

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Hệ thống quản lý nhắn tin đa kênh

PHÍ KHÁNH HUYỀN

huyen.pk151748@sis.hust.edu.vn

Ngành Kỹ thuật phần mềm

Giảng viên hướng dẫn: TS. Nguyễn Thị Thu Trang _____

Bộ môn: Công nghệ phần mềm

Viện: Công nghệ thông tin – Truyền thông

HÀ NỘI, 6/2020

Lời cam kết

Họ và tên sinh viên: Phí Khánh Huyền

Điện thoại liên lạc: (+84) 968 036 042

Email: huyen.pk151748@sis.hust.edu.vn

Lớp: CNTT 2.03 – K60

Hệ đào tạo: Đại học chính quy

Tôi – *Phí Khánh Huyền* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Nguyễn Thị Thu Trang*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 26 tháng 6 năm 2020

Tác giả ĐATN

Phí Khánh Huyền

Lời cảm ơn

Lời đầu tiên em xin gửi lời cảm ơn chân thành tới các thầy cô giáo trường Đại học Bách Khoa Hà Nội nói chung và thầy cô giáo trong Viện Công nghệ thông tin và Truyền thông nói riêng đã tạo cho em một môi trường học tập bổ ích, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập. Đặc biệt em xin gửi lời cảm ơn sâu sắc nhất tới Cô TS. Nguyễn Thị Thu Trang, Bộ môn Công nghệ phần mềm – Viện Công nghệ thông tin và Truyền thông – Trường Đại học Bách Khoa Hà Nội. Trong quá trình thực hiện nghiên cứu, Cô đã hướng dẫn và chỉ bảo em ân cần, những lời góp ý của Cô giúp em định hướng tốt hơn, đúng đắn hơn cho đề tài mình đã chọn.

Lời tiếp theo em xin dành lời cảm ơn tới các anh chị, bạn bè, các thành viên trong Lab 914 đã quan tâm, chia sẻ, ủng hộ về tinh thần lớn lao để em có động lực học tập và hoàn thành ĐATN này.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè đã luôn tạo điều kiện, giúp đỡ động viên em trong suốt quá trình học tập và hoàn thành ĐATN.

Trong quá trình xây dựng và hoàn thiện báo cáo cũng như đồ án tốt nghiệp, em đã cố gắng giảm thiểu, tránh các sai sót ở mức thấp nhất có thể. Tuy nhiên, không có gì là hoàn hảo và đồ án không thể tránh khỏi những thiếu sót, vì vậy rất mong các thầy cô và các bạn đọc góp ý để em có thể hoàn thiện hơn nữa sản phẩm này.

Một lần nữa, em xin chân thành cảm ơn!

Tóm tắt

Khách hàng là yếu tố quyết định đến sự thành công hay thất bại của doanh nghiệp. Chính vì thế hiểu và hỗ trợ khách hàng là mục tiêu hàng đầu mà bất kỳ doanh nghiệp nào cũng đều hướng tới. Để thực hiện được việc này thì doanh nghiệp cần tương tác với khách hàng nhiều hơn và livechat là giải pháp mà rất nhiều doanh nghiệp lựa chọn để tiếp cận với khách hàng tiềm năng. Livechat là giải pháp trò chuyện trực tuyến cho phép doanh nghiệp theo dõi và trò chuyện với khách truy cập website của mình, giúp kết nối và hỗ trợ những thông tin bổ ích nhất đối với từng khách hàng.

Vì vậy, mục đích của đề án tốt nghiệp là phát triển một hệ thống quản lý nhắn tin đa kênh và một cửa sổ livechat có thể tích hợp vào bất kỳ website nào. Hệ thống này cho phép tạo ra các ứng dụng quản lý nhắn tin đa kênh phục vụ cho việc kết nối giữa các ứng dụng nhắn tin bên thứ ba và chatbot (một hệ thống kết hợp với trí tuệ nhân tạo để tương tác với con người). Đồng thời, ứng dụng này cũng hỗ trợ các tính năng để doanh nghiệp quản lý và giao tiếp với khách hàng như thống kê số lượng khách hàng, nhắn tin hỗ trợ khách hàng, gửi tin nhắn hàng loạt trên nhiều kênh. Cửa sổ livechat cho phép người dùng có thể nhắn tin với doanh nghiệp thông qua các dạng tin nhắn như văn bản, hình ảnh, âm thanh, video.

Hệ thống quản lý nhắn tin đa kênh hướng đến một sản phẩm dành cho người dùng thực sự, do đó cần đáp ứng được lượng lớn người dùng trong tương lai. Chính vì thế, hệ thống đã được thiết kế theo mô hình kiến trúc nhiều dịch vụ nhỏ Microservices, phần backend sử dụng NodeJS và frontend sử dụng thư viện ReactJS để dễ dàng bảo trì và mở rộng. Do hệ thống cho phép kết nối nhiều kênh nhắn tin nên em đã thiết kế một mô hình nhắn tin hàng loạt đa kênh để doanh nghiệp dễ dàng tương tác với khách hàng của mình. Cửa sổ livechat đã được đóng gói thành một script để có thể nhúng dễ dàng vào các website.

Hiện tại hệ thống quản lý nhắn tin đa kênh và cửa sổ livechat đã được triển khai trên server với tên miền <https://dashboard.iristech.club> và <https://livechat.iristech.club>. Cả hai đều đã được đưa vào thử nghiệm trong các lĩnh vực ngân hàng (Vietcombank), tài chính (FE Credit) và viễn thông (Viettel). Không chỉ dừng lại đó, hệ thống vẫn đang trong quá trình phát triển và hướng tới nhiều chức năng chuyên nghiệp hơn.

Mục lục

Lời cam kết	ii
Lời cảm ơn.....	iii
Tóm tắt.....	iv
Mục lục	v
Danh mục hình vẽ.....	viii
Danh mục bảng.....	x
Danh mục các từ viết tắt	xii
Danh mục thuật ngữ.....	xiv
Chương 1 Giới thiệu đề tài.....	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp	2
1.4 Bố cục đồ án.....	3
Chương 2 Khảo sát và phân tích yêu cầu	4
2.1 Khảo sát hiện trạng.....	4
2.2 Tổng quan chức năng.....	6
2.2.1 Hệ thống quản lý nhắn tin đa kênh.....	6
2.2.2 Cửa sổ livechat trên nền tảng web.....	7
2.2.3 Biểu đồ use case phân rã.....	8
2.3 Đặc tả chức năng	9
2.3.1 Đặc tả use case Hỗ trợ trực tuyến.....	11

2.3.2	Đặc tả use case Tích hợp Zalo.....	12
2.3.3	Đặc tả use case Tạo chiến dịch quảng cáo	13
2.3.4	Đặc tả use case Gửi tin nhắn video.....	16
2.4	Yêu cầu phi chức năng.....	17
2.4.1	Tính dễ dùng	17
2.4.2	Tính dễ bảo trì	17
2.4.3	Tính khả chuyển	17
Chương 3 Công nghệ sử dụng.....		18
3.1	Frontend	18
3.1.1	ReactJS.....	18
3.1.2	Material UI.....	19
3.1.3	Styled-Components	19
3.2	Backend	20
3.2.1	Kiến trúc Microservices.....	20
3.2.2	NodeJS	21
3.2.3	MongoDB.....	21
3.3	Giao thức kết nối	21
3.3.1	Websocket.....	21
3.3.2	RabbitMQ.....	22
3.3.3	Rest API.....	22
3.4	DevOps	23
Chương 4 Phát triển và triển khai ứng dụng		24
4.1	Thiết kế kiến trúc.....	24
4.2	Thiết kế chi tiết frontend.....	25
4.2.1	Thiết kế mockup	25
4.2.2	Thiết kế thành phần giao diện	28
4.3	Thiết kế chi tiết backend.....	31

4.3.1	Luồng hoạt động.....	31
4.3.2	Thiết kế API.....	36
4.4	Thiết kế cơ sở dữ liệu	37
4.4.1	Cơ sở dữ liệu dịch vụ quản lý	38
4.4.2	Cơ sở dữ liệu dịch vụ nhắn tin	40
4.5	Xây dựng ứng dụng	42
4.5.1	Thư viện và công cụ sử dụng	42
4.5.2	Kết quả đạt được.....	43
4.5.3	Minh hoạ các chức năng chính.....	44
4.6	Kiểm thử và triển khai	48
Chương 5 Các giải pháp và đóng góp nổi bật.....		52
5.1	Thiết kế kiến trúc hệ thống sử dụng Microservices	52
5.1.1	Vấn đề.....	52
5.1.2	Giải pháp.....	53
5.2	Mô hình nhắn tin hàng loạt đa kênh	58
5.2.1	Vấn đề.....	58
5.2.2	Giải pháp.....	58
5.2.3	Kết quả.....	60
5.3	Phát triển cửa sổ livechat trên nền tảng web	63
Chương 6 Kết luận và hướng phát triển.....		65
6.1	Kết luận.....	65
6.2	Hướng phát triển.....	66
Tài liệu tham khảo.....		67
Phụ lục		A-1

Danh mục hình vẽ

Hình 1 Biểu đồ use case tổng quan hệ thống quản lý nhắn tin đa kênh	6
Hình 2 Biểu đồ use case tổng quan cửa sổ livechat.....	7
Hình 3 Biểu đồ use case phân rã quảng cáo.....	8
Hình 4 Biểu đồ use case phân rã tích hợp kênh nhắn tin	9
Hình 5 Biểu đồ use case phân rã thống kê	9
Hình 6 Mô hình Microservices.....	20
Hình 7 Kiến trúc tổng quan hệ thống quản lý nhắn tin đa kênh.....	25
Hình 8 Mockup layout của hệ thống quản lý nhắn tin đa kênh.....	26
Hình 9 Mockup cửa sổ livechat.....	27
Hình 10 Thiết kế thành phần giao diện màn hình thống kê	28
Hình 11 Thiết kế thành phần giao diện màn hình hỗ trợ trực tuyến.....	29
Hình 12 Thiết kế thành phần giao diện màn hình tích hợp.....	29
Hình 13 Thiết kế thành phần giao diện màn hình tạo nội dung chiến dịch quảng cáo.....	30
Hình 14 Biểu đồ trình tự gửi quảng cáo Facebook.....	32
Hình 15 Biểu đồ trình tự gửi quảng cáo Zalo	32
Hình 16 Biểu đồ trình tự gửi quảng cáo Viber.....	33
Hình 17 Biểu đồ trình tự Tạo chiến dịch quảng cáo.....	34
Hình 18 Biểu đồ trình tự tích hợp Zalo.....	35
Hình 19 Biểu đồ thực thể liên kết dịch vụ quản lý	38
Hình 20 Thiết kế tổng quan cơ sở dữ liệu dịch vụ quản lý	38
Hình 21 Biểu đồ thực thể liên kết dịch vụ nhắn tin.....	40

Hình 22 Thiết kế tổng quan cơ sở dữ liệu dịch vụ nhắn tin	41
Hình 23 Giao diện thống kê	45
Hình 24 Giao diện hỗ trợ trực tuyến.....	46
Hình 25 Giao diện tích hợp 1	46
Hình 26 Giao diện tích hợp 2	47
Hình 27 Giao diện cửa sổ livechat.....	48
Hình 28 Thiết kế kiến trúc Microservices.....	53
Hình 29 Kiến trúc dịch vụ Quản lý.....	54
Hình 30 Kiến trúc dịch vụ Nhắn tin	55
Hình 31 Kiến trúc dịch vụ Tải tập tin	56
Hình 32 Kiến trúc frontend hệ thống quản lý nhắn tin đa kênh	57
Hình 33 Thành phần trong giao diện tạo nội dung quảng cáo	59
Hình 34 Mô hình nhắn tin hàng loạt đa kênh.....	59
Hình 35 Giao diện tạo nội dung chiến dịch quảng cáo	61
Hình 36 Giao diện chọn đối tượng gửi quảng cáo.....	62
Hình 37 Giao diện đặt lịch gửi quảng cáo	62
Hình 38 Giao diện xác nhận gửi quảng cáo	63

Danh mục bảng

Bảng 1 So sánh ứng dụng LiveChat, tawk.to, vChat	4
Bảng 2 Danh sách use case.....	10
Bảng 3 Đặc tả use case Hỗ trợ trực tuyến	11
Bảng 4 Đặc tả use case Tích hợp Zalo.....	12
Bảng 5 Đặc tả use case Tạo chiến dịch quảng cáo	13
Bảng 6 Đặc tả use case Gửi tin nhắn video.....	16
Bảng 7 Bộ cấu hình chung cho giao diện quản lý	26
Bảng 8 Bảng màu của sổ livechat.....	28
Bảng 9 Danh sách API của hệ thống quản lý nhắn tin đa kênh.....	36
Bảng 10 Danh sách hàng đợi tin nhắn	37
Bảng 11 Thiết kế chi tiết cơ sở dữ liệu dịch vụ quản lý	39
Bảng 12 Thiết kế chi tiết cơ sở dữ liệu dịch vụ nhắn tin	41
Bảng 13 Danh sách thư viện và công cụ sử dụng.....	42
Bảng 14 Các đường dẫn các dịch vụ và giao diện của hệ thống	43
Bảng 15 Danh sách test case	48
Bảng 16 Thông số cấu hình server triển khai hệ thống.....	51
Bảng 17 Đặc tả use case Chia sẻ ứng dụng quản lý nhắn tin đa kênh	A-1
Bảng 18 Đặc tả use case Tích hợp Facebook Messenger	A-2
Bảng 19 Đặc tả use case Tích hợp Viber	A-2
Bảng 20 Đặc tả use case Thống kê số lượng khách hàng theo kênh nhắn tin.....	A-3
Bảng 21 Các trường hợp kiểm thử cho “Tạo nội dung chiến dịch quảng cáo”	B-5

Bảng 22 Các trường hợp kiểm thử cho “Gửi tin nhắn video”	B-8
---	-----

Danh mục các từ viết tắt

ĐATN	Đồ án tốt nghiệp
CSDL	Cơ sở dữ liệu
HTTP	HyperText Transfer Protocol
SPA	Single-page Application
API	Application Programming Interface
JSON	JavaScript Object Notation
UI	User Interface
HTML	Hypertext Markup Language
XML	Extensible Markup Language
CSS	Cascading Style Sheets
JSX	JavaScript XML
DOM	Document Object Model
AMQP	Advanced Message Queue Protocol

URL	Uniform Resource Locator
ODM	Object Data Model
RAM	Random-Access Memory
SSD	Solid-State Drive
DevOps	Software Development and Information Technology Operations

Danh mục thuật ngữ

Chatbot

Hệ thống kết hợp với trí tuệ nhân tạo để tương tác với con người

Livechat

Công cụ nhắn tin trực tuyến trên website

NU

Number of User – Số lượng khách hàng của hệ thống

DAU

Daily Active User – Khách hàng nhắn tin trong ngày đến hệ thống

Chương 1 Giới thiệu đề tài

Chương 1 giới thiệu những vấn đề thực tế dẫn tới lý do chọn đề tài, tổng quan về hệ thống quản lý nhắn tin đa kênh và cửa sổ livechat. Sau đó đưa ra mục tiêu và phạm vi của đề tài, định hướng giải pháp và bố cục trình bày của đồ án.

1.1 Đặt vấn đề

Khách hàng là yếu tố quan trọng trong sự tồn tại và phát triển của doanh nghiệp. Để đảm bảo rằng khách hàng hài lòng khi sử dụng sản phẩm/dịch vụ, họ cần nhận được sự hỗ trợ kịp thời và chính xác khi gặp khó khăn trong quá trình sử dụng. Hiện nay có rất nhiều kênh hỗ trợ khách hàng khác nhau như: hỗ trợ khách hàng trực tiếp, hỗ trợ khách hàng qua điện thoại, hỗ trợ khách hàng qua email, hỗ trợ khách hàng qua livechat, v.v.

Trong đó, livechat là giải pháp nhắn tin trực tuyến cho phép doanh nghiệp theo dõi và trò chuyện với khách truy cập website của mình. Thông qua đây, khách hàng sẽ được hỗ trợ nhanh và tiện lợi hơn rất nhiều so với việc trực tiếp gặp nhân viên hỗ trợ hay qua email và điện thoại, vì họ không cần mất công đến cửa hàng hay mất thời gian chờ đợi nhận được sự hỗ trợ. Với sự có mặt của livechat trên website, khách hàng sẽ nhanh chóng và dễ dàng kết nối được với công ty, doanh nghiệp. Điều này vừa tạo được thiện cảm với khách hàng lại vừa gia tăng cơ hội bán hàng.

Hiện nay đã có các ứng dụng livechat hỗ trợ khách hàng cho website như: tawk.to, LiveChat, vChat. Những ứng dụng này cung cấp đầy đủ chức năng nên có ở một phần mềm livechat, tuy nhiên, vẫn còn một số điểm hạn chế như tính năng tùy chỉnh nhiều nhưng doanh nghiệp cần có thêm tính năng đặc thù riêng, hoặc có hỗ trợ tích hợp nhưng hạn chế kênh kết nối. Bên cạnh đó, với những ứng dụng này, một tài khoản chỉ có thể quản lý một ứng dụng livechat.

Chính vì vậy em đã thực hiện đề tài phát triển “Hệ thống quản lý nhắn tin đa kênh” nhằm giải quyết các vấn đề nêu trên. Đồng thời, khi tự phát triển hệ thống riêng, em có thể thiết kế thêm tính năng mới phù hợp với yêu cầu của khách hàng.

1.2 Mục tiêu và phạm vi đề tài

Với mục đích đã trình bày ở phần 1.1, mục tiêu đặt ra là phát triển (i) hệ thống quản lý nhắn tin đa kênh và (ii) cửa sổ livechat có thể tích hợp với bất kỳ website nào. Trong đó, hệ thống quản lý cho phép tạo ra các ứng dụng để kết nối nhiều kênh nhắn tin và quản lý khách hàng

từ nhiều kênh khác nhau tại cùng một nơi. Với mỗi ứng dụng, hệ thống cần hiển thị các thông tin tổng quan về số lượng khách hàng cũng như số phiên hội thoại, qua đó doanh nghiệp có thể nắm bắt nhanh được tình hình hoạt động. Đồng thời hệ thống cũng cần có hướng dẫn chi tiết để người dùng có thể tự tích hợp cửa sổ livechat vào website dịch vụ của họ và có thể tích hợp với nhiều kênh nhắn tin khác nhau. Bên cạnh đó, doanh nghiệp có thể tương tác với khách hàng bằng cách để nhân viên trực tiếp nhắn tin hỗ trợ khách hàng hoặc kết nối với các chatbot tạo ra từ SmartDialog Platform (một nền tảng chatbot thông minh) để trả lời tự động. Doanh nghiệp cũng có thể tạo ra các chiến dịch quảng cáo với nội dung phong phú và hướng đến nhiều đối tượng khác nhau. Cửa sổ livechat cần dễ dàng tích hợp vào website và đáp ứng các hình thức tin nhắn cơ bản như: tin nhắn văn bản, tin nhắn hình ảnh, tin nhắn âm thanh, tin nhắn video và tin nhắn lựa chọn.

Để xây dựng được một hệ thống như trên, em cần tìm hiểu, lựa chọn và thiết kế một kiến trúc đáp ứng được nhiều người sử dụng cùng một lúc, có hiệu năng tốt. Đồng thời cung cấp được các tính năng như: thống kê, kết nối nhiều kênh nhắn tin, tích hợp với chatbot, quảng cáo trên nhiều tập khách hàng khác nhau. Bên cạnh đó, hệ thống phải là một sản phẩm có giao diện thân thiện, trải nghiệm của người dùng tốt và tương thích với các loại màn hình. Hệ thống cần được thiết kế để đáp ứng được các yêu cầu trên và phải dễ dàng tìm kiếm, sửa lỗi, triển khai, mở rộng thêm chức năng mới.

1.3 Định hướng giải pháp

Do hệ thống quản lý nhắn tin đa kênh có rất nhiều chức năng, mỗi chức năng có những đặc điểm chung để gom thành nhóm chức năng độc lập, bên cạnh đó, số lượng khách hàng nhắn tin đến hệ thống là rất nhiều, nhưng số lượng người sử dụng trang quản lý lại ít, nên cần thiết kế để có thể mở rộng hệ thống đáp ứng được số lượng lớn khách hàng, đồng thời không gây lãng phí tài nguyên. Vì vậy, em lựa chọn mô hình kiến trúc nhiều dịch vụ nhỏ Microservices để xây dựng hệ thống. Mỗi dịch vụ đảm nhiệm một chức năng riêng biệt và giao tiếp với nhau qua HTTP hoặc Message Queue tùy thuộc vào mục đích giao tiếp.

Hệ thống này cho phép kết nối với nhiều kênh nhắn tin khác nhau, mỗi kênh lại có những đặc điểm riêng, để có thể tiếp cận với rất nhiều khách hàng trên nhiều kênh như vậy là không hề đơn giản. Chính vì vậy, em đã thiết kế một mô hình nhắn tin hàng loạt đa kênh. Để đạt được hiệu quả giao tiếp với khách hàng, doanh nghiệp chỉ cần tạo ra nội dung tin nhắn, lựa chọn đối tượng gửi tin và mô hình nhắn tin hàng loạt sẽ giúp chuyển tin nhắn đó đến đúng đối tượng khách hàng mong muốn.

Để xây dựng được một hệ thống như vậy em đã lựa chọn sử dụng Single-page Application (SPA). Với SPA thì frontend và backend được xây dựng tách biệt hoàn toàn và làm việc thông qua API. Phần frontend em sử dụng ReactJS vì đây là một thư viện tập trung chủ yếu

phát triển vào giao diện và ReactJS phân chia mọi thứ thành các thành phần để thuận lợi cho việc tái sử dụng, mở rộng ứng dụng. Về phần backend, các dịch vụ đều được xây dựng bằng NodeJS vì việc xuất hiện của JavaScript xuyên suốt ứng dụng web từ client đến server đem lại rất nhiều lợi ích, rõ ràng nhất là về việc đồng nhất ngôn ngữ cũng như định dạng dữ liệu trao đổi (JSON). Việc lưu trữ dữ liệu ở các dịch vụ thì em lựa chọn MongoDB vì dữ liệu lưu trong Mongo có cấu trúc linh hoạt hơn SQL, hỗ trợ các cấu trúc phi chuẩn, tốc độ ghi và update nhanh hơn MySQL, phù hợp với nhu cầu của hệ thống. MongoDB với NodeJS sẽ tạo nên một sự kết hợp tốt cho việc xây dựng ứng dụng web.

Đối với cửa sổ livechat, để có thể tích hợp dễ dàng vào các website em đã phát triển cửa sổ này dưới dạng một thành phần và được đóng gói thành script có thể sử dụng ngay. Cụ thể, em lựa chọn xây dựng thành tập tin javascript để áp dụng vào trang html vì javascript là một ngôn ngữ lập trình cho website và nó được hỗ trợ bởi hầu hết các trình duyệt web cũng như các trình duyệt di động cho điện thoại thông minh. Tuy nhiên để hạn chế tối đa kích thước của tập tin javascript thì thay vì dùng thư viện hỗ trợ viết giao diện, em đã tự viết CSS cho cửa sổ livechat.

1.4 Bố cục đồ án

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

Trong Chương 2 em sẽ trình bày quá trình phân tích yêu cầu. Từ việc khảo sát hiện trạng đến việc đưa ra tổng quan các chức năng và đi sâu vào làm rõ một số chức năng chính thông qua các biểu đồ use case.

Trong Chương 3 em sẽ trình bày về kiến trúc, công nghệ sử dụng cho việc xây dựng hệ thống quản lý nhắn tin đa kênh và cửa sổ livechat. Phía frontend bao gồm các công nghệ như: ReactJS, Material UI, Styled-Components. Phía backend như NodeJS, RabbitMQ. Giao thức kết nối như: Websocket, RabbitMQ, REST API.

Trong Chương 4 em sẽ trình bày chi tiết quá trình phân tích thiết kế, xây dựng và triển khai hệ thống.

Trong Chương 5 em sẽ trình bày các giải pháp và đóng góp nổi bật của bản thân trong suốt quá trình làm ĐATN như xây dựng kiến trúc, phát triển mô hình nhắn tin hàng loạt đa kênh và cách giải quyết khi gặp khó khăn trong quá trình làm đồ án.

Trong Chương 6 em sẽ kết luận lại nội dung của đồ án, nêu ra các ưu, nhược điểm, những gì đã làm được, những gì chưa làm được. Đồng thời đề xuất các định hướng phát triển tương lai.

Sau đây em sẽ đi vào chi tiết từng phần của đồ án.

Chương 2 Khảo sát và phân tích yêu cầu

Chương 2 em sẽ trình bày khảo sát các ứng dụng livechat trên thị trường hiện nay. Từ đó đưa ra tổng quan chức năng của hệ thống quản lý nhắn tin đa kênh và đi vào chi tiết làm rõ từng chức năng.

2.1 Khảo sát hiện trạng

Hiện nay trên thị trường đã có rất nhiều ứng dụng livechat với các tính năng phong phú khác nhau. Trong đó, LiveChat và tawk.to là hai ứng dụng nổi bật nhất. Sau đây em sẽ phân tích ưu nhược điểm của ứng dụng LiveChat [12], ứng dụng tawk.to [13] và ứng dụng vChat [14] là một ứng dụng do người Việt phát triển.

Bảng 1 So sánh ứng dụng LiveChat, tawk.to, vChat

Tiêu chí	LiveChat	tawk.to	vChat
Giá cả	Trả phí	Miễn phí	Miễn phí và trả phí
Ngôn ngữ	45 ngôn ngữ (không có tiếng Việt)	27 ngôn ngữ (có tiếng Việt)	4 ngôn ngữ (có tiếng Việt)
Thị trường	Đa quốc gia	Đa quốc gia	Việt nam
Tích hợp nền tảng nhắn tin	Facebook Messenger, Twilio SMS	Không	Không
Tích hợp chatbot	Có	Có	Có
Thống kê	Có	Có	Có
Gửi tin nhắn hàng loạt	Không	Không	Không

LiveChat là nền tảng trò chuyện trực tuyến bao gồm ba thành phần chính là: cửa sổ trò chuyện, bảng điều khiển trên nền tảng web và ứng dụng vận hành đa nền tảng cho phép đăng nhập và trò chuyện với khách truy cập thông qua máy tính để bàn và thiết bị di động. LiveChat cung cấp một loạt các tính năng như: tích hợp bên thứ ba, chia sẻ tệp, lập lịch làm việc, thống kê và phân tích, báo cáo nâng cao, bảo mật dữ liệu, tùy chỉnh giao diện và nhiều hơn nữa. Mặc dù rất nhiều tính năng nhưng LiveChat không hề khó sử dụng mà rất trực quan và dễ điều hướng. Tuy nhiên, LiveChat không hỗ trợ tiếng Việt, ngoài tiếng Anh thì nó còn hỗ trợ thêm một số ngôn ngữ chính bao gồm tiếng Hà Lan, tiếng Pháp, tiếng Đức, tiếng Ý, tiếng Ba Lan, tiếng Bồ Đào Nha, tiếng Tây Ban Nha, tiếng Nga, tiếng Trung Quốc và tiếng Thái.

tawk.to là một ứng dụng nhắn tin và hỗ trợ trò chuyện trực tuyến tập trung vào giao tiếp giữa doanh nghiệp và khách hàng của họ. Với tawk.to, doanh nghiệp có thể tiếp cận khách hàng trực tiếp từ website hoặc ứng dụng di động với việc thiết lập đơn giản. tawk.to cho phép doanh nghiệp cá nhân hóa thương hiệu của mình bằng việc cho phép thay đổi logo, màu sắc cuộc trò chuyện. Trong khi trò chuyện với khách hàng, ứng dụng cho phép doanh nghiệp biết khách hàng sẽ viết gì trên cửa sổ trò chuyện trước khi họ gửi tin nhắn, từ đó có thể chuẩn bị câu trả lời trước khi khách hàng hỏi, giúp thu hẹp khoảng cách giao tiếp. Các tính năng thống kê, báo cáo hàng tháng, hàng ngày cho phép theo dõi sự quan tâm của khách hàng đối với website. Không quan trọng người dùng đang ở đâu hay nói ngôn ngữ gì, tawk.to có sẵn trong 27 ngôn ngữ và hoạt động tốt với Android, iOS, Windows và macOS. tawk.to có rất nhiều tính năng, tuy nhiên không phải doanh nghiệp nào cũng cần sử dụng tất cả các tính năng đó, vì vậy doanh nghiệp cần lựa chọn sao cho phù hợp với nhu cầu của mình. Ở thị trường Việt Nam, rất nhiều khách hàng đặc biệt là khách hàng trẻ thường có nhu cầu tiếp cận với doanh nghiệp thông qua các kênh nhắn tin như Facebook Messenger, Zalo, Viber, v.v. thay vì vào trang web của doanh nghiệp, tuy nhiên tawk.to không cung cấp tính năng tích hợp với các kênh nhắn tin khác.

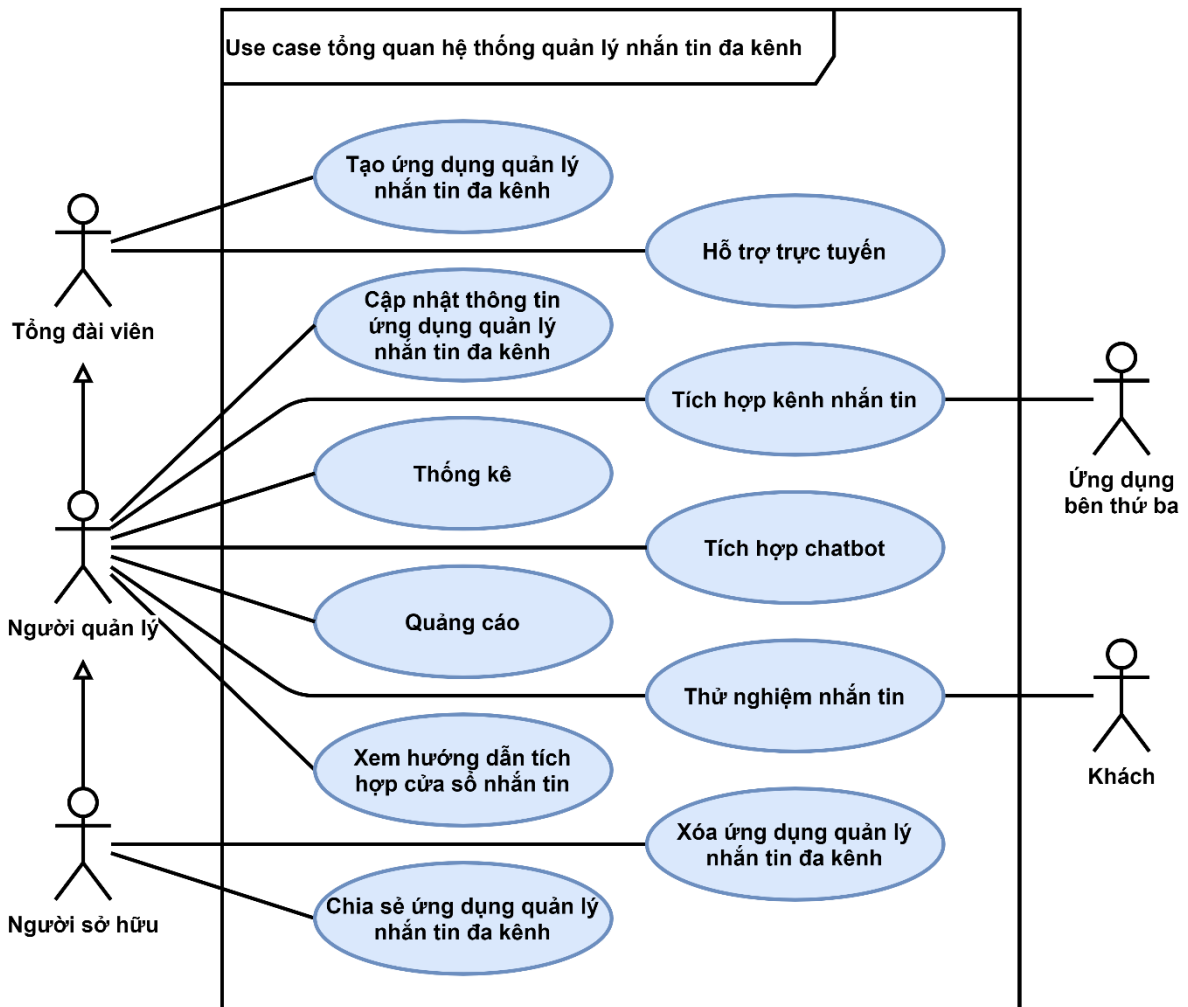
vChat là phần mềm nhắn tin trực tuyến trên website của Việt Nam. Phần mềm tích hợp khá đầy đủ các tính năng của phần mềm nhắn tin trực tuyến có trên thị trường hiện nay như cài đặt trả lời nhanh, chat bot thông minh, phân quyền cho nhân viên, gửi tin nhắn offline đến khách hàng, thống kê báo cáo (tổng số cuộc nhắn tin, số lượng tin nhắn, thời gian hỗ trợ, v.v.) để nâng cao chất lượng chăm sóc khách hàng tốt hơn nữa, theo dõi lịch sử truy cập của khách hàng (Đến từ đâu? Xem những sản phẩm nào? Địa điểm ở khu vực nào? v.v.) từ đó lựa chọn khách hàng tiềm năng nhất để chủ động tư vấn, tùy chỉnh cửa sổ nhắn tin theo phong cách riêng, v.v. Tuy nhiên, cũng giống như tawk.to, vChat không cung cấp tính năng tích hợp với các kênh nhắn tin khác. vChat cũng chưa cung cấp tính năng gửi tin nhắn hàng loạt đến khách hàng.

Từ những so sánh trên, em sẽ phát triển hệ thống quản lý nhắn tin đa kênh cung cấp các chức năng cơ bản như thống kê, tích hợp chatbot, nhắn tin hỗ trợ khách hàng. Đồng thời phát triển thêm tính năng kết nối với nhiều kênh nhắn tin bên thứ ba và gửi tin nhắn hàng loạt đến khách hàng.

2.2 Tổng quan chức năng

2.2.1 Hệ thống quản lý nhắn tin đa kênh

Hệ thống quản lý nhắn tin đa kênh gồm nhiều thành phần và chức năng khác nhau. **Hình 1** mô tả tổng quan chức năng của toàn bộ hệ thống.



Hình 1 Biểu đồ use case tổng quan hệ thống quản lý nhắn tin đa kênh

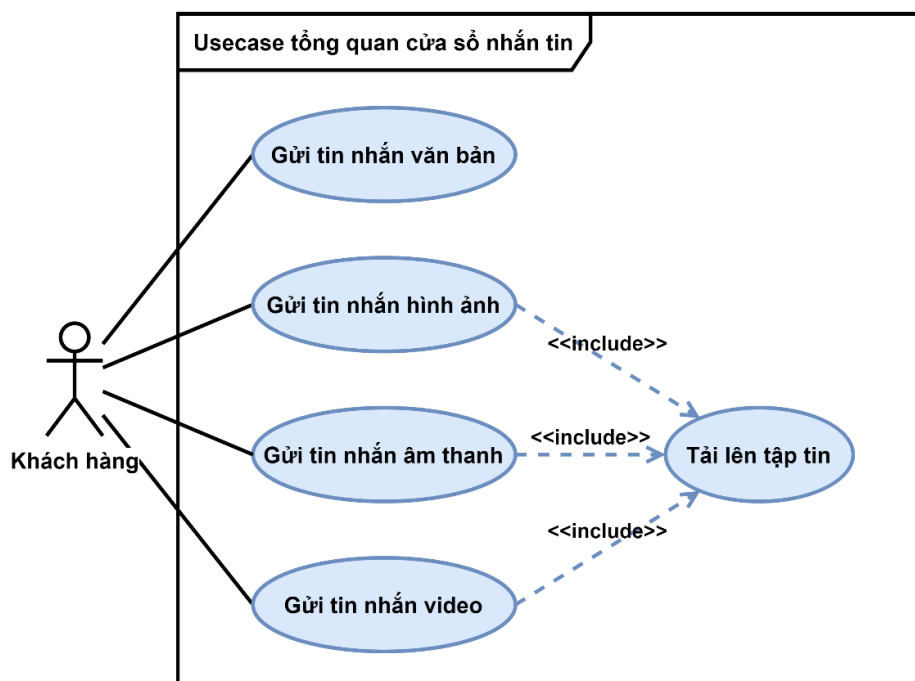
Hệ thống gồm 5 tác nhân, đó là: Người sở hữu, Người quản lý, Tổng đài viên, Khách và Ứng dụng bên thứ ba, mỗi tác nhân có những vai trò nhất định.

- Tổng đài viên sau khi đăng nhập vào hệ thống sẽ được cấp quyền nhắn tin hỗ trợ khách hàng và có thể tạo ứng dụng quản lý nhắn tin đa kênh của riêng mình.

- Người quản lý sẽ kế thừa mọi chức năng của tổng đài viên và có thể thực hiện các chức năng liên quan đến ứng dụng quản lý nhắn tin đa kênh như: cập nhật thông tin ứng dụng quản lý nhắn tin đa kênh, thống kê, tích hợp kênh nhắn tin, tích hợp chatbot, quảng cáo, xem hướng dẫn tích hợp của sổ livechat và thử nghiệm nhắn tin.
- Người sở hữu là người tạo ra ứng dụng quản lý nhắn tin đa kênh, giữ vai trò cao nhất điều khiển mọi hoạt động của hệ thống, có thêm quyền chia sẻ và xóa ứng dụng so với người quản lý.
- Ứng dụng bên thứ ba gồm Facebook Messenger, Zalo và Viber được tích hợp vào hệ thống để thêm kênh nhắn tin.
- Khách là người dùng không đăng nhập vào hệ thống, có được đường dẫn đến trang nhắn tin thử và thực hiện nhắn tin đến hệ thống.

2.2.2 Cửa sổ livechat trên nền tảng web

Cửa sổ livechat cho phép khách hàng nhắn tin với nhiều thể loại tin nhắn. **Hình 2** mô tả các loại tin nhắn được hệ thống chấp nhận. Cửa sổ livechat chỉ có một tác nhân tham gia, đó là Khách hàng. Khách hàng là người truy cập vào website đã tích hợp cửa sổ livechat và thực hiện nhắn tin trên đó. Hệ thống chấp nhận 4 loại tin nhắn chính, đó là: tin nhắn văn bản, tin nhắn hình ảnh, tin nhắn âm thanh và tin nhắn video. Trong đó để thực hiện việc gửi tin nhắn hình ảnh, âm thanh hoặc video thì trước hết cần tải tệp tin lên server để lấy đường dẫn tệp tin.

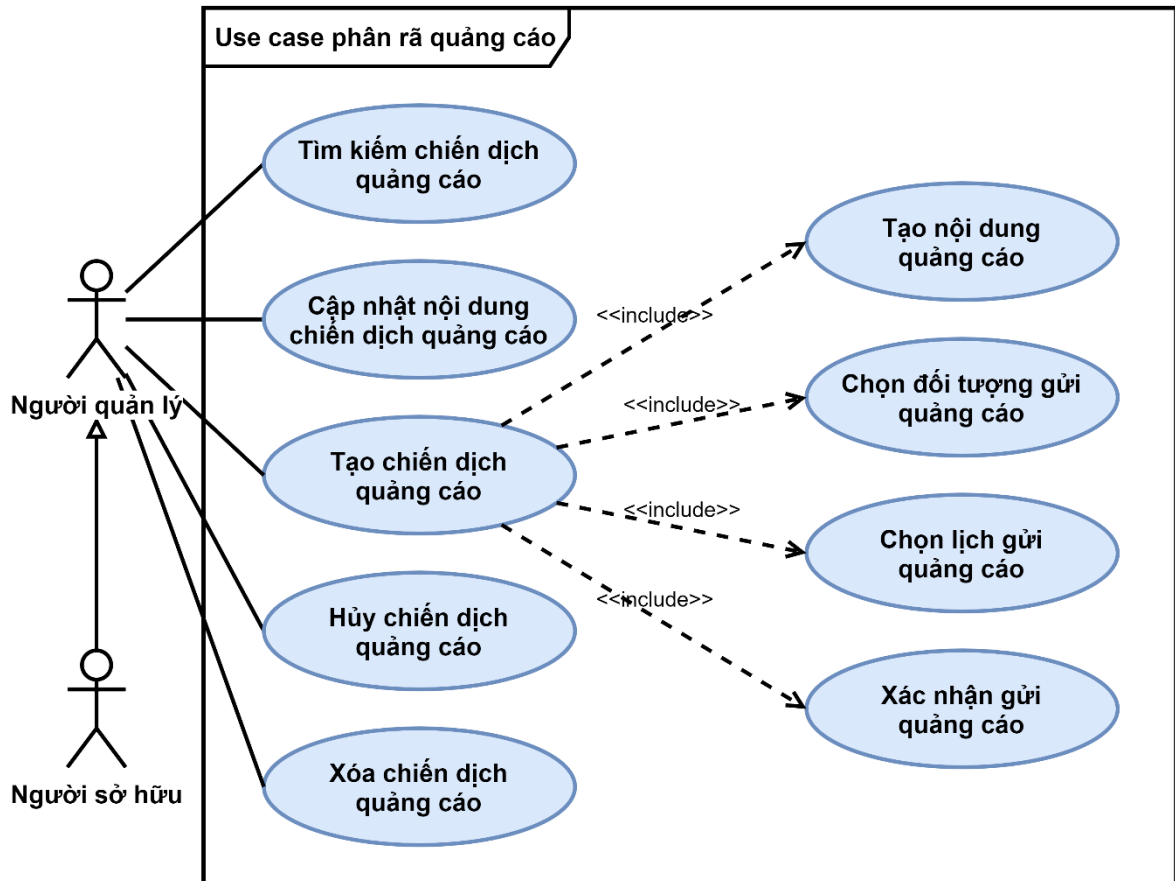


Hình 2 Biểu đồ use case tổng quan cửa sổ livechat

2.2.3 Biểu đồ use case phân rã

2.2.3.1 Biểu đồ use case phân rã quảng cáo

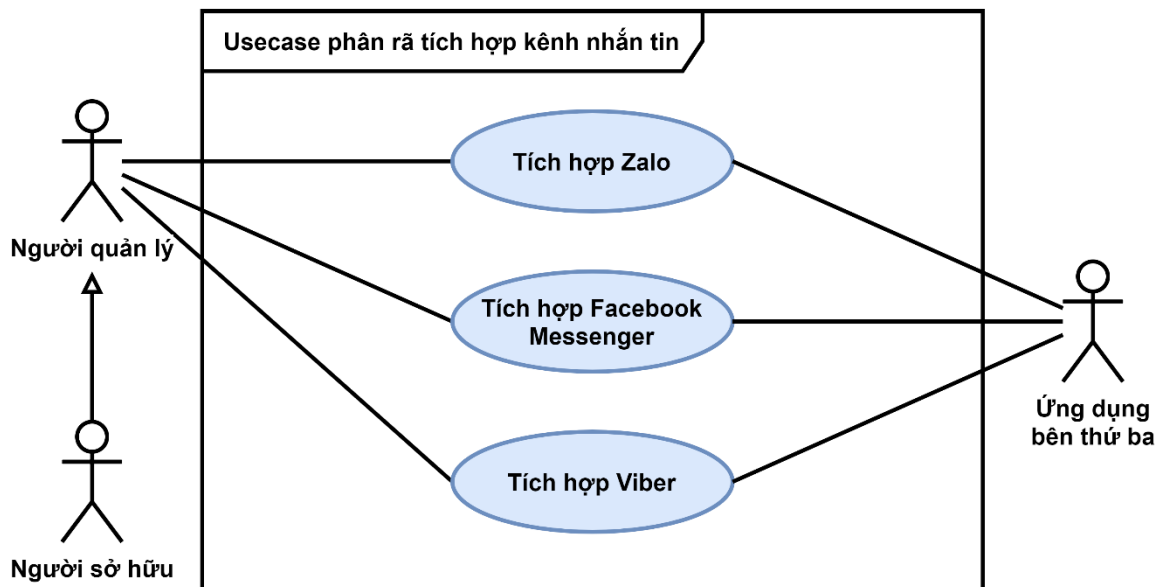
Với chức năng quảng cáo người dùng có thể Tạo chiến dịch quảng cáo, Tìm kiếm chiến dịch quảng cáo, Cập nhật nội dung chiến dịch quảng cáo, Hủy chiến dịch quảng cáo và Xóa chiến dịch quảng cáo như **Hình 3**. Trong đó, với chức năng tạo chiến dịch quảng cáo, người dùng cần thực hiện bốn bước: (i) tạo nội dung quảng cáo, (ii) chọn đối tượng gửi quảng cáo, (iii) chọn lịch gửi quảng cáo, (iv) xác nhận gửi quảng cáo.



Hình 3 Biểu đồ use case phân rã quảng cáo

2.2.3.2 Biểu đồ use case phân rã tích hợp kênh nhắn tin

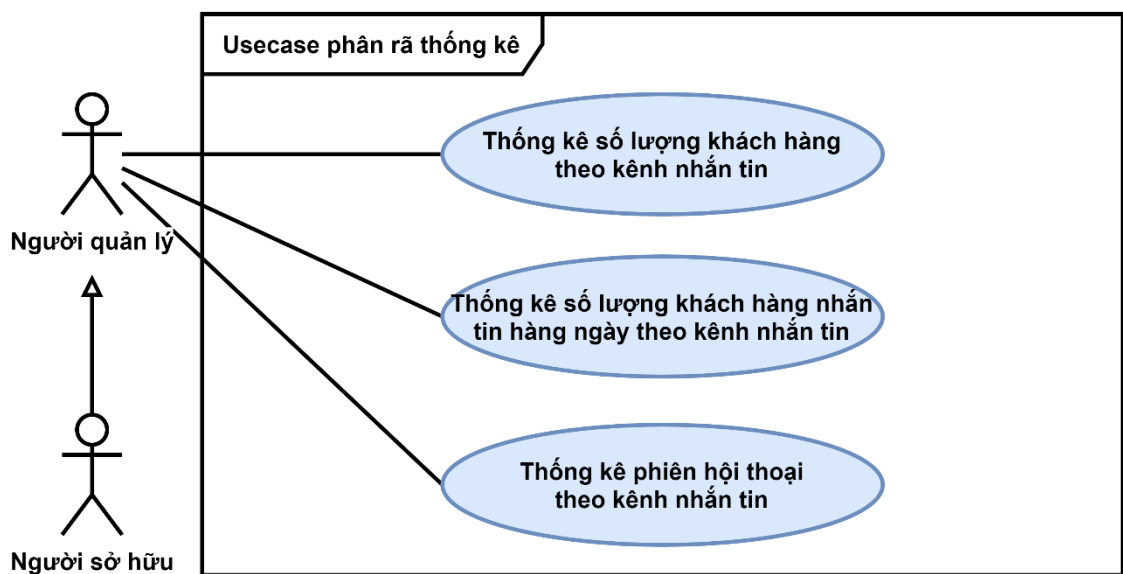
Hệ thống quản lý nhắn tin đa kênh cho phép tích hợp nhiều kênh nhắn tin. Ngoài kênh nhắn tin cửa sổ livechat thì người dùng có thể tích hợp thêm kênh nhắn tin từ Zalo, Facebook Messenger, Viber như trong **Hình 4** để quản lý toàn bộ thông tin khách hàng cũng như tin nhắn tại một nơi duy nhất.



Hình 4 Biểu đồ use case phân rã tích hợp kênh nhắn tin

2.2.3.3 Biểu đồ use case phân rã thống kê

Hình 5 là biểu đồ phân rã cho use case thống kê. Qua chức năng này, người dùng có thể nắm bắt nhanh được các thông tin về lượng khách hàng nhắn tin đến hệ thống theo các khoảng thời gian.



Hình 5 Biểu đồ use case phân rã thống kê

2.3 Đặc tả chức năng

Phần này sẽ đặc tả một số chức năng chính của hệ thống. **Bảng 2** liệt kê 23 use case được sử dụng. Do kích thước báo cáo có hạn nên em sẽ trình bày 4 use case: (i) Hỗ trợ trực tuyến,

(ii) Tích hợp Zalo, (iii) Lập lịch gửi quảng cáo, (iv) Gửi tin nhắn video. Các use case còn lại được trình bày trong phần Phụ lục.

Bảng 2 Danh sách use case

Hệ thống	Nhóm use case	Mã use case	Tên use case
Hệ thống quản lý nhắn tin đa kênh	Tổng quan	UC001	Tạo ứng dụng quản lý nhắn tin đa kênh
		UC002	Cập nhật thông tin ứng dụng quản lý nhắn tin đa kênh
		UC003	Chia sẻ ứng dụng quản lý nhắn tin đa kênh
		UC004	Xóa ứng dụng quản lý nhắn tin đa kênh
		UC005	Xem hướng dẫn tích hợp cửa sổ livechat
		UC006	Thử nghiệm nhắn tin
		UC007	Hỗ trợ trực tuyến
		UC008	Tích hợp chatbot
	Tích hợp kênh nhắn tin	UC009	Tích hợp Zalo
		UC010	Tích hợp Facebook Messenger
		UC011	Tích hợp Viber
	Thống kê	UC012	Thống kê số lượng khách hàng theo kênh nhắn tin
		UC013	Thống kê số lượng khách hàng nhắn tin hàng ngày theo kênh nhắn tin
		UC014	Thống kê phiên hội thoại theo kênh nhắn tin
	Quảng cáo	UC015	Tìm kiếm chiến dịch quảng cáo
		UC016	Cập nhật nội dung chiến dịch quảng cáo

Hệ thống	Nhóm use case	Mã use case	Tên use case
		UC017	Tạo chiến dịch quảng cáo
		UC018	Hủy chiến dịch quảng cáo quảng cáo
		UC019	Xóa chiến dịch quảng cáo quảng cáo
Cửa sổ livechat		UC020	Gửi tin nhắn văn bản
		UC021	Gửi tin nhắn hình ảnh
		UC022	Gửi tin nhắn âm thanh
		UC023	Gửi tin nhắn video

2.3.1 Đặc tả use case Hỗ trợ trực tuyến

Bảng 3 Đặc tả use case Hỗ trợ trực tuyến

Mã use case	UC007	Tên use case	Use case Hỗ trợ trực tuyến
Tác nhân	Tổng đài viên, Người quản lý, Người sở hữu		
Tiền điều kiện	<p>Người dùng đăng nhập vào hệ thống quản lý nhắn tin đa kênh với vai trò là Tổng đài viên, Người quản lý hoặc Người sở hữu.</p> <p>Đã có Khách hàng nhắn tin đến hệ thống.</p>		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Chọn Khách hàng cần hỗ trợ
	2.	Hệ thống	Hiển thị khung nhắn tin với khách hàng
	3.	Người dùng	Nhập tin nhắn văn bản
	4.	Hệ thống	Lưu tin nhắn vào cơ sở dữ liệu và hiển thị trên khung nhắn tin
	5.	Hệ thống	Gửi tin nhắn đến khách hàng

Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a.	Hệ thống	Thông báo lỗi nếu tin nhắn không được lưu
	3a1.	Người dùng	Chọn tin nhắn hình ảnh / âm thanh / video từ thiết bị sử dụng
	3a2.	Hệ thống	Tải tệp tin người dùng chọn lên server
	3a3.	Hệ thống	Lưu đường dẫn tệp tin vào cơ sở dữ liệu và hiển thị tin nhắn
	3a3a.	Hệ thống	Thông báo tải tệp tin lỗi
	3b1.	Người dùng	Chọn biểu tượng từ khung biểu tượng
	3b2.	Hệ thống	Lưu tin nhắn vào cơ sở dữ liệu và hiển thị trên khung nhắn tin
	3b2a.	Hệ thống	Thông báo lỗi nếu tin nhắn không được lưu
Hậu điều kiện	Không		

2.3.2 Đặc tả use case Tích hợp Zalo

Bảng 4 Đặc tả use case Tích hợp Zalo

Mã use case	UC009	Tên use case	Use case Tích hợp Zalo
Tác nhân	Người sử hữu, Người quản lý, Ứng dụng bên thứ ba		
Tiền điều kiện	<p>Người dùng đăng nhập vào hệ thống quản lý nhắn tin đa kênh với vai trò là Người quản lý hoặc Người sở hữu.</p> <p>Đã tạo ứng dụng Zalo trên trang https://developers.Zalo.me.</p> <p>Đã tạo Official Account trên trang https://oa.Zalo.me.</p> <p>Đã liên kết ứng dụng Zalo với Official Account.</p> <p>Cấu hình Official Account Callback Url trên ứng dụng Zalo (Callback Url lấy từ giao diện tích hợp Zalo).</p>		

Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu tích hợp Zalo và nhập ID ứng dụng đã tạo trên Zalo
	2.	Hệ thống	Kiểm tra các trường bắt buộc nhập
	3.	Hệ thống	Điều hướng đến trang xác nhận thông tin của Zalo
	4.	Người dùng	Chọn Official Account muốn tích hợp và tích chọn đồng ý tích hợp
	5.	Hệ thống	Lấy dữ liệu từ đường dẫn Callback Url, lưu cơ sở dữ liệu và thông báo tích hợp thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a.	Hệ thống	Thông báo lỗi chưa nhập thông tin
	3a.	Hệ thống	Thông báo lỗi đường dẫn chuyển hướng không chính xác
	5a.	Hệ thống	Thông báo lỗi không thể lấy được dữ liệu
Hậu điều kiện	Không		

2.3.3 Đặc tả use case Tạo chiến dịch quảng cáo

Bảng 5 Đặc tả use case Tạo chiến dịch quảng cáo

Mã use case	UC017	Tên use case	Use case Tạo chiến dịch quảng cáo
Tác nhân	Người sở hữu, Người quản lý		
Tiền điều kiện	Người dùng đăng nhập vào hệ thống quản lý nhắn tin đa kênh với vai trò là Người quản lý hoặc Người sở hữu.		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu tạo chiến dịch quảng cáo

	2.	Hệ thống	Hiển thị giao diện tạo nội dung chiến dịch quảng cáo
	3.	Người dùng	Nhập thông tin nội dung chiến dịch quảng cáo (mô tả dưới *)
	4.	Người dùng	Yêu cầu tiếp tục
	5.	Hệ thống	Lưu cơ sở dữ liệu nội dung chiến dịch quảng cáo và Hiển thị giao diện tạo chọn đối tượng gửi quảng cáo
	6.	Người dùng	Chọn đối tượng gửi quảng cáo
	7.	Người dùng	Yêu cầu tiếp tục
	8.	Hệ thống	Hiển thị giao diện chọn thời gian gửi quảng cáo
	9.	Người dùng	Chọn thời gian gửi quảng cáo
	10.	Người dùng	Yêu cầu tiếp tục
	11.	Hệ thống	Hiển thị giao diện xác nhận gửi quảng cáo (mô tả dưới **)
	12.	Người dùng	Yêu cầu lưu chiến dịch quảng cáo
	13.	Hệ thống	Lưu cơ sở dữ liệu và thông báo tạo chiến dịch quảng cáo thành công
	14.	Hệ thống	Gửi quảng cáo đến người dùng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a1.	Người dùng	Yêu cầu hủy tạo chiến dịch quảng cáo
	4a2.	Hệ thống	Kết thúc use case Tạo chiến dịch quảng cáo
	5a	Hệ thống	Thông báo không tạo được nội dung chiến dịch quảng cáo

	7a1.	Người dùng	Yêu cầu quay lại
	7a2.	Hệ thống	Quay lại bước 3
	7b1.	Người dùng	Yêu cầu hủy tạo chiến dịch quảng cáo
	7b2.	Hệ thống	Kết thúc use case Tạo chiến dịch quảng cáo
	10a1.	Người dùng	Yêu cầu quay lại
	10a2.	Hệ thống	Quay lại bước 6
	10b1.	Người dùng	Yêu cầu hủy tạo chiến dịch quảng cáo
	10b2.	Hệ thống	Kết thúc use case Tạo chiến dịch quảng cáo
	12a1.	Người dùng	Yêu cầu quay lại
	12a2.	Hệ thống	Quay lại bước 9
	12b1.	Người dùng	Yêu cầu hủy tạo chiến dịch quảng cáo
	12b2.	Hệ thống	Kết thúc use case Tạo chiến dịch quảng cáo
	13a.	Hệ thống	Thông báo không tạo được nội dung chiến dịch quảng cáo.
Hậu điều kiện	Không		

* Dữ liệu đầu vào khi tạo nội dung chiến dịch quảng cáo

STT	Trường dữ liệu	Mô tả	Bắt buộc
1.	Tiêu đề		Có
2.	Tin nhắn văn bản		Không
3.	Tin nhắn hình ảnh		Không
4.	Mô tả tin nhắn hình ảnh		Không

STT	Trường dữ liệu	Mô tả	Bắt buộc
5.	Tin nhắn âm thanh		Không
6.	Mô tả tin nhắn âm thanh		Không
7.	Tin nhắn video		Không
8.	Mô tả tin nhắn video		Không
9.	Tin nhắn catalog	Danh sách các tin nhắn hình ảnh	Không

** Dữ liệu đầu ra khi hiển thị giao diện xác nhận gửi chiến dịch quảng cáo

STT	Trường dữ liệu	Mô tả
1.	Tiêu đề	
2.	Nội dung	Danh sách các tin nhắn sẽ được gửi đến khách hàng
3.	Đối tượng gửi	Danh sách các kênh nhắn tin sẽ gửi tin nhắn quảng cáo
4.	Lịch gửi	Thời gian sẽ gửi quảng cáo đến khách hàng

2.3.4 Đặc tả use case Gửi tin nhắn video

Bảng 6 Đặc tả use case Gửi tin nhắn video

Mã use case	UC023	Tên use case	Use case Gửi tin nhắn video
Tác nhân	Khách hàng		
Tiền điều kiện	Không		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Khách hàng	Chọn video muốn gửi
	2.	Hệ thống	Tải video lên server upload và trả về đường dẫn video

	3	Hệ thống	Gửi đường dẫn video đến server nhắn tin và hiển thị video trên cửa sổ livechat của người dùng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a.	Hệ thống	Thông báo lỗi không đúng định dạng
	2b.	Hệ thống	Thông báo lỗi tệp tin quá lớn
Hậu điều kiện	Không		

2.4 Yêu cầu phi chức năng

2.4.1 Tính dễ dùng

Do người sử dụng hệ thống quản lý nhắn tin đa kênh là những người đã có hiểu biết ít nhiều về máy tính và công nghệ nên hệ thống cần được xây dựng sao cho giao diện tinh giản, hạn chế các công đoạn thao tác không cần thiết. Cần có hướng dẫn đầy đủ các bước khi tiến hành kết nối với kênh nhắn tin bên thứ ba. Các giao diện cần sử dụng icon hành động phổ biến mà không phải những icon lạ bắt người dùng phải học hoặc biết.

2.4.2 Tính dễ bảo trì

Hệ thống đang trong quá trình xây dựng và phát triển nên cần được thiết kế để dễ dàng sửa đổi và nâng cấp cho phù hợp với môi trường hoạt động và các yêu cầu mới. Các tính năng mới hay phiên bản mới sẽ được bổ sung và nâng cấp mà không ảnh hưởng đến sự hoạt động ổn định của hệ thống.

2.4.3 Tính khả chuyển

Hệ thống được xây dựng để có thể hoạt động tốt trên nhiều môi trường khác nhau như Laptop, PC, điện thoại Android, điện thoại IOS, máy tính bảng.

Chương 2 đã mô tả yêu cầu, giới thiệu tổng quan về chức năng, tiến hành phân tích một số use case và đặc tả một số chức năng chính của hệ thống. Ngoài ra, yêu cầu phi chức năng là một phần quan trọng đối với hệ thống nhằm nâng cao trải nghiệm người dùng. Để đạt được hiệu quả và các yêu cầu như thế, Chương 3 sẽ trình bày công nghệ sử dụng để xây dựng ứng dụng.

Chương 3 Công nghệ sử dụng

Từ quá trình phân tích yêu cầu trong Chương 2, Chương 3 sẽ giới thiệu về công nghệ sử dụng trong hệ thống quản lý nhắn tin đa kênh và cửa sổ livechat trên nền tảng web. Hệ thống quản lý được xây dựng thành hai phần riêng biệt frontend và backend. Phần frontend sử dụng ReactJS kết hợp với thư viện giao diện Material UI. Phần backend được thiết kế theo kiến trúc Microservices sử dụng NodeJS cùng với MongoDB để lưu trữ dữ liệu. Cửa sổ livechat được thiết kế theo dạng component sử dụng ReactJS, Styled-Components để viết CSS và Websocket để kết nối đến server socket.

3.1 Frontend

3.1.1 ReactJS

ReactJS [4] là một thư viện JavaScript để xây dựng giao diện người dùng, được xây dựng và phát triển bởi Facebook. ReactJS tạo ra những ứng dụng web hấp dẫn với tốc độ nhanh và hiệu quả cao với những mã hóa tối thiểu. Nó đang đóng một vai trò thiết yếu trong hệ sinh thái frontend. Các tính năng quan trọng của ReactJS như sau:

JSX (Javascript XML) - là một phần mở rộng cú pháp JavaScript. Đây là một cú pháp cho phép viết các mã HTML trong JavaScript để xác định thành phần giao diện người dùng và chức năng của chúng. JSX thực hiện việc tối ưu hóa trong khi biên dịch sang mã Javascript, các lỗi cũng sẽ được phát hiện ngay trong quá trình biên dịch, qua đó giúp việc gỡ lỗi trở nên dễ dàng hơn. Với JSX, các nhà phát triển có thể xây dựng các khối React UI và React Component bằng cú pháp của HTML. Do đó, JSX là tính năng tốt nhất được cung cấp bởi ReactJS vì tất cả mọi thứ viết ra đều là những thứ hiển thị trên màn hình.

Component-Based – React được xây dựng dựa trên Component. Mỗi component được thiết kế với một logic cụ thể. Chính vì thế, việc phát triển và tìm lỗi đơn giản hơn vì chỉ cần tập trung vào một phần nhỏ thay vì toàn bộ ứng dụng. Component gồm state và props. State duy trì trạng thái của component, khi state thay đổi thì toàn bộ component sẽ được cập nhật lại. Props thể hiện luồng dữ liệu một chiều trong react, dữ liệu được truyền từ component cha xuống component con thông qua props, trong một component thì props là bất biến. Hơn nữa, khi sử dụng component sẽ giúp tăng khả năng tái sử dụng khi viết code.

Luồng dữ liệu một chiều (One-way Data Binding) - ReactJS được thiết kế tuân theo luồng dữ liệu một chiều từ cha xuống con. Lợi ích của luồng dữ liệu một chiều cho phép kiểm soát tốt hơn trong toàn ứng dụng. Nếu luồng dữ liệu theo hướng khác thì nó yêu cầu các chức năng bổ sung. Đó là bởi vì các Component được cho là bất biến và dữ liệu trong chúng không thể thay đổi từ bên ngoài.

Virtual DOM - là tính năng chính cho phép ReactJS xây dựng các ứng dụng nhanh và có thể mở rộng. Nó chứa một thuật toán đối chiếu bộ nhớ giúp React tạo ra một đại diện của một trang web trong bộ nhớ ảo. Vì vậy, Virtual DOM là đại diện của DOM gốc. Bất cứ khi nào có thay đổi xảy ra trong ứng dụng web, toàn bộ giao diện người dùng sẽ được kết xuất lại trong Virtual DOM. Sau đó, nó kiểm tra sự khác biệt giữa DOM trước đó và DOM mới. Khi đã xong, DOM thật sẽ chỉ cập nhật những thứ đã thực sự thay đổi. Điều này làm cho ứng dụng nhanh hơn và không lãng phí bộ nhớ.

Vì ReactJS là một thư viện javascript nhẹ và chỉ hỗ trợ các thành phần UI nên có phải được bổ xung với các công cụ, thư viện, framework hoặc công nghệ khác để xây dựng thành một kiến trúc hoàn chỉnh.

3.1.2 Material UI

Material UI [7] là một thư viện gồm tập hợp các react component được Google viết cho ReactJS theo phong cách của Material Design. Material UI rất phù hợp để xây dựng các ứng dụng React một cách nhanh chóng vì nó là tập hợp các components của React và đáp ứng được hầu hết các components thông dụng như Layout, Button, Textfield, Icon, DatePicker, v.v.

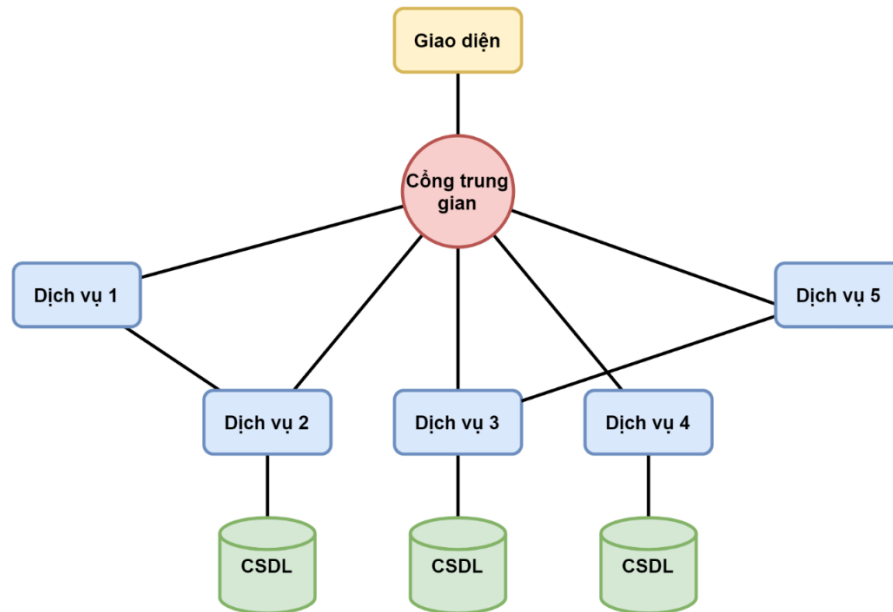
3.1.3 Styled-Components

Styled-Components [8] là một thư viện giúp tổ chức và quản lý code CSS một cách dễ dàng trong các ứng dụng React. Nó cho phép viết CSS bằng Javascript và được xây dựng với mục tiêu giữ cho các styles của các components trong React gắn liền với chính các components đó. Với Styled-Components, khi component được render thì style sẽ được tự động chèn vào, như vậy, người dùng sẽ phải tải ít mã nhất có thể khi load trang. Styled-Components cũng giúp tránh được một số lỗi so với sử dụng CSS thông thường, ví dụ như sẽ không bao giờ gặp lỗi liên quan đến tên class vì nó tạo ra các tên class là duy nhất. Việc gỡ lỗi cũng đơn giản hơn khi không cần tìm kiếm các file khác nhau để tìm ra các style ảnh hưởng đến component.

3.2 Backend

3.2.1 Kiến trúc Microservices

Ý tưởng của mô hình kiến trúc Microservices là chia nhỏ ứng dụng thành các dịch vụ nhỏ kết nối với nhau như **Hình 6**.



Hình 6 Mô hình Microservices

Mỗi dịch vụ nhỏ thực hiện một tập chức năng chuyên biệt. Các dịch vụ nhỏ này xây dựng các API để giao tiếp với nhau hoặc cho giao diện gọi tới. Các dịch vụ nhỏ được xây dựng sao cho có khả năng hoạt động độc lập mà không bị phụ thuộc hay phụ thuộc quá nhiều vào các dịch vụ khác trong hệ thống [2]. Các giao diện sẽ không kết nối trực tiếp đến từng dịch vụ mà thông qua một cổng kết nối trung gian. Cổng trung gian này có nhiệm vụ điều hướng, phân tải, cache, xác thực, kiểm tra quyền truy cập và theo dõi.

Ưu điểm của Microservices là giảm thiểu sự phức tạp trong các hệ thống lớn. Chia nhỏ ứng dụng một khối cồng kềnh thành các dịch vụ nhỏ dễ quản lý, bảo trì nâng cấp, thay đổi công nghệ mới. Dễ dàng mở rộng thêm các chức năng mới bằng cách thêm dịch vụ và kết nối tới cổng trung gian. [3]

Bên cạnh ưu điểm thì Microservices cũng có những nhược điểm. Microservices nhấn mạnh kích thước nhỏ gọn của dịch vụ, chính vì vậy khi chia quá nhỏ sẽ dẫn đến vụn vặt và khó kiểm soát, mỗi dịch vụ hầu như đều kết nối đến một cơ sở dữ liệu riêng làm cho dữ liệu bị phân tán quá mức cần thiết gây ra sự dư thừa và đôi khi không đảm bảo được tính toàn vẹn. Bên cạnh đó, nếu các dịch vụ bị phụ thuộc vào nhau quá nhiều thì sẽ gây ra sự chồng chéo

khó kiểm soát. Vì vậy, cần có cách phân chia dịch vụ hợp lý và cách kết nối phù hợp để tránh sự chồng chéo trên.

3.2.2 NodeJS

NodeJS [5] là một nền tảng được xây dựng dựa trên Javascript engine V8 của Google – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng. NodeJS được các công ty lớn sử dụng như: Amazon, Ebay, Microsoft, Trello, v.v. Ưu điểm của NodeJS là tốc độ thực thi và khả năng mở rộng. NodeJS tạo ra các ứng dụng có tốc độ xử lý nhanh với thời gian thực, có lượng lớn người dùng và cần mở rộng nhanh. NodeJS chạy đa nền tảng phía Server sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả. Cộng đồng sử dụng NodeJS rất lớn và hoàn toàn miễn phí.

Express là một framework phổ biến nhất của NodeJS được sử dụng cho việc xây dựng các ứng dụng web và các API. Với Express, việc phát triển ứng dụng được rút ngắn đi rất nhiều. Các đặc điểm nổi bật chính của Express có thể nhận thấy bao gồm việc đây là một framework tương đối nhỏ gọn và linh hoạt, cung cấp nhiều tính năng mạnh mẽ trên nền tảng web cũng như di động. Nó cung cấp vô số các phương thức HTTP và các middleware, nhờ đó giúp việc xây dựng các API trở nên nhanh và dễ dàng hơn. Express tạo ra một tầng mỏng các tính năng ứng dụng web cơ bản mà không làm mờ đi các tính năng hữu ích của NodeJS.

3.2.3 MongoDB

MongoDB là cơ sở dữ liệu phi quan hệ NoSQL, các dữ liệu trong MongoDB được lưu trữ dưới dạng JSON. MongoDB không sử dụng các hàng và cột thông thường mà cơ sở dữ liệu quan hệ hay dùng để liên kết. Nó là một kiến trúc được xây dựng trên các bộ sưu tập (collections) và tài liệu (documents). Đơn vị dữ liệu cơ bản trong cơ sở dữ liệu này bao gồm một tập hợp các cặp key-value. Nó cho phép các document có các trường và cấu trúc khác nhau, linh hoạt trong việc lưu trữ dữ liệu. Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau nên khi thêm, cập nhật, xóa không cần mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu hay không. Tuy nhiên cũng chính vì vậy nên khi thao tác với MongoDB phải hết sức cẩn thận. Bên cạnh đó, do dữ liệu được lưu dưới dạng key-value, các collection chỉ khác nhau về value do đó key sẽ bị lặp lại dẫn đến dư thừa dữ liệu.

3.3 Giao thức kết nối

3.3.1 Websocket

Websocket API [6] là công nghệ cho phép mở một kết nối 2 chiều giữa client và server. Với API này Client có thể gửi tin nhắn đến Server bất kỳ lúc nào và nhận phản hồi theo hướng sự kiện mà không cần gửi yêu cầu đến Server [6]. Nói cách khác, Server có thể chủ động

gửi thông tin đến Client mà không cần có yêu cầu từ Client và Client có thể gửi tin nhắn đến Server bất kỳ lúc nào, khác với HTTP là Client chủ động gửi yêu cầu đến Server sau đó chờ đợi để nhận được dữ liệu phản hồi. Websocket Client sử dụng Websocket API để giao tiếp với Websocket Server bằng giao thức Websocket.

3.3.2 RabbitMQ

RabbitMQ là một message broker sử dụng giao thức AMQP (Advanced Message Queue Protocol - Giao thức giao nhận tin nhắn sử dụng hàng đợi). Đây là chương trình đóng vai trò trung gian lưu trữ cũng như điều phối các yêu cầu (message) giữa người nhận (consumer) và người gửi (producer). Đối với hệ thống sử dụng Microservices thì việc gọi chéo giữa các dịch vụ là khá nhiều khiến cho luồng xử lý phức tạp, mức độ trao đổi dữ liệu giữa các dịch vụ tăng lên làm cho việc lập trình trở nên khó khăn hơn. Với RabbitMQ thì một người gửi không cần phải biết người nhận mà chỉ cần gửi yêu cầu đến các hàng đợi trong message broker, người nhận chỉ việc đăng ký nhận yêu cầu từ các hàng đợi này.

3.3.3 Rest API

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu cần cho ứng dụng ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, v.v đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các tài nguyên. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

REST hoạt động chủ yếu dựa vào giao thức HTTP, hoạt động này sẽ sử dụng những phương thức HTTP riêng.

- GET (SELECT): Trả về một dữ liệu hoặc một danh sách dữ liệu.
- POST (CREATE): Tạo mới một dữ liệu.
- PUT (UPDATE): Cập nhật thông tin cho dữ liệu.
- DELETE (DELETE): Xóa một dữ liệu.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

3.4 DevOps

Để triển khai cả phần frontend và backend của hệ thống một cách nhanh chóng, em đã sử dụng Docker.

Docker là một nền tảng cho các lập trình viên và nhà quản trị hệ thống dùng để xây dựng, triển khai và chạy các ứng dụng với container. Nó cho phép tạo các môi trường độc lập và tách biệt để khởi chạy và phát triển ứng dụng, môi trường này được gọi là container.

Dockerfile, Images, Docker containers, Docker Hub, Docker Compose là một số những khái niệm và thuật ngữ quan trọng khi sử dụng docker.

Dockerfile là danh sách các lệnh mà Docker Engine (công cụ để đóng gói ứng dụng) sẽ chạy để khởi tạo Images.

Images chứa mã nguồn cũng như tất cả các công cụ, thư viện và các phụ thuộc mà ứng dụng cần để chạy.

Docker containers là phiên bản chạy trực tiếp, thể hiện của Images. Trong khi Images là tập tin chỉ đọc thì containers là nội dung đã thực thi và người dùng có thể tương tác với chúng.

Docker Hub là kho lưu trữ các Images bao gồm cả công khai và riêng tư. Người dùng Docker Hub có thể chia sẻ images của mình theo ý muốn và cũng có thể tải xuống các Images để sử dụng cho bất kỳ ứng dụng docker nào.

Docker Compose là công cụ để định nghĩa và chạy nhiều container cho ứng dụng Docker. Với một hệ thống microservices thì khi khởi chạy sẽ phải chạy rất nhiều dịch vụ con, với Docker Compose sẽ tạo ra tệp YAML chỉ định dịch vụ nào sẽ được bao gồm trong ứng dụng và có thể triển khai các container bằng một lệnh duy nhất.

Chương 3 em đã trình bày các công nghệ sử dụng để xây dựng hệ thống. Để hiểu rõ hơn về kiến trúc và cách xây dựng, tổ chức và lưu trữ thông tin, Chương 4 em sẽ trình bày chi tiết về điều này.

Chương 4 Phát triển và triển khai ứng dụng

Trong chương 4 em sẽ trình bày chi tiết việc thiết kế từ kiến trúc đến giao diện và cơ sở dữ liệu, sau đó là quá trình xây dựng, kiểm thử và triển khai hệ thống.

4.1 Thiết kế kiến trúc

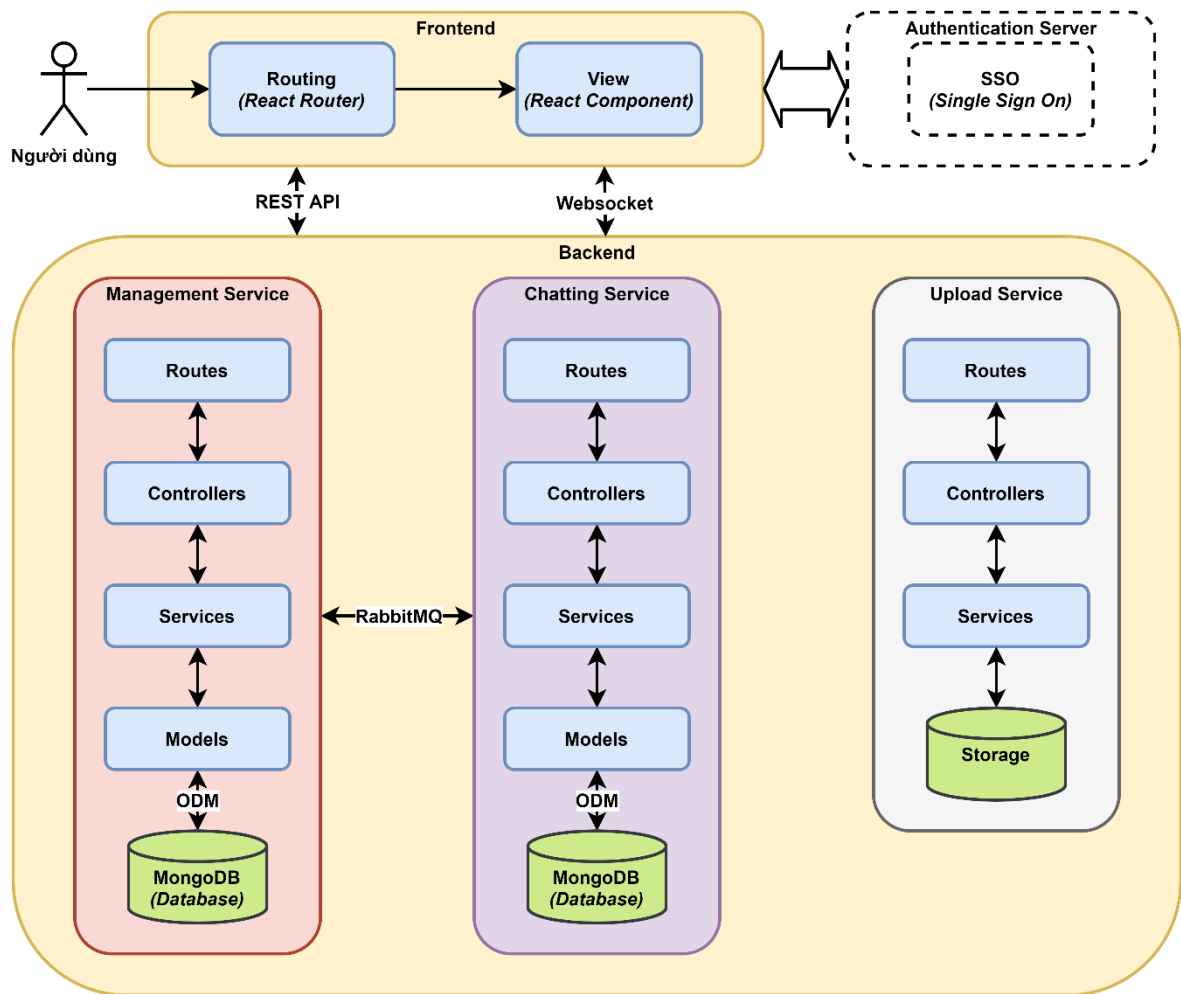
Hình 7 mô tả kiến trúc tổng quan của hệ thống quản lý nhắn tin đa kênh. Như đã trình bày ở Chương 3, hệ thống được tách biệt làm hai phần: Frontend và Backend.

Frontend sử dụng thư viện ReactJS để xây dựng các thành phần giao diện nhận và gửi dữ liệu đến máy chủ thông qua API. Thiết kế các thành phần giao diện sẽ được trình bày chi tiết trong 5.1.2.4.

Backend được thiết kế theo mô hình kiến trúc Microservices, trong đó bao gồm ba dịch vụ chính là: dịch vụ quản lý (Management Service), dịch vụ nhắn tin (Chatting Service) và dịch vụ tải lên tập tin (Upload Service). Mỗi dịch vụ có kiến trúc và cơ sở dữ liệu riêng và sẽ được trình bày chi tiết trong phần 5.1.2.1, 5.1.2.2 và 5.1.2.3. Các dịch vụ không truy vấn trực tiếp vào cơ sở dữ liệu MongoDB mà thực hiện thông qua ODM (Object Data Model) là biểu diễn dữ liệu của dưới dạng đối tượng trong JavaScript, sau đó đối chiếu đến nền cơ sở dữ liệu bên dưới. Nhờ vậy mà việc truy vấn trở nên đơn giản và dễ dàng hơn do có thể coi cơ sở dữ liệu như những đối tượng trong JavaScript thay vì sử dụng cú pháp thuần túy viết riêng cho cơ sở dữ liệu. Hơn nữa ODM còn cung cấp những tính năng để xác nhận và kiểm tra dữ liệu.

Phần Frontend sẽ kết nối tới các dịch vụ của phần Backend thông qua việc gọi API hoặc qua Websocket tùy mục đích. Giữa các dịch vụ giao tiếp với nhau qua RabbitMQ các hàng đợi tin nhắn. Nhờ cách thiết kế này mà các phần được tách riêng biệt và phát triển độc lập với nhau. Ở mỗi phần frontend và backend có thể tự do lựa chọn phối hợp nhiều loại công nghệ hay thậm chí là thay đổi cũng không làm ảnh hưởng đến nhau.

Ngoài ra, để đăng nhập thì hệ thống cần tương tác với dịch vụ xác thực (Authentication Server), đây là dịch vụ cho phép xác thực người dùng tập trung, đăng nhập một lần sử dụng nhiều nơi. Lý do sử dụng phương thức này là do hệ thống có dùng chung thông tin xác thực với SmartDialog Platform.



Hình 7 Kiến trúc tổng quan hệ thống quản lý nhắn tin đa kênh

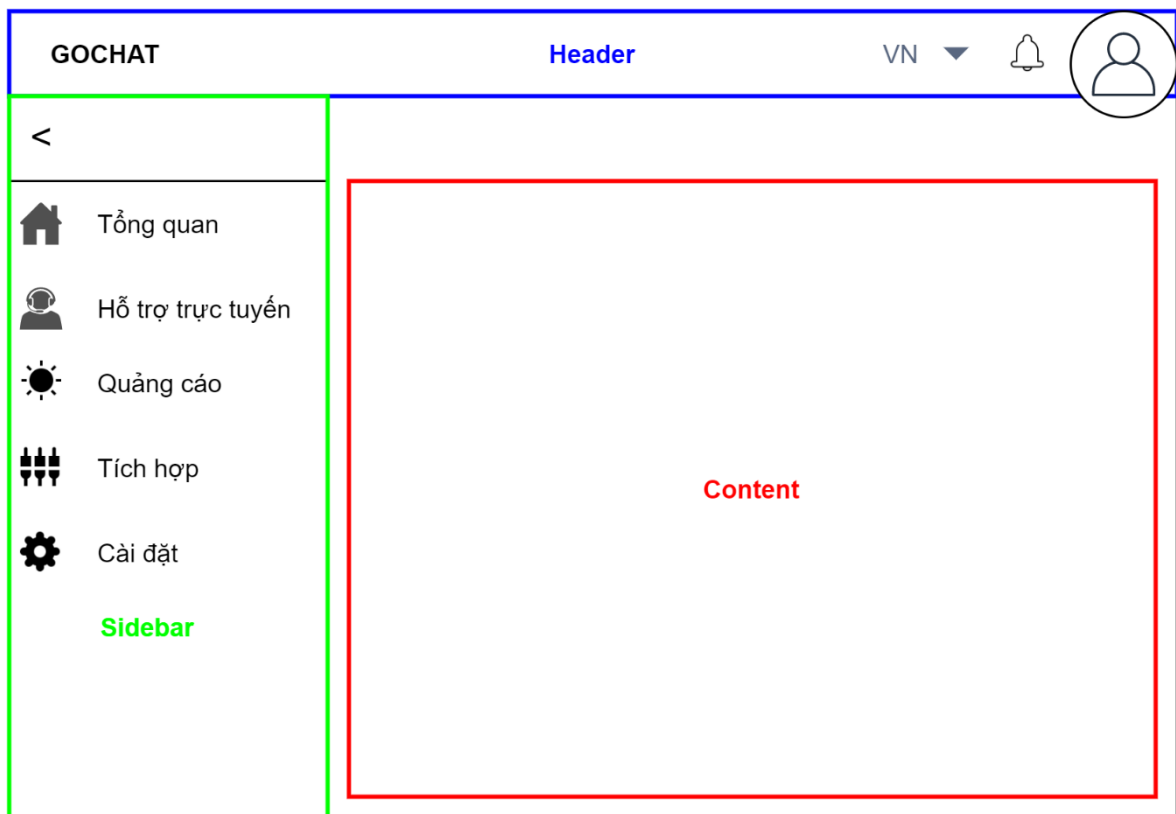
4.2 Thiết kế chi tiết frontend

4.2.1 Thiết kế mockup

4.2.1.1 Mockup layout của hệ thống quản lý nhắn tin đa kênh

Giao diện quản lý sử dụng thư viện React nên được thiết kế thành các component. Độ phân giải được dùng để thiết kế là 1920x1080, đây là độ phân giải phổ biến thường dùng cho thiết kế. Bố cục giao diện được thiết kế như **Hình 8** bao gồm Header, Sidebar và Content.

Tất cả các trang của hệ thống quản lý đều có Header và Sidebar chỉ thay đổi phần Content tùy thuộc vào đường dẫn người dùng truy cập để đảm bảo sự thống nhất về giao diện. Header là thanh tiêu đề của trang web, thông qua đây người dùng có thể thực hiện nhanh việc thay đổi ngôn ngữ hay đăng xuất. Sidebar là thanh bên trái của giao diện chứa nhiều thành phần nhỏ, mỗi thành phần sẽ dẫn đến một trang mới của hệ thống.



Hình 8 Mockup layout của hệ thống quản lý nhắn tin đa kênh

Màu sắc, font chữ, các nút, cách hiển thị thông báo đều được thiết kế đồng nhất nhằm đạt được tính nhất quán cho toàn bộ ứng dụng và gây hiệu ứng đẹp cho người dùng. **Bảng 7** trình bày rõ chi tiết của bộ cấu hình được sử dụng trong hệ thống quản lý.

Bảng 7 Bộ cấu hình chung cho giao diện quản lý

Thuộc tính	Cấu hình
Màu sắc	Màu chính: #000034 Màu khác: #f6a61f, #f16a73, #4991e2 #4a4a4a,
Font chữ	Helvetica, Aldrich
Bo góc	10px
Đổ bóng	0px 3px 3px -2px rgba(0,0,0,0.2), 0px 3px 4px 0px rgba(0,0,0,0.14), 0px 1px 8px 0px rgba(0,0,0,0.12)
Vị trí hiển thị popup thông báo	Chính giữa màn hình

Vị trí hiển thị thông báo khác	Góc trên bên phải màn hình
--------------------------------	----------------------------

4.2.1.2 Mockup cửa sổ livechat

Cửa sổ livechat cũng được thiết kế và xây dựng dựa trên thư viện ReactJS do đó giao diện sẽ được thiết kế thành các component như **Hình 9**.

Cửa sổ livechat được chia bố cục làm 3 phần. Trên cùng là phần tiêu đề hiển thị tên của doanh nghiệp sử dụng. Ở giữa là danh sách các tin nhắn. Dưới cùng là khu vực để nhập tin nhắn, ở đây người dùng có thể nhập tin nhắn văn bản hoặc chọn tệp hình ảnh, âm thanh, video từ thiết bị sử dụng.

Kích thước của cửa sổ livechat là 350x550, đây là kích thước phù hợp, vừa phải với màn hình được lựa chọn trong phần 4.2.1.1. Bảng màu được sử dụng trong cửa sổ livechat thể hiện qua **Bảng 8**. Các màu được phối theo kiểu Gradient cho phép tạo ra các phần tử có màu nền là các hiệu ứng chuyển màu giữa hai hay nhiều màu sắc được chỉ định.



Hình 9 Mockup cửa sổ livechat

Bảng 8 Bảng màu của sổ livechat

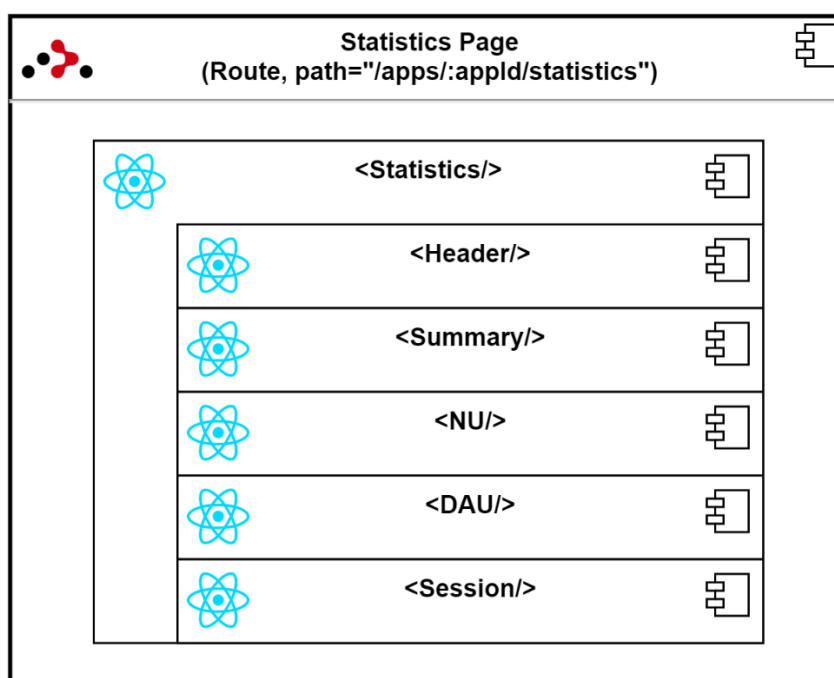
Tiêu đề	#418355	#82C13F
Danh sách tin nhắn	#6E8B49	#BDD1B0
Tin nhắn	#7FC781	#B3E59C
Nhập tin nhắn	#0F3F2D	#BABABA

4.2.2 Thiết kế thành phần giao diện

Phần này em sẽ trình bày thiết kế thành phần giao diện của bốn màn hình chính: thống kê, hỗ trợ trực tuyến, tích hợp và tạo nội dung chiến dịch quảng cáo.

4.2.2.1 Màn hình thống kê

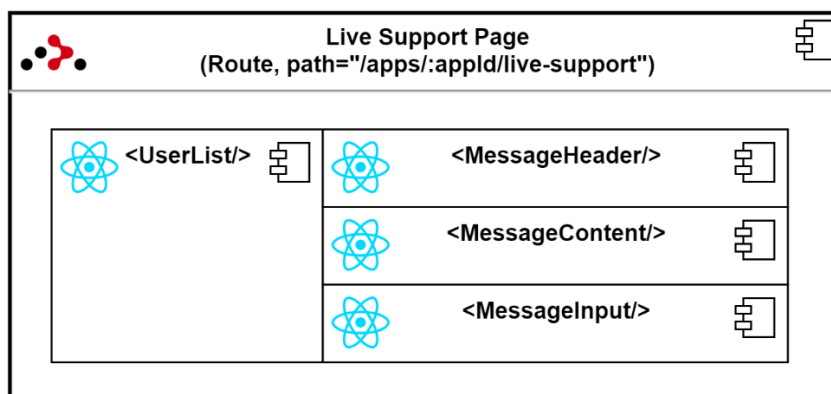
Màn hình thống kê được chia thành 5 thành phần con như **Hình 10**, đó là Header sẽ chứa các nút để lựa chọn khoảng thời gian thống kê, Summary là thành phần hiển thị các thông số tổng quan như số lượng khách hàng theo kênh nhắn tin, số lượng khách hàng nhắn tin hàng ngày theo kênh nhắn tin, số lượng phiên hội thoại theo kênh nhắn tin, NU (Number of User) hiển thị biểu đồ thống kê số lượng khách hàng theo kênh nhắn tin, DAU (Daily Active User) hiển thị biểu đồ thống kê số lượng khách hàng nhắn tin hàng ngày theo kênh nhắn tin, Session hiển thị biểu đồ thống kê số phiên hội thoại theo kênh nhắn tin.



Hình 10 Thiết kế thành phần giao diện màn hình thống kê

4.2.2.2 Màn hình hỗ trợ trực tuyến

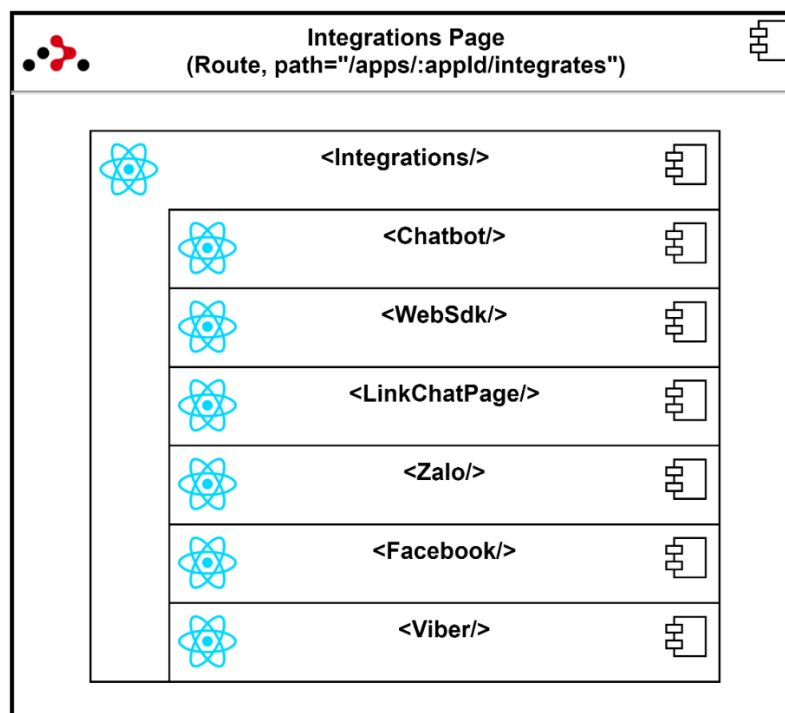
Màn hình hỗ trợ trực tuyến được chia làm bốn phần như **Hình 11**, bao gồm: (i) UserList, (ii) MessageHeader, (iii) MessageContent, (iv) MessageInput. Trong đó, UserList là danh sách khách hàng đã nhắn tin đến hệ thống trên tất cả các kênh nhắn tin và nằm ở bên trái của màn hình, các thành phần còn lại sẽ nằm bên phải màn hình. MessageHeader hiển thị tên của khách hàng, MessageContent là khung hiển thị danh sách các tin nhắn, MessageInput là khu vực để Tổng đài viên nhập tin nhắn trả lời khách hàng.



Hình 11 Thiết kế thành phần giao diện màn hình hỗ trợ trực tuyến

4.2.2.3 Màn hình tích hợp

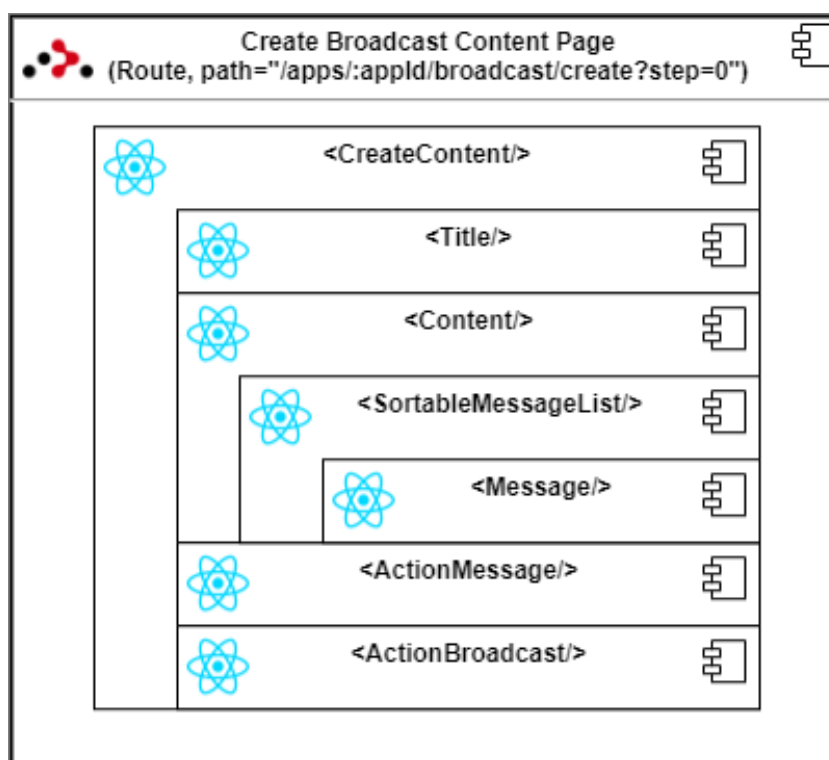
Màn hình tích hợp được chia thành 6 thành phần con như **Hình 12**, bao gồm:



Hình 12 Thiết kế thành phần giao diện màn hình tích hợp

- ChatBot là thành phần cho phép nhập token của chatbot để tích hợp hệ thống.
- WebSdk là thành phần hiển thị đoạn code để người dùng sử dụng khi tích hợp cửa sổ livechat vào website của họ.
- LinkChatPage sẽ hiển thị đường link đến trang thử nghiệm nhắn tin.
- Zalo là thành phần hiển thị hướng dẫn các bước tích hợp Zalo vào hệ thống và cho phép người dùng tích hợp Zalo.
- Facebook là thành phần hiển thị danh sách các trang Facebook người dùng đã tích hợp vào hệ thống và cho phép người dùng thực hiện hành động kết nối với các trang Facebook tại đây.
- Viber là thành phần cho phép người dùng nhập các thông tin để tích hợp Viber với hệ thống.

4.2.2.4 Màn hình tạo nội dung chiến dịch quảng cáo



Hình 13 Thiết kế thành phần giao diện màn hình tạo nội dung chiến dịch quảng cáo

Màn hình tạo nội dung chiến dịch quảng cáo bao gồm 4 thành phần chính: (i) tiêu đề quảng cáo (Title), (ii) nội dung quảng cáo (Content), (iii) các nút để thêm tin nhắn vào nội dung quảng cáo (ActionMessage) và (iv) các nút để thực hiện hành động với quảng cáo vừa tạo nội dung (ActionBroadcast) như trong **Hình 13**. Trong đó thành phần Content có thể chứa nhiều loại tin nhắn và có thể sắp xếp thứ tự được gửi đến người nhận. Có 5 loại tin nhắn được phép gửi tin nhắn, tương ứng với 5 nút trong thành phần ActionMessage đó là: tin nhắn văn bản, tin nhắn hình ảnh, tin nhắn âm thanh, tin nhắn video và tin nhắn catalog. Đối với

thành phần ActionBroadcast thì người dùng có thể hủy bỏ không tạo quảng cáo nữa, lưu nháp nội dung vừa tạo hoặc chuyển sang màn hình tiếp theo đó là chọn đối tượng gửi quảng cáo.

4.3 Thiết kế chi tiết backend

4.3.1 Luồng hoạt động

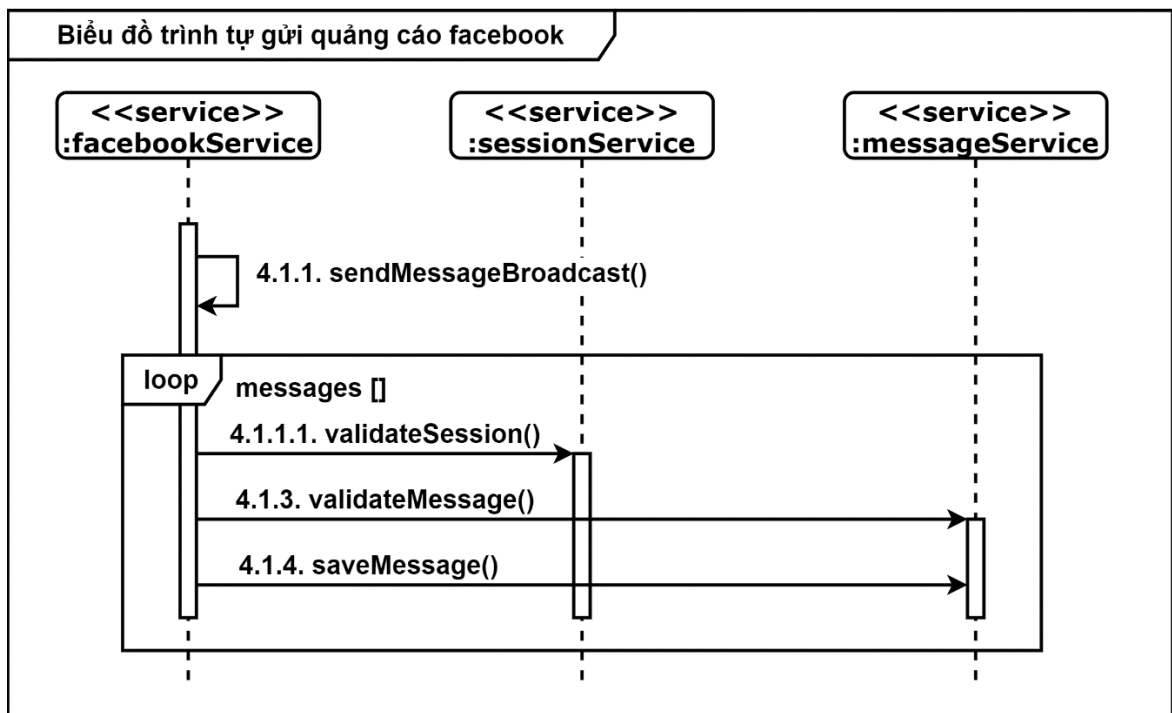
Phần này em sẽ trình bày biểu đồ trình thực các chức năng của hệ thống quản lý nhắn tin đa kênh. Tuy nhiên, do phạm vi báo cáo có giới hạn nên em sẽ trình bày biểu đồ trình tự của hai chức năng Tạo chiến dịch quảng cáo và Tích hợp Zalo.

4.3.1.1 Luồng hoạt động Tạo chiến dịch quảng cáo

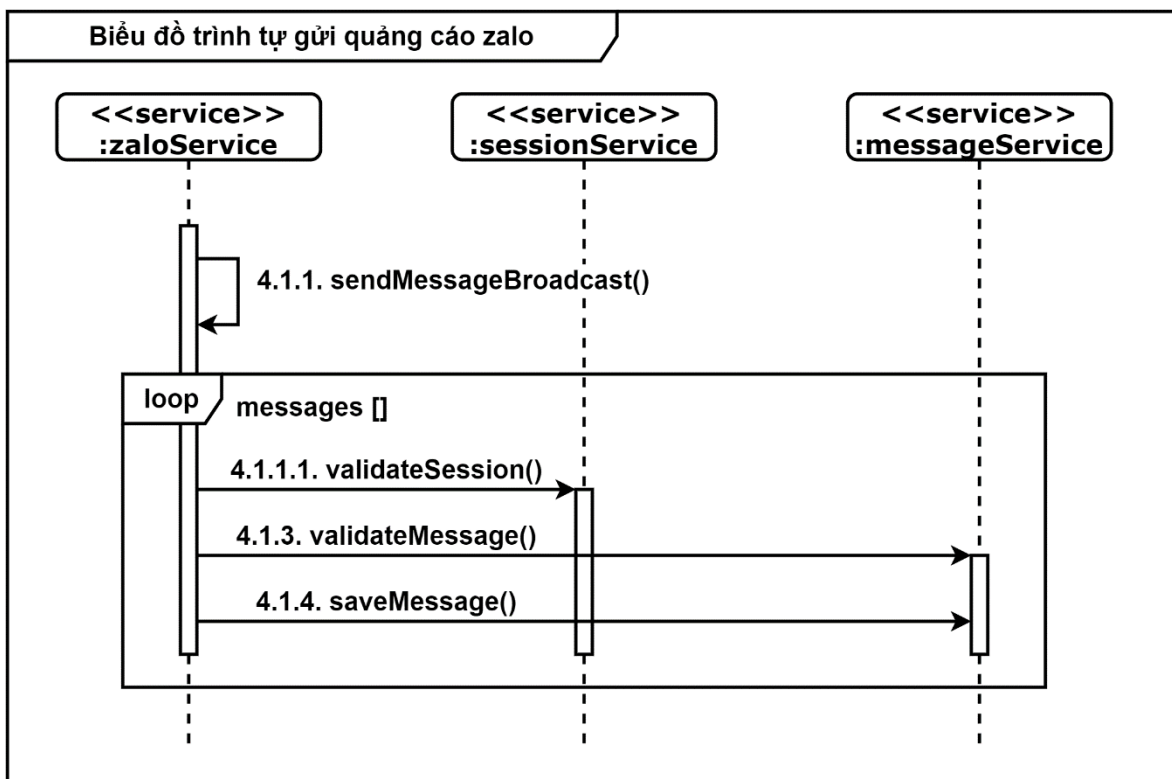
Hình 17 thể hiện trình tự xử lý việc tạo chiến dịch quảng cáo từ khi dịch vụ nhắn tin nhận được yêu cầu từ hàng đợi tin nhắn. Trước đó, người dùng cần phải thực hiện bốn bước (i) tạo nội dung chiến dịch quảng cáo, (ii) chọn đối tượng gửi quảng cáo, (iii) chọn lịch gửi quảng cáo, (iv) xác nhận gửi chiến dịch quảng cáo. Khi người xác nhận gửi chiến dịch quảng cáo thì yêu cầu đó sẽ được gửi đến và xử lý ở dịch vụ quản lý, kiểm tra thông tin đầu vào, lưu cơ sở dữ liệu sau đó mới gửi tin nhắn yêu cầu đến hàng đợi cho dịch vụ nhắn tin xử lý tiếp.

Khi dịch vụ nhắn tin nhận được tin nhắn yêu cầu xử lý quảng cáo thì sẽ gọi đến phương thức `handleBroadcastMessage` của `broadcastService`. Phương thức này sẽ lấy danh sách thông tin các đối tượng sẽ được gửi quảng cáo (`getBroadcastList`), sau đó xử lý gửi tin nhắn cho từng kênh nhắn tin (`sendBroadcast`). Chi tiết về luồng xử lý của Tạo chiến dịch quảng cáo sẽ được trình bày chi tiết trong phần 5.2.

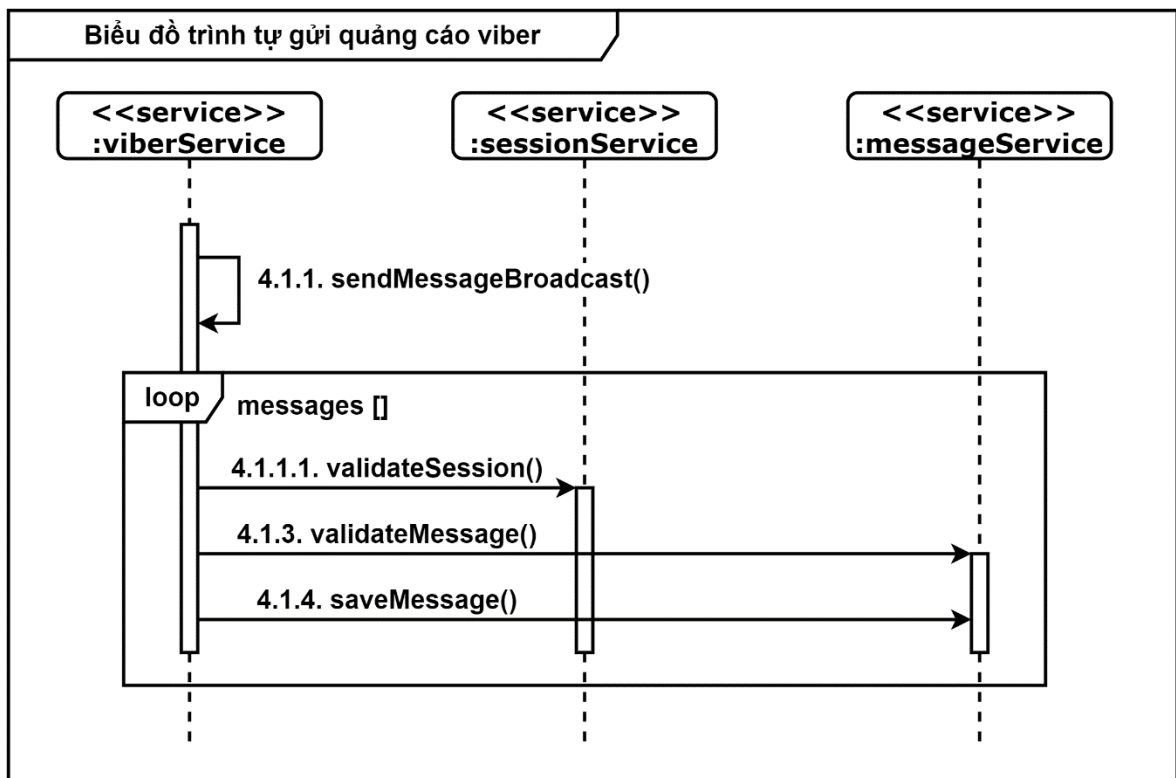
Do biểu đồ trình tự tạo chiến dịch quảng cáo dài và phức tạp, tuy nhiên có thể chia gọn thành các nhóm hành động xử lý riêng cho từng kênh nhắn tin. Chính vì vậy, em đã chia biểu đồ này thành ba biểu đồ con gửi quảng cáo Facebook (chi tiết thể hiện trong **Hình 14**), gửi quảng cáo Zalo (chi tiết thể hiện trong **Hình 15**) và gửi quảng cáo Viber (chi tiết thể hiện trong **Hình 16**).



Hình 14 Biểu đồ trình tự gửi quảng cáo Facebook



Hình 15 Biểu đồ trình tự gửi quảng cáo Zalo

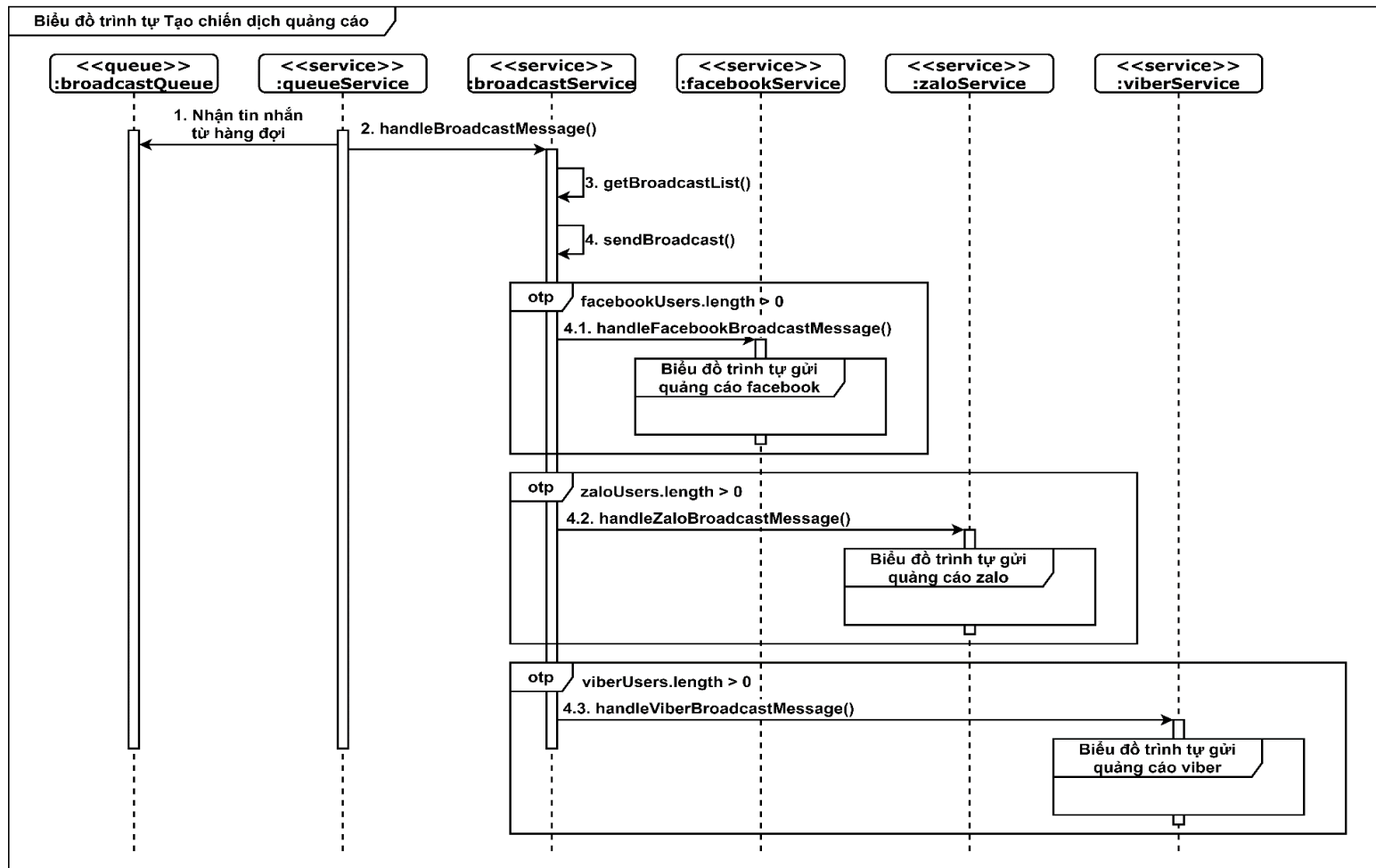


Hình 16 Biểu đồ trình tự gửi quảng cáo Viber

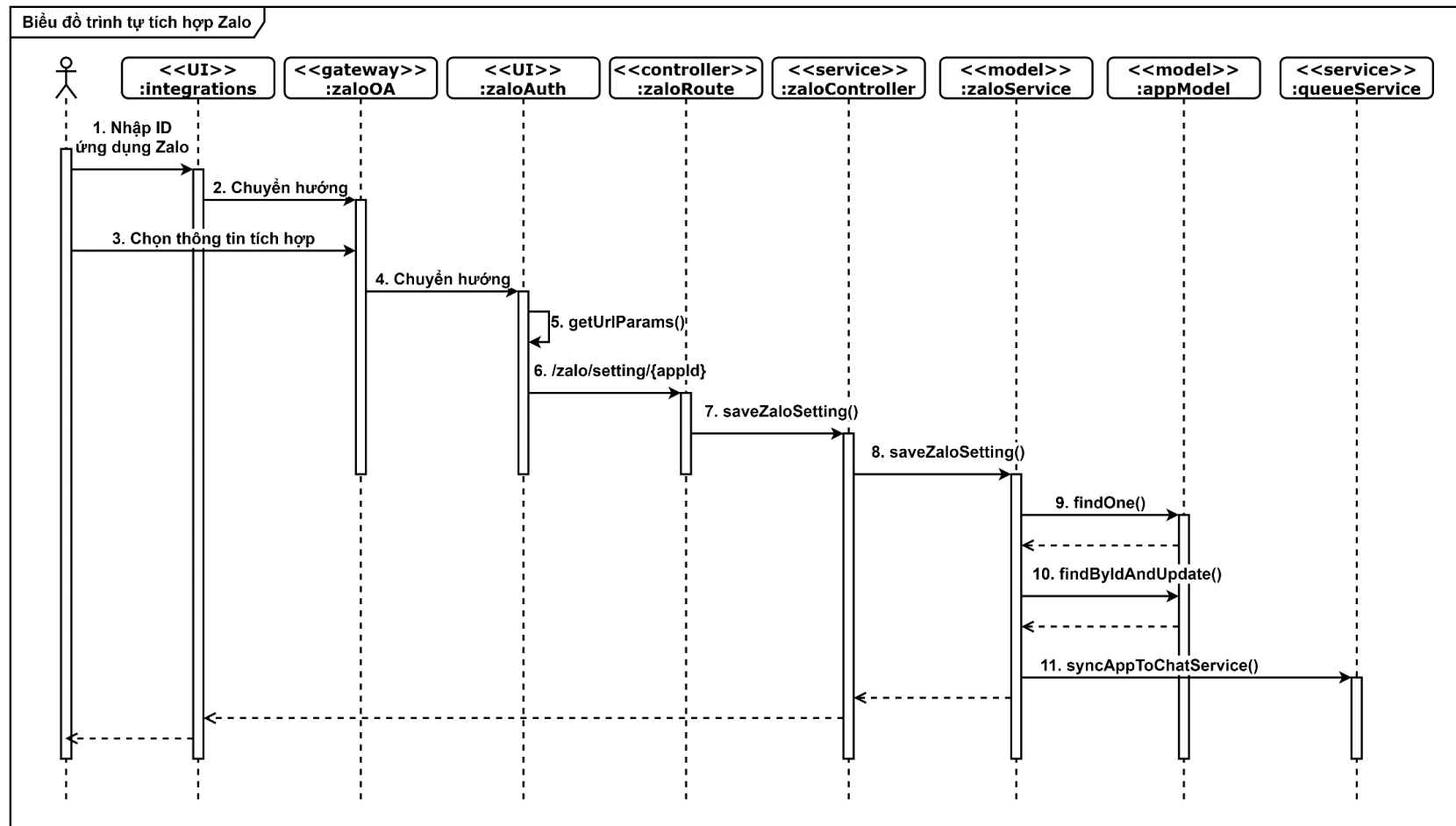
4.3.1.2 Luồng hoạt động tích hợp Zalo

Trình tự xử lý phía khi tích hợp Zalo vào ứng dụng quản lý nhắn tin đa kênh được thể hiện qua **Hình 18**.

1. Người dùng xác nhận cập nhật thông tin và nhập ID của ứng dụng đã tạo trên Zalo.
2. Hệ thống chuyển hướng đến trang chọn thông tin tích hợp của Zalo.
3. Người dùng chọn thông tin muốn tích hợp.
4. Server Zalo sẽ chuyển hướng về đường dẫn đã thiết lập trên trang của Zalo.
5. Hệ thống lấy tham số từ server Zalo truyền về qua đường dẫn đã thiết lập.
6. Hệ thống gửi HTTP Request đến cho dịch vụ quản lý.
7. Khi nhận được yêu cầu, router sẽ điều hướng đến controller chịu trách nhiệm xử lý yêu cầu đó.
8. Sau khi kiểm tra các thông tin đầu vào, controller sẽ gọi hàm đến service để xử lý tiếp.
9. Tại đây sẽ kiểm tra sự tồn tại về mặt dữ liệu.
10. Khi tất cả dữ liệu đều hợp lệ thì sẽ gọi hàm đến model để cập nhật cơ sở dữ liệu.
11. Sau khi cập nhật thành công sẽ gọi hàm để gửi yêu cầu đồng bộ dữ liệu sang dịch vụ nhắn tin bằng cách truyền lên rabbitmq.



Hình 17 Biểu đồ trình tự Tạo chiến dịch quảng cáo



Hình 18 Biểu đồ trình tự tích hợp Zalo

4.3.2 Thiết kế API

Như đã trình bày trong 4.1, các dịch vụ giao tiếp với nhau và với giao diện qua API hoặc RabbitMQ. Hệ thống cung cấp 15 API và sử dụng 2 hàng đợi tin nhắn, cụ thể API các dịch vụ cung cấp được mô tả trong **Bảng 9** và danh sách các hàng đợi tin nhắn được mô tả trong

Bảng 10.

Bảng 9 Danh sách API của hệ thống quản lý nhắn tin đa kênh

STT	Dịch vụ	Mục đích	Phương thức	Địa chỉ
1	Dịch vụ quản lý	Tạo ứng dụng	POST	/api/v1/apps
2		Lấy danh sách ứng dụng	GET	/api/v1/apps
3		Lấy thông tin ứng dụng theo ID	GET	/api/v1/apps/:appId
4		Cập nhật thông tin ứng dụng theo ID	PUT	/api/v1/apps/:appId
5		Xóa ứng dụng theo ID	DELETE	/api/v1/apps/:appId
6		Chia sẻ ứng dụng	POST	/api/v1/apps/:appId/assign-role
7	Dịch vụ nhắn tin	Lấy danh sách khách hàng	GET	/api/v1/users
8		Lấy thông tin khách hàng theo ID	GET	/api/v1/users/:userId
9		Lấy danh sách khách hàng kể từ một userId, sắp xếp theo thời gian tin nhắn cuối đến hệ thống	GET	/api/v1/skip-user

STT	Dịch vụ	Mục đích	Phương thức	Địa chỉ
10		Lấy danh sách tin nhắn của một khách hàng	GET	/api/v1/messages
11		Lấy danh sách tin nhắn kể từ một messageID, sắp xếp theo thời gian tin nhắn được tạo	GET	/api/v1/skip-messages
12		Lấy thông tin thống kê trong một ngày	GET	/api/v1/statistics/dashboard/by-hour
13		Lấy thông tin thống kê trong nhiều ngày	GET	/api/v1/statistics/dashboard/by-date
14	Dịch vụ tải tập tin	Tải tập tin dưới dạng file	POST	/api/v1/uploads/file
15		Tải ảnh dưới dạng base64	POST	/api/v1/uploads/base64

Bảng 10 Danh sách hàng đợi tin nhắn

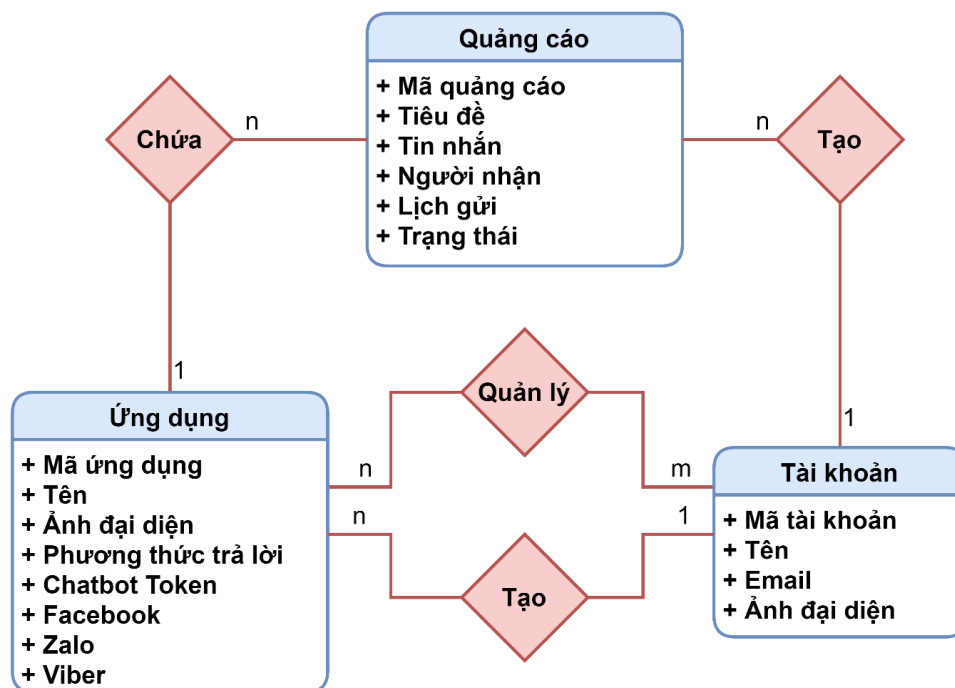
STT	Mục đích	Hàng đợi
1	CHAT_QUEUE	Dịch vụ quản lý gửi yêu cầu và trả kết quả cho dịch vụ nhắn tin
2	RESPONSE_CHAT_QUEUE	Dịch vụ nhắn tin gửi yêu cầu và kết quả cho dịch vụ quản lý

4.4 Thiết kế cơ sở dữ liệu

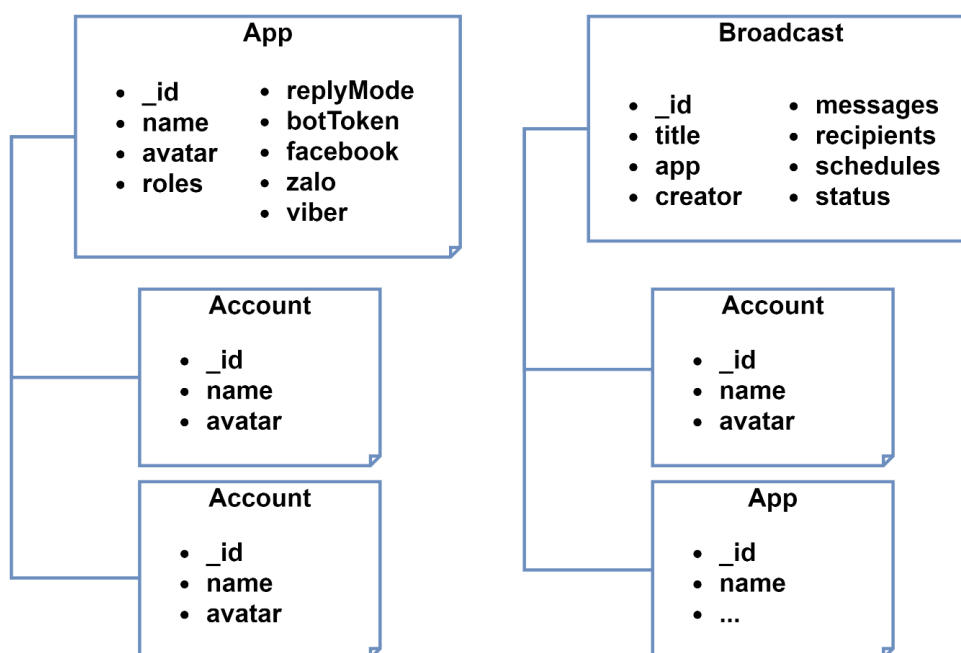
Như đã trình bày trong phần 4.1 về kiến trúc tổng quan, dịch vụ quản lý và dịch vụ nhắn tin sẽ có cơ sở dữ liệu riêng, dịch vụ tải lên tập tin không kết nối đến cơ sở dữ liệu. Do đó, trong hệ thống này, với ba dịch vụ sẽ có hai cơ sở dữ liệu.

4.4.1 Cơ sở dữ liệu dịch vụ quản lý

Hình 19 là biểu đồ thực thể liên kết của dịch vụ quản lý thể hiện mối quan hệ giữa Ứng dụng quản lý nhắn tin đa kênh, Tài khoản và Quảng cáo. Một tài khoản có thể quản lý nhiều ứng dụng và một ứng dụng có thể do nhiều tài khoản quản lý. Một ứng dụng có thể chứa nhiều quảng cáo nhưng một quảng cáo chỉ có thể thuộc về một ứng dụng, tương tự, một tài khoản có thể tạo nhiều quảng cáo nhưng một quảng cáo chỉ có thể do một người tạo.



Hình 19 Biểu đồ thực thể liên kết dịch vụ quản lý



Hình 20 Thiết kế tổng quan cơ sở dữ liệu dịch vụ quản lý

Từ biểu đồ thực thể liên kết em đã thiết kế cơ sở dữ liệu NoSQL MongoDB như **Hình 20**. Cơ sở dữ liệu gồm ba collections là App, Account và Broadcast. Collection App sẽ tham chiếu đến collection Account thông qua trường dữ liệu `_id` của Account. Collection Broadcast sẽ tham chiếu đến hai collection App và Account thông qua trường dữ liệu `_id` của hai collection này. Cụ thể chi tiết ý nghĩa của các trường dữ liệu trong các collection được mô tả qua **Bảng 11**.

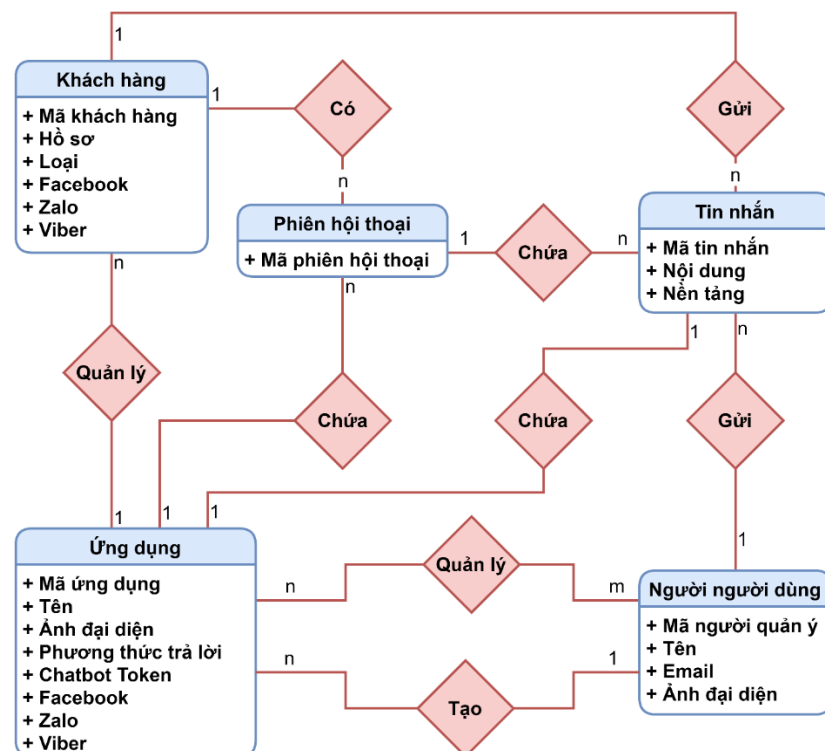
Bảng 11 Thiết kế chi tiết cơ sở dữ liệu dịch vụ quản lý

Collection	Tên trường	Kiểu dữ liệu	Ý nghĩa
Account	<code>_id</code>	ObjectId	ID của tài khoản trong hệ thống
	<code>name</code>	String	Tên người dùng của tài khoản
	<code>avatar</code>	String	Đường dẫn đến ảnh đại diện của người dùng
App	<code>_id</code>	String	ID của ứng dụng quản lý nhắn tin đa kênh
	<code>name</code>	String	Tên của ứng dụng quản lý nhắn tin đa kênh
	<code>avatar</code>	String	Đường dẫn đến ảnh đại diện của ứng dụng quản lý nhắn tin đa kênh
	<code>roles</code>	Array	Các quyền hiện có của ứng dụng quản lý nhắn tin đa kênh
	<code>replyMode</code>	String	Phương thức trả lời khi khách hàng nhắn tin đến ứng dụng quản lý nhắn tin đa kênh
	<code>botToken</code>	String	Token của chatbot bên hệ thống Smartdialog
	<code>Facebook</code>	Object	Thông tin tích hợp Facebook Messenger
	<code>Zalo</code>	Object	Thông tin tích hợp Zalo
	<code>Viber</code>	Object	Thông tin tích hợp Viber
Broadcast	<code>_id</code>	ObjectId	ID của chiến dịch quảng cáo
	<code>title</code>	String	Tiêu đề của chiến dịch quảng cáo

Collection	Tên trường	Kiểu dữ liệu	Ý nghĩa
	app	String	ID của ứng dụng quản lý nhắn tin đa kênh chứa quảng cáo
	creator	ObjectId	ID của tài khoản tạo chiến dịch quảng cáo
	messages	Object	Nội dung tin nhắn của chiến dịch quảng cáo
	recipients	Object	Thông tin nhóm người sẽ nhận được tin nhắn quảng cáo
	schedules	Object	Lịch gửi tin nhắn quảng cáo
	status	String	Trạng thái của chiến dịch quảng cáo

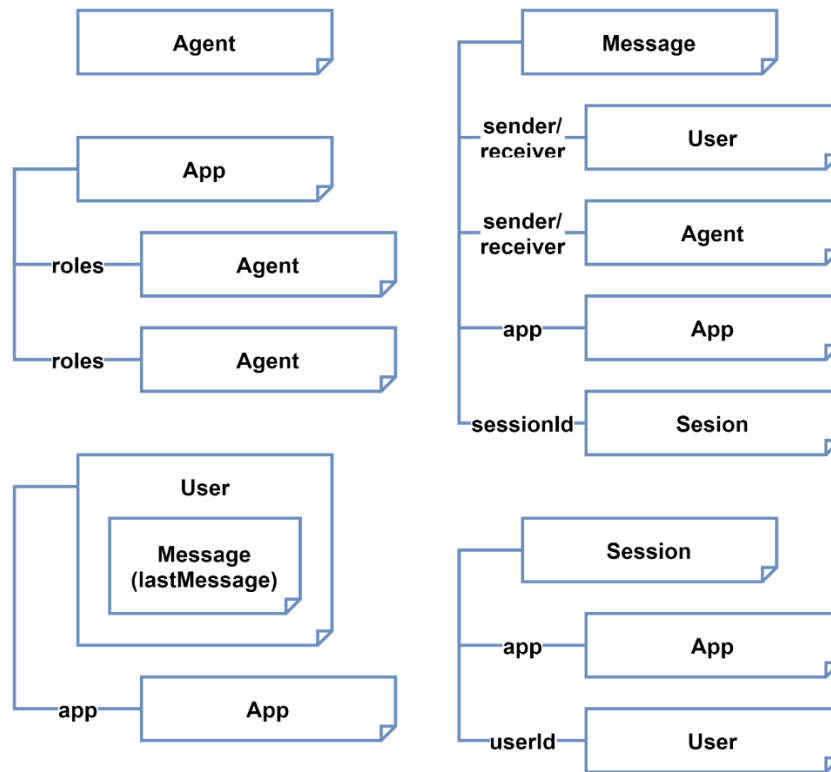
4.4.2 Cơ sở dữ liệu dịch vụ nhắn tin

Hình 21 là biểu đồ thực thể liên kết thể hiện mối quan hệ giữa ứng dụng quản lý nhắn tin đa kênh, người quản lý, khách hàng, tin nhắn và phiên hội thoại của dịch vụ nhắn tin. Tương tự như dịch vụ quản lý Một người quản lý có thể quản lý nhiều ứng dụng và một ứng dụng có thể do nhiều người cùng quản lý.



Hình 21 Biểu đồ thực thể liên kết dịch vụ nhắn tin

Từ biểu đồ thực thể liên kết em đã thiết kế cơ sở dữ liệu dịch vụ nhắn tin như **Hình 22**. Dịch vụ này có 5 collections là Agent, App, User, Message và Session. App và Agent là collection được đồng bộ từ dịch vụ quản lý, trong đó, Agent tương ứng với Account. User lưu trữ thông tin liên quan đến Khách hàng, Message lưu trữ các tin nhắn và Session lưu các phiên hội thoại, mỗi phiên hội thoại sẽ bắt đầu từ lúc Khách hàng gửi tin nhắn đầu tiên đến sau khi tin nhắn cuối cùng được gửi 10 phút.



Hình 22 Thiết kế tổng quan cơ sở dữ liệu dịch vụ nhắn tin

Thiết kế chi tiết cho các collection của dịch vụ nhắn tin được trình bày trong **Bảng 12**. Do collection App và Agent đã trình bày tại **Bảng 11** trong dịch vụ quản lý nên em sẽ chỉ thiết kế cho collection User, Message và Session.

Bảng 12 Thiết kế chi tiết cơ sở dữ liệu dịch vụ nhắn tin

Collection	Tên trường	Kiểu dữ liệu	Ý nghĩa
Session	_id	String	ID của phiên hội thoại
	userId	ObjectId	ID của khách hàng có phiên hội thoại tương ứng
	app	String	ID của ứng dụng chứa phiên hội thoại tương ứng

User	_id	ObjectId	ID của khách hàng
	type	String	Loại khách hàng (Anonymous, Facebook, Zalo, Viber)
	profile	Object	Thông tin chung của khách hàng (fullName, avatar, dateOfBirth, gender)
	Facebook	Object	Thông tin của khách hàng trên Facebook
	Zalo	Object	Thông tin của khách hàng trên Zalo
	Viber	Object	Thông tin của khách hàng trên Viber
Message	_id	ObjectId	ID của tin nhắn
	content	Object	Nội dung của tin nhắn
	platform	String	Nền tảng tin nhắn gửi đến
	receiver	Object	Thông tin người nhận tin nhắn
	sender	Object	Thông tin người gửi tin nhắn
	app	String	ID của ứng dụng chứa tin nhắn
	sessionId	String	ID của phiên hội thoại chứa tin nhắn

4.5 Xây dựng ứng dụng

4.5.1 Thư viện và công cụ sử dụng

Trong quá trình thực hiện đồ án, em có sử dụng một số công cụ hỗ trợ cho việc xây dựng frontend, backend và triển khai hệ thống như **Bảng 13** dưới đây.

Bảng 13 Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual Studio Code	https://code.visualstudio.com

Mục đích	Công cụ	Địa chỉ URL
Công cụ làm việc với CSDL MongoDB	Robo 3T	https://robomongo.org
Ngôn ngữ lập trình	Javascript	https://developer.mozilla.org/en-US/docs/Web/JavaScript
Nền tảng xây dựng backend	NodeJS	https://nodejs.org
Framework xây dựng backend	Express	https://expressjs.com
Thư viện xây dựng frontend	ReactJS	https://ReactJS.org
Thư viện React Component	Material-ui	https://material-ui.com/
Thư viện viết CSS cho ứng dụng React	Styled-Components	https://styled-components.com
Thư viện xử lý thời gian	Moment	https://momentjs.com
Thư viện gửi HTTP request	Axios	https://github.com/axios/axios
Triển khai hệ thống	Docker	https://www.docker.com

4.5.2 Kết quả đạt được

Cửa sổ livechat và hệ thống quản lý được xây dựng hướng tới một sản phẩm thật dùng thật. Hiện tại các dịch vụ cũng như các giao diện đã được xây dựng, triển khai lên server và gắn tên miền. Chi tiết về các tên miền được trình bày ở **Bảng 14** dưới đây.

Bảng 14 Các đường dẫn các dịch vụ và giao diện của hệ thống

Giao diện / Dịch vụ	Tên miền
Giao diện quản lý nhắn tin đa kênh	https://dashboard.irstech.club
Giao diện cửa sổ livechat	https://livechat.irstech.club
Dịch vụ quản lý	https://management.irstech.club

Dịch vụ nhắn tin	https://chat.iristech.club
Dịch vụ tải tập tin	https://cdn.iristech.club

4.5.3 Minh họa các chức năng chính

Trong phần này em sẽ minh họa một số chức năng chính của hệ thống quản lý nhắn tin đa kênh đa kênh và cửa sổ livechat trên nền tảng web. Cụ thể là các giao diện và chức năng sau đây (i) Thống kê, (ii) Hỗ trợ trực tuyến, (iv) Tích hợp và (v) Cửa sổ nhắn tin. Giao diện Tạo chiến dịch quảng cáo sẽ được trình bày trong phần 5.2.3.

4.5.3.1 Thống kê

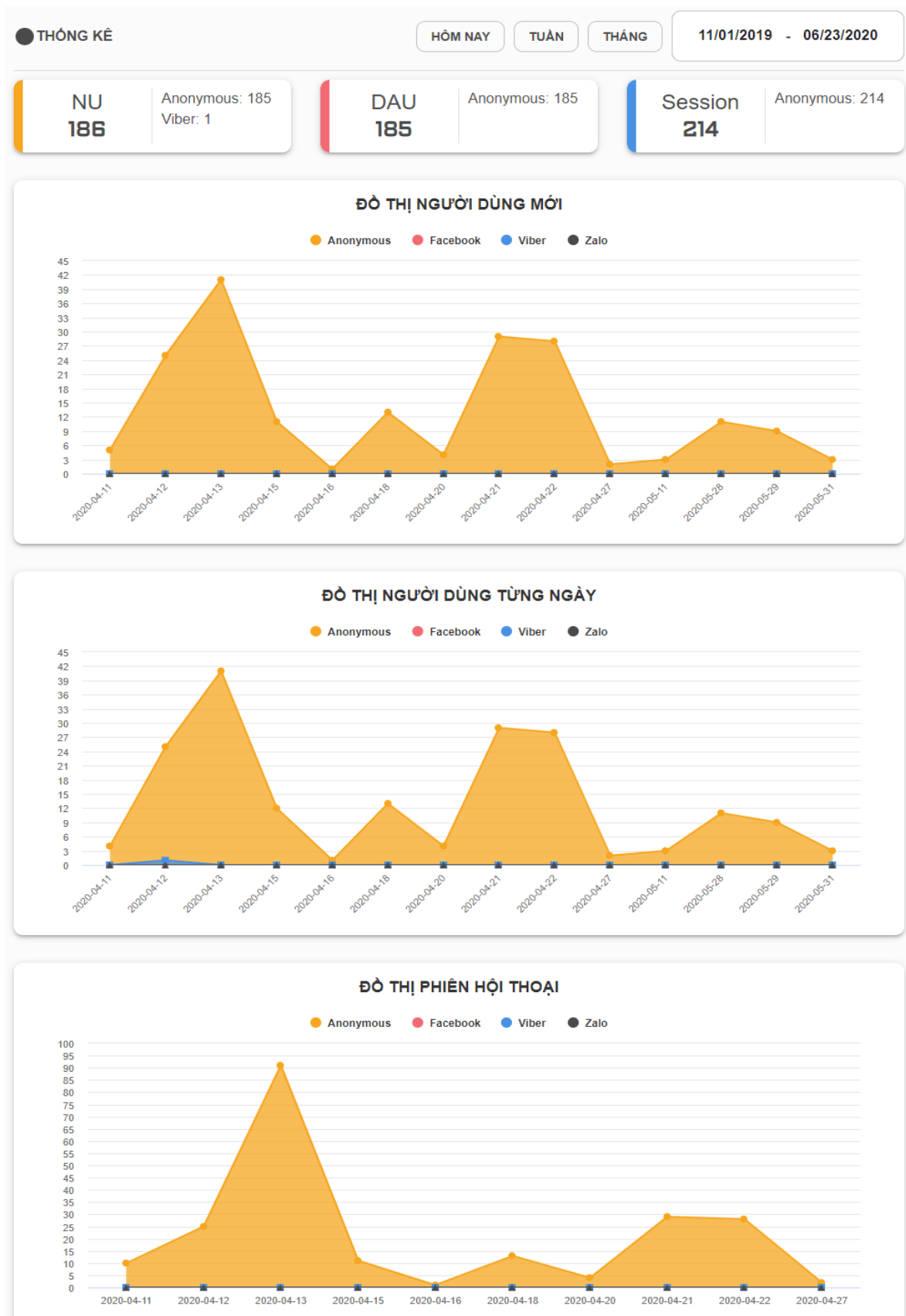
Hình 23 là giao diện thống kê của hệ thống quản lý nhắn tin đa kênh. Như đã thiết kế trong phần 4.2.2.1, màn hình thống kê có phần Header là các nút để chọn khoảng thời gian thống kê phía trên cùng, tiếp sau đó là phần Summary hiển thị các chỉ số tổng quan về số lượng khách hàng của ứng dụng, phần còn lại là ba biểu đồ chi tiết thể hiện số lượng người dùng (khách hàng) mới, số lượng người dùng (khách hàng) từng ngày và phiên hội thoại theo từng kênh nhắn tin.

4.5.3.2 Hỗ trợ trực tuyến

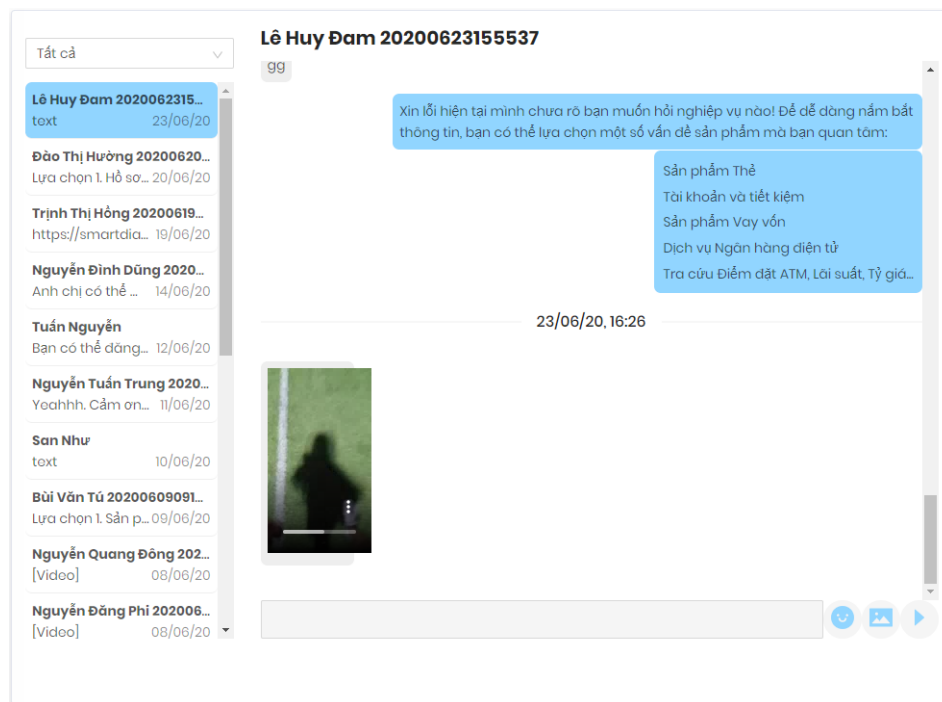
Chức năng hỗ trợ trực tuyến cho phép tổng đài viên nhắn tin với khách hàng của nhiều kênh nhắn tin trên một giao diện duy nhất. Tổng đài viên có thể chọn khách hàng từ cột phía bên trái, khi đó, nội dung tin nhắn giữa khách hàng với hệ thống sẽ được hiển thị phía bên phải và có thể nhắn tin tại đây. Tổng đài viên cũng có thể lọc khách hàng theo từng kênh nhắn tin (cửa sổ livechat, Facebook Messenger, Zalo, Viber) để tìm kiếm dễ dàng hơn.

4.5.3.3 Tích hợp

Chức năng tích hợp có giao diện như **Hình 25** và **Hình 26**. Mỗi khi người dùng muốn tích hợp thêm hoặc cập nhật các thông tin đã tích hợp thì sẽ vào đây để thực hiện hành động. Người dùng có thể thêm chatbot qua khung “Tích hợp chatbot”, lấy đoạn mã để tích hợp cửa sổ livechat vào website qua khung “Tích hợp vào trang web”, lấy đường dẫn đến trang nhắn tin thử khi tích hợp cửa sổ livechat qua khung “Trang nhắn tin của bạn”, thêm kênh nhắn tin Zalo, Facebook, Viber tại khung “Tích hợp Zalo”, “Tích hợp Facebook”, “Tích hợp Viber”



Hình 23 Giao diện thống kê



Hình 24 Giao diện hỗ trợ trực tuyến

Tích hợp Chatbot

Tích hợp với chatbot của bạn trên Smartdialog để trả lời các câu hỏi 24/7 một cách tự động, Giới thiệu với khách hàng các sản phẩm và dịch vụ của bạn, và cải thiện trải nghiệm của người dùng.

Chatbot Token

CẬP NHẬT

Tích hợp vào trang web

Tích hợp với chatbot của bạn trên Smartdialog để trả lời các câu hỏi 24/7 một cách tự động, Giới thiệu với khách hàng các sản phẩm và dịch vụ của bạn, và cải thiện trải nghiệm của người dùng.

- Sao chép và dán mã này trước thẻ đóng </body> trên mỗi trang của website.

```

<!-- Start of LiveChat code -->
<div id="live-chat" />
<script>window.appId="f4b0c7bf-ecf9-4719-a7e3-81dc13fccacd"</script>
<script type="text/javascript" src="https://dev-livechat.iristech.club/sdk.js">
</script>
<!-- End of LiveChat code -->

```
- Tải lại trang web của bạn. LiveChat sẽ xuất hiện ở góc dưới bên phải.

Trang nhắn tin của bạn

Chia sẻ liên kết trang trò chuyện của bạn để khách hàng có thể dễ dàng liên hệ với doanh nghiệp của bạn mọi lúc mọi nơi. [Thử ngay](#)

https://dev-dashboard.iristech.club/apps/f4b0c7bf-ecf9-4719-a7e3-81dc13fccacd/demo-chat

Hình 25 Giao diện tích hợp 1



Tích hợp Zalo

Tích hợp với ứng dụng của bạn trên Zalo để có thể xem và trả lời tin nhắn được gửi đến tài khoản Zalo trong Live Chat của bạn, giữ tất cả thông tin liên lạc khách hàng của bạn ở một nơi.

- 1 Tạo ứng dụng và Official Account theo hướng dẫn [tại đây](#).
- 2 Sao chép Đường dẫn chuyển hướng để cấu hình Official Account Callback Uri theo hướng dẫn [tại đây](#).
- 3 Nhập ID ứng dụng đã tạo trên Zalo và bắt đầu tích hợp

Đường dẫn chuyển hướng

`https://dev-dashboard.iristech.club/apps/f4b0c7bf-ecf9-4719-a7e3-81dc13fccacd/zalo/callback`



Official Account ID

Hiện không có dữ liệu.

CẬP NHẬT



Tích hợp Facebook

Tích hợp với ứng dụng của bạn trên Facebook để nhận tất cả tin nhắn từ trang Facebook của bạn trực tiếp trong Live Chat, giữ tất cả thông tin liên lạc khách hàng của bạn ở một nơi.

Danh sách trang đã kết nối

Hiện không có dữ liệu.

KẾT NỐI VỚI FACEBOOK



Tích hợp Viber

Tích hợp với ứng dụng của bạn trên Viber để có thể xem và trả lời tin nhắn được gửi đến tài khoản Viber trong Live Chat của bạn, giữ tất cả thông tin liên lạc khách hàng của bạn ở một nơi.

- 1 Tạo tài khoản Viber [tại đây](#).
- 2 Nhập mã xác thực tài khoản và bắt đầu tích hợp

Mã xác thực tài khoản

.....

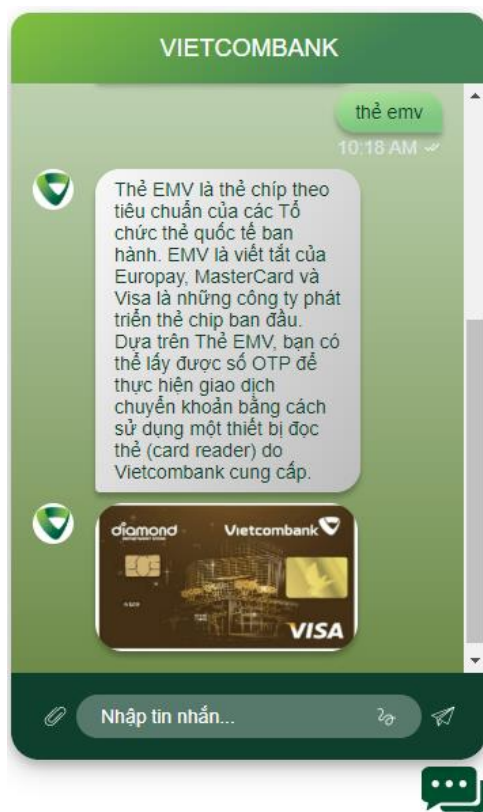


CẬP NHẬT

Hình 26 Giao diện tích hợp 2

4.5.3.4 Cửa sổ livechat

Hình 27 là giao diện cửa sổ livechat khi tích hợp vào website và người dùng đã nhấn tin trên đó. Đúng như thiết kế trong 4.2.1.2, cửa sổ bao gồm phần tiêu đề là tên của Vietcombank, ở giữa là danh sách các tin nhắn, dưới cùng là khung để người dùng nhập tin nhắn.



Hình 27 Giao diện cửa sổ livechat

4.6 Kiểm thử và triển khai

Để kiểm thử cho hệ thống quản lý nhắn tin đa kênh và cửa sổ livechat em đã tiến hành kiểm thử hộp đen dựa trên các màn hình hiển thị. Em đã tiến hành kiểm thử với 44 test case cụ thể trong **Bảng 15**. Chi tiết kiểm thử chức năng tạo nội dung quảng cáo và gửi tin nhắn video sẽ được trình bày trong phần Phụ lục.

Bảng 15 Danh sách test case

STT	Màn hình	Tên test case
1	Tổng quan	Kiểm tra tổng thể giao diện
2		Kiểm tra thông tin hiển thị

STT	Màn hình	Tên test case
3	Hỗ trợ trực tuyến	Kiểm tra thông tin hiển thị
4		Kiểm tra tải thêm khách hàng
5		Kiểm tra tải thêm tin nhắn
6		Kiểm tra nhắn tin
7	Danh sách chiến dịch quảng cáo	Kiểm tra hiển thị danh sách chiến dịch quảng cáo
8		Kiểm tra tìm kiếm chiến dịch quảng cáo
9		Kiểm tra chọn chiến dịch quảng cáo
10		Kiểm tra button tạo chiến dịch quảng cáo
11		Kiểm tra xóa chiến dịch quảng cáo
12	Tạo chiến dịch quảng cáo	Kiểm tra tạo chiến dịch quảng cáo thành công
13		Kiểm tra tạo chiến dịch quảng cáo với trạng thái Lưu nháp
14		Kiểm tra tạo chiến dịch quảng cáo với tiêu đề trống
15		Kiểm tra tạo chiến dịch quảng cáo không có tin nhắn
16		Kiểm tra tạo chiến dịch quảng cáo với tin nhắn văn bản
17		Kiểm tra tạo chiến dịch quảng cáo với tin nhắn hình ảnh
18		Kiểm tra tạo chiến dịch quảng cáo với tin nhắn âm thanh
19		Kiểm tra tạo chiến dịch quảng cáo với tin nhắn video
20		Kiểm tra tạo chiến dịch quảng cáo với tin nhắn catalog
21	Chọn đối tượng gửi quảng cáo	Kiểm tra tích chọn các đối tượng
22		Kiểm tra button Trở lại

STT	Màn hình	Tên test case
23		Kiểm tra button Tiếp theo
24		Kiểm tra button Hủy
25	Chọn lịch gửi quảng cáo	Kiểm tra tích chọn các loại lịch gửi
26		Kiểm tra button Trở lại
27		Kiểm tra button Tiếp theo
28		Kiểm tra button Hủy
29	Xác nhận gửi chiến dịch quảng cáo	Kiểm tra hiển thị dữ liệu trên màn hình
30		Kiểm tra button Lưu
31		Kiểm tra button Trở lại
32		Kiểm tra button Hủy
33	Tích hợp	Kiểm tra hiển thị dữ liệu trên màn hình
34		Kiểm tra tích hợp chatbot
35		Kiểm tra tích hợp Zalo
36		Kiểm tra tích hợp Facebook
37		Kiểm tra tích hợp Viber
38		Kiểm tra tích vào đường dẫn nhắn tin thử
39	Cài đặt	Kiểm tra thay đổi tên
40		Kiểm tra thay đổi ảnh đại diện
41		Kiểm tra thay đổi phương thức trả lời
42		Kiểm tra thêm quyền

STT	Màn hình	Tên test case
43		Kiểm tra xóa quyền
44	Cửa sổ livechat	Kiểm tra gửi tin nhắn văn bản
42		Kiểm tra gửi tin nhắn hình ảnh
43		Kiểm tra gửi tin nhắn âm thanh
44		Kiểm tra gửi tin nhắn video

Hệ thống quản lý nhắn tin đa kênh và cửa sổ livechat hiện nay đang trong quá trình hoàn thiện và đưa vào thử nghiệm sử dụng trong thực tế theo các đường dẫn trong **Bảng 14**. Cụ thể về một số thông tin liên quan đến cấu hình server triển khai như **Bảng 16** dưới đây.

Bảng 16 Thông số cấu hình server triển khai hệ thống

Tên cấu hình	Thông số
Hệ điều hành	Ubuntu 16.04
CPU	Intel Xeon Gold 5120 2.20GHz
RAM	200MB

Chương 4 đã trình bày quá trình thiết kế, xây dựng, kiểm thử và triển khai hệ thống dựa trên công nghệ sử dụng trong Chương 3 để đáp ứng các yêu cầu trong Chương 2. Trong Chương 5 em sẽ trình bày các giải pháp và đóng góp nổi bật của bản thân trong quá trình hoàn thành đồ án.

Chương 5 Các giải pháp và đóng góp nổi bật

Chương 5 sẽ trình bày về những điều tâm đắc của bản thân em trong suốt quá trình thực hiện đồ án này. Cụ thể là thiết kế kiến trúc hệ thống sử dụng Microservices, đưa ra mô hình nhắn tin hàng loạt đa kênh và phát triển cửa sổ livechat trên nền tảng web.

5.1 Thiết kế kiến trúc hệ thống sử dụng Microservices

5.1.1 Vấn đề

Hiện nay, hầu hết các hệ thống đều được xây dựng dưới dạng khối tập trung. Với các hệ thống vừa và nhỏ thì phương pháp này rất dễ viết, dễ kiểm thử và triển khai. Tuy nhiên, khi số lượng người dùng, số lượng yêu cầu tăng thêm, việc phát triển trong một khối duy nhất dẫn đến việc làm cho ứng dụng ngày càng trở lên cồng kềnh, phức tạp. Nếu phát sinh ra lỗi, rất khó để xác định lỗi nằm ở đâu. Khi đã tìm ra lỗi cũng mất thời gian cho việc sửa chữa và triển khai hoạt động trở lại. Điều này vừa gây ra sự lãng phí về mặt thời gian vừa gây ra sự lãng phí về mặt nhân lực và tiền của. Thêm vào đó, các hệ thống dạng khối thường chỉ có thể áp dụng một hoặc một vài ngôn ngữ, công nghệ và rất khó để chuyển đổi khi cần thiết do việc thay đổi dẫn theo việc toàn bộ nhân lực phải cập nhật ngôn ngữ, công nghệ mới, và việc thay đổi mã nguồn đồng loạt rõ ràng không hề đơn giản, và chưa chắc đã thành công.

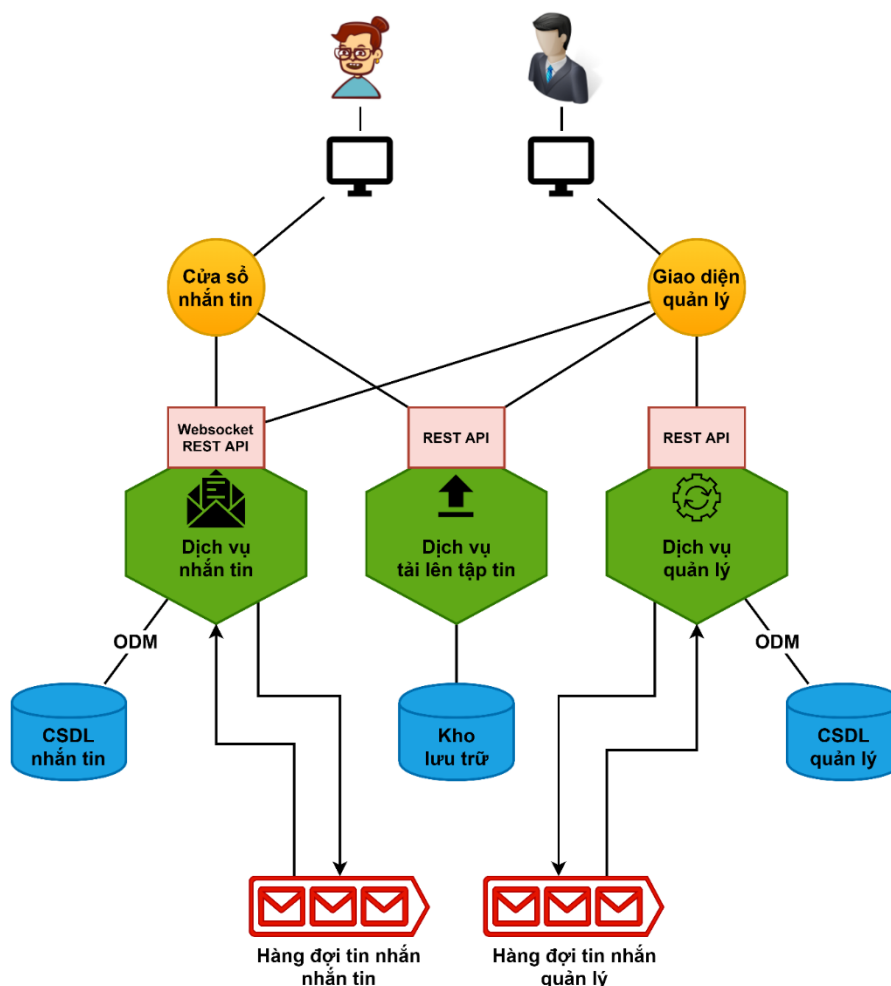
Xét về hệ thống quản lý nhắn tin đa kênh thì người dùng có thể được chia thành hai nhóm đối tượng chính. Một là khách hàng nhắn tin đến hệ thống thông qua cửa sổ livechat tích hợp trên website dịch vụ của doanh nghiệp hoặc các kênh nhắn khác (Facebook Messenger, Zalo, Viber) được tích hợp vào hệ thống. Hai là người sử dụng hệ thống quản lý ứng dụng. Hai nhóm đối tượng này có sự chênh lệch rất lớn về số lượng khi đưa vào sử dụng thực tế. Số lượng khách hàng nhắn tin đến hệ thống cùng lúc là rất nhiều trong khi số lượng người truy cập vào hệ thống quản lý là không đáng kể. Khi số lượng khách hàng nhắn tin tăng lên, hệ thống phải đối mặt với nguy cơ tắc nghẽn gây sập toàn bộ hệ thống. Có một số cách để giải quyết vấn đề trên, đó là tăng sức mạnh phần cứng cho server bằng cách lắp thêm RAM, lắp thêm chip, thay ổ cứng bằng ổ SSD hoặc thay vì tăng sức mạnh cho một server thì sẽ thêm nhiều server vào hệ thống chạy cùng một lúc. Tuy nhiên, số lượng khách hàng và người sử dụng hệ thống quản lý chênh lệch nhau rất nhiều nên khi mở rộng thì sẽ gây lãng phí,

không đủ tài nguyên phục vụ cho khách hàng trong khi tài nguyên dành cho người dùng hệ thống quản lý lại không dùng hết.

Bên cạnh đó, như đã trình bày trong phần 3.2.1, mô hình kiến trúc microservices sẽ chia nhỏ hệ thống thành các dịch vụ nhỏ, mỗi dịch vụ sẽ đảm nhiệm những chức năng khác nhau và hoạt động một cách độc lập nhất có thể. Chính vì thế em đã thiết kế hệ thống dựa trên mô hình kiến trúc microservices. Gom những chức năng phục vụ chủ yếu cho khách hàng và cho người sử dụng hệ thống quản lý tách thành các dịch vụ riêng. Khi đó có thể thoải mái mở rộng dịch vụ để phục vụ cho khách hàng mà vẫn giữ nguyên được dịch vụ cho người quản lý sử dụng.

5.1.2 Giải pháp

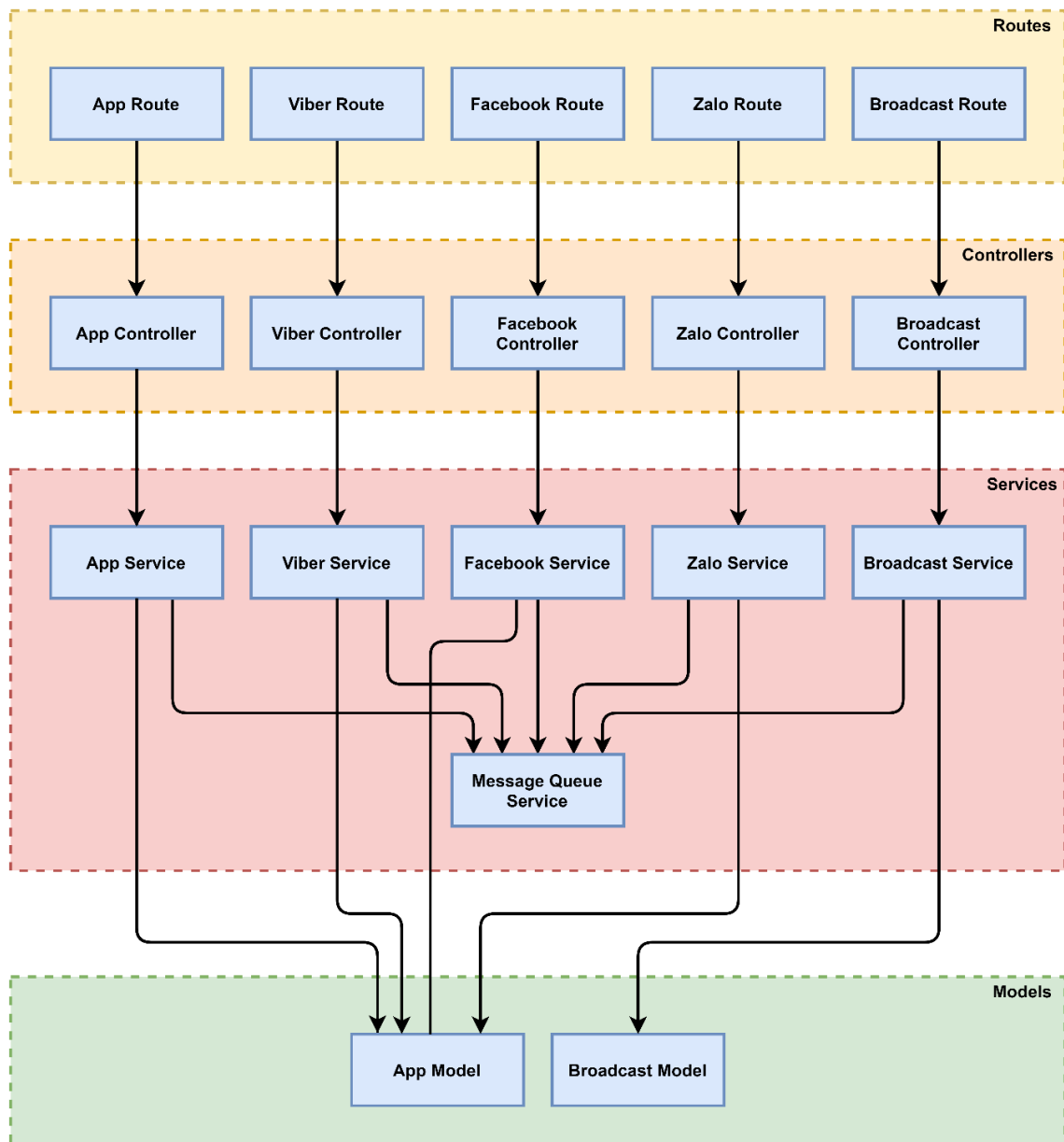
Nhận thấy ưu điểm mà kiến trúc Microservices đem lại, em đã thiết kế hệ thống dựa trên kiến trúc này. Cụ thể hệ thống được chia làm ba dịch vụ con là (i) dịch vụ nhắn tin, (ii) dịch vụ quản lý, (iii) dịch vụ tải lên tập tin như **Hình 28** bên dưới.



Hình 28 Thiết kế kiến trúc Microservices

5.1.2.1 Dịch vụ quản lý

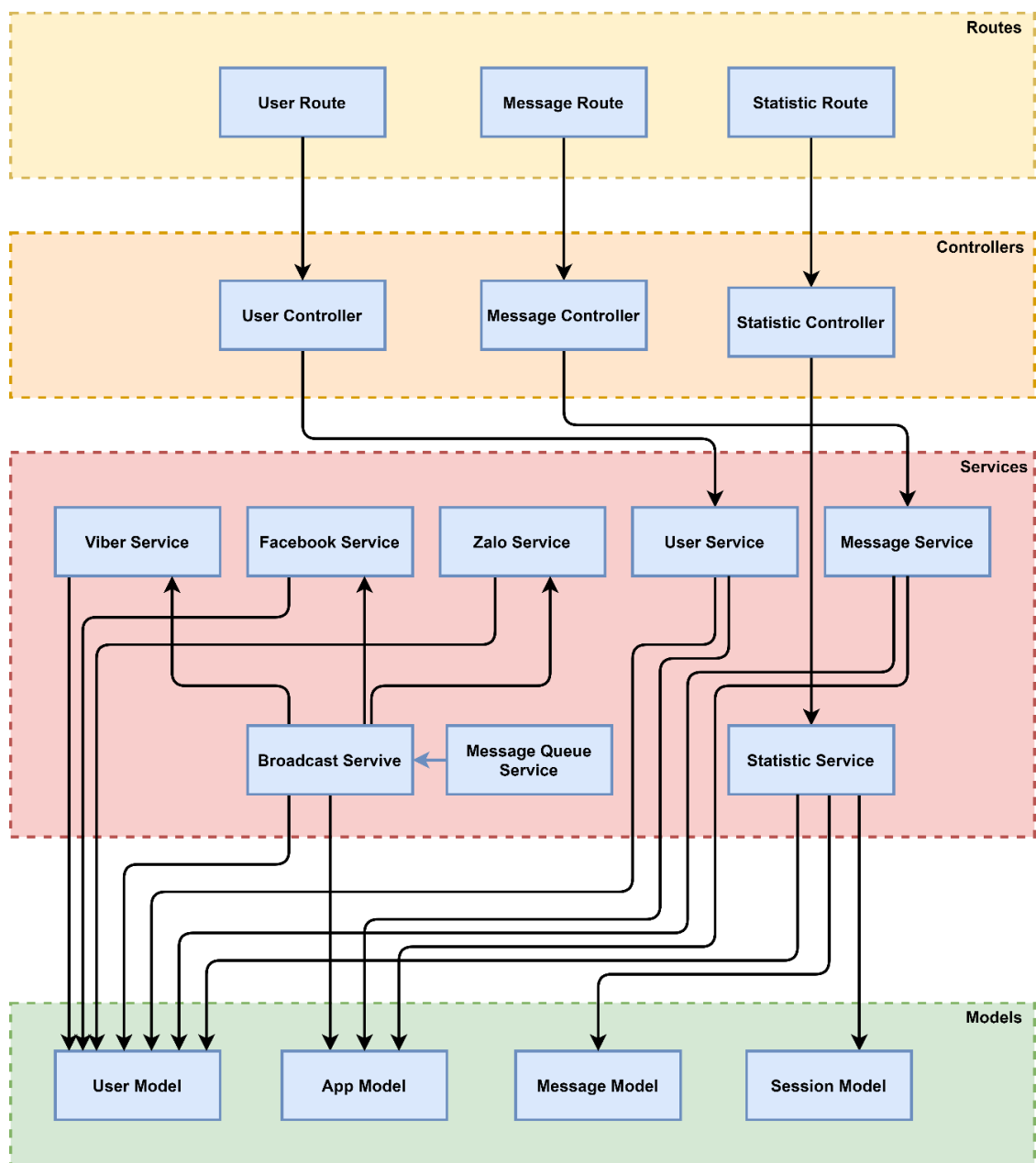
Dịch vụ quản lý chịu trách nhiệm quản lý và điều hành các ứng dụng, lưu các thông tin liên quan đến ứng dụng. Đây cũng là nơi trung gian để phục vụ cho việc tạo kết nối, giao tiếp giữa khách hàng, ứng dụng nhắn tin bên thứ ba với chatbot. Dịch vụ cũng hỗ trợ các giao thức, chuẩn kết nối cơ bản để các thành phần bên ngoài có thể dễ dàng giao tiếp với nhau như websocket, REST API. Bên cạnh đó, dịch vụ cũng phục vụ cho các nghiệp vụ như tạo chiến dịch quảng cáo, tạo nội dung quảng cáo gửi đến khách hàng. Kiến trúc chi tiết của dịch vụ quản lý được mô tả trong **Hình 29**.



Hình 29 Kiến trúc dịch vụ Quản lý

Dịch vụ quản lý sử dụng framework ExpressJS và được thiết kế bao gồm 4 thành phần là Routes, Controllers, Services, Models. Trong đó, Routes có nhiệm vụ nhận yêu cầu và điều hướng đến đúng Controller, Controllers nhận yêu cầu và xử lý xong sẽ gọi đến Services, Services thực hiện truy vấn dữ liệu đến cơ sở dữ liệu thông qua các Models và Models sẽ kết nối đến cơ sở dữ liệu MongoDB thông qua ODM. Dịch vụ quản lý cũng kết nối đến RabbitMQ để tương tác với các dịch vụ khác qua hàng đợi tin nhắn. Khi trong hàng đợi có tin nhắn đến sẽ dựa vào loại tin nhắn để gọi đến các Service phù hợp xử lý và trả về kết quả đẩy lại lên hàng đợi.

5.1.2.2 Dịch vụ nhắn tin



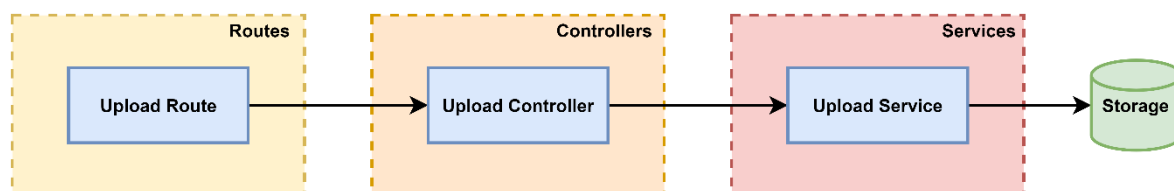
Hình 30 Kiến trúc dịch vụ Nhắn tin

Dịch vụ nhắn tin có nhiệm vụ chính là quản lý khách hàng và tin nhắn gửi đến hệ thống, tin nhắn này có thể do khách hàng gửi, tổng đài viên gửi hoặc chatbot gửi. Cụ thể dịch vụ nhắn tin sẽ tạo ra một khách hàng mới khi họ lần đầu nhắn tin đến hệ thống, quản lý tin nhắn của khách hàng theo từng phiên hội thoại, xử lý việc gửi tin nhắn từ một nơi đến nhiều kênh nhắn tin khác nhau và đây cũng là nơi để lấy các dữ liệu phục vụ cho việc thống kê. Kiến trúc chi tiết của dịch vụ nhắn tin được mô tả trong **Hình 30**.

Tương tự như dịch vụ Quản lý, dịch vụ Nhắn tin cũng cung cấp các REST API phục vụ cho việc thống kê và báo cáo, đồng thời cũng cung cấp Websocket để phục vụ cho việc gửi tin nhắn và kết nối đến RabbitMQ để giao tiếp với các dịch vụ khác. Kiến trúc của dịch vụ Nhắn tin cũng bao gồm 4 thành phần Routes, Controllers, Services và Models.

5.1.2.3 Dịch vụ tải lên tập tin

Dịch vụ tải lên tập tin chịu trách nhiệm lưu trữ các tập tin đa phương tiện. Lý do để tách chức năng tải tập tin thành dịch vụ riêng là vì đối với chức năng này không yêu cầu cao về cấu hình server mà cần kho lưu trữ lớn. Bên cạnh đó cả dịch vụ quản lý và dịch vụ nhắn tin đều cần tải tài liệu lên, thậm chí nếu tương lai hệ thống cần mở rộng hơn nữa và có thêm dịch vụ mới cũng cần tải tập tin lên thì tất cả các dịch vụ chỉ cần gọi đến dịch vụ tải lên tập tin để thực hiện chức năng này. Điều này đúng với tiêu chí của kiến trúc microservices đó là các dịch vụ hoạt động độc lập, có thể tách rời, dễ dàng triển khai, sửa đổi, mở rộng khi cần. Kiến trúc chi tiết của dịch vụ tải lên tập tin được mô tả trong **Hình 31**.



Hình 31 Kiến trúc dịch vụ Tải tập tin

Dịch vụ tải tập tin cung cấp REST API phục vụ cho việc tải tập tin và hình ảnh dạng base64 lên máy chủ. Kiến trúc của dịch vụ gồm 3 thành phần Routes, Controllers và Services giống dịch vụ quản lý và dịch vụ nhắn tin. Dịch vụ tải tập tin không kết nối với cơ sở dữ liệu mà lưu trữ dữ liệu trong kho lưu trữ (Storage).

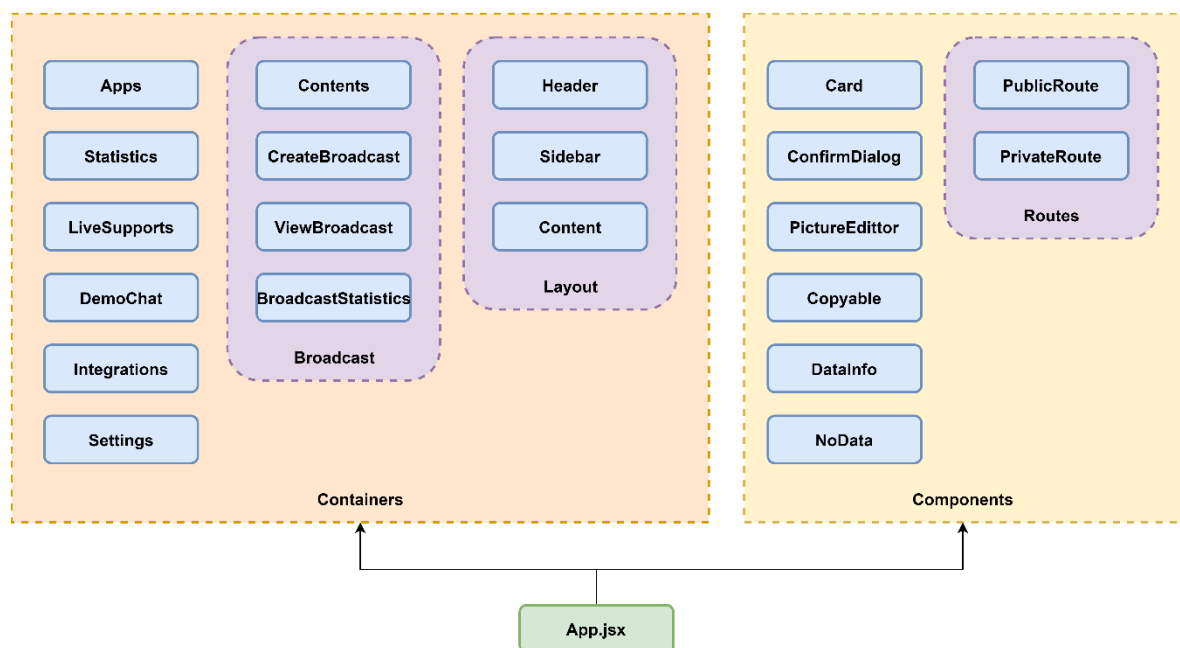
Khi hệ thống được thiết kế theo dạng microservices thì việc giao tiếp giữa các dịch vụ là vô cùng quan trọng. Để hoàn thành một tác vụ của người dùng thì kết nối và giao tiếp giữa các dịch vụ là cần thiết vì một tác vụ gồm nhiều tác động khác nhau lên các dịch vụ. Ví dụ như khi gửi tin nhắn quảng cáo thì trước hết sẽ xử lý tại dịch vụ quản lý sau đó mới gọi sang dịch vụ nhắn tin để tiến hành gửi tin nhắn đến khách hàng. Chính vì thế, em đã sử dụng mô hình hàng đợi tin nhắn để giao tiếp giữa các dịch vụ. Với mô hình này, các dịch vụ không trực

tiếp giao tiếp với nhau mà thông qua hệ thống hàng đợi tin nhắn, giao tiếp bất đồng bộ. Như **Hình 28**, dịch vụ quản lý gửi một tin nhắn đến “Hàng đợi tin nhắn quản lý”, dịch vụ nhắn tin đã đăng ký nhận tin nhắn từ hàng đợi này sẽ xử lý tin nhắn của dịch vụ quản lý.

5.1.2.4 Kiến trúc frontend

Ngoài phần backend thì hệ thống quản lý nhắn tin đa kênh còn có phần frontend. Phần frontend được thiết kế thành 2 phần chính là Containers và Components như **Hình 32**.

Phần containers chịu trách nhiệm kết xuất ra các giao diện hiển thị cho người dùng. Hệ thống có 8 containers là: (i) Apps kết xuất giao diện danh sách ứng dụng quản lý nhắn tin đa kênh của người dùng, (ii) Statistics kết xuất giao diện thống kê khách hàng, (iii) LiveSupports kết xuất giao diện nhắn tin hỗ trợ khách hàng, (iv) LiveDemo kết xuất giao diện nhắn tin thử qua cửa sổ livechat, (v) Integrations kết xuất giao diện tích hợp chatbot và các kênh nhắn tin (vi) Settings kết xuất giao diện cài đặt chung, (vii) Broadcasts là nhóm giao diện liên quan đến quảng cáo và (viii) Layout là bố cục của toàn bộ màn hình.. Nhóm giao diện Broadcast gồm: Contents chịu trách nhiệm hiển thị danh sách nội dung chiến dịch quảng cáo đã tạo, CreateBroadcast hiển thị giao diện tạo chiến dịch quảng cáo từ nội dung đến lập lịch gửi quảng cáo, ViewBroadcast hiển thị giao diện xem thông tin chiến dịch quảng cáo đã tạo và BroadcastStatistics hiển thị thống kê cho các chiến dịch quảng cáo.



Hình 32 Kiến trúc frontend hệ thống quản lý nhắn tin đa kênh

Phần Component là các thành phần dùng chung được sử dụng nhiều lần, container sẽ gọi đến component khi cần thiết. Có 7 thành phần là: Card, ConfirmDialog, PictureEditor, Copyable, DataInfo, NoData và Routes. Trong đó Routes gồm có PrivateRoute và PublicRoute. Route sẽ chịu trách nhiệm gọi đến controller phù hợp với đường dẫn trên trình

duyet mà người dùng yêu cầu. Với những đường dẫn nào được chỉ định là PrivateRoute thì chỉ truy cập được khi người dùng đã đăng nhập vào hệ thống.

5.2 Mô hình nhắn tin hàng loạt đa kênh

5.2.1 Vấn đề

Hiện nay, các doanh nghiệp đa phần không chỉ hoạt động trên website của mình mà còn ở trên các trang mạng xã hội, thậm chí tại đây có thể tiếp cận với lượng lớn khách hàng hơn là trên website, đặc biệt là đối tượng khách hàng trẻ. Vì vậy nhu cầu nhắn tin từ các trang mạng xã hội này là không hề nhỏ. Nắm bắt được xu thế trên, hệ thống quản lý nhắn tin đa kênh cho phép doanh nghiệp kết nối đến ứng dụng của mình trên các trang mạng xã hội bao gồm Facebook, Zalo và Viber. Khi khách hàng nhắn tin từ những trang mạng xã hội này, doanh nghiệp có thể xem và quản lý toàn bộ khách hàng và tin nhắn trên một trang quản lý duy nhất. Dữ liệu về tin nhắn và khách hàng được quản lý tập trung trong một dịch vụ nhắn tin.

Không chỉ chờ đợi khách hàng tìm đến mình mà hệ thống em xây dựng còn cho phép doanh nghiệp chủ động tiến đến gần hơn với khách hàng qua chức năng quảng cáo đa kênh, đa đối tượng với tin nhắn tùy chỉnh. Đây là chức năng đòi hỏi cần phải xử lý về mặt giao diện và logic hết sức phức tạp vì vậy em đã mất rất nhiều thời gian và công sức tìm hiểu, thiết kế và thực hiện để có thể tạo ra một chức năng tốt nhất.

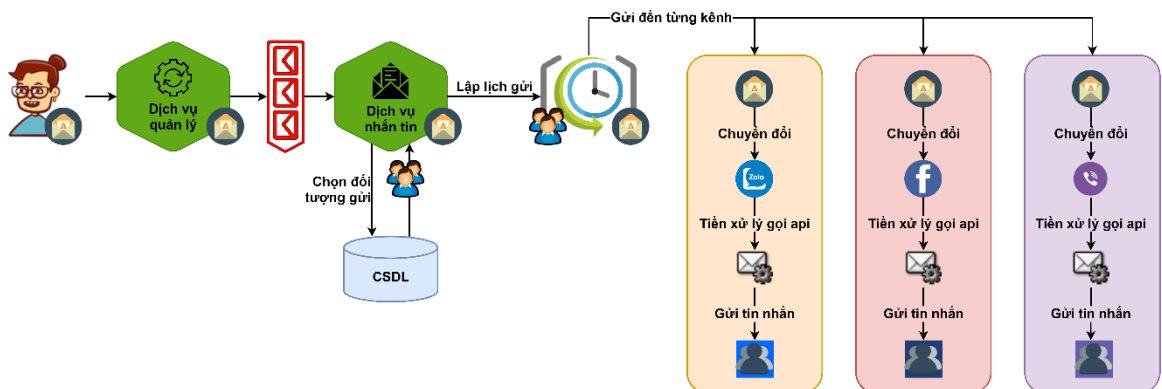
5.2.2 Giải pháp

Về mặt giao diện cần thiết kế để người dùng có thể dễ dàng tạo và tùy chỉnh nội dung chiến dịch quảng cáo cũng như việc chọn đối tượng gửi và đặt lịch gửi quảng cáo cần phải đơn giản và dễ sử dụng. Để đáp ứng được yêu cầu trên em đã chia chức năng này thành nhiều màn hình để tránh việc hiển thị quá nhiều thông tin trên một màn hình gây rối mắt. Tổng cộng chức năng quảng cáo có 5 màn hình bao gồm: (i) hiển thị danh sách nội dung quảng cáo đã tạo, (ii) tạo nội dung quảng cáo, (iii) chọn đối tượng gửi quảng cáo, (iv) đặt lịch gửi quảng cáo và (v) màn hình xác nhận các thông tin của chiến dịch quảng cáo, các màn hình này sẽ được trình bày trong phần 5.2.3.

Mỗi màn hình đều được xây dựng thành các component với cách sắp xếp, tổ chức hợp lý, dễ nâng cấp và có thể tái sử dụng. **Hình 33** là giao diện của chức năng tạo nội dung quảng cáo. Phần trên cùng là component Stepper cho người dùng biết hiện mình đang ở bước nào. Khi các bước thay đổi thì nội dung trong StepContent sẽ thay đổi tương ứng. Đối với bước đầu tiên tạo nội dung quảng cáo thì nội dung này sẽ gồm nhiều tin nhắn nằm trong SortableMessageList và nhóm các button để thêm tin nhắn vào trong danh sách các tin nhắn này.

Hình 33 Thành phần trong giao diện tạo nội dung quảng cáo

Hình 34 mô tả mô hình gửi tin nhắn hàng loạt đa kênh từ khi người dùng yêu cầu gửi quảng cáo. Yêu cầu này sẽ được gửi đến dịch vụ quản lý, tại đây sẽ tiến hành lưu cơ sở dữ liệu và gửi một tin nhắn đến hàng đợi tin nhắn. Sau khi dịch vụ nhận tin nhắn từ hàng đợi, tin nhắn sẽ lọc ra các tập đối tượng gửi và dựa vào lịch gửi sẽ tạo ra các công việc định kỳ theo lịch. Đây là phần xử lý khó nhất của việc lập lịch gửi quảng cáo. Với lịch gửi lặp lại hàng ngày nhưng kéo dài trong nhiều tháng, cần tạo nhiều công việc, mỗi công việc thực hiện trong vòng 1 tháng, riêng đối với tháng đầu và tháng cuối cần kiểm tra thêm ngày bắt đầu và ngày kết thúc để tạo công việc cho đúng lịch. Trong mỗi công việc cần lọc và xử lý gửi cho các tập đối tượng khác nhau, cụ thể tập đối tượng ở đây là theo các kênh nhắn tin. Mỗi kênh nhắn tin có logic xử lý việc gửi tin nhắn khác nhau. Đến đây, tin nhắn sẽ được tùy chỉnh theo kênh nhắn tin sao cho phù hợp với API từng kênh nhắn tin cung cấp.



Hình 34 Mô hình nhắn tin hàng loạt đa kênh

Tuy nhiên Facebook có giới hạn số lần gọi API trong 1 giây để ngăn hoạt động độc hại và trải nghiệm của người dùng kém. API gửi tin nhắn không có giới hạn cố định, vì vậy để tránh có một lượng lớn yêu cầu đột ngột tăng, hệ thống cần có cơ chế đáp ứng được. Để xử lý được việc này, em đã thực hiện gom các tin nhắn nhằm tạo ra một yêu cầu hàng loạt, bằng cách này từ nhiều lần gọi API gửi nhiều tin nhắn đã được chuyển thành một lần gọi cho nhiều tin nhắn. Nhưng số lượng tin nhắn trong một lần gửi vẫn bị giới hạn là 50 tin/lần [9], do đó, cần chia các yêu cầu hàng loạt thành nhóm 50 yêu cầu để gọi API gửi tin nhắn hàng loạt.

Zalo có giới hạn kích thước file khi gửi tin nhắn là 1GB [10], do đó, khi gửi tin nhắn đa phương tiện thì cần kiểm tra xem có vượt quá kích thước không rồi mới tiến hành gọi API gửi tin nhắn.

Đối với Viber thì sẽ giới hạn số lượng người dùng nhận được tin nhắn quảng cáo trong một lần gửi là 300 người [11]. Vì vậy, cần chia khách hàng thành các nhóm 300 người để thực hiện gọi API gửi tin nhắn hàng loạt.

5.2.3 Kết quả

Kết quả đạt được sau khi triển khai chức năng thể hiện ở các màn hình dưới đây.

Với bước 1, màn hình tạo nội dung chiến dịch quảng cáo được thiết kế để người dùng sử dụng một cách thuận tiện nhất. Khi người dùng vào trang tạo nội dung quảng cáo thì phần nội dung sẽ chỉ có khung nhập tin nhắn văn bản. Người dùng có thể nhấn vào các button phía dưới để thêm loại tin nhắn muốn gửi và kéo thả sắp xếp lại vị trí tin nhắn. **Hình 35** là giao diện khi đã thêm đủ năm loại tin nhắn hệ thống cho phép người dùng gửi, đó là: (i) tin nhắn văn bản, (ii) tin nhắn hình ảnh, (iii) tin nhắn âm thanh, (iv) tin nhắn video và (v) tin nhắn catalog.

Bước 2 là chọn đối tượng gửi quảng cáo, người dùng có thể chọn gửi cho khách hàng của kênh nhắn tin Facebook, Zalo, Viber như **Hình 36**.

Bước 3 là đặt lịch gửi quảng cáo, tại đây người dùng có thể đặt thời gian gửi quảng cáo đến khách hàng như **Hình 37**.

Cuối cùng sẽ có một bước xác nhận, người dùng có thể xem lại các thông tin của chiến dịch quảng cáo một cách tổng quan bao gồm tin nhắn sẽ gửi, đối tượng sẽ gửi, thời gian sẽ gửi như **Hình 38**. Nếu muốn tiếp tục gửi quảng cáo thì người dùng sẽ nhấn lưu để gửi hoặc nhấn trở lại để chỉnh sửa thông tin và nhấn hủy nếu không muốn gửi quảng cáo nữa, mọi hành động của người dùng sẽ không được lưu lại.

1

Tạo chiến dịch

2

Chọn đối tượng gửi

3

Đặt lịch gửi

4

Tiêu đề

Nhập tiêu đề

Nội dung

Tr

Văn bản

Nhập tin nhắn văn bản

Hình ảnh

Nhập mô tả hình ảnh

Âm thanh

TẢI LÊN

Nhập mô tả âm thanh

Video

TẢI LÊN

Nhập mô tả video

Catalogue

+ THÊM CATALOG

Tr

VĂN BẢN

HÌNH ẢNH

ÂM THANH

VIDEO

CATALOGUE

HỦY

LƯU NHÁP

TIẾP THEO

Hình 35 Giao diện tạo nội dung chiến dịch quảng cáo

61

☒ Tạo chiến dịch ·
 ☒ 2 Chọn đối tượng gửi ·
 ☐ 3 Đặt lịch gửi ·
 ☐ 4 Xác nhận chiến dịch

Kênh nhắn tin

☒ Facebook
 ☒ Zalo
 ☒ Viber

Hình 36 Giao diện chọn đối tượng gửi quảng cáo

☒ Tạo chiến dịch —
 ☒ Chọn đối tượng gửi —
 ☒ 3 Đặt lịch gửi —
 ☐ 4 Xác nhận chiến dịch

☒ Gửi ngay

☐ Một lần

Chọn thời gian
 2020-06-23 17:08

☐ Nhiều lần

☐ Hằng ngày

Ngày bắt đầu - Ngày kết thúc

Chọn thời gian
 17:08

☐ Hằng tuần

☐ Thứ hai

Chọn thời gian
 17:08

☐ Thứ ba

Chọn thời gian
 17:08

☐ Thứ tư

Chọn thời gian
 17:08

☐ Thứ năm

Chọn thời gian
 17:08

☐ Thứ sáu

Chọn thời gian
 17:08

☐ Thứ bảy

Chọn thời gian
 17:08

☐ Chủ nhật

Chọn thời gian
 17:08

☐ Hằng tháng

+ THÊM NGÀY

Hình 37 Giao diện đặt lịch gửi quảng cáo

✓ Tạo chiến dịch

✓ Chọn đối tượng gửi

✓ Đặt lịch gửi

4 Xác nhận chiến dịch

Tin nhắn


Tiêu đề

Khuyến mãi 02/9

Nội dung

Iristech khuyến mãi 10% tất cả các dịch vụ

Hình ảnh



Giảm 50% tất cả các mặt hàng túi

Đối tượng gửi

Kênh nhắn tin

Facebook

Zalo

Viber

Đặt lịch gửi

Nhiều lần

Hàng tuần

Thứ tư

vào lúc

08:30

Thứ bảy

vào lúc

09:30

HỦY

TRỞ LẠI

LƯU

Hình 38 Giao diện xác nhận gửi quảng cáo

5.3 Phát triển cửa sổ livechat trên nền tảng web

Dịch vụ hỗ trợ khách hàng là công cụ quan trọng để doanh nghiệp cạnh tranh hiệu quả, đồng thời hỗ trợ khách hàng chuyên nghiệp sẽ mang lại nguồn doanh thu vô cùng lớn. Nhận biết được việc này, hiện nay đa phần doanh nghiệp đều tập trung cải thiện dịch vụ hỗ trợ khách hàng một cách nhanh chóng và hiệu quả. Livechat chính là giải pháp nhiều doanh nghiệp lựa chọn. So với các công cụ hỗ trợ truyền thống thì Livechat giải quyết vấn đề nhanh hơn rất nhiều. khách hàng chỉ cần truy cập vào trang web của doanh nghiệp và bắt đầu một phiên hội thoại.

Tuy nhiên để có được những lợi ích trên thì quá trình tạo ra một cửa sổ livechat đẹp mắt, thu hút khách hàng, đồng thời cũng cần có thể dễ dàng tích hợp vào bất kỳ trang web nào là không hề đơn giản. Để nhúng được cửa sổ livechat vào các trang web khác em đã tìm hiểu cách thực hiện nó. Có hai cách để làm được việc này đó là sử dụng iframe và sử dụng javascript. Iframe là một thẻ HTML được sử dụng để nhúng nội dung HTML hay video vào trong một website, đây là một cách thuận tiện để hiển thị nội dung từ nguồn bên ngoài ở bất cứ đâu trên website. Tuy nhiên, iframe có thể làm chậm website và không phải tất cả các loại thiết bị đều hỗ trợ iframe. Chính vì vậy em đã lựa chọn sử dụng javascript để đưa cửa sổ livechat vào các website. Với cách làm này, cửa sổ livechat sẽ được xây dựng như một thành phần và được đóng gói thành tệp javascript, sau đó tệp javascript này sẽ được nhúng thông qua thẻ `<script>` trong trang html. Khi người dùng truy cập trang web, trình duyệt sẽ tự động thực thi đoạn mã được nhúng trong html của website.

Sau khi chọn được cách hiển thị cửa sổ livechat trên website, em tiến hành xây dựng một giao diện đẹp mắt và đảm bảo đáp ứng được khả năng phục vụ khách hàng. Để tránh gặp lỗi về giao diện cũng như hạn chế tối đa kích thước tệp javascript em đã không sử dụng các thư viện hỗ trợ UI để thực thi mà tự viết css. Về mặt chức năng thì khách hàng có thể gửi đa dạng các loại tin nhắn như tin nhắn văn bản, tin nhắn hình ảnh, tin nhắn âm thanh, tin nhắn video, ngoài ra hệ thống có thể gửi tin nhắn lựa chọn để định hướng câu hỏi cho khách hàng, tất cả các tin nhắn gửi đi và nhận về đều được đưa về dạng chuẩn để hiển thị một cách tốt nhất. Cửa sổ livechat cũng được kết nối websocket để việc kết nối luôn được duy trì.

Kết quả đạt được từ việc xây dựng cửa sổ livechat là tạo ra một cửa sổ với giao diện bắt mắt, dễ dàng tích hợp vào các website, hỗ trợ nhắn tin các loại tin nhắn cơ bản một cách nhanh chóng như **Hình 27** đã trình bày trong Chương 4.

Trên đây em đã trình bày những nội dung đóng góp mà bản thân em thấy tâm đắc nhất trong quá trình thực hiện đồ án tốt nghiệp. Đó là những khó khăn và cũng là kinh nghiệm em học hỏi, tích lũy được cho bản thân mình. Từ quá trình học hỏi từ bạn bè, các anh chị trong phòng lab và cô Nguyễn Thị Thu Trang mà mọi kỹ năng từ kỹ năng giao tiếp, phân tích thiết kế, lập trình đều được cải thiện đáng kể. Trong chương tiếp theo, em sẽ tổng kết lại nội dung của đồ án và đưa ra hướng phát triển cho hệ thống.

Chương 6 Kết luận và hướng phát triển

Trong chương này em sẽ trình bày kết quả cuối cùng của đồ án và hướng phát triển sau này cho sản phẩm.

6.1 Kết luận

Tổng kết lại, em đã trình bày về hệ thống quản lý nhắn tin đa kênh và cửa sổ livechat trên nền tảng web. Nêu ra các mô hình, kiến trúc và công nghệ sử dụng để phát triển hệ thống. Đồng thời hệ thống cũng đã được thiết kế để có thể linh hoạt sửa đổi, mở rộng. Và hiện tại, hệ thống quản lý nhắn tin đã được đưa vào thử nghiệm trong các lĩnh vực ngân hàng, tài chính và viễn thông với tên miền <https://dashboard.iristech.club>, cửa sổ livechat cũng đã được triển khai với tên miền <https://livechat.iristech.club>.

Trong quá trình phát triển hệ thống, em đã được tiếp xúc với những vấn đề liên quan đến việc mở rộng khi số lượng người dùng tăng lên, sự không thống nhất dữ liệu gọi API giữa các kênh nhắn tin bên thứ ba, khó khăn khi lập lịch gửi tin nhắn đến nhiều đối tượng khách hàng. Để giải quyết được những vấn đề trên, em đã thiết kế hệ thống theo kiến trúc microservices với nhiều dịch vụ con hoạt động độc lập, ít liên quan đến nhau. Em cũng đưa ra một mô hình nhắn tin hàng loạt đa kênh, khi thêm một kênh nhắn tin mới chỉ cần xử lý gửi tin nhắn hàng loạt theo kênh đó chứ không cần thực hiện cả một quy trình riêng cho kênh mới này. Bên cạnh đó, em đã học được cách nhúng một thành phần vào trang web bằng cách sử dụng javascript.

Ngoài việc học hỏi, trau dồi thêm về mặt kiến thức, em cũng đã học được một kỹ năng rất quan trọng đó là kỹ năng làm việc nhóm, đặc biệt là kỹ năng làm việc nhóm theo quy trình Scrum với phương pháp phát triển phần mềm Agile phổ biến trong ngành công nghệ thông tin. Với quy trình này, việc phát triển hệ thống được thực hiện một cách minh bạch, sau hai tuần (một Sprint) sẽ có một cuộc họp để tổng kết và đưa ra ý kiến của bản thân và lên kế hoạch cho sprint tiếp theo. Đồng thời, hằng ngày cũng có những cuộc họp ngắn để trình bày về những công việc đã và dự định làm, những khó khăn gặp phải để kịp thời giải quyết.

Tuy vậy, hệ thống được phát triển là chưa hoàn hảo, vẫn còn một số nhược điểm như chưa đạt độ ổn định cao, đôi khi vẫn xuất hiện các lỗi, một số chức năng so với sản phẩm hiện có trên thị trường chưa thực sự nổi bật. Ngoài ra, do trình độ của bản thân còn hạn chế nên một số đoạn code vẫn chưa được tối ưu làm ảnh hưởng đến hiệu năng. Hệ thống hiện tại vẫn

đang trong quá trình xây dựng và hoàn thiện, trong thời gian tới sẽ được nâng cấp, hoàn thiện và bàn giao đưa sản phẩm đến với người dùng cuối.

6.2 Hướng phát triển

Sản phẩm vẫn đang được phát triển và cải tiến trong tương lai. Em dự định sẽ cải tiến code để đạt được hiệu năng tốt hơn, nghiên cứu thêm về mô hình nhắn tin hàng loạt đa kênh để đưa ra được một mô hình tối ưu nhất. Hệ thống quản lý nhắn tin sẽ có thêm chức năng xuất báo cáo thống kê, giao diện cửa sổ livechat có thể được tùy chỉnh trên hệ thống quản lý sao cho phù hợp với giao diện, phong cách của từng doanh nghiệp. Đồng thời chức năng quảng cáo sẽ được cải tiến để nhắm đến tập người nhận chi tiết hơn.

Tài liệu tham khảo

- [1] Kirupa Chinnathambi, *Learning React: A hands-on guide to building web applications using React and Redux*, tái bản lần 2, Addison-Wesley, 2018
- [2] Kasun Indrasiri và Prabath Siriwardena, *Microservices for the Enterprise: Designing, Developing, and Deploying*, tái bản lần 2, Apress, 2018
- [3] Chris Richardson, *Microservices Patterns*, tái bản lần 1, Manning Publications, 2018
- [4] ReactJS, <https://reactjs.org>, truy cập lần cuối 24/06/2020
- [5] NodeJS, <https://nodejs.org>, truy cập lần cuối 24/06/2020
- [6] Websocket, https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API, lần truy cập cuối 24/06/2020
- [7] Material UI, <https://material-ui.com>, lần truy cập cuối 24/06/2020
- [8] Styled-Components, <https://styled-components.com>, lần truy cập cuối 24/06/2020
- [9] Facebook API, <https://developers.facebook.com/docs/graph-api/making-multiple-requests>, truy cập lần cuối 24/06/2020
- [10] Zalo API, <https://developers.zalo.me/docs/api/official-account-api-147>, truy cập lần cuối 24/06/2020
- [11] Viber API, <https://developers.viber.com/docs/api/rest-bot-api>, truy cập lần cuối 24/06/2020
- [12] LiveChat, <https://www.livechat.com>, truy cập lần cuối 24/06/2020
- [13] tawk.to, <https://www.tawk.to>, truy cập lần cuối 24/06/2020
- [14] vChat, <https://vchat.vn>, truy cập lần cuối 24/06/2020

Phụ lục

Phần này em sẽ đặc tả các use case Chia sẻ ứng dụng quản lý nhắn tin đa kênh, Tích hợp Facebook Messenger, Tích hợp Viber, Thống kê số lượng khách hàng theo kênh nhắn tiếp nói mục 2.3. Và trình bày kiểm thử cho hai chức năng Tạo nội dung chiến dịch quảng cáo và Gửi tin nhắn video.

A Đặc tả use case

A.1 Đặc tả use case Chia sẻ ứng dụng quản lý nhắn tin đa kênh

Bảng 17 Đặc tả use case Chia sẻ ứng dụng quản lý nhắn tin đa kênh

Mã use case	UC003	Tên use case	Chia sẻ ứng dụng quản lý nhắn tin đa kênh
Tác nhân	Người sở hữu		
Tiền điều kiện	Người dùng đăng nhập vào hệ thống quản lý nhắn tin đa kênh với vai trò là Người sở hữu.		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Nhập email người muốn chia sẻ
	2.	Hệ thống	Tìm kiếm tài khoản tương ứng với email
	3.	Người dùng	Chọn loại quyền muốn chia sẻ
	4.	Người dùng	Yêu cầu chia sẻ
	5.	Hệ thống	Cập nhật cơ sở dữ liệu và thông báo chia sẻ thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động

	5a.	Hệ thống	Thông báo chia sẻ thất bại
Hậu điều kiện	Không		

A.2 Đặc tả use case Tích hợp Facebook Messenger

Bảng 18 Đặc tả use case Tích hợp Facebook Messenger

Mã use case	UC010	Tên use case	Tích hợp Facebook Messenger
Tác nhân	Người quản lý, Người sở hữu		
Tiền điều kiện	Người dùng đăng nhập vào hệ thống quản lý nhắn tin đa kênh với vai trò là Người quản lý hoặc Người sở hữu.		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu tích hợp Facebook
	2.	Hệ thống	Điều hướng đến trang chọn thông tin của Facebook
	3.	Người dùng	Chọn ứng dụng Facebook muốn tích hợp
	4.	Hệ thống	Thông báo tích hợp thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a.	Hệ thống	Thông báo tích hợp thất bại
Hậu điều kiện	Không		

A.3 Đặc tả use case Tích hợp Viber

Bảng 19 Đặc tả use case Tích hợp Viber

Mã use case	UC011	Tên use case	Tích hợp Viber
--------------------	-------	---------------------	----------------

Tác nhân	Người quản lý, Người sở hữu		
Tiền điều kiện	Người dùng đăng nhập vào hệ thống quản lý nhắn tin đa kênh với vai trò là Người quản lý hoặc Người sở hữu. Đã có tài khoản bot của Viber		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Yêu cầu tích hợp Viber và nhập mã xác thực tài khoản bot của Viber
	2.	Hệ thống	Kiểm tra trường bắt buộc nhập
	3.	Hệ thống	Lưu cơ sở dữ liệu và thông báo tích hợp thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a.	Hệ thống	Thông báo lỗi chưa nhập thông tin
	3a.	Hệ thống	Thông báo tích hợp thất bại
Hậu điều kiện	Không		

A.4 Đặc tả use case Thống kê số lượng khách hàng theo kênh nhắn tin

Bảng 20 Đặc tả use case Thống kê số lượng khách hàng theo kênh nhắn tin

Mã use case	UC012	Tên use case	Thống kê số lượng khách hàng theo kênh nhắn tin
Tác nhân	Người quản lý, Người sở hữu		
Tiền điều kiện	Người dùng đăng nhập vào hệ thống quản lý nhắn tin đa kênh với vai trò là Người quản lý hoặc Người sở hữu của ứng dụng.		

Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Chọn khoảng thời gian thống kê
	2.	Hệ thống	Hiển thị thông tin thống kê trong khoảng thời gian người dùng chọn (mô tả dưới *)
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	1a.	Người dùng	Chọn thống kê trong ngày hôm nay
	1b.	Người dùng	Chọn thống kê trong vòng 1 tuần
	1c.	Người dùng	Chọn thống kê trong vòng 1 tháng
Hậu điều kiện	Không		

(*) Dữ liệu đầu ra khi hiển thị kết quả thống kê

STT	Trường dữ liệu	Mô tả
1.	NU	Tổng số lượng khách hàng của hệ thống
2.	Anonymous	Số khách hàng nhắn tin qua cửa sổ livechat
3.	Facebook	Số khách hàng nhắn tin qua Facebook
4.	Zalo	Số khách hàng nhắn tin qua Zalo
5.	Viber	Số khách hàng nhắn tin qua Viber
6	Biểu đồ	Biểu đồ chi tiết thống kê số lượng khách hàng nhắn tin trên các kênh theo từng ngày

B Kiểm thử

B.1 Kiểm thử chức năng Tạo nội dung chiến dịch quảng cáo

Bảng 21 trình bày các trường hợp kiểm thử cho chức năng Tạo nội dung chiến dịch quảng cáo với các dạng tin nhắn văn bản, hình ảnh, âm thanh, video và catalog

Bảng 21 Các trường hợp kiểm thử cho “Tạo nội dung chiến dịch quảng cáo”

ID	Dữ liệu kiểm thử	Quy trình kiểm thử	Kết quả mong muốn	Kết quả thực tế	Trạng thái
1	Tiêu đề = “Quảng cáo 01” Văn bản = “Test”	1. Nhấn button “Tạo chiến dịch mới” 2. Nhập tiêu đề 3. Nhập tin nhắn văn bản 4. Nhấn “Lưu nháp”	Hiển thị danh sách nội dung quảng cáo có chứa nội dung mới tạo	Hiển thị đúng danh sách cho chứa nội dung mới tạo	Pass
2	Tiêu đề = “” Văn bản = “Test”	1. Nhấn button “Tạo chiến dịch mới” 2. Bỏ trống tiêu đề 3. Nhập tin nhắn văn bản 4. Nhấn “Lưu nháp”	Thông báo “Tiêu đề là bắt buộc”	Thông báo “Tiêu đề là bắt buộc”	Pass
3	Tiêu đề = “Quảng cáo 03” Văn bản = “”	1. Nhấn button “Tạo chiến dịch mới” 2. Nhập tiêu đề 3. Bỏ trống tin nhắn văn bản 4. Nhấn “Lưu nháp”	Thông báo “Nhập ít nhất một nội dung tin nhắn”	Thông báo “Nhập ít nhất một nội dung tin nhắn”	Pass

ID	Dữ liệu kiểm thử	Quy trình kiểm thử	Kết quả mong muốn	Kết quả thực tế	Trạng thái
4	Tiêu đề = “Quảng cáo 04” Hình ảnh = tệp hình ảnh	1. Nhấn button “Tạo chiến dịch mới” 2. Nhập tiêu đề 3. Nhấn button “Hình ảnh” 4. Nhấn vào khung chọn hình ảnh để chọn hình ảnh từ thiết bị 5. Nhấn “Lưu nháp”	Hiển thị hình ảnh khi chọn từ thiết bị. Hiển thị danh sách nội dung quảng cáo có chứa nội dung mới tạo khi nhấn “Lưu nháp”	Hiển thị hình ảnh khi chọn từ thiết bị. Hiển thị danh sách nội dung quảng cáo có chứa nội dung mới tạo khi nhấn “Lưu nháp”	Pass
5	Tiêu đề = “Quảng cáo 05” Âm thanh = tệp âm thanh	1. Nhấn button “Tạo chiến dịch mới” 2. Nhập tiêu đề 3. Nhấn button “Âm thanh” 4. Nhấn button “Tải lên” để chọn tệp âm thanh từ thiết bị 5. Nhấn “Lưu nháp”	Hiển thị đường dẫn đến tệp âm thanh sau khi chọn từ thiết bị. Hiển thị danh sách nội dung quảng cáo có chứa nội dung mới tạo khi nhấn “Lưu nháp”	Hiển thị đường dẫn đến tệp âm thanh sau khi chọn từ thiết bị. Hiển thị danh sách nội dung quảng cáo có chứa nội dung mới tạo khi nhấn “Lưu nháp”	Pass
6	Tiêu đề = “Quảng cáo 06” Video = tệp video	1. Nhấn button “Tạo chiến dịch mới” 2. Nhập tiêu đề	Hiển thị đường dẫn đến tệp video sau khi chọn từ thiết bị.	Hiển thị đường dẫn đến tệp video sau khi chọn từ thiết bị.	Pass

ID	Dữ liệu kiểm thử	Quy trình kiểm thử	Kết quả mong muốn	Kết quả thực tế	Trạng thái
		3. Nhấn button “Video” 4. Nhấn button “Tải lên” để chọn tệp video từ thiết bị 5. Nhấn “Lưu nháp”	Hiện thị danh sách nội dung quảng cáo có chứa nội dung mới tạo khi nhấn “Lưu nháp”	Hiện thị danh sách nội dung quảng cáo có chứa nội dung mới tạo khi nhấn “Lưu nháp”	
7	Tiêu đề = “Quảng cáo 06” Hình ảnh = tệp hình ảnh	1. Nhấn button “Tạo chiến dịch mới” 2. Nhập tiêu đề 3. Nhấn button “Catalogue” 4. Nhấn vào khung chọn hình ảnh để chọn hình ảnh từ thiết bị 5. Lặp lại bước 2 và 4 6. Nhấn “Lưu nháp”	Hiện thị hình ảnh mỗi khi chọn từ thiết bị. Hiện thị danh sách nội dung quảng cáo có chứa nội dung mới tạo khi nhấn “Lưu nháp”	Hiện thị hình ảnh mỗi khi chọn từ thiết bị. Hiện thị danh sách nội dung quảng cáo có chứa nội dung mới tạo khi nhấn “Lưu nháp”	Pass

B.2 Kiểm thử chức năng Gửi tin nhắn video

Bảng 22 trình bày các trường hợp kiểm thử cho chức năng Gửi tin nhắn video trên cửa sổ livechat.

Bảng 22 Các trường hợp kiểm thử cho “Gửi tin nhắn video”

ID	Dữ liệu kiểm thử	Quy trình kiểm thử	Kết quả mong muốn	Kết quả thực tế	Trạng thái
1	Video = tệp video có kích thước nhỏ hơn 50MB	1. Nhấn biểu tượng chọn tệp tin 2. Chọn tệp video	Hiển thị video lên cửa sổ livechat	Hiển thị video trên cửa sổ livechat	Pass
2	Video = tệp video có kích thước lớn hơn 50MB	1. Nhấn biểu tượng chọn tệp tin 2. Chọn tệp video	Hiển thị thông báo “Đã xảy ra lỗi. Vui lòng thử lại.”	Hiển thị thông báo “Đã xảy ra lỗi. Vui lòng thử lại.”	Pass