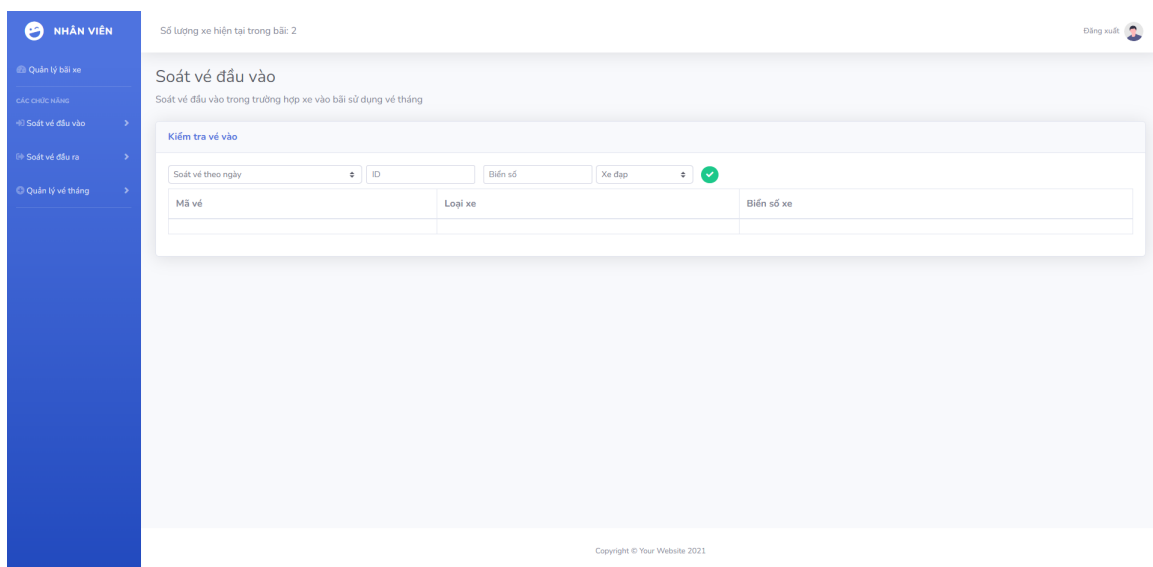
Hình 4.2.1.1b. Sơ đồ dịch chuyển màn hình của quản trị viên

4.2.1.2. Giao diện màn hình chính của nhân viên



Hình 4.2.1.2. Giao diện màn hình chính của nhân viên

4.2.1.3. Giao diện màn hình soát vé vào của nhân viên



Hình 4.2.1.3. Giao diện màn hình soát vé vào của nhân viên

4.2.1.4. Giao diện màn hình soát vé ra của nhân viên

NHÂN VIÊN Số lượng xe hiện tại trong bãi 2 Đăng xuất

Quản lý bãi xe
các chức năng
Soát vé đầu vào
Soát vé đầu ra
Quản lý vé tháng

Soát vé đầu ra

Soát vé đầu ra cho các xe đã vào bãi sử dụng vé ngày hoặc vé tháng

Kiểm tra và tính tiền

001

Loại vé	Thời gian vào	Thời gian ra	Loại xe	Biển số xe	Thành tiền
ngày	2022-08-08T01:45:59.209Z	2022-08-08T01:46:07.216Z	xe máy	29M1	5000 VNĐ

Xác nhận

Copyright © Your Website 2021

Hình 4.2.1.4. Giao diện màn hình soát vé ra của nhân viên

4.2.1.5. Giao diện màn hình tạo vé tháng của nhân viên

Tạo vé tháng

ID

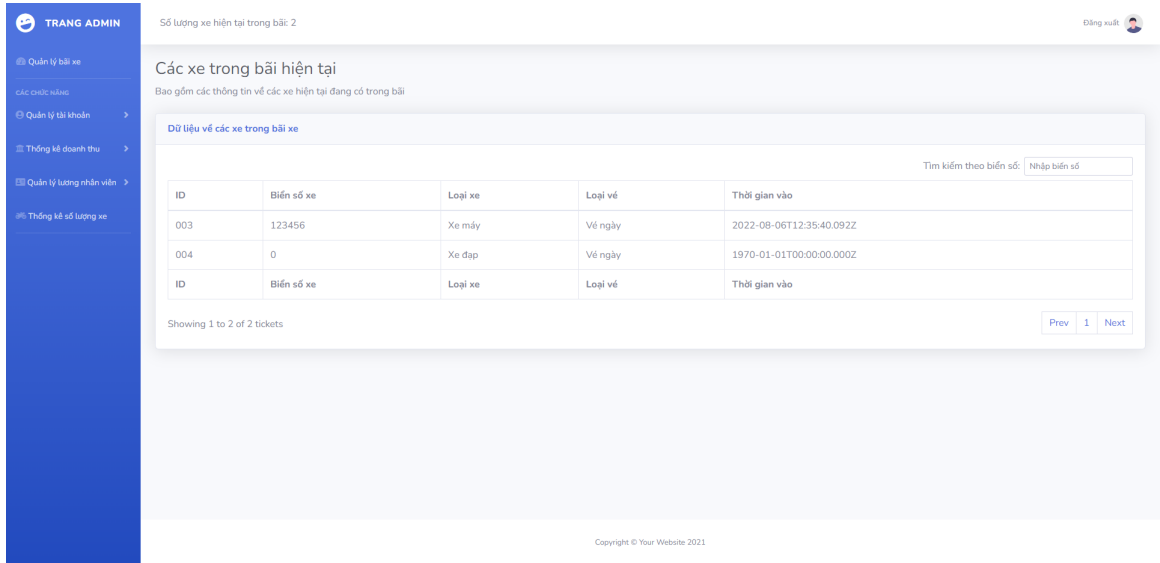
Biển số

Loại xe

Cancel OK

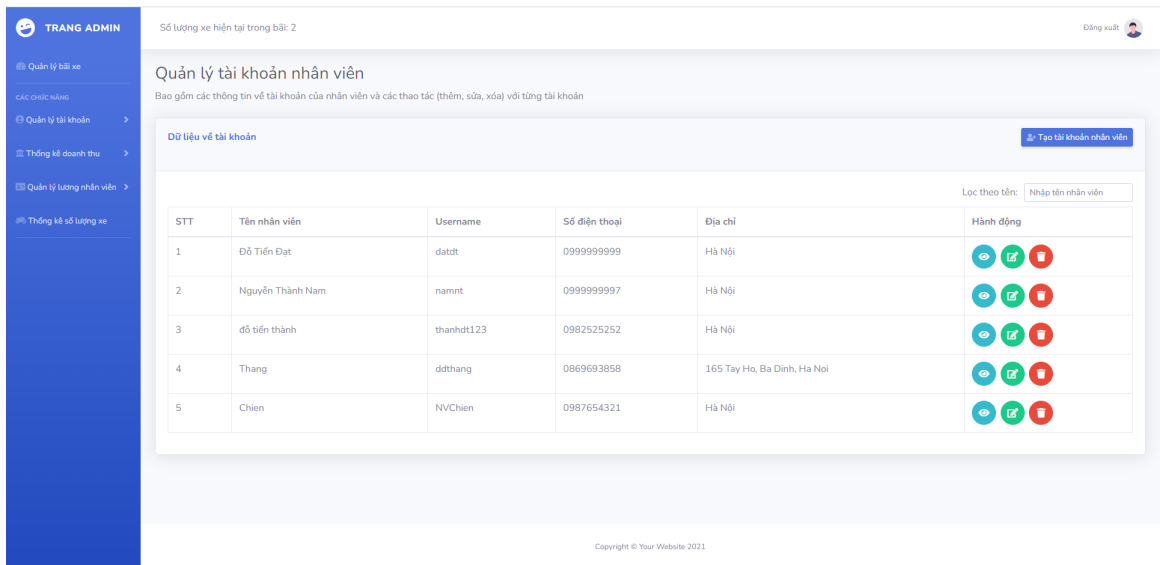
Hình 4.2.1.5. Giao diện màn hình tạo vé tháng của nhân viên

4.2.1.6. Giao diện màn hình chính của quản trị viên



Hình 4.2.1.6. Giao diện màn hình chính của quản trị viên

4.2.1.7. Giao diện màn hình quản lý tài khoản nhân viên



Hình 4.2.1.7. Giao diện màn hình quản lý tài khoản của nhân viên

4.2.1.8. Giao diện màn hình tạo tài khoản nhân viên

Tạo tài khoản nhân viên

×

Tên

Username

Mật khẩu

Mật khẩu cần ít nhất 5 ký tự

SĐT

Số điện thoại chỉ chứa số (bao gồm 10 chữ số)

Địa chỉ

Cancel

OK

Hình 4.2.1.8. Giao diện màn hình tạo tài khoản nhân viên

4.2.1.9. Giao diện màn hình quản lý lương, thời gian làm việc

TRANG ADMIN

Quản lý bán xe

CÁC CHỨC NĂNG

Quản lý tài khoản

Thống kê doanh thu

Quản lý lương nhân viên

Thống kê số lượng xe

Số lượng xe hiện tại trong bãi: 2

Đăng xuất

Quản lý lương nhân viên

Bao gồm các tác vụ xem, tính tiền lương cho nhân viên

Dữ liệu về thời gian làm việc và lương

Lọc theo tháng:

▼

✓

Lọc theo tên:

Nhập tên nhân viên

STT	Tên nhân viên	Tháng	Thời gian làm việc	Hệ số lương	Lương(VND)
1	Đỗ Tiến Đạt	12 / 2021	146.22	30000	4386600.00
2	Tiến Thành	12 / 2021	149.54	30000	4486200.00
3	Lê Tiến Đạt	12 / 2021	0.00	20000	0.00
4	Nguyễn Thành Nam	12 / 2021	0.00	20000	0.00
5	Thành	12 / 2021	0.00	25000	0.00

Copyright © Your Website 2021

Hình 4.2.1.9. Giao diện màn hình quản lý lương, thời gian làm việc

4.2.1.10. Giao diện màn hình thống kê doanh thu

TRANG ADMIN

Quản lý bãi xe

CÁC CHỨC NĂNG

Quản lý tài khoản

Thống kê doanh thu

Quản lý lương nhân viên

Thống kê số lượng xe

Số lượng xe hiện tại trong bãi: 2

Đăng xuất

Thống kê doanh thu

Thống kê doanh thu của bãi gửi xe theo tháng, bao gồm doanh thu của 6 tháng gần nhất.

Dữ liệu về doanh thu

Chọn tháng:

📅

✅

STT	Tháng	Số lượng vé ngày	Số lượng vé tháng	Doanh thu (VND)
1	8 / 2022	2	1	155000
2	7 / 2022	10	4	623000
3	6 / 2022	0	0	0
4	5 / 2022	0	0	0
5	4 / 2022	0	0	0
6	3 / 2022	0	0	0

Biểu đồ doanh thu

Dữ liệu về doanh thu

1.0

0.9

0.8

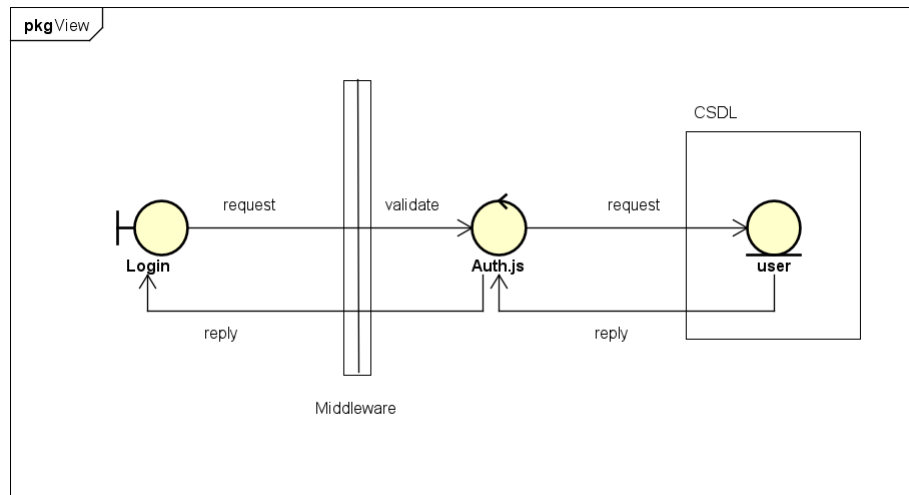
0.7

Hình 4.2.1.10. Giao diện màn hình thống kê doanh thu

4.2.2. Thiết kế module - API

Các module dưới đây được miêu tả theo đúng trình tự dòng của mã nguồn chương trình được cung cấp trong mỗi ảnh

4.2.2.1. Thiết kế module "Đăng nhập"



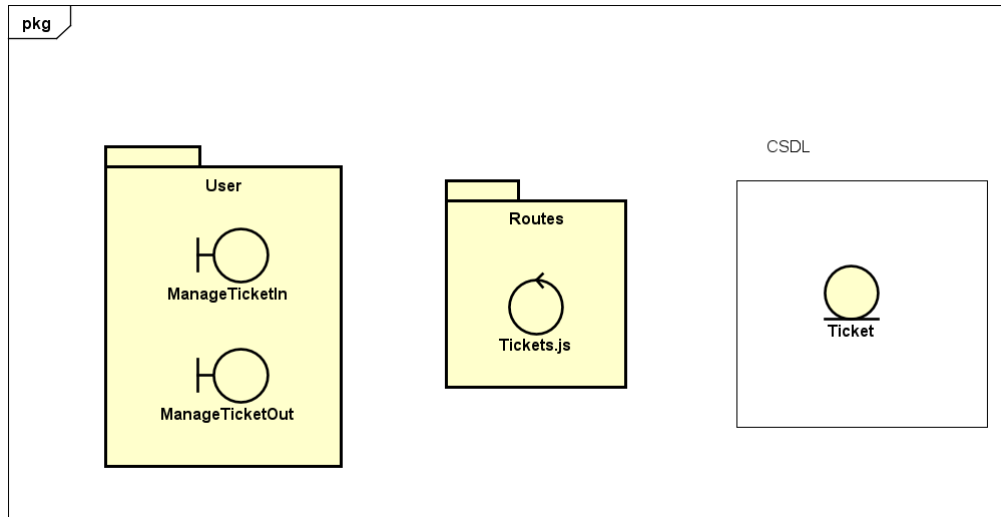
Hình 4.2.2.1a. Các thành phần tham gia module Đăng nhập

```
JS auth.js X
be > routes > JS auth.js > ...
1  const Joi = require('joi');
2  const bcrypt = require('bcrypt');
3  const { User } = require('../models/user')
4  const { Timekeeping } = require('../models/timekeeping')
5  const express = require('express');
6  const router = express.Router();
7
8  router.post('/', async (req, res) => {
9    const { error } = validate(req.body);
10   if (error) return res.status(400).send(error.details[0].message);
11
12   let user = await User.findOne({ username: req.body.username });
13   if (!user) return res.status(400).send("Invalid username or password")
14
15   const validPassword = await bcrypt.compare(req.body.password, user.password)
16   if (!validPassword) return res.status(400).send("Invalid username or password")
17
18   const timekeeping = new Timekeeping({
19     user_id: user._id
20   })
21   await timekeeping.save()
22
23   const token = user.generateAuthToken();
24   user.isActive = true;
25   await user.save();
26
27   res.json({ token: token, timekeeping_id: timekeeping._id, name: user.name, isAdmin: user.isAdmin });
28 })
29
```

Hình 4.2.2.1b. Đoạn mã lập trình xử lý đăng nhập của file auth.js

- Khi người dùng đăng nhập ở màn hình Login, middleware với file Admin.js sẽ kiểm tra tên đăng nhập để xác thực có phải tài khoản quản trị viên không
- Sau đó, file Auth.js sẽ kiểm tra tài khoản mật khẩu bằng cách truy xuất thông tin ở database với hàm findOne. Nếu sai, đưa ra mã lỗi 400. Nếu đúng, biến timekeeping sẽ được khởi tạo để tính thời gian làm việc sau đó lưu trữ dưới dạng mảng.

4.2.2.2. Thiết kế module "Soát vé" của nhân viên



Hình 4.2.2.2a. Các thành phần tham gia module Soát vé

```
JS auth.js X
be > routes > JS auth.js > ...
1  const Joi = require('joi');
2  const bcrypt = require('bcrypt');
3  const { User } = require('../models/user');
4  const { Timekeeping } = require('../models/timekeeping');
5  const express = require('express');
6  const router = express.Router();
7
8  router.post('/', async (req, res) => {
9    const { error } = validate(req.body);
10   if (error) return res.status(400).send(error.details[0].message);
11
12   let user = await User.findOne({ username: req.body.username });
13   if (!user) return res.status(400).send("Invalid username or password")
14
15   const validPassword = await bcrypt.compare(req.body.password, user.password)
16   if (!validPassword) return res.status(400).send("Invalid username or password")
17
18   const timekeeping = new Timekeeping({
19     user_id: user._id
20   })
21   await timekeeping.save()
22
23   const token = user.generateAuthToken();
24   user.isActive = true;
25   await user.save();
26
27   res.json({ token: token, timekeeping_id: timekeeping._id, name: user.name, isAdmin: user.isAdmin });
28 })
29
```

Hình 4.2.2.2b. Đoạn mã lập trình xử lý Soát vé đầu vào của file tickets.js

- Khi nhân viên nhập ID để soát vé vào, hệ thống sẽ kiểm tra ID của vé có hợp lệ không. Vé được sử dụng cũng có thể đã là vé tháng
- Nếu vé hợp lệ thì sẽ bắt đầu nhập thời gian vào khu vực gửi xe và số tiền cần trả
- Khi nhân viên nhập ID để khách gửi xe lấy xe ra ngoài, hệ thống sẽ truy xuất ID của thẻ có trong danh sách hiện tại, đồng thời truy xuất thời gian gửi xe tùy theo loại xe được gửi.

```
// soát vé đầu ra cho vé ngày
router.put('/out/:IDs', auth, async (req, res) => {
  const ticket = await Ticket.findOne({ IDs: req.params.IDs });

  if (!ticket) return res.status(404).send('ID vé không hợp lệ');

  if (!ticket.used) return res.status(400).send('Vé chưa soát vé đầu vào');

  if (ticket.ticket_type == 'ngày') {
    let license_plate = ticket.license_plate
    ticket.license_plate = ""
    ticket.used = false

    await ticket.save()

    const day = new Date(Date.now() + 7 * 60 * 60 * 1000).getDate() - ticket.time_in.getDate()
    // console.log(typeof day)

    const unit_price = ticket.vehicle_type == 'xe máy' ? 5000 : 3000

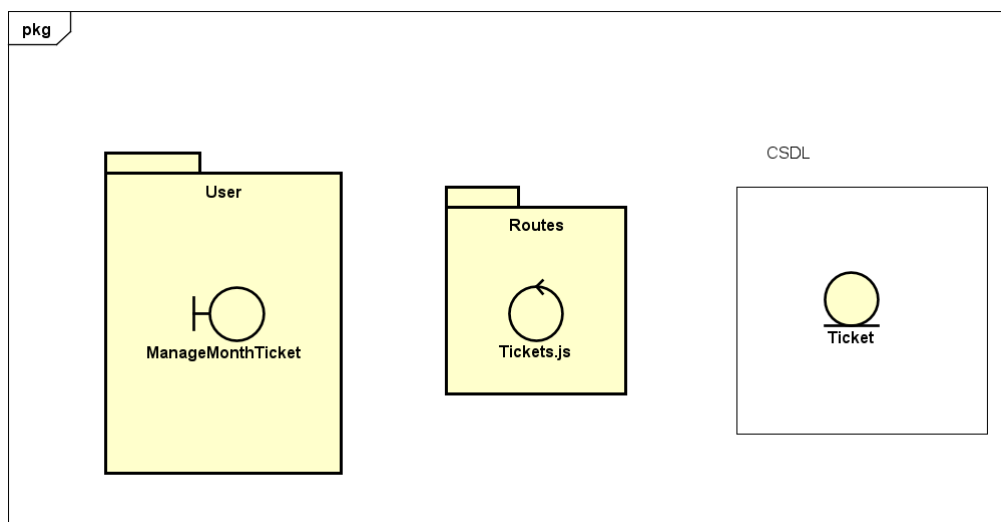
    const price = (day + 1) * unit_price

    const revenue = new Revenue({
      revenue: price,
      ticket_type: "ngày",
      vehicle_type: ticket.vehicle_type
    })
  }
})
```

Hình 4.2.2.2c. Đoạn mã lập trình xử lý Soát vé đầu ra của file tickets.js

- Sau đó, hệ thống sẽ kiểm tra thuộc tính ticket-type và đưa các thuộc tính như used và license-plate (biển số xe) về dạng không có dữ liệu
- Tiếp theo hệ thống sẽ kiểm tra trường vehicle-type để tính giá tiền gửi xe. Biến revenue sẽ được tính toán để có thể thống kê doanh thu

4.2.2.3. Thiết kế module "Quản lý vé tháng" của nhân viên



Hình 4.2.2.3a. Các thành phần tham gia module "Quản lý vé tháng của nhân viên"

```
// tạo vé tháng: kiểm tra IDs nếu vé ngày đang được sử dụng thì báo lỗi, nếu IDs đó là vé tháng thì báo lỗi,
// IDs đúng khi used=false và ticket_type == ngày]
router.put('/create_monthly_ticket/:IDs', async (req, res) => {
  const { error } = validate(req.body);
  if (error) return res.status(400).send(error.details[0].message);

  let ticket = await Ticket.findOne({ IDs: req.params.IDs })
  if (!ticket) return res.status(400).send('IDs is not found')

  if (ticket.used || ticket.ticket_type == 'tháng') return res.status(400).send('Ticket is used')

  ticket.license_plate = req.body.license_plate
  ticket.due_date = +new Date() + 7 * 60 * 60 * 1000 + 30 * 60 * 60 * 24 * 1000
  ticket.vehicle_type = req.body.vehicle_type
  ticket.ticket_type = 'tháng'

  await ticket.save()

  const price = req.body.vehicle_type === 'xe máy' ? 150000 : 50000

  const revenue = new Revenue({
    revenue: price,
    ticket_type: "tháng",
    vehicle_type: ticket.vehicle_type
  })
})
```

Hình 4.2.2.3b. Đoạn mã xử lý quy trình tạo vé tháng

- Khi nhân viên nhập ID của thẻ cứng để tạo vé tháng. ID phải là của vé ngày và chưa được sử dụng. Nếu không trả lại mã lỗi 400
- Khi tạo vé thành công, biến ticket-type sẽ đổi thành vé tháng và nhận loại hình xe gửi đồng thời tính giá tiền cần trả
- Nếu vé hợp lệ, hệ thống sẽ cộng thêm 30 ngày vào ngày tạo vé để tạo ra ngày hết hạn. Quy trình gia hạn vé cũng tương tự, sau khi ra hạn thành công biết due-date sẽ cộng thêm 30 ngày.

```
// gia hạn vé tháng
router.put('/renewal/:IDs', async (req, res) => {
  const ticket = await Ticket.findOne({ IDs: req.params.IDs, ticket_type: "tháng" });

  if (!ticket) return res.status(404).send('The monthly ticket with the given IDs was not found.');
```

```
  if (req.body.renewal) {
    ticket.due_date = +new Date(ticket.due_date) + 7 * 60 * 60 * 1000 + 30 * 60 * 60 * 24 * 1000

    await ticket.save()

    const price = ticket.vehicle_type === 'xe máy' ? 150000 : 50000

    const revenue = new Revenue({
      revenue: price,
      ticket_type: "tháng",
      vehicle_type: ticket.vehicle_type
    })

    await revenue.save()

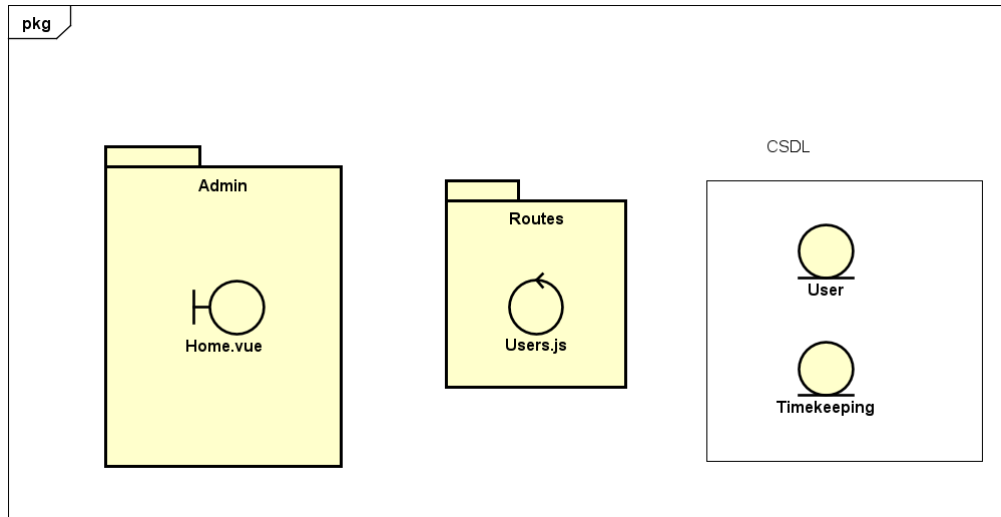
    res.send(ticket)
  }
  else {
    ticket.ticket_type = 'ngày'
    ticket.license_plate = '0'

    await ticket.save()

    res.send(ticket)
  }
})
```

Hình 4.2.2.3c. Đoạn mã xử lý quy trình gia hạn vé tháng

4.2.2.4. Thiết kế module "Đăng xuất"



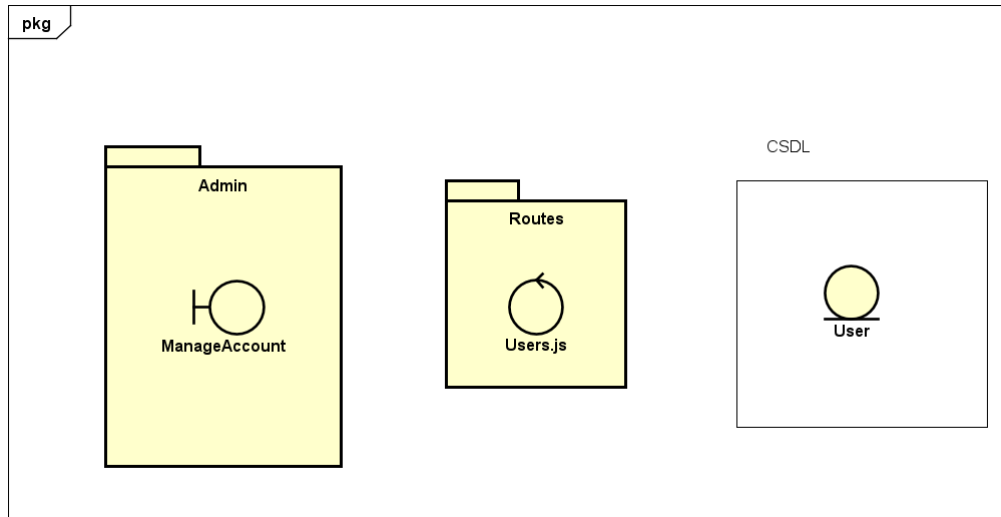
Hình 4.2.2.4a. Các thành phần tham gia module "Đăng xuất"

```
68
69 // đăng xuất
70 router.put('/logout/:timekeeping_id', auth, async (req, res) => {
71   console.log(1111)
72   let user = await User.findById(req.user._id)
73   user.isActive = false
74   await user.save()
75
76   let timekeeping = await Timekeeping.findById(req.params.timekeeping_id)
77   timekeeping.end_time = +new Date() + 7 * 60 * 60 * 1000,
78   console.log(timekeeping.end_time)
79
80   await timekeeping.save()
81   res.send(user)
82 })
83
```

Hình 4.2.2.4b. Đoạn mã lập trình xử lý chức năng "Đăng xuất"

- Khi nhân viên hoặc quản trị viên chọn nút đăng xuất, hệ thống sẽ đổi trường `isActive` của tác nhân sang `false` - không hoạt động. Sau đó thuộc tính `end-time` của `timekeeping` sẽ ghi nhận thời gian kết thúc phiên hoạt động để phục vụ việc tính lương ở mục 4.2.2.6

4.2.2.5. Thiết kế module "Quản lý tài khoản nhân viên" của quản trị viên



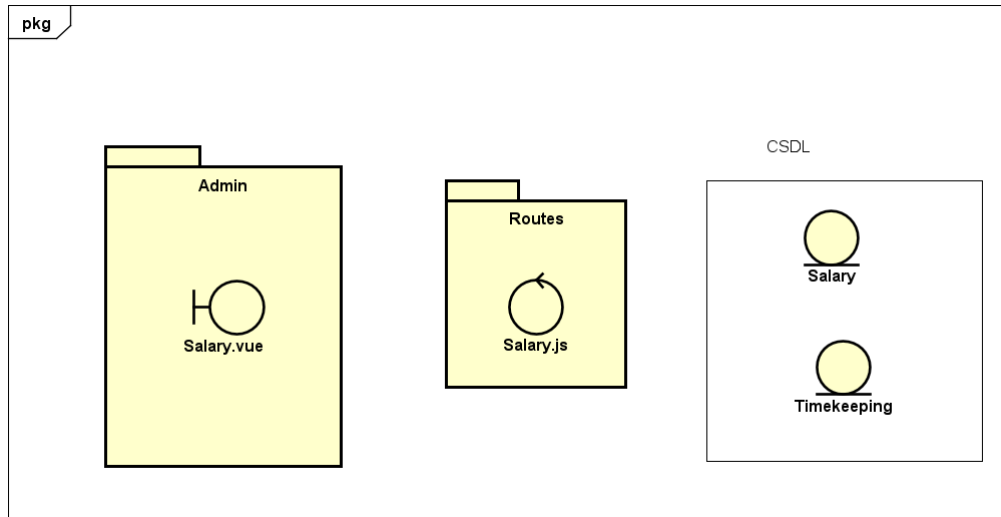
Hình 4.2.2.5a. Các thành phần tham gia module "Quản lý tài khoản nhân viên"

```
JS tickets.js M JS users.js X
be > routes > JS users.js > ...
9
10 // lấy toàn bộ danh sách nhân viên
11
12 router.get('/', [auth, admin], async (req, res) => {
13   const users = await User.find({ isAdmin: false }).select('-password -isAdmin')
14   res.send(users)
15 })
16
17 // tạo nhân viên mới
18 router.post('/create', [auth, admin], async (req, res) => {
19   const { error } = validate(req.body);
20   if (error) return res.status(400).send(error.details[0].message)
21
22   let user = await User.findOne({ username: req.body.username })
23   if (user) return res.status(400).send("username already exists")
24
25   user = new User(_.pick(req.body, ['name', 'username', 'password', 'phone_number', 'address']))
26
27   const salt = await bcrypt.genSalt(10);
28   user.password = await bcrypt.hash(user.password, salt);
29
30   await user.save();
31   const token = user.generateAuthToken();
32
33   res.header('x-auth-token', token).send(_.pick(user, ['_id', 'name', 'username']))
34 })
35
```

Hình 4.2.2.5b. Đoạn mã xử lý quy trình tạo tài khoản nhân viên

- Khi quản trị viên tạo tài khoản mới, hệ thống sẽ gửi request qua file JSON với các trường thông tin cần thiết. Trường mật khẩu sẽ được xử lý qua hàm băm trước khi lưu vào cơ sở dữ liệu
- Tương tự với quy trình chỉnh sửa tài khoản, các trường thông tin cũng sẽ được đưa qua tham số/param trong request để thay đổi.

4.2.2.6. Thiết kế module "Quản lý lương, thời gian làm việc của nhân viên"



Hình 4.2.2.6a. Các thành phần tham gia module "Quản lý lương, thời gian làm việc của nhân viên"

```
const timekeepings = await Timekeeping
  .find()
  .populate({
    path: 'user_id',
    match: {
      name: new RegExp(`.*${req.query.name}.*`, 'i')
    },
    select: 'name coefficients_salary -_id'
  })
// console.log(timekeepings)
let time = 0
let coefficients_salary = 0
let name = ''
for (let i in timekeepings) {
  // xét xem có đúng tên hay không
  if (timekeepings[i].user_id) {
    name = timekeepings[i].user_id.name
    coefficients_salary = timekeepings[i].user_id.coefficients_salary
    if (new Date(year, month, 1, 7) <= timekeepings[i].start_time &&
      new Date(year, month, 30, 7) > timekeepings[i].start_time)
      time += timekeepings[i].end_time - timekeepings[i].start_time
  }
}

time = time / 1000 / 60 / 60
salary = time * coefficients_salary
```

Hình 4.2.2.6b. Đoạn mã xử lý quy trình quản lý lương, thời gian làm việc của nhân viên

- Khi hệ thống truy xuất lương của nhân viên, kiểm tra tên nhân viên tương ứng. Sau đó, trích xuất các mảng time-keeping (mục 4.2.2.2) - thời gian làm việc. Biến time sẽ tính tổng các khoảng thời gian làm việc và nhân với lương theo giờ
- Sau đó đoạn chương trình sẽ lưu kết quả tính lương vào mảng theo dạng bảng và khi Admin yêu cầu truy xuất thì sẽ hiển thị kết quả. Trước khi lưu, hệ thống sẽ kiểm tra tháng tương ứng với kết quả, nếu tháng sử dụng là tháng hiện tại thì sẽ không lưu

```

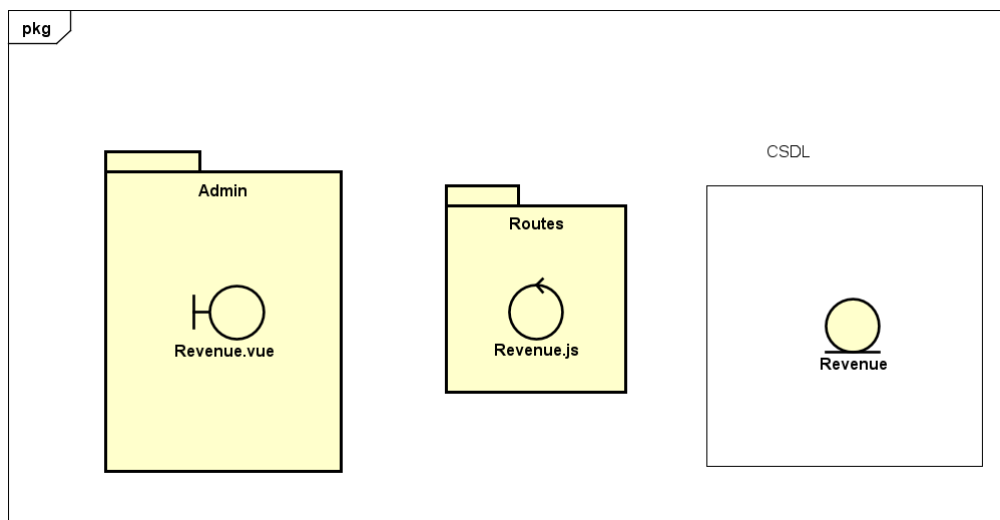
let nowMonth = new Date(new Date().getFullYear(), new Date().getMonth())
let searchMonth = new Date(year, month - 1)

let result = []
for (let key in users) {
  if (users[key].isAdmin) {
    let user = users[key]
    // console.log(users)
    result.push({ name: user.name, time: 0, salary: 0, coefficients_salary: user.coefficients_salary, month: month, year: year })
    for (let ind_tk in timekeepings) {
      // console.log(timekeepings[ind_tk])
      if (timekeepings[ind_tk].user_id) {
        if (String(timekeepings[ind_tk].user_id_id) == String(user._id)) {
          result[result.length - 1].time += timekeepings[ind_tk].end_time - timekeepings[ind_tk].start_time
        }
      }
    }
    result[result.length - 1].time = (result[result.length - 1].time / 1000 / 60 / 60).toFixed(2)
    result[result.length - 1].salary = (result[result.length - 1].time * user.coefficients_salary).toFixed(2)
    // lưu vào db, ko phải là tháng hiện tại thì mới lưu
    if (searchMonth < nowMonth) {
      salary = new Salary(result[result.length - 1])
      await salary.save()
    }
  }
}

```

Hình 4.2.2.6c. Đoạn mã xử lý kết quả tính lương của nhân viên

4.2.2.7. Thiết kế module "Thống kê doanh thu và số lượng xe"



Hình 4.2.2.7a. Các thành phần tham gia module "Thống kê doanh thu và số lượng xe"

```

JS tickets.js M JS revenues.js X
be > routes > JS revenues.js > ...
6
7 // tính doanh thu nhập vào ngày tháng
8 router.get('/', async (req, res) => {
9   if (!req.query.day && !req.query.month) {
10    // return res.status(400).send('No day, month')
11    const result = []
12    for (let month of [...Array(6).keys()]) {
13      let firstDate = new Date(new Date().getFullYear(), new Date().getMonth(), 1, 7)
14      firstDate.setMonth(firstDate.getMonth() - month)
15      let endDate = new Date(new Date().getFullYear(), new Date().getMonth() + 1, 1, 7)
16      endDate.setMonth(endDate.getMonth() - month)
17      let revenues = await Revenue.find({
18        time: {
19          $gte: firstDate,
20          $lt: endDate,
21        }
22      });
23
24      let revenue = 0
25      let day_ticket = 0
26      let month_ticket = 0
27      for (const i in revenues) {
28        revenue = revenue + revenues[i].revenue
29        if (revenues[i].ticket_type == "ngày") day_ticket += 1
30        else month_ticket += 1
31      }
32      result.push({ revenue: revenue, month: firstDate.getMonth() + 1, year: firstDate.getFullYear(), day_ticket: day_ticket, month_ticket: month_ticket })
33    }
34  }
35  return res.send(result)
36

```

Hình 4.2.2.7b. Đoạn mã xử lý thống kê doanh thu

- Khi người dùng chọn ngày tháng để hiển thị thống kê doanh thu, hệ thống sẽ truy xuất ngày bắt đầu và ngày kết thúc của tháng đó. Sau đó, các doanh thu/revenue sẽ được thu thập trong tháng đó từ mục 4.2.2.3 mỗi lần nhân viên soát vé ra.
- Hệ thống cũng kiểm tra số lượng mỗi loại vé được soát và chia ra số lượng vé ngày và vé tháng tính theo mỗi ngày.

```

72  });
73
74  // thống kê số lượng vé tháng vé ngày
75  router.get('/statistic', async (req, res) => {
76    // if (!req.query.day && !req.query.month) {
77    //   return res.status(400).send('No day, month')
78    // }
79
80    let month = req.query.month ? parseInt(req.query.month) - 1 : (new Date().getMonth())
81    let year = req.query.year ? parseInt(req.query.year) : (new Date().getFullYear())
82    let day = req.query.day ? parseInt(req.query.day) : null
83
84    if (day) {
85      if (new Date(year, month, day) > new Date()) {
86        console.log(new Date(year, month, day));
87        console.log(new Date());
88
89        return res.status(400).send('Hiện tại mới chỉ là ngày ${new Date().getDate()}')
90      }
91      revenues = await Revenue.find({
92        time: {
93          $gte: new Date(year, month, day, 7),
94          $lt: new Date(year, month, day + 1, 7),
95        }
96      });
97    }
98    else {
99      if (new Date(year, month) > new Date()) return res.status(400).send('Hiện tại mới chỉ là tháng ${new Date().getMonth() + 1}')
100      revenues = await Revenue.find({
101        time: {
102          $gte: new Date(year, month, 1, 7),
103          $lt: new Date(year, month, 30, 7),
104        }
105      });
106    }
107
108    let day_ticket = 0
109    let month_ticket = 0
110    for (const i in revenues) {
111      if (revenues[i].ticket_type == "ngày") day_ticket += 1

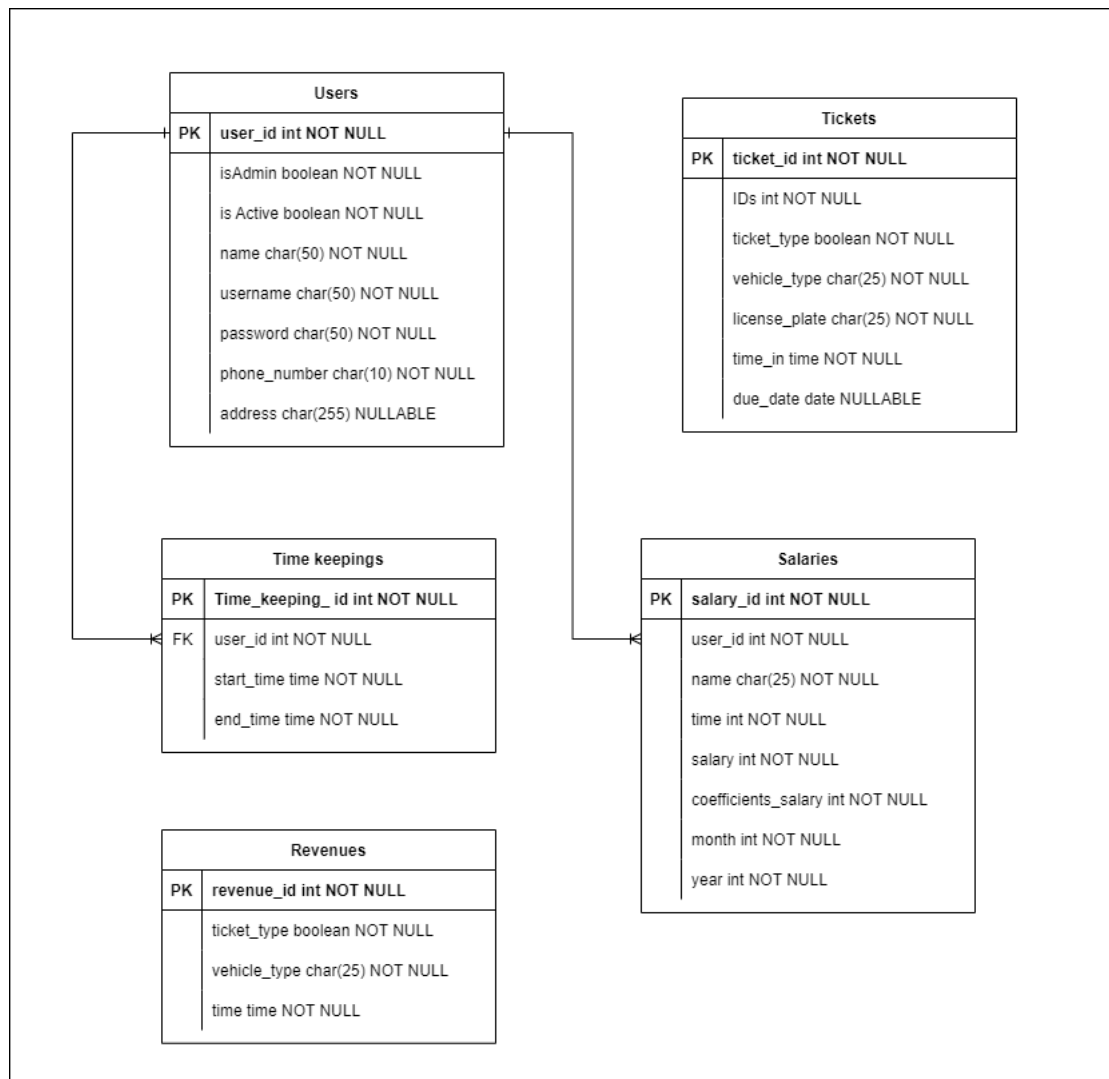
```

Hình 4.2.2.7c. Đoạn mã xử lý thống kê số lượng xe

- Sau khi người dùng chọn ngày để kiểm tra số lượng xe. Nếu ngày đó là ngày chưa diễn ra so với thời gian hiện tại, hệ thống sẽ báo lỗi. Nếu ngày tháng người dùng nhập vào hợp lệ, hệ thống sẽ chia ra 2 trường vé ngày và vé tháng để quản trị viên có thể thể đưa ra đánh giá

4.2.3. Thiết kế cơ sở dữ liệu

4.2.3.1. Sơ đồ liên kết thực thể



Hình 4.2.3.1 Sơ đồ các bảng trong cơ sở dữ liệu

4.2.3.2. Chi tiết các bảng trong cơ sở dữ liệu

Bảng User:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc	Khuôn dạng	Ghi chú
user-id	int		Khóa chính	Số nguyên	
isAdmin	boolean			Văn bản	
isActive	boolean			Văn bản	
name	char	50 ký tự		Văn bản	
username	char	50 ký tự		Văn bản	
password	char	50 ký tự		Văn bản	
phone-number	char	10 ký tự		Văn bản	
address	char	255 ký tự		Văn bản	

Bảng Tickets:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc	Khuôn dạng	Ghi chú
ticket-id	int		Khóa chính	Số nguyên	
IDs	int			Số nguyên	
used	boolean			Văn bản	
ticket-type	char	25 ký tự		Văn bản	
vehicle-type	char	25 ký tự		Văn bản	
time-in	time			Thời gian	
due-date	date			Ngày	

Bảng Salaries:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc	Khuôn dạng	Ghi chú
salary-id	int		Khóa chính	Số nguyên	
user-id	int		Khóa ngoài	Số nguyên	
name	char	25 ký tự		Văn bản	
time	int	25 ký tự		Số nguyên	
salary	int			Số nguyên	
coefficients-salary	int			Số nguyên	
month	int			Số nguyên	
year	int			Số nguyên	

Bảng Time-keepings:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc	Khuôn dạng	Ghi chú
time-keeping-id	int		Khóa chính	Số nguyên	
user-id	int		Khóa ngoài	Số nguyên	
start-time	time			Thời gian	
end-time	time			Thời gian	

Bảng Revenues:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc	Khuôn dạng	Ghi chú
revenue-id	int		Khóa chính	Số nguyên	
ticket-type	char			Văn bản	
vehicle-type	char	25 ký tự		Văn bản	
time	time			Thời gian	

4.3. Xây dựng ứng dụng

4.3.1. Công cụ sử dụng

Mục đích	Công cụ sử dụng	Địa chỉ URL
IDE lập trình	Visual Studio Code	https://code.visualstudio.com
Ngôn ngữ lập trình	JavaScript	
Framework	VueJS	https://vuejs.org/
Framework	Bootstrap	https://getbootstrap.com/
Cơ sở dữ liệu	MongoDB	https://www.mongodb.com/

4.3.2. Kết quả đạt được

Hệ thống sau khi được xây dựng có thể sử dụng các chức năng:

- Hiển thị danh sách các xe đang có trong khu vực quản lý
- Soát vé đầu vào và đầu ra
- Tạo, gia hạn và huỷ vé tháng
- Hiển thị danh sách tài khoản nhân viên
- Quản lý nhân viên: Tạo, sửa, xoá tài khoản nhân viên
- Thống kê doanh thu
- Thống kê số lượng xe và số lượng vé đã soát
- Thống kê lương và thời gian làm việc của nhân viên

4.3.3. Kiểm thử

4.3.3.1. Kiểm thử tính năng "Hiển thị danh sách các xe trong khu vực quản lý"

STT	Đầu vào	Đầu ra mong muốn	Kết quả
1	Nhân viên đăng nhập thành công	Hiển thị màn hình danh sách các xe trong khu vực quản lý	Đạt
2	Quản trị viên đăng nhập thành công	Hiển thị màn hình danh sách các xe trong khu vực quản lý	Đạt

4.3.3.2. Kiểm thử tính năng "Soát vé đầu vào"

STT	Đầu vào	Đầu ra mong muốn	Kết quả
1	Vé ngày ID = '001', license-plate= '29M1-12345', vehicle-type = 'xe máy'	Thông báo soát vé thành công	Đạt

4.3.3.3. Kiểm thử tính năng "Soát vé đầu ra"

STT	Đầu vào	Đầu ra mong muốn	Kết quả
1	ID = '001' và thẻ đã được soát vé đầu vào	Thông báo soát vé thành công	Đạt
2	ID = '001' và thẻ chưa được soát vé đầu vào	Thông báo vé chưa được soát vé đầu vào	Đạt
3	ID = '002' và thẻ là vé tháng	Thông báo vé là vé tháng	Đạt

4.3.3.4. Kiểm thử tính năng "Tạo vé tháng"

STT	Đầu vào	Đầu ra mong muốn	Kết quả
1	ID = '001', license-plate= '29M1-12345', vehicle-type = 'xe máy'	Thông báo tạo vé tháng thành công	Đạt
2	ID = '001', license-plate= '29M1-12345', vehicle-type = 'xe máy' nhưng thẻ đã là vé tháng	Thông báo thẻ đã là vé tháng	Đạt

4.3.3.5. Kiểm thử tính năng "Gia hạn vé tháng"

STT	Đầu vào	Đầu ra mong muốn	Kết quả
1	ID = '001'	Hiển thị thông báo gia hạn thành công	Đạt
2	ID = '001' với điều kiện thẻ đã được ra hạn từ trước	Thông báo thẻ đã được gia hạn từ trước	Đạt
3	ID = '002' với điều kiện thẻ không phải vé tháng	Thông báo thẻ không phải vé tháng	Đạt

4.3.3.6. Kiểm thử tính năng "Huỷ vé tháng"

1	ID = '001'	Hiển thị thông báo huỷ vé tháng thành công	Đạt
2	ID = '001' với điều kiện thẻ không phải vé tháng	Thông báo thẻ không phải vé tháng	Đạt

4.3.3.7. Kiểm thử tính năng "Tạo tài khoản nhân viên"

Kiểm thử tính năng "Tạo tài khoản nhân viên" và "Chỉnh sửa tài khoản nhân viên" tương đương nhau vì giống nhau phần đầu vào

STT	Đầu vào	Đầu ra mong muốn	Kết quả
1	Ten = 'Huong', username = 'huongkm', password = '123456', phone-number = '0123456789', address = '165 Doi can'	Thông báo tạo tài khoản nhân viên thành công	Đạt
2	Các trường như mục 1 nhưng vi phạm điều kiện ràng buộc	Hiển thị cảnh báo màu đỏ ở dưới các trường	Đạt

4.3.3.8. Kiểm thử tính năng "Thống kê doanh thu"

STT	Đầu vào	Đầu ra mong muốn	Kết quả
1	08-2022	Hiển thị màn hình doanh thu trong tháng 8/2022	Đạt
2	10-2022	Hiển thị thông báo hiện giờ mới chỉ là tháng (hiện tại) năm (hiện tại)	Đạt

4.3.3.9. Kiểm thử tính năng "Quản lý lương, thời gian làm việc của nhân viên"

STT	Đầu vào	Đầu ra mong muốn	Kết quả
1	08-2022	Hiển thị màn hình danh sách lương và tên nhân viên trong tháng 8/2022	Đạt
2	10-2022	Hiển thị thông báo hiện giờ mới chỉ là tháng (hiện tại) năm (hiện tại)	Đạt

4.3.3.10. Kiểm thử tính năng "Thống kê số lượng xe, vé tháng và vé ngày đã soát"

STT	Đầu vào	Đầu ra mong muốn	Kết quả
1	07-08-2022	Hiển thị màn hình số lượng xe, vé tháng, vé ngày trong ngày 07/08/2022	Đạt
2	01-10-2022	Hiển thị thông báo hiện giờ mới chỉ là ngày (hiện tại) tháng (hiện tại) năm (hiện tại)	Đạt

Chương 5

Đánh giá và đưa ra chương trình chạy thử

5.1. Các bước để khởi động chương trình

Bước 1: Cài đặt MongoDB và MongoDB Compass để truy cập cơ sở dữ liệu

Bước 2: Kiểm tra kết nối mạng, sau đó mở MongoDB Compass và kết nối đến cơ sở dữ liệu tạm thời tại link:

mongodb+srv://daily:12345@parking-system.viixj.mongodb.net/parking

Bước 3: Download bản zip hoặc clone dự án tại:

<https://github.com/ddthang2611/parking-system>

Bước 4: Mở 2 thư mục fe và be

Bước 5: Trong thư mục be, mở file .env bằng notepad hoặc bất cứ ứng dụng nào hỗ trợ. Sửa đường link mongodbURL tương ứng với đường link ở phần cơ sở dữ liệu tạm thời

Bước 5: Mở cửa sổ CMD hoặc cửa sổ terminal tại 2 thư mục trên

Bước 6: Chạy lệnh **npm install** trên 2 cửa sổ CMD/terminal

Bước 7: Sau khi các gói cài đặt đã hoàn tất

- Ở cửa sổ be: chạy lệnh **node index.js**

- Ở cửa sổ fe: chạy lệnh **npm run serve**

5.2. Đánh giá hiệu suất và đưa ra những điều cần khắc phục trong quá trình phát triển

- Về ưu điểm và thuận lợi:

- Hệ thống xử lý đúng nghiệp vụ, đúng tốc độ so với kỳ vọng của nhà phát triển. Các chức năng quản trị quản lý hoàn thành đúng nhiệm vụ.
- Giao diện sử dụng nhanh gọn. Các nút dễ hiểu, dễ thao tác, không xảy ra lỗi khi nhập nhiều dữ liệu.
- Các tài liệu để tham khảo, học hỏi có sẵn trên Internet cùng sự chỉ dẫn tận tình của thầy hướng dẫn giúp em yên tâm thực hiện đồ án.

- Về nhược điểm và bất lợi:

- Chưa tích hợp được phần cứng vào việc soát vé, hệ thống hiện tại mới chỉ xử lý các nghiệp vụ của phần mềm quản lý.
- Giao diện khá đơn giản, cần có sự bổ sung để tránh gây nhầm lẫn sau thời gian dài sử dụng.

Chương 6

Kết luận và hướng phát triển

6.1. Kết luận

Sản phẩm của em lần này có thể không mới trong số các ứng dụng đã có sẵn trên thị trường. Tuy nhiên, hệ thống lần này có thể nói là sản phẩm có tiềm năng phát triển khi vừa tích hợp được quy trình của nhân viên soát vé cũng như là quản lý khu vực. Hệ thống cũng có thể tích hợp, lắp ghép với hệ thống quản lý có sẵn của tòa nhà và khu đô thị. Đối với bản thân em cũng là một sự nỗ lực lớn về việc học hỏi kỹ năng lập trình cũng như phân tích thiết kế hệ thống.

Mặt khác, sự thành thạo trong việc vận dụng công nghệ chưa được cao. Sản phẩm chỉ dừng lại ở mức thỏa mãn quy trình quản lý bên trong phần mềm cũng như chưa thể tích hợp phần cứng như máy quét biển số hay quét ID thẻ nên các thao tác trong phần mềm đều bằng tay. Đây là phương pháp tạm thời khi chưa thể nâng cấp hệ thống, em sẽ cố gắng để khắc phục điểm yếu này.

6.2. Hướng phát triển

Sản phẩm sẽ được em tiếp tục duy trì và phát triển cũng như là bảo trì, tinh chỉnh những phần dư thừa. Như đã đề cập ở phần trên, mục tiêu hướng đến sẽ là phần cứng tích hợp trí tuệ nhân tạo nhận diện văn bản. Ngoài ra giao diện hệ thống cũng cần phải được nâng cấp và tạo ra sự khác biệt giữa các phân hệ trong hệ thống nhằm hướng đến việc nâng cao trải nghiệm cho người dùng.

TÀI LIỆU THAM KHẢO

- www.hocdauthau.com/mongodb-la-gi-khi-nao-nen-dung-mongodb/
- www.viblo.asia/p/tim-hieu-ve-mo-hinh-mvvm-maGK7vW95j2
- www.hoidapthutuchaiquan.vn/vue-js-la-gi/
- www.bizfly.vn/techblog/vuejs-la-gi.html
- www.learnlatex.org/vi/more-08
- www.stackoverflow.com
- www.getbootstrap.com
- <https://vi.wikipedia.org/>
- <https://thuonggiaonline.vn/sau-10-nam-luong-can-ho-chung-cu-tai-ha-noi-tang-gap-4-lan-28966.htm>