

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT  
BỘ MÔN CÔNG NGHỆ PHẦN MỀM



TRẦN CAO QUYỀN                  MSSV: 17110213  
NGUYỄN TRỌNG LUẬT                  MSSV: 17110181

Đề tài:

**TÌM HIỂU THUẬT TOÁN VÀ  
XÂY DỰNG ỨNG DỤNG GIAO HÀNG**

**TIÊU LUẬN CHUYÊN NGÀNH**

**GIÁO VIÊN HƯỚNG DẪN  
TS. HUỲNH XUÂN PHUNG**

**KHÓA 2017 - 2021**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM  
BỘ MÔN CÔNG NGHỆ PHẦN MỀM**



**TRẦN CAO QUYỀN**      **MSSV: 17110213**  
**NGUYỄN TRỌNG LUẬT**      **MSSV: 17110181**

**Đề tài:**

**TÌM HIỂU THUẬT TOÁN VÀ  
XÂY DỰNG ỨNG DỤNG GIAO HÀNG**

**TIÊU LUẬN CHUYÊN NGÀNH**

**GIÁO VIÊN HƯỚNG DẪN  
TS. HUỲNH XUÂN PHUNG**

**KHÓA 2017 - 2021**

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên 1: TRẦN CAO QUYỀN                  MSSV 1: 17110213

Họ và tên Sinh viên 2: NGUYỄN TRỌNG LUẬT    MSSV 2: 17110181

Tên đề tài: TÌM HIỂU THUẬT TOÁN VÀ XÂY DỰNG ỨNG DỤNG GIAO HÀNG

Họ và tên Giáo viên hướng dẫn: TS. HUỲNH XUÂN PHUNG

**NHẬN XÉT**

1. Về nội dung đề tài và khối lượng thực hiện:

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

2. Ưu điểm:

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

3. Khuyết điểm:

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

4. Đề nghị cho bảo vệ hay không? .....

5. Đánh giá loại: .....

6. Điểm: .....

*Tp. Hồ Chí Minh, ngày tháng năm 2020*

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

### **PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

Họ và tên Sinh viên 1: TRẦN CAO QUYỀN              MSSV 1: 17110213

Họ và tên Sinh viên 2: NGUYỄN TRỌNG LUẬT    MSSV 2: 17110181

Ngành: Công nghệ Thông tin

Tên đề tài: TÌM HIỂU THUẬT TOÁN VÀ XÂY DỰNG ỨNG DỤNG GIAO HÀNG

Họ và tên Giáo viên phản biện: .....

#### **NHẬN XÉT**

1. Về nội dung đề tài và khối lượng thực hiện:

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

2. Ưu điểm:

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

3. Khuyết điểm:

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

4. Đề nghị cho bảo vệ hay không? .....

5. Đánh giá loại: .....

6. Điểm: .....

*Tp. Hồ Chí Minh, ngày tháng năm 2020*

Giáo viên phản biện

(Ký & ghi rõ họ tên)

## **Lời cảm ơn**

*Học tập là quá trình rèn luyện, cung cấp kiến thức hành trang cho sinh viên bước vào công việc trong tương lai. Nhờ ơn những thầy cô đã dùi dắt, chỉ bảo những kiến thức, kinh nghiệm để áp dụng giúp cho chúng em hoàn thành tốt học phần.*

*Với lòng biết ơn sâu sắc nhất, chúng em xin chân thành cảm ơn các thầy cô giáo trong trường ĐH Sư Phạm Kỹ Thuật TP.HCM nói chung, các thầy cô trong khoa Đào tạo CLC nói riêng đã dạy dỗ cho em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt quá trình học tập.*

*Chúng em xin gửi lời cảm ơn chân thành đến thầy Huỳnh Xuân Phụng, giảng viên hướng dẫn Tiểu luận tốt nghiệp - trường ĐH Sư Phạm Kỹ Thuật TP.HCM người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt quá trình nghiên cứu.*

*Cuối cùng, chúng em xin chân thành cảm ơn gia đình và bạn bè đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên trong suốt quá trình học tập và hoàn thành tốt phần mềm.*

*Tp. Hồ Chí Minh, ngày 30/05/2020*

*Trần Cao Quyền – 17110213  
Nguyễn Trọng Luật – 17110181*

## MỤC LỤC

LỜI CẢM ƠN .....	5
ĐỀ CƯƠNG TIỂU LUẬN CHUYÊN NGÀNH .....	10
KẾ HOẠCH THỰC HIỆN .....	11
Phần 1: MỞ ĐẦU .....	12
1.1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI .....	12
1.2. MỤC ĐÍCH CỦA ĐỀ TÀI .....	13
1.3. CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU - ĐỐI TƯỢNG NGHIÊN CỨU - PHẠM VI NGHIÊN CỨU .....	14
1.4. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC .....	14
Phần 2: NỘI DUNG .....	15
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....	15
1.1. TỔNG QUAN VỀ HỆ THỐNG .....	15
1.1.1. KHÁI QUÁT VỀ HỆ THỐNG: .....	15
1.1.2. CÁC THÀNH PHẦN CỦA HỆ THỐNG: .....	15
1.2. NODEJS .....	17
1.2.1. KHÁI QUÁT VỀ NODEJS .....	17
1.2.2. ĐẶC ĐIỂM GIÚP NODEJS TRỎ NÊN MẠNH MẼ VÀ NỔI BẬT TRÊN THỊ TRƯỜNG GẦN ĐÂY? .....	17
1.2.3. NPM (NODEJS PACKAGE MANAGER): .....	20
1.2.3.1. Package manager là gì? .....	20
1.2.3.2. Tại sao lại sử dụng npm? .....	21
1.2.4. NHƯỢC ĐIỂM CỦA NODEJS .....	21
1.3. EXPRESSJS .....	21
1.3.1. KHÁI QUÁT VỀ EXPRESSJS .....	21
1.3.2. ĐẶC ĐIỂM CỦA EXPRESSJS .....	22
1.3.3. RESTFUL API .....	22
1.3.4. TẠI SAO LẠI SỬ DỤNG EXPRESS TRONG VIỆC QUẢN LÝ REST API .....	24
1.4. POSTGRES-SQL .....	24
1.4.1. KHÁI QUÁT VỀ RDBMS .....	24

1.4.2. KHÁI QUÁT VỀ POSTGRES-SQL.....	25
1.4.3. SEQUELIZE.....	25
1.5. REACT NATIVE .....	26
1.5.1. GIỚI THIỆU VỀ REACT NATIVE.....	26
1.5.2. ƯU / NHƯỢC CỦA REACT NATIVE .....	26
1.5.3. NGÔN NGỮ LẬP TRÌNH JAVASCRIPT.....	27
CHƯƠNG 2: PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU.....	29
2.1. KHẢO SÁT HIỆN TRẠNG.....	29
2.2. XÁC ĐỊNH YÊU CẦU .....	32
2.2.1. Xác định yêu cầu chức năng hệ thống: .....	33
2.2.2. Lược đồ Use case và đặc tả .....	35
2.2.3.1. Use case Đăng ký .....	40
2.2.3.2. Use case Đăng nhập.....	41
2.2.3.3. Use case Xem danh sách sản phẩm .....	42
2.2.3.4. Use case Xem sản phẩm.....	43
2.2.3.5. Use case Thêm giỏ hàng .....	45
2.2.3.6. Use case Xem giỏ hàng.....	46
2.2.3.7. Use case Chọn địa chỉ.....	47
2.2.3.8. Use case Thêm địa chỉ .....	49
2.2.3.9. Use case Đặt hàng.....	50
2.2.3.10. Use case Xem sản phẩm đã đặt.....	51
2.2.3.12 Use case Đăng nhập appgiaohang.....	54
2.2.3.13. Use case Xem danh sách chuỗi đơn hàng .....	55
2.2.3.14. Use case Đăng ký chuỗi đơn hàng .....	57
2.2.3.15. Use case Xem thông tin khách hàng .....	58
2.2.3.16. Use case Liên hệ khách hàng .....	59
2.2.3.17. Use case Xác nhận đơn hàng.....	61
2.2.3.18. Use case Xem tình trạng chuỗi đơn .....	62
2.2.3.19. Use case Xác nhận hoàn thành chuỗi đơn hàng .....	63
2.2.3.20. Use case Xem tình trạng và thông tin đơn hàng từng khách hàng .....	64

2.2.3.21 Use case Đăng xuất .....	66
2.2.3.22 Use case Xem thông tin cá nhân.....	67
<b>CHƯƠNG 3: THIẾT KẾ ỨNG DỤNG.....</b>	<b>69</b>
<b>3.1. THIẾT KẾ DỮ LIỆU.....</b>	<b>69</b>
3.2.1. Lược đồ quan hệ .....	69
3.2.2. Mô tả cơ sở dữ liệu .....	69
3.2.2.1. NguoiDung.....	69
3.2.2.2. LoaiSanPham.....	70
3.2.2.3. SanPham.....	71
3.2.2.4. Shipper.....	72
3.2.2.5. Diachi.....	73
3.2.2.5. DonHang.....	74
3.2.2.6. ChuoiGiaoHang.....	75
3.2.2.7. ThongBao.....	75
3.2.2.8. DSSanPham .....	76
3.3. LƯỢC ĐỒ TUẦN TỰ .....	77
3.3.1. Ứng dụng khách hàng: .....	77
3.3.1.1. Lược đồ Đăng ký.....	77
3.3.1.2. Lược đồ Đăng nhập .....	78
3.3.1.3. Lược đồ Đăng xuất.....	78
3.3.1.4. Lược đồ Chọn sản phẩm.....	79
3.3.1.5. Lược đồ Đặt hàng .....	80
3.3.1.6. Lược đồ Xem đơn hàng.....	80
3.3.1.8. Lược đồ Tìm kiếm theo tên sản phẩm .....	81
3.3.1.9. Lược đồ Tìm kiếm theo loại sản phẩm .....	82
3.3.2. Ứng dụng shipper .....	83
3.3.2.1. Lược đồ Đăng ký chuỗi giao hàng.....	83
3.3.2.2. Lược đồ Giao hàng .....	84
3.3.3. Lược đồ giữa hai Server:.....	85
3.4. GIAO DIỆN.....	86

3.4.1. Màn hình Danh sách sản phẩm .....	86
3.4.2. Màn hình Sản Phẩm .....	88
3.4.3. Màn hình Đăng nhập khách hàng .....	89
3.4.4. Màn hình Giỏ Hàng .....	90
3.4.5. Màn hình Cá nhân .....	92
3.4.6. Màn hình Chọn địa chỉ .....	93
3.4.7. Màn hình Thêm địa chỉ .....	94
3.4.8. Màn hình Sản phẩm đã mua.....	96
3.4.9. Màn hình Quản lý đơn hàng .....	97
3.4.10. Màn hình Thông tin khách hàng .....	98
3.4.11. Màn hình Đăng Ký .....	99
3.4.12. Màn hình Khởi đầu .....	100
3.4.13. Màn hình Map chưa đăng ký chuỗi đơn .....	101
3.4.14. Màn hình Map Khi đăng ký chuỗi .....	102
3.4.15. Màn hình Đăng ký giao hàng .....	103
3.4.16. Màn hình Tình trạng Chuỗi đơn .....	104
3.4.17. Màn hình Chuỗi đơn hàng .....	105
3.4.18. Màn hình menu .....	106
3.4.19. Màn hình Thông tin shipper .....	107
3.4.20. Màn hình Đăng xuất .....	108
<b>CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG.....</b>	<b>109</b>
4.1. CÀI ĐẶT ỨNG DỤNG .....	109
4.2. KIỂM THỬ ỨNG DỤNG .....	112
4.2.1. Ứng dụng khách hàng .....	112
4.2.2. Ứng dụng giao hàng .....	116
<b>Phần 3: KẾT LUẬN .....</b>	<b>119</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>120</b>

## ĐỀ CƯƠNG TIỂU LUẬN CHUYÊN NGÀNH

Họ và Tên SV thực hiện 1: TRẦN CAO QUYỀN                          MSSV: 17110213  
Họ và Tên SV thực hiện 2: NGUYỄN TRỌNG LUẬT                          MSSV: 17110181  
Thời gian làm tiểu luận từ: 05/08/2020 đến: 23/12/2020  
Chuyên ngành: Công nghệ phần mềm  
Tên tiểu luận: TÌM HIỂU THUẬT TOÁN VÀ XÂY DỰNG ỨNG DỤNG GIAO HÀNG  
GV hướng dẫn: TS. HUỲNH XUÂN PHUNG

### Nhiệm vụ của luận văn:

1. Nghiên cứu về React Native
2. Nghiên cứu về cách viết RESTFUL API và cách ứng dụng
3. Sử dụng thư viện Sequelize để tạo Model cho dữ liệu trong Postgres-SQL
4. Sử dụng Express để làm các công việc về phía server-side
5. Viết các RESTFUL API để thao tác dữ liệu giữa Express và Postgres-SQL
6. Dùng React Native để làm front-end application
7. Sử dụng thư viện hierarchical clustering (Python) hỗ trợ thuật toán chia đơn hàng
8. Ráp 4 thành phần lại để xây dựng ứng dụng đặt hàng và giao hàng.
9. Kiểm thử

### Đề cương viết luận văn:

## MỤC LỤC

### Phần 1: Mở đầu

- 1.1. Tính cấp thiết của đề tài
- 1.2. Mục đích của đề tài
- 1.3. Cách tiếp cận và phương pháp nghiên cứu
- 1.4. Kết quả dự kiến đạt được

### Phần 2: Nội dung

#### Chương 1: Cơ sở lý thuyết

- 1.1. React Native
- 1.2. Nodejs
- 1.3. Expressjs
- 1.4. Postgres-SQL
- 1.5. Python

Chương 2: Phân tích và mô hình hóa yêu cầu

Chương 3: Thiết kế ứng dụng

Chương 4: Cài đặt và kiểm thử ứng dụng

Phần 3: Kết luận

Tài liệu tham khảo

## KẾ HOẠCH THỰC HIỆN

STT	Thời gian	Công việc	Ghi chú
1	31/08/2020 – 06/09/2020	Tìm hiểu về NodeJs, React Native, Javascript Tạo đồ án trên github	
2	07/09/2020 – 13/09/2020	Tổng hợp các vấn đề, yêu cầu trong đề tài	
3	14/09/2020 – 20/09/2020	Giải quyết các vấn đề trong đề tài	
4	21/09/2020 – 27/09 /2020	Thiết Kế Giao Diện Ứng Dụng Mua Bán Đặt Hàng cho Khách Hàng và Ứng dụng Giao Hàng Cho Shipper	
5	28/09/2020 – 04/10/2020	Tiếp tục Xây dựng Giao Diện Tạo cơ bản database Nghiên cứu giải quyết vấn đề giao hàng	
6	05/10/2020 – 11/10/2020	Tiếp tục Xây dựng Giao Diện Cải tiến thuật toán Quay Demo	
7	12/10/2020 – 18/10/2020	Tiếp tục Xây dựng Giao Diện Cải tiến thuật toán Tìm hiểu thuật toán hierarchical clustering	

8	19/10/2020 – 25/10/2020	Tiếp tục Xây dựng Giao Diện	
9	26/10/2020 – 01/11/2020	Fetch Data cho app	
10	02/11/2020 – 08/11/2020	Triển khai thuật toán Giải quyết vấn đề giao hàng khi khách hàng chọn nhiều địa chỉ	
11	09/11/2020 – 15/11/2020	Tiếp tục Xây dựng Giao Diện Tạo cơ bản database Nghiên cứu giải quyết vấn đề giao hang	
12	16/11/2020 – 22/11/2020	Xây dựng hoàn thiện server	
13	23/11/2020 – 29/11/2020	Fetch data từ server về app	
14	30/11/2020 – 06/12/2020	Demo Liên kết các thành phần	
15	07/12/2020 – 13/12 /2020	Vẽ Sequence Diagram	
16	14/12/2020 – 20/12 /2020	Viết báo cáo	

TP HCM, ngày 15 tháng 3 năm 2020

Ý kiến của giáo viên hướng dẫn

Người viết đề cương

(Ký và ghi rõ họ tên)

## Phần 1: MỞ ĐẦU

### 1.1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Năm 2020 là năm nhiều biến cố nhất trong những năm gần đây trên toàn thế giới.

Đặc biệt là Dịch bệnh Covid-19 khởi nguồn từ Vũ Hán, Trung Quốc vào khoảng cuối tháng 12 năm 2019 kéo dài cho đến nay. Dịch bệnh đã kéo theo nhiều hệ lụy cho Thế Giới nói chung và Việt Nam nói riêng. Nhiều người bị thất nghiệp, nhiều công ty trên đà phá sản, ...

## PHẦN MỞ ĐẦU

Dịch bệnh vẫn đang tiếp tục xảy ra trên Thế Giới và chưa có dấu hiệu dừng lại. Biện pháp tạm thời và cần thiết của Thế giới nói chung và Việt Nam nói riêng là đeo khẩu trang và thực hiện giãn cách xã hội. Việc giãn cách xã hội đã kéo theo nhiều dịch vụ đang phát triển trở nên phát triển mạnh mẽ hơn. Đặc biệt là Thương mại điện tử là Logistic.

Trong thời kỳ 4.0 hiện tại, con người đang có xu hướng mua hàng online ngày càng nhiều, đặc biệt trong thời kỳ giãn cách xã hội việc mua hàng online càng bùng nổ. Người ta có thể mua được bất cứ thứ gì mình cần ở trên mạng thông qua các phương tiện xã hội như Facebook, hay các trang website, ứng dụng thương mại điện tử.

Do nhu cầu mua hàng online trở nên nhiều nên kéo theo ngành dịch vụ Logistic cũng phát triển theo. Hàng loạt các công ty được thành lập nhằm mục đích cung ứng nhu cầu của người mua hàng và bên bán hàng. Việc giao hàng hiện tại trở nên khá cần thiết cho nhu cầu của con người. Và việc giao hàng làm sao để thuận tiện cho bên Công ty, Người mua bán, và người giao hàng vẫn đang liên tục được các công ty nghiên cứu và phát triển.

Với các lý do trên, đề tài luận án được chọn là “**Tìm hiểu Thuật toán và Xây dựng ứng dụng Giao Hàng**”.

### 1.2. MỤC ĐÍCH CỦA ĐỀ TÀI

Tìm hiểu Restful API, ứng dụng Restful API và kết hợp với các thư viện có sẵn để xây dựng ứng dụng.

Áp dụng được công nghệ của các bên thứ 3 như GeoCoder, Google Service để làm cho đề tài thêm phong phú. Qua đó học tập các kỹ năng tích hợp công nghệ do bên thứ 3 cung cấp vào dự án.

## PHẦN MỞ ĐẦU

Xây dựng hệ thống giúp khách hàng đặt hàng và giúp người giao hàng có được lộ trình giao hàng hợp lý, nhanh chóng và thuận tiện.

### **1.3. CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU - ĐỐI TƯỢNG NGHIÊN CỨU - PHẠM VI NGHIÊN CỨU**

#### **Cách tiếp cận:**

Tiếp cận đề tài theo hướng tìm hiểu kiến thức lý thuyết thông qua tài liệu online, các khóa học online, video hướng dẫn. Từ đó tổng hợp kiến thức lý thuyết vào xây dựng một ứng dụng thực để có hiểu rõ, hiểu sâu nhất về đối tượng nghiên cứu.

#### **Đối tượng nghiên cứu:**

Tổng quan về Restful API, các nền tảng như React Native, Python.

Cách tích hợp các dịch vụ bên thứ 3 như là dịch vụ lưu trữ Heroku, Postgres-SQL.

**Phạm vi nghiên cứu:** Nghiên cứu lý thuyết về Restful API thông qua tài liệu trên mạng. Xây dựng hệ thống đặt hàng và giao hàng.

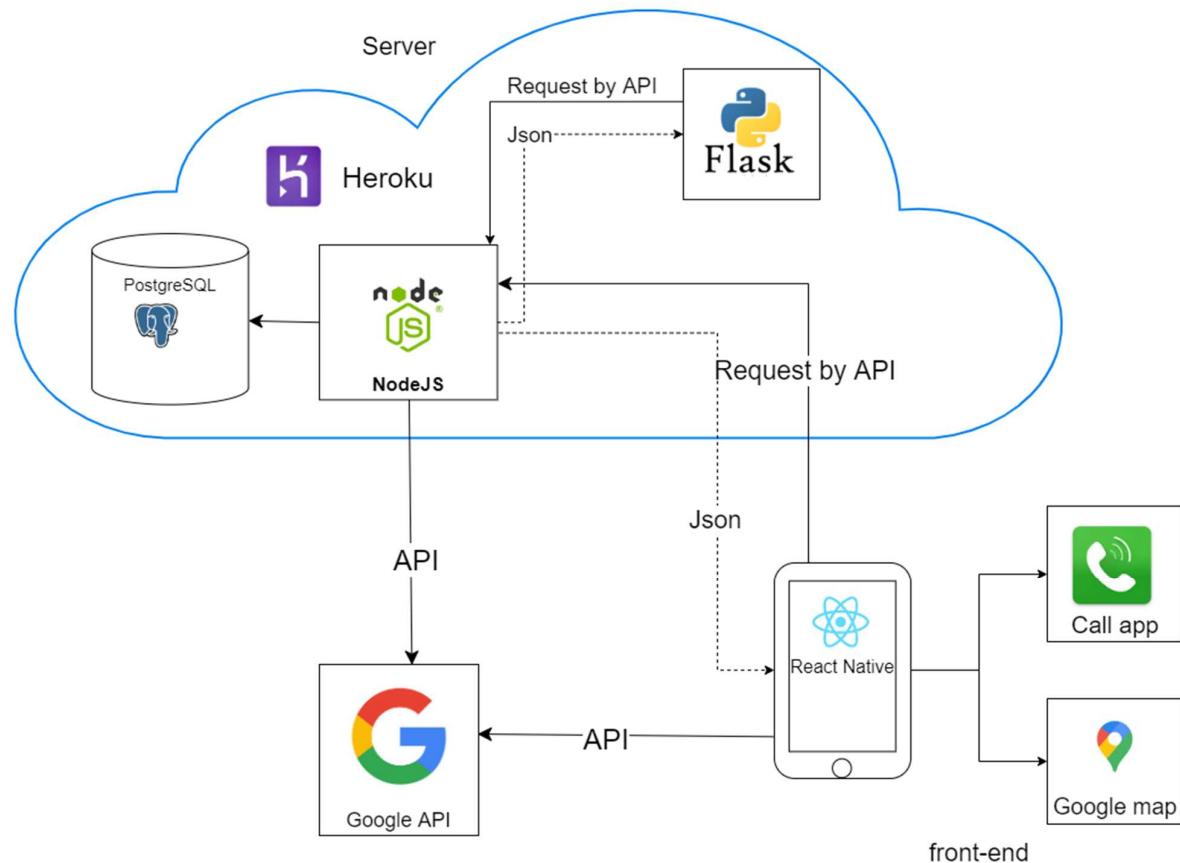
### **1.4. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC**

- Về lý thuyết hiểu rõ về các thành phần của hệ thống và cách sử dụng phát triển từng thành phần riêng biệt.
- Xây dựng một cơ sở dữ liệu phong phú.
- Xây dựng được một ứng dụng giao diện đẹp, dễ sử dụng.
- Source code rõ ràng, dễ bảo trì và phát triển về sau. React native phải phân chia thành các component riêng biệt, ExpressJS phải phân bố thư mục rõ ràng, các API phải được đặt trong từng file riêng biệt dễ dàng có thể tìm lại được khi cần thiết.
- Hệ thống quản lý đơn giản, tiện dụng và hiệu quả.

## Phần 2: NỘI DUNG

### CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

#### 1.1. TỔNG QUAN VỀ HỆ THỐNG



Hình 1: Tổng quan về hệ thống

#### 1.1.1. KHÁI QUÁT VỀ HỆ THỐNG:

MEAN STACK là một full-stack JavaScript framework thân thiện với người dùng cho việc xây dựng các trang web và ứng dụng động. Được cung cấp miễn phí, mã nguồn mở, xây dựng bằng JavaScript để phát triển ứng dụng Single Page Application (SPA). Đây là một trong những thế mạnh khiến MEAN chạy trên mọi cấp độ của ứng dụng, trở thành một cách hiệu quả và hiện đại để phát triển web. [3]

#### 1.1.2. CÁC THÀNH PHẦN CỦA HỆ THỐNG:

- Postgres-SQL - RDBMS

## CHƯƠNG 1 – CƠ SỞ LÝ THUYẾT

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở.

RDBMS là một "hệ thống phần mềm cho phép người dùng xác định, tạo, duy trì và kiểm soát quyền truy cập vào cơ sở dữ liệu". RDBMS là một phần mở rộng của từ viết tắt đó đôi khi được sử dụng khi cơ sở dữ liệu cơ bản là quan hệ.

- **ExpressJS - Web framework**

ExpressJS là một web application framework cho NodeJS, cung cấp các tính năng mạnh mẽ cho việc xây dựng một ứng dụng web đúng nghĩa hoặc ngược lại. Express cũng có thể sử dụng để xây dựng một API mạnh mẽ và thân thiện với người dùng, vì nó cung cấp rất nhiều tiện ích HTTP và middleware cho việc kết nối.

- **React Native - Front-end framework**

React Native là một framework được tạo bởi Facebook, cho phép developer xây dựng các ứng dụng di động trên cả Android và iOS chỉ với một ngôn ngữ lập trình duy nhất: JavaScript.

Sự ra đời của React Native là lời giải cho bài toán liên quan đến hiệu năng Hybrid và sự phức tạp khi phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động trước đó.

- **NodeJS - Web server (Back-end)**

Node.js nổi tiếng với việc xây dựng các ứng dụng thời gian thực có thể mở rộng và các ứng dụng web nhanh. Node.js có thể biên dịch mã JavaScript thành mã máy gốc trước khi thực thi mã. Tính năng này rất hữu ích trong việc xây dựng các ứng dụng web JavaScript phía máy chủ. Các ứng dụng dựa trên kết thúc mạnh mẽ phục vụ để chứa các chức năng quan trọng cao bằng cách sử dụng Node.js. Tính năng được ngưỡng mộ nhất của Node.js là nó có thể sử dụng JavaScript làm cả ngôn ngữ front end và back end.

## 1.2. NODEJS

### 1.2.1. KHÁI QUÁT VỀ NODEJS

- NodeJS là chữ N trong “MEAN”, tuy nằm ở cuối nhưng NodeJS lại là có vai trò quan trọng nhất, nó là nền tảng, là khởi nguồn cho cả stack này được tạo nên.
- NodeJS là một môi trường thực thi (runtime environment) được ra mắt vào năm 2009 phép JavaScript chạy trên Server-side.
- NodeJS là một mã nguồn mở, sử dụng mô hình event-driven và non-blocking I/O được build trên Google V8 JavaScript engine để thực thi mã lệnh. NodeJS thường được sử dụng trong việc xây dựng một môi trường cho các máy chủ và ứng dụng mạng, tuy nhiên khả năng và ứng dụng của NodeJS càng ngày càng được mở rộng.  
[5]
- NodeJS là nền tảng cơ bản nhất để xây dựng MEAN stack.

### 1.2.2. ĐẶC ĐIỂM GIÚP NODEJS TRỞ NÊN MẠNH MẼ VÀ NỐI BẬT TRÊN THỊ TRƯỜNG GẦN ĐÂY?

#### **Nhanh:**

Một trong những ưu điểm làm nổi bật NodeJS nhất là tốc độ. Vì được thực thi trên Google V8 engine, mã JavaScript sẽ được build thành ra mã Assembly nên tốc độ xử lý của các ứng dụng NodeJS nhanh hơn Java, Python, PHP. Và bên cạnh đó còn có các cơ chế Non-blocking I/O và Event Driven giúp mã lệnh thực thi trên NodeJS đạt hiệu năng cao hơn, ít tốn tài nguyên hơn. [6]

#### **Đơn giản:**

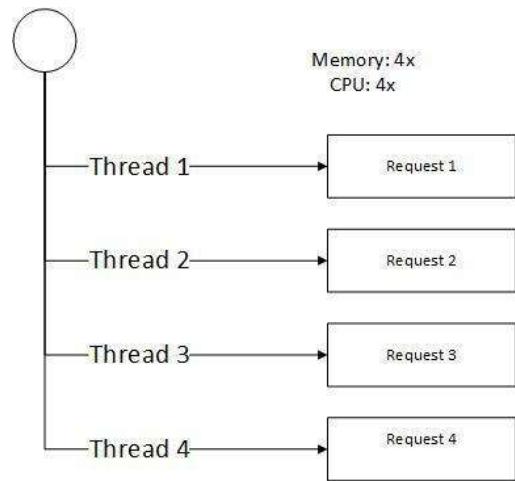
Không phải vì NodeJS là công nghệ mới mà việc học và sử dụng NodeJS khó khăn, ngược lại có thể rất dễ đối với những lập trình viên đã quen với JavaScript. Bởi vì NodeJS chỉ là một môi trường thực thi, còn mã lệnh được viết trên đó vẫn là JavaScript, vậy những lập trình viên Front-end đang sử dụng JavaScript đều có thể học và sử dụng NodeJS một cách nhanh chóng.

Một điều đơn giản hơn nữa của NodeJS là được ứng dụng trong MEAN stack, một Technical Stack được viết toàn bộ bằng JavaScript (một ngôn ngữ trước đây bị đánh giá thấp là chỉ có thể chạy trên trình duyệt web). Điều này giúp ích rất nhiều trong việc đỡ phải gắp rắc rối vì sử dụng nhiều ngôn ngữ lập trình một lúc.

### **Hiệu năng cao:**

Nhờ vào cơ chế Single Thread và Event Loop, mà JavaScript tiết kiệm tài nguyên hơn cho hệ thống

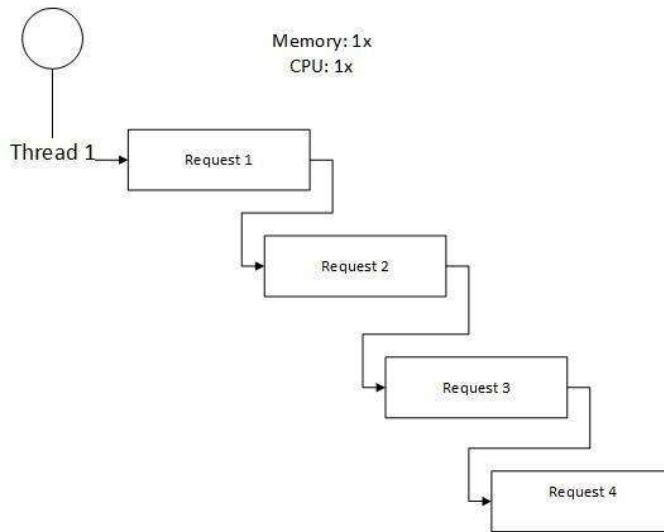
### **Mutil thread trên hệ thống cũ:**



*Hình 2: Multi thread trên hệ thống cũ*

Những hệ thống cũ, mỗi khi server tiếp nhận một request là bản thân server sẽ tự tách ra một Thread mới để xử lý request đó, việc này sẽ làm tốn rất nhiều tài nguyên nếu như số lượng request được tăng cao (1 ngàn đến 1 triệu request) vào cùng một thời điểm, dễ quá tải server.

### **Single thread trên những hệ thống mới:**



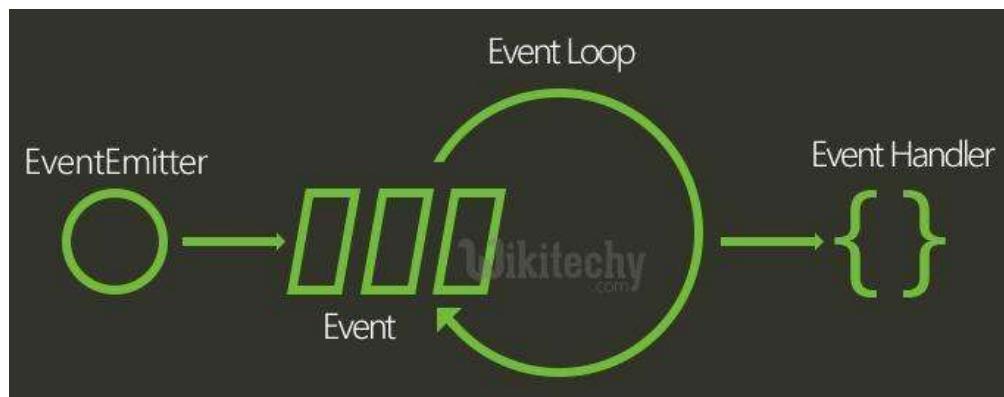
**Hình 3: Single Thread**

Với cơ chế Single Thread của NodeJS, thật ra ở bên trong của Thread đơn này, các tác vụ vẫn được xử lý đa luồng nhưng theo kiểu bất đồng bộ (asynchronous)

### Event driven

**Event:** Bất cứ điều gì xảy ra với ứng dụng được viết bằng JavaScript mà có thể phản hồi lại thì đều được gọi là event (sự kiện). JavaScript là ngôn ngữ hướng sự kiện. Có 2 loại sự kiện chính ở trong NodeJS:

- ❖ Sự kiện hệ thống: C++ core từ thư viện có tên libuv. (Ví dụ, đọc xong một tập tin).
- ❖ Sự kiện tùy chỉnh: core JavaScript



**Hình 4: Event driven NodeJS**

## CHƯƠNG 1 – CƠ SỞ LÝ THUYẾT

Event driven là cơ chế mà NodeJS sử dụng để điều khiển luồng sự kiện trong nó. Vì là cơ chế đơn luồng (Single Thread) vì vậy mọi hàm xử lý được viết trong nodejs đều trả về một Event.<sup>[8]</sup>

### 1.2.3. NPM (NODEJS PACKAGE MANAGER):

#### 1.2.3.1. Package manager là gì?

Package Manager (PM) hoặc package management system là một tập hợp các phần mềm dùng để quản lý và tự động hóa việc cài đặt, nâng cấp, gỡ bỏ các phần mềm hoặc thư viện (package).

Trong một dự án cần sử dụng một số thư viện thay vì tự code để sử dụng. Việc sử dụng các thư viện có sẵn trên mạng và nhúng vào dự án khác sẽ gặp những vấn đề như:

- ▲ Khi nhiều thành viên cùng làm chung một dự án, đôi lúc sử dụng những phiên bản riêng, xung đột lẫn nhau. Dự án có thể chạy ở máy này nhưng lại không chạy ở máy khác.
- ▲ Khi commit code vào source control, cần phải commit cả thư viện vào. Điều làm khiến cho dự án rất nhiều và nặng khi commit cũng như tải về.
- ▲ Khi deploy, việc copy cả một thư viện lên rất chậm và mất thời gian.
- ▲ Sử dụng PM, mọi vấn đề đều được giải quyết. Vậy PM xử lý vấn đề này như thế nào?

Thông thường, một hệ thống PM thường bao gồm các thành phần:

- ▲ Package Manager: Cài đặt trên máy developer, quản lý việc cài đặt các package
- ▲ Repository: Nơi chứa các package (trên mạng). Khi cần một package nào đó, PM sẽ tải package đó từ repository về.
- ▲ Local Package Database: Mỗi dự án sẽ có local package database riêng, chứa thông tin (metadata, bao gồm tên package, phiên bản, dependency) của các package trong dự án đó.
- ▲ Với 3 thành phần này, PM giải quyết các vấn đề trên rất dễ dàng:
  - ▲ Mỗi khi cần thư viện, dùng PM để cài đặt. PM sẽ tải thư viện từ Repository về, sau đó cập nhật Local Package Database.
  - ▲ Khi commit code, bạn chỉ cần commit Local Package Database, đây thường chỉ là 1 file text rất nhẹ.

## CHƯƠNG 1 – CƠ SỞ LÝ THUYẾT

- ▲ Khi deploy hoặc share code qua máy khác, sẽ sử dụng PM. Dựa theo các package trong Local Package Database, PM sẽ tải đúng package phiên bản tương ứng.
- ▲ Điều này đảm bảo mọi người đều dùng chung package, chung phiên bản. Code nằm trong source control cũng rất nhẹ, up nhanh tải nhanh mang lại hiệu quả cao.

### 1.2.3.2. Tại sao lại sử dụng npm?

NPM viết tắt của Node Package Manager là một công cụ quản lý các thư viện lập trình Javascript cho Node.js, công cụ này là thật sự cần thiết cho thế giới mã nguồn mở. Trong cộng đồng JavaScript, các lập trình viên chia sẻ hàng trăm nghìn các đoạn code giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các framework. Mỗi đoạn code này có thể phụ thuộc vào rất nhiều các mã nguồn mở khác, thật may mắn khi các công cụ quản lý thư viện ra đời, nếu không sẽ mất rất nhiều công sức trong việc quản lý các thư viện này [10].

### 1.2.4. NHUỘC ĐIỂM CỦA NODEJS

Dễ để cho lập trình viên có kinh nghiệm tìm hiểu nhưng lại khá khó cho người mới bắt đầu. Đặc biệt là những lập trình viên quen với kiểu lập trình tuần tự (Synchronous).

Vì nó là công nghệ mới, vậy nên thường chỉ thích hợp với những start up. Những dự án lớn, quan trọng của các công ty lớn sẽ ít chọn NodeJS để làm.

## 1.3. EXPRESSJS

### 1.3.1. KHÁI QUÁT VỀ EXPRESSJS

Express là chữ “E” trong “MEAN”. ExpressJS là một trong những Framework quan trọng nhất của NodeJS.

Khi xây dựng một web server, có rất nhiều đoạn mã về Routing, về Listening port, configure... phải lặp đi lặp lại rất nhiều và dài dòng, ExpressJS ra đời và giải quyết vấn đề về code dài, code không sạch của lập trình viên back-end khi xây dựng Web server. Đây là một framework tối giản để xây dựng một loạt các ứng dụng web và di động cũng như các API.

Rất nhiều ứng dụng phổ biến và các trang web như MySpace, Geekli.st, Klout, Segment.io và Yummly được xây dựng bởi Express.js. Express.js cung cấp các tính năng khác nhau, như đơn giản hóa nhiều định tuyến, tích hợp cơ sở dữ liệu, ...

### 1.3.2. ĐẶC ĐIỂM CỦA EXPRESSJS

#### Tối ưu việc tạo server:

Như đã được lưu ý ở trên, Node.js là một nền tảng không phải server. Điều này cho phép có thể sáng tạo trong việc thiết lập server và làm những việc mà các web server khác không thể làm. Nó cũng khiến cho việc tạo và chạy các trang web căn bản trở nên khó khăn do phải viết nhiều hơn những lệnh cấu hình.

Express loại bỏ những khó khăn này bằng cách thiết lập một web server để thực hiện việc lắng nghe các yêu cầu gửi đến và trả về các yêu cầu liên quan. Ngoài ra, nó cũng định nghĩa một cấu trúc thư mục. Một trong những thư mục được thành lập để chứa các file css, js, images, ...cho ứng dụng. Chúng ta phải cấu hình những file này trực tiếp trong Node.js, nhưng với Express sẽ làm luôn việc này.<sup>[12]</sup>

#### Routing URLs to responses:

Một trong những tính năng tuyệt vời của Express là cung cấp một giao diện đơn giản để điều hướng URL đến một đoạn code. Nó có thể phục vụ cho những trang web tĩnh, đọc thông tin từ database, hay ghi vào database. Giao diện đơn giản và nhất quán.

Những điều Express làm ở trên để làm giảm đi sự phức tạp của Node.js thuận, để viết code nhanh hơn, dễ dàng xem lại hơn.

#### Views: HTML responses

Express hỗ trợ một số các templating engine như: Jade, Handlebar, ... để dễ dàng xây dựng một trang HTML một cách thông minh, sử dụng lại các thành phần cũng như dữ liệu từ ứng dụng. Express biên dịch tổng hợp lại và hiển thị chúng ở browser như HTML.

#### Ghi nhớ người dùng với phiên hỗ trợ

Express có khả năng sử dụng các sessions để có thể xác nhận cá nhân những người đã truy cập trong tất cả các yêu cầu và trang.

### 1.3.3. RESTFUL API

RESTful API là giao diện lập trình ứng dụng (API), sử dụng HTTP method như GET, POST, PUT, DELETE, ... dữ liệu.

RESTful API – còn dịch vụ web Restful (RESTful web service) – dựa trên REST (**R**epresentational **S**tate **T**ransfer) là một kiểu kiến trúc được sử dụng trong việc giao tiếp giữa các máy tính (máy tính cá nhân và máy chủ của trang web) trong việc quản lý các

## CHƯƠNG 1 – CƠ SỞ LÝ THUYẾT

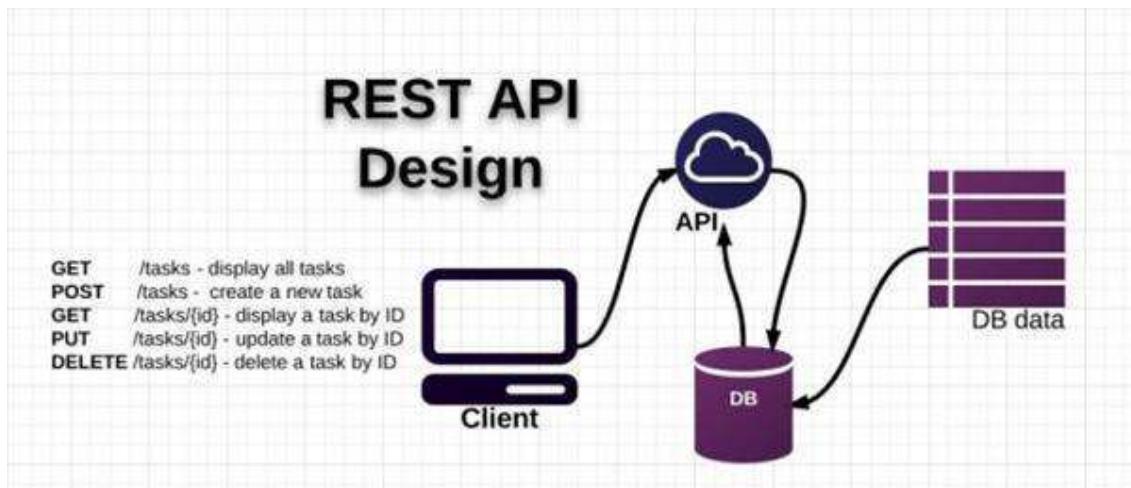
tài nguyên trên internet. Các ứng dụng sử dụng kiến trúc REST này thì sẽ được gọi là ứng dụng phát triển theo kiểu RESTful.<sup>[13]</sup>

Để được xem là một hệ thống REST, hệ thống cần đảm bảo được 5 ràng buộc (constraints) sau:

1. Kiến trúc client-server: Server sẽ là tập hợp các services nhỏ xử lý các request từ client, việc tách biệt giúp tăng tính linh hoạt của client cũng như khả năng mở rộng của server mà vẫn đảm bảo được giao tiếp.
2. Stateless (phi trạng thái): Server và client sẽ không lưu trạng thái của nhau, mỗi request và response sẽ chứa đầy đủ thông tin, tách biệt với nhau, điều này giúp hệ thống dễ phát triển và mở rộng
3. Cacheability (có khả năng cache): Các response có thể được lấy ra từ cache của server giúp giảm thiểu thời gian xử lý, các request phải đảm bảo tính duy nhất để response không bị nhầm lẫn với các request khác. Layered system (phân lớp hệ thống): Hệ thống được chia làm nhiều lớp, việc giao tiếp của 1 lớp chỉ được tiến hành với lớp ở trên và lớp ở dưới của nó, khả năng cân bằng tải (load balancing) và cache dữ liệu trong hệ thống cũng sẽ được cải thiện.
4. Uniform interface (chuẩn hoá các interface): Đây là ràng buộc quan trọng nhất của hệ thống REST.

Để đảm bảo ràng buộc này, hệ thống tập trung vào việc xử lý các tài nguyên (resource). Mỗi một resource sẽ được xác định (định danh) bằng một URI (Uniform Resource Identifier) riêng biệt.

Việc tạo ra quy ước chuẩn để giao tiếp giữa các thành phần trong hệ thống giúp tất cả các thiết bị REST đều có thể truy cập đến tài nguyên trên server, điều này lý giải vì sao rất nhiều API sử dụng REST.<sup>[13]</sup>



Hình 5: Thiết kế REST API

Để tạo một tài nguyên trên máy chủ, cần sử dụng phương thức POST.

Để truy xuất một tài nguyên, sử dụng GET.

Để thay đổi trạng thái một tài nguyên hoặc để cập nhật, sử dụng PUT.

Để huỷ bỏ hoặc xóa một tài nguyên, sử dụng DELETE.

### 1.3.4. TẠI SAO LẠI SỬ DỤNG EXPRESS TRONG VIỆC QUẢN LÝ REST API

NodeJS và nó còn đặc biệt hơn nữa bản thân nó được xây dựng để làm những công việc http method (post, get, put, ...) cũng là điều mà bản thân Rest API làm. Vậy còn gì thích hợp hơn việc dùng Express để quản lý và viết các API Restful.

Cộng đồng của ExpressJS rất lớn, vì vậy để có thể tìm ra các hướng dẫn xây dựng các web server với Express là Rest API là cực kì nhiều và cực kì đơn giản, điều này giúp các lập trình viên mới dễ dàng tiếp cận công nghệ.<sup>[15]</sup>

## 1.4. POSTGRES-SQL

### 1.4.1. KHÁI QUÁT VỀ RDBMS

Khi làm việc với database, những hệ quản trị cơ sở dữ liệu thường sử dụng: SQLServer, MySQL, Oracle, ... Điểm chung của những database này là đều sử dụng ngôn ngữ SQL để truy vấn dữ liệu. Ngôn ngữ này truy vấn trên nền một RDBMS (Hệ quản trị CSDL quan hệ) - hệ thống phần mềm cho phép người dùng xác định, tạo, duy trì và kiểm soát quyền truy cập vào cơ sở dữ liệu.

RDBMS đã là một lựa chọn phổ biến để lưu trữ thông tin trong cơ sở dữ liệu được sử dụng cho hồ sơ tài chính, sản xuất và thông tin hậu cần, dữ liệu nhân sự và các ứng dụng khác kể từ những năm 1980. Cơ sở dữ liệu quan hệ thường thay thế cơ sở dữ liệu phân cấp kẽ thưa và cơ sở dữ liệu mạng, vì RDBMS dễ triển khai và quản trị hơn.

### 1.4.2. KHÁI QUÁT VỀ POSTGRES-SQL

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay.

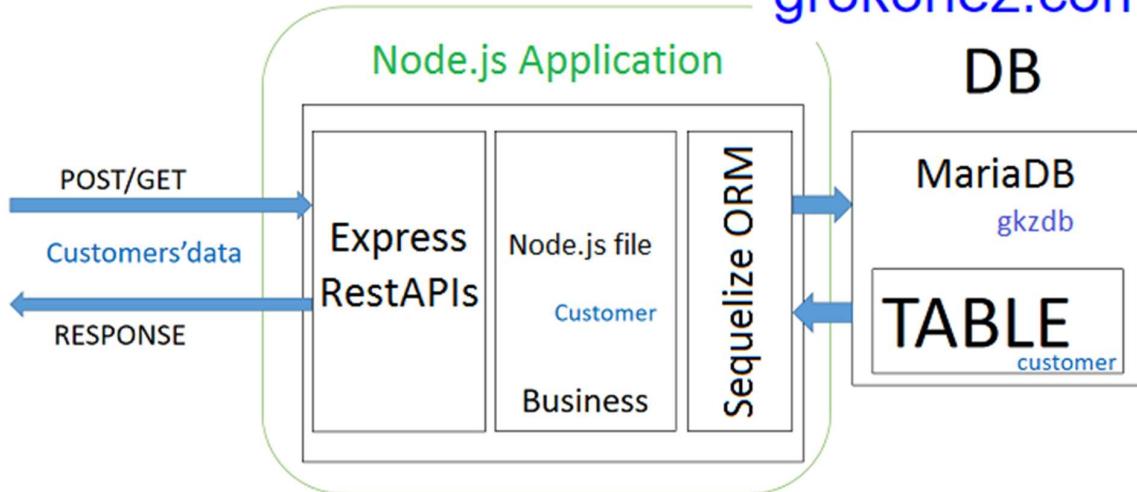
PostgreSQL được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, PostgreSQL sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.

PostgreSQL là một phần mềm mã nguồn mở miễn phí. Mã nguồn của phần mềm khả dụng theo license của PostgreSQL, một license nguồn mở tự do. Theo đó, bạn sẽ được tự do sử dụng, sửa đổi và phân phối PostgreSQL dưới mọi hình thức.

PostgreSQL không yêu cầu quá nhiều công tác bảo trì bởi có tính ổn định cao. Do đó, nếu bạn phát triển các ứng dụng dựa trên PostgreSQL, chi phí sở hữu sẽ thấp hơn so với các hệ thống quản trị dữ liệu khác.

### 1.4.3. SEQUELIZE

Sequelize là một ORM dành cho Node.js và io.js hỗ trợ bạn truy cập một cách dễ dàng đến PostgreSQL, MySQL, MariaDB, SQLite và MSSQL cùng với các tính năng như là relations, transaction, replication ...

**Hình 6: Mô hình hoạt động sequelize**

Khi xây dựng Back-end, Sequelize cung cấp các hàm giúp express thực hiện những công việc thông thường giữa web server và database như:

- + Kết nối cơ sở dữ liệu thông qua config.
- + Truy vấn trong cơ sở dữ liệu
- + Tạo mới và lưu
- + Xóa
- + Cập nhật

## 1.5. REACT NATIVE

### 1.5.1. GIỚI THIỆU VỀ REACT NATIVE

React Native là một framework được tạo bởi Facebook, cho phép developer xây dựng các ứng dụng di động trên cả Android và iOS chỉ với một ngôn ngữ lập trình duy nhất: JavaScript.

Sự ra đời của React Native là lời giải cho bài toán liên quan đến hiệu năng Hybrid và sự phức tạp khi phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động trước đó.

### 1.5.2. ƯU/ NHƯỢC CỦA REACT NATIVE

- **Ưu điểm:**

- + Sử dụng ít native code hơn
- + Thời gian phát triển ứng dụng được rút ngắn
- + Tiết kiệm chi phí nhân lực cho doanh nghiệp

## CHƯƠNG 1 – CƠ SỞ LÝ THUYẾT

- + Khả năng tái sử dụng code cao (75% – 80%)
  - + Trải nghiệm người dùng rất tốt
  - + Không cần nhiều kiến thức nền
- **Nhược điểm:**
    - + Chỉ hỗ trợ 2 nền tảng Android và IOS
    - + Hiệu năng sẽ thấp hơn với ứng dụng thuần native code
    - + Bảo mật không cao
    - + Có một số thư viện cần trả phí
    - + Khả năng tùy biến không thực sự tốt với một vài module

### 1.5.3. NGÔN NGỮ LẬP TRÌNH JAVASCRIPT

JavaScript là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua.

Nó cũng là một trong số 3 ngôn ngữ chính của lập trình web:

- + HTML: Giúp bạn thêm nội dung cho trang web.
- + CSS: Định dạng thiết kế, bố cục, phong cách, canh lề của trang web.
- + JavaScript: Cải thiện cách hoạt động của trang web.

JavaScript có thể học nhanh và dễ dàng áp dụng cho nhiều mục đích khác nhau, từ việc cải thiện tính năng của website đến việc chạy game và tạo phần mềm nền web. Hơn nữa, có hàng ngàn mẫu template JavaScript và ứng dụng ngoài kia, nhờ vào sự công hiến của cộng đồng, đặc biệt là Github.

- **Ưu điểm:**
  - + Không cần một compiler vì web browser có thể biên dịch nó bằng HTML.
  - + Nó dễ học hơn các ngôn ngữ lập trình khác.
  - + Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn.
  - + Nó có thể được gắn trên một số element của trang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới.
  - + JS hoạt động trên nhiều trình duyệt, nền tảng, vân vân.
  - + Bạn có thể sử dụng JavaScript để kiểm tra input và giảm thiểu việc kiểm tra thủ công khi truy xuất qua database.
  - + Nó giúp website tương tác tốt hơn với khách truy cập.
  - + Nó nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác.

● **Nhược điểm:**

- + Dễ bị khai thác;
- + Có thể được dùng để thực thi mã độc trên máy tính của người dùng;
- + Nhiều khi không được hỗ trợ trên mọi trình duyệt;
- + JS code snippets lớn;
- + Có thể bị triển khai khác nhau tùy từng thiết bị dẫn đến việc không đồng nhất.

## CHƯƠNG 2: PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

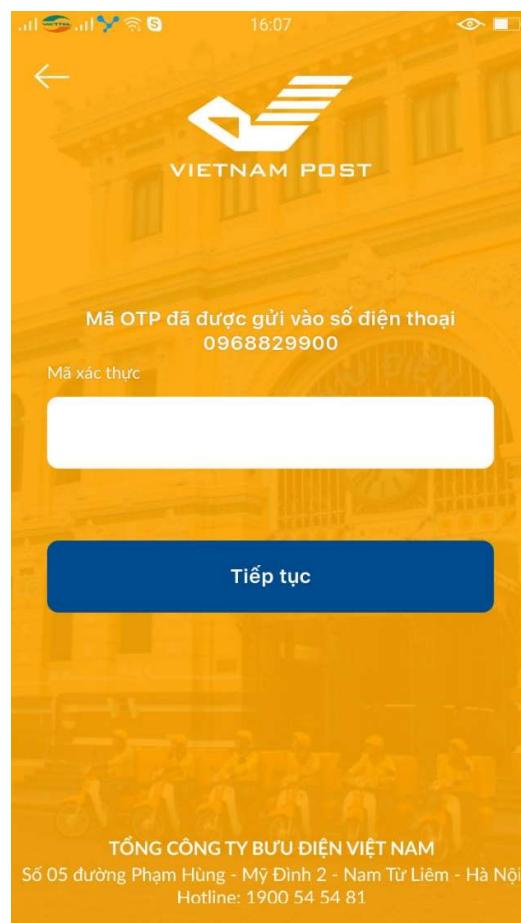
### 2.1. KHẢO SÁT HIỆN TRẠNG

Hiện nay số lượng khách hàng sử dụng các ứng dụng mua bán ngày càng nhiều.

Song các đơn hàng ngày càng tăng lên. Vì vậy nhu cầu giao nhận hàng cũng từ đó được tăng theo. Nhiều ứng dụng mua hàng như shopee, sendo đã liên kết với các công ty giao hàng để khách hàng có thể chọn ra cách giao hàng nhanh và phù hợp, hoặc tự tạo ra đội giao hàng của ứng dụng đó như tiki, lazada, ... để dễ dàng quản lý. Và hiện tại bây giờ có rất nhiều các đơn vị vận chuyển được mở ra nhằm đáp ứng nhu cầu của các đơn vị mua bán.

*Nhận xét riêng về từng đơn vị:*

**VNPost - EMS**



*Hình 7: VNPost*

#### Ưu điểm:

- Đại lý rộng khắp trên cả nước, dễ dàng đặt dịch vụ vận chuyển.
- Ít xảy ra vấn đề về mất hàng hoá, độ tin cậy, uy tín cao.

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

- Nhận hàng đến 5h30 chiều nên có thể xử lý các đơn hàng gấp trong ngày.

### **Khuyết điểm:**

- Giao hàng chậm hơn so với các dịch vụ cùng loại của công ty khác.
- Người gửi hàng phải mang hàng đến bưu điện đóng gói chứ không có dịch vụ mang hàng đi.
- Sau khi phát hàng 2 lần nếu vẫn gặp trục trặc không nhận được, người nhận phải tự đến lấy hàng.

### **LazadaExpress**



*Hình 8: LazadaExpress*

### **Ưu điểm:**

- Vì là cùng đơn vị nên giá giao hàng ko thay đổi trên từng vùng
- Mô hình vận chuyển của đơn vị mua bán

### **Khuyết điểm:**

- Giao hàng khá chậm

### **Viettel Post**



*Hình 9: ViettelPost*

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### Ưu điểm:

- Mang lưới rộng khắp cả nước.
- Giao nhận hàng tận nơi, không cần mang hàng ra bưu cục Viettel post.
- Khách hàng sử dụng dịch vụ chuyển phát trên địa bàn Hà Nội, lựa chọn Viettel post giúp tối ưu chi phí dịch vụ.

### Khuyết điểm:

- Cước phí vận chuyển: Giá hơi cao so với các dịch vụ cùng loại
- Chất lượng dịch vụ: Giao hàng khá chậm

### Giao Hàng Tiết Kiệm



*Hình 10: GiaoHangTietKiem*

### Ưu điểm:

- Chất lượng dịch vụ: Dịch vụ bảo hiểm hàng hoá
- Cước phí vận chuyển: Mức giá vận chuyển hấp dẫn, có nhiều chế độ ưu đãi

### Khuyết điểm:

- Chất lượng dịch vụ: Chỉ chuyển được trong nội thành Hà Nội, TP Hồ Chí Minh và Đà Nẵng

### Đánh giá chung:

Nhìn chung tất cả các ứng dụng giao hàng đều đáp ứng đủ nhu cầu khách hàng. Song vẫn còn một số vấn đề vẫn còn gây khó khăn cho khách hàng. Thứ nhất là về việc tra cứu. Đa số các dịch vụ vận chuyển này đều sẽ hợp tác với các dịch vụ mua bán. Vì vậy việc tra cứu sẽ gây bất lợi hơn, khách hàng phải vào website của dịch vụ vận chuyển để tra cứu chi tiết đơn hàng. Thứ hai là về thời gian giao hàng. Đa số các dịch vụ vận chuyển

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

đều không cho khách hàng có thể chọn lựa thời gian mà khách hàng rảnh mà khách hàng phải phụ thuộc hoàn toàn vào các đơn vị vận chuyển. Đôi khi khách hàng không có ở nơi nhận hàng dẫn đến một số bất lợi cho bên khách hàng và shipper.

### Mục tiêu:

Từ những vấn đề trên nhóm chúng em đã xây dựng thuật toán, và ứng dụng giúp khách hàng có thể thuận tiện trong việc chọn thời gian rảnh của mình. Giúp cho khách hàng và shipper giao nhận hàng chỉ trong một lần duy nhất. Ứng dụng sẽ thông báo cho khách hàng địa điểm giao hàng và thời gian đăng ký chính xác từng phút. Giúp khách hàng có thể chuẩn bị tiền và biết thời gian giao hàng.

## 2.2. XÁC ĐỊNH YÊU CẦU

### Lưu trữ:

- + Thông tin người dùng (khách hàng, shipper).
- + Đơn hàng.
- + Sản phẩm đã mua.
- + Địa chỉ.
- + Chuỗi giao hàng.

### Tác vụ:

- + Cho phép người dùng đăng ký tài khoản.
- + Thêm địa chỉ, cập nhật địa chỉ.
- + Đặt hàng.

### Giao diện:

- + Thuận tiện, dễ sử dụng.
- + Đẹp, thân thiện người dùng.

### Tìm kiếm:

- + Tìm kiếm, lựa chọn sản phẩm.
- + Lựa chọn chuỗi giao hàng.

### Quản lý:

- + Xem đơn hàng, danh sách sản phẩm.
- + Xem chuỗi giao hàng, bản đồ.

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### 2.2.1. Xác định yêu cầu chức năng hệ thống:

*Bảng 1: Xác định yêu cầu chức năng ứng dụng khách hàng*

STT	Nội dung	Mô tả chi tiết	Ghi chú
1	Đặt hàng	Lựa chọn sản phẩm và đặt hàng	Đơn hàng được tạo cùng với các sản phẩm đã mua
2	Xem đơn hàng	Khách hàng có thể xem các đơn hàng theo khoảng thời gian nào đó	
3	Xem các sản phẩm đã mua	Khách hàng có thể xem các sản phẩm đã mua trong từng đơn hàng	
4	Tìm kiếm sản phẩm theo tên sản phẩm	Khách hàng có thể tìm kiếm theo tên sản phẩm	
5	Tìm kiếm sản phẩm theo loại sản phẩm	Khách hàng có thể tìm kiếm sản phẩm theo từng loại sản phẩm	Khách hàng ánh vào loại sản phẩm muốn tìm
6	Xem thông tin sản phẩm	Khách hàng xem thông tin của từng sản phẩm	Khách hàng ánh vào sản phẩm muốn xem thông tin
7	Đăng nhập	Khách hàng đăng nhập để sử dụng ứng dụng	
8	Đăng xuất	Khách hàng đăng	

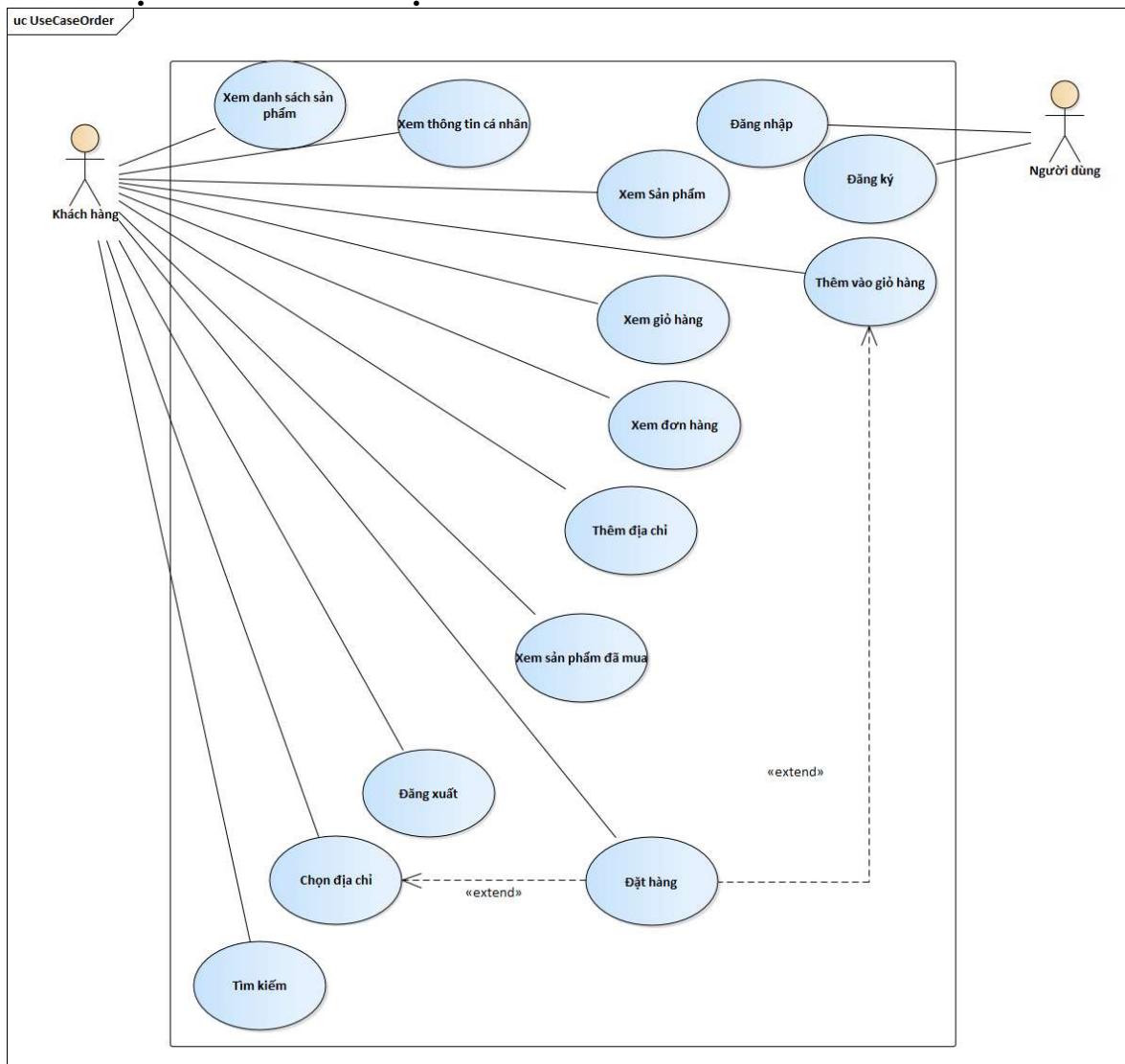
## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

		xuất khói ứng dụng	
9	Đăng ký	Khách hàng đăng ký tài khoản	

*Bảng 2: Xác định yêu cầu chức năng ứng dụng shipper*

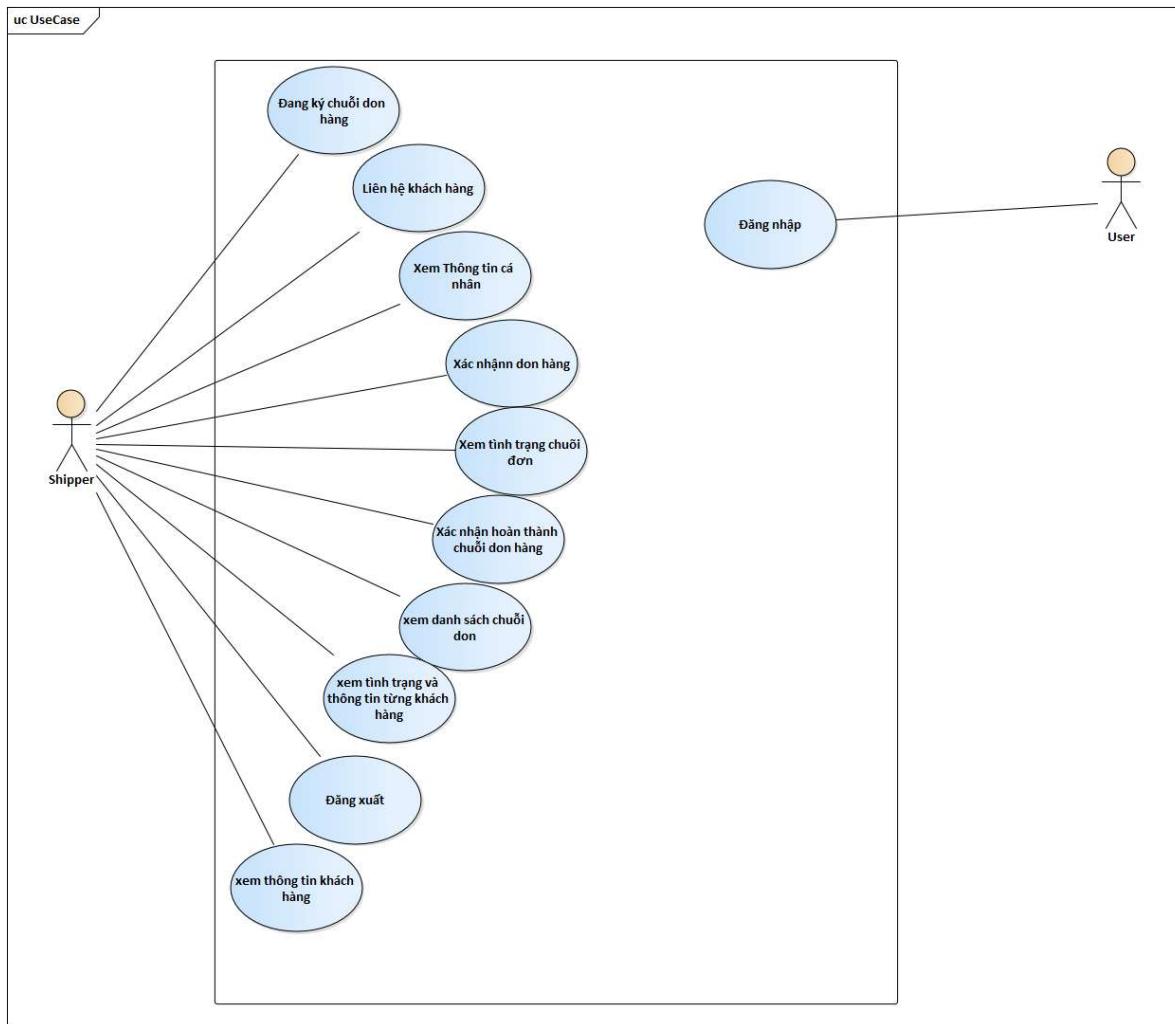
STT	Nội dung	Mô tả chi tiết	Ghi chú
1	Đặt hàng	Lựa chọn sản phẩm và đặt hàng	Đơn hàng được tạo cùng với các sản phẩm đã mua
2	Xem đơn hàng	Khách hàng có thể xem các đơn hàng theo khoảng thời gian nào đó	
3	Xem các sản phẩm đã mua	Khách hàng có thể xem các sản phẩm đã mua trong từng đơn hàng	
4	Tìm kiếm sản phẩm theo tên sản phẩm	Khách hàng có thể tìm kiếm theo tên sản phẩm	
5	Tìm kiếm sản phẩm theo loại sản phẩm	Khách hàng có thể tìm kiếm sản phẩm theo từng loại sản phẩm	Khách hàng ấn vào loại sản phẩm muốn tìm
6	Xem thông tin sản phẩm	Khách hàng xem thông tin của từng sản phẩm	Khách hàng ấn vào sản phẩm muốn xem thông tin

### 2.2.2. Lược đồ Use case và đặc tả



**Hình 11: Lược đồ Use Case và đặc tả**

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU



**Hình 12: Lược đồ Use Case**

Nhận diện tác nhân và chức năng trong sơ đồ Use case:

**Bảng 3: Nhận diện tác nhân và chức năng**

Tác nhân	Chức năng
Khách hàng	Xem sản phẩm
	Xem Danh sách sản phẩm
	Thêm sản phẩm vào giỏ hàng
	Xem giỏ hàng
	Chọn địa chỉ
	Thêm địa chỉ

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

	Xem sản phẩm đã mua
	Xem đơn hàng
	Đặt hàng
	Xem thông tin cá nhân
	Đăng xuất
Shipper	Đăng ký chuỗi đơn hàng
	Xem danh sách chuỗi đơn hàng
	Xem tình trạng chuỗi đơn hàng
	Xác nhận tình trạng chuỗi đơn hàng
	Xem tình trạng đơn hàng
	Xem thông tin, địa chỉ khách hàng
	Liên hệ khách hàng
	Xác nhận đơn hàng
	Xem thông tin cá nhân
	Đăng Xuất
User	Đăng nhập
	Đăng ký

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

**Bảng 4: Bảng mô tả chức năng**

STT	Chức năng	Mô tả
1	Xem danh sách sản phẩm	Khách hàng các sản phẩm đang được bán tại Ứng dụng
2	Xem sản phẩm	Khách hàng sau khi chọn vào sản phẩm muốn coi có thể coi được chi tiết sản phẩm mình vừa chọn
3	Thêm vào giỏ hàng	Khách hàng khi ưng ý với sản phẩm mình muốn có thể thêm sản phẩm đó vào giỏ hàng để tiến hành đặt hàng
4	Xem giỏ hàng	Khách hàng có thể xem được những sản phẩm mình đã cho vào từ trước để tiến hành đặt hàng sản phẩm
5	Chọn địa chỉ	Khách hàng sau khi thêm hàng vào giỏ hàng sẽ chọn địa chỉ của khách hàng vào để shipper có thể giao hàng đến địa chỉ mong muốn của khách hàng
6	Thêm địa chỉ	Nếu chưa có địa chỉ hoặc chưa có địa chỉ khách hàng muốn giao, khách hàng có thể thêm địa chỉ mới theo ý muốn của khách hàng
7	Đặt hàng	Khách hàng sau khi thêm sản phẩm vào giỏ hàng và chọn được địa chỉ mong muốn thì sẽ tiến hành đặt hàng
8	Xem đơn hàng	Khi đã đặt hàng thành công khách hàng có thể xem lại đơn hàng về giá và địa chỉ
9	Xem sản phẩm đã mua	Khách hàng sau khi đặt hàng có thể xem chi tiết đơn hàng coi đã mua những gì
10	Xem thông tin cá nhân	Khách hàng vào để xem thông tin của bản thân
11	Đăng xuất	Khách hàng sau khi sử dụng xong hoặc muốn chuyển tài khoản có thể đăng xuất tài khoản hiện tại

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

- |    |                                     |  |
|----|-------------------------------------|--|
| 12 | Xem danh sách chuỗi đơn hàng        | Shipper vào trang chuỗi đơn hàng để xem danh sách chuỗi đơn hàng   |
| 13 | Đăng ký chuỗi đơn hàng              | Shipper sau khi xem chuỗi đơn hàng ưng ý thì có thể tiến hàng đăng ký chuỗi đó để đi giao  |
| 14 | Xem thông tin địa chỉ khách hàng    | Shipper có thể xem được tên, địa chỉ, số điện thoại và số tiền thu từ khách hàng   |
| 15 | Liên hệ khách hàng                  | Shipper liên hệ với khách hàng thông qua số điện thoại   |
| 16 | Xác nhận đơn hàng                   | Shipper sau khi giao hàng thành công hay thất bại thì xác nhận đơn hàng của khách hàng   |
| 17 | Xem tình trạng đơn hàng trong chuỗi | Shipper có thể xem tình trạng của từng đơn hàng trong chuỗi, xem thử mình đã giao bao nhiêu đơn rồi và còn lại bao nhiêu đơn nữa |
| 18 | Xem tình trạng chuỗi đơn hàng       | Shipper có thể xem xem mình đã đăng ký thành công chuỗi đơn hàng chưa, và giao đã hoàn thành hết chưa.                           |
| 19 | Xác nhận hoàn thành chuỗi đơn hàng  | Sau khi giao hàng hết, Shipper tiến hành xác nhận hoàn thành chuỗi đơn hàng đó để có thể đăng ký chuỗi đơn hàng khác             |
| 20 | Xem thông tin cá nhân               | Shipper xem thông tin cá nhân của bản thân   |
| 21 | Đăng xuất                           | Shipper thoát tài khoản hiện tại   |
| 22 | Đăng nhập                           | Người dùng có thể đăng nhập vào ứng dụng   |
| 23 | Đăng ký                             | Khi người dùng chưa có tài khoản thì sẽ tiến hành đăng ký  |

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### Đặc tả Use Case

#### 2.2.3.1. Use case Đăng ký

USE CASE - RT_12			
Use Case No.	REG_01	Use Case Version	2.0
Use Case Name	Người dùng đăng ký tài khoản		
Author	Trần Cao Quyền		
Date	10/12/2020	Priority	normal

**Actor:**

- Người dùng

**Summary:**

- Người dùng đăng ký tài khoản

**Goal:**

- Người dùng đăng ký tài khoản

**Triggers:**

- Khách hàng ấn vào ô tìm kiếm

**Preconditions:**

- Màn hình đang ở Trang đăng nhập

**Post conditions:**

- Success: đăng ký thành công
- Fail: Hệ thống hiển thị Warn message.

**Main Success Scenario:**

Step	Actor Action	System Response
1	Người dùng chọn mục đăng ký	Ứng dụng chuyển đến màn hình đăng ký
2	Người dùng điền thông tin	Ứng dụng hiện thông tin người dùng vừa điền
3	Người dùng ấn nút đăng ký	Ứng dụng thông báo đăng ký thành công và chuyển đến trang đăng nhập

**Alternative Scenario:**

Trùng username

No	Actor Action	System Response
3	Người dùng ấn nút đăng ký	Ứng dụng thông báo đăng ký trùng tên người dùng

**Relationships:**

**Business Rules:**

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### 2.2.3.2. Use case Đăng nhập

USE CASE - RT_12			
Use Case No.	LOGIN_01	Use Case Version	2.0
Use Case Name	Đăng nhập		
Author	Trần Cao Quyền		
Date	10/12/2020	Priority	normal

**Actor:**

- User

**Summary:**

- User đăng nhập

**Goal:**

- User đăng nhập thành công

**Triggers:**

- User ấn vào nút đăng nhập

**Preconditions:**

- Điền tên đăng nhập
- Điền mật khẩu

**Post conditions:**

- Success: Truy cập vào app tới Screen tiếp theo
- Fail: Thông báo sai mật khẩu

**Main Success Scenario:**

Step	Actor Action	System Response
1	User điền thông tin tài khoản	Thông tin tài khoản được điền vào khung nhập một cách chính xác
2	User nhấn nút đăng nhập	Đăng nhập thành công, dẫn tới màn hình tiếp theo

**Alternative Scenario:**

**Sai mật khẩu:**

No	Actor Action	System Response

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

1	User nhấn nút đăng nhập	Hệ thống hiển thị thông báo sai mật khẩu
<b>Sai tên tài khoản</b>		
<b>No</b>	<b>Actor Action</b>	<b>System Response</b>
1	User nhấn nút đăng nhập	Hệ thống hiển thị thông báo không tìm thấy tài khoản
<b>Relationships:</b> RES_02.		
<b>Business Rules:</b>		

### 2.2.3.3. Use case Xem danh sách sản phẩm

USE CASE – RT_12			
<b>Use Case No.</b>	SEE_LIST_PRODUCT_01	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Khách hàng xem danh sách sản phẩm		
<b>Author</b>	Trần Cao Quyền		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal
<b>Actor:</b> <ul style="list-style-type: none"> <li>Khách hàng</li> </ul> <b>Summary:</b> <ul style="list-style-type: none"> <li>Khách hàng xem sản phẩm</li> </ul> <b>Goal:</b> <ul style="list-style-type: none"> <li>Khách hàng xem được các sản phẩm</li> </ul>			

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### Triggers:

- Khách hàng ấn vào trang chủ

### Preconditions:

- 

### Post conditions:

- Success: Danh sách sản phẩm hiện ra
- Fail: Hệ thống hiển thị Warn message.

### Main Success Scenario:

Step	Actor Action	System Response
1	Khách hàng ấn vào mục trang chủ	Ứng dụng hiện danh sách sản phẩm

### Alternative Scenario:

#### Không có sản phẩm:

No	Actor Action	System Response
1	Khách hàng ấn vào mục trang chủ	Ứng dụng hiện danh sách sản phẩm trống

### Relationships:

### Business Rules:

- Người đăng nhập với tư cách là Khách hàng.

#### 2.2.3.4. Use case Xem sản phẩm

USE CASE – RT\_12

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

<b>Use Case No.</b>	SEE _PRODUCT_01	<b>Use Case Version</b>	2.0						
<b>Use Case Name</b>	Khách hàng xem chi tiết sản phẩm								
<b>Author</b>	Trần Cao Quyền								
<b>Date</b>	10/12/2020	<b>Priority</b>	normal						
<p><b>Actor:</b></p> <ul style="list-style-type: none"> <li>Khách hàng</li> </ul> <p><b>Summary:</b></p> <ul style="list-style-type: none"> <li>Khách hàng xem chi tiết sản phẩm</li> </ul> <p><b>Goal:</b></p> <ul style="list-style-type: none"> <li>Khách hàng xem được sản phẩm</li> </ul> <p><b>Triggers:</b></p> <ul style="list-style-type: none"> <li>Khách hàng chọn sản phẩm</li> </ul> <p><b>Preconditions:</b></p> <ul style="list-style-type: none"> <li>Trong danh sách sản phẩm có sản phẩm</li> </ul> <p><b>Post conditions:</b></p> <ul style="list-style-type: none"> <li>Success: hiện chi tiết sản phẩm</li> <li>Fail: Hệ thống hiển thị Warn message.</li> </ul> <p><b>Main Success Scenario:</b></p> <table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Khách hàng chọn sản phẩm</td><td>Ứng dụng hiện chi tiết sản phẩm gồm hình ảnh sản phẩm, giá, mô tả, kích cỡ, khối lượng, khuyến mãi</td></tr> </tbody> </table> <p><b>Alternative Scenario:</b></p> <p><b>Relationships:</b></p> <p><b>Business Rules:</b></p> <ul style="list-style-type: none"> <li>Người đăng nhập với tư cách là Khách hàng.</li> </ul>				Step	Actor Action	System Response	1	Khách hàng chọn sản phẩm	Ứng dụng hiện chi tiết sản phẩm gồm hình ảnh sản phẩm, giá, mô tả, kích cỡ, khối lượng, khuyến mãi
Step	Actor Action	System Response							
1	Khách hàng chọn sản phẩm	Ứng dụng hiện chi tiết sản phẩm gồm hình ảnh sản phẩm, giá, mô tả, kích cỡ, khối lượng, khuyến mãi							

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### 2.2.3.5. Use case Thêm giỏ hàng

USE CASE – RT_12												
Use Case No.	ADD_TO_CART_01	Use Case Version	2.0									
Use Case Name	Khách hàng thêm sản phẩm vào giỏ hàng											
Author	Trần Cao Quyền											
Date	10/12/2020	Priority	normal									
<p><b>Actor:</b></p> <ul style="list-style-type: none"> <li>Khách hàng</li> </ul> <p><b>Summary:</b></p> <ul style="list-style-type: none"> <li>Khách hàng thêm sản phẩm mình cần mua vào giỏ hàng</li> </ul> <p><b>Goal:</b></p> <ul style="list-style-type: none"> <li>Sản phẩm khách chọn được đưa vào giỏ hàng</li> </ul> <p><b>Triggers:</b></p> <ul style="list-style-type: none"> <li>Khách hàng ấn vào hình giỏ hàng</li> </ul> <p><b>Preconditions:</b></p> <ul style="list-style-type: none"> <li>Trong danh sách sản phẩm có sản phẩm</li> </ul> <p><b>Post conditions:</b></p> <ul style="list-style-type: none"> <li>Success: sản phẩm được chọn vào giỏ hàng</li> <li>Fail: Hệ thống hiển thị Warn message.</li> </ul> <p><b>Main Success Scenario:</b></p> <table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Khách hàng ấn vào mục trang chủ</td><td>Ứng dụng hiện danh sách sản phẩm</td></tr> <tr> <td>2</td><td>Khách hàng chọn sản phẩm</td><td>Ứng dụng hiện chi tiết sản phẩm gồm hình tên sản phẩm, giá, mô tả, khích cỡ, khối lượng, khuyến mãi</td></tr> </tbody> </table>				Step	Actor Action	System Response	1	Khách hàng ấn vào mục trang chủ	Ứng dụng hiện danh sách sản phẩm	2	Khách hàng chọn sản phẩm	Ứng dụng hiện chi tiết sản phẩm gồm hình tên sản phẩm, giá, mô tả, khích cỡ, khối lượng, khuyến mãi
Step	Actor Action	System Response										
1	Khách hàng ấn vào mục trang chủ	Ứng dụng hiện danh sách sản phẩm										
2	Khách hàng chọn sản phẩm	Ứng dụng hiện chi tiết sản phẩm gồm hình tên sản phẩm, giá, mô tả, khích cỡ, khối lượng, khuyến mãi										

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

3	Khách hàng chọn biểu tượng giỎ hàng	Ứng dụng thêm sản phẩm đã chọn vào giỎ hàng
---	--	--

**Alternative Scenario:**

**Không có sản phẩm:**

**Relationships:**

**Business Rules:**

- Người đăng nhập với tư cách là Khách hàng.

### 2.2.3.6. Use case Xem giỏ hàng

USE CASE – RT_12			
Use Case No.	SEE _PRODUCT_CART_01	Use Case Version	2.0
Use Case Name	Khách hàng xem sản phẩm trong giỏ hàng		
Author	Trần Cao Quyền		
Date	10/12/2020	Priority	normal

**Actor:**

- Khách hàng

**Summary:**

- Khách hàng xem các sản phẩm đã chọn trong giỏ hàng

**Goal:**

- Khách hàng xem được danh sách sản phẩm

**Triggers:**

- Khách hàng ấn vào biểu tượng giỏ hàng

**Preconditions:**

- Trong danh sách sản phẩm có sản phẩm

**Post conditions:**

- Success: Danh sách sản phẩm hiện ra
- Fail: Hệ thống hiển thị Warn message.

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### Main Success Scenario:

Step	Actor Action	System Response
1	Khách hàng ấn vào mục trang chủ	Ứng dụng hiện danh sách sản phẩm
2	Khách hàng ấn vào biểu tượng giỏ hàng	Ứng dụng chuyển đến màn hình giỏ hàng hiện ra các sản phẩm đã được chọn

### Alternative Scenario:

Chưa thêm sản phẩm vào giỏ hàng:

No	Actor Action	System Response
1	Khách hàng ấn vào biểu tượng giỏ hàng	Ứng dụng chuyển đến màn hình giỏ hàng hiện ra màn hình trống

### Relationships:

### Business Rules:

- Người đăng nhập với tư cách là Khách hàng.

### 2.2.3.7. Use case Chọn địa chỉ

USE CASE – RT_12			
<b>Use Case No.</b>	SEL_ADDRESS_01	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Khách hàng chọn địa chỉ		
<b>Author</b>	Trần Cao Quyền		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal
<b>Actor:</b>			
<ul style="list-style-type: none"> <li>• Khách hàng</li> </ul>			
<b>Summary:</b>			

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

- Khách hàng chọn địa chỉ nhận hàng

### Goal:

- Khách hàng chọn được địa chỉ mong muốn

### Triggers:

- Khách hàng ấn vào địa chỉ

### Preconditions:

- Trong danh sách sản phẩm có sản phẩm
- Đã có sản phẩm trong giỏ hàng
- Đang trong mục giỏ hàng

### Post conditions:

- Success: Địa chỉ hiện trên màn hình giỏ hàng
- Fail: Hệ thống hiển thị Warn message.

### Main Success Scenario:

Step	Actor Action	System Response
1	Khách hàng ấn chọn địa chỉ	Ứng dụng hiện danh sách địa chỉ
2	Khách hàng chọn địa chỉ	Ứng dụng hiện thông báo muốn chọn hay cập nhật địa chỉ
3	Khách hàng ấn chọn	Ứng dụng xác nhận địa chỉ và hiện địa chỉ vừa được chọn lên màn hình Giỏ hàng

### Alternative Scenario:

#### Khách hàng chưa đăng nhập:

No	Actor Action	System Response
1	Khách hàng ấn chọn địa chỉ	Ứng dụng hiện thông báo chưa đăng nhập

### Relationships:

### Business Rules:

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

- Người đăng nhập với tư cách là Khách hàng.

### 2.2.3.8. Use case Thêm địa chỉ

USE CASE – RT_12									
<b>Use Case No.</b>	ADD_ADDRESS_01	<b>Use Case Version</b>	2.0						
<b>Use Case Name</b>	Khách hàng thêm địa chỉ								
<b>Author</b>	Trần Cao Quyền								
<b>Date</b>	10/12/2020	<b>Priority</b>	normal						
<b>Actor:</b> <ul style="list-style-type: none"> <li>• Khách hàng</li> </ul> <b>Summary:</b> <ul style="list-style-type: none"> <li>• Khách hàng thêm địa chỉ và thời gian giao hàng</li> </ul> <b>Goal:</b> <ul style="list-style-type: none"> <li>• Khách hàng thêm được địa chỉ của mình và thời gian giao hàng mong muốn</li> </ul> <b>Triggers:</b> <ul style="list-style-type: none"> <li>• Khách hàng ấn thêm địa chỉ</li> </ul> <b>Preconditions:</b> <ul style="list-style-type: none"> <li>• Trong danh sách sản phẩm có sản phẩm</li> <li>• Trong giỏ hàng đã được thêm sản phẩm</li> <li>• Đang ở trong màn hình giỏ hàng</li> </ul> <b>Post conditions:</b> <ul style="list-style-type: none"> <li>• Success: thêm địa chỉ thành công</li> <li>• Fail: Hệ thống hiển thị Warn message.</li> </ul> <b>Main Success Scenario:</b> <table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Khách hàng ấn vào chọn địa chỉ địa chỉ</td><td>Ứng dụng mở lên danh sách địa chỉ</td></tr> </tbody> </table>				Step	Actor Action	System Response	1	Khách hàng ấn vào chọn địa chỉ địa chỉ	Ứng dụng mở lên danh sách địa chỉ
Step	Actor Action	System Response							
1	Khách hàng ấn vào chọn địa chỉ địa chỉ	Ứng dụng mở lên danh sách địa chỉ							

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

2	Khách hàng chọn thêm địa chỉ mới	Ứng dụng chuyển đến màn hình thêm địa chỉ
3	Khách hàng nhập địa chỉ và khung giờ giao hàng	
4	Khách hàng ấn lưu	Ứng dụng lưu thông tin và chuyển và màn hình gio hàng

**Alternative Scenario:**

**Relationships:**

**Business Rules:**

- Người đăng nhập với tư cách là Khách hàng.

### 2.2.3.9. Use case Đặt hàng

USE CASE – RT_12			
Use Case No.	ORDER_01	Use Case Version	2.0
Use Case Name	Khách hàng đặt hàng		
Author	Trần Cao Quyền		
Date	10/12/2020	Priority	normal

**Actor:**

- Khách hàng

**Summary:**

- Khách hàng đặt hàng

**Goal:**

- Khách hàng đặt được sản phẩm

**Triggers:**

- Khách hàng ấn đặt hàng

**Preconditions:**

- Trong danh sách sản phẩm có sản phẩm
- Đã có sản phẩm trong gio hàng

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

- Đang trong mục giỏ hàng
- Đã được chọn địa chỉ

### Post conditions:

- Success: chuyển màn hình đặt hàng thành công
- Fail: Hệ thống hiển thị Warn message.

### Main Success Scenario:

Step	Actor Action	System Response
1	Khách hàng ấn đặt hàng	Ứng dụng lưu thông tin đơn hàng, và chuyển đến màn hình đặt hàng thành công

### Alternative Scenario:

### Relationships:

### Business Rules:

- Người đăng nhập với tư cách là Khách hàng.

### 2.2.3.10. Use case Xem sản phẩm đã đặt

USE CASE – RT_12			
Use Case No.	SEE_ORDER_01	Use Case Version	2.0
Use Case Name	Khách hàng xem đơn hàng đã đặt		
Author	Trần Cao Quyền		
Date	10/12/2020	Priority	normal
<b>Actor:</b> <ul style="list-style-type: none"> <li>• Khách hàng</li> </ul> <b>Summary:</b> <ul style="list-style-type: none"> <li>• Khách hàng xem đơn hàng đã đặt</li> </ul> <b>Goal:</b> <ul style="list-style-type: none"> <li>• Khách hàng xem được các đơn hàng đã đặt</li> </ul> <b>Triggers:</b>			

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

- Khách hàng chọn mục sản phẩm đã mua

### Preconditions:

- Khách hàng đã đăng nhập

### Post conditions:

- Success: Địa chỉ hiện trên màn hình giờ hàng
- Fail: Hệ thống hiển thị Warn message.

### Main Success Scenario:

Step	Actor Action	System Response
1	Khách hàng ấn mục cá nhân	Ứng dụng chuyển đến màn hình cá nhân
2	Khách hàng chọn mục sản phẩm đã mua	Ứng dụng chuyển đến màn hình sản phẩm đã mua và hiện thông tin các đơn hàng đã đặt

### Alternative Scenario:

#### Khách hàng chưa đặt hàng:

No	Actor Action	System Response
1	Khách hàng chọn mục sản phẩm đã mua	Ứng dụng chuyển đến màn hình sản phẩm đã mua và hiện màn hình trống

### Relationships:

### Business Rules:

- Người đăng nhập với tư cách là Khách hàng.

#### 2.2.3.11. Use case Tìm kiếm

USE CASE – RT_12			
Use Case No.	SEARCH_01	Use Case Version	2.0
Use Case Name	Khách hàng tìm kiếm sản phẩm		
Author	Trần Cao Quyền		

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

Date	10/12/2020	Priority	normal
<b>Actor:</b>			
<ul style="list-style-type: none"> <li>Khách hàng</li> </ul>			
<b>Summary:</b>			
<ul style="list-style-type: none"> <li>Khách hàng tìm kiếm sản phẩm</li> </ul>			
<b>Goal:</b>			
<ul style="list-style-type: none"> <li>Khách hàng tìm kiếm sản phẩm</li> </ul>			
<b>Triggers:</b>			
<ul style="list-style-type: none"> <li>Khách hàng ấn vào ô tìm kiếm</li> </ul>			
<b>Preconditions:</b>			
<ul style="list-style-type: none"> <li>Màn hình đang ở trang chủ</li> </ul>			
<b>Post conditions:</b>			
<ul style="list-style-type: none"> <li>Success: Tìm kiếm thành công</li> <li>Fail: Hệ thống hiển thị Warn message.</li> </ul>			
<b>Main Success Scenario:</b>			
Step	Actor Action	System Response	
1	Khách hàng ấn vào ô tìm kiếm	Ứng dụng mở bàn phím	
2	Khách hàng nhập tên sản phẩm	Ứng dụng hiện các ký tự khách hàng vừa nhập	
3	Khách hàng ấn icon tìm kiếm trên bàn phím	Ứng dụng tìm kiếm theo tên khách hàng nhập vào và hiện các sản phẩm có tên đó	
<b>Alternative Scenario:</b>			
<b>Không có sản phẩm</b>			
No	Actor Action	System Response	
1	Khách hàng ấn icon tìm kiếm trên bàn phím	Ứng dụng tìm kiếm theo tên khách hàng nhập vào và hiện màn hình trống	

**Relationships:**

**Business Rules:**

- Người đăng nhập với tư cách là Khách hàng.

### 2.2.3.12 Use case Đăng nhập appgiaohang

#### USE CASE – RT\_12

<b>Use Case No.</b>	LOGIN_01	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Đăng nhập		
<b>Author</b>	Nguyễn Trọng Luật		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal

**Actor:**

- User

**Summary:**

- User đăng nhập

**Goal:**

- User đăng nhập thành công

**Triggers:**

- User ấn vào nút đăng nhập

**Preconditions:**

- Điền tên đăng nhập
- Điền mật khẩu

**Post conditions:**

- Success: Truy cập vào app tới Screen tiếp theo
- Fail: Thông báo sai mật khẩu

**Main Success Scenario:**

<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
1	User điền thông tin tài khoản	Thông tin tài khoản được điền vào khung nhập một cách chính xác

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

2	User nhấn nút đăng nhập	Đăng nhập thành công, dẫn tới màn hình tiếp theo
---	-------------------------	--

**Alternative Scenario:**

**Sai mật khẩu:**

No	Actor Action	System Response
1	User nhấn nút đăng nhập	Hệ thống hiển thị thông báo sai mật khẩu

**Sai tên tài khoản**

No	Actor Action	System Response
1	User nhấn nút đăng nhập	Hệ thống hiển thị thông báo không tìm thấy tài khoản

**Relationships:** RES\_02.

**Business Rules:**

### 2.2.3.13. Use case Xem danh sách chuỗi đơn hàng

USE CASE – RT_12			
<b>Use Case No.</b>	SEE_ORDER_01	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Shipper xem danh sách chuỗi đơn		
<b>Author</b>	Trần Cao Quyền		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal

**Actor:**

- Shipper

**Summary:**

- Shipper Xem chuỗi đơn hàng

**Goal:**

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

- Shipper Xem được chuỗi giao hàng.

### Triggers:

- Shipper ấn vào chuỗi đơn hàng đó.

### Preconditions:

- Trong danh sách có chuỗi đơn hàng
- shipper đã đăng nhập với tài khoản của mình.

### Post conditions:

- Success: Danh sách chuỗi đơn hiện ra
- Fail: Hệ thống hiển thị Warn message.

### Main Success Scenario:

Step	Actor Action	System Response
1	Shipper chọn mục listOrder	Hệ thống hiển thị danh sách chuỗi đơn hàng gồm số lượng, và danh sách gồm id và thời gian bắt đầu giao
2	Shipper chọn từng chuỗi	Hệ thống hiển thị chi tiết chuỗi đó bao gồm số lượng đơn hàng và chi tiết địa chỉ từng đơn

### Alternative Scenario:

#### Không có chuỗi đơn hàng:

No	Actor Action	System Response
1	Shipper chọn mục listOrder	Hệ thống hiển thị danh sách chuỗi đơn hàng gồm số lượng là 0 và không hiện danh sách gồm id và thời gian bắt đầu giao

**Relationships:** RES\_02, RES\_ORDER\_02, SEE\_INF\_CUS\_03, CALL\_04,  
 SUBMIT\_STT\_CUS\_05, SEE\_STT\_ORDER\_06,  
 SUBMIT\_ORDER\_07, SEE\_STT\_CUS\_08

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### **Business Rules:**

- Người đăng nhập với tư cách là shipper.

#### 2.2.3.14. Use case Đăng ký chuỗi đơn hàng

##### USE CASE – RT\_12

<b>Use Case No.</b>	RES_ORDER_02	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Shipper Đăng ký chuỗi đơn		
<b>Author</b>	Trần Cao Quyền		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal

##### **Actor:**

- Shipper

##### **Summary:**

- Shipper Đăng ký chuỗi đơn hàng mà mình muốn

##### **Goal:**

- Shipper Đăng ký được chuỗi giao hàng.

##### **Triggers:**

- Shipper ấn vào nút chọn của chuỗi đơn hàng đó.

##### **Preconditions:**

- Trong danh sách có chuỗi đơn hàng
- shipper đã đăng nhập với tài khoản của mình.
- Shipper đã chọn vào chuỗi mình muốn

##### **Post conditions:**

- Success: Hiện thông báo đăng ký thành công
- Fail: Hệ thống hiển thị Warn message.

##### **Main Success Scenario:**

<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>
1	Shipper ấn nút chọn của chuỗi đó	Hệ thống put id shipper lên server và thông báo "Đã Đăng Ký Thành Công"

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### **Alternative Scenario:**

#### **Đã đăng ký đơn trước đó:**

No	Actor Action	System Response
1	Shipper ấn nút chọn của chuỗi đó	Hệ thống thông báo “Bạn Đã Đăng Ký Chuỗi Trước Đó”

#### **Đã có shipper đăng ký đơn đó:**

No	Actor Action	System Response
1	Shipper ấn nút chọn của chuỗi đó	Hệ thống thông báo "Đã có Shipper khác đăng ký chuỗi đơn này"

**Relationships:** SEE\_ORDER\_01, RES\_ORDER\_02, SEE\_INF\_CUS\_03,  
CALL\_04, SUBMIT\_STT\_CUS\_05, SEE\_STT\_ORDER\_06,  
SUBMIT\_ORDER\_07, SEE\_STT\_CUS\_08

#### **Business Rules:**

- Người đăng nhập với tư cách là shipper.

### 2.2.3.15. Use case Xem thông tin khách hàng

USE CASE – RT_12			
<b>Use Case No.</b>	SEE_INF_CUS_03	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Shipper xem thông tin khách hàng		
<b>Author</b>	Trần Cao Quyền		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal

#### **Actor:**

- Shipper

#### **Summary:**

- Shipper Xem thông tin khách hàng

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### Goal:

- Thông tin khách hàng hiện ra trên màn hình.

### Triggers:

- Shipper chọn mục MapScreen.

### Preconditions:

- shipper đã đăng nhập với tài khoản của mình.

### Post conditions:

- Success: Thông tin khách hàng hiện ra trên màn hình
- Fail: Hệ thống hiển thị Warn message.

### Main Success Scenario:

Step	Actor Action	System Response
1	Shipper chọn mục MapScreen	Hệ thống hiển thị thông tin của khách hàng bao gồm họ tên, số điện thoại, địa chỉ, số tiền thanh toán

### Alternative Scenario:

#### Chưa đăng ký chuỗi:

No	Actor Action	System Response
1	Shipper chọn mục MapScreen	Hệ thống hiển thị màn hình có hình ảnh chưa có chuỗi nào được đăng ký

**Relationships:** SEE\_ORDER\_01, RES\_ORDER\_02, CALL\_04,  
SUBMIT\_STT\_CUS\_05, SEE\_STT\_ORDER\_06, SUBMIT\_ORDER\_07,  
SEE\_STT\_CUS\_08

### Business Rules:

- Người đăng nhập với tư cách là shipper.

### 2.2.3.16. Use case Liên hệ khách hàng

USE CASE – RT\_12

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

<b>Use Case No.</b>	CALL_04	<b>Use Case Version</b>	2.0						
<b>Use Case Name</b>	Shipper Liên hệ khách hàng								
<b>Author</b>	Trần Cao Quyền								
<b>Date</b>	10/12/2020	<b>Priority</b>	normal						
<p><b>Actor:</b></p> <ul style="list-style-type: none"> <li>• Shipper</li> </ul> <p><b>Summary:</b></p> <ul style="list-style-type: none"> <li>• Shipper Liên hệ với khách hàng trên đơn hàng</li> </ul> <p><b>Goal:</b></p> <ul style="list-style-type: none"> <li>• Ứng dụng chuyển đến ứng dụng điện thoại kèm số điện thoại khách hàng</li> </ul> <p><b>Triggers:</b></p> <ul style="list-style-type: none"> <li>• Shipper ấn vào nút liên hệ.</li> </ul> <p><b>Preconditions:</b></p> <ul style="list-style-type: none"> <li>• Shipper đã đăng ký chuỗi đơn hàng trước đó</li> <li>• Shipper đã đăng nhập với tài khoản của mình.</li> </ul> <p><b>Post conditions:</b></p> <ul style="list-style-type: none"> <li>• Success: Hiện thông báo đăng ký thành công</li> <li>• Fail: Hệ thống hiển thị Warn message.</li> </ul> <p><b>Main Success Scenario:</b></p> <table border="1"> <thead> <tr> <th><b>Step</b></th><th><b>Actor Action</b></th><th><b>System Response</b></th></tr> </thead> <tbody> <tr> <td>1</td><td>Shipper nhấn nút liên hệ</td><td>Hệ thống chuyển sang ứng dụng điện thoại kèm số điện thoại của khách hàng</td></tr> </tbody> </table> <p><b>Alternative Scenario:</b></p> <p><b>Relationships:</b> SEE_ORDER_01, RES_ORDER_02, SEE_INF_CUS_03, SUBMIT_STT_CUS_05, SEE_STT_ORDER_06, SUBMIT_ORDER_07, SEE_STT_CUS_08</p> <p><b>Business Rules:</b></p>				<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>	1	Shipper nhấn nút liên hệ	Hệ thống chuyển sang ứng dụng điện thoại kèm số điện thoại của khách hàng
<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>							
1	Shipper nhấn nút liên hệ	Hệ thống chuyển sang ứng dụng điện thoại kèm số điện thoại của khách hàng							

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

- Người đăng nhập với tư cách là shipper.

### 2.2.3.17. Use case Xác nhận đơn hàng

#### USE CASE – RT\_12

<b>Use Case No.</b>	SUBMIT_STT_CUS_05	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Shipper Xác nhận đơn hàng		
<b>Author</b>	Trần Cao Quyền		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal

#### Actor:

- Shipper

#### Summary:

- Shipper xác nhận giao hàng

#### Goal:

- Shipper xác nhận thành công

#### Triggers:

- Shipper ấn vào nút xác nhận.

#### Preconditions:

- Shipper đã đăng ký chuỗi đơn hàng trước đó
- shipper đã đăng nhập với tài khoản của mình.

#### Post conditions:

- Success: Hiện thông báo đăng ký thành công
- Fail: Hệ thống hiển thị Warn message.

#### Main Success Scenario:

Step	Actor Action	System Response
1	Shipper nhấn nút xác nhận	Hệ thống hiện cửa sổ gồm 2 nút thành công và thất bại
2	Shipper chọn thành công	Hệ thống put tình trạng đơn hàng thành thành công, sau đó chuyển thông tin đến khách hàng tiếp theo

**Alternative Scenario:**

**Giao hàng thất bại:**

No	Actor Action	System Response
1	Shipper chọn thất bại	Hệ thống put tình trạng đơn hàng thành thất bại, sau đó chuyển thông tin đến khách hàng tiếp theo

**Relationships:** RES\_02.

**Business Rules:** SEE\_ORDER\_01, RES\_ORDER\_02, SEE\_INF\_CUS\_03, CALL\_04, SEE\_STT\_ORDER\_06, SUBMIT\_ORDER\_07,,SEE\_STT\_CUS\_08

- Người đăng nhập với tư cách là shipper.

### 2.2.3.18. Use case Xem tình trạng chuỗi đơn

USE CASE – RT_12			
Use Case No.	SEE_STT_ORDER_06	Use Case Version	2.0
Use Case Name	Shipper Xem tình trạng chuỗi đơn		
Author	Trần Cao Quyền		
Date	10/12/2020	Priority	normal

**Actor:**

- Shipper

**Summary:**

- Shipper Xem tình trạng chuỗi đơn

**Goal:**

- Shipper xem được tình trạng chuỗi đơn hàng

**Triggers:**

- Shipper ấn vào mục CheckStatus.

**Preconditions:**

- shipper đã đăng nhập với tài khoản của mình.

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### **Post conditions:**

- Success: Xem được trình trạng
- Fail: Hệ thống hiển thị Warn message.

### **Main Success Scenario:**

Step	Actor Action	System Response
1	Shipper ấn vào mục CheckStatus	Hệ thống hiển thị các mục tiếp nhận chuỗi, đang thực hiện, hoàn tất

### **Alternative Scenario:**

**Relationships:** SEE\_ORDER\_01, RES\_ORDER\_02, SEE\_INF\_CUS\_03, CALL\_04, SUBMIT\_STT\_CUS\_05, SUBMIT\_ORDER\_07,,SEE\_STT\_CUS\_08

### **Business Rules:**

- Người đăng nhập với tư cách là shipper.

### **2.2.3.19. Use case Xác nhận hoàn thành chuỗi đơn hàng**

USE CASE – RT_12			
<b>Use Case No.</b>	SUBMIT_ORDER_07	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Shipper Xác nhận hoàn thành chuỗi đơn hàng		
<b>Author</b>	Trần Cao Quyền		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal

### **Actor:**

- Shipper

### **Summary:**

- Shipper Xác nhận hoàn thành chuỗi đơn hàng

### **Goal:**

- Shipper xác nhận giao hoàn tất

### **Triggers:**

- Shipper ấn vào nút xác nhận hoàn tất .

### **Preconditions:**

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

- Shipper đã đăng ký chuỗi đơn hàng trước đó
- shipper đã đăng nhập với tài khoản của mình.
- Đang ở màn hình CheckStatus

### **Post conditions:**

- Success: Hiện thông báo Xác Nhận Thành Công
- Fail: Hệ thống hiển thị Warn message.

### **Main Success Scenario:**

Step	Actor Action	System Response
1	Shipper nhấn nút xác nhận giao hoàn tất	Hệ thống put lên server và hiện thông báo Xác Nhận Thành Công, sau đó chuyển các dòng về đầu x

### **Alternative Scenario:**

### **Chưa hoàn thành xong:**

No	Actor Action	System Response
1	Shipper nhấn nút xác nhận giao hoàn tất	Hệ thống hiện thông báo Đảm bảo rằng bạn đã giao hết đơn

**Relationships:** SEE\_ORDER\_01, RES\_ORDER\_02, SEE\_INF\_CUS\_03, CALL\_04, SUBMIT\_STT\_CUS\_05, SEE\_STT\_ORDER\_06, SEE\_STT\_CUS\_08

### **Business Rules:**

- Người đăng nhập với tư cách là shipper.

### **2.2.3.20. Use case Xem tình trạng và thông tin đơn hàng từng khách hàng**

USE CASE – RT_12			
Use Case No.	SEE_STT_CUS_08	Use Case Version	2.0
Use Case Name	Shipper xem tình trạng và thông tin từng khách hàng		
Author	Trần Cao Quyền		

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

Date	10/12/2020	Priority	normal		
<b>Actor:</b>					
<ul style="list-style-type: none"> <li>Shipper</li> </ul>					
<b>Summary:</b>					
<ul style="list-style-type: none"> <li>Shipper xem tình trạng và thông tin từng khách hàng</li> </ul>					
<b>Goal:</b>					
<ul style="list-style-type: none"> <li>Shipper xem tình trạng và thông tin</li> </ul>					
<b>Triggers:</b>					
<ul style="list-style-type: none"> <li>Shipper nhấp vào mục Logistics.</li> </ul>					
<b>Preconditions:</b>					
<ul style="list-style-type: none"> <li>Shipper đã đăng nhập với tài khoản của mình.</li> </ul>					
<b>Post conditions:</b>					
<ul style="list-style-type: none"> <li>Success: vào được màn hình</li> <li>Fail: Hệ thống hiển thị Warn message.</li> </ul>					
<b>Main Success Scenario:</b>					
<b>Step</b>	<b>Actor Action</b>	<b>System Response</b>			
1	Shipper nhấp vào mục Logistics	Hệ thống hiển thị thông tin bao gồm số lượng đơn và danh sách từng đơn bao gồm tên, số tiền địa chỉ và tình trạng			
<b>Alternative Scenario:</b>					
<b>Chưa đăng ký chuỗi:</b>					
<b>No</b>	<b>Actor Action</b>	<b>System Response</b>			
1	Shipper nhấp vào mục Logistics	Hệ thống hiện ra thông tin số lượng: 0			
<b>Relationships:</b> SEE_ORDER_01, RES_ORDER_02, SEE_INF_CUS_03, CALL_04, SUBMIT_STT_CUS_05, SEE_STT_ORDER_06, SUBMIT_ORDER_07,					

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### Business Rules:

- Người đăng nhập với tư cách là shipper.

#### 2.2.3.21 Use case Đăng xuất

USE CASE – RT_12			
<b>Use Case No.</b>	LOGOUT_01	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Đăng xuất		
<b>Author</b>	Nguyễn Trọng Luật		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal

#### Actor:

- Shipper

#### Summary:

- Shipper đăng xuất

#### Goal:

- Shipper đăng xuất thành công

#### Triggers:

- Shipper ấn vào đăng xuất

#### Preconditions:

- Shipper đã đăng nhập

#### Post conditions:

- Success: Đăng xuất thành công
- Fail: Đăng xuất thất bại

#### Main Success Scenario:

Step	Actor Action	System Response

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

1	Shipper nhấn nút đăng xuất	Hệ thống thông báo đăng xuất thành công
---	----------------------------	---

**Alternative Scenario:**

**Exceptions:**

No	Actor Action	System Response
1	Shipper nhấn nút đăng xuất	Hệ thống thông báo đăng xuất thất bại

**Relationships:** RES\_02.

**Business Rules:**

- Người đăng nhập với tư cách là shipper.

### 2.2.3.22 Use case Xem thông tin cá nhân

USE CASE – RT_12			
<b>Use Case No.</b>	PROFILE_01	<b>Use Case Version</b>	2.0
<b>Use Case Name</b>	Thông tin cá nhân		
<b>Author</b>	Nguyễn Trọng Luật		
<b>Date</b>	10/12/2020	<b>Priority</b>	normal

**Actor:**

- Shipper

**Summary:**

- Shipper xem thông tin cá nhân

**Goal:**

- Hệ thống hiển thị thông tin cá nhân của shipper

**Triggers:**

- Shipper nhấn vào profile

**Preconditions:**

- Shipper đã đăng nhập

## CHƯƠNG 2 – PHÂN TÍCH VÀ MÔ HÌNH HÓA YÊU CẦU

### **Post conditions:**

- Success: Hệ thống hiển thị chính xác thông tin cá nhân của shipper
- Fail: Hệ thống không hiển thị hoặc hiển thị không chính xác thông tin cá nhân của shipper

### **Main Success Scenario:**

Step	Actor Action	System Response
1	Shipper nhấp vào profile	Hệ thống hiển thị chính xác thông tin cá nhân của shipper

### **Alternative Scenario:**

#### **Hiển thị sai thông tin:**

No	Actor Action	System Response
1	Shipper nhấp vào profile	Hệ thống hiển thị không chính xác thông tin cá nhân của shipper

#### **Không hiển thị:**

No	Actor Action	System Response
1	Shipper nhấp vào profile	Hệ thống không hiển thị thông tin cá nhân của shipper

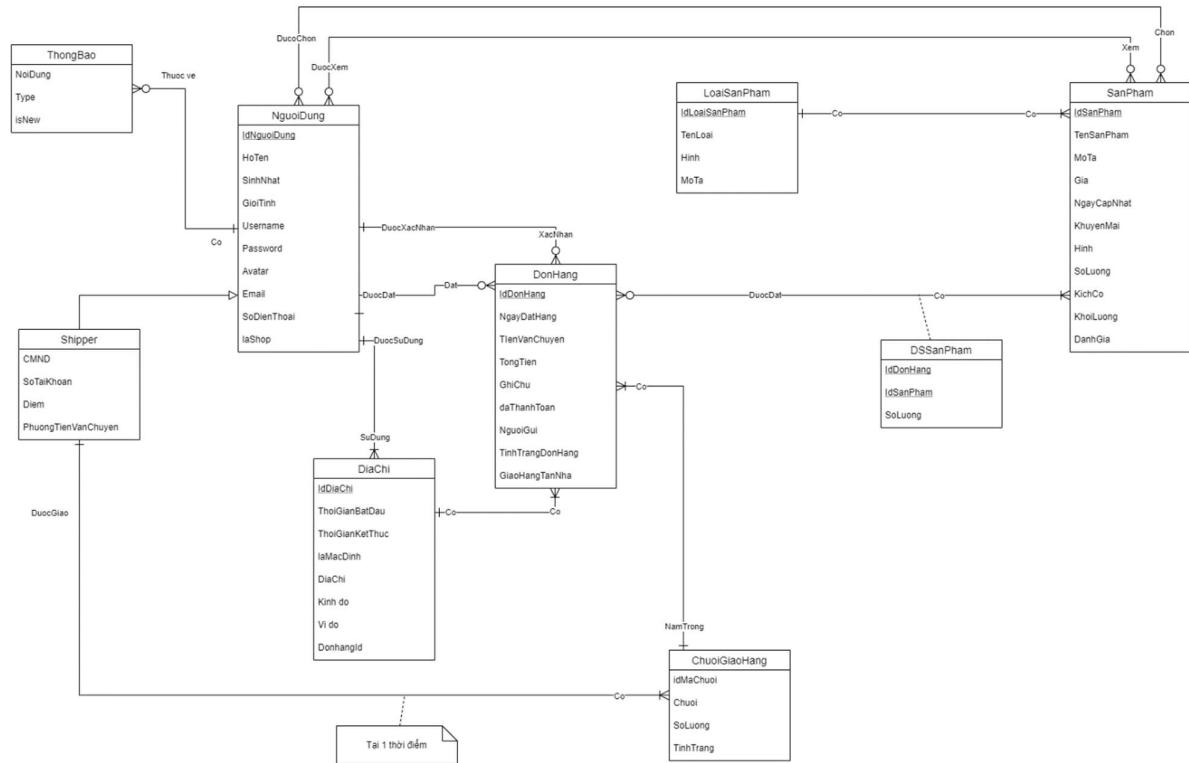
### **Business Rules:**

- Người đăng nhập với tư cách là shipper.

## CHƯƠNG 3: THIẾT KẾ ỨNG DỤNG

### 3.1. THIẾT KẾ DỮ LIỆU

#### 3.2.1. Lược đồ quan hệ



Hình 13: Lược đồ quan hệ của database schema

#### 3.2.2. Mô tả cơ sở dữ liệu

##### 3.2.2.1. NguoiDung

Bảng 5: Bảng thuộc tính NguoiDung

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	<b>id</b>	Int	<b>Khóa chính</b>	<b>id</b> do PostgreSQL tự sinh	
2	<b>HoTen</b>	String		<b>Họ tên người dùng</b>	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

3	SinhNhat	Datetime		Ngày sinh người dùng	
4	GioiTinh	String		Giới tính người dùng	
5	Username	String		Tên tài khoản người dùng	
6	Password	String		Mật khẩu người dùng	
7	Avatar	String		Hình đại diện người dùng	
8	Email	String		Email người dùng	
9	SDT	String		SDT người dùng	
10	isDisable	Boolean		Kiểm tra xem tài khoản còn active hay không	

#### 3.2.2.2. LoaiSanPham

Bảng 6: Bảng thuộc tính LoaiSanPham

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	id	Int	Khóa chính	id do PostgresSQL tự sinh	
2	TenLoai	String		Tên loại sản	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

				<b>phẩm</b>	
3	Hinh	String		Hình tượng trung cho loại sản phẩm	
4	MoTa	String		Mô tả cho từng loại sản phẩm	

#### 3.2.2.3. SanPham

*Bảng 7 : Bảng thuộc tính SanPham*

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	id	Int	Khóa chính	id do PostgresSQL tự sinh	
2	TenSanPham	String		Tên của sản phẩm	
3	MoTa	String		Mô tả về sản phẩm	
4	Gia	Decimal		Giá của sản phẩm	
5	NgayCapNhat	Datetime		Ngày đăng sản phẩm	
6	KhuyenMai	Decimal		Tỉ lệ giảm giá	
7	Hinh	String		Hình ảnh sản phẩm	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

8	SoLuong	Int		Số lượng sản phẩm	
9	KichCo	Decimal		Size của sản phẩm	
10	KhoiLuong	Decimal		Khối lượng sản phẩm	
11	DanhGia	String		Đánh giá dành cho sản phẩm	

#### 3.2.2.4. Shipper

Bảng 8 : Bảng thuộc tính Shipper

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	id	Int	Khóa chính	id do PostgresSQ L tự sinh	
2	CMND	String		CMND của shipper	
3	STK	String		Số tài khoản của shipper	
4	Diem	Int		Điểm đánh giá dành cho shipper	
5	PhuongTien VanChuyen	String		Xác định dung tích xe	

### 3.2.2.5. Diachi

*Bảng 9: Bảng thuộc tính DiaChi*

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	<b>id</b>	Int	Khóa chính	<b>id do PostgresSQ L tự sinh</b>	
2	<b>TenDiaChi</b>	String		<b>Tên địa chỉ</b>	
3	<b>KinhDo</b>	String		<b>Kinh độ của địa chỉ</b>	
4	<b>ViDo</b>	String		<b>Vĩ độ của địa chỉ</b>	
5	<b>ThoiGianBatDau</b>	Datetime		<b>Thời gian bắt đầu trong khung giờ rảnh của người nhận</b>	
6	<b>ThoiGianKetThuc</b>	Datetime		<b>Thời gian kết thúc trong khung giờ rảnh của người nhận</b>	
7	<b>DonhangId</b>	Int		<b>Lưu trữ mã đơn hàng mà địa chỉ được sử dụng</b>	<b>Dùng để liệt kê các địa chỉ của đơn hàng mà khách hàng</b>

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

					đăng ký
8	laMacDinh	Boolean		Biến kiểm tra	

#### 3.2.2.5. DonHang

Bảng 10: Bảng thuộc tính DonHang

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	id	Int	Khóa chính	id do PostgresSQ L tự sinh	
2	NgayDatHàng	Datetime		Ngày đặt đơn	Tìm kiếm đơn hàng theo ngày
3	TongTien	Decimal		Tiền khách hàng phải trả khi nhận hàng	
4	GhiChu	String		Ghi chú cho đơn hàng	
5	DanhGia	String		Đánh giá đơn hàng	
6	TinhTrangDon	String		Trạng thái của đơn hàng	
7	daThanhToan	Boolean		Kiểm tra khách hàng đã thanh	

## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

				toán hay chưa	
--	--	--	--	------------------	--

### 3.2.2.6. ChuoiGiaoHang

Bảng 11: Bảng thuộc tính ChuoiGiaoHang

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	id	Int	Khóa chính	id do PostgreSQL tự sinh	
2	Chuoi	String		Lưu trữ chuỗi	Lưu trữ lộ trình giao hàng dưới dạng stringify của object JSON
3	ThoiGianBatDau	String		Thời gian bắt đầu giao hàng	Được hiển thị cho shipper lựa chọn
4	SoLuong	Int		Số lượng đơn trong chuỗi	
5	isShipped	Boolean		Kiểm tra chuỗi đã giao	

### 3.2.2.7. ThongBao

Bảng 12: Bảng thuộc tính ThongBao

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
-----	------------	------	--------------	---------	---------

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

1	<b>id</b>	Int	Khóa chính	<b>id do PostgresSQL tự sinh</b>	
2	<b>NoiDung</b>	String		<b>Nội dung thông báo</b>	
3	<b>Type</b>	String		<b>Loại thông báo</b>	
4	<b>isNew</b>	Boolean		<b>Kiểm tra thông báo mới</b>	

#### 3.2.2.8. DSSanPham

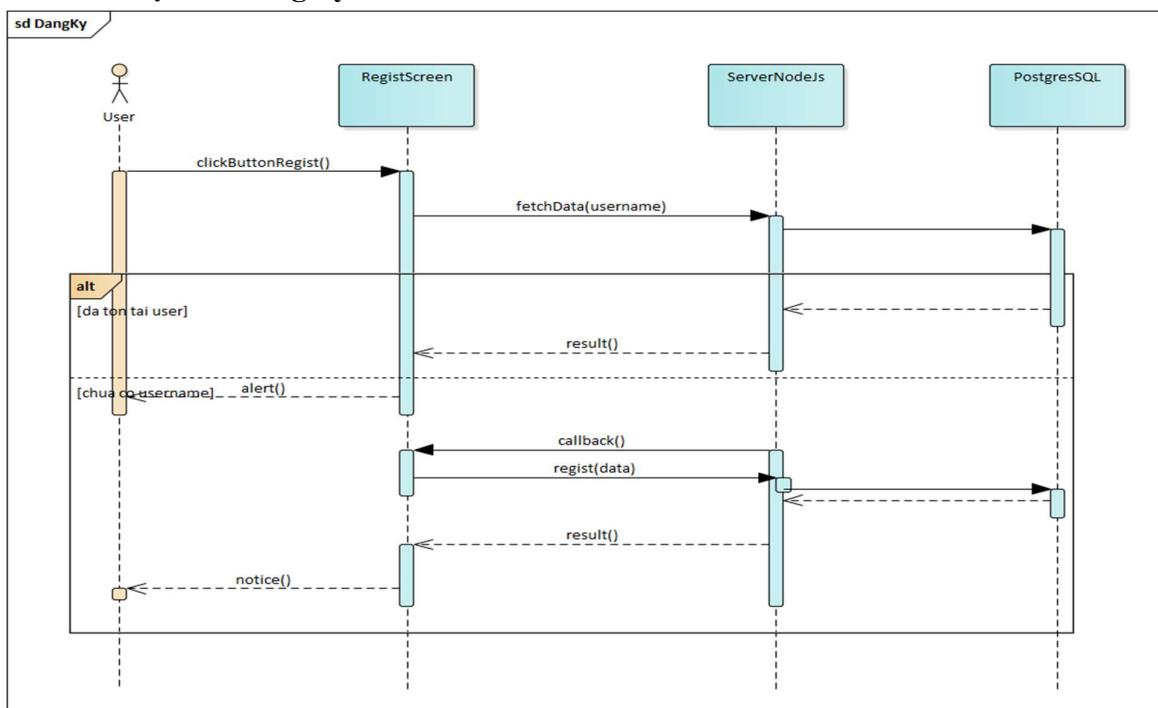
*Bảng 13: Bảng thuộc tính DSSanPham*

STT	Thuộc tính	Kiểu	Miền giá trị	Ý nghĩa	Ghi chú
1	<b>id</b>	Int	Khóa chính	<b>id do PostgreSQL tự sinh</b>	
2	<b>SoLuong</b>	Int		<b>Số lượng của từng sản phẩm trong đơn hàng mà khách hàng đã mua</b>	

### 3.3. LUẬC ĐỒ TUẦN TỰ

#### 3.3.1. Ứng dụng khách hàng:

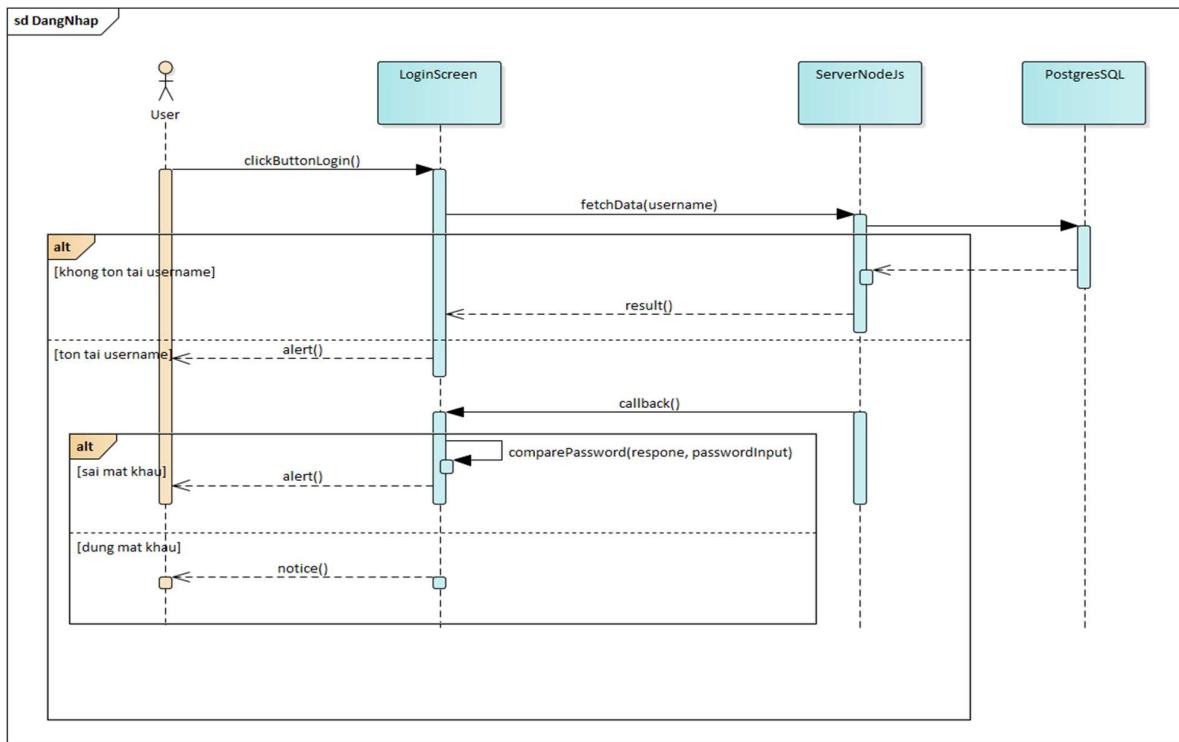
##### 3.3.1.1. Lược đồ Đăng ký



Hình 14: Lược đồ tuần tự Đăng kí

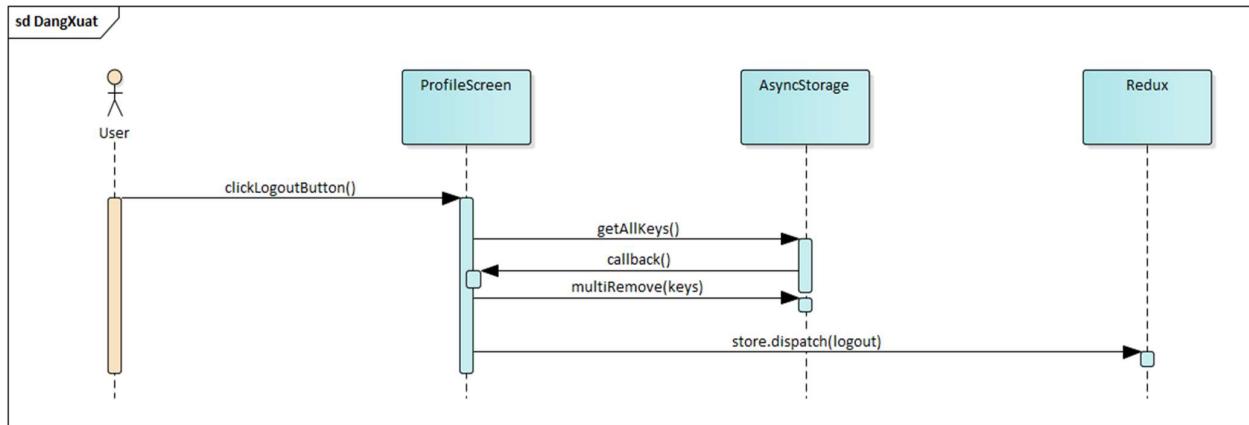
## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

### 3.3.1.2. Lược đồ Đăng nhập



*Hình 15: Lược đồ tuần tự Đăng nhập*

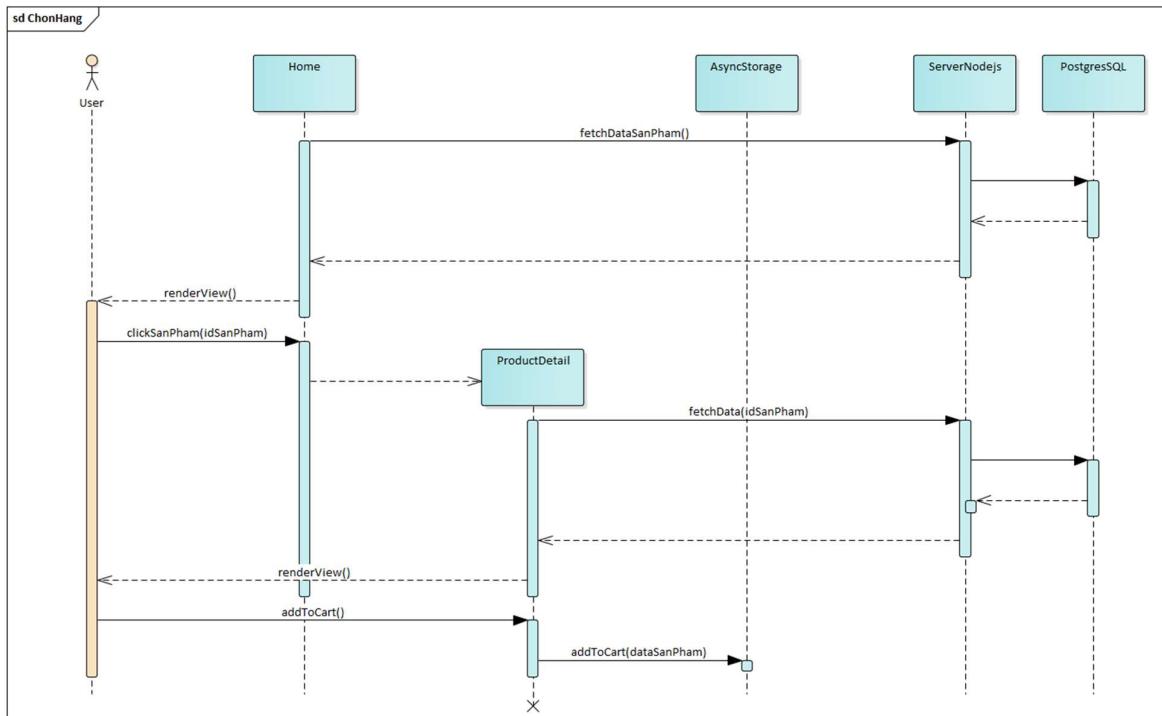
### 3.3.1.3. Lược đồ Đăng xuất



*Hình 16: Lược đồ tuần tự Đăng xuất*

## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

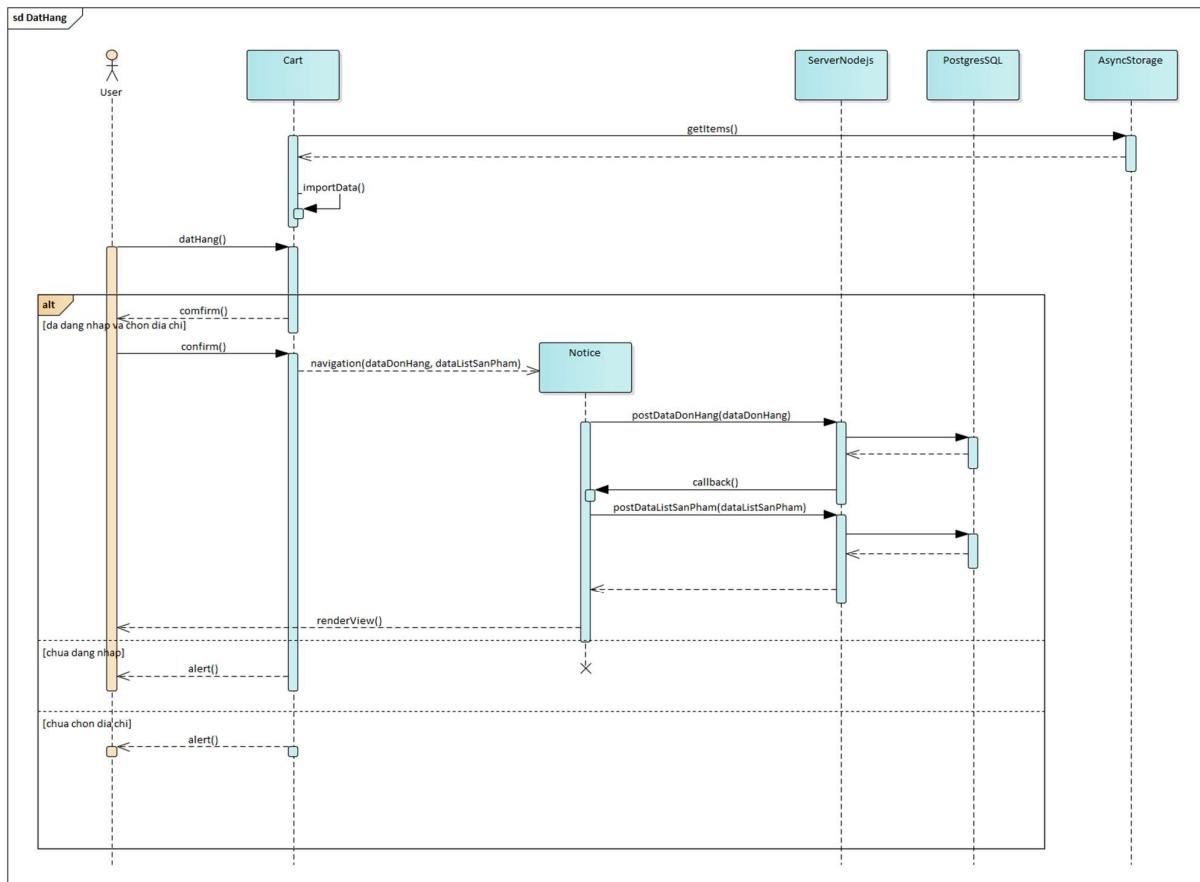
### 3.3.1.4. Lược đồ Chọn sản phẩm



Hình 17: Lược đồ tuần tự Chọn sản phẩm

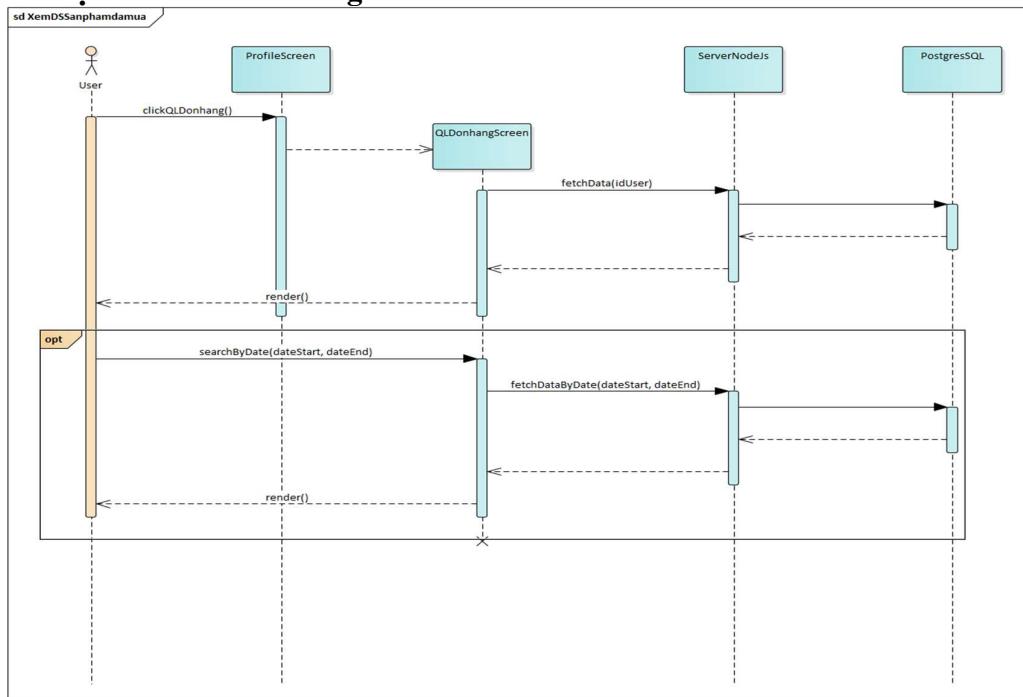
## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

### 3.3.1.5. Lược đồ Đặt hàng



**Hình 18: Lược đồ tuần tự Cập nhật thông tin tài khoản doanh nghiệp**

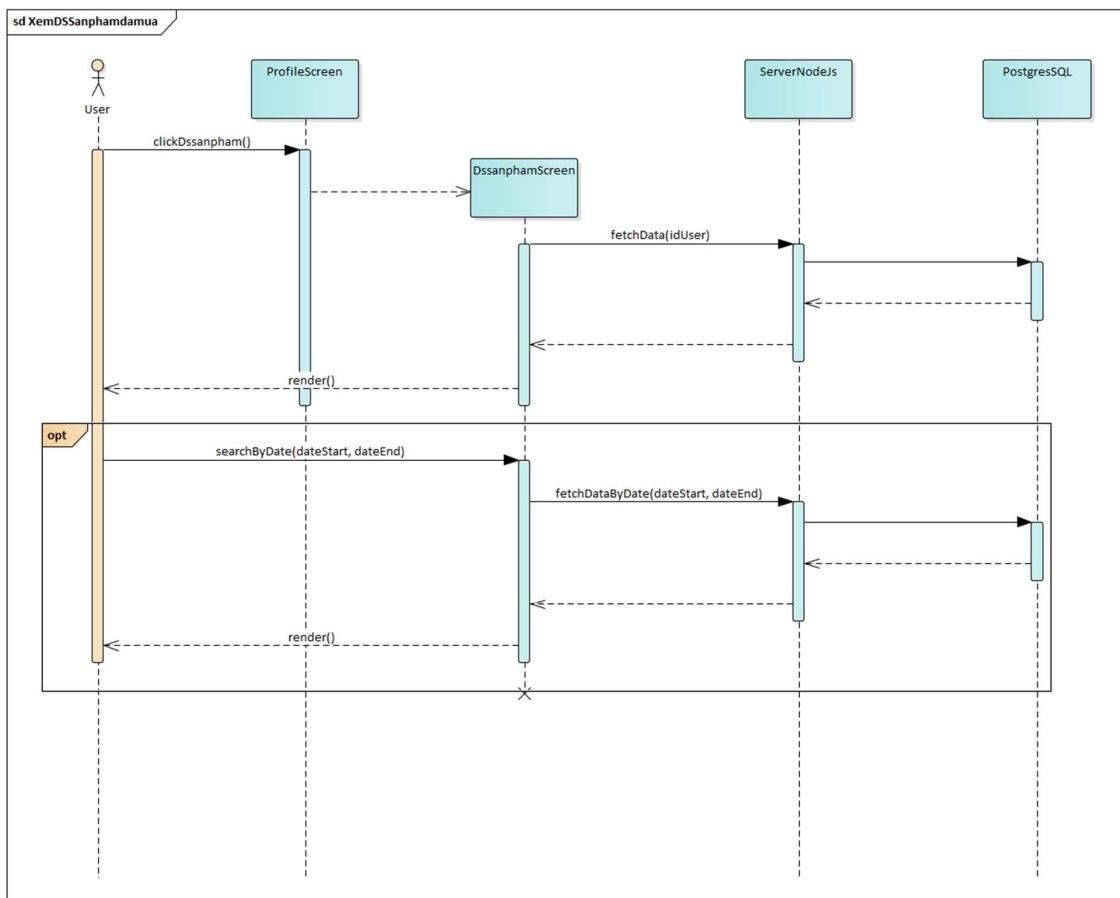
### 3.3.1.6. Lược đồ Xem đơn hàng



**Hình 19: Lược đồ tuần tự Xem đơn hàng**

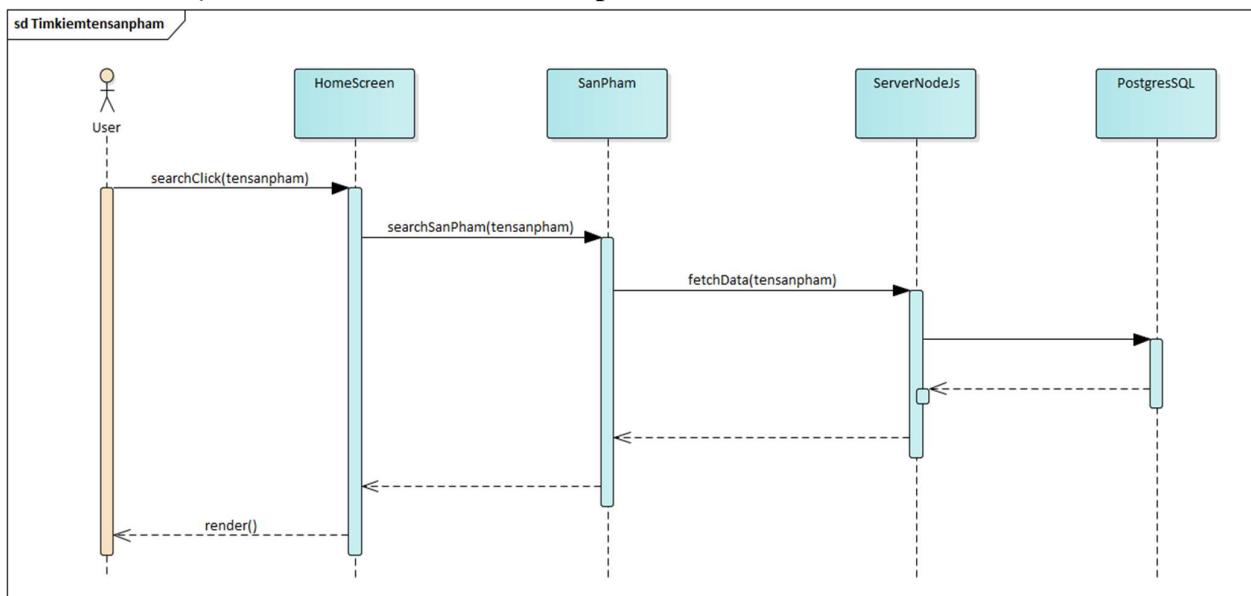
## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

### 3.3.1.7. Lược đồ Xem các sản phẩm đã mua



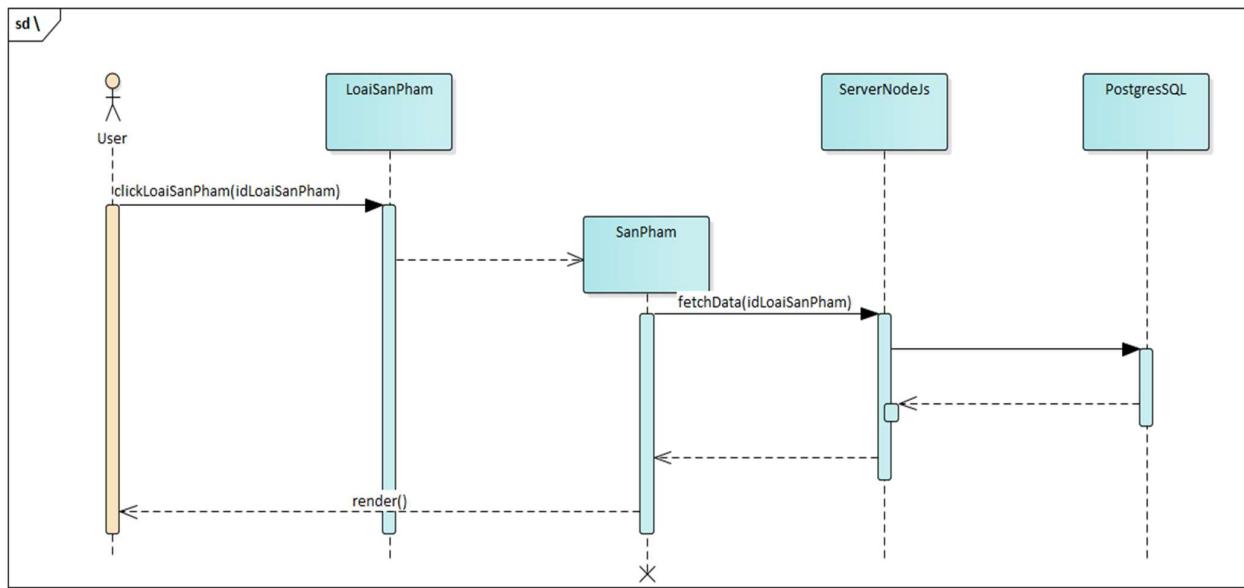
Hình 20: Lược đồ tuần tự Xem các sản phẩm đã mua

### 3.3.1.8. Lược đồ Tìm kiếm theo tên sản phẩm



Hình 21: Lược đồ tuần tự Tìm theo tên sản phẩm

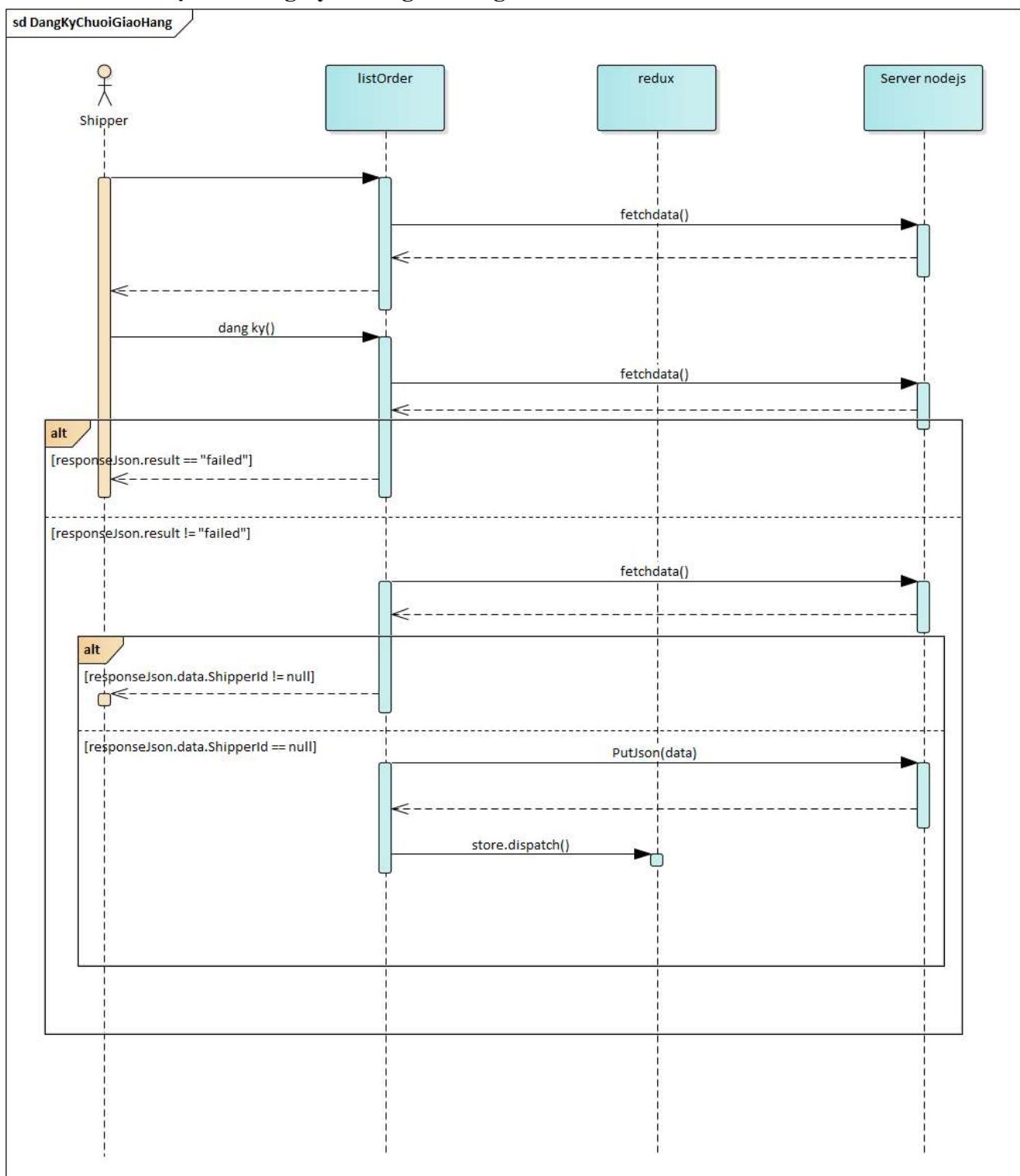
### 3.3.1.9. Lược đồ Tìm kiếm theo loại sản phẩm



Hình 22: Lược đồ tuần tự Tìm theo loại sản phẩm

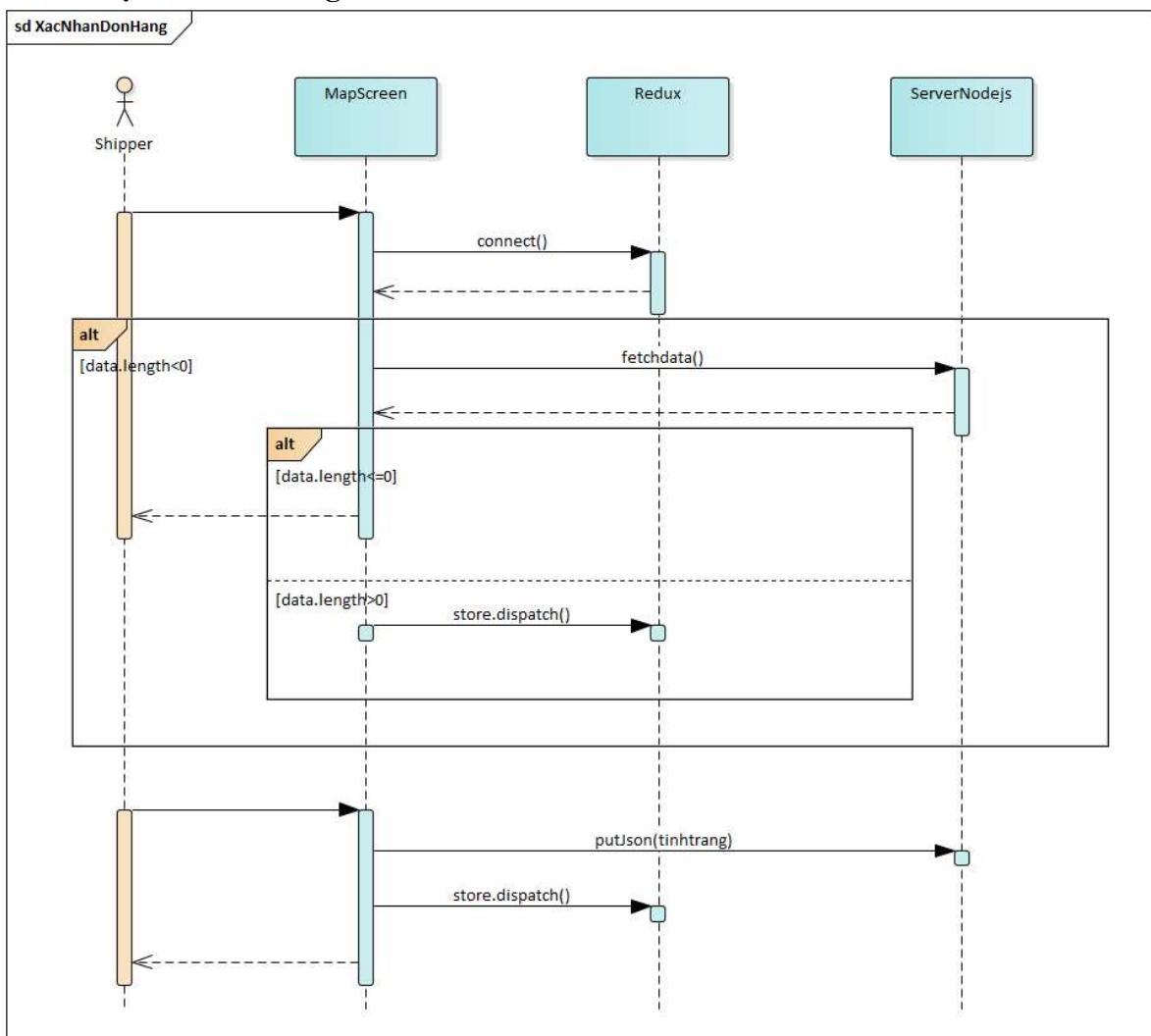
### 3.3.2. Ứng dụng shipper

#### 3.3.2.1. Lược đồ Đăng ký chuỗi giao hàng



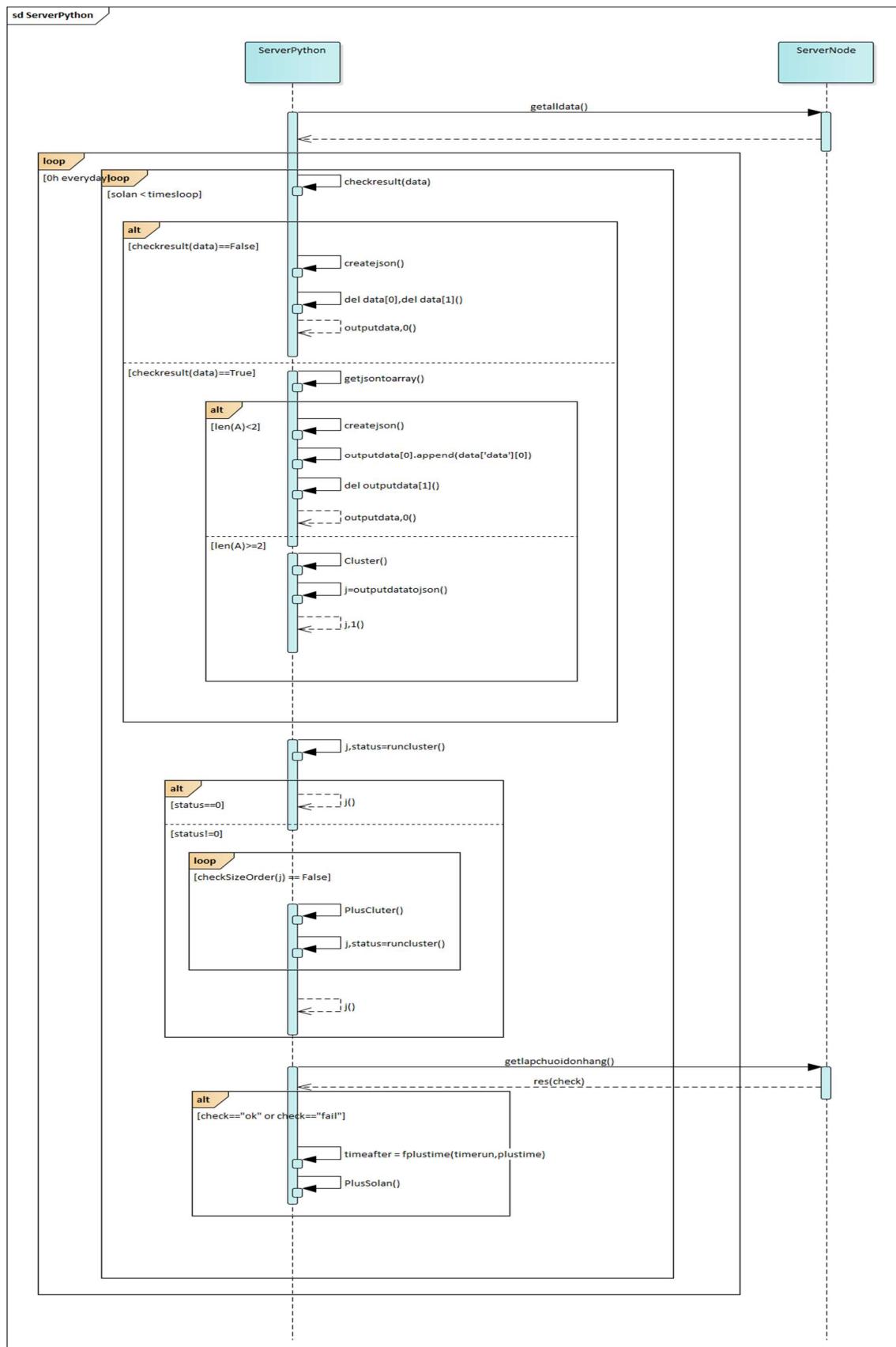
Hình 23: Lược đồ tuần tự Đăng ký chuỗi giao hàng

### 3.3.2.2. Lược đồ Giao hàng



Hình 24: Lược đồ tuần tự Xác nhận đơn hàng

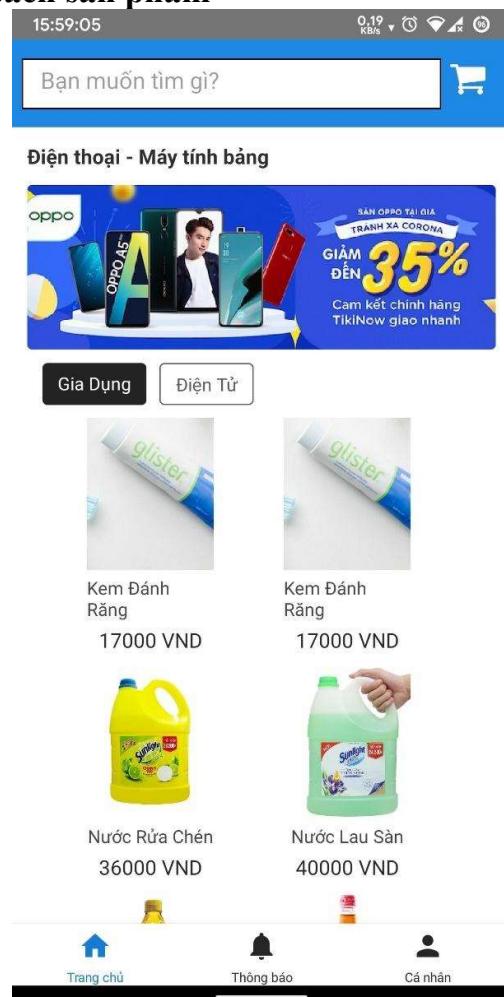
### 3.3.3. Lược đồ giữa hai Server:



Hình 25: Lược đồ tuần tự giữa hai server

### 3.4. GIAO DIỆN

#### 3.4.1. Màn hình Danh sách sản phẩm



*Hình 26: Màn hình Danh sách sản phẩm*

Các đối tượng trong màn hình:

*Bảng 14: Bảng thuộc tính Giao diện*

STT	Loại	Ý nghĩa	Mô tả
1	LoaiSanPham	Hiển thị các loại sản phẩm	
2	SanPham	Hiển thị các sản phẩm	
3	TextInput	Nhập tên sản phẩm tìm kiếm	
4	Navigation Bottom -	Chuyển tới màn hình	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

	Thông báo	thông báo	
5	Navigation Bottom - Cá nhân	Chuyển tới màn hình cá nhân	
6	Icon	Chuyển tới giỏ hàng	

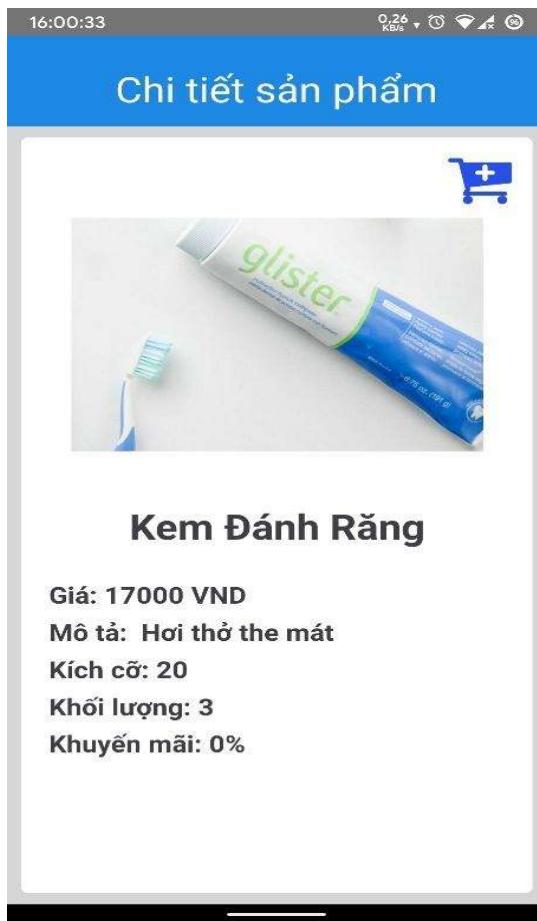
Sơ đồ biến cố:



Hình 27: Sơ đồ biến cố màn hình Đăng ký tài khoản ứng viên

## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

### 3.4.2. Màn hình Sản Phẩm



Hình 28: Màn hình Sản phẩm

Các đối tượng trong màn hình:

Bảng 15: Đối tượng trong màn hình sản phẩm

STT	Loại	Ý nghĩa	Mô tả
1	Icon	Thêm vào giỏ hàng	

Sơ đồ biến cố:



Hình 29: Sơ đồ biến cố màn hình sản phẩm

### 3.4.3. Màn hình Đăng nhập khách hàng



*Hình 30: Màn hình Đăng nhập khách hàng*

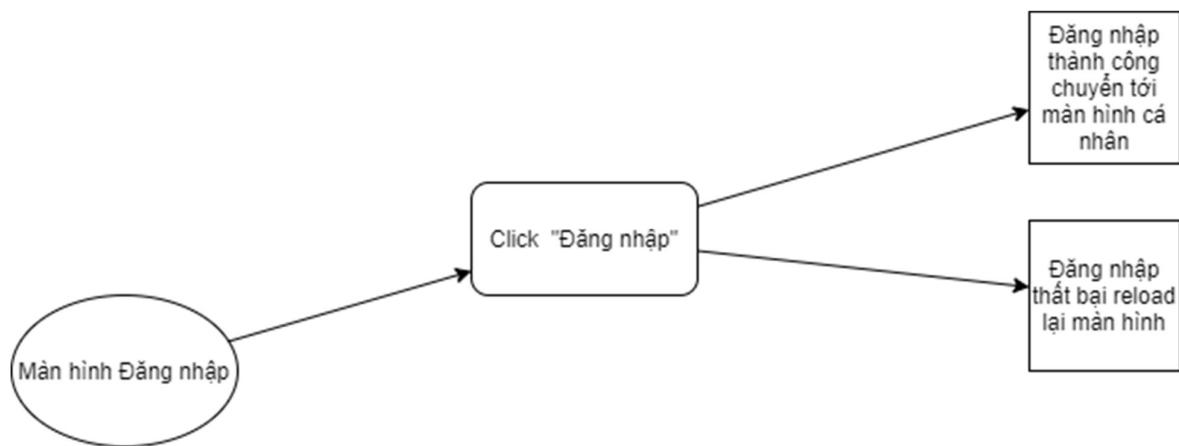
Các đối tượng trong màn hình

*Bảng 16: Đối tượng trong màn hình đăng nhập khách hàng*

STT	Loại	Ý nghĩa	Mô tả
1	TextInput	Nhập tên tài khoản	
2	TextInput	Nhập mật khẩu	
3	TouchableOpacity	Nhấn đăng nhập	1. Đăng nhập thành công chuyển tới màn hình cá nhân 2. Đăng nhập thất bại reload lại màn hình

Sơ đồ biến cố:

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG



Hình 31: Sơ đồ biến cố màn hình Đăng nhập

#### 3.4.4. Màn hình Giỏ Hàng



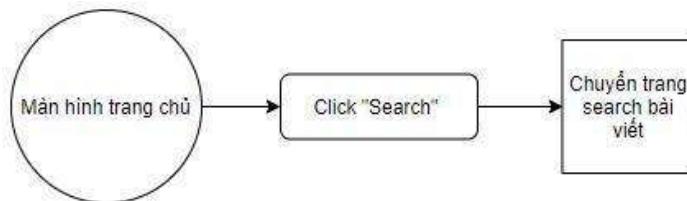
Hình 32: Màn hình Giỏ hàng

Các đối tượng trong màn hình:

**Bảng 17: Đối tượng trong màn hình Trang chủ**

STT	Loại	Ý nghĩa	Mô tả
1	TouchableOpacity “x”	Xóa sản phẩm khỏi giỏ hàng	
2	TouchableOpacity “-”	Giảm số lượng sản phẩm	
3	TouchableOpacity “+”	Tăng số lượng sản phẩm	
4	TouchableOpacity “show detail”	Chuyển tới màn hình thông tin sản phẩm	
5	TouchableOpacity “Chọn địa chỉ”	Hiển thị danh sách địa chỉ	1. Nếu chưa đăng nhập hiển thông báo chưa đăng nhập
6	TouchableOpacity “Đặt hàng”	Chuyển tới màn hình đặt hàng	1. Nếu chưa đăng nhập hiển thông báo chưa đăng nhập 2. Nếu chưa chọn địa chỉ hiển thông báo chưa chọn địa chỉ

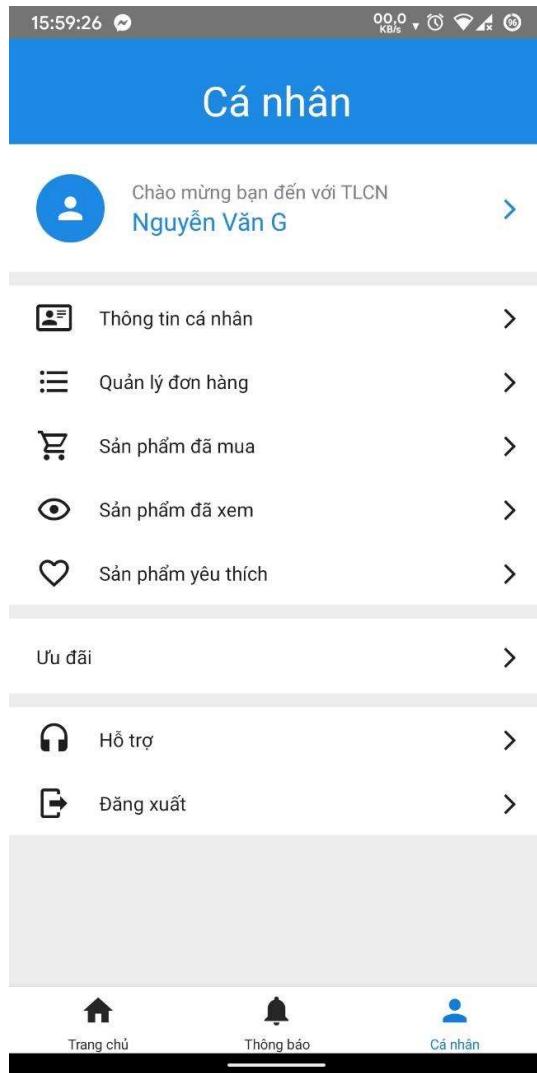
**Sơ đồ biến cố:**



**Hình 33: Sơ đồ biến cố màn hình Trang chủ**

## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

### 3.4.5. Màn hình Cá nhân



*Hình 34: Màn hình Cá nhân*

Các đối tượng trong màn hình:

*Bảng 18: Đối tượng trong màn hình cá nhân*

STT	Loại	Ý nghĩa	Mô tả
1	ProfileItem “Đăng nhập”	Đăng nhập	1. Nếu đã đăng nhập sẽ hiện tên người dùng 2. Nếu chưa đăng nhập sẽ chuyển tới màn hình đăng nhập

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

2	ProfileItem “Thông tin cá nhân”	Chuyển tới màn hình thông tin cá nhân	
3	ProfileItem “Quản lý đơn hàng”	Chuyển tới màn hình quản lý đơn hàng	
4	ProfileItem “Sản phẩm đã mua”	Chuyển tới màn hình sản phẩm đã mua	
5	ProfileItem “Đăng xuất”	Đăng xuất	

#### 3.4.6. Màn hình Chọn địa chỉ



Hình 35: Màn hình Chọn địa chỉ

**Các đối tượng trong màn hình***Bảng 19: Đối tượng trong màn hình Chọn địa chỉ*

STT	Loại	Ý nghĩa	Mô tả
1	TouchableOpacity “Thêm địa chỉ mới”	Chuyển tới màn hình thêm địa chỉ	
2	DiaChi	Chọn địa chỉ	1. Nếu chưa chọn địa chỉ thì thêm địa chỉ vào bộ lưu trữ 2. Nếu đã chọn địa chỉ trùng khớp sẽ hiện thông báo cảnh báo người dùng 3. Nếu bộ lưu trữ đã có 3 địa chỉ sẽ hiện thông báo cảnh báo người dùng

**3.4.7. Màn hình Thêm địa chỉ***Hình 36: Màn hình Thêm địa chỉ*

**Các đối tượng trong màn hình***Bảng 20: Đối tượng trong màn hình Review về nhà tuyển dụng*

STT	Loại	Ý nghĩa	Mô tả
1	TextInput	Nhập số nhà	
2	TextInput	Nhập đường	
3	TextInput	Nhập phường	
4	TextInput	Nhập quận	
5	TouchableOpacity “Ngày”	Chọn ngày	
6	TouchableOpacity “Bắt đầu”	Chọn khung giờ bắt đầu	
7	TouchableOpacity “Kết thúc”	Chọn khung giờ kết thúc	
8	TouchableOpacity “Lưu”	Nhấn lưu	1. Nếu khung giờ không đủ 3 tiếng sẽ hiện cảnh báo cho người dùng

### 3.4.8. Màn hình Sản phẩm đã mua



*Hình 37: Màn hình Sản phẩm đã mua*

Các đối tượng trong màn hình

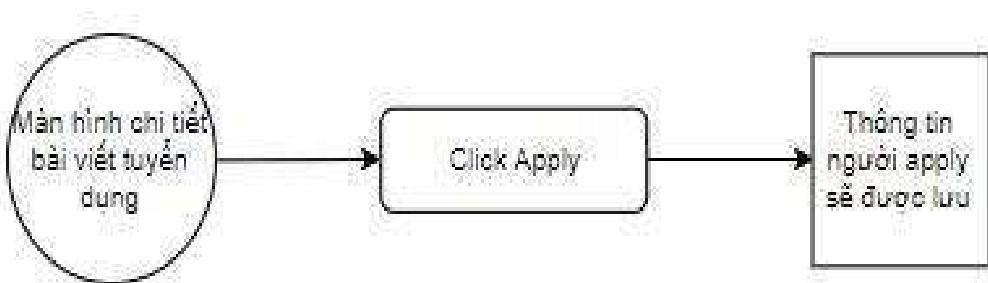
*Bảng 21: Đối tượng trong màn hình Chi tiết bài viết tuyển dụng*

STT	Loại	Ý nghĩa	Mô tả
1	Pagination	Trang sản phẩm	
2	TouchableOpacity “Tìm theo ngày”	Hiển thị khung tìm theo ngày	
3	TouchableOpacity “Bắt đầu”	Chọn ngày bắt đầu	
4	TouchableOpacity “Kết thúc”	Chọn ngày kết thúc	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

5	TouchableOpacity “Tim”	Nhấn Tìm	
---	------------------------	----------	--

Sơ đồ biến cố:



Hình 38: Sơ đồ biến cố màn hình Chi tiết bài viết tuyển dụng

#### 3.4.9. Màn hình Quản lý đơn hàng



Hình 39: Màn hình Quản lý đơn hàng

Các đối tượng trong màn hình

Bảng 22: Đối tượng trong màn hình Thống kê bài viết tuyển dụng

STT	Loại	Ý nghĩa	Mô tả
1	Pagination	Trang sản phẩm	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

2	TouchableOpacity “Tìm theo ngày”	Hiển thị khung tìm theo ngày	
3	TouchableOpacity “Bắt đầu”	Chọn ngày bắt đầu	
4	TouchableOpacity “Kết thúc”	Chọn ngày kết thúc	
5	TouchableOpacity “Tìm”	Nhấn Tìm	

#### 3.4.10. Màn hình Thông tin khách hàng



*Hình 40: Màn hình Thông tin khách hàng*

Các đối tượng trong màn hình:

## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

### 3.4.11. Màn hình Đăng Ký



*Hình 41: Màn hình Đăng ký*

Các đối tượng trong màn hình:

*Bảng 23: Đối tượng trong màn hình Đăng ký*

STT	Loại	Ý nghĩa	Mô tả
1	TextInput	Nhập họ tên	
2	TextInput	Nhập giới tính	
3	TextInput	Nhập username	
4	TextInput	Nhập password	
5	TextInput	Nhập confirm password	
6	TextInput	Nhập SDT	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

7	TextInput	Nhập Email	
8	TouchableOpacity “Sinh nhật”	Chọn ngày sinh	
9	TouchableOpacity “Avatar”	Chọn ảnh đại diện	
10	TouchableOpacity “Đăng ký”	Nhấn đăng ký	<ol style="list-style-type: none"><li>Thiếu thông tin sẽ thông báo cho người dùng</li><li>Tên tài khoản đã tồn tại sẽ hiển thị thông báo cho người dùng</li></ol>

#### 3.4.12. Màn hình Khởi đầu



**Hình 42: Màn hình Khởi đầu**

## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

Các đối tượng trong màn hình:

### 3.4.13. Màn hình Map chưa đăng ký chuỗi đơn



Hình 43: Màn hình Map chưa đăng ký chuỗi

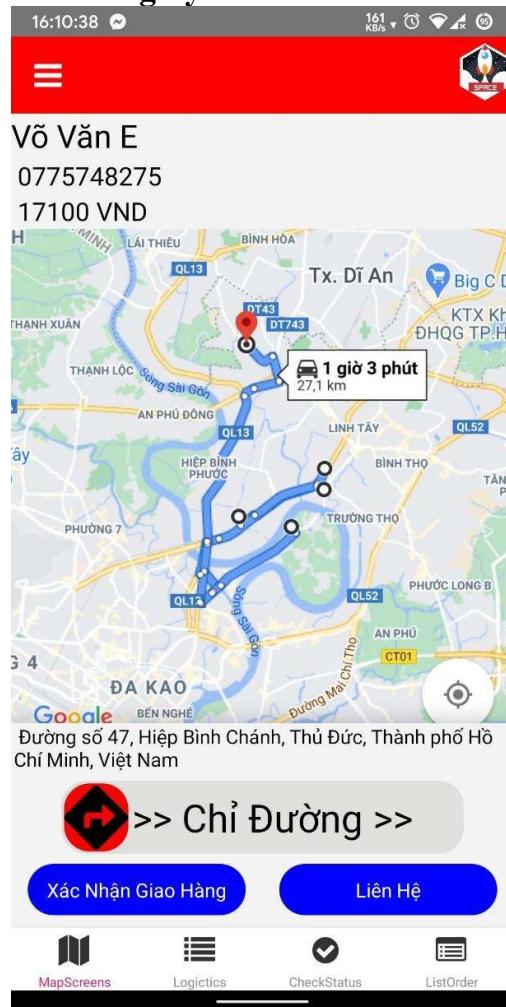
Các đối tượng trong màn hình:

Bảng 24: Đối tượng trong màn hình Map chưa đăng ký chuỗi

STT	Loại	Ý nghĩa	Mô tả
1	Navigation Bottom “Logictics”	Chuyển tới màn hình Logictics	
2	Navigation Bottom “CheckStatus”	Chuyển tới màn hình CheckStatus	
3	Navigation Bottom “ListOrder”	Chuyển tới màn hình ListOrder	

## CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

### 3.4.14. Màn hình Map Khi đăng ký chuỗi



*Hình 44: Màn hình Đã đăng ký chuỗi*

Các đối tượng trong màn hình:

*Bảng 25: Đối tượng trong màn hình Đã đăng ký chuỗi*

STT	Loại	Ý nghĩa	Mô tả
1	Slide “Chỉ đường”	Mở ứng dụng map trên điện thoại	
2	TouchableOpacity “Xác nhận giao hàng”	Xác nhận giao hàng	
3	TouchableOpacity	Mở chức năng gọi	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

	“Liên hệ”	điện của điện thoại	
--	-----------	---------------------	--

#### 3.4.15. Màn hình Đăng ký giao hàng



*Hình 45: Màn hình Đăng ký chuỗi đơn hàng*

Các đối tượng trong màn hình:

*Bảng 26: Đối tượng trong màn hình Đăng ký chuỗi đơn hàng*

STT	Loại	Ý nghĩa	Mô tả
1	TouchableOpacity “Chọn”	Chọn chuỗi đơn hàng	
2	TouchableOpacity	Xác nhận giao	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

	“Xác nhận giao hàng”	hàng	
3	TouchableOpacity “Liên hệ”	Mở chức năng gọi điện của điện thoại	

#### 3.4.16. Màn hình Tình trạng Chuỗi đơn



Hình 46: Màn hình Tình trạng chuỗi đơn

Các đối tượng trong màn hình:

Bảng 27: Đối tượng trong màn hình Tình trạng chuỗi đơn

STT	Loại	Ý nghĩa	Mô tả

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

1	TouchableOpacity “Xác nhận hoàn tất”	Nhấn xác nhận hoàn tất chuỗi đơn hàng	
---	---	---------------------------------------	--

#### 3.4.17. Màn hình Chuỗi đơn hàng



*Hình 47: Màn hình Chuỗi đơn hàng*

Các đối tượng trong màn hình:

*Bảng 28: Đối tượng trong màn hình Chuỗi đơn hàng*

STT	Loại	Ý nghĩa	Mô tả
1	Slide “Chỉ đường”	Mở ứng dụng map trên điện thoại	

### CHƯƠNG 3 – THIẾT KẾ ỨNG DỤNG

2	TouchableOpacity “Xác nhận giao hàng”	Xác nhận giao hàng	
3	TouchableOpacity “Liên hệ”	Mở chức năng gọi điện của điện thoại	

#### 3.4.18. Màn hình menu



*Hình 48: Màn hình Menu*

### 3.4.19. Màn hình Thông tin shipper



*Hình 49: Màn hình Thông tin shipper*

### 3.4.20. Màn hình Đăng xuất



*Hình 50: Màn hình Đăng xuất*

Các đối tượng trong màn hình:

*Bảng 29: Đối tượng trong màn hình Đăng xuất*

STT	Loại	Ý nghĩa	Mô tả
1	TouchableOpacity “Đăng xuất”	Nhấn Đăng xuất	

## CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG

### 4.1. CÀI ĐẶT ỨNG DỤNG

**Yêu cầu tiên quyết:**

- + Máy phải có cài đặt npm
- + Máy phải cài đặt IDE là Microsoft Visual Studio Code (để dễ dàng sử dụng)
- + Máy có cài đặt NodeJS
- + Máy phải cài đặt React Native
- + Máy phải cài đặt máy ảo (Android Studio, Genymotion, link tham khảo:  
[https://reactnative.dev/docs/environment-setup?fbclid=IwAR0unDzVgnKRCosSP1H0m0emNRcRxcJIXsQxo32ofnKiK6wc\\_3-gU\\_dslw](https://reactnative.dev/docs/environment-setup?fbclid=IwAR0unDzVgnKRCosSP1H0m0emNRcRxcJIXsQxo32ofnKiK6wc_3-gU_dslw) )

**Các bước cài đặt:**

Ở bài hướng dẫn này, nhóm sử dụng IDE là Microsoft Visual Studio Code:

- + Bước 1:

Clone hoặc download project từ link github:

<https://github.com/UTEK17TieuLuanChuyenNganh/GiaoHang>

- + Bước 2:

Một thư mục có tên là “GiaoHang-master” sẽ xuất hiện sau khi clone project từ github hoặc giải nén file đính kèm báo cáo. Tiến hành vào thư mục “GiaoHang-master”

- + Bước 3:

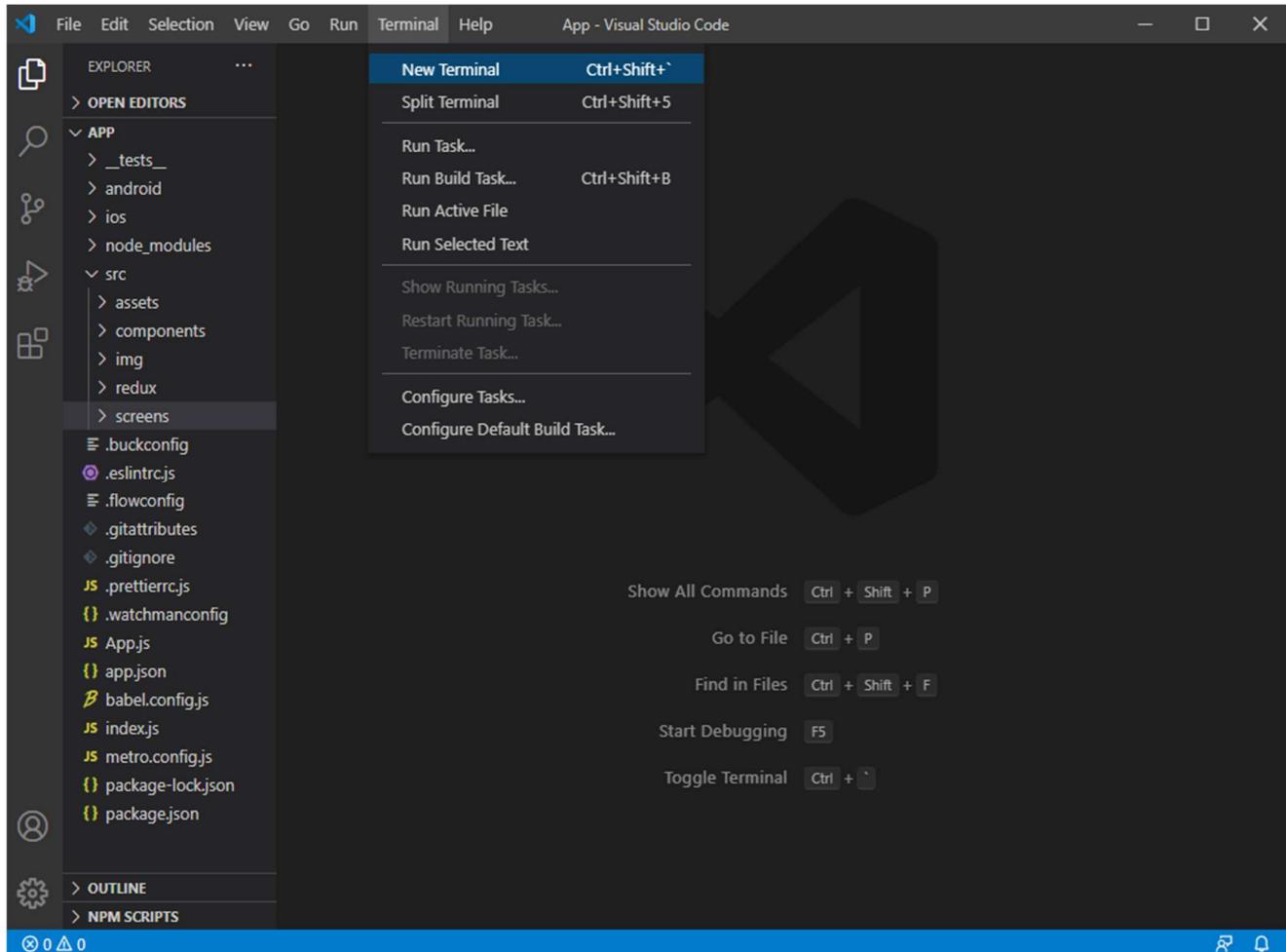
Sau khi vào được thư mục “GiaoHang-master” (như hình), chúng ta chọn ứng dụng muốn sử dụng và click chuột phải chọn “Open with Code”

*Một số máy không mở được theo kiểu mở nhanh này thì có thể mở thẳng Microsoft Visual Studio Code và tìm đến thư mục “App” hoặc “AppGiaoHang” (tài khoản shipper mặc định: caoquyen1913-123).*

## CHƯƠNG 4 – CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG

### Bước 4:

Sau khi mở được project bằng Microsoft Visual Studio Code, chọn “Terminal” trên thanh taskbar, chọn tiếp “New Terminal” (*Hoặc đơn giản là bấm tổ hợp phím Ctrl + `*)

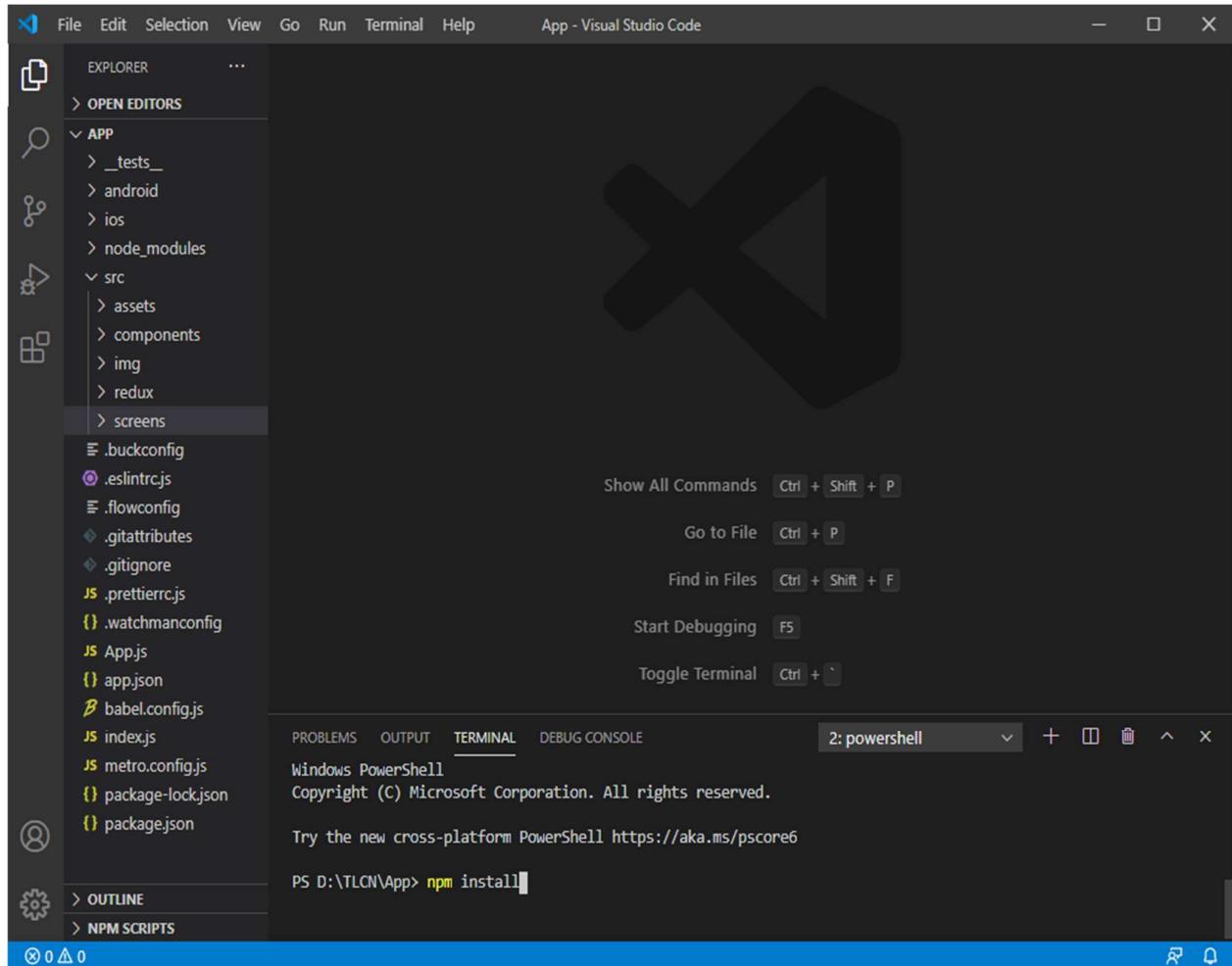


**Hình 51: Giao diện Visual Studio Code**

## CHƯƠNG 4 – CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG

### Bước 5:

Sau khi cửa sổ Command Line tích hợp hiện ra, chạy lệnh “npm install” để cập nhật những thư viện, package còn thiếu trên máy mà Project cần



**Hình 52: Màn hình Intergrate Command Line**

### Bước 6:

Sau khi lệnh npm install kết thúc, gõ dòng lệnh “npx react-native run-android” và đợi ứng dụng chạy.

## 4.2. KIỂM THỬ ỨNG DỤNG

### 4.2.1. Ứng dụng khách hàng

ID	Test Case Description	Test steps	Expected Output	Result
SignUp_User_001	Đăng ký tài khoản người dùng	1. Nhập thông tin người dùng 2. Nhấn nút Đăng ký	Đăng ký thành công và chuyển sang màn hình đăng nhập	Pass
Login_User_01	Đăng nhập khách hàng	1. Nhập thông tin người dùng 2. Nhấn nút Đăng nhập	Đăng nhập thành công và chuyển sang màn hình Profile	Pass
Logout_User_001	Đăng xuất	1. Vào giao diện Profile 2. Nhấn Đăng xuất	Đăng xuất thành công, thông tin đăng nhập của người dùng được xóa khỏi ứng dụng	Pass
Search_Product_001	Tìm sản phẩm theo loại sản phẩm	1. Nhấn vào loại sản phẩm 2. Nhấn vào sản phẩm mong muốn	Chuyển sang màn hình ProductDetail và hiển thị thông tin sản phẩm mong muốn	Pass
Search_Product_002	Tìm sản phẩm theo tên sản phẩm	1. Nhập tên sản phẩm vào thanh tìm kiếm	Các sản phẩm có tên gần giống với tên	Pass

CHƯƠNG 4 – CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG

		2. Nhấn tìm	khách nhập sẽ hiện ra	
AddToCart_001	Thêm sản phẩm vào giỏ hàng	1. Chọn sản phẩm muốn thêm 2. Nhấn vào biểu tượng giỏ hàng 3. Quay trở về trang Home 4. Nhấn vào biểu tượng giỏ hàng trên góc phải	Sản phẩm vừa thêm vào được hiển thị	Pass
RemoveFromCart_001	Xóa sản phẩm khỏi giỏ hàng	1. Nhấn vào dấu “X” trên góc phải sản phẩm trong giỏ hàng	Sản phẩm xóa khỏi giỏ hàng	Pass
AddAddress_001	Thêm địa chỉ	1. Đăng nhập 2. Chọn sản phẩm và thêm vào giỏ hàng 3. Nhấn vào biểu tượng giỏ hàng ở trang Home 4. Nhấn chọn địa chỉ 5. Nhấn thêm	Sau khi thêm đầy đủ thông tin địa chỉ và nhấn nút Lưu, giao diện sẽ trở về màn hình giỏ hàng, khi nhấn vào chọn địa chỉ sẽ hiển thị ra địa chỉ vừa nhập	Pass

CHƯƠNG 4 – CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG

		địa chỉ 6. Thêm thông tin địa chỉ 7. Nhấn Lưu		
SelectAddress_001	Chọn địa chỉ	1. Đăng nhập 2. Chọn sản phẩm và thêm vào giỏ hàng 3. Nhấn vào biểu tượng giỏ hang ở trang Home 4. Nhấn chọn địa chỉ 5. Chọn địa chỉ mong muốn 6. Xác nhận	Sau khi chọn địa chỉ và xác nhận, địa chỉ đã chọn sẽ được hiển thị trên màn hình (Người dùng có thể chọn tối đa 3 địa chỉ khác nhau với 3 khung giờ khác nhau để thuận tiện cho việc nhận hàng mà người dùng mong muốn)	Pass
Order_001	Đặt hàng	1. Chọn sản phẩm thêm vào giỏ hàng 2. Chọn số lượng sản phẩm mong muốn 3. Chọn địa chỉ 4. Nhấn nút Đặt hàng	Đặt hàng thành công, giao diện sẽ chuyển sang màn hình thông báo đặt hang thành công	Pass

CHƯƠNG 4 – CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG

Profile_User_01	Xem thông tin người dùng	1. Vào Profile 2. Chọn Thông tin cá nhân	Thông tin cá nhân của người dùng hiển thị	Pass
OrderManage_001	Xem đơn hàng	1. Vào Profile 2. Chọn Quản lý đơn hàng	Thông tin các đơn hàng được hiển thị	Pass
OrderManage_002	Xem đơn hàng theo ngày	1. Vào Profile 2. Chọn Quản lý đơn hàng 3. Chọn Tìm theo ngày 4. Nhập khoảng thời gian muốn tìm 5. Nhấn Tìm	Thông tin các đơn hàng trong khoảng thời gian được nhập sẽ hiển thị trên giao diện	Pass
ProductManag e_001	Xem sản phẩm đã mua	1. Vào Profile 2. Chọn Sản phẩm đã mua	Thông tin các đơn hàng và sản phẩm được hiển thị	Pass
ProductManag e_002	Xem sản phẩm đã mua theo ngày	1. Vào Profile 2. Chọn Sản phẩm đã mua 3. Chọn Tìm theo ngày 4. Nhập khoảng thời gian muốn tìm 5. Nhấn Tìm	Thông tin các đơn hàng và sản phẩm trong khoảng thời gian được nhập sẽ hiển thị trên giao diện	Pass
Notication_001	Xem thông báo	1. Đăng nhập	Hiển thị tất cả	Pass

## CHƯƠNG 4 – CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG

		2. Vào Notification	các thông báo của người dùng	
--	--	------------------------	---------------------------------	--

### 4.2.2. Ứng dụng giao hàng

ID	Test Case Description	Test steps	Expected Output	Result
Login_Shipper_001	Đăng nhập shipper	1. Nhập thông tin người dùng 2. Nhấn nút Đăng nhập	Đăng nhập thành công và chuyển sang màn hình MapScreen	Pass
Logout_Shipper_001	Đăng xuất	1. Vào Menu 2. Nhấn Cài đặt 3. Nhấn Đăng xuất	Đăng xuất thành công, thông tin đăng nhập của shipper được xóa khỏi ứng dụng	Pass
Profile_Shipper_001	Xem thông tin shipper	1. Vào Menu 2. Chọn Hồ sơ	Thông tin cá nhân của shipper được hiển thị	Pass
ListOrder_001	Xem chuỗi đơn hàng	1. Đăng nhập 2. Chọn ListOrder	Danh sách các chuỗi đơn hàng được hiện ra cho shipper lựa chọn	Pass
RegistListOrder_001	Đăng ký chuỗi đơn hàng	1. Đăng nhập 2. Chọn ListOrder 3. Chọn chuỗi mong muốn 4. Nhấn Chọn	Chuỗi đơn hàng được lưu vào bộ nhớ, thông báo đăng ký thành công	Pass

CHƯƠNG 4 – CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG

CheckStatus_001	Kiểm tra trạng thái	1. Đăng nhập 2. Chọn CheckStatus	Trạng thái hiện tại hiển thị	Pass
CheckStatus_002	Kiểm tra trạng thái khi đã đăng ký chuỗi đơn hàng	1. Đăng nhập 2. Chọn chuỗi đơn hàng 3. Nhấn Chọn 4. Vào Check Status	Trạng thái Tiếp nhận chuỗi và Đang thực hiện được check	Pass
Logitics_001	Danh sách đơn hàng	1. Đăng nhập 2. Vào Logistic	Danh sách đơn hàng hiện tại hiển thị (nếu không có đơn hàng nào thì hiển thị trống)	Pass
Logitisc_002	Danh sách đơn hàng sau khi đăng ký chuỗi đơn hàng	1. Đăng nhập 2. Chọn chuỗi đơn hàng 3. Vào Logitisc	Danh sách các đơn hàng trong chuỗi đơn hàng được hiển thị	Pass
SelectOrder_001	Tiếp nhận đơn hàng	1. Đăng nhập 2. Chọn chuỗi đơn hàng 3. Vào Logitisc 4. Chọn đơn hàng mong muốn	Chuyển sang màn hình MapScreen và hiển thị lộ trình tới địa chỉ đơn hàng	Pass
Map_001	Bản đồ	1. Đăng nhập	Giao diện bản đồ hiện ra (nếu không có đơn	Pass

CHƯƠNG 4 – CÀI ĐẶT VÀ KIỂM THỬ ỨNG DỤNG

			hàng nào đang giao sẽ hiển thị chưa tiếp nhận đơn)	
RouteSlide_001	Thanh chỉ đường	1. Tiếp nhận đơn hàng 2. Kéo thanh chỉ đường	Mở ứng dụng chỉ đường (nếu đã cài đặt trong điện thoại, thông thường đã có sẵn trên điện thoại)	Pass
Call_001	Gọi điện thoại người nhận hàng	1. Tiếp nhận đơn 2. Nhấn Liên hệ	Mở giao diện gọi điện của điện thoại với số điện thoại người nhận được nhập sẵn	Pass
ConfirmStatus Order_001	Xác nhận kết quả giao hàng	1. Tiếp nhận đơn 2. Nhấn Xác nhận giao hàng 3. Chọn kết quả của việc giao hàng (Thành công/Thất bại)	Cập nhật trạng thái của đơn hàng bên Logistic và bản đồ hiển thị lộ trình của đơn hàng tiếp theo	Pass

## Phần 3: KẾT LUẬN

### 1. Kết quả đạt được:

- Kỹ năng làm việc nhóm tốt, kỹ năng giao tiếp trong team, phân công công việc khi làm dự án ngắn ngày và kỹ năng quản lý dự án theo mô hình Agile Scrum.
- Kỹ năng search tài liệu, kỹ năng tìm kiếm giải pháp trên mạng cho những khúc mắc trong quá trình xây dựng đồ án.
- Học được nhiều kỹ năng trong việc xây dựng ứng dụng android, xây dựng được một web server REST API có thể phục vụ cho ứng dụng web lẫn ứng dụng di động.
- Có cái nhìn tổng quan nhất về việc xây dựng ứng dụng web ở vị trí một lập trình viên full-stack.
- Ứng dụng chạy ổn định, giao diện đẹp, tiện dụng. Đáp ứng được nhu cầu của khách hàng lẫn shipper.

### 2. Ưu điểm:

- Ứng dụng có giao diện thân thiện, dễ dàng sử dụng, thân thiện với người dùng.
- Có các công cụ tìm kiếm giúp đáp ứng được các nhu cầu tìm kiếm của khách hàng.
- Đặt hàng nhanh chóng.
- Hệ thống tạo lộ trình giúp shipper có được lộ trình ngắn nhất, tiết kiệm được thời gian và chi phí vận chuyển.
- Shipper dễ dàng lựa chọn chuỗi đơn hàng thích hợp với bản thân.

### 3. Khuyết điểm:

- Vì Google Service (Location API, Routing API) tính phí nên phải sử dụng các viễn khác để thay thế nên mang theo những hạn chế nhất định.
- Hệ thống deploy miễn phí nên có thể xử lý chậm.
- Ứng dụng chạy phụ thuộc khá nhiều vào tốc độ Internet.

### 4. Hướng phát triển tiếp theo:

- Tìm kiếm theo vị trí gần nhất so với bưu cục.
- Tạo hệ thống quản lý shipper, hàng hóa.
- Tạo các chương trình khuyến mãi, tri ân khách hàng.
- Tìm hướng giải quyết tốt hơn cho các đơn hàng có khung giờ không phù hợp.

## TÀI LIỆU THAM KHẢO

### 1. Link github project:

Link: <https://github.com/UTEK17TieuLuanChuyenNganh/GiaoHang>

### 2. Tài liệu React Native:

Link:

- + Trang chủ React Native: <https://reactnative.dev/docs/getting-started>
- + Khóa học trên mạng:

[https://www.youtube.com/watch?v=WbFcHNbz\\_z&list=PLJ5qtRQovuEO\\_kK\\_wekZt5HXItiqZXeLVY](https://www.youtube.com/watch?v=WbFcHNbz_z&list=PLJ5qtRQovuEO_kK_wekZt5HXItiqZXeLVY)

### 3. Tài liệu NodeJs:

Link:

- + Khóa học căn bản:  
[https://sites.google.com/view/tailieucntt2020/javascript/NodeJS\\_can\\_ban\\_BachNgocToan?fbclid=IwAR2MkcZsfmFF6Yyrxqvjdj891eqdVqOR7NZoi\\_bLGIDMI93-nOkYWuTKdL6r4](https://sites.google.com/view/tailieucntt2020/javascript/NodeJS_can_ban_BachNgocToan?fbclid=IwAR2MkcZsfmFF6Yyrxqvjdj891eqdVqOR7NZoi_bLGIDMI93-nOkYWuTKdL6r4)

### 4. Tài liệu Python:

### 5. Tài liệu Map:

Link:

- + Tài liệu Microsoft: <https://docs.microsoft.com/en-us/bingmaps/rest-services/routes/calculate-a-distance-matrix>

### 6. Thư viện Geocoder:

Link: <https://github.com/devfd/react-native-geocoder>

### 7. Thư viện Hierarchical Clustering:

Link: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html>