TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Hệ thống xây dựng và quản lý chatbot theo luật

TRƯƠNG HỮU TIẾN
Tien.th173398@sis.hust.edu.vn

Ngành Công nghệ thông tin

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

Bộ môn: Công nghệ phần mềm
Viện: Công nghệ thông tin – Truyền thông

HÀ NỘI, 6/2021

Lời cam kết

Họ và tên sinh viên: Trương Hữu Tiến

Điện thoại liên lạc: (+84) 378 935 817 Email: tien.th173398@sis.hust.edu.vn

Lớp: CNTT 10 – K62 Hệ đào tạo: Đại học chính quy

Tôi – *Trương Hữu Tiến* – 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 11 tháng 06 năm 2021

Tác giả ĐATN

Trương Hữu Tiến

Lời cảm ơn

Lời đầu tiên em xin gửi lời cảm ơn chân thành nhất tới TS. Nguyễn Thị Thu Trang. Cô đã luôn nhiệt tình, tận tâm hướng dẫn và giúp đỡ em trong suốt quá trình tham gia làm đồ án tốt nghiệp.

Em cũng xin gửi lời cảm ơn tới những người anh chị, bạn bè, thành viên trên lab 914, đặc biệt là anh Phạm Quang Minh đã luôn trao đổi, góp ý, chia sẻ kinh nghiệm làm việc với em trong suốt thời gian qua.

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

Trong quá trình xây dựng, phát triển và hoàn thiện báo cáo cũng như đồ án tốt nghiệp, em khó có thể tránh khỏi những sai sót. Em rất mong thầy cô và các bạn đọc góp ý để em có thể hoàn thiện hơn nữa sản phẩm này.

Một lần nữa em xin trân thành cảm ơn !

Tóm tắt

Ngày nay, với sự phổ biến của các ứng dụng nhắn tin thì nhu cầu tư vấn thông tin, đặt hàng online giữa khách hàng và các đơn vị, tổ chức, doanh nghiệp thông qua tin nhắn đang ngày càng gia tăng. Những công việc này tưởng chừng rất đơn giản nhưng với số lượng tương tác khổng lồ thì công việc này đang khiến các tổ chức, doanh nghiệp tốn rất nhiều thời gian, chi phí và nguồn nhân lực để giải quyết. Chatbot đang là giải pháp hàng đầu cho vấn đề này, việc sử dụng chatbot sẽ hỗ trợ người dùng tự động tương tác với khách hàng, giúp tối ưu hoá lợi ích của các tổ chức, doanh nghiệp. Có rất nhiều nghiệp vụ đơn gian cần sử dụng chatbot như tư vấn một mặt hàng nào đó, đặt hàng online, tư vấn tuyển sinh. Vấn đề đặt ra là làm sao có thể nhanh chóng xây dựng được các bot đơn giản, dễ dàng mà không cần tạo nhiều dữ liệu cho bot. Xuất phát từ hiện trạng trên, đồ án tốt nghiệp này đã phát triển hệ thống xây dựng và quản lý chatbot theo luật với các công cụ trực quan, giúp nhanh chóng tạo bot. Hệ thống này không đòi hỏi người dùng cung cấp nhiều dữ liệu nhưng bot có thể hiểu và đáp trả người dùng với những yêu cầu cơ bản.

Khi nhận được một tin nhắn, hệ thống chatbot cần xác định được ý định khách hàng và trích xuất các thông tin cần thiết, từ đó đưa ra các phản hồi thích hợp, để có thể thực hiện được quy trình đó hệ thống cần có các tập dữ liệu có sẵn. Vì vậy, đồ án đề xuất giải pháp cần quản lý các thông tin về ý định, thực thể, các phản hồi của bot và kịch bản hội thoại. Để xác định được ý muốn của khách hàng trong câu nói có nằm trong tập câu đầu vào đã được xây dựng, hệ thống sử dụng Elasticsearch để hỗ trợ tìm kiếm, đánh giá và cho ra kết quả tương đồng nhất với mong muốn của người dùng. Công cụ quản lý kịch bản đóng vai trò rất quan trọng trong hệ thống chatbot, vì vậy hệ thống sẽ sử dụng thư viện React-Diagram để hỗ trợ xây dựng giao diện và thiết kế công cụ quản lý kịch bản dưới dạng biểu đồ luồng, giúp người dùng trực quan hoá giao diện và thiết kế được kịch bản theo mong muốn.

Hiện tại hệ thống xây dựng và quản lý chatbot đã được triển khai và kiểm thử tại https://rbc-bot.iristech.club. Hệ thống vẫn đang trong quá trình thử nghiệm, phát triển lắng nghe ý kiến từ người dùng để ngày càng tối ưu hơn.

Mục lục

Danh mục hình vẽ

Danh mục bảng

Danh mục các từ viết tắt

API	Application Programming Interface Giao diện lập trình ứng dụng
HTML	HyperText Markup Language Ngôn ngữ đánh dấu siêu văn bản
CNTT	Công nghệ thông tin
ĐATN	Đồ án tốt nghiệp
SV	Sinh viên
DFD	Data Flow Diagram

Danh mục thuật ngữ

Browser	Trình duyệt
Client	Máy khách
Server	Máy chủ
Intent	Ý định của câu đầu vào
Entity	Thực thể
Action	Hành động của chatbot
DOM	Mô hình các đối tượng trong tài liệu HTML
User	Người dùng

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 xây dựng chatbot theo luật. 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

Hiện nay với sự phát triển mạnh mẽ các ứng dụng nhắn tin đã giúp cho sự tương tác giữa các đơn vị, tổ chức, doanh nghiệp và khách hàng của mình ngày càng thuận lợi. Tuy nhiên một số lượng lớn tin nhắn từ khách hàng có xu hướng lặp lại đã khiến cho các đơn vị, tổ chức, doanh nghiệp phải tốn rất nhiều chi phí, thời gian và nguồn nhân lực để phản hồi các tin nhắn này.

Chẳng hạn như trong năm 2020, Trường Đại học Bách Khoa Hà Nội đã thay đổi phương thức tuyển sinh. Tuy đã tổ chức những chương trình giới thiệu về hình thức tuyển sinh mới, cũng như cập nhật thông tin trên các trang web, nhà trường vẫn không thể tránh khỏi hàng ngàn những câu hỏi giống nhau trong đợt tuyển sinh này. Đây không phải vấn đề riêng với Trường Đại học Bách Khoa Hà Nội, các phòng tuyển sinh nói chung, các đơn vị, tổ chức, chủ cửa hàng bán hàng online hay các trang web cũng gặp khó khăn khi xử lý lượng tương tác không lồ này. Với sự phát triển của các ứng dụng nhắn tin, chatbot thực sự là cách đơn giản nhất để giải quyết vấn đề, đặc biệt trong bối cảnh bị ảnh hưởng bởi đại dịch Covid, nhu cầu tìm kiếm thông tin online ngày một tăng lên. Chatbot là một chương trình được lập trình để tương tác với con người, là công cụ thay thế cho nhân viên để tư vấn trả lời cho những gì khách hàng thắc mắc.

Ngày nay, theo hướng tiếp cận chatbot sẽ có 2 loại: (i) chatbot sử dụng các thuật toán học máy, học sâu (AI chatbot) và chatbot dựa vào các quy luật (Rule-based chatbot). AI chatbot là chatbot được trang bị thêm thêm công nghệ trí tuệ nhân tạo (AI), giúp bot có khả năng tự học từ những cuộc trò chuyện thực tế với người dùng. Chatbot loại này thường hướng tới xử lý các nghiệp vụ phức tạp, phù hợp với doanh nghiệp lớn như ngân hàng, tài chính. Ngoài ra, chatbot cần một lượng lớn dữ liệu huấn luyện, đòi hỏi thời gian công sức khá lớn của người xây dựng bot. Chatbot theo luật trả lời người dùng theo luật, thường được xây dựng dựa trên ngôn ngữ sử dụng hàng ngày giữa khách hàng và các tổ chức, doanh nghiệp. Chatbot loại này không cần quá nhiều dữ liệu nên rất dễ để xây dựng, phù hợp để xử lý các nghiệp

vụ đơn đơn giản, thường là phục vụ một nhiệm vụ cụ thể như tư vấn tuyển sinh, tư vấn bán hàng.

Vì vậy, trong khuôn khổ đồ án, em sẽ phát triển hệ thống xây dựng và quản lý chatbot theo luật để giải quyết vấn đề nêu trên. Hệ thống này không đòi hỏi người dùng cung cấp nhiều dữ liệu nhưng bot có thể hiểu và đáp trả người dùng với những yêu cầu cơ bản.

1.2 Mục tiêu và phạm vi đề tài

Với vấn đề đã trình bày ở mục 1.1, mục tiêu của đồ án là phát triển hệ thống xây dựng và quản lý Chatbot theo luật, hệ thống sẽ giúp người dùng tạo ra một chatbot hỗ trợ trả lời tin nhắn tự động tới khách hàng khi họ tìm kiếm thông tin. Hệ thống có thể tích hợp trên các nền tảng nhắn tin phổ biến hiện nay như Facebook Messenger, Zalo giúp người dùng có thể tiếp cận, cung cấp thông tin, chăm sóc khách hàng trên các nền tảng này.

Để đảm bảo tính đơn giản, dễ giúp người dùng dễ xây dựng mà không cần quá nhiều dữ liệu, hỗ trợ hiệu quả với các bài toán nghiệp vụ đơn giản cũng như chăm sóc khách hàng, chatbot được xây dựng sẽ sử dụng loại chatbot theo luật, với tập câu trả lời được xây dựng từ trước phù hợp nhất với ý định đầu vào.

Đối tượng của đồ án là chủ cửa hàng bán hàng online, các doanh nghiệp có nhu cầu sử dụng để giải quyết các nghiệp vụ đơn giản. Đối tượng này chủ yếu không rành về công nghệ, muốn giảm tải thời gian và chi phí để giải đáp nhu cầu thông tin của khách hàng. Vì vậy hệ thống xây dựng chatbot theo luật cần có giao diện trực quan, dễ dùng, hỗ trợ người dùng phản hồi, cung cấp thông tin cho khách hàng.

Từ đây, khi nói đến người dùng, ta sẽ hiểu là người sử dụng hệ thống để xây dựng chatbot. Khi nói đến khách hàng, ta nói tới người tương tác với chatbot đó, là khách hàng của người dùng.

1.3 Định hướng giải pháp

Để hoàn thành mục tiêu như trên, em đề xuất một số giải pháp. Đầu tiên, khi nhận được một tin nhắn, hệ thống sẽ xác định ý định, trích xuất các thông tin và đưa ra phản hồi phù hợp, để có thể thực hiện được quy trình trên hệ thống cần phải có sẵn dữ liệu. Vì vậy, đồ án đề xuất cần quản lý các thông tin về ý định, thực thể để trích xuất thông tin, các phản hồi của bot tới khách hàng và các kịch bản hội thoại của bot.

Để đáp ứng nhu cầu xây dựng và quản lý chatbot em sẽ phát triển hệ thống trên nền tảng web, theo mô hình client-server. Hệ thống sẽ bao gồm hai phần chính chúng giao tiếp với nhau qua RESTful API là backend và frontend.

Về phía backend, để dễ dàng mở rộng, phát triển và tích hợp với các dịch vụ của bên thứ ba, em sẽ sử dụng kiến trúc Microservices [1]. Đây là một kiến trúc chia ứng dụng thành nhiều dịch vụ (service) nhỏ chạy độc lập, chúng tách biệt nhau về mặt mã nguồn, về hoạt động và dữ liệu. Hệ thống chatbot sẽ chỉ có một dịch vụ, dịch vụ này sẽ được viết bằng ngôn ngữ Javascript dựa trên nền tảng NodeJS [3]. Về việc lưu trữ dữ liệu, hệ thống sẽ sử dụng hệ quản trị cơ sở dữ liệu MongoDB [4] vì đây là một kiểu cơ sở dữ liệu phi quan hệ có cấu trúc linh hoạt, tốc độ đọc ghi nhanh rất phù hợp với hệ thống nhắn tin cần cập nhật dữ liệu liên tục. Ngoài ra, em cũng sử dụng thêm công nghệ của Redis [7], cũng là một công nghệ lưu trữ dữ liệu để có thể tăng tốc độ cho hệ thống và quản lý phiên người dùng truy cập. Để dễ dàng giao tiếp và tích hợp với các dịch vụ khác, hệ thống chatbot sẽ sử dụng RabbitMQ [5], đây là một công nghệ giúp các dịch vụ giao tiếp với nhau qua hàng đợi tin nhắn. Elasticsearch [6] sẽ là một giải pháp rất hữu ích để hỗ trợ truy vấn tìm kiếm được ý định chính xác nhất của khách hàng.

Về phía frontend, em sẽ phát triển trang web theo hướng Single Page Application, làm tăng trải nghiệm người dùng, hiệu năng của trang web. Toàn bộ mã nguồn của web sẽ được tải lên lần đầu tiên sau khi chúng ra truy cập website. Ở những lần sau, khi chuyển trang khác client sẽ gửi những yêu cầu để lấy những dữ liệu cần thiết, việc này đem đến cho người dùng web tốt hơn, giảm thời gian tải lại toàn bộ trang web cồng kềnh, tiết kiệm băng thông cũng như thời gian chờ đợi. Để xây dựng một trang web như vậy em sử dụng ReactJS [8] vì công cụ này chia trang web thành các thành phần con dễ dàng cho tái sử dụng và mở rộng. Với yêu cầu xây dựng công cụ quản lý kịch bản nhắn tin, em sử dụng thư viện React-Diagram [9], thư viện này cung cấp rất nhiều các phương thức, các API để thuận tiện xây dựng biểu đồ luồng giúp trực quan hoá luồng kịch bản nhắn tin mà người dùng tạo.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp, bố cục các phần được trình bày như sau.

Trong Chương 2, em sẽ trình bày tổng quan các chức năng trong hệ thống và đặc tả một số chức năng chính.

Từ các yêu cầu và chức năng tổng quan đã đưa ra, trong Chương 3 em sẽ trình bày về các công nghệ sử dụng chính trong việc xây dựng và phát triển hệ thống.

Trong Chương 4, em sẽ trình bày chi tiết về 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 xây dựng và phát triển ĐATN như giải pháp xác định ý định người dùng với Elasticsearch, thiết kế giao diện kịch bản luồng hội thoại trực quan.

Trong Chương 6, em sẽ tổng kết nội dung chính, những gì đã làm được trong trong đồ án và hướng phát triển của hệ thống trong tương lai.

Chương 2 Khảo sát và phân tích yêu cầu

Trong Chương 2 em sẽ trình bày khảo sát các ứng chatbot trên thị trường hiện nay. Từ đó đưa ra các chức năng tổng quan và đi vào phân tích, làm rõ các chức năng đó.

2.1 Khảo sát hiện trạng

Trên thị trường hiện nay đã có nhiều hệ thống hỗ trợ xây dựng chatbot theo luật. Trong đó, Manychat [10], Chatfuel [11] là các ứng dụng nổi bật về hướng tiếp cận này, sau đây em sẽ phân tích ưu nhược điểm của các ứng dụng trên.

Bảng 1 so sánh ứng dụng Manychat và Chatfuel

Tiêu chí	**Manychat**	**Chatfuel**
Giá cả	Miễn phí và trả phí	Miễn phí và trả phí
Ngôn ngữ	Tiếng Anh	Tiếng Anh
Thị trường	Đa quốc gia	Đa quốc gia
Tích hợp nền tảng nhắn tin	Facebook Messenger	Facebook Messenger, Instagram, Twitter, …
Thống kê	Có	Có
Công cụ quản lý kịch bản	Có	Có

Manychat là một nền tảng giúp người dùng dễ dàng tạo chatbot cho riêng mình trên facebook. Ứng dụng tập trung vào các doanh nghiệp vừa và nhỏ, giúp họ tận dụng các ứng dụng nhắn tin phổ biến để thu hút khách hàng. Cách hoạt động của Manychat là dùng những từ khóa định nghĩa sẵn để điều hướng khách hàng theo kịch bản đã thiết lập sẵn. Ngoài ra nó còn giúp người dùng phản hồi cho khách hàng bằng tin nhắn dưới dạng văn bản, hình ảnh từ đó giúp khách hàng nhanh chóng tiếp nhận thông tin. Manychat cung cấp các chức năng như: tích hợp với nhiều tảng khác, quản lý thiết lập kịch bản, quản lý nội dung trò chuyện,

quản lý các từ khoá để điều hướng kịch bản, xây dựng chiến dịch chăm sóc khách hàng và còn nhiều hơn nữa. Mặc dù rất nhiều chức năng nhưng Manychat không hỗ trợ tiếng Việt nên sẽ gây ra một chút khó khăn với những người dùng có vốn tiếng Anh không tốt.

Chatfuel là một công cụ mạnh mẽ và hữu ích để đơn giản hoá việc tạo chatbot cho người dùng. Chatfuel cung cấp khá nhiều tính năng của một ứng dụng chatbot hiện nay như cài đặt kết nối fanpage, thêm người quản lý, thiết lập và xây dựng tình huống theo kịch bản, tích hợp được với nhiều nền tảng khác, đo lường cường độ hoạt động của chatbot và còn nhiều nữa những tính năng hữu ích. Trong đó, kịch bản của Chatfuel gồm các thành phần chính: ý định người dùng, thực thể, phản hồi của bot và điều kiện đầu vào cùng với giao diện trực quan giúp người dùng có thể tạo luồng kịch bản một cách dễ dàng. Cũng giống như Manychat, ngôn ngữ sử dụng của Chatfuel là tiếng Anh, nhưng nếu bỏ qua vấn đề đó thì đây chắc chắn là một trong những công cụ mạnh nhất để tạo chatbot mà người dùng không thể nào bỏ qua. Đặc biệt là các doanh nghiệp hoặc cá nhân làm kinh doanh online thì việc sử dụng chatbot sẽ cực kì hữu ích và nó hoàn toàn miễn phí.

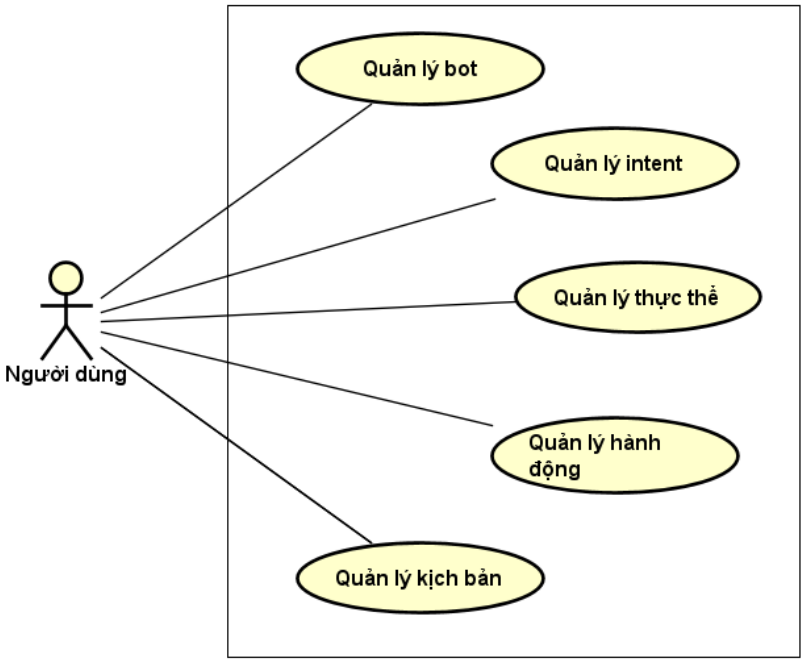
Từ những khảo sát ở trên, em sẽ phát triển hệ thống xây dựng chatbot cung cấp các chức năng như sau: thống kê, tích hợp chatbot, quản lý bot, quản lý tập ý định, quản lý thực thể, quản lý hành động (phản hồi của bot), công cụ quản lý kịch bản.

2.2 Tổng quan chức năng

Phần 2.2 này có nhiệm vụ tóm tắt các chức năng của phần mềm. Trong phần này, sinh viên lưu ý chỉ mô tả chức năng mức cao (tổng quan) mà không đặc tả chi tiết cho từng chức năng. Đặc tả chi tiết được trình bày trong phần 2.3.

2.2.1 Biểu đồ use case tổng quan

Hệ thống xây dựng và quản lý chatbot theo luật gồm nhiều thành phần và chức năng khác nhau. Tổng quan các chức năng của toàn bộ hệ thống chatbot được mô tả trong **Hình 1**.

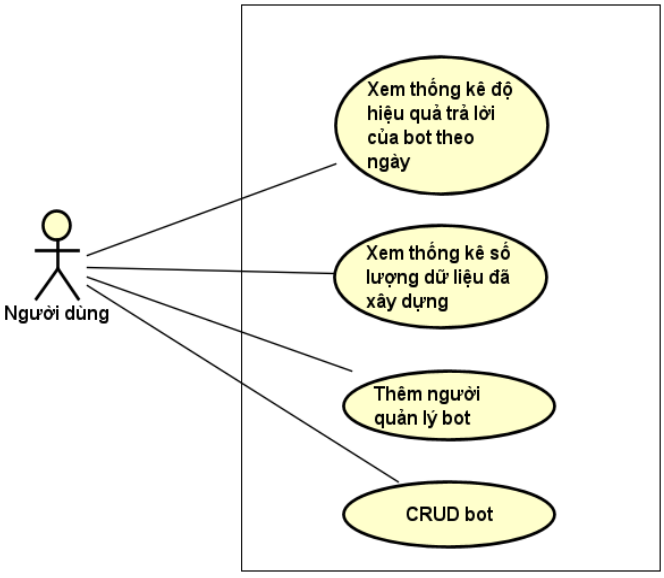


Hình 1 Biểu đồ use case tổng quan hệ thống xây dựng chatbot theo luật.

Hệ thống gồm một tác nhân đó là Người dùng, Người dùng sẽ đóng vai trò là người sử dụng Web để xây dựng và quản lý chatbot.

2.2.2 Biểu đồ use case phân rã

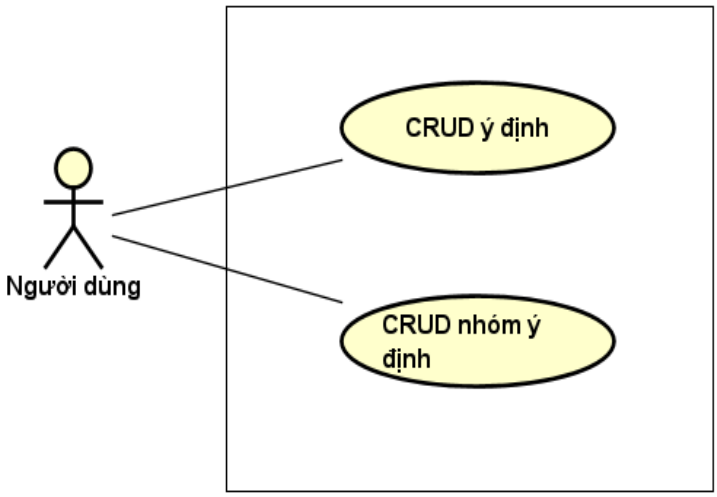
a) Biểu đồ use case phân rã "Quản lý bot"



Hình 2 Biểu đồ use case phân rã "Quản lý bot".

Hình 2 là biểu đồ use case phân rã quản lý bot của người dùng. Người dùng có thể thêm, sửa, xóa, tìm kiếm, thêm người xây dựng và quản lý và xem thống kê hiệu quả trả lời của bot theo ngày, xem số lượng dữ liệu xây dựng trong bot.

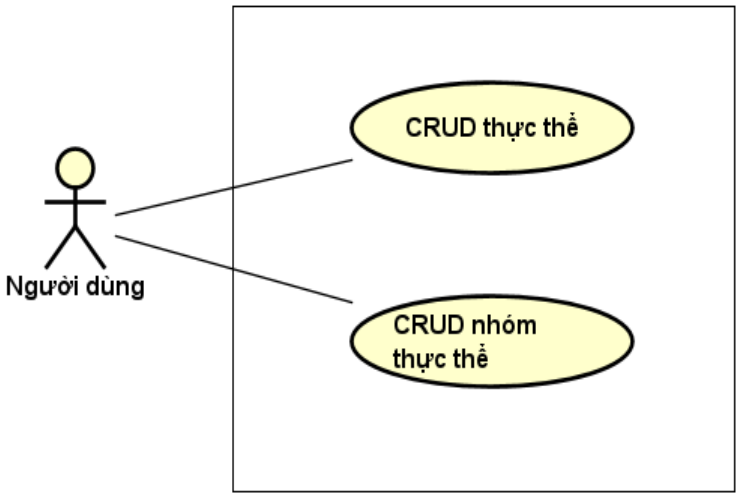
b) Biểu đồ use case phân rã "Quản lý ý định"



Hình 3 Biểu đồ use case phân rã "Quản lý ý định".

Hình 3 Biểu đồ use case phân rã quản lý tập ý định của người dùng. Người dùng có thể xem danh sách, thêm, sửa, xóa, xem chi tiết ý định, và quản lý ý định theo nhóm như gộp các ý định vào một nhóm.

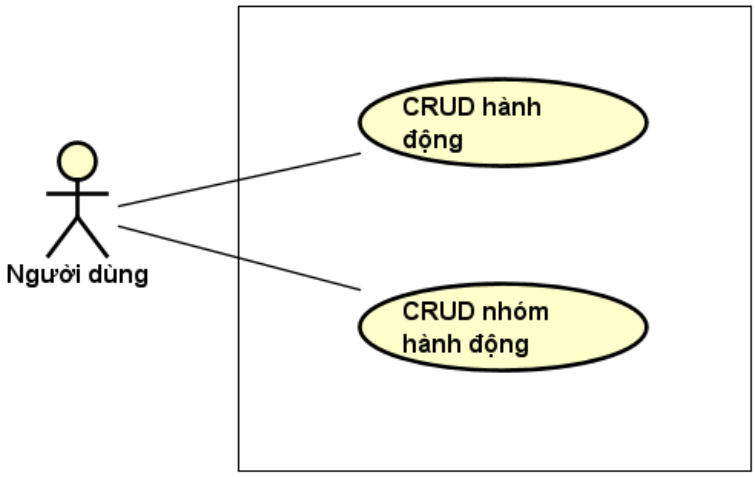
c) Biểu đồ use case phân rã "Quản lý thực thể"



Hình 4 Biểu đồ use case phân rã "Quản lý thực thể".

Hình 4 là biểu đồ use case phân rã quản lý tập thực thể của người dùng. Người dùng có thể xem danh sách thực thể, xem chi tiết, tìm kiếm, thêm, sửa, xóa, thực thể và quản lý thực thể theo nhóm.

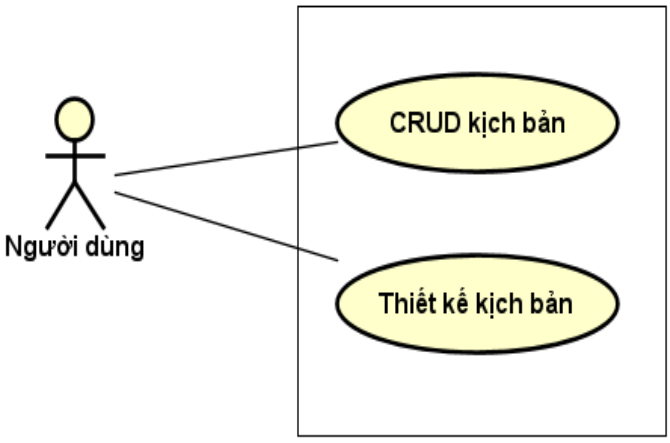
d) Biểu đồ use case phân rã "Quản lý hành động"



Hình 5 Biểu đồ use case phân rã "Quản lý hành động".

Hình 5 là biểu đồ use case phân rã cho chức năng quản lý hành động của người dùng. Người dùng có thể xem danh sách, xem chi tiết, tìm kiếm, thêm, sửa, xóa và quản lý hành động theo nhóm.

e) Biểu đồ use case phân rã "Quản lý kịch bản"

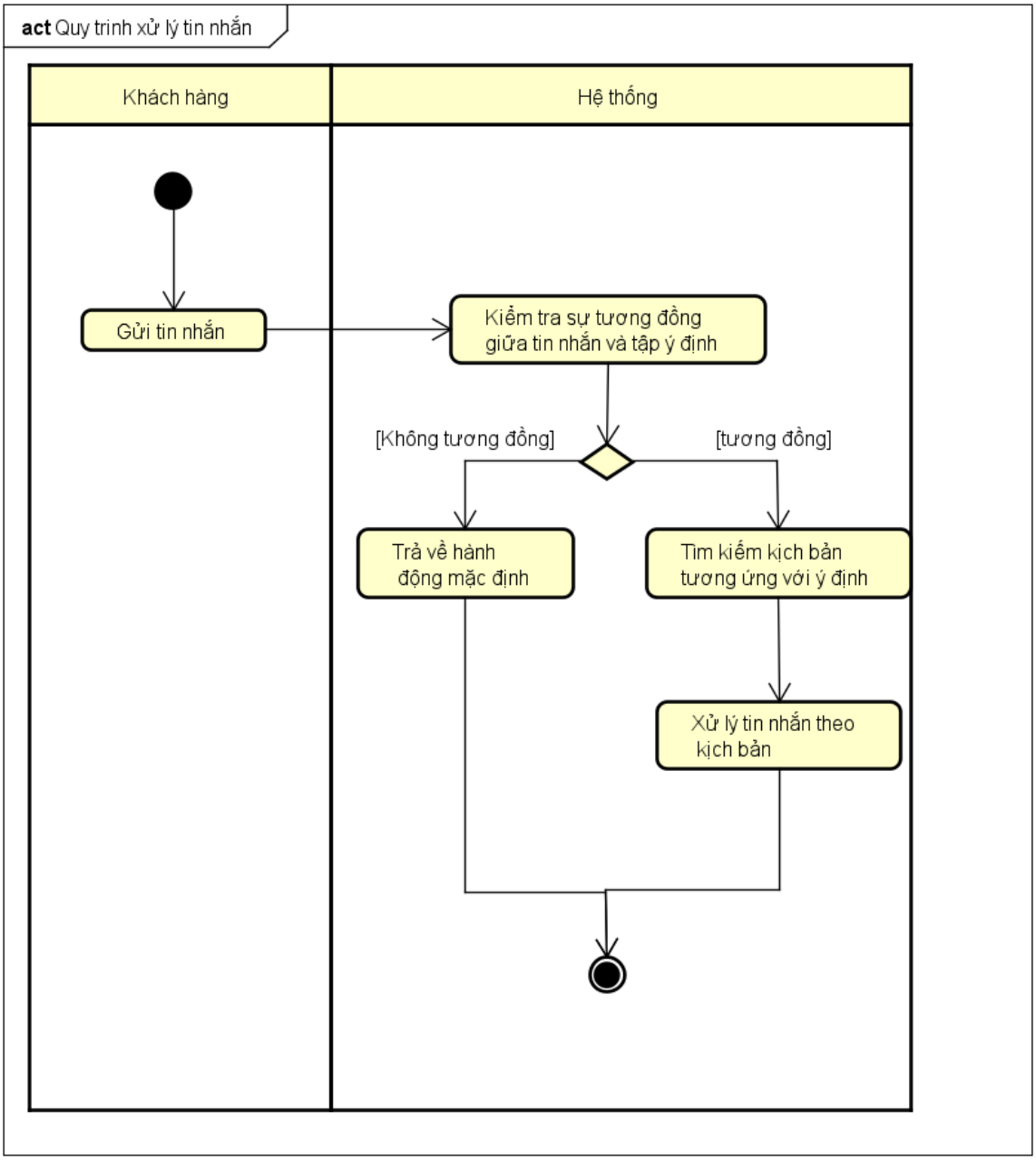


Hình 6 Biểu đồ use case phân rã "Quản lý kịch bản".

Hình 6 là biểu đồ use case cho chức năng quản lý kịch bản của người dùng. Hệ thống cung cấp các chức năng người dùng có thể xem danh sách, xem chi tiết, tìm kiếm, thêm, sửa, xóa kịch bản và thiết kế kịch bản.

2.2.3 Quy trình nghiệp vụ

Quy trình nghiệp vụ xử lý tin nhắn khi khách hàng gửi đến được biểu diễn trong **Hình 7**. Sau khi hệ thống tiếp nhận tin nhắn được gửi đến từ khách hàng, hệ thống kiểm tra nội dung tin nhắn có tương đồng với tập ý định đã được xây dựng từ trước hay không. Nếu không, chatbot sẽ trả về một hành động mặc định. Nếu không bot sẽ trích xuất các tham số (nếu có) từ nội dung tin nhắn của khách hàng và phản hồi lại cho khách hàng theo kịch bản đã xây dựng từ trước.



Hình 7 Quy trình xử lý tin nhắn của khách hàng.

2.3 Đặc tả chức năng

Trong phần này em sẽ mô tả một vài chức năng chính của hệ thống. **Bảng 2** dưới đây sẽ liệt kê 31 use case của hệ thống. Do kích thước báo cáo có hạn nên em sẽ trình bày 3 use case: tạo ý định, tạo hành động, vẽ kịch bản.

Bảng 2 Danh sách các use case

Nhóm use case	**Mã use case**	**Tên use case**
Bot	UC001	Tạo bot
	UC002	Cập nhật thông tin bot
	UC003	Chia sẻ quyền quản lý bot
	UC004	Xóa bot
	UC005	Xem thống kê độ hiệu quả của bot
	UC006	Xem thống kê số lượng dữ liệu của bot
	UC007	Tìm kiếm bot
Ý định	**UC008**	**Tạo ý định**
	UC009	Cập nhật ý định
	UC010	Xoá ý định
	UC011	Tạo nhóm ý định
	UC012	Sửa tên nhóm
	UC013	Xoá nhóm ý định
	UC014	Tìm kiếm ý định
	UC015	**Tạo hành động**

Nhóm use case	Mã use case	Tên use case
Hành động	UC016	Cập nhật hành động
	UC017	Xoá hành động
	UC018	Tạo nhóm hành động
	UC019	Sửa tên nhóm hành động
	UC020	Xoá nhóm hành động
	UC021	Tìm kiếm hành động
Thực thể	UC022	Tạo thực thể
	UC023	Cập nhật thực thể
	UC024	Xoá thực thể
	UC025	Tạo nhóm thực thể
	UC026	Sửa tên nhóm thực thể
	UC027	Xoá nhóm thực thể
Kịch bản	UC028	Tạo kịch bản
	UC029	Sửa tên kịch bản
	UC030	**Thiết kế kịch bản**
	UC031	Xoá kịch bản

2.3.1 Đặc tả use case "Tạo ý định"

Bảng 3 đặc tả use case tạo ý định, use case sẽ được thực hiện bởi người dùng và tiền điều kiện là người dùng phải đăng nhập vào hệ thống, đây là use case xảy ra khi người dùng yêu cầu tạo ý định.

Bảng 3 Đặc tả use case "Tạo ý định"

<table>
<tr><td>Mã use case</td><td>UC007</td><td>Tên use case</td><td>Use case Tạo ý định</td></tr>
<tr><td>Tác nhân</td><td colspan="3">Người dùng</td></tr>
<tr><td>Tiền điều kiện</td><td colspan="3">Người dùng đăng nhập vào hệ thống xây dựng chatbot với vai trò là chủ của bot hoặc người chỉnh sửa.</td></tr>
<tr><td>Luồng sự kiện chính (Thành công)</td><td colspan="3"><table>
<tr><th>STT</th><th>Thực hiện bởi</th><th>Hành động</th></tr>
<tr><td>1.</td><td>Người dùng</td><td>Chọn chức năng tạo ý định</td></tr>
<tr><td>2.</td><td>Hệ thống</td><td>Hiển thị màn hình tạo ý định</td></tr>
<tr><td>3.</td><td>Người dùng</td><td>Nhập và chọn các thông tin của ý định (mô tả bên dưới *)</td></tr>
<tr><td>4.</td><td>Người dùng</td><td>Yêu cầu tạo ý định</td></tr>
<tr><td>5.</td><td>Hệ thống</td><td>Kiểm tra các thông tin của ý định</td></tr>
<tr><td>6.</td><td>Hệ thống</td><td>Lưu thông tin vào cơ sở dữ liệu</td></tr>
<tr><td>7.</td><td>Hệ thống</td><td>Thông báo lưu ý định thành công</td></tr>
</table></td></tr>
<tr><td>Luồng sự kiện thay thế</td><td colspan="3"><table>
<tr><th>STT</th><th>Thực hiện bởi</th><th>Hành động</th></tr>
<tr><td>6.</td><td>Hệ thống</td><td>Thông báo lỗi: ý định trùng tên</td></tr>
</table></td></tr>
</table>

* Các trường thông tin của ý định

Bảng 4 mô tả các trường thông tin khi tạo ý định, bao gồm các thông tin về tên tên trường dữ liệu, mô tả của các trường và các trường bắt buộc phải có thông tin.

Bảng 4 Các trường thông tin của ý định

STT	Trường dữ liệu	Mô tả	Bắt buộc
1.	Tên		có

STT	Trường dữ liệu	Mô tả	Bắt buộc
2.	Tên nhóm	Là ô select để người dùng lựa chọn	có
3.	Câu mẫu		
4.	Yêu cầu tham số	Là ô checkbox	
5.	Tên tham số		
6.	Thực thể	Là ô select để người dùng lựa chọn	
7.	Phản hồi	Phím bấm để mở ra popup chỉnh sửa phản hồi	
7a.	Hành động hỏi lại	Là ô select để người dùng lựa chọn	
7b.	Số lần hỏi lại		
7c.	Hành động khi quá số lần hỏi lại	Là ô select để người dùng lựa chọn	
8.	Kích hoạt hành động đi cùng	Ô checkbox	
9.	Hành động đi cùng	Là ô select để người dùng lựa chọn	

2.3.2 Đặc tả use case “Tạo hành động”

Bảng 5 đặc tả use case tạo hành động, use case sẽ được thực hiện bởi người dùng và tiền điều kiện là người dùng phải đăng nhập vào hệ thống, đây là use case xảy ra khi người dùng yêu cầu tạo hành động.

Bảng 5 Đặc tả use case “Tạo hành động”

Mã use case	UC014	Tên use case	Use case Tạo hành động
Tác nhân	Người dùng		

<table>
<tr><td>Tiền điều kiện</td><td colspan="3">Người dùng đăng nhập vào hệ thống xây dựng chatbot</td></tr>
<tr><td>Luồng sự kiện chính (Thành công)</td><td colspan="3"><table>
<tr><th>STT</th><th>Thực hiện bởi</th><th>Hành động</th></tr>
<tr><td>1.</td><td>Người dùng</td><td>Chọn chức năng tạo hành động</td></tr>
<tr><td>2.</td><td>Hệ thống</td><td>Hiển thị màn hình tạo hành động</td></tr>
<tr><td>3.</td><td>Người dùng</td><td>Nhập các trường thông tin (mô tả ở phía dưới*)</td></tr>
<tr><td>4.</td><td>Người dùng</td><td>Chọn thẻ hành động (danh sách các thẻ hành động được mô tả phía dưới **)</td></tr>
<tr><td>5.</td><td>Người dùng</td><td>Yêu cầu tạo hành động</td></tr>
<tr><td>6.</td><td>Hệ thống</td><td>Kiểm tra thông tin hành động</td></tr>
<tr><td>7.</td><td>Hệ thống</td><td>Lưu thông tin vào cơ sở dữ liệu</td></tr>
<tr><td>8.</td><td>Hệ thống</td><td>Thông báo lưu hành động thành công</td></tr>
</table></td></tr>
<tr><td>Luồng sự kiện thay thế</td><td colspan="3"><table>
<tr><th>STT</th><th>Thực hiện bởi</th><th>Hành động</th></tr>
<tr><td>4a.</td><td>Hệ thống</td><td>Thông báo lỗi: cần nhập đủ các trường thông tin cần thiết của node nếu Người quản lý nhập thiếu</td></tr>
</table></td></tr>
</table>

* Các trường thông tin của hành động

Bảng 6 mô tả các trường thông tin khi tạo hành động, bao gồm các thông tin về tên tên trường dữ liệu, mô tả của các trường và các trường bắt buộc phải có thông tin.

Bảng 6 Các trường thông tin của hành động

STT	Trường dữ liệu	Mô tả	Bắt buộc
1.	Tên hành động		có
2.	Tên nhóm	Là ô select để người dùng lựa chọn	có

** Dữ liệu đầu vào của các thẻ hành động

Bảng 7 mô tả chi tiết các dữ liệu đầu vào của các thẻ hành động (i) nhập văn bản, (ii) Hình ảnh, (iii) âm thanh, (iv) video, (v) category, (vi) json api.

Bảng 7 Dữ liệu đầu vào của các thẻ hành động

STT	Trường dữ liệu	Mô tả	Bắt buộc
Nhập văn bản	Văn bản	Nội dung dạng văn bản phản hồi tới khách hàng	
Hình ảnh	Tải lên	Một nút bấm để người dùng tải ảnh lên	
	Nhập link ảnh	Nếu không bấm nút thì người dùng có thể nhập trực tiếp đường dẫn	
	Mô tả	Mô tả hình ảnh	
Âm thanh	Tải lên	Một nút bấm để người dùng tải tệp âm thanh lên, tệp âm thanh có định dạng mp3	
	Nhập đường dẫn âm thanh	Nếu không bấm nút thì người dùng có thể nhập trực tiếp đường dẫn	
	Mô tả	Mô tả tệp âm thanh	
Video	Tải lên	Một nút bấm để người dùng tải tệp video lên, tệp âm thanh có định dạng mp3	
	Nhập đường dẫn video	Nếu không bấm nút thì người dùng có thể nhập trực tiếp đường dẫn	
	Mô tả	Mô tả video	
Category	Tên lựa chọn		
	Giá trị		

STT	Trường dữ liệu	Mô tả	Bắt buộc
JSON API	Phương thức	Phương thức gọi api đến hệ thống thứ ba	
	Đường dẫn	Địa chỉ của spi sẽ gọi đến gọi	
	Tiêu đề header	Tiêu đề header khi gọi api	
	Giá trị tiêu đề header	Giá trị tiêu đề header khi gọi api	
	Tiêu đề body	Tiêu đề body khi gọi api	
	Giá trị body	Giá trị body khi gọi api	
	Chạy thử	Nút gọi thử api và giá trị nhận được sẽ hiển thị ở ô phản hồi	
	Tên slot	Tên sot sẽ lấy giá trị từ Chuỗi Json nhận được khi gọi api	
	Giá trị slot	Giá trị sẽ lấy ra trong chuỗi json nhận được khi gọi api	

2.3.3 Đặc tả use case “Thiết kế kịch bản”

Bảng 8 đặc tả use case thiết kế kịch bản, use case sẽ được thực hiện bởi người dùng và tiền điều kiện là người dùng phải đăng nhập vào hệ thống, đây là use case xảy ra khi người dùng muốn thiết kế một kịch bản nhắn tin.

Bảng 8 Đặc tả use case “Thiết kế kịch bản”.

Mã use case	UC030	Tên use case	Use case thiết kế kịch bản
Tác nhân	Người dùng		
Tiền điều kiện	Người dùng đăng nhập vào hệ thống xây dựng chatbot		

Luồng sự kiện chính (Thành công)	**STT**	**Thực hiện bởi**	**Hành động**
	1.	Người quản lý	Chọn chức năng kịch bản tin nhắn
	2.	Hệ thống	Hiển thị giao diện kịch bản tin nhắn
	3.	Người dùng	Thực hiện lặp lại các thao tác tạo node và cấu hình cho node để tạo ra luồng kịch bản tin nhắn
	4.	Người dùng	Thực thao tác chọn loại node từ thanh menu và yêu cầu tạo node cho luồng kịch bản (mô tả ở phía dưới *)
	5.	Hệ thống	Hiển thị node trên giao diện
	6.	Người dùng	Nhập và chọn thông tin của node (mô tả phía dưới **)
	7.	Người dùng	Thực hiện các thao tác nối giữa các node với nhau để tạo một kịch bản hoàn chỉnh
	8.	Người dùng	Yêu cầu tạo kịch bản
	9.	Hệ thống	Kiểm tra các thông tin của kịch bản
	10.	Hệ thống	Lưu thông tin vào cơ sở dữ liệu
	11.	Hệ thống	Thông báo lưu kịch bản thành công
Luồng sự kiện thay thế	**STT**	**Thực hiện bởi**	**Hành động**
	3a.	Người dùng	Người dùng tạo node mới bằng cách kéo đường dẫn từ node hiện tại ra vị trí bất kì chọn node sẽ tạo ở menu
	6a.	Người dùng	Tạo mới nội dung cho node
	6b.	Người dùng	Chỉnh sửa nội dung của node
	6c.	Người dùng	Yêu cầu xoá node

	6d.	Người dùng	Yêu cầu nhân bản node hiện tại	
	7a.	Hệ thống	Thông báo lỗi: một node Không thể nối với hai node hành động	
	7b.	Hệ thống	Thông báo lỗi: node điều kiện không thể nối với ý định	
	7c.	Hệ thống	Thông báo lỗi: không thể nối hai node ý định với nhau	
	7d.	Người dùng	Di chuyển node đến vị trí thích hợp	

* Dữ liệu của các loại node

Bảng 9 mô tả các dữ liệu đầu vào của các loại node (i) ý định, (ii) điều kiện, (iii) hành động.

Bảng 9 Dữ liệu đầu vào của các loại node

Node	**Trường dữ liệu**	**Mô tả**	**Bắt buộc**
Ý định	Tên ý định	Là ô autocomplete người dùng có thể nhập và tìm kiếm để chọn ý định đã tạo từ trước	có
Điều kiện	Tham số		có
	Toán tử so sánh	là ô select để chọn giá trị của tham số so sánh với giá trị nhập vào	
	Giá trị		
	Toán tử quan hệ	Là ô select để chọn quan hệ giữa các điều kiện con	
Hành động	Tên hành động	Là ô autocomplete người dùng có thể nhập và tìm kiếm để chọn hành động đã tạo từ trước	
	Hành động hỏi lại		

Node	Trường dữ liệu	Mô tả	Bắt buộc
	Hành động khi quá số vòng lặp	Là ô autocomplete người dùng có thể nhập và tìm kiếm để chọn hành động đã tạo từ trước	
	Số lần hỏi lại	Là ô input nhập số	

2.4 Yêu cầu phi chức năng

2.4.1 Tính dễ dùng

Hệ thống sử dụng giao diện web đơn giản, đẹp, thân thiện với người dùng, có sự thống nhất về bố cục và màu sắc của các phân tử trong một trang và giữa các trang với nhau.

2.4.2 Tính dễ nâng cấp và bảo trì

Hệ thống vẫn đang trong giai đoạn phát triển và mở rộng nên cần thiết kế để dễ dàng sửa đổi và nâng cấp sao cho phù hợp mỗi khi có yêu cầu mới. Các tính năng nếu cần sửa đổi hoặc thêm mới cũng sẽ không ảnh hưởng gì đến sự hoạt động của hệ thống.

Chương 2 đã trình bày tổng quan các chức năng đặc tả một số chức năng chính trong hệ thống, Chương 3 em sẽ giới thiệu các công nghệ chính được sử dụng trong việc xây dựng hệ thống.

Chương 3 Công nghệ sử dụng

Từ quá trình phân tích yêu cầu trong Chương 2, Chương 3 em sẽ giới thiệu cộng nghệ sử dụng trong hệ thống xây dựng và quản lý chatbot trên nền tảng web. Nội dung này bao gồm công nghệ sử dụng phía Frontend, công nghệ sử dụng phía Backend.

3.1 Frontend

3.1.1 ReactJS

ReactJS [8] là một thư viện JavaScript rất phổ biến để xây dựng giao diện phía người dùng .Nó cho tốc độ phản hồi tuyệt vời khi user nhập liệu bằng cách sử dụng phương pháp mới để render trang web. Components của công cụ này được phát triển bởi Facebook. Nó được ra mắt như một công cụ JavaScript mã nguồn mở vào năm 2013. Hiện tại, nó đã đi trước các đối thủ chính như Angular và Bootstrap, hai thư viện JavaScript bán chạy nhất thời bấy giờ. Dưới đây là một số tính năng quan trọng và nổi bật của ReactJS:

JSX (JavaScript XML) – là cú pháp mở rộng của javascript, Facebook sử dụng JSX để biểu thị UI Components. JSX được đánh giá là sử dụng đơn giản hơn JavaScript và cho phép trích dẫn HTML cũng như việc sử dụng các cú pháp thẻ HTMl để render các subcomponent. JSX tối ưu hoá code khi biên soạn, vì vậy nó nhanh hơn so với code javascript tương đương.

Component-based – React được xây dựng dựa trên các component. Mỗi component được thiết kế với các trạng thái và thành phần giao diện khác nhau. Component sẽ bao gồm state và props. State đóng vai trò lưu trữ thông tin, thể hiện các trạng thái của component bằng các giá trị, khi giá trị state thay đổi thì toàn bộ component sẽ được cập nhật lại. Props là dữ liệu được truyền từ component cha xuống component con, trong component giá trị props là bất biến. Ngoài ra, component còn hỗi trả khả năng tái sử dụng code.

Single-way data flow (Luồng dữ liệu một chiều) – ReactJS không có những module chuyên dụng để xử lý data, vì vậy ReactJS chia nhỏ view thành các component nhỏ có mối quan hệ chặt chẽ với nhau. Luồng dữ liệu trong ReactJS là luồng dữ liệu một chiều từ cha xuống con. Việc ReactJS sử dụng one-way data flow có thể gây khó khăn cho những người muốn tìm hiểu và ứng dụng vào trong các dự án. Tuy nhiên, cơ chế này sẽ phát huy được ưu điểm của mình khi cấu trúc cũng như chức năng của view trở nên phức tạp thì ReactJS sẽ phát huy được vai trò của mình.

Virtual DOM – Được thiết kế để giải quyết vấn đề cập nhật DOM thường xuyên một cách có hiệu năng cao hơn. Vì nó là bản thiết kế của DOM nên có thể cập nhật thường xuyên mà không cần sử dụng đến DOM API. Khi tất cả các thay đổi được cập nhật trong DOM ảo, React sẽ kiểm tra xem thay đổi cụ thể nào được áp dụng vào DOM gốc (bởi thuật toán so sánh diff của React), và áp dụng thay đổi này vào chính xác chỗ cần thay đổi. Điều này sẽ làm cho hệ thống chatbot nhanh hơn mà không bị lãng phí bộ nhớ.

3.1.2 Material UI

Material UI [12] là một thư viện các React Component đã được tích hợp thêm cả Google's Material Design. Material cung cấp khá đầy đủ các component để có thể tạo ra một trang web nhanh Chóng hơn, với những button, textfield, ... được thiết kế sẵn, rất tiện lợi. Đây là một thư viện rất tốt cho việc xây dựng giao diện của hệ thống.

3.1.3 React Diagram

React Diagram [9] là công cụ chuyên hỗ trợ biểu diễn các thành phần trên giao diện dưới dạng biểu đồ.

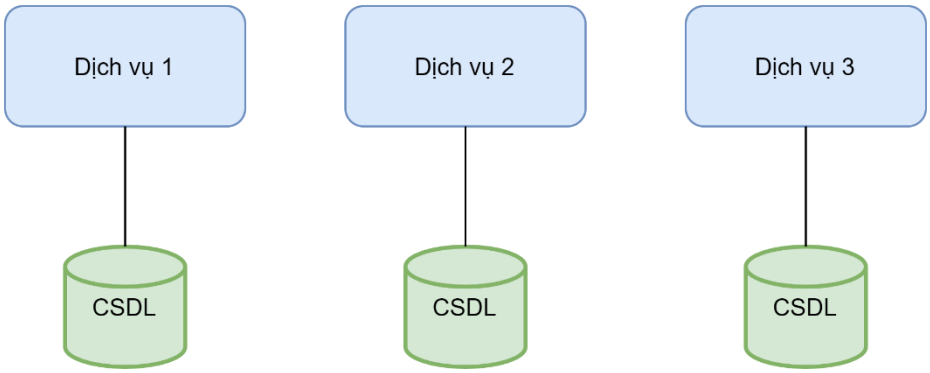
Để tạo ra biểu đồ cần sử dụng các đối tượng DiagramEngine, DiagramModel, NodeModel và Link. Trong đó NodeModel là các thành phần sẽ được biểu diễn, hiển thị trên giao diện, Link giúp kết nối các node để tạo thành một biểu đồ, DiagramModel sẽ chứa NodeModel và Link để hiển thị các thành phần này lên giao diện, đồng thời nó còn cung cấp các API hỗ trợ thao tác giữa các Node và Link với nhau và cuối cùng DiagramEngine sẽ chứa tất cả các đối tượng trên.

Thư viện cung cấp rất nhiều các component, các phương thức, API hữu ích để người dùng có thể tự tuỳ chỉnh giao diện theo mong muốn của mình.

3.2 Backend

3.2.1 Kiến trúc Microservices

Microservices [1] là một kiến trúc phần mềm. Các module trong phần mềm này được chia thành các dịch vụ (service) rất nhỏ (microservice) và có cơ sở dữ liệu riêng như **Hình 8** [2]. Mỗi service được đặt trên một server riêng để dễ dàng nâng cấp. giao tiếp với nhau thông qua giao thức HTTP hoặc sử dụng MessageQueue.



Hình 8 Minh hoạ kiến trúc xây dựng theo mô hình Microservices.

Ưu điểm của kiến trúc Microservices là dễ dàng nâng cấp vào bảo trì, khi một dịch vụ bị lỗi thì toàn bộ hệ thống vẫn hoạt động bình thường, dễ dàng mở rộng thêm chức năng mới bằng cách tạo mới một dịch vụ và kết nối tới cổng trung gian.

3.2.2 NodeJS

NodeJS [3] là một nền tảng giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng, được xây dựng và phát triển trên ngôn ngữ Javascript. NodeJS được phát triển bởi Ryan Dahl và năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux. Nhờ cơ chế xử lý bất đồng bộ, NodeJS có tốc độ xử lý rất nhanh, nó có thể xử lý hàng ngàn kết nối cùng một lúc mà không gặp khó khăn nào. NodeJS chạy đa nền tảng phía server sử dụng kiến trúc hướng sự kiện Event-driven, với cộng đồng sử dụng lớn và hoàn toàn miễn phí.

ExpressJS là một framework cung cấp tính năng mạnh mẽ để phát triển web và mobile, nó được phát triển dựa trên nền tảng NodeJS. Framework hỗ trợ Rest API, cho phép định nghĩa Middleware hỗ trợ cho việc tổ chức và tái sử dụng code, nó giúp định nghĩa routes và các request method đến server một cách dễ dàng.

3.2.3 MongoDB

MongoDB [4] là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL NoSql và được hàng triệu người dùng. Nó là một CSDL hướng tài liệu (document), các dữ liệu được lưu trữ trong document dưới dạng key-value thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh. So với cơ sở dữ liệu quan hệ thì trong MongoDB bộ sưu tập (collection) ứng với bảng (table), tài liệu (document) ứng với hàng (row). Các thông tin liên quan được lưu trữ cùng nhau, ngoài ra dữ liệu cũng được ghi đệm lên RAM, hạn chế truy cập vào ở cứng nên tốc độ truy vấn trong MongoDB rất nhanh chóng. Do dữ liệu không có sự ràng buộc vì vậy khi thêm, sửa xoá chúng ta không cần mất thời gian kiểm tra về các ràng buộc dữ liệu như

trong cơ sở dữ liệu quan hệ. Và cũng do không có sự ràng buộc nên khi truy truy vấn chúng ta cũng phải rất cẩn thận nếu không sẽ dẫn đến sai lệch dữ liệu.

3.2.4 Redis

Redis [7] (Remote Dictionary Server) là một mã nguồn mở dùng để lưu trữ dữ liệu có cấu trúc có thể sử dụng như một cơ sở dữ liệu NoSql. Ưu điểm lớn nhất của redis là lưu trữ dữ liệu trên Ram, nên việc đọc ghi dữ liệu rất nhanh chóng, thích hợp để làm bộ nhớ đệm.

3.2.5 Elasticsearch

Elasticsearch [6] là một search engine, có khả năng tìm kiếm nhanh chóng. Nó hoạt động như một web server và đồng thời giao tiếp thông qua RESTful do vây nên nó không phụ thuộc vào client viết bằng gì hay hệ thống hiện tại của bạn viết bằng gì. Nên việc tích hợp nó vào hệ thống rất dễ dàng. Ưu điểm của elasticsearch là tìm kiếm dữ liệu nhanh chóng, mạnh mẽ dựa trên Apache lucene (near-realtime searching), có khả năng phân tích dữ liệu và đặc biệt elasticsearch hỗ trợ tìm kiếm mở (fuzzy), tức là từ khoá tìm kiếm có thể bị sai lỗi chính tả hay không đúng cú pháp thì vẫn có khả năng elasticsearch trả về kết quả tốt.

3.3 Giao thức kết nối

3.3.1 RabbitMQ

RabbitMQ [5] là một message-broker sử dụng giao thức AMQP (Advanced Message Queuing Protocol – giao thức truyền nhận tin nhắn sử dụng hàng đợi). Nó đóng vai trò trung gian, giúp lưu trữ và điều phối yêu cầu giữa người nhận và người gửi. Trong hệ thống microservice việc các dịch vụ gọi chéo nhau là khá nhiều khiến cho luồng xử lý phức tạp làm cho việc lập trình trở nên khó khăn hơn, RabbitMQ được sử dụng để giải quyết các vấn đề như vậy. Giao tiếp trong RabbitMQ được thực hiện bằng cách người gửi (producer) gửi tin nhắn có chứa thông tin vào trong một hàng đợi và người nhận (consumer) sẽ chỉ việc lắng nghe hàng đợi và nhận tin nhắn để xử lý.

3.3.2 Rest API

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng dưới dạng phổ biến như JSON hay XML.

REST (REpresentational State Transfer) - là một dạng chuyển đổi cấu trúc dữ liệu, là một phong cách kiến trúc cho việc thiết kế các ứng dụng có kết nối. Nó sử dụng giao thức HTTP đơn giản để giao tiếp giữa các máy tính với nhau. 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, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn để thiết kế API cho các ứng dụng web, để tiện cho việc quản lý các tài nguyên (resource). Nó được sử dụng phổ biến để giao tiếp giữa các ứng dụng (web, mobile, …) với nhau.

REST có chức năng quan trọng nhất là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các tài nguyên. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

Chưởng 3 em đã giới thiệu về các công nghệ sử dụng trong hệ thống. để hiểu rõ hơn về kiến trúc và cách thức hệ thống hoạt động, em xin trình bày cụ thể trong chương 4.

Chương 4 Phát triển và triển khai ứng dụng

Sau khi trình bày các công nghệ ở Chương 3, Chương 4 em sẽ trình bày chi tiết về thiết kế kiến trúc đến giao diện và cơ sở dữ liệu, sau đó là quá trình xây dựng, phát triển, kiểm thử và triển khai hệ thống.

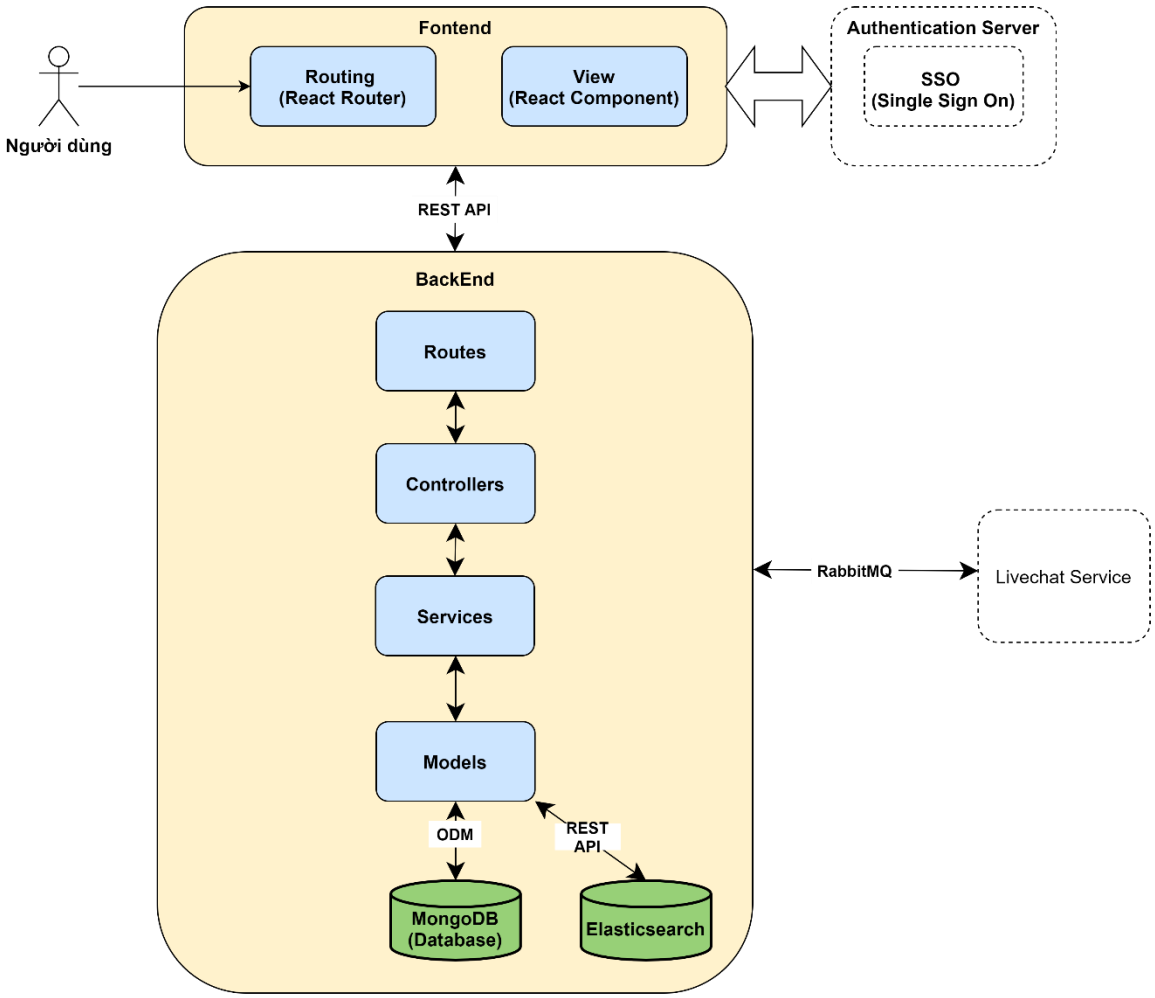
4.1 Thiết kế kiến trúc

4.1.1 Kiến trúc tổng quan

Hình 9 mô tả kiến trúc tổng quan của hệ thống xây dựng và quản lý chatbot theo luật, bao gồm các thành phần của Frontend, Backend, các công cụ kết nối giữa Frontend và Backend, giữa Backend và cơ sở dữ liệu, giữa Backend và Elasticsearch và giữa dịch vụ (service) của hệ thống chatbot với các dịch vụ (service) của các bên liên quan.

Frontend sẽ sử dụng ReactJS để xây dựng các thành phần trang web và tương tác với backend thông qua API. Trong mục 4.1.1 sẽ trình bày chi tiết thiết kế các thành phần giao diện.

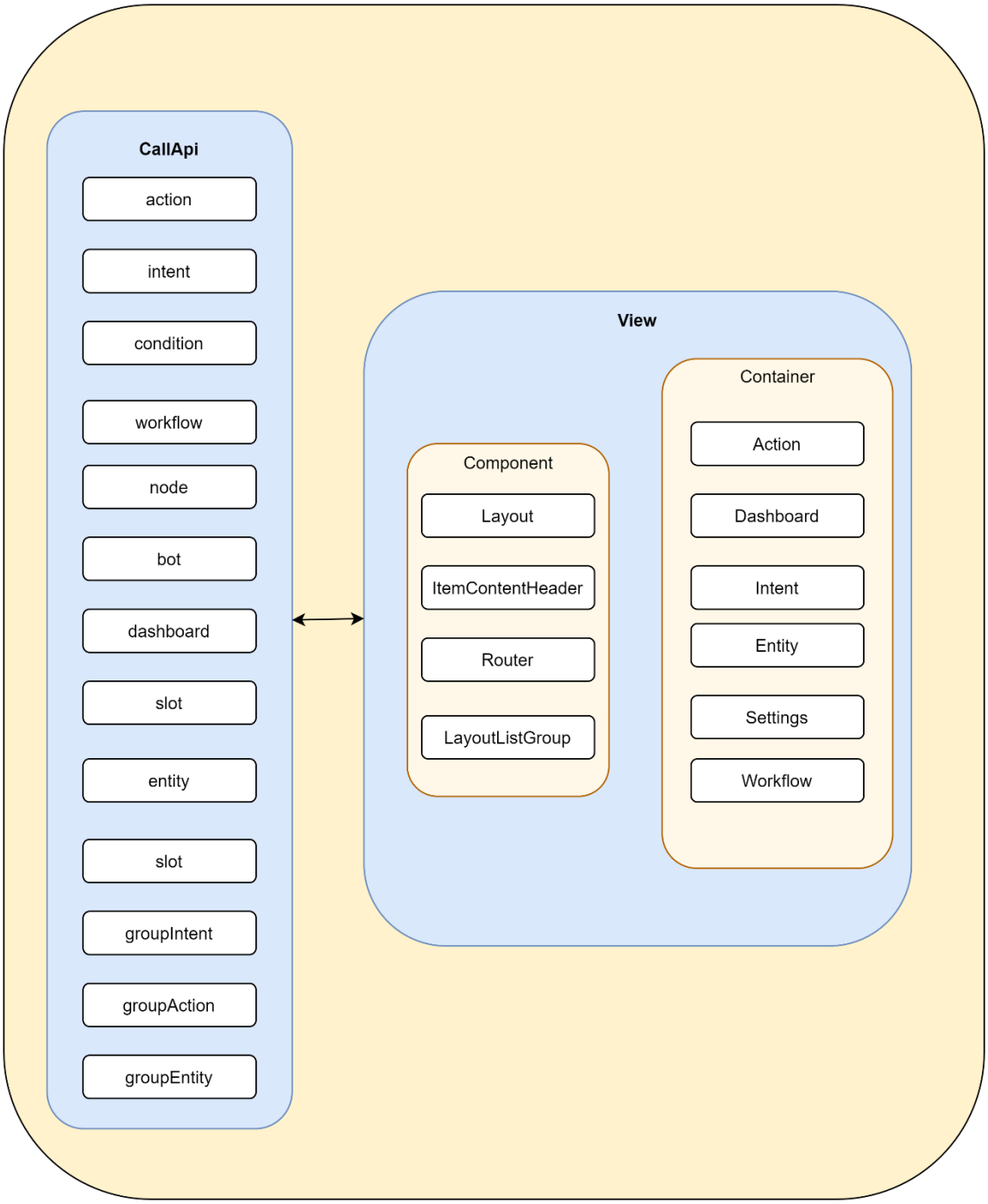
Backend sẽ sử dụng kiến trúc Microservices để có thể tái sử dụng các dịch vụ có sẵn như livechat service và account service. Dịch vụ của hệ thống sẽ giao tiếp với với các dịch vụ khác thông qua RabbitMQ, và dịch vụ sẽ không truy vấn trực tiếp vào cơ sở dữ liệu mà sẽ thực hiện truy vấn thông qua ODM(Object Data Model). ODM là là biểu diễn dữ liệu trong CSDL dưới dạng các đối tượng Javascript, điều này giúp hệ thống giao tiếp với CSDL trở nên đơn giản và dễ dàng hơn. Để giao tiếp với công cụ tìm kiếm Elasticsearch, hệ thống sẽ sử dụng các API. Hệ thống sẽ sử dụng chung thông tin xác thực nên khi đăng nhập hệ thống sẽ phải kết nối đến dịch vụ xác thực (Authentication Server), đây là dịch vụ cho phép xác thực người dùng tập trung trong kiến trúc microservice, đăng nhập một lần, sử dụng ở nhiều nơi.



Hình 9 Kiến trúc tổng quan hệ thống xây dựng chatbot theo luật.

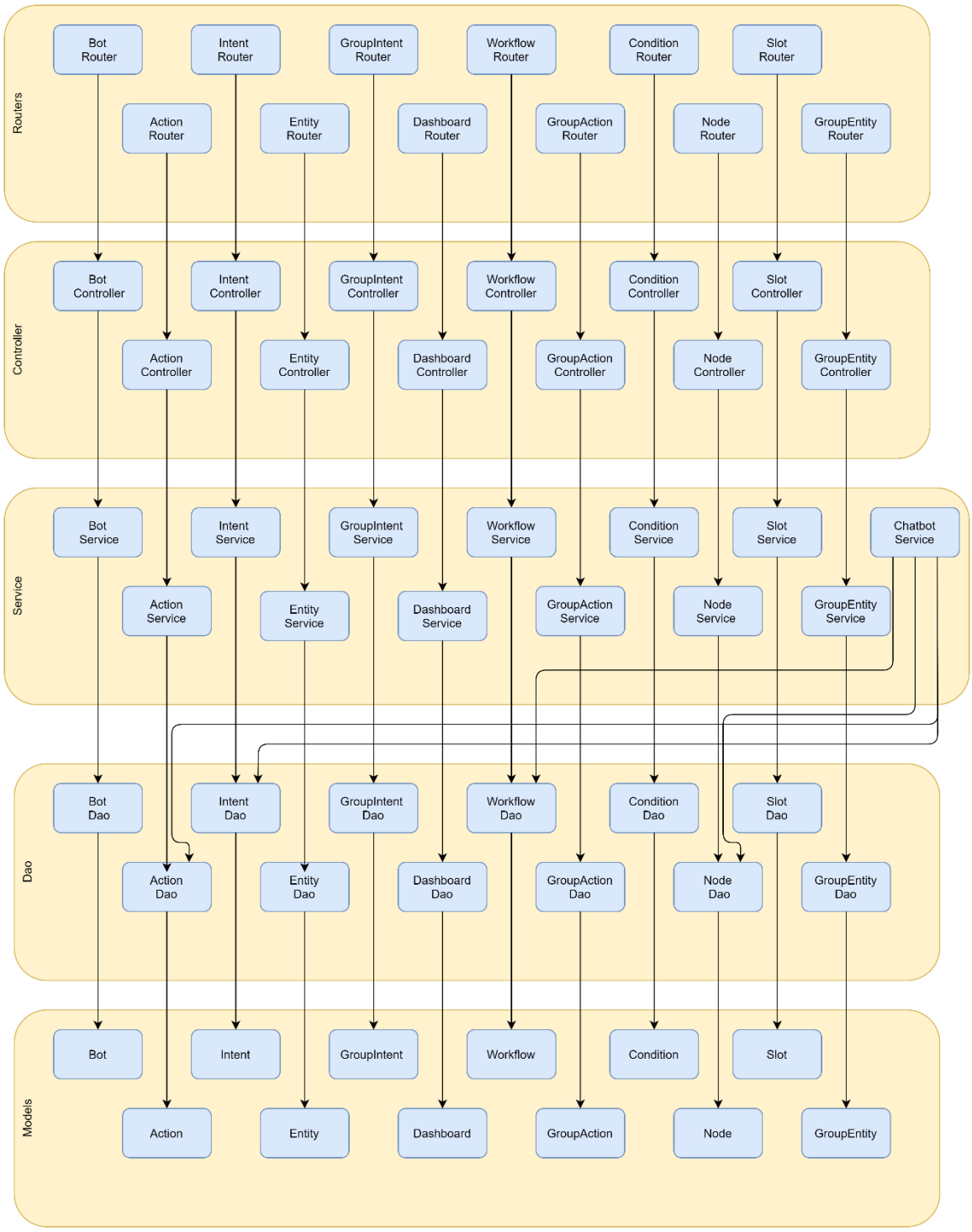
4.1.2 Thiết kế kiến trúc frontend

Hình 10 là thiết kế kiến trúc frontend của hệ thống. Phần container chịu trách nhiệm hiển thị giao diện cho người dùng, trong đó có 7 containers là: Action hiển thị giao diện các thông tin về action, Dashboard hiển thị thống kê các thông tin, về mức độ hiệu quả của bot, Intent hiển thị giao diện các thông tin về intent, Entity hiển thị giao diện các thông tin về entity, Settings hiển thị thông tin bot, chia sẻ quyền quản lý bot và Workflow hiển thị thông tin về kịch bản.



Hình 10 Thiết kế kiến trúc frontend hệ thống xây dựng chatbot theo luật.

4.1.3 Thiết kế kiến trúc backend



Hình 11 Thiết kế kiến trúc backend hệ thống xây dựng chatbot theo luật.

Hình 11 là thiết kế kiến trúc backend của hệ thống, backend sử dụng Nodejs để xây dựng và được thiết kế thành 5 thành phần: Routers, Controllers, Service, Daos và Models.

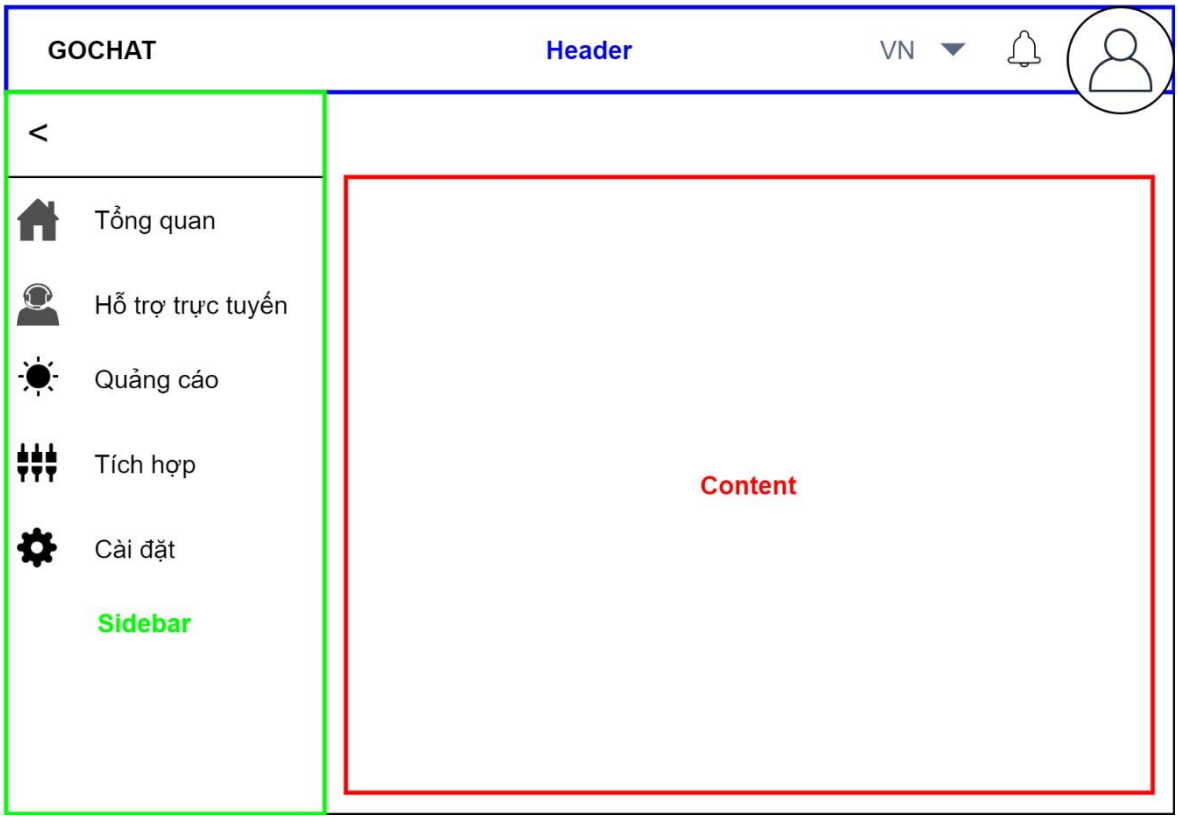
Khi client gửi yêu cầu lên server, các Routers điều hướng yêu cầu đến Controller tương ứng để xử lý, Controller sẽ tiếp nhận yêu cầu và gọi Service để xử lý logic nghiệp vụ, Daos có nhiệm vụ cung cấp các phương thức để thao tác với cơ sở dữ liệu mỗi khi Service yêu cầu và Model sẽ chứa các đối tượng ánh xạ đến cơ sở dữ liệu, các đối tượng này tương ứng với một collection trong cơ sở dữ liệu MongoDB. Việc thiết kế như vậy sẽ giúp các thành phần chỉ thực hiện một nhiệm vụ nhất định, giúp code dễ bảo trì và mở rộng.

4.2 Thiết kế chi tiết

4.2.1 Thiết kế chi tiết frontend

4.2.1.1 Thiết kế mockup

Giao diện được thiết kế theo độ phân giải 1920x1080, đây là độ phân giải phổ biến thường dùng trong thiết kế web. Bố cục của trang web sẽ bao gồm các thành phần Header, Sidebar, Content. Các thành phần được biểu diễn trong **Hình 12**.



Hình 12 Mockup layout của hệ thống xây dựng chatbot.

Header là thanh tiêu đề nằm trên cùng của trang web, nó hiển thị logo thông tin người dùng, người dùng có thể thông qua đây để thay đổi ngôn ngữ và đăng xuất. Sidebar là thanh bên trái của màn hình, nó giúp người dùng điều hướng trang web theo ý muốn và Content sẽ hiển thị nội dung tương ứng.

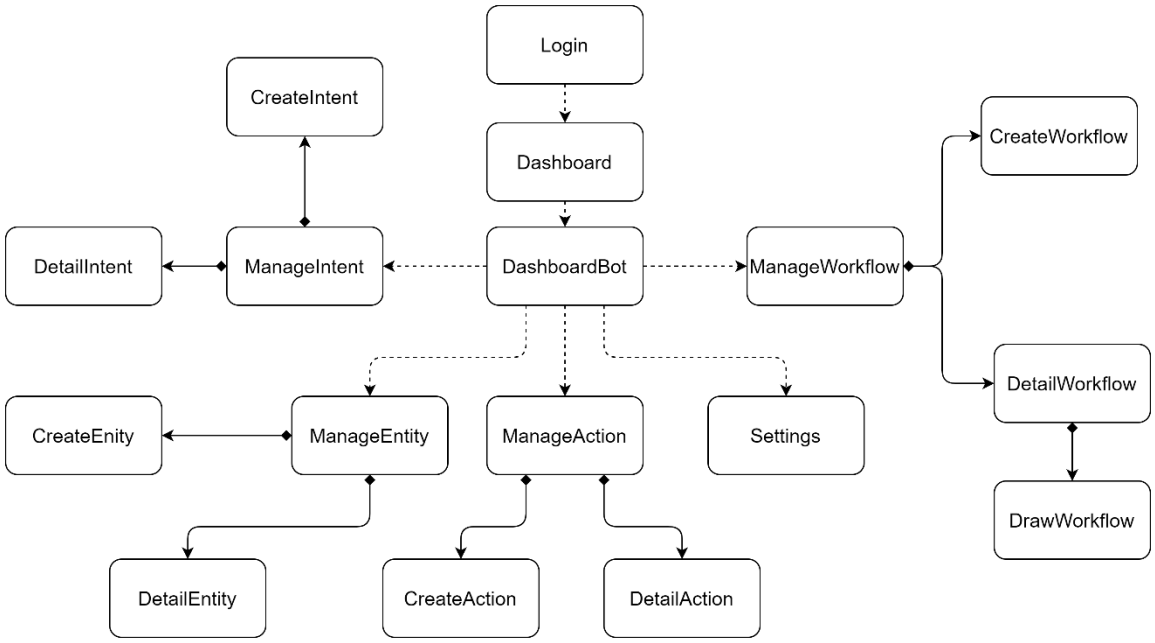
Màu sắc, font chữ, cách nút, hiển thị các thành phần và thông báo được thiết kế hoà hợp nhằm đặt được tính nhất quán cho trải nghiệm người dùng. **Bảng 10** sẽ trình bày các cấu hình được sử dụng trong hệ thống.

Bảng 10 Cấu hình các thuộc tính trên giao diện

Thuộc tính	**Cấu hình**
Màu sắc	Màu chính: #fff Màu khác: #2196f3, #eee, rgba(0, 0, 0, 0.87)
Font chữ	Helvetica, Aldrich
Bo góc	10px
Đổ bóng	rgb(0 0 0 / 42%) 0px 10px 30px -12px, rgb(0 0 0 / 12%) 0px 4px 25px 0px, rgb(0 0 0 / 20%) 0px 8px 10px -5px
Vị trì hiển thị popup thông báo	Góc dưới bên phải màn hình

4.2.1.2 Biểu đồ dịch chuyển màn hình

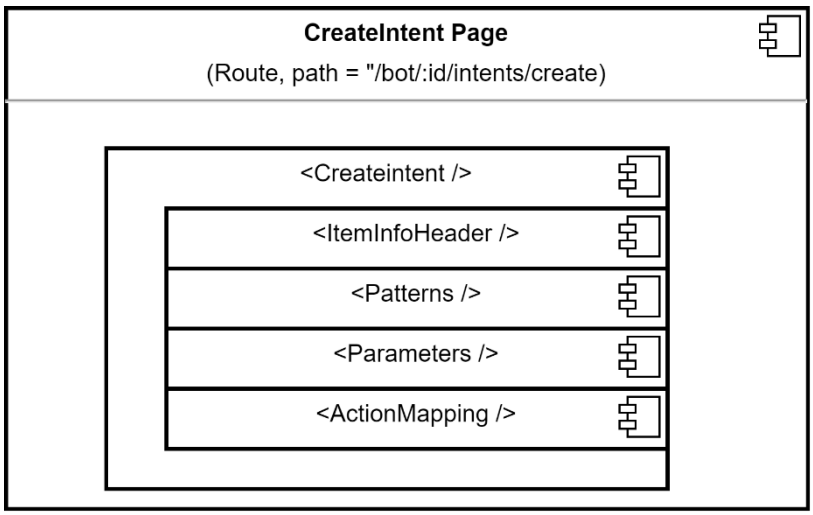
Sau khi đăng nhập vào hệ thống, người dùng sẽ được dịch chuyển đến màn hình Dashboard, sau đó người dùng có thể chọn bot và chuyển đến màn hình DashboardBot và từ đó người dùng có thể chuyển đến các màn hình quản lý intent, action, entity, màn hình quản lý kịch bản và màn hình quản lý thông tin của bot. **Hình 13** là biểu đồ dịch chuyển màn hình trong hệ thống xây dựng chatbot theo luật. mỗi một hình chữ nhật tương ứng với một màn hình, các hướng mũi tên sẽ giải thích sự dịch chuyển giữa các màn hình, mũi tên nét liền có đuôi là một hình thoi biểu thị màn hình dịch chuyển tới là một màn hình con của màn hình hiện tại.



Hình 13 Biểu đồ dịch chuyển màn hình hệ thống xây dựng chatbot theo luật.

4.2.1.3 Thiết kế các thành phần giao diện

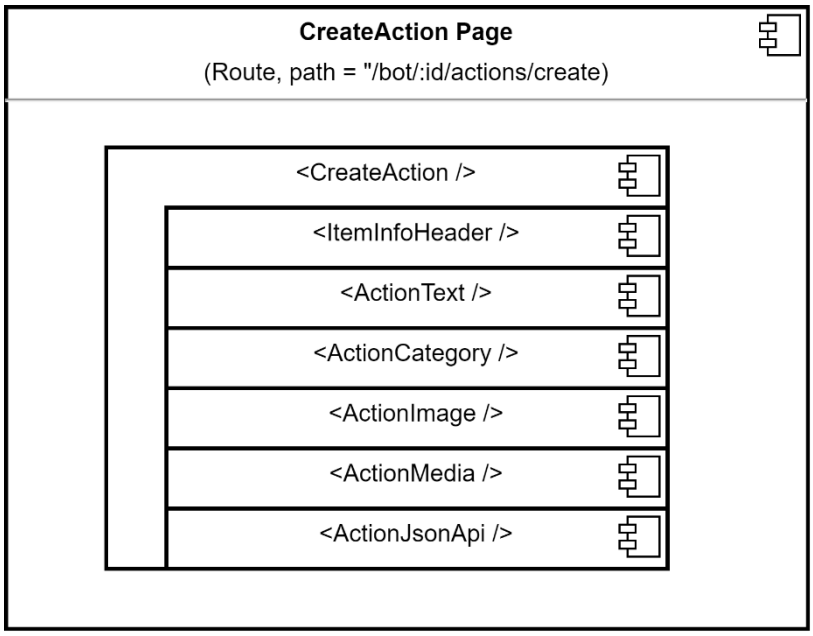
a) Màn hình tạo intent



Hình 14 Thiết kế thành phần giao diện màn hình tạo intent.

Màn hình tạo intent sẽ gồm bốn thành phần chính như **Hình 14**, thành phần ItemInfoHeader thể hiện thông tin tên và nhóm của intent, nút tạo intent mới. Pattern là thành phần hiển thị các câu mẫu, tập ý định do người dùng nhập từ bàn phím vào ô input. Parameters sẽ hiển thị thông tin các tham số, thực thể và ô checkbox để người dùng muốn yêu cầu trích xuất thông tin trong câu hỏi của khách hàng. ActionMapping là thành phần hiển thị action sẽ thực hiện sau khi vào intent này.

b) Màn hình tạo action

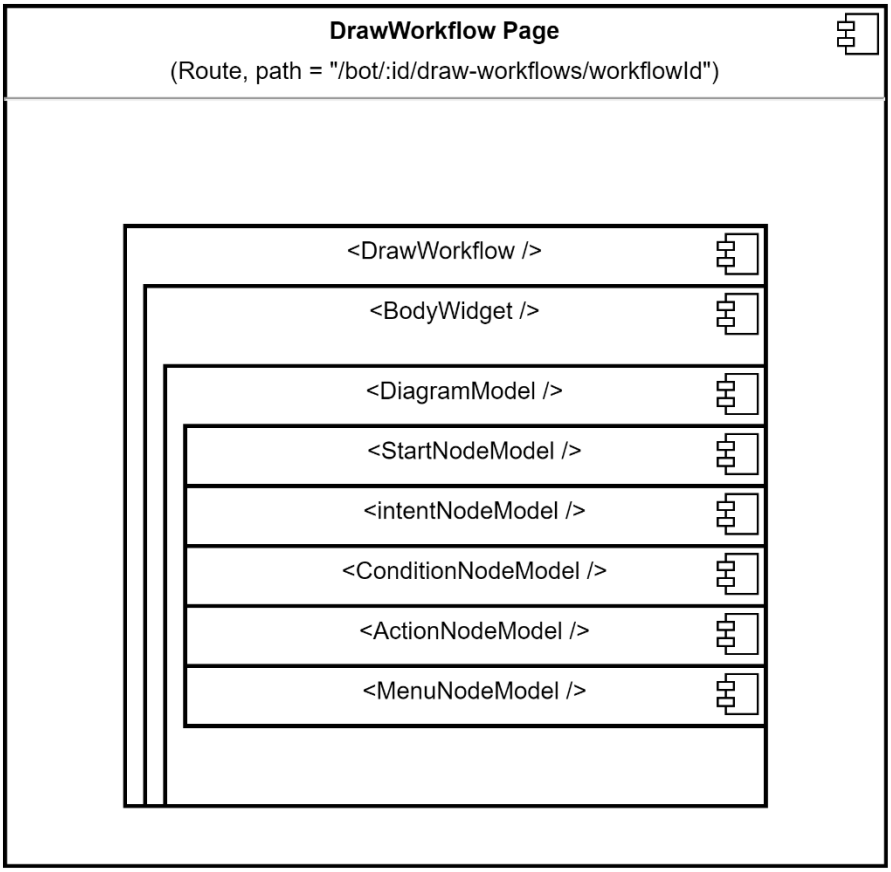


Hình 15 Thiết kế thành phần giao diện tạo action.

Màn hình tạo action gồm sáu thành phần chính như **Hình 15.** Thành phần ItemInfoHeader thể hiện thông tin tên và nhóm của action, nút tạo action mới. ActionText là thành phần dùng để nhập và hiển thị nội dung phản hồi dưới dạng chữ. ActionCategory dùng để nhập vào hiển thị nội dung phản hồi dưới dạng lựa chọn. ActionImage và ActionMedia dùng để nhập hoặc chọn file và hiển thị đường dẫn file. ActionJsonApi để cài đặt các thông số của một api, api này dùng để lấy thông tin từ một hệ thống khác và người dùng có thể gọi thử api, nhận kết quả mẫu và cấu hình các tham số để nhận giá trị nhận được từ kết quả đó.

c) Màn hình vẽ workflow

Màn hình vẽ workflow gồm các thành phần chính như **Hình 16.** BodyWidget là thành phần hiển thị danh sách các lựa chọn để thao tác với biểu đồ. DiagramModel là component cung cấp bởi thư viện, nó có chức năng hiển thị kịch bản hội thoại dưới dạng biểu đồ luồng. StartNodeModel là thành phần giúp hiển đâu là nơi bắt đầu của kịch bản. IntentNodeModel, ConditionNodeModel, ActionNodeModel, MenuNodeModel, là những thành phần được hiển thị khi người dùng thêm hoặc thao tác với kịch bản, các node trên chứa các thông tin có thể điều hướng luồng hội thoại,



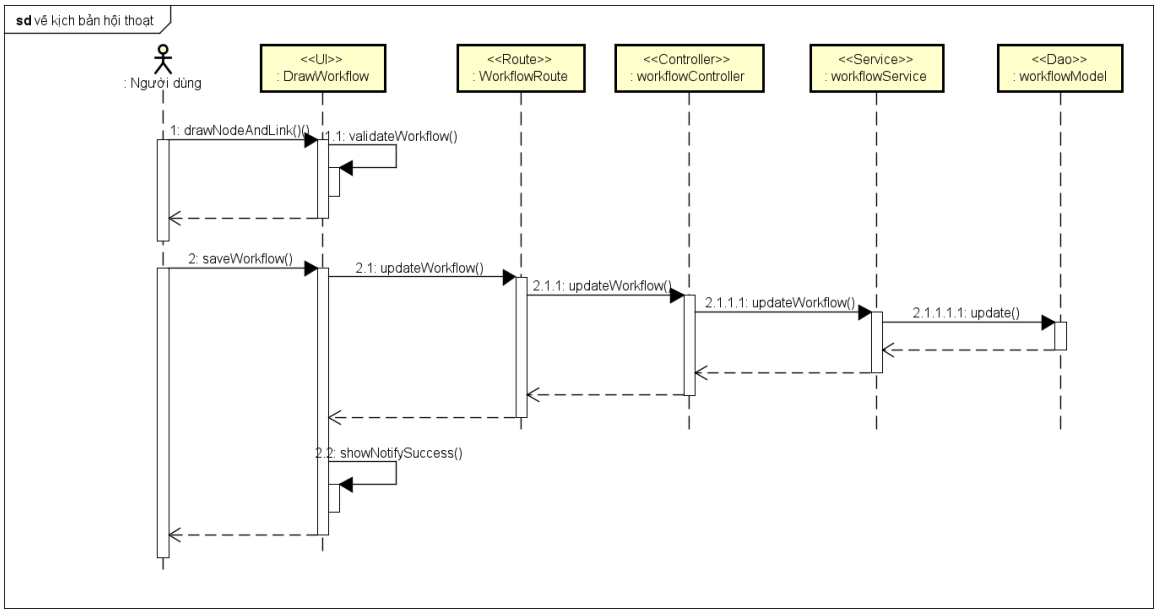
Hình 16 Thiết kế thành phần vẽ workflow.

4.2.2 Thiết chi tiết backend

4.2.2.1 Luồng hoạt động

Trong phần này em xin trình bày luồng hoạt động khi người dùng vẽ kịch bản hội thoại như **Hình 17.**

Trước hết, người dùng cần tạo kịch bản và đặt tên cho kịch bản, tiếp theo người dùng vào giao diện vẽ kịch bản. Sau đó tạo các node intent, condition, action, cấu hình thông tin cho các node đó, nối liên kết giữa các node với nhau, tiếp tục lặp đi lặp lại như vậy cho đến khi hoàn tất kịch bản. Cứ mỗi lần người dùng yêu cầu tạo node trên giao diện, node sẽ được được tạo trực tiếp trong cơ sở dữ liệu.



Hình 17 Biểu đồ trình tự vẽ kịch bản hội thoại.

4.2.2.2 Thiết kế API

Như đã trình bày ở mục 4.1, dịch vụ giao tiếp với frontend qua API, và giao tiếp với các dịch vụ liên quan qua RabbitMQ. Chi tiết tất cả các API trong hệ thống chatbot được mô tả trong **Bảng 11.**

Bảng 11 Danh sách API của hệ thống xây dựng chatbot theo luật

STT	Mục đích	Phương thức	Địa chỉ
1	Tạo bot	POST	/api/v1/bots
2	Lấy danh sách bot	GET	/api/v1/bots
3	Lấy thông tin bot theo ID	GET	/api/v1/bots/:id
4	Cập nhật thông tin bot theo ID	PUT	/api/v1/bots/:id
5	Xóa bot theo ID	DELETE	/api/v1/bots/:id
6	Chia sẻ bot	PUT	/api/v1/bots/:id/add-user/:userId
7	Ngừng chia sẻ bot với một user	PUT	/api/v1/bots/:id/remove-user/:userId

STT	Mục đích	Phương thức	Địa chỉ
8	Lấy bot bằng botToken	GET	/api/v1/verify-bot-token
9	Tạo intent	POST	/api/v1/intents
10	Cập nhật intent	PUT	/api/v1/intents/:id
11	Lấy thông tin intent theo ID	GET	/api/v1/intents/:id
12	Lấy danh sách intent theo botID	GET	/api/v1/intents
13	Xoá intent theo ID	DELETE	/api/v1/intents/:id
14	Cập nhật câu mẫu trong intent	PUT	/api/v1/intents/patterns/:id
15	Thêm câu mẫu	PUT	/api/v1/intents/:id/addUsersay
16	Xoá câu mẫu	PUT	/intents/:id/removeUsersay
17	Tạo action	POST	/api/v1/actions
18	Cập nhật action	PUT	/api/v1/ actions /:id
19	Lấy thông tin action theo ID	GET	/api/v1/actions/:id
20	Lấy danh sách action theo botID	GET	/api/v1/actions
21	Xoá action theo ID	DELETE	/api/v1/actions/:id
22	Tạo condition	POST	/api/v1/conditions
23	Cập nhật condition	PUT	/api/v1/conditions/:id
24	Lấy thông tin condition theo ID	GET	/api/v1/conditions/:id

STT	Mục đích	Phương thức	Địa chỉ
25	Lấy số lượng dữ liệu trong bot	GET	/api/v1/dashboards/statisticWorkingData
26	Lấy thống kê hiệu quả bot theo ngày	GET	/api/v1/dashboards
27	Tạo entity	POST	/api/v1/entities
28	Cập nhật entity	PUT	/api/v1/entities/:id
29	Lấy thông tin entity theo ID	GET	/api/v1/entities/:id
30	Lấy danh sách entity theo botID	GET	/api/v1/entities
31	Xoá entity theo ID	DELETE	/api/v1/entities/:id
32	Tạo workflow	POST	/api/v1/workflow
33	Cập nhật workflow	PUT	/api/v1/workflows/:id
34	Lấy thông tin workflow theo ID	GET	/api/v1/workflows/:id
35	Lấy danh sách workflow theo botID	GET	/api/v1/workflows
36	Xoá workflow theo ID	DELETE	/api/v1/workflows/:id
37	Tạo node	POST	/api/v1/nodes
38	Cập nhật node	PUT	/api/v1/nodes/:id
39	Lấy thông tin node theo ID	GET	/api/v1/nodes/:id
40	Lấy danh sách node theo botID	GET	/api/v1/nodes

STT	Mục đích	Phương thức	Địa chỉ
41	Xoá node theo ID	DELETE	/api/v1/nodes/:id
42	Tạo groupIntent	POST	/api/v1/groupIntents
43	Cập nhật groupIntent	PUT	/api/v1/groupIntents/:id
44	Lấy thông tin groupIntent theo ID	GET	/api/v1/groupIntents/:id
45	Lấy danh sách groupIntent theo botID	GET	/api/v1/groupIntents/getGroupAndItems
46	Xoá groupIntent theo ID	DELETE	/api/v1/groupIntents/:id
47	Tạo groupAction	POST	/api/v1/groupActions
48	Cập nhật groupAction	PUT	/api/v1/groupActions/:id
49	Lấy thông tin groupAction theo ID	GET	/api/v1/groupActions/:id
50	Lấy danh sách groupAction theo botID	GET	/api/v1/groupActions/getGroupAndItems
51	Xoá groupAction theo ID	DELETE	/api/v1/groupActions/:id
52	Tạo groupEntity	POST	/api/v1/groupEntities
53	Cập nhật groupEntity	PUT	/api/v1/groupEntities/:id
54	Lấy thông tin groupEntity theo ID	GET	/api/v1/groupEntities/:id
55	Lấy danh sách groupEntity theo botID	GET	/api/v1/groupEntities/getGroupAndItems
56	Xoá groupEntity theo ID	DELETE	/api/v1/groupEntities/:id

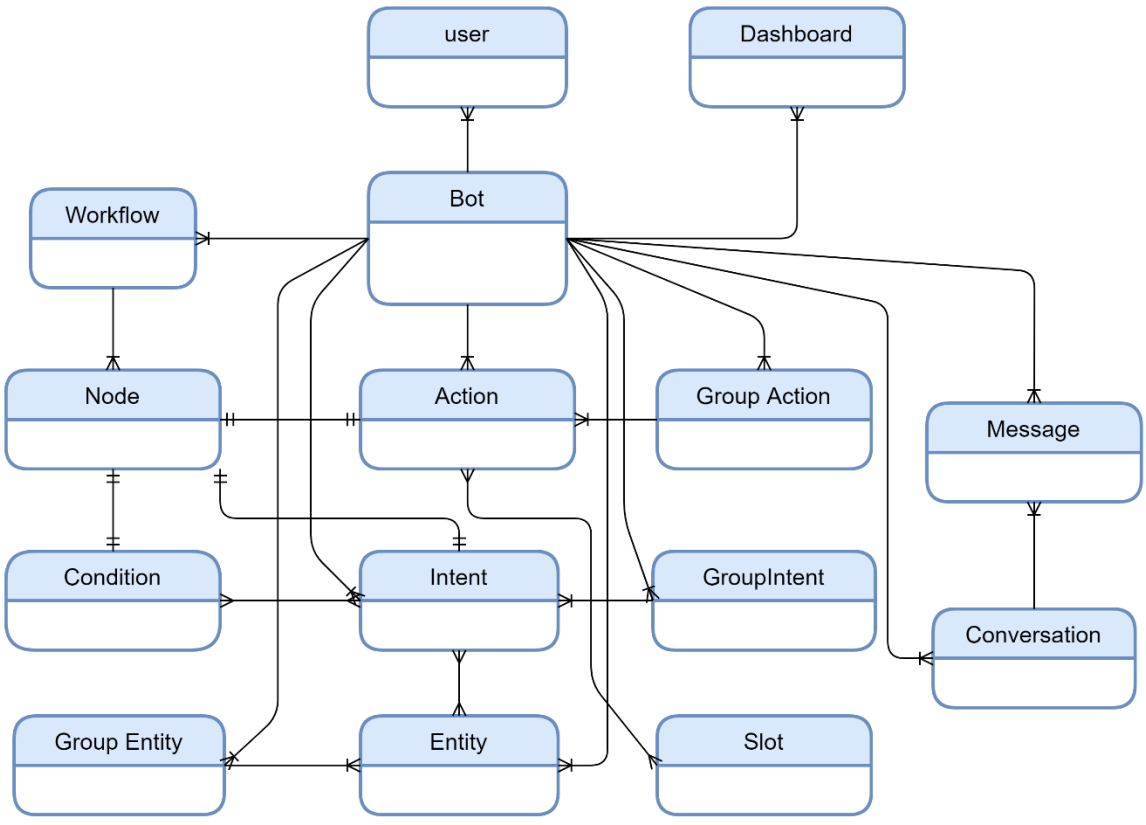
Bảng 12 là danh sách tên và mục đích của các hàng đợi, hàng đợi này là các tin nhắn (message) lần lượt và xử lý chúng theo thứ tự.

Bảng 12 Danh sách hàng đợi tin nhắn

STT	Hàng đợi	Mục đích
1	OUTPUT_QUEUE	Nhận tin nhắn đến của khách hàng
2	LOG_MESSAGE_QUEUE	Xử lý nội dung hội thoại giữa bot và khách hàng
3	USER_EXCHANGE	Đồng bộ thông tin người dùng

4.2.3 Thiết kế cơ sở dữ liệu

4.2.3.1 Biểu đồ thực thể liên kết



Hình 18 Biểu đồ thực thể liên kết hệ thống xây dựng chatbot theo luật.

Hình 18 là biểu đồ thực thể liên kết mà em xây dựng cho hệ thống, ý nghĩa của các thực thể được mô tả trong **Bảng 13**.

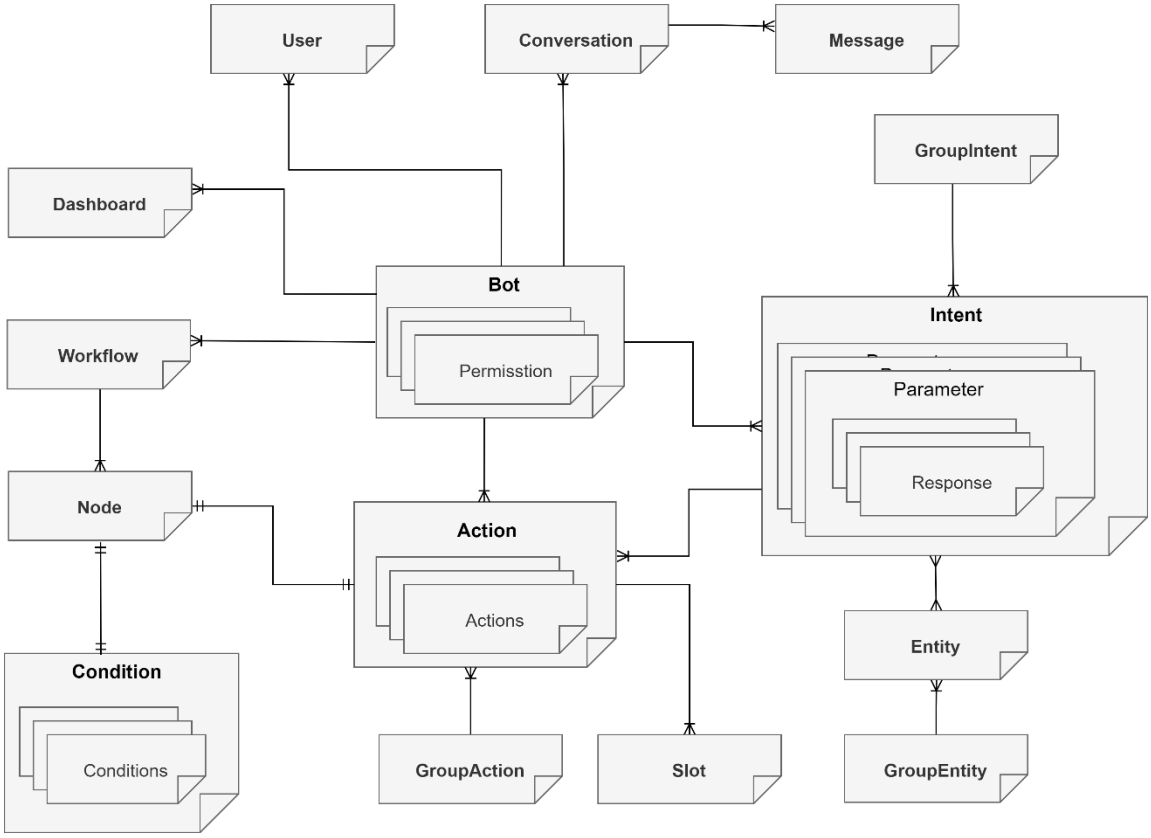
Bảng 13 Giải thích ý nghĩa của các thực thể.

Tên thực thể	Ý nghĩa của thực thể
Bot	Là bot
Workflow	Chứa kịch bản hội thoại
Node	Thông tin chi tiết một node của kịch bản
Condition	Thông tin nội dung của node điều kiện
Intent	Thông tin của intent
Entity	Thông tin về dữ liệu cần trích xuất
GroupEntity	Thông tin nhóm của thực thể
User	Thông tin người dùng
Action	Các phản hồi của bot khi khách hàng nhắn tin tới
Slot	Tham số khi bot thực hiện hành động gọi api
Dashboard	Thông tin thống kê các thông số của bot
GroupIntent	Thông tin nhóm của intent
GroupAction	Thông tin nhóm của action
Conversation	Lịch sử cuộc hội thoại giữa bot và khách hàng
Message	Tin nhắn của bot và khách hàng

4.2.3.2 Thiết kế CSDL trên MongoDB

Từ biểu đồ thực thể liên kết em đã xây dựng cơ sở dữ liệu phi quan hệ NoSQL MongoDB bao gồm 15 collection đó là: Intent, Action, Entity, Condition, User, Workflow, Bot, Node,

Message, Conversation, Dashboard, Slot, GroupIntent, GroupAction, GroupEntity. **Hình 19** là thiết kế cơ sở dữ liệu hệ thống xây dựng và quản lý chatbot. Collection Bot sẽ tham chiếu _id tới tất cả collection còn lại, và để hình vẽ trực quan quan hơn em sẽ chỉ vẽ liên kết giữa bot với các collection quan trọng.



Hình 19 Thiết kế tổng quan cơ sở dữ liệu hệ thống xây dựng và quản lý chatbot

Mô tả chi tiết về các collection intent, action, node em sẽ trình bày trong Chương 5, các collection còn lại em sẽ trình bày mô tả chi tiết ở phần phụ lục.

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Trong quá trình xây dựng đồ án em có sử dụng các công cụ hỗ trợ xây dựng và phát triển frontend, backend như trong **Bảng 14**.

Bảng 14 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
Tìm kiếm ý định	Elasticsearch	https://www.elastic.co
Thư viện xây dựng frontend	ReactJS	https://ReactJS.org
Thư viện React Component	Material-ui	https://material-ui.com/
Thư viện hỗ trợ xây dựng biểu đồ luồng	React diagram	https://github.com/projectstorm/react-diagrams
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

4.3.2 Kết quả đạt được

Sau khi nghiên cứu và triển khai, em đã phát triển được hệ thống xây dựng và quản lý chatbot theo luật và đã thực hiện kết nối tới dịch vụ livechat để hỗ trợ tích hợp với các ứng dụng nhắn tin. Các chức năng chính của hệ thống là: (i) Quản lý bot, quản lý ý định, (ii) Quản lý thực thể, (iii) Quản lý hành động, (iv) Quản lý kịch bản. Các thống kê về ứng dụng được trình bày ở trong **Bảng 15**.

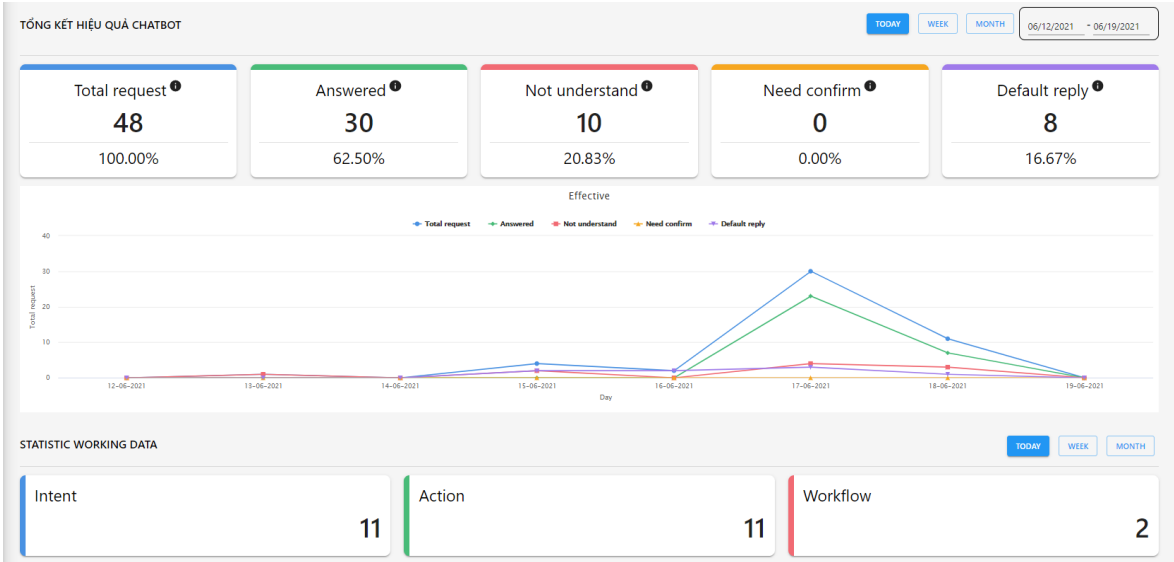
Bảng 15 Thống kê ứng dụng

Thông tin	Thống kê
Số lớp tronng mã nguồn (file jsx)	73 lớp
Số gói trực tiếp chứa mã nguồn	30 gói
Số lượng file js react	57 file
Số lượng file backend	127 file
Số bảng CSDL được sử dụng trong hệ thống	15 bảng

4.3.3 Minh hoạ các chức năng chính

Trong phần này em sẽ minh hoạ một số chức năng chính của hệ thống xây dựng và quản lý chatbot theo luật. Cụ thể là các giao diện (i) thống kê, (ii) tạo intent, (iii) tạo action.

4.3.3.1 Thống kê

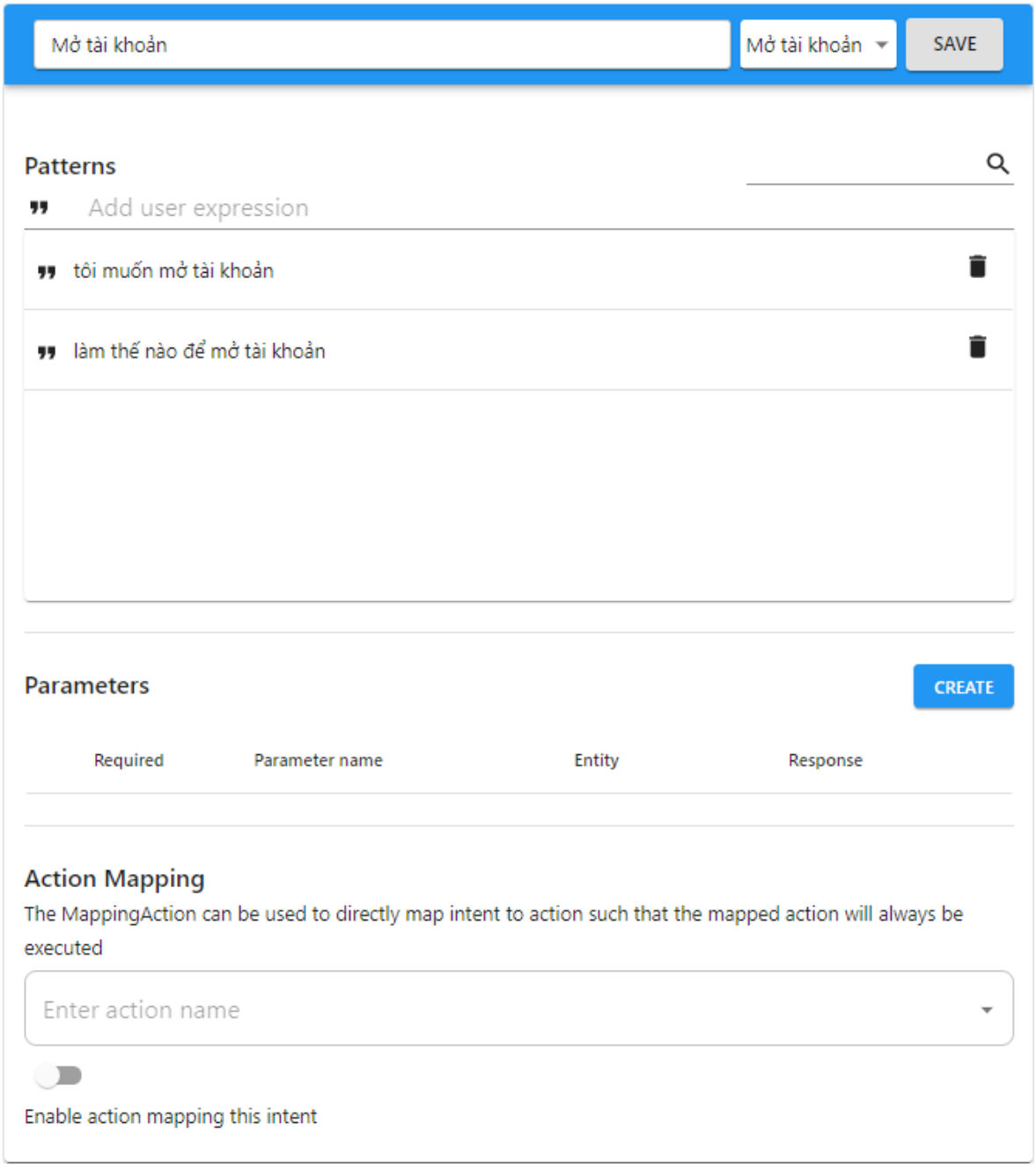


Hình 20 Giao diện thống kê

Hình 20 là giao diện thống kê của hệ thống xây dựng chatbot theo luật. Khi truy cập trang web này người dùng có thể xem báo cáo số liệu về sự tương tác giữa khách hàng và chatbot, ở đây người dùng có thể xem thống kê theo tiêu giữa ngày, tuần, tháng và năm. Các thông tin thống kê sẽ bao gồm tổng số tin nhắn mà khách hàng gửi đến theo bộ lọc thời gian, số

yêu cầu bot trả lời thành công, số yêu cầu bot không hiểu, số yêu cầu bot cần xác nhận hỏi lại, số yêu cầu bot trả lời mặc định.

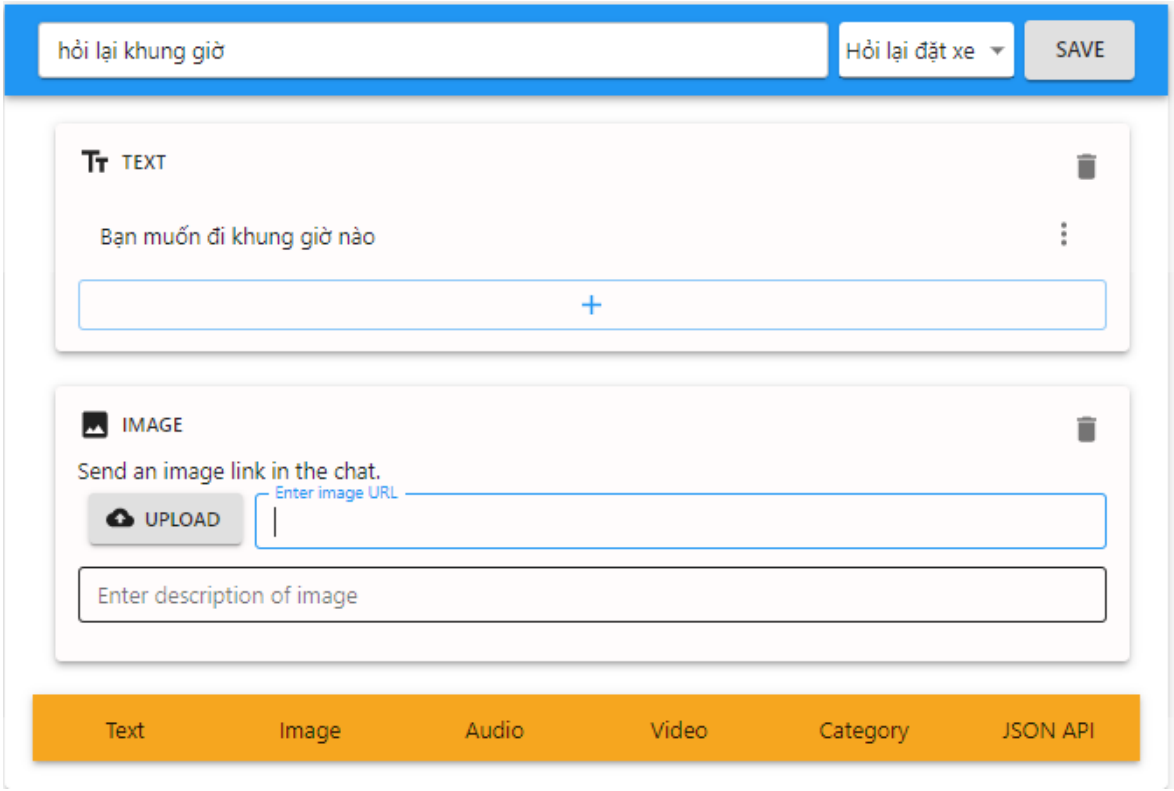
4.3.3.2 Tạo intent



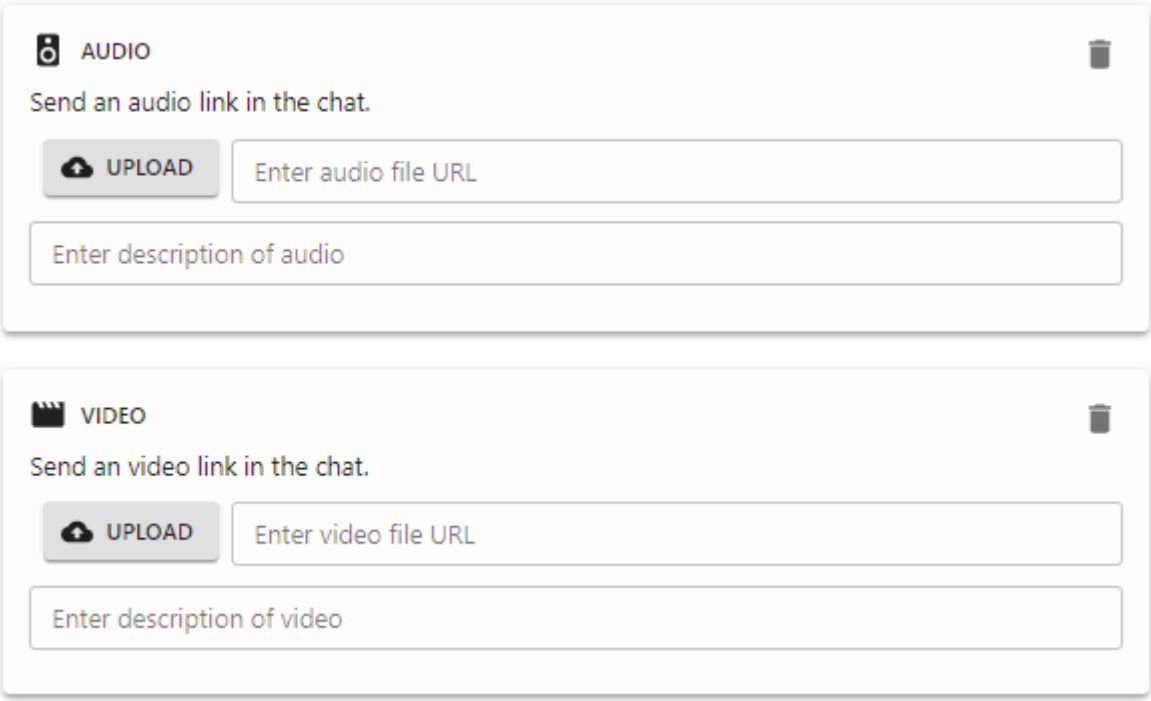
Hình 21 Giao diện tạo intent

Hình 21 là giao diện tạo intent của hệ thống. ở đây người dùng có thể thêm các mẫu câu, thêm các tham số để trích xuất thông tin, cấu hình phản hồi khi khách hàng gửi nội dung tin nhắn không đủ thông tin, và cài đặt action đi kèm với intent này nếu cần.

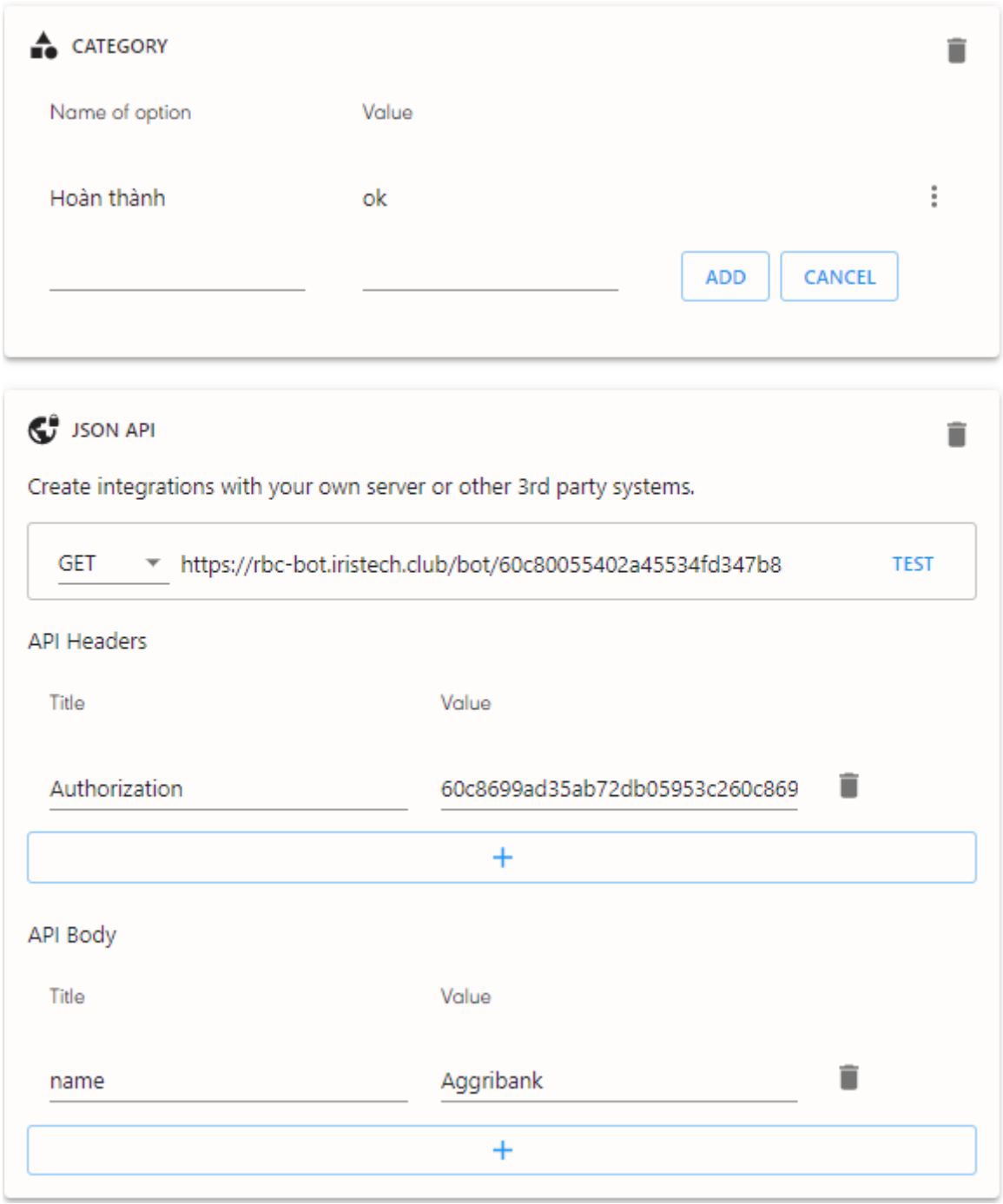
4.3.3.3 Tạo action



Hình 22 Màn hình tạo action với thẻ Text và Image



Hình 23 Màn hình tạo action với thẻ Audio và Video



Hình 24 Màn hình tạo action với thẻ Json API và Category

Hình 22 là giao diện tạo action của hệ thống với hai thẻ Text và Image, trong đó thẻ Text sẽ hỗ trợ nhập các phản hồi của bot bằng tin nhắn văn bản, thẻ Image hỗ trợ nhập các phản hồi của bot bằng hình ảnh và mô tả hình ảnh đó.

Hình 23 là giao diện tạo action với thẻ Audio và Video, trong đó thẻ Audio hỗ trợ nhập phản hồi của bot bằng âm thanh, thẻ Video hỗ trợ nhập phản hồi của bot bằng video.

Hình 24 là giao diện tạo action với thẻ Json API và Category, trong đó thẻ Category sẽ hỗ trợ phản hồi của bot bằng cách đưa ra các lựa chọn cho khách hàng, người dùng có thể tuỳ ý nhập vào các lựa chọn và giá trị thích hợp. Thẻ Json API là thẻ hỗ trợ bot gọi API để lấy thông tin từ một hệ thống khác, thẻ cung cấp các nút chọn phương thức, nhập đường dẫn, nhập các header và body cần thiết, và có thể gọi thử API bằng cách nhấn nút TEST để cài đặt các tham số nhận được từ thông tin API trả về.

4.4 Kiểm thử

4.4.1 Kiểm tra tính tương thích

Bảng 16 Thống kê kiểm thử tương thích.

Thiết bị	Thông số kỹ thuật	Giao diện	Chức năng
Asus gl553	Màn hình: 15.6’’ HD, 16GB RAM	Đạt	Đạt
HP pavilion 15	Màn hình: 15.6’’ FHD, 9GB RAM	Đạt	Đạt

4.4.2 Kiểm thử hộp đen

Em sẽ sử dụng phương pháp kiểm thử hộp đen dựa trên các tính năng trên giao diện hiển thị, trong **Bảng 17**, em sẽ trình bày 39 test case mà em đã tiến hành kiểm thử.

Bảng 17 Danh sách các test case

STT	Màn hình	Tên test case
1.	Quản lý bot	Kiểm tra hiển thị thống kê hiệu quả bot theo thời gian
2.		Kiểm tra hiển thị thống kê dữ liệu của bot
3.		Kiểm tra tạo bot
4.		Kiểm tra xem thông tin của bot
5.		Kiểm tra thêm người dùng xây dựng và quản lý bot
6.		Kiểm tra hiển thị thông tin bot

STT	Màn hình	Tên test case
7.		Kiểm tra xoá bot
8.	Quản lý intent	Kiểm tra hiển thị thông tin intent
9.		Kiểm tra cập nhật thông tin intent
10.		Kiểm tra hiển thị danh sách nhóm intent
11.		Kiểm tra thêm nhóm intent
12.		Kiểm tra cập nhật nhóm intent
13.		Kiểm tra xoá intent
14.		Kiểm tra tạo intent
15.		Kiểm tra tìm kiếm intent và nhóm intent
16.		Kiểm tra xoá nhóm intent
17.	Quản lý entity	Kiểm tra hiển thị thông tin entity
18.		Kiểm tra cập nhật thông tin entity
19.		Kiểm tra hiển thị danh sách nhóm entity
20.		Kiểm tra thêm nhóm entity
21.		Kiểm tra cập nhật nhóm entity
22.		Kiểm tra xoá entity
23.		Kiểm tra tạo entity
24.		Kiểm tra tìm kiếm entity và nhóm entity
25.		Kiểm tra xoá nhóm entity
26.		Kiểm tra hiển thị thông tin action

STT	Màn hình	Tên test case
27.	Quản lý action	Kiểm tra cập nhật thông tin action
28.		Kiểm tra hiển thị danh sách nhóm action
29.		Kiểm tra thêm nhóm action
30.		Kiểm tra cập nhật nhóm action
31.		Kiểm tra xoá action
32.		Kiểm tra tạo action
33.		Kiểm tra tìm kiếm action và nhóm action
34.		Kiểm tra xoá nhóm action
35.	Quản lý kịch bản	Kiểm tra hiển thị danh sách kịch bản
36.		Kiểm tra tạo mới kịch bản
37.		Kiểm tra cập nhật kịch bản
38.		Kiểm tra xoá kịch bản
39.		Kiểm tra thiết kế kịch bản

4.5 Triển khai

Hiện tại hệ thống và giao diện đã được xây dựng, triển khai lên server tại tên miền https://rbc-bot.iristech.club. Các thông tin cụ thể liên quan đến cấu hình server triển khai được mô tả trong **Bảng 18**.

Bảng 18 Thông số cấu hình server triển khai hệ thống

Tên cấu hình	Thông số
Hệ điều hành	Ubuntu 16.04

Tên cấu hình	Thông số
CPU	Intel Xeon Gold 5120 2.20GHz
RAM	200MB

Chương 5 Các giải pháp và đóng góp nổi bật

Chương 5 em sẽ trình bày những điều tâm đắc nhất của em trong quá trình thực hiện đồ án này. Cụ thể là giải pháp thiết kế chatbot theo luật, giải pháp xác định ý định người dùng với Elasticsearch.

5.1 Giải pháp xây dựng và quản lý chatbot theo luật

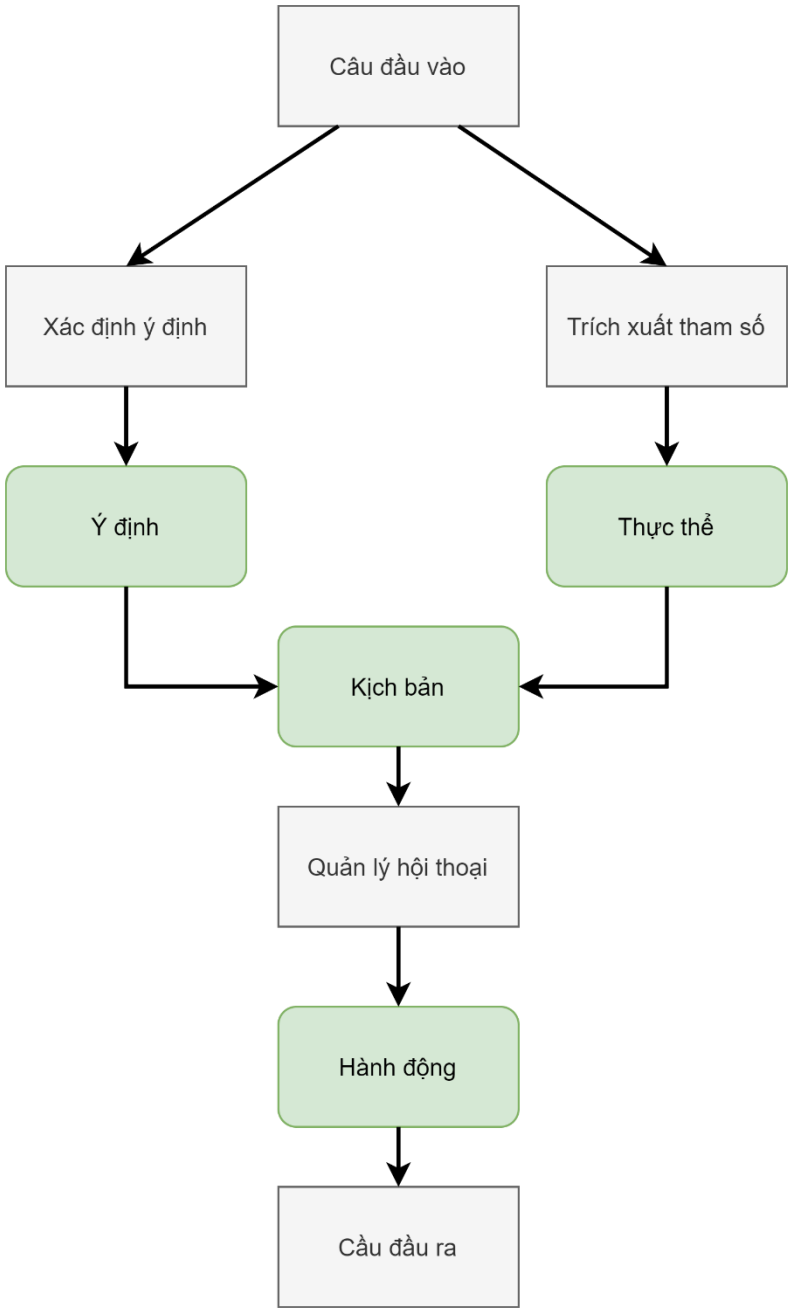
5.1.1 Vấn đề

Hiện nay quy trình tương tác giữa bot và khách hàng thường diễn ra như sau: Khách hàng gửi tin nhắn, bot sẽ kiểm tra nội dung tin nhắn và tìm kiếm trong tập dữ liệu, trích xuất thông tin và trả về phản hồi tương ứng với dữ liệu tìm được cho khách hàng, sau đó khách hàng có thể tiếp tục tương tác với bot lặp đi lặp lại như vậy cho đến khi nhận đủ thông tin.

Từ quy trình như vậy, hệ thống xây dựng chatbot theo luật cần xây dựng được cơ sở dữ liệu có thể quản lý được các câu hỏi mẫu mà khách hàng sẽ hỏi, quản lý được các được các kiểu trích xuất thông tin từ nội dung câu hỏi, quản lý được các phản hồi của tới khách hàng, và đặc biệt có thể quản lý được các kịch bản mà bot sẽ tương tác với người dùng. Đặc biệt làm giảm thiểu thời gian xây dựng dữ liệu người dùng ít nhất có thể và hỗ trợ người dùng quản lý tập trung các dữ liệu giống nhau về một mặt nội dung nào đó.

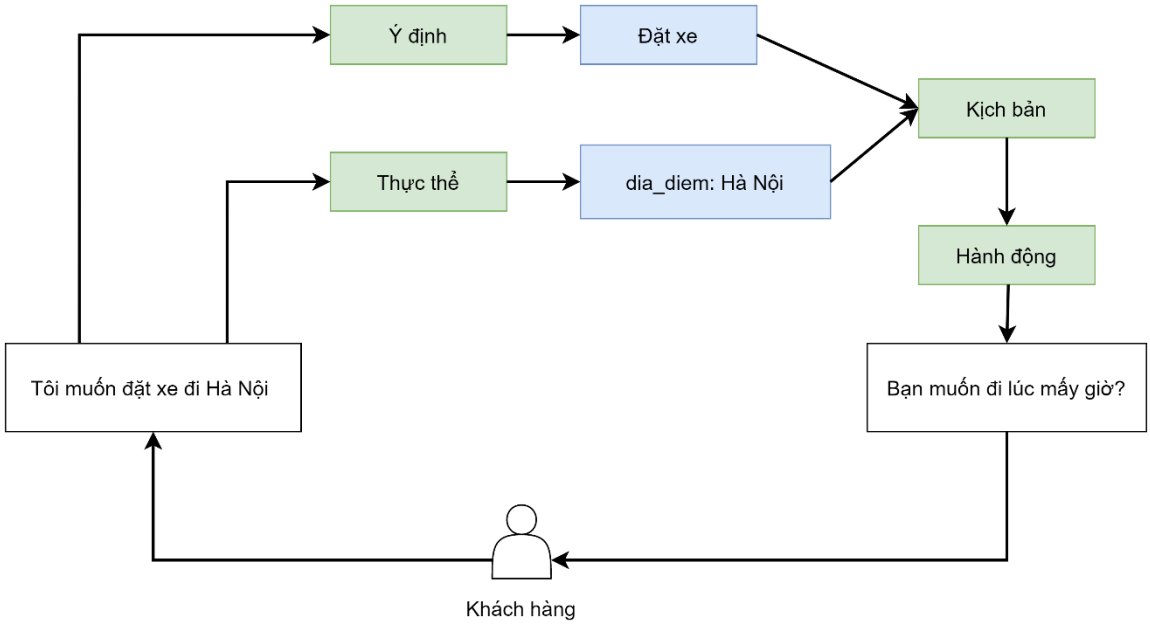
5.1.2 Giải pháp tổng quan

Từ những vấn đề trên em đã thiết kế mô hình kiến trúc hệ thống xây dựng và quản lý chatbot trong **Hình 25.** Trong hình các thành phần màu xanh sẽ tương ứng với các đối tượng dữ liệu mà hệ thống cần xây dựng và quản lý. Khi khách hàng gửi câu đầu vào, hệ thống sẽ xác định ý định khách hàng dựa trên tập ý định trong cơ sở dữ liệu, trích xuất thông tin từ câu đầu vào bằng thực thể, sau đó thành phần quản lý hội thoại có nhiệm vụ xác định bối cảnh kịch bản nhắn tin. Khi có ý định, thông tin đầu vào và kịch bản thành phần quản lý hội thoại sẽ kiểm tra điều kiện giá trị thông tin đầu vào, kế tiếp là lựa chọn hành động phù hợp tới khách hàng.



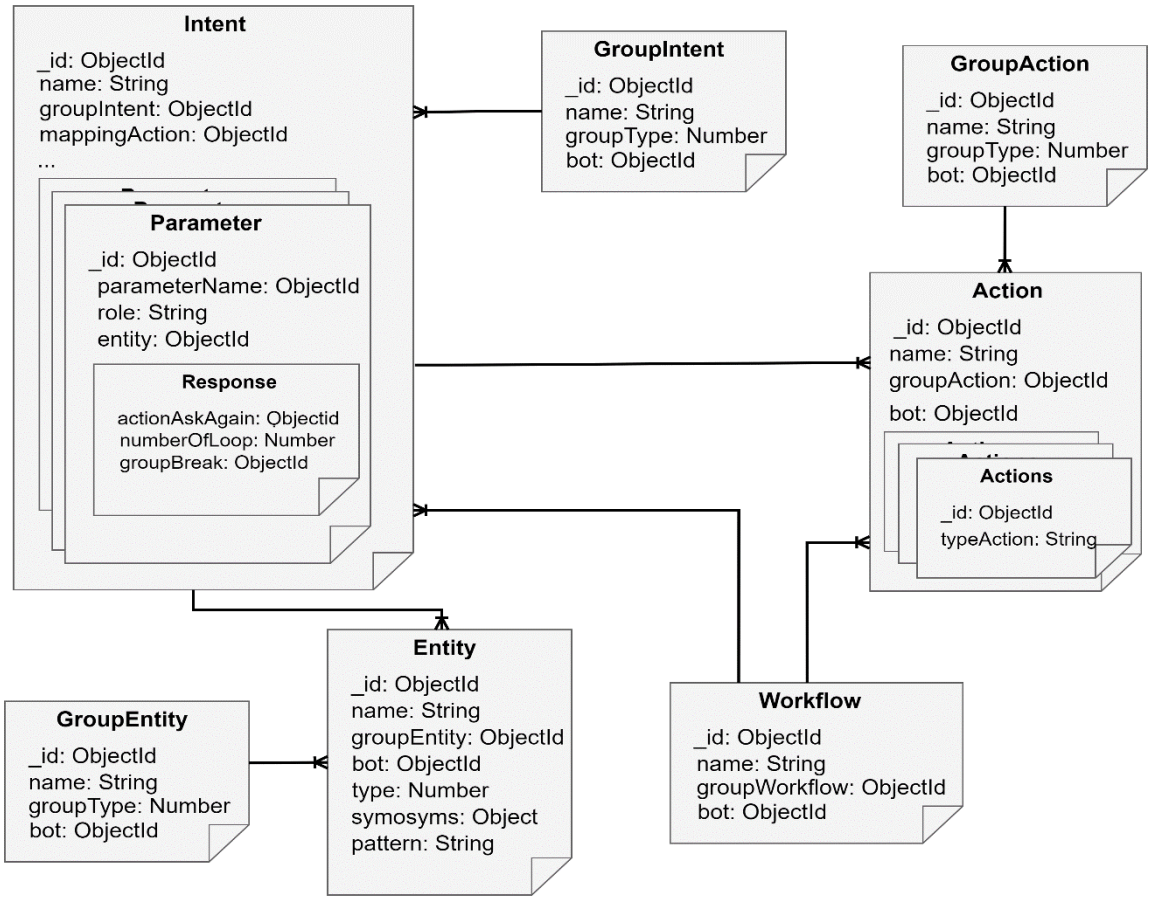
Hình 25 Mô hình kiến trúc hệ thống chatbot

Hình 26 là hình mô tả các dữ liệu trong kiến trúc hệ thống chatbot dưới một kịch bản hội thoại đặt xe. Khách hàng sẽ gửi một câu đầu vào "Tôi muốn đặt xe đi Hà Nội", hệ thống xác định được: ý định (Đặt xe), thực thể (Hà Nội). Sau đó xác định được kịch bản và kiểm tra điều kiện, hệ thống xe kiểm tra điểm đến (Hà Nội) và trả về các hành động thích hợp ở đây là "Bạn muốn đi lúc mấy giờ" cho khách hàng, sau đó là chuỗi tin nhắn hội thoại theo kịch bản giữa bot và khách hàng.



Hình 26 Mô tả các dữ liệu trong một kịch bản nhắn tin

5.1.3 Thiết kế dữ liệu



Hình 27 Thiết kế cơ sở dữ liệu cho các thành phần cơ bản của chatbot

Hình 27 là kết quả thiết kế dữ liệu cho các thành phần cơ bản của chatbot, bao gồm 7 collection Intent, GroupIntent, Entity, GroupEntity, Workflow, Action, GroupAction. Dưới đây là thiết kế chi tiết các collection Intent, Action các mô tả chi tiết về collection Workflow, Node em sẽ trình bày ở mục 5.4.2.3, các collection còn lại em sẽ trình bày chi tiết ở phần phụ lục.

Thiết kế chi tiết collection Action

```
{
    name: String,               // tên action
    actions: [                  // danh sách các thẻ action
      {
        typeAction: String,     // loại action TEXT, MEDIA, CATEGORY, API, LOOP
        text: [String],
        media: {
          typeMedia: String,    // IMAGE, VIDEO, AUDIO
          url: String,          // đường dẫn
          description: String,  // mô tả
        },
        options: [              // action kiểu lựa chọn
          {
            name: String,       // tên lựa chọn
            value: String,      // giá trị
          },
        ],
        api: {                  // action gọi api
          method: String,       // phương thức GET, POST
          url: String,          // đường dẫn
          headers: [
            {
              title: String,    // tên tiêu đề
              value: String,    // giá trị
            },
          ],
          body: [               // nội dung
            {
              title: String,    // tên trường
              value: String,    // giá trị
            },
          ],
          response: {},         // kết quả nhận được từ api
          parameters: [         // tham số
            {
              slot: {
                type: ObjectId,
                ref: 'Slot',
              },
              name: String,    // tên tham số
              value: String,   // giá trị
            },
          ],
        },
      },
    ],
    groupAction: {              // ID group action
      type: ObjectId,
      ref: 'GroupAction',
    },
    bot: {                      // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
    createBy: {                 // người tạo
      type: ObjectId,
      ref: 'User',
    },
}
```

Thiết kế chi tiết collection Intent

```
{
    id: ObjectId,                // ID của ý định
    name: String,                // tên ý định
    patterns: [String],          // danh sách các câu mẫu
    isMappingAction: Boolean,    // kết nối với hành động hay không
    mappingAction: {             // ID cau hành động được kết nối
      type: ObjectId,
      ref: 'Action',
    },
    parameters: [                // danh sách tham số trích xuất
      {
        parameterName: String,   // tên tham số trích xuất
        required: Boolean,       // yêu cầu trích xuất tham số này
        entity: {                // ID của thực thể
          type: ObjectId,
          ref: 'Entity',
        },
        response: {              // phản hồi của bot khi không trích xuất được thông tin
          actionAskAgain: {      // ID của hành động hỏi lại
            type: ObjectId,
            ref: 'Entity',
          },
          numberOfLoop: Number,  // số lần hỏi lại
          actionBreak: {         // ID của hành động khi quá số lần hỏi lại
            type: ObjectId,
            ref: 'Entity',
          },
        },
      },
    ],
    groupIntent: {                // ID nhóm ý định
      type: ObjectId,
      ref: 'GroupIntent',
    },
    bot: {                        // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
    createBy: {                   // ID của người tạo
      type: ObjectId,
      ref: 'User',
    },
}
```

5.2 Xác định ý định người dùng với Elasticsearch

5.2.1 Vấn đề

Thông thường, khi khách hàng truy cập hệ thống chatbot khách hàng thường mong muốn hệ thống sẽ đưa ra những hành động hỗ trợ mình về một vấn đề nào đó. Để việc phản hồi được chính xác hệ thống cần xác định chính xác ý định (intent) đó của khách hàng. Việc này sẽ quyết định đoạn hội thoại sau đó giữa khách hàng và chatbot sẽ diễn ra như thế nào. Vì vậy bài toán xác định ý định của khách hàng đóng vai trò rất quan trọng trong hệ thống chatbot theo luật.

Hiện nay các hệ cơ sở dữ liệu chỉ hỗ trợ truy vấn dạng LIKE để tìm kiếm. Khi sử dụng truy vấn LIKE để tìm kiếm ý định thì kết quả trả về sẽ là những ý định chỉ chứa giá trị đầu vào, chưa kể nội dung tin nhắn của khách hàng gửi đến rất đa dạng, khách hàng có thể thêm, bớt hoặc nhập sai từ trong tin nhắn. Nếu làm như vậy người dùng sẽ phải xây dựng dữ liệu ý

định (intent) rất nhiều mà chỉ khác nhau bởi một hoặc nhiều từ, điều này làm mất thời gian và trùng lặp dữ liệu.

5.2.2 Giải pháp

Với những vấn đề nêu ở mục 5.2.1, em sẽ sử dụng công cụ search engine là Elasticsearch để giải quyết vấn đề ở trên.

Khi tìm kiếm, Elasticsearch trả về cho chúng ta ngoài cá kết quả tìm được, còn có đánh giá độ liên quan(relevance) của kết quả dựa trên giá trị thực dương _score. ES sẽ sắp xếp các kết quả của query theo thứ tự _score giảm dần. Đây chính là điểm có thể giải quyết vấn đề ở mục 5.2.1.

Thuật toán đánh giá được sử dụng rất phổ biến trong Elasticsearch có tên gọi là term frequency/inverse document frequency, gọi tắt là TF/IDF, bao gồm các yếu tố đánh giá: Term frequency, Inverse document frequency, Field-length norm.

Term frequency – yếu tố này đánh giá tần suất xuất hiện của term trong field. Càng xuất hiện nhiều, relevance càng cao. Và dĩ nhiên một field mà từ khoá xuất hiện 5 lần sẽ cho relevance cao hơn là field mà từ khoá chỉ xuất hiện một lần.

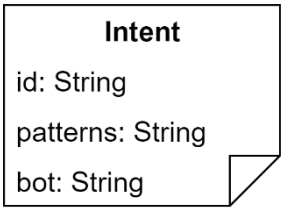
Inverse document frequency – yếu tố này đánh giá tần suất xuất hiện của term trên toàn bộ index (tương đương với database trong mysql) . Điểm đánh giá ở đây là càng xuất hiện nhiều càng ít thích hợp.

Field-length-norm – yếu tố này đánh giá độ dài của field. Field càng ngắn, thì term sẽ có giá trị càng cao và ngược lại.

Số điểm cuối cùng là tích các kết quả của ba thuật toán trên. Như vậy khi sử dụng elasticsearch em có thể tìm kiếm những ý định của khách hàng một cách dễ dàng, ngay cả khi khách hàng nhập sai từ, thiếu dấu, thừa hoặc ít từ hơn thì elasticsearch vẫn cho ra câu mẫu tìm được liên quan nhất có thể.

5.2.3 Kết quả

Với giải pháp sử dụng Elasticsearch để tìm kiếm ý định, mỗi lần tạo mới và cập nhật intent thì intent sẽ được lưu trữ trong elasticsearch theo cấu trúc trong **Hình 28**.



Hình 28 Cấu trúc lưu trữ intent trong elasticsearch

Trong đó id là id của ý định, patterns là tập các câu mẫu mà người dùng quy định, bot là id của bot quản lý intent này.

5.3 Xác định thực thể

5.3.1 Vấn đề

Bên cạnh việc xác định ý định của khách hàng trong hội thoại, chúng ta cần trích xuất thông tin cần thiết trong đó. Ví dụ khách hàng muốn đặt vé xe, hệ thống cần biết thông tin về địa điểm khởi hành và địa điểm khách muốn đến, ngày và giờ xuất phát. Các thông tin cần trích xuất trong nội dung tin nhắn thường phải là các thực thể thuộc một loại nào đó.

5.3.2 Giải pháp

Để tiếp cận bài toán trích xuất dữ liệu ở mục 5.3.1 em sẽ sử dụng Regular expression (regex) và từ đồng nghĩa để trích xuất thông tin.

Trong đó Regular expression là biểu thức chính quy được dùng để xử lý chuỗi nâng cao thông qua biểu thức của riêng nó, những biểu thức này có nguyên tắc riêng và khi sử dụng phải tuân theo nguyên tắc này thì mới hoạt động được. Nguyên tắc hoạt động của biểu thức regex là so khớp dựa vào khuôn mẫu, khuôn mẫu được xây dựng từ các quy tắc căn bản của biểu thức. Dựa vào các nguyên tắc của regex người dùng có thể tự định nghĩa các thực thể cho riêng mình.

Thực thể từ đồng nghĩa là sử dụng từ đầu vào tìm kiếm được trong câu hội thoại và quy đổi ra từ đồng nghĩa khác được quy định bởi người dùng, trong đó đầu vào có thể là danh sách các từ được cài đặt trong thực thể.

5.3.3 Kết quả

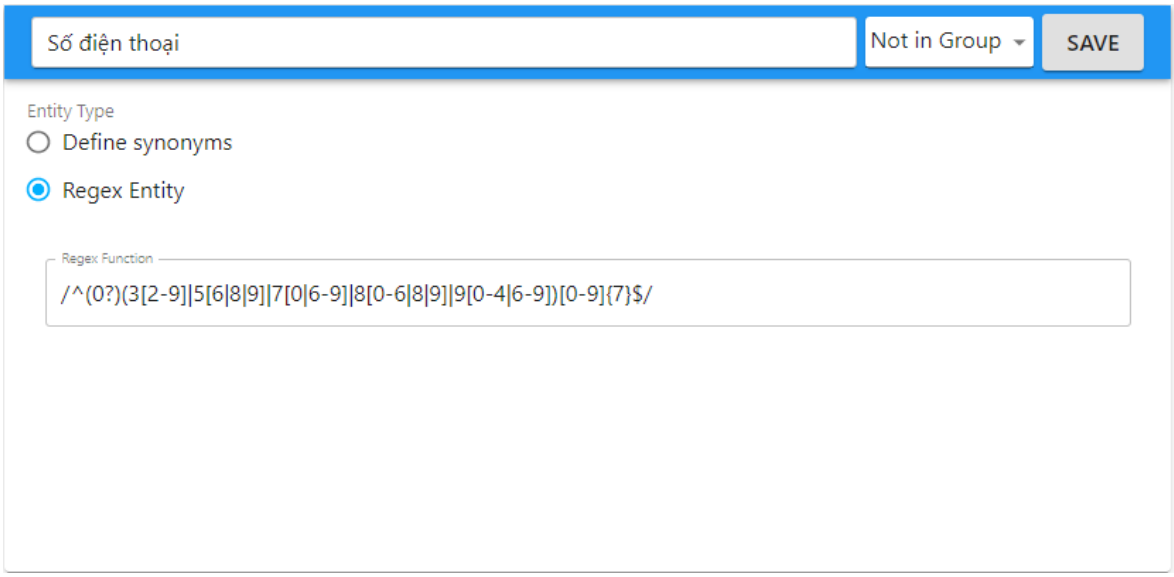
So sánh sử dụng Regex và thuật toán so khớp chuỗi (knuth-Morris-Pratt), cả hai phương pháp đều có điểm chung là tìm kiếm sự xuất hiện của một từ trong một xâu văn bản, trong đó thuật toán so khớp xâu có độ phức tạp o(n) với n là độ dài xâu văn bản. Để trực quan hơn em sẽ kiểm thử hiệu năng của hai phương pháp trong **Bảng 19** với bài toán đầu vào là một

xâu văn bản có độ dài m và một danh sách có n từ, bài toán sử đầu ra là từ đầu tiên trong n từ đó xuất hiện trong xâu.

Bảng 19 Kiểm thử và so sánh hiệu năng của phương pháp regex và so khớp chuỗi

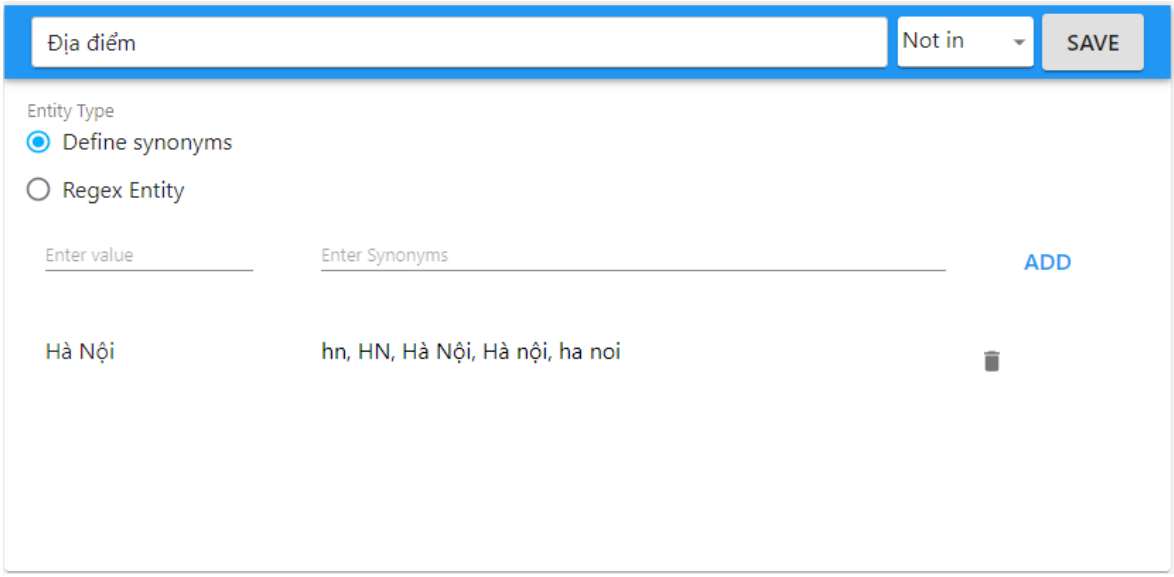
Độ dài xâu văn bản	Độ dài danh sách các từ	Kết quả		Thời gian xử lý (ms)	
		Regex	So khớp chuỗi	Regex	So khớp chuỗi
1.000	1.000	Đạt	Đạt	0,15	7
1.0000	10.000	Đạt	Đạt	1,20	630
100.000	100.000	Đạt	Đạt	12,00	120.000

Nhìn vào kết quả kiểm thử ta có thể thấy hiệu năng của regex rất tốt, hơn nữa nhược điểm của so khớp xâu là từ đầu vào phải rõ ràng, vì vậy ta cần xây dựng n rất lớn nhưng nếu sử dụng các nguyên tắc của Regex chúng ta có thể viết rất ngắn gọn. **Hình 29** là giao diện xây dựng thực thể bằng Regex. Mục đích xác định thực thể này là xác định số điện thoại trong câu nói của khách hàng. Trong đó có các nguyên tắc về độ dài bắt buộc, các đầu số hợp lệ của số điện thoại.



Hình 29 Giao diện định nghĩa thực thể bằng Regex.

Hình 30 là giao diện xây dựng thực thể bằng từ đồng nghĩa. Trong đó các từ có thể xuất hiện trong câu nói của khách hàng như HN, Hà Nội, ha noi sẽ được quy đổi ra giá trị Hà Nội.



Hình 30 Giao diện định nghĩa thực thể bằng từ đồng nghĩa.

5.4 Phát triển công cụ quản lý kịch bản nhắn tin

5.4.1 Vấn đề

Trong hệ thống xây dựng chatbot theo luật, kịch bản có thể coi là phần khá quan trọng, nó giúp điều hướng luồng hội thoại theo cài đặt của người dùng. Vì vậy để người dùng có thể nhanh chóng dễ dàng xây dựng được kịch bản thì công cụ quản lý kịch bản cần đáp ứng được các yêu cầu sau.

Giao diện của kịch bản cần trực quan, dễ hình dung. Bất kỳ ai khi xem kịch bản hội thoại cũng hiểu luồng và nội dung của kịch bản. Khi số lượng node tăng lên, thì giao diện vẫn phải thiết kế đảm bảo để người dùng vẫn có thể kiểm soát được luồng hội thoại. Sự tương tác với các thành phần trong giao diện phải đảm bảo sự linh hoạt, dễ dùng và tiện lợi.

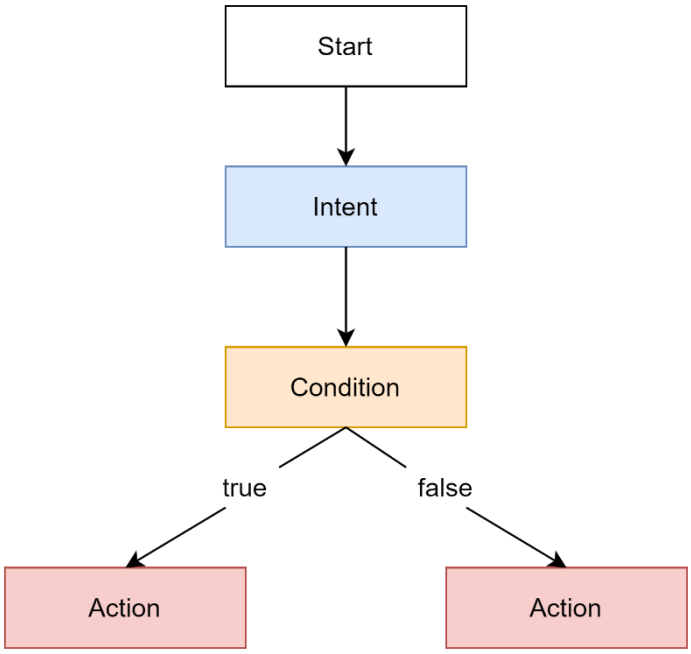
5.4.2 Giải pháp

Từ những yêu cần nêu ở mục 5.4.1 em sẽ phát triển công cụ quản lý kịch bản hội thoại để giải quyết vấn đề đó.

5.4.2.1 Thiết kế kịch bản và lựa chọn thư viện phù hợp

Để giải quyết vấn đề trên em đã thiết kế công cụ quản lý kịch bản theo biểu đồ luồng (Data Flow Diagram, viết tắt là DFD). DFD là một mô hình hệ thống cân xứng cả dữ liệu và tiến trình. Nó chỉ ra cách thông tin vận chuyển từ tiến trình này sang một tiến trình khác. Điều quan trọng nhất là các thông tin nào cần phải có trước khi cho một tiến trình cần thực hiện.

Trong công cụ thiết kế kịch bản hội thoại, DFD giúp người dùng có thể phân tích, thiết kế, biểu đạt theo nội dung yêu cầu của kịch bản. Ngoài ra khi nhìn vào DFD bất kì ai nhìn cũng có thể tham gia đóng góp, chỉnh sửa kịch bản, cho dù họ không phải người xây dựng kịch bản từ đầu.



Hình 31 Mô hình kịch bản chatbot cơ bản trong DFD

Các thành phần trong DFD được biểu diễn trong **Hình 31**. Trong đó bao gồm bốn node: start, intent, condition, action. Node start đóng vai trò là nơi bắt đầu của kịch bản, node intent thể hiện ý định của khách hàng khi vào kịch bản, node condition có nhiệm vụ lấy các thông tin trước đó(các tham số trong intent) mà khách hàng cung cấp trước đó để so sánh và đưa ra hướng đi phù hợp với kết quả so sánh. Trong node condition để lấy lấy các thông tin trước đó em sẽ sử dụng thuật toán duyệt theo chiều sâu(Depth-first search) đề lấy các thông tin, quá trình duyệt sẽ dừng lại khi gặp node start. Node action có nhiệm vụ biểu diễn các phản hồi của bot tới người dùng. Giữa các node sẽ được nối với nhau bằng các link có chưa mũi tên để biểu thị luồng dịch chuyển thông tin.

Để vẽ được biểu đồ này em lựa chọn thư viện React Diagram. Đây là thư viện rất hữu ích để giúp tạo ra biểu đồ luồng hội thoại. Thư viện hỗ trợ chỉnh sửa màu sắc, kích thước, hình dạng các hình khối, các liên kết trong biểu đồ. Hơn nữa, thư viện cũng cung cấp các tài liệu hướng dẫn, tài liệu API đầy đủ để tuỳ chỉnh các sự kiện như chọn chuột, di chuột và ấn nút bàn phím, …. Từ đó em có thể thiết kế giao diện trực quan, cài đặt sự tương tác giữa các thành phần trong giao diện một cách linh hoạt, dễ dàng sử dụng.

5.4.2.2 Thiết kế các node

Trong kịch bản hội thoại sẽ có các node để biểu thị, mỗi node sẽ thể hiện vai trò riêng trong luồng kịch bản. Để tuân theo quy trình hệ thống chatbot theo luật em sẽ thiết kế ba loại node cho kịch bản như sau: node intent là node thể hiện ý định người dùng, node condition để thể hiện sự so sánh các thông tin trích xuất được từ trước với giá trị người dùng cấu hình và node action thể hiện sự phản hồi của bot tới khách hàng. Các thông tin chi tiết của từng node sẽ được mô tả trong **Bảng 20**.

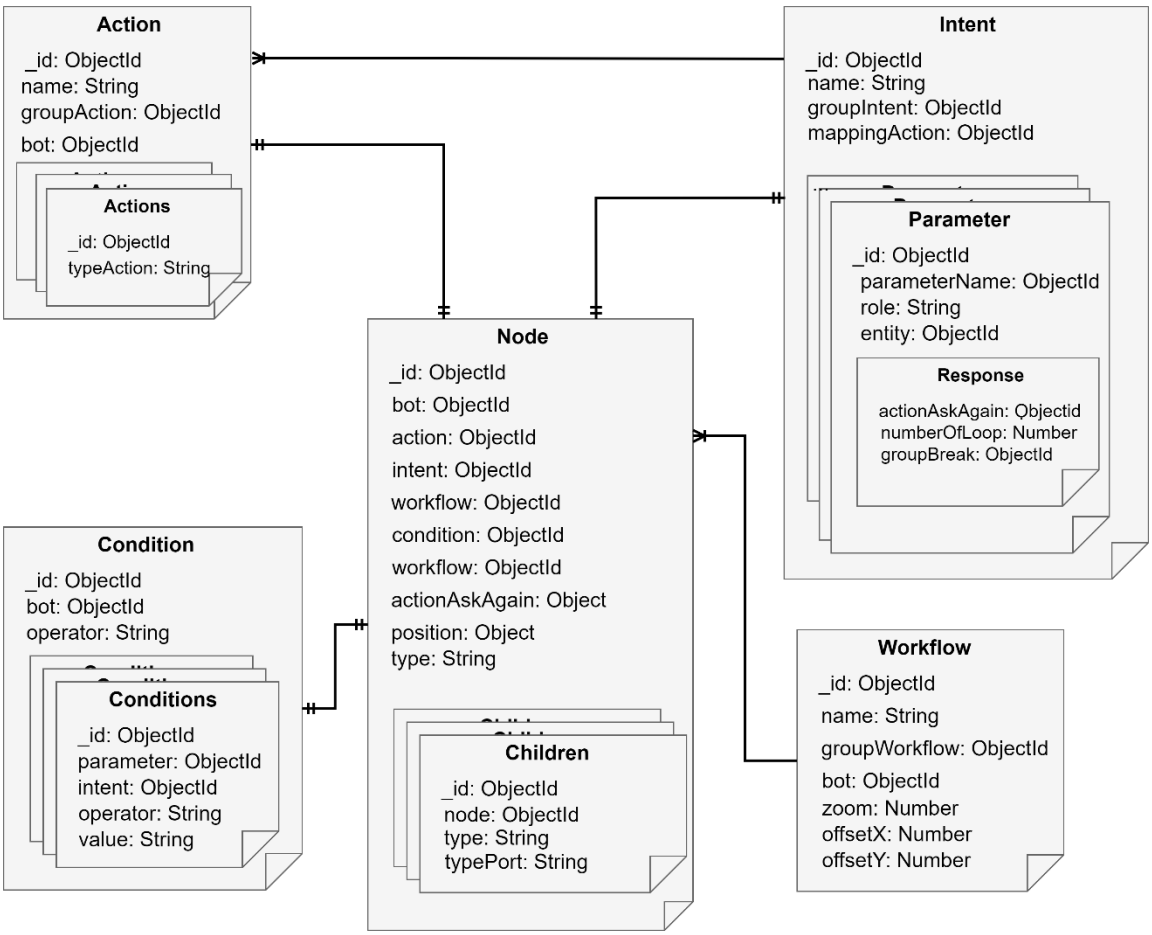
Bảng 20 Bảng mô tả chi tiết các node trong kịch bản hội thoại.

Node	**Trường dữ liệu**	**Mô tả**
Intent	name	Là ô select giúp người dùng chọn các ý định(intent) đã được tạo từ trước.
Condition	parameter	Tên tham số được trích xuất từ trước.
	subOperator	Ô chọn toán tử so sánh
	value	Giá trị dùng để so sánh với giá trị tham số
	operator	Toán tử nối các điều kiện con với nhau (nếu có nhiều hơn một điều kiện con)
Action	name	Là ô select giúp người dùng chọn các hành động(action) đã được tạo từ trước
	actionAskAgain	Ô select giúp người dùng chọn hành động hỏi lại khi khách hàng gửi sai ý định trong kịch bản
	numberOfLoop	Số lần hỏi lại
	Action when too number loop	Ô select giúp người dùng chọn hành động khi quá số lần hỏi lại

5.4.2.3 Thiết kế cơ sở dữ liệu quản lý kịch bản

Mặc dù sử dụng hệ cơ sở dữ liệu dạng NoSql. Nhưng do việc tạo và cập nhật dữ liệu thường xuyên và hơn nữa để việc một dữ liệu có thể sử dụng được ở nhiều nơi nên e sẽ thiết kế cơ sở dữ liệu quản lý kịch bản như sau: collection node sẽ chứa workflowId, như vậy khi lấy

tất cả dữ liệu của một kịch bản, em sẽ phải join các bảng workflow, node, intent, condition và action. **Hình 32** sẽ minh hoạ cơ sở dữ liệu em đã thiết kế.



Hình 32 Thiết kế cơ sở dữ liệu quản lý kịch bản hội thoại

Do các collection intent, action đã được em mô tả ở mục 5.1.3 nên dưới đây em sẽ chỉ mô tả chi tiết collection node, hai collection condition, workflow em sẽ trình bày ở phần phụ lục.

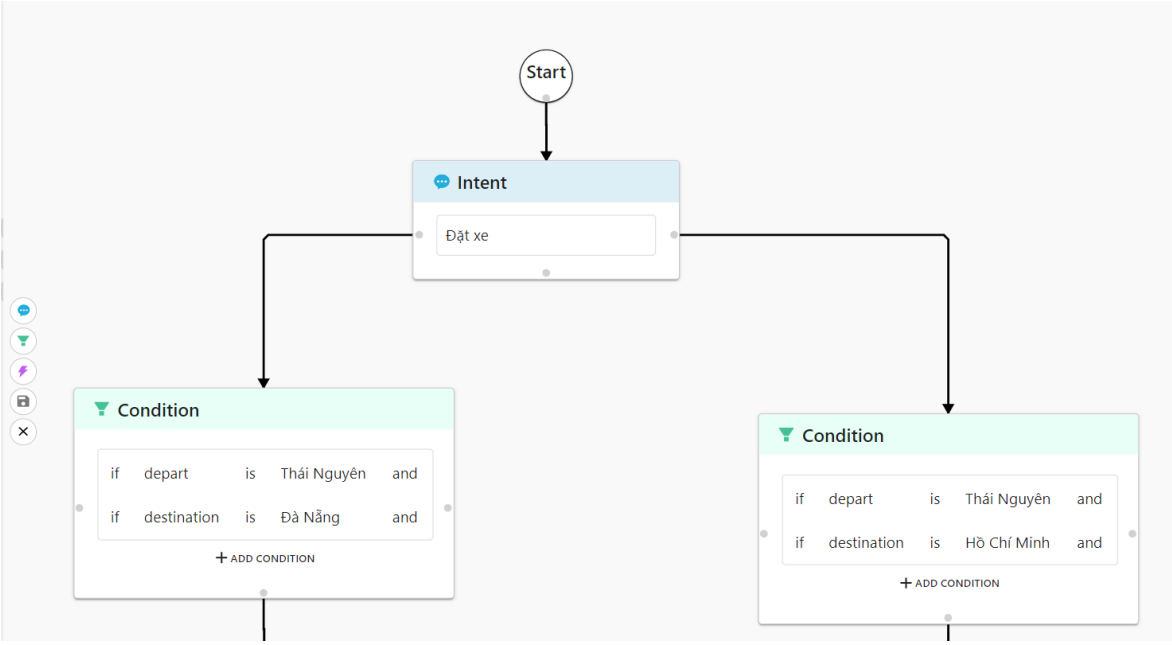
Thiết kế chi tiết collection Node

```
{
    type: String,              // loại node
    intent: {                  // ID của intent nếu loại node là intent
      type: ObjectId,
      ref: 'Intent',
    },
    action: {                  // ID của action nếu loại node là action
      type: ObjectId,
      ref: 'Action',
    },
    condition: {               // ID của condition nếu loại node là condition
      type: ObjectId,
      ref: 'Condition',
    },
    actionAskAgain: {          // thông tin action hỏi lại nếu loại node là action
      numberOfLoop: Number,    // số lần hỏi lại
      actionFail: {            // ID của hành động khi quá số lần hỏi lại
        type: ObjectId,
        ref: 'Action',
      },
      actionAskAgain: {        // ID của hành động hỏi lại
        type: ObjectId,
        ref: 'Action',
      },
    },
    parent: [                  // danh sách node cha
      {
        node: {                 // ID node cha
          type: ObjectId,
          ref: 'Node',
        },
        type: {                // loại của node cha
          type: String,
        },
      },
    ],
    children: [                // danh sách node con
      {
        node: {                // ID của node con
          type: ObjectId,
          ref: 'Node',
        },
        type: {                // loại node con
          type: String,
        },
        typePort: String,      // loại cổng đầu ra nối đến node con
      },
    ],
    position: {                 // vị trí của node
      x: Number,                // toạ độ x
      y: Number,                // toạ độ y
    },
    workflow: {                // ID của workflow
      type: ObjectId,
      ref: 'Workflow',
    },
    bot: {                     // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
}
```

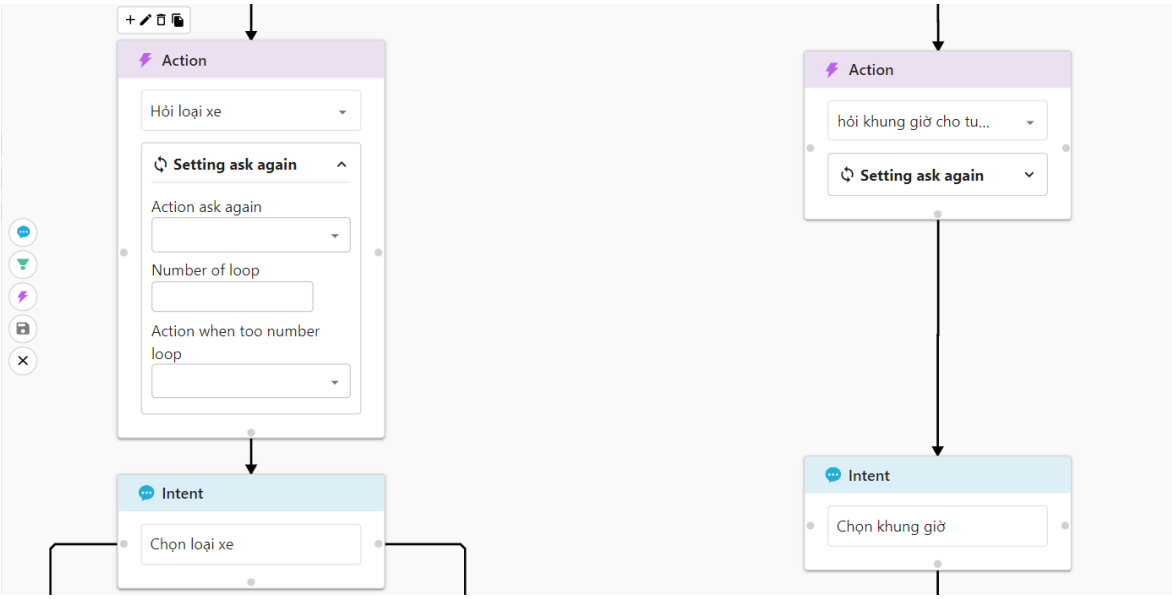
5.4.3 Kết quả

Em đã phát triển được công cụ trực quan giúp người dùng có thể dễ dàng quan sát được luồng kịch bản như **Hình 33 và Hình 34**. Thành phần của màn hình bao gồm bao gồm một node start và các node intent, action, condition và một thanh menu. Trong thanh menu người

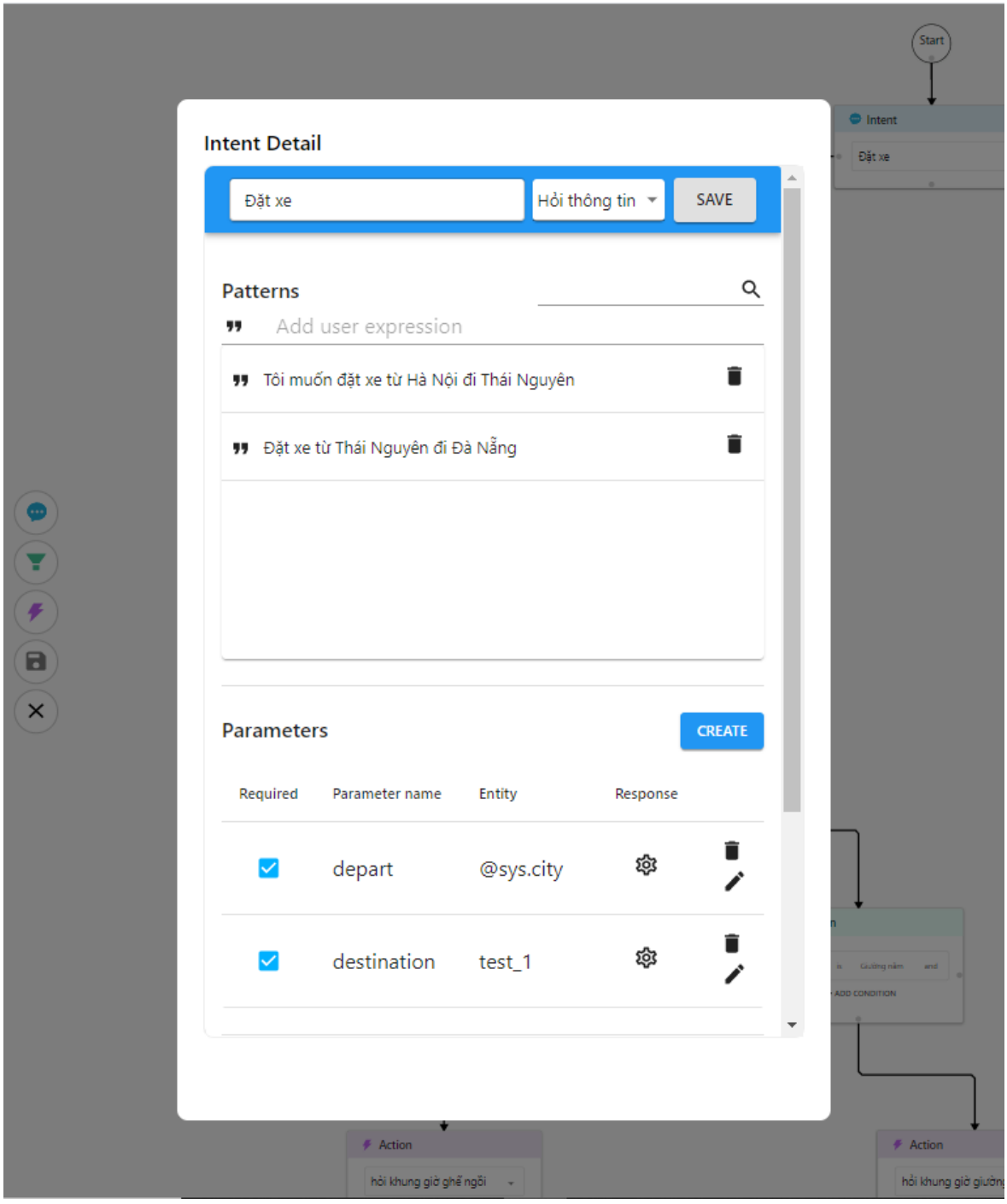
dùng có thể thêm các loại node vào kịch bản, lưu kịch bản và thoát khỏi màn hình thiết kế. Người dùng có thể di chuyển node đến các vị trí thích hợp và cài đặt nội dung cho node sao cho phù hợp với nội dung kịch bản. Khi chọn vào một node trên biểu đồ, ở trên cùng của node đó sẽ hiển thị lên các tuỳ chọn: tạo mới nội dung của node, sửa nội dung của node, nhân bản node và xoá node (Trong **Hình 33** – node Action). Việc thiết kế giao diện kịch bản theo biểu đồ luồng hoạt động như trên giúp người dùng có cái nhìn tổng quan về kịch bản hội thoại, qua đó giúp người dùng có thể thuận tiện các thao tác thêm, sửa, xoá, cập nhật node trong kịch bản.



Hình 33 Màn hình thiết kế kịch bản

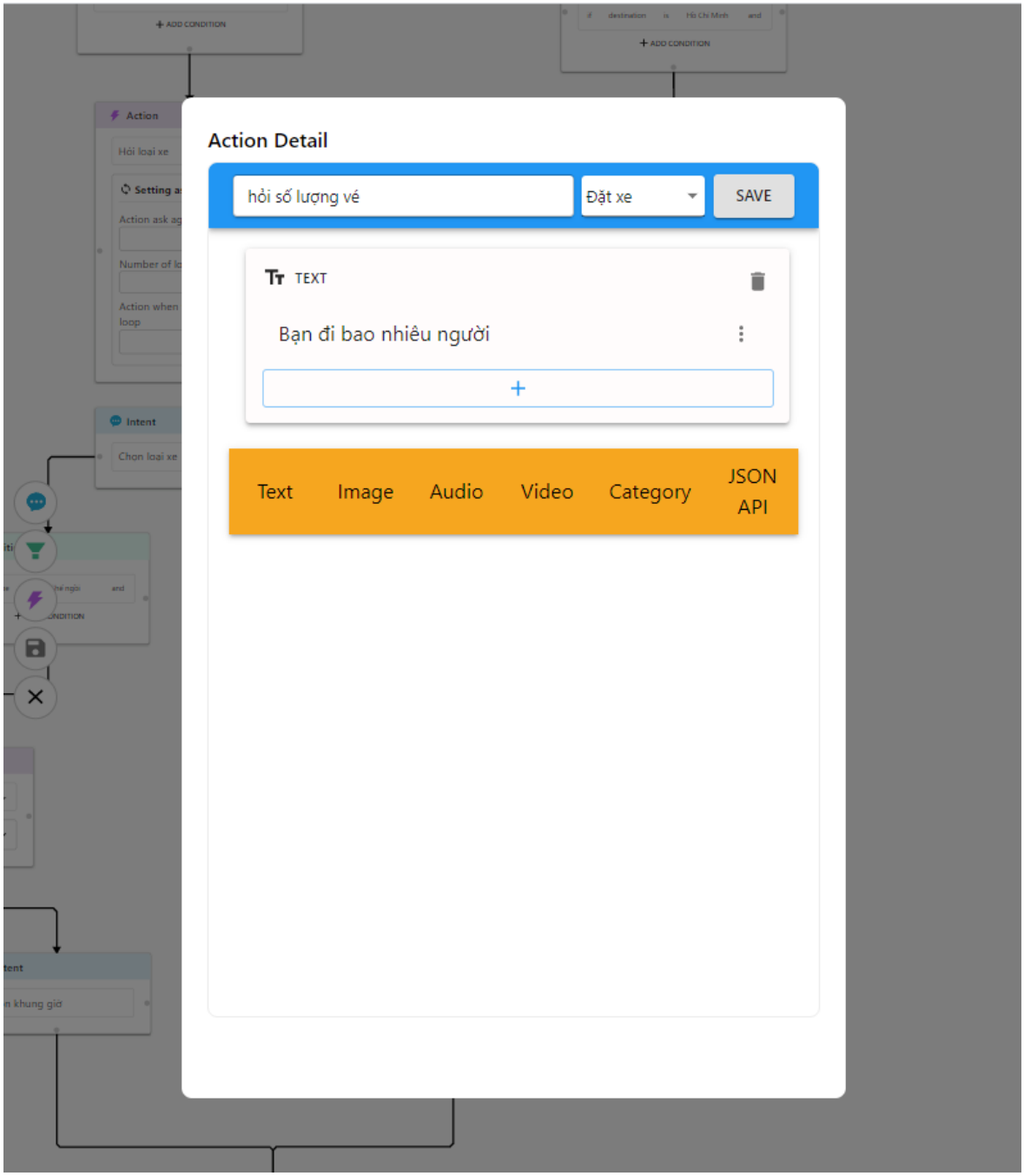


Hình 34 Màn hình thiết kế kịch bản



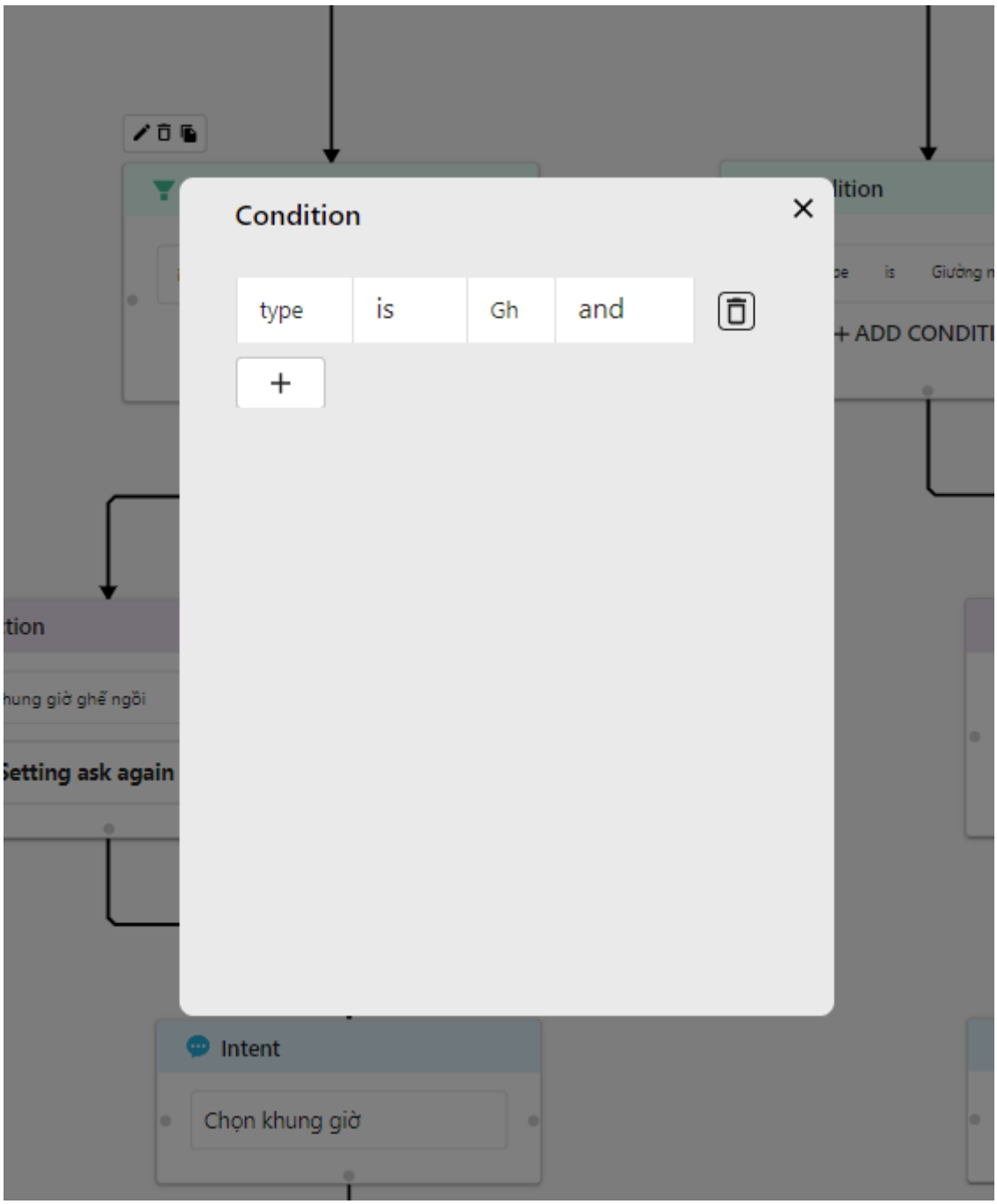
Hình 35 Màn hình thiết kế kịch bản khi sửa ý định

Hình 35 là màn hình chỉnh sửa ý định trong thiết kế kịch bản, các thông tin và thao tác của màn hình giống màn hình tạo ý định được mô tả trong mục 4.3.3.2.



Hình 36 Màn hình thiết kế kịch bản khi sửa action

Hình 36 là màn hình sửa hành động trong thiết kế kịch bản, các thông tin và thao tác của màn hình giống màn hình tạo ý định được mô tả trong mục 4.3.3.3.



Hình 37 Màn hình thiết kế kịch bản khi thêm hoặc sửa điều kiện

Hình 37 là màn hình sửa hoặc thêm điều kiện, ở đây người dùng có thể chọn các tham số được trích xuất trong các ý định trước đó (với điều kiện là node intent phải có luồng chạy tới node condition), người dùng chọn toán tử nhập giá trị để so sánh với tham số và chọn toán tử để liên kết các điều kiện con.

Chương 6 Kết luận và hướng phát triển

6.1 Kết luận

Sau quá trình tìm hiểu, phân tích, thiết kế và xây dựng hệ thống, em đã phát triển được hệ thống xây dựng chatbot theo luật. Hệ thống được xây dựng theo mô hình Microservice nên việc rất thuận tiện cho việc phát triển và mở rộng sau này. Hệ thống hiện tại đã chạy và thử nghiệm tại địa chỉ http://rbc-bot.iristech.club.

Qua quá trình phát triển, em đã xây dựng hệ thống có thể quản lý các thông tin như ý định, thực thể, các phản hồi của bỏ và các kịch bản hội thoại. Trong quá trình tương tác giữa bot và khách hàng, em đã sử dụng Elasticsearch để xác định ý định của người dùng. Công cụ này hỗ trợ truy vấn bằng thuật toán, giúp đánh giá được độ tương đồng của tin nhắn khách hàng với tập ý định được xây dựng sẵn. Trong vấn đề trích xuất thông tin, em đã sử dụng Regex và Từ đồng nghĩa để người dùng có thể tự định nghĩa thực thể cho riêng mình. Để phát triển công cụ quản lý kịch bản luồng hội thoại, em đã tìm hiểu, phân tích và so sánh các hệ thống chatbot trên thị trường, tìm hiểu các thư viện hỗ trợ. Qua đó em đã phát triển công cụ quản lý kịch bản nhắn tin bằng thư viện React Diagram. Nhằm tối ưu hoá giao diện và trải nghiệm người dùng, em đã thiết kế kịch bản thể hiện dưới dạng biểu đồ luồng hoạt động. Người dùng có thể cấu hình những kịch bản hội thoại phức tạp mà vẫn không gây khó khăn khi sử dụng. Với hệ thống người dùng có thể dễ dàng quản lý các câu hỏi mẫu, các hành động của bot, tự định nghĩa các thông tin cần trích xuất và quản lý chúng và người dùng cũng có thể thêm thành viên tham gia xây dựng chatbot.

Tuy nhiên, việc xây dựng hệ thống trong thời gian có ngắn nên còn những hạn chế và thiếu sót, nhưng hệ thống vẫn sẽ được xây dựng và phát triển, lắng nghe các ý kiến từ người dùng để hoàn hoàn thiện hơn.

6.2 Hướng phát triển

Hệ thống vẫn sẽ được xây dựng và phát triển trong tương lai. Trước hết em sẽ cải tiến thêm về UI/UX để nâng cao trải nghiệm người dùng, sau đó em sẽ thêm chức năng real-time khi vẽ kịch bản, phát triển phân quyền quản lý bot chi tiết hơn và tiếp tục lắng nghe các ý kiến từ người dùng để hoàn thiện sản phẩm.

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] NodeJS, *https://nodejs.org*, lần truy cập cuối 10/06/2021

[4] MongoDB, https://mongodb.com, lần truy cập cuối 10/06/2021

[5] RabbitMQ, https://rabbitmq.com, lần truy cập cuối 10/06/2021

[6] Elasticsearch, https://www.elastic.co, lần truy cập cuối 10/06/2021

[7] Redis, https://redis.io, lần truy cập cuối 10/06/2021

[8] ReactJS, https://reactjs.org, truy cập lần cuối 10/06/2021

[9] React-Diagram, https://github.com/projectstorm/react-diagrams, lần truy cập cuối 10/06/2021

[10] Manychat, https://manychat.com, lần truy cập cuối 10/06/2021

[11] Chatfuel, https://chatfuel.com, lần truy cập cuối 10/06/2021

[12] Material UI, https://material-ui.com, lần truy cập cuối 10/06/2021

[13] Kirupa Chinnathambi, Learning React: A hands-on guide to building web applications using React and Redux, tái bản lần 2, Addison-Wesley, 2018

Phụ lục

Phần này em sẽ mô tả thiết kế chi tiết cho từng collection trong cơ sở dữ liệu, bao gồm các collection .

A. Thiết kế chi tiết các collection trong cơ sở dữ liệu

A.1 Thiết kế chi tiết collection Bot

```
{
    id: ObjectId,           // ID của bot
    name: String,           // tên bot
    description: String,    // mô tả bot
    imageUrl: String,       // đường link ảnh của bot
    createBy: {             // ID người tạo
      type: ObjectId,
      ref: 'User',
    },
    botToken: String,       // một mã định danh của bot
    permissions: [          // quyền truy cập bot
      {
        user: {             // ID của user được truy cập
          type: ObjectId,
          ref: 'User',
        },
        role: String,       // loại quyền truy cập bot
      },
    ],
}
```

A.2 Thiết kế chi tiết collection Workflow

```
{
    name: String,          // tên workflow
    zoom: Number,          // độ phóng to thu nhỏ
    offsetX: Number,       // độ lệch so với trục x
    offsetY: Number,       // độ lệch so với trục y
    groupWorkflow: {       // ID nhóm workflow
      type: ObjectId,
      ref: 'GroupWorkflow',
    },
    bot: {                 // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
    createBy: {            // ID của người tạo
      type: ObjectId,
      ref: 'User',
    },
}
```

A.3 Thiết kế chi tiết collection User

```
{
    id: ObjectId,    // ID của user
    name: String,    // tên user
    avatar: String,  // avatar
    email: String,   // email
}
```

A.4 Thiết kế chi tiết collection Slot

```
{
    id: ObjectId,     // ID của slot
    name: String,     // tên slot
    dataType: Number, // Number, Floar, Text,...
    slotType: Number, // 1 default
    bot: {            // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
}
```

A.5 Thiết kế chi tiết collection Message

```
{
    id: ObjectId,              // ID của message
    conversation: {            // ID của conversation
      type: ObjectId,
      ref: 'Conversation',
    },
    from: String,              // BOT, USE
    bot: {                     // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
    status: String,            // trạng thái phản hồi của bot
    message: {
      text: String,            // văn bản
      attachment: {            // đính kèm
        type: { type: String }, // IMAGE, AUDIO, VIDEO, FILE, CATEGORY
        payload: {              // nội dung đính kèm
          url: String,          // đường dẫn
          elements: [           // các phần tử
            {
              label: String,    // tên nhãn hiển thị
              value: String,    // giá trị của tên nhãn
            },
          ],
        },
      },
    },
}
```

A.6 Thiết kế chi tiết collection Conversation

```
{
    id: ObjectId,          // ID của conversation
    sessionId: String,     // ID của phiên làm việc
    bot: {                 // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
    workflow: {            // ID của workflow
      type: ObjectId,
      ref: 'Workflow',
    },
}
```

A.7 Thiết kế chi tiết collection GroupEntity

```
{
    id: ObjectId,         // ID của nhóm entity
    name: String,         // tên group
    groupType: Number,    // 1: DEFAULT, 2: GROUP, 3: GROUP_SINGLE,
    bot: {                // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
}
```

A.8 Thiết kế chi tiết collection Dashboard

```
{
    id: ObjectId,                   // ID của dashboard
    bot: {                          // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
    totalUsersay: Number,           // tổng lượt yêu cầu khách hàng trong 1 ngày
    answeredUsersay: Number,        // số lần thành công
    notUnderstandUsersay: Number,   // số lần bot không hiểu
    defaultUsersay: Number,         // trả lời mặc định
    needConfirmUsersay: Number,     // hỏi lại
}
```

A.9 Thiết kế chi tiết collection Entity

```
{
    id: ObjectId,           // ID của entity
    name: String,           // tên entity
    type: Number,           // loại entity
    pattern: String,        // chuỗi regex
    synonyms: [             // từ đồng nghĩa
      {
        input: [String],    // các từ đầu vào
        output: String,     // từ đầu ra
      },
    ],
    groupEntity: {          // ID của group entity
      type: ObjectId,
      ref: 'GroupEntity',
    },
    bot: {                  // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
    createBy: {             // ID cảu người tạo
      type: ObjectId,
      ref: 'User',
    },
}
```

A.10 Thiết kế chi tiết collection Condition

```
{
    id: ObjectId,           // ID của điều kiện
    conditions: [
      {
        parameter: {        // ID của parameter trong intent
          type: ObjectId,
          ref: 'Intent',
        },
        intent: {           // ID của intent
          type: ObjectId,
          ref: 'Intent',
        },
        operator: String,   // toán tử so sánh
        value: String,      // giá trị so sánh
      },
    ],
    operator: String,       // toán tử nối các điều kiện
    createBy: {             // ID của người tạo
      type: ObjectId,
      ref: 'User',
    },
    bot: {                  // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
}
```

A.11 Thiết kế chi tiết collection GroupIntent

```
{
    id: ObjectId,         // ID của nhóm intent
    name: String,         // tên group
    groupType: Number,    // 1: DEFAULT, 2: GROUP, 3: GROUP_SINGLE,
    bot: {                // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
}
```

A.12 Thiết kế chi tiết collection GroupAction

```
{
    id: ObjectId,         // ID của nhóm action
    name: String,         // tên group
    groupType: Number,    // 1: DEFAULT, 2: GROUP, 3: GROUP_SINGLE,
    bot: {                // ID của bot
      type: ObjectId,
      ref: 'Bot',
    },
}
```