

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

## **ĐỒ ÁN TỐT NGHIỆP**

### **Hệ thống xây dựng dữ liệu huấn luyện từ hội thoại trong chatbot**

**NGUYỄN ANH TUẤN**

`tuan.na154101@sis.hust.edu.vn`

**Ngành Kỹ thuật phần mềm**

**Giảng viên hướng dẫn:** TS. Nguyễn Thị Thu Trang \_\_\_\_\_

**Bộ môn:** Công nghệ phần mềm

**Viện:** Công nghệ thông tin – Truyền thông

**HÀ NỘI, 06/2020**

# Lời cam kết

Họ và tên sinh viên: Nguyễn Anh Tuấn

Điện thoại liên lạc: (+84) 372048460

Email: tuan.na154101@sis.hust.edu.vn

Lớp: CNTT 2.4 Khóa 60

Hệ đào tạo: Đại học chính quy

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

*Hà Nội, ngày 26 tháng 06 năm 2020*

Tác giả ĐATN

*Nguyễn Anh Tuấn*

# Lời cảm ơn

Đầu tiên, em xin phép gửi lời cảm ơn sâu sắc nhất tới TS Nguyễn Thị Thu Trang. Cô đã tận tâm chỉ bảo, giúp đỡ em trong suốt quá trình làm đồ án. Những lời khuyên, kiến thức cô truyền dạy là hành trang quý báu cho em trên con đường sự nghiệp sau này.

Em cũng xin chân thành cảm ơn Ban lãnh đạo nhà trường, các thầy cô trong trường Đại học Bách Khoa Hà Nội nói chung và những thầy cô trong Viện Công nghệ thông tin và Truyền thông nói riêng. Trong suốt 5 năm giảng đường Bách Khoa, nhờ công lao giảng dạy thầy cô, em đã hiểu thế nào là một công dân tốt, một kỹ sư tốt. Đó chính là mục tiêu để em phấn đấu hàng ngày.

Em cảm ơn toàn thể anh, chị, bạn bè cùng học tập và làm việc tại phòng thí nghiệm 914. Mọi người đã nhiệt tình giúp đỡ, chia sẻ, động viên em trong suốt thời gian qua.

Sau cùng, em xin gửi lời cảm ơn tới gia đình và những bạn bè thân thiết đã luôn bên cạnh em trong suốt quá trình học tập này.

Trong quá trình thực hiện đồ án tốt nghiệp, em đã cố gắng nhưng cũng không tránh khỏi các sai sót. Em mong thầy cô góp ý để em có thể chỉnh sửa và hoàn thiện, không tái phạm những lỗi này về sau.

Em xin chân thành cảm ơn!

# Tóm tắt

Ngày nay, chatbot và nền tảng tạo chatbot đang được quan tâm và đầu tư mạnh mẽ. Để triển khai chatbot một cách hiệu quả, quá trình tạo dữ liệu huấn luyện và đánh giá hoạt động của bot trong thực tế là việc vô cùng quan trọng. Tuy nhiên, một số nền tảng sẵn có trên thế giới và Việt Nam còn có các hạn chế về hai vấn đề trên. Hạn chế đó là chưa tận dụng được nguồn hội thoại để làm dữ liệu huấn luyện, chưa có quy trình xây dựng dữ liệu đảm bảo chất lượng. Điều đó đặt ra bài toán cần có một hệ thống xây dựng dữ liệu huấn luyện từ hội thoại khắc phục các vấn đề trên.

Hướng tiếp cận của đề án là tập trung phát triển cơ chế và quy trình xây dựng dữ liệu huấn luyện từ hội thoại. Hội thoại sẽ được tạo bằng hai cách chat tương tác trực tiếp với bot và bóc tách từ lịch sử trò chuyện của người dùng. Quy trình sẽ hướng tới áp dụng trong các công ty lớn, nhiều phòng ban, nhiều người làm dữ liệu. Để đạt được điều này, dữ liệu sẽ được quản lý theo trạng thái, theo nhóm và theo các cấp bậc quản lý. Khi đó, chỉ có dữ liệu đạt trạng thái chấp nhận mới được đưa vào huấn luyện. Nhờ vào quá trình xây dựng dữ liệu từ hội thoại, hệ thống cũng đưa ra được các chỉ số, các biểu đồ, các dạng báo cáo để đánh giá tính hiệu quả của bot trong thực tế.

Do hệ thống có nhiều người sử dụng với nhiều vai trò, em phát triển trên nền tảng web với công nghệ sử dụng là ReactJS, ExpressJS và cơ sở dữ liệu MongoDB. Hiện tại, hệ thống đã được triển khai thực tế trên tên miền <https://smartdialog.iristech.club> và thử nghiệm trong các lĩnh vực ngân hàng (VietcomBank), tài chính (FE Credit).

# Mục lục

<b>Lời cam kết .....</b>	<b>ii</b>
<b>Lời cảm ơn .....</b>	<b>iii</b>
<b>Tóm tắt .....</b>	<b>iv</b>
<b>Mục lục .....</b>	<b>v</b>
<b>Danh mục hình vẽ.....</b>	<b>ix</b>
<b>Danh mục bảng.....</b>	<b>xi</b>
<b>Danh mục các từ viết tắt.....</b>	<b>xii</b>
<b>Danh mục thuật ngữ .....</b>	<b>xiii</b>
<b>Chương 1 Giới thiệu đề tài .....</b>	<b>1</b>
1.1 Đặt vấn đề .....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp .....	2
1.4 Bố cục đồ án .....	3
<b>Chương 2 Khảo sát và phân tích yêu cầu .....</b>	<b>4</b>
2.1 Kịch bản điển hình của chatbot.....	4
2.2 Khảo sát hiện trạng .....	5
2.3 Tổng quan chức năng.....	7
2.3.1 Biểu đồ use case tổng quan .....	7
2.3.2 Biểu đồ use case phân rã Quản lý đơn vị hội thoại .....	8
2.3.3 Biểu đồ use case phân rã Quản lý hội thoại .....	9
2.3.4 Biểu đồ use case phân rã Duyệt hội thoại .....	10

2.4 Đặc tả chức năng.....	11
2.4.1 Đặc tả use case Tạo hội thoại từ lịch sử chat .....	12
2.4.2 Đặc tả use case Tạo hội thoại qua chat tương tác.....	14
2.4.3 Đặc tả use case Chấp nhận hội thoại .....	14
2.4.4 Đặc tả use case Xuất báo cáo .....	15
2.5 Yêu cầu phi chức năng.....	19
2.5.1 Tính dễ dùng.....	19
2.5.2 Khả năng mở rộng .....	19
2.6 Kết chương.....	19
<b>Chương 3 Công nghệ sử dụng.....</b>	<b>20</b>
3.1 Frontend .....	20
3.1.1 ReactJS và NextJS .....	20
3.1.2 Material UI .....	21
3.1.3 Redux.....	22
3.2 Backend .....	22
3.2.1 NodeJS và ExpressJS .....	22
3.2.2 MongoDB .....	23
3.2.3 REST API.....	23
3.3 Kết chương.....	24
<b>Chương 4 Phát triển và triển khai ứng dụng .....</b>	<b>25</b>
4.1 Thiết kế kiến trúc .....	25
4.1.1 Kiến trúc tổng quan .....	25
4.1.2 Kiến trúc Frontend.....	26
4.1.3 Kiến trúc Backend .....	27
4.2 Thiết kế chi tiết Frontend.....	28
4.3 Thiết kế chi tiết Backend .....	31
4.3.1 Luồng hoạt động.....	31

4.3.2 Thiết kế API .....	34
4.4 Thiết kế cơ sở dữ liệu .....	36
4.4.1 Biểu đồ thực thể liên kết.....	36
4.4.2 Thiết kế tổng quan cơ sở dữ liệu .....	37
4.4.3 Thiết kế chi tiết collection .....	37
4.5 Xây dựng ứng dụng .....	40
4.5.1 Thư viện và công cụ sử dụng .....	40
4.5.2 Kết quả đạt được.....	41
4.5.3 Minh họa các chức năng chính .....	41
4.6 Kiểm thử và triển khai .....	50
4.7 Kết chương.....	50
<b>Chương 5 Các giải pháp và đóng góp nổi bật.....</b>	<b>51</b>
5.1 Thiết kế quy trình đảm bảo chất lượng dữ liệu huấn luyện .....	51
5.1.1 Đặt vấn đề.....	51
5.1.2 Quy trình xây dựng và phê duyệt dữ liệu huấn luyện .....	51
5.1.3 Xây dựng thành phần quản lý quyền linh hoạt.....	54
5.2 Giải pháp bóc tách và tạo dữ liệu huấn luyện từ lịch sử chat .....	55
5.2.1 Đặt vấn đề.....	55
5.2.2 Đề xuất hiển thị lịch sử chat theo đơn vị hội thoại.....	55
5.2.3 Quy trình xây dựng dữ liệu huấn luyện từ lịch sử chat thông qua các đơn vị hội thoại .....	56
5.3 Giải pháp tạo dữ liệu huấn luyện qua chat tương tác.....	58
5.3.1 Đặt vấn đề.....	58
5.3.2 Giải pháp .....	58
5.4 Đánh giá tính hiệu quả của bot bằng dữ liệu huấn luyện từ hội thoại .....	60
5.4.1 Đặt vấn đề.....	60
5.4.2 Giải pháp .....	60
5.5 Kết chương.....	61

<b>Chương 6 Kết luận và hướng phát triển .....</b>	<b>62</b>
6.1 Kết luận.....	62
6.2 Hướng phát triển .....	62
<b>Tài liệu tham khảo .....</b>	<b>64</b>
<b>Phụ lục.....</b>	<b>A-1</b>



# Danh mục hình vẽ

<b>Hình 1</b> Các thành phần của chatbot [4] .....	4
<b>Hình 2</b> Chức năng tạo dữ liệu huấn luyện từ lịch sử trò chuyện của Dialogflow .....	6
<b>Hình 3</b> Chức năng tạo dữ liệu huấn luyện từ lịch sử trò chuyện của FPT.AI Conversation.6	
<b>Hình 4</b> Biểu đồ use case tổng quan .....	8
<b>Hình 5</b> Biểu đồ use case phân rã Quản lý đơn vị hội thoại .....	9
<b>Hình 6</b> Biểu đồ use case phân rã Quản lý hội thoại .....	10
<b>Hình 7</b> Biểu đồ use case phân rã Duyệt hội thoại .....	11
<b>Hình 8</b> Quản lý trạng thái khi không có Redux và có Redux .....	22
<b>Hình 9</b> Kiến trúc tổng quan của hệ thống .....	25
<b>Hình 10</b> Kiến trúc Frontend .....	26
<b>Hình 11</b> Các thành phần của Frontend .....	27
<b>Hình 12</b> Kiến trúc Backend.....	27
<b>Hình 13</b> Mockup màn hình “Xem danh sách lịch sử chat” .....	28
<b>Hình 14</b> Mockup màn hình “Xem chi tiết lịch sử chat” .....	29
<b>Hình 15</b> Mockup màn hình “Tạo hội thoại từ lịch sử chat” .....	29
<b>Hình 16</b> Mockup màn hình “Tạo báo cáo” .....	30
<b>Hình 17</b> Mockup màn hình “Xem dashboard tình hình hoạt động của bot” .....	30
<b>Hình 18</b> Mockup màn hình “Phân quyền cho quản trị dữ liệu” .....	31
<b>Hình 19</b> Biểu đồ trình tự Tạo hội thoại từ lịch sử chat .....	32
<b>Hình 20</b> Biểu đồ trình tự Tạo hội thoại qua chat tương tác.....	33
<b>Hình 21</b> Biểu đồ thực thể quan hệ.....	36

<b>Hình 22</b>	Thiết kế tổng quan cơ sở dữ liệu .....	37
<b>Hình 23</b>	Chức năng Xem danh sách lịch sử hội thoại .....	42
<b>Hình 24</b>	Chức năng Xem chi tiết lịch sử trò chuyện .....	42
<b>Hình 25</b>	Chức năng Tạo hội thoại từ lịch sử chat .....	43
<b>Hình 26</b>	Chức năng thêm đơn vị hội thoại .....	44
<b>Hình 27</b>	Tạo hội thoại bằng chat tương tác .....	45
<b>Hình 28</b>	Chức năng chấp nhận hội thoại .....	46
<b>Hình 29</b>	Chức năng xuất báo cáo .....	47
<b>Hình 30</b>	Chức năng xem thông số hoạt động của bot .....	48
<b>Hình 31</b>	Chức năng phân quyền cho quản trị dữ liệu.....	49
<b>Hình 32</b>	Quy trình xây dựng và phê duyệt hội thoại.....	52
<b>Hình 33</b>	Minh họa thành phần quản lý phân quyền linh hoạt .....	54
<b>Hình 34</b>	Mô tả khái niệm đơn vị hội thoại .....	56
<b>Hình 35</b>	Các chức năng quản lý đơn vị hội thoại.....	56
<b>Hình 36</b>	Quy trình tạo và duyệt hội thoại từ lịch sử chat .....	57
<b>Hình 37</b>	Chức năng chat tương tác với bot .....	58
<b>Hình 38</b>	Quy trình tạo dữ liệu huấn luyện qua chat tương tác .....	59
<b>Hình 39</b>	Đánh giá độ hiệu quả của bot qua dữ liệu huấn luyện từ hội thoại.....	60
<b>Hình 40</b>	Chức năng thống kê tính hiệu quả của bot dựa vào hội thoại .....	61

# Danh mục bảng

<b>Bảng 1</b> Kết quả khảo sát các nền tảng chatbot phổ biến .....	7
<b>Bảng 2</b> Danh sách use case.....	11
<b>Bảng 3</b> Dữ liệu đầu vào khi lưu hội thoại .....	13
<b>Bảng 4</b> Dữ liệu đầu vào báo cáo “Câu hỏi khách hàng theo trạng thái” .....	16
<b>Bảng 5</b> Dữ liệu đầu vào báo cáo “Ý định của khách hàng” .....	17
<b>Bảng 6</b> Dữ liệu đầu vào báo cáo “Câu hỏi khách hàng theo nhóm ý định” .....	18
<b>Bảng 7</b> Dữ liệu đầu vào khi xuất báo cáo.....	18
<b>Bảng 8</b> Danh sách API trong hệ thống .....	34
<b>Bảng 9</b> Thiết kế chi tiết Script Collection .....	38
<b>Bảng 10</b> Thiết kế chi tiết LogRequest Collection .....	39
<b>Bảng 11</b> Thiết kế chi tiết GroupScript Collection.....	39
<b>Bảng 12</b> Thiết kế chi tiết Manage Collection.....	39
<b>Bảng 13</b> Thiết kế chi tiết History Collection.....	40
<b>Bảng 14</b> Thư viện và công cụ sử dụng .....	41
<b>Bảng 15</b> Thông số chính của hệ thống .....	41
<b>Bảng 16</b> Dữ liệu đầu ra khi hiển thị danh sách hội thoại .....	A-1
<b>Bảng 17</b> Dữ liệu đầu vào khi lọc danh sách lịch sử chat.....	A-2
<b>Bảng 18</b> Dữ liệu đầu ra các mục đã phân quyền cho quản trị dữ liệu.....	A-5
<b>Bảng 19</b> Các trường hợp kiểm thử cho Tạo hội thoại từ lịch sử .....	B-6
<b>Bảng 20</b> Các trường hợp kiểm thử cho Tạo hội thoại qua chat tương tác.....	B-8
<b>Bảng 21</b> Các trường hợp kiểm thử cho Duyệt hội thoại.....	B-8

# Danh mục các từ viết tắt

<b>API</b>	Application Programming Interface Giao diện lập trình ứng dụng
<b>NLP</b>	Natural Language Processing Xử lý ngôn ngữ tự nhiên
<b>SSR</b>	Server-side rendering Render phía server
<b>CSR</b>	Client-side rendering Render phía client
<b>HTML</b>	HyperText Markup Language Ngôn ngữ đánh dấu siêu văn bản
<b>CNTT</b>	Công nghệ thông tin
<b>ĐATN</b>	Đồ án tốt nghiệp

# Danh mục thuật ngữ

<b>Client</b>	Máy khách
<b>Server</b>	Máy chủ
<b>DOM</b>	Mô hình các đối tượng trong tài liệu HTML
<b>Render</b>	Biểu diễn dữ liệu thành đồ họa
<b>Intent</b>	Ý định của một câu đầu vào
<b>Action</b>	Hành động phản hồi của chatbot

# Chương 1 Giới thiệu đề tài

Chương 1 giới thiệu những vấn đề thực tiễn dẫn tới lý do chọn đề tài, tổng quan về hệ thống xây dựng dữ liệu huấn luyện từ hội thoại trong chatbot. Sau đó em sẽ trình bày về phạm vi, mục tiêu, định hướng giải pháp và bố cục trình bày của đề án.

## 1.1 Đặt vấn đề

Những năm gần đây, những thành tựu của trí tuệ nhân tạo cùng với sự bùng nổ của các ứng dụng nhắn tin đã và đang thúc đẩy sự tăng trưởng và phát triển của chatbot. Hiểu một cách đơn giản, chatbot là một chương trình máy tính mô phỏng cuộc trò chuyện của con người. Từ yêu cầu đầu vào, chatbot phán đoán ý định, mong muốn của người dùng, phản hồi hành động tương ứng. Chatbot có thể hỗ trợ doanh nghiệp trong nhiều lĩnh vực khác nhau như chăm sóc khách hàng, marketing trực tuyến, v.v

Hoạt động của chatbot dựa trên giải thuật xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP). Qua đó, chatbot có thể tự động trả lời từ dữ liệu con người đã dạy. Chatbot có hoạt động tốt hay không, có “thông minh” hay không, dữ liệu huấn luyện đóng vai trò quan trọng. Việc xây dựng và huấn luyện chatbot, hiện nay trên thế giới và Việt Nam, đã có một số nền tảng phổ biến như Chatfuel [1], Dialogflow [2] và FPT.AI Conversation [3], v.v Tuy nhiên các nền tảng trên có một số hạn chế về tạo dữ liệu từ hội thoại để huấn luyện. Đó là chưa tận dụng được hoặc chưa tận dụng triệt để lịch sử hội thoại của người dùng, chưa có quy trình phê duyệt đảm bảo chất lượng dữ liệu, việc đánh giá chất lượng chatbot triển khai trong thực tế còn chưa rõ ràng. Các vấn đề kể trên khiến việc tạo dữ liệu để huấn luyện còn gặp nhiều khó khăn, số lượng dữ liệu có thể ít, chất lượng không đảm bảo, dẫn tới khả năng bot có thể phản hồi không tốt với người dùng.

Do đó, em chọn đề tài “Hệ thống xây dựng dữ liệu huấn luyện từ hội thoại trong chatbot” để cải tiến chất lượng, số lượng dữ liệu hội thoại huấn luyện, đánh giá và nâng cao độ hiệu quả của chatbot trong thực tế.

## 1.2 Mục tiêu và phạm vi đề tài

Dựa vào những vấn đề thực tiễn và mục đích nêu trên, đề án tập trung vào các mục tiêu bao gồm i) xây dựng cơ chế tạo dữ liệu huấn luyện từ hội thoại, ii) thiết kế phương pháp đảm

bảo chất lượng dữ liệu để đưa vào huấn luyện, iii) cung cấp công cụ trực quan hóa hiệu quả hoạt động của chatbot trong thực tế sử dụng.

Với mục tiêu như vậy, em phân tích, thiết kế và xây dựng hệ thống với những chức năng chính bao gồm người làm dữ liệu có thể tạo dữ liệu huấn luyện từ hội thoại một cách trực quan sinh động và dễ sử dụng, dữ liệu được xây dựng phải tuân theo một quy trình để được đưa vào huấn luyện. Các dữ liệu đã xây dựng cần phải theo một quy trình khắt khe để được chấp nhận đưa vào huấn luyện. Ngoài ra, để quan sát tính hiệu quả của bot trong thực tế, hệ thống cung cấp các bảng biểu và các dạng báo cáo thích hợp. Qua đó, công việc tạo và huấn luyện chatbot trở nên dễ dàng hơn và đạt hiệu quả mong muốn.

### **1.3 Định hướng giải pháp**

Để đạt được mục tiêu như trên, em đề xuất và thiết kế một số giải pháp. Đầu tiên, do hệ thống dựa trên hội thoại để xây dựng dữ liệu huấn luyện nên nguồn hội thoại là vô cùng quan trọng. Để có được nguồn hội thoại phong phú và dồi dào, em đưa ra hai hướng giải quyết. Thứ nhất là hỗ trợ bóc tách lịch sử chat của người dùng để đưa vào xây dựng dữ liệu huấn luyện. Phương pháp này giúp người làm dữ liệu có nguồn hội thoại khách quan từ người dùng. Thứ hai là cung cấp công cụ chat cho chính người làm dữ liệu để tương tác, chỉnh sửa hành vi của bot trong quá trình chat trực tiếp. Cách này giúp người làm dữ liệu có được những đoạn hội thoại một cách chủ quan theo mong muốn của mình.

Để đảm bảo chất lượng dữ liệu huấn luyện, em thiết kế quy trình xây dựng và phê duyệt dữ liệu một cách nghiêm ngặt. Dữ liệu được phân chia theo nhóm và theo các cấp bậc quản lý. Cấp bậc quản lý được chia làm hai cấp: quản trị dữ liệu và biên tập dữ liệu. Dữ liệu sẽ được trải qua năm trạng thái: đang xây dựng, chờ phê duyệt, đang phê duyệt, trả lại, chấp nhận. Khi đó, chỉ những dữ liệu ở trạng thái chấp nhận mới được đưa vào huấn luyện. Quy trình như vậy sẽ rất thích hợp với các công ty lớn, nhiều phòng ban, nhiều nhân viên cùng làm dữ liệu cho chatbot. Để xây dựng được quy trình như vậy, giao diện hệ thống cũng đòi hỏi phải thể hiện phù hợp theo từng vai trò. Nếu không giải quyết khéo léo, việc mở rộng mã nguồn sau này sẽ gặp khó khăn khi phát sinh các yêu cầu mới, vai trò mới. Do vậy, em xây dựng một thành phần quản lý quyền một cách linh hoạt và tập trung. Cách xử lý logic về quyền hiển thị được tập trung tập trung tại một chỗ. Sau này khi có thay đổi, ta chỉ cần chỉnh sửa tại một chỗ mà không ảnh hưởng nhiều tới các thành phần khác trong mã nguồn.

Cuối cùng, hệ thống sẽ thiết kế tính năng thống kê và báo cáo tình hình xây dựng dữ liệu và hoạt động của bot trong thực tế. Tính năng này sẽ đưa ra các dạng biểu đồ và báo cáo thích hợp cho người quản trị để dàng xem xét và đánh giá.

Với nhiều người sử dụng trên nhiều máy tính khác nhau, chức năng cần giao diện phức tạp như vậy, em đề xuất triển khai hệ thống trên nền tảng web. Hệ thống được chia làm 2 phần

frontend và backend riêng biệt tương tác với nhau qua REST API giúp quá trình mở rộng hệ thống sau này dễ dàng hơn. Phần frontend em sử dụng framework của React là NextJS. NextJS có ưu điểm dễ dàng cấu hình dự án, tối ưu tải Javascript cho từng đường dẫn, tăng tốc độ tải trang. Phần backend em sử dụng framework ExpressJS của NodeJS. NodeJS có ưu điểm bất đồng bộ, hướng sự kiện, có hiệu năng tốt. Ngoài ra, cùng một hệ sinh thái Javascript, việc phát triển ứng dụng sẽ nhanh hơn, ít lỗi hơn. Phần cơ sở dữ liệu em sử dụng cơ sở dữ liệu MongoDB. Nó là cơ sở dữ liệu có tính mở rộng cao, thích hợp với dữ liệu lớn, tùy biến cao như lịch sử trò chuyện, hội thoại.

## **1.4 Bố cục đồ án**

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

Chương 2, em sẽ khảo sát các nền tảng tạo và huấn luyện chatbot sẵn có trên thị trường. Qua quá trình khảo sát, em sẽ phân tích yêu cầu, trình bày tổng quan chức năng và làm rõ các chức năng chính của hệ thống.

Tiếp theo, chương 3, em sẽ mô tả về những công nghệ áp dụng trong đồ án này, lý do lựa chọn công nghệ đó để giải quyết bài toán.

Chương 4 mô tả chi tiết kiến trúc phát triển và triển khai ứng dụng bao gồm các quá trình thiết kế, xây dựng, kiểm thử và triển khai.

Trong chương 5, em sẽ trình bày tất cả những đóng góp, giải pháp mà em thấy tâm đắc nhất trong suốt quá trình làm ĐATN. Đó là những kinh nghiệm, bài học giúp em phát triển bản thân trên con đường kỹ sư sau này.

Cuối cùng, chương 6 là kết luận, ưu điểm, nhược điểm, những gì đã làm được và chưa làm được của em. Ngoài ra, em đề xuất các định hướng phát triển cho hệ thống trong tương lai để hệ thống hoàn thiện hơn.

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

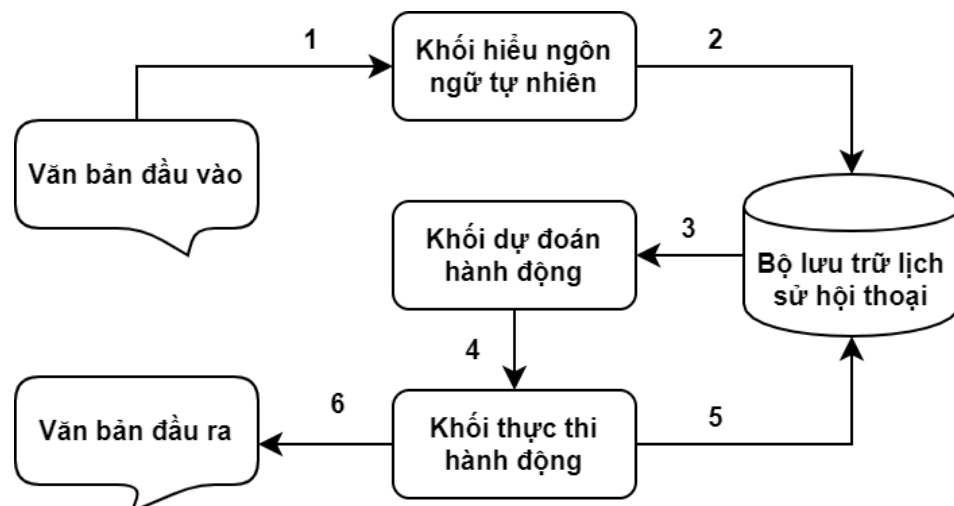


## Chương 2 Khảo sát và phân tích yêu cầu

Chương 2 em sẽ mô tả cách thức hoạt động của chatbot để làm rõ các khái niệm liên quan. Sau đó, em sẽ khảo sát các nền tảng tương tự trên thị trường để phân tích yêu cầu cho hệ thống.

### 2.1 Kịch bản diễn hình của chatbot

Trước khi khảo sát các nền tảng khác, em sẽ trình bày rõ ràng hơn về các khái niệm, cách thức, kịch bản hoạt động của chatbot. Chatbot là chương trình máy tính có khả năng tương tác với người dùng bằng cách giao tiếp tự nhiên [4].



**Hình 1** Các thành phần của chatbot [4]

**Hình 1** mô tả thứ tự xử lý và các thành phần trong một hệ thống chatbot. Người dùng đưa yêu cầu đầu vào qua các dạng như văn bản, giọng nói. Khi đó, thành phần hiểu ngôn ngữ tự nhiên trích xuất ra thông tin bao gồm ý định của người dùng và các thông tin quan trọng trong câu hay còn gọi là các thực thể. Các thông tin này sẽ được ghi lại vào bộ lưu trữ lịch sử hội thoại ở bước 2. Đến bước 3, khối dự đoán hành động nhận thông tin các trạng thái từ bộ lưu trữ để đưa ra các hành động tiếp theo sao cho phù hợp và thực thi hành động đó ở bước 4. Hành động đã dự đoán sẽ được ghi chép lại ở bộ lưu trữ hội thoại tại bước 5 và trả về phản hồi cho người dùng ở bước 6.

Ý định (intent) là những mong muốn, mục đích của người dùng. Ý định gắn với những nghiệp vụ của doanh nghiệp. Ví dụ với chatbot về ngân hàng đó là hỏi thông tin thẻ, thông

tin tài khoản, tra cứu ATM. Còn với chatbot về bán hàng, ý định có thể là tra cứu sản phẩm, mua bán sản phẩm, v.v

Thực thể (entity) là những thông tin có ý nghĩa được trích xuất từ yêu cầu đầu vào người dùng. Ví dụ trong câu “Cho mình hỏi địa điểm ATM quận cầu giấy”, “quận cầu giấy” là một thực thể chỉ địa điểm. Để huấn luyện cho bot hiểu ý định và trích xuất được thực thể, người làm dữ liệu phải cung cấp cho bot tập câu mẫu. Câu mẫu là những câu có sẵn phù hợp với ý định và đã được đánh dấu thực thể nếu có. Ví dụ với ý định “Hướng dẫn sử dụng ngân hàng điện tử” thì tập câu mẫu có thể là “làm sao để đổi mật khẩu trên mobile banking”, “hướng dẫn tôi cách hủy dịch vụ khác trên internet banking”, “chuyển tiền trong internet banking như thế nào”, v.v

Hành động (action) mô tả phản hồi của của bot sau khi phân tích yêu cầu. Hành động có thể gồm nhiều dạng như chữ, hình ảnh, video, audio phản hồi lại người dùng hay các lệnh thực thi như gửi email, tắt đèn, v.v để tác động vào bên thứ ba.

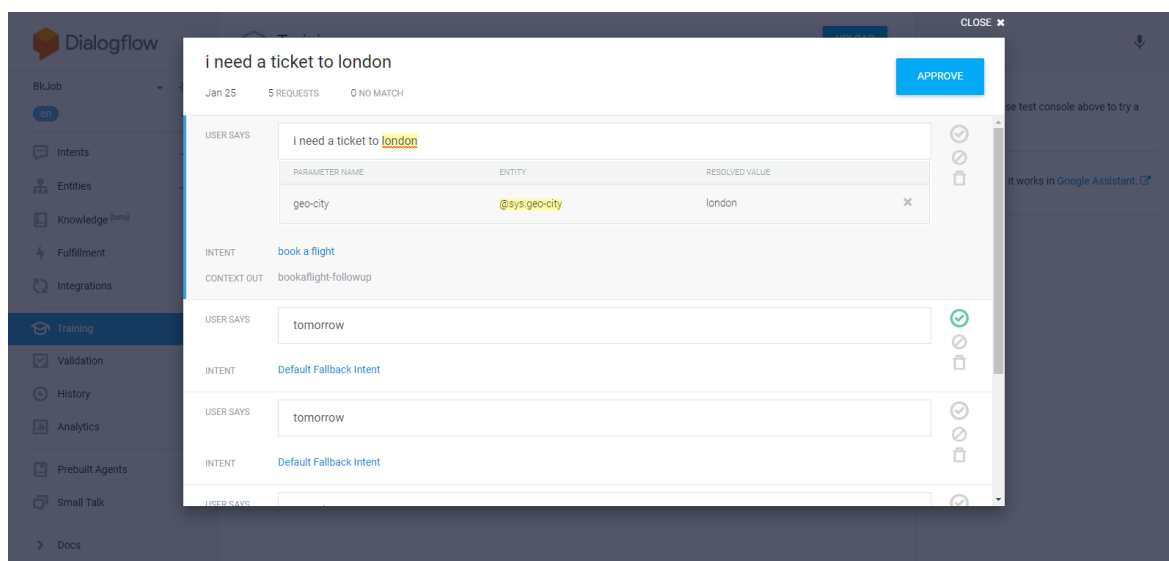
Để thuận tiện cho việc đánh giá và chỉnh sửa, em đề xuất hiển thị hội thoại thành các đơn vị hội thoại. Đơn vị hội thoại bao gồm một câu đầu vào của người dùng đi kèm với thông tin về ý định, thực thể được trích xuất, tiếp theo là những hành động phản hồi của bot. Hiểu một cách đơn giản hơn, đơn vị hội thoại là một thành phần gồm hỏi và đáp. Chi tiết về khái niệm đơn vị hội thoại sẽ được trình bày ở chương 5. Từ hội thoại, dữ liệu huấn luyện có thể xây dựng bằng nhiều cách. Các cách đó bao gồm chuyển các câu đầu vào trong hội thoại thành câu mẫu hoặc sử dụng chính hội thoại đó để huấn luyện bot theo kịch bản

## 2.2 Khảo sát hiện trạng

Trong khuôn khổ đồ án, em khảo sát ba nền tảng chatbot phổ biến, có nhiều người sử dụng. Hai nền tảng trên thế giới là Chatfuel và Dialogflow, một nền tảng của Việt Nam là FPT.AI Conversation. Hiện nay, Chatfuel chưa có chức năng tạo dữ liệu huấn luyện từ lịch sử trò chuyện. Hai nền tảng còn lại Dialogflow và FPT. AI Conversation đã có cơ chế tận dụng lịch sử trò chuyện làm dữ liệu huấn luyện. Tuy nhiên, cơ chế đó còn một số nhược điểm. Các nhược điểm đó bao gồm chưa có chỉnh sửa hành động trong hội thoại, chưa có cơ chế hiển thị hội thoại một cách rõ ràng trực quan để người làm dữ liệu có thể dễ dàng đánh giá và chỉnh sửa dữ liệu.

**Hình 2** mô tả chức năng tạo dữ liệu huấn luyện từ lịch sử trò chuyện của nền tảng Dialogflow. Ta thấy nền tảng này chỉ đang cho phép chỉnh sửa ý định và thực thể, chưa cho thay đổi hành động mà bot phản hồi. Tương tự như vậy, ở nền tảng FPT.AI Conversation được mô tả ở **Hình 3**, hiện nay cũng chưa cho phép chỉnh sửa lại hành động. Cách thức hiển thị lịch sử trò chuyện của cả hai nền tảng này là theo từng yêu cầu đầu vào của người dùng. Cách hiển thị như vậy khiến người làm dữ liệu khó theo dõi được toàn bộ ngữ cảnh của cuộc

trò chuyện để làm dữ liệu huấn luyện. Ngoài ra, cả ba nền tảng trên đều chưa cung cấp công cụ chat tương tác giúp người làm dữ liệu vừa tạo hội thoại vừa chỉnh sửa hành vi của bot. Điều này dẫn tới bỏ qua một cách tạo dữ liệu từ hội thoại một cách chủ động, nhanh chóng. Các nền tảng này cũng chưa phát triển một quy trình xây dựng và phê duyệt dữ liệu. Việc thiếu sót quy trình này dẫn tới khả năng các dữ liệu huấn luyện chất lượng không đồng đều, ảnh hưởng tới quá trình huấn luyện của bot. Cuối cùng, do không phân chia dữ liệu theo các nhóm dữ liệu và các cấp bậc quản lý, các nền tảng này chưa thực sự phù hợp trong các công ty lớn có nhiều phòng ban, nhiều người cùng làm dữ liệu. Với những kết quả đã khảo sát, để tiện theo dõi em tổng hợp và mô tả ở **Bảng 1**.



**Hình 2** Chức năng tạo dữ liệu huấn luyện từ lịch sử trò chuyện của Dialogflow

Thời gian yêu cầu	Username	Người dùng nói	Bot nhận diện	Update Intent	Cách xử lý	Độ tin cậy	
20:38:06 26-02-2020	[42864] tuan	tôi muốn tìm việc ở hà nội	tam_biet	<a href="#">Sửa</a>	Bot không hiểu - câu trả lời mặc định	0.30	<a href="#">Chi tiết</a>
20:36:56 26-02-2020	[42864] tuan	có việc nào ở hà nội không	tam_biet	<a href="#">Sửa</a>	Bot không hiểu - câu trả lời mặc định	0.26	<a href="#">Chi tiết</a>
20:29:07 26-02-2020	[42864] tuan	địa điểm ở đâu	tam_biet	<a href="#">Sửa</a>	Bot không hiểu - câu trả lời mặc định	0.33	<a href="#">Chi tiết</a>
20:28:49 26-02-2020	[42864] tuan	các diễn giả là ai	show_tat_ca_di_en_gia	<a href="#">Sửa</a>	Bot hiểu - trả lời theo kịch bản	0.80	<a href="#">Chi tiết</a>
20:28:39 26-02-2020	[42864] tuan	cho tôi xem lịch trình buổi sáng	lich_trinh_buoi_sang	<a href="#">Sửa</a>	Bot hiểu - trả lời theo kịch bản	0.99	<a href="#">Chi tiết</a>
20:28:15 26-02-2020	[42864] tuan	xin chào	chao_hoi	<a href="#">Sửa</a>	Bot hiểu - trả lời theo kịch bản	1.00	<a href="#">Chi tiết</a>

**Hình 3** Chức năng tạo dữ liệu huấn luyện từ lịch sử trò chuyện của FPT.AI Conversation

**Bảng 1** Kết quả khảo sát các nền tảng chatbot phổ biến

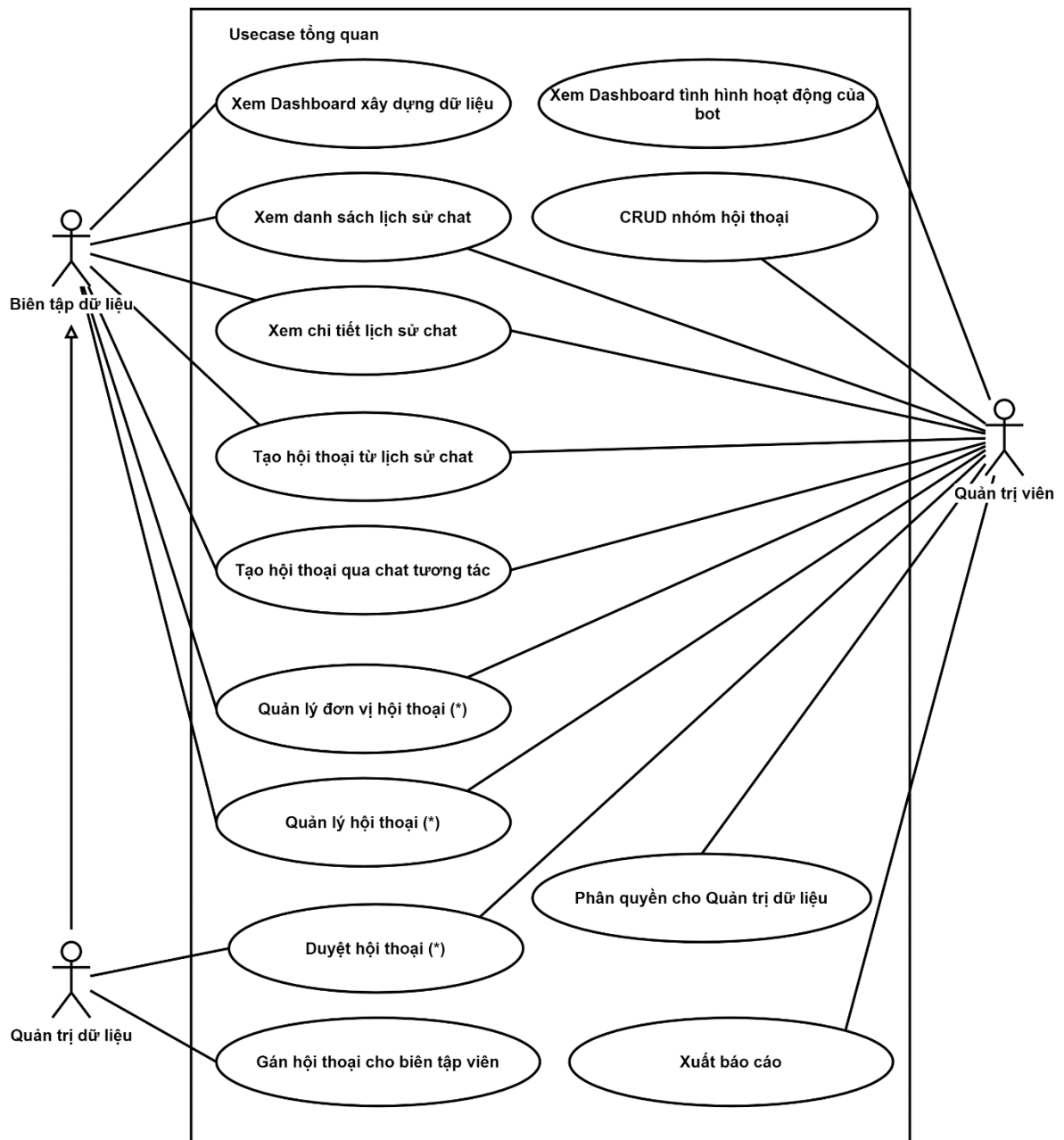
Chức năng	Chatfuel	Dialogflow	FPT.AI Conversation
Tạo dữ liệu huấn luyện từ lịch sử chat	Không có	Đã có nhưng chưa cho sửa hành động	Đã có nhưng chưa cho sửa hành động
Tạo hội thoại qua chat tương tác (vừa tạo hội thoại vừa chỉnh sửa dữ liệu)	Không có	Không có	Không có
Quy trình xây dựng và duyệt dữ liệu huấn luyện	Không có	Không có	Không có

Sau khi khảo sát ba nền tảng trên, em nhận thấy các nền tảng này còn một số chức năng cần hoàn thiện và phát triển thêm để có thể tạo dữ liệu huấn luyện từ hội thoại một cách dễ dàng và đạt hiệu quả cao nhất. Do vậy, em đã phân tích, thiết kế yêu cầu của hệ thống để khắc phục được những điểm còn hạn chế đó. Chi tiết về hệ thống này sẽ được trình bày ở phần tiếp theo.

## 2.3 Tổng quan chức năng

### 2.3.1 Biểu đồ use case tổng quan

**Hình 4** miêu tả use case tổng quan của hệ thống. Hệ thống xây dựng dữ liệu từ hội thoại có ba loại tác nhân chính bao gồm biên tập dữ liệu, quản trị dữ liệu và quản trị viên. Quản trị viên có quyền tạo các nhóm hội thoại, phân quyền các nhóm hội thoại và nhóm biên tập viên cho người quản trị dữ liệu, xem dashboard các thông số của bot khi hoạt động trong thực tế và xuất các dạng báo cáo thích hợp. Đối với biên tập dữ liệu, tác nhân này cũng có dashboard của riêng mình. Đó là bảng danh sách những dữ liệu do mình xây dựng ra theo các trạng thái khác nhau. Ngoài ra, biên tập viên có thể xem danh sách lịch sử chat của người dùng, xem chi tiết một lịch sử chat, tạo hội thoại qua lịch sử chat và chat trực tiếp với bot để vừa xây dựng hội thoại vừa chỉnh sửa hành vi của bot. Sau khi tạo xong hội thoại từ hai cách trên, biên tập viên có thể quản lý các hội thoại này. Ngoài ra, do cơ chế hiển thị hội thoại theo các đơn vị hội thoại, biên tập viên sẽ có thêm chức năng quản lý các đơn vị hội thoại đó. Tiếp đến tác nhân cuối cùng, quản trị dữ liệu là những người quản lý các biên tập viên. Ngoài các chức năng kế thừa từ biên tập dữ liệu, quản trị dữ liệu còn có thể duyệt hội thoại được tạo ra để đưa vào huấn luyện, gán dữ liệu huấn luyện của biên tập viên này cho biên tập viên khác.

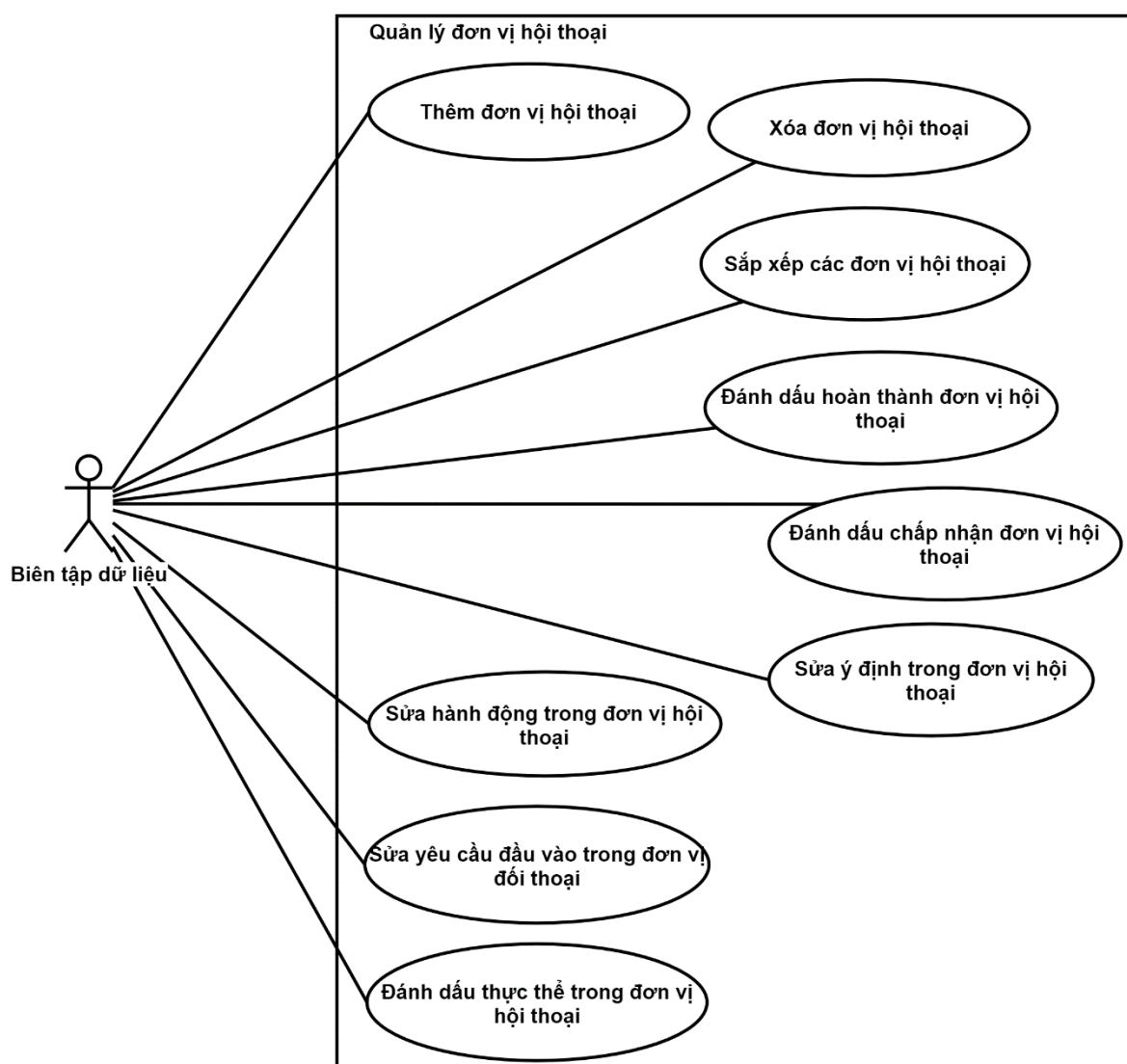


**Hình 4** Biểu đồ use case tổng quan

### 2.3.2 Biểu đồ use case phân rã Quản lý đơn vị hội thoại

**Hình 5** mô tả use case phân rã Quản lý đơn vị hội thoại. Do nhu cầu tạo dữ liệu huấn luyện từ hội thoại kéo theo nhu cầu cần có các chức năng để chỉnh sửa một đoạn hội thoại tương ứng. Quản lý đơn vị hội thoại sẽ bao gồm các chức năng chỉnh sửa trên các đơn vị hội thoại khác nhau. Các chức năng đó bao gồm thêm đơn vị hội thoại (thêm vào cuối hội thoại hoặc thêm vào bên trên, bên dưới đơn vị hội thoại đã chọn), di chuyển, sắp xếp các đơn vị hội thoại. Chức năng sửa đơn vị hội thoại bao gồm sửa ý định, hành động, nội dung và đánh dấu lại thực thể của yêu cầu đầu vào. Ngoài ra, để duyệt một hội thoại đưa vào huấn luyện thì

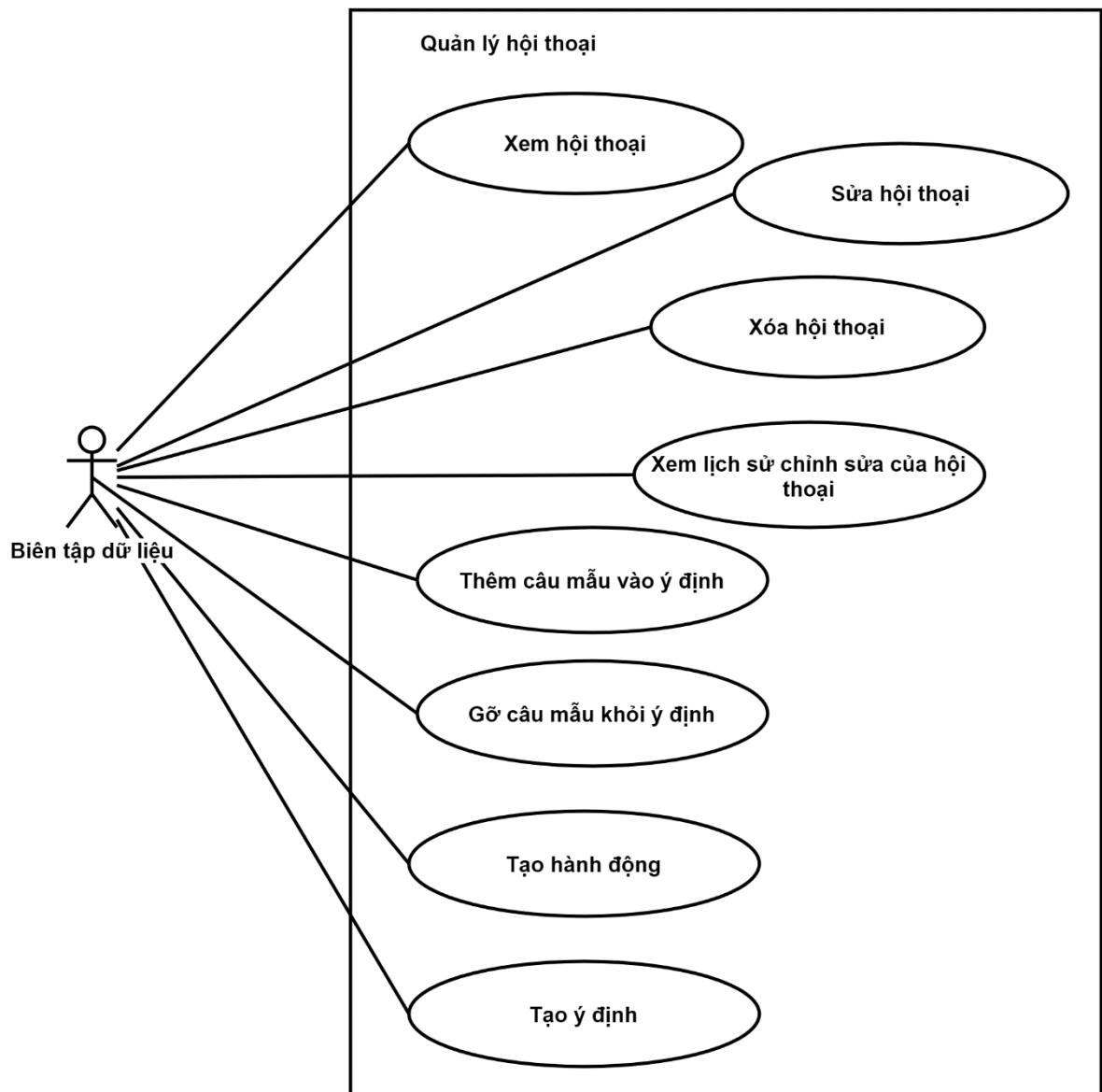
biên tập viên cũng cần phải duyệt lần lượt từng đơn vị hội thoại bên trong. Chỉ khi nào tất cả đơn vị hội thoại được chấp nhận thì hội thoại mới được chấp nhận.



**Hình 5** Biểu đồ use case phân rã Quản lý đơn vị hội thoại

### 2.3.3 Biểu đồ use case phân rã Quản lý hội thoại

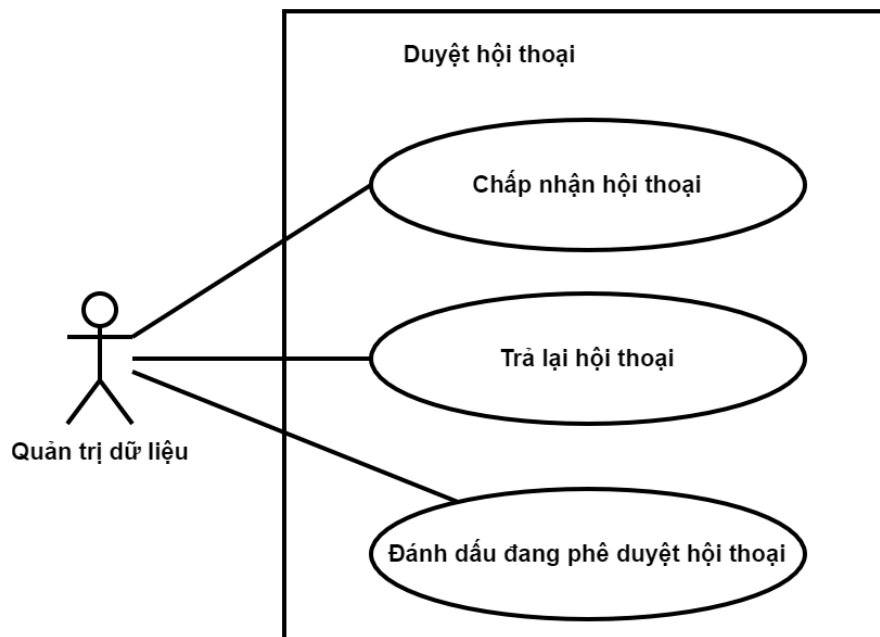
**Hình 6** là biểu đồ phân rã cho use case Quản lý hội thoại. Sau khi các hội thoại được xây dựng qua bóc tách lịch sử chat hoặc qua chat tương tác trực tiếp với bot, biên tập dữ liệu có thể quản lý các đoạn hội thoại này. Khi đó, biên tập dữ liệu có thể xem nội dung hội thoại, xem lịch sử chỉnh sửa. Ngoài hai chức năng xem, biên tập dữ liệu có thể tác động tới hội thoại qua hai chức năng khác là sửa và xóa hội thoại. Trong quá trình xây dựng dữ liệu hội thoại, nếu cần sử dụng ý định và hành động mà chúng chưa tồn tại, biên tập viên có thể tạo thêm ý định hoặc hành động mới để sử dụng. Cuối cùng trong phần quản lý hội thoại, họ có thể thêm câu mẫu hoặc gỡ câu mẫu tương ứng với ý định để hoàn thiện dữ liệu ý định. Đây cũng một trong hai dạng dữ liệu huấn luyện xây dựng từ hội thoại.



**Hình 6** Biểu đồ use case phân rã Quản lý hội thoại

### 2.3.4 Biểu đồ use case phân rã Duyệt hội thoại

**Hình 7** mô tả use case phân rã Duyệt hội thoại. Use case này phân rã thành ba use case: chấp nhận hội thoại, trả lại hội thoại và đánh dấu đang phê duyệt hội thoại. Sau khi biên tập viên tạo và hoàn thành dữ liệu, quản trị dữ liệu có thể bắt đầu thực hiện duyệt. Khi hội thoại đảm bảo về mặt chất lượng, quản trị dữ liệu sẽ chọn chức năng chấp nhận. Khi đó, dữ liệu sẽ đạt trạng thái chấp nhận và được đưa vào huấn luyện. Nếu dữ liệu không đạt chất lượng và quản trị dữ liệu không thể chỉnh sửa hết thì họ chọn chức năng trả lại hội thoại và đi kèm tin nhắn để biên tập dữ liệu sửa và hoàn thành chúng. Trong trường hợp còn lại, khi quản trị dữ liệu chưa duyệt xong, họ có thể đánh dấu hội thoại đó thành trạng thái đang phê duyệt. Khi ở trạng thái này, biên tập viên sẽ không được phép chỉnh sửa, đảm bảo tính nhất quán về mặt dữ liệu cho lần duyệt tới của quản trị viên.



**Hình 7** Biểu đồ use case phân rã Duyệt hội thoại

## 2.4 Đặc tả chức năng

Phần này em sẽ đặc tả một số chức năng chính của Hệ thống xây dựng dữ liệu từ hội thoại. Sau quá trình phân tích yêu cầu, hệ thống của em bao gồm 30 use case được liệt kê ở **Bảng 2**. Tuy nhiên do khuôn khổ đề án có hạn, em sẽ đặc tả chi tiết các use case “Tạo hội thoại từ lịch sử chat”, “Tạo hội thoại qua chat tương tác”, “Chấp nhận hội thoại” và “Xuất báo cáo”. Một số use case khác sẽ được đặc tả trong phần phụ lục.

**Bảng 2** Danh sách use case

Mã use case	Tên use case	Mã use case	Tên use case
UC001	Tạo hội thoại từ lịch sử chat	UC002	Tạo hội thoại qua chat tương tác
UC003	Chấp nhận hội thoại	UC004	Xuất báo cáo
UC005	Xem danh sách lịch sử chat	UC006	Sửa hành động trong đơn vị hội thoại
UC007	Phân quyền cho quản trị dữ liệu	UC008	Xem chi tiết lịch sử chat
UC009	Thêm đơn vị hội thoại	UC010	Xóa đơn vị hội thoại



Mã use case	Tên use case	Mã use case	Tên use case
UC011	Sắp xếp các đơn vị hội thoại	UC012	Đánh dấu hoàn thành đơn vị hội thoại
UC013	Đánh dấu chấp nhận đơn vị hội thoại	UC014	Sửa ý định trong đơn vị hội thoại
UC015	Xem dashboard xây dựng dữ liệu	UC016	Sửa yêu cầu đầu vào đơn vị hội thoại
UC017	Đánh dấu thực thể trong đơn vị hội thoại	UC018	Xem hội thoại
UC019	Xóa hội thoại	UC020	Xem lịch sử chỉnh sửa của hội thoại
UC021	Thêm câu mẫu vào ý định	UC022	Gỡ câu mẫu khỏi ý định
UC023	Tạo hành động	UC024	Tạo ý định
UC025	Trả lại hội thoại	UC026	Đánh dấu đang phê duyệt hội thoại
UC027	Xem Dashboard tình hình hoạt động của bot	UC028	CRUD nhóm hội thoại
UC029	Phân quyền cho quản trị dữ liệu	UC030	Gán hội thoại cho biên tập

#### 2.4.1 Đặc tả use case Tạo hội thoại từ lịch sử chat

Mã Use case	UC001	Tên Use case	Tạo hội thoại từ lịch sử chat
Tác nhân	Biên tập dữ liệu, quản trị dữ liệu, quản trị viên		
Tiền điều kiện	Người dùng đăng nhập, đang ở chức năng xem chi tiết lịch sử trò chuyện		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Tác nhân	Chọn chức năng chuyển thành hội thoại
	2.	Hệ thống	Mở tab mới chức năng hội thoại

	3.	Hệ thống	Hiển thị nội dung hội thoại gồm các đơn vị hội thoại
	4.	Tác nhân	Đánh dấu các đơn vị là hoàn thành
	5.	Tác nhân	Chọn chức năng hoàn thành
	6.	Hệ thống	Kiểm tra dữ liệu có hợp lệ
	7.	Hệ thống	Hiển thị form nhập thông tin để lưu (mô tả phía dưới *)
	8.	Tác nhân	Nhập dữ liệu và ấn lưu
	9.	Hệ thống	Kiểm tra tính hợp lệ dữ liệu
	10.	Hệ thống	Thông báo lưu thành công
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	5a1.	Tác nhân	Chỉnh sửa đơn vị hội thoại
	5a2.	Hệ thống	Hệ thống tự động lưu dữ liệu với tên mặc định, nhóm mặc định
	5b1.	Tác nhân	Chọn chức năng lưu nháp
	5b2.	Hệ thống	Đi đến bước 5
<b>Hậu điều kiện</b>	Không		

\* Dữ liệu đầu vào khi lưu hội thoại

**Bảng 3** Dữ liệu đầu vào khi lưu hội thoại

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Tên hội thoại	Tên hiển thị của hội thoại	Có	Tên chỉ chứa kí tự tiếng việt, số và gạch dưới	Mở thẻ visa
2.	Nhóm	Nhóm hội thoại	Không	Có trong hệ thống	Mở thẻ

### 2.4.2 Đặc tả use case Tạo hội thoại qua chat tương tác

Mã Use case	UC002	Tên Use case	Tạo hội thoại bằng chat tương tác
Tác nhân	Biên tập dữ liệu, quản trị dữ liệu, quản trị viên		
Tiền điều kiện	Người dùng đăng nhập		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Tác nhân	Chọn chức năng tạo hội thoại
	2.	Hệ thống	Hiển thị giao diện chat
	3.	Tác nhân	Nhập yêu cầu đầu vào
	4.	Hệ thống	Phân tích yêu cầu đầu vào
	5.	Hệ thống	Hiển thị câu trả lời của bot
	6.	Hệ thống	Tự động lưu với tên và nhóm mặc định
	7.	Tác nhân	Chọn chức năng hoàn thành
	8.	Hệ thống	Kiểm tra tính hợp lệ dữ liệu
	9.	Hệ thống	Hiển thị form nhập thông tin để lưu (đã mô tả ở <b>Bảng 3</b> )
	10.	Tác nhân	Nhập dữ liệu và ấn lưu
	11.	Hệ thống	Kiểm tra tính hợp lệ dữ liệu
	12.	Hệ thống	Thông báo lưu thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	7a1.	Tác nhân	Chỉnh sửa đơn vị hội thoại
	7a2.	Hệ thống	Hệ thống tự động lưu dữ liệu
Hậu điều kiện	Không		

### 2.4.3 Đặc tả use case Chấp nhận hội thoại

Mã Use case	UC003	Tên Use case	Chấp nhận hội
-------------	-------	--------------	---------------

			thoại
<b>Tác nhân</b>	Quản trị dữ liệu, quản trị viên		
<b>Tiền điều kiện</b>	Người dùng đăng nhập với vai trò quản trị dữ liệu, quản trị viên		
<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Tác nhân	Chọn một hội thoại
	2.	Hệ thống	Hiển thị nội dung hội thoại
	3.	Tác nhân	Đánh dấu các khối là chấp nhận
	4.	Hệ thống	Chọn chức năng chấp nhận
	5.	Hệ thống	Kiểm tra tính hợp lệ dữ liệu
	6.	Hệ thống	Hiển thị form nhập thông tin để lưu (mô tả ở <b>Bảng 3</b> )
	7.	Tác nhân	Nhập dữ liệu và ấn lưu
	8.	Hệ thống	Kiểm tra tính hợp lệ dữ liệu
	9.	Hệ thống	Thông báo lưu thành công
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	4a1.	Tác nhân	Chỉnh sửa đơn vị hội thoại
	4a2.	Hệ thống	Hệ thống tự động lưu dữ liệu thành trạng thái đang phê duyệt
	5a1.	Hệ thống	Hệ thống thông báo dữ liệu không hợp lệ
	8a1.	Hệ thống	Thông báo dữ liệu không hợp lệ
<b>Hậu điều kiện</b>	Không		

#### 2.4.4 Đặc tả use case Xuất báo cáo

<b>Mã Use case</b>	UC004	<b>Tên Use case</b>	Xuất báo cáo
<b>Tác nhân</b>	Quản trị viên		

<b>Tiền điều kiện</b>	Người dùng đăng nhập với vai trò quản trị viên		
<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Tác nhân	Chọn trang báo cáo
	2.	Hệ thống	Hiển thị giao diện trang báo cáo
	3.	Tác nhân	Chọn loại báo cáo
	4.	Hệ thống	Hiển thị các trường có thể hiện trong báo cáo (mô tả phía dưới *)
	5.	Tác nhân	Chọn các trường cần hiện trong báo cáo
	6.	Tác nhân	Chọn dạng file xuất, khoảng thời gian, logo và sửa tiêu đề (mô tả phía dưới **)
	7.	Tác nhân	Chọn chức năng xuất báo cáo
	8.	Hệ thống	Thống kê dữ liệu, trả về báo cáo tương ứng
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	7a1.	Tác nhân	Chọn chức năng xem trước
	7a2.	Hệ thống	Hiển thị báo cáo dưới dạng xem trước
<b>Hậu điều kiện</b>	Không		

\* Dữ liệu đầu vào khi tạo báo cáo

**Bảng 4** Dữ liệu đầu vào báo cáo “Câu hỏi khách hàng theo trạng thái”

Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
Câu hỏi trả lời được	Các câu hỏi bot trả về hành động khác mặc định	Không	True hoặc False	True
Câu hỏi không trả lời được	Câu hỏi bot trả lời hành động mặc định	Không	True hoặc False	True
Câu hỏi bị hỏi lại	Câu hỏi bot không hiểu phải hỏi lại	Không	True hoặc False	True

Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
Câu hỏi đã duyệt	Câu hỏi mà khi tạo hội thoại, biên tập viên không sửa	Không	True hoặc False	True
Câu hỏi chỉ sửa ý định	Câu hỏi mà khi tạo hội thoại, biên tập viên sửa ý định	Không	True hoặc False	True
Câu hỏi chỉ sửa hành động	Câu hỏi mà khi tạo hội thoại, biên tập viên sửa hành động	Không	True hoặc False	True
Câu hỏi sửa cả ý định và hành động	Câu hỏi mà khi tạo hội thoại, Biên tập viên sửa cả ý định và hành động	Không	True hoặc False	True
Được yêu cầu nhiều nhất	Ý định được phán đoán từ yêu cầu người dùng nhiều nhất	Không	True hoặc False	True
Được yêu cầu ít nhất	Ý định được phán đoán từ yêu cầu người dùng ít nhất	Không	True hoặc False	True
Được dự đoán tốt nhất	Ý định khi tạo hội thoại huấn luyện được sửa ít nhất	Không	True hoặc False	True
Được dự đoán tồi nhất	Ý định khi tạo hội thoại huấn luyện được sửa nhiều nhất	Không	True hoặc False	True

**Bảng 5** Dữ liệu đầu vào báo cáo “Ý định của khách hàng”

Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
Được yêu cầu nhiều nhất	Ý định được phán đoán từ yêu cầu	Không	True hoặc False	True

Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
	người dùng nhiều nhất			
Được yêu cầu ít nhất	Ý định được phán đoán từ yêu cầu người dùng ít nhất	Không	True hoặc False	True
Được dự đoán tốt nhất	Ý định khi tạo hội thoại huấn luyện được sửa ít nhất	Không	True hoặc False	True
Được dự đoán tồi nhất	Ý định khi tạo hội thoại huấn luyện được sửa nhiều nhất	Không	True hoặc False	True

**Bảng 6** Dữ liệu đầu vào báo cáo “Câu hỏi khách hàng theo nhóm ý định”

Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
Nhóm ý định	Các nhóm ý định được định nghĩa từ đầu	Không	Các nhóm ý định có trong hệ thống	Mở thẻ

**\*\*** Dữ liệu đầu vào khi xuất báo cáo

**Bảng 7** Dữ liệu đầu vào khi xuất báo cáo

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Tiêu đề	Tiêu đề báo cáo	Có	Khác rỗng	Báo cáo yêu cầu người dùng
2.	Logo	Logo trong báo cáo	Không	Không	
3	Khoảng thời gian	Khoảng thời gian lọc dữ liệu	Có	Tuần, Tháng hoặc khoảng thời gian cố định	Tuần

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
4	Dạng file xuất	Dạng file báo cáo	Có	Excel, PDF, CSV	Excel

## 2.5 Yêu cầu phi chức năng

Để phần mềm được sử dụng một cách hiệu quả, ngoài sự đầy đủ về mặt chức năng, các yêu cầu phi chức năng cũng đóng vai trò quan trọng.

### 2.5.1 Tính dễ dùng

Hệ thống tạo dữ liệu huấn luyện từ hội thoại là một hệ thống có những chức năng, quy trình và các khái niệm tương đối mới mẻ với người dùng không chuyên. Do vậy, hệ thống cần phải cung cấp các giao diện trực quan, cách tiếp cận đơn giản và dễ sử dụng. Ngoài ra, do sự phụ thuộc lẫn nhau giữa các dữ liệu huấn luyện nên hệ thống cũng có các cảnh báo hướng dẫn nếu biên tập viên thực hiện các tác vụ sai nghiệp vụ, sai quy trình hoặc những tác vụ có ảnh hưởng lớn tới các mục dữ liệu khác.

### 2.5.2 Khả năng mở rộng

Hệ thống trong tương lai có thể thay đổi về quy trình tạo và phê duyệt dữ liệu. Điều đó dẫn tới khả năng thay đổi về giao diện để đáp ứng với từng vai trò người sử dụng khác nhau. Do vậy hệ thống khi xây dựng và phát triển phải đảm bảo tính dễ mở rộng. Khi đó, các tính năng phát triển sau này sẽ không ảnh hưởng quá nhiều tới các tính năng đã xây dựng của hệ thống.

## 2.6 Kết chương

Chương này em đã khảo sát các vấn đề của các nền tảng sẵn có trên thị trường. Từ thực tiễn đó, em đã phân tích yêu cầu đề tài, giới thiệu chức năng tổng quan và tiến hành đặc tả các use case quan trọng. Ngoài ra, chương này còn mô tả thêm các yêu cầu phi chức năng để nâng cao trải nghiệm người dùng. Để đạt được yêu cầu như trên, Chương 3 sẽ trình bày các công nghệ sử dụng trong hệ thống này.



## Chương 3 Công nghệ sử dụng

Với yêu cầu của hệ thống trình bày trong chương 2, chương 3 em sẽ mô tả và làm rõ các công nghệ và lý do sử dụng các công nghệ đó trong đồ án của mình.

### 3.1 Frontend

Để thiết kế và phát triển giao diện web nhanh chóng và thân thiện với người sử dụng. Em lựa chọn framework của ReactJS là NextJS, cùng với thư viện đi kèm là Redux, Material UI.

#### 3.1.1 ReactJS và NextJS

Vào thời kì đầu của lập trình web, trang web được hoạt động theo cơ chế Server-Side Rendering (SSR). Khi người dùng vào một trang web, trình duyệt sẽ gửi yêu cầu tới Web Server. Web Server sẽ nhận yêu cầu, xử lý nghiệp vụ, trả về file HTML, CSS, Javascript cho người dùng. Sau khi trình duyệt nhận toàn bộ HTML, Javascript sẽ bắt đầu chạy để chỉnh sửa DOM (các đối tượng trong HTML), tạo hiệu ứng, ... Cơ chế này có một nhược điểm. Server đóng nhiều vai trò khi vừa xử lý nghiệp vụ, kết nối cơ sở dữ liệu, vừa phụ trách trả tài nguyên cho Client. Thời gian phản hồi giữa Client và Server tăng, ảnh hưởng không tốt đến trải nghiệm người dùng. Nhược điểm thứ hai, với chức năng giao diện phức tạp, việc sử dụng Javascript thao tác trực tiếp vào DOM khiến công việc lập trình trở nên khó khăn, dễ phát sinh lỗi. Ngoài ra, khi đó hiệu suất của ứng dụng cũng giảm do HTML được render lại nhiều lần.

Ngày nay, để khắc phục các nhược điểm trên, xu hướng mới trong lập trình web là Client-side Rendering (CSR). Việc render HTML, CSS được chuyển trách nhiệm về phía trình duyệt. Server đóng vai trò cung cấp dữ liệu qua API. Nổi bật trong xu hướng đó là sử dụng ReactJS. ReactJS là thư viện Javascript, được phát triển với mục đích xây dựng giao diện người dùng [5]. Thư viện này được Facebook phát triển và có cộng đồng sử dụng lớn mạnh. Đến thời điểm tháng 6 2020, mã nguồn React trên Github đạt 150.000 sao.

Tư tưởng chính của React là chia nhỏ các giao diện thành các thành phần (Component). Component có thể chứa các component con. Mỗi component chứa trạng thái (state) và thuộc tính (props) của chính nó. Khi state và props thay đổi, component sẽ được render lại. Sự khác biệt của state và props là state được quản lý bởi component, có thể thay đổi bởi chính

nó, còn props thì được truyền từ ngoài vào, bản thân component không thể tự thay đổi được. Điểm giống nhau của state và props là đều là biến của Javascript, có thể là chuỗi, đối tượng, số, v.v Chính vì vậy, lập trình viên muốn thay đổi giao diện chỉ cần thao tác với các biến của Javascript mà không cần thao tác vào DOM. Tư tưởng lập trình trở nên rõ ràng, mạch lạc, mã nguồn cũng có khả năng mở rộng, tái sử dụng hơn.

Ưu điểm nổi bật khác của React đó là Virtual DOM (DOM ảo). Website càng ngày càng phức tạp, DOM Tree cũng phức tạp theo. Việc tìm kiếm, thao tác, chỉnh sửa trực tiếp trên DOM với tần suất lớn làm giảm hiệu năng của trang web. Virtual DOM là giải pháp cho vấn đề trên. Virtual DOM là bản sao lưu của DOM, là nơi mà các component thực sự trên đó. Khi component render lại, React sẽ tính toán và biết được thay đổi nào cần cập nhật trên DOM thật. Qua đó, việc cập nhật trên DOM thật được giảm thiểu một cách đáng kể. Hiệu năng trang web được tăng lên.

ReactJS tuy có nhiều ưu điểm kể trên nhưng cơ bản đây chỉ là thư viện xây dựng giao diện. Để phát triển trang web một cách hoàn chỉnh, lập trình viên cần kết hợp thêm nhiều thư viện như react-router (cấu hình điều hướng trang web), webpack (đóng gói và quản lý tài nguyên Javascript, CSS), v.v Việc kết hợp các thư viện như vậy, nếu thực hiện không tốt, hiệu suất của trang web không những không tăng lên mà còn giảm đi. Trong đó, có thể kể đến một vấn đề điển hình của một số trang web sử dụng Client-Side Rendering hay gặp phải. Đó là trình duyệt phải tải một file Javascript rất nặng chứa nhiều code không cần thiết để chạy ngay lúc đó. Do vậy, em sử dụng NextJS [6], là một framework của ReactJS. NextJS có những ưu điểm sau i) tự động routing qua thư mục “page”, ii) tự động chia nhỏ mã nguồn tải về theo từng trang, iii) cài đặt sẵn môi trường và thư viện liên quan, iv) tự động tải lại trang khi có thay đổi trong lập trình. Qua đó, việc lập trình với thư viện ReactJS trở nên dễ dàng và năng suất hơn.

### **3.1.2 Material UI**

Lợi ích lớn nhất của việc sử dụng ReactJS nói riêng và các thư viện, framework Javascript hiện đại nói chung là tái sử dụng mã nguồn. Do tư tưởng chia giao diện theo component như đã trình bày ở trên, ta có thể dễ dàng sử dụng các bộ component được định nghĩa sẵn. Material UI là một trong những bộ component như thế.

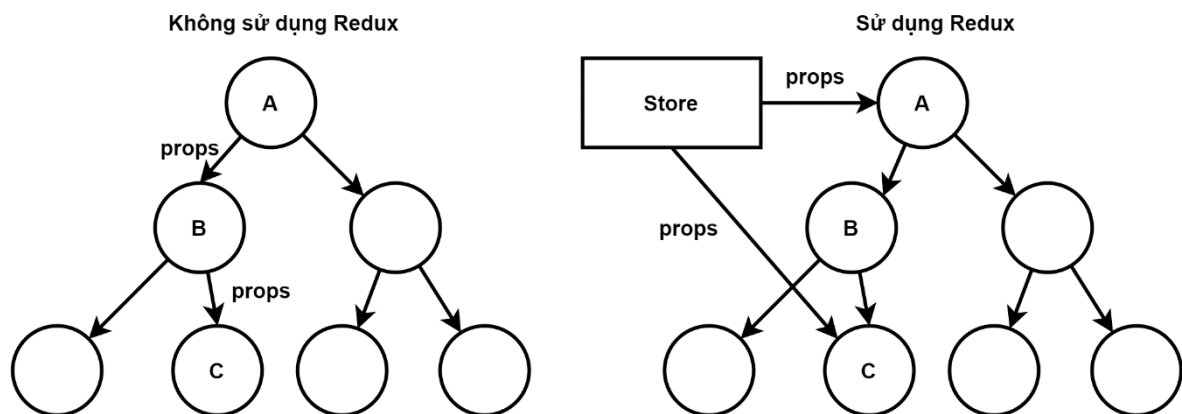
Material UI là thư viện mã nguồn mở, thiết kế theo tư tưởng Material Design [7]. Thư viện cung cấp sẵn các mẫu thiết kế về form, button, dialog, select... và các tương tác Javascript trong đó. Từ đó, việc lập trình giao diện trở nên nhanh chóng và dễ dàng hơn, không phải viết lại toàn bộ mã nguồn từ đầu.

Đây là một trong những thư viện UI phổ biến nhất hiện nay, cộng đồng sử dụng lớn, giao diện khá phù hợp với một trang quản lý nên em lựa chọn sử dụng.

### 3.1.3 Redux

Thông thường, khi sử dụng React, các component giao tiếp với nhau theo cơ chế như sau. Component ở mức cao muốn truyền trạng thái xuống mức thấp phải truyền tuần tự props qua các component ở mức trung gian. Tương tự như vậy, các component ở mức thấp khi muốn thay đổi trạng thái ở mức cao, cũng truyền lần lượt các sự kiện lên các component ở mức bên trên. Với những chức năng đơn giản, cách làm như vậy đã đáp ứng được nhu cầu. Tuy nhiên, với những chức năng phức tạp, trạng thái chia sẻ chung qua nhiều component, việc thực hiện như vậy làm mã nguồn trở nên phức tạp, khó mở rộng, khó bảo trì.

Redux [8] là thư viện quản lý trạng thái trong ứng dụng dùng để giải quyết vấn đề trên. Redux quản lý trạng thái tập trung tại một nơi duy nhất gọi là store. Tất cả các component cần props nào sẽ lắng nghe và nhận trực tiếp từ store này mà không cần thông qua các component trung gian. Ngược lại, các component muốn thay đổi trạng thái nào sẽ thực hiện một action để store nhận và thay đổi state qua các hàm thay đổi trạng thái là reducer.



**Hình 8** Quản lý trạng thái khi không có Redux và có Redux

**Hình 8** mô tả cho ta thấy lợi ích khi sử dụng Redux. Với trường hợp không dùng Redux, khi component C muốn nhận trạng thái component A, component A phải truyền props xuống component B rồi mới đến C. Mặc dù, component B không sử dụng props này. Còn trường hợp sử dụng Redux, component A và C đều nhận state từ store mà không cần qua component B. Khi store thay đổi, component A và C đều có thể nhận được sự thay đổi đó.

## 3.2 Backend

### 3.2.1 NodeJS và ExpressJS

NodeJS là một nền tảng chạy trên môi trường V8 Javascript runtime [9]. Đây là một trình thông dịch Javascript cực nhanh của Chrome. Nhờ có NodeJS, Javascript đã có thể sử dụng làm ngôn ngữ lập trình phía Backend thay vì chỉ lập trình Frontend như trước kia. NodeJS được xây dựng và phát triển từ năm 2009, phần core được viết hầu hết bằng C++. Do vậy

tốc độ xử lý và hiệu năng của nó đạt hiệu suất khá cao. Ngoài ra, với cơ chế non-blocking IO, kiến trúc hướng sự kiện, NodeJS còn giúp ứng dụng có tốc độ xử lý nhanh, theo thời gian thực, đáp ứng lượng truy cập lớn, dễ dàng mở rộng. NodeJS cũng là lựa chọn của rất nhiều công ty lớn như Amazon, Ebay, Microsoft, v.v và có một cộng đồng đông đảo với nhiều thư viện tốt.

Framework NodeJS em lựa chọn là ExpressJS [10]. ExpressJS là framework phổ biến nhất của NodeJS để xây dựng các ứng dụng web [11]. Mã nguồn đạt 48.900 sao trên Github, đứng đầu trong các framework của NodeJS. Ưu điểm nổi bật của ExpressJS là nhỏ gọn, linh hoạt, cung cấp nhiều tính năng mạnh mẽ. Framework cung cấp cơ chế xây dựng các phương thức HTTP như GET, POST, DELETE, PUT, ... và các middleware. Nhờ vậy, việc xây dựng các API giao tiếp với client trở nên nhanh chóng và dễ dàng hơn rất nhiều.

Để phát triển API cho Frontend thì ngôn ngữ và framework cho Server còn nhiều lựa chọn khác như Spring của Java hay Laravel của PHP. Tuy vậy em lựa chọn ExpressJS của NodeJS một phần vì những ưu điểm trên, một phần là do cùng ngôn ngữ Javascript. Việc lập trình, triển khai, cài đặt nhờ vậy sẽ đơn giản, hạn chế phát sinh lỗi hơn.

### **3.2.2 MongoDB**

NoSQL [12] là dạng cơ sở dữ liệu mã nguồn mở, hay còn có tên gọi khác là “not only SQL”. Đó là loại cơ sở dữ liệu mà không truy vấn bằng ngôn ngữ SQL, không có chuẩn chung thống nhất như SQL. NoSQL được sử dụng rộng rãi xuất phát từ nhu cầu lưu trữ dữ liệu cực lớn, truy vấn tốc độ cao mà không đòi hỏi quá nhiều về năng lực phần cứng như tài nguyên hệ thống và khả năng chịu lỗi.

MongoDB [13] là một trong những cơ sở dữ liệu phổ biến của NoSQL. MongoDB hỗ trợ đa nền tảng Windows, Linux, v.v và đa ngôn ngữ Javascript, C#, Java, v.v Dữ liệu trong MongoDB được lưu dưới dạng documents là các BSON (Binary JSON). Đây là dạng dữ liệu có cấu trúc rất linh động và không có quan hệ, giúp truy vấn nhanh và dễ dàng thay đổi.

Hệ thống cần truy vấn từ lịch sử tin nhắn rất nhiều, cùng với đó dạng tin nhắn rất linh động và có khả năng thay đổi trong tương lai. Để đạt được yêu cầu trên hệ thống cần có một cơ sở dữ liệu truy vấn nhanh và linh động trong việc thay đổi. Qua các phân tích ở trên, em quyết định chọn MongoDB làm cơ sở dữ liệu cho ứng dụng.

### **3.2.3 REST API**

REST API là một chuẩn dùng trong việc thiết kế API để quản lý tài nguyên ứng dụng [14]. Đây là một chuẩn phổ biến nhất hiện nay. Trọng tâm của REST quy định các cách phương thức HTTP (GET, POST, PUT, DELETE, v.v) và cách định dạng các URL cho ứng dụng web. Ví dụ như GET để truy xuất dữ liệu, POST để tạo mới dữ liệu, PUT để cập nhật dữ

liệu, DELETE để xóa một dữ liệu. Các định dạng dữ liệu phổ biến của REST là JSON, XML. Trong phạm vi của đề tài, dữ liệu trả về dưới dạng JSON. Client khi muốn lấy dữ liệu từ Server sẽ gọi một HTTP Request để nhận HTTP Response tương ứng. Việc sử dụng REST API như vậy giúp phát triển hai thành phần Backend và Frontend một cách độc lập, tăng tốc quá trình phát triển dự án. Đồng thời, công việc mở rộng, bảo trì sau này sẽ đơn giản hơn.

### **3.3 Kết chương**

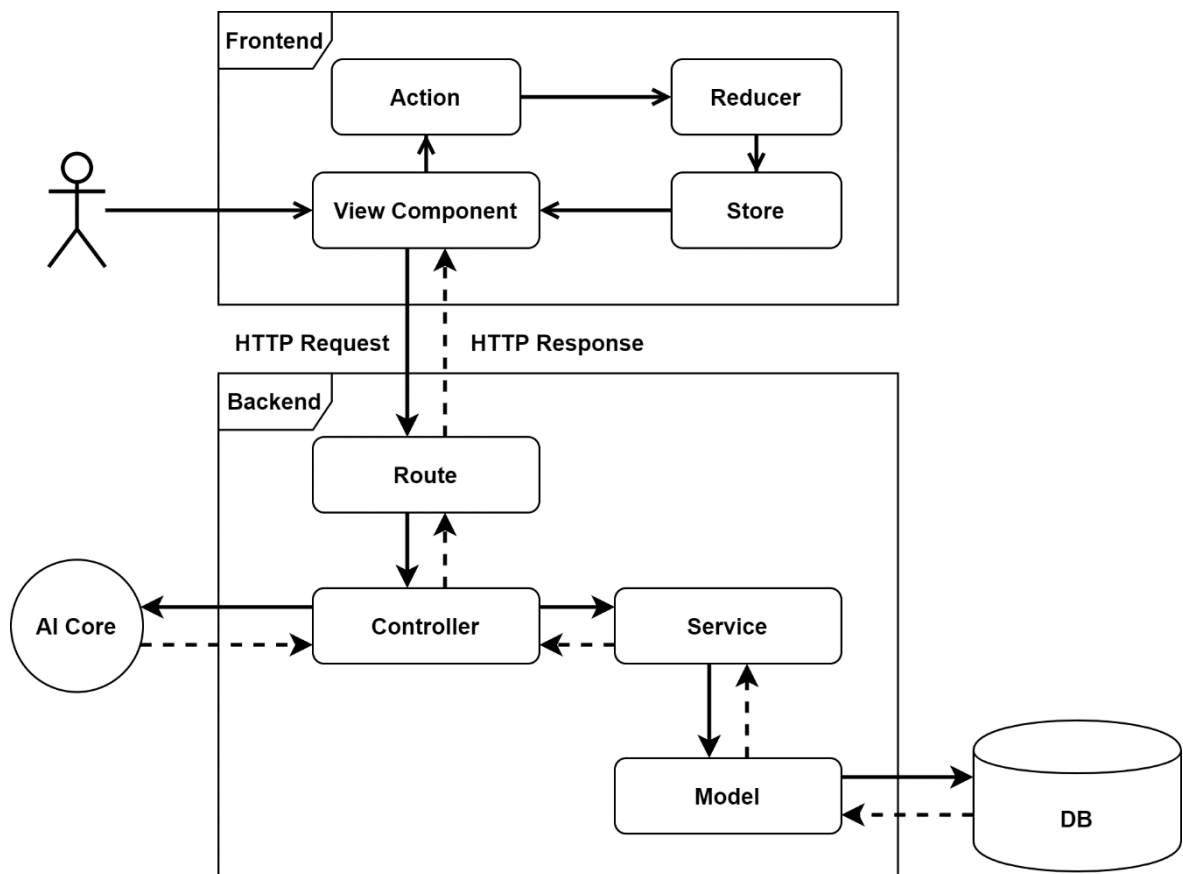
Chương 3 em đã trình bày lý thuyết, phân tích ưu nhược điểm của các công nghệ sử dụng trong đồ án của mình. Hệ thống được chia làm hai phần i) Frontend với NextJS, ReduxJS, Material UI, ii) Backend với ExpressJS, MongoDB. Tiếp theo, em sẽ làm rõ cách áp dụng những công nghệ trên vào hệ thống của mình.

## Chương 4 Phát triển và triển khai ứng dụng

Chương 3 đã trình bày công nghệ áp dụng cho ứng dụng. Chương 4 này sẽ miêu tả chi tiết các công nghệ đó được triển khai theo kiến trúc như thế nào. Đồng thời, chương này cũng trình bày kết quả làm được trong quá trình xây dựng đồ án và xây dựng một số trường hợp kiểm thử kèm theo.

### 4.1 Thiết kế kiến trúc

#### 4.1.1 Kiến trúc tổng quan

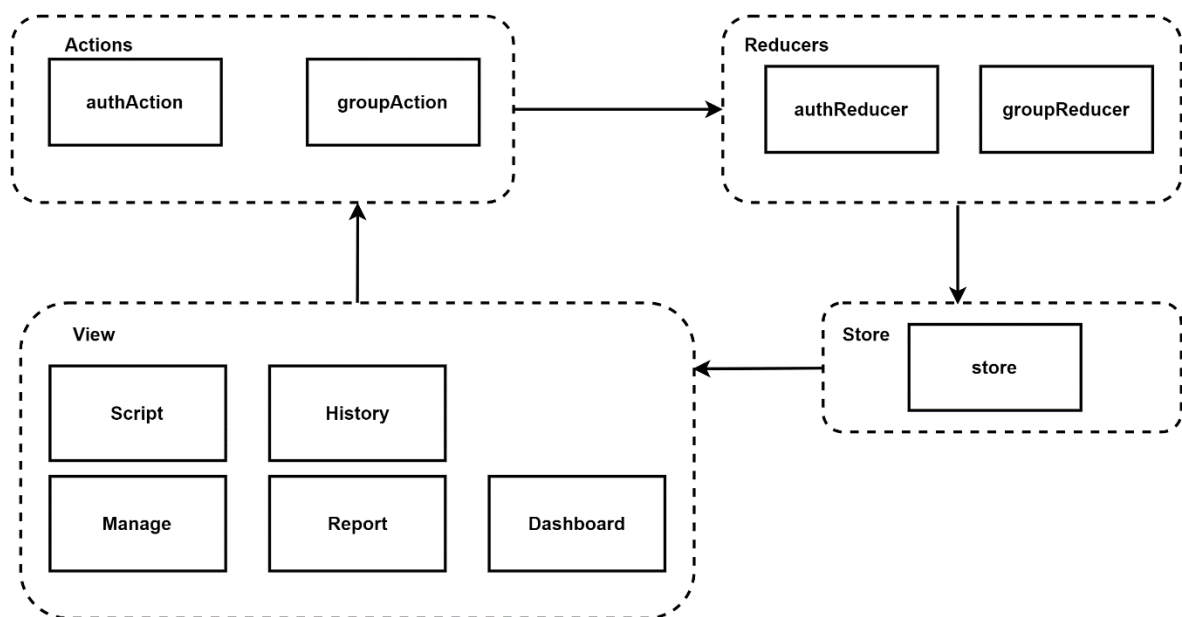


**Hình 9** Kiến trúc tổng quan của hệ thống

**Hình 9** mô tả kiến trúc tổng quan của hệ thống. Hệ thống được chia thành hai thành phần Frontend và Backend giao tiếp với nhau qua giao thức HTTP. Bên Frontend, em sử dụng Redux để quản lý các trạng thái chung của ứng dụng. Khi một component View muốn thay đổi sẽ gửi một Action. Khi đó Reducer sẽ nhận được Action và xử lý logic để cập nhật trạng

thái trong Store. View sẽ lắng nghe sự thay đổi đó và thực hiện render lại. Khi muốn giao tiếp với Backend, phía Frontend sẽ gửi yêu cầu lên Backend qua các endpoint. Các endpoint này được quản lý trong Route. Với mỗi route là một Controller để xử lý. Controller sẽ gọi Service để xử lý nghiệp vụ. Service này có thể nằm bên trong hệ thống hoặc ở hệ thống AI Core. AI Core là một dịch vụ bên ngoài đã được đóng gói để xử lý tin nhắn khi chat tương tác và không nằm trong phạm vi của đề án này. Các Service bên trong hệ thống sẽ kết nối với cơ sở dữ liệu qua Model và xử lý logic của nghiệp vụ. Sau toàn bộ quá trình trên, Frontend sẽ nhận HTTP Response và hiển thị giao diện cho người dùng.

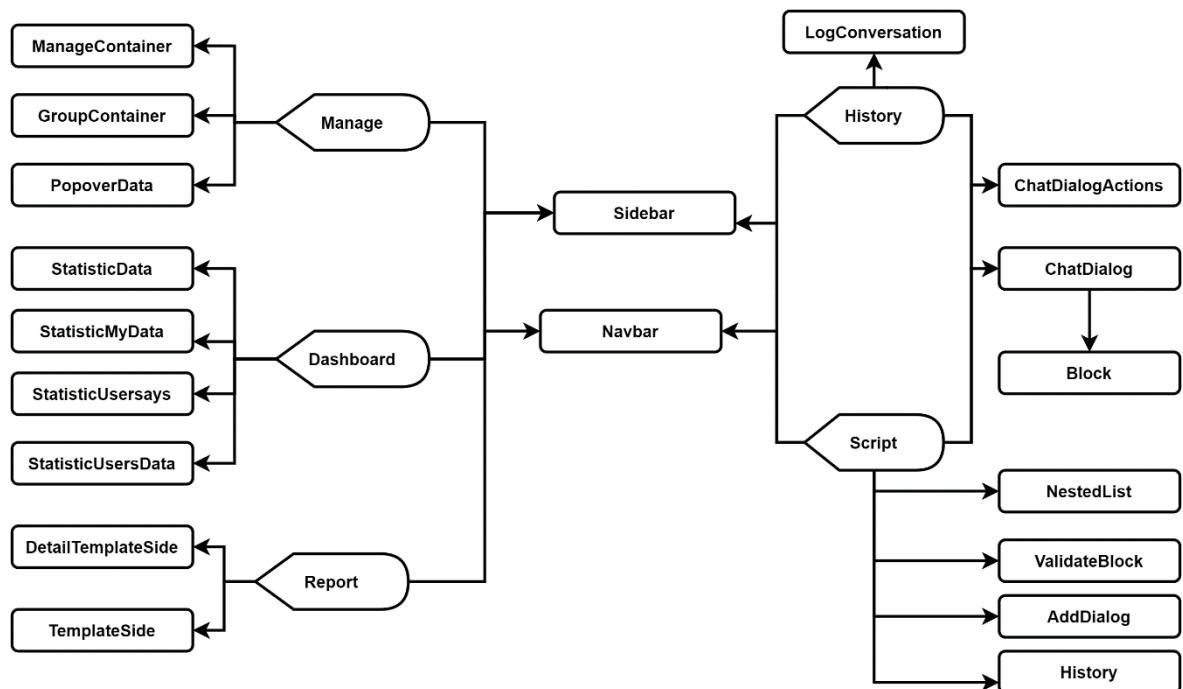
#### 4.1.2 Kiến trúc Frontend



**Hình 10** Kiến trúc Frontend

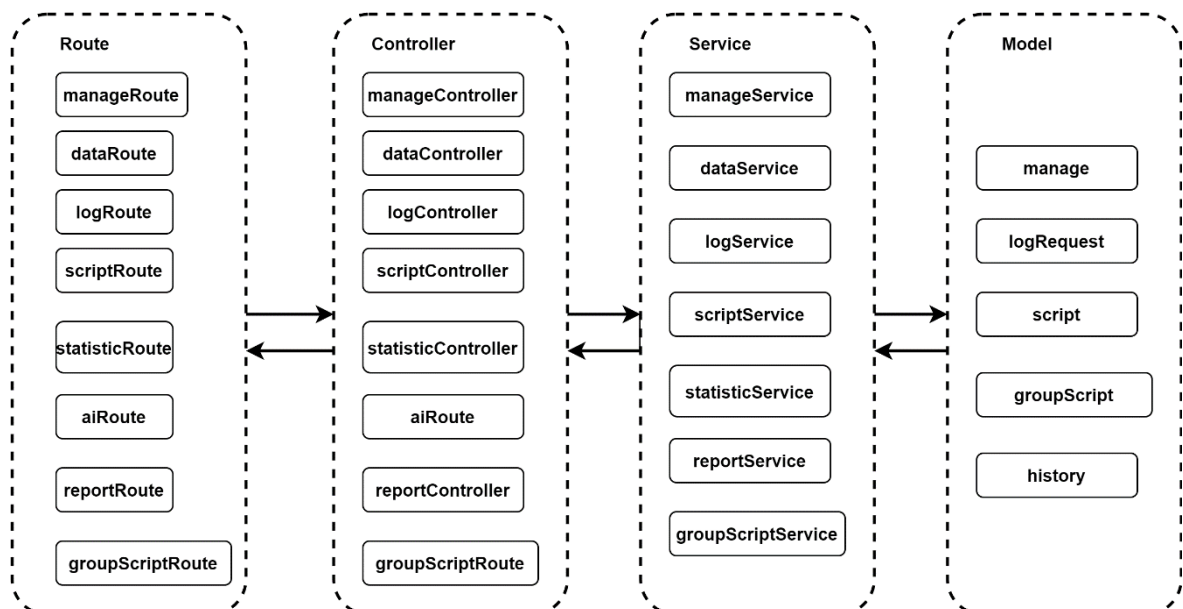
**Hình 10** mô tả kiến trúc Frontend của hệ thống tạo dữ liệu huấn luyện từ hội thoại. Kiến trúc này gồm bốn thành phần: View, Actions, Reducers và Store. View chứa các React Component, có năm component chính tương ứng với các màn hình chính là Script, History, Manage, Report và Dashboard. Trong đó, component Script hiển thị màn hội thoại, nơi quản lý các hội thoại để huấn luyện. Component History ứng với màn hình xem danh sách lịch sử chat và xem chi tiết lịch sử chat. Cuối cùng, các màn hình tạo báo cáo, phân quyền cho quản trị dữ liệu, xem tình hình hoạt động của bot sẽ lần lượt ứng với các component Report, Manage và Dashboard. Hệ thống sử dụng Redux để quản lý hai trạng thái của ứng dụng là xác thực, phân quyền và nhóm các hội thoại. Sở dĩ Redux chỉ quản lý hai phần này do chúng được chia sẻ và sử dụng nhiều giữa các component khác nhau. Để quản lý hai trạng thái trên, kiến trúc có thêm các thành phần nằm trong Actions là groupAction và authAction. Khi cần thay đổi trạng thái, các component ở View sẽ tạo một hành động nằm trong groupAction và authAction. Khi đó, các Reducer là groupReducer, authReducer sẽ nhận được sự kiện và xử

lý logic để thay đổi trạng thái ở store. Khi store thay đổi trạng thái, các component kết nối tới store sẽ nhận được các thay đổi tương ứng. Chi tiết các thành phần trong màn hình được mô tả ở **Hình 11**. Nhờ vào tính tái sử dụng của React, em tái sử dụng được các thành phần như Sidebar, Navbar, ChatDialogActions, ChatDialog, Block.



**Hình 11** Các thành phần của Frontend

### 4.1.3 Kiến trúc Backend



**Hình 12** Kiến trúc Backend



Như đã nêu ở phần kiến trúc tổng quan, Backend được chia làm 4 thành phần Route, Controller, Service, Model. Chi tiết của mỗi thành phần được mô tả ở **Hình 12**. Luồng xử lý dữ liệu sẽ đi lần lượt từ Route, Controller, Service, Model. Mỗi service sẽ tương ứng với một nghiệp vụ trong hệ thống.

## 4.2 Thiết kế chi tiết Frontend

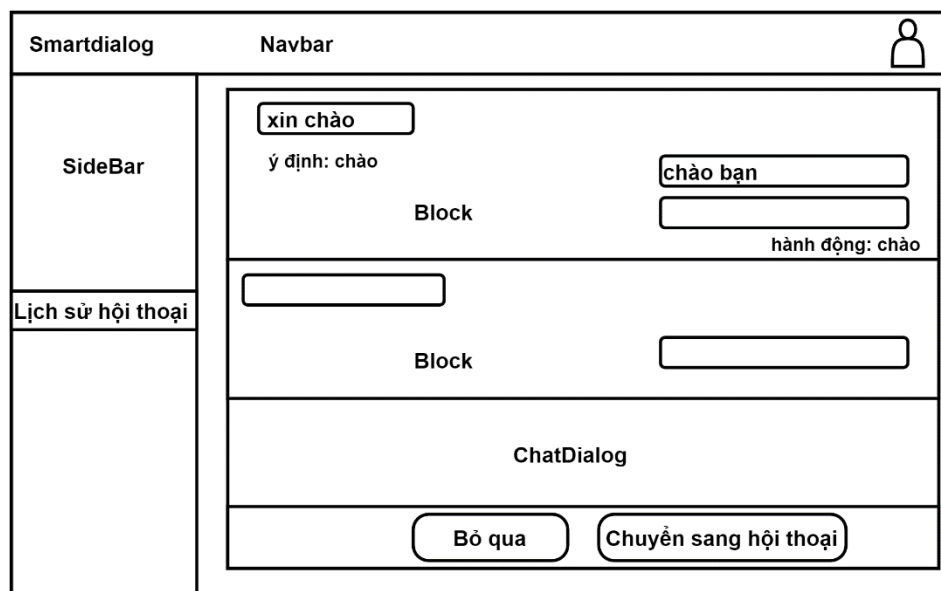
Hệ thống của em gồm các màn hình chính sau: Màn hình “Xem danh sách lịch sử chat”, “Xem chi tiết lịch sử chat”, “Tạo hội thoại từ lịch sử”, “Xem dashboard tình hình hoạt động của bot”, “Tạo báo cáo” và “Phân quyền cho quản trị viên”.

**Hình 13** mô tả mockup của màn hình “Xem danh sách lịch sử chat”. Màn này có ba thành phần chính là Navbar, Sidebar và LogConversation. Navbar thể hiện tên hệ thống và các chức năng liên quan tới người dùng. Sidebar là thanh điều hướng tới các chức năng khác nhau. LogConversation bao gồm các thành phần tìm kiếm và thành phần hiển thị danh sách chat. Thành phần tìm kiếm gồm các trường tìm kiếm về khoảng thời gian, kênh, trạng thái, v.v. Danh sách hiển thị bao gồm tên đoạn hội thoại, thời gian bắt đầu và trạng thái.

Smartdialog		Navbar										
SideBar	<div> <div>15/06/2020 - 21/06/2020</div> <div>Facebook</div> <div>Chưa xem</div> </div>											
	<table border="1"> <thead> <tr> <th>Đoạn hội thoại</th> <th>Bắt đầu</th> <th>Trạng thái</th> </tr> </thead> <tbody> <tr> <td>mở thẻ</td> <td>19-06-2020</td> <td>Chưa xem</td> </tr> <tr> <td>xin chào</td> <td>19-06-2020</td> <td>Chưa xem</td> </tr> </tbody> </table>				Đoạn hội thoại	Bắt đầu	Trạng thái	mở thẻ	19-06-2020	Chưa xem	xin chào	19-06-2020
Đoạn hội thoại	Bắt đầu	Trạng thái										
mở thẻ	19-06-2020	Chưa xem										
xin chào	19-06-2020	Chưa xem										
Lịch sử hội thoại	LogConversation											
		Số hội thoại mỗi trang: 10		1/3								

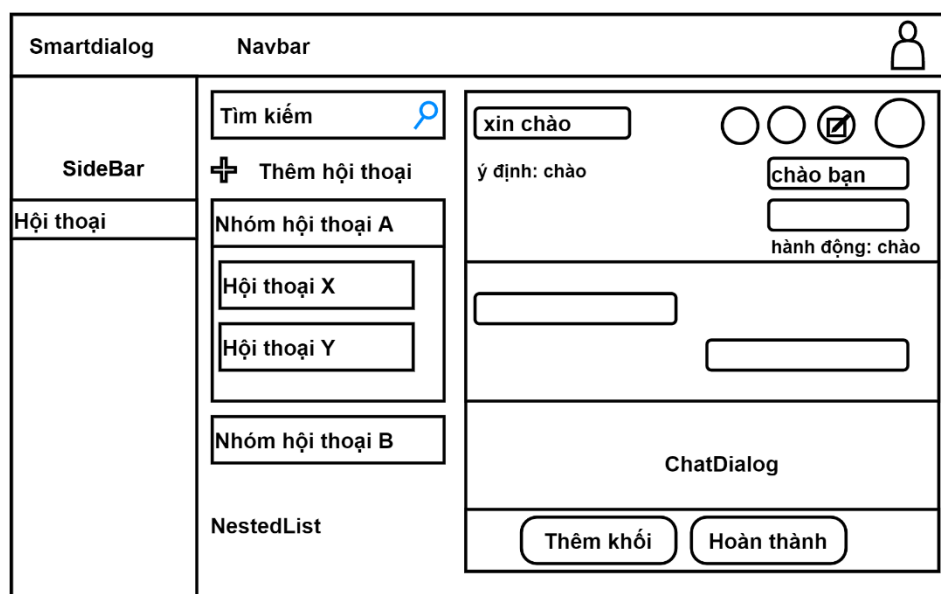
**Hình 13** Mockup màn hình “Xem danh sách lịch sử chat”

Sau khi có được danh sách, người dùng chọn một đoạn hội thoại để xem chi tiết lịch sử chat. Màn hình “Xem chi tiết lịch sử chat” được mô tả ở **Hình 14**. Màn hình này bao gồm thành phần hiển thị danh sách các đơn vị hội thoại. Phía dưới thành phần đó là các nút bấm bao gồm đánh dấu bỏ qua lịch sử chat và chuyển lịch sử chat này thành hội thoại. Sau khi ấn nút chuyển thì màn hình nhảy ra tab mới để bắt đầu xây dựng dữ liệu



**Hình 14** Mockup màn hình “Xem chi tiết lịch sử chat”

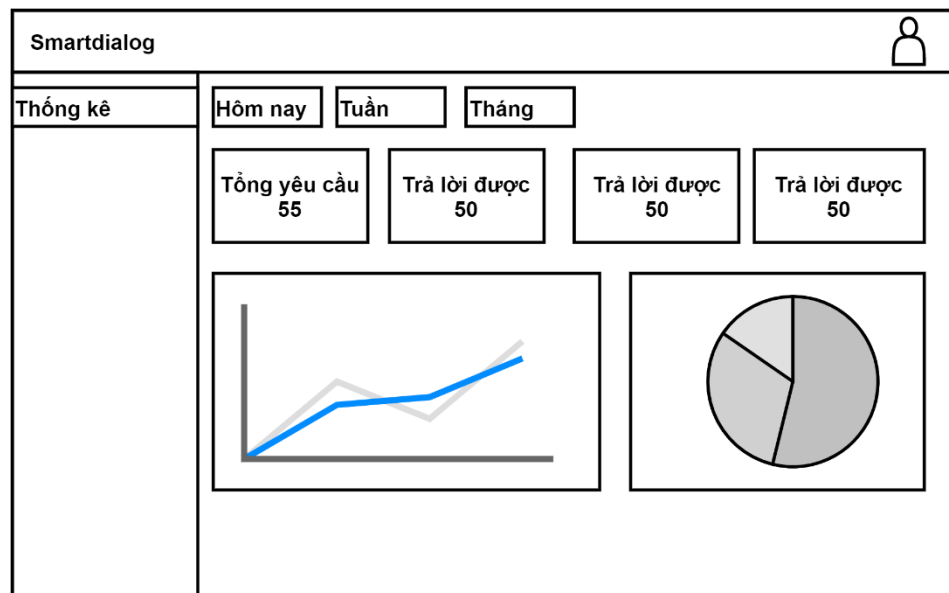
Chức năng “Tạo hội thoại từ lịch sử” được mô tả ở **Hình 15**. Giao diện vẫn hiển thị ChatDialog giống giao diện “Xem chi tiết lịch sử chat”. Tuy vậy, các đơn vị giờ có thêm các hành động để thay đổi. Cùng với đó, phía bên trái ChatDialog, có thành phần NestedList để liệt kê và tìm kiếm các hội thoại đã có. Ngoài ra, NestedList này có nút “Thêm hội thoại” để chuyển sang chức năng “Tạo hội thoại qua chat tương tác”.



**Hình 15** Mockup màn hình “Tạo hội thoại từ lịch sử chat”

**Hình 16** mô tả mockup màn hình Tạo báo cáo. Màn hình này gồm hai thành phần. Thành phần bên trái để chọn mẫu báo cáo, các trường hiển thị. Thành phần bên phải để xem trước hoặc xuất báo cáo.

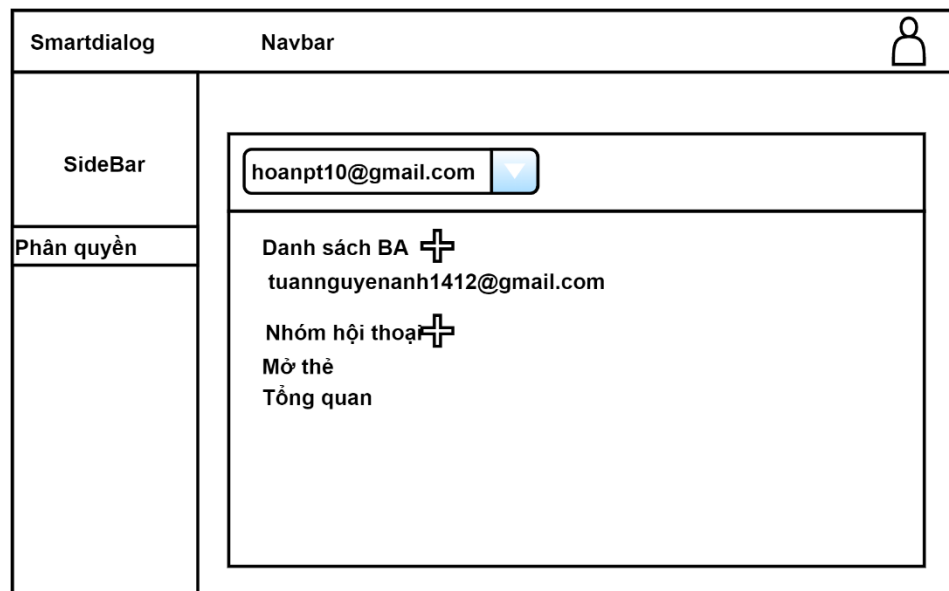
**Hình 16** Mockup màn hình “Tạo báo cáo”



**Hình 17** Mockup màn hình “Xem dashboard tình hình hoạt động của bot”

**Hình 17** mô tả giao diện trang dashboard của quản trị viên. Ở trang này sẽ hiển thị các biểu đồ, các thông số mà hệ thống đã dựa vào dữ liệu lịch sử và dữ liệu huấn luyện. Từ đó quản trị viên có thể đánh giá độ hiệu quả của bot.

Màn hình em mô tả giao diện cuối cùng là phân quyền cho quản trị dữ liệu. **Hình 18** mô tả mockup màn hình này. Màn hình gồm thành phần hiển thị quản trị dữ liệu đang được phân quyền. Thành phần thứ hai là các biên tập dữ liệu và các nhóm dữ liệu phân quyền cho quản trị dữ liệu đó. Nếu cần chỉnh sửa, quản trị viên chọn nút dấu cộng và thêm bớt các mục dữ liệu tương ứng.

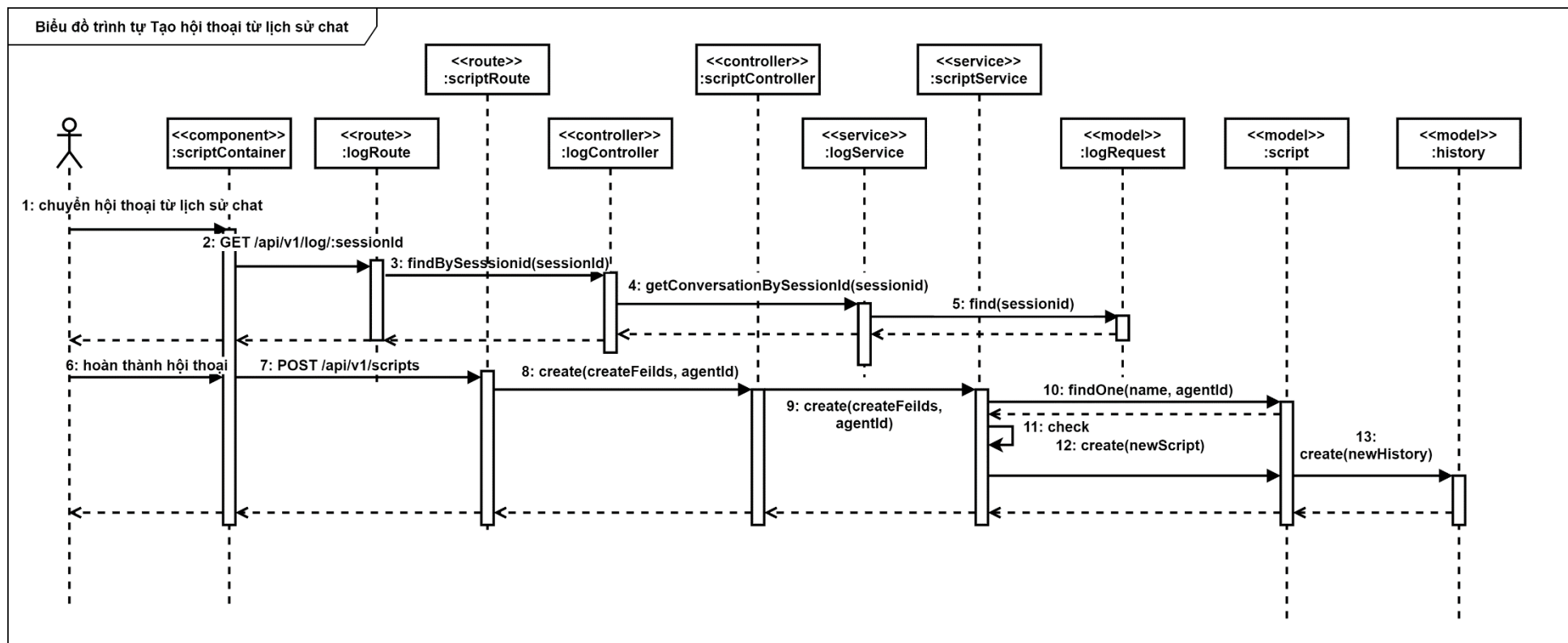


**Hình 18** Mockup màn hình “Phân quyền cho quản trị dữ liệu”

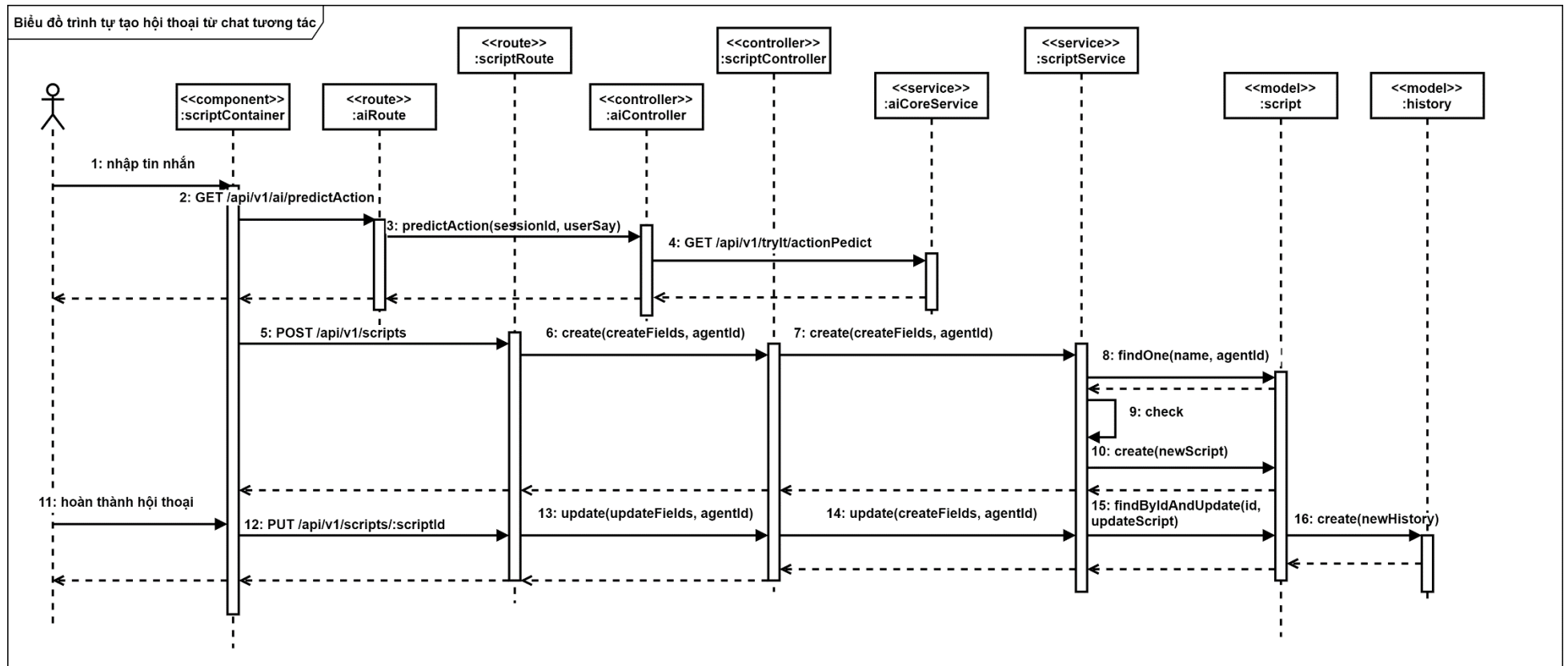
## 4.3 Thiết kế chi tiết Backend

### 4.3.1 Luồng hoạt động

Do phạm vi báo cáo có giới hạn nên phần này em chỉ trình bày biểu đồ trình tự hai use case chính là “Tạo hội thoại huấn luyện từ lịch sử chat” và “Tạo hội thoại huấn luyện từ chat tương tác”. Với use case “Tạo hội thoại huấn luyện từ lịch sử chat”, người biên tập dữ liệu kích hoạt chức năng qua giao diện. Lúc này component phía frontend sẽ gửi một HTTP Request lên server và được LogRoute tiếp nhận. Sau đó LogRoute chuyển đến LogController và yêu cầu được chuyển tiếp đến LogService. LogService có nhiệm vụ lấy chi tiết lịch sử chat ở cơ sở dữ liệu qua LogRequest rồi dữ liệu được trả về giao diện hiển thị. Sau đó, người làm dữ liệu chọn chức năng hoàn thành để tạo một hội thoại. Sau khi chọn chức năng này, giao diện tiếp tục gửi một HTTP Request đến ScriptRoute. ScriptRoute sẽ chuyển tiếp đến ScriptController, ScriptService. ScriptService kiểm tra trùng lặp trước rồi tạo một hội thoại và một phiên bản lịch sử của hội thoại đó. Kết quả sẽ trả về người dùng vẫn là HTTP Response. Với use case “Tạo hội thoại qua chat tương tác”, người làm dữ liệu bắt đầu bằng cách nhập và gửi tin nhắn qua giao diện. Hệ thống sẽ xử lý tin nhắn qua dịch vụ bên ngoài là AICoreService. Sau khi tin nhắn trả về, hệ thống sẽ tự động lưu thành hội thoại có trạng thái nháp. Quá trình diễn ra cho đến khi người dùng chọn chức năng hoàn thành hội thoại ở giao diện. Lúc này, component bên giao diện sẽ gửi yêu cầu lên server. Luồng đi của yêu cầu sẽ tương tự là ScriptRoute, ScriptController và ScriptService. ScriptService có nghiệp vụ cập nhật nội dung hội thoại với tên và các trường liên quan, đồng thời lưu thêm một phiên bản lịch sử của hội thoại để người làm dữ liệu có thể kiểm tra và truy vấn sau này.



**Hình 19** Biểu đồ trình tự Tạo hội thoại từ lịch sử chat



**Hình 20** Biểu đồ trình tự Tạo hội thoại qua chat tương tác

### 4.3.2 Thiết kế API

Sau khi thiết kế kiến trúc và thiết kế chi tiết các luồng hoạt động, em sẽ mô tả các API sử dụng trong hệ thống này. Qua quá trình hoàn thiện ĐATN, em thiết kế được 30 API cho hệ thống xây dựng dữ liệu huấn luyện từ hội thoại với nhiều nghiệp vụ khác nhau như tạo hội thoại, thống kê, báo cáo, v.v.

**Bảng 8** liệt kê danh sách API với 3 trường là mục đích, phương thức và địa chỉ. Địa chỉ đường dẫn chính là địa chỉ để Frontend giao tiếp tới.

**Bảng 8** Danh sách API trong hệ thống

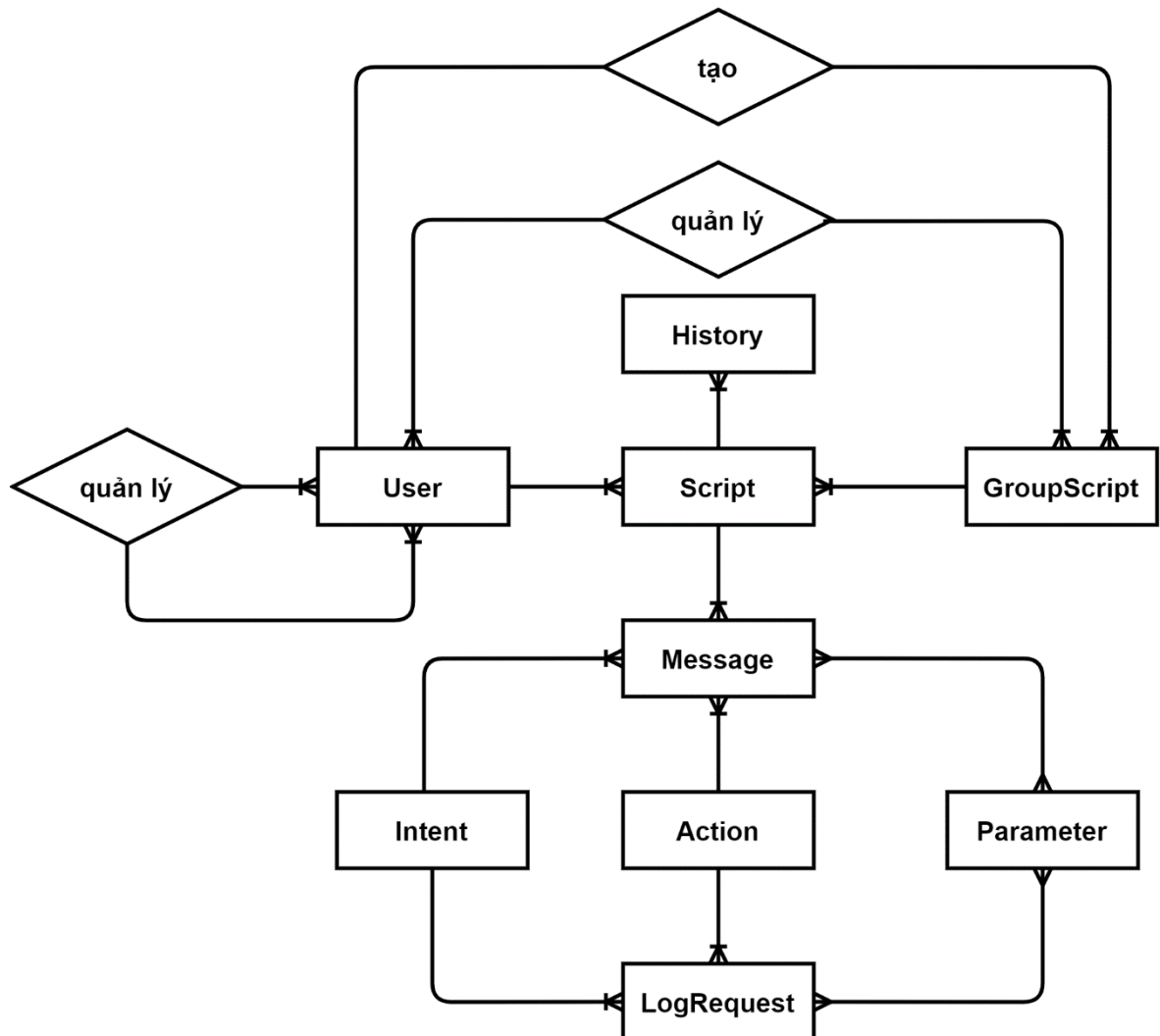
Mục đích	Phương thức	Địa chỉ
Lấy chi tiết một hội thoại	GET	/api/v1/scripts/:scriptId
Tạo một hội thoại	POST	/api/v1/scripts
Sửa một hội thoại	PUT	/api/v1/scripts/:scriptId
Xóa một hội thoại	DELETE	/api/v1/scripts/:scriptId
Gán hội thoại cho biên tập viên	PUT	/api/v1/:scriptId/assign
Xem lịch sử chỉnh sửa	GET	/api/v1/:scriptId/history
Xem danh sách lịch sử chat	GET	/api/v1/logs
Xem chi tiết lịch sử chat	GET	/api/v1/logs/:sessionId
Đánh dấu bỏ qua lịch sử chat	POST	/api/v1/logs/:sessionId/validate
Dự đoán hành động khi người dùng nhấn tin	GET	/api/v1/ai/predict/action
Lấy danh sách hội thoại và nhóm	GET	/api/v1/groupScripts/getGroupAndItems
Tạo một nhóm hội thoại	POST	/api/v1/groupScripts
Sửa một nhóm hội thoại	PUT	/api/v1/groupScripts/:groupScriptId

Mục đích	Phương thức	Địa chỉ
Xóa nhóm hội thoại	DELETE	/api/v1/groupScripts/:groupScriptId
Lấy danh sách biên tập viên và nhóm dữ liệu được phân quyền của quản trị dữ liệu	GET	/api/v1/manages/:agentId
Cập nhật danh sách biên tập viên và nhóm dữ liệu được phân quyền của quản trị dữ liệu	PUT	/api/v1/manages/:agentId
Thống kê số lượng tin nhắn của người dùng trong một khoảng thời gian	GET	/api/v1/statistics/usersays
Thống kê dữ liệu số lượng dữ liệu xây dựng và phê duyệt theo thời gian	GET	/api/v1/statistics/data
Thống kê số lượng yêu cầu bị chỉnh sửa khi xây dựng dữ liệu từ hội thoại	GET	/api/v1/statistics/scriptFromLog
Lấy danh sách dữ liệu xây dựng của bản thân	GET	/api/v1/data/createdBy
Lấy danh sách dữ liệu xây dựng của các biên tập viên mình quản lý	GET	/api/v1/data/userIds
Thêm câu mẫu vào ý định	PUT	/api/v1/intents/:intentId/addUsersay
Gỡ câu mẫu khỏi ý định	PUT	/api/v1/intents/:intentId/removeUsersay
Xem trước báo cáo	POST	/api/v1/reports/dataPreview
Xuất báo cáo	POST	/api/v1/reports



## 4.4 Thiết kế cơ sở dữ liệu

### 4.4.1 Biểu đồ thực thể liên kết

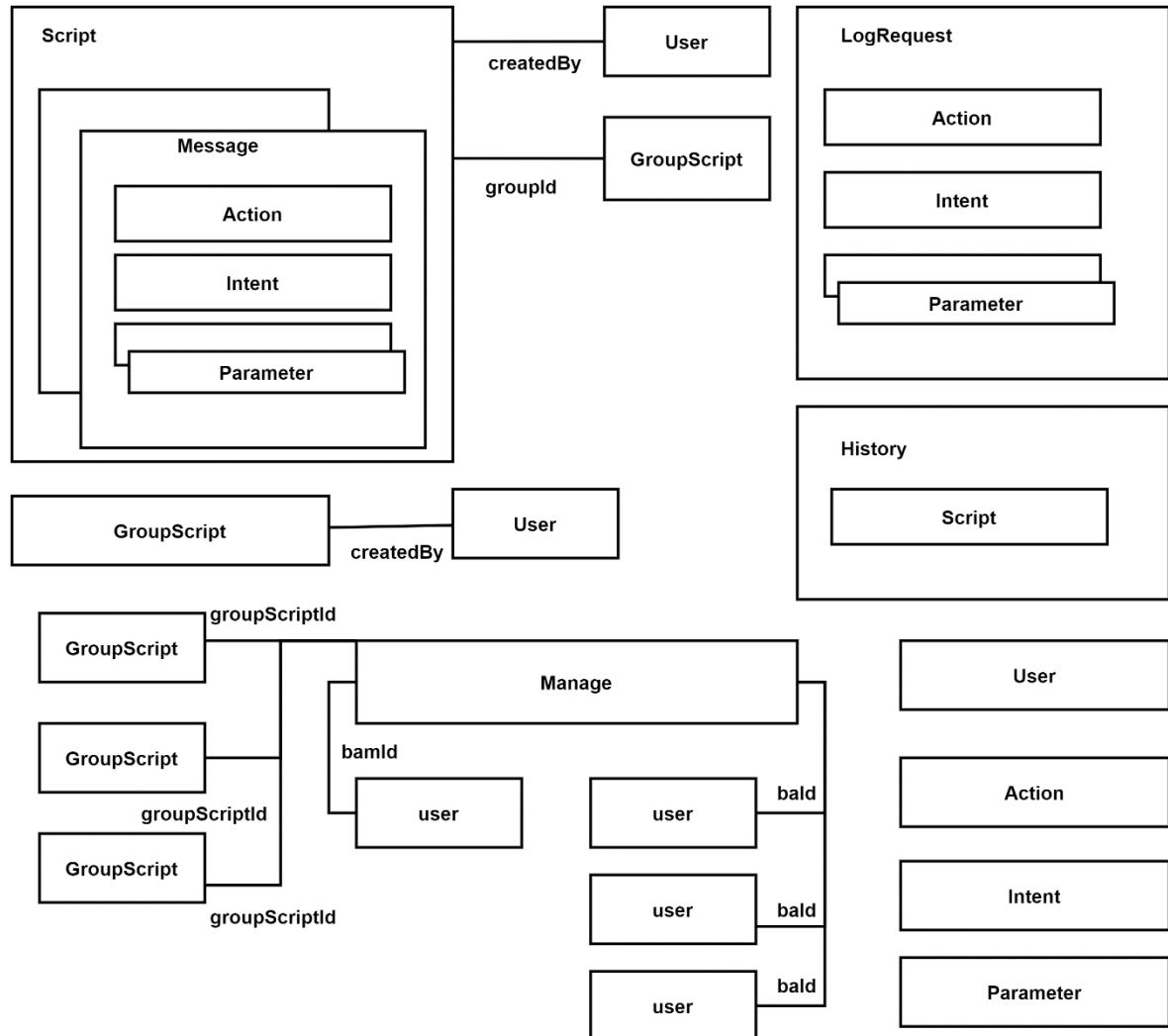


**Hình 21** Biểu đồ thực thể quan hệ

**Hình 21** thể hiện biểu đồ thực thể liên kết bao gồm các thực thể Script, LogRequest, Intent, Action, Paramters, Message, GroupScript và User. Thực thể Script hướng tới mô tả các hội thoại được tạo từ lịch sử chat hoặc từ chat tương tác. Một hội thoại (Script) sẽ chứa nhiều tin nhắn (Message) và nhiều phiên bản lịch sử chỉnh sửa (History). Trong thực thể Message này có thể chứa một ý định (Intent), một hành động (Action) hoặc nhiều tham số (Parameter). Thực thể Intent chỉ ý định của yêu cầu đầu vào, Action chỉ hành động phản hồi của bot còn Parameter biểu diễn các thực thể được tham số hóa trong câu đầu vào của người dùng. Tương tự như vậy với Log Request thì có chứa một ý định, một hành động hoặc nhiều tham số. Thực thể LogRequest này để lưu lịch sử dự đoán ý định và phản hồi của bot khi người dùng nhấn tới các kênh chat như Facebook, Zalo. Thực thể GroupScript tương ứng với nhóm hội

thoại. Một nhóm hội thoại bao gồm không hoặc nhiều hội thoại. Thực thể User chỉ người dùng hệ thống với vai trò quản trị viên, quản trị dữ liệu và biên tập dữ liệu. Người dùng này có thể tạo hội thoại, tạo nhóm hội thoại, quản lý các nhóm hội thoại hoặc quản lý nhóm người dùng khác.

#### 4.4.2 Thiết kế tổng quan cơ sở dữ liệu



**Hình 22** Thiết kế tổng quan cơ sở dữ liệu

Dựa vào biểu đồ thực thể liên kết, em thiết kế tổng quan cơ sở dữ liệu theo dạng NoSQL được mô tả **Hình 22**. Với cơ sở dữ liệu không quan hệ NoSQL, ta không nhất thiết phải tuân theo các chuẩn hóa như cơ sở dữ liệu quan hệ. Từ thiết kế tổng quan cơ sở dữ liệu, em đi vào thiết kế chi tiết cho từng collection.

#### 4.4.3 Thiết kế chi tiết collection

Do giới hạn của độ dài ĐATN cũng như giới hạn phạm vi hệ thống là tạo dữ liệu huấn luyện từ hội thoại, em sẽ trình bày một số collection chính. Các collection này gắn liền với các

chức năng chính của hệ thống. Các collection đó bao gồm Script để biểu diễn hội thoại, LogRequest thể hiện yêu cầu của người dùng và phản hồi của bot trong lịch sử, GroupScript để biểu diễn nhóm hội thoại, Manage để lưu thông tin xem quản trị dữ liệu quản lý những biên tập viên nào, các nhóm dữ liệu nào.

**Bảng 9** Thiết kế chi tiết Script Collection

Tên trường	Kiểu dữ liệu	Ý nghĩa
_id	ObjectId	Id của hội thoại
mainMessages	Array Object	Danh sách message của hội thoại
name	String	Tên của hội thoại dùng để huấn luyện
displayName	String	Tên hiển thị của hội thoại
groupId	String	Id của nhóm hội thoại
sessionId	String	Session Id của phiên hội thoại lịch sử
status	String	Trạng thái của hội thoại
trainingScript	String	Dạng dữ liệu dùng để huấn luyện
agentId	String	Id của chatbot
createdBy	ObjectId	Id của người tạo
approvedBy	ObjectId	Id của người duyệt
createdAt	Timestamp	Thời gian tạo
updatedAt	Timestamp	Thời gian cập nhật

**Bảng 10** Thiết kế chi tiết LogRequest Collection

Tên trường	Kiểu dữ liệu	Ý nghĩa
_id	ObjectId	Id của logRequest
nlu	Object	Thông tin về intent và các parameter
action	Object	Thông tin về action
result	Object	Nội dung trả về của hành động
userSay	String	Nội dung yêu cầu đầu vào của người dùng
sessionId	String	Id của phiên hội thoại lịch sử
status	String	Trạng thái của logRequest
agentId	String	Id của chatbot
createdAt	Timestamp	Thời gian tạo

**Bảng 11** Thiết kế chi tiết GroupScript Collection

Tên trường	Kiểu dữ liệu	Ý nghĩa
_id	ObjectId	Id của Group Script
name	String	Tên hiển thị của nhóm
agentId	String	Id của chatbot
createdAt	Timestamp	Thời gian tạo

**Bảng 12** Thiết kế chi tiết Manage Collection

Tên trường	Kiểu dữ liệu	Ý nghĩa
_id	ObjectId	Id của Manage

bamId	ObjectId	Id của người quản trị dữ liệu
baIds	Array ObjectId	Danh sách người biên tập dữ liệu được quản trị dữ liệu quản lý
groupScripts	Array ObjectId	Danh sách nhóm hội thoại được quản trị dữ liệu quản lý
agentId	String	Id của chatbot
createdAt	Timestamp	Thời gian tạo

**Bảng 13** Thiết kế chi tiết History Collection

Tên trường	Kiểu dữ liệu	Ý nghĩa
_id	ObjectId	Id của History
type	String	Kiểu dữ liệu cần lưu lịch sử. Ở đây là Script
data	Object	Dữ liệu document Script cần lưu
userId	Id	Người hoàn thành hoặc phê duyệt dữ liệu
agentId	String	Id của chatbot
createdAt	Timestamp	Thời gian tạo
updatedAt	Timestamp	Thời gian cập nhật

## 4.5 Xây dựng ứng dụng

### 4.5.1 Thư viện và công cụ sử dụng

Trong quá trình phát triển và hoàn thiện đồ án, em sử dụng một số công cụ và ngôn ngữ lập trình như sau. Các framework này đã được mô tả ở chương 3, ngoài ra em sử dụng thêm các công cụ khác như Robo3T, Visual Studio Code.

**Bảng 14** Thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
GUI kết nối MongoDB	Robo3T	<a href="https://robomongo.org">https://robomongo.org</a>
IDE lập trình	Visual Studio Code	<a href="https://code.visualstudio.com">https://code.visualstudio.com</a>
Framework Frontend	NextJS	<a href="https://nextjs.org">https://nextjs.org</a>
Framework Backend	ExpressJS	<a href="https://expressjs.com">https://expressjs.com</a>
Cơ sở dữ liệu	MongoDB	<a href="https://www.mongodb.com">https://www.mongodb.com</a>

#### 4.5.2 Kết quả đạt được

Hiện tại hệ thống đã được hoàn thành và đóng gói. Hệ thống đã được tích hợp trong nền tảng tạo và huấn luyện chatbot Smartdialog, triển khai thử nghiệm với đối tác Vietcombank trong lĩnh vực ngân hàng và đối tác FE Credit trong lĩnh vực tài chính. Các chức năng chính của ứng dụng là i) xem danh sách lịch sử trò chuyện người dùng ii) tạo hội thoại huấn luyện từ lịch sử trò chuyện và chat tương tác iii) quy trình duyệt hội thoại để đưa vào huấn luyện iv) thống kê và báo cáo tình trạng dữ liệu huấn luyện và triển khai của bot trong thực tế. Các thông số được trình bày ở **Bảng 15**.

**Bảng 15** Thông số chính của hệ thống

Thông số	Giá trị
Số lượng file mã nguồn Backend	53
Số lượng file mã nguồn Frontend	164
Số lượng API	25
Số lượng component	80

#### 4.5.3 Minh họa các chức năng chính

Phần này em sẽ trình bày và mô tả một số chức năng chính như xem danh sách lịch sử chat, tạo hội thoại từ lịch sử chat, tạo hội thoại từ chat tương tác, chấp nhận hội thoại huấn luyện, xuất báo cáo, xem dashboard tình hình hoạt động của bot và phân quyền cho quản trị dữ liệu. Đây là những chức năng có màn hình giao diện khác nhau tuy nhiên cùng chung chuẩn thiết kế đảm bảo tính thống nhất của hệ thống.

### 4.5.3.1 Xem danh sách lịch sử chat

SMART DIALOG

27/05/2020 00:00 - 26/06/2020 23:59 ▼ Tất cả kênh ▼ Tất cả trạng thái ▼ Tìm theo từ khóa XÓA LỌC

☐ Bot không hiểu ☐ Bot không hiểu hỏi lại ☐ Bot trả lời mặc định

Ý định ▼ Hành động ▼

hồ sơ tat toan so tiet kiem	1	0	17-06-2020 14:03	Chưa xem
huỷ sms giúp mình	3	0	17-06-2020 13:51	Chưa xem
huỷ sms giúp anh được không	3	0	17-06-2020 13:49	Chưa xem
khoá thẻ giúp mình với	3	0	17-06-2020 13:48	Chưa xem
Chuyển nhầm tiền có lấy lại được không	2	0	17-06-2020 13:46	Chưa xem
Chuyển nhầm tiền có lấy lại được không	2	0	17-06-2020 13:45	Chưa xem
bây giờ anh cần huỷ sms	1	0	17-06-2020 13:43	Chưa xem

Số hội thoại mỗi trang 10 ▼ 1-10 của 70 |< < 1 / 7 > >|

Hình 23 Chức năng Xem danh sách lịch sử hội thoại

04-06-2020 13:32

U hồ sơ phát hành thẻ ghi nợ như nào nhỉ

Ý định: Hồ sơ phát hành thẻ (100%)

04-06-2020 13:32

Vui lòng cho mình biết bạn đang muốn hỏi hình thức ghi nợ nào nhé!

Lựa chọn 1. Nội địa (Ghi nợ nội địa, Ghi nợ trong nước)  
Lựa chọn 2. Quốc tế (Ghi nợ Visa, Ghi nợ Master, ...)

Lựa chọn 1  
Lựa chọn 2

Hành động: Xác nhận hình thức ghi nợ (100%)

04-06-2020 13:33

U nội địa

Ý định: inform 1 tham số Nhóm thẻ (97%)

04-06-2020 13:33

Để phát hành thẻ ghi nợ nội địa, bạn chỉ cần chuẩn bị giúp mình bản sao giấy CMND hoặc Hộ chiếu. Khi mang ra phòng giao dịch Vietcombank, giao dịch viên sẽ hướng dẫn bạn viết thêm các giấy:

- Yêu cầu phát hành thẻ ghi nợ nội địa;
- Giấy đề nghị mở tài khoản (nếu như bạn chưa có tài khoản tại Vietcombank)

Chỉ cần thế thôi bạn nhé.

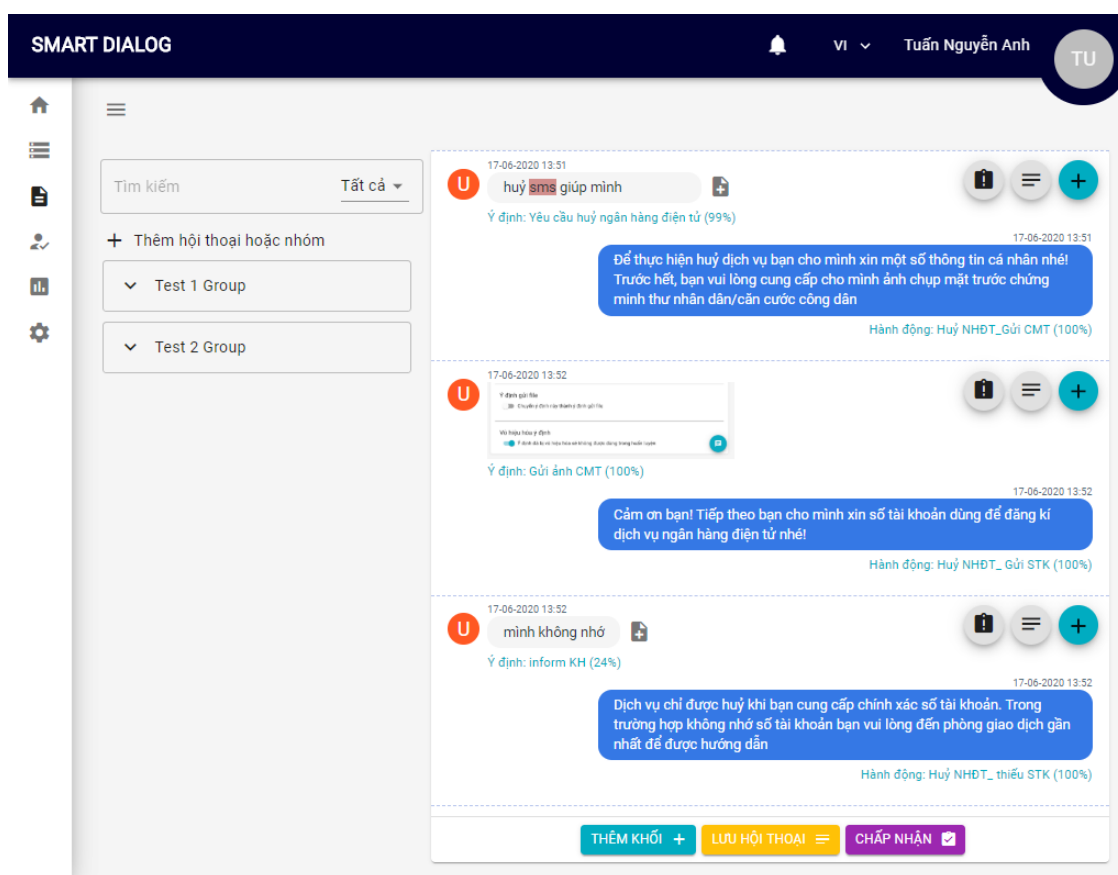
← QUAY LẠI BỎ QUA CHUYỂN THÀNH HỘI THOẠI

Hình 24 Chức năng Xem chi tiết lịch sử trò chuyện

Đây là chức năng được dùng nhiều nhất trong ứng dụng. Sau một khoảng thời gian, người làm dữ liệu, quản lý bot đều có nhu cầu xem người dùng tương tác với bot như thế nào. Do đoạn hội thoại tăng dần theo thời gian rất nhanh nên em đã thêm nhiều lựa chọn để lọc và tìm đoạn hội thoại một cách nhanh chóng. Các lựa chọn lọc đó bao gồm lọc theo khoảng thời gian, kênh chat, trạng thái, từ khóa, ý định xuất hiện trong hội thoại lịch sử.

Sau khi xem danh sách lịch sử chat, người dùng click chọn một hàng trong bảng để xem chi tiết lịch sử chat. Màn hình sẽ không chỉ hiển thị nội dung chat mà còn hiển thị một số thông số khác như ý định, thực thể trích xuất thì câu đầu vào của người dùng, hành động bot trả lại và độ tự tin của bot khi thực hiện các hành động trên. Phía dưới của nội dung lịch sử chat là các chức năng như quay lại để trở về danh sách lịch sử chat, bỏ qua để đánh dấu đoạn hội thoại không thích hợp để làm dữ liệu huấn luyện và chuyển thành hội thoại để bắt đầu quá trình xây dựng và phê duyệt dữ liệu.

#### 4.5.3.2 Tạo hội thoại từ lịch sử chat



**Hình 25** Chức năng Tạo hội thoại từ lịch sử chat

Từ giao diện xem chi tiết lịch sử chat, sau khi chọn chức năng chuyển thành hội thoại, màn hình chuyển sang trang tạo hội thoại từ lịch sử chat được mô tả ở **Hình 25**. Thành phần bên



trái để hiển thị và tìm kiếm tất cả các hội thoại có trong hệ thống. Các hội thoại được hiển thị nằm trong các nhóm hội thoại, thể hiện rõ ràng tính chất phân lớp của dữ liệu.

Chức năng tìm kiếm có các dạng tìm kiếm bao gồm từ khóa và trạng thái, giúp quản trị viên cũng như biên tập viên tập trung vào các hội thoại liên quan tới mình hơn. Thành phần bên phải hiển thị lại nội dung của lịch sử chat. Lúc này, người làm dữ liệu sẽ có đầy đủ chức năng để chỉnh sửa. Tại đây, hội thoại được biểu diễn có sự khác biệt khi được chia thành các đơn vị. Người làm dữ liệu sẽ có quyền chỉnh sửa ở các đơn vị này. Các chức năng có thể chỉnh sửa bao gồm thêm đơn vị hội thoại mới, xóa, di chuyển đơn vị hội thoại đã có. Chức năng thêm đơn vị hội thoại được mô tả ở **Hình 26**. Khi đó người sử dụng có thể thêm yêu cầu đầu vào, đánh dấu thực thể, ý định và hành động tương ứng. Người dùng thêm ý định hoặc hành động bằng cách ấn icon bút chì bên cạnh hiện popup chỉnh sửa. Bên cạnh đó, icon bên góc phải trên để thêm nhanh câu mẫu vào ý định. Sau khi hoàn thành, người làm dữ liệu chọn nút lưu. Sau khi hệ thống thực hiện xử lý, đơn vị hội thoại sẽ được thêm vào hội thoại.

Trong quá trình chỉnh xây dựng và chỉnh sửa này, cứ mỗi một hành động, hệ thống sẽ thực hiện lưu tự động hội thoại. Điều đó đảm bảo dữ liệu luôn ở trạng thái mới nhất. Nhờ vậy, biên tập viên có thể dễ dàng xây dựng dữ liệu dễ dàng hơn.

Tên tham số	Thực thể	Giá Trị
Tên tham số	Thực thể	Giá Trị
Ngân hàng điện tử	Ngân hàng điện tử	sms

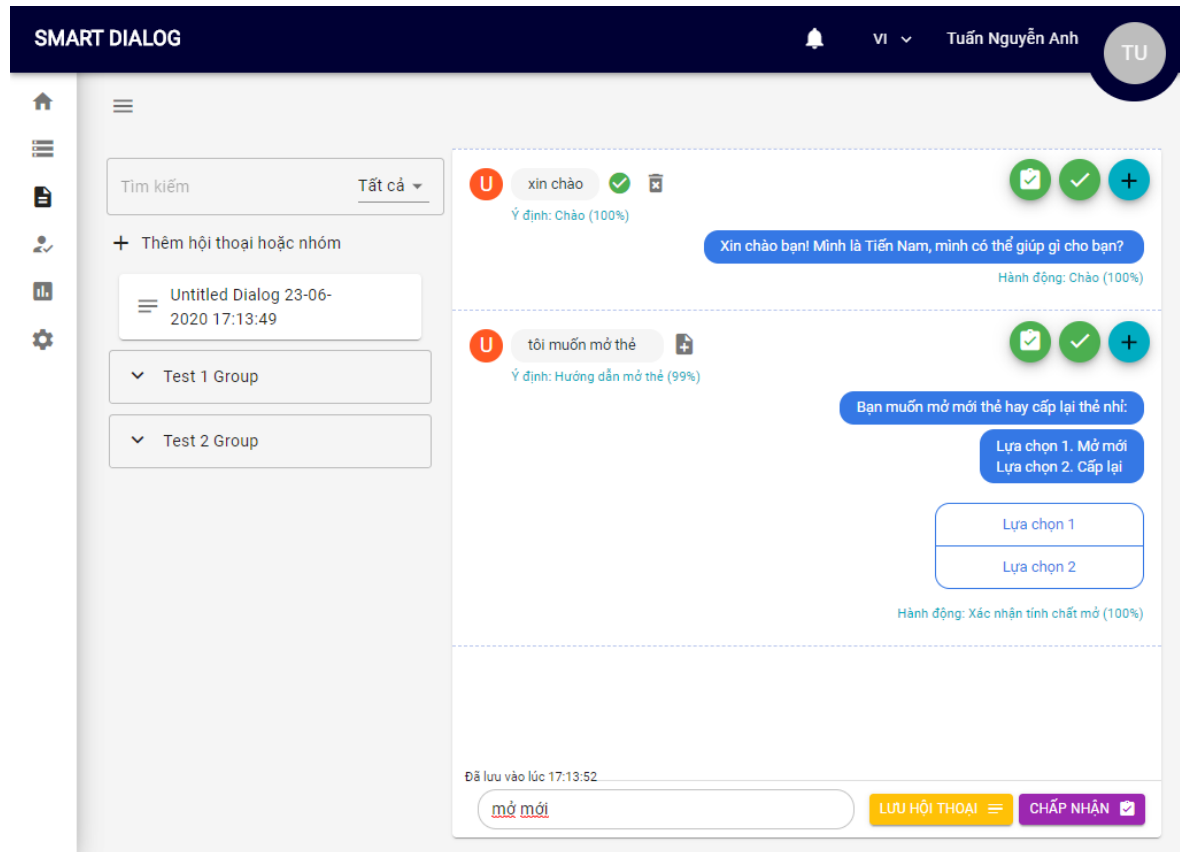
Ý định ✎  
Yêu cầu huỷ ngân hàng điện tử (99%)

Hành động ✎  
Huỷ NHĐT\_Gửi CMT (100%) 🗑

HỦY LƯU

**Hình 26** Chức năng thêm đơn vị hội thoại

### 4.5.3.3 Tạo hội thoại qua chat tương tác

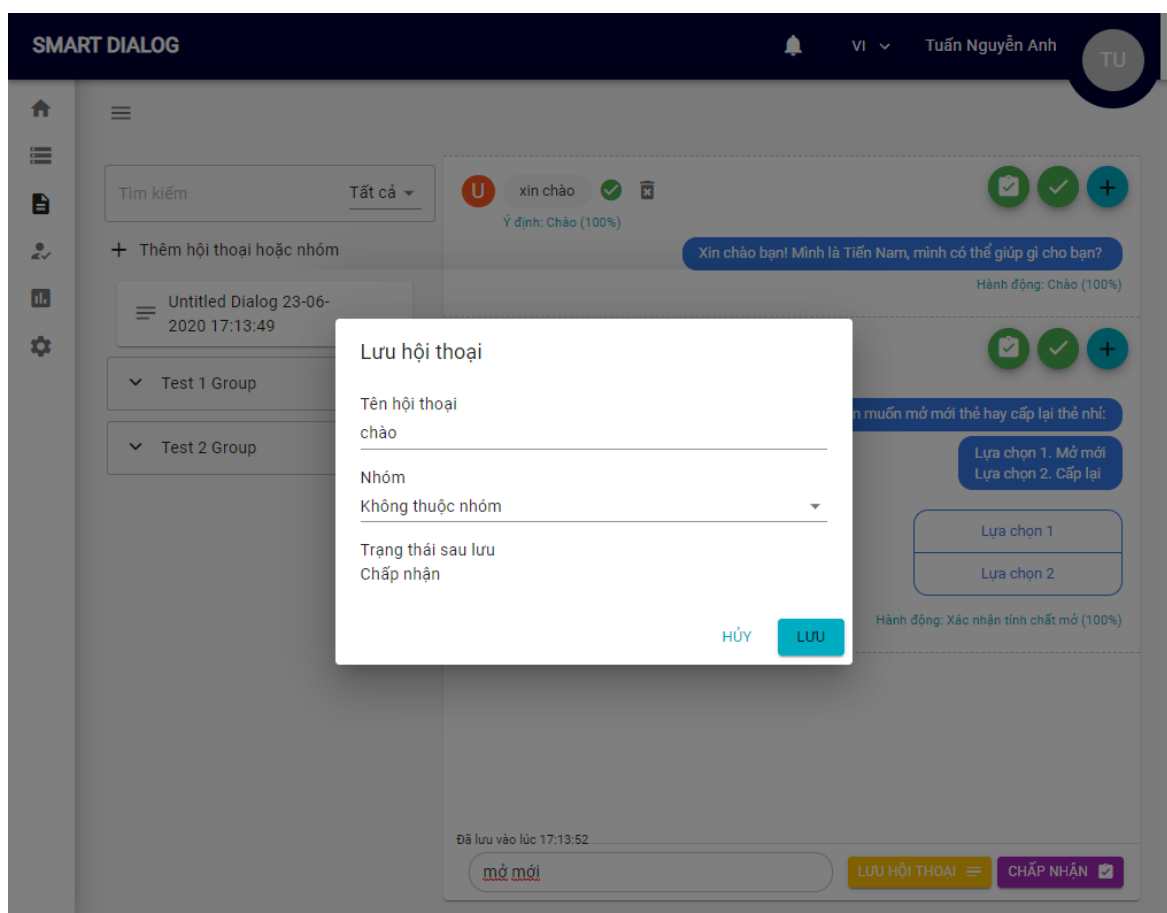


**Hình 27** Tạo hội thoại bằng chat tương tác

Ngoài tạo hội thoại bằng cách chỉnh sửa dữ liệu từ lịch sử, người làm dữ liệu có thể chat trực tiếp với bot. Chức năng chat tương tác này hiển thị giao diện y hệt người dùng chat ngoài đời thực. Câu trả lời của bot được hiển thị một cách trực quan, bao gồm dạng chữ, hình ảnh, âm thanh, video. Hội thoại lúc này cũng hiển thị thành các đơn vị hội thoại, với mỗi đơn vị hội thoại ta cũng có các chức năng chỉnh sửa như đã mô tả khi tạo hội thoại từ lịch sử chat. Do vậy người dùng vừa có thể chat vừa chỉnh sửa hành vi cho bot. Cách này sẽ rất là tiện cho người làm dữ liệu, họ chủ động chọn hội thoại đầu vào theo ý của mình. Đây là chức năng khác biệt so với các nền tảng đã khảo sát ở Chương 2.

### 4.5.3.4 Chấp nhận hội thoại huấn luyện

Quy trình cụ thể để chấp nhận hội thoại huấn luyện được mô tả kỹ hơn ở chương 5. Khi quản trị dữ liệu hoặc quản trị viên thấy hội thoại đảm bảo chất lượng để đưa vào huấn luyện, họ sẽ chọn chức năng chấp nhận để đổi trạng thái cho hội thoại. Khi đó, màn hình sẽ hiển thị popup giúp người làm dữ liệu có thể thay đổi tên và nhóm dữ liệu. Sau đó, người dùng ấn nút “Lưu” để chuyển trạng thái hội thoại sang chấp nhận. **Hình 28** mô tả giao diện của chức năng “Chấp nhận hội thoại huấn luyện” này.



**Hình 28** Chức năng chấp nhận hội thoại

#### 4.5.3.5 Xuất báo cáo

Để đánh giá chất lượng chatbot triển khai trong thực tế, hệ thống cung cấp chức năng xuất báo cáo. **Hình 29** mô tả chi tiết chức năng này. Quản trị viên có thể chọn dạng báo cáo, chọn trường cần hiển thị, khoảng thời gian và dạng file xuất ra. Hệ thống cung cấp ba dạng báo cáo là “Câu hỏi khách hàng theo trạng thái”, “Ý định khách hàng”, “Câu hỏi khách hàng theo nhóm ý định”. Mỗi dạng báo cáo lại có nhiều trường khác nhau. Các trường của báo cáo được mô tả chi tiết ở chương 2. Quản trị viên không nhất thiết phải xuất tất cả các trường đó mà có thể chủ động lược bỏ một số trường để báo cáo trở nên gọn nhẹ hơn. Ở dưới mục loại báo cáo là khoảng thời gian lọc, khoảng thời gian này tính theo tuần, tháng hoặc theo ngày bắt đầu và ngày kết thúc cụ thể tùy thuộc vào mục đích của quản trị viên. Ở phía bên phải màn hình, quản trị viên có thể xem trước dữ liệu báo cáo xuất ra như thế nào. Ngoài ra, quản trị viên có thể đổi tên báo cáo xuất ra, thay đổi logo ở trên góc trái phải của màn hình. Nếu thấy dữ liệu đó cần thiết, quản trị viên chọn chức năng xuất để tải xuống file báo cáo. Các dạng file báo cáo bao gồm file Excel, PDF và CSV. Chính nhờ việc cấu hình các loại file báo cáo như vậy giúp quản trị viên có thể dễ dàng nắm được tình hình hoạt động, hiệu quả của bot trong thực tế để cải tiến chúng.

SMART DIALOG

VI

Tuấn Nguyễn Anh

TU

TẠO MẪU

MẪU ĐÃ TẠO

Tên mẫu

Chọn số liệu

☒ Câu hỏi khác hàng theo trạng thái
 

☐ Tất cả
 ☒ Câu hỏi trả lời được
 ☒ Câu hỏi không trả lời được
 ☒ Câu hỏi bị hỏi lại
 ☐ Câu hỏi đã duyệt đúng
 ☐ Câu hỏi chỉ sửa ý định
 ☐ Câu hỏi chỉ sửa hành động
 ☐ Câu hỏi sửa cả ý định và hành động

☐ Ý định khách hàng
 ☐ Câu hỏi khách hàng theo nhóm ý định

Thời gian lọc dữ liệu

Một tuần

Loại file

☒ EXCEL
 ☐ PDF
 ☐ CSV

Tên mẫu:

Dữ liệu chọn: Câu hỏi trả lời được & Câu hỏi không trả lời được & Câu hỏi bị hỏi lại

Khoảng thời gian lọc dữ liệu: Một tuần

Dạng file xuất: EXCEL

XEM TRƯỚC

LƯU MẪU

XUẤT BÁO CÁO

LOGO

Báo cáo câu hỏi khách hàng

Thời điểm tạo

Khoảng thời gian

Tổng số

Câu hỏi trả lời được

Tổng số: 50

Câu hỏi	Hành động	Thời gian
can giay to gi de tat toan vay	ho_so_tat_toan_vay	19/06/2020 16:55
xin chào bạn hiền	chao	19/06/2020 16:53
cho tôi hỏi tí được không	hoi	19/06/2020 16:54
cho tôi hỏi tí được không	hoi	19/06/2020 16:54
Sản phẩm Thẻ	thong_tin_the	19/06/2020 09:30

Số dữ liệu mỗi trang 5 1-5 của 50 1 / 10

Câu hỏi không trả lời được

Tổng số: 2

Câu hỏi	Hành động	Thời gian
ksajld	default_fallback_action	19/06/2020 09:30
kjdlskf	default_fallback_action	19/06/2020 09:30

Số dữ liệu mỗi trang 5 1-2 của 2 1 / 1

Câu hỏi hỏi lại

Tổng số: 3

Câu hỏi	Thời gian
xin chao ban	19/06/2020 16:53
cho minh chao ban duoc khong	19/06/2020 16:53
cho toi loi ti duoc khong	19/06/2020 16:54

Số dữ liệu mỗi trang 5 1-3 của 3 1 / 1

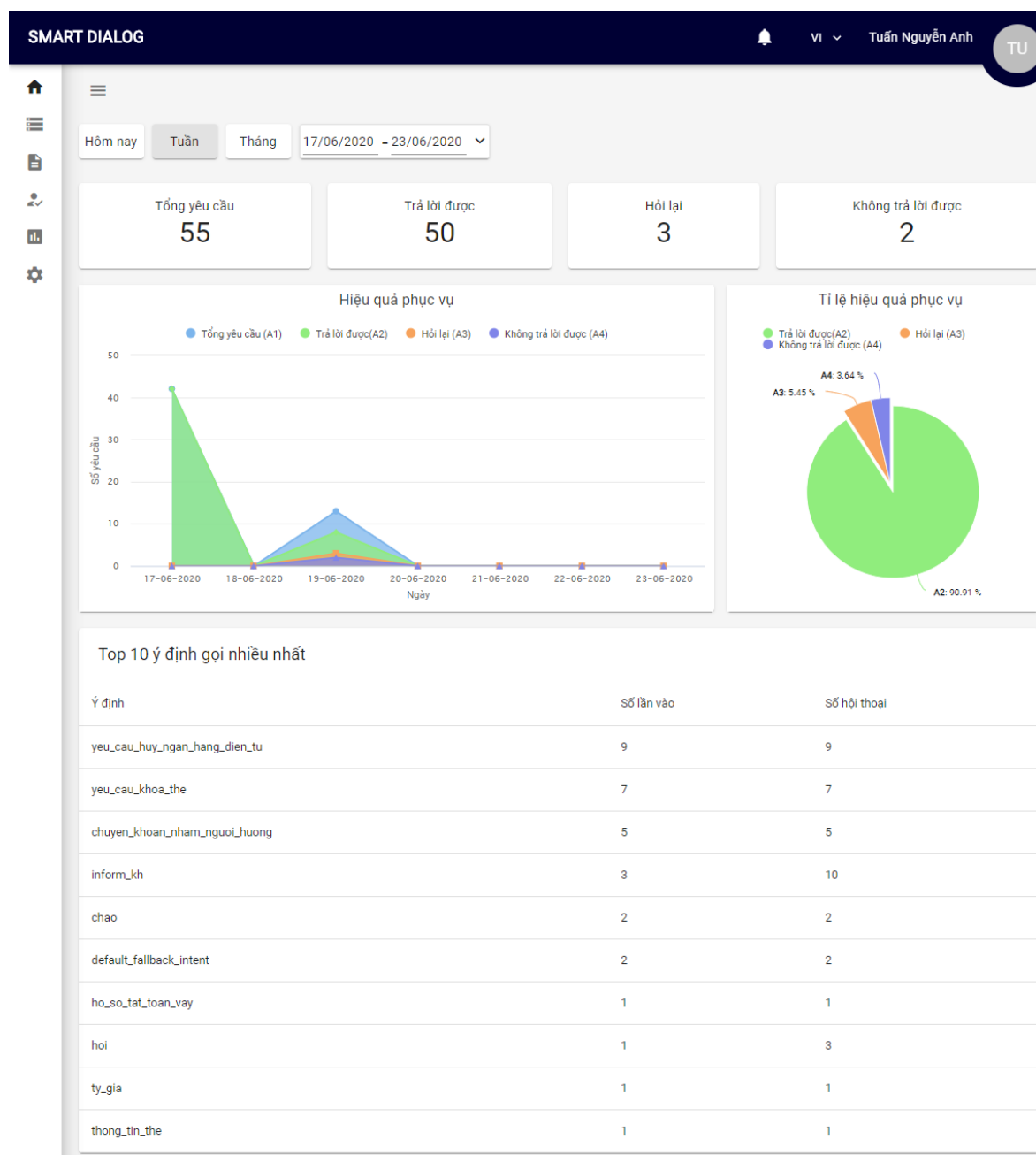
**Hình 29** Chức năng xuất báo cáo

#### 4.5.3.6 Xem dashboard tình hình hoạt động của bot

Ngoài xuất báo cáo để đánh giá hoạt động của bot, quản trị viên có thể nhanh chóng xem qua dashboard. Trên dashboard có các loại thống kê như sau: thống kê yêu cầu người dùng theo thời gian, thống kê ý định được gọi nhiều nhất, thống kê tính hiệu quả của bot qua dữ

liệu huấn luyện từ hội thoại. Trong đó, chức năng thống kê tính hiệu quả của bot là chức năng đặc biệt của hệ thống sẽ được mô tả ở chương 5.

**Hình 30** mô tả một phần giao diện của dashboard này. Ta có thể hiện tại bot thống kê theo tuần và có 55 yêu cầu người dùng trong đó có 50 yêu cầu trả lời được, 3 yêu cầu hỏi lại và 2 yêu cầu không trả lời được. Các con số đó được hiển thị qua hai dạng biểu đồ là biểu đồ miền và biểu đồ tròn. Ngoài giao diện xuất hiện trên ảnh ta còn có các biểu đồ ở bên dưới nữa. Nhờ vậy quản trị viên có thể đánh giá bot bằng nhiều thông số đảm bảo độ chính xác cao.

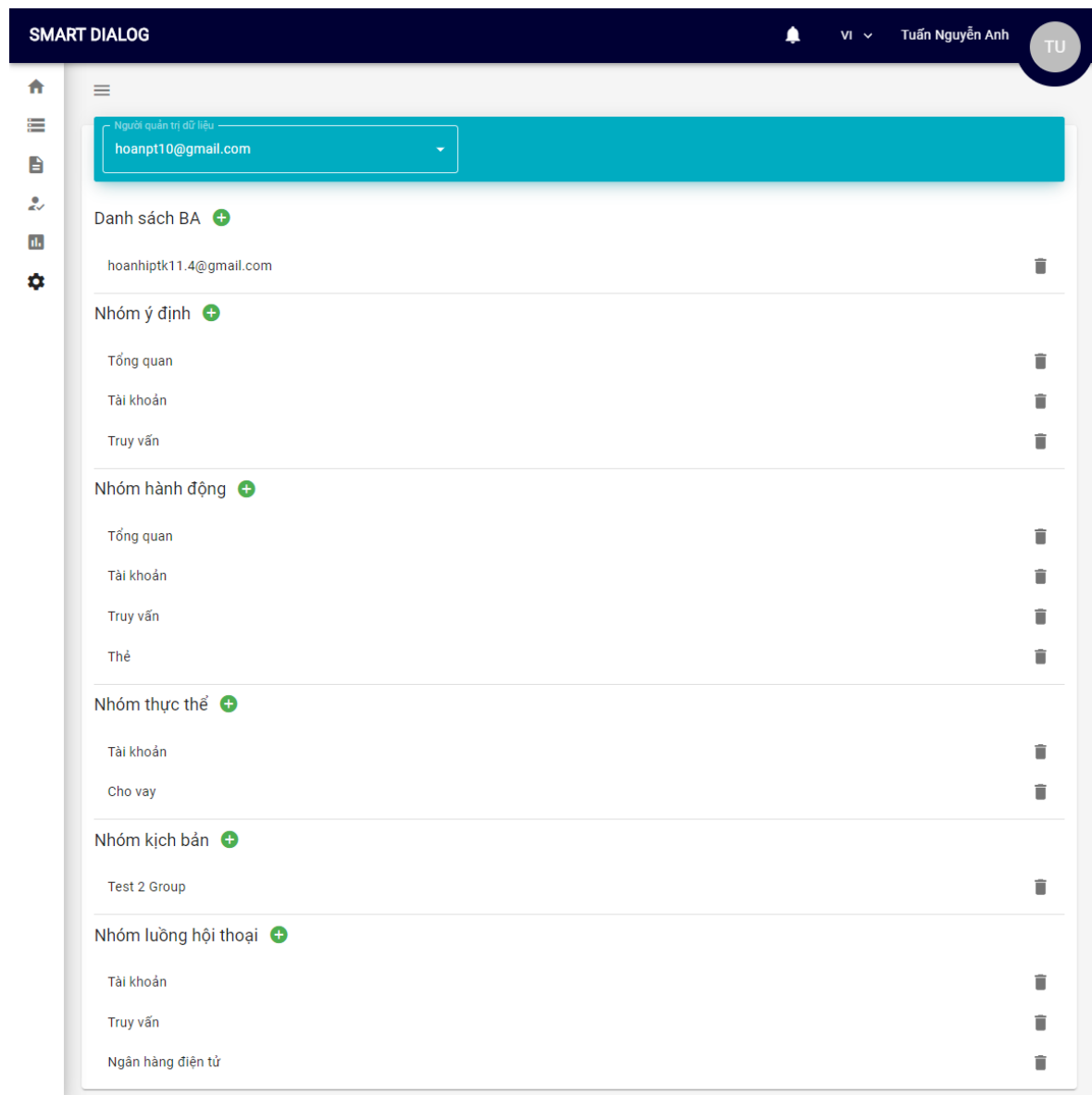


**Hình 30** Chức năng xem thông số hoạt động của bot

#### 4.5.3.7 Chức năng phân quyền cho quản trị dữ liệu

Quản trị viên sẽ có chức năng phân quyền cho quản trị dữ liệu. Khi vào màn hình này, quản trị viên sẽ chọn email quản trị dữ liệu muốn phân quyền. Khi đó màn hình sẽ hiện thi các biên tập viên hoặc các nhóm dữ liệu đã phân cho người này. Từ đó, quản trị viên có thể thêm bớt chỉnh sửa các mục dữ liệu đó.

Muốn chỉnh sửa, quản trị viên ấn vào nút dấu cộng ở mỗi mục dữ liệu để hiển thị popup. Sau đó, quản trị viên chọn mục dữ liệu cần thêm và ấn nút lưu. Để xóa dữ liệu thì người dùng ấn icon delete ở phía bên phải màn hình. Sau khi phân quyền thành công thì quản trị dữ liệu sẽ được quản lý các nhóm dữ liệu hoặc các biên tập viên tương ứng.



**Hình 31** Chức năng phân quyền cho quản trị dữ liệu

## 4.6 Kiểm thử và triển khai

Trong quá trình phát triển ĐATN, em đã xây dựng 10 test case cho quy trình chính của hệ thống là xây dựng và phê duyệt hội thoại với tỉ lệ đạt là 100%. Do giới hạn của ĐATN, em sẽ trình bày chi tiết các test case ở Phụ lục.

Hệ thống đã được tích hợp vào nền tảng Smartdialog và triển khai thử nghiệm với đối tác VietcomBank trong lĩnh vực ngân hàng và đối tác FE Credit trong lĩnh vực tài chính. Hệ thống đang triển khai trên tên miền <https://smartdialog.iristech.club> và <https://dev-smartdialog.iristech.club>.

## 4.7 Kết chương

Trong chương này, em đã trình bày cách áp dụng công nghệ được mô tả ở chương 3 để thiết kế và triển khai hệ thống. Qua đó, người đọc hình dung rõ hơn hệ thống được phát triển và vận hành như thế nào. Tiếp theo, chương 5 em sẽ mô tả các đóng góp, giải pháp nổi bật của mình trong hệ thống này.

## Chương 5 Các giải pháp và đóng góp nổi bật

Chương 5 sẽ trình bày các giải pháp và đóng góp nổi bật của em trong suốt quá trình hoàn thành đề án này. Đó là những giải pháp, cơ chế giúp xây dựng, đảm bảo chất lượng dữ liệu huấn luyện từ hội thoại và trực quan hóa tình hình hoạt động của chatbot trong thực tế.

### 5.1 Thiết kế quy trình đảm bảo chất lượng dữ liệu huấn luyện

#### 5.1.1 Đặt vấn đề

Trong quá trình triển khai, dữ liệu huấn luyện được xây dựng ngày càng nhiều với nhiều người làm dữ liệu khác nhau. Điều đó phát sinh ra trường hợp người làm tốt, người làm chưa tốt. Nếu tất cả dữ liệu đó đều được đưa vào huấn luyện thì dẫn tới khả năng bot hoạt động không hiệu quả, phản hồi sai nghiệp vụ gây phản cảm cho người dùng và gây thiệt hại cho công ty. Một vấn đề khác khi có nhiều người làm dữ liệu trên hệ thống, đặc biệt trong các công ty lớn nhiều phòng ban là sự xung đột và dư thừa dữ liệu. Ví dụ như biên tập viên của phòng ban A có thể làm một mục dữ liệu đã có trong phòng ban B và ngược lại. Ngoài ra, việc không nằm trong phòng ban B, không nắm rõ hoàn toàn nghiệp vụ của bên đó thì dữ liệu xây dựng có thể không tốt. Do vậy, ta cần phải có thêm cơ chế để phân nhóm người làm dữ liệu để khắc phục vấn đề trên.

#### 5.1.2 Quy trình xây dựng và phê duyệt dữ liệu huấn luyện

Giải pháp em đưa ra là đề xuất quy trình xây dựng và phê duyệt dữ liệu huấn luyện được mô tả chi tiết ở **Hình 32**. Dữ liệu huấn luyện trải qua năm trạng thái với hai cấp bậc quản lý là quản trị dữ liệu và biên tập dữ liệu. Năm trạng thái đó là đang xây dựng, chờ phê duyệt, đang phê duyệt và trả lại. Chỉ đến khi nào ở trạng thái chấp nhận, hội thoại đó mới được đưa vào huấn luyện. Quản trị dữ liệu chịu trách nhiệm duyệt các hội thoại do biên tập dữ liệu tạo ra. Nhờ vậy, dữ liệu sẽ đạt chất lượng cao và ổn định do có nhiều lượt xem xét và chỉnh sửa. Để giải quyết bài toán phân nhóm người làm dữ liệu, người quản trị viên của hệ thống sẽ phân quyền cho quản trị dữ liệu. Quản trị dữ liệu sẽ được quản lý một số nhóm dữ liệu và một nhóm các biên tập viên. Nhóm dữ liệu và biên tập viên này tương đương với trách nhiệm của một phòng ban trong công ty lớn. Như vậy cả hai vấn đề đã nêu trên đều được giải quyết.





Trạng thái bắt đầu là trạng thái đang xây dựng. Lúc này, biên tập viên và quản trị dữ liệu có thể chỉnh sửa và lưu lại. Khi hoàn thiện công việc chỉnh sửa, biên tập viên và quản trị dữ liệu chọn chức năng hoàn thành. Khi đó, hội thoại sẽ chuyển trạng thái ứng với mỗi vai trò người làm dữ liệu. Với biên tập viên, hội thoại chuyển trạng thái thành chờ phê duyệt. Với quản trị dữ liệu, trạng thái lúc đó là chấp nhận. Với trạng thái chấp nhận, hội thoại sẽ được đưa vào huấn luyện và kết thúc quy trình.

Khi hội thoại đang ở trạng thái chờ phê duyệt, biên tập viên nếu có nhu cầu vẫn có thể sửa được và sẽ lưu lại theo hai cách: một là lưu nháp để về trạng thái đang xây dựng, hai là hoàn thành để giữ nguyên trạng thái chờ phê duyệt.

Quản trị dữ liệu sẽ được duyệt một hội thoại khi hội thoại đó đang ở trạng thái chờ phê duyệt. Khi này quản trị dữ liệu có ba sự lựa chọn để thao tác. Lựa chọn thứ nhất, khi duyệt chưa xong, muốn đánh dấu cho biên tập viên không thể vào sửa được nữa thì quản trị dữ liệu chọn chức năng lưu để đưa hội thoại về trạng thái đang phê duyệt. Lựa chọn thứ hai, khi thấy hội thoại này có nhiều vấn đề mà bản thân không sửa hết được, quản trị dữ liệu chọn trả lại cho biên tập viên. Với hành động này, quản trị dữ liệu phải có kèm theo lý do trả lại để biên tập viên biết được vấn đề dữ liệu mình xây dựng là gì để tìm cách sửa thỏa mãn. Khi ở trạng thái trả lại, biên tập viên có thể sửa và hoàn thiện, đưa dữ liệu về trạng thái chờ phê duyệt. Lựa chọn cuối cùng, nếu hội thoại huấn luyện đã được xây dựng ổn, quản trị dữ liệu sẽ chọn chức năng chấp nhận.

Tuy nhiên, khi quản trị dữ liệu chấp nhận hội thoại thì hội thoại sẽ chưa chuyển sang trạng thái chấp nhận ngay mà lúc này sẽ nảy sinh ra một vấn đề như sau. Đó là hội thoại là dạng dữ liệu phụ thuộc vào nhiều dạng dữ liệu khác như ý định, hành động, thực thể. Các dạng dữ liệu này cũng có các trạng thái và quy trình xây dựng phê duyệt như trên. Việc chấp nhận hội thoại khi chưa chấp nhận các dạng dữ liệu kia là điều không hợp lý. Do vậy, trước khi hội thoại chuyển sang trạng thái chấp nhận, hệ thống sẽ kiểm tra tất cả các dữ liệu phụ thuộc xem đã chấp nhận hết chưa rồi mới cho chấp nhận hội thoại. Qua rất nhiều bước như vậy, hệ thống đảm bảo rằng dữ liệu đưa vào huấn luyện là những dữ liệu chất lượng nhất, phù hợp nhất. Điều đó giúp bot trở nên “thông minh” hơn, hoạt động tốt hơn.

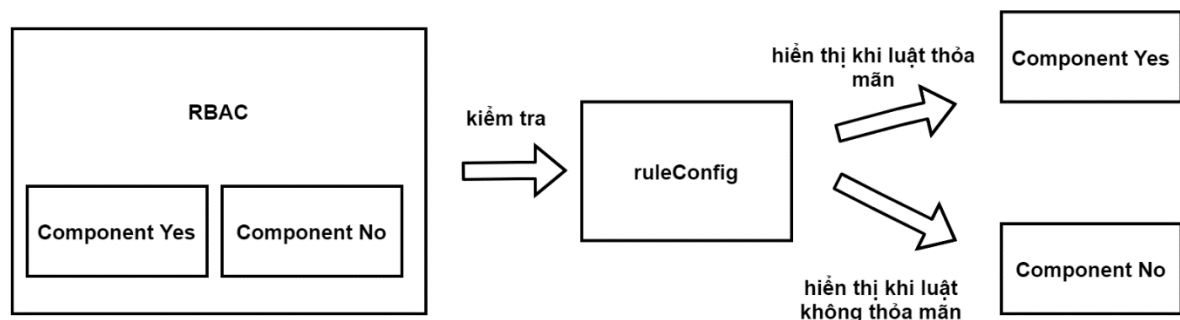
Ngoài ra, quy trình có một số điểm lưu ý. Đó là khi xây dựng hội thoại chỉ được dùng các ý định, hành động, thực thể không ở trạng thái đang xây dựng hoặc trả lại. Việc hạn chế hai trạng thái trên có những ưu điểm như sau. Đó là hạn chế các dữ liệu phụ thuộc kém chất lượng. Khi đó, hội thoại tạo ra sẽ có khả năng được chấp nhận cao hơn do các dữ liệu phụ thuộc kia cũng có khả năng được chấp nhận cao. Thứ hai do không hạn chế hết các trạng thái giúp các biên tập viên không phải chờ đợi lẫn nhau để xây dựng dữ liệu. Để tạo hội thoại, họ không cần phải đợi các ý định, hành động hay thực thể được chấp nhận hết. Chính vì vậy, số lượng hội thoại huấn luyện sẽ được tăng về mặt số lượng. Lưu ý thứ hai là các

biên tập viên có thể xem dữ liệu lẫn nhau nhưng không được sửa của nhau. Điều này giúp mọi người không bị trùng lặp và không xảy ra tranh chấp khi xây dựng dữ liệu. Điều này nảy sinh ra một vấn đề. Đó là một biên tập viên đang làm dở một mục dữ liệu ở trạng thái đang xây dựng mà nghỉ việc thì mục dữ liệu đó sẽ không ai có quyền được sửa. Do vậy em phát triển thêm chức năng giúp quản trị dữ liệu có thể gán dữ liệu của biên tập viên đó cho người khác, khắc phục tình trạng nêu trên. Lưu ý cuối cùng, do có nhiều người cùng thao tác lên một đoạn hội thoại nên khi hội thoại chuyển trạng thái sang đang phê duyệt, chấp nhận hay trả lại em đều lưu lại một phiên bản của hội thoại đó. Sau này, khi truy vết, mọi người đều có thể xem được ai đã sửa và sửa như thế nào. Tất cả các chức năng thêm vào đó giúp quy trình xây dựng và phê duyệt dữ liệu hoàn thiện hơn, đạt hiệu quả tốt hơn.

### 5.1.3 Xây dựng thành phần quản lý quyền linh hoạt

Để có quy trình như trên thì giao diện cần thể hiện chức năng phù hợp với mỗi vai trò. Ví dụ với vai trò biên tập viên, chức năng chấp nhận không được hiển thị. Nếu theo phương án code thông thường, với mỗi chức năng như vậy, ta dùng câu lệnh rẽ nhánh để hiển thị thì mã nguồn sẽ trở nên phức tạp, khó mở rộng. Sau này khi yêu cầu thay đổi, ta phải sửa rất nhiều chỗ khác trong mã nguồn.

Tận dụng tính phân chia giao diện thành các thành phần (component) của React, giải pháp em đưa ra là thiết kế một component có tên là RBAC (Rule based access control). Tất cả các component cần được hiển thị theo dữ liệu phân quyền đều được bọc component RBAC này. Khi đó, chúng sẽ được kiểm tra qua một hàm để xem xét có được hiển thị hay không. Hàm này có đầu vào là luật. Đầu ra là biến trạng thái đúng hoặc sai. Danh sách các luật được cấu hình tại một chỗ duy nhất. Nhờ vậy khi yêu cầu thay đổi, ta chỉ cần thay đổi logic của luật tại một chỗ, toàn bộ giao diện sẽ thay đổi theo mà không cần thay đổi quá nhiều mã nguồn. Mã nguồn trở nên linh hoạt hơn rất nhiều.



**Hình 33** Minh họa thành phần quản lý phân quyền linh hoạt

## 5.2 Giải pháp bóc tách và tạo dữ liệu huấn luyện từ lịch sử chat

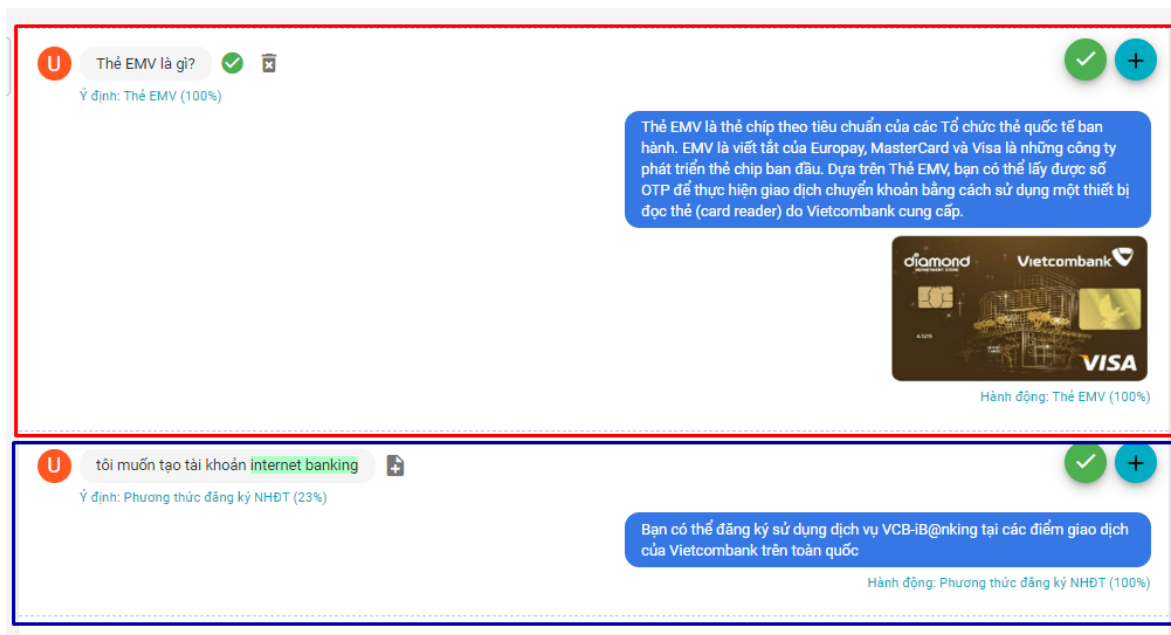
### 5.2.1 Đặt vấn đề

Dữ liệu lịch sử chat của người dùng là nguồn dữ liệu dồi dào và mang tính thực tế cao. Điều đó đặt ra nhu cầu chuyển đổi nguồn dữ liệu này thành dữ liệu huấn luyện. Trên thực tế khảo sát ở chương 2, các nền tảng Dialogflow và FPT.AI Conversation đã có cơ chế này. Tuy vậy các nền tảng đó còn một số điểm hạn chế. Hạn chế ở đây là các nền tảng chỉ hiển thị lịch sử trò chuyện thành danh sách các yêu cầu đầu vào của người dùng. Yêu cầu này chỉ có nội dung người dùng gửi lên và ý định, thực thể mà bot trích xuất ra. Điều này dẫn tới bất tiện cho người làm dữ liệu là không thể bao quát được ngữ cảnh của toàn bộ cuộc trò chuyện vì không rõ các hành động bot trả lời trong cả đoạn hội thoại là gì và cũng không chỉnh sửa được các hành động đó. Từ đó đặt ra yêu cầu hệ thống phải cung cấp dữ liệu lịch sử chat một cách trực quan, cơ chế chỉnh sửa rõ ràng để người làm dữ liệu có thể đánh giá đoạn hội thoại lịch sử đó đúng chưa, đúng ở đâu, sai chỗ nào để chuyển đổi thành dữ liệu huấn luyện có ích.

### 5.2.2 Đề xuất hiển thị lịch sử chat theo đơn vị hội thoại

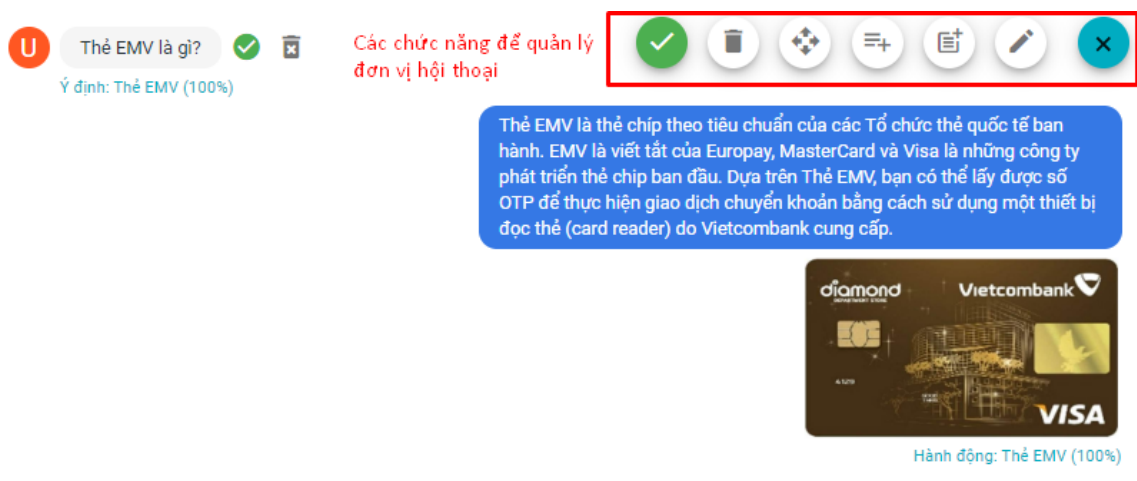
Để thuận tiện, trực quan hóa cho người làm dữ liệu huấn luyện từ lịch sử chat, em hiển thị một đoạn hội thoại theo các đơn vị hội thoại thay vì chỉ hiển thị theo các tin nhắn thông thường. Đơn vị hội thoại ở đây được định nghĩa là một danh sách bao gồm một yêu cầu đầu vào người dùng và các hành động phản hồi của bot tương ứng với yêu cầu đó.

**Hình 34** mô tả chi tiết về đơn vị hội thoại. Đoạn hội thoại trong hình được chia làm hai đơn vị hội thoại. Ví dụ với đơn vị đầu tiên được tô viền màu đỏ, thành phần thứ nhất là nội dung yêu cầu người dùng “thẻ emv là gì”. Ý định dự đoán là “Thẻ EMV” với độ tự tin 100%. Hành động trả về là “Thẻ EMV” với độ tự tin cũng là 100%. Hành động này gồm hai nội dung phản hồi là chữ và hình ảnh. Với đơn vị thứ hai được tô màu xanh nước biển cũng tương tự như vậy, nội dung yêu cầu là “tôi muốn tạo tài khoản internet banking” với ý định đoán ra là “Phương thức đăng ký NHĐT”, độ tự tin là 23%. Ở yêu cầu này, có một thực thể được trích xuất là “Ngân hàng điện tử” hiển thị màu xanh lá cây. Hành động trả về là “Phương thức đăng ký NHĐT” với một nội dung phản hồi là chữ, độ tự tin 100%. Tương tự với các đoạn hội thoại khác cũng được chia thành một, hai hay nhiều đơn vị hội thoại như vậy. Với cách phân chia đó, rõ ràng người làm dữ liệu sẽ dễ dàng đánh giá với một yêu cầu của người dùng, ý định bot đoán ra có đúng không, hành động phản hồi có chính xác không. Nhờ vậy, việc tạo dữ liệu huấn luyện từ lịch sử chat trở nên dễ dàng, trực quan hơn khi ta sẽ tập trung vào chỉnh sửa từng đơn vị hội thoại một.



**Hình 34** Mô tả khái niệm đơn vị hội thoại

### 5.2.3 Quy trình xây dựng dữ liệu huấn luyện từ lịch sử chat thông qua các đơn vị hội thoại

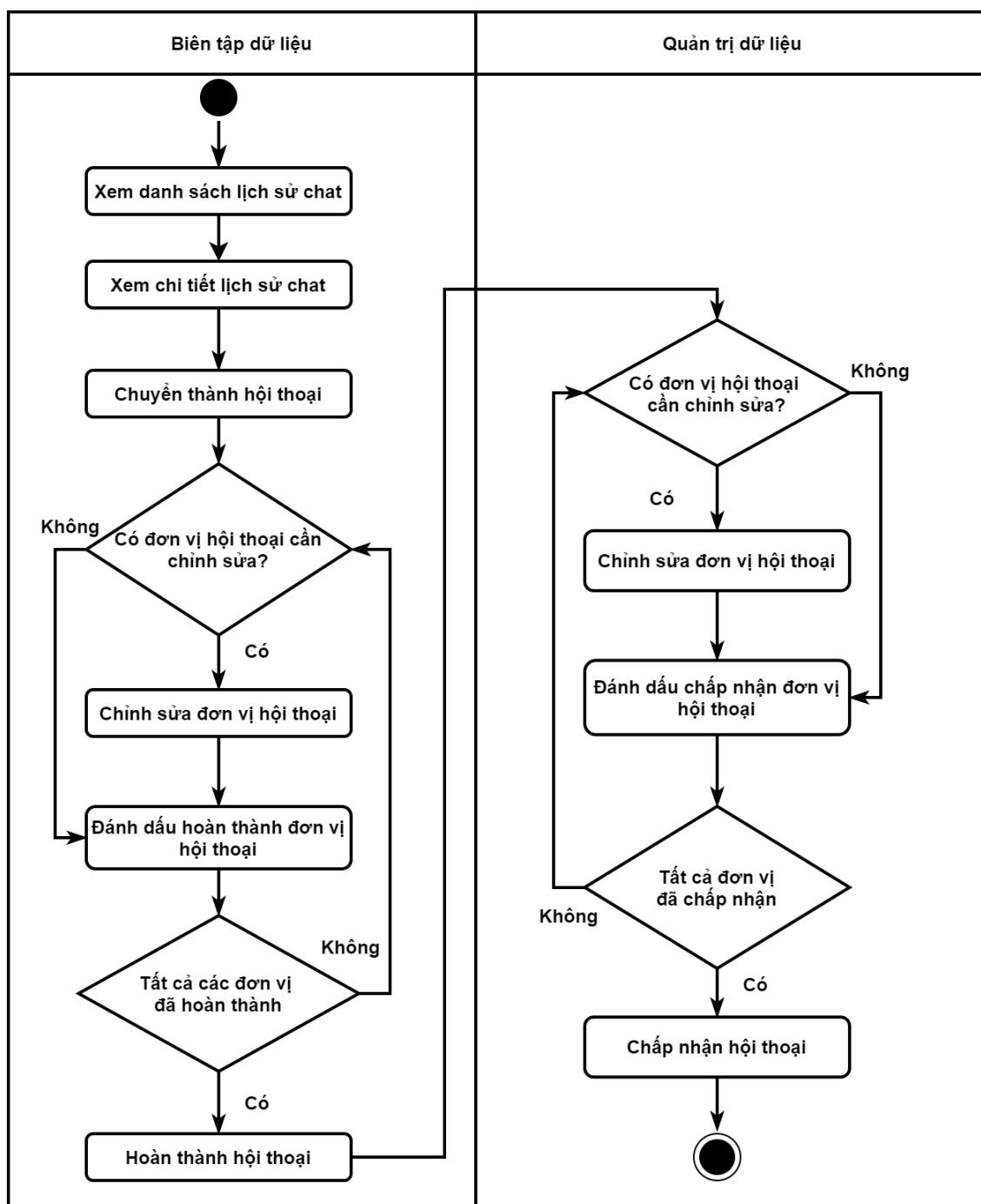


**Hình 35** Các chức năng quản lý đơn vị hội thoại

Sau khi bóc tách lịch sử chat và hiển thị thành các đơn vị hội thoại, việc quản lý hội thoại sẽ xoay quanh việc quản lý các đơn vị hội thoại. **Hình 35** mô tả các chức năng trong quản lý đơn vị hội thoại. Các chức năng đó bao gồm như xóa đơn vị hội thoại, di chuyển đơn vị hội thoại, thêm đơn vị bên trên, bên dưới và chỉnh sửa đơn vị hội thoại. Việc cung cấp nhiều chức năng như vậy giúp việc chỉnh sửa hội thoại trở nên dễ dàng và hiệu quả hơn.

Do việc quản lý hội thoại xoay quanh việc quản lý các đơn vị hội thoại nên dù đã có quy trình xây dựng và phê duyệt dữ liệu trình bày ở 5.1, em vẫn đề xuất thêm một quy trình nữa

để đảm bảo chất lượng dữ liệu. Đó là xây dựng dữ liệu huấn luyện từ lịch sử chat thông qua các đơn vị hội thoại.



**Hình 36** Quy trình tạo và duyệt hội thoại từ lịch sử chat

**Hình 36** mô tả quy trình tạo và duyệt hội thoại từ lịch sử chat dựa vào các đơn vị hội thoại. Do lịch sử chat có thể bao gồm nhiều đơn vị hội thoại, người làm dữ liệu có thể bỏ qua và không xem xét một số đơn vị nhất định. Do vậy, để đảm bảo chất lượng dữ liệu, em thiết kế quy trình đảm bảo người làm dữ liệu cần đánh dấu từng đơn vị hội thoại một để hoàn thành

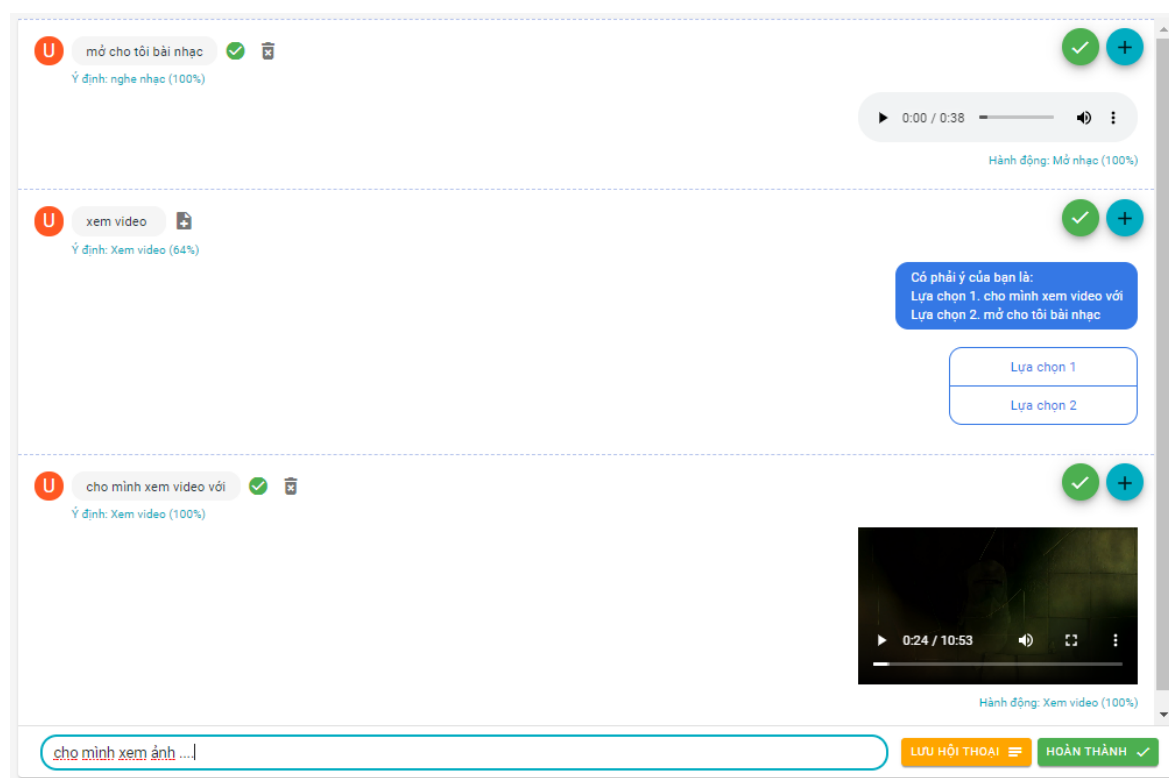
hoặc chấp nhận dữ liệu. Từ đó, biên tập viên hoặc quản trị dữ liệu sẽ tập trung kiểm tra và không bỏ sót đơn vị hội thoại nào.

## 5.3 Giải pháp tạo dữ liệu huấn luyện qua chat tương tác

### 5.3.1 Đặt vấn đề

Nguồn hội thoại từ lịch sử chat là nguồn hội thoại thụ động. Người làm dữ liệu đôi khi không thể tìm được các hội thoại phù hợp với mục đích của mình để xây dựng dữ liệu. Vấn đề này có thể tạm thời giải quyết bằng cách sử dụng các nền tảng nhắn tin để chat và sau đó tạo dữ liệu huấn luyện từ lịch sử. Cách làm đó có nhiều điểm bất tiện là phải thực hiện thêm bước tạo hội thoại, tạo xong mới chỉnh sửa được. Ngoài ra khi làm như vậy, hệ thống sẽ tăng lượng hội thoại ảo cho bot dẫn tới chỉ số thống kê khác sai lệch. Điều này dẫn tới cần phải có giải pháp cho người làm dữ liệu vừa tạo hội thoại vừa chỉnh sửa hành vi của bot.

### 5.3.2 Giải pháp

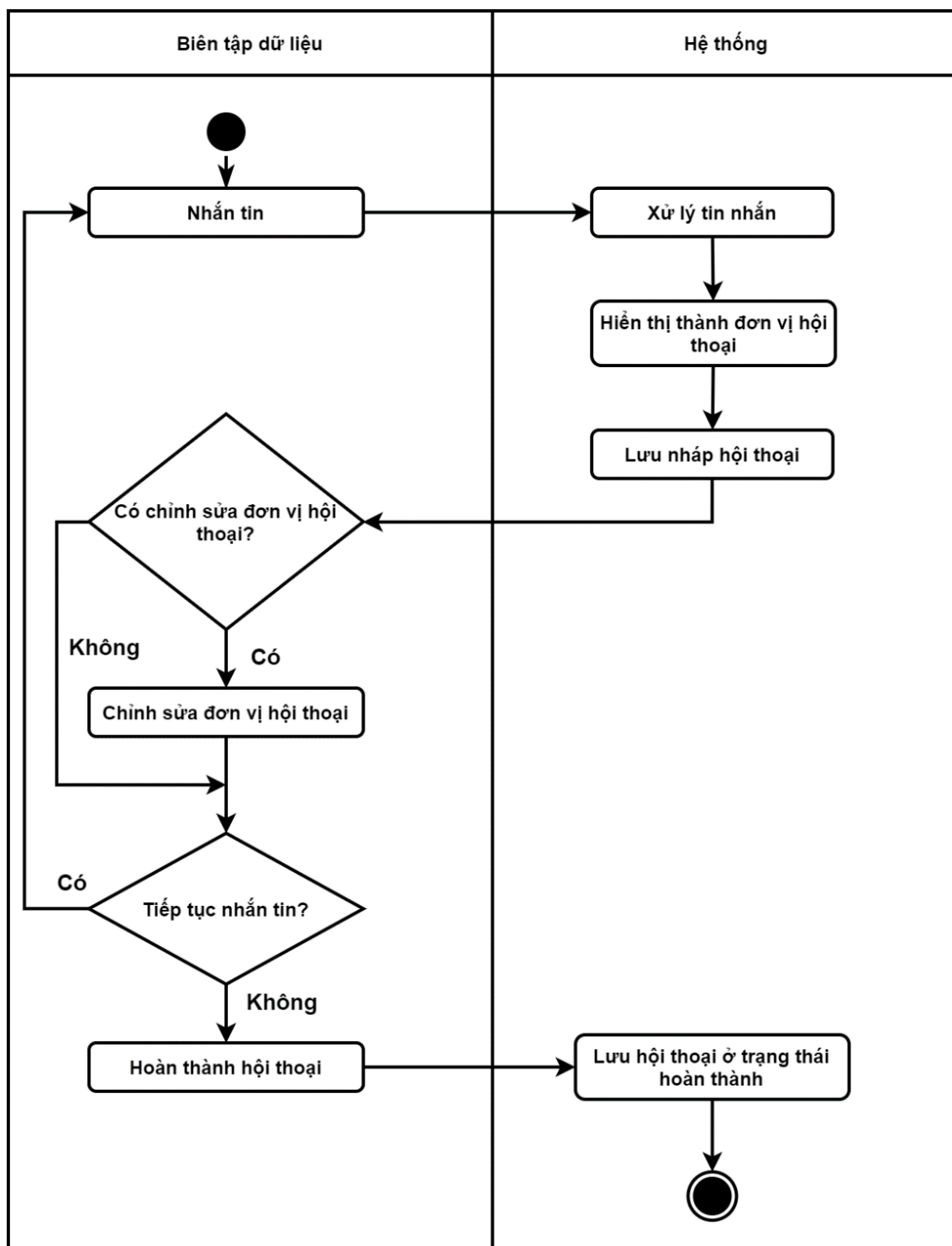


**Hình 37** Chức năng chat tương tác với bot

Giải pháp em đưa ra là phát triển một công cụ chat trực tiếp cho người làm dữ liệu vừa nhắn tin tương tác với bot vừa xây dựng dữ liệu. Giao diện chức năng mô tả ở **Hình 37**.

Tính năng chat này nằm trực tiếp trong hệ thống và hiển thị dữ liệu chat thành các đơn vị hội thoại. Do vậy, người làm dữ liệu có thể sử dụng đầy đủ các tính năng của quản lý đơn vị

hội thoại. Các dạng phản hồi của bot như chữ, ảnh, audio, video đều được thể hiện một cách đầy đủ, giúp người dùng đạt trải nghiệm như các nền tảng nhắn tin thông thường khác. Quá trình tạo dữ liệu từ chat tương tác sẽ được mô tả ở **Hình 38**.



**Hình 38** Quy trình tạo dữ liệu huấn luyện qua chat tương tác



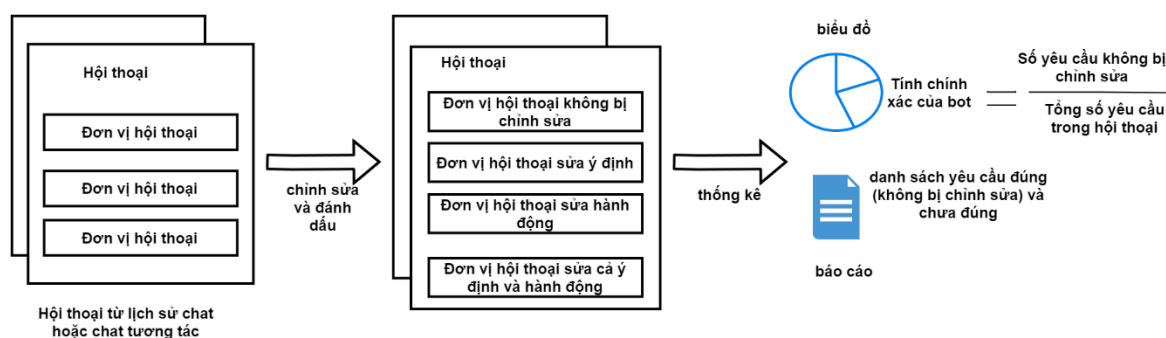
## 5.4 Đánh giá tính hiệu quả của bot bằng dữ liệu huấn luyện từ hội thoại

### 5.4.1 Đặt vấn đề

Để đánh giá hoạt động triển khai bot trong thực tế, nhiều hệ thống khác thường đưa ra thông số dựa chủ yếu vào dữ liệu lịch sử chat. Thông số đó thường là tỉ lệ giữa số yêu cầu bot trả về hành động với tổng số yêu cầu người dùng. Điều đó phần nào đã đánh giá được khả năng đáp ứng nhu cầu của chatbot đó. Tuy nhiên, việc bot phản hồi được yêu cầu người dùng không tương đương với bot trả lời đúng hay không. Ta cần thêm chỉ số khác để đánh giá được tính chính xác, độ hiệu quả trả lời câu hỏi của bot trong thực tế.

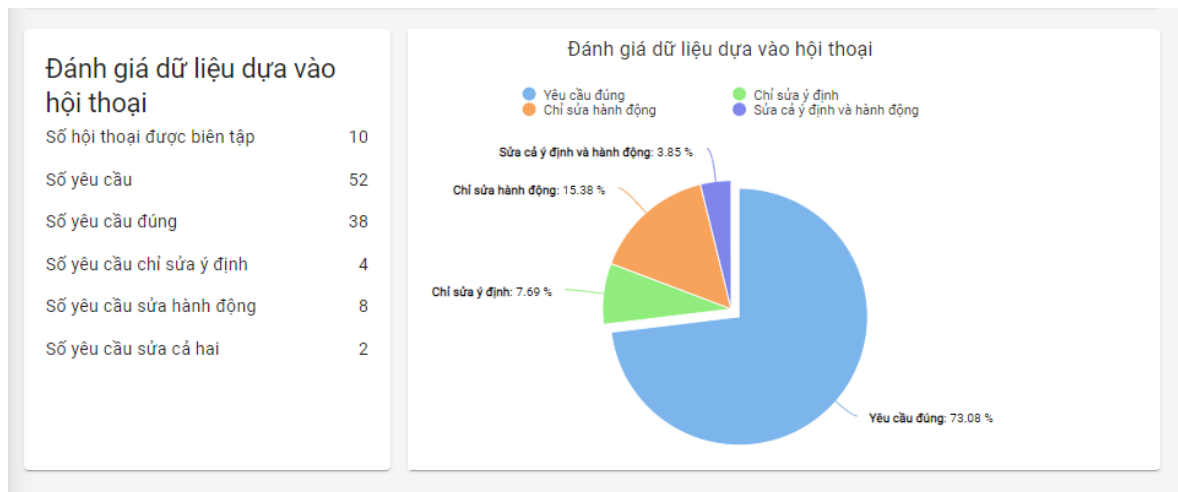
### 5.4.2 Giải pháp

Giải pháp em đưa ra ở đây là dựa vào sự chỉnh sửa các đơn vị hội thoại sau khi xây dựng hội thoại huấn luyện để đánh giá tính chính xác của bot. **Hình 39** mô tả quá trình đánh giá đó. Ban đầu, ta có các hội thoại từ hai nguồn là lịch sử chat và chat tương tác. Hội thoại này được tách thành các đơn vị hội thoại. Sau đó, người làm dữ liệu sẽ xem xét từng đơn vị hội thoại trong một hội thoại đó. Nếu yêu cầu đầu vào, ý định đoán ra và bot phản hồi đã khớp với nhau thì không cần chỉnh sửa, nếu không thì người làm dữ liệu sẽ chỉnh sửa lại cho đúng. Quá trình chỉnh sửa này sẽ được hệ thống đánh dấu và ghi lại. Do vậy, hệ thống sẽ thống kê được đơn vị hội thoại nào không bị chỉnh sửa, đơn vị nào sửa ý định, sửa hành động hoặc sửa cả ý định lẫn hành động. Chính các đơn vị không bị chỉnh sửa mới tương ứng với yêu cầu mà bot trả lời đúng, khác với yêu cầu bot trả lời được như dữ liệu từ lịch sử. Do vậy, từ các số liệu thống kê trên, hệ thống sẽ xuất ra các biểu đồ và báo cáo phù hợp.



**Hình 39** Đánh giá độ hiệu quả của bot qua dữ liệu huấn luyện từ hội thoại

**Hình 40** mô tả cách đánh giá này áp dụng trong hệ thống của em. Với 10 hội thoại được biên tập và 52 yêu cầu người dùng, ta thấy được số yêu cầu đúng (các yêu cầu không bị chỉnh sửa) là 38. Do vậy tỉ lệ chính xác của chatbot đạt được là 73.08%.



**Hình 40** Chức năng thống kê tính hiệu quả của bot dựa vào hội thoại

## 5.5 Kết chương

Chương 5 em đã trình bày những giải pháp và những nội dung đóng góp mà em thấy tâm đắc nhất trong quá trình hoàn thành ĐATN. Các giải pháp bao gồm bóc tách dữ liệu từ lịch sử và dữ liệu huấn luyện từ lịch sử, phát triển giải pháp tạo dữ liệu huấn luyện qua chat tương tác, thiết kế quy trình đảm bảo chất lượng dữ liệu và đánh giá hiệu quả của bot bằng dữ liệu huấn luyện. Các giải pháp trên đã giúp hệ thống có thể xây dựng, đảm bảo chất lượng dữ liệu huấn luyện từ hội thoại và trực quan hóa tình hình hoạt động của chatbot trong thực tế. Sau đây, chương 6 sẽ tổng kết nội dung đồ án và đưa ra hướng phát triển cho tương lai.

# Chương 6 Kết luận và hướng phát triển

## 6.1 Kết luận

Như vậy, em đã trình bày xong về hệ thống tạo dữ liệu huấn luyện từ hội thoại. So với các nền tảng khác trên thị trường, hệ thống đã đạt được các cải tiến như sau: i) tận dụng triệt để dữ liệu lịch sử chat để làm dữ liệu huấn luyện, ii) phát triển công cụ để người làm dữ liệu vừa chat trực tiếp vừa chỉnh sửa hành vi của bot, iii) thiết kế quy trình xây dựng và phê duyệt dữ liệu phù hợp với các công ty lớn nhiều phòng ban, iv) đánh giá tình hình hoạt động của bot trong thực tế qua dữ liệu lịch sử và dữ liệu huấn luyện. Hiện tại, hệ thống đã được tích hợp vào nền tảng Smartdialog triển khai thử nghiệm với đối tác Vietcombank, FECredit.

Tuy vậy, hệ thống vẫn còn một số điểm cần hoàn thiện để quá trình làm dữ liệu đạt hiệu quả cao hơn. Đó là chức năng chat tương tác còn chưa cho người dùng gửi file, hệ thống chưa có chức năng undo, redo khi xây dựng hội thoại, chưa có nhiều dạng báo cáo cũng như khả năng tái sử dụng các mẫu báo cáo sẵn có.

Trong quá trình phát triển hệ thống, em đã gặp một số vấn đề và có được các bài học kinh nghiệm sâu sắc sau khi giải quyết. Đó là do đây là hệ thống có các tính năng, khái niệm mới nên cần phát triển các chức năng cần trực quan, thân thiện với người sử dụng. Tính UI/UX của hệ thống cần được đặc biệt lưu ý. Đây là một lĩnh vực mà rất dễ bỏ qua khi làm phần mềm. Thứ hai, do quy trình tạo và phê duyệt dữ liệu không phải được quy định ngay từ đầu mà được thay đổi qua nhiều quá trình họp và xem xét, em cần thiết kế mã nguồn có tính mở rộng cao để đáp ứng được sự thay đổi liên tục đó. Cuối cùng, do hệ thống nằm trong nền tảng Smartdialog nên em được làm việc cùng nhiều anh, chị khác. Kỹ năng làm việc nhóm và trình bày ý tưởng cũng cần phải cải thiện. Đây là những kinh nghiệm quý giá đối với sinh viên để bước chân vào con đường sự nghiệp sau này.

## 6.2 Hướng phát triển

Hệ thống xây dựng dữ liệu huấn luyện hội thoại vẫn đang cải tiến và phát triển trong tương lai. Em sẽ tiếp tục lắng nghe các ý kiến của thầy cô, người sử dụng để hoàn thiện sản phẩm.

Đầu tiên, em sẽ cải tiến các chức năng chat tương tác để người dùng có thể gửi file khi chat, đồng thời cũng để xây dựng hội thoại huấn luyện có yêu cầu đầu vào là audio, ảnh, video. Thứ hai là phát triển tính năng undo, redo khi xây dựng dữ liệu hội thoại. Tiếp theo, em sẽ

phát triển các dạng báo cáo cho quản trị dữ liệu và biên tập viên để có thể dễ dàng quản lý và báo cáo dữ liệu mình đang xây dựng. Ngoài ra, em cần cải thiện một số phần về giao diện giúp hệ thống thân thiện với người sử dụng hơn.

Trên đây là những kết luận và hướng phát triển của dự án. Em rất mong có những góp ý, nhận xét của mọi người để hệ thống ngày càng phát triển hơn.

## Tài liệu tham khảo

- [1] “Chatfuel.” [Online]. Available: <https://chatfuel.com>. [Accessed: 23-June-2020].
- [2] “Dialogflow.” [Online]. Available: <https://dialogflow.com>. [Accessed: 23-June-2020].
- [3] “FPT.AI Conversation.” [Online]. Available <https://fpt.ai>. [Accessed: 23-June-2020].
- [4] S. Arora, K. Batra, and S. Singh, “Dialogue System: A Brief Review,”  
ArXiv13064134 Cs, Jun. 2013.
- [5] “ReactJS.” [Online]. Available: <https://reactjs.org>. [Accessed: 23-June-2020].
- [6] “NextJS.” [Online]. Available: <https://nextjs.org>. [Accessed: 23-June-2020].
- [7] “MaterialUI.” [Online]. Available: <https://material-ui.com>. [Accessed: 23-June-2020].
- [8] “Redux.” [Online]. Available: <https://redux.js.org>. [Accessed: 23-June-2020].
- [9] “NodeJS.” [Online]. Available: <https://nodejs.org>. [Accessed: 23-June-2020]
- [10] “ExpressJS”. [Online]. Available: <https://expressjs.com>. [Accessed: 23-June-2020]
- [11] Aaron Kili, “18 Best NodeJS Frameworks for Developers in 2020”. [Online].  
Available: <https://www.tecmint.com/best-nodejs-frameworks-for-developers/>.  
[Accessed: 23-June-2020]
- [12] “What is NoSQL?”. [Online]. Available: <https://www.mongodb.com/nosql-explained>. [Accessed: 23-June-2020].
- [13] “MongoDB”. [Online]. Available: <https://www.mongodb.com>. [Accessed: 23-June-2020].
- [14] Mark Massé, REST API Design Rulebook, 2012

# Phụ lục

## A Đặc tả use case

### A.1 Đặc tả use case Xem danh sách lịch sử chat

<b>Mã Use case</b>	UC005	<b>Tên Use case</b>	Xem danh sách lịch sử chat
<b>Tác nhân</b>	Biên tập dữ liệu, quản trị dữ liệu, quản trị viên		
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập		
<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Tác nhân	Chọn trang lịch sử
	2.	Hệ thống	Lấy danh sách hội thoại từ lịch sử
	3.	Hệ thống	Hiển thị danh sách hội thoại (mô tả phía dưới *)
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	4a1.	Tác nhân	Lựa chọn các thuộc tính để lọc (mô tả phía dưới **)
	4a2.	Hệ thống	Lấy danh sách hội thoại đã lọc
	4a3.	Hệ thống	Hiển thị danh sách hội thoại
<b>Hậu điều kiện</b>	Không		

\* Dữ liệu đầu ra khi hiển thị danh sách hội thoại ở bảng dưới

**Bảng 16** Dữ liệu đầu ra khi hiển thị danh sách hội thoại

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
1.	Tên hội thoại	Là nội dung yêu	Chuỗi	Xin chào

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
		cầu đầu tiên		
2.	Số yêu cầu	Số yêu cầu trong hội thoại	Số	10
3.	Số yêu cầu không phù hợp	Các yêu cầu mà bot không trả lời được hoặc hỏi lại	Số	1
4.	Bắt đầu	Thời gian bắt đầu đoạn hội thoại	Chuỗi	06-05-2020 18:03
5.	Trạng thái	Các trạng thái gắn với dữ liệu huấn luyện	Chuỗi	Chưa xem

**\*\* Dữ liệu đầu vào khi chọn các thuộc tính lọc danh sách hội thoại**

**Bảng 17** Dữ liệu đầu vào khi lọc danh sách lịch sử chat

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Khoảng thời gian	Khoảng thời gian bắt đầu hội thoại	Có	Thời gian kết thúc lớn hơn thời gian bắt đầu	30/04/2020 00:00 – 07/05/2020 23:59
2.	Kênh	Các kênh tương tác	Không	Lựa chọn trong các kênh Smartdialog, Facebook, Zalo	Facebook
3.	Trạng thái	Trạng thái của hội thoại	Không	Lựa chọn trong trạng thái	Chưa xem
4.	Từ khóa	Từ nằm trong yêu cầu người dùng	Không	Không có	visa
5.	Ý định	Ý định xuất hiện trong hội	Không	Lựa chọn trong các	Mở thẻ visa

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
		thoại		ý định của hệ thống	
6.	Hành động	Hành động xuất hiện trong hội thoại	Không	Lựa chọn trong các hành động của hệ thống	Hướng dẫn mở thẻ visa
7.	Bot không hiểu	Hội thoại có ý định mặc định	Không	True hoặc False	True
8	Bot không hiểu hỏi lại	Hội thoại có trường hợp bot hỏi lại	Không	True hoặc False	True
9	Bot trả lời mặc định	Hội thoại có hành động mặc định	Không	True hoặc False	True
11	Số hội thoại mỗi trang	Số hội thoại hiển thị mỗi trang	Không	Lựa chọn hai giá trị 10, 25	10
12	Số trang	Chỉ số trang	Không	Số tự nhiên dương	1

## A.2 Đặc tả use case Sửa hành động trong đơn vị hội thoại

<b>Mã Use case</b>	UC006	<b>Tên Use case</b>	Sửa hành động trong đơn vị hội thoại
<b>Tác nhân</b>	Biên tập dữ liệu, quản trị dữ liệu, quản trị viên		
<b>Tiền điều kiện</b>	Người dùng đăng nhập với vai trò quản trị dữ liệu, quản trị viên đang trong chức năng xem hội thoại		
<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Tác nhân	Chọn chức năng sửa đơn vị hội thoại
	2.	Hệ thống	Hiển thị giao diện hiện chỉnh sửa đơn vị hội thoại



	3.	Tác nhân	Chọn chức năng sửa hành động
	4.	Hệ thống	Hiển thị danh sách hành động của hệ thống
	5.	Tác nhân	Chọn một hoặc nhiều hành động
	6.	Tác nhân	Chọn chức năng sửa
	7.	Hệ thống	Hiển thị danh sách hành động được sửa trong đơn vị hội thoại
	8.	Tác nhân	Chọn chức năng lưu đơn vị hội thoại
	9.	Hệ thống	Hiển thị các hành động mới của đơn vị hội thoại
Luồng sự kiện thay thế	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	8a1.	Tác nhân	Kéo thả các hành động vừa chọn
	8a2.	Hệ thống	Hiển thị các hành động theo danh sách đã sắp xếp
	8b1.	Tác nhân	Chọn chức năng hủy
	8b2.	Hệ thống	Hiển thị đơn vị hội thoại cũ
Hậu điều kiện	Không		

### A.3 Đặc tả use case Phân quyền cho quản trị dữ liệu

Mã Use case	UC007	Tên Use case	Phân quyền cho quản trị dữ liệu
Tác nhân	Quản trị viên		
Tiền điều kiện	Người dùng đăng nhập với vai trò quản trị viên		
Luồng sự kiện chính (Thành công)	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Tác nhân	Chọn trang phân quyền
	2.	Hệ thống	Hiển thị giao diện phân quyền

	3.	Tác nhân	Chọn email một quản trị dữ liệu bất kì
	4.	Hệ thống	Hiển thị danh sách biên tập dữ liệu, nhóm dữ liệu đã được phân cho quản trị dữ liệu đó (mô tả phía dưới *)
	5.	Tác nhân	Chọn thêm biên tập viên
	6.	Hệ thống	Hiển thị danh sách biên tập viên
	7.	Tác nhân	Chọn một hay nhiều biên tập viên
	8.	Tác nhân	Chọn chức năng thêm
	9.	Hệ thống	Lưu vào hệ thống
	10.	Hệ thống	Hiển thị danh sách biên tập viên đã được phân cho quản trị dữ liệu
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	5a1.	Tác nhân	Chọn thêm nhóm dữ liệu
	5a2.	Hệ thống	Hiển thị danh sách nhóm dữ liệu
	5a3	Tác nhân	Chọn nhóm dữ liệu cần thêm
	5a4	Tác nhân	Chọn chức năng thêm
	5a5	Hệ thống	Đi đến bước 8
<b>Hậu điều kiện</b>	Không		

\* Dữ liệu đầu ra các mục đã phân quyền cho quản trị dữ liệu

**Bảng 18** Dữ liệu đầu ra các mục đã phân quyền cho quản trị dữ liệu

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
1.	Nhóm biên tập dữ liệu	Các biên tập dữ liệu do Quản trị dữ liệu đó quản lý	Danh sách	<a href="mailto:tuangmail.com">tuangmail.com</a> , hoan@gmail.com

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
2.	Nhóm ý định	Các nhóm ý định mà quản trị dữ liệu đó quản lý	Danh sách	Mở thẻ, Khóa thẻ
3	Nhóm hành động	Các nhóm hành động mà quản trị dữ liệu đó quản lý	Danh sách	Mở thẻ, Khóa thẻ
4	Nhóm hội thoại	Các nhóm hội thoại mà quản trị dữ liệu đó quản lý	Danh sách	ATM, mở tài khoản
5	Nhóm luồng hội thoại	Các nhóm luồng hội thoại mà Quản trị dữ liệu đó quản lý	Danh sách	ATM, mở tài khoản

## B Kiểm thử

### B.1 Kiểm thử Tạo hội thoại từ lịch sử

**Bảng 19** Các trường hợp kiểm thử cho Tạo hội thoại từ lịch sử

ID	Mô tả	Quy trình kiểm thử	Kết quả mong muốn	Kết quả thực tế	Trạng thái
1	Tạo hội thoại từ lịch sử chat với tên rỗng	Chọn chức năng “Chuyển sang hội thoại” từ chi tiết lịch sử chat	Thông báo không thành công	Thông báo không thành công	Đạt
		Chọn lưu nháp			
		Nhập tên rỗng			
		Ấn lưu			

ID	Mô tả	Quy trình kiểm thử	Kết quả mong muốn	Kết quả thực tế	Trạng thái
2	Tạo hội thoại từ lịch sử với nhóm không được phân quyền	Chọn chức năng “Chuyển sang hội thoại” từ chi tiết lịch sử chat	Không hiển thị nhóm được phân quyền	Không hiển thị nhóm được phân quyền	Đạt
		Chọn lưu nháp			
		Chọn nhóm để lưu			
3	Hoàn thành hội thoại từ lịch sử với nhóm khi chưa đánh dấu hoàn thành các đơn vị hội thoại	Chọn chức năng “Chuyển sang hội thoại” từ chi tiết lịch sử chat	Thông báo cần đánh dấu hoàn thành hết các đơn vị hội thoại	Thông báo cần đánh dấu hoàn thành hết các đơn vị hội thoại mới được hoàn thành	Đạt
		Chọn hoàn thành			
		Nhập tên, nhóm			
		Ấn lưu			
4	Hoàn thành hội thoại từ lịch sử chat	Chọn chức năng “Chuyển sang hội thoại” từ chi tiết lịch sử chat	Thông báo thành công, hiển thị thêm một hội thoại ở danh sách bên trái	Thông báo thành công, hiển thị thêm một hội thoại nữa ở danh sách bên trái	Đạt
		Đánh dấu hoàn thành các đơn vị hội thoại			
		Chọn hoàn thành			
		Nhập tên, nhóm			
		Ấn lưu			

## B.2 Kiểm thử chức năng Tạo hội thoại qua chat tương tác

**Bảng 20** Các trường hợp kiểm thử cho Tạo hội thoại qua chat tương tác

ID	Mô tả	Quy trình kiểm thử	Kết quả mong muốn	Kết quả thực tế	Trạng thái
1	Tự động lưu hội thoại khi chat tương tác	Chọn chức năng Tạo hội thoại	Hiển thị nội dung hội thoại và tự động lưu với tên mặc định	Hiển thị nội dung hội thoại và tự động lưu với tên mặc định	Đạt
		Nhấn “Xin chào” tại ô nhập tin nhắn			
2	Gửi tin nhắn rỗng khi chat tương tác	Đặt trỏ chuột vào ô nhập tin nhắn. Ấn phím Enter	Không gửi và xử lý tin nhắn rỗng. Hội thoại giữ nguyên nội dung	Hội thoại giữ nguyên nội dung	Đạt
3	Tự động đánh dấu hoàn thành đơn vị hội thoại	Nhập tin nhắn, đợi hệ thống trả về kết quả	Thêm một đơn vị đã đánh dấu hoàn thành ở hội thoại	Thêm một đơn vị đã đánh dấu hoàn thành ở hội thoại	Đạt

**B.3 Kiểm thử chức năng Duyệt hội thoại****Bảng 21** Các trường hợp kiểm thử cho Duyệt hội thoại

ID	Mô tả	Quy trình kiểm thử	Kết quả mong muốn	Kết quả thực tế	Trạng thái
1	Duyệt hội thoại khi hội thoại đang ở trạng thái nhấp	Chọn một hội thoại ở trạng thái nhấp	Không hiển thị các hành động Chấp nhận, Trả lại, Lưu hội thoại	Không hiển thị các hành động trên.	Đạt
2	Chấp nhận hội thoại khi còn thực thể, hành động, ý định	Chọn một hội thoại có các dữ liệu phụ thuộc chưa ở trạng thái chấp nhận	Thông báo tên các dữ liệu phụ thuộc chưa chấp nhận.	Thông báo tên các dữ liệu phụ thuộc chưa chấp nhận.	Đạt

ID	Mô tả	Quy trình kiểm thử	Kết quả mong muốn	Kết quả thực tế	Trạng thái
	không ở trạng thái Chấp nhận	Ấn nút chấp nhận			
3	Chấp nhận hội thoại khi chưa chấp nhận hết các đơn vị hội thoại	Chọn hội thoại đang ở trạng thái Chờ phê duyệt	Thông báo cần duyệt hết đơn vị hội thoại trước khi chấp nhận	Thông báo cần duyệt hết đơn vị hội thoại trước khi chấp nhận	Đạt
		Ấn nút chấp nhận			