## TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

# ĐỒ ÁN TỐT NGHIỆP

# Xây dựng hệ thống quản lý vận chuyển hành khách

#### NGUYỄN ĐÌNH DƯƠNG

duong.nd173071@sis.hust.edu.vn

Ngành Công nghệ thông tin Chuyên ngành Công nghệ thông tin

Giảng viên hướng dẫn:	PGS.TS. Trần Đình Khang	
	1 02.12. 11 2 12	

**Bộ môn:** Hệ thống thông tin

Viện: Công nghệ thông tin – Truyền thông

## Lời cam kết

Họ và tên sinh viên: Nguyễn Đình Dương

Điện thoại liên lạc: 0352748300 Email: duong.nd173071@sis.hust.edu.vn

Lớp: CNTT.09 - K62 Hệ đào tạo: Cử nhân kỹ thuật

Tôi – Nguyễn Đình Dương – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của PGS.TS Trần Đình Khang. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những thông tin tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm nếu có những sai phạm vi phạm quy chế của nhà trường.

Hà Nội, ngày 09 tháng 02 năm 2022

Tác giả ĐATN

Nguyễn Đình Dương

#### Lời cảm ơn

Bốn năm đại học trôi qua một cách thật nhanh, tuy không dài những cũng là một chặng đường đủ để bản thân em có những trải nghiệm mới, được tiếp xúc với những con người mới, tư duy mới, và hơn cả là những bài học kinh nghiệm trong cuộc sống.

Qua Đồ án tốt nghiệp này, em muốn gửi lời cảm ơn tới tất cả mọi người, tới các thầy cô, bạn bè ở Viện Công nghệ thông tin và Truyền thông – nay là Trường Công nghệ thông tin và Truyền thông – Đại học Bách khoa Hà Nội đã chỉ bảo cũng như cho em những bài học quý giá trong hơn bốn năm qua, em xin gửi lời cảm ơn đặc biệt tới gia đình - những người đã luôn ở bên, đồng hành cùng em qua mọi thử thách, mọi buồn vui. Cuối cùng, em xin gửi lời cảm ơn chân thành nhất tới thầy Trần Đình Khang đã giúp em trong quá trình thực hiện đồ án, thầy luôn có những hướng dẫn, chỉ dạy kịp thời với những thắc mắc, khó khăn mà em đã gặp phải.

Em xin chân thành cảm ơn.

## Tóm tắt

Cuộc sống hiện nay ngày càng hiện đại, internet cũng như các thiết bị thông minh được ứng dụng trong rất nhiều lĩnh vực đời sống. Bên cạnh đó, dân số ngày càng đông, chính vì thế nhu cầu đi lại của người dân cũng ngày càng lớn. Tuy nhiên hiện nay đa phần người dân khi muốn di chuyển đến những tỉnh thành khác cũng như các công ty vận tải vẫn giữ hệ thống, phương pháp cũ từ ngày trước là khách sẽ gọi đặt vé thông qua số điện thoại nhà xe/phòng vé hoặc ra tận bến xe để mua vé. Chính từ những điều thực tế đó đã đặt ra cần có một hệ thống giúp các công ty vận tải quản lý hành khách, chuyến đi một cách thuận tiện hơn cũng như giúp hành khách có thể đặt vé thông qua chiếc điện thoại thông minh của mình.

Hệ thống quản lý dành cho các công ty vận tải được xây dựng trên nền tảng web, các nhân viên công ty có thể truy cập bất cứ lúc nào. Hệ thống đặt vé cho hành khách được xây dựng trên nền tảng Android và trong tương lai sẽ mở rộng thành đa nền tảng, giúp người dùng có những trải nghiệm tốt nhất, thuận tiện nhất.

# Mục lục

Lời cam kết	ii
Lời cảm ơn Error! Bo	okmark not defined.
Mục lục	iv
Danh mục hình vẽ	vii
Chương 1 Giới thiệu đề tài	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi đề tài	1
1.3 Định hướng giải pháp	2
1.4 Bố cục đồ án	2
Chương 2 Khảo sát và phân tích yêu cầu	3
2.1 Khảo sát hiện trạng	3
2.2 Tổng quan chức năng	5
2.2.1 Biểu đồ use case tổng quan	5
2.2.2 Biểu đồ use case phân rã	6
2.2.3 Quy trình nghiệp vụ	12
2.3 Đặc tả chức năng	15
2.3.1 Đặc tả use case Đăng nhập bằng số điện thoại	15
2.3.2 Đặc tả use case Đặt vé	16
2.3.3 Đặc tả use case Thực hiện chuyến đi	18
2.3.4 Đặc tả use case Tạo chuyến đi	19
2.4 Giải pháp quản lý chuyến đi cho các công ty vận tải	23
2.4.1 Đặt vấn đề	23

	2.4.2 Giải pháp đưa ra	. 23
2.5 Y	êu cầu phi chức năng	. 25
	2.5.1 Yêu cầu về bảo mật	. 25
	2.5.2 Yêu cầu giao diện	. 26
	2.5.3 Yêu cầu khác	. 26
Chươ	ơng 3 Công nghệ sử dụng	. 27
3.1 Se	erver	. 27
	3.1.1 NodeJS	. 27
	3.1.2 MySQL	. 27
3.2 W	Vebsite	. 28
	3.2.1 VueJS	. 28
3.3 A	pp mobile	. 28
	3.3.1 Java	. 28
	3.3.2 RESTful API	. 29
	3.3.3 Firebase	. 29
Chươ	ơng 4 Phát triển và triển khai ứng dụng	31
4.1 Tl	hiết kế kiến trúc	. 31
	4.1.1 Lựa chọn kiến trúc phần mềm	. 31
	4.1.2 Thiết kế tổng quan	. 32
	4.1.3 Thiết kế chi tiết gói	. 33
4.2 Tl	hiết kế chi tiết	. 35
	4.2.1 Thiết kế giao diện	. 35
	4.2.2 Thiết kế lớp	. 37
	4.2.3 Thiết kế cơ sở dữ liệu	. 37
4.3 X	ây dựng ứng dụng	. 42
	4.3.1 Thư viện và công cụ sử dụng	. 42
	4.3.2 Kết quả đạt được	. 43

4.3.3 Minh hoạ các chức năng chính	44
4.4 Kiểm thử	48
4.4.1 Kiểm thử tương thích	48
4.4.2 Kiểm thử chức năng	48
4.5 Triển khai	49
Chương 5 Kết luận và hướng phát triển	51
5.1 Kết luận	51
5.2 Hướng phát triển	51

# Danh mục hình vẽ

Hình 1 Biểu đồ usecase tổng quan	5
<b>Hình 2</b> Biểu đồ usecase phân rã Quản lý thông tin cá nhân	6
<b>Hình 3</b> Biểu đồ usecase phân rã Theo dõi lịch sử hoạt động	7
<b>Hình 4</b> Biểu đồ usecase phân rã Tìm kiếm	8
<b>Hình 5</b> Biểu đồ usecase phân rã Đặt vé	9
<b>Hình 6</b> Biểu đồ usecase phân rã Quản lý chuyến đi	10
<b>Hình 7</b> Biểu đồ usecase phân rã Quản lý người dùng	11
<b>Hình 8</b> Biểu đồ quy trình nghiệp vụ Đặt vé	12
<b>Hình 9</b> Biểu đồ quy trình nghiệp vụ Quản lý chuyến đi	14
Hình 22. Hình minh họa chức năng "Ghép chuyến đi"	24
Hình 23. Sơ đồ thuật toán ghép xe	25
Hình 10. Mô hình Client-Server	28
Hình 11 Tổng quan kiến trúc phần mềm	31
<b>Hình 12</b> Ví dụ biểu đồ phụ thuộc gói	33
Hình 13 Ví dụ thiết kế gói	34
<b>Hình 14.</b> Giao diện màn hình app mobile	35
<b>Hình 15.</b> Giao diện website	36
Hình 16. Sơ đồ thiết kế cơ sở dữ liệu	38
<b>Hình 17.</b> Giao diện đăng nhập app mobile	44
<b>Hình 18.</b> Giao diện tìm kiếm vé xe	45
Hình 19. Giao diện chọn chuyến và chọn chỗ	46

Hình	20.	Giao	diện	Quản	1ý	người	dùng	và	Quản	lý	công	ty	vận	tải	của	Admin	hệ
thống		•••••															. 47
Hình	21.	Giao d	diện q	uản lý	xe					••••							. 48

# Chương 1 Giới thiệu đề tài

#### 1.1 Đặt vấn đề

Từ xưa đến nay, nhu cầu đi lại của con người lúc nào cũng có và nhu cầu ấy càng ngày càng tăng cùng với sự phát triển của xã hội. Công nghệ được ứng dụng nhiều lĩnh vực trong cuộc sống và đặc biệt trong nhiều năm trở lại đây, công nghệ thông tin được ứng dụng trong ngành Giao thông vận tải, điều đó làm cho cuộc sống của con người tiện lợi hơn.

Ngày trước nếu muốn di chuyển trong tỉnh/thành phố, người ta có thể gọi taxi hay ra đường bắt xe ôm thì bây giờ chỉ cần một chiếc điện thoại, bạn có thể đặt ngay cho mình một chuyến xe đến đúng nơi yêu cầu, thậm chí, tiện lợi hơn bạn có thể trả tiền trong tài khoản mà k cần đến tiền mặt, hay việc bạn có thông tin đầy đủ của tài xế, vị trí của xe cũng làm cho mỗi chuyến đi trở nên an toàn hơn.

Tuy là tiện lợi như vậy nhưng để phát triển thành mô hình lớn không phải điều dễ. Xuất phát từ những điều trên ĐATN của em sẽ xây dựng một hệ thống cho những chuyến đi đường dài – khi hiện nay đa số người dân di chuyển qua các tỉnh vẫn dùng phương thức cũ là ra bến xe mua vé hay gọi điện cho phòng vé. Hệ thống sẽ giúp khách hàng có thể đặt vé bằng điện thoại cá nhân mà không cần phải ra bến, các nhà xe có thể giảm bớt nhân công khi khâu đặt vé đã có hệ thống hỗ trợ,...

#### 1.2 Mục tiêu và phạm vi đề tài

Nhìn vào tình hình hiện nay, ở Việt Nam, chúng ta đã có một số bên triển khai việc mua bán vé xe đường dài online như Vexere, Vetaugiare24h, Megabus,... hay các công ty lớn như Be, Zalo cũng phát triển chức năng này và tích hợp luôn trong app. Tuy nhiên, về phía người dùng, loại hình này chưa thực sự được khách hàng quan tâm mà chỉ một số bộ phận nhỏ, chủ yếu là sinh viên biết đến và sử dụng thử. Còn về phía các công ty vận tải, hiện nay nhiều nhà xe đã hợp tác với các app và sử dụng phần mềm quản lý bán vé của bên cung cấp dịch vụ nên việc đồng bộ hệ thống mua/bán vé cũng dễ dàng hơn.

Với những vấn đề đặt ra ở mục 1.1 cũng như tình hình thực tế nêu ở trên, mục tiêu của đồ án này là xây dựng được hệ thống mua bán vé xe đường dài bao gồm phần mềm cho nhà xe để quản lý việc bán vé, doanh thu,... và phần mềm cho người dùng sử dụng để đặt vé theo nhu cầu.

#### 1.3 Định hướng giải pháp

Từ mục tiêu đã đề ra ở mục 1.2, giao diện quản lý cho các công ty vận tải sẽ được xây dựng trên nền tảng web, với web, nhân viên nhà xe sẽ dễ dàng sử dụng, đặc biệt với các thiết bị màn hình lớn có bố cục rộng rãi sẽ giúp việc quản lý được hiệu quả, tránh những sai sót không cần thiết hơn so với làm việc qua màn hình nhỏ.

Bên cạnh đó, giao diện cho người dùng/khách hàng sẽ được xây dựng trên app mobile, cụ thể trong ĐATN này sẽ xây dựng trên Android vì với tình hình hiện nay, mỗi người đa số đều có cho mình một chiếc smartphone, nên việc xây dựng app mobile sẽ giúp phần mềm tiếp cận được với nhiều khách hàng hơn và chính nó cũng mang lại sự tiện lợi cho người dùng khi chỉ cần vài thao tác đơn giản là đã có ngay vé xe cho mình.

ĐATN sử dụng Nodejs để xây dựng phía server, MySQL làm cơ sở dữ liệu, giao diện web xây dựng bằng PHP, Android Java để xây dựng app mobile, RESTful API để giao tiếp client-server và Firebase để xác thực người dùng. Ngoài ra em sử dụng Github để quản lý mã nguồn một cách an toàn và hiệu quả.

#### 1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 em trình bày về khảo sát của em với các ứng dụng, hệ thống cho phép người dùng đặt vé xe đường dài. Sau đó đưa ra các phân tích, đặc tả các chức năng, nghiệp vụ sẽ xây dựng trong hệ thống của mình.

Trong Chương 3, em sẽ giới thiệu về các công nghệ em sử dụng trong đồ án tốt nghiệp này, các công nghệ để xây dựng giao diện web, giao diện app mobile, back-end và triển khai hệ thống.

Đến Chương 4, em sẽ trình bày chi tiết về phân tích thiết kế hệ thống, các thiết kế kiến trúc, database, giao diện,...

Chương 5 trình bày về những giải pháp, đóng góp nổi bật của mình trong suốt quá trình làm đồ án.

Cuối cùng, Chương 6 là chương sẽ nêu lên những thứ em đã đạt được sau khi làm ĐATN, những thiếu sót, hạn chế của hệ thống cũng như của bản thân để từ đó rút ra kinh nghiệm và có định hướng phát triển sản phẩm thêm trong tương lai.

# Chương 2 Khảo sát và phân tích yêu cầu

#### 2.1 Khảo sát hiện trạng

Hiện nay trên thị trường có một số ứng dụng cho phép người dùng đặt vé xe đường dài và dưới đây em xin đưa ra một vài so sánh giữa các ứng dụng này.

	Vexere	Be	Vetaugiare24h	Megabus
Giao diện app mobile cho người dùng	Có	Có	Không	Không
Tích hợp khai báo thông tin y tế	Không	Không	Không	Không
Thanh toán điện tử	Có	Có	Có	Có
Người dùng được chọn vị trí ghế/giường	Có	Có	Không	Có
Cho phép người dùng đăng ký gửi thêm hàng hóa, xe cộ	Không	Không	Có	Không
Cho phép người dùng đánh giá trải nghiệm của mình về các nhà xe	Có	Không	Không	Có
Thông báo cho khách hàng khi sắp đến giờ khởi hành	Có	Có	Không	Không
Phần mềm quản lý cho các nhà xe	Có	Không	Không	Không

Vexere là hệ thống thương mại dịch vụ dành cho khách hàng muốn đặt vé và cả các nhà xe (công ty vận tải) hay đại lý bán vé cho nhà xe. Đây là một hệ thống khá hoàn chỉnh và nổi

lên trong vài năm trở lại đây, tuy nhiên qua quá trình sử dụng vẫn có thể thấy một số chức năng còn phức tạp hoặc chưa được phát triển như việc tích hợp khai báo y tế thay vì sau khi khách lên xe rồi phải ghi vào giấy khai báo của nhà xe.

Be khởi điểm ban đầu là dịch vụ xe ôm, taxi điện tử. Về sau phát triển thêm nhiều chức năng khác tạo thành hệ sinh thái riêng. Tuy nhiên khi phát triển chức năng đặt vé xe đường dài thì còn chưa được chú trọng, chức năng này chỉ phát triển đơn giản là cho người dùng đặt chỗ và thanh toán tiền vé. Người dùng không có cái đánh giá khách quan về các nhà xe, bên phía nhà xe cũng chưa được hỗ trợ hệ thống quản lý.

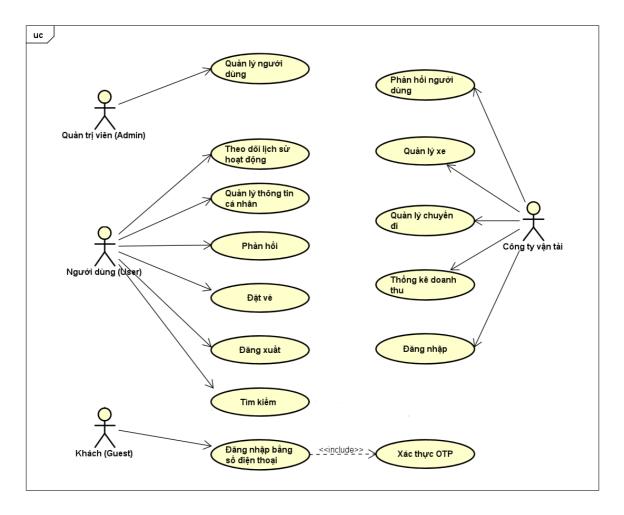
Vetaure 24h là hệ thống đặt vé xe được phát triển cách đây khá lâu, vì vậy cho đến thời điểm hiện tại thì hệ thống hơi lỗi thời, giao diện nhìn chưa bắt mắt và cũng chưa có app mobile cho người dùng, khách hàng đặt vé không được chọn chỗ ngồi.

Megabus là hệ thống cho phép người dùng đặt vé xe, vé máy bay, phòng nghỉ trên cả nước. Các chức năng cơ bản vận hàng tốt tuy nhiên chưa có giao diện app mobile cho người dùng, chưa tích hợp khai báo y tế trong mùa dịch, các nhà xe đăng ký còn hạn chế dẫn đến việc tìm chuyến đi còn khó khăn ở một số tuyến.

Qua tìm hiểu, so sánh những ưu, nhược điểm của các hệ thống đặt vé khác, bên cạnh các chức năng cơ bản như đặt vé, đăng nhập, đăng ký,... hệ thống của em sẽ phát triển thêm một số chức năng hỗ trợ cho người dùng như thông báo cho người dùng khi sắp đến giờ khởi hành, tích hợp khai báo thông tin y tế bằng cách quét mã QR, cho phép người dùng đánh giá các chuyến đi của mình để người dùng sau có cái nhìn khách quan hơn trước khi quyết định đặt vé của nhà xe nào,...

## 2.2 Tổng quan chức năng

#### 2.2.1 Biểu đồ use case tổng quan



Hình 1 Biểu đồ usecase tổng quan

Người dùng (user) là những người đăng nhập vào hệ thống bằng số điện thoại, người dùng sẽ được quyền tìm kiếm, đặt vé xe và thực hiện một số chức năng khác.

Quản trị viên (Admin) là những tài khoản đặc biệt của hệ thống, quản trị viên sẽ là bên quản lý hệ thống: quản lý người dùng, quản lý các công ty vận tải.

Khách (Guest) là những người truy cập vào hệ thống nhưng chưa đăng nhập, để có thể thực hiện được các chức năng khác của hệ thống, khách cần thực hiện chức năng đăng nhập.

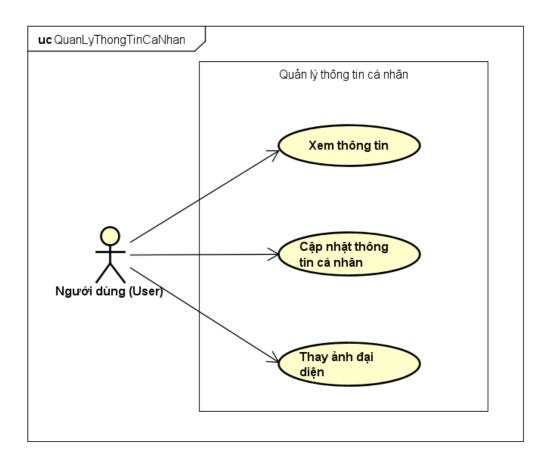
Công ty vận tải là các nhà xe, được cung cấp tài khoản để quản lý, phản hồi người dùng, cập nhật tình hình vé xe,...

Hệ thống có chức năng chính là Đặt vé, cho phép người dùng đặt vé xe của một nhà xe bất kỳ. Bên cạnh đặt vé, người dùng cũng có thể thực hiện các chức năng liên quan như gửi phản

hồi, đánh giá của bản thân. Đối với bên các công ty vận tải, chức năng chính là Quản lý chuyến đi, chức năng cho phép nhà xe tạo/sửa/hoãn các chuyến xe, ghép các chuyến xe ít người để có thể tiết kiệm chi phí.

#### 2.2.2 Biểu đồ use case phân rã

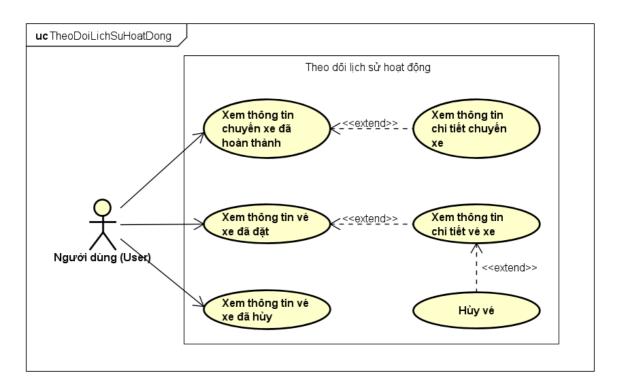
## 2.2.2.1 Biểu đồ use case phân rã Quản lý thông tin cá nhân



Hình 2 Biểu đồ usecase phân rã Quản lý thông tin cá nhân

**Hình 2** mô tả chức năng quản lý thông tin cá nhân, với 1 tác nhân là Người dùng(User), người dùng có thể xem thông tin, chỉnh sửa/cập nhật thông tin bản thân, thay đổi ảnh đại diện của tài khoản.

## 2.2.2.2 Biểu đồ usecase phân rã Theo dõi lịch sử hoạt động



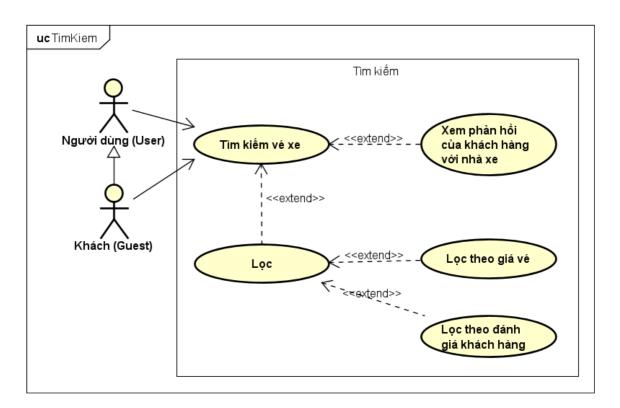
Hình 3 Biểu đồ usecase phân rã Theo dõi lịch sử hoạt động

Hình 3 mô tả chức năng theo dõi lịch sử hoạt động của người dùng.

Người dùng xem thông tin các vé xe mình vừa đặt và chi tiết từng vé đó, nếu vẫn còn trong thời gian hiệu lực người dùng có thể hủy vé đó theo chính sách của hệ thống.

Người dùng xem thông tin các chuyến xe mình đã đi khi đặt vé qua hệ thống, từ đó có thể đánh giá về chuyến đi nếu chưa đánh giá. Ngoài ra, người dùng cũng có thể xem thông tin các vé xe mình đã hủy.

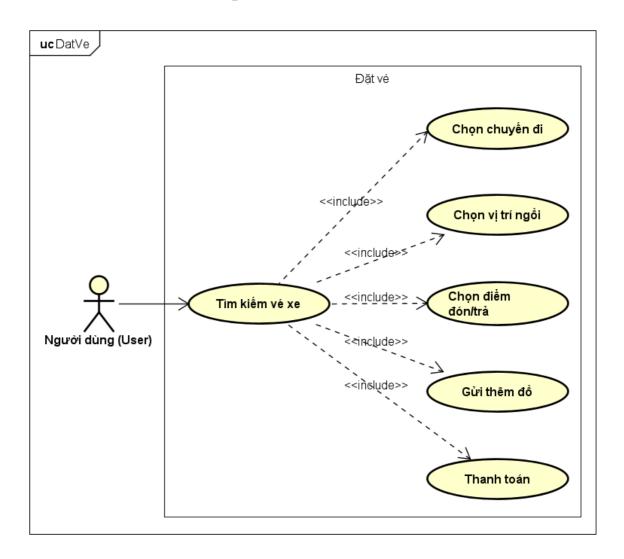
## 2.2.2.3 Biểu đồ usecase phân rã Tìm kiếm



Hình 4 Biểu đồ usecase phân rã Tìm kiếm

**Hình 4** mô tả chức năng tìm kiếm vé xe với 2 tác nhân là người dùng và khách, chức năng này cho phép tìm kiếm vé xe theo một vài thông số nhất định: nơi đi/đến, thời gian. Từ những kết quả đó, người dùng có thể xem phản hồi của những người dùng khác về dịch vụ của từng nhà xe. Ngoài ra, người dùng có thể lọc kết quả theo giá vé (tăng dần/giảm dần) hoặc lọc theo số điểm khách hàng đánh giá.

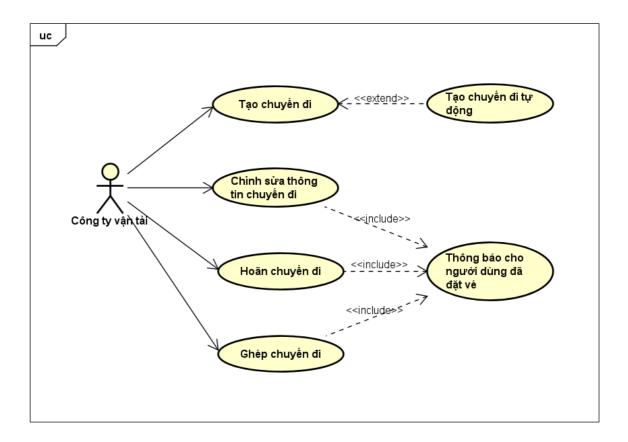
## 2.2.2.4 Biểu đồ usecase phân rã Đặt vé



Hình 5 Biểu đồ usecase phân rã Đặt vé

**Hình 5** mô tả chức năng đặt vé của người dùng. Khách sau khi tìm kiếm nếu muốn đặt vé thì phải đăng nhập mới thực hiện được chức năng này. Chức năng đặt vé bao gồm: Chọn chuyến đi từ những kết quả tìm kiếm được, chọn vị trí ngồi trên xe, chọn điểm đón/trả (điểm đón/trả sẽ là vị trí chi tiết hơn so với nơi đến/đi, có thể là quận/huyện, xã hay một địa điểm nổi tiếng ở địa phương), có thể gửi thêm đồ (vali, xe máy,...), thanh toán.

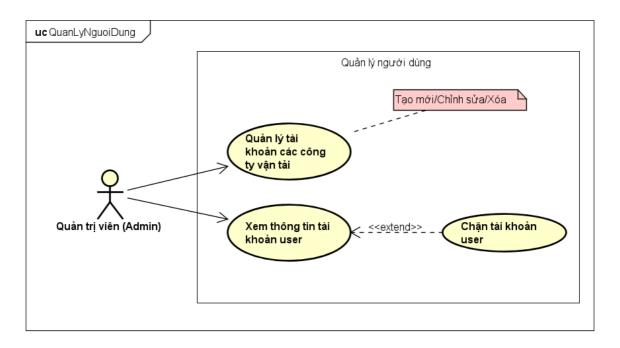
## 2.2.2.5 Biểu đồ usecase phân rã Quản lý chuyến đi



Hình 6 Biểu đồ usecase phân rã Quản lý chuyến đi

Hình 6 mô tả chức năng quản lý chuyến đi của Công ty vận tải, chức năng này bao gồm tạo chuyến đi mới – các nhà xe có thể để hệ thống tạo tự động đối với những chuyến xe có tần suất lặp lại đều đặn, chỉnh sửa thông tin chuyến đi (đối với thông tin quan trọng như thời gian nếu thay đổi phải đưa ra thông báo cho người dùng đã đặt vé chuyến đi đó), hoãn chuyến đi (vì những lý do đặc biệt như có Thông tư của thành phố cấm các dịch vụ vận tải, khi hoãn phải thông báo cho người dùng đã đặt vé và hoàn tiền nếu đã thanh toán), với những chuyến đi trong ngày nếu có ít người đặt vé thì có thể ghép các chuyến với nhau để tiết kiệm chi phí.

## 2.2.2.6 Biểu đồ usecase phân rã Quản lý người dùng

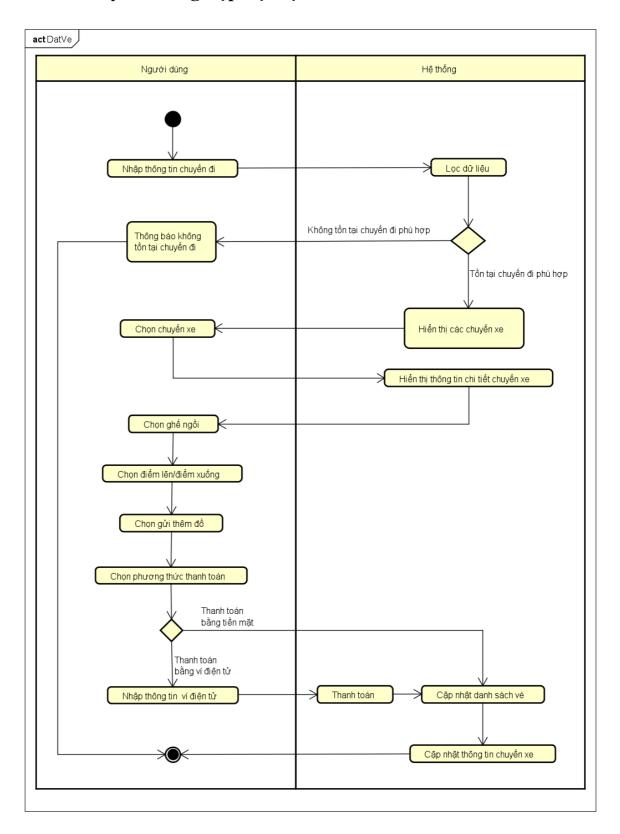


Hình 7 Biểu đồ usecase phân rã Quản lý người dùng

**Hình 7** mô tả chức năng quản lý người dùng của Quản trị viên hệ thống, Quản trị viên có thể tạo mới, chỉnh sửa, xóa tài khoản cho các công ty vận tải hợp tác với hệ thống, xem thông tin tài khoản của người dùng trong hệ thống, đối với những người dùng bỏ chuyến không thanh toán quá số lần quy định sẽ bị chặn khỏi hệ thống.

#### 2.2.3 Quy trình nghiệp vụ

## 2.2.3.1 Quy trình nghiệp vụ Đặt vé



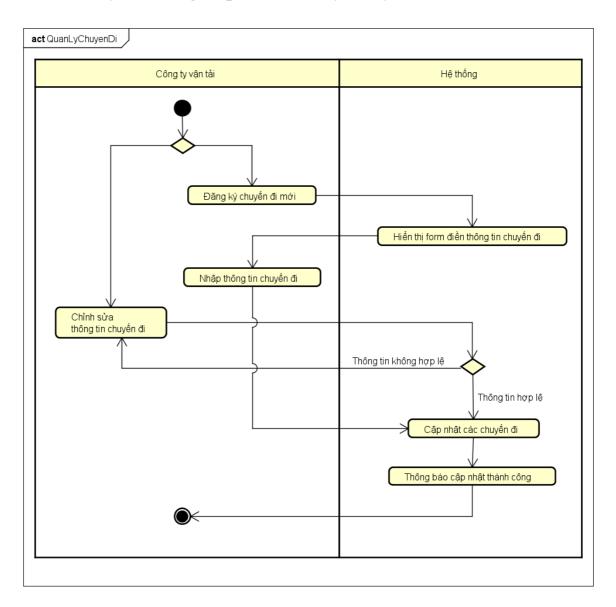
Hình 8 Biểu đồ quy trình nghiệp vụ Đặt vé

Hình 8 mô tả quy trình nghiệp vụ Đặt vé, đây là một nghiệp vụ quan trọng của hệ thống bên phía user. Để đặt vé, khách hàng cần đăng nhập tài khoản cá nhân, tìm kiếm chuyến đi phù hợp với nhu cầu của bản thân thông qua các thông tin: Nơi đến/đi (tỉnh/thành), thời gian đi. Dựa vào thông tin đó, hệ thống sẽ đưa ra các chuyến đi trùng thông tin người dùng tìm kiếm, tại đây người dùng có thể lọc các kết quả tìm được theo giá vé, điểm đánh giá. Sau khi chọn được chuyến đi, người dùng chọn các thông tin chi tiết hơn: vị trí ngồi, điểm lên/xuống (các điểm trung chuyển của nhà xe nếu có, quận/huyện/xã hay ở một địa điểm nổi tiếng tại địa phương), gửi thêm đồ (tùy theo quy định của nhà xe sẽ miễn phí hoặc lấy kèm phí), chọn phương thức thanh toán. Với phương thức thanh toán tiền mặt, khách hàng sẽ thanh toán khi lên xe hay tại quầy vé bến xe.

Những khách hàng đặt vé và chọn thanh toán tiền mặt nếu bỏ chuyến sẽ có cảnh báo từ hệ thống tới tài khoản người dùng, nếu bỏ chuyến 3 lần hệ thống sẽ khóa tài khoản.

Những khách hàng đặt vé và thanh toán qua ví điện tử, khi hủy vé trong thời gian quy định (trước 1 ngày), sẽ được hoàn lại một số tiền nhất định (80% tiền vé).

#### 2.2.3.2 Quy trình nghiệp vụ Quản lý chuyến đi



Hình 9 Biểu đồ quy trình nghiệp vụ Quản lý chuyến đi

**Hình 9** mô tả quy trình nghiệp vụ Quản lý chuyến đi, đây là quy trình quan trọng của hệ thống bên phía công ty vận tải.

Với các chuyến đi mới, nhân viên nhà xe tạo theo form thông tin hệ thống cho sẵn hoặc có thể đặt lịch tạo tự động với những chuyến lặp lại định kỳ.

Với các chuyến đi đã tạo và cần chỉnh sửa thông tin, nhân viên có thể chỉnh sửa một số thông tin nhất định: có thể chỉnh sửa giờ xuất phát nhưng không được đổi ngày xuất phát nếu đã có khách hàng đặt vé ở chuyến đi đó, đổi xe,...

Với các chuyến đi có quá ít khách đặt, nhà xe có thể cân nhắc để ghép các chuyến đi của mình hoặc ghép với chuyến của một số nhà xe khác có cùng tuyến đường đi chuyển. Tuy

nhiên, với việc ghép chuyến đi này, nhà xe sẽ phải có thông báo sớm với khách đặt vé để khách có thể cân nhắc, nếu không đồng ý, khách hàng sẽ được hoàn tiền 100% khi đã thanh toán.

## 2.3 Đặc tả chức năng

#### 2.3.1 Đặc tả use case Đăng nhập bằng số điện thoại

Mã Use case	UC001		Tên Use case	Đăng nhập bằng số điện thoại	
Tác nhân	Khách				
Tiền điều kiện	Không				
Luồng sự kiện chính	STT	Thực hiện bởi	Hành đ	ộng	
(Thành công)	1.	Khách	Yêu cầu form đăng nhập		
	2.	Hệ thống	Hiển thị màn hình đăng r	nhập	
	3.	Khách	Nhập số điện thoại và xá	c nhận	
	4.	Hệ thống	Gửi mã OTP tới số điện thoại khách vừa nhập		
	5.	Hệ thống	Hiển thị màn hình nhập mã OTP		
	6.	Khách	Nhập mã OTP và xác nhận		
	7.	Hệ thống	Kiểm tra mã OTP khách	vừa nhập	
	8.	Hệ thống	Thông báo đăng nhập thành công và hiển thị màn hình chính		
,					
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động		
	3.a.	Hệ thống	Thông báo nhập sai số đi	ện thoại	
	3.b.	Khách	Thực hiện luồng sự kiện	3	
	7.a.	Hệ thống	Thông báo mã OTP chưa đúng		

Hậu điều kiện	Không
---------------	-------

 $\ast$  Dữ liệu đầu vào của form đăng nhập gồm các trường dữ liệu sau:

**Bảng 1.** Dữ liệu đầu vào use case đăng nhập bằng số điện thoại

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Số điện thoại	Nội dung dạng văn bản	Có	Gồm 9 ký tự nếu ký tự đầu tiên không phải "0" Gồm 10 ký tự nếu ký tự đầu tiên là "0"	37954456 037954456

#### 2.3.2 Đặc tả use case Đặt vé

Mã Use case	UC002		Tên Use case	Đặt vé			
Tác nhân	Người	Người dùng					
Tiền điều kiện							
Luồng sự kiện chính	STT	STT Thực hiện bởi Hành động					
(Thành công)	1.	Người dùng	Nhập thông tin chuyển đi	và tìm kiếm			
	2.	Hệ thống	Hiển thị kết quả tìm kiếm				
	3.	Người dùng	Chọn chuyến đi				
	4.	Hệ thống	Hiển thị màn hình chọn v	rị trí ngồi			
	5.	Người dùng	Chọn vị trí ngồi và xác nh	hận			
	6.	Hệ thống	Hiển thị màn hình chọn v	rị trí đón/trả			
	7.	Người dùng	Chọn vị trí đón/trả				
	8.	Hệ thống	Hiển thị màn hình gửi thế	êm đồ			
	9.	Người dùng	Lựa chọn gửi thêm hoặc	bỏ qua			

	10.	Hệ thống	Hiển thị màn hình chọn phương thức thanh toán
	11.	Người dùng	Chọn phương thức thanh toán
	12.	Hệ thống	Hiển thị màn hình thanh toán
	13.	Người dùng	Xác nhận thanh toán
	14.	Hệ thống	Thông báo thanh toán thành công
	15.	Hệ thống	Thông báo đặt vé thành công
	16.	Hệ thống	Lưu lại thông tin vé và cập nhật thông tin chuyển đi
Luồng sự kiện			
thay thế	STT	Thực hiện bởi	Hành động
	12.a.	Hệ thống	Thông báo khách đã chọn thanh toán bằng tiền mặt và đặt vé thành công
	12.b.	Khách	Lưu lại thông tin vé và cập nhật thông tin chuyển đi
Hậu điều kiện	Không		

 $<sup>{}^{*}</sup>$  Dữ liệu đầu vào các thông tin khi đặt vé gồm các trường dữ liệu sau:

**Bảng 2.** Dữ liệu đầu vào use case Đặt vé

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Vị trí ngồi	Tập hợp các toggle button, mỗi button thể hiện cho một vị trí trên xe	Có	Có ít nhất 1 button được chọn	
2.	Vị trí	Các radio button, 1	Có	Với mỗi điểm	

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
	đón/trả	ô điền nội dung văn bản với mỗi lựa chọn đón và trả,		đón/trả đều tích 1 lựa chọn	
3.	Gửi thêm đồ	Checkbox các lựa chọn: xe đạp, xe điện, xe máy, khác (với lựa chọn khác thêm 1 ô điền nội dung văn bản)	Không		
4.	Chọn phương thức thanh toán	2 lựa chọn Ví điện tử và Tiền mặt	Có	Chọn 1 trong 2 lựa chọn	

## 2.3.3 Đặc tả use case Thực hiện chuyển đi

Mã Use case	UC003		Tên Use case	Thực hiện chuyến đi	
Tác nhân	Người	dùng			
Tiền điều kiện	Người	dùng đã đặt vé th	nành công		
Luồng sự kiện chính				ộng	
(Thành công)	1.	Hệ thống	Thông báo trước khi đến	giờ xuất phát 1 tiếng	
	2.	Hệ thống	Hệ thống Thông báo đến giờ khởi hành, hiển thị ma hình quét mã QR		
	3.	Người dùng	Quét mã QR để khai báo	thông tin y tế	
	4.	Hệ thống	Hiển thị màn hình kết thứ giờ xuống xe	íc chuyến đi khi đến	
	5.	Người dùng	Xác nhận đã hoàn thành	chuyến đi	

	6.	Hệ thống	Hiển thị màn hình đánh giá chuyến đi	
	7.	Người dùng	Đánh giá, phản hồi của bản thân về chuyến đi, nhà xe, nhân viên,	
8. Hệ thống		Hệ thống	Cập nhật chuyến đi, phản hồi	
Luồng sự kiện				
<u> </u>				
thay thế	STT	Thực hiện bởi	Hành động	
thay thế	7.a.	Thực hiện bởi Người dùng	Hành động  Bỏ qua bước đánh giá và sẽ đánh giá sau	
thay thế				
thay thế	7.a.	Người dùng	Bổ qua bước đánh giá và sẽ đánh giá sau	

<sup>\*</sup> Dữ liệu đầu vào các thông tin khi đặt vé gồm các trường dữ liệu sau:

Bảng 3. Dữ liệu đầu vào chức năng quét mã QR

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Mã QR	Ånh mã QR được sinh ra cho mỗi chuyến đi	Có		

#### 2.3.4 Đặc tả use case Tạo chuyến đi

Mã Use case	UC004		Tên Use case	Tạo chuyển đi	
Tác nhân	Công ty vận tải				
Tiền điều kiện					
Luồng sự kiện					
chính	STT	Thực hiện bởi	Hành độ	òng	
(Thành công)	1.	Công ty vận tải	Chọn chức năng Quản lý	chuyến đi	

	2.	Hệ thống	Hiển thị màn hình Quản lý chuyến đi
	3.	Công ty vận tải	Chọn Thêm mới tuyến đường (nếu chưa có)
	4.	Hệ thống	Hiển thị màn hình Thêm mới tuyến đường
	5.	Công ty vận tải	Điền thông tin tuyến đường mới
	6.	Hệ thống	Cập nhật thông tin tuyến đường mới cho nhà xe
	7.	Hệ thống	Hiển thị màn hình thêm các điểm trung chuyển cho tỉnh/thành 1
	8.	Công ty vận tải	Nhập các điểm trung chuyển và xác nhận
	9.	Hệ thống	Hiển thị màn hình thêm các điểm trung chuyển cho tỉnh/thành 2
	10.	Công ty vận tải	Nhập các điểm trung chuyển và xác nhận
	11.	Hệ thống	Thông báo thêm tuyến đường mới thành công
	12.	Công ty vận tải	Chọn Thêm mới chuyến đi
	13.	Hệ thống	Hiển thị màn hình Thêm mới chuyến đi
	14.	Công ty vận tải	Điền thông tin chuyến đi mới
	15.	Hệ thống	Cập nhật chuyến đi
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3.a.	Công ty vận tải	Thực hiện luồng hoạt động 7 (nếu vẫn sử dụng tuyến đường cũ)
	12.a.	Công ty vận tải	Chọn chức năng tự động tạo chuyển đi theo lịch trình

lịch trình

chuyến đi

12.b. Hệ thống

Hiển thị màn hình điền thông tin tự động tạo

	12.c.	Công ty vận tải	Nhập thông tin
	12.d.	Hệ thống	Cập nhật thông tin, tự tạo chuyến đi theo lịch trình
Hậu điều kiện	Không		

<sup>\*</sup> Dữ liệu đầu vào use case Tạo chuyến đi gồm các trường dữ liệu sau:

Bảng 4. Dữ liệu đầu vào chức năng Thêm mới tuyến đường

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Nơi đi/đến	Dropdown button gồm các tỉnh thành trên cả nước	Có	Chọn 1 tỉnh thành	
2.	Điểm trung chuyển	Trường nhập văn bản	Không		

Bảng 5. Dữ liệu đầu vào chức năng Thêm mới chuyến đi

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
3.	Xe	Dropdown button gồm các xe mà công ty vận tải sở hữu	Có		
4.	Tuyến đường	2 Dropdown button gồm nơi đến, nơi đi của nhà xe	C6		

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
5.	Ngày giờ xuất phát	Button lịch	C6		
6.	Vị trí ngồi	Checkbox vị trí ghế ngồi trong xe (dành cho những ghế đã được khách đặt hàng trước)	Không		

**Bảng 6.** Dữ liệu đầu vào chức năng Tạo chuyến đi theo lịch trình

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
7.	Xe	Dropdown button gồm các xe mà công ty vận tải sở hữu	Có		
8.	Tuyến đường	2 dropdown button gồm nơi đến, nơi đi mà công ty vận tải chạy xe	Có		
9.	Giờ xuất phát	Button lịch	Có		
10.	Tần suất	Dropdown gồm: mỗi ngày, mỗi tuần, mỗi tháng	Có		

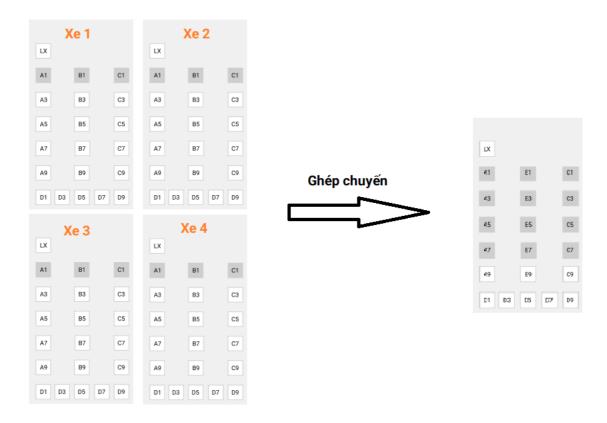
#### 2.4 Giải pháp quản lý chuyến đi cho các công ty vận tải

#### 2.4.1 Đặt vấn đề

Với những hệ thống đặt vé xe hiện nay, mỗi nhà xe sẽ có khá nhiều chuyến đi trong 1 ngày, vì mô hình kinh doanh bây giờ của đa số các công ty vận tải khá lớn, không gói gọn lại trong một vài chiếc xe mà có rất nhiều xe thuộc sở hữu của công ty, nên việc sắp xếp nhiều chuyến trong một ngày cũng xem như là một giải pháp để nhà xe có thể chở được nhiều khách hơn, khách hàng có nhiều lựa chọn khung giờ đi trong ngày hơn, qua đó doanh thu của nhà xe cũng tốt hơn. Nhưng cũng từ đó, nảy sinh ra một vấn đề ảnh hưởng khá lớn đến lợi nhuận của các công ty vận tải, mặc dù có thể doanh thu cao hơn nhưng việc có nhiều chuyến đi trong ngày cũng làm tăng chi phí vận hành hệ thống tăng lên nhiều lần. Chính vì thế, với những ngày có lượng khách ít, rất có thể có nhiều chuyến có khách đặt nhưng mỗi chuyến xe chỉ có 2-3 người đặt, dẫn đến việc nhà xe bắt buộc phải hủy chuyến hoặc chấp nhận chạy xe chỉ với số ít khách. Cả hai phương án này đều khiến cho phía công ty vận tải chịu lỗ khá nặng cho mỗi ngày như vậy.

#### 2.4.2 Giải pháp đưa ra

Để giải quyết vấn đề vừa nêu trên, hệ thống của em đã phát triển thêm một tính năng "Ghép chuyến đi". Thay vì hủy chuyến hay chấp nhận chạy chuyến ít người, các nhà xe có thể ghép những chuyến đi ít người đấy lại với nhau, tạo thành một chuyến có đông người hơn, qua đó cũng tiết kiệm chi phí cũng như đảm bảo lợi nhuận hơn rất nhiều.



Hình 10. Hình minh họa chức năng "Ghép chuyến đi"

Để đảm bảo sự trải nghiệm của khách hàng, những chuyến xe được ghép cùng nhau phải thỏa mãn: các chuyến xe cùng ngày, cùng một loại xe (ghế ngồi, giường nằm, limousine,...), cùng nhà xe. (1) Nhân viên nhà xe sẽ tự chủ động chọn chức năng ghép chuyến đi khi cần thiết (nên trước khoảng 1-2 tiếng so với thời gian xuất phát chuyến xe sớm nhất). Sau khi các chuyến xe được ghép, hệ thống sẽ gửi thông báo tới người dùng để người dùng nắm được thông tin.

Dưới đây là cách mà hệ thống sẽ tính toán và thực hiện ghép chuyến ngay sau khi có yêu cầu từ phía nhà xe.

Giả sử nhà xe A có n chuyến xe chưa xuất phát trong ngày hôm nay, dĩ nhiên cả n chuyến xe này đều thỏa mãn các tính chất (1) vừa nêu ở trên. Xe thứ i (i = 1,2,3,...,n), sẽ có  $a_i$  chỗ ngồi/giường và  $b_i$  vé đã được đặt.

Ta có thể tính lợi nhuận của nhà xe bằng công thức:

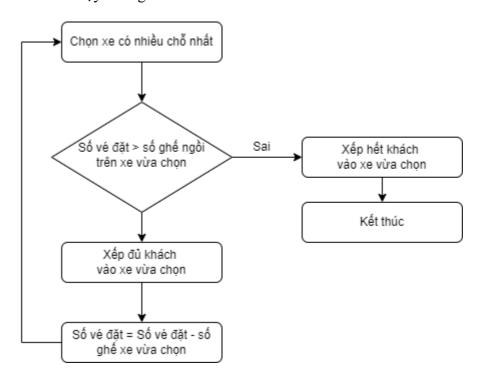
Lợi nhuận = 
$$\sum_{i=1}^{n} (b_i \times giá vé) - \sum_{i=1}^{n} (chi phí vận hành)$$

Vì cùng một loại xe, cùng loại dịch vụ nên giá vé của mỗi xe đều giống nhau.

Chi phí vận hành bao gồm tiền xăng, lương tài xế, nhân viên, chi phí phát sinh,... Đấy là những khoản nó sẽ thay đổi với từng xe, từng chuyến đi khác nhau và rất khó để có thể ước lượng chính xác được thông số này nên để đơn giản hơn chúng ta se ước lượng chi phí vận hành trung bình mỗi xe (điều này sẽ dễ hơn rất nhiều cho các công ty vận tải).

Lợi nhuận 
$$=\sum_{i=1}^{n}(b_i\times giá\ v\acute{e})-n\times chi phí vận hành trung bình mỗi xe$$

Vì với số lượng khách đã đăng ký vé  $(b_i)$  là cố định nên để có lợi nhuận tốt, ta cần bớt đi lượng  $n \times chi phí vận hành trung bình mỗi xe$  hay một cách tường minh hơn, chúng ta cần giảm số xe cần chay xuống mức tối thiểu.



Hình 11. Sơ đồ thuật toán ghép xe

Ta sẽ lần lượt xếp khách đã đặt vé vào các xe có số chỗ nhiều nhất cho đến khi hết khách.

#### 2.5 Yêu cầu phi chức năng

#### 2.5.1 Yêu cầu về bảo mật

Để đảm bảo tính bảo mật, chính chủ cũng như thuận tiện cho phía người dùng, hệ thống chỉ cho phép người dùng đăng nhập bằng số điện thoại thông qua xác thực OTP.

#### 2.5.2 Yêu cầu giao diện

Úng dụng, website cần có giao diện thân thiện, dễ nhìn, dễ sử dụng đối với người dùng. App mobile cần phù hợp với các điện thoại sử dụng hệ điều hành Android hiện nay. Sử dụng các hình ảnh, icon liên quan đến chức năng của hệ thống để tạo sự đơn giản, dễ liên tưởng.

#### 2.5.3 Yêu cầu khác

Hệ thống cần được xây dựng để dễ dàng sửa đổi, mở rộng, linh động theo nhu cầu thực tế của người dùng, của các công ty vận tải.

## Chương 3 Công nghệ sử dụng

#### 3.1 Server

#### **3.1.1 NodeJS**

NodeJS là một nền tảng (platform) được xây dựng dựa trên V8 Javascript Engine – trình thông dịch thực thi mã Javascript. Đây là công cũ mạnh mẽ để xây dựng server hay các ứng dụng web một cách nhanh chóng, hiệu quả. NodeJS được viết bằng C++ do đó nó cho tốc độ xử lý và hiệu năng khá cao.

NodeJS có thể chạy được trên nhiều hệ điều hành: Linux, Windows, Mac OS.

NodeJS có một cộng đồng lập trình viên sử dụng lớn, đó là một lợi thế, khi sử dụng NodeJS có thể tham khảo hoặc được giải đáp những thắc mắc, những lỗi gặp phải trong quá trình sử dụng.

#### **3.1.2 MySQL**

MySQL là một hệ thống quản trị cơ sở dữ liệu quan hệ mã nguồn mở (RDMS), dựa trên ngỗn ngữ truy vấn có cấu trúc (SQL) và được hỗ trợ bởi tập đoàn Oracle.

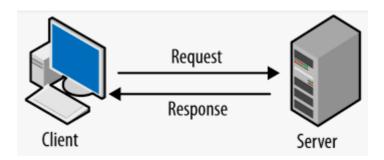
MySQL được sử dụng rộng rãi bởi khả năng dễ sử dụng, hoạt động rất nhanh và tốt với các tập dữ liệu lớn và được sử dụng vởi các trang web lớn như Google, Twitter, Facebook,...

Tính năng nổi bật của MySQL:

- MySQL được phát hành theo giấy phép mã nguồn mở.
- MySQL lý tưởng cho tất cả ứng dụng lớn và nhỏ.
- MySQL có tốc độ xử lý, truy vấn nhanh, an toàn, dễ mở rộng và sử dụng.
- MySQL hoạt động trên đa nên tảng với hầu hết các ngôn ngữ (PHP, JavaScript, Java, C, Kotlin,...
- MySQL hỗ trợ cơ sở dữ liệu lớn, lên tới 50 triệu hàng trong một bảng, giới hạn kích thước mặc định một bảng là 4GB và có thể tăng giới hạn đến 8 triệu TB.
- MySQL có tiêu chuẩn bảo mật mã hóa thông tin cao, đảm bảo an toàn dữ liệu

Các tính năng này khiến MySQL phù hợp cho sản phẩm ĐATN này.

MySQL hoạt động dựa trên mô hình Client-Server được mô tả ở hình dưới:



Hình 12. Mô hình Client-Server

Máy client sẽ liên lạc với server trong mạng nhất định, MySQL tạo ra bảng để lưu trữ, định nghĩa dữ liệu. Client sẽ gửi các yêu cầu đến server và server sẽ phản hồi lại kết quả sau khi truy vấn dữ liệu từ database.

#### 3.2 Website

#### **3.2.1 VueJS**

Vue.js là một framework Javascript được tạo bởi Evan You, thích hợp để xây dựng UI cho các single-page application (SPA). Các trang web viết bằng Vue có khả năng lấy dữ liệu từ server mà không cần tải lại trang giúp tăng trải nghiệm người dùng.

Vue.js là một framework linh động dùng để xây dựng giao diện người dùng. Khác với các monolithic framework, Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước. Khi phát triển lớp giao diện, người dùng chỉ cần dùng thư viện lõi của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn.

#### 3.3 App mobile

#### 3.3.1 Java

Java là một ngôn ngữ lập trình hiện đại, bậc cao, hướng đối tượng, bảo mật và mạnh mẽ. Java ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995. Mặc dù càng ngày càng xuất hiện nhiều ngôn ngữ lập trình mới đi kèm những thư viện giúp công việc của lập trình viên bớt nặng nhọc tuy nhiên Java vẫn đang là một ngôn ngữ lập trình được rất đông lập trình viên trên thế giới sử dụng.

Một số điểm nổi bật của Java:

 Hướng đối tượng: Trong Java, mọi thứ đều là một Object. Java có thể dễ dàng mở rộng và bảo trì vì nó được xây dựng dựa trên mô hình Object.

- Bảo mật: Với tính năng an toàn của Java, nó cho phép phát triển các hệ thống không có virut, giả mạo. Các kỹ thuật xác thực dựa trên mã hoá khóa công khai.
- Đa luồng: Với tính năng đa luồng của Java có thể viết các chương trình có thể thực hiện nhiều tác vụ đồng thời. Tính năng thiết kế này cho phép các nhà phát triển xây dựng các ứng dụng tương tác có thể chạy tron tru hơn.
- Thông dịch: Mã byte Java được dịch trực tiếp tới các máy tính gốc và không được lưu trữ ở bất cứ đâu.
- Hiệu năng cao: Với việc sử dụng trình biên dịch Just-In-Time, Java cho phép thực hiện hiệu năng cao.
- Tính an toàn: Ngôn ngữ lập trình Java yêu cầu chặt chẽ về kiểu dữ liệu, dữ liệu phải được khai báo tường minh, không sử dụng con trỏ và các phép toán với con trỏ, Java kiểm soát chặt chẽ việc truy nhập đến mảng, chuỗi, không cho phép sử dụng các kỹ thuật tràn do đó các truy nhập sẽ không vượt quá kích thước của mảng hoặc chuỗi. Quá trình cấp phát và giải phóng bộ nhớ được thực hiện tự động. Cơ chế xử lý lỗi giúp việc xử lý và phục hồi lỗi dễ dàng hơn.

#### 3.3.2 RESTful API

RESTful API là một tiêu chuẩn trong việc thiết kế API cho các ứng dụng client-server để thuận tiện cho việc quản lý tài nguyên và truyền tải qua phương thức HTTP.

Thành thần chính của RESTful API:

- API (Application Programming Interface): là tập hợp các quy tắc để tương tác giữa các thành phần ứng dụng. API trả về dạng dữ liệu JSON hoặc XML.
- REST( Representational State Transfer): là dạng chuyển đổi cấu trúc để viết API. Nó tạo tương tác giữa các máy với nhau qua HTTP và quy định việc sử dụng các phương thức HTTP và dạng URL cho các ứng dụng.

Trong đồ án này thì dữ liệu trả về sẽ luôn là dạng JSON.

#### 3.3.3 Firebase

Firebase là một dịch vụ hệ thống backend được Google cung cấp sẵn cho ứng dụng. Dịch vụ này giúp phát triển, triển khai và mở rộng quy mô của ứng dụng Mobile nhanh hơn, dù là Android hay IOS, Firebase vẫn có thể hỗ trợ tốt. Trong một số dự án, Firebase là các API đơn giản, không cần backend hay server. Tuy nhiên, đồ án này chỉ sử dụng một số tính năng của firebase như xác thực số điện thoại.

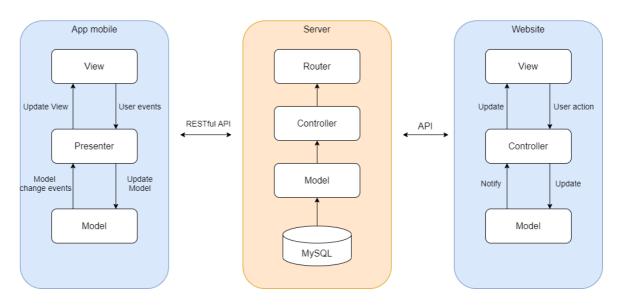
Ưu điểm:

- Triển khai ứng dụng nhanh chóng: Firebase tiết kiệm rất nhiều thời gian quản lý và đồng bộ tất cả dữ liệu cho người dùng. Đó là nhờ người dùng không phải quan tâm đến phần backend cùng các API tốt, hỗ trợ đa nền tảng.
- Sự ổn định: Được viết dựa trên nền tảng cloud cung cấp bởi Google, các công cụ luôn đảm bảo độ ổn định tối đa. Bên cạnh đó, quá trình nâng cấp hay bảo trì Server cũng diễn ra nhanh và đơn giản hơn.

# Chương 4 Phát triển và triển khai ứng dụng

### 4.1 Thiết kế kiến trúc

#### 4.1.1 Lựa chọn kiến trúc phần mềm



Hình 13 Tổng quan kiến trúc phần mềm

Hệ thống được triển khai như mô hình được mô tả ở **Hình 11**. Server sẽ sử dụng (truy xuất, thêm, sửa, xóa) dữ liệu từ database thông qua các model, các chức năng logic được thực thi tại controller và router sẽ định tuyến, cung cấp phương thức giao tiếp cho app cũng như web. Giao diện app mobile (cho khách hàng) và web (cho công ty vận tải) hiển thị dữ liệu được nhận từ server, đồng thời nhận các yêu cầu của người dùng gửi về server để xử lý. App mobile được xây dựng theo mô hình MVP, web được xây dựng theo mô hình MVC.

#### 4.1.1.1 Mô hình MVC

MVC là từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình chia source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

Model là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là

một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,...

View là phần giao diện dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website. View có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller.

Controller là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view, từ đó đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

#### **4.1.1.2 Mô hình MVP**

MVP là một mô hình kiến trúc lập trình gần giống như MVC. MVP có nhiều điểm được kế thừa từ MVC trong đó Controller được thay bằng Presenter. Mô hình này chia ứng dụng thành 3 phần chính: Model, View và Presenter. MVP là một mô hình kiến trúc hướng giao diện người dùng, được thiết kế để tạo thuận lợi cho việc kiểm thử đơn vị (unit testing) và tăng tính tách biệt giữa tầng dữ liệu và tầng hiển thị dữ liệu trong mô hình MVC.

Model, tương tự như ở mô hình MVC, là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Model bao gồm các lớp mô tả business logic, định nghĩa business rules cho dữ liệu.

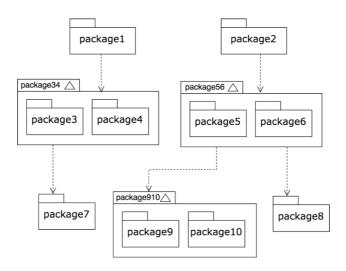
View là thành phần tương tác trực tiếp và nhận yêu cầu từ người dùng như XML, Activity, fragments. Nó không bao gồm bất kỳ việc xử lý logic nào.

Presenter sẽ nhận input của người dùng thông qua View, rồi xử lý dữ liệu của người dùng với sự trợ giúp của Model và trả kết quả về View. Presenter giao tiếp với View qua interface. Interface được định nghĩa trong lớp Presenter. Trong cấu trúc MVP, Presenter thao túng Model và cập nhật ở View. View và Presenter tách biệt với nhau hoàn toàn và giao tiếp với nhau qua thông qua interface. Vì nếu tách riêng từng phần ở View sẽ dễ dàng cho việc kiểm thử ứng dụng ở MVP.

### 4.1.2 Thiết kế tổng quan

Sinh viên vẽ biểu đồ gói UML (UML package diagram), nêu rõ sự phụ thuộc giữa các gói (package). SV cần vẽ các gói sao cho chúng được phân theo các tầng rõ ràng, không được sắp đặt package lộn xộn trong hình vẽ. Sinh viên chú ý các quy tắc thiết kế (Các gói không phụ thuộc lẫn nhau, gói tầng dưới không phụ thuộc gói tầng trên, không phụ thuộc bỏ qua

tầng, v.v.) và cần giải thích sơ lược về mục đích/nhiệm vụ của từng package. SV tham khảo ví dụ minh họa trong Hình 14.



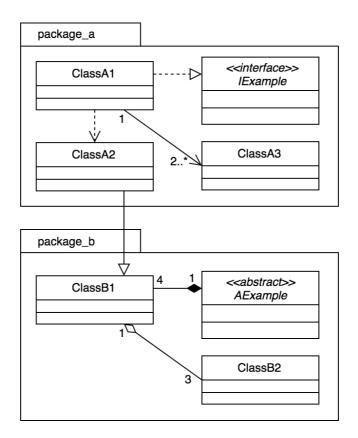
Hình 14 Ví dụ biểu đồ phụ thuộc gói

#### 4.1.3 Thiết kế chi tiết gói

Sinh viên thiết kế và lần lượt vẽ biểu đồ thiết kế cho từng package, hoặc một nhóm các package liên quan để giải quyết một vấn đề gì đó. Khi vẽ thiết kế gói, sinh viên chỉ cần đưa tên lớp, không cần chỉ ra các thành viên phương thức và thuộc tính. SV tham khảo ví dụ minh họa trong Hình 15.

Sinh viên cần vẽ rõ ràng quan hệ giữa các lớp trong biểu đồ. Các quan hệ bao gồm: phụ thuộc (dependency), kết hợp (association), kết tập (aggregation), hợp thành (composition), kế thừa (inheritance), và thực thi (implementation). Các quan hệ này đều đã được minh họa trong Hình 15.

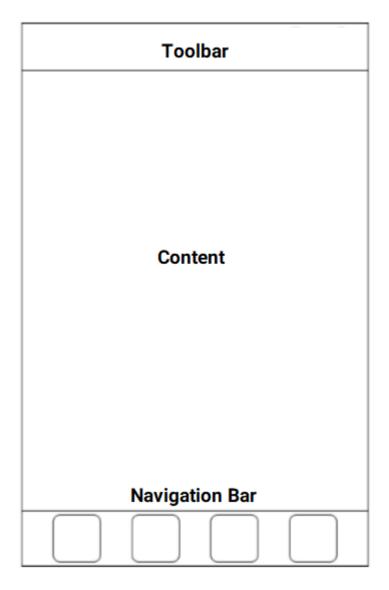
Sau khi vẽ hình minh họa, sinh viên cần giải thích ngắn gọn về thiết kế của mình.



**Hình 15** Ví dụ thiết kế gói

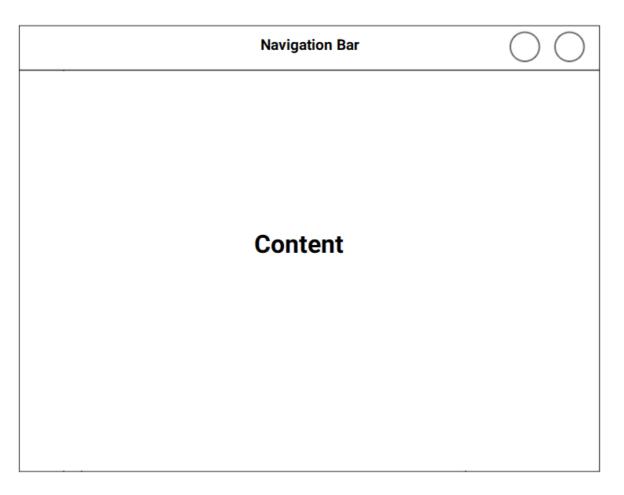
# 4.2 Thiết kế chi tiết

#### 4.2.1 Thiết kế giao diện



Hình 16. Giao diện màn hình app mobile

App mobile được thiết kế trên nền tảng Android, phù hợp với hầu hết màn hình điện thoại android hiện tại, nhưng tương thích nhất màn hình có độ phân giải là 1280x720 pixels. Bố cục gồm ba phần chính là Toolbar, Content, và Navigation Bar. Toolbar chứa thanh tiêu đề của màn hình, đồng thời có thể có các giao diện tương tác nhanh tại đây. Content là nội dung chính, là các fragment hiển thị các thông tin chính của từng màn hình. Navigation Bar là thành phần cố định điều hướng các màn hình. Content và Toolbar thay đổi phù hợp với từng nội dung hiển thị.



Hình 17. Giao diên website

Giao diện web được thiết kế responsive, phù hợp với nhiều kích thước màn hình khác nhau.

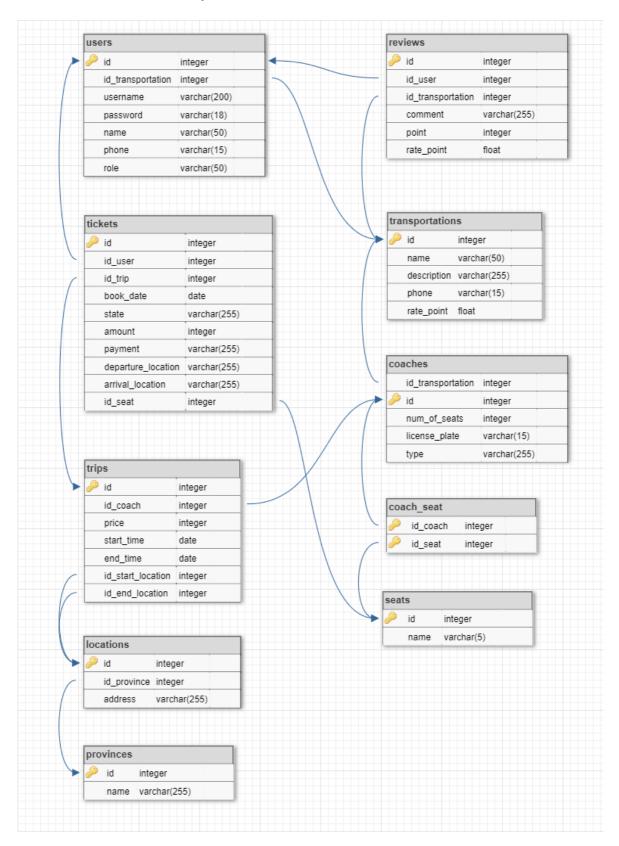
Website không yêu cầu load lại trang sau mỗi thao tác nên có tính tương tác cao với người dùng. Ngoài ra, giao diện ứng dụng sử dụng các icon thể hiện các chức năng một cách trực quan, giúp người dùng có thể dễ dàng làm quen và sử dụng hệ thống.

Tông màu chủ đạo của ứng dụng là xanh dương, mang lại cảm giác dễ chịu cho người sử dụng, không bị loá mắt khi sử dụng lâu dài.

Bố cục đơn giản với 2 phần chính: Navigation Bar là thanh điều hướng giúp người dùng có thể chuyển qua các chức năng khác nhanh chóng, thuận tiện. Content là nội dung ứng với từng chức năng.

# 4.2.2 Thiết kế lớp

#### 4.2.3 Thiết kế cơ sở dữ liệu



# Hình 18. Sơ đồ thiết kế cơ sở dữ liệu

Chi tiết các bảng trong cơ sở dữ liệu:

**Bảng 7.** Thiết kế chi tiết bảng users

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh người dùng
username	varchar(200)		Tên đăng nhập
password	varchar(18)		Mật khẩu
phone	varchar(15)		Số điện thoại
name	varchar(50)		Họ tên người dùng
role	varchar(50)		Chức năng của người dùng
id_transportation	int	Khóa ngoại	Định danh công ty vận tải mà người dùng là nhân viên (chỉ có với user có role là nhân viên nhà xe)

**Bảng 8.** Thiết kế chi tiết bảng transportations

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh công ty vận tải
name	varchar(255)		Tên công ty vận tải
description	varchar(255)		Thông tin thêm về nhà xe
phone	varchar(15)		Số điện thoại
rate_point	float		Điểm người dùng đánh giá

Bảng 9. Thiết kế chi tiết bảng coach\_type

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh loại xe
name	varchar(255)		Tên loại xe
description	varchar(255)		Mô tả

Bảng 10. Thiết kế chi tiết bảng coaches

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh xe
id_transportation	int	Khóa ngoại	Định danh công ty vận tải
license_plate	varchar(15)		Số điện thoại
num_of_seats	int		Số chỗ trên xe
id_type	int	Khóa ngoại	Loại xe

Bảng 11. Thiết kế chi tiết bảng seats

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh ghế
name	varchar(10)		Tên ghế
id_coach_type	int	Khóa ngoại	Định danh loại xe

Bảng 12. Thiết kế chi tiết bảng coach\_seat

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id_coach	int	Khóa ngoại	Định danh xe
id_seat	int	Khóa ngoại	Định danh ghế

**Bảng 13.** Thiết kế chi tiết bảng provinces

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh tỉnh/thành
name	varchar(255)		Tên tỉnh/thành

Bảng 14. Thiết kế chi tiết bảng locations

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh địa điểm
id_province	int	Khóa ngoại	Định danh tỉnh/thành
address	varchar(255)		Địa chỉ
name	varchar(255)		Tên địa điểm

**Bảng 15.** Thiết kế chi tiết bảng trips

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh chuyến đi
id_coach	int	Khóa ngoại	Định danh xe
price	int		Giá vé
start_time	date		Thời gian xuất phát
end_time	date		Thời gian kết thúc dự kiến

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id_start_location	int	Khóa ngoại	Định danh địa điểm xuất phát
id_end_location	int	Khóa ngoại	Định danh địa điểm kết thúc

**Bảng 16.** Thiết kế chi tiết bảng payments

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh đơn thanh toán
method	varchar(255)		Phương thức thanh toán
amount	int		Tổng tiền thanh toán
paid	int		Đã thanh toán/Chưa thanh toán

**Bảng 17.** Thiết kế chi tiết bảng tickets

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh vé
id_trip	int	Khóa ngoại	Định danh chuyến đi
id_user	int	Khóa ngoại	Định danh người dùng
id_seat	int	Khóa ngoại	Định danh ghế
book_date	date		Thời gian đặt vé
state	varchar(255)		Trạng thái vé
id_payment	int	Khóa ngoại	Định danh đơn thanh toán
departure_location	varchar(255)		Địa điểm trung chuyển đón (nếu có)
arrival_location	varchar(255)		Địa điểm trung chuyển xuống

Tên tr	ường	Kiểu dữ liệu	Ràng buộc	Mô tả
				(nếu có)

Bảng 18. Thiết kế chi tiết bảng reviews

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	Khóa chính	Định danh đánh giá
id_user	int	Khóa ngoại	Định danh người dùng
id_transportation	int	Khóa ngoại	Định danh công ty vận tải
comment	varchar(255)		Bình luận đánh giá
point	int		Điểm đánh giá

# 4.3 Xây dựng ứng dụng

### 4.3.1 Thư viện và công cụ sử dụng

Thư viện và công cụ được sử dụng cho việc hoàn thành đồ ản được mô tả trong **Bảng 17** dưới đây:

Bảng 19. Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual Studio Code	https://code.visualstudio.com/
IDE lập trình	Android Studio	https://developer.android.com/
Xây dựng backend	NodeJS	https://nodejs.org/
Xây dựng frontend web	VueJS	https://vuejs.org
Database hệ thống	MySQL	https://www.mysql.com/

Xác thực số điện thoại người dùng	Firebase	https://firebase.google.com/
Xây dựng frontend web	VueJS	https://vuejs.org

# 4.3.2 Kết quả đạt được

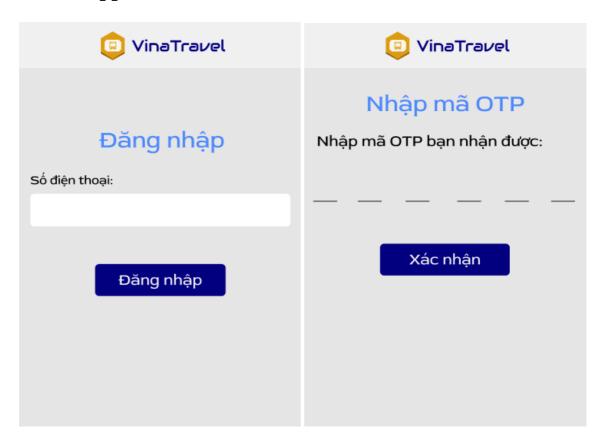
Qua quá trình tìm hiểu, em đã xây dựng được một hệ thống đặt vé xe đường dài, hệ thống giúp các công ty vận tải có thể tự quản lý các chuyến đi, doanh số bán hàng,... và từ những thông tin của các nhà xe đó, người dùng có thể đặt vé một cách nhanh chóng, thuận tiện.

Dưới đây là một số thống kê về các thông số của source code hệ thống:

Thông tin	Số liệu
Số dòng code	Khoảng 15000 dòng
Số file mã nguồn	Khoảng 300 files
Dung lượng mã nguồn	Front-end Web: 3MB, Mobile: 3MB, Back-end: 7MB

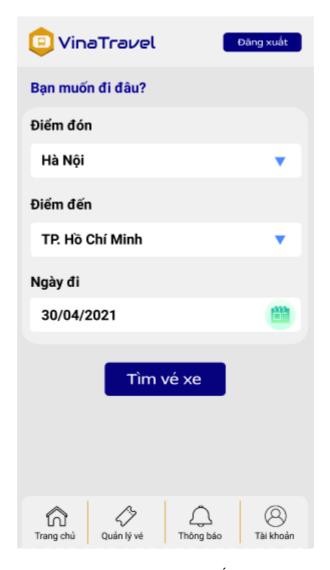
#### 4.3.3 Minh hoạ các chức năng chính

### **4.3.3.1** App mobile



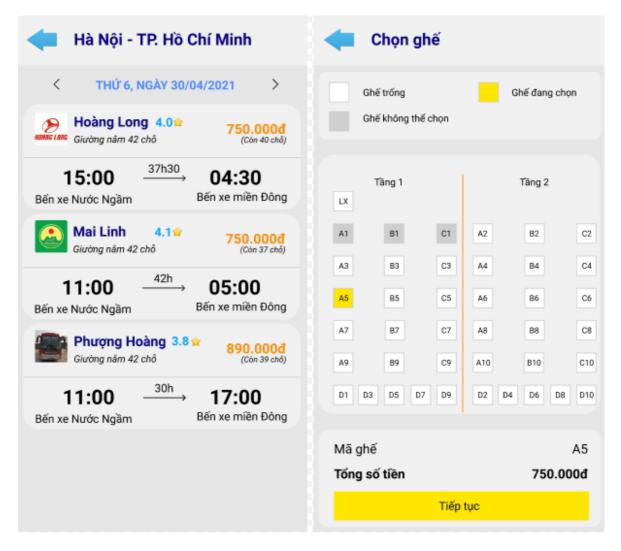
Hình 19. Giao diện đăng nhập app mobile

Úng dụng cho phép người dùng đăng nhập bằng số điện thoại, sau khi nhập số điện thoại, một mã OTP sẽ được gửi về số người dùng đã nhập để có thể xác thực người dùng đang sở hữu số điện thoại đó. Với cách đăng nhập này, người dùng có thể bớt đi nỗi lo sợ khi quên mật khẩu như ở các hệ thống khác. Giao diện được thiết kế tinh giản, không màu mè, tạo cảm giác dễ chịu, dễ nhìn cho người dùng.



Hình 20. Giao diện tìm kiếm vé xe

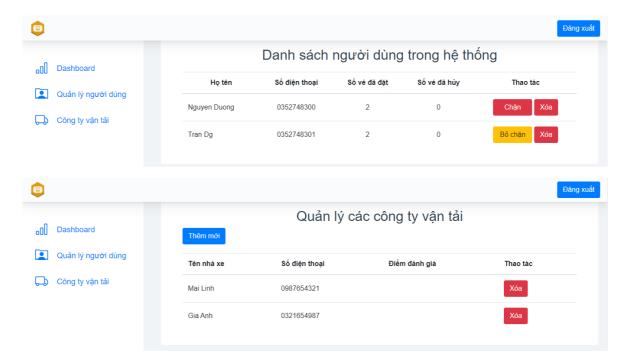
Sau khi đăng nhập thành công, người dùng sẽ được chuyển đến màn hình trang chủ, cũng chính là màn hình tìm kiếm vé xe – một trong những chức năng quan trọng nhất của app. Ở đây, người dùng sẽ nhập 3 thông tin cơ bản của chuyến xe cần tìm: điểm đón, điểm đến và ngày đi, từ đó, hệ thống sẽ hiển thị ra những kết quả phù hợp với yêu cầu của người dùng.



Hình 21. Giao diện chọn chuyến và chọn chỗ

Hệ thống sẽ tìm kiếm những chuyến đi phù hợp với yêu cầu của người dùng và hiển thị lên màn hình "Chọn chuyến đi", người dùng click vào chuyến đi mình muốn và chọn chỗ trên xe. Ngoài ra, để hoàn thành chức năng đặt vé, người dùng sẽ điền thêm một vài thông tin như chỗ lên, chỗ xuống, gửi kèm hàng.

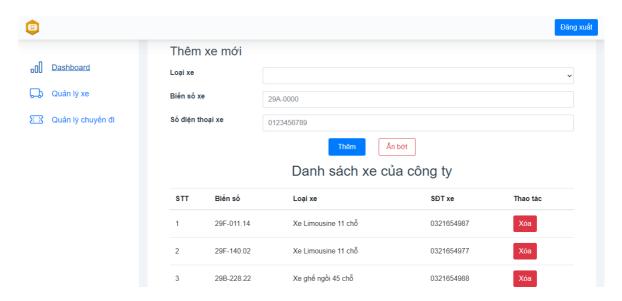
# 4.3.3.2 Web cho Admin hệ thống



Hình 22. Giao diện Quản lý người dùng và Quản lý công ty vận tải của Admin hệ thống

Với các công ty vận tải, Admin có thể quản lý các thông tin, thêm mới hoặc xóa một công ty bất kỳ khỏi hệ thống. Còn đối với người dùng, ngoài chức năng xem, xóa người dùng, Admin có thể chặn/bỏ chặn người dùng đó, với những người dùng có số lần hủy vé, bùng vé nhiều hệ thống sẽ tự động chặn, và để khôi phục người dùng phải liên lạc với hệ thống qua email rồi từ đó, admin sẽ xem xét tình huống và đưa ra quyết định bỏ chặn hay không. Những tài khoản bị chặn sẽ không thể đăng nhập cũng như sử dụng các chức năng của hệ thống.

### 4.3.3.3 Web cho phía công ty vận tải



Hình 23. Giao diện quản lý xe

Các công ty vận tải sẽ có trách nhiệm quản lý các xe của mình, mỗi xe sẽ có biển số xe riêng để phân biệt và có số điện thoại riêng để khi gần đến giờ xuất phát hoặc khi có thắc mắc, người dùng có thể gọi cho tài xế hoặc lơ xe của chuyến họ đã đặt.

#### 4.4 Kiểm thử

#### 4.4.1 Kiểm thử tương thích

Thiết bị	Thông số	Giao diện	Chức năng
0 52	Màn hình: IPS LCD 5.5" Full HD		
Oppo F3	RAM: 4GB		
	Màn hình: P-OLED 5.5" Full HD+		
Google Pixel 3	RAM: 4GB		

Bảng 20. Thống kê kiểm thử tương thích

### 4.4.2 Kiểm thử chức năng

Để kiểm thử các chức năng chính của hệ thống, em tiến thành kiểm thử bằng phương pháp kiểm thử hộp đen.

Chức năng	Test case	Kết quả
	Tìm kiếm vé xe	
Đặt vé	Chọn chỗ trên xe, cập nhật giá tiền	
	Chọn điểm lên, xuống, điền thêm thông tin khác	
0 ' 1' '	Cập nhật thông tin vé đã đặt, đã đi, đã hủy	
Quản lý vé	Hủy vé xe	

Bảng 21. Kiểm thử chức năng Đặt vé của người dùng

Chức năng	Test case	Kết quả
0 1/	Thêm xe mới	
Quản lý xe	Cập nhật thông tin xe	
	Thêm chuyến đi	
Quản lý chuyến đi	Ghép chuyến đi	
	Hoãn chuyến đi	
Thông báo	Gửi thông báo tới người dùng	

Bảng 22. Kiểm thử chức năng quản lý phía công ty vận tải

# 4.5 Triển khai

Úng dụng mobile được triển khai trên thiết bị di động chạy hệ điều hành Android. Server được triển khai trên localhost tại laptop cá nhân. Bảng dưới đây mô tả thông tin chi tiết các thiết bị đã được dùng để triển khai hệ thống.

Thiết bị	Vai trò	Cấu hình
Onna F2	App mobile cho người dùng	Màn hình: IPS LCD 5.5" Full HD
Oppo F3		RAM: 4GB

Thiết bị	Vai trò	Cấu hình
Laptop HP 15-ay169TX Z6X61PA	Server	Intel Core i7-7500U (2.70 GHz Up to 3.50GHz, 4MB)  8GB RAM DDR4
Laptop HP 15-ay169TX Z6X61PA	Giao diện web	Màn hình: 15.6" HD 8GB RAM DDR4

# Chương 5 Kết luận và hướng phát triển

# 5.1 Kết luận

Sau thời gian một kỳ học tìm hiểu, phân tích, nghiên cứu với sự hướng dẫn của thầy Trần Đình Khang, em đã xây dựng được cơ bản một hệ thống đặt vé xe đường dài . Hệ thống đã giải quyết được một số vấn đề lớn trong thời buổi hiện nay như (i) cung cấp môi trường quản lý dễ dàng, tiện lợi cho các công ty vận tải, (ii) cung cấp ứng dụng giúp người dùng đặt vé một cách nhanh chóng, thuận tiện thay vì đặt vé như cách truyền thống, (iii) cung cấp giải pháp cho vấn đề ghép các chuyến đi để tránh lãng phí tài nguyên.

Trong quá trình phát triển hệ thống, em cũng được tiếp xúc và học tập, trau dồi về nhiều công nghệ, kiến thức mới như một số framework phát triển frontend, backend: VueJS, Android, NodeJS, RESTful API, quy trình triển khai, xây dựng một hệ thống,...

Tuy nhiên, vì thời gian có hạn cũng như kiến thức, khả năng của bản thân còn hạn chế nên hệ thống còn có nhiều thiếu sót: giao diện chưa thực sự hài hòa về bố cục hay màu sắc, trải nghiệm người dùng ở một vài chức năng còn chưa tốt,...

# 5.2 Hướng phát triển

Trong tương lai, hệ thống sẽ được nâng cấp và hoàn thiện hơn nữa về các tính năng cũng như trải nghiệm cho người dùng. Hệ thống sẽ có thể tích hợp thêm chức năng thanh toán từ các ngân hàng hay ví điện tử, tích hợp bản đồ Google/Apple giúp người dùng có thể định vị vị trí của xe, của bản thân trong thời gian thực.

Ngoài ra, ứng dụng mobile sẽ được phát triển trên các nền tảng khác, cũng như nghiệp vụ của các công ty vân tải sẽ được thay đổi để phù hợp với nhu cầu thực tế hơn nữa.