

TRƯỜNG ĐH KHTN – ĐHQG.TPHCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

SAIGON PARKING PROJECT

VŨ TƯỜNG BÁCH
HUỲNH THANH BÌNH
PHẠM VIẾT MINH ĐẠO
VŨ HẢI
ĐẶNG ĐỨC TÀI



ĐỒ ÁN TỐT NGHIỆP CỦ NHÂN CNTT
TPHCM, tháng 08 năm 2020

TRƯỜNG ĐH KHTN – ĐHQG.TPHCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

SAIGON PARKING PROJECT

GVHD: ThS. Ngô Ngọc Đăng Khoa

1653005 – Vũ Tường Bách

1653006 – Huỳnh Thanh Bình

1653012 – Phạm Viết Minh Đạo

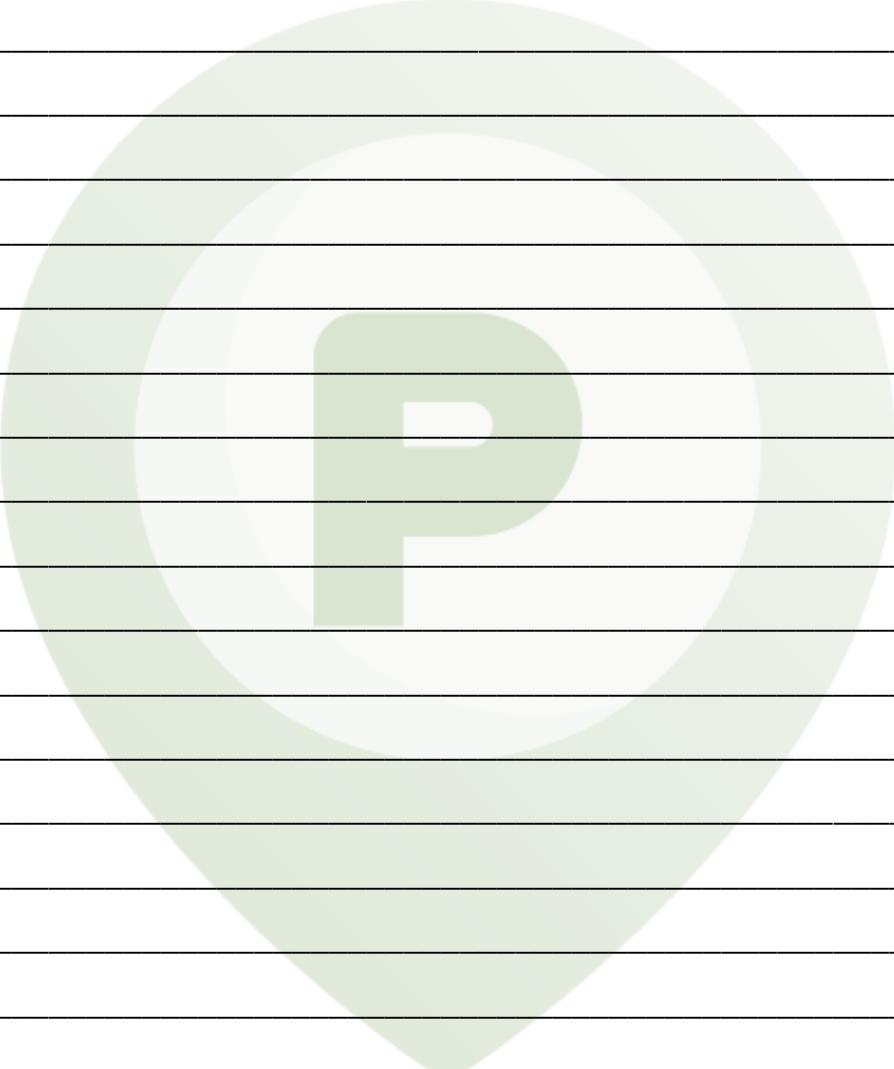
1653020 – Vũ Hải

1653076 – Đăng Đức Tài



ĐỒ ÁN TỐT NGHIỆP CỦ NHÂN CNTT
TPHCM, tháng 08 năm 2020

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN



P

TP.HCM, ngày tháng 8 năm 2020

GVHD

ThS. Ngô Ngọc Đăng Khoa

Nhận xét của giáo viên

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

The image features a large, bold, light green letter 'P' centered on a white background. The 'P' is filled with a solid green color and has a thin black outline. It is positioned within a circular area defined by three concentric rings. The innermost ring is a very light green, the middle ring is a medium shade of green, and the outermost ring is a slightly darker shade. The entire graphic is overlaid on a background of horizontal grey ruling lines, similar to those found in a notebook or ledger.

TP.HCM, ngày tháng 8 năm 2020

GVPB

Nhận xét của giáo viên

LỜI CẢM ƠN

Nhóm chúng em xin gửi lời tri ân sâu sắc nhất đến thầy Ngô Ngọc Đăng Khoa – người đã nhiệt tình giúp đỡ, tận tâm đóng góp ý kiến, định hướng cho chúng em cách tốt nhất để hoàn thành đồ án tốt nghiệp này. Chúng em xin cảm ơn quý Thầy/Cô đã tận tình giảng dạy, trang bị cho chúng em những kiến thức quý báu, cũng như đã tạo điều kiện cho chúng em được học tập và rèn luyện trong môi trường tốt nhất. Đó là hành trang, cũng là cơ sở vững vàng giúp chúng em tiến bước trên con đường phía trước. Đồng thời, chúng em cũng bày tỏ lòng biết ơn đối với quý ban lãnh đạo nhà trường, đã trực tiếp hoặc gián tiếp giúp đỡ, tạo cơ hội cho chúng em thực hiện tốt đồ án tốt nghiệp này trong suốt thời gian vừa qua.

Cũng không thể không nhắc đến sự hợp tác, giúp đỡ tận tình từ phía các chủ xe, tài xế xe ô tô, các bãi xe trên địa bàn thành phố, đã bỏ thời gian quý báu để tham gia những khảo sát cũng như những buổi phỏng vấn trực tiếp cùng chúng em, góp phần giúp chúng em hoàn thiện thu thập yêu cầu để đưa ra hướng tiếp cận cũng như giải quyết bài toán một cách hiệu quả nhất.

Bên cạnh đó, chúng em xin gửi lời cảm ơn chân thành đến gia đình, và bạn bè đã ủng hộ, động viên, giúp đỡ chúng em những lúc khó khăn cũng như trong suốt thời gian học tập và nghiên cứu.

Thời gian thực hiện đồ án khó khăn, đầy thử thách, nhưng đem lại cho chúng em nhiều kinh nghiệm đáng quý, giúp chúng em học hỏi được nhiều điều từ kiến thức đến những kỹ năng trong đời sống mà khó có sách vở nào có thể cung cấp hết được.

Chúng em đã dành hết mọi tâm huyết, nỗ lực để hoàn thành đồ án. Tuy nhiên, với điều kiện về thời gian cũng như kinh nghiệm còn hạn chế của sinh viên, chúng em không thể tránh được những thiếu sót. Chúng em kính mong nhận được sự thông cảm, chỉ bảo và đóng góp ý kiến của quý thầy/ cô để chúng em hoàn thiện hơn.

Một lần nữa, chúng em xin trân trọng cảm ơn và gửi lời chúc sức khỏe, hạnh phúc đến Quý Thầy / Cô !

TP.HCM, ngày tháng 8 năm 2020,

Nhóm thực hiện dự án Saigon Parking

Vũ Tường Bách – Huỳnh Thanh Bình – Vũ Hải

Phạm Viết Minh Đạo – Đăng Đức Tài



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN CÔNG NGHỆ PHẦN MỀM

ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

Tên đề tài: SAIGON PARKING

Loại đề tài: Công nghệ có ứng dụng demo

Giáo viên hướng dẫn: ThS. Ngô Ngọc Đăng Khoa

Thời gian thực hiện: từ ngày 01/02/2020 đến ngày 16/08/2020

Sinh viên thực hiện:

1653005 – Vũ Tường Bách

1653006 – Huỳnh Thanh Bình

1653012 – Phạm Việt Minh Đạo

1653020 – Vũ Hải

1653076 – Đăng Đức Tài

Giới thiệu đề tài:

Hiện nay, đa số mọi người đều đi làm đi chơi,... bằng xe máy hoặc xe ô tô. Với mật độ dân số ngày càng đông, các tuyến đường ngày càng chật hẹp, sáng sớm đi làm thì kẹt xe, tối đến đi chơi lại quá đông, cho nên việc đỗ xe cũng ảnh hưởng không ít tới vấn đề giao thông, vấn đề thời gian và nhiều vấn đề khác. Thay vì đến bãi đỗ xe mà nhận ra rằng đã hết chỗ, đi đến một địa điểm mới mà ta chưa biết nên đỗ xe ở đâu hay tìm bãi đỗ xe vào các ngày lễ hoặc trong giờ cao điểm,... chúng ta cần một ứng dụng để giải quyết bài toán trên một cách hiệu quả.

Mục tiêu đề tài:

- Phát triển được một ứng dụng giúp các tài xế có thể tìm được bãi đỗ xe còn trống gần nhất hoặc gần địa điểm sắp đến.
- Người dùng có thể đặt trước để giữ chỗ đỗ xe, tránh được việc hết chỗ giữ, nhất là vào giờ cao điểm.
- Ứng dụng có tích hợp tính năng tính khoảng cách, chỉ đường từ vị trí hiện tại đến bãi đậu xe và từ bãi đậu xe đến địa điểm mà người dùng muốn đến
- Ứng dụng được tạo ra nhằm giúp người dùng tiết kiệm thời gian thay vì phải tự tìm chỗ để giữ xe.

Phạm vi đề tài:

Phạm vi nghiên cứu hướng tới 2 nhóm đối tượng chính:

- Nhóm có nhu cầu sử dụng dịch vụ đỗ xe: chủ xe ô tô, tài xế xe ô tô.
- Nhóm có nhu cầu cung cấp dịch vụ đỗ xe: các bãi xe, các nhà cung cấp phần mềm quản lý bãi xe

Khu vực nghiên cứu là thành phố Hồ Chí Minh.

Nội dung đề tài:

- Xây dựng ứng dụng tìm kiếm bãi xe và đặt trước chỗ đậu xe với các tính năng phụ trợ đi kèm cho tài xế.
- Ứng dụng gồm các module sau:
 - 1 hệ thống backend được phát triển theo kiến trúc microservice
 - 1 website giả lập ứng dụng bên bãi xe hỗ trợ nhận yêu cầu đặt chỗ bên phía app khách hàng, quản lý lượng yêu cầu đặt chỗ tùy theo hiện trạng bãi xe (chấp nhận/từ chối yêu cầu đặt chỗ, cập nhật số chỗ trong bãi xe), kiểm tra tính đúng đắn của khách hàng (có phải xe ô tô đó là xe đã đặt chỗ trước trên hệ thống), kết thúc quy trình đặt chỗ sau khi kiểm tra thành công và cho xe vào bãi đỗ
 - 1 website admin quản lý tài khoản khách hàng, tài khoản bãi xe với các chức năng cơ bản: thêm, xóa, sửa đổi thông tin (bãi xe, khách hàng), cập nhật đổi thông tin mới.
 - 1 website giới thiệu về ứng dụng, các chức năng chính và hiển thị các bãi đậu xe ô tô trong thành phố Hồ Chí Minh, đăng ký và kích hoạt tài khoản người dùng, đặt lại mật khẩu người dùng
 - 1 app mobile chạy trên hệ điều hành Android giúp các bác tài xế ô tô tương tác và sử dụng các chức năng cần thiết cho việc tìm và đặt trước chỗ đậu xe ô tô
 - 1 app mobile giả lập máy quét QR, kiểm tra mã QR được gửi về app khách hàng có đúng với mã QR mà server đã gửi hay không (mã QR được hiểu như là một Booking ID nhằm kiểm tra xe vào bãi là xe đã đặt chỗ trước thông qua ứng dụng)

Phương pháp thực hiện:

- Khảo sát ứng dụng, giải pháp tương tự
- Khảo sát, lấy yêu cầu tài xế thông qua survey
- Lấy ý kiến của bãi xe thông qua phỏng vấn trực tiếp
- Phân tích yêu cầu, nghiệp vụ của phần mềm
- Thiết kế kiến trúc, cơ sở dữ liệu hệ thống
- Thiết kế prototype cho mỗi module của dự án

Kế hoạch thực hiện:

01/02/2020 – 16/02/2020	<ul style="list-style-type: none"> • họp nhóm giới thiệu và phân tích thế mạnh của từng thành viên • họp nhóm với giáo viên hướng dẫn để bắt tay vào việc làm đồ án • phân tích nghiệp vụ hệ thống • nghiên cứu các công nghệ liên quan • tìm kiếm các giải pháp, ứng dụng tương tự trên thị trường
17/02/2020 – 01/03/2020	<ul style="list-style-type: none"> • phỏng vấn bãi xe và khảo sát tài xế xe ô tô • phân tích ưu, nhược điểm các giải pháp hiện có • đề xuất hướng chức năng, hướng xây dựng dự án saigonparking • tìm kiếm nguồn dữ liệu mẫu • quyết định công nghệ thực hiện dự án

02/03/2020 – 15/03/2020	<ul style="list-style-type: none"> thiết kế prototype các module frontend thiết kế logo nhận diện dự án, logo cho các ứng dụng thiết kế nghiệp vụ hệ thống và các quy trình hoạt động thiết kế cơ sở dữ liệu hệ thống thiết kế kiến trúc hệ thống hướng dẫn sử dụng các công nghệ mới
16/03/2020 – 29/03/2020	<ul style="list-style-type: none"> cài đặt backend theo kiến trúc monolith cài đặt cơ sở dữ liệu cho hệ thống thiết kế và xây dựng API cho những chức năng cơ bản xây dựng những chức năng cơ bản của mobile app cho tài xế xây dựng những chức năng cơ bản của website chính thức của dự án
30/03/2020 – 12/04/2020	<ul style="list-style-type: none"> kiểm thử backend và các API cơ bản kiểm thử những chức năng cơ bản của website chính thức của dự án kiểm thử những chức năng đã cài đặt của mobile app cho tài xế xây dựng những chức năng cơ bản của mobile app cho tài xế (tt) bảo vệ API với xác thực và phân quyền tại backend nghiên cứu hướng chuyển đổi backend sang kiến trúc microservice nghiên cứu sử dụng Docker để đóng gói và triển khai service cài đặt Envoy Proxy để chuyển đổi protocol grpc-web sang grpc
13/04/2020 – 26/04/2020	<ul style="list-style-type: none"> thiết kế và chuyển đổi backend sang kiến trúc microservice thiết kế và cài đặt cơ sở dữ liệu theo hướng microservice chuyển đổi API theo hướng kiến trúc microservice cài đặt Kong làm API Gateway cho hệ thống đóng gói và triển khai các service với Docker, Docker Compose nghiên cứu Amazon S3 và Amazon RDS để lưu trữ dữ liệu kiểm thử những chức năng đã cài đặt của mobile app cho tài xế xây dựng những chức năng cơ bản của web app dành cho admin
27/04/2020 – 10/05/2020	<ul style="list-style-type: none"> kiểm thử backend mới cài đặt theo hướng microservice kiểm thử API sau khi chuyển sang hướng microservice kiểm thử những chức năng đã cài đặt của web app dành cho admin xây dựng những chức năng cơ bản của web app dành cho admin (tt) triển khai sử dụng Amazon S3 và Amazon RDS để lưu trữ dữ liệu nghiên cứu sử dụng Message Queue để giao tiếp bất đồng bộ giữa các microservice ở backend
11/05/2020 – 31/05/2020	<ul style="list-style-type: none"> kiểm thử những chức năng đã cài đặt của web app dành cho admin kiểm thử những chức năng đã cài đặt của mobile app cho tài xế xây dựng những chức năng chính của website chính thức của dự án xây dựng những chức năng cơ bản của web app dành cho bãi xe cài đặt RabbitMQ làm Message Queue để giao tiếp bất đồng bộ giữa các microservice ở backend nghiên cứu về Service Discovery cũng như cách thức triển khai

01/06/2020 – 14/06/2020	<ul style="list-style-type: none"> • triển khai Service Discovery bằng cách cài đặt Consul làm Service Registry, không cần cấu hình / lưu trữ hard-code địa chỉ của các microservice trong hệ thống mạng nội bộ • kiểm thử những chức năng đã cài đặt của web app dành cho bãi xe • xây dựng những chức năng cơ bản của web app dành cho bãi xe (tt) • nghiên cứu hướng triển khai backend lên cloud server (GCP) • nghiên cứu áp dụng WebSocket cho gửi nhận real-time
15/06/2020 – 28/06/2020	<ul style="list-style-type: none"> • triển khai backend lên GCP sử dụng Google Virtual Machine • cấu hình domain và SSL cho cloud server • tiến hành xây dựng service liên quan đến nghiệp vụ booking • cài đặt WebSocket để giao tiếp 2 chiều giữa client và server • xây dựng chức năng liên quan booking cho các module frontend • xây dựng mobile app giả lập máy quét QR cho bãi xe • đóng gói các web app bằng NGINX để triển khai lên cloud server
29/06/2020 – 26/07/2020	<ul style="list-style-type: none"> • sửa lỗi và hoàn thiện mobile app dành cho tài xế • sửa lỗi và hoàn thiện mobile app giả lập máy quét QR cho bãi xe • sửa lỗi và hoàn thiện web app dành cho admin • sửa lỗi và hoàn thiện web app dành cho bãi xe • sửa lỗi và hoàn thiện web chính thức của dự án • sửa lỗi và hoàn thiện backend của dự án • lên dàn ý cho báo cáo, slide thuyết trình và kịch bản demo
27/07/2020 – 16/08/2020	<ul style="list-style-type: none"> • tinh chỉnh giao diện của tất cả các app phía frontend • tinh chỉnh toàn bộ mã nguồn của dự án • hoàn thành báo cáo dự án, slide và kịch bản demo nộp GVHD • hoàn thiện toàn bộ theo hướng dẫn của giảng viên và quay clip demo

Kết quả đạt được:

- Ứng dụng cơ bản đáp ứng được các nhu cầu về việc tìm bãi đậu và đặt trước chỗ của tài xế xe ô tô
- Hiểu được nghiệp vụ, quy trình hoạt động của các bãi giữ xe trên địa bàn thành phố Hồ Chí Minh
- Áp dụng được kiến trúc microservice trong xây dựng hệ thống – kiến trúc đang là xu hướng hiện tại
- Triển khai được dự án lên cloud server, cấu hình thành công domain và SSL cho server.
- Ứng dụng được các công nghệ mới trong dự án như gRPC, ReactJS, Kong, Consul, RabbitMQ, ...

Ý kiến của giảng viên hướng dẫn Chữ ký của giảng viên hướng dẫn ThS. Ngô Ngọc Đăng Khoa	TP. HCM, ngày tháng 8 năm 2020 Nhóm sinh viên thực hiện Vũ Tường Bách – Huỳnh Thanh Bình – Vũ Hải Phạm Viết Minh Đạo – Đăng Đức Tài
---	--

MỤC LỤC

Nhận xét của giáo viên hướng dẫn

Nhận xét của giáo viên phản biện

Lời cảm ơn

Đề cương đồ án tốt nghiệp

TRA CÚU

Mục lục	ix
Danh mục hình ảnh	ix
Danh mục bảng biểu	xiii
Thuật ngữ, ký hiệu, từ viết tắt	xv

CHƯƠNG 1 – TỔNG QUAN DỰ ÁN

1.1. Phát biểu vấn đề	1
1.2. Mục tiêu của dự án.....	1
1.3. Nội dung dự án	2
1.4. Kế hoạch thực hiện	2

CHƯƠNG 2 – KHẢO SÁT HIỆN TRẠNG

2.1. Khảo sát nhu cầu thực tế	5
2.1.1. Nhu cầu của tài xế	5
2.1.2. Nhu cầu của bãi xe	5
2.2. Khảo sát nghiệp vụ của bãi xe	6
2.2.1. Nghiệp vụ kiểm tra xe ô tô ra/vào bãi	6
2.3. Khảo sát ứng dụng tương tự	6
2.3.1. Ứng dụng iParking của CIS	6
2.3.2. Ứng dụng MyParking của Viettel	6
2.3.3. Ứng dụng PakMe của PakMe	6
2.3.4. Phân tích ưu, nhược điểm của các ứng dụng	6
2.4. Danh sách yêu cầu	8
2.4.1. Yêu cầu chức năng	8
2.4.2. Yêu cầu phi chức năng	8
2.5. Giải pháp đề xuất	9
2.5.1. Tìm kiếm và gợi ý bãi xe quanh vị trí cố định, kết hợp lọc kết quả tìm theo khoảng cách, giá cả	9

2.5.2. Cho phép đặt trước chỗ đậu xe thông qua ứng dụng và xem lại lịch sử đỗ xe	9
2.5.3. Cho phép tương tác với bãi xe thông qua ứng dụng: gửi/nhận tin nhắn văn bản,	9
2.5.4. Cho phép đánh giá, xem đánh giá bãi xe thông qua ứng dụng	9
2.5.5. Chỉ đường/ điều hướng đến bãi xe / địa điểm thứ ba trực tiếp trên ứng dụng	9
2.5.6. Chỉ đường/ điều hướng đến bãi xe / địa điểm thứ ba thông qua ứng dụng Google Map	9
2.5.7. Gợi ý các địa điểm hỗ trợ liên quan cho tài xế xe ô tô (trạm xăng, trạm dừng, ...).....	9
2.5.8. Đơn giản hóa việc hoàn tất quá trình đặt chỗ cho cả đôi bên bằng phương pháp scan mã QR	9
2.5.9. Cân bằng tải để hệ thống có khả năng cùng lúc xử lý số lượng lớn yêu cầu từ người dùng	9

CHƯƠNG 3 – THIẾT KẾ HỆ THỐNG

3.1. Tổng quan kiến trúc microservice	10
3.1.1. Kiến trúc monolithic là gì	10
3.1.2. Kiến trúc microservice là gì	10
3.1.3. So sánh kiến trúc monolithic và kiến trúc microservice	10
3.1.4. API gateway là gì	11
3.1.5. Service Discovery là gì	11
3.1.5.1. Khái niệm về truy tìm dịch vụ	11
3.1.5.2. Client-side service discovery là gì	12
3.1.5.3. Server-side service discovery là gì	13
3.2. Kiến trúc dự án	13
3.2.1. Các modul của dự án	13
3.2.1.1. Backend	13
3.2.1.1.1. Auth Service	13
3.2.1.1.2. Mail Service	13
3.2.1.1.3. Parking-lot Service	14
3.2.1.1.4. Booking Service	14
3.2.1.1.5. Contact Service	14
3.2.1.2. Website danh cho quản trị viên	14
3.2.1.3. Website chính thức của dự án	14
3.2.1.4. Website giả lập ứng dụng quản lý bãi giữ xe ô tô	14
3.2.1.5. Ứng dụng di động dành cho tài xế ô tô	14
3.2.1.6. Ứng dụng di động giả lập thiết bị scan mã QR tại bãi giữ xe	15
3.2.2. Kiến trúc tổng quan (architecture)	15
3.3. Mô hình hoạt động	16
3.4. Mô hình use-case (use-case diagram)	17
3.4.1. Tổng quan	17

3.4.2. Đặc tả chi tiết use-case	19
3.4.2.1. Use-case sử dụng chung	19
3.4.2.1.1. UC 1: Đăng nhập	19
3.4.2.1.2. UC 2: Đăng xuất	20
3.4.2.1.3. UC 3: Xem thông tin cá nhân	21
3.4.2.1.4. UC 4: Thay đổi mật khẩu	22
3.4.2.2. Use-case của khách hàng	22
3.4.2.2.1. UC 5: Tạo tài khoản	22
3.4.2.2.2. UC 6: Kích hoạt tài khoản	23
3.4.2.2.3. UC 7: Quên mật khẩu	24
3.4.2.2.4. UC 8: Xác thực bằng email	24
3.4.2.2.5. UC 9: Xem bản đồ bãi đỗ xe	25
3.4.2.2.6. UC 10: Hiện danh sách các bãi đỗ đang có trên bản đồ	26
3.4.2.2.7. UC 11: Xem thông tin chi tiết của bãi đỗ xe	26
3.4.2.2.8. UC 12: Chỉ đường đến vị trí bãi đỗ xe	27
3.4.2.2.9. UC 13: Đặt chỗ đỗ xe	28
3.4.2.2.10. UC 14: Xác thực đặt chỗ bằng QR-code	29
3.4.2.2.11. UC 15: Chat hỗ trợ khách hàng	30
3.4.2.2.11. UC 16: Đánh giá bãi đỗ xe	31
3.4.2.2.11. UC 17: Xem các dịch vụ ôtô khác	32
3.4.2.2.14. UC-18: Xem lịch sử đặt chỗ bãi xe	32
3.4.2.3. Use-case của bãi xe	33
3.4.2.3.1. UC 19: Xem dữ liệu thống kê của bãi đỗ	33
3.4.2.3.2. UC 20: Cập nhật thông tin bãi đỗ	33
3.4.2.3.3. UC 21: Phản hồi lại khách hàng về việc đặt chỗ	34
3.4.2.3.4. UC 22: Tư vấn khách hàng	35
3.4.2.3.5. UC 23: Hoàn tất đặt chỗ đỗ xe	36
3.4.2.3.6. UC 24: Xác nhận khách đỗ xe bằng cách quét QR-code	36
3.4.2.4. Use-case của admin	37
3.4.2.4.1. UC 25: Xem dữ liệu thống kê	37
3.4.2.4.2. UC 26: Tạo tài khoản người dùng	38
3.4.2.4.3. UC 27: Xem thông tin người dùng	38
3.4.2.4.4. UC 28: Kích hoạt người dùng	39
3.4.2.4.5. UC 29: Vô hiệu hóa người dùng	40
3.4.2.4.6. UC 30: Cấp quyền quản lý bãi đỗ xe	40

3.4.2.4.7. UC 31: Xem thông tin bãi đỗ xe	41
3.4.2.4.8. UC 32: Xoá bãi đỗ xe	42
3.5. Các quy trình hoạt động chính (activity diagram)	42
3.5.1. Quy trình tạo tài khoản và xác thực người dùng	42
3.5.2. Quy trình người dùng đặt chỗ bãi đỗ xe	42
3.5.3. Quy trình chỉ đường người dùng và xác thực đặt chỗ bằng QR code	43
3.6. Danh sách các chức năng	44
3.6.1. Danh sách chức năng dành cho ứng dụng web của Admin	44
3.6.2. Danh sách chức năng dành cho ứng dụng web của dự án	46
3.6.3. Danh sách chức năng dành cho ứng dụng di động của tài xế	47
3.6.4. Danh sách chức năng dành cho ứng dụng web giả lập ứng dụng quản lý bãi giữ xe	47
3.6.5. Danh sách chức năng dành cho ứng dụng app giả lập ứng dụng quét mã QR	48
3.7. Cơ sở dữ liệu của dự án (database diagram)	48
3.7.1. Tổng quan	48
3.7.2. UUID (GUID) là gì	48
3.7.3. Cơ sở dữ liệu của User service	49
3.7.4. Cơ sở dữ liệu của Auth service	49
3.7.5. Cơ sở dữ liệu của Parking-lot service	50
3.7.6. Cơ sở dữ liệu của Booking service	51

CHƯƠNG 4 – GIAO TIẾP GIỮA CÁC THÀNH PHẦN HỆ THỐNG

4.1. Protobuf là gì	52
4.2. So sánh Protobuf và JSON	52
4.3. Unidirectional communication là gì	53
4.4. Bidirectional communication là gì.....	53
4.5. gRPC là gì	53
4.6. So sánh gRPC và REST	55
4.7. Websocket là gì	55
4.8. So sánh Websocket và HTTP	56
4.9. Message Queue là gì	56
4.10. Cân bằng tải hệ thống	57
4.10.1. Giao tiếp đồng bộ là gì	58
4.10.2. Giao tiếp bất đồng bộ là gì	58
4.10.3. Cân bằng tải là gì	58
4.10.4. Các thuật toán cân bằng tải thông dụng	58

4.10.5. Client-side load-balancing và server-side load-balancing	58
4.10.6. Cân bằng tải giao tiếp từ bên ngoài hệ thống	59
4.10.7. Cân bằng tải giao tiếp từ bên ngoài hệ thống đối với websocket	60
4.10.7.1. Mở kết nối WebSocket với server	60
4.10.7.2. Gửi nhận message với WebSocket	61
4.10.8. Cân bằng tải giao tiếp từ bên trong hệ thống (đồng bộ)	62
4.10.9. Cân bằng tải giao tiếp từ bên trong hệ thống (bất đồng bộ)	62
4.11. Danh sách các API cung cấp	63
4.11.1. API WebSocket	63
4.11.2. API gRPC	63
4.11.2.1. Auth Service API	63
4.11.2.2. User Service API	64
4.11.2.3. Parking Lot Service API	65
4.11.2.4. Booking Service API	66
4.11.2.5. Contact Service API	68
4.12. Vấn đề xác thực và phân quyền	69
4.12.1. Interceptor là gì	69
4.12.2. JSON Web Token (JWT) là gì	69
4.12.3. Xác thực và phân quyền với JWT	70
4.12.3.1. Sử dụng JWT với gRPC	71
4.12.3.2. Sử dụng JWT với WebSocket	71

CHƯƠNG 5 – TRIỂN KHAI THỰC NGHIỆM

5.1. Lựa chọn công nghệ	73
5.1.1. Tổng quan công nghệ sử dụng trong dự án	73
5.1.2. Java	74
5.1.3. Spring Boot	74
5.1.4. Hibernate	74
5.1.5. Consul	75
5.1.6. RabbitMQ	75
5.1.7. Kong	75
5.1.8. Envoy	75
5.1.9. Docker	75
5.1.10. NGINX	76
5.1.11. NPM	76

5.1.12. Maven	76
5.1.13. Gradle	76
5.1.14. ReactJS	76
5.1.15. Microsoft SQL Server	77
5.1.16. Github.....	77
5.1.17. Amazon Web Service.....	77
5.1.17.1. Amazon RDS	77
5.1.17.2. Amazon S3	77
5.1.18. Google Cloud Platform	78
5.1.18.1. Google Maps API	78
5.1.18.2. Google Places API	78
5.1.18.3. Google Directions API	78
5.1.18.4. Google Cloud Compute Engine	78
5.1.19. IDE	78
5.1.19.1. IntelliJ IDEA	78
5.1.19.2. Android Studio	78
5.1.19.3. Visual Studio Code	78
5.1.20. Các thư viện, tiện ích và công cụ hỗ trợ	79
5.1.20.1. SQL Server Management Studio	79
5.1.20.2. Lombok	79
5.1.20.3. MapStruct	79
5.1.20.4. SonarLint	79
5.1.20.5. AndroidLint	79
5.1.20.6. ESLint	79
5.1.20.7. SaveActions	79
5.1.20.8. Multirun	79
5.1.20.9. Docker Compose	79
5.2. Lưu trữ mã nguồn với Github	80
5.3. Đóng gói dự án sử dụng Docker, Docker Compose và NGINX	80
5.4. Triển khai dự án với Google Cloud Platform và Amazon Web Service	81
5.5. Sử dụng Github Actions cho CI/CD	82
5.6. Giới thiệu các chức năng của hệ thống (screenshots)	84
5.6.1. Website chính thức của dự án	84
5.6.1.1. Đăng nhập	84
5.6.1.2. Đăng ký	84

5.6.1.3. Quên mật khẩu	85
5.6.1.4. Xem vị trí các bãi xe	86
5.6.1.5. Xem thông tin chi tiết bãi xe	86
5.6.1.6. Bình luận đánh giá.....	87
5.6.1.7. Xem thông tin cơ bản về chức năng	88
5.6.2. Website dành cho quản trị viên	89
5.6.2.1. Chức năng đăng nhập	89
5.6.2.2. Chức năng thống kê	89
5.6.2.3. Chức năng quản lý User	90
5.6.2.4. Chức năng quản lý bãi xe	91
5.6.2.5. Chức năng đổi mật khẩu	95
5.6.2.6. Chức năng thêm User quản lý bãi xe	95
5.6.3. Ứng dụng di động dành cho tài xế xe ô tô	96
5.6.3.1. Chức năng đăng ký/đăng nhập	97
5.6.3.2 Chức năng tìm kiếm bãi xe quanh vị trí hiện tại	98
5.6.3.3. Chức năng tìm kiếm bãi xe quanh địa điểm cần đến	98
5.6.3.4. Chức năng hiển thị các địa điểm khác.....	98
5.6.3.5. Chức năng chỉ đường đến bãi xe bằng Google Direction API.....	101
5.6.3.6. Chức năng chỉ đường đến bãi xe bằng ứng dụng hệ thống	101
5.6.3.7 Chức năng tìm địa điểm thứ ba	102
5.6.3.8 Chức năng đặt trước chỗ đậu xe	104
5.6.3.9. Chức năng chat với bãi xe	106
5.6.3.10 Chức năng đánh giá booking service	107
5.6.3.11. Chức năng xem lịch sử đặt chỗ	107
5.6.4. Giả lập ứng dụng quản lý bãi giữ xe ô tô	108
5.6.4.1. Chức năng đăng nhập	108
5.6.4.2. Chức năng thống kê.....	108
5.6.4.3. Chức năng phản hồi yêu cầu đặt chỗ	109
5.6.4.4. Chức năng kết thúc quy trình đặt chỗ	109
5.6.4.5. Chức năng cập nhật số lượng chỗ	110
5.6.5. Giả lập máy quét mã QR	110
5.6.5.1. Chức năng đăng nhập	110
5.6.5.2. Chức năng đăng xuất	111
5.6.5.3. Chức năng quét mã QR để kết thúc booking	111

CHƯƠNG 6 – KẾT LUẬN, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận	113
6.1.1. Về mặt lý thuyết	113
6.1.2. Về mặt thực nghiệm	113
6.2. Đánh giá	113
6.2.1. Ưu điểm	113
6.2.2. Hạn chế	114
6.3. Hướng phát triển	114
6.3.1. Hướng phát triển backend	114
6.3.2. Hướng phát triển ứng dụng dành cho tài xế	114
6.3.3. Hướng phát triển ứng dụng dành cho bãi xe	114
6.3.4. Hướng phát triển ứng dụng admin	114

CHƯƠNG 7 – PHỤ LỤC VÀ TÀI LIỆU THAM KHẢO

7.1. Phụ lục	115
7.1.1. Môi trường đã sử dụng để phát triển các module của dự án	115
7.1.2. Hướng dẫn cài đặt dự án (dành cho thực nghiệm / kiểm thử tại máy local)	115
7.1.3. Phiên bản của các thư viện, công cụ, plugin sử dụng trong dự án	116
7.1.4. Chi phí cài đặt và vận hành dự án	117
7.1.4.1. Amazon RDS	117
7.1.4.2. Amazon S3	118
7.1.4.3. Google Place API	119
7.1.4.4. Google Map API	122
7.1.4.5. Google Directions API	123
7.1.4.6. Google Virtual Machine	123
7.1.4.7. Gmail	123
7.1.4.8. Domain và SSL	123
7.1.4.9. Chi phí khảo sát thực tế	124
7.1.5. Kết quả khảo sát	124
7.1.6. Giới thiệu nhóm phát triển dự án Saigon Parking	129
7.1.7. Phân công công việc	130
7.2. Tài liệu tham khảo	131

DANH MỤC HÌNH ẢNH

Hình 3.1.5.1.a. Minh họa bài toán truy tìm dịch vụ và cân bằng tải với kiến trúc microservice.....	11
Hình 3.1.5.1.b. Minh họa kiến trúc microservice và vai trò của API Gateway, Service Discovery	12
Hình 3.1.5.2. Minh họa truy tìm dịch vụ, sử dụng phương pháp Client-side service discovery	12
Hình 3.1.5.3. Minh họa truy tìm dịch vụ, sử dụng phương pháp Server-side service discovery.....	13
Hình 3.2.2. Kiến trúc tổng quan của hệ thống saigonparking	15
Hình 3.3. Mô hình hoạt động của hệ thống saigonparking	16
Hình 3.4.1. Sơ đồ use-case của hệ thống saigonparking.....	17
Hình 3.5.1. Quy trình tạo tài khoản và xác thực người dùng	43
Hình 3.5.2. Quy trình người dùng đặt chỗ bãi xe.....	43
Hình 3.5.3. Quy trình chỉ đường người dùng và xác thực đặt chỗ bằng QR code.....	44
Hình 3.7.3. Cơ sở dữ liệu của User Service.....	49
Hình 3.7.4. Cơ sở dữ liệu của Auth Service	49
Hình 3.7.5. Cơ sở dữ liệu của Parking Lot Service	50
Hình 3.7.6. Cơ sở dữ liệu của Booking Service	51
Hình 4.2. Minh họa việc so sánh performance giữa Protocol Buffer và JSON	53
Hình 4.5.a. Minh họa về cơ chế RPC – Remote Procedure Call	54
Hình 4.5.b. Minh họa về cách thức cài đặt và sử dụng gRPC.....	55
Hình 4.7.c. Minh họa toàn bộ quá trình giao tiếp bằng giao thức WebSocket	55
Hình 4.9.a. Minh họa mô hình hoạt động của RabbitMQ.....	57
Hình 4.9.b. Minh họa kết nối giữa các microservice với RabbitMQ trong hệ thống SaigonParking	57
Hình 4.10.6. Minh họa cân bằng tải giao tiếp từ bên ngoài hệ thống (với hệ thống saigonparking)	59
Hình 4.10.7.1. Minh họa mở kết nối WebSocket với server (với hệ thống saigonparking, đã cân bằng tải) .	60
Hình 4.10.7.2. Minh họa gửi nhận message với WebSocket (với hệ thống saigonparking, đã cân bằng tải) .	61
Hình 4.10.8. Minh họa cân bằng tải giao tiếp đồng bộ bên trong hệ thống saigonparking.....	62
Hình 4.10.9. Minh họa cân bằng tải giao tiếp bất đồng bộ bên trong hệ thống saigonparking	62
Hình 4.12.1. Minh họa cách thức sử dụng interceptor trong việc xác thực và phân quyền	69
Hình 4.12.2. Minh họa nội dung của chuỗi JWT	70
Hình 4.12.3. Minh họa nội dung 1 access token của hệ thống saigonparking	70
Hình 5.1.1. Tổng quan công nghệ sử dụng trong dự án saigonparking	73

Hình 5.4. Minh họa triển khai hệ thống saigonparking lên cloud-server sử dụng AWS và GCP	81
Hình 5.5.a. Minh họa trạng thái của Github Actions trong lịch sử commit của Github saigonparking	82
Hình 5.5.b. Minh họa 1 phiên làm việc của Github Actions với 1 commit cụ thể (build + test + deploy)	83
Hình 5.6.1.1.a. Hình chức năng đăng nhập và kích hoạt tài khoản	84
Hình 5.6.1.2.a. Hình chức năng đăng ký	85
Hình 5.6.1.2.b. Hình thông báo đăng ký thành công	85
Hình 5.6.1.3.a Hình chức năng quên mật khẩu	86
Hình 5.6.1.4.a. Hình bản đồ chính	86
Hình 5.6.1.5.a. Hình thông tin chi tiết bãi xe	87
Hình 5.6.1.6.a. Hình chức năng đánh giá và xem đánh giá	87
Hình 5.6.1.7.a. Hình trang chủ của hệ thống	88
Hình 5.6.2.1.a. Hình chức năng đăng nhập	89
Hình 5.6.2.2.a. Hình chức năng xem thống kê	89
Hình 5.6.2.3.a. Hình chức năng xem danh sách User	90
Hình 5.6.2.3.b. Hình thông tin chi tiết User theo từng Role	91
Hình 5.6.2.3.c. Hình chức năng thêm User	92
Hình 5.6.2.4.a. Hình chức năng xem danh sách bãi xe	93
Hình 5.6.2.4.b. Hình thông tin chi tiết của bãi xe	93
Hình 5.6.2.4.c. Hình chức năng quản lý danh sách User quản lý bãi xe	94
Hình 5.6.2.4.d. Hình chức năng thêm bãi xe mới	94
Hình 5.6.2.5.a. Hình chức năng đổi mật khẩu	95
Hình 5.6.2.6.a Hình chức năng thêm User thành người quản lý bãi xe	95
Hình 5.6.2.6.b. Hình ảnh thông báo User đã có quản lý 1 bãi xe nào đó rồi	96
Hình 5.6.2.6.c. Hình thông báo có thể thêm 1 User thành quản lý bãi xe	96
Hình 5.6.3. Hình đại diện ứng dụng di động dành cho tài xế xe ô tô	96
Hình 5.6.3.1.a. Màn hình xác nhận cho phép sử dụng camera điện thoại	97
Hình 5.6.3.1.b. Màn hình đăng nhập	97
Hình 5.6.3.1.c. Màn hình đăng ký	97
Hình 5.6.3.1.d: Màn hình bản đồ	98
Hình 5.6.3.2.a. Màn hình hiển thị bãi xe quanh vị trí hiện tại khi nhấn nút "FIND PARKING LOT"	99
Hình 5.6.3.2.b. Màn hình hiển thị bãi xe quanh vị trí hiện tại khi bản đồ tải lên dữ liệu bãi xe	99
Hình 5.6.3.2.c. Hiển thị danh sách các bãi xe quanh vị trí hiện tại khi kéo màn hình từ dưới lên	99

Hình 5.6.3.3.a. Màn hình tìm kiếm địa điểm cần đến	99
Hình 5.6.3.3.b. Bản đồ di chuyển đến địa điểm vừa tìm và hiển thị các bãi xe gần đó	99
Hình 5.6.3.3.c. Hiển thị danh sách các bãi xe quanh địa điểm vừa tìm khi kéo màn hình từ dưới lên.....	99
Hình 5.6.3.4.a. Hiển thị danh sách các địa điểm khác.....	100
Hình 5.6.3.4.b. Hiển thị danh sách các trạm xăng	100
Hình 5.6.3.4.c. Hiển thị danh sách nhà hàng	100
Hình 5.6.3.4.d. Hiển thị danh sách bệnh viện.....	100
Hình 5.6.3.5.a. Màn hình hiển thị thông tin chi tiết bãi xe	101
Hình 5.6.3.5.b. Màn hình chỉ đường đến bãi xe	101
Hình 5.6.3.6.a. Màn hình Đặt Chỗ	102
Hình 5.6.3.6.b. Màn hình chỉ đường tới bãi đỗ	102
Hình 5.6.3.7.a. Tìm địa điểm cần đến.....	103
Hình 5.6.3.7.b. Chọn bất kì bãi xe xung quanh địa điểm đó	103
Hình 5.6.3.7.c. Màn hình hiển thị thông tin chi tiết bãi xe	103
Hình 5.6.3.7.d. Màn hình Đặt Chỗ	103
Hình 5.6.3.7.e. Màn hình chỉ đường tới bãi đỗ	104
Hình 5.6.3.7.f. Màn hình chỉ đường tới địa điểm thứ ba.....	104
Hình 5.6.3.8.a. Nhập thông tin cần thiết để đặt chỗ.....	105
Hình 5.6.3.8.b. Màn hình Đặt Chỗ khi đang đợi bãi xe Accept (Status: PROCESSING)	105
Hình 5.6.3.8.c. Bãi xe từ chối yêu cầu đặt chỗ	105
Hình 5.6.3.8.d. Bãi xe chấp nhận yêu cầu đặt chỗ (Status: ACCEPTED)	105
Hình 5.6.3.8.e. Hệ thống hoàn thành Booking	106
Hình 5.6.3.9.a. Màn hình Đặt Chỗ	106
Hình 5.6.3.9.b. Màn hình Chat	106
Hình 5.6.3.10.a: Màn hình đánh giá.....	107
Hình 5.6.3.10.b: Màn hình xem đánh giá của người dùng khác	107
Hình 5.6.3.11.a. Left slide menu	108
Hình 5.6.3.11.b: Màn hình lịch sử booking	108
Hình 5.6.4.1.a. Hình chức năng đăng nhập.....	108
Hình 5.6.4.2.a. Hình thống kê số lượng booking và đánh giá.....	109
Hình 5.6.4.3.a. Hình phản hồi yêu cầu booking người dùng.....	109
Hình 5.6.4.4.a. Hình hoàn tất booking.....	110

Hình 5.6.4.5.a. Hình cập nhật số lượng chỗ	110
Hình 5.6.5. Hình đại diện cho ứng dụng giả lập máy quét mã QR.....	110
Hình 5.6.5.1.a. Màn hình chờ của app.....	111
Hình 5.6.5.1.b. Màn hình quét mã QR để đăng nhập	111
Hình 5.6.5.1.c. Màn hình sau khi đăng nhập thành công.....	111
Hình 5.6.5.2.a. Màn hình đăng xuất.....	113
Hình 5.6.5.3.a. Màn hình quét mã QR để kết thúc booking	113
Hình 7.1.4.1.a: Chi phí cài đặt Amazon RDS.....	118
Hình 7.1.4.2.a: Chi phí cài đặt Amazon S3	119
Hình 7.1.4.3.a: Chi phí cài đặt Google Direction API	122
Hình 7.1.4.4.a: Chi phí cài đặt Google Map API.....	122
Hình 7.1.4.5.a: Chi phí cài đặt Google Direction API	123
Hình 7.1.4.8.a: chi phí cài đặt Domain và SSL	124
Hình 7.1.5.a: biểu đồ thống kê họ tên và độ tuổi người tham gia khảo sát	124
Hình 7.1.5.b: biểu đồ thống kê thói quen và mật độ sử dụng ô tô của người tham gia khảo sát.....	125
Hình 7.1.5.c – 7.1.5.d: biểu đồ thống kê khảo sát thói quen tìm kiếm bãi đỗ xe và thanh toán của người tham gia khảo sát.....	126
Hình 7.1.5.e – 7.1.5.f: biểu đồ khảo sát mong muốn và nhu cầu sử dụng ứng dụng tìm bãi đỗ của khách hàng	127
Hình 7.1.5.h – 7.1.5.g: biểu mẫu khảo sát nhân viên bãi xe	128

DANH MỤC BẢNG BIỂU

Bảng 1. Thuật ngữ, ký hiệu và từ viết tắt	xv
Bảng 2.3.4.a. Phân tích ưu, nhược điểm của ứng dụng iParking	6
Bảng 2.3.4.b. Phân tích ưu, nhược điểm của ứng dụng MyParking	7
Bảng 2.3.4.c. Phân tích ưu, nhược điểm của ứng dụng PakMe	7
Bảng 2.4.1. Danh sách các yêu cầu chức năng	8
Bảng 2.4.2. Danh sách yêu cầu phi chức năng	8
Bảng 3.1.3. So sánh kiến trúc monolithic và kiến trúc microservice	11
Bảng 3.4.1. Danh sách các use-case của hệ thống saigonparking	19
Bảng 3.6.1. Danh sách chức năng dành cho ứng dụng web của Admin	46
Bảng 3.6.2. Danh sách chức năng dành cho ứng dụng web của dự án	46
Bảng 3.6.3. Danh sách chức năng dành cho ứng dụng di động của tài xế	47
Bảng 3.6.4. Danh sách chức năng dành cho ứng dụng web giả lập ứng dụng quản lý bãi giữ xe	48
Bảng 3.6.5. Danh sách chức năng dành cho ứng dụng app giả lập máy quét mã QR	48
Bảng 4.7.a. Các sự kiện chính và các event handler tương ứng trong WebSocket	55
Bảng 4.7.b. Các phương thức chính trong WebSocket	56
Bảng 7.1.3. Thông kê phiên bản của các thư viện, công cụ, plugin đã sử dụng trong dự án	117



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



THUẬT NGỮ, KÝ HIỆU VÀ TỪ VIẾT TẮT

Alphabet	Thuật ngữ, ký hiệu, Từ viết tắt	Định nghĩa
A	Architecture / Architect	Kiến trúc phần mềm / kiến trúc hệ thống
	Administrator / Admin	Quản trị viên hệ thống
	API	Các phương thức, giao thức để kết nối với ứng dụng khác
	API Gateway	(Cổng kết nối API) là một máy chủ truy xuất duy nhất vào hệ thống
	Application / App	Ứng dụng
	Android	Một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng
	Activity	Hoạt động
	Asynchronous / Async	Bất đồng bộ
	Assign / Assignment	Phân công
	Auto complete	Tự động hoàn tất
	AMQP	Advanced Message Queuing Protocol
	AWS	Amazon Web Service
B	Bug	Lỗi
	Backend	Lập trình trên server, xử lý các thông tin từ frontend
	Binary	Nhi phân
	Block	Khóa lại
	Booking	Đặt chỗ
	Bidirectional communication	Giao tiếp 2 chiều
C	CI	Continuous Integration
	CD	Continuous Delivery
	Client	Khách hàng, máy khách
	Compile	Biên dịch
	Comment	Chú thích
	Component	Thành phần
	Consume	Tiêu thụ
	Communication	Giao tiếp

Alphabet	Thuật ngữ, ký hiệu, Từ viết tắt	Định nghĩa
D	Code / Source code	Mã nguồn
	CH Play	Cửa hàng ứng dụng của Google cho điện thoại dùng hệ điều hành android (tên gọi khác: Play Store)
	CSR	Certificate signing request
	Chat	Nhắn tin
E	DB / Database	Cơ sở dữ liệu
	Dashboard	Bảng điều khiển
	Diagram	Biểu đồ, lược đồ
	Download	Tải xuống
	Document	Tài liệu
	Data	Dữ liệu
	Dependency / Dependencies	Phụ thuộc – cách nói khác của Library
	Define / Definition	Định nghĩa
	Deploy / Deployment	Triển khai
	Discover / Discovery	Khám phá
	Domain	Tên miền
	DNS	Hệ thống phân giải tên miền
	Deserialize	Quá trình chuyển đổi từ nhị phân sang đối tượng
	Desktop	Máy tính cá nhân
F	DevOps	Một công việc trong ngành công nghệ phần mềm. Lập trình viên DevOps là người kết nối hai giai đoạn phát triển và vận hành trong chu kỳ phát triển phần mềm. Công việc của Devops Engineer bao gồm deploy, optimizing, monitoring, analysis một phần mềm hệ thống
	Demo	Bản thử nghiệm
E	Environment / Env	Môi trường
	Emulator	Giả lập
	External	Ngoại bộ, bên ngoài hệ thống
	Event	Sự kiện
	Error	Lỗi
	Exception	Ngoại lệ
F	Filter	Bộ lọc

Alphabet	Thuật ngữ, ký hiệu, Từ viết tắt	Định nghĩa
G	Format	Dạng
	Frontend	Hệ thống giao diện người dùng và lập trình phía người dùng
	Framework	Là các đoạn code đã được viết sẵn, cấu thành nên một bộ khung và các thư viện lập trình được đóng gói
G	Gateway	Cách nói ngắn gọn của API Gateway
	GCP	Google Cloud Platform
	Global	Mạng tính toàn cầu
	Generate	Phát sinh
	GRPC	Google Remote Procedure Calls
H	Host	Máy chủ
	Human	Con người
	Handshake	Bắt tay
I	Instance	Thực thể, thể hiện
	Interface	Giao diện
	IDE	Integrated Development Environment
	Implement / Impl	Cài đặt
	Identifier / ID	Mã định danh
	Insert	Thêm vào
	Internal	Nội bộ, bên trong hệ thống
	Interceptor	Can thiệp
J	JSON	JavaScript Object Notation - là một kiểu định dạng dữ liệu kiểu bản cấu trúc sử dụng các cặp key – value để dữ liệu sử dụng. JSON hỗ trợ các cấu trúc dữ liệu như đối tượng và mảng
	JWT	JSON Web Token
K	Key	Khóa
L	Local	Cục bộ
	Layout	Bố cục trang
	Load-balancing / LB	Cân bằng tải
	Latitude / Lat	Vĩ độ
	Longitude / Lng	Kinh độ
	Library	Thư viện

Alphabet	Thuật ngữ, ký hiệu, Từ viết tắt	Định nghĩa
M	MSSQL	Microsoft SQL Server
	Microservice / μService / μS	Kiến trúc microservice
	Monolithic / Monolith	Nguyên khối
	Monitor	Giám sát
	Mapping	Chuyển đổi / Ánh xạ
	Mobile	Di động
	Mobile App	Ứng dụng di động
	Multithreads	Đa luồng
	MPA	Multi Page Application <⇒ Single Page Application Multi-page application
	Message	Tin nhắn
	Message Queue / MQ	Hàng đợi tin nhắn
	Milestone	Cột mốc
	Module	Thành phần
N	NGINX	Một web server mã nguồn mở có cấu trúc bất đồng bộ (asynchronous) và hướng sự kiện (event-driven). NGINX thường được dùng để deploy một website
	Negative User	Người dùng có những tác động tiêu cực, gây hại hệ thống
O	ORM	Ánh xạ quan hệ đối tượng (Object Relational Mapping)
	OOP	Lập trình hướng đối tượng (Object Oriented Programming)
	Open-source	Mã nguồn mở
	Optimize	Tối ưu hóa
P	Parameter / Param	Tham số
	Pagination	Phân trang
	Parking	Đậu/ Đỗ xe
	Protocol	Giao thức
	Protobuf	(Google) Protocol Buffer
	Protected	Được bảo vệ
	Prototype	Nguyên mẫu
	Public	Công khai
	Private	Riêng tư
	Plain text	Văn bản thô (không được bảo mật bằng SSL/TLS)

Alphabet	Thuật ngữ, ký hiệu, Từ viết tắt	Định nghĩa
	Plugin	Bổ sung
	Publish	Xuất bản
Q	Query	Truy vấn
	QR Code	Mã đọc nhanh
	Queue	Hàng đợi
R	Random	Ngẫu nhiên / Cơ chế phát sinh ngẫu nhiên
	Round-robin	Cơ chế phát sinh Round-robin
	RDB	Relational Database – Cơ sở dữ liệu quan hệ
	RDBMS	Relational Database Management System – Hệ quản trị cơ sở dữ liệu quan hệ
	REST	Representational State Transfer
	Render	Kiến tạo, xuất ra
	Readable	Có thể đọc được
	Reload	Tải lại
	Review	Nhìn lại / Đánh giá
	Role	Vai trò
	Rating	Đánh giá
	Request / Req	Yêu cầu
S	Response / Res	Phản hồi, trả lời
	Stack	Ngăn xếp
	Schema	Lược đồ
	Serialize	Quá trình chuyển đổi từ nhị phân thành đối tượng
	Session	Phiên làm việc
	System	Hệ thống
	Subdomain	Tên miền phụ
	Scanner	Máy quét
	Service	Dịch vụ
	Subscribe	Theo dõi
	SQL	Structured Query Language
	SSL	Secure Sockets Layer
	SPA	Single Page Application <> Multi Page Application
	SSR	Server side render

Alphabet	Thuật ngữ, ký hiệu, Từ viết tắt	Định nghĩa
	SMS	Tin nhắn
	SSMS	SQL Server Management Studio
	Server	Máy chủ
	Side	Phía, khía cạnh
	Spam	Gây phiền
	Strategy / Strategies	Chiến lược / chiến thuật
	Synchronous / Sync	Đồng bộ
T	Test	Kiểm tra, kiểm thử
	Test-case	Trường hợp kiểm tra
	Technology	Công nghệ
	Thread	Luồng
U	UI	User Interface – Giao diện người dùng
	UX	User Experience – Trải nghiệm người dùng
	UUID	Mã ID duy nhất trên toàn cầu
	Use-case	Danh sách các trường hợp, sự kiện tương tác giữa người dùng và hệ thống để hoàn thành mục tiêu đề ra.
	User	Người dùng
	Unique	Duy nhất
	Upload	Tải lên
	Universal	Mang tính vũ trụ
V	Unidirectional communication	Giao tiếp 1 chiều
	Virtual Machine / VM	Máy ảo
	Value	Giá trị
W	Version	Phiên bản
	Website / Web	Trang mạng
X	Workflow	Luồng làm việc
	XML	Extensible Markup Language - ngôn ngữ đánh dấu mở rộng

Bảng 1. Thuật ngữ, ký hiệu và từ viết tắt

Chương 1 | TỔNG QUAN DỰ ÁN

Chương này giới thiệu tổng quát dự án của nhóm, bao gồm nhiều vấn đề xoay quanh (mục tiêu của dự án này là gì, thực trạng hiện nay có phù hợp để xây dựng dự án hay không, nội dung và kế hoạch thực hiện dự án)

1.1. Phát biểu vấn đề

- Hiện nay, đa số mọi người đều đi làm đi chơi,... bằng xe máy hoặc xe ô tô. Với mật độ dân số ngày càng đông, các tuyến đường ngày càng chật hẹp, sáng sớm đi làm thì kẹt xe, tối đến đi chơi lại quá đông, cho nên việc đỗ xe cũng ảnh hưởng không ít tới vấn đề giao thông, vấn đề thời gian và nhiều vấn đề khác.
- Mặt khác, xe ô tô là một trong những phương tiện di chuyển công khênh và không phổ biến lắm ở nước ta do chi phí mua một chiếc xe ô tô quá đắt so với thu nhập bình quân của người Việt Nam. Vì thế, không phải bãi xe nào trong thành phố đều có nơi giữ xe ô tô cũng như các quán ăn, khu vui chơi,... có mặt bằng trung bình và nhỏ không đủ diện tích để đậu một hay nhiều chiếc xe ô tô cùng một lúc. Chủ sở hữu xe ô tô luôn phải tính toán, tìm hiểu rất kỹ trước địa điểm mà mình sẽ đến có chỗ đỗ xe ô tô hay không, và nếu không có thì chủ xe ô tô phải biết khoảng cách từ địa điểm muôn đến tới bãi đậu xe gần nhất, chưa kể đến các ngoại lệ phát sinh (bãi xe hết chỗ, bãi xe đóng cửa,...). Và thật sự nếu chủ xe không có kinh nghiệm về đường sá Việt Nam cũng như không có các ứng dụng công nghệ cao hỗ trợ thì chỉ từ một việc đơn giản là đi ăn uống, hoặc đi chơi cùng với gia đình trên con xe ô tô mới tậu được đã trở nên khó khăn và khó chịu đến đường nào

➔ Năm bắt được xu thế cũng nhu cầu của các chủ sở hữu xe ô tô hiện nay, nhóm em quyết định xây dựng một ứng dụng có thể giải quyết được những vấn đề khó khăn nêu trên cũng như giúp các chủ bãi đỗ xe có thể tăng thêm lượng khách đậu bãi, cải thiện thu nhập bình quân của họ

1.2. Mục tiêu của dự án

- Phát triển được một ứng dụng giúp người dùng có thể tìm được bãi đỗ xe còn trống gần nhất hoặc ở một địa điểm nào đó.
- Người dùng có thể đặt trước để giữ chỗ đỗ xe trong khoảng thời gian dự trù, tránh được việc hết chỗ giữ nhất là vào giờ cao điểm.
- Ứng dụng có tích hợp tính năng tính khoảng cách, chỉ đường từ vị trí hiện tại đến bãi đậu xe và từ bãi đậu xe đến địa điểm mà người dùng muốn đến
- Ứng dụng được tạo ra nhằm giúp người dùng tiết kiệm thời gian thay vì phải tự đi tìm chỗ để giữ xe.

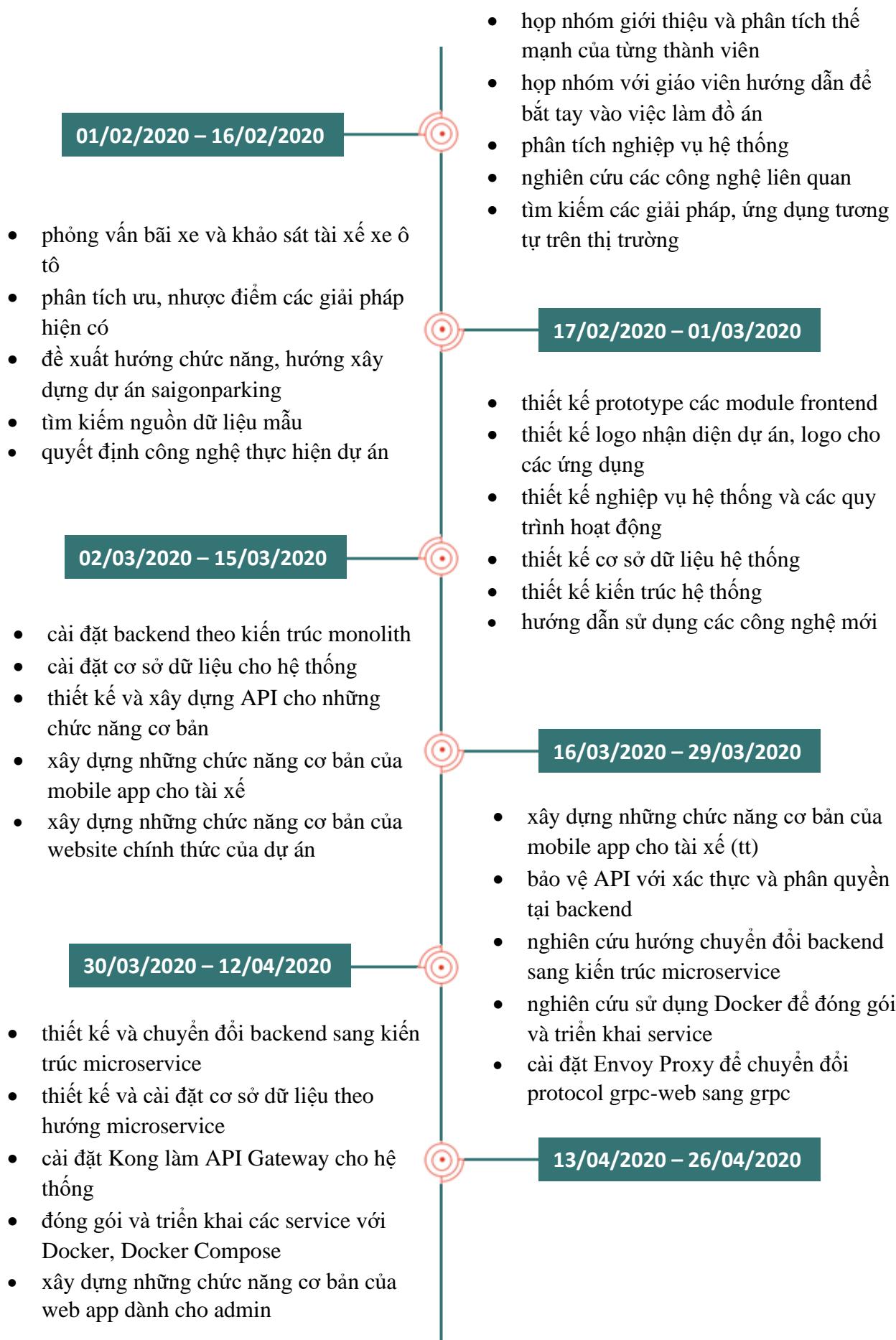
1.3. Nội dung của dự án

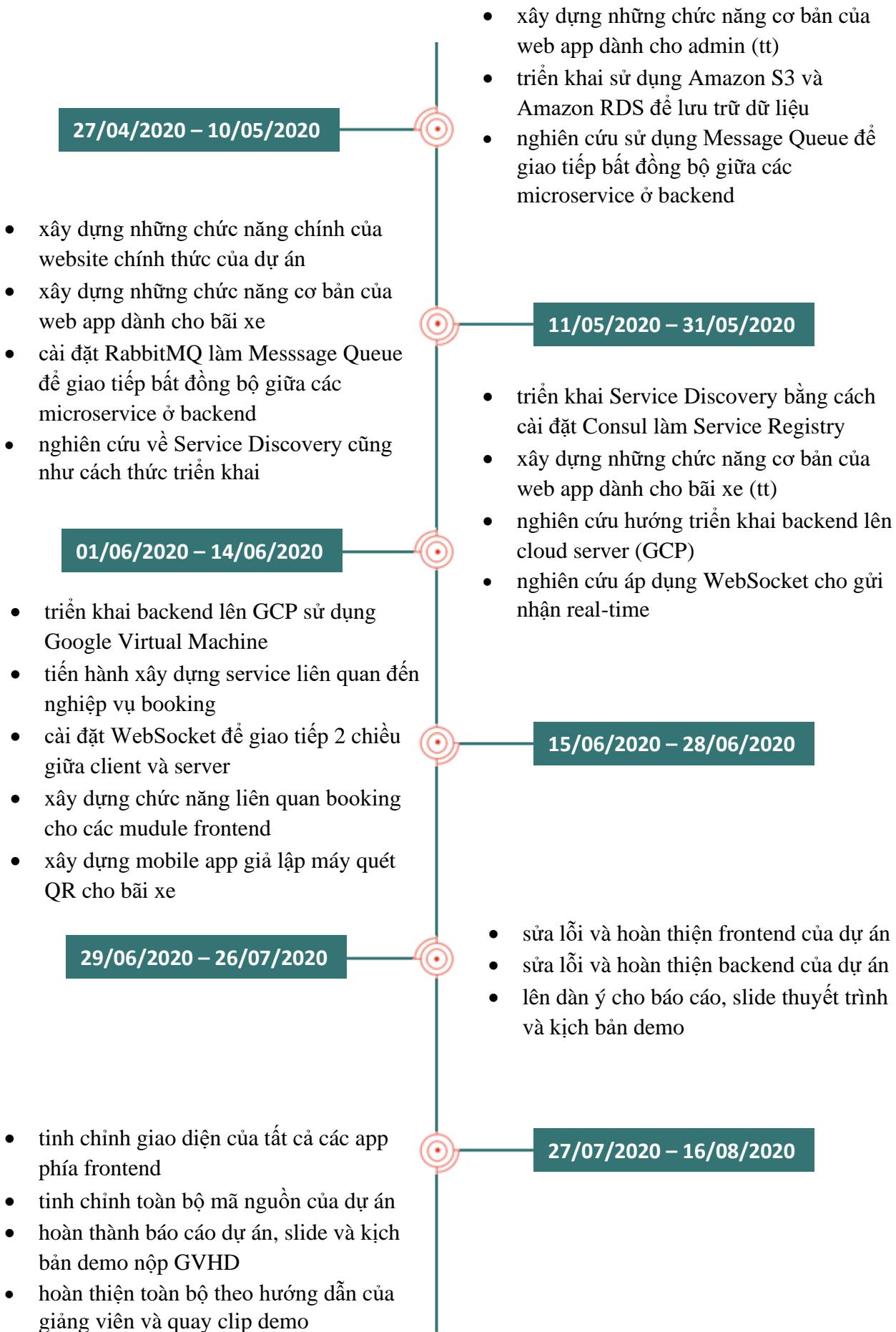
- Xây dựng ứng dụng tìm kiếm bãi xe và đặt trước chỗ đậu xe với các tính năng đi kèm như đã đề cập trong đề mục 1.2 của báo cáo.

- Ứng dụng được cài đặt trên các môi trường:

- 1 app mobile chạy trên hệ điều hành Android giúp khách hàng tương tác và sử dụng các chức năng cần thiết cho việc tìm và đặt trước chỗ đậu xe ô tô
- 1 website giả lập ứng dụng bên phía bãi xe hỗ trợ nhận yêu cầu đặt chỗ bên phía app khách hàng, quản lý lượng yêu cầu đặt chỗ tùy theo hiện trạng bãi xe (chấp nhận/từ chối yêu cầu đặt chỗ, cập nhật số chỗ trong bãi xe), kiểm tra tính đúng đắn của khách hàng (có phải xe ô tô đó là xe đã đặt chỗ trước trên hệ thống), kết thúc quy trình đặt chỗ sau khi kiểm tra thành công và cho xe vào bãi đỗ
- 1 website admin quản lý tài khoản khách hàng, tài khoản bãi xe với các chức năng cơ bản: thêm, xóa, sửa đổi tượng (bãi xe, khách hàng), cập nhật đổi tượng mới.
- 1 website giới thiệu về ứng dụng, các chức năng chính và hiển thị các bãi đậu xe ô tô trong thành phố Hồ Chí Minh
- 1 app giả lập kiểm tra mã QR Code được gửi về app khách hàng có đúng với mã QR mà Server đã gửi hay không (mã QR được hiểu như là một Booking ID nhằm kiểm tra xe vào bãi là xe đã đặt chỗ trước thông qua ứng dụng)

1.4. Kế hoạch thực hiện





Chương 2 KHẢO SÁT HIỆN TRẠNG

Nội dung chương 2 mô tả lại quá trình khảo sát nhu cầu thực tế của khách hàng và những người có liên quan đến nghiệp vụ hệ thống

2.1. Khảo sát nhu cầu thực tế

2.1.1. Nhu cầu của chủ xe ô tô

Thông qua khảo sát online các cá nhân hiện đang sở hữu và sử dụng xe ô tô thường xuyên (form khảo sát nhóm tự chuẩn bị), nhóm em đúc kết được những thông tin sau:

- Khách hàng tiềm năng nằm trong độ tuổi từ 18-50
 - Hơn 70% trong số họ có sử dụng bãi đỗ xe ô tô thường xuyên
 - Thời gian khách hàng đậu xe khá đa dạng, dao động từ 1-6h (chiếm khoảng 70%), 30% còn lại thường giữ xe qua đêm
 - Khách hàng thường tìm bãi đậu xe thông qua kinh nghiệm bản thân và tìm bãi đậu gần với vị trí hiện tại của mình nhất
 - 91% khách hàng thanh toán chi phí đậu xe bằng tiền mặt trực tiếp
 - Hơn 58% khách hàng có nhu cầu đặt trước chỗ đậu xe, 42% còn lại không có nhu cầu đó
 - Có rất nhiều khó khăn trong việc tìm kiếm chỗ đậu xe ô tô, thông thường là không biết bãi đậu xe gần vị trí hiện tại, đến bãi đậu xe nhưng đã hết chỗ, ...
 - 90% người tham gia khảo sát mong muốn có một ứng dụng di động có thể đề xuất chỗ đậu xe gần nhất, cho biết số lượng chỗ còn trống cũng như có thể đặt trước chỗ đậu xe
- ➔ Đồ án được triển khai đúng theo nhu cầu thực tế của các cá nhân đang sở hữu xe ô tô, có ứng dụng thực tiễn cao

2.1.2. Nhu cầu của bãi xe

Thông qua hình thức phỏng vấn trực tiếp nhân viên bãi xe tòa nhà Vina Giày, nhóm em thu thập được những thông tin sau:

- Các bãi xe tư nhân và bãi xe tòa nhà nhỏ có lượng xe ô tô ra vào bãi khá ít. Thu nhập chủ yếu của bãi xe là thông qua giữ xe máy phổ thông
 - Khi được đề xuất một ứng dụng dành cho khách hàng có thể tìm kiếm và đặt trước chỗ đậu xe trong bãi, đa số nhân viên bãi xe đều tán thành vì việc đó góp phần cải thiện số lượng xe ô tô vào bãi và cải thiện lợi nhuận của bãi xe
- ➔ Đồ án được triển khai đúng theo nhu cầu thực tế hiện nay, có ứng dụng thực tiễn cao

2.2. Khảo sát nghiệp vụ của bãi xe

2.2.1 Nghiệp vụ kiểm tra xe ô tô ra/vào bãi

B1: Nhân viên bãi xe sẽ dùng phần mềm quản lý của bãi xe kiểm tra xe ô tô vào

B2: Nhân viên bãi xe sẽ hướng dẫn chủ xe điều khiển phương tiện hướng biển số và khuôn mặt vào đúng tầm quét của camera

B3: Nhân viên sẽ lấy thẻ xe quét vào khe scan ID (mỗi thẻ giữ xe sẽ có ID riêng biệt). Lúc này trên hệ thống sẽ lưu lại ID thẻ xe, biển số xe, nhận dạng khuôn mặt, ngày giờ xe vào bãi. Và thẻ xe vừa quét sẽ được đưa cho chủ phương tiện

B4: Khi chủ phương tiện lấy xe ra khỏi bãi, nhân viên sẽ lấy thẻ xe đã cấp cho chủ xe và tiến hành quét thẻ. Nếu các thông số (ID thẻ xe, biển số xe, khuôn mặt) trùng với thông số ban đầu, hệ thống sẽ thông báo cho xe ra và kết thúc nghiệp vụ. Mọi thông tin về ngày giờ ra vào bãi, biển số xe,... sẽ được lưu lại trong hệ thống

2.3. Khảo sát ứng dụng tương tự

2.3.1. Ứng dụng iParking của CIS

- Ứng dụng tìm bãi đỗ xe thông minh được phát triển bởi công ty CIS và được Sở Giao Thông Vận Tải Hà Nội chọn làm ứng dụng thí điểm để giải quyết bài toán đỗ xe, dừng xe trái phép trên những tuyến đường quan trọng tại thủ đô

- Tìm kiếm bãi đỗ nhanh chóng
- Ứng dụng nhận về nhiều phản hồi khá tốt từ phía người dùng
- Giao diện, các tính năng đơn giản dễ sử dụng
- Chưa kiểm soát được quy trình thanh toán phí giữ xe. Thay vì nhập mã của điểm đỗ và thanh toán tiền phí trông giữ xe qua ứng dụng iParking, người gửi xe vẫn thanh toán bằng hình thức rất thủ công là trả tiền trực tiếp cho đơn vị bãi xe

2.3.2. Ứng dụng MyParking của Viettel

- Ứng dụng đỗ xe thông minh được xây dựng và phát triển bởi tập đoàn viễn thông quân đội Viettel – một trong những tập đoàn có lịch sử lâu đời và nền móng vững chắc tại Việt Nam

- Ứng dụng được hỗ trợ đầu tư rất lớn về công nghệ và tiền bạc
- Các phân hệ quản lý và ứng dụng cho khách hàng được trình bày chi tiết, rõ ràng trên website chính thức của ứng dụng
- Ứng dụng chưa giải quyết triệt để về nghiệp vụ ngoài mang chiều hướng tiêu cực: nhân viên bãi xe thu “tiền tươi” thay vì hướng dẫn tài xế cài đặt ứng dụng My Parking để trả tiền phí đậu xe

2.3.3. Ứng dụng PakMe của PakMe

- Ứng dụng Pakme chính thức hoạt động từ ngày 10/1/2017. Việc tải, đăng ký và sử dụng ứng dụng này khá đơn giản. Hầu hết người dùng sở hữu điện thoại thông minh đều có thể sử dụng dễ dàng, chỉ mất khoảng 7 phút cho tất cả các bước, tải, đăng ký và bắt đầu sử dụng, giao diện tương đối đơn giản, trực quan, dễ dùng.

2.3.4. Phân tích ưu, nhược điểm của các ứng dụng

- Ứng dụng iParking

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> - Giao diện đẹp, tường minh dễ sử dụng - Các marker bãi xe được đặt rõ ràng trên map của ứng dụng - Có mở rộng thêm chức năng tìm kiếm các địa điểm khác có liên đới tới xe ô tô, vận tải (trạm xăng, chỗ vá lốp, chỗ rửa xe,...) 	<ul style="list-style-type: none"> ❖ Chức năng chỉ đường tới bãi xe phải liên kết với google map navigation ❖ Chưa có tính năng hiển thị danh sách các bãi xe gần đây ❖ Chưa có tính năng đề xuất các bãi xe mới ❖ Chưa có tính năng đếm ngược cảnh báo cho khách khi sắp hết giờ đỗ xe

Bảng 2.3.4.a. Phân tích ưu, nhược điểm của ứng dụng iParking

- **Ứng dụng MyParking**

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> ❖ Cho phép sử dụng chức năng tìm kiếm bãi khi không đăng nhập ❖ Cho phép booking ở hầu hết các thành phố lớn ❖ Hỗ trợ thanh toán qua ViettelPay, SMS 	<ul style="list-style-type: none"> ❖ Giao diện rườm rà, gây khó chịu cho người sử dụng ❖ Chưa đặt các marker bãi xe trên app, gây khó khăn cho khách hàng trong việc tìm kiếm bãi xe ❖ Chế độ auto complete trên thanh search địa điểm còn nhiều bất cập (không auto complete được địa điểm cần đến, không xóa được lịch sử search khi người dùng bấm nút Back hoặc bấm kí tự delete)

Bảng 2.3.4.b. Phân tích ưu, nhược điểm của ứng dụng MyParking

- **Ứng dụng PakMe**

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> ❖ Ứng dụng đặt chỗ đỗ xe ô tô đầu tiên và lớn nhất Việt Nam ❖ Lái xe theo điều hướng trực tiếp từ trợ lý ảo trong ứng dụng Pakme ❖ Thanh toán tự động qua tài khoản trả trước 	<ul style="list-style-type: none"> ❖ Ứng dụng đã ngừng khởi động Server tại Việt Nam ❖ Đánh giá của người dùng trên CH Play không thực sự tốt ❖ Ứng dụng không cung cấp được tình trạng còn hay hết chỗ của bãi đỗ xe, điều này dễ khiến người dùng tốn công di chuyển đến bãi đỗ.

Bảng 2.3.4.c. Phân tích ưu, nhược điểm của ứng dụng PakMe

2.4. Danh sách yêu cầu

2.4.1. Yêu cầu chức năng

#	Yêu cầu
Về phía khách hàng	
1	Gọi ý bãi giữ xe xung quanh vị trí hiện tại / vị trí muốn đến
2	Chỉ đường từ vị trí hiện tại đến bãi giữ xe / đến vị trí muốn đến
3	Xem thông tin chi tiết của bãi xe
4	Xem đánh giá, đánh giá bãi giữ xe
5	Đặt trước chỗ đậu xe
6	Xem lịch sử đặt chỗ
Về phía bãi xe	
7	Cập nhật số lượng chỗ trong bãi
8	Gửi thông báo đến server khi bãi xe đã hết chỗ
9	Quét kiểm tra mã QR đặt chỗ của khách hàng
Về phía quản trị viên	
10	Quản lý thông tin khách hàng: thêm, xóa, sửa khách hàng
11	Quản lý thông tin bãi xe: thêm, xóa, sửa bãi xe

Bảng 2.4.1. Danh sách các yêu cầu chức năng

2.4.2. Yêu cầu phi chức năng

#	Yêu cầu
1	Ứng dụng có độ tin cậy cao
2	Ứng dụng có độ bảo mật cao
3	Giao diện thân thiện, dễ sử dụng
4	Hiệu suất nhanh chóng, mượt mà
5	Bảo trì, cập nhật ứng dụng dễ dàng
6	Khả năng mở rộng cao

Bảng 2.4.2. Danh sách yêu cầu phi chức năng

2.5. Giải pháp đề xuất

2.5.1. Tìm kiếm và gợi ý bãi xe quanh vị trí cố định, kết hợp lọc kết quả tìm theo khoảng cách, giá cả

- Ứng dụng sẽ đề xuất bãi xe đã được sắp xếp theo khoảng cách, giá cả quanh vị trí cố định của người dùng hoặc quanh vị trí mà người dùng bấm tìm kiếm trên ứng dụng.

2.5.2. Cho phép đặt trước chỗ đậu xe thông qua ứng dụng và xem lại lịch sử đỗ xe

- Sau khi tìm được bãi xe phù hợp, người dùng có thể đặt trước chỗ đậu xe một cách nhanh chóng. Toàn bộ lịch sử đỗ xe sẽ được hệ thống lưu lại và người dùng có thể xem lại bất kỳ lúc nào.

2.5.3. Cho phép tương tác với bãi xe thông qua ứng dụng: gửi/nhận tin nhắn văn bản

- Trong quá trình đặt trước chỗ đậu xe, người dùng có thể chat với bãi xe để biết thêm về thông tin chi tiết bãi đỗ, giá cả, tổng quan bãi xe,... và bãi xe cũng có thể gửi lại tin nhắn để giải đáp những câu hỏi của người dùng.

2.5.4. Cho phép đánh giá, xem đánh giá bãi xe thông qua ứng dụng

- Người dùng có thể xem đánh giá bãi xe của người dùng khác và có thể tự mình đánh giá bãi xe đó. Điều này giúp cho những người sử dụng ứng dụng có thể tham khảo trước chất lượng và có cái nhìn khách quan nhất về bãi xe mà họ dự định sẽ đến.

2.5.5. Chỉ đường/ điều hướng đến bãi xe / địa điểm thứ ba trực tiếp trên ứng dụng

- Sau khi chọn được bãi xe phù hợp, người dùng có thể sử dụng chức năng chỉ đường đến bãi xe và đến địa điểm thứ ba trực tiếp trên ứng dụng mà không cần chuyển đến chức năng chỉ đường của Google Map.

2.5.6. Chỉ đường/ điều hướng đến bãi xe / địa điểm thứ ba thông qua ứng dụng Google Map

- Nếu người dùng thích sử dụng chức năng chỉ đường tới bãi xe tích hợp điều hướng thông minh của Google Map hơn, ứng dụng cũng có thể đáp ứng được yêu cầu đó.

2.5.7. Gợi ý các địa điểm hỗ trợ liên quan cho tài xế xe ô tô (trạm xăng, trạm dừng, ...)

- Ứng dụng tích hợp thêm chức năng gợi ý các địa điểm liên quan hỗ trợ cho phuơng tiện của họ (trạm xăng, vá lốp xe, sửa xe, gọi cứu hộ,...) nhằm kiểm soát các trường hợp không mong muốn khi lưu thông xe ô tô trên đường đến bãi đỗ.

2.5.8. Đơn giản hóa việc hoàn tất quá trình đặt chỗ cho cả đôi bên bằng phương pháp scan mã QR

- Bãi xe và người sử dụng dịch vụ đặt chỗ có thể nhanh chóng kết thúc quy trình bằng cách scan mã QR được gửi vào điện thoại của người dùng sau khi đặt chỗ thành công.

2.5.9. Cân bằng tải để hệ thống có khả năng cùng lúc xử lý số lượng lớn yêu cầu từ người dùng

- Hệ thống máy chủ sử dụng cân bằng tải, đảm bảo có thể cùng lúc xử lý lượng lớn yêu cầu gửi về từ người dùng, đồng thời SSL cũng được sử dụng để bảo mật dữ liệu giao tiếp 2 chiều giữa người dùng và hệ thống.

Chương 3 THIẾT KẾ HỆ THỐNG

Chương này nói về thiết kế của hệ thống saigonparking, bao gồm:

- *kiến trúc hệ thống*
- *mô hình hoạt động hệ thống*
- *cơ sở dữ liệu hệ thống*
- *nghiệp vụ hệ thống*
- *danh sách các module và chức năng tương ứng*

3.1. Tổng quan kiến trúc microservice

3.1.1. Kiến trúc monolithic là gì

Monolithic là từ dùng để diễn tả một mẫu nguyên khối bao gồm cơ sở dữ liệu, giao diện người dùng phía client và ứng dụng phía máy chủ. Tất cả các bộ phận của phần mềm được hợp nhất và tất cả các chức năng của phần mềm được quản lý ở một nơi

3.1.2. Kiến trúc microservice là gì

Microservice là một loại kiến trúc phần mềm hướng dịch vụ, tập trung vào việc xây dựng một loạt các thành phần tự quản lý tạo nên ứng dụng. Không giống như các ứng dụng nguyên khối được xây dựng dưới dạng một đơn vị không thể chia tách, các ứng dụng microservice bao gồm nhiều thành phần độc lập output ra các API

3.1.3. So sánh kiến trúc monolithic và kiến trúc microservice

	Monolithic	Microservice
Ưu điểm	<ul style="list-style-type: none"> - Phát triển và triển khai đơn giản - Hiệu suất tốt hơn 	<ul style="list-style-type: none"> - Các dịch vụ nhỏ có thể được xây dựng, thử nghiệm và triển khai độc lập - Dễ dàng phát triển, thử nghiệm và triển khai - Tăng hiệu suất - Hiệu suất tăng cho phép các lập trình viên cập nhật các thành phần hệ thống mà không làm giảm ứng dụng - Khả năng mở rộng theo chiều ngang
Nhược điểm	<ul style="list-style-type: none"> - Codebase trở nên cồng kềnh theo thời gian - Khó áp dụng công nghệ mới - Kém linh hoạt 	<ul style="list-style-type: none"> - Phức tạp - Tính nhất quán dữ liệu và quản lý giao dịch trở nên khó khăn hơn vì mỗi service có cơ sở dữ liệu - Nghẽn cổ chai

Bảng 3.1.3. So sánh kiến trúc monolithic và kiến trúc microservice

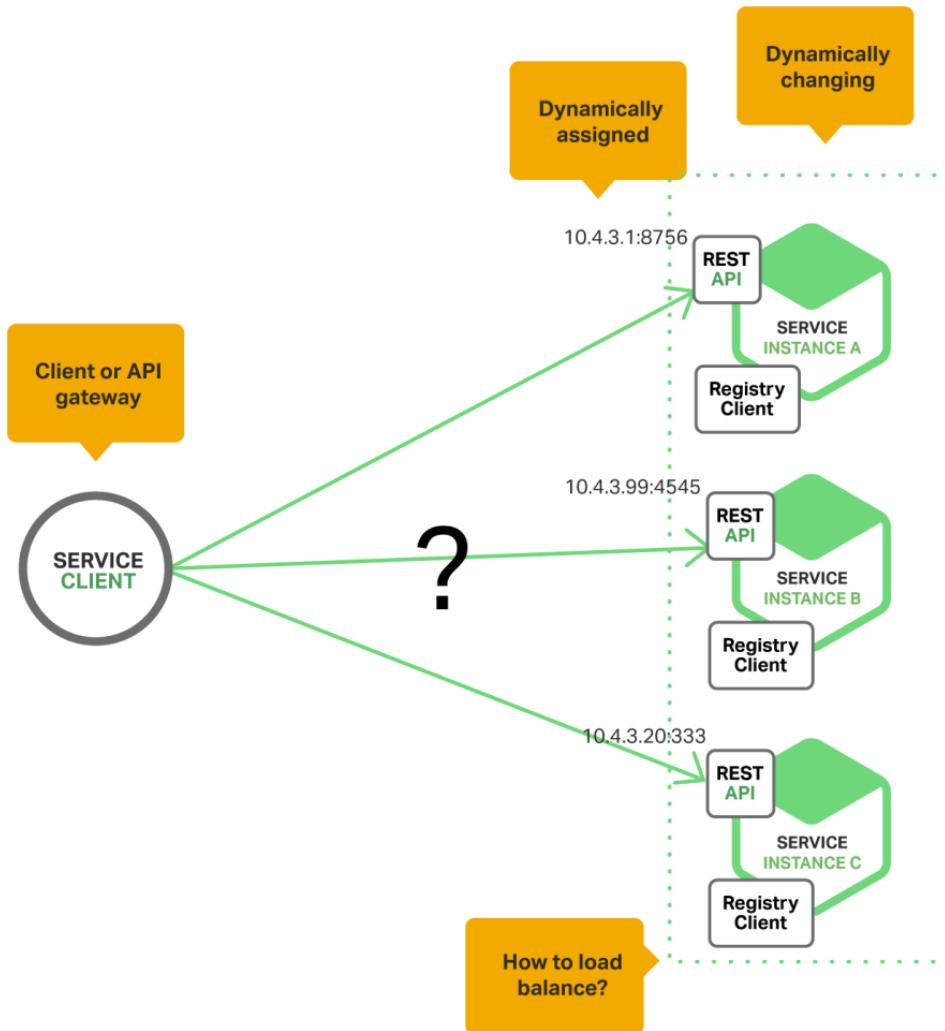
3.1.4. API gateway là gì

API Gateway có thể coi là một cổng trung gian, nó là cổng vào duy nhất tới hệ thống microservices

API gateway sẽ nhận các requests từ phía client, chỉnh sửa, xác thực và điều hướng chúng đến các API cụ thể trên các services phía sau

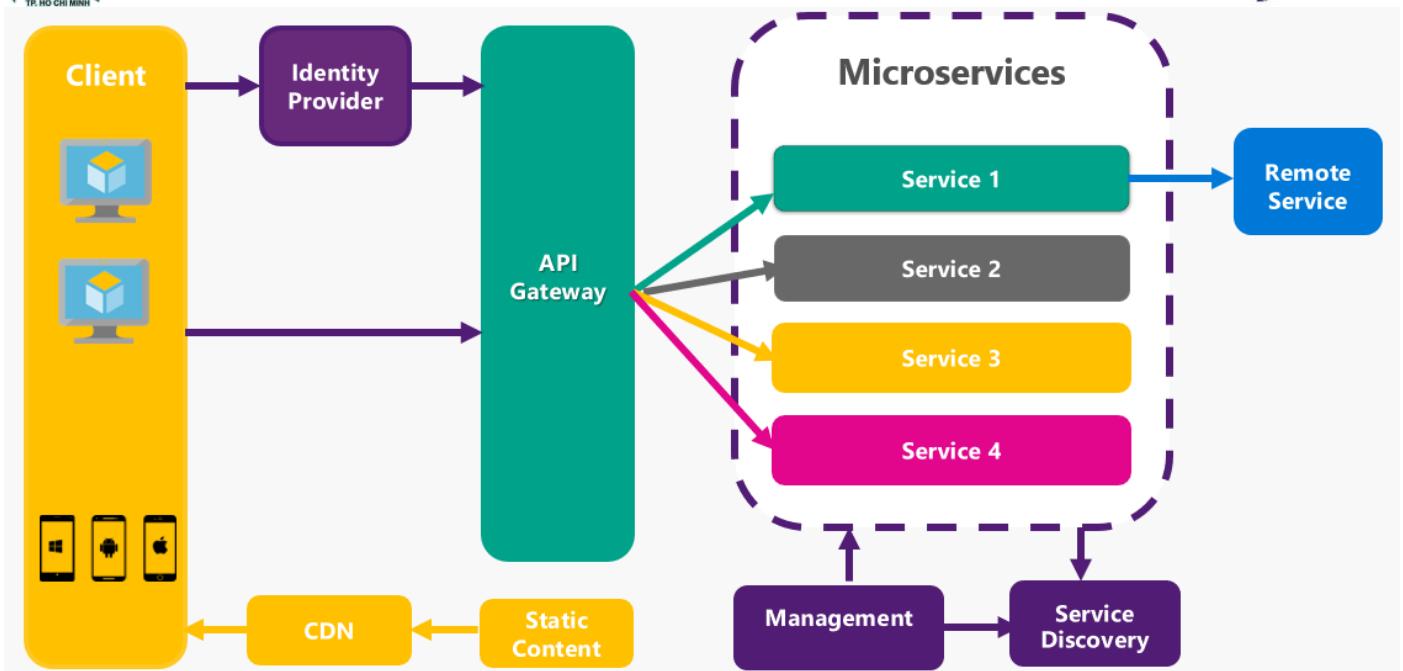
3.1.5. Service Discovery là gì

3.1.5.1. Khái niệm về truy tìm dịch vụ



Hình 3.1.5.1.a. Minh họa bài toán truy tìm dịch vụ và cân bằng tải với kiến trúc microservice

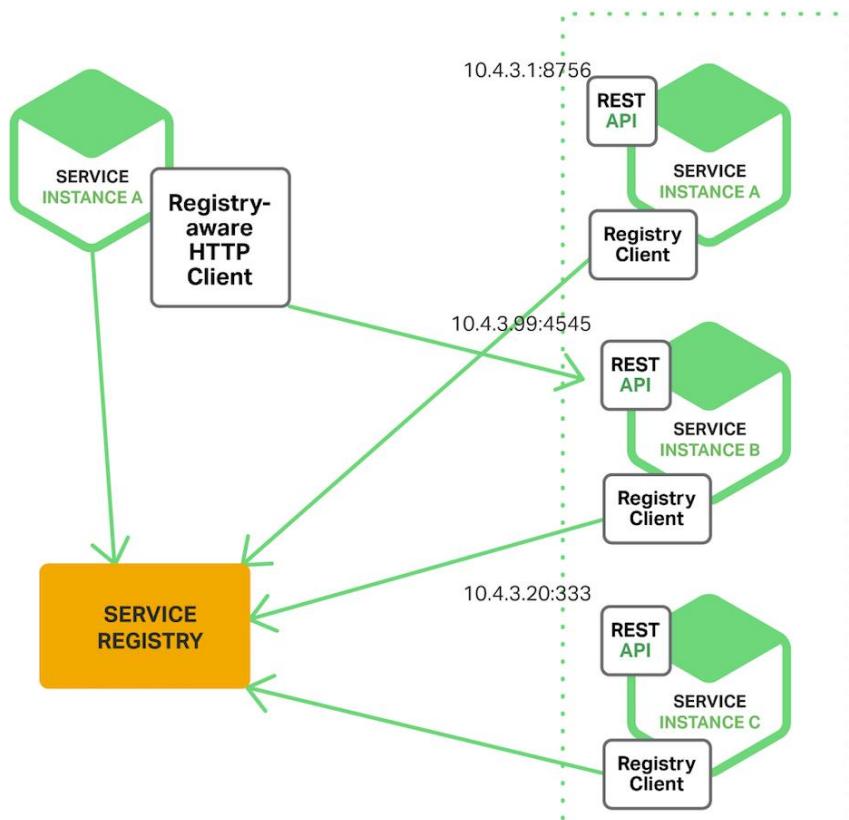
Truy tìm dịch vụ hay Service Discovery là việc tìm địa chỉ IP/port của một service trong hệ thống. Việc tương chừng như dễ dàng này lại trở nên phức tạp khi trong hệ thống có nhiều service (mô hình microservice). Đặc biệt khi chạy trên cloud (AWS, Azure, GCP...), IP của các service thay đổi thường xuyên hơn do việc tạo/xóa máy ảo (hay container) là chuyện thường ngày.



Hình 3.1.5.1.b. Minh họa kiến trúc microservice và vai trò của API Gateway, Service Discovery

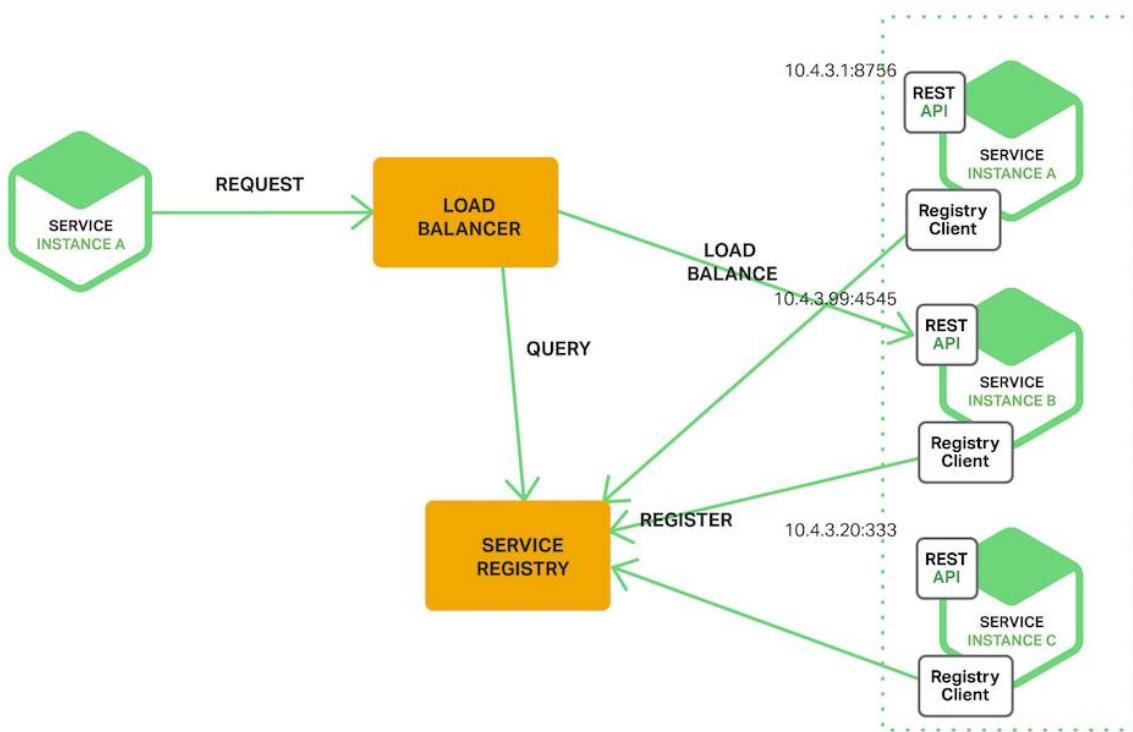
Service discovery có 2 loại thường gặp là Client-side service discovery & Server-side service discovery.

3.1.5.2. Client-side service discovery là gì



Hình 3.1.5.2. Minh họa truy tìm dịch vụ, sử dụng phương pháp Client-side service discovery

3.1.5.3. Server-side service discovery là gì



Hình 3.1.5.3. Minh họa truy tìm dịch vụ, sử dụng phương pháp Server-side service discovery

3.2. Kiến trúc dự án

3.2.1. Các modul của dự án

Dự án saigonparking gồm **6 module**, đó là:

- **1** backend theo hướng kiến trúc microservice
- **5** app frontend (**2** mobile app + **3** web app)

3.2.1.1. Backend

Backend hệ thống saigonparking được thiết kế gồm **2 phần**:

- **Phần phụ trợ:** **1** API Gateway (Kong), **1** Service Discovery (Consul), **1** Message Queue (RabbitMQ)
- **Phần chính:** gồm **6** service, mỗi service là **1** Spring Boot Application, xử lý **1** nhóm nghiệp vụ riêng

3.2.1.1.1. User Service:

User Service sẽ xử lý tất cả nghiệp vụ liên quan đến tài khoản người dùng, bao gồm: thêm/xóa/sửa tài khoản người dùng, tìm kiếm dữ liệu người dùng, kích hoạt/hủy kích hoạt người dùng, thống kê số lượng người dùng,...

Giao tiếp: gRPC, MessageQueue

3.2.1.1.2. Auth Service

Auth Service sẽ xử lý tất cả nghiệp vụ liên quan đến việc xác thực người dùng, bao gồm: đăng nhập tài khoản, đăng ký tài khoản, quên mật khẩu, cấp mới token (access token, refresh token) ...

Giao tiếp: gRPC, MessageQueue

3.2.1.1.3. Mail Service

Mail Service sẽ xử lý tất cả nghiệp vụ liên quan đến việc gửi mail từ hệ thống, bao gồm: gửi mail kèm link kích hoạt tài khoản cho người dùng, gửi mail kèm link đặt lại mật khẩu cho người dùng.

Giao tiếp: MessageQueue

3.2.1.1.4. Parking-lot Service

Parking Lot Service sẽ xử lý tất cả nghiệp vụ liên quan đến bãi xe, bao gồm: thêm/xóa/sửa bãi xe, tìm kiếm bãi xe, gợi ý bãi xe, lấy thông tin chi tiết bãi xe, lấy số lượng chỗ còn trống của bãi xe, thống kê số lượng bãi xe,...

Giao tiếp: gRPC, MessageQueue

3.2.1.1.5. Booking Service

Booking Service sẽ xử lý tất cả nghiệp vụ liên quan đến booking (đặt chỗ), bao gồm: thêm/xóa/sửa booking, thêm/xóa/sửa booking history, lấy dữ liệu chi tiết 1 booking, xem lại lịch sử booking, tìm kiếm booking thỏa điều kiện, thống kê số lượng booking,...

Giao tiếp: gRPC, MessageQueue

3.2.1.1.6. Contact Service

Contact Service sẽ xử lý tất cả nghiệp vụ sử dụng kết nối 2 chiều (WebSocket), bao gồm: kiểm tra người dùng có đang online hay không, kiểm tra bãi xe có đang online hay không, tạo kết nối 2 chiều với máy chủ, gửi nhận tin nhắn 2 chiều giữa client và server, nhắn tin qua lại giữa khách hàng và bãi xe, quét mã QR ,...

Giao tiếp: gRPC, MessageQueue, WebSocket

3.2.1.2. Website dành cho quản trị viên

- Thông kê
- Quản lý bãi xe
- Quản lý người dùng

3.2.1.3. Website chính thức của dự án

- Giới thiệu ứng dụng
- Đăng ký đăng nhập kích hoạt tìm lại mật khẩu
- Tìm kiếm, xem thông bãi xe

3.2.1.4. Website giả lập ứng dụng quản lý bãi giữ xe ô tô

- Quản lý đặt chỗ
- Quản lý xe vào ra
- Giao tiếp với khách hàng

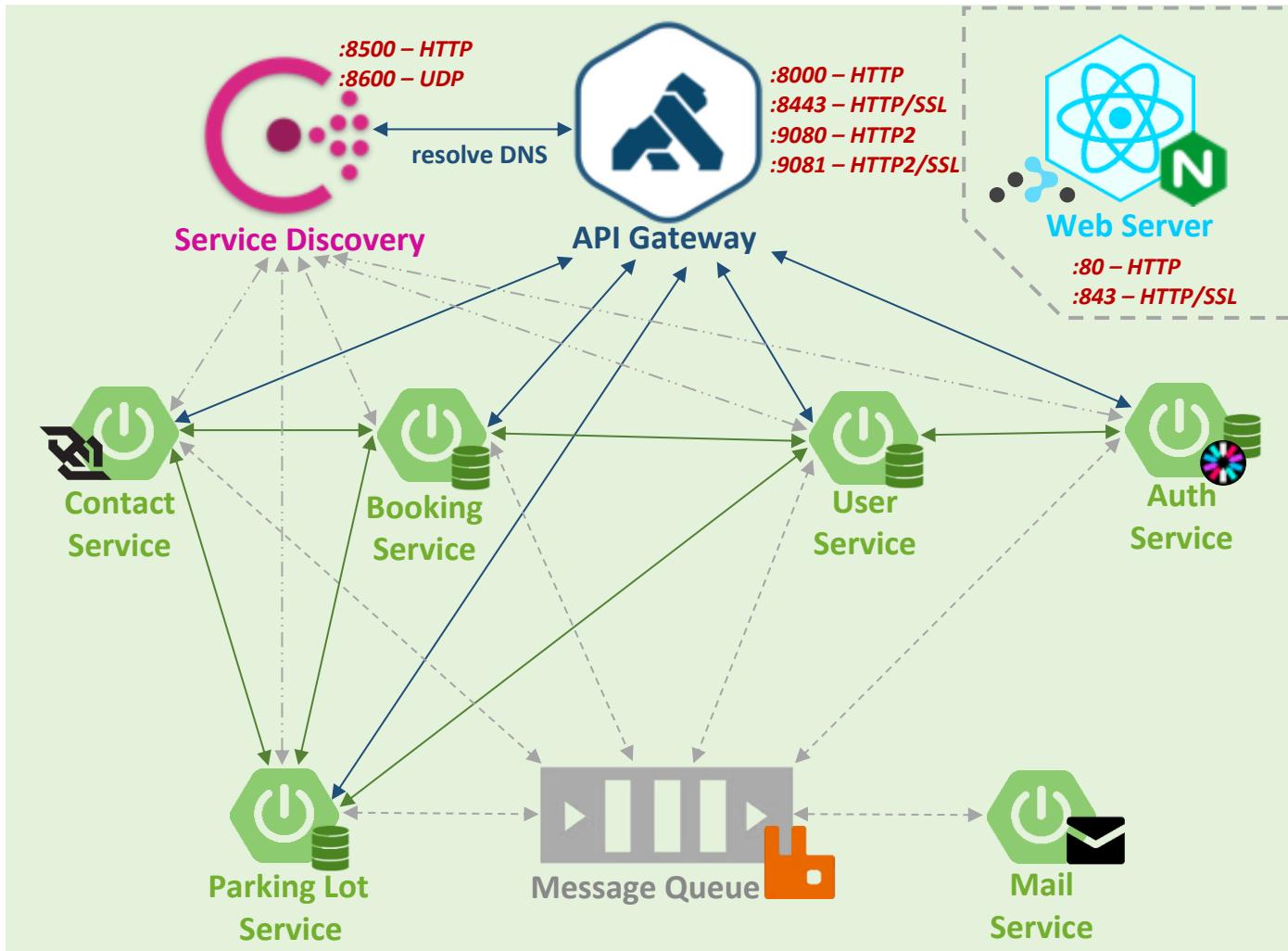
3.2.1.5. Ứng dụng di động dành cho tài xế xe ô tô

- Tìm các bãi đỗ
- Xem thông tin các bãi đỗ
- Đặt chỗ
- Chỉ đường
- Giao tiếp với bãi xe
- Tìm các trạm đỗ xăng, ...

3.2.1.6. Ứng dụng di động giả lập thiết bị scan mã QR tại bãi giữ xe

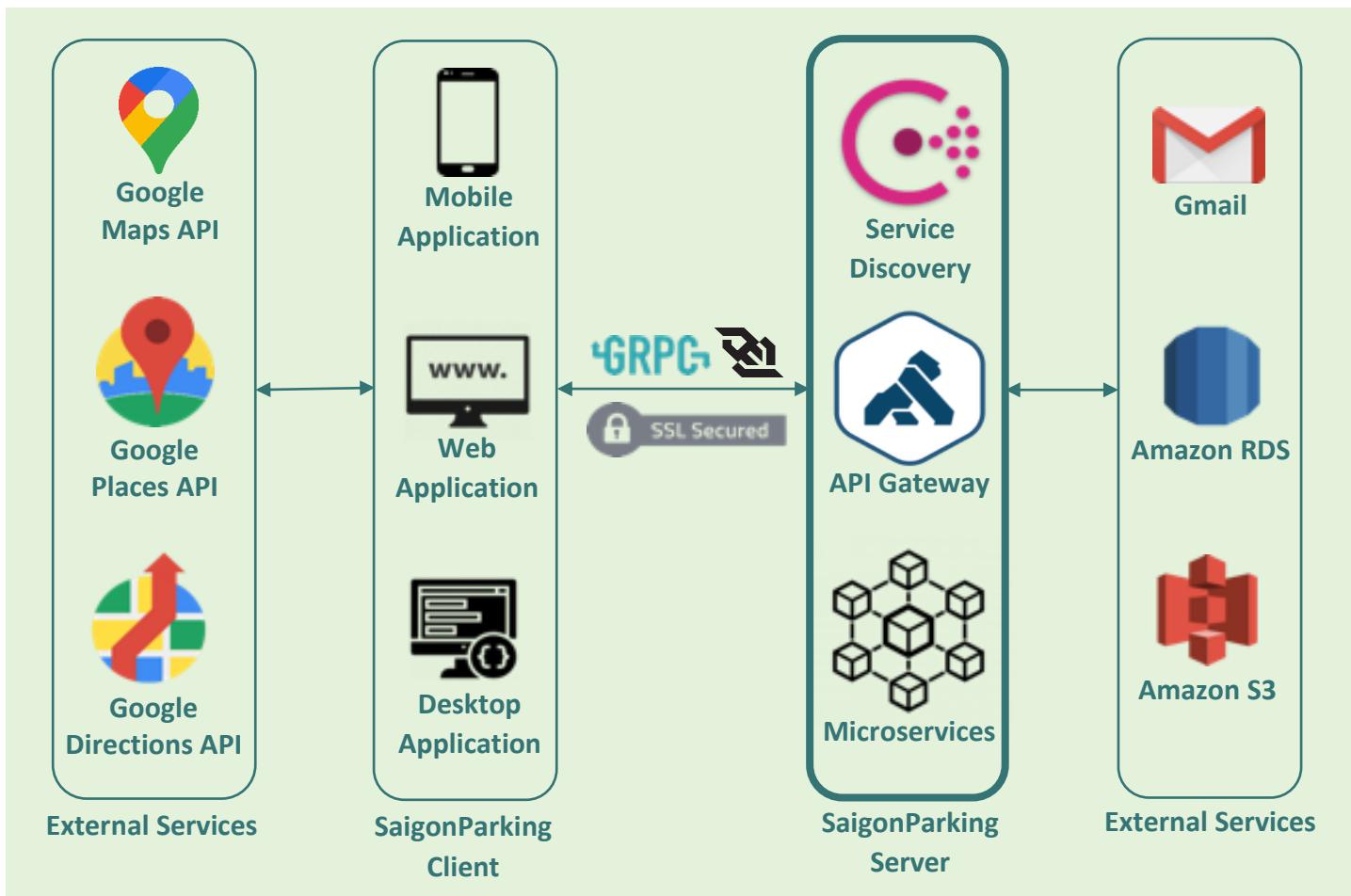
- Hỗ trợ giúp cho bãi xe xác thực việc đặt chỗ

3.2.2. Kiến trúc tổng quan (architecture)



Hình 3.2.2. Kiến trúc tổng quan của hệ thống saigonparking

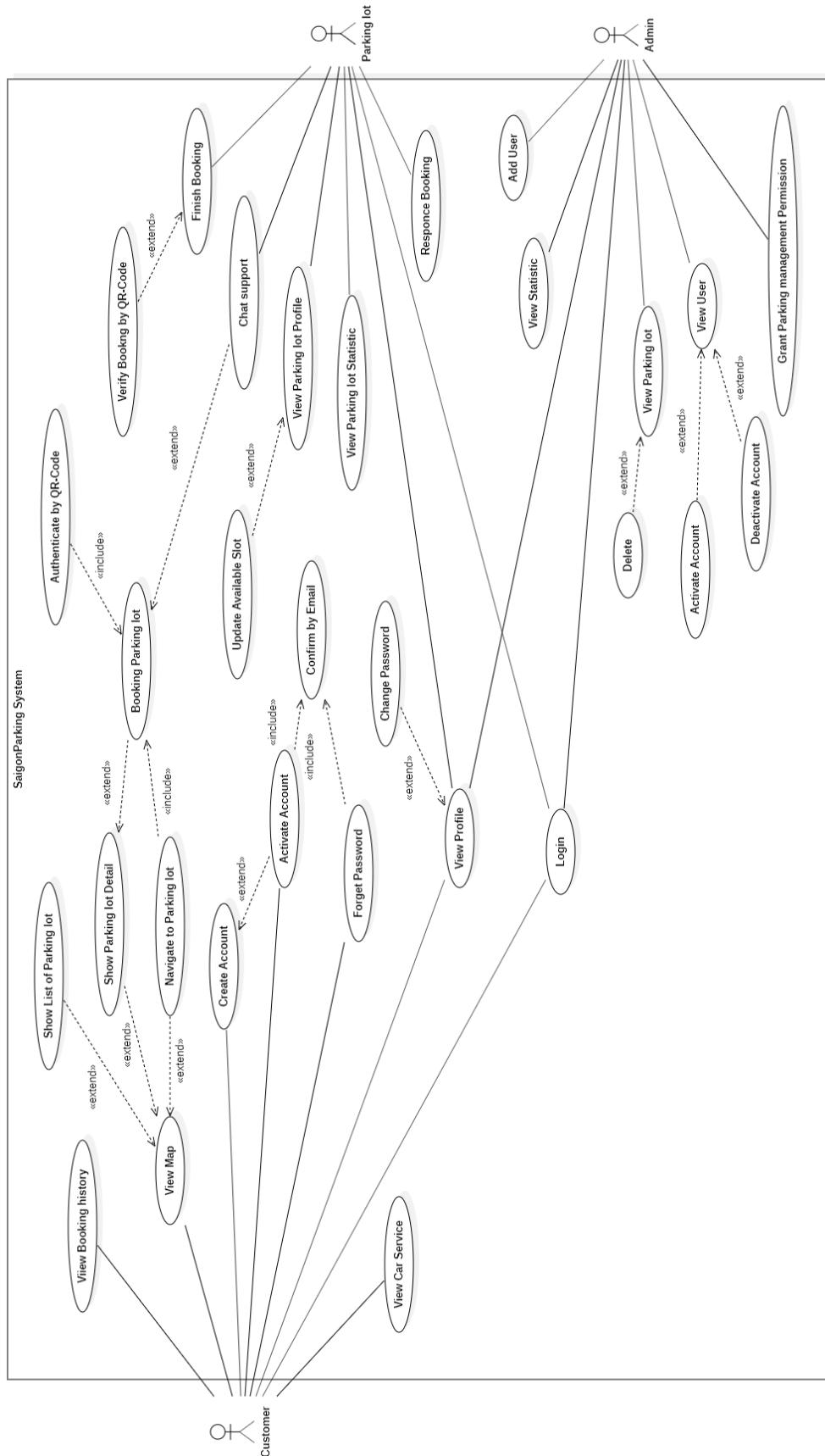
3.3. Mô hình hoạt động (application workflow)



Hình 3.3. Mô hình hoạt động của hệ thống saigonparking

3.4. Mô hình use-case (use-case diagram)

3.4.1. Tổng quan



Hình 3.4.1. Sơ đồ use-case của hệ thống saigonparking

STT	Tên Use-case	Tên Use-case tiếng Việt	Ý nghĩa /Ghi chú
01	Login	Đăng nhập	
02	Logout	Đăng xuất	
03	View Profile	Xem thông tin cá nhân	
04	Change Password	Thay đổi mật khẩu	
05	Create Account	Tạo tài khoản	
06	Activate Account	Kích hoạt tài khoản	
07	Forget Password	Quên mật khẩu	
08	Confirm by Email	Xác thực bằng email	Nhằm hạn chế hiện tượng tạo tài khoản ma làm nặng hệ thống.
09	View Map	Xem bản đồ bãi đỗ xe	Sử dụng bản đồ của Google.
10	Show List of Parking lot	Hiện danh sách các bãi đỗ đang có trên bản đồ	
11	Show Parking lot Detail	Xem thông tin chi tiết của bãi đỗ xe	
12	Navigate to Parking lot	Chỉ đường đến vị trí bãi đỗ xe	Dùng dẫn đường của ứng dụng hay của Google Map điều hướng người dùng đến bãi đỗ xe hoặc vị trí thứ ba.
13	Book Parking lot	Đặt chỗ đỗ xe	Yêu cầu chọn bãi xe đang hoạt động và còn dư chỗ đỗ.
14	Authenticate by QR-Code	Xác thực đặt chỗ bằng QR-code	Chỉ dành cho người dùng, tài xế.
15	Chat support	Chat hỗ trợ khách hàng	Chỉ dành cho người dùng, tài xế.
16	Review Parking lot	Đánh giá bãi xe	Đánh giá chất lượng và thái độ của bãi đỗ xe.
17	View Car Service	Xem các dịch vụ ôtô khác	Chỉ dành cho người dùng. Hiển thị vị trí các dịch vụ gần chỗ đã chọn (trạm xăng, chỗ sửa xe, v.v...)
18	View Booking History	Xem lịch sử đặt chỗ bãi xe	Hiện danh sách đặt chỗ mà người dùng đã thực hiện.
19	View Statistic Parking lot	Xem dữ liệu thống kê của bãi đỗ	Chỉ dành cho quản lý bãi xe. Hiển thị lượng người sử dụng, tình trạng và số lượt đặt chỗ ,v,v...
20	Update available Slot	Cập nhật thông tin bãi đỗ	
21	Response to Booking	Phản hồi lại khách hàng về việc đặt chỗ	Chấp nhận hoặc từ chối booking.
22	Chat support	Tư vấn khách hàng	Chỉ dành cho quản lý bãi xe.
23	Finish Booking	Hoàn tất đặt chỗ đỗ xe	Dùng khi bãi xe không có máy quét mã QR-Code
24	Verify Booking by QR-Code	Xác nhận khách đỗ xe bằng cách quét QR-code	Chỉ dành cho quản lý bãi xe.
25	View Statistic	Xem thống kê	Chỉ dành cho Admin
26	Create User	Tạo tài khoản người dùng	Chỉ dành cho Admin

27	View User	Xem danh sách thông tin người dùng	Tìm kiếm, hiển thị danh sách người dùng
28	Activate Account	Kích hoạt người dùng	
29	Deactivate Account	Vô hiệu hóa người dùng	
30	Grant Parking lot management Premission	Cấp quyền quản lý bãi đỗ xe	Cấp quyền cho người dùng trở thành quản lý bãi đỗ xe.
31	View Parking lot	Xem danh sách thông tin bãi đỗ xe	Hiển thị danh sách bãi xe, có thanh tìm kiếm và bộ lọc,,
32	Delete Parking lot	Xoá bãi đỗ xe	

Bảng 3.4.1. Danh sách các use-case của hệ thống saigonparking

3.4.2. ĐẶC TẢ CHI TIẾT USE-CASE

3.4.2.1. Use-case sử dụng chung

3.4.2.1.1. UC-01: Đăng nhập

Tên của Use Case	ĐĂNG NHẬP
Tóm tắt	Use Case dùng để cấp phép người dùng đăng nhập vào hệ thống.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng khởi động ứng dụng. Hiển thị màn hình đăng nhập gồm có khung điền tên tài khoản, mật khẩu, nút bấm đăng nhập vào hệ thống, thao tác tạo tài khoản, quên mật khẩu và kích hoạt tài khoản. Nhập thông tin đăng nhập, mật khẩu và nhấn nút đăng nhập. Khi người dùng thực hiện xong các thao tác yêu cầu, thông tin sẽ gửi về server thông qua gRPC, hệ thống Auth Service sẽ xác thực tài khoản người dùng có hợp lệ. Hiển thị màn hình chính của ứng dụng. Ở các lần đăng nhập sau, hệ thống tự động thực hiện thay cho người dùng.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Tại bước nhập thông tin đăng nhập. <ul style="list-style-type: none"> A3.1: Nếu người dùng chưa có tài khoản: <ul style="list-style-type: none"> Nhấn thao tác tạo tài khoản, bắt đầu thực hiện UC-05 (Tạo tài khoản) A3.2: Nếu người dùng có tài khoản nhưng chưa kích hoạt

	<ul style="list-style-type: none"> + Nhấn thao tác kích hoạt tài khoản, bắt đầu thực hiện UC-06 (Kích hoạt tài khoản) • A3.2: Nếu người dùng có tài khoản nhưng quên mật khẩu <ul style="list-style-type: none"> + Nhấn thao tác quên mật khẩu, bắt đầu thực hiện UC-07 (Quên mật tài khoản) • A4: Tại bước gửi thông tin đăng nhập về server <ul style="list-style-type: none"> • A4.1: Đăng nhập không thành công. • A4.2: Màn hình hiển thị thông báo lỗi, yêu cầu người dùng đăng nhập lại. • A5: Tại bước hiển thị màn hình chính của ứng dụng. <ul style="list-style-type: none"> - A5.1: Nếu người đăng nhập là tài xế <ul style="list-style-type: none"> + Hiển thị màn hình trang chủ tài xế - A5.2: Nếu người đăng nhập là quản lý bãi đỗ xe <ul style="list-style-type: none"> + Hiển thị màn hình trang chủ quản lý bãi đỗ xe. - A5.3: Nếu người đăng nhập là Admin hệ thống <ul style="list-style-type: none"> + Hiển thị màn hình trang chủ Admin. • Trả về gói tin thông báo đăng nhập thành công, báo lỗi hay đăng nhập thất bại.
Ngữ cảnh	Màn hình đăng nhập vào ứng dụng hay website.
Điều kiện tiên quyết	Người dùng phải mở ứng dụng hay website
Điều kiện kết thúc	<ul style="list-style-type: none"> • Đăng nhập thành công.
Điểm mở rộng	<ul style="list-style-type: none"> • Hệ thống sẽ sử dụng Token để xác thực người dùng ở các lần đăng nhập sau.

3.4.2.1.2. UC-02: Đăng xuất

Tên của Use Case	DĂNG XUẤT
Tóm tắt	Use Case dùng để cấp phép người dùng đăng xuất khỏi tài khoản đang sử dụng.
Dòng sự kiện	<ul style="list-style-type: none"> • Use-case này bắt đầu khi người dùng đang chạy ứng dụng, website và đã đăng nhập tài khoản. • Bấm vào Icon góc trên bên phải, hiển thị một menu có nhiều thao tác để lựa chọn trong đó có thao tác đăng xuất. • Nhấn thao tác đăng xuất.

Dòng sự kiện khác	<ul style="list-style-type: none"> Đăng xuất thành công Thoát khỏi màn hình hiện tại và quay về màn hình đăng nhập của ứng dụng hay website
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Tại bước nhấn thao tác đăng xuất. <ul style="list-style-type: none"> A3.1: Chọn các thao tác khác trong menu. <ul style="list-style-type: none"> Chuyển sang màn hình khác theo thao tác mà người dùng đã lựa chọn. A3.2: Đóng menu và quay về trang hiện hành.
Ngữ cảnh	Menu lựa chọn các thao tác.
Điều kiện tiên quyết	Người dùng phải thực hiện xong UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Ứng dụng sẽ xoá thông tin về token và các thông tin lưu trữ tạm trong ứng dụng.
Điểm mở rộng	Không có

3.4.2.1.3. UC-03: Xem thông tin cá nhân

Tên của Use Case	XEM THÔNG TIN CÁ NHÂN
Tóm tắt	Use Case cho phép người dùng kiểm tra thông tin cá nhân của tài khoản đang sử dụng
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng đang chạy ứng dụng, website và đã đăng nhập tài khoản. Bấm vào Icon góc trên bên phải, hiển thị một menu có nhiều thao tác để lựa chọn trong đó có thao tác xem thông tin cá nhân. Hiển thị màn hình thông tin cá nhân bao gồm các trường thông tin cá nhân, nút đổi mật khẩu và nút thay đổi thông tin cá nhân.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Tại bước hiển thị màn hình thông tin cá nhân <ul style="list-style-type: none"> Người dùng có thể lựa chọn các thao tác khác trên màn hình thông tin cá nhân. Kích hoạt Use-case tương ứng theo thao tác người dùng đã chọn.
Ngữ cảnh	Màn hình xem thông tin cá nhân.
Điều kiện tiên quyết	Người dùng phải thực hiện xong UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị màn hình thông tin cá nhân.

Điểm mở rộng	Không có
--------------	----------

3.4.2.1.4. UC-04: Thay đổi mật khẩu

Tên của Use Case	THAY ĐỔI MẬT KHẨU
Tóm tắt	Use Case cho phép người dùng thay đổi mật khẩu của tài khoản đang sử dụng.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng thực hiện xong UC-03 (Xem thông tin cá nhân). Nhập vô khung mật khẩu mới và xác nhận mật khẩu, nhấn nút đổi mật khẩu. Khi người dùng thực hiện xong các thao tác yêu cầu, thông tin sẽ gửi về server thông qua gRPC, hệ thống Auth Service sẽ cập nhập lại thông tin mật khẩu.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Tại bước gửi gói tin về server. <ul style="list-style-type: none"> Tài khoản đã được kích hoạt. Hiển thị thông báo lỗi, yêu cầu người dùng nhập lại mật khẩu thay đổi.
Ngữ cảnh	Màn hình thay đổi mật khẩu
Điều kiện tiên quyết	Người dùng phải thực hiện UC-03 (Xem thông tin cá nhân)
Điều kiện kết thúc	<ul style="list-style-type: none"> Nhận gói tin thông báo đổi mật khẩu thành công.
Điểm mở rộng	Không có

3.4.2.2. Use-case của khách hàng

3.4.2.2.1. UC-05: Tạo tài khoản

Tên của Use Case	TẠO TÀI KHOẢN
Tóm tắt	Use Case cấp phép người dùng đăng ký tạo tài khoản trên hệ thống

Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng khởi động ứng dụng, website và chọn tạo tài khoản. Hiển thị màn hình đăng ký gồm có khung điền tên tài khoản, mật khẩu và nút bấm đăng ký tài khoản. Khi người dùng thực hiện xong các thao tác yêu cầu, thông tin sẽ gửi về server thông qua gRPC, hệ thống Auth Service sẽ xác thực tài khoản đăng ký của người dùng có hợp lệ.
Dòng sự kiện khác	<ul style="list-style-type: none"> Kích hoạt UC-06 (Kích hoạt tài khoản).
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Tại bước gửi gói tin về server. <ul style="list-style-type: none"> Tài khoản đã tồn tại. Hiển thị thông báo lỗi, yêu cầu người dùng nhập lại thông tin tạo tài khoản.
Ngữ cảnh	Màn hình đăng ký tài khoản.
Điều kiện tiên quyết	Người dùng phải đang chạy ứng dụng, website
Điều kiện kết thúc	<ul style="list-style-type: none"> Nhận gói tin thông báo đăng ký tài khoản của người dùng thành công.
Điểm mở rộng	Không có

3.4.2.2.2. UC-06: Kích hoạt tài khoản

Tên của Use Case	KÍCH HOẠT TÀI KHOẢN
Tóm tắt	Use Case dùng để kích hoạt tài khoản người dùng mục đích hạn chế việc tạo các tài khoản ảo làm ảnh hưởng hệ thống.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng mở màn hình kích hoạt tài khoản và đã tạo ít nhất một tài khoản. Nhập tên tài khoản và nhấn nút kích hoạt. Khi người dùng thực hiện xong các thao tác yêu cầu, gói tin sẽ gửi về server thông qua gRPC, hệ thống Auth Service sẽ xác thực gói tin có hợp lệ và gửi email để người dùng kích hoạt tài khoản.
Dòng sự kiện khác	<ul style="list-style-type: none"> Gửi mail về email người dùng đã đăng ký.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Tại bước gửi gói tin về server. <ul style="list-style-type: none"> Tài khoản đã được kích hoạt.

	<ul style="list-style-type: none"> - Hiển thị thông báo lỗi, yêu cầu người dùng nhập lại thông tin kích hoạt tài khoản.
Ngữ cảnh	Màn hình kích hoạt tài khoản.
Điều kiện tiên quyết	Người dùng phải mở ứng dụng và sở hữu ít nhất một tài khoản đã đăng ký để kích hoạt.
Điều kiện kết thúc	<ul style="list-style-type: none"> • Nhận thông báo gửi email kích hoạt tài khoản thành công.
Điểm mở rộng	Không có

3.4.2.2.3. UC-07: Quên mật khẩu

Tên của Use Case	Quên mật khẩu
Tóm tắt	Use Case dùng để cho phép người dùng cài đặt lại mật khẩu mới trong trường hợp người dùng quên mật khẩu.
Dòng sự kiện	<ul style="list-style-type: none"> • Use-case này bắt đầu khi người dùng mở màn hình đăng nhập tài khoản và có sử dụng ít nhất một tài khoản đã kích hoạt . • Nhấn thao tác quên mật khẩu tài khoản. • Hiển thị màn hình tái tạo mật khẩu gồm tên tài khoản. • Nhập tên tài khoản và nhấn nút quên mật khẩu. • Khi người dùng thực hiện xong các thao tác yêu cầu, gói tin sẽ gửi về server thông qua gRPC, hệ thống Auth Service sẽ xác thực gói tin có hợp lệ và gửi email để người dùng cài đặt lại mật khẩu tài khoản.
Dòng sự kiện khác	<ul style="list-style-type: none"> • Gửi mail về email người dùng đã đăng kí.
Dòng sự kiện phụ	<ul style="list-style-type: none"> • A5: Tại bước gửi gói tin về server. <ul style="list-style-type: none"> - Tài khoản không tồn tại. - Hiển thị thông báo lỗi, yêu cầu người dùng nhập lại thông tin kích hoạt tài khoản.
Ngữ cảnh	Màn hình quên mật khẩu
Điều kiện tiên quyết	Người dùng khởi động ứng dụng.
Điều kiện kết thúc	<ul style="list-style-type: none"> • Nhận thông báo cài đặt lại mật khẩu của tài khoản người dùng thành công.
Điểm mở rộng	Không có

3.4.2.2.4. UC-08: Xác thực bằng email

Tên của Use Case	XÁC THỰC BẰNG EMAIL
------------------	---------------------

Tóm tắt	Use Case dùng để cho phép người dùng xác thực bằng email thực hiện đăng ký tài khoản.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng đã thực hiện UC-06 (Kích hoạt tài khoản) hoặc UC-07 (Quên mật khẩu) và đang mở email cá nhân (gmail, yahoo, hotmail, outlook, etc). Email sẽ được gửi từ địa chỉ saigonparkingvn@gmail.com, trong đó có 1 đường dẫn để thực hiện xác thực bằng email. Nhấn vào đường dẫn xác thực. Hiển thị màn hình xác thực bằng email thành công.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Nhấn vào đường dẫn xác thực. <ul style="list-style-type: none"> Đường dẫn hết hạn. Hiển thị thông báo lỗi.
Ngữ cảnh	Trang web email cá nhân của người dùng.
Điều kiện tiên quyết	Người dùng đã thực hiện UC-06 (Kích hoạt tài khoản) hoặc UC-07 (Quên mật khẩu) và đang trong trang web email cá nhân.
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị thông báo xác thực bằng email thành công..
Điểm mở rộng	Không có

3.4.2.2.5. UC-09: Xem bãi đỗ xe trên bản đồ

Tên của Use Case	XEM BẢN ĐỒ BÃI ĐỖ XE
Tóm tắt	Use Case dùng để cho phép tìm kiếm tra cứu bãi đỗ xe trên bản đồ Google thông qua các API
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng thực hiện xong UC-01 (Đăng nhập) Hiển thị màn hình chính bản đồ. Gửi gói tin vị trí (tại điểm chính giữa màn hình bản đồ hiện tại) thông qua gRPC về server, hệ thống Parking Lot Service sẽ xử lý và trả về danh sách bãi xe trong phạm vi gần đó. Hiển thị các bãi xe trong phạm vi lân cận.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A2: Tại bước hiển thị bản đồ.

	<ul style="list-style-type: none"> - A2.1: Thao tác kéo thả bản đồ. - A2.2: Tìm kiếm địa điểm bằng Search của Google Map. - A2.3: Nhấn vào icon bãi đỗ xe thực hiện UC-11 (Xem thông tin chi tiết bãi xe) • A3: Tại bước gửi gói tin về server <ul style="list-style-type: none"> - Lấy danh sách bãi xe thất bại, hiển thị thông báo lỗi.
Ngữ cảnh	Màn hình chính của tài xế.
Điều kiện tiên quyết	Người dùng đã thực hiện UC-01 (Đăng nhập) thuộc dạng tài xế.
Điều kiện kết thúc	<ul style="list-style-type: none"> • Hiển thị màn hình bản đồ và các bãi xe gần vị trí đã chọn.
Điểm mở rộng	Không có

3.4.2.2.6. UC-10: Hiện danh sách các bãi đỗ đang có trên bản đồ

Tên của Use Case	HIỆN DANH SÁCH CÁC BÃI ĐỖ ĐANG CÓ TRÊN BẢN ĐỒ
Tóm tắt	Use Case dùng để cho phép người dùng xem danh sách của các bãi xe hiển thị trên bản đồ.
Dòng sự kiện	<ul style="list-style-type: none"> • Use-case này bắt đầu khi người dùng đã thực hiện UC-09 (Xem bãi đỗ xe trên bản đồ) • Nhấn chọn menu hiển thị danh sách bãi xe. • Hiển thị danh sách bãi xe tại vị trí đã chọn.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> • Chọn nhấn một bãi xe trên danh sách, thực hiện UC-11 (Xem thông tin chi tiết bãi xe). • Đóng menu, quay về màn hình chính
Ngữ cảnh	Màn hình chính.
Điều kiện tiên quyết	Người dùng đã thực hiện UC-09 (Xem bản đồ bãi đỗ xe)
Điều kiện kết thúc	<ul style="list-style-type: none"> • Hiển thị danh sách các bãi xe gần vị trí đã chọn
Điểm mở rộng	Không có

3.4.2.2.7. UC-11: Xem thông tin chi tiết của bãi đỗ xe

Tên của Use Case	XEM THÔNG TIN CHI TIẾT CỦA BÃI ĐỖ XE
------------------	--------------------------------------

Tóm tắt	Use Case dùng để cho phép người dùng xem thông tin chi tiết của bãi đỗ xe
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng đã thực hiện xong UC-01 (Đăng nhập). Nhấn vào bãi đỗ xe bất kỳ. Gửi gói tin Id bãi đỗ thông qua gRPC về server, hệ thống Parking Lot Service sẽ xử lý và trả về thông tin chi tiết của bãi xe đó Hiển thị màn hình thông tin chi tiết bãi đỗ đã chọn và các nút thao tác khác.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A2: Tại bước nhấn vào bãi xe bất kỳ <ul style="list-style-type: none"> A2.1:Nhấn chọn bất kỳ Icon bãi xe trên bản đồ. A2.2: Chọn một bãi xe đang hiển thị từ UC-10 (Hiện danh sách các bãi đỗ đang có trên bản đồ). A3: Tại bước gửi gói tin về server <ul style="list-style-type: none"> Lấy thông tin bãi xe thất bại, hiển thị thông báo lỗi. Đóng màn hình thông tin chi tiết quay về màn hình chính. Thực hiện giao tác khác trên màn hình thông tin chi tiết. <ul style="list-style-type: none"> Chọn thao tác chỉ đường bắt đầu UC-12 (Chỉ đường đến vị trí bãi đỗ xe) Chọn thao tác đặt chỗ bắt đầu UC-13 (Đặt chỗ đỗ xe) Chọn thao tác đánh giá bãi xe bắt đầu UC-16 (Đánh giá chất lượng và thái độ của bãi xe).
Ngữ cảnh	Màn hình xem thông tin chi tiết bãi đỗ xe.
Điều kiện tiên quyết	Người dùng đã thực hiện UC-09 (Xem bản đồ bãi đỗ)
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị thông tin chi tiết của bãi xe đã chọn.
Điểm mở rộng	Không có

3.4.2.2.8. UC-12: Chỉ đường đến vị trí bãi đỗ xe

Tên của Use Case	CHỈ ĐƯỜNG ĐẾN VỊ TRÍ BÃI ĐỖ XE
Tóm tắt	Use Case dùng để chỉ đường người dùng đến bãi xe đã chọn.

Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng thực hiện xong UC-11 (Xem thông tin chi tiết bãi đỗ xe) Nhấn thao tác chỉ đường. Chuyển về màn hình chính. Ứng dụng sẽ vẽ đường đi từ vị trí hiện tại đến bãi xe người dùng đã chọn.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A2: Tại bước nhấn thao tác chỉ đường <ul style="list-style-type: none"> A2.1: Sử dụng chỉ đường của ứng dụng. A2.2: Sử dụng chỉ đường của Google Map Nếu lựa chọn chỉ đường đến vị trí không phải bãi xe, sẽ vẽ thêm chỉ đường đi bộ cho người dùng đi đến điểm đó. A3: Tại bước ứng dụng vẽ đường đi <ul style="list-style-type: none"> Vẽ đường thất bại, hiển thị thông báo lỗi. Đóng màn hình thông tin chi tiết bãi xe và quay về màn hình chính.
Ngữ cảnh	Màn hình chính và màn hình xem thông tin chi tiết bãi đỗ.
Điều kiện tiên quyết	<ul style="list-style-type: none"> Người dùng đã thực hiện UC-01 (Đăng nhập) và chọn một bãi xe hoặc một vị trí cố định trên bản đồ. Mở định vị vị trí điện thoại.
Điều kiện kết thúc	<ul style="list-style-type: none"> Người dùng tắt chỉ đường. Chỉ đường đến vị trí, bãi xe thành công.
Điểm mở rộng	Không có

3.4.2.2.9. UC-13: Đặt chỗ đỗ xe

Tên của Use Case	ĐẶT CHỖ ĐỖ XE
Tóm tắt	Use Case dùng để cho phép người dùng đặt chỗ bãi đỗ xe.

Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng đã thực hiện xong UC-11 (Xem thông tin chi tiết bãi đỗ xe) Nhấn thao tác đặt chỗ. Chọn thời lượng đặt chỗ là bao nhiêu giờ, nhập biển số xe và xác nhận đặt chỗ. Gửi gói tin đặt chỗ bãi đỗ thông qua gRPC về server, hệ thống Booking Service sẽ xử lý và gửi về phía bãi xe để xác nhận yêu cầu đặt chỗ. Hiển thị thông báo đặt chỗ thành công và mã QR-code dùng ở UC-14 (Xác thực đặt chỗ bằng QR-code).
Dòng sự kiện khác	<ul style="list-style-type: none"> Gửi yêu cầu đặt chỗ về phía quản lý bãi xe để phản hồi yêu cầu đặt chỗ. (UC-20)
Dòng sự kiện phụ	<ul style="list-style-type: none"> A4: Tại bước gửi gói tin về server, bãi xe. <ul style="list-style-type: none"> - A4.1: Nếu bãi xe từ chối. <ul style="list-style-type: none"> + Gửi gói tin về server thông qua gRPC và Booking Service sẽ huỷ đặt chỗ, trả về thông báo đặt chỗ thất bại. + Màn hình người dùng hiển thị thông báo đặt chỗ thất bại. - A4.2: Nếu bãi xe chấp nhận. <ul style="list-style-type: none"> + Gửi gói tin về server thông qua gRPC và Booking Service sẽ chấp nhận đặt chỗ, trả về thông báo đặt chỗ thành công cho người dùng và lưu thông tin đặt chỗ ở phía bãi xe. + Màn hình người dùng hiển thị thông báo đặt chỗ thành công - Gửi gói tin thất bại, hiển thị thông báo lỗi. - Đóng màn hình đặt chỗ đỗ xe quay về màn hình chính Thực hiện UC-12 (Chỉ đường đến vị trí bãi đỗ xe)
Ngữ cảnh	Màn hình đặt chỗ đỗ xe.
Điều kiện tiên quyết	Người dùng đã thực hiện UC-11 (Xem thông tin chi tiết của bãi đỗ xe)
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị thông báo đặt chỗ thành công.
Điểm mở rộng	Không có

3.4.2.2.10. UC-14: Xác thực đặt chỗ bằng QR-code

Tên của Use Case	XÁC THỰC ĐẶT CHỖ BẰNG QR-CODE
------------------	-------------------------------

Tóm tắt	Use Case dùng để cho phép người dùng xác thực đặt chỗ bằng QR-code thông qua quét mã ở bãi xe.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng hoàn tất UC-13 (Đặt chỗ đỗ xe) và đang ở bãi xe đã đặt chỗ. Hiện màn hình đặt chỗ có hiển thị QR-Code. Người quản lý bãi xe sẽ quét QR-Code. Xác nhận thành công, người dùng lái xe vào bãi đỗ.
Dòng sự kiện khác	<ul style="list-style-type: none"> Đồng thời thực hiện UC 22 (Xác nhận khách đỗ xe bằng cách quét QR-code) ở phía quản lý bãi đỗ xe.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Tại bước người quản lý bãi xe sẽ quét QR-Code. <ul style="list-style-type: none"> A3.1: Yêu cầu đặt chỗ hết hạn.. A3.2: Xác nhận thất bại mã QR-Code không trùng khớp.
Ngữ cảnh	Người dùng đi đến bãi xe.
Điều kiện tiên quyết	Người dùng đã thực hiện UC-13 (Đặt chỗ đỗ xe) và đã đến bãi đỗ xe đã đặt chỗ.
Điều kiện kết thúc	<ul style="list-style-type: none"> Xác nhận QR-Code thành công, thất bại hay phát sinh lỗi.
Điểm mở rộng	Không có

3.4.2.2.11. UC-15: Chat hỗ trợ khách hàng

Tên của Use Case	CHAT HỖ TRỢ KHÁCH HÀNG
Tóm tắt	Use Case dùng để cho phép người dùng liên hệ với bãi đỗ xe sau khi đặt chỗ để hỏi đáp các vấn đề.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng hoàn tất UC-13 (Đặt chỗ đỗ xe) Nhấn thao tác chat hỗ trợ khách hàng. Gửi gói tin đặt chỗ bãi đỗ thông qua gRPC về server, hệ thống Contact Service sẽ xử lý và gửi về phía bãi xe để xác nhận yêu cầu kết nối Socket. Hiển thị màn hình chat. Chat với phía quản lý bãi đỗ xe.
Dòng sự kiện khác	<ul style="list-style-type: none"> Kích hoạt UC-21 (Tư vấn khách hàng) ở phía bãi đỗ xe.

Dòng sự kiện phụ	<ul style="list-style-type: none"> Kết nối Socket thất bại. Hiển thị thông báo lỗi, người dùng có thể yêu cầu kết nối Chat lại.
Ngữ cảnh	Màn hình chat .
Điều kiện tiên quyết	Người dùng đã thực hiện UC-13 (Đặt chỗ đỗ xe).
Điều kiện kết thúc	<ul style="list-style-type: none"> Người dùng kết thúc chat với phía quản lý bãi xe.
Điểm mở rộng	Không có

3.4.2.2.12. UC-16: Đánh giá bãi đỗ xe

Tên của Use Case	ĐÁNH GIÁ BÃI ĐỖ XE
Tóm tắt	Use Case dùng để cho phép người dùng đánh giá chất lượng và dịch vụ của bãi xe
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng đã thực hiện UC-13 (Đặt chỗ đỗ xe) Sau khi hoàn thành đặt chỗ (đã được nhân viên bãi xe quét mã QR xác nhận), ứng dụng khách hàng sẽ hiển thị khung thông báo Booking Finish kèm nút Back ở góc phải dưới khung thông báo Khách hàng nhấn nút Back, ứng dụng sẽ chuyển sang màn hình đánh giá bãi đỗ xe Hiển thị màn hình đánh giá bãi xe: <ul style="list-style-type: none"> Nếu người dùng không muốn đánh giá, nhấn nút Feedback Later -> Quay trở lại màn hình bản đồ, kết thúc đánh giá bãi xe Nếu người dùng muốn đánh giá, nhập nội dung đánh giá, chấm điểm cho bãi đỗ xe sau đó nhấn nút Send Feedback Ứng dụng quay trở về màn hình xem thông tin chi tiết bãi xe Nếu người dùng muốn xem đánh giá vừa rồi, nhấn nút Comment -> Hiển thị danh sách đánh giá của bãi xe mới được cập nhật, bao gồm đánh giá của chính bản thân.

Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A6: Tại bước gửi gói tin về server. <ul style="list-style-type: none"> Gửi gói tin thất bại, phát sinh lỗi. Hiển thị thông báo lỗi.
Ngữ cảnh	Màn hình đánh giá bãi xe.
Điều kiện tiên quyết	<ul style="list-style-type: none"> Người dùng đã thực hiện UC-13 (Đặt chỗ đỗ xe)
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị các đánh giá của bãi xe bao gồm cả cái người dùng vừa thực hiện.
Điểm mở rộng	Không có

3.4.2.2.13. UC-17: Xem các dịch vụ ôtô khác

Tên của Use Case	XEM CÁC DỊCH VỤ ÔTÔ KHÁC
Tóm tắt	Use Case dùng để cho phép người dùng xem các dịch vụ ôtô khác và vị trí của các dịch vụ đó
Dòng sự kiện	<ul style="list-style-type: none"> Use-case bắt đầu khi người dùng đã thực hiện UC-01 (Đăng nhập). Chọn lựa dịch vụ ôtô bất kỳ. Hiển thị vị trí các dịch vụ ôtô trên bản đồ.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	Không có.
Ngữ cảnh	Màn hình chính.
Điều kiện tiên quyết	Người dùng đã thực hiện UC-01 (Đăng nhập)
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị vị trí các dịch vụ ôtô trên bản đồ
Điểm mở rộng	Không có

3.4.2.2.14. UC-18: Xem lịch sử đặt chỗ bãi xe.

Tên của Use Case	XEM LỊCH SỬ ĐẶT CHỖ BÃI XE
Tóm tắt	Use Case dùng để cho phép người dùng xem lịch sử các lần đặt chỗ bãi xe.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi người dùng đã thực hiện UC-01 (Đăng nhập) Nhấn hiện menu lựa chọn thao tác.

	<ul style="list-style-type: none"> Chọn thao tác xem lịch sử đặt chỗ bãi xe. Gửi gói tin lấy lịch sử đặt chỗ bãi xe dùng thông qua gRPC về server, hệ thống Booking Service sẽ xử lý và gửi lại cho người dùng danh sách lịch sử đặt chỗ. Hiển thị màn hình lịch sử đặt chỗ bãi xe của người dùng.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A4: Tại bước gửi gói tin về server. <ul style="list-style-type: none"> Gửi gói tin thất bại, phát sinh lỗi. Hiển thị thông báo lỗi.
Ngữ cảnh	Màn hình chính.
Điều kiện tiên quyết	Người dùng đã thực hiện UC-01 (Đăng nhập)
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị màn hình lịch sử đặt chỗ bãi xe của người dùng..
Điểm mở rộng	Không có

3.4.2.3. Use-case của bãi xe

3.4.2.3.1. UC-19: Xem dữ liệu thống kê của bãi đỗ

Tên của Use Case	XEM DỮ LIỆU THỐNG KÊ CỦA BÃI ĐỖ
Tóm tắt	Use Case dùng để cho phép quản lý bãi xe xem thống kê tình trạng, số lượng đặt chỗ thành công.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi quản lý thực hiện xong UC-01 (Đăng nhập) Hiển thị màn hình thống kê của bãi xe.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	Không có
Ngữ cảnh	Trang chủ của quản lý bãi xe. (Dashboard)
Điều kiện tiên quyết	Quản lý đã thực hiện xong UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị thông tin thống kê bãi đỗ xe. Quản lý đăng xuất hoặc thực hiện các thao tác khác.
Điểm mở rộng	Không có

3.4.2.3.2. UC-20: Cập nhập thông tin bãi đỗ

Tên của Use Case	CẬP NHẬP THÔNG TIN BÃI ĐỖ
------------------	---------------------------

Tóm tắt	Use Case dùng để cho phép quản lý thay đổi thông tin bãi đỗ xe.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi quản lý thực hiện xong UC-01 (Đăng nhập) Di chuyển chuột về bên trái để hiện menu Dashboard. Nhấn chọn thao tác cập nhật thông tin bãi đỗ. Hiển thị màn hình cập nhật thông tin bãi đỗ. Chỉnh sửa các trường thông tin cần đổi Nhấn nút cập nhật thông tin Gửi gói tin cập nhật bãi xe qua gRPC về server, hệ thống Parking lot Service sẽ xử lý và gửi lại cho phía quản lý bãi xe thông tin đã được cập nhật. Hiển thị thông báo thành công với màn hình có thông tin đã được cập nhật.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Tại bước gửi gói tin về server. <ul style="list-style-type: none"> Gửi gói tin thất bại hay phát sinh lỗi. Hiển thị thông báo lỗi.
Ngữ cảnh	Màn hình cập nhật thông tin bãi xe của quản lý bãi.
Điều kiện tiên quyết	Quản lý đã thực hiện xong UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị thông báo cập nhật bãi xe thành công.
Điểm mở rộng	Không có

3.4.2.3.3. UC-21: Phản hồi lại khách hàng về việc đặt chỗ

Tên của Use Case	PHẢN HỒI LẠI KHÁCH HÀNG VỀ VIỆC ĐẶT CHỖ
Tóm tắt	Use Case dùng để cho phép quản lý phản hồi về việc chấp nhận hay từ chối việc đặt chỗ bãi xe của tài xế.

Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi quản lý thực hiện xong UC-01 (Đăng nhập) Nhận yêu cầu đặt chỗ bãi xe từ tài xế. Chấp nhận hoặc từ chối việc đặt chỗ. Gửi gói tin phản hồi đặt chỗ bãi xe qua gRPC về server, hệ thống Parking lot Service sẽ xử lý và gửi lại cho phía tài xế gói tin phản hồi đặt chỗ. Quay về màn hình hiện tại.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A3: Tại bước gửi gói tin về server. <ul style="list-style-type: none"> Gửi gói tin thất bại hay phát sinh lỗi. Hiển thị thông báo lỗi.
Ngữ cảnh	Message thông báo.
Điều kiện tiên quyết	Quản lý đã thực hiện xong UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Gửi gói tin phản hồi khách hàng về việc đặt chỗ ở bãi.
Điểm mở rộng	Không có

3.4.2.3.4. UC-22: Tư vấn khách hàng

Tên của Use Case	TƯ VẤN KHÁCH HÀNG
Tóm tắt	Use Case dùng để cho phép quản lý tiếp nhận liên hệ từ người dùng.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi quản lý bãi xe hoàn tất UC-01 (Đăng nhập) và đã chấp nhận yêu cầu đặt chỗ của tài xế. Tài xế liên hệ với quản lý bãi xe bằng UC-15 (Tư vấn khách hàng). Nhận gói tin yêu cầu kết nối chat từ tài xế thông qua gRPC từ server, hệ thống Contact Service sẽ xử lý và xác nhận kết nối Socket với tài xế. Hiển thị màn hình chat. Chat với tài xế đã đặt chỗ về các vấn đề..
Dòng sự kiện khác	Không có.

Dòng sự kiện phụ	<ul style="list-style-type: none"> Kết nối Socket thất bại. Hiển thị thông báo lỗi, người dùng có thể yêu cầu kết nối Chat lại.
Ngữ cảnh	Màn hình chat .
Điều kiện tiên quyết	Người dùng, tài xế phải thực hiện UC-15 (Tư vấn khách hàng) để liên hệ với quản lý bãi xe.
Điều kiện kết thúc	<ul style="list-style-type: none"> Người dùng kết thúc chat với phía quản lý bãi xe.
Điểm mở rộng	Không có

3.4.2.3.5. UC 23: Hoàn tất đặt chỗ đỗ xe

Tên của Use Case	HOÀN TẤT ĐẶT CHỖ ĐỖ XE
Tóm tắt	Use Case dùng để cho phép quản lý bãi đỗ xe hoàn tất đặt chỗ đỗ xe
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi quản lý thực hiện xong UC-01 (Đăng nhập), không có thiết bị, dụng cụ quét mã và có tài xế ở bãi xe đã đặt chỗ thành công. Quản lý bãi xe nhấn nút hoàn tất đặt chỗ bãi xe. Gửi gói tin thông báo hoàn tất đặt chỗ bãi xe về server thông qua gRPC, hệ thống Booking Service nhận và kết thúc quá trình đặt chỗ. Hiển thị thông báo hoàn tất đặt chỗ.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	Không có
Ngữ cảnh	Màn hình chính của quản lý xe.
Điều kiện tiên quyết	Quản lý bãi xe đã thực hiện UC-01 (Đăng nhập) nhưng không có thiết bị quét mã QR-Code.
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị thông báo hoàn tất đặt chỗ bãi xe
Điểm mở rộng	Không có

3.4.2.3.6. UC 24: Xác nhận khách đỗ xe bằng cách quét QR-code

Tên của Use Case	XÁC NHẬN KHÁCH ĐỖ XE BẰNG CÁCH QUÉT QR-CODE
Tóm tắt	Use Case dùng để cho phép quản lý bãi xe quét QR-CODE để kết thúc quy trình đặt chỗ

Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi quản lý thực hiện xong UC-01 (Đăng nhập), có dụng cụ quét mã QR-Code và có tài xế ở bãi xe đang thực hiện UC-14 (Xác thực đặt chỗ bằng QR-code). Tài xế hiển thị thông tin đặt chỗ và mã QR-Code cho quản lý bãi. Quản lý quét mã QR-Code của tài xế. Gửi mã quét về ứng dụng để xác thực với gói tin đặt chỗ bãi đỗ, xác thực thành công.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A4: Tại bước xác thực mã QR-Code. <ul style="list-style-type: none"> A4.1: Mã QR-Code không trùng khớp. A4.2: Lịch đặt chỗ bãi xe đã hết hạn. Hiển thị thông báo lỗi.
Ngữ cảnh	Màn hình thông tin đặt chỗ của quản lý.
Điều kiện tiên quyết	Quản lý đã thực hiện UC-01 (Đăng nhập) và có thiết bị quét mã QR-Code.
Điều kiện kết thúc	<ul style="list-style-type: none"> Xác nhận gói tin đặt chỗ với mã QR-Code thành công.
Điểm mở rộng	Không có

3.4.2.4. Use-case của admin

3.4.2.4.1. UC 25: Xem dữ liệu thống kê

Tên của Use Case	XEM DỮ LIỆU THỐNG KÊ
Tóm tắt	Use Case dùng để cho phép Admin xem thống kê thông tin hoạt động của ứng dụng.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi quản lý thực hiện xong UC-01 (Đăng nhập) Hiển thị màn thống kê của Admin.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	Không có
Ngữ cảnh	Trang chủ của Admin. (Dashboard)
Điều kiện tiên quyết	Admin đã thực hiện xong UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị thông tin thống kê bãi đỗ xe.

	<ul style="list-style-type: none"> Admin đăng xuất hoặc thực hiện các thao tác khác.
Điểm mở rộng	Không có

3.4.2.4.2. UC 26: Tạo tài khoản người dùng

Tên của Use Case	TẠO TÀI KHOẢN NGƯỜI DÙNG
Tóm tắt	Use Case dùng để cho phép Admin tạo tài khoản người dùng.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi Admin thực hiện xong UC-01 (Đăng nhập) Nhấn chọn thao tác tạo tài khoản người dùng. Điền vào các trường thông tin để tạo tài khoản. Nhấn nút tạo tài khoản. Gửi gói tin tạo tài khoản người dùng về server thông qua gRPC, hệ thống User Service xử lý và gửi về thông báo tạo tài khoản thành công. Hiển thị thông báo tạo tài khoản thành công.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A5: Tại bước gửi gói tin về server. <ul style="list-style-type: none"> Hiển thị thông báo lỗi, yêu cầu người dùng nhập lại thông tin tạo tài khoản.
Ngữ cảnh	Màn hình tạo tài khoản dùng.
Điều kiện tiên quyết	Admin đã thực hiện xong UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị thông báo tạo tài khoản thành công.
Điểm mở rộng	Không có

3.4.2.4.3. UC 27: Xem danh sách người dùng

Tên của Use Case	XEM DANH SÁCH NGƯỜI DÙNG
Tóm tắt	Use Case dùng để cho phép Admin xem danh sách của người dùng.

Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi Admin thực hiện xong UC-01 (Đăng nhập). Nhấn chọn nút quản lý người dùng trên menu. Gửi gói tin yêu cầu lấy danh sách người dùng về server thông qua gRPC, hệ thống User Service xử lý và gửi về danh sách người dùng. <p>Hiển thị danh sách người dùng.</p>
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A2: Tại bước chọn thao tác người dùng. <ul style="list-style-type: none"> A2.1: Nhấn vào nút Activated, thực hiện UC-28 (Kích hoạt người dùng) A2.2: Nhấn vào nút Deactivated, thực hiện UC-29 (Vô hiệu hóa người dùng). A2.3: Chọn một dòng thông tin người dùng bất kỳ, hiện Popup gồm thông tin của người dùng đó và các thao tác khác Admin có thể tìm kiếm người dùng bằng thanh tìm kiếm hoặc các filter.
Ngữ cảnh	Màn hình quản lý danh sách người dùng của admin.
Điều kiện tiên quyết	Admin đã thực hiện UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển danh sách quản lý người dùng có phân trang.
Điểm mở rộng	Không có

3.4.2.4.4. UC 28: Kích hoạt người dùng

Tên của Use Case	KÍCH HOẠT NGƯỜI DÙNG
Tóm tắt	Use Case dùng để cho phép Admin kích hoạt tài khoản người dùng.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi Admin thực hiện xong UC-27 (Xem thông tin danh sách người dùng). Chọn một tài khoản người dùng đang có trạng thái Deactivated và nhấn nút Deactivated để chuyển sang trạng thái Activated. Gửi gói tin kích hoạt người dùng về server thông qua gRPC, hệ thống User Service xử lý và xác nhận kích hoạt tài khoản người dùng thành công. Vô hiệu hóa nút kích hoạt của tài khoản người dùng đó.

Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	
Ngữ cảnh	Màn hình quản lý danh sách người dùng của admin.
Điều kiện tiên quyết	Admin đã thực hiện UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Người dùng đã chuyển sang trạng thái Activated.
Điểm mở rộng	Không có

3.4.2.4.5. UC 29: Vô hiệu hoá người dùng

Tên của Use Case	VÔ HIỆU HOÁ NGƯỜI DÙNG
Tóm tắt	Use Case dùng để cho phép Admin vô hiệu hóa tài khoản người dùng.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi Admin thực hiện xong UC-27 (Xem thông tin danh sách người dùng). Chọn một tài khoản người dùng đang có trạng thái Activated và nhấn nút Activated để chuyển sang trạng thái Deactivated. Gửi gói tin vô hiệu hóa người dùng về server thông qua gRPC, hệ thống User Service xử lý và xác nhận vô hiệu hóa tài khoản người dùng thành công. Vô hiệu hóa nút vô hiệu hóa của tài khoản người dùng đó.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	
Ngữ cảnh	Màn hình quản lý danh sách người dùng của Admin.
Điều kiện tiên quyết	Admin đã thực hiện UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Người dùng đã chuyển sang trạng thái Deactivated.
Điểm mở rộng	Không có

3.4.2.4.6. UC 30: Cấp quyền quản lý bãi đỗ xe

Tên của Use Case	CẤP QUYỀN QUẢN LÝ BÃI ĐỖ XE
Tóm tắt	Use Case dùng để cho phép Admin cấp quyền quản lý bãi xe cho tài khoản người dùng

Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi Admin thực hiện xong UC-01 (Đăng nhập). Nhấn thao tác cấp quyền quản lý bãi xe. Chọn tài khoản người dùng kèm theo bãi xe muốn cấp phép quản lý. .Nhấn nút cấp quyền quản lý bãi xe. Hiển thị thông báo cấp quyền thành công.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> Nếu người dùng đã quản lý 1 bãi xe nào khác, huỷ yêu cầu cấp quyền.
Ngữ cảnh	Màn hình cấp quyền quản lý bãi xe của Admin
Điều kiện tiên quyết	Admin đã thực hiện UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị thông báo cấp quyền thành công.
Điểm mở rộng	Không có

3.4.2.4.7. UC 31: Xem danh sách bãi đỗ xe

Tên của Use Case	XEM THÔNG TIN BÃI ĐỖ XE
Tóm tắt	Use Case dùng để cho phép Admin xem thông tin của bãi đỗ xe.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi Admin thực hiện xong UC-01 (Đăng nhập). Nhấn chọn thao tác bãi đỗ xe trên menu. Gửi gói tin yêu cầu lấy danh sách thông tin bãi đỗ xe về server thông qua gRPC, hệ thống User Service xử lý và gửi về danh sách thông tin bãi đỗ xe. Hiển thị danh sách thông tin bãi đỗ xe có.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	<ul style="list-style-type: none"> A2: Tại bước chọn thao tác người dùng. <ul style="list-style-type: none"> Chọn dòng thông tin người dùng bất kì, nhấn nút Delete thực hiện UC-32 (Xoá bãi đỗ xe) Admin có thể tìm kiếm bãi đỗ xe bằng thanh tìm kiếm hoặc các filter.
Ngữ cảnh	Màn hình quản lý thông tin bãi đỗ xe của admin.
Điều kiện tiên quyết	Admin đã thực hiện UC-01 (Đăng nhập).

Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển danh sách quản lý thông tin bãi đỗ xe có.
Điểm mở rộng	Không có

3.4.2.4.8. UC 32: Xoá bãi đỗ xe

Tên của Use Case	XOÁ BÃI ĐỖ XE
Tóm tắt	Use Case dùng để cho phép Admin xoá bãi đỗ xe.
Dòng sự kiện	<ul style="list-style-type: none"> Use-case này bắt đầu khi Admin thực hiện xong UC-31 (Xem thông tin bãi đỗ xe) Chọn bãi đỗ xe muốn xoá. Nhấn nút xoá bãi đỗ xe. Gửi gói tin yêu cầu xoá các bãi đỗ xe đã chọn về server thông qua gRPC, hệ thống Parking lot Service xử lý và gửi về danh sách thông tin bãi đỗ xe đã được cập nhập theo trang. Hiển thị danh sách bãi đỗ xe đã được cập nhập và có phân trang.
Dòng sự kiện khác	Không có.
Dòng sự kiện phụ	Không có
Ngữ cảnh	Màn hình quản lý danh sách bãi đỗ xe của admin.
Điều kiện tiên quyết	Admin đã thực hiện UC-01 (Đăng nhập).
Điều kiện kết thúc	<ul style="list-style-type: none"> Hiển thị danh sách bãi xe đã được cập nhập và có phân trang.
Điểm mở rộng	Không có.

3.5. Các quy trình hoạt động chính (activity diagram)

3.5.1. Quy trình tạo tài khoản và xác thực người dùng.

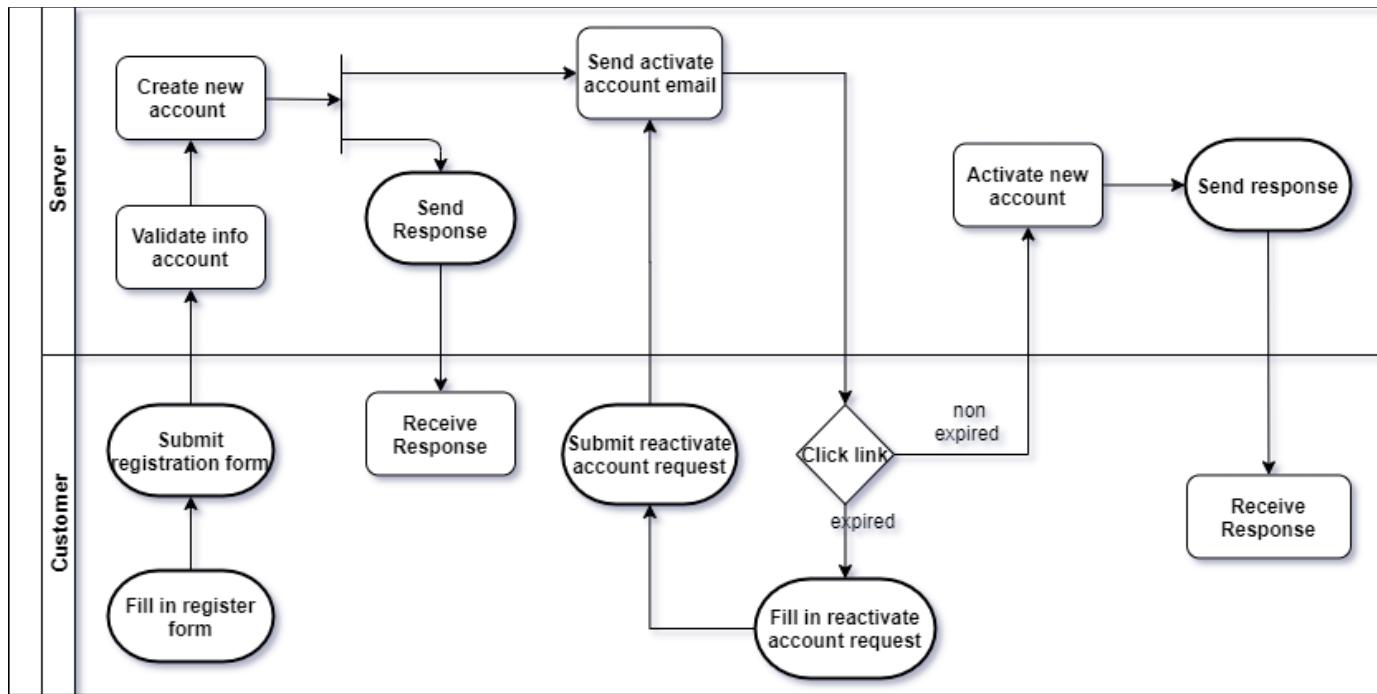
Người dùng điền đầy đủ thông tin tạo tài khoản vào form đăng ký tài khoản, ứng dụng sẽ gửi yêu cầu đăng ký tài khoản. Hệ thống kiểm tra thông tin tài khoản và tạo mới tài khoản cho người dùng, Mail Service sẽ gửi mail xác thực về email dùng để tạo tài khoản đó, người dùng kích hoạt tài khoản thông qua email đã nhận.

Trong trường hợp email xác thực hết hạn, người dùng tái kích hoạt tài khoản, ứng dụng sẽ gửi yêu cầu xác thực.

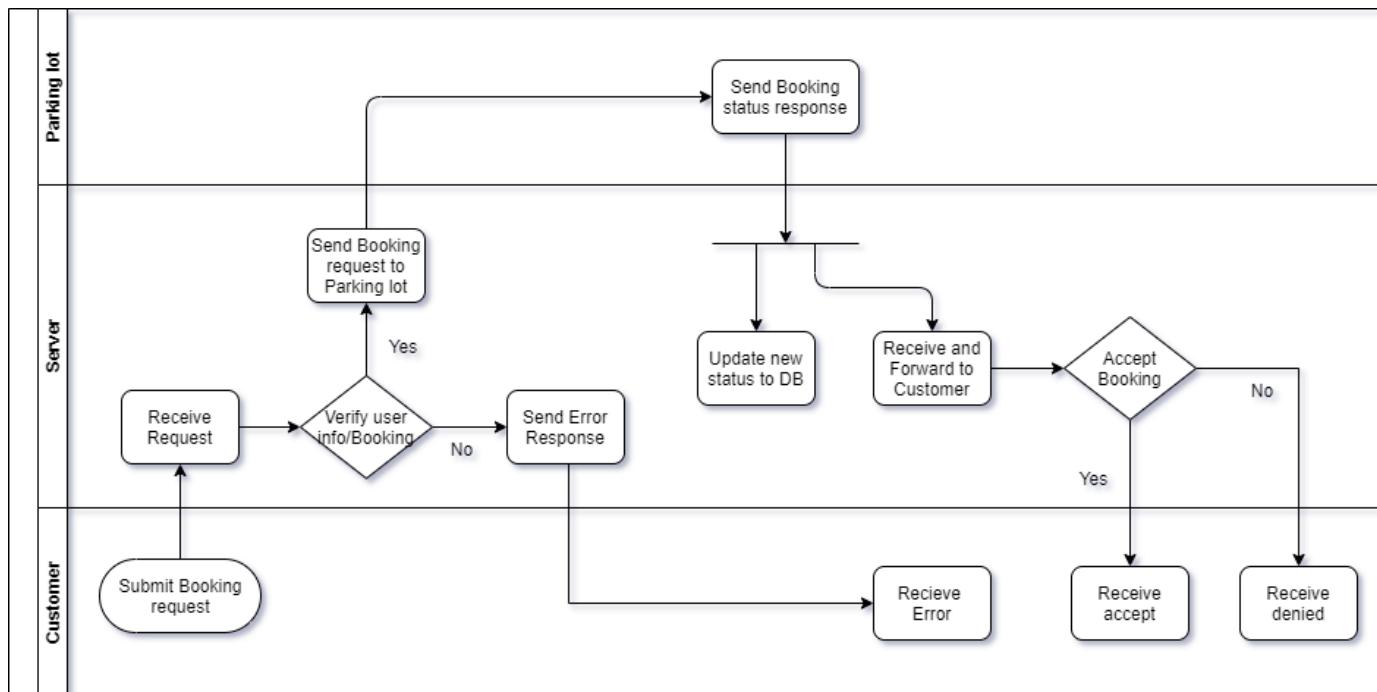
3.5.2. Quy trình người dùng đặt chỗ bãi đỗ xe.

Người dùng gửi thông tin bãi xe muốn đặt chỗ và thông tin chứng thực về hệ thống. Hệ thống sẽ xác thực thông tin tài khoản, nếu thông tin là hợp lệ, hệ thống sẽ gửi thông tin đặt chỗ cho phía bãi giữ xe xác nhận. Bãi xe chấp nhận đặt chỗ và gửi thông báo về người dùng.

Trong trường hợp bãi xe từ chối, hệ thống sẽ hiển thị thông báo đặt chỗ thất bại, người dùng quay về bước chọn bãi xe đặt chỗ.



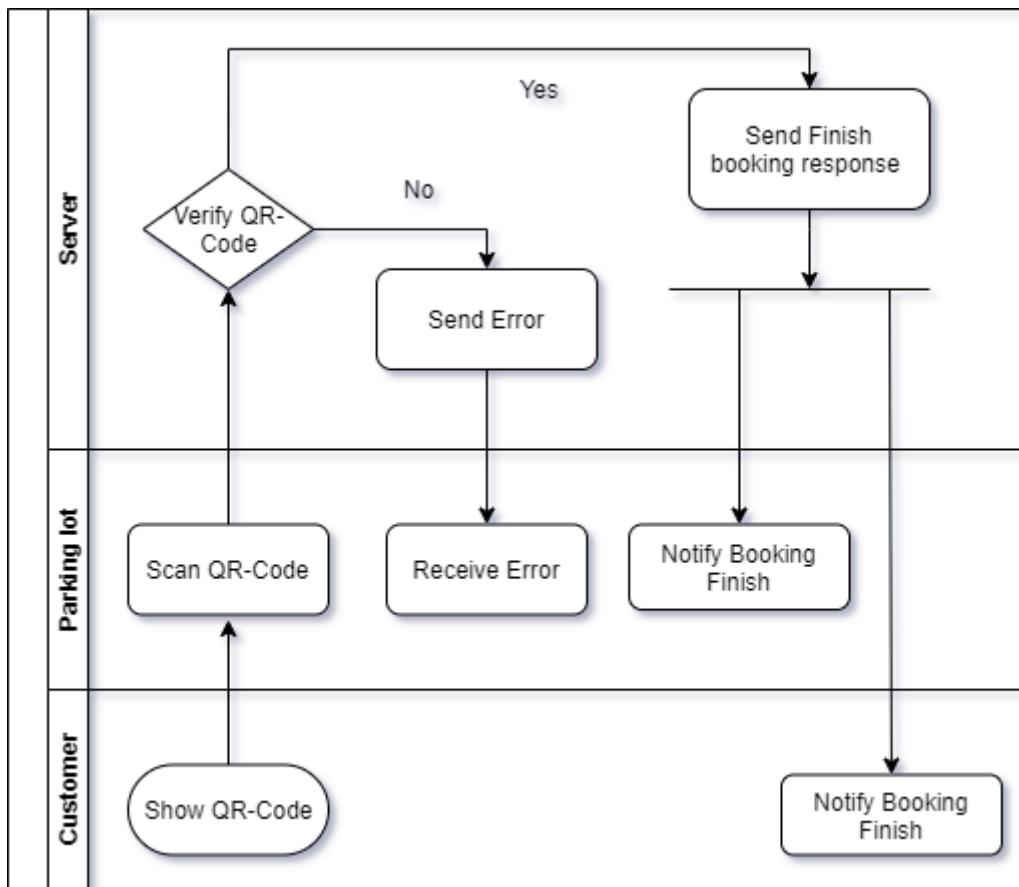
Hình 3.5.1. Quy trình tạo tài khoản và xác thực người dùng



Hình 3.5.2. Quy trình người dùng đặt chỗ bãi xe

3.5.3. Quy trình xác thực đặt chỗ bằng QR-Code.

Khi đến bãi xe, người dùng đưa mã QR-Code cho quản lý bãi xe quét mã đối chiếu và kết thúc đặt chỗ bãi đỗ..



Hình 3.5.3. Quy trình chỉ đường người dùng và xác thực đặt chỗ bằng QR code

3.6. Danh sách chức năng

3.6.1. Danh sách chức năng dành cho ứng dụng web của Admin

#	Feature	Note	Screen
Nhóm chức năng liên quan tới user-role			
1	Xem danh sách user-role		1
2	Tạo mới 1 user-role		1
Nhóm chức năng liên quan tới tài khoản của admin			
3	Xem danh sách admin		1
4	Tạo mới admin	mặc định role admin	1
5	Đăng nhập	trả về token	1
6	Quên mật khẩu		1
7	Đăng xuất	popup cảnh báo	0
8	Cập nhật thông tin tài khoản		1

#	Feature	Note	Screen
9	Xem thông tin tài khoản	có last_sign_in	1
Nhóm chức năng liên quan tới user			
10	Xem danh sách user (không phân loại / phân loại theo role)	có thể tìm theo filter	1
11	Tạo mới 1 user	chọn user-role	1
12	Xem thông tin cụ thể 1 user		1
13	Cập nhật thông tin 1 user		1
14	Xóa 1 user	popup	0
15	Kích hoạt 1 user	popup	
16	Hủy kích hoạt 1 user	popup	
Nhóm chức năng liên quan tới bãi xe			
17	Xem danh sách các bãi xe	có thể tìm theo filter	1
18	Tạo mới 1 bãi xe		1
19	Xem thông tin cụ thể 1 bãi xe		1
20	Cập nhật thông tin 1 bãi xe		1
21	Xóa 1 bãi xe	popup	0
22	Xem danh sách đánh giá 1 bãi xe	chỉ hiện rating	1
23	Xem thông tin cụ thể 1 đánh giá	rating + comment	1
24	Xem danh sách bãi xe đề nghị		1
25	Xem thông tin cụ thể 1 bãi xe đề nghị		1
26	Duyệt 1 bãi xe đề nghị		
Nhóm chức năng liên quan tới thông tin các điểm cấm dừng, đỗ xe và khung giờ tương ứng			
27	Xem danh sách các điểm cấm dừng, đỗ xe	có thể tìm theo filter	1
28	Tạo mới 1 điểm cấm dừng đỗ xe (để hiển thị lên map)		1
29	Xem thông tin cụ thể 1 điểm cấm dừng đỗ xe		1
30	Cập nhật thông tin 1 điểm cấm dừng đỗ xe		1
31	Xóa thông tin 1 điểm cấm dừng đỗ xe	popup	0

#	Feature	Note	Screen
Nhóm chức năng khác			
32	Admin dashboard (trang chủ admin)		1

Bảng 3.6.1. Danh sách chức năng dành cho ứng dụng web của Admin

3.6.2. Danh sách chức năng dành cho ứng dụng web của dự án

#	Feature	Note	Screen	
Nhóm chức năng liên quan tới tài khoản khách hàng				
1	Tạo mới tài khoản	gửi mail kích hoạt	1	
2	Đăng nhập	trả về token	1	
3	Quên mật khẩu	gửi mail xác nhận	1	
4	Đăng xuất	popup cảnh báo	0	
5	Cập nhật thông tin tài khoản		1	
6	Xem thông tin tài khoản	có last_sign_in	1	
Nhóm chức năng liên quan tới bãi xe				
7	Xem thông tin chi tiết bãi xe (khi nhấn vào marker trên map)	gồm đơn giá giữ xe	1	
8	Xem đánh giá chi tiết bãi xe (5 rating mới nhất) + btnMore	rating + comment	1	
9	Đánh giá 1 bãi xe (hiện chưa giới hạn số lần đánh giá)	rating + comment		
Nhóm chức năng liên quan tới sử dụng bản đồ				
10	Tìm kiếm bãi giữ xe quanh đây (req: LatLang + Radius)	quét theo zoom	1	
11	Tìm kiếm bãi giữ xe quanh điểm đến (req: LatLang + Radius)	quét theo zoom		
Nhóm chức năng liên quan tới sử dụng bản đồ (nâng cao)				
12	Dẫn đường đến 1 bãi xe cụ thể (vẽ trực tiếp lên bản đồ)	dùng polyline	1	
Nhóm chức năng nâng cao				
17	Xem offline bản đồ các bãi giữ xe		1	

Bảng 3.6.2. Danh sách chức năng dành cho ứng dụng web của dự án

3.6.3. Danh sách chức năng dành cho ứng dụng di động của tài xế

#	Feature	Note	Screen
Nhóm chức năng liên quan tới tài khoản khách hàng			
1	Tạo mới tài khoản	gửi mail kích hoạt	1
2	Đăng nhập	trả về token	1
3	Đăng xuất	popup cảnh báo	0
4	Xem thông tin tài khoản	có last_sign_in	1
5	Xem lịch sử booking		1
6	Chat với bãi xe khi đang chờ yêu cầu booking		1
Nhóm chức năng liên quan tới bãi xe			
7	Xem thông tin chi tiết bãi xe (khi nhấn vào marker trên map)	gồm đơn giá giữ xe	1
8	Xem đánh giá chi tiết bãi xe (5 rating mới nhất) + btnMore	rating + comment	1
9	Đánh giá bãi xe sau khi hoàn thành nghiệp vụ booking (hiện chưa giới hạn số lần đánh giá)	rating + comment	
10	Đặt trước chỗ đậu xe trong bãi		1
Nhóm chức năng liên quan tới sử dụng bản đồ			
11	Tìm kiếm bãi giữ xe quanh đây (req: LatLang + Radius)	quét theo zoom	1
12	Tìm kiếm bãi giữ xe quanh điểm đến (req: LatLang + Radius)	quét theo zoom	
13	Tìm kiếm các địa điểm khác (bệnh viện, trạm xăng, nhà hàng,...)	quét theo zoom	1
Nhóm chức năng liên quan tới sử dụng bản đồ (nâng cao)			
14	Dẫn đường đến 1 bãi xe cụ thể (vẽ trực tiếp lên bản đồ)	dùng polyline	1
15	Dẫn đường đến địa điểm thứ ba	dùng polyline	

Bảng 3.6.3. Danh sách chức năng dành cho ứng dụng di động của tài xế

3.6.4. Danh sách chức năng dành cho ứng dụng web giả lập ứng dụng quản lý bãi giữ xe

#	Feature	Note	Screen
Nhóm chức năng liên quan tới API dành cho bên thứ 3 - cung cấp phần mềm quản lý cho các bãi xe			
1	Cập nhật tổng số lượng chỗ (int newCapacity)	API	1

2	Cập nhật số lượng chỗ còn trống (int newAvailability)	(giả lập app bãi xe)	
3	Tăng số lượng chỗ trống lên 1 chỗ (empty)		
4	Giảm số lượng chỗ trống xuống 1 chỗ (empty)		
5	Tiếp nhận yêu cầu đặt chỗ từ ứng dụng khách hàng		1
6	Hủy yêu cầu đặt chỗ		1
7	Chấp nhận yêu cầu đặt chỗ		1
8	Hoàn thành yêu cầu đặt chỗ (bằng cách quét mã QR hoặc bấm hoàn thành yêu cầu trên web giả lập)		1

Bảng 3.6.4. Danh sách chức năng dành cho ứng dụng web giả lập ứng dụng quản lý bãi giữ xe

3.6.5. Danh sách chức năng dành cho ứng dụng app giả lập ứng dụng quét mã QR

#	Feature	Note	Screen
Nhóm chức năng liên quan tới API dành cho bên thứ 3 - cung cấp phần mềm quản lý cho các bãi xe			
1	Đăng nhập	API (giả lập app bãi xe)	1
2	Đăng xuất		
3	Quét kiểm tra mã QR được gửi vào app tài xế		

Bảng 3.6.5. Danh sách chức năng dành cho ứng dụng app giả lập máy quét mã QR

3.7. Cơ sở dữ liệu của dự án (database diagram)

3.7.1. Tổng quan

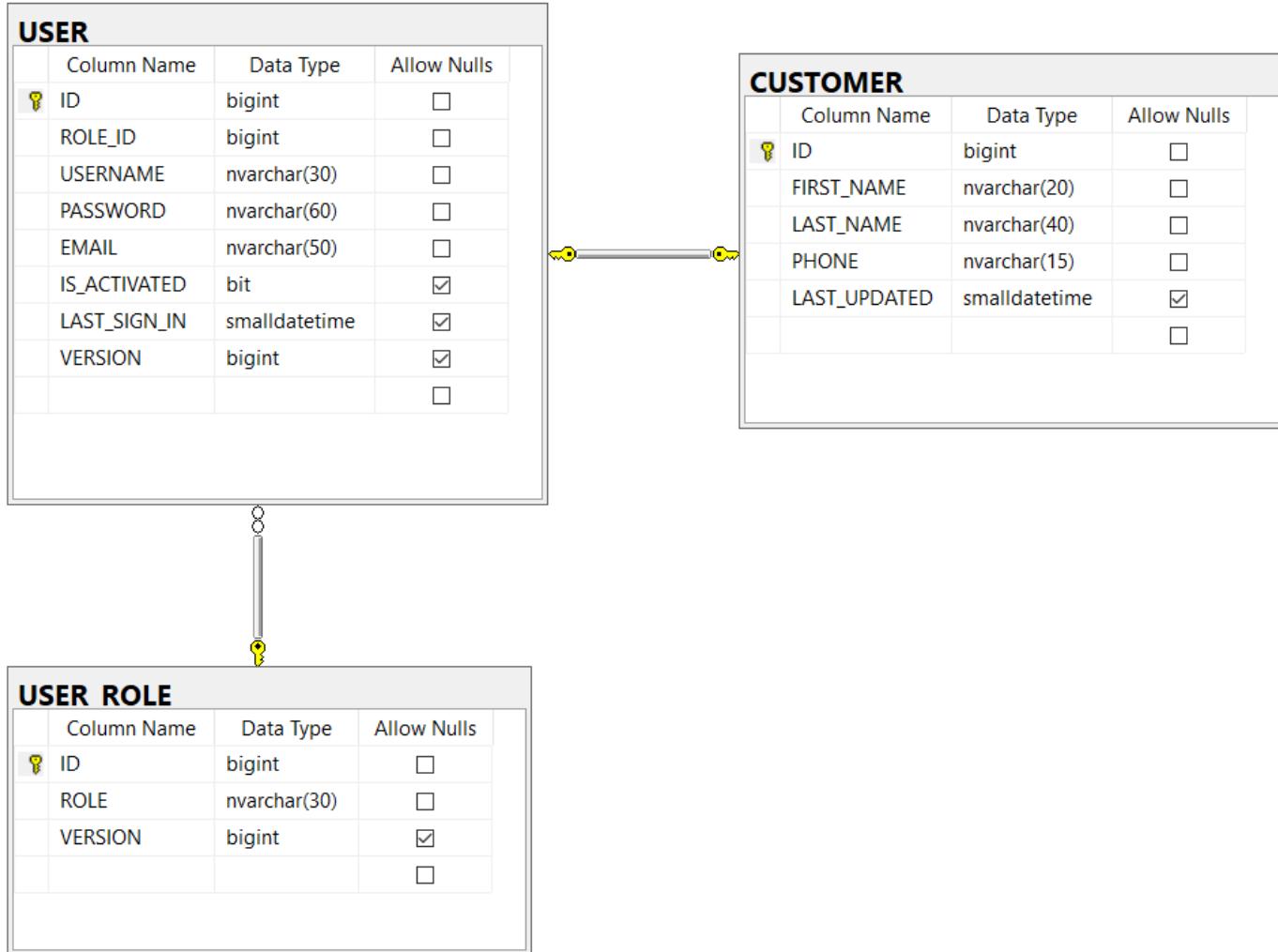
- Dự án Saigon Parking sử dụng hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) của Microsoft (Microsoft SQL Server - MSSQL), được triển khai trên môi trường đám mây sử dụng dịch vụ RDS của Amazon (Amazon Relational Database Service)
- Cơ sở dữ liệu của dự án là cơ sở dữ liệu quan hệ (RDB), gồm 16 bảng, thuộc 4 service, trong đó:
 - Auth Service có 1 bảng
 - Booking Service có 4 bảng
 - User Service có 3 bảng
 - Parking Lot Service có 8 bảng

3.7.2. UUID (GUID) là gì

- UUID** (Universal Unique Identifier) hay **GUID** (Global Unique Identifier) là một định danh duy nhất trên toàn cầu, hay duy nhất toàn thể vũ trụ. UUID ra đời chủ yếu để che giấu đi khóa chính tự động tăng của các bảng trong cơ sở dữ liệu, thường được cài đặt là Natural ID – định danh tự nhiên của bảng đó. Theo đó, đối với những bảng dữ liệu không muốn lặp ID gốc (ID số nguyên tự động tăng), người ta dùng UUID để public ra bên ngoài và dùng ID gốc cho những truy vấn nội bộ hệ thống.
- Ví dụ: **552d10d2-db0f-11ea-87d0-0242ac130003** là 1 UUID.

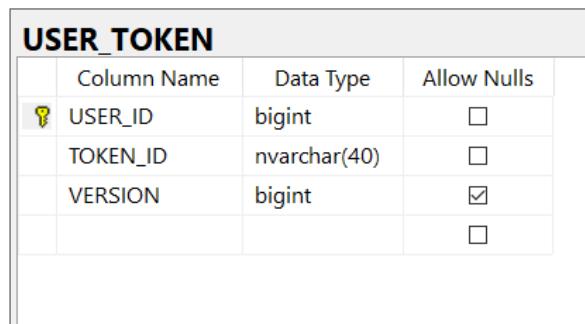
- Trong khuôn khổ dự án, UUID được sử dụng làm mã đặt chỗ đậu/đỗ xe thông qua ứng dụng và được mã hóa thành mã QR (Quick Reading code) để khách hàng cũng như bãi xe có thể dễ dàng tương tác.

3.7.3. Cơ sở dữ liệu của User service



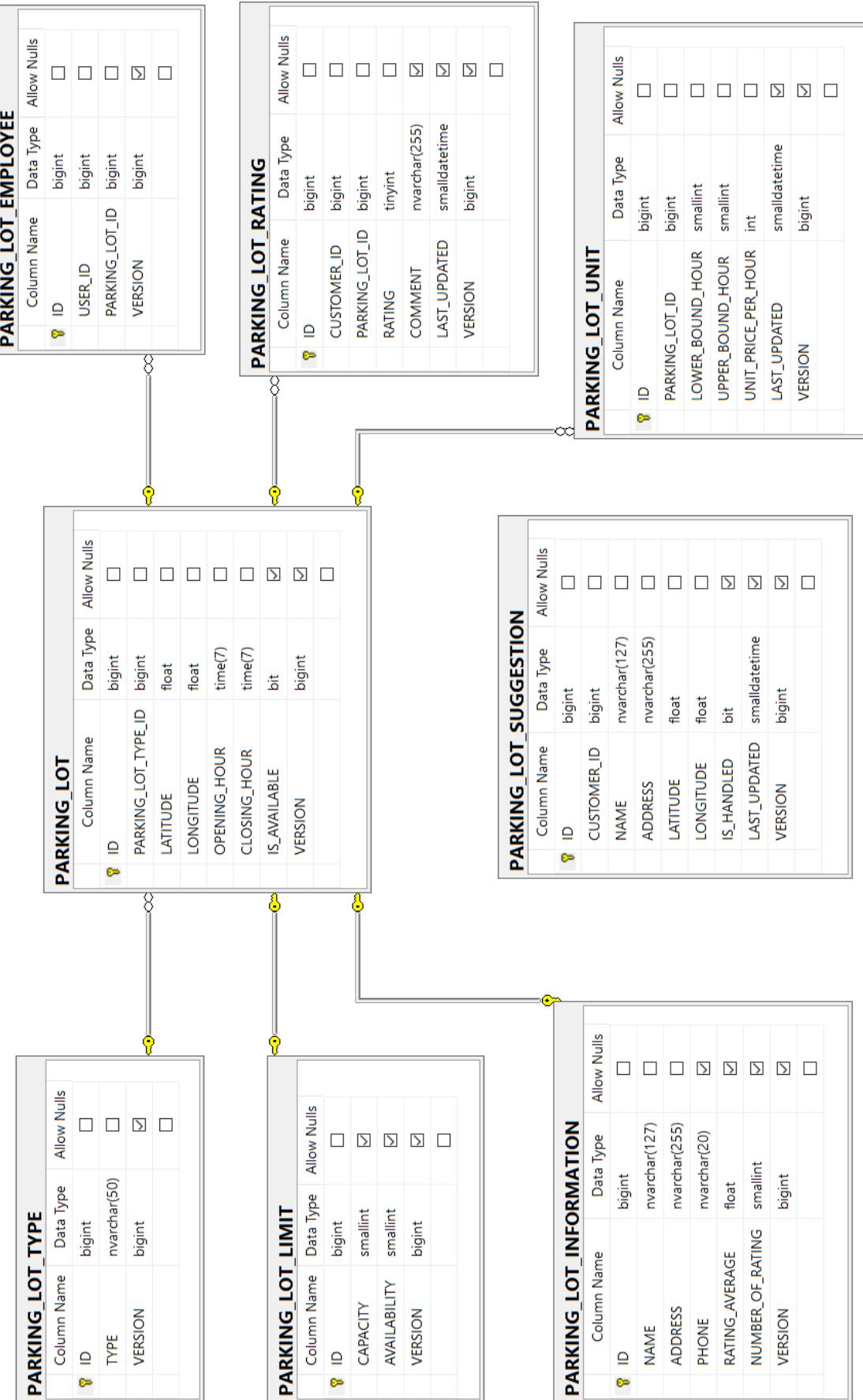
Hình 3.7.3. Cơ sở dữ liệu của User Service

3.7.4. Cơ sở dữ liệu của Auth service



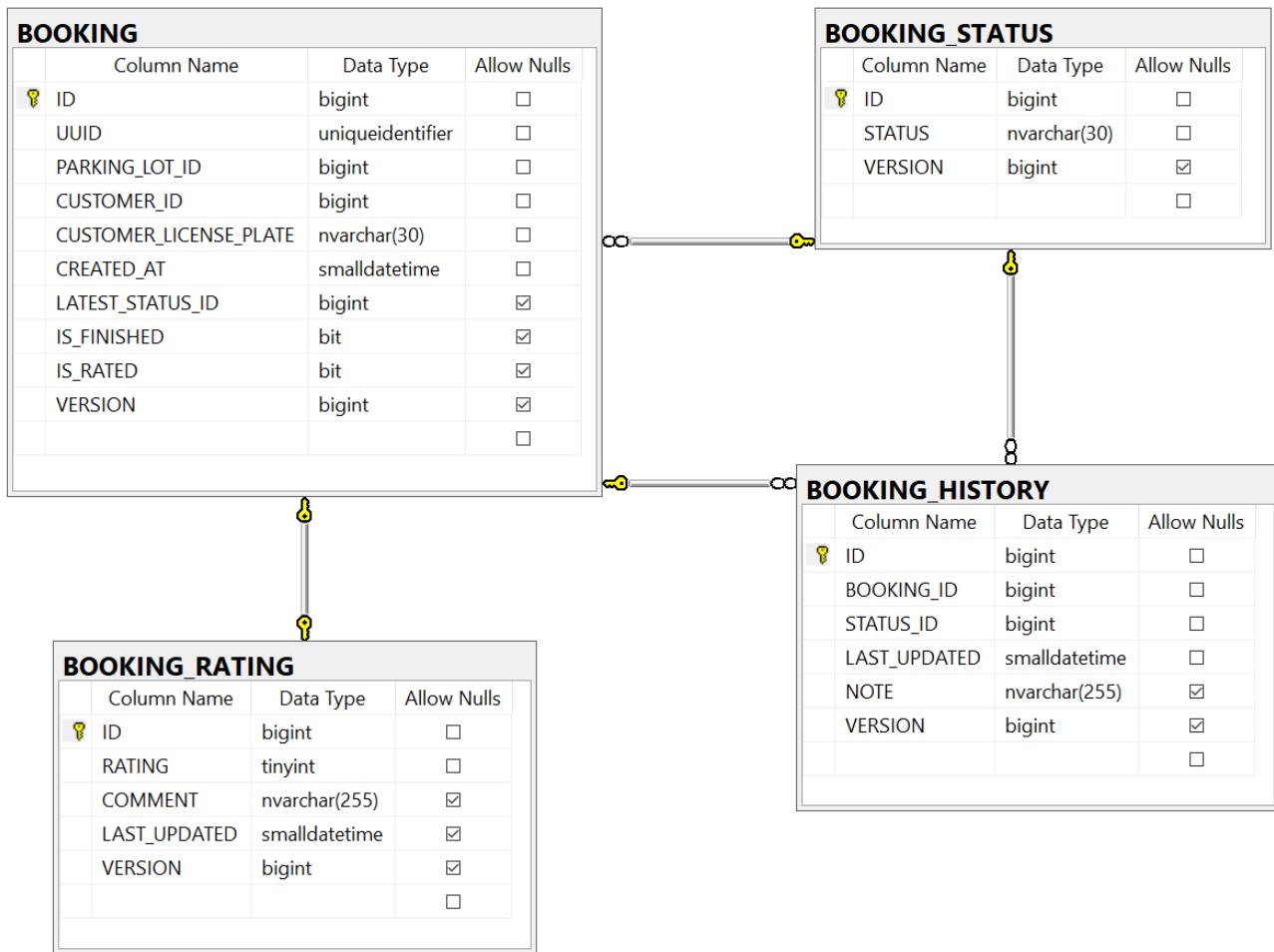
Hình 3.7.4. Cơ sở dữ liệu của Auth Service

3.7.5. Cơ sở dữ liệu của Parking-lot service



Hình 3.7.5. Cơ sở dữ liệu của Parking Lot Service

3.7.6. Cơ sở dữ liệu của Booking service



Hình 3.7.6. Cơ sở dữ liệu của Booking Service

Chương 4 GIAO TIẾP GIỮA CÁC THÀNH PHẦN HỆ THỐNG

Chương này nói về cách thức giao tiếp giữa các module trong hệ thống saigonparking, bao gồm các nội dung:

- giao tiếp nội bộ và giao tiếp từ bên ngoài
- giao tiếp 1 chiều và giao tiếp 2 chiều
- giao tiếp đồng bộ và giao tiếp bất đồng bộ
- cân bằng tải các giao tiếp
- vấn đề xác thực và phân quyền
- danh sách các API được cung cấp

4.1. Protobuf là gì

Protobuf (hay protocol buffer) là một platform được phát triển bởi Google dùng để serialize object thành dạng byte stream (nhị phân) và ngược lại deserialize byte stream (nhị phân) thành object (object sẽ tùy thuộc vào từng ngôn ngữ khác nhau). Phiên bản protobuf tính đến thời điểm viết báo cáo này là: v3.12.4.

Định nghĩa của protobuf sẽ được lưu trữ dưới dạng file với định dạng *.proto. Object trong file proto sẽ được định nghĩa dưới dạng struct (protobuf gọi là message). Để có thể phiên dịch các file proto ra các ngôn ngữ tương ứng, Google cung cấp 1 trình biên dịch dành cho protobuf, đó là protobuf compiler, còn gọi là protoc.

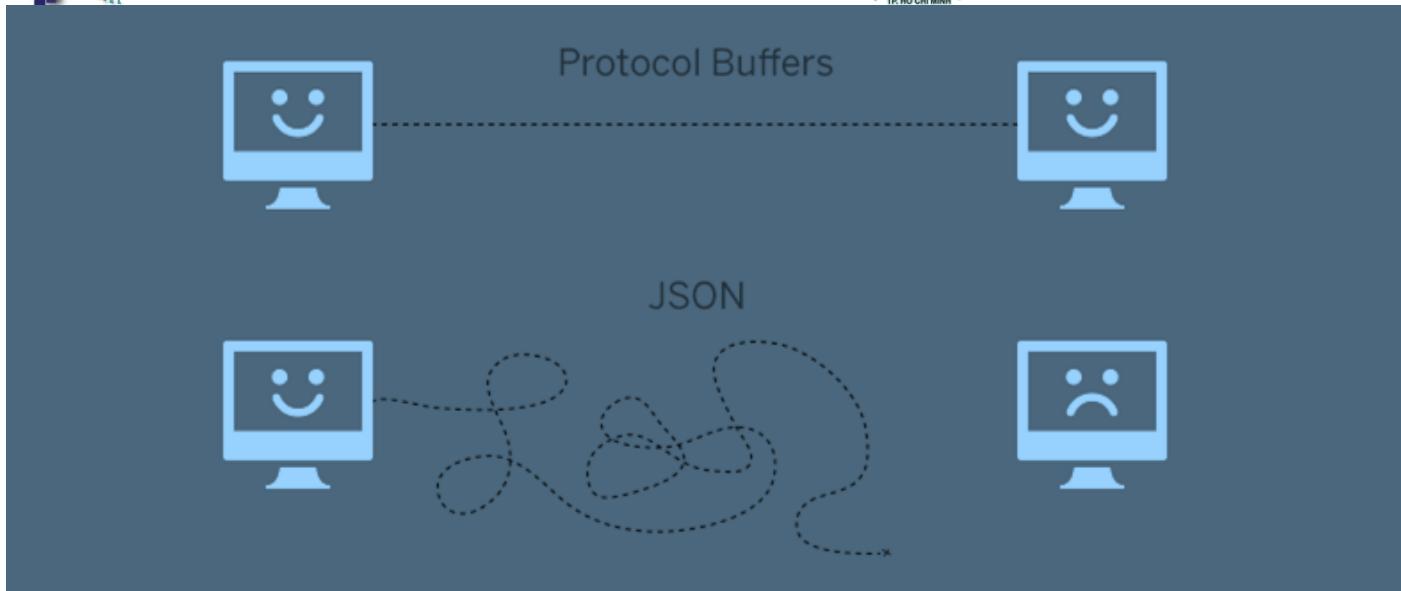
Do có sẵn compiler để biên dịch ra các ngôn ngữ khác nhau, protobuf đang dần trở thành lựa chọn hàng đầu để xây dựng các API dùng để giao tiếp giữa các module với nhau như giữa client với server hay giữa các microservice.

4.2. So sánh Protobuf và JSON

Đặc tính tiêu biểu của JSON là human readable – con người có thể đọc được, do vậy, dữ liệu sau khi serialize thành JSON vẫn nặng hơn so với protobuf, có thể phân giải mà không cần biết schema.

Trái với JSON, Protobuf không dành cho con người (con người không đọc được), vì là dữ liệu dạng nhị phân (binary). Tuy vậy do ở dạng nhị phân, protobuf có tốc độ xử lý rất nhanh, hơn hết sẽ rất khó để deserialize protobuf thành object nếu không biết schema. Nói cách khác, cần có shema (định nghĩa message) để có thể deserialize dữ liệu nhị phân thành object (object tùy vào từng ngôn ngữ khác nhau).

Do các đặc tính kể trên, nếu sử dụng protobuf thay cho JSON để gửi nhận dữ liệu giữa các module không những có thể cải thiện tốc độ xử lý cũng như gửi nhận dữ liệu mà còn tăng tính bảo mật cho dữ liệu được gửi (vì sẽ không thể deserialize mà không biết schema), tránh việc lộ dữ liệu khi đi qua các trung gian trên môi trường mạng giữa bên gửi và bên nhận.



Hình 4.2. Minh họa việc so sánh performance giữa Protocol Buffer và JSON

Nhóm em chọn sử dụng protobuf để serialize cũng như deserialize object khi gửi nhận thông qua môi trường mạng, dùng cho tất cả các trường hợp giao tiếp trong và ngoài hệ thống (gửi nhận với gRPC, gửi nhận với WebSocket, gửi nhận với Message Queue).

4.3. Unidirectional communication là gì

Giao tiếp 1 chiều (Unidirectional communication) là giao tiếp mà dữ liệu chỉ có thể gửi đi từ 1 phía.

Ví dụ: giao tiếp sử dụng REST API là giao tiếp 1 chiều vì chỉ có client gửi message (request) được cho server mà server thì không thể chủ động gửi message đến cho client được.

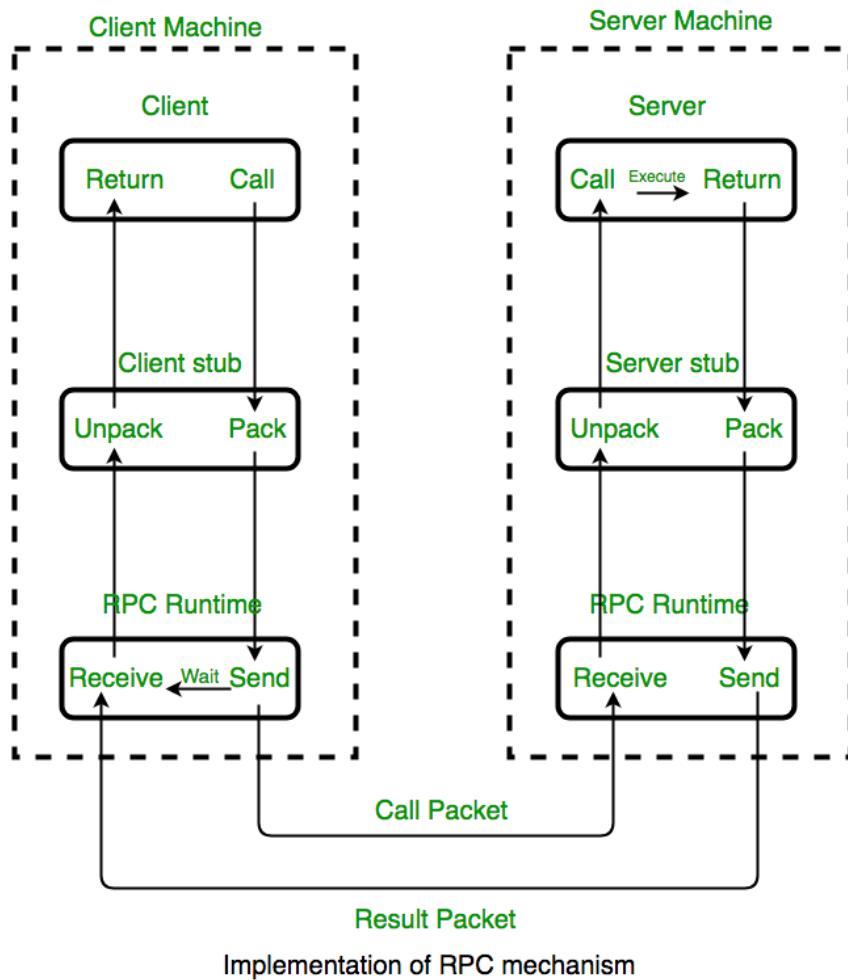
4.4. Bidirectional communication là gì

Khác với giao tiếp 1 chiều, giao tiếp 2 chiều (Bidirectional communication) hỗ trợ message có thể gửi đi từ cả 2 phía (mỗi bên vừa có thể là người nhận vừa có thể là người gửi).

Ví dụ: giao tiếp sử dụng WebSocket là giao tiếp 2 chiều vì cả client và server đều có thể gửi message cho nhau sau khi kết nối WebSocket được mở thành công.

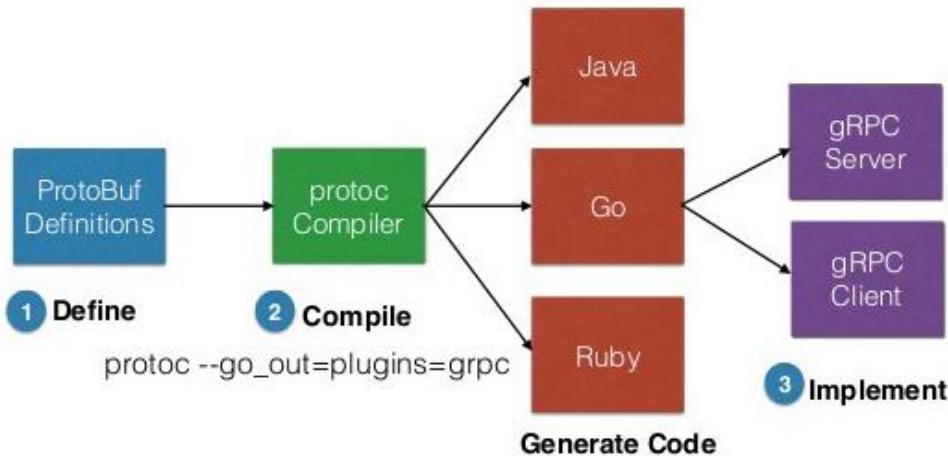
4.5. gRPC là gì

gRPC (gRPC Remote Procedure Call) là platform của Google dành riêng cho RPC và là sự kết hợp hoàn hảo giữa Protobuf của Google với HTTP/2. Hình dưới đây sẽ minh họa về cơ chế RPC:



Hình 4.5.a. Minh họa về cơ chế RPC – Remote Procedure Call

gRPC thường dùng để giao tiếp giữa các server với nhau (service to service), ngoài ra gRPC còn có thể sử dụng trong giao tiếp client - server (client to server). Có thể nói gRPC giúp cho mọi module giao tiếp tốt với nhau, bất kể đó là giao tiếp 1 chiều hay 2 chiều, bất kể đó là giao tiếp đồng bộ hay bất đồng bộ (giao tiếp đồng bộ và bất đồng bộ sẽ được trình bày trong những phần kế tiếp). Sở dĩ được như vậy là vì gRPC dùng cơ chế RPC thay cho REST, dùng Protobuf để gửi nhận dữ liệu giữa các module thay cho JSON hay XML truyền thống (ưu điểm của Protobuf đã được trình bày ở phần trước). Ngoài ra, gRPC còn hỗ trợ tốt trong việc xác thực và phân quyền người dùng (authentication and authorization), kiểm tra sự khả dụng của service (qua Health Checking API – kiểm tra service còn “sống” hay đã “chết”),... Cách thức cài đặt và sử dụng gRPC sẽ được minh họa qua sơ đồ sau đây:



Hình 4.5.b. Minh họa về cách thức cài đặt và sử dụng gRPC

Bước 1: Định nghĩa tất cả các message và service cần sử dụng trong các file proto.

Bước 2: Dùng protoc để generate định nghĩa trong file proto ra code tương ứng từng ngôn ngữ khác nhau.

Bước 3: Dùng những file vừa generate ra được để tiến hành đóng gói thành API, cũng như cài đặt server.

4.6. So sánh gRPC và REST

“RESTful và gRPC về bản chất là giống nhau, đều là công cụ RPC (remote procedure call). Tuy nhiên, chỉ khác nhau cách cài đặt (implement): gRPC sử dụng binary trong đóng gói dữ liệu để truyền thông (protobuf), còn RESTful thì tự do, bạn có thể lựa chọn (thường là text-based: xml hoặc json). Nhờ đó mà gRPC thường có tốc độ cao và độ trễ thấp hơn.” – Theo TechTalk - Có nên chuyển APIs từ REST sang gRPC?

Vì những ưu điểm vượt trội kể trên và vì muốn tìm hiểu cũng như trải nghiệm công nghệ mới, gRPC được chúng em đưa vào sử dụng trong dự án thay cho REST.

4.7. WebSocket là gì

Là giao thức hỗ trợ giao tiếp 2 chiều giữa client và server, WebSocket có độ trễ thấp, kích thước gói tin nhỏ thực hiện qua TCP thay vì HTTP nên thường được dùng cho các ứng dụng real-time như gaming, chat app.

Tuy được thiết kế riêng cho các ứng dụng Web, WebSocket vẫn hỗ trợ các loại ứng dụng khác.

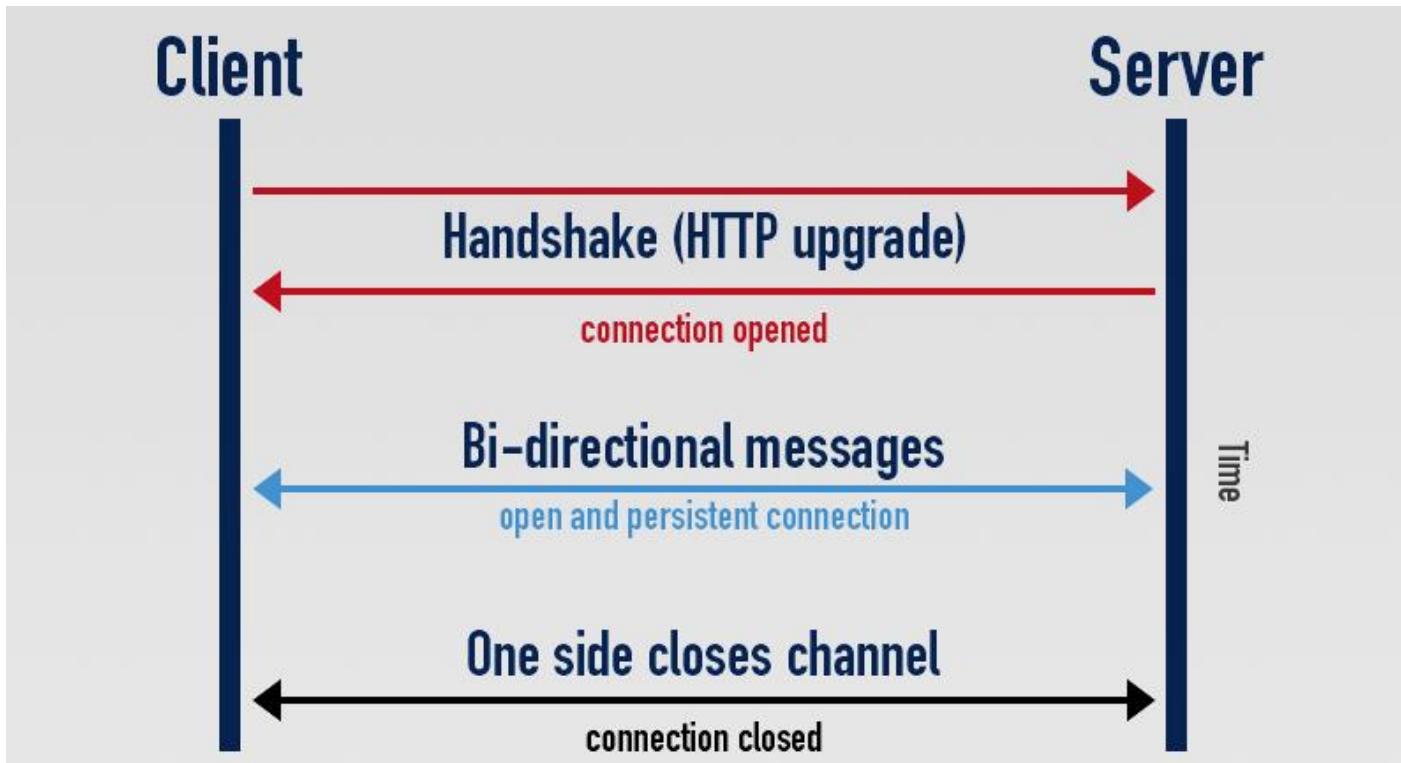
Có 4 sự kiện chính tương ứng với 4 event listener / event handler trong WebSocket, và 2 phương thức đó là:

Sự kiện	Event handler / Event listener	Mô tả
open	onOpen	(Khi) kết nối WebSocket được mở
message	onMessage	(Khi) có tin nhắn đến (dạng text hoặc dạng nhị phân)
error	onError	(Khi) có lỗi xảy ra trong việc giao tiếp giữa client và server
close	onClose	(Khi) kết nối WebSocket đóng lại

Bảng 4.7.a. Các sự kiện chính và các event handler tương ứng trong WebSocket

Phương thức	Mô tả
send	Gửi tin nhắn (dạng text hoặc dạng nhị phân)
close	Đóng kết nối WebSocket

Bảng 4.7.b. Các phương thức chính trong WebSocket



Hình 4.7.c. Minh họa toàn bộ quá trình giao tiếp bằng giao thức WebSocket

4.8. So sánh WebSocket và HTTP

Khác với HTTP, giao thức chuẩn của WebSocket là **ws://** (không bảo vệ với SSL) hay **wss://** (bảo vệ với SSL) và WebSocket hỗ trợ giao tiếp 2 chiều trong khi HTTP thì không. Ngoài ra, giao thức WebSocket là giao thức có trạng thái (stateful) khác với HTTP là không trạng thái (stateless). Điều đó có nghĩa là sau khi mở kết nối thành công, kết nối WebSocket sẽ được duy trì cho đến khi một trong 2 phía (client hoặc server) đóng kết nối hoặc do kết nối ở tầng dưới bị đóng lại (mất kết nối mạng).

Chúng em chọn WebSocket cho giao tiếp 2 chiều giữa client và server, đồng thời chúng em cũng chọn gửi nhận tin nhắn nhị phân thay vì tin nhắn văn bản để có thể tăng tính bảo mật và cải thiện tốc độ gửi nhận dữ liệu. (Tin nhắn nhị phân ở đây chính là protobuf object ứng với các ngôn ngữ khác nhau sau khi được serialize).

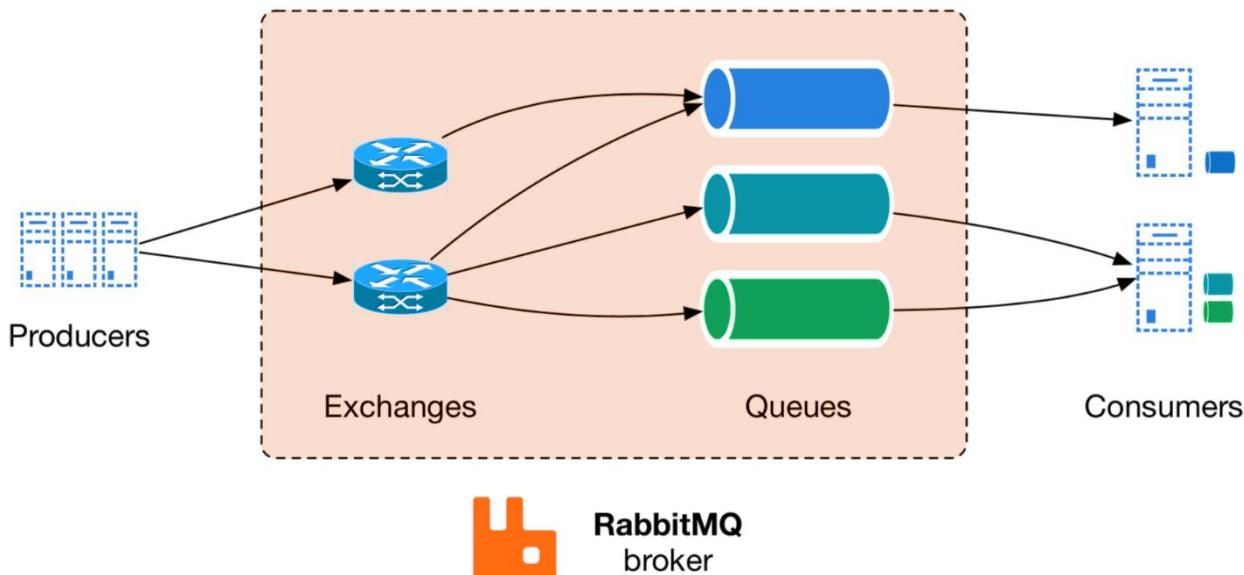
4.9. Message Queue là gì

Message Queue (MQ) có thể hiểu theo nghĩa đen của nó là hàng đợi tin nhắn. Định nghĩa queue chính là hàng đợi, có nghĩa là vào trước thì ra trước, vào sau thì ra sau (FIFO – First In First Out). Message Queue thường được sử dụng trong kiến trúc microservice và chủ yếu được dùng trong giao tiếp bất đồng bộ giữa các service với nhau. Việc sử dụng MQ giúp chúng ta dễ dàng xây dựng hệ thống phân tán cũng như scaling hệ thống.

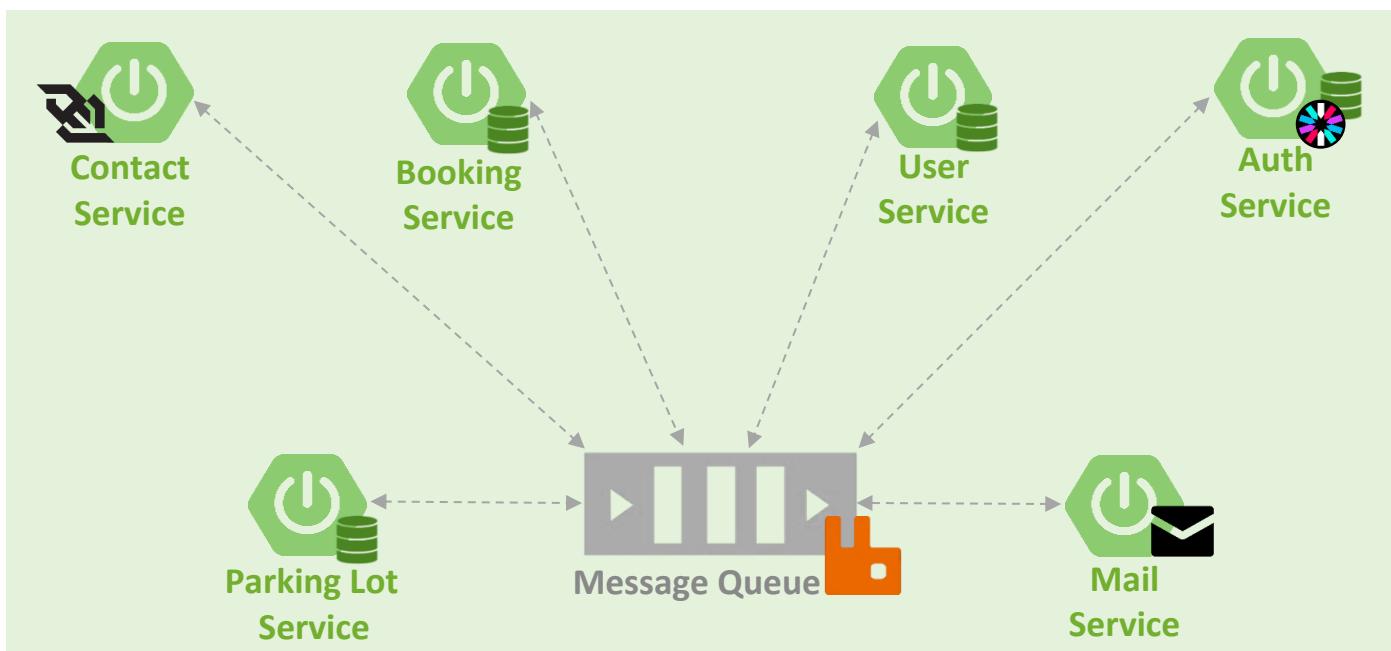
Đối với MQ, chúng em lựa chọn RabbitMQ để xây dựng hệ thống vì RabbitMQ có độ khó vừa phải, có nhiều tài liệu tham khảo trong quá trình cài đặt cũng như sử dụng, hỗ trợ tốt mô hình publish-subscribe...

Dữ liệu gửi nhận qua RabbitMQ của chúng em cũng là Protobuf object. Việc sử dụng Protobuf sẽ giúp cho tốc độ serialize, deserialize, cũng như gửi nhận dữ liệu qua Queue tăng đáng kể.

Mô hình hoạt động của RabbitMQ sẽ được minh họa sau đây:



Hình 4.9.a. Minh họa mô hình hoạt động của RabbitMQ



Hình 4.9.b. Minh họa kết nối giữa các microservice với RabbitMQ trong hệ thống SaigonParking

4.10. Cân bằng tải hệ thống

Với một hệ thống có thể có hàng nghìn user, việc phải tiếp nhận và xử lý đồng thời một lượng lớn request từ client đòi hỏi saigonparking phải có khả năng chịu tải cao, bất kể protocol đó là gRPC hay WebSocket, bất kể đó là giao tiếp đồng bộ hay bất đồng bộ,... Bài toán cân bằng tải hệ thống sẽ được trình bày sau đây...

4.10.1. Giao tiếp đồng bộ là gì

Giao tiếp đồng bộ (synchronous communication) là những giao tiếp mang tính real-time, xử lý trong thời gian thực, cần phản hồi ngay tức khắc. Đối với gRPC, giao tiếp đồng bộ sẽ block thread hiện tại (thread dùng để gọi request) cho đến khi nhận được phản hồi từ phía máy chủ gRPC (gRPC server).

gRPC sẽ được sử dụng để giao tiếp đồng bộ giữa các client bên ngoài hệ thống cũng như giao tiếp nội bộ giữa các microservice. Việc sử dụng gRPC thay cho REST sẽ cải thiện đáng kể tốc độ giao tiếp.

4.10.2. Giao tiếp bất đồng bộ là gì

Trái với giao tiếp đồng bộ, giao tiếp bất đồng bộ (asynchronous communication) là những giao tiếp không cần phản hồi ngay tức khắc (hoặc không cần phản hồi). Đối với gRPC, giao tiếp bất đồng bộ sẽ không block thread hiện tại (thread dùng để gọi request) và sẽ dùng 1 đối tượng là **StreamObserver** để handle khi request được máy chủ gRPC (gRPC server) xử lý xong và phản hồi về.

Giao tiếp bất đồng bộ giúp cho người nhận (ở đây là máy chủ) sẽ chủ động trong việc phản hồi (có thể trả lời hoặc không trả lời, ...) thay vì bị động (bắt buộc phản hồi) khi có cuộc gọi đến.

gRPC, WebSocket, MessageQueue sẽ được sử dụng để giao tiếp bất đồng bộ giữa các client bên ngoài hệ thống với hệ thống, cũng như giao tiếp nội bộ giữa các microservice. Riêng Message Queue sẽ được dùng riêng để giao tiếp bất đồng bộ giữa các microservice với nhau. Việc sử dụng Message Queue sẽ đảm bảo được các gói tin gửi đi sẽ không bị mất và sẽ lần lượt được bến nhận xử lý cho đến hết.

4.10.3. Cân bằng tải là gì

“*Cân bằng tải là một phương pháp phân phối khói lượng tải trên nhiều máy tính hoặc một cụm máy tính để có thể sử dụng tối ưu các nguồn lực, tối đa hóa thông lượng, giảm thời gian đáp ứng và tránh tình trạng quá tải trên máy chủ.*” – Theo Viettel IDC

4.10.4. Các thuật toán cân bằng tải thông dụng

Việc lựa chọn 1 máy chủ thích hợp để xử lý request phụ thuộc nhiều yếu tố. Ở mức cơ bản, có 2 thuật toán cân bằng tải được sử dụng, đó là random và round-robin.

- **Đối với random**, việc lựa chọn cũng như tên gọi của nó: lựa chọn máy chủ một cách ngẫu nhiên!
- **Đối với round-robin**, các máy chủ lần lượt sẽ được lựa chọn để xử lý request, khi đã luân phiên hết 1 vòng, sẽ quay trở lại máy đầu tiên, lặp lại như vậy cho những vòng kế tiếp.

Round-robin (RR) là thuật toán cân bằng tải mà hệ thống chúng em đã sử dụng, do RR phù hợp cho các hệ thống vừa và nhỏ, dễ dàng cấu hình, hoạt động tương đối hiệu quả.

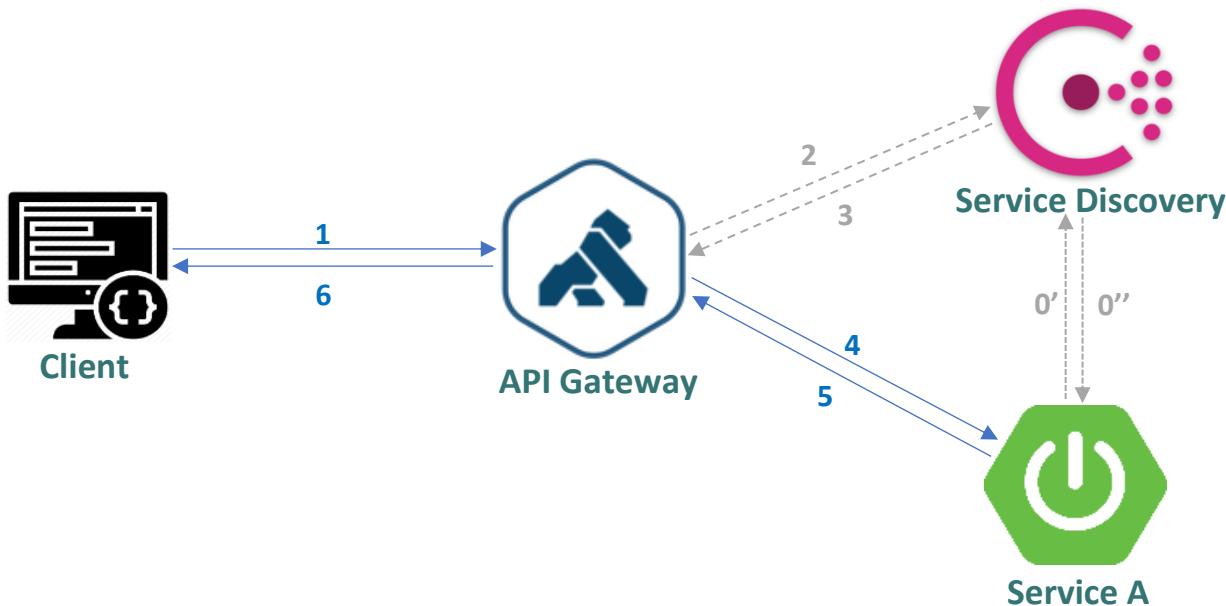
4.10.5. Client-side load-balancing và server-side load-balancing

Client-side load-balancing (cân bằng tải phía client) và server-side load-balancing (cân bằng tải phía server) là 2 cách thức cân bằng tải sử dụng **phương pháp service-discovery (truy tìm dịch vụ)** đã được đề cập ở chương trước. Theo đó:

- **Client-side load-balancing** sử dụng phương pháp truy tìm dịch vụ phía client (client-side service discovery), thường được dùng để cân bằng tải giao tiếp nội bộ giữa các microservice.
- **Server-side load-balancing** sử dụng phương pháp truy tìm dịch vụ phía server (server-side service discovery), thường được dùng để cân bằng tải giao tiếp từ bên ngoài hệ thống (qua gateway).

2 cách thức cân bằng tải kể trên sẽ được chúng em áp dụng thực tế vào việc cân bằng tải cho hệ thống này.

4.10.6. Cân bằng tải giao tiếp từ bên ngoài hệ thống



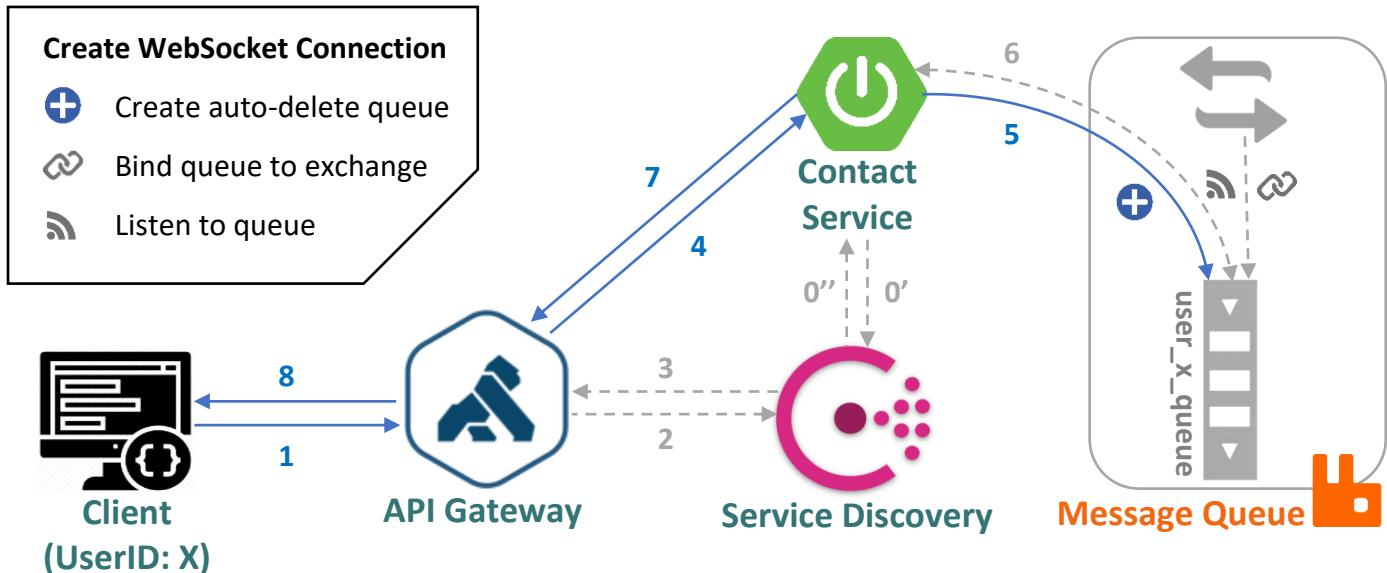
Hình 4.10.6. Minh họa cân bằng tải giao tiếp từ bên ngoài hệ thống (với hệ thống saigonparking)

- 0' Mỗi instance của service A đăng ký chính bản thân với Service Discovery khi instance được khởi động. (service_id + service_host + service_port + protocol)
- 0'' Service Discovery định kỳ kiểm tra sự khả dụng của những instance của service A thông qua API Health Checking Protocol
- 1 Client gửi request đến API gateway
- 2 Gateway ánh xạ request path thành domain name (VD: /com.bht.saigonparking.api.grpc.auth.AuthService/ → grpc-saigon-parking-auth-service.service.saigonparking-dc.consul). Lúc này Service Discovery đóng vai trò như 1 DNS resolver, Gateway sẽ truy vấn xuống Service Discovery để phân giải domain name vừa có được thành địa chỉ instance của Service A (địa chỉ gồm host + port).
- 3 Service Discovery phân giải domain name trả về địa chỉ 1 instance của service A trong hệ thống backend thông qua cơ chế cân bằng tải randomized round robin (lựa chọn 1 instance trong list những instance đang khả dụng của Service A bằng thuật toán randomized round robin (*), trả về địa chỉ của instance đó)
- 4 Gateway chuyển tiếp request từ client đến instance của service A dựa vào địa chỉ instance của service A vừa nhận được từ Service Discovery.
- 5 Instance của service A xử lý request từ client và gửi kết quả trả về cho Gateway
- 6 Gateway chuyển kết quả trả lại cho client.

(*): Theo “**Load Balancing Strategies for Consul**”, <https://hashicorp.com/blog/load-balancing-strategies-for-consul/>

4.10.7. Cân bằng tải giao tiếp từ bên ngoài hệ thống đối với websocket

4.10.7.1. Mở kết nối WebSocket với server

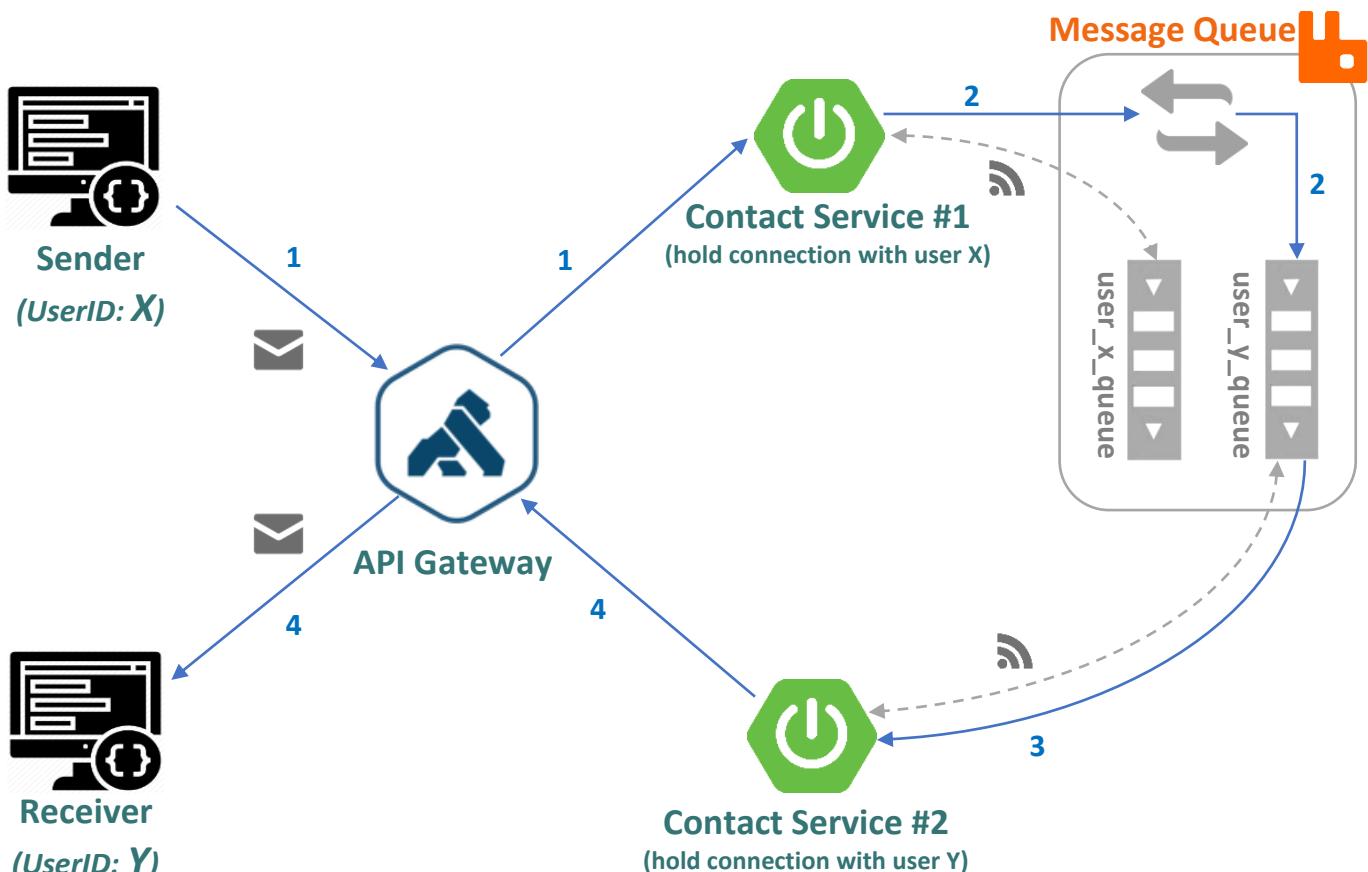


Hình 4.10.7.1. Minh họa mở kết nối WebSocket với server (với hệ thống saigonparking, đã cân bằng tải)

- 0' Mỗi instance của Contact Service đăng ký chính bản thân với Service Discovery khi instance được khởi động (service_id + service_host + service_port + protocol)
- 0'' Service Discovery định kỳ kiểm tra sự khả dụng của những instance của Contact Service thông qua API Health Checking Protocol
- 1 Client gửi request handshake đến API gateway
- 2 Gateway ánh xạ request path thành domain name (ở đây là /contact → saigon-parking-contact-service.saigonparking-dc.consul). Lúc này Service Discovery đóng vai trò như 1 DNS resolver, Gateway sẽ truy vấn xuống Service Discovery để phân giải domain name vừa có được thành địa chỉ instance của Contact Service (địa chỉ gồm host + port).
- 3 Service Discovery phân giải domain name trả về địa chỉ 1 instance của Contact Service trong hệ thống backend thông qua cơ chế cân bằng tải randomized round robin (lựa chọn 1 instance trong list những instance đang khả dụng của Contact Service bằng thuật toán randomized round robin (*), trả về địa chỉ của instance)
- 4 Gateway chuyển tiếp request từ client đến instance của Contact Service dựa vào địa chỉ instance của Contact Service vừa nhận được từ Service Discovery. Instance của Contact Service nhận được request, tiến hành xác thực request (quá trình xác thực request sẽ được trình bày trong phần sau) và bắt đầu quá trình handshake để mở kết nối WebSocket 2 chiều với client.
- 5 Instance của Contact Service đăng ký 1 auto-delete queue (queue tự động xóa khi không còn ai lắng nghe) cho client với cách đặt tên queue thống nhất cho hệ thống saigonparking là user_{user_id}_queue.
- 6 Sau khi tạo queue thành công, bind queue vừa tạo được với contact exchange (exchange này được chúng em định nghĩa thuộc loại topic exchange – sẽ tự động routing message đến đúng queue đã bind theo routing key, chúng em cũng định nghĩa: với routing key là user_{user_id}_queue.#, message sẽ được routing đến queue có tên là user_{user_id}_queue, việc này để đảm bảo message luôn gửi đúng đến người nhận). Kế tiếp, instance của Contact Service sẽ đăng ký lắng nghe queue vừa tạo. Kể từ lúc này, message khi đến queue sẽ được Message Queue gửi cho instance của Contact Service vì đã đăng ký lắng nghe message.
- 7 Instance của service A xử lý request handshake thành công, mở kết nối và gửi kết quả trả về cho Gateway

8 Gateway chuyển kết quả handshake thành công trở lại cho client. Kết nối WebSocket 2 chiều được mở!

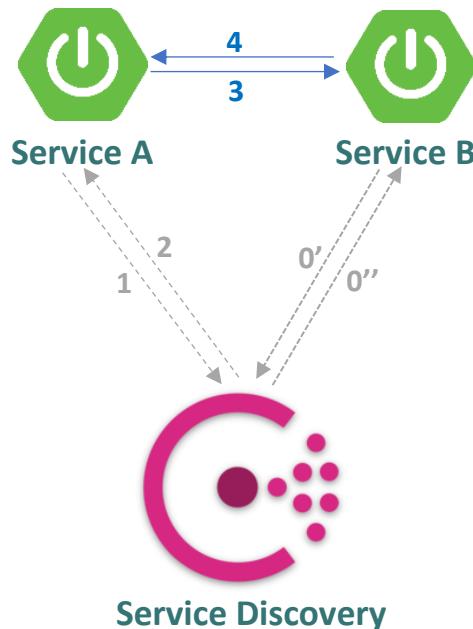
4.10.7.2. Gửi nhận message với WebSocket



Hình 4.10.7.2. Minh họa gửi nhận message với WebSocket (với hệ thống saigonparking, đã cân bằng tải)

- 0' Client bên nhận (Receiver) mở kết nối WebSocket với Server.
Instance thứ 2 của Contact Service (instance mở connection với Receiver) tiến hành đăng ký queue mới (auto-delete queue) cho Receiver (**user_y_queue**) và đăng ký lắng nghe queue đó.
- 0'' Client bên nhận (Sender) mở kết nối WebSocket với Server
Instance thứ 1 của Contact Service (instance mở connection với Sender) tiến hành đăng ký queue mới (auto-delete queue) cho Sender (**user_x_queue**) và đăng ký lắng nghe queue đó.
- 1 Sender tiến hành gửi message (binary mesage) đến Receiver thông qua instance 1 của Contact Service.
- 2 Message được instance 1 của Contact Service publish xuống queue của Receiver.
- 3 Message Queue sẽ publish message đến instance thứ 2 của Contact Service, vì instance này trước đó đã đăng ký lắng nghe queue của Receiver (**user_y_queue**)
- 4 Instance thứ 2 của Contact Service nhận được message và parse user ID của Receiver từ tên của queue gửi (UserID: Y), sau đó lấy ra session của user có ID là Y và tiến hành gửi message đến cho Receiver.

4.10.8. Cân bằng tải giao tiếp từ bên trong hệ thống (đồng bộ)

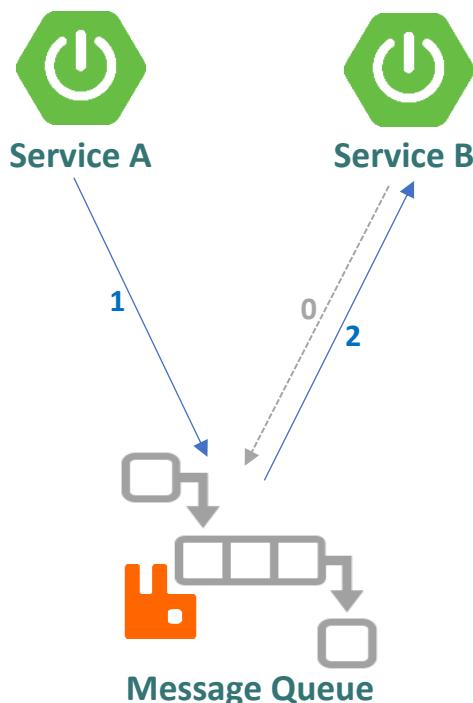


- 0' Mỗi instance của service B đăng ký chính bản thân với Service Discovery khi instance được khởi động.
(service_id + service_host + service_port + protocol)
- 0'' Service Discovery định kỳ kiểm tra sự khả dụng của những instance của service B thông qua API Health Checking Protocol
- 1 Instance của service A truy vấn list địa chỉ (host + port) những instance của service B trong hệ thống backend (through qua id của service B đã đăng ký với Service Discovery)
- 2 Service Discovery trả về list địa chỉ những instance Service B khả dụng (những instance còn sống)
- 3 Instance của service A tiến hành lựa chọn 1 instance của Service B (through qua cơ chế round-robin) và thực hiện gọi API đến instance đó.
- 4 Instance của service B trả kết quả cho instance của service A.

Hình 4.10.8. Minh họa cân bằng tải giao tiếp đồng bộ bên trong hệ thống saigonparking

4.10.9. Cân bằng tải giao tiếp từ bên trong hệ thống (bất đồng bộ)

- 0 Instance của service B đăng ký lắng nghe queue X của Message Queue
- 1 Instance của service A publish message đến queue X thông qua exchange Y của Message Queue (X đã bind với Y trước đó, Y là exchange kiểu topic)
- 2 Message Queue sẽ lựa chọn 1 listener của queue X theo cơ chế Round-Robin để publish message đến listener đó, ở đây là 1 instance của service B
- 3 Instance của service B nhận message từ Message Queue và xử lý message đó



Hình 4.10.9. Minh họa cân bằng tải giao tiếp bất đồng bộ bên trong hệ thống saigonparking

4.11. Danh sách các API cung cấp

API nhóm chúng em xây dựng gồm 2 nhóm: nhóm API dành cho WebSocket và nhóm API dành cho gRPC:

4.11.1. API WebSocket

#	API URL	Mô tả
1	/contact	URL dành cho mobile client gọi request để mở kết nối WebSocket
2	/contact/web?token=	URL dành cho web client gọi request để mở kết nối WebSocket
3	/contact/qrsscanner	URL dành cho app giả lập máy quét QR để mở kết nối WebSocket

4.11.2. API gRPC

API dành cho gRPC được phân thành các nhóm sau đây:

#	Nhóm	Mô tả
1	Protected	Nhóm các API được bảo vệ (cần đính kèm access token vào lời gọi API)
2	Admin only	Nhóm các API dành riêng cho Admin
3	Internal use	Nhóm các API dùng gọi nội bộ hệ thống (giữa các microservices với nhau)
4	External use	Nhóm các API dùng gọi từ bên ngoài hệ thống (qua API gateway)

4.11.2.1. Auth Service API

#	API / gRPC service	Protected	Admin only	Internal use	External use	Mô tả
1	checkUsernameAlreadyExist				X	Kiểm tra xem usernme đó đã tồn tại hay chưa
2	checkEmailAlreadyExist				X	Kiểm tra xem email đó đã tồn tại hay chưa
3	validateUser				X	Kiểm tra đăng nhập hợp lệ (username, password, role) và cấp access token + refresh token nếu đăng nhập hợp lệ
4	registerUser				X	Đăng ký tài khoản khách hàng mới
5	sendResetPasswordEmail				X	Gửi mail kèm link để user đặt lại mật khẩu
6	sendActivateAccountEmail				X	Gửi mail kèm link để user kích hoạt tài khoản
7	generateNewToken	X			X	Cấp mới access token cho user (cần đính kèm refresh token đã cấp cho user trước đó)

#	API / gRPC service	Protected	Admin only	Internal use	External use	Mô tả
8	activateNewAccount	X			X	Kích hoạt tài khoản cho user (cần đính kèm token đã cấp cho user trước đó thông qua link đính kèm trong email)

4.11.2.2. User Service API

#	API / gRPC service	Protected	Admin only	Internal use	External use	Mô tả
1	countAllUserGroupByRole	X	X		X	Đếm số lượng user theo user role
2	countAllUser	X	X		X	Đếm số lượng user, có hoặc không có filter
3	getAllUser	X	X		X	Lấy tất cả user, có hoặc không có filter, có pagination
4	getUserById	X	X		X	Lấy thông tin user, biết ID
5	getUserByUsername	X		X	X	Lấy thông tin user, biết username
6	getCustomerById	X	X	X	X	Lấy thông tin khách hàng, biết ID
7	getCustomerByUsername	X		X	X	Lấy thông tin khách hàng, biết username
8	mapToUsernameMap	X		X		Ánh xạ danh sách user ID thành danh sách username tương ứng
9	checkUsernameAlreadyExist	X	X	X	X	Kiểm tra username đó đã tồn tại hay chưa
10	checkEmailAlreadyExist	X	X	X	X	Kiểm tra email đó đã tồn tại hay chưa
11	createUser	X	X	X	X	Tạo mới 1 tài khoản user
12	createCustomer	X	X	X	X	Tạo mới 1 tài khoản khách hàng
13	updateCustomer	X			X	Cập nhật thông tin khách hàng
14	updatePassword	X			X	Cập nhật password của user
15	activateUser	X	X	X	X	Kích hoạt tài khoản user
16	deactivateUser	X	X	X	X	Hủy kích hoạt tài khoản user
17	deleteUserById	X	X	X	X	Xóa 1 user, biết user ID
18	deleteMultiUserById	X	X	X	X	Xóa nhóm user, biết danh sách ID
19	getEmployeeManageParkingLotList	X	X		X	Lấy danh sách các nhân viên quản lý bãi xe, biết ID của bãi xe

4.11.2.3. Parking Lot Service API

#	API / gRPC service	Protected	Admin only	Internal use	External use	Mô tả
1	countAllParkingLotGroupByType	X	X		X	Đếm số lượng bãi xe theo loại
2	getParkingLotIdByAuthorizationHeader	X			X	Lấy ID của bãi xe khi biết ID của nhân viên (ID nhân viên được parse từ chuỗi JWT)
3	getParkingLotIdByParkingLotEmployeeId	X	X	X		Lấy ID của bãi xe khi biết ID của nhân viên (ID nhân viên được truyền vào request body)
4	countAllParkingLot	X	X		X	Đếm số lượng bãi xe, có hoặc không có filter
5	getAllParkingLot	X	X		X	Lấy danh sách những bãi xe, có hoặc không có filter, có pagination
6	checkLimit	X			X	Lấy số lượng chỗ còn trống trong bãi xe
7	checkAvailability	X			X	Cho trước ID bãi xe, kiểm tra xem bãi xe có ID đó có còn khả dụng hay không (còn chỗ, còn mở cửa)
8	checkUnavailability	X			X	Cho trước danh sách ID của các bãi xe, kiểm tra xem các bãi xe đó bãi nào không còn khả dụng (không còn chỗ, không còn trong giờ mở cửa)
9	getParkingLotById	X			X	Lấy thông tin chi tiết của bãi xe biết ID của bãi xe
10	getTopParkingLotInRegionOrderByDistanceWithName	X			X	Lấy top top những bãi xe trong vùng giới hạn có tâm và bán kính, kèm theo tên bãi xe
11	getTopParkingLotInRegionOrderByDistanceWithoutName	X			X	Lấy top top những bãi xe trong vùng giới hạn có tâm và bán kính, không kèm theo tên bãi xe
12	deleteParkingLotById	X	X		X	Xóa 1 bãi xe khi biết ID của bãi
13	deleteMultiParkingLotById	X	X		X	Xóa nhiều bãi xe (truyền vào danh sách ID của các bãi xe)
14	countAllParkingLotHasRatings	X	X		X	Đếm số bãi xe có bình luận, có hoặc không có filter
15	getAllParkingLotHasRatings	X	X		X	Lấy tất cả bãi xe có bình luận, có hoặc không có filter, có pagination

#	API / gRPC service	Protected	Admin only	Internal use	External use	Mô tả
16	countAllRatingsOfParkingLot	X			X	Đếm tất cả bình luận về bãi xe đó, có hoặc không có filter
17	getAllRatingsOfParkingLot	X			X	Lấy tất cả bình luận về bãi xe đó, có hoặc không có filter, có pagination
18	getParkingLotRatingCountGroupByRating	X			X	Đếm số lượt đánh giá về bãi xe, gom nhóm theo đánh giá
19	updateParkingLotAvailability	X			X	Cập nhật số lượng chỗ còn trống của bãi xe
20	mapToParkingLotNameMap	X		X		Ánh xạ từ danh sách ID của những bãi xe thành danh sách tên của những bãi xe tương ứng
21	getParkingLotNameByParkingLotId	X		X		Lấy tên bãi xe khi biết ID của bãi xe đó
22	createNewRating	X			X	Tạo mới 1 đánh giá
23	createNewParkingLot	X	X		X	Tạo mới 1 bãi xe
24	addEmployeeOfParkingLot	X	X		X	Thêm tài khoản nhân viên vào danh sách tài khoản nhân viên quản lý bãi xe
25	removeEmployeeOfParkingLot	X	X		X	Loại bỏ tài khoản nhân viên ra khỏi danh sách tài khoản nhân viên quản lý bãi xe
26	checkEmployeeAlreadyManageParkingLot	X	X		X	Kiểm tra xem tài khoản nhân viên đó có quản lý bãi xe nào hay không
27	getEmployeeManageParkingLotIdList	X	X		X	Lấy danh sách nhân viên quản lý bãi xe đó
28	getParkingLotManagedByEmployee	X			X	Lấy dữ liệu của bãi xe mà được quản lý bởi nhân viên đó

4.11.2.4. Booking Service API

#	API / gRPC service	Protected	Admin only	Internal use	External use	Mô tả
1	createBooking	X	X	X		Tạo mới booking
2	updateBookingStatus	X	X	X		Cập nhật trạng thái của booking
3	deleteBookingById	X	X		X	Xóa booking

#	API / gRPC service	Protected	Admin only	Internal use	External use	Mô tả
4	countAllBookingGroupByStatus	X	X		X	Gom nhóm tất cả booking theo trạng thái để đếm
5	getCustomerOnGoingBooking	X			X	Lấy dữ liệu booking hiện tại của customer
6	checkCustomerHasOnGoingBooking	X			X	Kiểm tra xem customer có đang có booking diễn ra hay không
7	generateBookingQrCode	X			X	Tạo mã QR từ UUID của booking
8	finishBooking	X		X	X	Kết thúc booking
9	getAllBooking	X			X	Lấy dữ liệu tất cả booking, có hoặc không có filter, có pagination
10	countAllBooking	X	X		X	Đếm tất cả booking, có hoặc không có filter
11	getAllBookingOfCustomer	X	X		X	Lấy dữ liệu tất cả booking của 1 khách hàng, có pagination, nếu không truyền ID của khách hàng, ID khách hàng sẽ được parse từ chuỗi JWT string đi kèm request
12	countAllBookingOfCustomerById	X	X		X	Đếm tất cả booking của 1 khách hàng, cần truyền ID của khách hàng
13	countAllBookingOfCustomerByAuthorizationHeader	X			X	Đếm tất cả booking của 1 khách hàng, ID khách hàng sẽ được parse từ JWT string đi kèm request
14	getAllBookingOfParkingLot	X			X	Lấy dữ liệu tất cả booking của 1 bãi xe, có hoặc không có filter, có pagination
15	countAllBookingOfParkingLot	X			X	Đếm tất cả booking của 1 bãi xe, có hoặc không có filter
16	getAllOnGoingBookingOfParkingLot	X			X	Lấy dữ liệu tất cả booking đang diễn ra của bãi xe
17	countAllOnGoingBookingOfParkingLot	X			X	Đếm số booking đang diễn ra của bãi xe
18	getBookingDetailByBookingId	X			X	Lấy dữ liệu của booking thông qua booking UUID
19	countAllBookingOfParkingLotGroupByStatus	X			X	Đếm số booking của bãi xe, gom nhóm theo trạng thái hiện tại của booking
20	createOrUpdateBookingRating	X			X	Đánh giá hoặc cập nhật đánh giá booking sau khi đã hoàn thành
21	deleteBookingRatingByBookingId	X			X	Xóa đánh giá booking, biết UUID

#	API / gRPC service	Protected	Admin only	Internal use	External use	Mô tả
22	countAllBookingRatingOfParkingLotGroupByRating	X			X	Đếm số lượt booking rating của bãi xe theo rating (biết ID bãi xe)
23	countAllBookingRatingOfParkingLot	X			X	Đếm số lượt booking rating của bãi xe, có hoặc không có filter
24	getAllBookingRatingOfParkingLot	X			X	Lấy dữ liệu tất cả booking rating của bãi xe, có hoặc không có filter, có pagination
25	countAllBookingOfParkingLotInDay	X			X	Đếm số lượng booking của bãi xe trong ngày đó, cần cung cấp ngày và ID bãi xe
26	getAllBookingOfParkingLotInDay	X			X	Lấy dữ liệu tất cả booking của bãi xe trong ngày đó, có pagination, cần cung cấp ngày và ID bãi xe
27	countAllBookingOfParkingLotFromOneDayToAnotherDay	X			X	Đếm tổng số booking của bãi xe từ ngày này qua ngày khác, cần cung cấp 2 ngày và ID bãi xe
28	getAllBookingOfParkingLotFromOneDayToAnotherDay	X			X	Lấy dữ liệu tất cả booking của bãi xe từ ngày này qua ngày khác, có pagination, cần cung cấp 2 ngày và ID bãi xe
29	countAllBookingByNumberLicensePlate	X	X		X	Đếm số lượng booking biết biển số xe của khách
30	getAllBookingByNumberLicensePlate	X	X		X	Lấy dữ liệu tất cả booking biết biển số xe của khách, có pagination

4.11.2.5. Contact Service API

#	API / gRPC service	Protected	Admin only	Internal use	External use	Mô tả
1	checkUserOnlineById	X			X	Kiểm tra 1 user nào đó có đang online (có kết nối WebSocket với máy chủ) hay không
2	checkParkingLotOnlineByParkingLotId	X			X	Kiểm tra 1 bãi xe nào đó có đang online (có kết nối WebSocket với máy chủ) hay không
3	generateSocketConnectQrCode	X			X	Tạo mã QR để kết nối WebSocket với máy chủ

4.12. Vấn đề xác thực và phân quyền

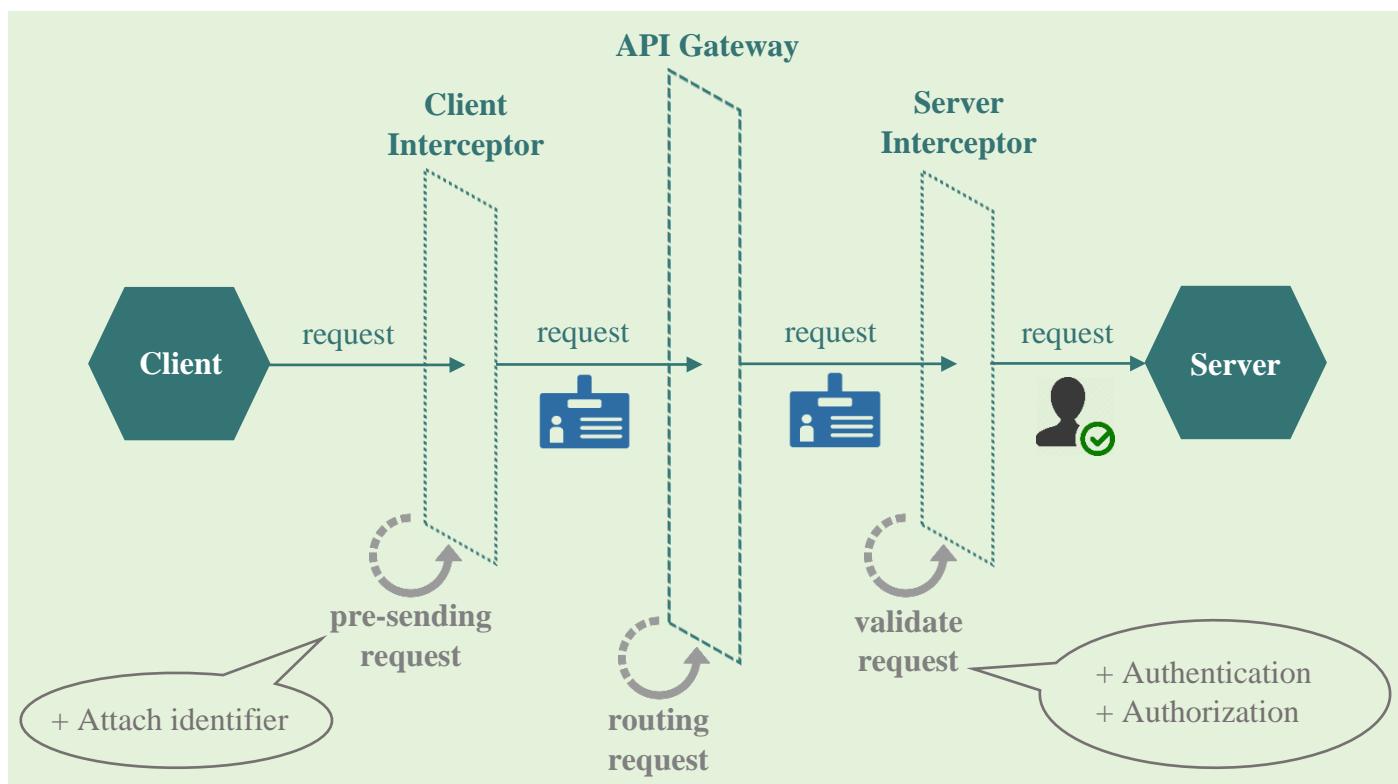
Để bảo vệ tài nguyên của mỗi hệ thống, việc xác thực và phân quyền là không thể thiếu với mỗi máy chủ cung cấp dịch vụ trên hệ thống mạng internet. Dưới đây là phần trình bày về giải pháp cũng như công nghệ mà nhóm em đã ứng dụng để giải quyết bài toán cấp thiết này.

4.12.1. Interceptor là gì

Để giao tiếp với server, client sẽ thực hiện gọi request đến server thông qua API public mà server đã cung cấp. Thông thường, đi kèm với mỗi request, chúng ta hay bổ sung thêm những trường cố định như thêm ‘authorization’ vào trong header để định danh client, v.v... Ngoài ra, trước khi server xử lý 1 request, sẽ có quá trình tiếp nhận request. Theo đó, việc xác thực và phân quyền người dùng sẽ diễn ra tại đây. Interceptor theo đúng nghĩa của nó, được sinh ra để can thiệp, là hàm trung gian trong quá trình giao tiếp giữa client và server. Nếu nhìn theo khía cạnh, có thể phân interceptor ra thành 2 loại:

- **Client Interceptor:** interceptor nằm bên phía client, thường dùng để thêm những trường cố định, những thông tin bổ sung cho request trước khi request được gửi đi từ client, thường thấy thông tin được gửi kèm sẽ là định danh của client, gửi thông qua header ‘authorization’ của request đó...
- **Server Interceptor:** interceptor nằm bên phía server, thường dùng để kiểm tra tính hợp lệ của request, trước khi chính thức được server xử lý. Nếu request không hợp lệ, server interceptor sẽ ngắt ngay lần gọi API đó, và trả về mã lỗi tương ứng.

Hình dưới đây sẽ mô tả cách thức chúng em sử dụng interceptor trong việc xác thực và phân quyền:



Hình 4.12.1. Minh họa cách thức sử dụng interceptor trong việc xác thực và phân quyền

4.12.2. JSON Web Token (JWT) là gì

JWT có thể hiểu như là 1 định danh mà server cấp cho client. Với mỗi lần gọi đến các API được bảo vệ, client cần phải đính kèm chuỗi JWT trong ‘authorization’ header của request để server có thể xác thực được chính xác client nào đang gọi đến để có những ứng xử phù hợp.

Các thông tin trong chuỗi JWT được định dạng bằng JSON. Nội dung của chuỗi gồm có 3 phần là header, payload và signature, mỗi phần được ngăn cách nhau bằng dấu “.”.

Hình dưới đây là ví dụ cơ bản của JWT (nguồn: <https://blog.larapulse.com/web/jwt>)



Hình 4.12.2. Minh họa nội dung của chuỗi JWT

4.12.3. Xác thực và phân quyền với JWT

Sử dụng Interceptor kết hợp cùng JWT là giải pháp nhóm em đã lựa chọn để giải quyết vấn đề xác thực và phân quyền. Quá trình này sẽ có chút khác nhau đối với 2 phương thức giao tiếp là WebSocket và gRPC, tuy nhiên, chúng đều có điểm tương đồng là trước tiên client phải được cấp chuỗi JWT thông qua việc đăng nhập hoặc làm mới token.

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzUxMiJ9.eyJqdGkiOiI1NDIzOD
MwNDUwNTUzODk0MjE4QDE10Tc00DEzNjM20DIiL
CJpc3Mi0iJ3d3cuc2FpZ29ucGFya2luZy53dGYi
LCJyb2xlIjoiUEFSS010R19MT1RfRU1QTE9ZRUU
iLCJmYWMi0jk2MywiY2xhc3NpZmljYXRpb24i0i
JBQ0NFU1NfVE9LRU4iLCJzdWIi0iI1NDIz0DMwN
DUwNTUzODk0MjE4IiwiaWF0IjoxNTk3NDgxMzYz
LCJleHAi0jE10Tc00DMxNjN9.-4LSEevHFsdxmq
cWs-3rdMBgy7aeswLSngfWUzIQIp2C-
qLujBucSNWEoMqyzSW6y_75CYZ2wughu0MB13wc
Rg
```

Decoded EDIT THE PAYLOAD AND SECRET

```

HEADER: ALGORITHM & TOKEN TYPE
{
    "alg": "HS512"
}

PAYLOAD: DATA
{
    "jti": "5423830450553894218@1597481363682",
    "iss": "www.saigonparking.wtf",
    "role": "PARKING_LOT_EMPLOYEE",
    "fac": 963,
    "classification": "ACCESS_TOKEN",
    "sub": "5423830450553894218",
    "iat": 1597481363,
    "exp": 1597483163
}

VERIFY SIGNATURE
HMACSHA512(
    base64UrlEncode(header) + "." +
    base64UrlEncode(payload),
    your-256-bit-secret
)  secret base64 encoded

```

Hình 4.12.3. Minh họa nội dung 1 access token của hệ thống saigonparking

4.12.3.1. Sử dụng JWT với gRPC

Đối với gRPC, việc xác thực và phân quyền sẽ diễn ra như sau:

Về phía client, client sẽ được phân thành **2 loại**: gRPC-Web client và gRPC client (non-gRPC-Web client). Sở dĩ được phân như vậy do gRPC-Web client khác với gRPC client thông thường: hiện phiên bản Web chưa hỗ trợ cài đặt client interceptor và hơn hết, gRPC-Web request phải đi qua trung gian là proxy server (mặc định là Envoy) để chuyển đổi protocol từ gRPC-Web thành gRPC thường và ngược lại.

- Với **gRPC client**: client sẽ tiến hành cài đặt 1 client interceptor với nội dung: trước khi mỗi 1 request được gửi đi từ phía ứng dụng của client, client interceptor sẽ đính kèm chuỗi JWT trong ‘authorization’ header của request đó.
- Với **gRPC-Web client**: việc đính kèm chuỗi JWT trong ‘authorization’ header phải thực hiện thủ công bằng cách tự đính kèm vào request mỗi khi thực hiện gọi API đến server do không có interceptor hỗ trợ.

Về phía server, server cũng tiến hành cài đặt 1 server interceptor với nội dung: với mỗi request được gửi đến server, server interceptor sẽ kiểm tra tính hợp lệ của request thông qua việc phân giải chuỗi JWT để có được định danh của client, user role tương ứng của client, trước khi request được server chính thức xử lý.

Lần gọi API sẽ ngắt ngay khi phát hiện request không hợp lệ (chuỗi JWT sai format, sai chữ ký hay user role không có quyền gọi API đó) và trả về cho client mã lỗi tương ứng. Nếu tất cả đều hợp lệ, server sẽ xử lý request đó và trả response về cho client kèm theo status thành công.

4.12.3.2. Sử dụng JWT với WebSocket

Đối với WebSocket, việc xác thực và phân quyền sẽ diễn ra như sau:

Về phía client, để thiết lập kết nối socket mới, client sẽ tiến hành gửi HTTP request để handshake với server đồng thời upgrade connection thành WebSocket. Việc cần làm là phải đính kèm chuỗi JWT vào trong request được gửi đi ở bước này. Việc này sẽ khác nhau đối với **2 loại** client phân biệt là client web và client mobile.

- Với **mobile client**, tương tự như gRPC, client sẽ đính kèm JWT vào ‘authorization’ header của request.
- Với **web client**, chúng em đã khảo sát trên nhiều thư viện và cộng đồng, kết quả là hiện các thư viện kết nối WebSocket phiên bản Web đều chưa hỗ trợ việc đính kèm header vào handshake request. Giải pháp chúng em đưa ra ở đây là đính kèm chuỗi JWT lên trên request URL. Theo đó, URL sẽ có dạng như sau:

`wss://<gateway_host>:<gateway_port>/contact/web?token=<JWT_string>`

Ví dụ: `wss://saigonparking.wtf:8443/contact/web?token=abcdefghijklmnopqrstuvwxyz`

Về phía server, server sẽ tiến hành cài đặt 1 handshake interceptor với nội dung: khi request handshake được gửi đến server, handshake interceptor sẽ kiểm tra tính hợp lệ của request thông qua việc phân giải chuỗi JWT để có được định danh của client, user role tương ứng của client, trước khi đồng ý mở kết nối với client.

Nếu có lỗi xảy ra trong quá trình can thiệp handshake request, ví dụ như chuỗi JWT không hợp lệ hay user X với role Y không có quyền mở kết nối socket với server, Handshake Interceptor sẽ ngắt ngay request handshake đó và trả về mã lỗi tương ứng cho client.

Handshake interceptor cũng được cài đặt thành **2 loại**, tương ứng với web client và mobile client.

Chương 5 TRIỂN KHAI THỰC NGHIỆM

Nội dung của chương 5 trình bày:

- Tổng quan những công nghệ sử dụng trong dự án (giới thiệu công nghệ, ưu điểm của công nghệ đó, tại sao lại dùng chúng vào trong dự án)
- Triển khai các công nghệ đó trong dự án như thế nào
- Giới thiệu các chức năng của hệ thống

5.1. Lựa chọn công nghệ

5.1.1. Tổng quan công nghệ sử dụng trong dự án

Languages	
IDEs	
Dependencies Management	
Framework	
Coding support	
Database, ORM	
Communication	
Communication Support	
Deployment	
Plugins	

Hình 5.1.1. Tổng quan công nghệ sử dụng trong dự án saigonparking

5.1.2. Java

- “Là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì bien dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.” [3]

- Những ưu điểm của Java so với các ngôn ngữ khác:

+ **Hiệu năng cao** Java đạt được hiệu suất cao thông qua việc sử dụng byteCode có thể dễ dàng dịch sang mã máy. Với việc sử dụng các trình biên dịch JIT (Just-In-Time), Java mang lại hiệu năng cao.

+ **Năng động** - Java năng động hơn C hoặc C++ vì nó được thiết kế để thích nghi với môi trường đang phát triển.

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

+ **Nền tảng độc lập** - Không giống nhiều ngôn ngữ lập trình khác, khi Java được biên dịch, nó không được biên dịch thành ngôn ngữ máy nền tảng cụ thể, thay vào mã byte - nền tảng độc lập. Mã byte này được thông dịch bởi máy ảo (JVM) trên nền tảng nào đó mà nó đang chạy.

+ **Bảo mật** - Với tính năng an toàn của Java, nó cho phép phát triển các hệ thống không có virus, giả mạo. Các kỹ thuật xác thực dựa trên mã hoá khóa công khai.

+ **Mạnh mẽ** - Java có một hệ thống quản lý bộ nhớ mạnh. Nó giúp loại bỏ lỗi vì nó kiểm tra Code trong quá trình biên dịch và runtime.

+ **Đa luồng** - Java hỗ trợ nhiều luồng thực thi, bao gồm một tập hợp các nguyên hàm đồng bộ hóa. Điều này làm cho lập trình với các chủ đề dễ dàng hơn nhiều.

+ **Thông dịch** - Mã byte Java được dịch trực tiếp tới các máy tính gốc và không được lưu trữ ở bất cứ đâu.

5.1.3. Spring Boot

- Spring Boot là một module của Spring Framework. Nó giúp cho các lập trình viên đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.

- Ưu điểm của Java Spring Boot:

- Có các tính năng của Spring Framework.
- Giúp tạo được ứng dụng độc lập dựa trên Spring, có thể tự chạy được (java -jar)
- Cung cấp nhiều plugin
- Chuẩn cho Microservices (Cloud support; giảm việc setup, config; các thư viện hỗ trợ...)
- Spring boot không yêu cầu cấu hình XML và không sinh code cấu hình

5.1.4. Hibernate

” - *Hibernate là một thư viện ORM (Object Relational Mapping) mã nguồn mở giúp lập trình viên viết ứng dụng Java có thể map các objects với hệ quản trị cơ sở dữ liệu quan hệ, và hỗ trợ thực hiện các khái niệm lập trình hướng đối tượng với cơ sở dữ liệu quan hệ*

- *Hiểu ngắn gọn thì Hibernate sẽ là một layer đứng trung gian giữa ứng dụng và database, và chúng ta sẽ giao tiếp với Hibernate thay vì giao tiếp với database* ” [4]

- Ưu điểm khi sử dụng Hibernate:

- Tìm kiếm và sắp xếp nhanh.
- Làm việc được với dữ liệu lớn.

- Làm việc trên nhóm dữ liệu
- Truy vấn cơ sở dữ liệu độc lập
- Tạo bảng tự động
- Đơn giản lệnh join phức tạp

5.1.5. Consul

- Là một giải pháp mạng lưới dịch vụ cung cấp đầy đủ các tính năng với truy tìm dịch vụ, configuration. Mỗi tính năng có thể được sử dụng riêng biệt hay sử dụng tất cả để xây dựng một mạng lưới dịch vụ. Consul yêu cầu dữ liệu và hỗ trợ cho cả proxy và các model tích hợp.

5.1.6. RabbitMQ

- Là một phần mềm nơi các queue được định nghĩa, phục vụ cho ứng dụng với mục đích vận chuyển một hoặc nhiều message.

- Trong phạm vi đề tài, chúng em sử dụng RabbitMQ vì những lợi ích mà nó mang lại phù hợp với yêu cầu và phạm vi quản lý các request được gửi đến server trong đồ án. Đơn cử là tác vụ xử lý các request từ khách hàng về server. Queue sẽ lưu trữ những request đó, không bị thiếu sót request nào khi số lượng của chúng trở nên vô cùng lớn.

5.1.7. Kong

" - Là một open-source API Gateway và platform, Kong được viết bằng ngôn ngữ Lua và xây dựng trên NGINX. Kong hỗ trợ nhiều plugins giúp cho việc triển khai microservices dễ dàng hơn như authentication, rate-limiting, transformation, logging, ... Nhà phát triển có thể tự viết plugins cho Kong bằng Lua tùy vào nhu cầu sử dụng" [5]

- Ưu điểm của Kong:

- + Mở rộng dễ dàng
- + Hiệu năng tốt
- + Nhiều plugins
- + Miễn phí

5.1.8. Envoy

- Là một dịch vụ proxy được thiết kế cho các ứng dụng đám mây (Cloud-Native Applications)
 - Trong phạm vi đồ án, Envoy đóng vai trò là proxy tiếp nhận và chuyển tiếp các protocol (giao thức) vào gateway

5.1.9. Docker

- Là một nền tảng để cung cấp cách để xây dựng, triển khai và chạy ứng dụng dễ dàng hơn bằng cách sử dụng các containers (trên nền tảng ảo hóa). Khi cần deploy lên bất kỳ server nào chỉ cần chạy container của Docker thì ứng dụng sẽ được khởi chạy ngay lập tức. Ban đầu Docker được viết bằng Python, hiện tại đã chuyển sang GoLang.

- Ưu điểm của Docker:

- + Tốc độ xử lí nhanh
- + Có thể khởi chạy container trên mọi hệ thống.
- + Container có thể build và loại bỏ nhanh hơn máy ảo.
- + Dễ dàng thiết lập môi trường làm việc.

- + Thích hợp cho các hệ thống được triển khai theo kiến trúc Microservices

5.1.10. NGINX

- NGINX, đọc là “engine-ex,” là một phần mềm web server mã nguồn mở sử dụng kiến trúc đơn luồng, hướng sự kiện. Ban đầu nó dùng để phục vụ web HTTP. Tuy nhiên, ngày nay nó cũng được dùng làm reverse proxy, HTTP load balancer và email proxy như IMAP, POP3, và SMTP.

- Ưu điểm của NGINX:

- “+ Có khả năng xử lý cùng một lúc hơn 10.000 kết nối với bộ nhớ thấp
- + Dùng bộ nhớ đệm Cache để tăng tốc Proxy ngược, cân bằng tải đơn giản và khả năng chịu lỗi
- + Hỗ trợ tăng tốc với bộ nhớ đệm của WSGI, SCGI, FastCGI và các máy chủ Memcached
- + Có cấu hình linh hoạt và khả năng lưu lại nhật ký truy vấn
- + Hạn chế tỷ lệ đáp ứng truy vấn
- + Tương thích và hỗ trợ IPv6
- + Hỗ trợ WebSockets” [6]

5.1.11. NPM

” - NPM (Node Package Manager) là một công cụ quản lý các thư viện lập trình Javascript cho Node.js. Trong cộng đồng ngôn ngữ Javascript, các lập trình viên đã chia sẻ hàng trăm nghìn thư viện với các đoạn code đã thực hiện sẵn một chức năng nào đó. Nó 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”. [7]

- Về cơ bản, NPM khá giống với Maven, sự khác biệt ở đây là NPM quản lý các thư viện Javascript trong khi đó Maven quản lý các thư viện Java.

5.1.12. Maven

- Là chương trình quản lý dự án được sử dụng nhiều bởi các lập trình viên Java, nó thực hiện các công việc như khởi tạo project, biên dịch, đóng gói, chạy test,... một cách tự động và nhanh chóng. Nó đặc biệt hữu dụng với các dự án tầm trung và lớn.

5.1.13. Gradle

- Là một công cụ xây dựng tự động hóa để phát triển phần mềm đa ngôn ngữ. Gradle cung cấp một mô hình linh hoạt có thể hỗ trợ toàn bộ vòng đời phát triển của mã nguồn từ biên dịch, kiểm tra cho đến đóng gói mã để xuất bản

- Lý do lựa chọn: ”Gradle được tích hợp với Android Studio hỗ trợ quản lý dependency, multi-project. Đồng thời tích hợp các test services phổ biến như AppThwack, TestDroid và Manymo. Ngoài ra, có thể tạo nhiều APK từ 1 module và upload chúng lên Google Play và các app store khác với các phiên bản khác nhau (free, paid,...)” [8]

5.1.14. ReactJS

- Là một framework dùng để xây dựng giao diện người dùng (User Interface-UI) với concept đang rất trending hiện nay là Single Page Application.). React là một mã nguồn mở được xây dựng bởi chính Facebook. Rất nhiều thương hiệu hàng đầu thế giới đang sử dụng React, như là Facebook và Instagram.

Single Page Application: là một ứng dụng hoạt động trên browser và không cần bắt buộc phải reload lại khi sử dụng. Hiểu một cách đơn giản, thì toàn bộ resource của web bao gồm các file CSS, Javascript, master

layout hay cấu trúc web page sẽ được load lần đầu tiên khi chúng ta bắt đầu duyệt một website A nào đó. Ở những lần sau, khi chuyển trang khác, client sẽ gửi những ajax request để lấy dữ liệu cần thiết (thường là phần nội dung). Việc này mang đến trải nghiệm cho người dùng web tốt hơn, giảm thời gian phải load lại toàn bộ trang web công kênh, tiết kiệm băng thông cũng như thời gian chờ đợi. Việc này là trái ngược hoàn toàn với trang web truyền thống (multi page application) khi toàn bộ trang web phải load lại mỗi khi chuyển trang.

- **Ưu điểm của Reactjs:** Reactjs tạo ra cho chính nó DOM ảo (Virtual DOM) – nơi mà các component thực sự tồn tại trên đó. DOM ảo tạo ra bản cache cấu trúc dữ liệu của ứng dụng trên bộ nhớ. Sau đó, ở mỗi vòng lặp, nó liệt kê những thay đổi và sau đó là cập nhật lại sự thay đổi trên DOM của trình duyệt một cách hiệu quả. Điều này giúp cải thiện tốc độ tải web rất nhiều vì trang web chỉ render tất cả dữ liệu ở lần đầu tiên. Vào các lần tải sau, Reactjs chỉ render những content nào thực sự thay đổi trên trang web (thường là phần nội dung (body) của trang web).

5.1.15. Microsoft SQL Server

“- Là một phần mềm được phát triển bởi Microsoft dùng để lưu trữ dữ liệu dựa trên chuẩn RDBMS, và nó cũng là một hệ quản trị cơ sở dữ liệu quan hệ đối tượng (ORDBMS).

- SQL Server cung cấp đầy đủ công cụ để quản lý, từ giao diện GUI cho đến việc sử dụng ngôn ngữ truy vấn SQL. Ngoài ra điểm mạnh của nó là Microsoft có khai thác nền tảng kết hợp hoàn hảo với SQL Server như ASP.NET, C# xây dựng Winform, bởi vì nó hoạt động hoàn toàn độc lập.”[9]

5.1.16. Github

- Là một hệ thống quản lý dự án và phiên bản code, hoạt động giống như một mạng xã hội cho lập trình viên. Các lập trình viên có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để có thể làm việc.

- Một số tính năng nổi bật của Github:

+ Là công cụ giúp quản lý source code tổ chức theo **dạng dữ liệu phân tán**.

+ Giúp **đồng bộ source code** của team lên 1 server.

+ Hỗ trợ các thao tác **kiểm tra source code** trong quá trình làm việc (diff, check modifications, show history, merge source, ...)

5.1.17. Amazon Web Service

5.1.17.1. Amazon RDS

Amazon Relational Database Service (Amazon RDS) là một dịch vụ được quản lý giúp bạn dễ dàng thiết lập, vận hành và thay đổi quy mô cơ sở dữ liệu quan hệ trên đám mây. Dịch vụ này cung cấp tính năng hiệu quả về chi phí và có thể thay đổi quy mô, trong khi vẫn quản lý được các tác vụ quản trị cơ sở dữ liệu tiêu tốn nhiều thời gian, giúp bạn thoải mái tập trung vào các ứng dụng và công việc của bạn

5.1.17.2. Amazon S3

Amazon Simple Storage Service (Amazon S3) là một dịch vụ lưu trữ đối tượng cung cấp khả năng thay đổi theo quy mô, tính khả dụng của dữ liệu, bảo mật và hiệu năng hàng đầu trong lĩnh vực. Điều này có nghĩa là khách hàng thuộc mọi quy mô và lĩnh vực có thể sử dụng dịch vụ này để lưu trữ và bảo vệ bất kỳ lượng dữ liệu nào cho nhiều trường hợp sử dụng khác nhau, chẳng hạn như trang web, ứng dụng di động, sao lưu và khôi phục, lưu trữ, ứng dụng doanh nghiệp, thiết bị IoT và phân tích dữ liệu lớn. Amazon S3 cung cấp các tính năng quản lý dễ sử dụng, nhờ đó, bạn có thể tổ chức dữ liệu và cấu hình các kiểm soát truy cập được tinh chỉnh để đáp ứng yêu cầu cụ thể của doanh nghiệp, tổ chức và yêu cầu về tuân thủ. Amazon S3 được

thiết kế để đảm bảo độ bền 99,99999999% (11 9's) và lưu trữ dữ liệu của hàng triệu ứng dụng cho các công ty trên toàn thế giới

5.1.18. Google Cloud Platform

Google Cloud Platform được viết tắt là GCP là 1 nền tảng của điện toán đám mây cho phép tổ chức và các doanh nghiệp tạo ra, xây dựng và hoạt động những ứng dụng của mình trên hệ thống google tạo ra. Những ứng dụng phổ biến của Google sử dụng Google Cloud Platform hiện đang rất phổ biến chính là: Youtube, Chrome, Google Apps, Google Maps, Google Search,...

5.1.18.1. Google Maps API

Google Maps là dịch vụ bản đồ trực tuyến phổ biến nhất hiện nay, được phát hành và hỗ trợ bởi Google, cho phép người dùng có thể tìm đường, tìm địa điểm trên ứng dụng này (website và app). Bên cạnh đó Google Maps API được Google giới thiệu và hướng dẫn các lập trình sử dụng nền tảng Maps này trên các ứng dụng website và app mobile

5.1.18.2. Google Places API

Google Place API là một trong hàng trăm API tuyệt vời mà Google cung cấp cho nhà phát triển phần mềm, với thông tin chi tiết về hơn 100 triệu địa điểm. Google Place API gồm có 3 công cụ: Place Picker, Place Autocomplete, Place Details: Place Picker cho phép người dùng tìm kiếm các địa điểm xung quanh vị trí hiện tại của họ. Place Autocomplete giúp người dùng dễ dàng nhập tên địa điểm và địa chỉ, tự động hoàn thành các truy vấn của người dùng khi họ nhập. Place Detail cho phép người dùng lấy thông tin chi tiết về một địa điểm như địa chỉ, số điện thoại, trang web, hình ảnh,...

5.1.18.3. Google Directions API

Google Maps Directions API là một dịch vụ tính toán đường đi giữa các vị trí. Qua đó bạn có thể xem hướng dẫn đường đi cụ thể cho một số phương thức vận tải, bao gồm lái xe, đi bộ hoặc đi xe đạp

5.1.18.4. Google Cloud Compute Engine

Google Compute Engine cung cấp các máy ảo có thể tùy chỉnh cao với các tính năng ưu việt nhất. Google Compute Engine cung cấp một loạt các tùy chọn tính toán có thể mở rộng cho phù hợp với nhu cầu của người dùng. Là dịch vụ toàn cầu, cân bằng tải, khả năng phục hồi đến các máy ảo cá thể đơn lẻ linh hoạt. Google Compute Engine cung cấp các máy ảo chạy trong các trung tâm dữ liệu sáng tạo của Google và mạng cáp quang toàn cầu. Tính toán Công cụ và công việc của Công cụ hỗ trợ cho phép mở rộng, cân bằng tải điện toán đám mây.

5.1.19. IDE

5.1.19.1. IntelliJ IDEA

IntelliJ IDEA là một trình IDE dùng để lập trình Java (nó cũng được sử dụng để lập trình một số ngôn ngữ khác như Node.js, python...)

5.1.19.2. Android Studio

Đây là IDE (Môi trường phát triển tích hợp) chính thức cho nền tảng Android, được phát triển bởi Google và được sử dụng để tạo phần lớn các ứng dụng mà bạn có thể sử dụng hàng ngày.

5.1.19.3. Visual Studio Code

Visual Studio Code (VS Code hay VSC) là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở chính là những ưu điểm vượt trội khiến VS Code ngày càng được ứng dụng rộng rãi

5.1.20. Các thư viện, tiện ích và công cụ hỗ trợ

5.1.20.1. SQL Server Management Studio

- SQL Server Management Studio (SSMS) là một môi trường tích hợp để quản lý bất kỳ cơ sở hạ tầng SQL nào, từ SQL Server đến Azure SQL Database. SSMS cung cấp các công cụ để cấu hình, giám sát và quản lý các phiên bản của SQL Server và cơ sở dữ liệu. Sử dụng SSMS để triển khai, giám sát và nâng cấp các data-tier-components, cũng như dùng để viết các câu truy vấn và các script.

- Ưu điểm của SSMS so với các công cụ trực quan khác dùng để viết truy vấn và quản lí cơ sở dữ liệu:

- + Dễ làm quen và sử dụng

- + Cài đặt ứng dụng miễn phí

- + Được bảo trợ bởi một trong những tập đoàn công nghệ lớn nhất toàn cầu (Microsoft)

5.1.20.2. Lombok

- Là một bộ thư viện viết bằng Java, giúp chúng ta generate code một cách tự động. Lombok được sử dụng để giảm mã cho các đối tượng model/ entity. Lombok sử dụng chú thích (annotation) để đánh dấu các phương thức cần generate. Ví dụ: @Getter được sử dụng để tạo getter cho các field, @Setter để tạo setter cho các field, ...

5.1.20.3. MapStruct

- Là một trình tạo mã giúp đơn giản hóa đáng kể việc ánh xạ giữa các kiểu Java Bean. Mã ánh xạ được tạo sử dụng các lệnh gọi phương thức đơn giản. **MapStruct** sinh ra để giải quyết vấn đề mapping dữ liệu giữa các Java Bean một cách tự động, thuận tiện và nhanh chóng.

5.1.20.4. SonarLint

- Một Plugin giúp phát hiện các vấn đề khi viết code. SonarLint giúp code Java được clean hơn, giảm thời gian review code, các lỗi code không cần thiết, nâng chất lượng dự án tốt hơn

5.1.20.5. AndroidLint

- AndroidLint hay còn gọi là Lint Tool là một công cụ phân tích code tĩnh với chức năng phát hiện và tối ưu các lỗi tiềm ẩn bên trong source code để từ đó cải thiện về tính chính xác, độ bảo mật, hiệu năng, khả năng tái sử dụng, cũng như đơn giản hóa source code giúp cho người khác dễ dàng hơn trong việc đọc hiểu source code của dự án.

5.1.20.6. ESLint

- Một Plugin giúp phát hiện các vấn đề khi viết code. ESLint giúp code Javascript được clean hơn, giảm thời gian review code, các lỗi code không cần thiết, nâng chất lượng dự án tốt hơn

5.1.20.7. SaveActions

- Một plugin của IntelliJ giúp tự động format code theo chuẩn, giúp code clean hơn, dễ nhìn hơn.

5.1.20.8. Multirun

- Một plugin của IntelliJ giúp cho lập trình viên có thể cấu hình thử tự khởi động của các ứng dụng, các service khi chạy (debug) code với IntelliJ do thường có sự phụ thuộc giữa các ứng dụng với nhau.

5.1.20.9. Docker Compose

- là công cụ dùng để định nghĩa và chạy multi Docker container, thay thế cho lệnh docker run từng container.

5.2. Lưu trữ mã nguồn với Github

Tên Repository	Địa chỉ Repository	Mô tả Repository
saigonparking	https://github.com/huynhthanhbinh/saigonparking	lưu trữ mã nguồn backend, DevOps và document dự án
saigonparking-web	https://github.com/huynhthanhbinh/saigonparking-web	dùng để lưu trữ mã nguồn ứng dụng web
saigonparking-android	https://github.com/huynhthanhbinh/saigonparking-android	lưu trữ mã nguồn ứng dụng di động dành cho tài xế xe ô tô
saigonparking-qrscanner	https://github.com/huynhthanhbinh/saigonparking-qrscanner	lưu trữ mã nguồn ứng dụng di động giả lập quét mã QR

5.3. Đóng gói dự án sử dụng Docker, Docker Compose và NGINX

Có tổng cộng **11** module đã được đóng gói thành các docker image bao gồm:

- 6 microservice (Auth, Booking, Contact, Mail, Parking Lot, User)
- NGINX Web Server sau khi đã tùy biến để phù hợp với hệ thống saigonparking (*)
- Envoy Proxy sau khi đã tùy biến để phù hợp với hệ thống saigonparking
- Kong Gateway sau khi đã tùy biến để phù hợp với hệ thống saigonparking
- Consul Service Discovery sau khi đã tùy biến để phù hợp với hệ thống saigonparking
- RabbitMQ sau khi đã tùy biến để phù hợp với hệ thống saigonparking

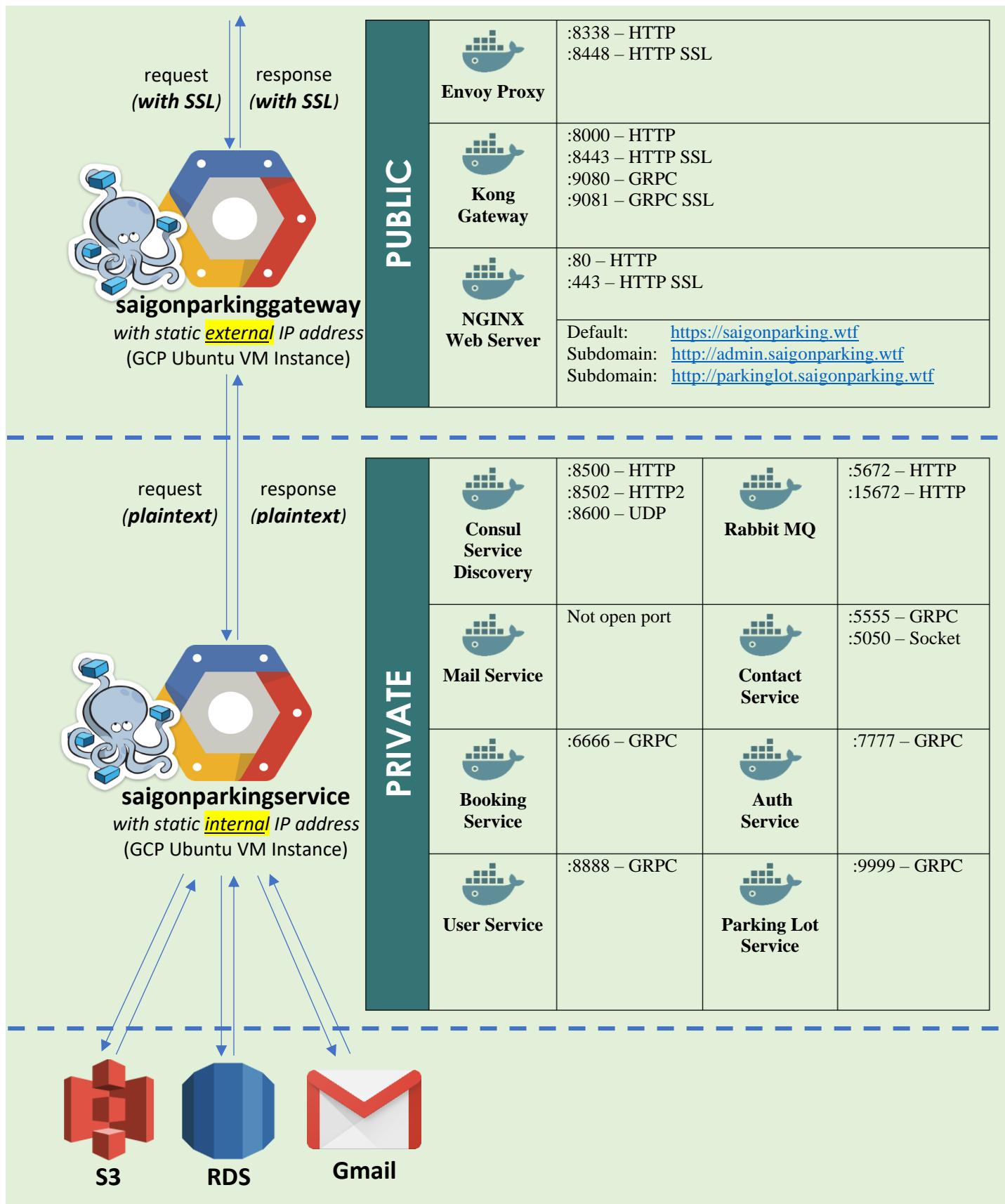
(*): 3 Web App sẽ được đóng gói lại thành 1 image duy nhất sử dụng image của **NGINX**, với **3** domain:

#	Domain	Đường dẫn
1	Domain chính (website chính thức của dự án)	https://saigonparking.wtf https://www.saigonparking.wtf
2	Subdomain admin (website dành cho quản trị viên hệ thống)	http://admin.saigonparking.wtf
3	Subdomain parkinglot (website giả lập ứng dụng dành cho bãi xe)	http://parkinglot.saigonparking.wtf

Sau khi đã đóng gói thành công, chúng em viết sẵn file docker-compose để cấu hình tất cả biến môi trường, bind tất cả các port cần thiết, cũng như thứ tự để start các service. Nhờ có docker-compose, việc start, shut down hệ thống và health check các service sẽ vô cùng dễ dàng. Dưới đây là các lệnh thường được dùng:

#	Lệnh / Thao tác	Mô tả
1	docker-compose up	Khởi động hệ thống (khởi động tất cả service)
2	docker-compose ps	Health Check tất cả service được định nghĩa trong compose
3	Ctrl + C (khi các service đang chạy)	Tắt hệ thống (tắt tất cả microservice)
4	docker-compose rm	Xóa tất cả container đã chạy định nghĩa trong compose

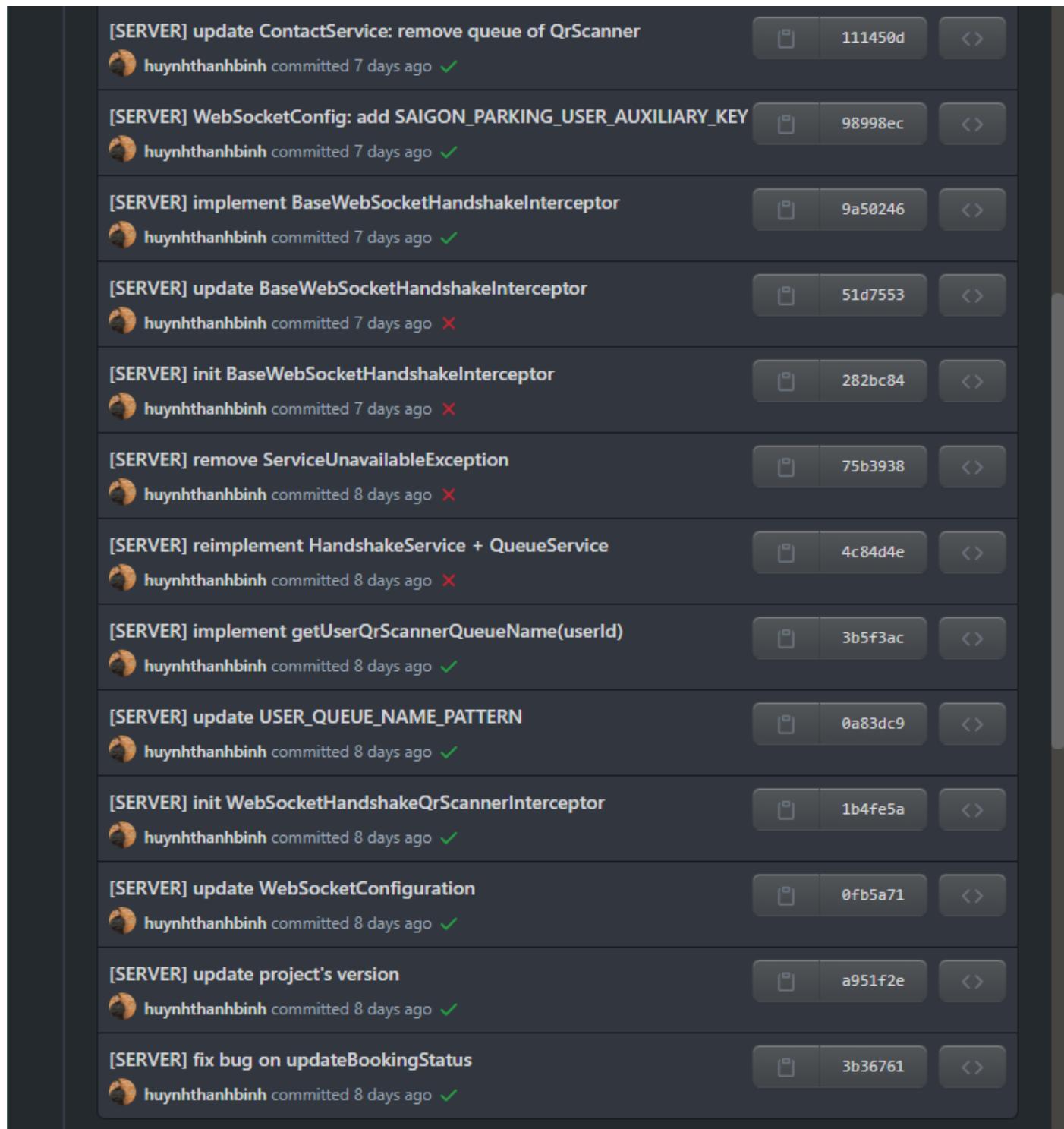
5.4. Triển khai dự án với Google Cloud Platform và Amazon Web Service



Hình 5.4. Minh họa triển khai hệ thống saigonparking lên cloud-server sử dụng AWS và GCP

5.5. Sử dụng Github Actions cho CI/CD

Có rất nhiều tool để CI/CD trong đó phải kể đến Jenkins, Travis CI, Circle CI, Gitlab CI/CD, Azure DevOps. Tuy nhiên, do chúng em sử dụng Github để lưu trữ mã nguồn nên để tiện sử dụng và dễ cấu hình, chúng em lựa chọn Github Actions để thực hiện CI/CD.



Hình 5.5.a. Minh họa trạng thái của Github Actions trong lịch sử commit của Github saigonparking

Search or jump to...

Pull requests Issues Marketplace Explore

huynhthanhbinh / saigonparking Private

Unwatch 1

Code Issues Pull requests Actions Projects Wiki Security

[SERVER] update Kong gateway's version

master · a633ba8

saigonparking-master / build-and-deploy

on: push

Search logs

build-and-deploy

- ▶ ✓ Set up job
- ▶ ✓ Run actions/checkout@v2
- ▶ ✓ Set up JDK 1.8
- ▶ ✓ Check project version
- ▶ ✓ Clean with Maven
- ▶ ✓ Build with Maven
- ▶ ✓ Login Docker Hub
- ▶ ✓ Build parkinglot-service image
- ▶ ✓ Build user-service image
- ▶ ✓ Build auth-service image
- ▶ ✓ Build mail-service image
- ▶ ✓ Build contact-service image
- ▶ ✓ Build booking-service image
- ▶ ✓ Build message-queue image
- ▶ ✓ Build service-discovery image
- ▶ ✓ Build api-gateway image
- ▶ ✓ Build web-proxy image
- ▶ ✓ Check Docker images
- ▶ ✓ Push parkinglot-service image
- ▶ ✓ Push user-service image

Hình 5.5.b. Minh họa 1 phiên làm việc của Github Actions với 1 commit cụ thể (build + test + deploy)

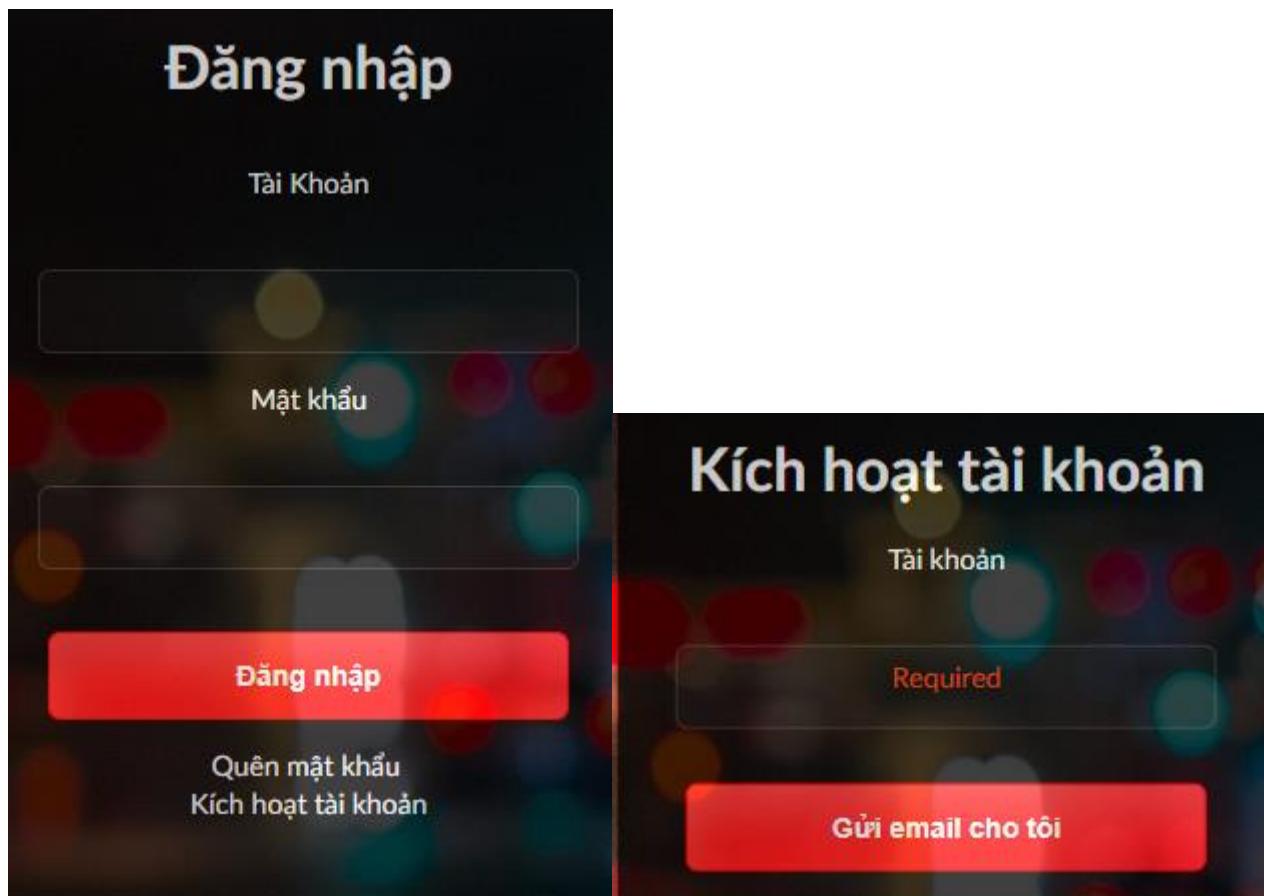
5.6. Giới thiệu các chức năng của hệ thống

5.6.1. Website chính thức của dự án

5.6.1.1. Đăng nhập

Người dùng cần nhập đầy đủ thông tin (Tài khoản và Mật khẩu) khi tạo được để có thể đăng nhập và sử dụng nhiều chức năng khác.

Trong đó, nếu tài khoản người dùng chưa được kích hoạt, người dùng có thể nhấn vào kích hoạt để thực hiện nhập tên tài khoản và nhấn xác nhận, sau đó, hệ thống sẽ gửi về email được dùng để tạo tài khoản đó 1 đường dẫn kích hoạt tài khoản



Hình 5.6.1.1.a Hình chức năng đăng nhập và kích hoạt tài khoản

5.6.1.2. Đăng ký

Người dùng cần nhập đủ các thông tin sau để có thể tiến hành đăng ký tài khoản

Lưu ý: Người dùng cần nhập đúng email tồn tại để hệ thống có thể gửi mail xác nhận kích hoạt tài khoản

Đăng ký

Họ Required

Tên Required

Tên tài khoản Required

Mật khẩu Required

Xác nhận mật khẩu Required

Email Required

Số điện thoại Required

Đăng ký

Đặt lại

Hình 5.6.1.2.a Hình chức năng đăng ký

Sau khi đăng ký thành công, hệ thống sẽ gửi mail về email đã được dùng để tạo tài khoản

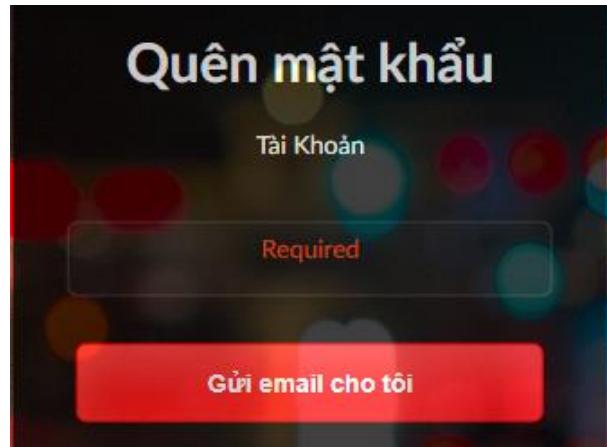


Hình 5.6.1.2.b Hình thông báo đăng ký thành công

5.6.1.3. Quên mật khẩu

Người dùng cần nhập chính xác tài khoản, hệ thống sẽ gửi email xác nhận đặt lại mật khẩu đến email mà người dùng sử dụng để tại tài khoản

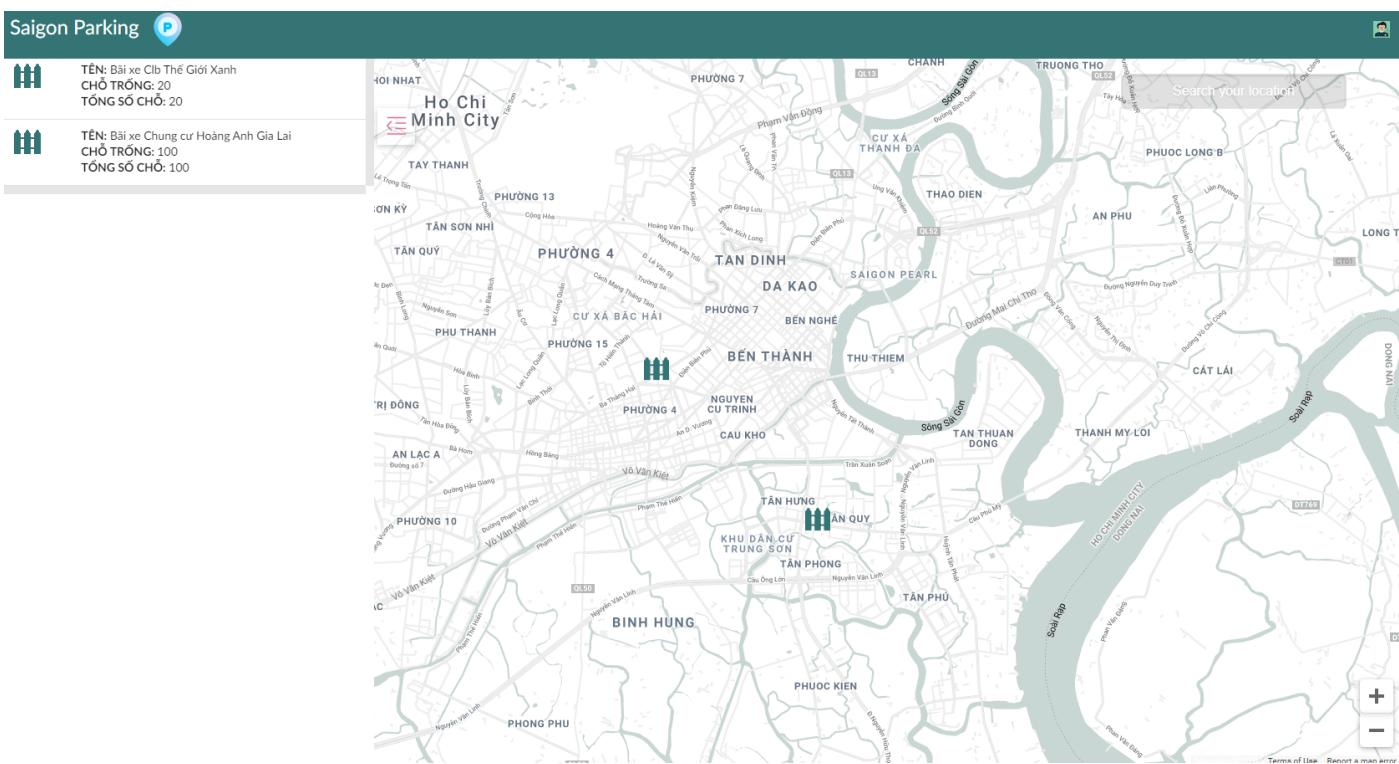
Lưu ý: Quên mật khẩu chỉ hoạt động đối với tài khoản đã được đăng ký và xác thực kích hoạt



Hình 5.6.1.3.a Hình chức năng quên mật khẩu

5.6.1.4. Xem vị trí các bãi xe

Sau khi đăng nhập thành công, người dùng có thể sử dụng được tính năng xem bản đồ và biết được vị trí của những bãi xe có trên hệ thống



Hình 5.6.1.4.a Hình bản đồ chính

5.6.1.5. Xem thông tin chi tiết bãi xe

Người dùng có thể chọn 1 bãi xe để xem thông tin chi tiết của bãi xe đó

The figure consists of two side-by-side screenshots of a mobile application for parking management.

Screenshot 1 (Left):

- ID:** 209
- xem bình luận**
- Rating: ★★★★☆
- Details:
 - NAME: Bãi xe đường Trần Bình Trọng
 - ADDRESS: Đường Trần Bình Trọng, Quận 5, Hồ Chí Minh
 - PHONE:
 - TYPE: 3
 - OPEN: 06:00:00
 - CLOSE: 23:55:56
 - AVAILABLE SLOT: 178
 - TOTAL SLOT: 178

Screenshot 2 (Right):

- ID:** 219
- xem bình luận**
- Rating: ★★★★☆
- Details:
 - NAME: Bãi xe đường Đông Du
 - ADDRESS: Đường Đông Du, Quận 1, Hồ Chí Minh
 - PHONE:
 - TYPE: 3
 - OPEN: 06:00:00
 - CLOSE: 23:55:56
 - AVAILABLE SLOT: 29
 - TOTAL SLOT: 29

Hình 5.6.1.5.a. Hình thông tin chi tiết bãi xe

5.6.1.6. Bình luận đánh giá

Người dùng có thể xem được bình luận của từng bãi xe và có thể đánh giá dịch vụ của bãi

This screenshot shows a list of comments for a parking lot, with a total of 24 comments displayed.

Commenter	Rating	Date
pvluc	★★★★☆	2020-06-23 18:57:00.0
đẹp	★★★★☆	2020-06-23 19:05:00.0
vmdang	★★★★☆	2020-06-23 19:18:00.0
tiện; có camera	★★★★☆	2020-06-23 19:18:00.0
vhai	★★★★☆	2020-06-23 19:18:00.0
sạch; đẹp; tiện	★★★★☆	2020-06-23 19:18:00.0
pttduy	★★★★☆	2020-06-23 19:05:00.0
gần với các quán ăn. nhiều chỗ	★★★★☆	2020-06-23 19:05:00.0
đậu	★★★★☆	2020-06-23 19:05:00.0
pvluc	★★★★☆	2020-06-23 19:05:00.0
có camera; ồn; rộng	★★★★☆	2020-06-23 19:05:00.0
vhai	★★★★☆	2020-06-23 19:05:00.0
bình thường. an toàn v.v..	★★★★☆	2020-06-23 19:05:00.0
tdphuc	★★★★☆	2020-06-23 19:18:00.0
có camera; ồn; rộng	★★★★☆	2020-06-23 19:18:00.0

This screenshot shows a rating and commenting interface.

COMMENT RATING

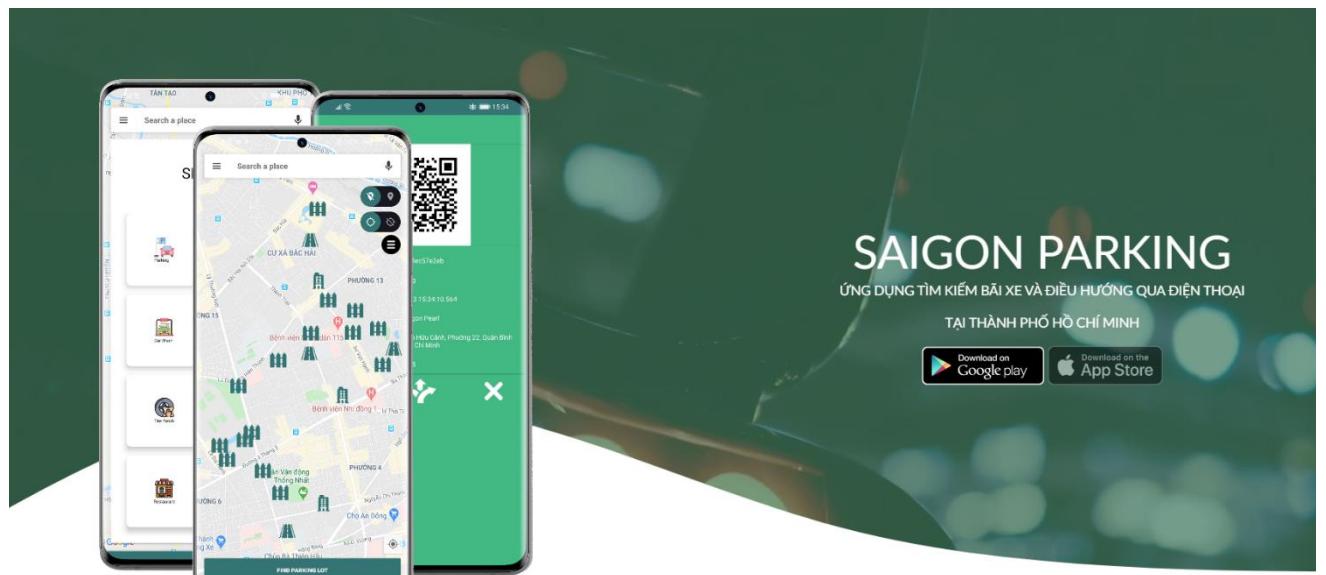
Rating scale: ★ ★ ★ ★ ☆

Text input field: 0/255

Submit button

Hình 5.6.1.6.a Hình chức năng đánh giá và xem đánh giá

5.6.1.7. Xem thông tin cơ bản về chức năng



TÍNH NĂNG

- Tim kiếm hiệu quả**: Tim kiếm các bãi giữ xe xung quanh nhanh chóng và hiệu quả.
- Tim kiếm xung quanh**
- Đặt chỗ**: Đặt chỗ nhanh chóng cho bãi xe Tân Hoàng Cầu.
- Xác nhận đến nơi dễ dàng**: Xác nhận đến nơi dễ dàng bằng mã QR.
- Contact us**: Các cách liên hệ bao gồm Email, Điện thoại, và Zalo.
- Team Developer**: Danh sách thành viên phát triển:
 - Xử lý và hoàn thành tính năng cho ứng dụng và website: Vũ Tường Bách (Website, Android Developer)
 - Xử lý dữ liệu và thực thi tính năng ở Backend: Huỳnh Thành Bình (Backend Developer)
 - Phân tích nghiệp vụ, kiểm thử và đánh giá: Phạm Việt Minh Đạo (Tester, BA)
 - Xử lý và hoàn thành tính năng cho ứng dụng và website: Đặng Đức Tài (Website, Android Developer)
 - Thiết kế giao diện website dành cho khách hàng, nhân viên quản lý bãi xe, và ứng dụng android: Vũ Hải (Website, Android Designer)

Không cần thiết phải đăng nhập, người dùng có thể xem được thông tin cơ bản của sản phẩm dự án, chức năng và thông tin liên hệ

Hình 5.6.1.7.a Hình trang chủ của hệ thống

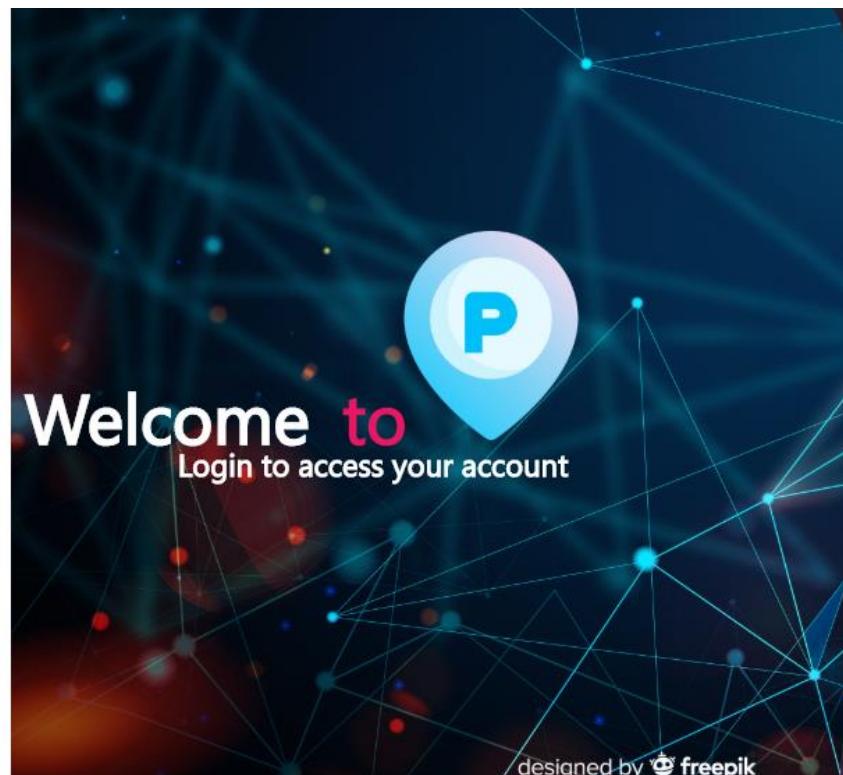
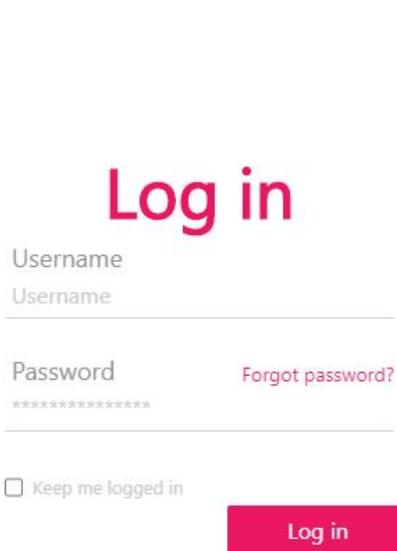
5.6.2. Website dành cho quản trị viên

5.6.2.1. Chức năng đăng nhập

Người dùng đăng nhập vào hệ thống quản trị viên bằng tên đăng nhập và mật khẩu đã được cấp sẵn do quản trị viên

Người dùng cần phải nhập đủ những thông tin sau để đăng nhập:

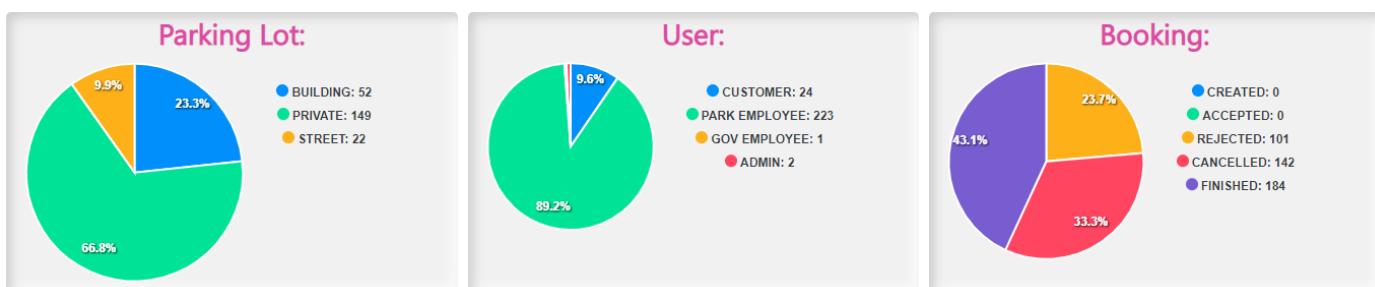
- ❖ Username: Tên đăng nhập được cấp
- ❖ Password: Mật khẩu được cấp



Hình 5.6.2.1.a Hình chức năng đăng nhập

5.6.2.2. Chức năng thống kê

Người dùng quản trị viên có thể xem được thống kê tổng thể về số lượng đặt chỗ, số lượng bãi xe và số lượng người dùng trên hệ thống

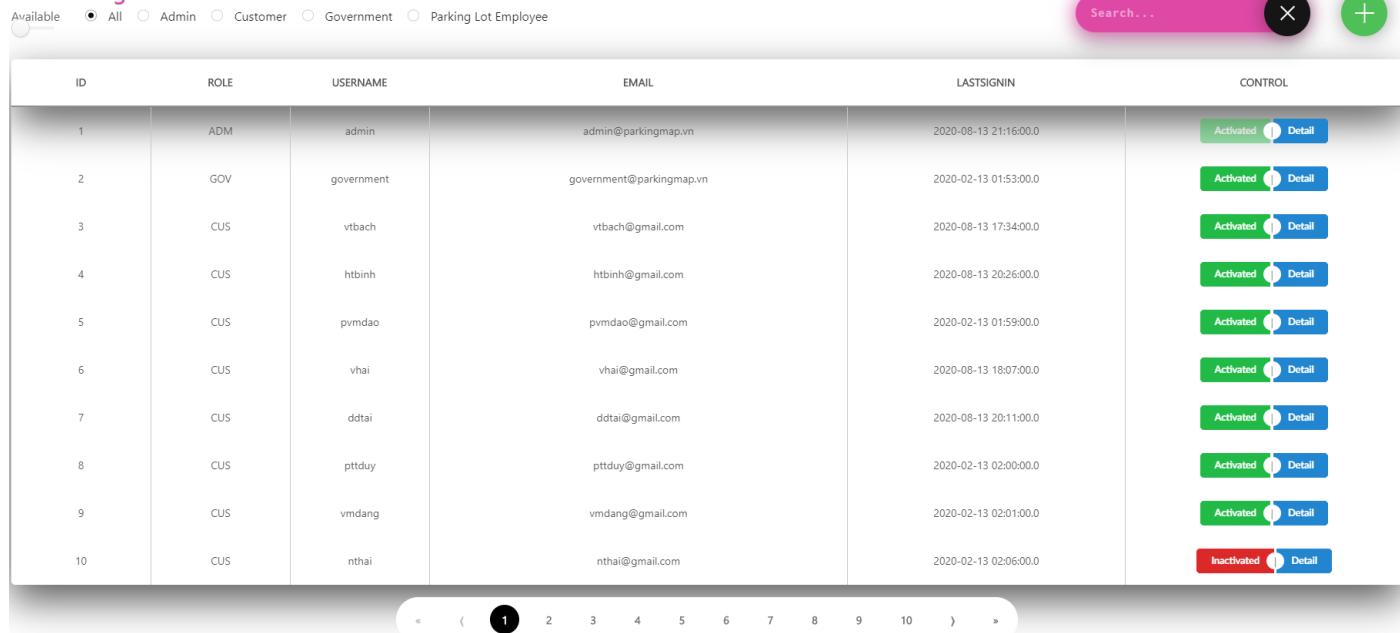


Hình 5.6.2.2.a Hình chức năng xem thống kê

5.6.2.3. Chức năng quản lý User

Người dùng quản trị viên có thể xem tổng thể thông tin cơ bản của từng User và được phân trang và cho phép người dùng có thể kích hoạt hoặc vô hiệu hóa tài khoản, xem thông tin chi tiết User, thêm User mới, tìm kiếm User theo nhiều kiểu dữ liệu

User Manager



The screenshot shows a table titled "User Manager" with the following data:

ID	ROLE	USERNAME	EMAIL	LASTSIGNIN	CONTROL
1	ADM	admin	admin@parkingmap.vn	2020-08-13 21:16:00.0	Activated Detail
2	GOV	government	government@parkingmap.vn	2020-02-13 01:53:00.0	Activated Detail
3	CUS	vtbach	vtbach@gmail.com	2020-08-13 17:34:00.0	Activated Detail
4	CUS	htbinh	htbinh@gmail.com	2020-08-13 20:26:00.0	Activated Detail
5	CUS	pvmadao	pvmadao@gmail.com	2020-02-13 01:59:00.0	Activated Detail
6	CUS	vhai	vhai@gmail.com	2020-08-13 18:07:00.0	Activated Detail
7	CUS	dttai	dttai@gmail.com	2020-08-13 20:11:00.0	Activated Detail
8	CUS	pttduy	pttduy@gmail.com	2020-02-13 02:00:00.0	Activated Detail
9	CUS	vmdang	vmdang@gmail.com	2020-02-13 02:01:00.0	Activated Detail
10	CUS	nthai	nthai@gmail.com	2020-02-13 02:06:00.0	Inactivated Detail

Hình 5.6.2.3.a Hình chức năng xem danh sách User

- ❖ Chức năng kích hoạt hoặc vô hiệu hóa tài khoản: Nhấn nút Activated để chuyển thành Deactivated đối với tài khoản đang được kích hoạt hoặc nhấn nút Deactivated để chuyển thành Activated đối với tài khoản đang bị vô hiệu hóa
- ❖ Chức năng xem thông tin chi tiết User: nhấp vào nút Detail để hiện ra cửa sổ thông tin chi tiết User (Do User có 3 4 loại nên mỗi User sẽ có cấu trúc thông tin khác nhau tùy thuộc vào loại User đó):

ADMIN		GOVERNMENT EMPLOYEE	
Activated	true	Activated	true
Last sign in	2020-08-13 21:16:00.0	Last sign in	2020-08-13 21:16:00.0
Role	ADMIN	Username	admin
Username	admin	Email	admin@parkingmap.vn
Email	admin@parkingmap.vn	close	
close			

CUSTOMER

Activated	<input type="text" value="true"/>
Last sign in	<input type="text" value="2020-08-13 17:34:00.0"/>
Username	<input type="text" value="vtbach"/>
Email	<input type="text" value="vtbach@gmail.com"/>
First Name	<input type="text" value="Bách"/>
Last Name	<input type="text" value="Vũ Tường"/>
Phone	<input type="text" value="0123456789"/>
<input type="button" value="close"/>	

PARKING LOT EMPLOYEE

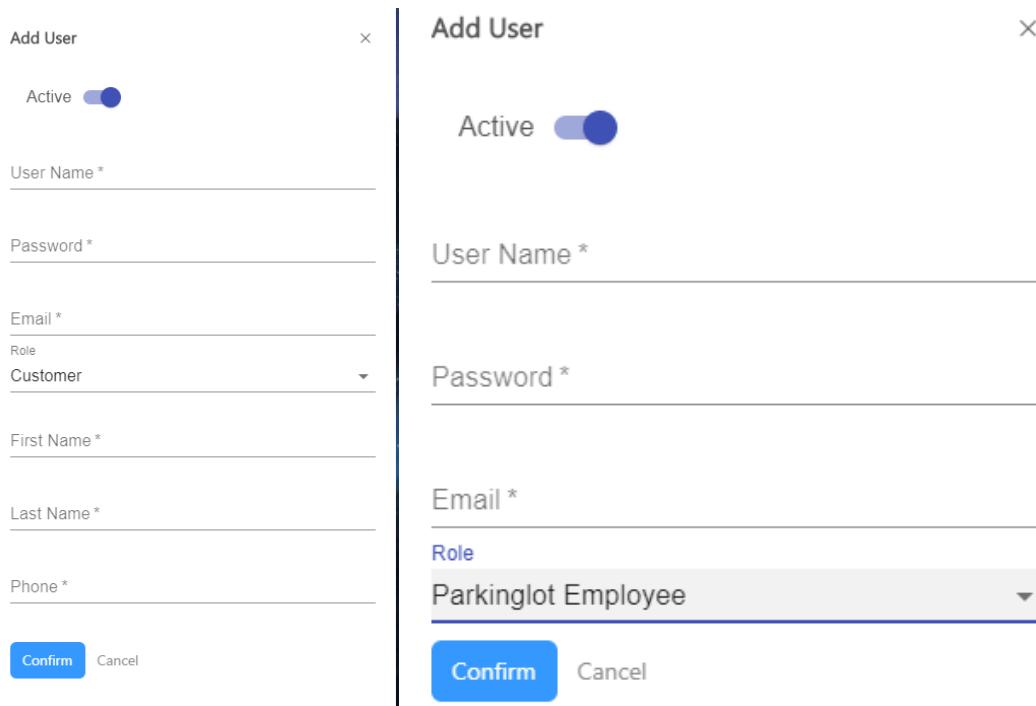
Activated	<input type="text" value="true"/>
Last sign in	<input type="text" value="2020-02-19 18:53:00.0"/>
Username	<input type="text" value="bxtonhthmost"/>
Email	<input type="text" value="bxtonhthmost@gmail.co"/>
Management	<input type="text" value="6: Bãi xe Toà nhà The Mo"/>
<input type="button" value="close"/>	

Hình 5.6.2.3.b Hình thông tin chi tiết User theo từng Role

- ❖ Chức năng tìm kiếm User theo nhiều điều kiện sau:

- + Available: Mặc định là tất cả sẽ xuất ra toàn bộ User – Mở sẽ xuất ra những User nào hiện đang có trạng thái tài khoản là Deactivated
- + All, Admin, Customer, Government, Parking Lot Employee: Mặc định là All sẽ xuất ra toàn bộ User – Chỉ được chọn 1 trong những thông tin còn lại sẽ xuất ra những User có vai trò như trên (VD: chọn Customer sẽ xuất ra danh sách những User có vai trò là khách hàng,.. ...)

- ❖ Chức năng thêm User mới:



The screenshot displays two identical user addition forms side-by-side. Each form has a header 'Add User' and a 'Close' button. A 'Active' toggle switch is set to 'On'. Below it is a 'User Name *' input field. The left form's 'Role' dropdown is set to 'Customer', while the right form's is set to 'Parkinglot Employee'. Both forms also have 'First Name *', 'Last Name *', 'Phone *', and 'Email *' input fields, as well as 'Password *' fields. At the bottom of each form are 'Confirm' and 'Cancel' buttons.

Hình 5.6.2.3.c Hình chức năng thêm User

+ Đối với lựa chọn Role là Customer, người dùng cần phải nhập những thông tin sau mới có thể nhấn nút Confirm để hoàn tất thêm User:

1. User Name: Chữ không dấu và số và không được trùng với những User khác đã có trong dữ liệu
2. Password: Chữ không dấu và số
3. Email: Đúng định dạng email và không được trùng với những User khác đã có trong dữ liệu
4. First Name: Tên
5. Last Name: Họ
6. Phone: Số điện thoại là số và có 10 kí tự

+ Đối với lựa chọn Role không phải là Customer, người dùng cần phải nhập những thông tin sau mới có thể nhấn nút Confirm để hoàn tất thêm User:

1. User Name: Chữ không dấu và số và không được trùng với những User khác đã có trong dữ liệu
2. Password: Chữ không dấu và số
3. Email: Đúng định dạng email và không được trùng với những User khác đã có trong dữ liệu

5.6.2.4. Chức năng quản lý bãi xe

Người dùng quản trị viên có thể xem tổng thể thông tin cơ bản của từng Bãi xe và được phân trang và cho phép người dùng có thể xóa bãi xe, xem thông tin chi tiết bãi xe, thêm bãi xe mới, tìm kiếm bãi xe theo nhiều kiểu dữ liệu và quản lý nhân viên của bãi xe đó

Parking Manager

 Available All Building Private Street



ID	NAME	TYPE	AVAILABILITY	CAPACITY	CONTROL
1	Bãi xe Chung cư Ngõ Tát Tố	Private	30	30	
2	Bãi xe Saigon Pearl	Private	300	300	
3	Bãi xe Hồ bơi Yết Kiêu	Private	40	40	
4	Bãi xe Chung cư Mỹ Đức	Private	150	150	
5	Bãi xe Công Ty TNHH Sơn Hải	Private	20	20	
6	Bãi xe Tòa nhà The Morning Star	Private	100	100	
7	Bãi xe Chung cư Thủý Lợi 4	Private	100	100	
8	Bãi xe Công ty TNHH Minh Tâm	Private	40	40	
9	Bãi xe Chung cư Mỹ Phước	Private	100	100	
10	Bãi xe Cửa hàng xăng dầu Bạch Đằng	Private	10	10	

< (1 2 3 4 5 6 7 8 9 10) >

Hình 5.6.2.4.a Hình chức năng xem danh sách bãi xe

- ❖ Chức năng xem thông tin chi tiết bãi xe:



- ★★★★☆
- NAME: Bãi xe Sân vận động Phú Thọ
- ADDRESS: 221 Lý Thường Kiệt, Phường 9, Quận 11, Hồ Chí Minh
- PHONE:
- TYPE: Private
- OPEN: 08:00:00
- CLOSE: 23:55:56
- AVAILABLESLOT: 60
- TOTALSLOT: 60

close

Hình 5.6.2.4.b. Hình thông tin chi tiết của bãi xe

- ❖ Chức năng xóa bãi xe hiện có: nhấn nút DELETE vào bãi xe cần xóa và xác nhận xóa
- ❖ Chức năng quản lý nhân viên của bãi xe đó: nhấn vào nút M-E (Management Employee) để xem danh sách những user đang quản lý bãi xe được chọn và có thể chọn Remove để xóa User khỏi danh

sách hoặc chọn Delete để xóa hoàn toàn User đó khỏi dữ liệu

Manager Employee

13: bxchcungtato

Remove **Delete**

270: bxsape2

Remove **Delete**

Ok

Cancel

Hình 5.6.2.4.c Hình chức năng quản lý danh sách User quản lý bãi xe

- ❖ Chức năng thêm bãi xe mới:

Add ParkingLot

Name *

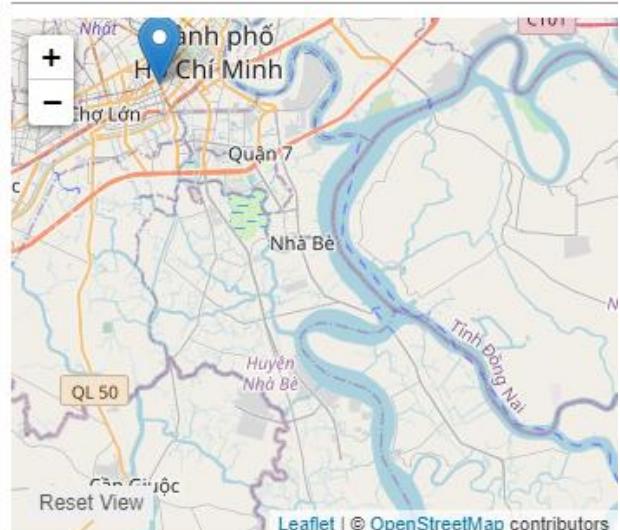
10.762273961494273

Address *

Lng *

466.6819882392884

Opening Hour *



Closing Hour *

Total Slot *

0

Type

Building

Phone

Confirm **Cancel**

Hình 5.6.2.4.d Hình chức năng thêm bãi xe mới
Người dùng cần nhập đầy đủ các thông tin sau để có thể nhấn nút Confirm để hoàn thành thêm bãi xe:

1. Name: Tên bãi xe
2. Address: Địa chỉ bãi xe
3. Opening Hour: Giờ mở cửa
4. Closing Hour: Giờ đóng cửa

5. Total Slot: Số lượng chỗ của bãi
6. Type: Loại bãi xe (có 3 loại: Building, Private, Street)
7. Phone: (Tùy chọn) Số điện thoại
8. Lat, Lng: tọa độ chính xác của bãi xe (Người dùng có thể dùng bản đồ nhỏ bên dưới để xác định bãi xe bằng mắt và nhấn chuột để đặt vị trí của bãi xe)

5.6.2.5. Chức năng đổi mật khẩu

Người dùng nhập mật khẩu mới và nhập lại 1 lần nữa mật khẩu đó vào ô Confirm Password và nhấn Update để cập nhật mật khẩu mới

Hình 5.6.2.5.a Hình chức năng đổi mật khẩu

5.6.2.6. Chức năng thêm User quản lý bãi

Người dùng chọn 1 bãi xe cần để thêm 1 User với Role là Parking Lot Employee thành User Để quản lý bãi xe đó:

ID	NAME	TYPE	CAPACITY
1	Bãi xe Chung cư Ngõ Tát Tố	Private	30
2	Bãi xe Saigon Pearl	Private	300
3	Bãi xe Hồ bơi Yết Kiêu	Private	40
4	Bãi xe Chung cư Mỹ Đức	Private	150
5	Bãi xe Công Ty TNHH Sơn Hải	Private	20
6	Bãi xe Tòa nhà The Morning Star	Private	100
7	Bãi xe Chung cư Thủ Lợi 4	Private	100
8	Bãi xe Công ty TNHH Minh Tâm	Private	40
9	Bãi xe Chung cư Mỹ Phước	Private	100
10	Bãi xe Cửa hàng xăng dầu Bách Đằng	Private	10

Hình 5.6.2.6.a Hình chức năng thêm User thành người quản lý bãi xe

Người dùng chọn 1 User để thêm, ở bước này, người dùng cần chọn đúng User chưa quản lý bãi xe nào hết, nếu User đó đã có quản lý 1 bãi xe nào khác rồi thì kết quả sẽ hiển thị ra như thế này:

Choose Parking Lot _____

Employee: bxchcungtato - ID: 13

Can't add to **Bãi xe Chung cư Ngô Tất Tố** because User already have another parking lot
Please choose again!

Hình 5.6.2.6.b Hình ảnh thông báo User đã có quản lý 1 bãi xe nào đó rồi

Nếu User chưa quản lý bãi xe nào hết thì người dùng có thể nhấn Confirm để hoàn tất việc thêm User trở thành quản lý 1 bãi giữ xe:

Choose Parking Lot _____

Employee: 16530202 - ID: 272

Do you want to add to **Bãi xe Chung cư Ngô Tất Tố** ?
Please Confirm!

Hình 5.6.2.6.c Hình thông báo có thẻ thêm 1 User thành quản lý bãi xe

5.6.3. Ứng dụng di động dành cho tài xế xe ô tô



Hình 5.6.3. Hình đại diện ứng dụng di động dành cho tài xế xe ô tô

5.6.3.1. Chức năng Đăng ký/đăng nhập

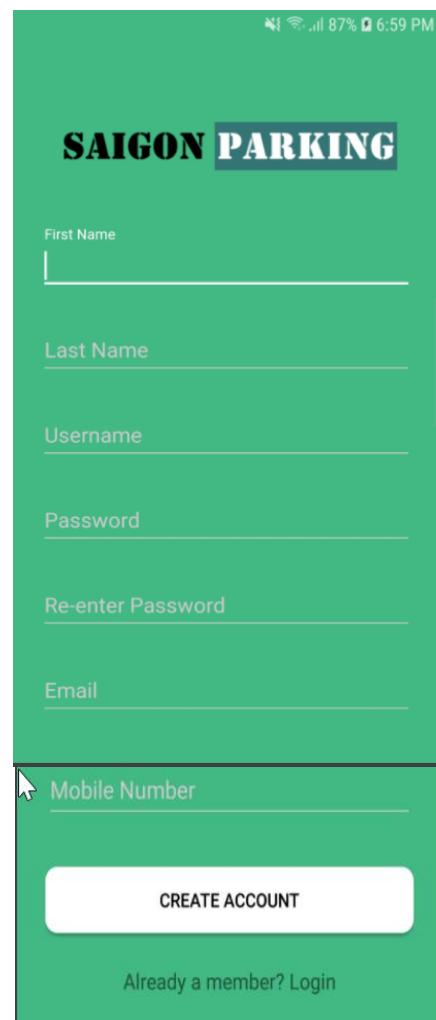
- Cho phép người dùng đăng ký và đăng nhập tài khoản Saigon Parking. Người dùng phải đăng ký tài khoản để có thể đăng nhập và sử dụng các dịch vụ mà ứng dụng Saigon Parking cung cấp.
- Ở mỗi màn hình đăng ký/đăng nhập của ứng dụng, người dùng phải nhập đầy đủ các ô thông tin được yêu cầu để có thể đăng ký/đăng nhập thành công



Hình 5.6.3.1.a. Màn hình xác nhận cho phép sử dụng camera điện thoại



Hình 5.6.3.1.b. Màn hình đăng nhập



Hình 5.6.3.1.c. Màn hình đăng ký

Sau khi đăng nhập thành công, ứng dụng sẽ chuyển sang màn hình bản đồ. Ở màn hình này người dùng có thể tìm kiếm bãi xe quanh vị trí hiện tại hoặc xung quanh địa điểm mà họ dự định tới.



Hình 5.6.3.1.d: Màn hình bản đồ

5.6.3.2. Chức năng tìm kiếm bãi xe quanh vị trí hiện tại

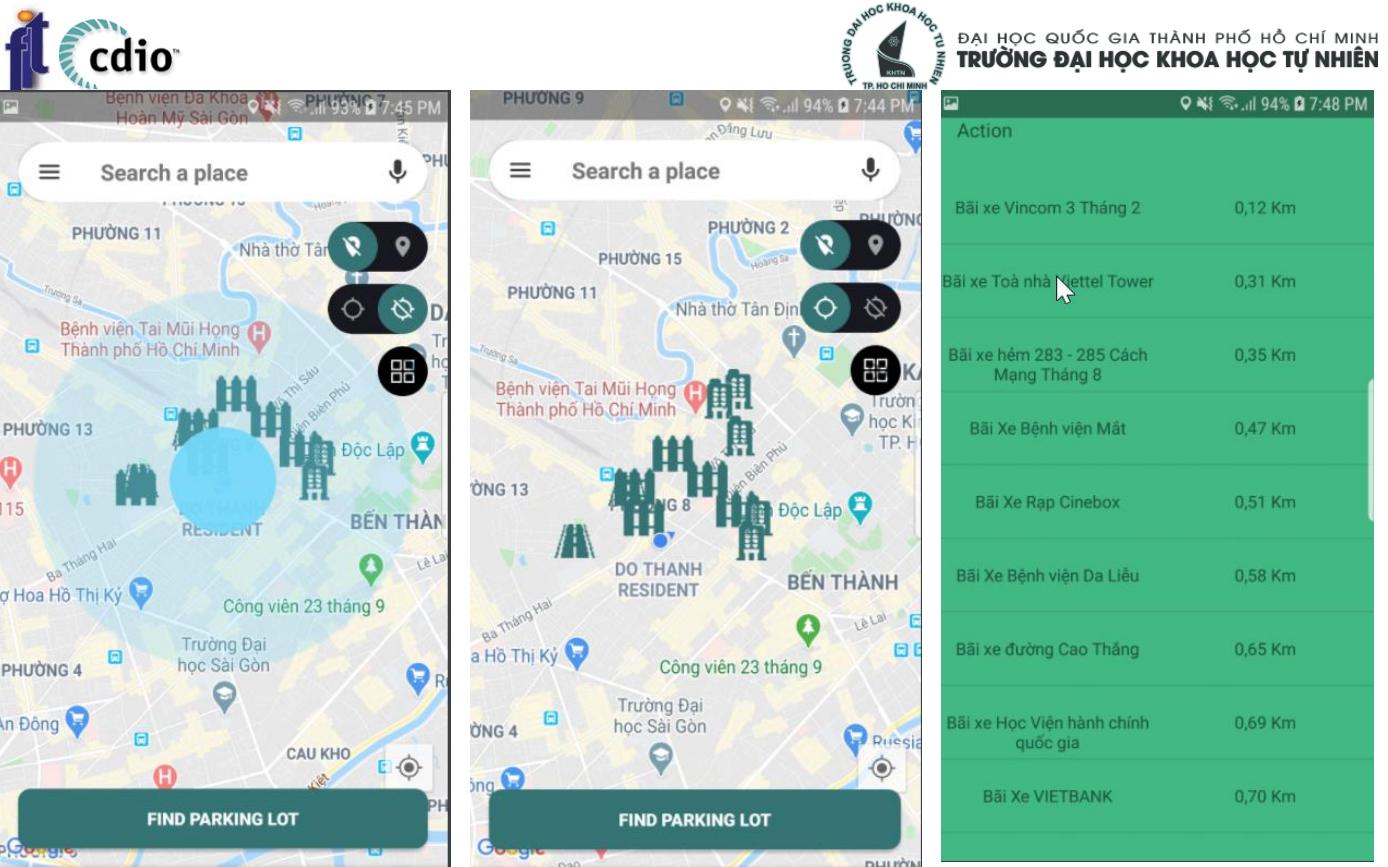
- Cho phép người dùng tìm kiếm bãi xe quanh vị trí hiện tại của họ bằng hai lựa chọn sau:
 - + Nhấn nút “FIND PARKING LOT” trên màn hình bản đồ.
 - + Nhấn, di chuyển ngón tay trên màn hình và nháy ra. Các bãi xe sẽ tự động được tải lên

5.6.3.3. Chức năng tìm kiếm bãi xe quanh địa điểm cần đến

- Cho phép người dùng tìm kiếm bãi xe quanh địa điểm cần đến bằng cách:
 - + Nhấn tìm kiếm địa điểm bằng thanh Search trên đầu màn hình bản đồ và nhấn Enter. Ứng dụng sẽ di chuyển màn hình đến địa điểm vừa tìm và hiện danh sách các bãi xe gần đó

5.6.3.4. Chức năng hiển thị các địa điểm khác

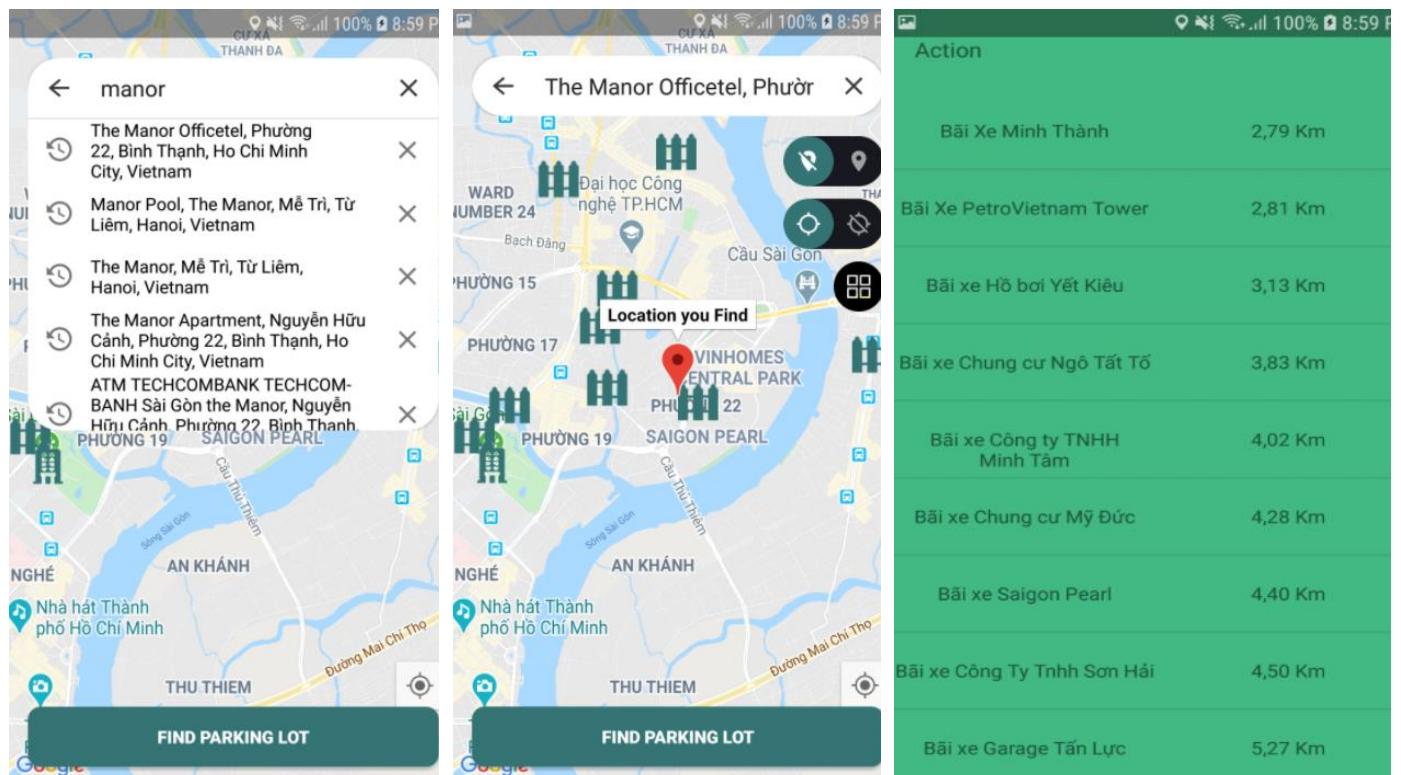
- Cho phép người dùng xem vị trí các địa điểm có liên quan đến phương tiện ô tô (trạm xăng, chỗ vá lốp, bệnh viện, nhà hàng,...)



Hình 5.6.3.2.a. Màn hình hiển thị bãi xe quanh vị trí hiện tại khi nhấn nút “FIND PARKING LOT”

Hình 5.6.3.2.b. Màn hình hiển thị bãi xe quanh vị trí hiện tại khi bắn đồ tài lên dữ liệu bãi xe

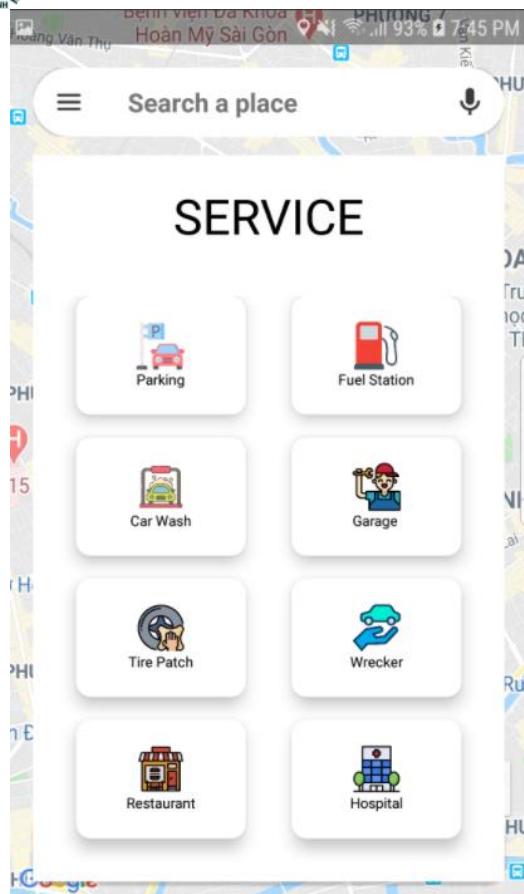
Hình 5.6.3.2.c. Hiển thị danh sách các bãi xe quanh vị trí hiện tại khi kéo màn hình từ dưới lên



Hình 5.6.3.3.a. Màn hình tìm kiếm địa điểm cần đến

Hình 5.6.3.3.b. Bản đồ di chuyển đến địa điểm vừa tìm và hiển thị các bãi xe gần đó

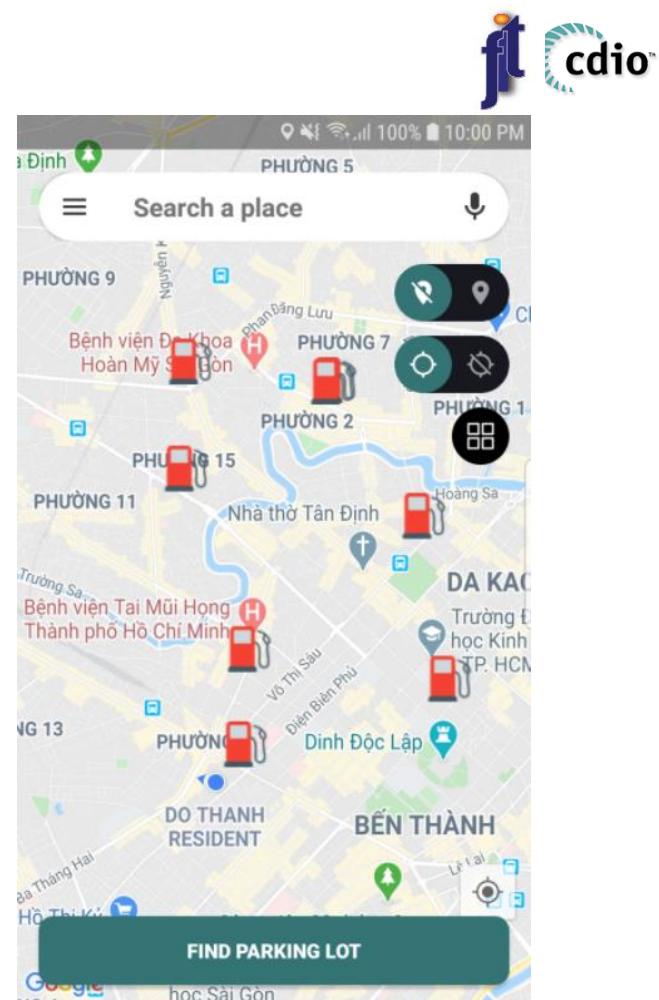
Hình 5.6.3.3.c. Hiển thị danh sách các bãi xe quanh địa điểm vừa tìm khi kéo màn hình từ dưới lên



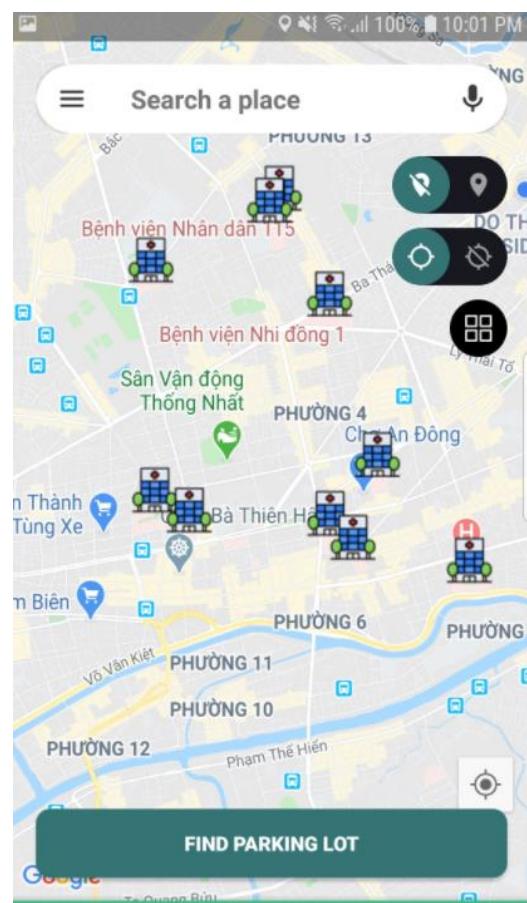
Hình 5.6.3.4.a. Hiển thị danh sách các địa điểm khác



Hình 5.6.3.4.c. Hiển thị danh sách nhà hàng



Hình 5.6.3.4.b. Hiển thị danh sách các trạm xăng



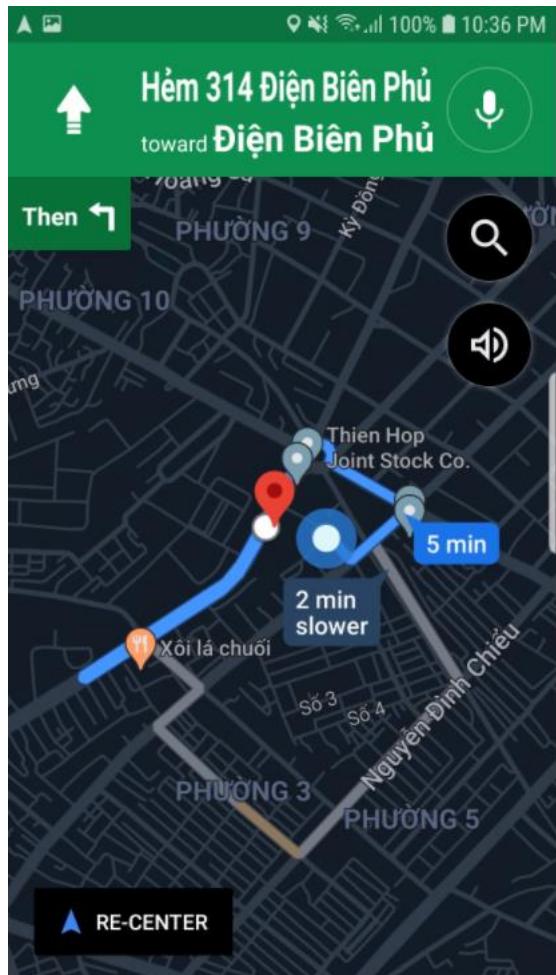
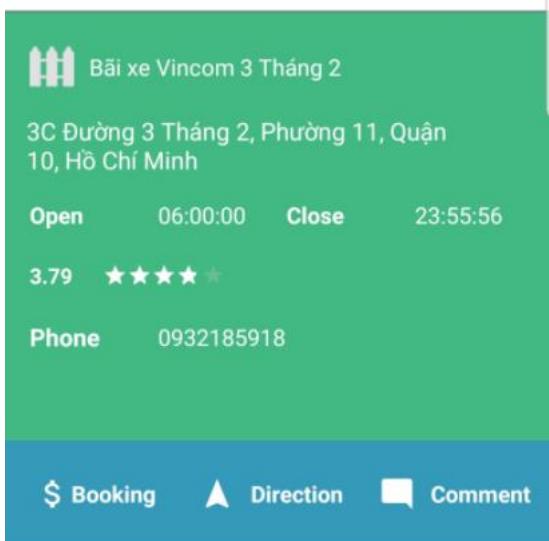
Hình 5.6.3.4.d. Hiển thị danh sách bệnh viện

5.6.3.5. Chức năng chỉ đường đến bãi xe bằng Google Direction API

- Chức năng khả dụng sau khi người dùng bấm vào biểu tượng bãi xe hiện trên bản đồ. Ứng dụng sẽ chuyển đến màn hình chi tiết bãi xe đó.
- Người dùng bấm vào nút “Direction”, ứng dụng sẽ chuyển đến chức năng điều hướng của Google Map. Người dùng chỉ cần theo chỉ dẫn của điều hướng + điều khiển phương tiện theo đường vẽ trên bản đồ cho đến khi tới bãi đỗ.



Availability: 60/60



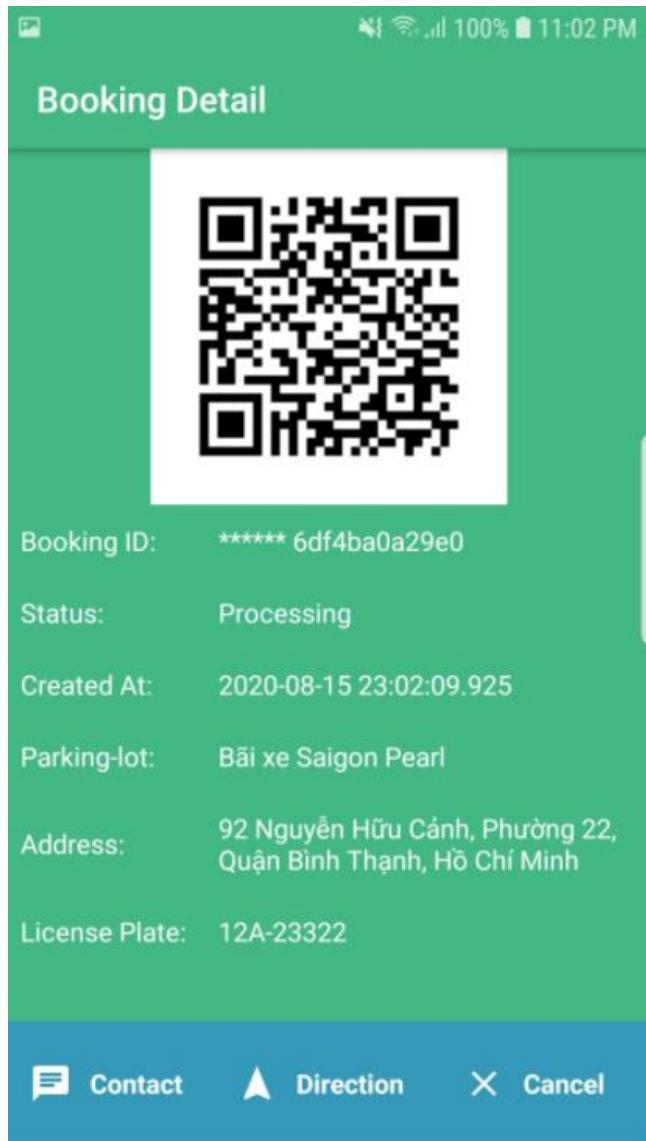
Hình 5.6.3.5.a. Màn hình hiển thị thông tin chi tiết bãi xe

Hình 5.6.3.5.b. Màn hình chỉ đường đến bãi xe

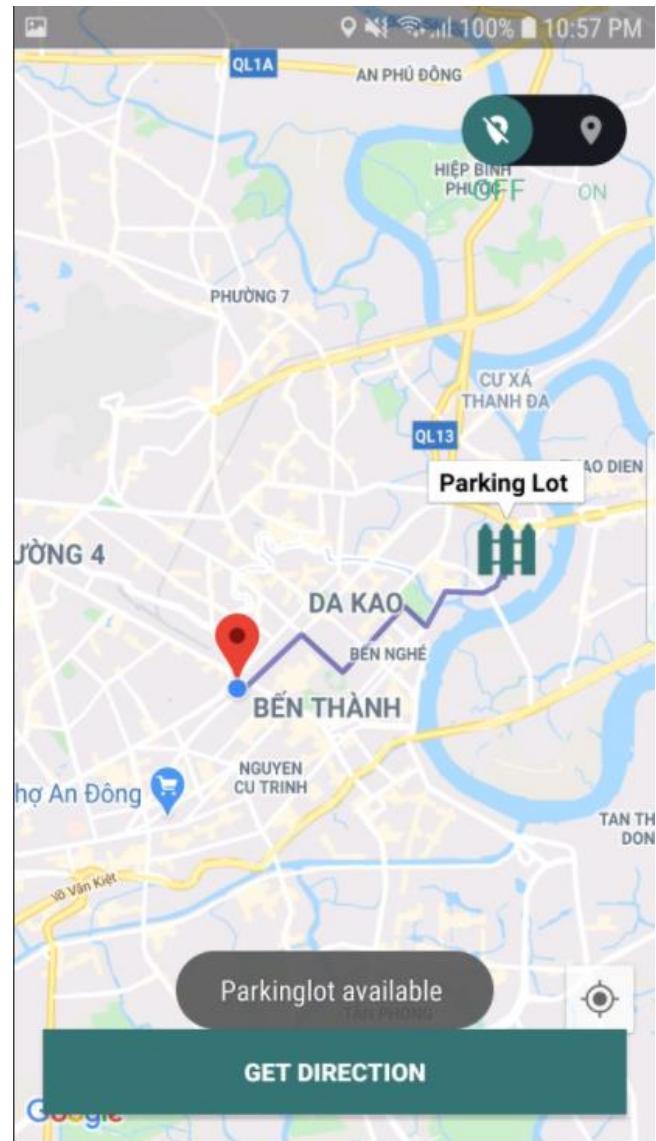
5.6.3.6. Chức năng chỉ đường đến bãi xe bằng ứng dụng hệ thống

- Để chủ động hơn trong việc quản lý hệ thống, tránh việc người dùng phải chuyển qua các ứng dụng thứ ba chỉ để thực hiện chức năng chỉ đường đến bãi xe, nhóm em đã phát triển chức năng chỉ đường mà không cần phải sử dụng Direction API của Google.
- Chức năng khả dụng khi người dùng bấm vào nút “Booking” trong màn hình hiển thị thông tin chi tiết bãi xe. Vì chức năng chỉ đường của hệ thống khá đặc biệt (sẽ được giới thiệu ở đề mục 5.6.3.7) nên chỉ khi người dùng có đặt trước chỗ đậu xe trong bãi đỗ, chức năng chỉ đường này mới được bật lên.

- Sau khi bấm nút “Booking”, hệ thống sẽ chuyển sang màn hình Đặt Chỗ. Lúc này người dùng bấm vào nút “Direction” trên màn hình, hệ thống sẽ chuyển sang màn hình Chỉ Đường với đường đi đã được vẽ theo khoảng cách ngắn nhất tính từ vị trí người dùng đến vị trí bãi xe



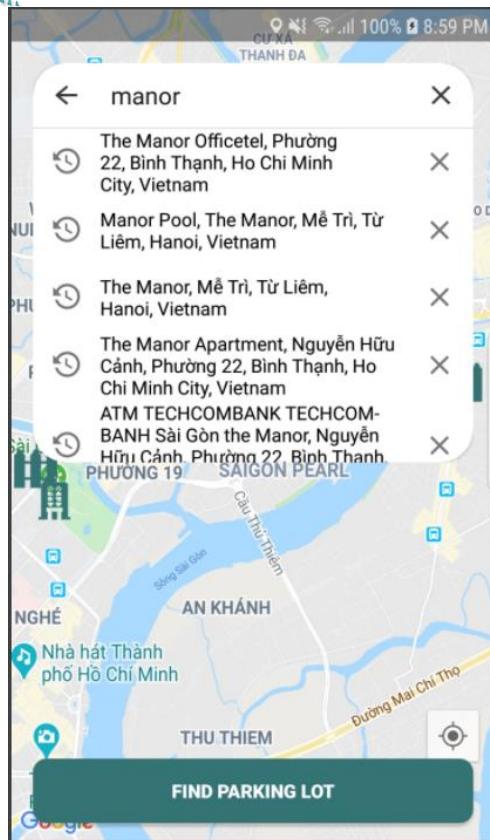
Hình 5.6.3.6.a. Màn hình Đặt Chỗ



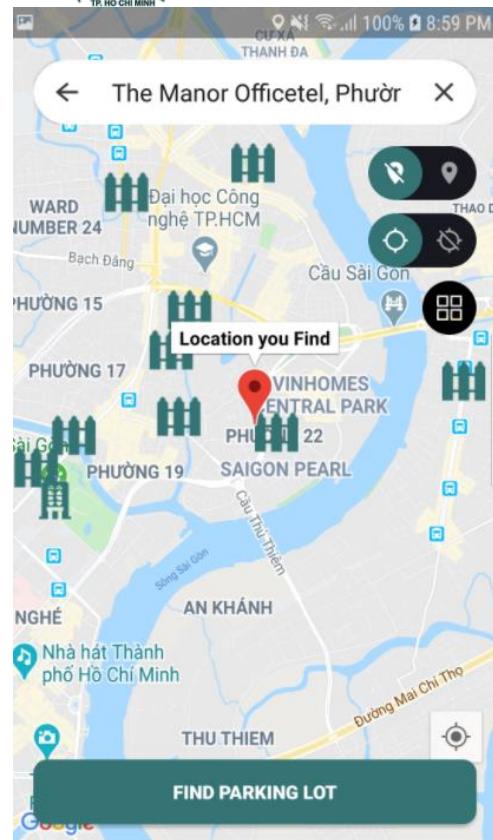
Hình 5.6.3.6.b. Màn hình chỉ đường tới bãi đỗ

5.6.3.7. Chức năng tìm địa điểm thứ ba

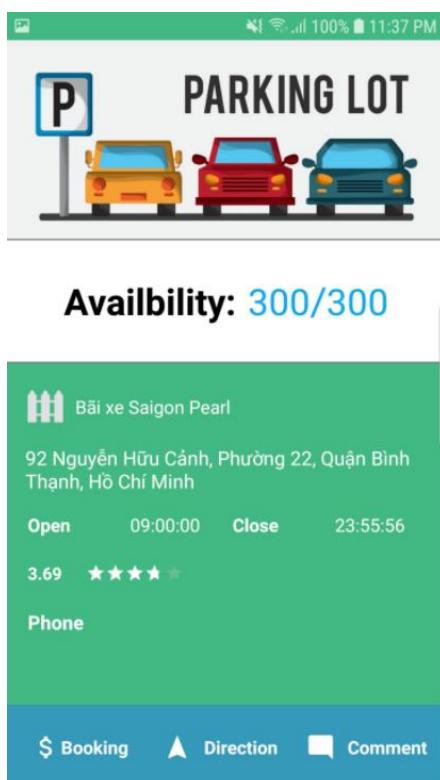
- Chức năng này được xây dựng khi người dùng muốn di chuyển tới một địa điểm nào đó (quán ăn, tòa nhà,...) nhưng tại địa điểm đó không có bãi xe. Vì thế người dùng cần phải tìm đường đi từ vị trí hiện tại tới bãi xe gần với địa điểm cần đến nhất và đường đi từ bãi xe tới địa điểm đó
- Chức năng khả dụng khi người dùng bật chế độ “MODE ON: FIND DESTINATION” (tìm địa điểm thứ ba) trên màn hình bản đồ. Sau đó flow của ứng dụng được chạy giống như chức năng chỉ đường đến bãi xe bằng ứng dụng hệ thống. Sau khi chủ xe điều khiển phương tiện tới bãi xe cần đến, ứng dụng sẽ vẽ thêm đường đi tới địa điểm thứ ba



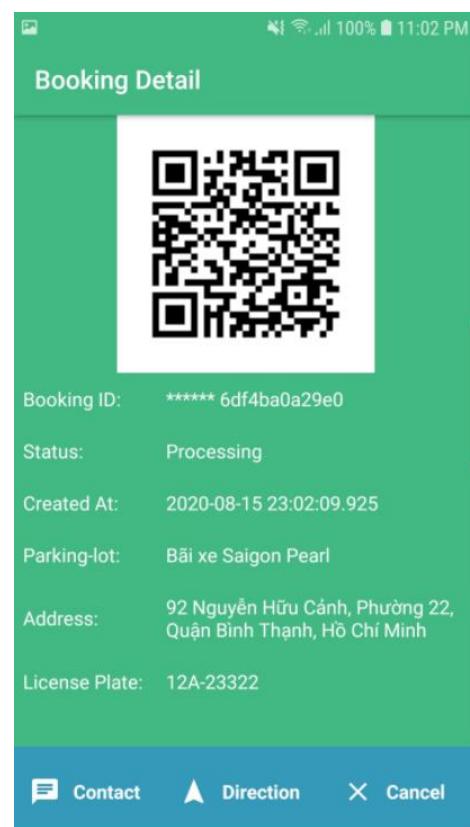
Hình 5.6.3.7.a. Tìm địa điểm cần đến



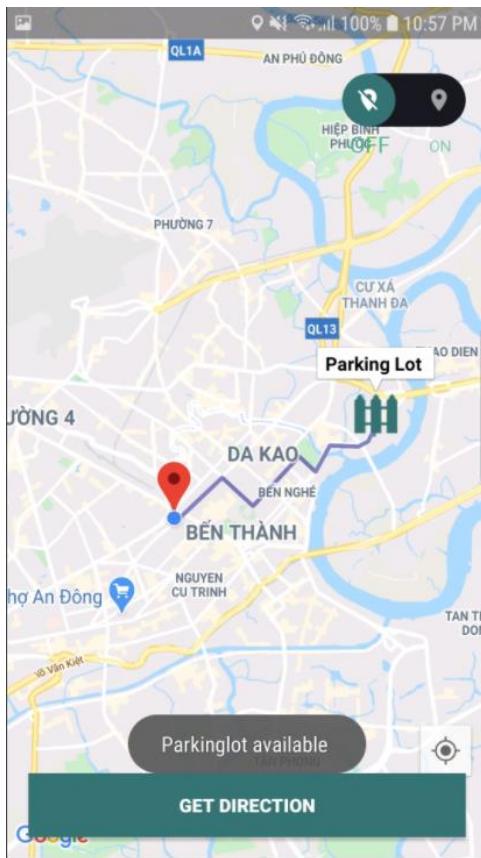
Hình 5.6.3.7.b. Chọn bất kì bãi xe xung quanh địa điểm đó



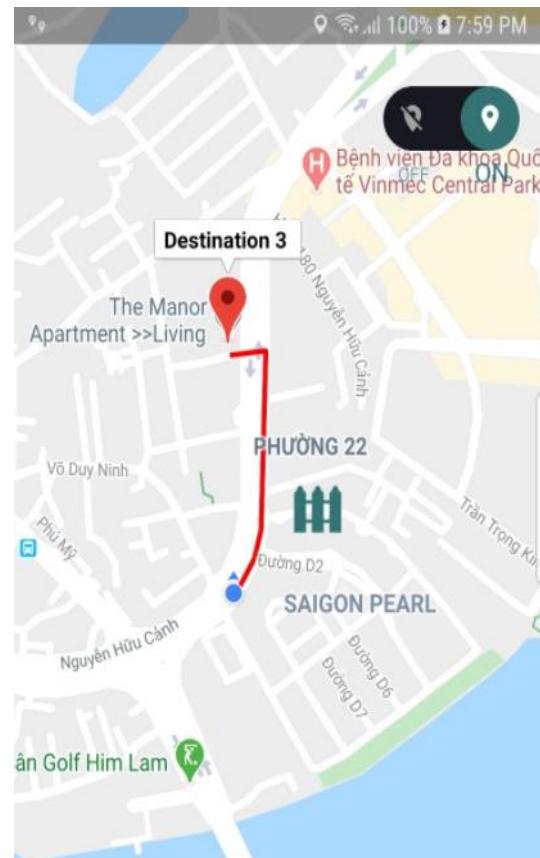
Hình 5.6.3.7.c. Màn hình hiển thị thông tin chi tiết bãi xe



Hình 5.6.3.7.d. Màn hình Đặt Chỗ



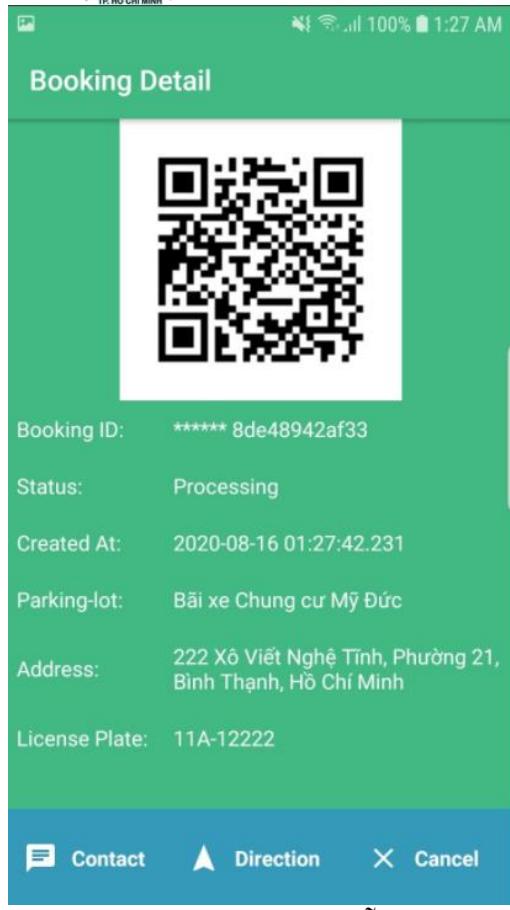
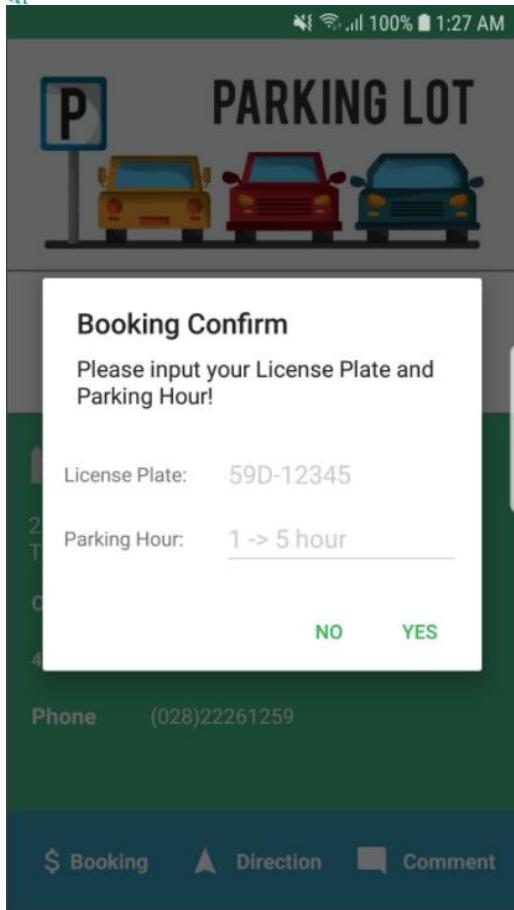
Hình 5.6.3.7.e. Màn hình chỉ đường tới bãi đỗ



Hình 5.6.3.7.f. Màn hình chỉ đường tới địa điểm thứ ba

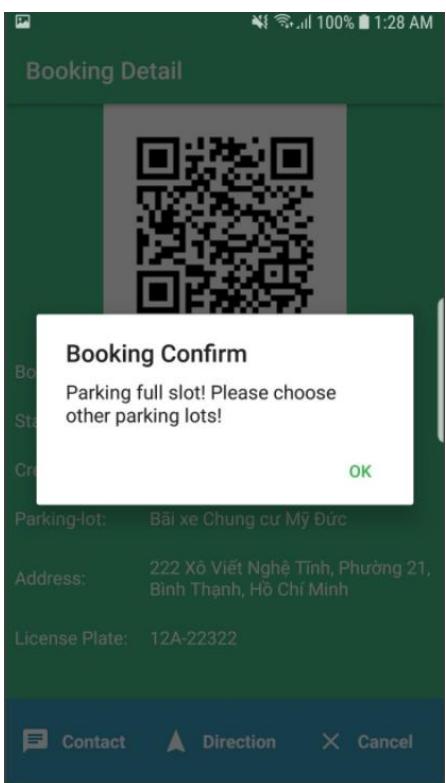
5.6.3.8. Chức năng đặt trước chỗ đậu xe

- Khi người dùng bấm Booking, ứng dụng sẽ gửi request đến Server và Server sẽ gửi request đó cho bãi xe. Bãi xe có thể chọn chấp nhận hoặc hủy bỏ yêu cầu đó.
 - + Nếu bãi xe hủy bỏ yêu cầu ứng dụng sẽ thông báo đến người dùng tìm bãi xe khác và quay trở lại màn hình bản đồ
 - + Nếu bãi xe chấp nhận ứng dụng sẽ chuyển sang trạng thái "Accept". Chỉ khi chủ phương tiện chạy xe đến bãi và check in mã QR hoặc biển số xe cho nhân viên bãi xe, yêu cầu Booking đó mới hoàn thành

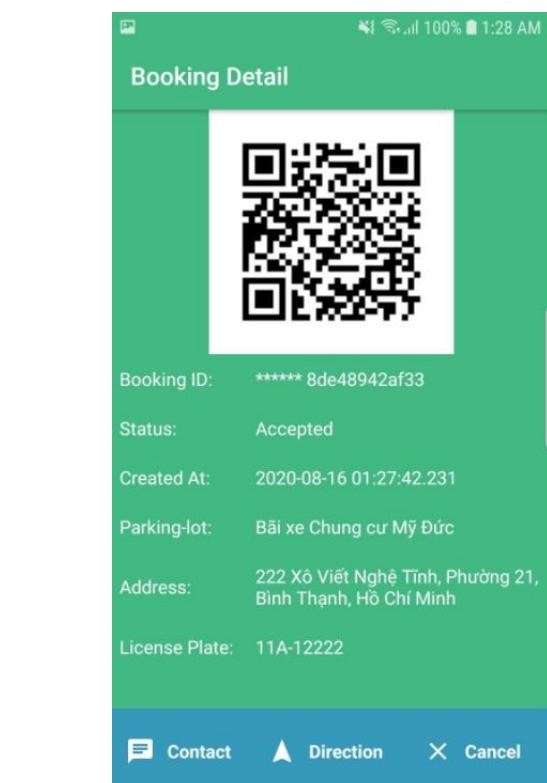


Hình 5.6.3.8.a. Nhập thông tin cần thiết để đặt chỗ

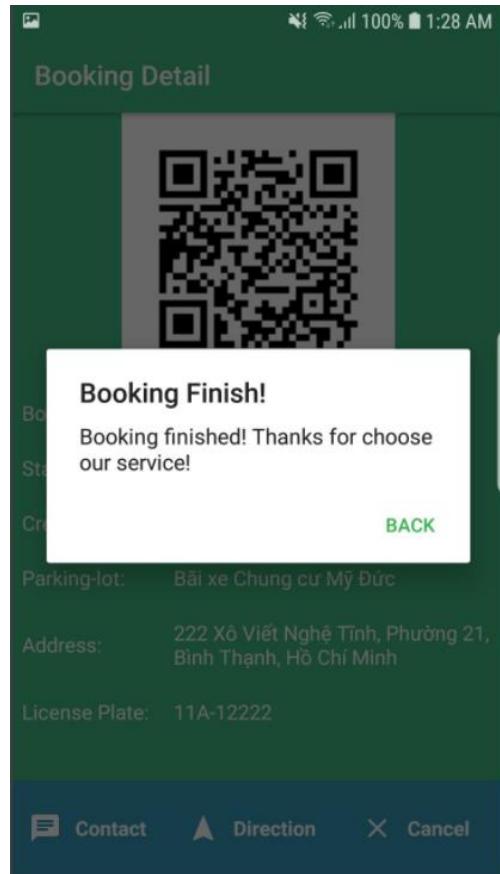
Hình 5.6.3.8.b. Màn hình Đặt Chỗ khi đang đợi bãi xe Accept (Status: PROCESSING)



Hình 5.6.3.8.c. Bãi xe từ chối yêu cầu đặt chỗ



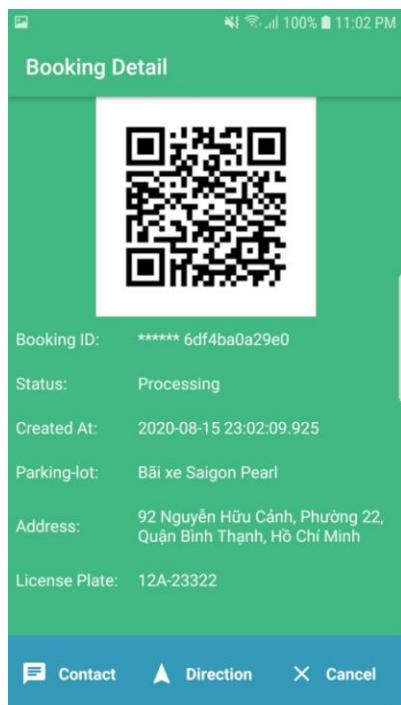
Hình 5.6.3.8.d. Bãi xe chấp nhận yêu cầu đặt chỗ (Status: ACCEPTED)



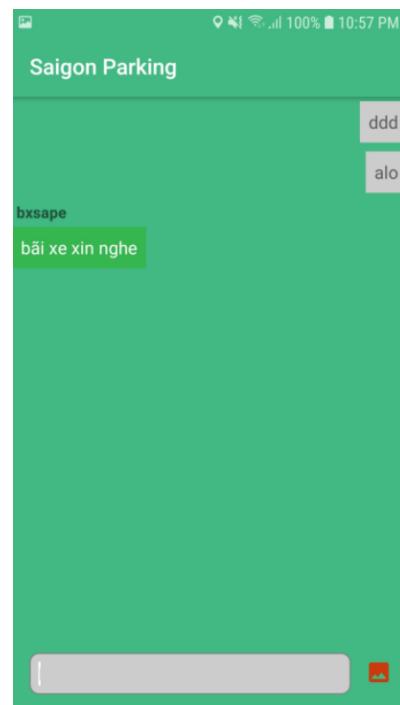
Hình 5.6.3.8.e. Hệ thống hoàn thành Booking

5.6.3.9. Chức năng chat với bãi xe

- Chức năng khả dụng khi người dùng bấm vào nút “Contact” trong màn hình Đặt Chỗ.
- Dùng để nhắn tin với bãi xe để biết thêm thông tin chi tiết về bãi đỗ



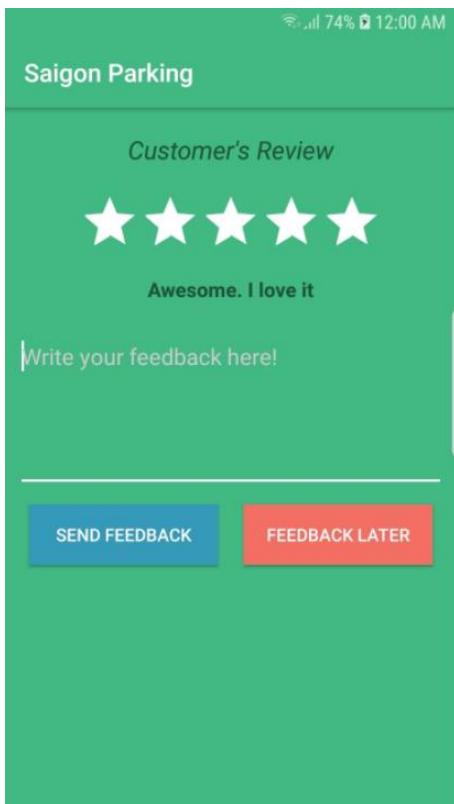
Hình 5.6.3.9.a. Màn hình Đặt Chỗ



Hình 5.6.3.9.b. Màn hình Chat

5.6.3.10. Chức năng đánh giá booking service

- Sau khi Booking thành công, ứng dụng sẽ hiện màn hình đánh giá bãi xe. Người dùng có thể viết bình luận và cho đánh giá dựa trên trải nghiệm của họ khi sử dụng dịch vụ đặt chỗ tại bãi xe đó hoặc chọn đánh giá sau (FEEDBACK LATER) và quay trở lại màn hình chính.



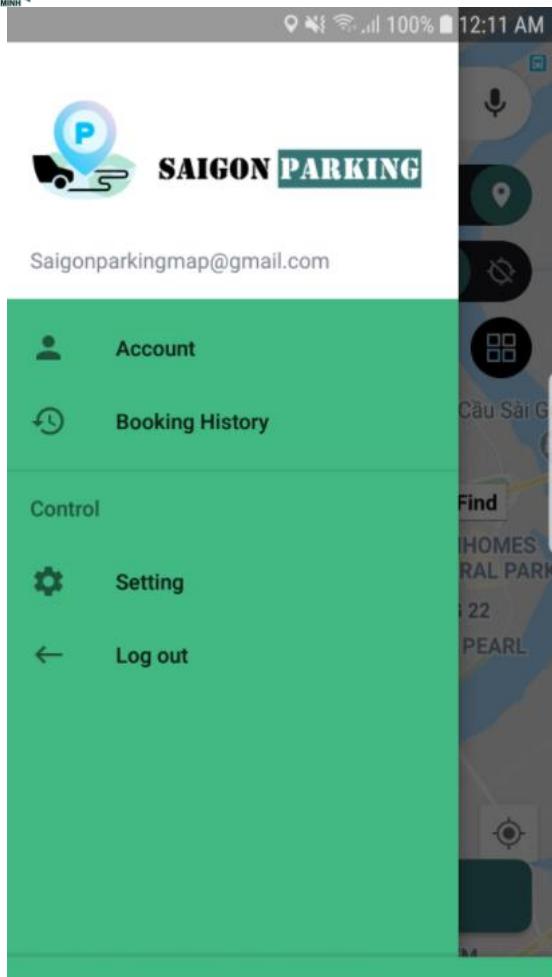
Hình 5.6.3.10.a: Màn hình đánh giá



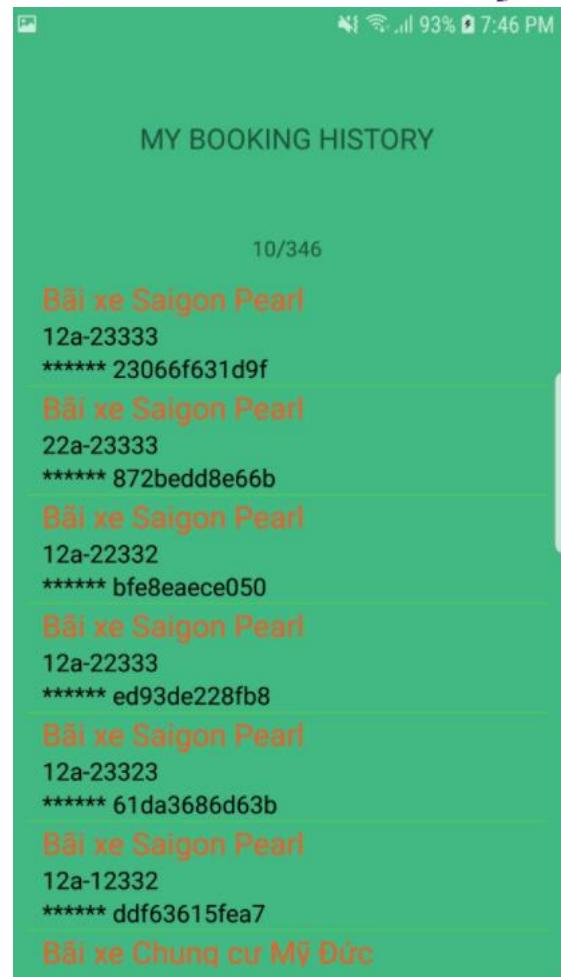
Hình 5.6.3.10.b: Màn hình xem đánh giá của người dùng khác

5.6.3.11. Chức năng xem lịch sử đặt chỗ

- Khách hàng có thể xem lại lịch sử đặt chỗ của mình khi nhấn nút Booking History trong màn hình bản đồ



Hình 5.6.3.11.a. Left slide menu



Hình 5.6.3.11.b: Màn hình lịch sử booking

5.6.4. Giả lập ứng dụng quản lý bãi giữ xe ô tô

5.6.4.1. Chức năng đăng nhập

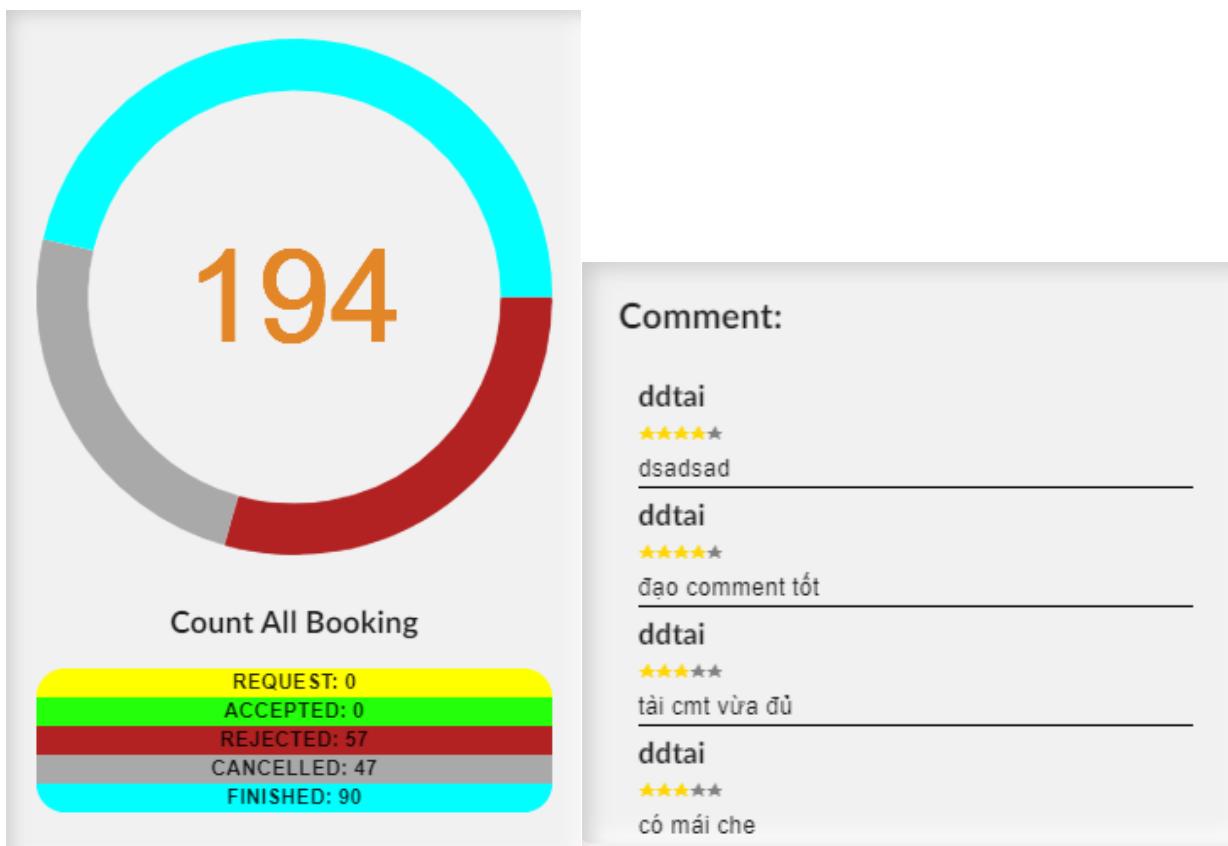
Người dùng cần nhập đủ thông tin User Name và Password được cấp do quản trị viên để đăng nhập

The form has a header message "Please Login to continue!". It contains two input fields: "User Name" and "Password", both represented by empty text input boxes. In the bottom right corner of the form area, there is a green "Submit" button with a white checkmark icon.

Hình 5.6.4.1.a. Hình chức năng đăng nhập

5.6.4.2. Chức năng thống kê

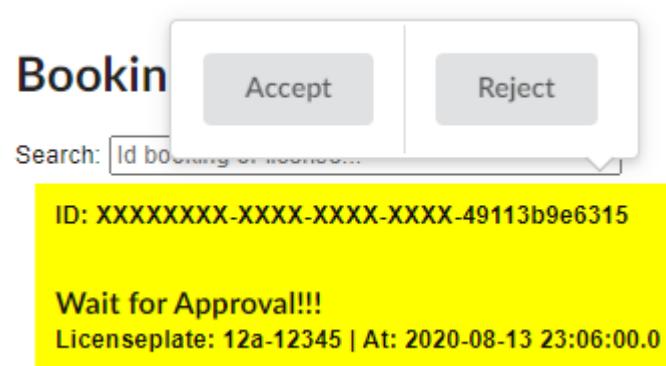
Người dùng có thể xem được thống kê số lượng đặt chỗ theo trạng thái và danh sách bình luận đối với bãi xe



Hình 5.6.4.2.a. Hình thống kê số lượng booking và đánh giá

5.6.4.3. Chức năng phản hồi yêu cầu đặt chỗ

Người dùng chọn Accept hoặc Reject để xác nhận hay từ chối yêu cầu đặt chỗ từ khách hàng



Hình 5.6.4.3.a. Hình phản hồi yêu cầu booking người dùng

5.6.4.4. Chức năng kết thúc quy trình đặt chỗ

Người dùng có thể nhấn vào yêu cầu đặt chỗ đã được chấp nhận ở trên để hoàn tất đặt chỗ khi xe đã vào bãi

Booking:

Search:

ID: XXXXXXXX-XXXX-XXXX-XXXX-49113b9e631

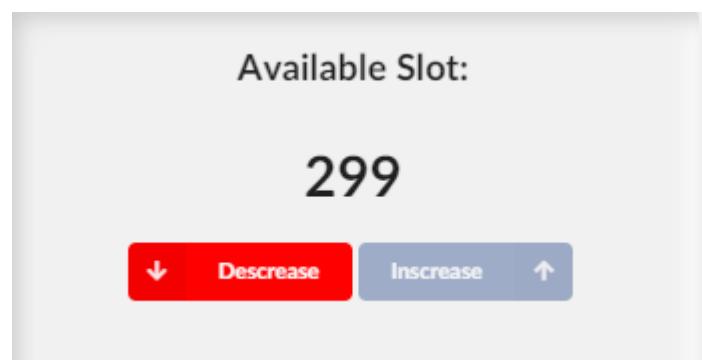
Click to Finish

Customer ongoing...
License Plate: 12a-12345 | At: 2020-08-13 23:06:00.0

Hình 5.6.4.4.a. Hình hoàn tất booking

5.6.4.5. Chức năng cập nhật số lượng chỗ

Người dùng có thể tăng giảm số lượng chỗ trống hiện tại để cập nhật dữ liệu bãi xe



Hình 5.6.4.5.a. Hình cập nhật số lượng chỗ

5.6.5. Giả lập máy quét mã QR



Hình 5.6.5. Hình đại diện cho ứng dụng giả lập máy quét mã QR

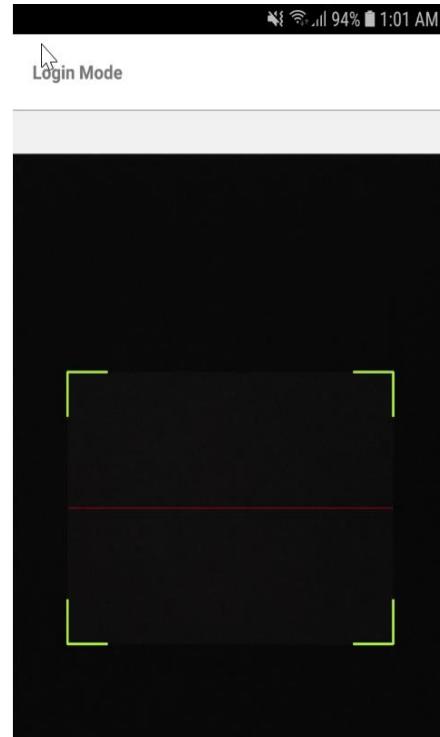
5.6.5.1. Chức năng đăng nhập

- Cho phép bãi xe đăng nhập bằng tài khoản bãi xe (access token của bãi xe được mã hóa thành mã QR). Khi người dùng quét mã QR được cấp trên trang web bãi xe, ứng dụng sẽ chuyển đổi từ Login Mode sang Finish Booking Mode

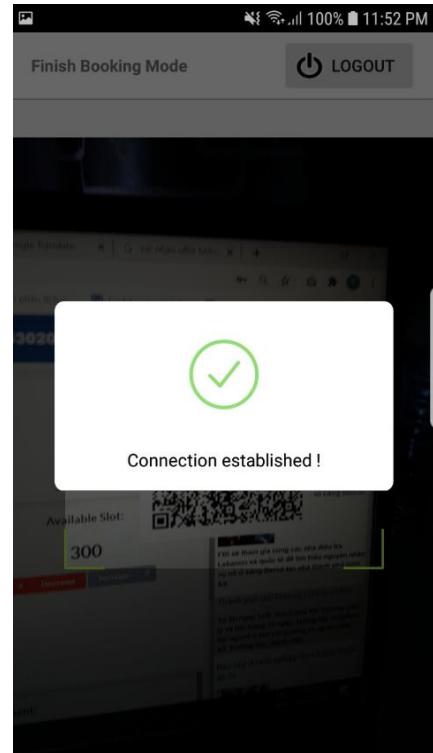
- Ở màn hình chờ, sau khi người dùng bấm nút SCAN QR CODE TO LOGIN, ứng dụng sẽ chuyển sang màn hình đăng nhập



Hình 5.6.5.1.a. Màn hình chờ của app



Hình 5.6.5.1.b. Màn hình quét mã QR để đăng nhập



Hình 5.6.5.1.c. Màn hình sau khi đăng nhập thành công

5.6.5.2. Chức năng đăng xuất

- Được sử dụng khi nhân viên bãi xe cần thoát ứng dụng để kết thúc phiên đăng nhập (chuyển giao ca làm cho nhân viên bãi xe khác)

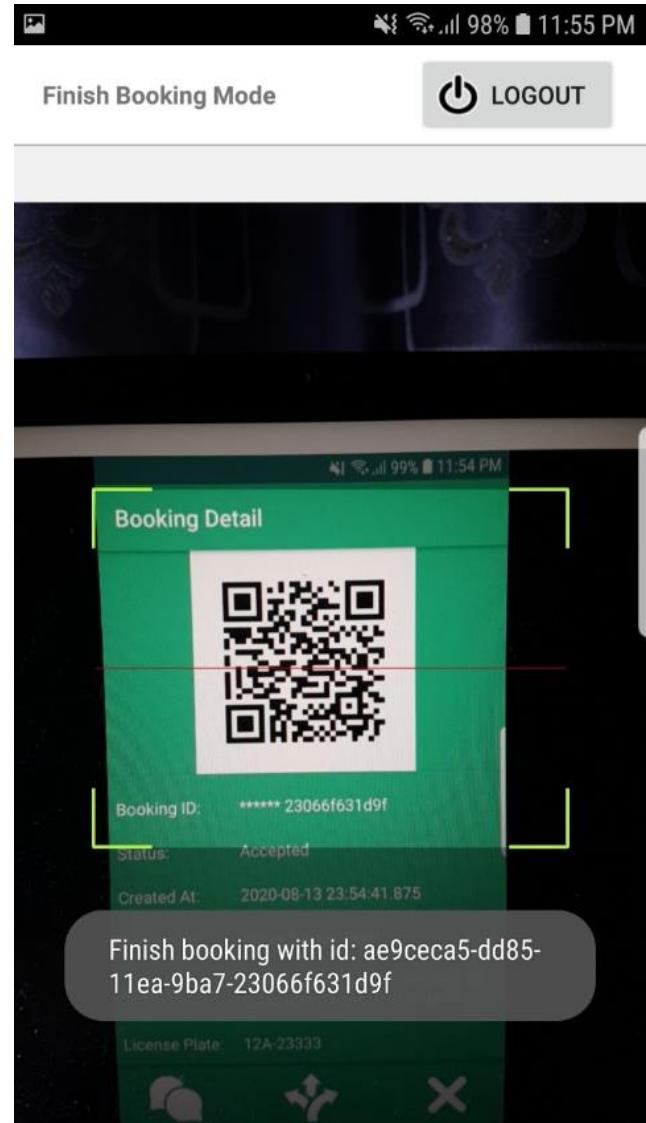
- Sau khi bấm nút LOGOUT ở màn hình quét mã QR, ứng dụng sẽ quay về màn hình chờ và hiện thông báo đóng kết nối

5.6.5.3. Chức năng quét mã QR để kết thúc booking

Sau khi khách hàng tới bãi xe, nhân viên sẽ thực hiện việc quét mã QR trên app khách hàng để kiểm tra mã booking có thật sự đúng hay không. Nếu quét đúng mã QR thì trên màn hình đăng nhập sẽ hiện thông báo tìm thấy ID kèm mã số. Ngay lúc này trên app khách hàng và web bãi xe sẽ đồng loạt hoàn thành booking và kết thúc quy trình



Hình 5.6.5.2.a. Màn hình đăng xuất



Hình 5.6.5.3.a. Màn hình quét mã QR để kết thúc booking

Chương 6 KẾT LUẬN, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

Nội dung của chương 6 đúc kết lại những giá trị cốt lõi đã đạt được khi xây dựng hệ thống, đánh giá ưu nhược điểm của ứng dụng và suy nghĩ hướng phát triển của ứng dụng trong tương lai

6.1. Kết luận

6.1.1. Về mặt lý thuyết

- Ứng dụng được xây dựng cơ bản đáp ứng các nhu cầu thực tế về việc tìm bến đậu và đặt trước chỗ của chủ xe ô tô
- Tăng thêm lượng khách và thu nhập cho các bến xe có liên kết với hệ thống
- Nghiệp vụ, quy trình hệ thống tìm bến xe, đặt trước chỗ nhìn chung đã được cài đặt thành công vào ứng dụng
- Quy trình quản lý tài khoản người dùng, tài khoản bến xe rõ ràng, dễ kiểm soát và bảo trì hệ thống khi có sự cố

6.1.2. Về mặt thực nghiệm

- Ứng dụng chạy thành công trên các môi trường Android, Opera, Google Chrome, Firefox (tùy vào mỗi phân hệ mà nhóm em sẽ chạy môi trường phù hợp với phân hệ đó)
- Phân hệ Website (Admin, Bến xe, website chính thức của hệ thống) được deploy thành công và có link host riêng biệt
- Phân hệ Android được upload thành công trên CH Play

6.2. Đánh giá

6.2.1. Ưu điểm

➤ Cơ bản giải quyết được mục tiêu đề ra:

- + Xây dựng Ứng dụng trên thiết bị di động tìm kiếm bến xe quanh vị trí hiện tại của người dùng hoặc quanh vị trí mà người dùng tìm kiếm trên app; định vị và chỉ đường/diều hướng từ vị trí hiện tại đến bến xe và địa điểm kế tiếp
- + Công nghệ hóa quy trình ra/vào bến xe bằng hình thức đặt trước chỗ đậu và quét mã QR để xác nhận kết thúc quy trình đặt chỗ. Mã QR sẽ được khởi tạo và gửi đến app di động dành cho người dùng sau khi bến xe nhận được yêu cầu đặt chỗ. Trong quá trình đợi bến xe xác nhận hay từ chối yêu cầu, người dùng không thể đặt trước chỗ đậu các bến xe khác nhằm tránh tình trạng boom, spam chức năng hệ thống
- + Web admin dùng để quản lý bến xe và tài khoản người dùng cơ bản quản lý được các yêu cầu cơ bản của một ứng dụng quản lý: thêm, xóa sửa tài khoản; thống kê số lượng yêu cầu đặt chỗ, số lượng tài khoản đăng ký; thống kê được số lượng yêu cầu đặt chỗ thành công và bị từ chối

6.2.2. Hạn chế

- + Chưa kiểm soát được khoảng thời gian di chuyển từ lúc người dùng bấm đặt chỗ trước cho đến khi họ tới bãi xe
- + Chưa xử lý được nghiệp vụ sau khi hoàn thành booking và những luồng xử lý ngoại lệ sau nó
- + Chưa kiểm soát được những khách hàng có ý muốn boom, spam ứng dụng (đặt trước chỗ nhưng không tới, đặt trước chỗ và hủy tác vụ nhiều lần một lúc...)
- + Giao diện phần mềm chưa thực sự chuyên nghiệp và tiện dụng, còn đơn giản.
- + Ứng dụng chưa có chức năng thanh toán trước qua thẻ ngân hàng, thẻ tín dụng,...

6.3. Hướng phát triển

6.3.1. Hướng phát triển backend

- ✓ Dùng thêm Redis để cache dữ liệu, cải thiện tốc độ xử lý của hệ thống.
- ✓ Triển khai hệ thống sử dụng Docker Swarm hoặc Kubernetes thay cho Docker-compose hiện tại.
- ✓ Liên kết với kho dữ liệu bãi giữ xe, luật dừng đỗ, các đường cấm dừng đậu ... của Sở Giao Thông Vận Tải Thành Phố.

6.3.2. Hướng phát triển ứng dụng dành cho tài xế

- + Phát triển chức năng thanh toán trước qua thẻ ngân hàng, thẻ tín dụng, ...
- + Xử lý nghiệp vụ sau khi hoàn thành booking và những luồng xử lý ngoại lệ sau nó
- + Chính sửa, hoàn thành giao diện phần mềm chuyên nghiệp, hiệu ứng đặc sắc hơn

6.3.3. Hướng phát triển ứng dụng dành cho bãi xe

- + Xử lý và lưu lại thời gian dự kiến mà xe sẽ tới bãi tính từ lúc xác nhận đặt chỗ cho đến khi hoàn thành quy trình
- + Xử lý nghiệp vụ sau khi hoàn thành booking và những luồng xử lý ngoại lệ sau nó

6.3.4. Hướng phát triển ứng dụng admin

- + Xử lý hệ thống và lưu lại những negative user vào danh sách đen
- + Có những chế tài và xử phạt hợp lý đối với các negative user (khóa tài khoản, thông báo nhắc nhở với những người dùng đó,...)

Chương 7 PHỤ LỤC VÀ TÀI LIỆU THAM KHẢO

Chương này nói về các vấn đề ngoài lề, chưa được đề cập trong các chương trước

7.1. Phụ lục

7.1.1. Môi trường đã sử dụng để phát triển các module của dự án

- **Ứng dụng di động và ứng dụng giả lập quét mã QR:**
 - Phát triển bằng Android Studio IDE và cài đặt ứng dụng trên điện thoại sử dụng hệ điều hành Android
- **Website chính thức của dự án, website admin và website giả lập ứng dụng dành cho bãi xe:**
 - Phát triển bằng Visual Studio Code chạy trên Windows 10, framework ReactJS, triển khai bằng NGINX và sau cùng được đóng gói thành Docker Image.
- **Backend hệ thống:**
 - Được build bằng IntelliJ IDEA Ultimate chạy trên Ubuntu 18.04, Java 8 (JDK1.8), sử dụng Maven để quản lý các thư viện và biên dịch project, sau cùng được đóng gói thành các Docker Image.

7.1.2. Hướng dẫn cài đặt dự án (dành cho thực nghiệm / kiểm thử tại máy local)

❖ Cài đặt và kiểm thử backend

- Tải xuống và cài đặt docker và docker-compose
- Tải xuống và cài đặt Microsoft SQL Server 2017 và SSMS
- Tải xuống mã nguồn backend của saigonparking
- Chuyển đến thư mục dev trong thư mục source code vừa tải về
- Mở thư mục database, sau đó mở thư mục backup
- Mở SSMS và restore toàn bộ các file backup có trong thư mục
- Trở về thư mục dev, chỉnh sửa cấu hình trong file .env cho phù hợp với máy local
- Mở terminal trong thư mục dev và chạy lệnh sau để khởi động hệ thống: docker-compose up

❖ Cài đặt và kiểm thử webserver

- Tải xuống mã nguồn frontend web của saigonparking
- Mở terminal trong thư mục vừa tải về và chạy lệnh sau để khởi động web-server: docker-compose up

❖ Cài đặt và kiểm thử ứng dụng dành cho tài xế

- Tải xuống mã nguồn saigonparking-android
- Vào thư mục app trong thư mục vừa tải về, sửa IP lại thành IP local của máy tính đang chạy backend trong file build.gradle (sửa GATEWAY_HOST)
- Build APK bằng Android Studio
- Cài đặt và test APK trên điện thoại dùng hệ điều hành android

❖ Cài đặt và kiểm thử ứng dụng giả lập máy quét QR

- Tải xuống mã nguồn saigonparking-qrscanner

- Vào thư mục app trong thư mục vừa tải về, sửa IP lại thành IP local của máy tính đang chạy backend trong file build.gradle (sửa GATEWAY_HOST)
- Build APK bằng Android Studio
- Cài đặt và test APK trên điện thoại dùng hệ điều hành android

7.1.3. Phiên bản của các thư viện, công cụ, plugin sử dụng trong dự án

#	Thư viện, công cụ, plugin	Phiên bản
1	Microsoft SQL Server	2017
2	SQL Server Management Studio	18.6
3	Docker	19.3.12
4	Docker Compose	1.26.2
5	Envoy Proxy	1.12.2
6	Kong Gateway	2.1.2
7	Consul Service Discovery	1.6
8	RabbitMQ	3.8.3
9	NGINX	1.19.0
10	JDK	1.8
11	Maven	3.6.3
12	Gradle	3.5.3
13	NPM	6.12.1
14	Maven Compiler Plugin	3.8.1
15	Maven OS Plugin	1.6.2
16	Maven Ant Run Plugin	3.0.0
17	Maven Docker Plugin	1.2.2
18	Maven Protobuf Plugin	0.6.1
19	Maven Resources Plugin	3.1.0
20	Spring Boot	2.2.5
21	Spring Cloud	2.2.3
22	Spring Framework	5.2.4
23	Javax	4.0.1
24	Java JWT	0.11.2
25	Hibernate	5.4.19
26	MSSQL JDBC	8.2.2.jre8
27	GRPC	1.30.2
28	GRPC Web	1.2

29	Protobuf	3.12.2
30	GRPC Spring Boot Starter	3.5.6
31	Netty Tomcat Native Boring SSL	2.0.30
32	JUnit	4.13
33	Log4J	2.13.3
34	Lombok	1.18.12
35	MapStruct	1.3.1
36	Amazon Web Service Sdk	1.11.829
37	Zxing	3.4.0
38	Apache Common Lang	3.11
39	Java UUID Generator	4.0.1
40	SonarLint	4.9
41	Multirun	1.10
42	SaveActions	2.0.0
43	GrepConsole	11.0
44	ReactJS	16.13.1
45	React Router DOM	16.13.1
46	React Scripts	3.4.1
47	React Modal	3.11.2
48	React Notifications	1.6.0
49	React Bootstrap	1.0.0
50	Android Studio	3.5.3
51	IntelliJ IDEA	2020.1
52	Visual Studio Code	1.48.0

Bảng 7.1.3. Thông kê phiên bản của các thư viện, công cụ, plugin đã sử dụng trong dự án

7.1.4. Chi phí cài đặt và vận hành dự án

7.1.4.1 Amazon RDS

“- Amazon Relational Database Service (Amazon RDS) là một dịch vụ được quản lý giúp bạn dễ dàng thiết lập, vận hành và thay đổi quy mô cơ sở dữ liệu quan hệ trên đám mây.

- Cung cấp tính năng hiệu quả về chi phí và có thể thay đổi quy mô, trong khi vẫn quản lý được các tác vụ quản trị cơ sở dữ liệu tiêu tốn nhiều thời gian, giúp người dùng thoải mái tập trung vào các ứng dụng và công việc của họ.”[1]

- Chi phí cài đặt: Amazon RDS cho phép sử dụng miễn phí 1 năm. Các năm sau đó số tiền phải trả phụ thuộc vào việc sử dụng các dịch vụ của Amazon RDS. (Database Storage, Backup Storage, Data Transfer,...)

- Chi phí cho dịch vụ lưu trữ database của Amazon RDS:

Database Storage

For each DB Instance class, Amazon RDS provides you the ability to select up to 16 TB of associated storage capacity for your primary data set.

General Purpose (SSD) Storage

Amazon RDS provides you the ability to select from 20 GiB to 16 TiB of associated General Purpose (SSD) storage capacity for your primary data set. With General Purpose (SSD), you will be charged for storage you provision. However, you will not be charged for the I/Os you consume.

Single-AZ Deployment	Multi-AZ Deployment
Region: Asia Pacific (Singapore)	
General Purpose (SSD) Storage	\$0.138 per GB-month
Single-AZ Deployment	Multi-AZ Deployment
Region: Asia Pacific (Singapore)	
General Purpose (SSD) Storage	\$0.276 per GB-month

Hình 7.1.4.1.a: Chi phí cài đặt Amazon RDS

7.1.4.2 Amazon S3

.” - Amazon Simple Storage Service (Amazon S3) là một dịch vụ lưu trữ đối tượng cung cấp khả năng thay đổi theo quy mô, tính khả dụng của dữ liệu, bảo mật và hiệu năng hàng đầu trong lĩnh vực. Điều này có nghĩa là khách hàng thuộc mọi quy mô và lĩnh vực có thể sử dụng dịch vụ này để lưu trữ và bảo vệ bất kỳ lượng dữ liệu nào cho nhiều trường hợp sử dụng khác nhau, chẳng hạn như trang web, ứng dụng di động, sao lưu và khôi phục, lưu trữ, ứng dụng doanh nghiệp, thiết bị IoT và phân tích dữ liệu lớn”[2]

- Trong phạm vi đồ án tốt nghiệp, nhóm em sử dụng Amazon S3 để lưu trữ và bảo vệ dữ liệu hình ảnh các bãi xe

- Chi phí cài đặt: miễn phí 1 năm đầu sử dụng. Các năm sau đó, chi phí để sử dụng dịch vụ Amazon S3 được tính như sau:

Storage Requests and data retrievals Data transfer Management and replication

You pay for storing objects in your S3 buckets. The rate you're charged depends on your objects' size, how long you stored the objects during the month, and the storage class—S3 Standard, S3 Intelligent-Tiering, S3 Standard - Infrequent Access, S3 One Zone - Infrequent Access, S3 Glacier, and S3 Glacier Deep Archive, and Reduced Redundancy Storage (RRS). You pay a monthly monitoring and automation fee per object stored in the S3 Intelligent-Tiering storage class to monitor access patterns and move objects between access tiers in S3 Intelligent-Tiering.



There are per-request ingest fees when using PUT, COPY, or lifecycle rules to move data into any S3 storage class. Consider the ingest or transition cost before moving objects into any storage class. Estimate your costs using the [AWS Pricing Calculator](#).

Region: Asia Pacific (Singapore) ▾

[Storage pricing](#)

S3 Standard - General purpose storage for any type of data, typically used for frequently accessed data	
First 50 TB / Month	\$0.025 per GB
Next 450 TB / Month	\$0.024 per GB
Over 500 TB / Month	\$0.023 per GB
S3 Intelligent - Tiering * - Automatic cost savings for data with unknown or changing access patterns	
Frequent Access Tier, First 50 TB / Month	\$0.025 per GB
Frequent Access Tier, Next 450 TB / Month	\$0.024 per GB
Frequent Access Tier, Over 500 TB / Month	\$0.023 per GB
Infrequent Access Tier, All Storage / Month	\$0.02 per GB
Monitoring and Automation, All Storage / Month	\$0.0025 per 1,000 objects
S3 Standard - Infrequent Access * - For long lived but infrequently accessed data that needs millisecond access	
All Storage / Month	\$0.02 per GB
S3 One Zone - Infrequent Access * - For re-createable infrequently accessed data that needs millisecond access	
All Storage / Month	\$0.016 per GB
S3 Glacier ** - For long-term backups and archives with retrieval option from 1 minute to 12 hours	
All Storage / Month	\$0.005 per GB
S3 Glacier Deep Archive ** - For long-term data archiving that is accessed once or twice in a year and can be restored within 12 hours	
All Storage / Month	\$0.002 per GB

Hình 7.1.4.2.a: Chi phí cài đặt Amazon S3

7.1.4.3 Google Place API

“- Là một trong những API Google cung cấp cho nhà phát triển phần mềm, với thông tin chi tiết về hơn 100 triệu địa điểm. Google Place API gồm có 3 công cụ: Place Picker, Place Autocomplete, Place Details:

- *Place Picker* cho phép người dùng tìm kiếm các địa điểm xung quanh vị trí hiện tại của họ.
- *Place Autocomplete* giúp người dùng dễ dàng nhập tên địa điểm và địa chỉ, tự động hoàn thành các truy vấn của người dùng khi họ nhập.
- *Place Detail* cho phép người dùng lấy thông tin chi tiết về một địa điểm như địa chỉ, số điện thoại, trang web, hình ảnh,...”[3]

- Chi phí cài đặt: miễn phí 200USD đầu tiên. Sau 200USD dùng thử, chi phí phải chi trả khi sử dụng các công cụ của Google Place API được tính dựa trên số lượng request đến server của Google. Chi phí cho mỗi request được tính như sau:

Places

APIs in Places include [Places API](#), [Places SDK for Android](#); [Places SDK for iOS](#); [Places Library](#), [Maps JavaScript API](#); [Geocoding API](#); [Geolocation API](#); [Time Zone API](#); and [Elevation API](#).



Note: Many Places requests also return [Places data](#), which is charged in addition to the cost of the Places request. Review the various Places SKUs in the table below for more details.

SKU	\$200 monthly credit Equivalent free usage	Monthly volume range (Price per thousand)
		0–100,000 100,001–500,000 500,001+
Autocomplete requests		
Autocomplete - Per Request	Up to 70,000 requests	\$2.83 \$2.27
Place Autocomplete - Per Request are typically followed by a Places Details call. See below for pricing.		
Query Autocomplete - Per Request	Up to 70,000 requests	\$2.83 \$2.27
CONTACT SALES for volume discounts.		

Query Autocomplete - Per Request are typically followed by a [Places Details](#) call (if user selects a place result), or by a [Places Text Search](#) (if user selects a query, not a result). See below for pricing.

Autocomplete requests, per session – Cost based on Autocomplete request + Places Details request + Places data			
Autocomplete (included with Places Details) - Per Session + Places Details + Basic Data	Up to 11,000 sessions	\$0.00 \$17.00 \$0.00 ----- \$17.00	\$0.00 \$13.60 \$0.00 ----- \$13.60
Total cost:			
Autocomplete (included with Places Details) - Per Session + Places Details + Basic Data + Contact Data	Up to 10,000 sessions	\$0.00 \$17.00 \$0.00 \$3.00 ----- \$20.00	\$0.00 \$13.60 \$0.00 \$2.40 ----- \$16.00
Total cost:			
Autocomplete (included with Places Details) - Per Session + Places Details + Basic Data + Atmosphere Data	Up to 9,000 sessions	\$0.00 \$17.00 \$0.00 \$5.00 ----- \$22.00	\$0.00 \$13.60 \$0.00 \$4.00 ----- \$17.60
Total cost:			
CONTACT SALES for volume discounts.			

Autocomplete (included with Places Details) - Per Session		\$0.00	\$0.00	
+ Places Details		\$17.00	\$13.60	
+ Basic Data	Up to 8,000 sessions	\$0.00	\$0.00	
+ Contact Data		\$3.00	\$2.40	
+ Atmosphere Data		\$5.00	\$4.00	
Total cost:		\$25.00	\$20.00	
Autocomplete requests, per session – when request does not also include a Places Details request				
Autocomplete without Places Details - Per Session	Up to 11,000 sessions	\$17.00	\$13.60	CONTACT SALES
Places Details requests – Cost based on request + data				
Places Details		\$17.00	\$13.60	
+ Basic Data	Up to 11,000 calls	\$0.00	\$0.00	
Total cost:		\$17.00	\$13.60	
Places Details		\$17.00	\$13.60	
+ Basic Data	Up to 10,000 calls	\$0.00	\$0.00	
+ Contact Data		\$3.00	\$2.40	
Total cost:		\$20.00	\$16.00	
Places Details		\$17.00	\$13.60	CONTACT SALES
+ Basic Data	Up to 9,000 calls	\$0.00	\$0.00	
+ Atmosphere Data		\$5.00	\$4.00	
Total cost:		\$22.00	\$17.60	
Places Details		\$17.00	\$13.60	
+ Basic Data	Up to 8,000 calls	\$0.00	\$0.00	
+ Contact Data		\$3.00	\$2.40	
+ Atmosphere Data		\$5.00	\$4.00	
Total cost:		\$25.00	\$20.00	
Places Details – ID Refresh	Unlimited	\$0.00	\$0.00	
Find Place requests – Cost based on request + data				
Find Place		\$17.00	\$13.60	
+ Basic Data	Up to 11,000 calls	\$0.00	\$0.00	
Total cost:		\$17.00	\$13.60	
Find Place		\$17.00	\$13.60	
+ Basic Data	Up to 10,000 calls	\$0.00	\$0.00	
+ Contact Data		\$3.00	\$2.40	
Total cost:		\$20.00	\$16.00	
Find Place		\$17.00	\$13.60	CONTACT SALES
+ Basic Data	Up to 9,000 calls	\$0.00	\$0.00	
+ Atmosphere Data		\$5.00	\$4.00	
Total cost:		\$22.00	\$17.60	
Find Place		\$17.00	\$13.60	
+ Basic Data	Up to 8,000 calls	\$0.00	\$0.00	
+ Contact Data		\$3.00	\$2.40	
+ Atmosphere Data		\$5.00	\$4.00	
Total cost:		\$25.00	\$20.00	

Find Place – ID only	Unlimited	\$0.00	\$0.00
Find Current Place calls – Cust based on call + data			
Find Current Place + Basic Data	Up to 6,500 calls	\$30.00 \$0.00 ----- \$30.00	\$24.00 \$0.00 ----- \$24.00
Total cost:			\$24.00
Find Current Place + Basic Data + Contact Data + Atmosphere Data	Up to 5,000 calls	\$30.00 \$0.00 \$3.00 \$5.00 ----- \$38.00	\$24.00 \$0.00 \$2.40 \$4.00 ----- \$30.40
Total cost:			\$30.40
Other Places requests (Note: Nearby and Text Search requests return all data types by default, triggering all data SKUs.)			
Places Photo	Up to 28,000 calls	\$7.00	\$5.60
Places - Nearby Search + Basic Data + Contact Data + Atmosphere Data	Up to 5,000 calls	\$32.00 \$0.00 \$3.00 \$5.00 ----- \$40.00	\$25.60 \$0.00 \$2.40 \$4.00 ----- \$32.00
Total cost:			\$32.00
Places - Text Search + Basic Data + Contact Data + Atmosphere Data	Up to 5,000 calls	\$32.00 \$0.00 \$3.00 \$5.00 ----- \$40.00	\$25.60 \$0.00 \$2.40 \$4.00 ----- \$32.00
Total cost:			\$32.00

CONTACT SALES
 for volume discounts.

Hình 7.1.4.3.a: Chi phí cài đặt Google Direction API

7.1.4.4 Google Map API

- Là ứng dụng được Google lập trình để sử dụng Maps trên điện thoại và website. Google Map API còn hỗ trợ nhiều dịch vụ khác của Google đặc biệt là dò đường và chỉ đường; hiển thị bản đồ đường sá và những địa điểm (kinh doanh, trường học, bệnh viện, cây ATM...) trong khu vực cũng như khắp nơi trên thế giới.
- Chi phí cài đặt: miễn phí 200USD đầu tiên. Sau 200USD dùng thử, chi phí phải trả được tính theo số lần loading map và số lượng request chỉ đường dò đường từ client gửi tới server Google. Chi phí cài đặt và vận hành được tính như sau:

Maps

APIs in Maps include: [Maps SDK for Android](#); [Maps SDK for iOS](#); [Maps JavaScript API](#); [Maps Static API](#); [Street View API](#); [Maps Embed API](#); and [Maps URLs](#).

SKU	\$200 monthly credit Equivalent free usage	Monthly volume range (Price per thousand)		
		0–100,000	100,001–500,000	500,001+
Mobile Native Static Maps	Unlimited loads	\$0.00	\$0.00	
Mobile Native Dynamic Maps	Unlimited loads	\$0.00	\$0.00	
Embed	Unlimited loads	\$0.00	\$0.00	
Static Maps	Up to 100,000 loads	\$2.00	\$1.60	
Dynamic Maps	Up to 28,000 loads	\$7.00	\$5.60	
Local Context Map beta	Requires enabling Dynamic Maps	\$0.00 during beta	\$0.00 during beta	
Static Street View	Up to 28,000 panos	\$7.00	\$5.60	
Dynamic Street View	Up to 14,000 panos	\$14.00	\$11.20	

CONTACT SALES
 for volume discounts.

Hình 7.1.4.4.a: Chi phí cài đặt Google Map API

7.1.4.5 Google Direction API

Google Maps Directions API là một dịch vụ tính toán đường đi giữa các vị trí. Qua đó người dùng có thể xem hướng dẫn đường đi cụ thể cho một số phương thức vận tải, bao gồm lái xe ô tô, xe máy, đi bộ hoặc đi xe đạp.

- Chi phí cài đặt: miễn phí 200USD đầu tiên. Sau 200USD dùng thử, chi phí phải chi trả phụ thuộc vào số request chỉ đường từ người dùng đến server Google. Mỗi request có bảng giá như sau:

Routes

APIs in Routes include: [Directions API](#); [Distance Matrix API](#); and [Roads API](#).

SKU	\$200 monthly credit Equivalent free usage	Monthly volume range (Price per thousand)		
		0–100,000	100,001–500,000	500,001+
Directions	Up to 40,000 calls	\$5.00	\$4.00	
Directions Advanced	Up to 20,000 calls	\$10.00	\$8.00	
Distance Matrix	Up to 40,000 elements	\$5.00	\$4.00	
Distance Matrix Advanced	Up to 20,000 elements	\$10.00	\$8.00	
Roads - Route Traveled	Up to 20,000 calls	\$10.00	\$8.00	
Roads - Nearest Road	Up to 20,000 calls	\$10.00	\$8.00	
Roads - Speed Limits	Up to 2,000 elements	\$20.00	\$16.00	

[CONTACT SALES](#)

for volume discounts.

Hình 7.1.4.5.a: Chi phí cài đặt Google Direction API

7.1.4.6 Google Virtual Machine

(Xem lại đề mục 5.4 để biết rõ hơn về các service ở tầng public và tầng private)

#	VM	Chi phí ước tính	Cấu hình
1	saigonparkinggateway (máy dùng để chạy các service ở tầng public)	31 USD/tháng (mở máy 24/7)	+ hệ điều hành Ubuntu 18.04 + 1 core 6GB RAM + 20GB SSD + static <i>external</i> IP
2	saigonparkingservice (máy dùng để chạy các service tầng private)	57 USD/tháng (mở máy 24/7)	+ hệ điều hành Ubuntu 18.04 + 2 core 10GB RAM + 20GB SSD + static <i>internal</i> IP

Bảng 7.1.4.6.a: Chi phí cài đặt Google Virtual Machine

7.1.4.7 Gmail

- Dịch vụ email trực tuyến do Google cung cấp tới người dùng.
- Chi phí cài đặt: được Google cung cấp miễn phí

7.1.4.8 Domain và SSL (sử dụng dịch vụ của name.com)

- Miễn phí cho 1 năm đầu tiên. Các năm tiếp theo: 29.99USD/năm
- 1 Domain: *.wtf (domain dùng cho đồ án: saigonparking.wtf)
- 1 SSL dành cho default domain (không dùng được cho subdomain)



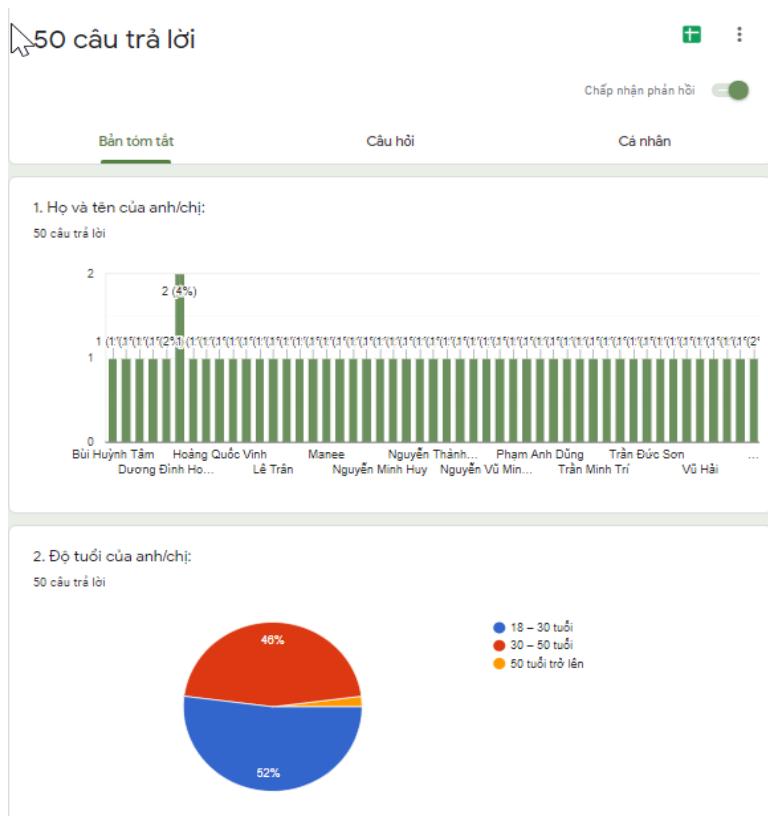
Hình 7.1.4.8.a: chi phí cài đặt Domain và SSL

7.1.4.9 Chi phí khảo sát thực tế

- Quà tặng cho nhân viên bãi xe: 50.000VNĐ

7.1.5. Kết quả khảo sát

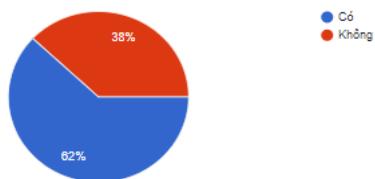
- **Kết quả khảo sát online người thường xuyên sử dụng xe ô tô làm phương tiện di chuyển và có nhu cầu tìm kiếm bãi đỗ**



Hình 7.1.5.a: biểu đồ thống kê họ tên và độ tuổi người tham gia khảo sát

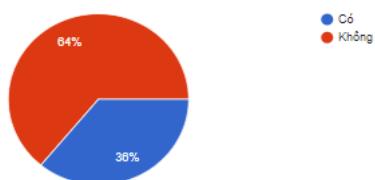
3. Anh/chị có sử dụng xe ôtô thường xuyên hay không ?

50 câu trả lời



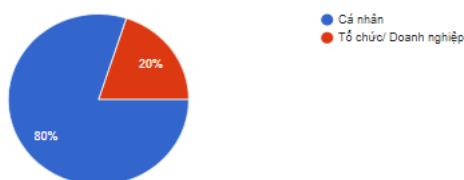
4. Xe ôtô có phải là phương tiện di chuyển chính của anh/chị hay không ?

50 câu trả lời



5. Xe của anh/chị là xe cá nhân hay xe của tổ chức/doanh nghiệp ?

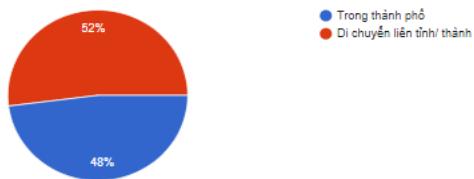
50 câu trả lời



Hình 7.1.5.b: biểu đồ thống kê thói quen và mật độ sử dụng ô tô của người tham gia khảo sát

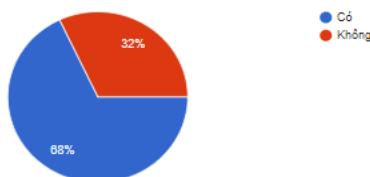
6. Anh/chị thường dùng xe ôtô để di chuyển:

50 câu trả lời



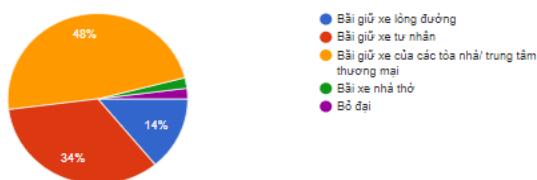
7. Anh/chị có thường sử dụng bãi giữ xe hay không ?

50 câu trả lời



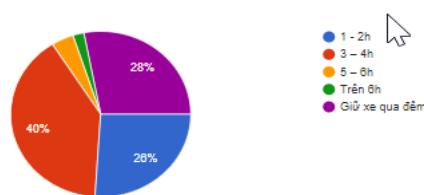
8. Loại bãi giữ xe anh/chị thường sử dụng ?

50 câu trả lời



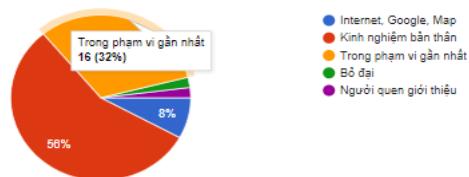
9. Anh/chị thường giữ xe ôtô trong bao lâu ?

50 câu trả lời



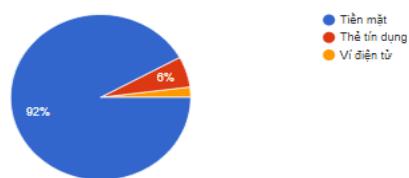
10. Anh/chị thường tìm kiếm chỗ đậu xe thông qua đâu ?

50 câu trả lời



11. Anh/chị thường thanh toán chỗ đậu xe ôtô thông qua hình thức nào ?

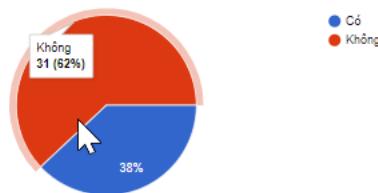
50 câu trả lời



Hình 7.1.5.c – 7.1.5.d: biểu đồ thống kê khảo sát thói quen tìm kiếm bãi đỗ xe và thanh toán của người tham gia khảo sát

12. Anh/chị có nhu cầu đặt trước chỗ đậu xe trong thời gian xác định hay không ?

50 câu trả lời



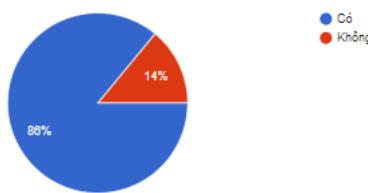
13. Anh/chị thường gặp khó khăn gì trong việc tìm kiếm chỗ đậu xe ôtô ?

35 câu trả lời

- khó kiếm nơi để gửi oto
- Nhiều đường bị cấm đậu
- Dựa theo kinh nghiệm bản thân để tìm chỗ đậu xe nên đôi khi không tìm được chỗ đậu xe ô tô
- Chỗ đậu xe không cho đậu ô tô, chỉ cho đậu xe máy
- Khu vực trung tâm khó
- Nhiều khi hết chỗ giữ xe
- Khó chịu
- Thường hết chỗ đỗ xe
- Không nha

14. Anh/chị có muốn có một ứng dụng di động có thể để xuất chỗ đậu xe gần nhất, so sánh giá giữa các bãi, cũng như cho biết số lượng chỗ còn trống trong bãi hay không ?

50 câu trả lời



15. Nếu có, anh chị có đề xuất, cũng như mong muốn có thêm tính năng nào hay không ?

23 câu trả lời

- Không
- không
- Chưa biết
- Bình luận, đánh giá
- Tìm được chỗ đậu trong vòng 5 phút
- Địa điểm đậu xe uy tín
- Ứng dụng nên thể hiện nhanh chóng các bãi đậu xe còn chỗ trống gần vị trí hiện tại nhất
- Không chỉ ô tô mà trực thăng, du thuyền, limousine, kiệu, cưới voi, trâu, chó... đều tìm được bãi đỗ
- gửi qua mạng hay fb

Hình 7.1.5.e – 7.1.5.f: biểu đồ khảo sát mong muốn và nhu cầu sử dụng ứng dụng tìm bãi đỗ của khách hàng

- Kết quả khảo sát thực tế bãi xe thông qua phỏng vấn trực tiếp

FORM KHẢO SÁT NHU CẦU CÁC BÁI XE Ô TÔ TRÊN ĐỊA BẢN THÀNH PHỐ HỒ CHÍ MINH	
KHẢO SÁT NHU CẦU CÁC BÁI GIỮ XE Ô TÔ HIỆN NAY <p>Khảo sát này hướng đến nhu cầu tìm kiếm cho bãi đậu xe ô tô hiện nay của tập đối tượng là các bãi giữ xe ô tô trong khu vực thành phố Hồ Chí Minh nhằm phát triển một ứng dụng di động phục vụ cho nhu cầu tìm kiếm kiểm báu ô tô; từ đó nâng cao số lượng xe ô tô ra vào bãi, mang lại lợi ích to lớn về mặt kinh tế cho chủ bãi xe ô tô.</p> <p>Khảo sát này được tạo bởi nhóm sinh viên ngành Công Nghệ Thông Tin thuộc trường Đại học Khoa Học Tự Nhiên TP.HCM nhằm hướng đến một sản phẩm thực tế phục vụ cho nhu cầu của xã hội. Chân thành cảm ơn anh/chị đã dành thời gian tham gia khảo sát!</p>	
1.Họ và tên của anh/chị:	<u>Marylin Vinh</u>
2.Độ tuổi của anh/chị:	<input checked="" type="radio"/> 18 - 30 tuổi <input type="radio"/> 30 - 50 tuổi <input type="radio"/> 50 tuổi trở lên
3.Anh/chị đang là nhân viên của bãi xe:	<u>Nhân viên Vinh</u>
4.Địa chỉ bãi xe mà anh/chị đang công tác:	<u>180 Lý Chính Thắng</u>
5.Bãi giữ xe ô tô của anh/chị thuộc loại:	<input type="radio"/> Bãi xe lòng đường <input type="radio"/> Bãi xe tur nhàn <input checked="" type="radio"/> Bãi xe tòe nhả
6.Lượng khách giữ xe ô tô ở bãi xe của anh/chị trong một ngày:	<input checked="" type="radio"/> Nhiều <input type="radio"/> Trung bình <input type="radio"/> Ít
7.Bãi xe mà anh/chị thường xuyên đến:	<u>Khoảng</u>
8.Thời gian anh/chị thường xuyên đến bãi xe:	<u>10h - 17h</u>
9.Anh/chị có muốn cải tiến thêm API (phần mềm mở rộng) vào phần mềm bãi xe hiện tại mà anh/chị đang sử dụng không? (API này được cung cấp từ phía nhà phát triển ứng dụng cho phép bãi xe có thể nhận được yêu cầu đặt trước chỗ từ phía khách hàng, quyết định nhận yêu cầu hoặc hủy tuy thuộc vào số lượng chỗ còn trong trong bãi)	<input checked="" type="radio"/> Có <input type="radio"/> Không
10.Anh/chị có muốn cập nhật thêm tính năng check xe ô tô đã đặt trước chỗ đậu khi vào bãi bằng mã QR hoặc qua biển số xe không?	<input checked="" type="radio"/> Có <input type="radio"/> Không
11.Anh/chị có đề xuất, cần như mong muốn có thêm tính năng nào hay không?	<u>App dành cho khách hàng</u>
12.App dành cho bãi xe:	<u>Khoảng</u>
13.Plugin check xe ô tô vào bãi bằng QR Code hoặc qua biển số xe:	<u>Khoảng</u>

<p>7.Bãi xe mà anh/chị đang công tác quy định số giờ tối đa để giữ xe ô tô trong ban đêm:</p> <p><input checked="" type="radio"/> 1 - 2h <input checked="" type="radio"/> 3 - 4h <input type="radio"/> 5 - 6h <input type="radio"/> Trên 6h <input type="radio"/> Giữ xe qua đêm</p>	
<p>8.Anh/chị có muốn có một ứng dụng di động cho khách hàng sử dụng có thể tìm kiếm bãi đậu xe của anh/chị hay không?</p> <p><input checked="" type="radio"/> Có <input type="radio"/> Không</p> <p>9.Anh/chị có muốn cải tiến thêm API (phần mềm mở rộng) vào phần mềm bãi xe hiện tại mà anh/chị đang sử dụng không? (API này được cung cấp từ phía nhà phát triển ứng dụng cho phép bãi xe có thể nhận được yêu cầu đặt trước chỗ từ phía khách hàng, quyết định nhận yêu cầu hoặc hủy tuy thuộc vào số lượng chỗ còn trong trong bãi)</p> <p><input checked="" type="radio"/> Có <input type="radio"/> Không</p> <p>10.Anh/chị có muốn cập nhật thêm tính năng check xe ô tô đã đặt trước chỗ đậu khi vào bãi bằng mã QR hoặc qua biển số xe không?</p> <p><input checked="" type="radio"/> Có <input type="radio"/> Không</p> <p>11.Anh/chị có đề xuất, cần như mong muốn có thêm tính năng nào hay không?</p> <p>App dành cho khách hàng</p>	
<p>12.App dành cho bãi xe:</p> <p>Khoảng</p>	
<p>13.Plugin check xe ô tô vào bãi bằng QR Code hoặc qua biển số xe:</p> <p>Khoảng</p>	

TP.HCM, Ngày 25/7/2019, Số: Năm 2019
Họ và tên người làm khảo sát:
Marylin

Hình 7.1.5.h – 7.1.5.g: biểu mẫu khảo sát nhân viên bãi xe

7.1.6. Giới thiệu nhóm phát triển dự án Saigon Parking

Nhóm gồm 5 thành viên và thầy giáo hướng dẫn đề án:

- Thầy Ngô Ngọc Đăng Khoa với vai trò là giáo viên hướng dẫn đề án
- Phạm Viết Minh Đạo với vai trò là Business Analyst và Tester
- Huỳnh Thanh Bình với vai trò là Backend và DevOps Developer
- Vũ Tường Bách với vai trò là Frontend Website và Frontend Android
- Vũ Hải với vai trò là Frontend Website và Frontend Android
- Đăng Đức Tài với vai trò là Frontend Website và Frontend Android
- Đồng thời, Phạm Viết Minh Đạo còn có vai trò quản lý chi phí, phân chia và quản lý task và deadline, lịch meeting hoặc gặp gỡ thầy giáo hướng dẫn phụ trách

7.1.7. Phân công công việc

#	MSSV	Họ và tên	Phân công
1	1653005	Vũ Tường Bách	<ul style="list-style-type: none"> - Quản lý tiến độ thực hiện dự án - Quản lý team frontend - Phân tích nghiệp vụ hệ thống - Xây dựng và test app mobile dành cho khách hàng giữ xe - Viết báo cáo dự án
2	1653006	Huỳnh Thanh Bình	<ul style="list-style-type: none"> - Thiết kế kiến trúc hệ thống - Xây dựng cơ sở dữ liệu hệ thống - Xây dựng backend hệ thống - Xây dựng API - Triển khai dự án lên host thực tế - Viết báo cáo dự án
3	1653012	Phạm Việt Minh Đạo	<ul style="list-style-type: none"> - Thu thập, thống kê, nhập liệu data - Thiết kế use-case diagram cho dự án - Thiết kế activity diagram cho dự án - Thiết kế prototype cho các ứng dụng - Viết báo cáo dự án
4	1653020	Vũ Hải	<ul style="list-style-type: none"> - Khảo sát nhóm ứng dụng tương tự, đã có trên thị trường, phân tích ưu, khuyết điểm, đề xuất hướng thực hiện dự án - Thiết kế và xử lý các tài nguyên liên quan đồ họa - Xây dựng và test website quản trị viên - Xây dựng và test website giả lập ứng dụng bãi xe - Viết báo cáo dự án
5	1653076	Đặng Đức Tài	<ul style="list-style-type: none"> - Khảo sát nhu cầu client, đề xuất chức năng cần triển khai - Thông kê yêu cầu chức năng, phi chức năng của dự án - Xây dựng và test app mobile dành cho khách hàng giữ xe - Xây dựng và test app mobile giả lập máy quét QR code - Viết báo cáo dự án

7.2. Tài liệu tham khảo

--- Saigon Parking Project ---
Lớp 16CLC1 - Khóa 2016
Khoa Công Nghệ Thông Tin
ĐH KHTN – ĐHQG.TPHCM