

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO TỔNG HỢP
GIAI ĐOẠN 1 + GIAI ĐOẠN 2

KHẢO SÁT FUTA BUS

NHÓM 20HTTT2_8
QUẢN TRỊ CƠ SỞ DỮ LIỆU HIỆN ĐẠI
Thành phố Hồ Chí Minh – 2024

MỤC LỤC

MỤC LỤC	2
THÔNG TIN NHÓM	3
GIAI ĐOẠN 1	4
I. USE CASE HỆ THỐNG	4
II. ĐẶC TẢ USE CASE HỆ THỐNG	6
III. TẦN SUẤT GIAO DỊCH	13
IV. GUI (BẢN THỬ)	15
V. CÁC THÀNH PHẦN DỮ LIỆU TRONG FUTA BUS	19
GIAI ĐOẠN 2	21
I. USE CASE HỆ THỐNG (UPDATE)	21
II. ĐẶC TẢ USE CASE HỆ THỐNG (UPDATE)	23
III. LẬP BẢNG PHÂN TÍCH ƯU/NHƯỢC ĐIỂM CỦA TỪNG LOẠI NOSQL	31
a) Bảng ưu điểm của các loại NoSQL	31
b) Bảng ưu điểm của các CSDL NoSQL	36
c) Bảng nhược điểm của các loại NoSQL	40
d) Bảng nhược điểm của các CSDL NoSQL	44
IV. PHÂN TÍCH CÁC CHỨC NĂNG ĐÃ MÔ TẢ TRONG HỆ THỐNG	48
V. THIẾT KẾ CƠ SỞ DỮ LIỆU CHO TỪNG LOẠI	55
VI. GIAO DIỆN CHỨC NĂNG	59
a) Giao diện chính	59
b) Giao diện đặt vé và thanh toán	60
c) Giao diện hiển thị hóa đơn	61
d) Giao diện tra cứu vé đã đặt	62
VII. THIẾT KẾ CHỨC NĂNG	63
a) Đặt vé	63
b) Thanh Toán	65
c) Hóa đơn	68
d) Tra cứu vé đã đặt	70

THÔNG TIN NHÓM

Giai đoạn 1

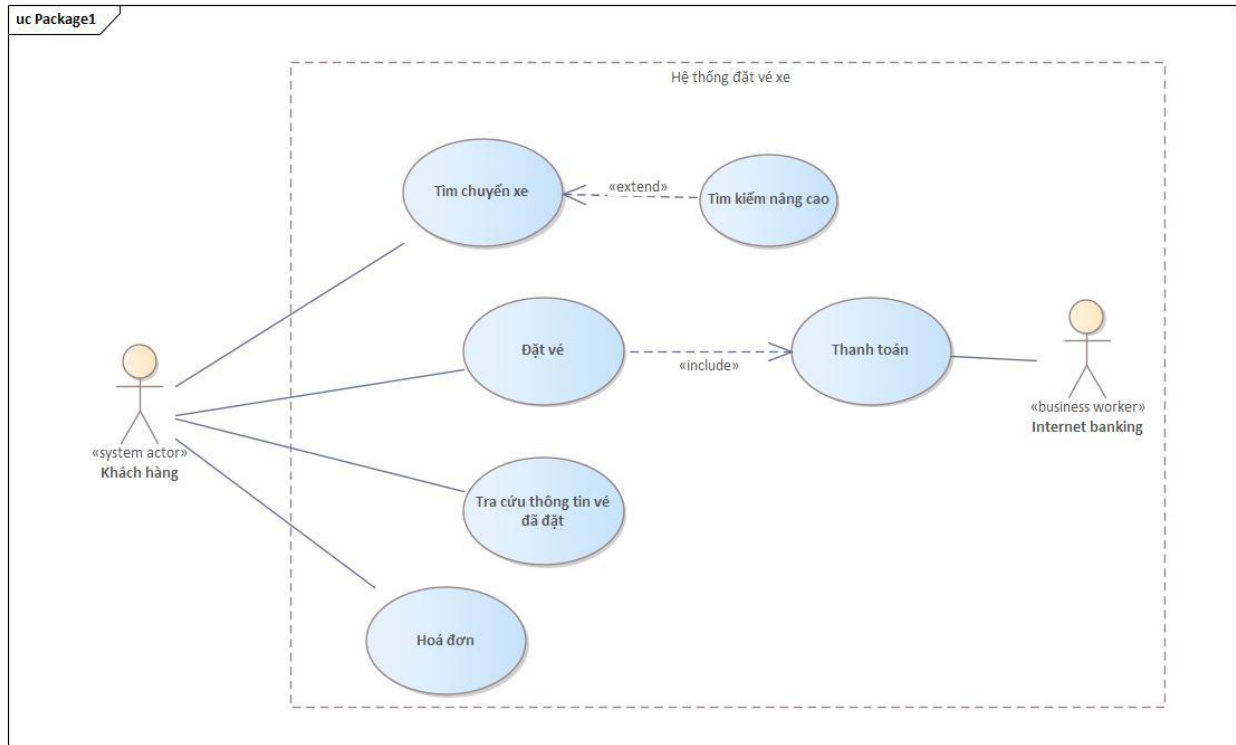
Họ tên	MSSV	Công việc	Tiến độ
Phạm Thanh Long	20127235	Use case hệ thống	100%
Đỗ Phương Nam	20127418	Tần suất giao dịch	100%
Quách Đỗ Gia Huy	20127401	Use case hệ thống	100%
Lê Hoàng Long	18127047	GUI	100%

Giai đoạn 2

Họ tên	MSSV	Công việc	Tiến độ
Phạm Thanh Long	20127235	Tính năng Đặt vé + Tổng hợp báo cáo	100%
Đỗ Phương Nam	20127418	Tính năng Tra cứu vé đã đặt + báo cáo	100%
Quách Đỗ Gia Huy	20127401	Tính năng Thanh toán + báo cáo	100%
Lê Hoàng Long	18127047	Tính năng Hoá đơn + báo cáo	100%

GIẢI ĐOẠN 1

I. USE CASE HỆ THỐNG



➤ **Mô tả chi tiết về quy trình tương tác giữa các Use case:**

1. **Tìm chuyến xe:** Khách hàng bắt đầu bằng việc tìm kiếm chuyến xe phù hợp. Trong quá trình này, họ có thể sử dụng chức năng “Tìm kiếm nâng cao” để lọc kết quả tìm kiếm theo các tiêu chí cụ thể như giờ đi, loại xe, hàng ghế, tầng, v.v. Và cũng có thể sử dụng “Tra cứu thông tin chuyến xe” để xem trước chuyến xe.
2. **Đặt vé:** Sau khi tìm thấy chuyến xe phù hợp, khách hàng tiếp tục đến use case “Đặt vé”. Tại đây, họ chọn số lượng vé, cung cấp thông tin liên lạc và thông tin đón trả, sau đó đặt vé.
3. **Thanh toán:** Sau khi đặt vé, khách hàng tiếp tục đến use case “Thanh toán”. Tại đây, họ chọn hình thức thanh toán (thẻ tín dụng, ví điện tử, v.v.) và cung cấp thông tin thanh toán. Hệ thống sau đó xác nhận thông tin thanh toán và hoàn tất giao dịch. Trong quá trình này, hệ thống có thể tương tác với “Internet Banking” để xử lý giao dịch thanh toán.
4. **Hóa đơn:** Sau khi thanh toán thành công, hệ thống tạo hóa đơn cho khách hàng. Khách hàng có thể tra cứu và xác thực hóa đơn thông qua hệ thống quản lý hóa đơn.
5. **Tra cứu thông tin vé đã đặt:** Cuối cùng, sau khi đã đặt và thanh toán vé, khách hàng có thể tra cứu thông tin về vé đã đặt.

*Ghi chú: **Đăng nhập/ đăng ký** không phải là 1 Use case bởi không cần đăng nhập thì hệ thống vẫn cho đặt vé, cùng với đó là Use case này không có tính hoàn thiện, không đem lại giá trị như những Use case trên*

II. ĐẶC TẢ USE CASE HỆ THỐNG

Tên use case	Tìm chuyến xe
Tóm tắt	Khách hàng tìm kiếm và xem thông tin chi tiết về các chuyến xe có sẵn.
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào hệ thống tìm chuyến xe. 2. Hệ thống hiển thị danh sách các chuyến xe có sẵn 3. Khách hàng chọn chuyến xe và nhập thông tin tìm kiếm (điểm đi, điểm đến, ngày giờ). 4. Hệ thống tìm kiếm và hiển thị danh sách chuyến xe phù hợp với thông tin tìm kiếm của khách hàng, bao gồm thông tin về nơi đi, nơi đến, thời gian khởi hành, giá vé và số ghế còn trống. 5. Khách hàng xem chi tiết về một chuyến xe bằng cách chọn chuyến xe trong danh sách. 6. Hệ thống hiển thị thông tin chi tiết về chuyến xe, bao gồm lịch trình, tài xế, loại xe và các thông tin khác.
Dòng sự kiện phụ	<p>A1. Tại bước 4, nếu không có chuyến xe phù hợp,</p> <ul style="list-style-type: none"> - Hệ thống sẽ không xuất ra các chuyến và thông báo cho khách hàng - Không có các bước sau
Điều kiện tiên quyết	Hệ thống phải có danh sách các chuyến xe có sẵn và thông tin chi tiết về các chuyến xe đó .
Hậu điều kiện	Danh sách chuyến xe phù hợp được hiển thị cho khách hàng. Khách hàng có thể xem thông tin chi tiết về chuyến xe và tiếp tục các hành động khác như đặt vé.

Tên use case	Tìm kiếm nâng cao
Tóm tắt	Khách hàng tìm kiếm chuyến xe với các tiêu chí tùy chọn để tìm kiếm chuyến xe theo nhu cầu cụ thể.
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào hệ thống tìm chuyến xe 2. Khách hàng chọn tùy chọn tìm kiếm nâng cao. 3. Khách hàng cung cấp các tiêu chí cho bộ lọc tìm kiếm như giờ đi , loại xe , hàng ghế , tầng 4. Hệ thống tìm kiếm và hiển thị danh sách chuyến xe phù hợp với bộ lọc tìm kiếm của khách hàng, 5. Khách hàng xem chi tiết về một chuyến xe bằng cách chọn chuyến xe trong danh sách. 6. Hệ thống hiển thị thông tin chi tiết về chuyến xe, bao gồm lịch trình, tài xế, loại xe và các thông tin khác.
Dòng sự kiện phụ	<p>A1. Tại bước 4, nếu không có bộ lọc được cung cấp</p> <ul style="list-style-type: none"> - Hệ thống sẽ hiển thị tất cả các chuyến xe có sẵn mà không áp dụng bộ lọc tìm kiếm. <p>A2. Tại bước 4, nếu không có chuyến xe nào phù hợp với bộ lọc, hệ thống sẽ thông báo cho khách hàng.</p>
Điều kiện tiên quyết	Hệ thống phải có danh sách các chuyến xe có sẵn và thông tin chi tiết về các chuyến xe.
Hậu điều kiện	Danh sách chuyến xe phù hợp với tìm kiếm nâng cao được hiển thị cho khách hàng.

Tên use case	Đặt vé
Tóm tắt	Sau khi khách hàng sử dụng Tìm chuyến xe phù hợp với nhu cầu và giờ giấc, khách hàng sẽ tiếp tục quá trình là đặt vé của chuyến xe đó.
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Sau khi chọn chuyến xe, hệ thống hiển thị giao diện đặt vé bao gồm: Chọn ghế - Thông tin khách hàng – Điểm đón/trả - Thanh toán 2. Khách hàng chọn số lượng ghế phù hợp với nhu cầu 3. Khách hàng điền thông tin khách hàng cần thiết và đồng ý điều khoản 4. Khách hàng chọn điểm đón/trả được hiển thị sẵn trên hệ thống 5. Khách hàng có thể chọn thanh toán với số tiền đã hiển thị trên hệ thống sau khi điền các thông tin cần thiết
Dòng sự kiện phụ	<p>A2: Tại bước 2, nếu khách hàng không chọn ghế</p> <ul style="list-style-type: none"> - Hệ thống sẽ không thanh toán được (ít nhất 1 ghế và tối đa 5 ghế) <p>A3: Tại bước 3, nếu khách hàng không điền đầy đủ thông tin hoặc không đồng ý điều khoản</p> <ul style="list-style-type: none"> - Hệ thống sẽ không thanh toán được
Điều kiện tiên quyết	Khách hàng phải điền đầy đủ thông tin trên trang đó và đồng ý điều khoản để có thể sang bước tiếp theo
Hậu điều kiện	Hệ thống sẽ lưu thông tin của trang Đặt vé và đưa khách hàng tới trang Thanh toán

Tên use case	Thanh toán
Tóm tắt	Sau khi khách hàng điền đầy đủ thông tin cho vé và bấm Thanh toán, khách hàng sẽ tiếp tục quá trình là thanh toán bằng các phương thức khác nhau.
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị các thông tin đã xác nhận và các phương thức thanh toán khác nhau 2. Khách hàng chọn phương thức thanh toán 3. Khách hàng quét mã QR để thanh toán
Dòng sự kiện phụ	<p>A3.1: Tại bước 3, nếu khách hàng không thanh toán trong thời gian chờ giữ ghế</p> <ul style="list-style-type: none"> - Hệ thống sẽ huỷ ghế - Quay lại Usecase đặt vé <p>A3.2: Tại bước 3, nếu khách hàng có nhu cầu chuyển đổi vé thì chỉ có thể chuyển 1 lần duy nhất</p> <p>A3.3: Tại bước 3, nếu khách hàng có nhu cầu thay đổi hoặc huỷ vé đã thanh toán:</p> <ul style="list-style-type: none"> - Liên hệ với tổng đài 1900 6067 hoặc quầy vé - Vé huỷ chỉ trả lại chi phí từ 10% - 30% tùy thuộc thời gian huỷ vé so với giờ khởi hành ghi trên vé và số lượng vé cá nhân/tập thể áp dụng theo các quy định hiện hành
Điều kiện tiên quyết	Khách hàng phải thanh toán để có vé
Hậu điều kiện	Hệ thống xác nhận thanh toán và gửi vé qua mail mà khách hàng đã điền thông tin từ trước

Tên use case	Tra cứu thông tin vé
Tóm tắt	Có thể thực hiện sau khi UC Thanh toán đã hoàn thành, khách hàng tra cứu thông tin vé đã được gửi
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none">1. Hệ thống gửi thông tin vé qua mail mà khách hàng đã cho trước2. Khách hàng vào mail kiểm tra thông tin vé
Dòng sự kiện phụ	A2: Tại bước 2, nếu khách hàng kiểm tra thông tin vé không phù hợp hoặc sai sót <ul style="list-style-type: none">- Báo lên hệ thống hoặc CSKH để giải quyết
Điều kiện tiên quyết	Không có
Hậu điều kiện	Khách hàng có vé trên mail và dùng vé đó để sử dụng khi thời điểm khởi hành

Tên use case	Hóa đơn
Tóm tắt	Hệ thống cho phép người dùng tra cứu và xác thực hóa đơn
Tác nhân	Người dùng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng truy cập vào hệ thống quản lý hóa đơn. 2. Hệ thống hiển thị giao diện quản lý hóa đơn cho người dùng. 3. Người dùng chọn chức năng "Tra cứu hóa đơn". 4. Hệ thống yêu cầu người dùng nhập thông tin cần tra cứu 5. Người dùng cung cấp thông tin tra cứu. 6. Hệ thống truy vấn cơ sở dữ liệu và hiển thị danh sách hóa đơn phù hợp với thông tin tra cứu. 7. Người dùng có thể chọn một hóa đơn từ danh sách để xem chi tiết.
Dòng sự kiện thay thế	<ol style="list-style-type: none"> 1a. Người dùng truy cập vào hệ thống quản lý hóa đơn. 2a. Hệ thống hiển thị giao diện quản lý hóa đơn cho người dùng. 3a. Người dùng chọn chức năng "Xác thực hóa đơn". 4a. Hệ thống yêu cầu người dùng nhập thông tin hóa đơn cần xác thực 5a. Người dùng cung cấp thông tin xác thực. 6a. Hệ thống kiểm tra thông tin xác thực và cập nhật trạng thái xác thực của hóa đơn tương ứng. 7a. Hệ thống thông báo cho người dùng kết quả xác thực thành công hoặc thất bại.
Dòng sự kiện phụ	<p>A1. Tại bước 6, nếu không tìm thấy hóa đơn phù hợp</p> <ul style="list-style-type: none"> - Hệ thống sẽ thông báo cho người dùng.

	<ul style="list-style-type: none">- Không có các bước sau A2. Tại bước 7a , nếu thông tin xác thực không hợp lệ <ul style="list-style-type: none">- Hệ thống sẽ thông báo lỗi cho người dùng và yêu cầu người dùng nhập lại thông tin.
Điều kiện tiên quyết	Khách hàng đã hoàn thành thanh toán và có mã vé chuyển xe. Hệ thống phải có cơ sở dữ liệu hóa đơn và thông tin xác thực
Hậu điều kiện	Người dùng có thể tra cứu và xác thực hóa đơn thông qua giao diện hệ thống quản lý hóa đơn.

III. TẦN SUẤT GIAO DỊCH

a) Quản trị viên

SỐ THỨ TỰ	CHỨC NĂNG	TẦN SUẤT
QTV1	Thêm thông tin chuyến xe	1-60/ngày
QTV2	Xoá thông tin chuyến xe	0-10/ngày
QTV3	Cập nhật giá vé	0-60/ngày
QTV4	Cập nhật thời gian hành trình	1-120/ngày
QTV5	Thêm thông tin loại xe	1-60/ngày
QTV6	Xoá thông tin loại xe	0-10/ngày
QTV7	Cập nhật lịch sử đặt vé	60-300/ngày
QTV8	Cập nhật thông tin tài khoản	0-10/ngày
QTV9	Thêm tài khoản	0-10/ngày
QTV10	Xoá tài khoản	0-3/ngày
QTV11	Thêm thông tin tin tức liên quan Futabus	1-2/ngày
QTV12	Xoá thông tin tin tức Futabus	0-1/ngày
QTV13	Thêm các chương trình khuyến mãi	1-2/tháng
QTV14	Cập nhật các chương trình khuyến mãi	1-2/tháng
QTV15	Xoá các chương trình khuyến mãi	0-1/tháng
QTV16	Cập nhật hoá đơn khách hàng	60-300/ngày

b) Nhân viên

SỐ THỨ TỰ	CHỨC NĂNG	TẦN SUẤT
NV1	Xem lịch sử đặt vé	20-30/giờ
NV2	Xem danh sách đặt vé	20-30/giờ
NV3	Xem thông tin tài khoản	0-20/ngày
NV4	Xem hồ sơ thanh toán của khách hàng	30-40/giờ
NV5	Cập nhập liên hệ với khách hàng	30-40/giờ
NV6	Xem lịch trình của khách hàng	20-30/giờ
NV7	Xem thông tin liên hệ của khách hàng	20-30/giờ
NV8	Xem đánh giá của khách hàng	10-20/giờ
NV9	Xem hoá đơn khách hàng	30-40/giờ
NV10	Xem danh sách khách hàng được áp dụng khuyến mãi	30-40/giờ

c) Khách hàng

SỐ THỨ TỰ	CHỨC NĂNG	TẦN SUẤT
KH1	Đăng ký tài khoản	10-50/ngày
KH2	Xem thông tin chuyến xe	20-100/ngày
KH3	Xem loại xe	20-100/ngày
KH4	Xem lịch sử đặt vé	30-60/ngày
KH5	Xem tình trạng đặt vé	30-80/ngày
KH6	Xem thông tin hoá đơn thanh toán	30-80/ngày
KH7	Xem lịch sử thanh toán	20-60/ngày
KH8	Xem thông tin liên quan tới Futabus	0-30/ngày

KH9	Xem các chương trình khuyến mãi	1-300/tháng
KH10	Đặt vé xe	30-80/ngày
KH11	Thanh toán	30-80/ngày
KH12	Liên hệ nhà xe	20-100/ngày

d) Quản trị viên, Nhân viên

SỐ THỨ TỰ	CHỨC NĂNG	TẦN SUẤT
Login	Đăng nhập	1-20/ngày

✓ Các tần suất giao dịch thường xuyên**Quản trị viên**

- Thêm thông tin chuyến xe
- Cập nhật thời gian hành trình
- Cập nhật lịch sử đặt vé
- Cập nhật hoá đơn khách hàng

Nhân viên

- Xem danh sách đặt vé
- Xem hoá đơn khách hàng
- Xem thông tin liên hệ của khách hàng
- Xem lịch trình của khách hàng

Khách hàng

- Đặt vé xe
- Thanh toán
- Xem thông tin chuyến xe
- Liên hệ nhà xe


IV. GUI (BẢN THỬ)

➤ Đăng nhập/đăng ký



➤ Trang chủ

Trang chủ

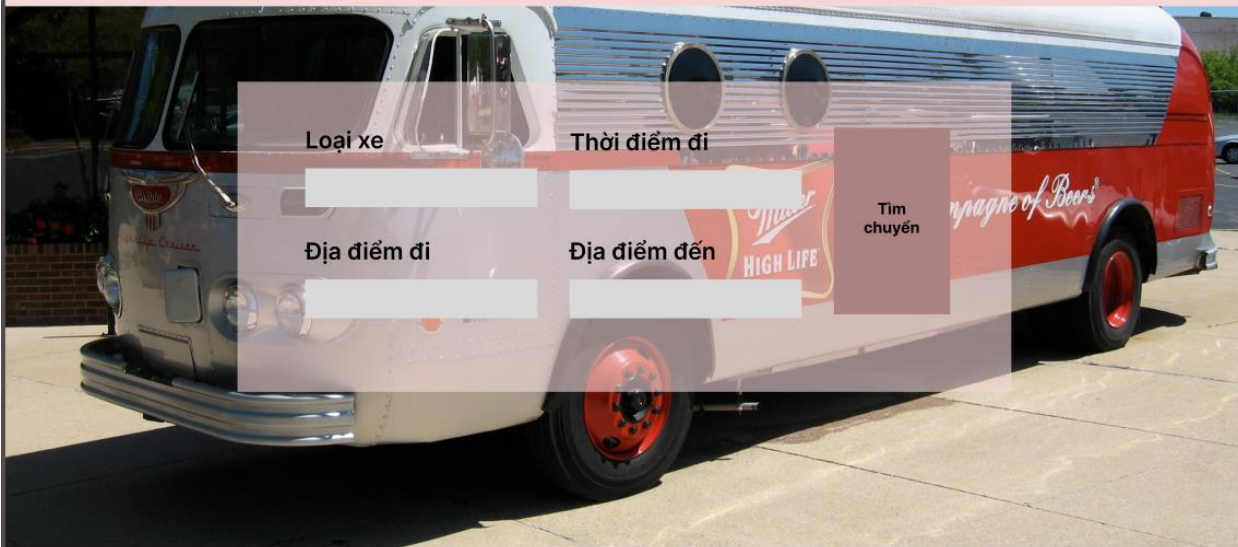
 **FUTA BUS**

ĐẶT VÉ ĐĂNG NHẬP ĐĂNG KÝ

Loại xe Thời điểm đi

Địa điểm đi Địa điểm đến

Tìm chuyến



➤ Đặt vé

Đặt vé

 **FUTA BUS**

TÌM CHUYẾN ĐẶT VÉ VÉ ĐÃ ĐẶT ĐĂNG XUẤT

Vé đã chọn

Loại xe Số lượng

Thời điểm đi Thời điểm đến

Địa điểm đi Địa điểm đến

Bạn hãy xác nhận thông tin chính xác và đặt vé

ĐẶT VÉ



➤ Tìm chuyến xe

Tìm chuyến xe

FUTA BUS

TÌM CHUYẾN ĐẶT XE VÉ ĐÃ ĐẶT ĐĂNG XUẤT

Loại xe
 Thời điểm đi
 Thời điểm đến
 Địa điểm đi
 Địa điểm đến

TÌM KIẾM NÂNG CAO

TÌM CHUYẾN

Loại xe	Thời điểm đi	Thời điểm đến	Địa điểm đi	Địa điểm đến	...
Bus 24 chỗ	09:00 24/02/2023	17:00 24/02/2023	123 ABC, P.x, Tp. y	345, P.z, Tp. t	...

➤ Tìm kiếm nâng cao

Tìm kiếm nâng cao

FUTA BUS

TÌM CHUYẾN ĐẶT XE VÉ ĐÃ ĐẶT ĐĂNG XUẤT


Loại xe
 Thời điểm đi
 Thời điểm đến
 Địa điểm đi
 Địa điểm đến
 Khoảng cách
 Giá tiền
 Trạm nghỉ chân

TÌM CHUYẾN

Loại xe	Thời điểm đi	Thời điểm đến	Địa điểm đi	Địa điểm đến	...
Bus 24 chỗ	09:00 24/02/2023	17:00 24/02/2023	123 ABC, P.x, Tp. y	345, P.z, Tp. t	...

➤ Hoá đơn

Hóa đơn

 **FUTA BUS**

TÌM CHUYẾN ĐẶT XE VÉ ĐÃ ĐẶT ĐĂNG XUẤT

Số hóa đơn		
Vé đã đặt	Loại xe	Số lượng
Thời điểm đi	Thời điểm đến	Hình thức thanh toán
Địa điểm đi	Địa điểm đến	Tổng giá tiền

➤ Thanh toán

Thanh toán

 **FUTA BUS**

TÌM CHUYẾN ĐẶT XE VÉ ĐÃ ĐẶT ĐĂNG XUẤT

Vé đã chọn	Loại xe	Số lượng
Hình thức thanh toán	Tổng giá tiền	
Thời điểm đi	Thời điểm đến	
Địa điểm đi	Địa điểm đến	

Bạn hãy xác nhận lần cuối thông tin chính xác và thanh toán vé đã chọn

THANH TOÁN

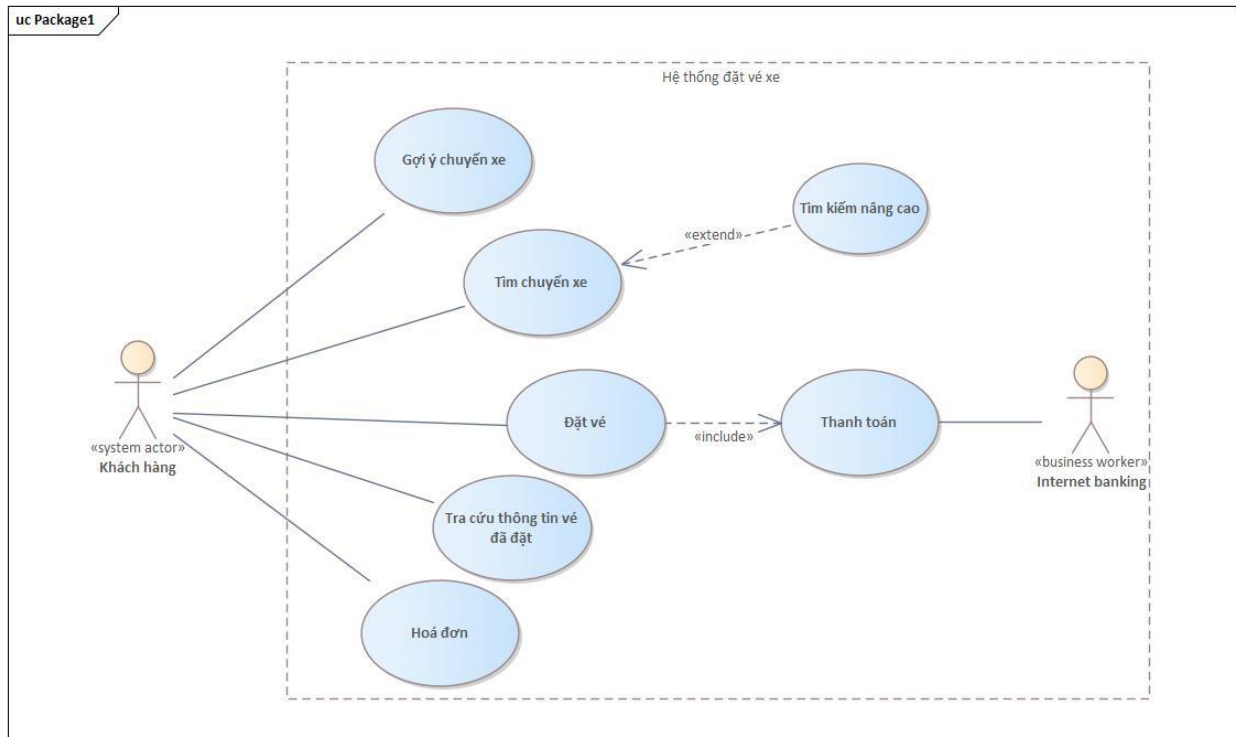
➤ **Danh sách vé đã đặt****V. CÁC THÀNH PHẦN DỮ LIỆU TRONG FUTA BUS**

1. **Lịch trình chạy xe:** Đây là thông tin về các chuyến xe, thời gian khởi hành, điểm đến, và lịch trình dự kiến. Lịch trình chạy xe giúp quản lý việc vận hành và đảm bảo tính chính xác của dịch vụ.
Dữ liệu có trong Use case: **Tìm chuyến xe**
2. **Thông tin về tài xế:** Bao gồm tên, số điện thoại, và thông tin liên quan đến tài xế. Điều này quan trọng để theo dõi và quản lý tài xế trong quá trình hoạt động.
Dữ liệu có trong Use case: **Tìm chuyến xe**
3. **Dữ liệu về vé và giá vé:** Bao gồm thông tin về loại vé, giá vé, số lượng vé đã bán, và các chương trình khuyến mãi. Điều này giúp quản lý doanh thu và tối ưu hóa giá vé.
Dữ liệu có trong Use case: **Đặt vé, Tra cứu thông tin vé đã đặt, Thanh toán, Hoá đơn**
4. **Thông tin về bến xe và trạm trung chuyển:** Bao gồm địa chỉ, tên, và thông tin liên quan đến các bến xe và trạm trung chuyển. Điều này hỗ trợ quản lý và điều phối hoạt động vận chuyển.
Dữ liệu có trong Use case: **Tìm chuyến đi, Tìm kiếm nâng cao**
5. **Thông tin khách hàng:** Thông tin này bao gồm các thông tin cá nhân như tên, giới tính, tuổi, email, Điều này giúp quản lý khách hàng dễ dàng cũng như là tệp khách hàng tiềm năng, mức độ sử dụng dịch vụ.
Dữ liệu có trong Use case: **Không có**

6. **Thông tin về loại xe:** Cung cấp nhiều loại xe, bao gồm các loại xe ghế ngồi, xe giường nằm, xe limosine. Mỗi loại xe có thiết kế và tiện nghi khác nhau để phục vụ nhu cầu khách hàng
Dữ liệu có trong Use case: **Không có**

GIẢI ĐOẠN 2

I. USE CASE HỆ THỐNG (UPDATE)



➤ **Mô tả chi tiết về quy trình tương tác giữa các Use case:**

1. **Tìm chuyến xe:** Khách hàng bắt đầu bằng việc tìm kiếm chuyến xe phù hợp. Trong quá trình này, họ có thể sử dụng chức năng “Tìm kiếm nâng cao” để lọc kết quả tìm kiếm theo các tiêu chí cụ thể như giờ đi, loại xe, hàng ghế, tầng, v.v. Và cũng có thể sử dụng “Tra cứu thông tin chuyến xe” để xem trước chuyến xe.
2. **Đặt vé:** Sau khi tìm thấy chuyến xe phù hợp, khách hàng tiếp tục đến use case “Đặt vé”. Tại đây, họ chọn số lượng vé, cung cấp thông tin liên lạc và thông tin đón trả, sau đó đặt vé.
3. **Thanh toán:** Sau khi đặt vé, khách hàng tiếp tục đến use case “Thanh toán”. Tại đây, họ chọn hình thức thanh toán (thẻ tín dụng, ví điện tử, v.v.) và cung cấp thông tin thanh toán. Hệ thống sau đó xác nhận thông tin thanh toán và hoàn tất giao dịch. Trong quá trình này, hệ thống có thể tương tác với “Internet Banking” để xử lý giao dịch thanh toán.
4. **Hóa đơn:** Sau khi thanh toán thành công, hệ thống tạo hóa đơn cho khách hàng. Khách hàng có thể tra cứu và xác thực hóa đơn thông qua hệ thống quản lý hóa đơn.
5. **Tra cứu thông tin vé đã đặt:** Cuối cùng, sau khi đã đặt và thanh toán vé, khách hàng có thể tra cứu thông tin về vé đã đặt.

6. Gợi ý chuyển xe: Nếu người dùng chưa biết tìm chuyến xe như thế nào để phù hợp, hệ thống sẽ gợi ý các chuyến xe thường xuyên được đặt và sử dụng nhiều nhất, người dùng có thể tham khảo các chuyến xe đó để cân nhắc các lựa chọn tốt hơn.

*Ghi chú: **Đăng nhập/ đăng ký** không phải là 1 Use case bởi không cần đăng nhập thì hệ thống vẫn cho đặt vé, cùng với đó là Use case này không có tính hoàn thiện, không đem lại giá trị như những Use case trên*

II. ĐẶC TẢ USE CASE HỆ THỐNG (UPDATE)

Tên use case	Tìm chuyến xe
Tóm tắt	Khách hàng tìm kiếm và xem thông tin chi tiết về các chuyến xe có sẵn.
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào hệ thống tìm chuyến xe. 2. Hệ thống hiển thị danh sách các chuyến xe có sẵn 3. Khách hàng chọn chuyến xe và nhập thông tin tìm kiếm (điểm đi, điểm đến, ngày giờ). 4. Hệ thống tìm kiếm và hiển thị danh sách chuyến xe phù hợp với thông tin tìm kiếm của khách hàng, bao gồm thông tin về nơi đi, nơi đến, thời gian khởi hành, giá vé và số ghế còn trống. 5. Khách hàng xem chi tiết về một chuyến xe bằng cách chọn chuyến xe trong danh sách. 6. Hệ thống hiển thị thông tin chi tiết về chuyến xe, bao gồm lịch trình, tài xế, loại xe và các thông tin khác.
Dòng sự kiện phụ	<p>A1. Tại bước 4, nếu không có chuyến xe phù hợp,</p> <ul style="list-style-type: none"> - Hệ thống sẽ không xuất ra các chuyến và thông báo cho khách hàng - Không có các bước sau
Điều kiện tiên quyết	Hệ thống phải có danh sách các chuyến xe có sẵn và thông tin chi tiết về các chuyến xe đó.
Hậu điều kiện	Danh sách chuyến xe phù hợp được hiển thị cho khách hàng. Khách hàng có thể xem thông tin chi tiết về chuyến xe và tiếp tục các hành động khác như đặt vé.

Tên use case	Tìm kiếm nâng cao
Tóm tắt	Khách hàng tìm kiếm chuyến xe với các tiêu chí tùy chọn để tìm kiếm chuyến xe theo nhu cầu cụ thể.
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào hệ thống tìm chuyến xe 2. Khách hàng chọn tùy chọn tìm kiếm nâng cao. 3. Khách hàng cung cấp các tiêu chí cho bộ lọc tìm kiếm như giờ đi , loại xe , hàng ghế , tầng 4. Hệ thống tìm kiếm và hiển thị danh sách chuyến xe phù hợp với bộ lọc tìm kiếm của khách hàng, 5. Khách hàng xem chi tiết về một chuyến xe bằng cách chọn chuyến xe trong danh sách. 6. Hệ thống hiển thị thông tin chi tiết về chuyến xe, bao gồm lịch trình, tài xế, loại xe và các thông tin khác.
Dòng sự kiện phụ	<p>A1. Tại bước 4, nếu không có bộ lọc được cung cấp</p> <ul style="list-style-type: none"> - Hệ thống sẽ hiển thị tất cả các chuyến xe có sẵn mà không áp dụng bộ lọc tìm kiếm. <p>A2. Tại bước 4, nếu không có chuyến xe nào phù hợp với bộ lọc, hệ thống sẽ thông báo cho khách hàng.</p>
Điều kiện tiên quyết	Hệ thống phải có danh sách các chuyến xe có sẵn và thông tin chi tiết về các chuyến xe.
Hậu điều kiện	Danh sách chuyến xe phù hợp với tìm kiếm nâng cao được hiển thị cho khách hàng.

Tên use case	Đặt vé
Tóm tắt	Sau khi khách hàng sử dụng Tìm chuyến xe phù hợp với nhu cầu và giờ giấc, khách hàng sẽ tiếp tục quá trình là đặt vé của chuyến xe đó.
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Sau khi chọn chuyến xe, hệ thống hiển thị giao diện đặt vé bao gồm: Chọn ghế - Thông tin khách hàng – Điểm đón/trả - Thanh toán 2. Khách hàng chọn số lượng ghế phù hợp với nhu cầu 3. Khách hàng điền thông tin khách hàng cần thiết và đồng ý điều khoản 4. Khách hàng chọn điểm đón/trả được hiển thị sẵn trên hệ thống 5. Khách hàng có thể chọn thanh toán với số tiền đã hiển thị trên hệ thống sau khi điền các thông tin cần thiết
Dòng sự kiện phụ	<p>A2: Tại bước 2, nếu khách hàng không chọn ghế</p> <ul style="list-style-type: none"> - Hệ thống sẽ không thanh toán được (ít nhất 1 ghế và tối đa 5 ghế) <p>A3: Tại bước 3, nếu khách hàng không điền đầy đủ thông tin hoặc không đồng ý điều khoản</p> <ul style="list-style-type: none"> - Hệ thống sẽ không thanh toán được
Điều kiện tiên quyết	Khách hàng phải điền đầy đủ thông tin trên trang đó và đồng ý điều khoản để có thể sang bước tiếp theo
Hậu điều kiện	Hệ thống sẽ lưu thông tin của trang Đặt vé và đưa khách hàng tới trang Thanh toán

Tên use case	Thanh toán
Tóm tắt	Sau khi khách hàng điền đầy đủ thông tin cho vé và bấm Thanh toán, khách hàng sẽ tiếp tục quá trình là thanh toán bằng các phương thức khác nhau.
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị các thông tin đã xác nhận và các phương thức thanh toán khác nhau 2. Khách hàng chọn phương thức thanh toán 3. Khách hàng quét mã QR để thanh toán
Dòng sự kiện phụ	<p>A3.1: Tại bước 3, nếu khách hàng không thanh toán trong thời gian chờ giữ ghế</p> <ul style="list-style-type: none"> - Hệ thống sẽ huỷ ghế - Quay lại Usecase đặt vé <p>A3.2: Tại bước 3, nếu khách hàng có nhu cầu chuyển đổi vé thì chỉ có thể chuyển 1 lần duy nhất</p> <p>A3.3: Tại bước 3, nếu khách hàng có nhu cầu thay đổi hoặc huỷ vé đã thanh toán:</p> <ul style="list-style-type: none"> - Liên hệ với tổng đài 1900 6067 hoặc quầy vé - Vé huỷ chỉ trả lại chi phí từ 10% - 30% tùy thuộc thời gian huỷ vé so với giờ khởi hành ghi trên vé và số lượng vé cá nhân/tập thể áp dụng theo các quy định hiện hành
Điều kiện tiên quyết	Khách hàng phải thanh toán để có vé
Hậu điều kiện	Hệ thống xác nhận thanh toán và gửi vé qua mail mà khách hàng đã điền thông tin từ trước

Tên use case	Tra cứu thông tin vé
Tóm tắt	Có thể thực hiện sau khi UC Thanh toán đã hoàn thành, khách hàng tra cứu thông tin vé đã được gửi
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none">1. Hệ thống gửi thông tin vé qua mail mà khách hàng đã cho trước2. Khách hàng vào mail kiểm tra thông tin vé
Dòng sự kiện phụ	A2: Tại bước 2, nếu khách hàng kiểm tra thông tin vé không phù hợp hoặc sai sót <ul style="list-style-type: none">- Báo lên hệ thống hoặc CSKH để giải quyết
Điều kiện tiên quyết	Không có
Hậu điều kiện	Khách hàng có vé trên mail và dùng vé đó để sử dụng khi thời điểm khởi hành

Tên use case	Hóa đơn
Tóm tắt	Hệ thống cho phép người dùng tra cứu và xác thực hóa đơn
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào hệ thống quản lý hóa đơn. 2. Hệ thống hiển thị giao diện quản lý hóa đơn cho người dùng. 3. Khách hàng chọn chức năng "Tra cứu hóa đơn". 4. Hệ thống yêu cầu người dùng nhập thông tin cần tra cứu 5. Khách hàng cung cấp thông tin tra cứu. 6. Hệ thống truy vấn cơ sở dữ liệu và hiển thị danh sách hóa đơn phù hợp với thông tin tra cứu. 7. Khách hàng có thể chọn một hóa đơn từ danh sách để xem chi tiết.
Dòng sự kiện thay thế	<ol style="list-style-type: none"> 1a. Khách hàng truy cập vào hệ thống quản lý hóa đơn. 2a. Hệ thống hiển thị giao diện quản lý hóa đơn cho người dùng. 3a. Khách hàng chọn chức năng "Xác thực hóa đơn". 4a. Hệ thống yêu cầu người dùng nhập thông tin hóa đơn cần xác thực 5a. Khách hàng cung cấp thông tin xác thực. 6a. Hệ thống kiểm tra thông tin xác thực và cập nhật trạng thái xác thực của hóa đơn tương ứng. 7a. Hệ thống thông báo cho người dùng kết quả xác thực thành công hoặc thất bại.
Dòng sự kiện phụ	<p>A1. Tại bước 6, nếu không tìm thấy hóa đơn phù hợp</p> <ul style="list-style-type: none"> - Hệ thống sẽ thông báo cho người dùng.

	<ul style="list-style-type: none">- Không có các bước sau A2. Tại bước 7a , nếu thông tin xác thực không hợp lệ <ul style="list-style-type: none">- Hệ thống sẽ thông báo lỗi cho người dùng và yêu cầu người dùng nhập lại thông tin.
Điều kiện tiên quyết	Khách hàng đã hoàn thành thanh toán và có mã vé chuyển xe. Hệ thống phải có cơ sở dữ liệu hóa đơn và thông tin xác thực
Hậu điều kiện	Người dùng có thể tra cứu và xác thực hóa đơn thông qua giao diện hệ thống quản lý hóa đơn.

Tên use case	Gợi ý chuyển xe
Tóm tắt	Hệ thống cho phép người dùng tham khảo các chuyến xe được thường xuyên đặt và sử dụng dịch vụ tốt nhất
Tác nhân	Khách hàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào Hệ thống tìm chuyến xe. 2. Hệ thống hiển thị danh sách các chuyến xe thường xuyên đặt nhiều nhất 3. Khách hàng có thể truy cập các chuyến xe đó và tiếp tục use case Đặt vé
Dòng sự kiện thay thế	Không có
Dòng sự kiện phụ	Không có
Điều kiện tiên quyết	Hệ thống phải có danh sách các chuyến xe có sẵn và lọc ra các chuyến xe được đặt nhiều nhất để gợi ý chính xác.
Hậu điều kiện	Người dùng có thể tham khảo các chuyến xe gợi ý để cân nhắc chuyến xe phù hợp với nhu cầu

III. LẬP BẢNG PHÂN TÍCH ƯU/NHUỢC ĐIỂM CỦA TỪNG LOẠI NOSQL

a) Bảng ưu điểm của các loại NoSQL

➤ Column ưu điểm

Ưu điểm	Column Family
Xử lý dữ liệu	Cơ sở dữ liệu column-family có thể xử lý cả dữ liệu cấu trúc và phi cấu trúc.
Linh hoạt	Các hàng trong cùng column-family có thể có số lượng cột khác nhau, và có thể thêm các cột mới vào một hàng nếu cần. Điều này tránh việc điền các giá trị mặc định cho các cột mới cho các hàng hiện có, như xảy ra với cơ sở dữ liệu quan hệ.
Tốc độ	Thông tin của các cột liên quan trong cùng column-family được lưu trữ cùng nhau trên đĩa, giúp ghi và truy xuất thông tin nhanh hơn nếu dữ liệu được lưu trữ ở các phần khác nhau của một đĩa.
Khả năng mở rộng	Cơ sở dữ liệu column-family mở rộng theo chiều ngang bằng cách chia sẻ trên nhiều máy chủ.
Xử lý lượng dữ liệu lớn	Cơ sở dữ liệu column-family được thiết kế để xử lý lượng dữ liệu lớn. Điều này là do tốc độ và khả năng mở rộng theo chiều ngang, cũng như khả năng nén dữ liệu hiệu quả.
Không có khóa ngoại	Cơ sở dữ liệu column-family không sử dụng khóa ngoại, điều này giúp tăng tốc độ truy vấn nhưng cũng giảm tính linh hoạt trong việc liên kết dữ liệu giữa các bảng.
Truy vấn	Truy cập chỉ những cột cần thiết giúp truy vấn nhanh hơn. Cơ sở dữ liệu cột cũng rất hiệu quả khi thực hiện truy vấn tổng hợp trên lượng dữ liệu lớn.
Nén dữ liệu	Cơ sở dữ liệu column-family có khả năng nén dữ liệu cao, giảm chi phí lưu trữ.

Chỉ mục	Cơ sở dữ liệu column-family sử dụng chỉ mục cấp cột, được tối ưu hóa cho việc truy xuất dữ liệu dựa trên cột, có thể cải thiện hiệu suất truy vấn.
Phân tích	Cơ sở dữ liệu column-family phù hợp cho các tác vụ phân tích/BI với các tổng hợp trên các tập dữ liệu lớn.

➤ Document ưu điểm

Ưu điểm	Document
Tính linh hoạt	Cho phép lưu trữ dữ liệu phi cấu trúc, linh hoạt hơn so với các cơ sở dữ liệu quan hệ truyền thống. Điều này giúp bạn dễ dàng lưu trữ các loại dữ liệu khác nhau, chẳng hạn như JSON, XML, và BSON.
Khả năng mở rộng	Có thể mở rộng dễ dàng để đáp ứng nhu cầu dữ liệu lớn. Dữ liệu có thể được phân tán theo chiều ngang trên nhiều máy chủ, giúp cải thiện hiệu suất và khả năng truy cập
Hiệu suất cao	Tối ưu hoá cho các truy vấn dữ liệu phi cấu trúc, giúp truy vấn dữ liệu hiệu quả và nhanh chóng dù có lượng lớn dữ liệu
Dễ sử dụng	Document store cung cấp API đơn giản và dễ sử dụng, giúp bạn dễ dàng phát triển ứng dụng
Truy vấn nhanh	Tối ưu hóa cho các truy vấn dữ liệu phi cấu trúc. Điều này giúp bạn có thể truy vấn dữ liệu của mình một cách nhanh chóng và hiệu quả, ngay cả khi bạn có một lượng lớn dữ liệu
Nhúng dữ liệu	Cho phép lưu trữ các dữ liệu liên quan vào 1 document. Qua đó có thể tránh được quá trình phải join dữ liệu (ghép 2 bảng vào nhau)
Tính đa hình	Khả năng lưu trữ các loại dữ liệu khác nhau như khác cấu trúc hoặc khác thuộc tính trong cùng một collection, tăng tính linh hoạt, giảm thiểu phức tạp

➤ Graph ưu điểm

Ưu điểm	Graph Store
Mô hình hóa mối quan hệ phức tạp	Cho phép mô hình hóa mối quan hệ giữa các đối tượng (nút) bằng các cạnh và thuộc tính của chúng. Điều này rất hữu ích khi cần biểu diễn các mối quan hệ phức tạp giữa các đối tượng trong dữ liệu.
Truy vấn hiệu suất cao với dữ liệu liên quan nhiều mối quan hệ	Truy vấn những mối quan hệ liên quan nhiều mức độ có thể được thực hiện hiệu quả hơn so với các cơ sở dữ liệu quan hệ truyền thống.
Độ linh hoạt cao trong thiết kế schema	Không giống như các cơ sở dữ liệu quan hệ có schema cố định, các cơ sở dữ liệu đồ thị thường linh hoạt hơn trong việc thay đổi mô hình dữ liệu vì chúng không yêu cầu schema cứng nhắc.
Phân tích mạnh mẽ về mối quan hệ	Thường hỗ trợ các thuật toán phân tích đồ thị mạnh mẽ như tìm kiếm theo độ sâu, độ rộng, hoặc tính toán các thuật toán đồ thị phức tạp (ví dụ như tìm đường đi ngắn nhất).
Khả năng mở rộng	Có thể dễ dàng mở rộng thêm mới các loại đối tượng và mối quan hệ vào đồ thị mà không cần phải thay đổi cấu trúc cơ sở dữ liệu quá nhiều.
Khả năng truy vấn đa dạng	Với cấu trúc đồ thị, bạn có thể thực hiện nhiều loại truy vấn khác nhau, bao gồm tìm kiếm đường đi, phân tích mạng lưới, xác định các mẫu trong dữ liệu.

➤ Key-Value ưu điểm

Ưu điểm	Key-Value
Đơn giản và dễ hiểu	Cấu trúc dữ liệu key-value rất đơn giản và dễ sử dụng. Mỗi giá trị được lưu trữ dưới dạng cặp key-value.
Truy xuất nhanh	Với key, bạn có thể truy cập dữ liệu một cách nhanh chóng. Cấu trúc này thường có hiệu suất tốt trong việc truy xuất dữ liệu.
Phù hợp cho việc lưu trữ dữ liệu đơn giản	Key-value stores thích hợp cho việc lưu trữ dữ liệu đơn giản như cài đặt cache, lưu trữ phi tập trung, hoặc quản lý session.
Khả năng mở rộng dễ dàng	Hầu hết các hệ thống key-value stores có khả năng mở rộng tốt, cho phép bạn mở rộng lưu trữ mà không gặp phải nhiều vấn đề hiệu suất.
Hiệu suất cao	Với cấu trúc đơn giản và khả năng truy cập nhanh, key-value stores thường có hiệu suất cao, phù hợp cho các ứng dụng có yêu cầu về hiệu suất.
Dễ dàng tích hợp với các hệ thống khác	Key-value stores thường được thiết kế để dễ dàng tích hợp với các hệ thống khác, giúp cho việc phát triển ứng dụng và mở rộng hệ thống trở nên dễ dàng hơn.

b) Bảng ưu điểm của các CSDL NoSQL**➤ Cassandra ưu điểm**

Ưu điểm	Cassandra
Dấu thời gian cho mỗi cột	Trong Cassandra, mỗi cột đều có một dấu thời gian đi kèm, điều này hữu ích để giải quyết các xung đột ghi và xử lý dữ liệu lỗi thời.
Chuyển giao nhiệm vụ khi một nút bị hỏng	Khi một nút bị hỏng, nó sẽ không ảnh hưởng đến việc đọc hoặc ghi, các nhiệm vụ sẽ được chuyển giao cho một nút khác trong cụm.
Kiến trúc	Cassandra được xây dựng như một cơ sở dữ liệu phân tán ngang hàng nơi tất cả các nút đều quan trọng như nhau.
Tính sẵn sàng cao	Cassandra được thiết kế để có tính sẵn sàng cao. Điều này là do không có node chủ và bất kỳ node nào đều có thể chấp nhận đọc và ghi.
Nhất quán	Cassandra có thể được cấu hình để có nhất quán cuối cùng hoặc nhất quán mạnh vì nó hỗ trợ những gì được gọi là nhất quán có thể điều chỉnh.
Hiệu suất cao	Cassandra được thiết kế để mang lại hiệu suất cao, độ trễ thấp và thông lượng cao. Nó được tối ưu hóa cho khối lượng công việc ghi nhiều và có thể xử lý hàng triệu lần ghi mỗi giây.
Khả năng mở rộng	Cassandra hỗ trợ khả năng mở rộng do kiến trúc phân tán của nó.

➤ MongoDB ưu điểm

Ưu điểm	MongoDB
Mô hình dữ liệu linh hoạt	Không yêu cầu schema cố định, cho phép lưu trữ dữ liệu có cấu trúc phức tạp và linh hoạt như JSON, BSON. Phù hợp với các ứng dụng có yêu cầu dữ liệu đa dạng và thay đổi linh hoạt
Khả năng mở rộng	Có thể mở rộng theo chiều ngang bằng cách thêm các nút máy chủ vào cụm MongoDB để tăng khả năng xử lý và lưu trữ, cung cấp các tính năng như Sharding và Replication để đảm bảo sự tin cậy và hiệu suất cao
Truy vấn	Truy vấn dữ liệu phi cấu trúc nhanh chóng nhờ cấu trúc dữ liệu và thuật toán tối ưu. Hỗ trợ indexing cho phép truy vấn hiệu quả hơn
Dễ quản lý	Có công cụ quản lý như MongoDB Compass, cho phép quản lý cơ sở dữ liệu, xem và chỉnh sửa dữ liệu, tạo truy vấn.
Tính sẵn sàng	Cung cấp khả năng Replication đảm bảo tính sẵn sàng cao. Khả năng phục hồi dữ liệu tốt trong trường hợp lỗi xảy ra
Hiệu suất cao	Có thể lưu document trong document, giúp giảm thiểu việc truy xuất và ghi dữ liệu từ nhiều bảng, cải thiện tốc độ truy vấn và thời gian phản hồi
Tích hợp	Cung cấp API phong phú cho các ngôn ngữ lập trình như Java, Python, NodeJS, Giúp việc phát triển ứng dụng dễ dàng và thuận tiện

➤ Neo4j ưu điểm

Ưu điểm	Neo4j
Tốc độ truy vấn cao	Neo4j cung cấp các thuật toán tối ưu để truy vấn dữ liệu đồ thị, điều này giúp tăng tốc quá trình truy vấn so với một số cơ sở dữ liệu quan hệ truyền thống.
Dễ dàng mở rộng	Có khả năng mở rộng tốt với việc thêm các nút và mối quan hệ mới mà không ảnh hưởng nhiều đến hiệu suất của hệ thống.
Hỗ trợ các tính năng ACID	Có khả năng đảm bảo tính toàn vẹn của dữ liệu với các tính năng ACID
Tính toán đồ thị	Neo4j cung cấp các tính năng tính toán trên đồ thị cho phép phân tích và khám phá cấu trúc đồ thị một cách hiệu quả

➤ Redis ưu điểm

Ưu điểm	Redis
Hiệu suất cao	Redis là một cơ sở dữ liệu trong bộ nhớ, nhanh chóng và có thể đạt được thời gian phản hồi rất thấp cho các hoạt động đọc và ghi.
Dễ sử dụng	Redis có cú pháp đơn giản và dễ sử dụng, hỗ trợ một loạt các cấu trúc dữ liệu cơ bản như chuỗi, danh sách, bản đồ và tập hợp.
Hỗ trợ đa dạng cấu trúc dữ liệu	Redis hỗ trợ nhiều loại cấu trúc dữ liệu, cho phép lưu trữ và truy xuất dữ liệu theo nhiều cách khác nhau, phù hợp với các nhu cầu khác nhau.
Khả năng mở rộng	Redis có khả năng mở rộng tốt, có thể được mở rộng theo cách dọc (dựa trên kích thước máy chủ) hoặc theo cách ngang (dựa trên phân phối).
Hỗ trợ cơ sở dữ liệu chính trị	Redis hỗ trợ các kiểu dữ liệu phức tạp như bản đồ, danh sách và tập hợp, giúp giải quyết nhiều vấn đề khó khăn trong lập trình ứng dụng.
Sự linh hoạt trong việc triển khai	Redis có thể triển khai trên nhiều môi trường khác nhau, từ máy tính để bàn đến các hệ thống phân tán và điện toán đám mây.
Cộng đồng hỗ trợ lớn	Redis có một cộng đồng rộng lớn, cung cấp tài liệu, hướng dẫn và hỗ trợ cho người dùng, giúp giải quyết các vấn đề và tìm hiểu sâu hơn về nền tảng.

c) Bảng nhược điểm của các loại NoSQL**➤ Column nhược điểm**

Nhược điểm	Column Family
Mô hình dữ liệu phức tạp	Thiết kế column-family đòi hỏi kế hoạch cẩn thận. Việc chọn các cột và khóa phù hợp là quan trọng để truy vấn hiệu quả. Nếu sửa đổi các lược đồ của column-family, điều này có thể tốn kém.
Khả năng truy vấn hạn chế	Khó khăn với các truy vấn phức tạp do thiếu các phép JOIN và tổng hợp.
Nhất quán cuối cùng	Hầu hết cơ sở dữ liệu column-family ưu tiên khả năng sẵn có và chịu lỗi hơn so với nhất quán nghiêm ngặt, dẫn đến nhất quán cuối cùng.
Quá tải lưu trữ	Lưu trữ dữ liệu thừa (các hàng với các cột biến đổi) có thể dẫn đến hiệu quả lưu trữ không tốt.
Hỗ trợ ACID hạn chế	Cơ sở dữ liệu column-family hy sinh giao dịch ACID để mở rộng.
Không phù hợp cho OLTP	Chi phí phát sinh và khả năng không thực hiện nhanh các truy vấn nhanh.
Cập nhật	Việc sửa đổi và xóa dữ liệu chậm hơn và phức tạp hơn.

➤ Document nhược điểm

Nhược điểm	Document
Truy vấn phức tạp hạn chế	Truy vấn theo các quan hệ phức tạp giữa các tài liệu có thể không được tối ưu, không cung cấp khả năng tham chiếu giữa các document, giảm tính nhất quán và hiệu quả của truy vấn.
Không hỗ trợ ACID	Điều này gây khó khăn trong việc đảm bảo dữ liệu nhất quán và toàn vẹn dữ liệu
Phục hồi và backup	Quy trình này thực hiện phức tạp hơn CSDL quan hệ
Thay đổi cấu trúc	Dữ liệu trong Document có sự thay đổi cấu trúc sẽ gây khó khăn trong việc quản lý và duy trì dữ liệu, cấu trúc thay đổi và cập nhật Document trở nên phức tạp và chi phí, tốn thời gian

➤ Graph nhược điểm

Nhược điểm	Graph Store
Khả năng mở rộng theo chiều dọc hạn chế	Có thể gặp khó khăn hơn trong việc mở rộng theo chiều dọc (vertical scaling), đặc biệt khi đối mặt với dữ liệu lớn và nhu cầu tăng cường hiệu suất.
Khó khăn trong mô hình hóa dữ liệu phi cấu trúc	Các cơ sở dữ liệu đồ thị tốt hơn khi sử dụng cho các mô hình dữ liệu có tính cấu trúc hoặc dễ dàng được phân tích. Trong trường hợp dữ liệu phi cấu trúc hoặc không có mô hình rõ ràng, việc quản lý và truy vấn có thể trở nên phức tạp hơn.
Khả năng lưu trữ phức tạp	Với các đồ thị lớn, một số cơ sở dữ liệu đồ thị có thể đòi hỏi nhiều tài nguyên về lưu trữ và xử lý để duy trì tính hiệu quả của hệ thống.
Chi phí lưu trữ	Vì cấu trúc đồ thị phức tạp hơn so với bảng và các mối quan hệ đơn giản trong cơ sở dữ liệu quan hệ, cơ sở dữ liệu đồ thị có thể yêu cầu nhiều không gian lưu trữ hơn. Điều này có thể là vấn đề đối với các ứng dụng lưu trữ lớn.
Khó khăn trong việc sử dụng cho các truy vấn không phải mối quan hệ	Trong các trường hợp mà mối quan hệ không phải là trọng tâm chính của truy vấn, sử dụng cơ sở dữ liệu đồ thị có thể không hiệu quả như các hệ thống cơ sở dữ liệu khác, ví dụ như các hệ thống cơ sở dữ liệu dạng cột (columnar databases) cho các truy vấn phân tích số liệu.

➤ Key-value nhược điểm

Nhược điểm	Key-Value
Hạn chế trong việc truy vấn phức tạp	Key-value stores thường không hỗ trợ các truy vấn phức tạp như các cơ sở dữ liệu quan hệ. Việc thực hiện các truy vấn phức tạp có thể phức tạp và không hiệu quả.
Thiếu tính linh hoạt trong việc tìm kiếm	Trong một số trường hợp, việc tìm kiếm dựa trên giá trị của các key có thể không hiệu quả hoặc không linh hoạt.
Khó khăn trong việc duy trì tính nhất quán	Trong một số trường hợp, việc đảm bảo tính nhất quán của dữ liệu trong key-value stores có thể phức tạp và đòi hỏi các biện pháp đặc biệt.
Giới hạn trong việc xử lý các loại dữ liệu	Key-value stores thường chỉ hỗ trợ một số loại dữ liệu cụ thể. Điều này có thể gây hạn chế trong việc xử lý các loại dữ liệu đa dạng.
Khả năng mở rộng có thể gặp hạn chế	Mặc dù nhiều key-value stores có khả năng mở rộng tốt, nhưng trong một số trường hợp, việc mở rộng có thể gặp phải hạn chế do các vấn đề về quy mô.
Không phù hợp cho các tác vụ phức tạp như ACID	Trong các ứng dụng yêu cầu tính toàn vẹn của dữ liệu như các giao dịch, key-value stores có thể không phù hợp do thiếu tính ACID.

d) Bảng nhược điểm của các CSDL NoSQL**➤ Cassandra nhược điểm**

Nhược điểm	Cassandra
Cập nhật	Việc sửa đổi và xóa dữ liệu trong Cassandra có thể chậm hơn và phức tạp hơn.
Không hỗ trợ đầy đủ các tính năng ACID	Cassandra có thể không hỗ trợ đầy đủ các tính năng ACID nhưng thay vào đó hỗ trợ các thuộc tính BASE.
Mô hình dữ liệu phức tạp	Việc thiết kế cơ sở dữ liệu Cassandra đòi hỏi kế hoạch cẩn thận. Việc chọn các cột và khóa phù hợp là quan trọng để truy vấn hiệu quả.
Khả năng truy vấn hạn chế	Khó khăn với các truy vấn phức tạp do thiếu các phép JOIN và tổng hợp, không hỗ trợ truy vấn phạm vi trên khóa phân vùng.

➤ MongoDB nhược điểm

Nhược điểm	MongoDB
Truy vấn phức tạp hạn chế	Không hỗ trợ join như CSDL quan hệ nhưng vẫn có thể sử dụng join nếu biết sử dụng. Tuy nhiên sẽ giảm hiệu suất và chi phí
Dung lượng Document	16MB là dung lượng tối đa của từng Document
Tính nhất quán khi phân vùng dữ liệu	MongoDB sử dụng eventual consistency, có thể dẫn đến tình trạng dữ liệu không nhất quán trong một số trường hợp.
Cần quản lý bộ nhớ và tài nguyên	MongoDB sử dụng bộ nhớ để lưu cache dữ liệu và một số hoạt động như sort và aggregation có thể yêu cầu một lượng lớn bộ nhớ, tránh gây ảnh hưởng đến hiệu suất và hệ thống

➤ Neo4j nhược điểm

Nhược điểm	Neo4j
Tốn nhiều tài nguyên	Để duy trì một cơ sở dữ liệu đồ thị, đôi khi có thể tốn nhiều tài nguyên so với các hệ thống cơ sở dữ liệu khác.
Khả năng xử lý dữ liệu lớn:	Khi dữ liệu trở nên rất lớn và phức tạp, hiệu suất của Neo4j có thể bị ảnh hưởng.
Khó khăn trong việc chuyển đổi từ cơ sở dữ liệu quan hệ	Nếu bạn có dữ liệu lớn đã được lưu trữ trong một hệ quản trị cơ sở dữ liệu quan hệ, việc chuyển đổi sang Neo4j có thể gặp khó khăn và tốn nhiều thời gian.

➤ Redis nhược điểm

Nhược điểm	Redis
Có hạn chế về dung lượng	Redis lưu trữ dữ liệu trong bộ nhớ, điều này có thể gây ra hạn chế về dung lượng lưu trữ so với các cơ sở dữ liệu đĩa.
Không hỗ trợ truy vấn phức tạp	Redis không cung cấp ngôn ngữ truy vấn phức tạp như SQL, nó tập trung chủ yếu vào các phương pháp truy xuất dữ liệu theo cấu trúc.
Có thể bị mất dữ liệu trong trường hợp sự cố	Vì Redis lưu trữ dữ liệu trong bộ nhớ, nó có thể mất dữ liệu khi có sự cố như mất điện hoặc lỗi phần cứng.
Khả năng mở rộng có thể gặp hạn chế	Mặc dù Redis có khả năng mở rộng tốt, nhưng việc mở rộng có thể gặp hạn chế với việc quản lý bộ nhớ hoặc vấn đề hiệu suất.
Không hỗ trợ ghi dữ liệu đĩa mặc định	Redis thường không được cấu hình để lưu trữ dữ liệu trên đĩa mặc định, điều này có thể là một vấn đề đối với một số ứng dụng yêu cầu lưu trữ dữ liệu bền vững.
Khó khăn trong việc quản lý cụm	Việc quản lý một cụm Redis phân tán có thể phức tạp hơn so với một máy chủ đơn lẻ, đặc biệt là khi cần đảm bảo tính nhất quán và hiệu suất.

IV. PHÂN TÍCH CÁC CHỨC NĂNG ĐÃ MÔ TẢ TRONG HỆ THỐNG

➤ Đặt vé

Phân tích	Mô tả
Mục đích	Tính năng Đặt vé giúp người dùng có thể tham khảo các tuyến xe phù hợp với nhu cầu cũng như xem được các chỗ ngồi còn trống trên tuyến xe đó
Phù hợp với loại dữ liệu	Dữ liệu có cấu trúc linh hoạt như Document, có thể có các cấu trúc khác nhau trong cùng Collection
Yêu cầu lưu trữ	<ol style="list-style-type: none"> Hệ thống cần lưu trữ thông tin về vé, chuyến xe, khách hàng Các thông tin quản lý bao gồm mã vé, mã chuyến xe, loại vé (một chiều, khứ hồi), số lượng vé, giá vé, tình trạng vé <p>→ cần lưu trữ các thông tin về vé, đảm bảo hệ thống có khả năng mở rộng đáp ứng nhu cầu tăng trong tương lai</p>
Khả năng mở rộng	<ol style="list-style-type: none"> Hệ thống cần được thiết kế để có thể mở rộng theo chiều ngang Thiết kế cấu trúc dữ liệu ảnh hưởng đến hiệu suất truy vấn và khả năng mở rộng
Nhu cầu truy xuất	<p>Hệ thống cần hỗ trợ truy vấn nhanh chóng và hiệu quả để lấy thông tin về tuyến xe, điểm đi, điểm đến và kiểm tra sự khả dụng của vé.</p> <ol style="list-style-type: none"> Truy vấn theo ID vé Truy vấn theo ngày tháng, tuyến đường, loại vé Truy vấn theo thông tin khách hàng Truy vấn theo tuyến xe
Trường hợp sử dụng	Khi người dùng cần đặt vé của FUTA bus, tiết kiệm thời gian, chi phí đi lại

➤ **Tìm kiếm nâng cao**

Phân tích	Mô tả
Mục đích	Cung cấp khả năng tìm kiếm chuyến xe dựa trên nhiều tiêu chí chi tiết.
Phù hợp với loại dữ liệu	Dữ liệu cần lưu trữ gồm thông tin về các chuyến xe, lịch trình, địa điểm, giá vé và các thuộc tính khác liên quan đến việc tìm kiếm.
Yêu cầu lưu trữ	<ol style="list-style-type: none"> 1. Hệ thống cần có thông tin lưu trữ từ chức năng Tìm chuyến xe 2. Các thông tin này sẽ lấy các thông tin lưu trữ và lọc ra keyword
Khả năng mở rộng	Hệ thống cần có khả năng mở rộng ngang để xử lý lượng lớn yêu cầu tìm kiếm từ người dùng đồng thời và để có thể mở rộng dữ liệu khi hệ thống phát triển.
Nhu cầu truy xuất	Dữ liệu cần có khả năng truy xuất nhanh chóng và phức tạp để hỗ trợ việc tìm kiếm của người dùng.
Trường hợp sử dụng	Trong chức năng “Tìm chuyến xe” có thể nhấn thêm nút “tìm kiếm nâng cao” để thêm vào các tiêu chí để tìm kiếm

➤ **Gợi ý chuyển xe**

Phân tích	Mô tả
Mục đích	Dựa vào các thông tin liên quan đến vé mà những người dùng khác trước đó đã đặt trước đó để có thể đưa ra những gợi ý giúp người dùng hiện tại có thể lựa chọn vé đúng với ý mình hơn
Phù hợp với loại dữ liệu	Dữ liệu có cấu trúc có khả năng đảm bảo khả năng truy xuất cao và thêm thông tin một cách linh hoạt như Graph Store
Yêu cầu lưu trữ	Hệ thống cần lưu trữ thông tin về “ID vé”, “ID chuyển xe”, “ngày khởi hành”, “ngày về”, “loại vé”, “tuyến xe”, “giá vé” với tần suất truy xuất nhanh, cao và với số lượng lớn
Khả năng mở rộng	<ol style="list-style-type: none"> 1. Hệ thống cần được thiết kế để có thể mở rộng theo chiều ngang 2. Thiết kế cấu trúc dữ liệu ảnh hưởng đến hiệu suất truy vấn và khả năng mở rộng
Nhu cầu truy xuất	<p>Hệ thống cần đảm bảo khả năng truy xuất nhanh chóng với tần suất cao và với số lượng lớn tại:</p> <ol style="list-style-type: none"> 1. Truy xuất theo ngày khởi hành, ngày về, loại vé 2. Truy xuất theo tuyến xe 3. Truy xuất theo giá vé
Trường hợp sử dụng	Khi nhấn vào “Gợi ý” sẽ hiển thị ra thông tin liên quan bao gồm “ngày khởi hành”, “ngày về”, “loại vé”, “tuyến xe”, “giá vé” để người dùng có thể xem xét và lựa chọn đặt vé

➤ **Tìm chuyến xe**

Phân tích	Mô tả
Mục đích	<p>Tính năng Tìm kiếm Chuyến xe giúp người dùng tìm kiếm các chuyến xe phù hợp với nhu cầu, thời gian di chuyển của họ. Người dùng có thể nhập điểm đón, điểm đến, xem xét thời gian mong muốn và hệ thống sẽ cung cấp các lựa chọn chuyến xe phù hợp với các tiêu chí đó.</p> <p>Các thông tin này sẽ được lưu trữ sẽ có các trường dữ liệu khác nhau, dữ liệu được nhập vào rất rộng và đa dạng, do vậy cần có cấu trúc lưu trữ linh hoạt, có thể lưu thông tin dễ dàng và không hạn chế.</p>
Phù hợp với loại dữ liệu	<ol style="list-style-type: none"> 1. Lịch trình xe: Bao gồm thông tin về thời gian khởi hành, thời gian di chuyển, giá vé, điểm đón và điểm trả,... của các chuyến xe. 2. Vé xe: Cho biết mã vé dành cho chuyến xe đó (nếu đã đặt), các thông tin người dùng cần phải cung cấp, giá vé, loại vé, tình trạng vé và ghế ngồi.
Yêu cầu lưu trữ	<ol style="list-style-type: none"> 1. Hệ thống cần lưu trữ thông tin về các chuyến xe 2. Các thông tin quản lý bao gồm mã chuyến xe, điểm đi, điểm đến, ngày giờ khởi hành, số vé,, tìm kiếm gần đây , ngày về nếu chức năng khứ hồi <p>-> cần lưu trữ số lượng lớn dữ liệu chuyến xe</p>
Khả năng mở rộng	Hệ thống cần hỗ trợ khả năng mở rộng ngang để đáp ứng nhu cầu tăng lượng người dùng và lượng dữ liệu.
Nhu cầu truy xuất	<p>Hệ thống phải có khả năng truy xuất nhanh chóng thông tin về các tuyến xe dựa trên yêu cầu tìm kiếm của người dùng.</p> <p>1. Truy vấn theo ngày tháng, tuyến đường, loại vé</p>
Trường hợp sử dụng	Khi nhấn vào “Tìm chuyến xe ” và nhập vào các thông tin xe cần đi , sẽ xuất ra các chuyến xe phù hợp

➤ **Hoá đơn**

Phân tích	Mô tả
Mục đích	Lưu trữ thông tin chi tiết về các hoá đơn sau khi khách hàng đã thanh toán.
Phù hợp với loại dữ liệu	Dữ liệu cấu trúc bao gồm thông tin về vé, số lượng, tổng tiền, ngày xuất hoá đơn, thông tin liên hệ của khách hàng.
Yêu cầu lưu trữ	Lưu trữ lượng lớn thông tin hoá đơn Hệ thống cần lưu trữ thông tin về vé , thông tin khách hàng , thông tin hóa đơn Các thông tin quản lý bao gồm mã hóa đơn , mã vé , slve , tổng tiền , ngày xuất hd , thông tin khách hàng. → cần lưu trữ các thông tin về hóa đơn người dùng, đảm bảo hệ thống có khả năng mở rộng đáp ứng nhu cầu tăng trong tương lai
Khả năng mở rộng	Mở rộng theo chiều ngang để đáp ứng nhu cầu tăng lượng dữ liệu hoá đơn.
Nhu cầu truy xuất	Truy vấn nhanh chóng và chính xác thông tin hoá đơn theo yêu cầu truy vấn mã hóa đơn của người dùng.
Trường hợp sử dụng	Khi nhấn vào “Chi tiết hóa đơn” sẽ hiển thị ra thông tin chi tiết liên quan đến hóa đơn người dùng sau khi thanh toán.

➤ **Thanh toán**

Phân tích	Mô tả
Mục đích	Xử lý và lưu trữ thông tin về các giao dịch thanh toán của khách hàng.
Phù hợp với loại dữ liệu	Dữ liệu thanh toán bao gồm thông tin về phương thức thanh toán, số tiền thanh toán, thời gian giao dịch, trạng thái của giao dịch, và thông tin liên quan đến khách hàng và vé đặt.
Yêu cầu lưu trữ	Lưu trữ lượng lớn thông tin thanh toán Dữ liệu thanh toán cần được lưu trữ một cách an toàn và bảo mật, đảm bảo tính toàn vẹn và khả năng truy cứu nhanh chóng
Khả năng mở rộng	Mở rộng theo chiều ngang để đáp ứng xử lý lưu trữ một lượng lớn các giao dịch
Nhu cầu truy xuất	Dữ liệu thanh toán cần có khả năng truy xuất nhanh chóng để hỗ trợ các chức năng như xem lịch sử thanh toán của khách hàng, kiểm tra trạng thái.
Trường hợp sử dụng	Nhấn vào “Thanh toán” giúp người dùng thanh toán vé

➤ **Tra cứu thông tin vé đã đặt**

Phân tích	Mô tả
Mục đích	Cung cấp khả năng tra cứu thông tin về các vé đã đặt trước đó. của khách hàng.
Phù hợp với loại dữ liệu	Dữ liệu cấu trúc bao gồm thông tin về vé, chuyến xe, ngày đặt, thông tin khách hàng.
Yêu cầu lưu trữ	Lưu trữ lượng lớn thông tin vé đã đặt với cấu trúc dữ liệu có thể thay đổi theo thời gian
Khả năng mở rộng	Mở rộng ngang để đáp ứng nhu cầu tăng lượng dữ liệu vé đã đặt.
Nhu cầu truy xuất	Truy vấn nhanh chóng cho người dùng
Trường hợp sử dụng	Nhấn vào “Tra cứu vé” sẽ hiển thị ra thông tin chi tiết liên quan đến vé người dùng đã đặt

V. THIẾT KẾ CƠ SỞ DỮ LIỆU CHO TỪNG LOẠI

1. COLUMN

CREATE TABLE HoaDon(IDHoaDon text , maVe text, soLuongVe int , TongTien int, NgayXuatHD timestamp, Hoten text ,sdt text ,email text,PRIMARY KEY(IDHoaDon));

idhoaddon	email	hoten	mave	ngayxuathd	sdt	soluongve	tongtien
HD1942	hovancd@gmail.com	Ho Van CD	3	2024-05-02 14:14:09.000000+0000	0982623123	1	15000000
HD6097	hovancd@gmail.com	Ho Van Cfqf	3	2024-05-02 14:18:37.000000+0000	06256561	1	15000000

CREATE TABLE ThanhToan(IDThanhToan uuid, soLuongVe int ,TongTien int, Ngaythanhtoan Date , Hoten text ,sdt text ,email text, HìnhThucThanhToan text,mave text , PRIMARY KEY(IDThanhToan));


idthanhtoan	email	hinhthucthanhtoan	hoten	mave	ngaythanhtoan	sdt	soluongve	tongtien
40a0069c-d20a-4e30-ad87-a95b909a7572	2e2j@gmail.com	Thẻ visa	abc	2	2024-05-02	05615188	1	79006
0a261703-532a-4e9d-b0c3-13bd494de369	lethib@example.com	Thẻ visa	Le Thị C	2	2024-05-02	0987654321	1	300000

2. DOCUMENT

```
DoAn> db.DatVe.insertOne({
  _id: ObjectId(),
  VeID: int ,
  ChuyenXeID: int,
  GheID: int,
  ThôngTinKhachHang: { Hoten: string, SDT:string, Email: string},
  LoạiVe: ["Khứ hồi", "Một chiều"],
  GiaVe: int,
  TìnhTrạngVe:string,
  NgàyKhởiHành: Date,
  NgàyVe:Date,
});
```

```
_id: ObjectId('6631314f57ac969ffe3458af')
VeID : 1
ChuyenXeID : "CX001"
GheID : "G001"
▸ ThôngTinKhachHang : Object
▸ LoạiVe : Array (1)
GiaVe : 500000
TìnhTrạngVe : "Đã đặt"
NgàyKhởiHành : 2024-05-01T00:00:00.000+00:00
NgàyVe : 2024-05-10T00:00:00.000+00:00
```

```
DoAn> db.ChuyenXe.insertOne({
  VeID : int,
  ChuyenXeID :int,
  GiaVe : int ,
  NgàyĐi : Date,
  NgàyVe: Date,
  Mota : string
})
```


	<pre>_id: ObjectId('6631fa362b603337a0a5ad2e') VeID : 1 ChuyenXeID : 1 GiaVe : 79006 NgàyĐi : 2024-04-30T17:00:00.000+00:00 NgàyVe : 1970-01-01T00:00:00.000+00:00 MoTa : "TPHCM -> CanTho"</pre>
	<pre>_id: ObjectId('66333ae2e7b8ebf6ba6386de') VeID : 2 ChuyenXeID : 2 GiaVe : 79006 NgàyĐi : 2024-05-20T00:00:00.000+00:00 NgàyVe : 1970-01-01T00:00:00.000+00:00 MoTa : "TPHCM -> DaLat"</pre>
	<pre>_id: ObjectId('66333c6be7b8ebf6ba6386df') VeID : 3 ChuyenXeID : 3 GiaVe : 50000 NgàyĐi : 2024-05-25T00:00:00.000+00:00 NgàyVe : 2024-05-30T00:00:00.000+00:00 MoTa : "HN -> TPHCM"</pre>

3. KEY-VALUE

```
HMSET ve:<id> maVe "null" ngayDat "null" HoTen "null" sdt "null"
```

```
new HashEntry("SDT", sdtValue),  
new HashEntry("Email", emailValue),  
new HashEntry("NgayKhoiHanh", ngayKhoiHanh),  
new HashEntry("NgayVe", ngayVe),  
new HashEntry("LoaiVe", loaiVe),  
new HashEntry("GiaVe",
```

4. GRAPH

```
create ()
```

VI. GIAO DIỆN CHỨC NĂNG

a) Giao diện chính


The screenshot displays the 'FUTA Bus' application window. The interface is divided into several sections:

- Top Bar:** Contains the application name 'FUTA Bus' and standard window control buttons (minimize, maximize, close).
- Form Fields:**
 - Thông tin khách hàng (Customer Information):** Includes input fields for 'Họ tên' (Full Name), 'SĐT' (Phone Number), and 'Email'.
 - ID Vé (Ticket ID):** An input field.
 - ID Chuyến xe (Bus Trip ID):** An input field.
 - Mô tả (Description):** A larger text area for additional details.
- Buttons:**
 - 'Gọi ý' (Suggest) is located near the top right.
 - 'Tra cứu vé đã đặt' (Search for booked tickets) is located near the top right.
 - 'Xem danh sách chuyến xe' (View bus trip list) is a button on the left side.
 - 'Đặt vé' (Book ticket) is a button at the bottom left.
- Dropdowns and Inputs:**
 - 'Ghế ID' (Seat ID) and 'Loại vé' (Ticket type) are dropdown menus.
 - 'Ngày đi' (Departure date) and 'Ngày về' (Return date) are date input fields.
 - 'Giá vé' (Ticket price) is a numeric input field.
- Main Content Area:** A large, empty gray rectangular area, likely intended for displaying a list of bus trips or search results.

b) Giao diện đặt vé và thanh toán

Thanh Toán

Chi tiết vé xe



ID Vé Xe

Chuyến xe ID

Tuyến xe

Loại vé

Ngày khởi hành

Ngày về

Ghế


Thông tin khách hàng


Họ Tên


SĐT

Email

Phương thức thanh toán

☐  Momo

☐  Thẻ ATM

☐  Thẻ Visa

Chi tiết giá

Giá vé

Phí thanh toán

Tổng tiền

Chi Tiết Hóa Đơn

Đóng

c) Giao diện hiển thị hóa đơn

Form1

Hóa Đơn

Mã Hóa Đơn

Bảng hóa đơn

Chi tiết hóa đơn

Mã Hóa Đơn Họ tên KH SDT

Email Số lượng vé Mã Vé

Tổng tiền Ngày Xuất HD

d) Giao diện tra cứu vé đã đặt

Nhập mã vé tra cứu

Thông tin vé

Họ tên	<input type="text"/>	SĐT	<input type="text"/>
Email	<input type="text"/>	Loại vé	<input type="text"/>
Ngày Khởi Hành	<input type="text"/>	Ngày về	<input type="text"/>
Giá vé	<input type="text"/>		

VII. THIẾT KẾ CHỨC NĂNG

a) Đặt vé

FUTA Bus

Thông tin khách hàng

Họ tên:

SĐT: Email:

ID Vé: ID Chuyển xe:

Gọi ý: Tra cứu vé đã đặt:

Mô tả:

Xem danh sách chuyển xe

Ghế ID:

Loại vé:

Ngày đi:

Ngày về:

Giá vé:

	Id	VeiD	ChuyenXeID	GiaVe	NgàyĐi	NgàyVề	MôT
▶	6631fa362b6033...	1	1	79006	4/30/2024 5:00 ...	1/1/1970	TPHC
	66333ae2e7b8e...	2	2	79006	5/20/2024	1/1/1970	TPHC
	66333c6be7b8e...	3	3	50000	5/25/2024	5/30/2024	HN ->

- Người dùng sẽ chọn danh sách chuyển xe được truy vấn từ collection ChuyenXe trong database DoAn

```
static MongoClient client = new MongoClient("mongodb://127.0.0.1:27017");
static IMongoDatabase db = client.GetDatabase("DoAn");
static IMongoCollection<ChuyenXe> collection1 = db.GetCollection<ChuyenXe>("ChuyenXe");
2 references
public string Mota{ get; private set; }
```

- Nhập vào các trường thông tin cần thiết (Họ tên ,SĐT , Email , Ghế , Loại vé). Nhấn vào Đặt vé để thanh toán vé.
- Khi đó các thông tin vé đã nhập của người dùng sẽ fill vào các box

```
private void button1_Click_1(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(hoten.Text) || string.IsNullOrEmpty(sdt.Text) || string.IsNullOrEmpty(email.Text))
    {
        MessageBox.Show("Nhập đầy đủ thông tin (Họ tên, SĐT, Email) trước khi đặt vé !!!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    else
    {
        Checkout checkoutForm = new Checkout();
        checkoutForm.FillData(textBox1.Text, textBox2.Text, textBox7.Text, comboBox2.Text, textBox8.Text, comboBox1.Text, textBox4.Text, hoten.Text, sdt.Text, email.Text, Mota);
        checkoutForm.Show();
    }
}
```

```
public void FillData(string idVe, string idChuyenXe, string ngayDi, string ghe, string ngayVe, string loiVe, string giaVe, string hoTen, string soDienThoai, string email, string motachuyenxe)
{
    idvexe.Text = idVe;
    tuyenxe.Text = motachuyenxe;
    ngaykh.Text = ngayDi;
    this.ghe.Text = ghe;
    idchuyenxe.Text = idChuyenXe;
    loiave.Text = loiVe;
    ngayve.Text = ngayVe;
    giave.Text = giaVe;
    hoten.Text = hoTen;
    sdt.Text = soDienThoai;
    this.email.Text = email;
    int giaveInt = int.Parse(giave.Text);
    vat.Text = "0";
    int vatInt = int.Parse(vat.Text);
    int total = giaveInt + vatInt;
    tongtien.Text = total.ToString();
    Total = total;
    xuathoadon.Visible = false;
    xemhd.Visible = false;
    exit.Visible = false;
}
```

- Khi tiến hành đặt vé , hệ thống sẽ kiểm tra xem ghế với chuyến xe đã được đặt trước đó hay không.

```
var gheID = comboBox2.Text;
var chuyenxeID = textBox2.Text;
var datVeCollection = db.GetCollection<BsonDocument>("DatVe");
var filter = Builders<BsonDocument>.Filter.And(
    Builders<BsonDocument>.Filter.Eq("GheID", gheID),
    Builders<BsonDocument>.Filter.Eq("ChuyenXeID", chuyenxeID)
);
var result = datVeCollection.Find(filter).Any();
```


- Nếu không trùng , thì tiếp tục cho người dùng thanh toán .

b) Thanh Toán

- Xác nhận các thông tin và chọn phương thức thanh toán .Người dùng nhấn vào nút thanh toán

Thanh Toán

Chi tiết vé xe




ID Vé Xe: Chuyển xe ID:


Tuyến xe: Loại vé:


Ngày khởi hành: Ngày về:

Ghế:

Phương thức thanh toán

☐  Momo

☐  Thẻ ATM

☐  Thẻ Visa

Chi tiết giá

Giá vé:

Phí thanh toán:

Tổng tiền:

Thông tin khách hàng

Họ Tên:

SĐT:

Email:

Thanh Toán

- Dữ liệu thông tin đặt vé của người dùng sẽ ghi xuống vào MongoDB để lưu trữ

```
private void InsertMongo()
{
    string mave = idvexe.Text;
    string hoten = hoten.Text;
    string sdtValue = sdt.Text;
    string emailValue = email.Text;

    Datve = database.GetCollection<BsonDocument>("DatVe");
    var document = new BsonDocument
    {
        { "_id", ObjectId.GenerateNewId() },
        { "VeID", mave },
        { "ChuyenXeID", idchuyenxe.Text },
        { "GheID", ghe.Text },
        { "ThongTinKhachHang", new BsonDocument
            {
                { "Hoten", hoten },
                { "SDT", sdtValue },
                { "Email", emailValue }
            }
        },
        { "LoaiVe", loaive.Text },
        { "GiaVe", Total },
        { "TinhTrangVe", "Đã đặt" },
        { "NgayKhoiHanh", ngaykh.Text },
        { "NgayVe", ngayve.Text }
    };

    Datve.InsertOne(document);
}
```

```
_id: ObjectId('6634ff00854f1f93999b0116')
VeID: "1"
ChuyenXeID: "1"
GheID: "G4"
ThongTinKhachHang: Object
LoaiVe: "Khứ hồi"
GiaVe: 79006
TinhTrangVe: "Đã đặt"
NgayKhoiHanh: "4/30/2024 5:00:00 PM"
NgayVe: "1/1/1970 12:00:00 AM"
```

- Lưu trữ dữ liệu ThanhToan vào cassandra

```
private void InsertCheckOutCass()
{
    string mave = idvexe.Text;
    string hoTen = hoten.Text;
    string sdtValue = sdt.Text;
    string emailValue = email.Text;
    int soLuongVe = 1;
    int tongTien = Total;
    DateTime ngayThanhToan = DateTime.Now;
    string hinhThucThanhToan = "";
    if (momo.Checked)
    {
        hinhThucThanhToan = "Momo";
    }
    else if (atm.Checked)
    {
        hinhThucThanhToan = "Thẻ atm";
    }
    else if (visa.Checked)
    {
        hinhThucThanhToan = "Thẻ visa";
    }

    var query = $"INSERT INTO ThanhToan (IDThanhToan, soLuongVe, TongTien, Ngaythanhtoan, Hoten, sdt, email, HinhThucThanhToan,mave) " +
        $"VALUES (uuid(), {soLuongVe}, {tongTien}, '{ngayThanhToan:yyyy-MM-dd}', '{hoTen}', '{sdtValue}', '{emailValue}', '{hinhThucThanhToan}', '{mave}')";

    session.Execute(query);
    MessageBox.Show("Thanh toán thành công !!!");
}
```

idthanhtoan	email	hinhthucthanhtoan	hoten	mave	ngaythanhtoan	sdt	soluongve	tongtien
40a0069c-d20a-4e30-ad87-a95999a7572	2e2j@gmail.com	Thẻ visa	abc	2	2024-05-02	05615188	1	79006
0a261703-532a-4e9d-b0c3-13bd494de369	lethib@example.com	Thẻ visa	Le Thi C	2	2024-05-02	0987654321	1	300000
c143b71c-0304-4b4a-b056-32a59521221	lethib@example.com	Thẻ visa	Le Thi B	2	2024-05-02	0987654321	1	300000
0ab25a0d-0b44-426d-bf61-7865f8498e7a	nguyenvana@example.com	Momo	Nguyen Van A	1	2024-05-02	0123456789	1	500000
17cc5e2a-14b1-46fb-92d9-42a08b268c66	nguyenvana@example.com	Thẻ visa	Nguyen Van A	1	2024-05-02	0123456789	1	500000
60489217-a089-4983-bc01-135cab05741b	lethib@example.com	Thẻ visa	Le Thi B	2	2024-05-02	0987654321	1	300000
eb02ccf6-dc86-481b-b48a-a28de88aebc	hovancd@gmail.com	Thẻ atm	Ho Van Cfqf	3	2024-05-02	06256561	1	1500000
55bb1ccf-d301-4aa3-ae3e-18497b50826c	abc@gmail.com	Thẻ atm	Nguyen Van A	2	2024-05-02	0345848	1	79006
f8c56f69-7047-444d-945e-0c9d5a5afa45	nguyenvana@example.com	Thẻ visa	Nguyen Van A	1	2024-05-02	0123456789	1	500000
edb7b312-c6d3-46cf-ac81-d2b17b7a3161	nguyenvana@example.com	Thẻ visa	Nguyen Van d	1	2024-05-02	55641	1	500000
5d9afb88-3654-4fee-891e-4c40c038bc3	nguyenvana@example.com	Momo	Nguyen Van A	1	2024-05-02	0123456789	1	500000
e3baed5f-d6ad-44ad-a87a-6b161017190e	nguyenvana@example.com	Thẻ atm	Nguyen Van A	1	2024-05-02	0123456789	1	500000
0bf74295-b6a0-4404-bb5b-ea71d3acd050	hrhn@gmail.com	Thẻ atm	Nguyen B	1	2024-05-02	062565614	1	500000
4051c2f4-0d06-4476-93f9-4dae6fd4960c	nguyenvana@example.com	Thẻ visa	Nguyen Van A	1	2024-05-02	0123456789	1	500000
4a145ba9-891d-4ec3-8275-267e0432d5d9	nguyenvana@example.com	Thẻ visa	Nguyen Van A	1	2024-05-02	0123456789	1	500000
c817cf9b-a32b-4ba9-b0c6-633f0741976e	nguyenvana@example.com	Momo	Nguyen Van A	1	2024-05-02	0123456789	1	500000
253a0dee-c650-42a3-b034-a0823f3d1c77	hovancd@gmail.com	Thẻ visa	Ho Van CD	3	2024-05-02	0982623123	1	1500000
d48451c9-5623-456f-b50c-238ebfea250b	abc@gmail.com	Thẻ atm	ggg	2	2024-05-03	05616515	1	79006
eaab4723f-4530-40e3-0739-468d30f01dc3	qffqf	Thẻ atm	abvv	2	2024-05-03	055	1	79006
8362f892-4997-4be7-b923-320bbdaa4f3f	nva@gmail.com	Thẻ atm	Nguyễn Văn A	1	2024-05-03	09015369123	1	79006

- Xác nhận truy xuất hóa đơn sẽ lưu trữ Hóa Đơn người dùng vào cassandra

```
private void xuathoadon_Click(object sender, EventArgs e)
{
    string idhoaddon = GenerateUniqueInvoiceID();

    string mave = idvexe.Text;
    string hoTen = hoten.Text;
    string sdtValue = sdt.Text;
    string emailValue = email.Text;
    int soLuongVe = 1;
    int tongTien = Total;
    DateTime ngayxhd = DateTime.Now;
    string timestamp = ngayxhd.ToString("yyyy-MM-dd HH:mm:ss");

    var query = $"INSERT INTO HoaDon (idhoaddon, email, hoten, mave, ngayxuat, sdt, soluongve, tongtien) " +
        $"VALUES ({idhoaddon}, '{emailValue}', '{hoTen}', '{mave}', '{timestamp}', '{sdtValue}', {soLuongVe}, {tongTien})";

    session.Execute(query);
    MessageBox.Show("Xuất hóa đơn thành công !!!");
    xemhd.Visible = true;
    xuathoadon.Visible = false;
    exit.Visible = true;
}
```

idhoadon	email	hoten	mave	ngayxuathd	sdt	soluongve	tongtien
HD1942	hovancd@gmail.com	Ho Van CD	3	2024-05-02 14:14:09.000000+0000	0982623123	1	15000000
HD6897	hovancd@gmail.com	Ho Van Cfqf	3	2024-05-02 14:18:37.000000+0000	06256561	1	15000000
HD3212	zezj@gmail.com	abc	2	2024-05-02 23:00:17.000000+0000	05615188	1	79006
HD2350	nguyenvana@example.com	Nguyen Van A	1	2024-05-02 12:01:22.000000+0000	0123456789	1	500000
HD8105	lethib@example.com	Le Thi B	2	2024-05-02 14:06:29.000000+0000	0987654321	1	300000
HD1667	abc@gmail.com	Nguyễn Văn A	2	2024-05-02 22:30:25.000000+0000	0345848	1	79006
HD5496	nguyenvana@example.com	Nguyen Van A	1	2024-05-02 12:45:25.000000+0000	0123456789	1	500000
HD5311	qffqf	abvv	2	2024-05-03 00:13:36.000000+0000	055	1	79006
HD5332	nguyenvana@example.com	Nguyen Van A	1	2024-05-02 12:48:25.000000+0000	0123456789	1	500000
HD1530	nva@gmail.com	Nguyễn Văn A	1	2024-05-03 22:19:01.000000+0000	09015369123	1	79006
HD5881	lethib@example.com	Le Thi B	2	2024-05-02 14:05:44.000000+0000	0987654321	1	300000
HD3561	nguyenvana@example.com	Nguyen Van A	1	2024-05-02 12:11:02.000000+0000	0123456789	1	500000

c) Hóa đơn

Người dùng xem chi tiết hóa đơn

```
private void LoadDataGrid()
{
    var result = _session.Execute(query);

    var columns = new Dictionary<string, string>
    {
        { "idhoaddon", "Mã Hóa Đơn" },
        { "mave", "Mã vé" },
        { "soluongve", "Số lượng vé" },
        { "tongtien", "Tổng tiền" },
        { "ngayxuathd", "Ngày Xuất HD" },
        { "hoten", "Họ Tên KH" },
        { "sdt", "SDT" },
        { "email", "Email" },
    };

    foreach (var column in columns)
    {
        dataGrid.Columns.Add(column.Key, column.Value);
        dataGrid.Columns[column.Key].ReadOnly = true;
    }

    foreach (var row in result)
    {
        load_data(row);
    }
}

private ISession _session;
string query = "SELECT * FROM HoaDon";

1 reference
public HoaDon()
{
    InitializeComponent();
    InitializeCassandra();
    LoadDataGrid();
}

1 reference
private void InitializeCassandra()
{
    var cluster = Cluster.Builder()
        .AddContactPoint("127.0.0.1").WithPort(9042)
        .Build();

    _session = cluster.Connect("da");
}
```

- Dữ liệu được query cassandra qua bảng da. Từ đó đổ dữ liệu vào datagrid. Khi đó người dùng xem chi tiết hóa đơn

Hóa Đơn

Mã Hóa Đơn

Bảng hóa đơn

	Mã Hóa Đơn	Mã vé	Số lượng vé	Tổng tiền	Ngày Xuất HD	Họ Tên KH
▶	HD1530	1	1	79006	2024-05-03 ...	Nguyễn Văn
	HD1667	2	1	79006	2024-05-02 ...	Nguyễn Văn
	HD1942	3	1	15000000	2024-05-02 ...	Ho Van CD
	HD2359	1	1	500000	2024-05-02 ...	Nguyen Va

Chi tiết hóa đơn

Mã Hóa Đơn Họ tên KH SDT

Email Số lượng vé Mã Vé

Tổng tiền Ngày Xuất HD

- Từ đó người dùng có thể tìm kiếm hóa đơn

```
private void btnSearch_Click(object sender, EventArgs e)
{
    string maHoaDon = tkHD.Text;
    var query = $"SELECT * FROM HoaDon WHERE idhoadon = '{maHoaDon}' ALLOW FILTERING";
    var result = _session.Execute(query);
    dataGrid.Rows.Clear();

    foreach (var row in result)
    {
        load_data(row);
    }
}
```

- Sử dụng truy vấn bằng mã hóa đơn người dùng nhập vào.

d) Tra cứu vé đã đặt

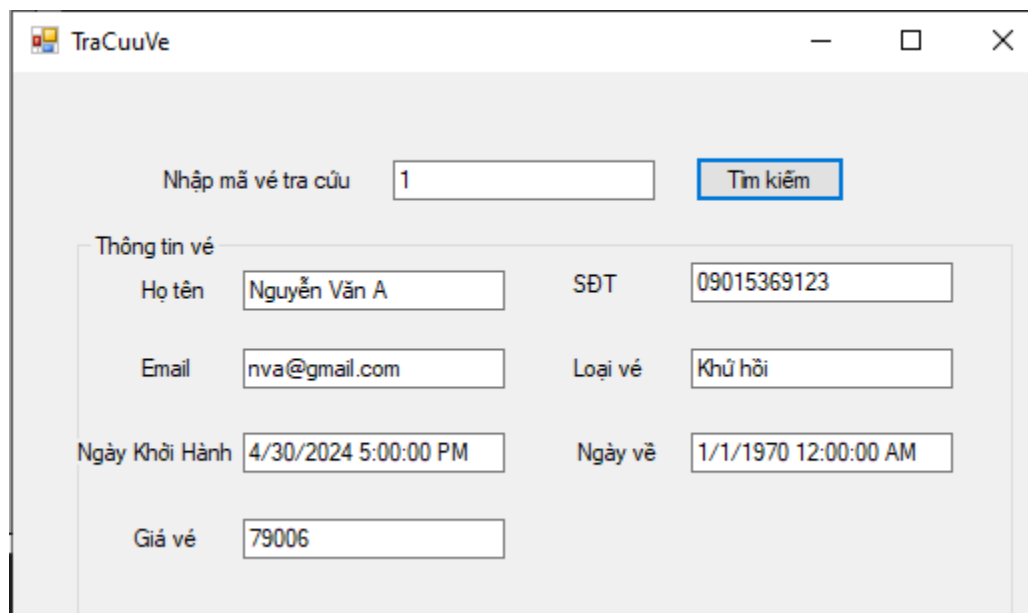
The screenshot shows a Windows application window titled "TraCuuVe". It contains a search form with the following elements:

- A label "Nhập mã vé tra cứu" followed by a text input field and a "Tìm kiếm" button.
- A section titled "Thông tin vé" containing several input fields:
 - Row 1: "Họ tên" and "Số ĐT" (Phone Number).
 - Row 2: "Email" and "Loại vé" (Ticket Type).
 - Row 3: "Ngày Khởi Hành" (Departure Date) and "Ngày về" (Return Date).
 - Row 4: "Giá vé" (Ticket Price).

- Dữ liệu được query theo mã vé người dùng nhập vào

```
private void button1_Click(object sender, EventArgs e)
{
    string mave = txtMaVe.Text;

    var hashEntries = db.HashGetAll(mave);
    if (hashEntries.Length > 0)
    {
        txtHoTen.Text = getHash(hashEntries, "HoTen");
        txtSDT.Text = getHash(hashEntries, "SDT");
        txtEmail.Text = getHash(hashEntries, "Email");
        txtNgayKhoiHanh.Text = getHash(hashEntries, "NgayKhoiHanh");
        txtNgayVe.Text = getHash(hashEntries, "NgayVe");
        txtLoaiVe.Text = getHash(hashEntries, "LoaiVe");
        txtGiaVe.Text = getHash(hashEntries, "GiaVe");
    }
}
```



The screenshot shows a web application window titled "TraCuuVe" with standard Windows window controls (minimize, maximize, close). The interface is in Vietnamese and contains the following elements:

- A search section with the label "Nhập mã vé tra cứu" (Enter ticket code to search), a text input field containing the number "1", and a blue button labeled "Tìm kiếm" (Search).
- A section titled "Thông tin vé" (Ticket information) containing several input fields:
 - "Họ tên" (Full name) with the value "Nguyễn Văn A".
 - "SĐT" (Phone number) with the value "09015369123".
 - "Email" with the value "nva@gmail.com".
 - "Loại vé" (Ticket type) with the value "Khứ hồi" (Round trip).
 - "Ngày Khởi Hành" (Departure date) with the value "4/30/2024 5:00:00 PM".
 - "Ngày về" (Return date) with the value "1/1/1970 12:00:00 AM".
 - "Giá vé" (Ticket price) with the value "79006".