

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

**HỒ THẢO HIỀN - 1412169**

**TÌM HIỂU CÔNG CỤ SMART CHATBOT VÀ XÂY DỰNG  
ỨNG DỤNG HỖ TRỢ TRA CỨU TÀI NGUYÊN HỌC TẬP**

**KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT**

**GIÁO VIÊN HƯỚNG DẪN**

**TS. PHẠM NGUYỄN CƯỜNG**

**KHÓA 2014-2018**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**Xếp loại:**

**Tp.HCM, ngày... tháng...năm...**

**Giáo viên hướng dẫn**

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**Khóa luận đáp ứng yêu cầu của Khóa luận cử nhân CNTT**

**Xếp loại:**

**Tp.HCM, ngày... tháng...năm...**

**Giáo viên phản biện**

## **LỜI CẢM ƠN**

Xin chân thành cảm ơn và tri ân thầy – TS. Phạm Nguyễn Cương, người đã trực tiếp hướng dẫn, định hướng và đồng hành với khóa luận của em trong suốt thời gian vừa qua. Những góp ý chân thành của thầy cũng như việc không ngừng tạo điều kiện thoải mái nhất là tiền đề để nhóm em có thể hoàn thành khóa luận này. Một lần nữa, em xin được gửi đến thầy lời cảm ơn chân thành nhất.

Em cũng xin gửi lời cảm ơn đến quý Thầy Cô trường Đại học Khoa Học Tự Nhiên – Đại học Quốc Gia Tp. Hồ Chí Minh đã tận tình giảng dạy em trong suốt 4 năm học vừa qua. Đặc biệt là quý Thầy Cô khoa Công nghệ thông tin vì đã đem lại cho em những kiến thức nền tảng quý báu giúp em không chỉ đạt kết quả tốt trong học tập mà còn có thể ứng dụng vào trong thực tiễn công việc hằng ngày.

Em cũng xin chân thành cảm ơn Khoa Công nghệ thông tin, trường Đại học Khoa Học Tự Nhiên, Đại học Quốc Gia Tp. Hồ Chí Minh đã tạo điều kiện thuận lợi cho em trong quá trình học tập cũng như thực hiện đề tài tốt nghiệp.

Do khả năng và thời gian thực hiện có hạn, em đã cố gắng hoàn thiện đề tài hết sức, tuy nhiên, những thiếu sót và hạn chế là điều khó có thể tránh khỏi. Chính vì vậy, em rất mong nhận được sự góp ý và cảm thông của quý Thầy Cô.

Cuối cùng, em xin trân trọng cảm ơn và chúc sức khỏe quý Thầy Cô!

Thành phố Hồ Chí Minh, ngày ... tháng... năm 2018

**Hồ Thảo Hiền**

**Khoa Công Nghệ Thông Tin**

**Bộ môn Hệ Thống Thông Tin**

**ĐỀ CƯƠNG CHI TIẾT**

**Tên đề tài:** Tìm hiểu công cụ smart chatbot và xây dựng ứng dụng hỗ trợ tra cứu tài nguyên học tập

**Giáo viên hướng dẫn:** TS. Phạm Nguyễn Cương

**Thời gian thực hiện:** 01/01/2018 đến ngày 30/07/2018

**Sinh viên thực hiện:** 1412169 – Hồ Thảo Hiền

**Loại đề tài:** Tìm hiểu công nghệ có ứng dụng minh họa

**Nội dung đề tài:**

**Về phần nghiên cứu lý thuyết:**

Tìm hiểu các khái niệm liên quan đến chatbot, nguyên lý hoạt động của chatbot và các mô hình phổ biến hiện nay được dùng để xây dựng chatbot.

Tìm hiểu một số nền tảng, công nghệ hỗ trợ xây dựng chatbot và phân nhóm, so sánh các nền tảng mà sinh viên đã biết.

**Về phần ứng dụng minh họa:**

Trải qua quá trình tìm hiểu về khả năng ứng dụng và lợi ích của chatbot trong lĩnh vực hỗ trợ học tập, sinh viên đã tìm hiểu và cài đặt thử nghiệm một chatbot hỗ trợ tra

cứu tài nguyên học tập với tên gọi là eBot. Ý tưởng của ứng dụng dựa trên một trong số các xu hướng phát triển chatbot trong lĩnh vực giáo dục.

### **Về phần công nghệ:**

Để xây dựng ứng dụng minh họa, sinh viên đã sử dụng các công nghệ và công cụ cần thiết để cài đặt và vận hành ứng dụng tích hợp các công nghệ này, cụ thể là:

- Sử dụng Dialogflow – một nền tảng hỗ trợ xử lý ngôn ngữ tự nhiên được hỗ trợ bởi Google để phân tích thông điệp từ người dùng.
- Sử dụng MySQL để lưu trữ dữ liệu liên quan đến môn học.
- Sử dụng NodeJS để xây dựng server cho ứng dụng.

### **Phương pháp thực hiện:**

- Về lý thuyết:
  - Tìm hiểu các tài liệu khoa học liên quan đến khái niệm và cách thức hoạt động của chatbot.
  - Thử nghiệm một số công cụ, nền tảng dùng để xây dựng chatbot phổ biến đang được phát triển
  - Dựa vào kiến thức cá nhân để mô hình hóa hệ thống, phân tích cách thức thiết kế.
- Về kỹ thuật:
  - Sử dụng nền tảng server-side NodeJS
  - Ngôn ngữ lập trình: Python, Javascript
  - Sử dụng RESTful API để xử lý yêu cầu người dùng.

### **Kết quả đạt được:**

- Tìm hiểu được các khái niệm, lợi ích mà chatbot mang lại.
- Hiểu được nguyên lý hoạt động của chatbot và các mô hình phổ biến dùng để xây dựng chatbot.

- Tổng hợp, phân nhóm, so sánh được các nền tảng, công nghệ phổ biến đang được dùng để xây dựng chatbot hiện nay.
- Xây dựng được ứng dụng eBot với giao diện là Facebook Messenger hỗ trợ giúp đỡ thông tin mà sinh viên cần biết khi học đại học.

**Kế hoạch thực hiện:**

Stt	Từ ngày	Đến ngày	Nội dung công việc
1	11/12/2017	03/2018	<p>Nhận đề tài. Lên kế hoạch thực hiện công việc.</p> <p>Đọc tài liệu tìm hiểu về chatbot. Đề xuất kế hoạch tự nghiên cứu và thực hiện luận văn.</p> <p>Xác định mục tiêu, hướng đi cụ thể và phạm vi của đề tài.</p> <p>Tìm hiểu các công nghệ đã và đang được cộng đồng sử dụng để xây dựng chatbot hiện nay.</p> <p>Viết chương 1 luận văn.</p>
2	10/03/2018	31/03/2018	<p>Tìm hiểu khái niệm smart chatbot.</p> <p>Những điều kiện và kiến thức nền tảng cần có để xây dựng một smart chatbot.</p>

			<p>Tính chất cơ bản của một smart chatbot, nguyên lí hoạt động chung của một smart chatbot.</p> <p>Tổng hợp, phân tích, so sánh, đánh giá ưu, nhược điểm,... của các công cụ, công nghệ.</p> <p>Viết mục lục sơ bộ và chương 2 của luận văn.</p>
3	01/04/2018	15/04/2018	<p>Tìm hiểu các hướng xây dựng chatbot trong lĩnh vực giáo dục hiện nay.</p> <p>Làm khảo sát và xác định các tính năng cần có của ứng dụng.</p> <p>Viết chương 3 luận văn.</p>
4	16/04/2018	25/04/2018	<p>Chọn ngôn ngữ lập trình, môi trường phát triển và công cụ để phát triển ứng dụng chatbot hỗ trợ tài nguyên học tập. Giải thích nguyên nhân cho sự lựa chọn những giải pháp đó.</p> <p>Thực hiện khảo sát, lên danh sách chức năng cần cài đặt.</p> <p>Viết chương 4 luận văn</p>



5	26/04/2018	31/05/2018	Thiết kế kiến trúc hệ thống cho ứng dụng và cơ sở dữ liệu của ứng dụng  Viết tiếp chương 4 luận văn
6	01/06/2018	20/06/2018	Cài đặt, cập nhật và sửa lỗi ứng dụng.  Viết chương 4 luận văn
7	21/06/2018	25/06/2018	Tổng kết, đánh giá những kiến thức thu được, sản phẩm thu được.  Viết chương 5 luận văn.
8	26/06/2018	30/06/2018	Hoàn thiện báo cáo cuốn.
<p><b>Xác nhận của GVHD</b></p> <p><b>TS. Phạm Nguyễn Cương</b></p>			<p><b>Ngày .....tháng.....năm.....</b></p> <p><b>SV thực hiện</b></p> <p><b>Hồ Thảo Hiền</b></p>

## MỤC LỤC

<b>CHƯƠNG 1: GIỚI THIỆU .....</b>	<b>1</b>
1. Giới thiệu đề tài.....	1
2. Lí do chọn đề tài.....	2
3. Mục tiêu của luận văn .....	2
4. Phạm vi đề tài.....	3
5. Cấu trúc khóa luận .....	4
<b>CHƯƠNG 2: LÝ THUYẾT NỀN TẢNG CỦA VIỆC XÂY DỰNG CHATBOT VÀ NGUYÊN TẮC VẬN HÀNH CỦA CHATBOT. ....</b>	<b>6</b>
1. Định nghĩa chatbot.....	6
2. Phân loại chatbot.....	7
3. Các khái niệm cơ sở liên quan đến việc xây dựng chatbot.....	9
3.1. Giới thiệu khái niệm trí thông minh nhân tạo (artificial intelligent) .....	10
3.2. Giới thiệu khái niệm máy học (machine learning).....	10
3.3. Khái niệm xử lí ngôn ngữ tự nhiên (natural language processing).....	12
3.4. Các khái niệm về xử lí ngôn ngữ tự nhiên trong chatbot.....	12
3.4.1. Khái niệm ý định (intent) .....	12
3.4.2. Khái niệm thực thể (entity) .....	13
3.4.3. Khái niệm ngữ cảnh (context) .....	14
4. Nguyên tắc vận hành của chatbot .....	15
4.1. Phân tích nội dung đầu vào .....	16
4.1.1. Phân loại ý định (intent classification).....	17
4.1.2. Nhận dạng thực thể (named-entity recognition) .....	19
4.2. Xây dựng phản hồi .....	20
4.3. Các mô hình được dùng để xây dựng chatbot.....	21
4.3.1. Mô hình dựa trên chẩn đoán (pattern-based heuristic model).....	21
4.3.2. Mô hình dựa trên truy xuất (retrieval-based models).....	23
4.3.3. Mô hình tự phát sinh (generative models).....	24
4.4. Tổng kết và đánh giá các mô hình .....	25

5.	Tổng hợp, so sánh một số nền tảng, công nghệ sử dụng xây dựng chatbot .....	28
5.1.	Các nhóm nền tảng, công nghệ xây dựng chatbot .....	28
5.2.	So sánh các nhóm nền tảng, công nghệ phổ biến dùng để xây dựng chatbot	29

### **CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ GIẢI PHÁP CHO ỨNG DỤNG CHATBOT HỖ TRỢ HỌC TẬP.....33**

1.	Tổng quan về các hướng xây dựng chatbot trong lĩnh vực giáo dục.....	33
2.	Tổng quan về ứng dụng .....	35
3.	Yêu cầu và ngữ cảnh của ứng dụng .....	35
4.	Giải pháp xây dựng phản hồi .....	40
5.	Cấu trúc cơ sở dữ liệu dùng để xây dựng phản hồi .....	42

### **CHƯƠNG 4: CÀI ĐẶT ỨNG DỤNG CHATBOT HỖ TRỢ HỌC TẬP .....45**

1.	Sơ đồ tổng quát hoạt động của ứng dụng.....	45
2.	Cài đặt ứng dụng .....	46
2.1.	Quá trình chuẩn bị.....	46
2.1.1.	Về dữ liệu .....	47
2.1.1.1.	Xây dựng dữ liệu và huấn luyện mô hình trên Dialogflow .....	47
2.1.1.2.	Cơ sở dữ liệu .....	51
2.1.2.	Về giao diện.....	51
2.1.3.	Về cấu hình ứng dụng.....	53
2.1.3.1.	Tạo ứng dụng .....	53
2.1.3.2.	Liên kết agent trên Dialogflow và ứng dụng Facebook với ứng dụng chatbot eBot thông qua mã truy cập (token) .....	55
2.1.3.3.	Cấu hình webhook.....	55
2.2.	Quá trình cài đặt chi tiết .....	56
2.2.1.	Cấu trúc thư mục của chương trình:.....	56
2.2.2.	Chi tiết cài đặt các hàm .....	57
2.2.2.1.	Cài đặt và cấu hình cơ sở dữ liệu .....	58
2.2.2.2.	Cài đặt webhook.....	58

2.2.2.3. Cài đặt chức năng.....	61
<b>CHƯƠNG 5: TỔNG KẾT, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>66</b>
1. Kết quả đạt được và hạn chế.....	66
1.1. Về lý thuyết.....	66
1.2. Về ứng dụng.....	66
1.3. Hạn chế.....	67
2. Hướng phát triển .....	67
<b>PHỤ LỤC.....</b>	<b>68</b>
1. AIML .....	68
2. Chatscript .....	71
3. Dialogflow .....	73
4. Chi tiết cài đặt các hàm trong chương trình.....	74
5. Chi tiết thuộc tính các bảng trong cơ sở dữ liệu .....	81
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>86</b>

## DANH MỤC BẢNG

Bảng 1: Thông tin được trích xuất từ ví dụ.....	19
Bảng 2: Bảng đánh giá các mô hình được dùng để xây dựng phản hồi cho chatbot .....	26
Bảng 3: Phân loại các nền tảng xây dựng chatbot. ....	28
Bảng 4: So sánh đặc điểm của ba nền tảng lớn được dùng để xây dựng chatbot .....	31
Bảng 5: Bảng danh sách các yêu cầu hỗ trợ có bình chọn trên 80% .....	38
Bảng 6: Các tính năng của ứng dụng eBot.....	39
Bảng 7: Danh sách các bảng trong cơ sở dữ liệu .....	44
Bảng 8: Danh sách các intent được dùng để huấn luyện mô hình trên Dialogflow.....	49
Bảng 9: Danh sách các entity được dùng để rút trích thông tin trong các câu mẫu trong mỗi intent trên Dialogflow .....	49
Bảng 10: Danh sách file và thư mục được dùng để xây dựng ứng dụng .....	57
Bảng 11: Danh sách các hàm được cài đặt trong chương trình .....	62
Bảng 12: Cấu trúc dữ liệu bảng INTENT_RESPONSE .....	81
Bảng 13: Cấu trúc dữ liệu bảng NGANH .....	82
Bảng 14: Cấu trúc dữ liệu bảng CHUYENNGANH .....	82
Bảng 15: Cấu trúc dữ liệu bảng MONHOC .....	82
Bảng 16: Cấu trúc dữ liệu bảng NOIDUNGCACBUOIHOC .....	83
Bảng 17: Cấu trúc dữ liệu bảng BAITAP .....	83
Bảng 18: Cấu trúc dữ liệu bảng SINHVIEN.....	84
Bảng 19: Cấu trúc dữ liệu bảng DIEMSO .....	84
Bảng 20: Cấu trúc dữ liệu bảng MONHOCTHAYTHE.....	84
Bảng 21: Cấu trúc dữ liệu bảng MONHOCTRUOC .....	85
Bảng 22: Cấu trúc dữ liệu bảng CHUYENNGANH_MONHOC.....	85

## DANH MỤC HÌNH ẢNH

Hình 1: Phân loại chatbot.....	7
Hình 2: Các nhóm thuật toán máy học.....	11
Hình 3: Minh họa một entity .....	14
Hình 4: Kiến trúc chung của chatbot .....	16
Hình 5: Phân loại văn bản bằng các thuật toán học có giám sát.....	18
Hình 6: Mô hình dựa trên truy xuất .....	23
Hình 7: Mô hình tự phát sinh .....	25
Hình 8: Danh sách câu hỏi liên quan đến đến giáo vụ và kết quả bình chọn .....	36
Hình 9: Danh sách câu hỏi liên quan đến đến quản trị, hỗ trợ.....	36

Hình 10: Danh sách câu hỏi liên quan đến nội dung môn học .....	37
Hình 11: Giao diện cung cấp nhắc nhở đối với các tham số bắt buộc.....	42
Hình 12: Lược đồ ER của cơ sở dữ liệu của ứng dụng eBot .....	43
Hình 13: Sơ đồ hoạt động của ứng dụng chatbot eBot .....	45
Hình 14: Minh họa phân nhóm nội dung vào các intent.....	50
Hình 15: Minh họa về việc xác định tham số trong câu hỏi .....	51
Hình 16: Giao diện tin nhắn hội thoại Messenger của Facebook. ....	52
Hình 17: Ứng dụng NodeJS tối thiểu.....	54
Hình 18: Chương trình chạy bằng lệnh node .....	54
Hình 19: Chương trình chạy bằng lệnh nodemon .....	54
Hình 20: Minh họa đưa mã truy cập vào code để xác thực .....	55
Hình 21: Minh họa tên miền tùy chọn được tạo ra bởi Ngrok.....	56
Hình 22: Nội dung tập tin dùng để cấu hình cơ sở dữ liệu .....	58
Hình 23: Đoạn mã dùng để đăng ký và xác thực webhook .....	58
Hình 24: Giao diện để cung cấp mã xác thực .....	59
Hình 25: Đoạn mã dùng để tạo điểm kết nối cuối để nhận yêu cầu POST.....	60
Hình 26: Sơ đồ thực thi của ứng dụng .....	61
Hình 27: Mã nguồn của hàm confirmActionInfo.....	63
Hình 28: Mã nguồn của hàm doAction.....	64
Hình 29: Một phần mã nguồn của hàm responseToUser.....	64
Hình 30: Hàm gửi tin nhắn dạng text đến người dùng.....	74
Hình 31: Hàm gửi hành động đang gõ phím đến người dùng .....	74
Hình 32: Hàm gửi biểu mẫu đến người dùng.....	75
Hình 33: Hàm dùng để tìm kiếm kết quả từ Google và trả về cho người dùng.....	76
Hình 34: Thông tin được gửi để xác nhận với người dùng.....	77
Hình 35: Hàm kiểm tra thông tin được xác nhận.....	78
Hình 36: Hàm trả về kết quả cho người dùng.....	79
Hình 37: Hàm thay đổi các tham số trong câu truy vấn.....	80
Hình 38: Hàm thay đổi các tham số trong procedure. ....	80
Hình 39: Sơ đồ ER của cơ sở dữ liệu.....	81

## CHƯƠNG 1: GIỚI THIỆU

*Mục tiêu của chương một là để giới thiệu đề tài, nêu bật lên được lợi ích của chatbot và tầm quan trọng của chatbot trong lĩnh vực giáo dục, đồng thời cũng nêu rõ lý do của việc chọn lựa đề tài. Tiếp đến là phần trình bày mục tiêu cần đạt được của luận văn. Cuối cùng sẽ nêu ra phạm vi nghiên cứu về lý thuyết và phạm vi thực hiện ứng dụng sẽ được xây dựng.*

### 1. Giới thiệu đề tài

Trong thời điểm đổi mới đào tạo, quản lý giáo dục, cải thiện môi trường học tập được chú trọng như hiện nay, việc đẩy mạnh và mở rộng công tác tư vấn, hỗ trợ giúp cho sinh viên học tập và làm việc hiệu quả hơn. Ngoài ra, việc sở hữu điện thoại thông minh hay máy tính cá nhân hiện nay đối với sinh viên đã trở nên rất phổ biến. Do đó, sử dụng một phần mềm chạy đa nền tảng để hỗ trợ sinh viên mọi lúc mọi nơi, tại mọi thời điểm trong những vấn đề học tập sẽ dễ dàng, hiệu quả và ít tốn chi phí hơn so với cách truyền thống. Chatbot hỗ trợ học tập có thể được xem là một giải pháp hiệu quả.

Chatbot (hay còn được gọi là talkbot, chatter bot,...) là một chương trình được lập trình nhằm quản lý và mô phỏng lại cách thức con người thực hiện một cuộc trò chuyện thông qua 2 hình thức chính là giọng nói và văn bản.

Theo [1], các lợi ích mà chatbot có thể mang lại cho con người là:

- Luôn sẵn sàng 24/7 để phục vụ cho con người.
- Khả năng xử lý đồng thời và giao tiếp với nhiều người cùng lúc.
- Giúp tiết kiệm chi phí cho các dịch vụ hỗ trợ người dùng.
- Đảm bảo sự hài lòng với người được giao tiếp mà không phụ thuộc vào cảm xúc.
- Hỗ trợ nhiều ngôn ngữ khác nhau.
- Tự động hóa các công việc lặp đi lặp lại.

Xuất phát từ những nhu cầu và những thuận lợi mà chatbot mang lại, việc chọn đề tài “Tìm hiểu công cụ smart chatbot và xây dựng ứng dụng hỗ trợ tra cứu tài nguyên học tập” để nghiên cứu và cài đặt thử là một quyết định hợp lý.

## **2. Lí do chọn đề tài**

Nhằm mục đích hệ thống hóa các kiến thức của bản thân trong quá trình học tập vào trong một sản phẩm có ý nghĩa thực tế, có ích trong tương lai, sinh viên lựa chọn đề tài: “Tìm hiểu công cụ smart chatbot và xây dựng ứng dụng hỗ trợ tra cứu tài nguyên học tập”.

Vì còn là sinh viên, hiểu rõ những khúc mắc mà bản thân đã trải qua trong những năm học đại học đã tạo ra sự nhìn nhận tốt hơn về những chức năng cần có khi cài đặt ứng dụng chatbot hỗ trợ tra cứu tài nguyên học tập. Ngoài ra, quá trình nghiên cứu và cài đặt ứng dụng tạo cơ hội tốt cho việc tìm hiểu thêm về các khái niệm, công nghệ mới có thể giúp ích cho công việc trong tương lai.

Bên cạnh đó, chatbot hiện đang là một chủ đề được cộng đồng rất quan tâm vì lợi ích to lớn mà nó mang lại. Chatbot hiện đang là một nền tảng có tiềm năng rất lớn trong lĩnh vực giáo dục. So sánh với cách hỗ trợ học tập truyền thống thì chatbot rẻ hơn, giúp con người tiết kiệm nhiều thời gian hơn, dễ tiếp cận hơn và luôn sẵn sàng để hỗ trợ người học mọi lúc mọi nơi [2]. Do đó, nếu được phát triển tốt, ứng dụng sẽ có khả năng được áp dụng vào thực tế và trở thành ứng dụng có thể giúp đỡ nhà trường trong công tác tư vấn, hỗ trợ sinh viên về một số vấn đề học tập.

## **3. Mục tiêu của luận văn**

Để hoàn thành tốt đề tài luận văn, bản luận văn và sản phẩm ứng dụng cuối cùng sẽ đảm bảo tối thiểu các mục tiêu sau đây:

- **Về lý thuyết:**
  - Bản luận văn sẽ trình bày rõ ràng về những khái niệm cơ sở trong việc xây dựng chatbot và nguyên lý vận hành chung của một chatbot. Đồng thời



cũng sẽ trình bày rõ ràng một số mô hình thường được dùng để xây dựng chatbot.

- Trình bày rõ ràng các nền tảng, công nghệ phổ biến được dùng để xây dựng chatbot và phân nhóm các nền tảng công nghệ. Sau khi phân nhóm, luận văn cũng sẽ nêu lên ưu, nhược điểm của ba nền tảng phát triển bot phổ biến, được sự hỗ trợ của các tập đoàn lớn như Google, Microsoft và Facebook. Những nền tảng này đã được thử nghiệm trong suốt quá trình làm luận văn để chọn ra nền tảng phù hợp nhất.

- **Về ứng dụng:**

- Trình bày kết quả khảo sát yêu cầu người dùng về các tính năng cần có của ứng dụng do sinh viên thực hiện trên hơn 100 bạn sinh viên trường đại học Khoa học tự nhiên TP HCM.
- Sản phẩm ứng dụng của luận văn là một ứng dụng chatbot được triển khai trên nền tảng Messenger của Facebook, đảm bảo luôn sẽ có câu trả lời cho người dùng trong mọi trường hợp.

#### **4. Phạm vi đề tài**

Yêu cầu ban đầu của đề tài là: “Tìm hiểu công cụ smart chatbot và xây dựng ứng dụng hỗ trợ tra cứu tài nguyên học tập”. Đề tài này sẽ thực hiện ba công việc:

- Khái quát định nghĩa của một chatbot, các khái niệm cơ sở liên quan đến việc xây dựng chatbot và nguyên tắc vận hành chung của một chatbot.
- Thực hiện tổng hợp, so sánh, đánh giá các công nghệ, nền tảng phổ biến nhất hiện nay được sử dụng để xây dựng một chatbot. Các công nghệ, nền tảng được so sánh dưới dạng bảng thông qua các yếu tố như: giấy phép sử dụng, tài liệu, khả năng hỗ trợ huấn luyện các mẫu và cộng đồng hỗ trợ của nền tảng đó. Sau khi so sánh, luận văn cũng sẽ nêu ra những nhận định và lý giải lý do chọn một nền tảng nhất định để xây dựng ứng dụng.

- Thiết kế giải pháp và cài đặt một chatbot với tên gọi là eBot với khả năng hỗ trợ tra cứu tài nguyên học tập. Thay vì phải xử lý nội dung và ngữ cảnh cho một cuộc đối thoại dài, ứng dụng chỉ tập trung vào việc trả lời một cách ngắn gọn câu hỏi đơn cuối cùng của người dùng về vấn đề học tập và hỗ trợ người dùng thực hiện một số hành động đơn giản. Sản phẩm luận văn được thực hiện một cách toàn diện nên có thể áp dụng và mở rộng các thư viện có sẵn cũng như cài đặt thô để đáp ứng được các yêu cầu đặc thù nhằm đạt được các mục tiêu và yêu cầu chức năng chuyên biệt.

## 5. Cấu trúc khóa luận

Dưới đây là tóm tắt nội dung của toàn bộ luận văn. Nội dung mỗi chương được liệt kê chi tiết như sau:

### - **Chương 1: Mở đầu.**

Trình bày sơ lược về đề tài, lý do tại sao nhóm lại chọn đề tài này và đề tài này giúp giải quyết vấn đề gì trong thực tế, đồng thời giới thiệu mục tiêu và phạm vi của đề tài.

### - **Chương 2: Tổng quan về chatbot, lý thuyết nền tảng của chatbot, tổng hợp và phân tích các nền tảng xây dựng chatbot phổ biến.**

- Khái niệm chatbot và phân loại các loại chatbot.
- Cơ sở lý thuyết liên quan đến việc xây dựng chatbot.
- Các khái niệm xử lý ngôn ngữ tự nhiên liên quan trực tiếp đến việc xây dựng chatbot như ý định (intent), thực thể (entity), ngữ cảnh (context).
- Nguyên lý vận hành của chatbot bao gồm hai quá trình: phân tích nội dung đầu vào và tạo phản hồi.
- Ba mô hình phổ biến dùng để xây dựng chatbot dựa trên cách phân tích ý định của người dùng và cách xây dựng câu trả lời.
- Tổng hợp, so sánh các công nghệ, nền tảng phổ biến dùng để xây dựng chatbot.

- Phân chia các nhóm công nghệ, nền tảng phổ biến được sử dụng để xây dựng chatbot hiện nay.
  - So sánh các công nghệ, nền tảng được trải nghiệm (Dialogflow, Microsoft Bot Framework và Luis, Wit.ai) trong quá trình tìm hiểu làm luận văn.
- **Chương 3: Phân tích và thiết kế giải pháp cho ứng dụng chatbot hỗ trợ học tập.**
- Tổng quan về các hướng xây dựng chatbot trong lĩnh vực giáo dục và tổng quan về ứng dụng eBot sẽ được xây dựng.
  - Quá trình và kết quả của khảo sát để lấy yêu cầu người dùng.
  - Giải pháp xây dựng phản hồi cho ứng dụng chatbot eBot.
- **Chương 4: Cài đặt và thử nghiệm cho ứng dụng chatbot hỗ trợ học tập.**
- Trình bày sơ đồ hoạt động trong thực tế của ứng dụng.
  - Trình bày chi tiết về quá trình cài đặt ứng dụng bao gồm quá trình chuẩn bị và cài đặt chi tiết.
- **Chương 5: Tổng kết và đánh giá.**
- Trình bày kết quả và kinh nghiệm đạt được sau khi làm đề tài và nêu những vấn đề, khó khăn đã gặp phải, đề xuất hướng khắc phục. Bên cạnh đó, phần này cũng nêu lên hướng phát triển của đề tài và những đề xuất nhằm cải tiến ứng dụng trong tương lai.

## **CHƯƠNG 2: LÝ THUYẾT NỀN TẢNG CỦA VIỆC XÂY DỰNG CHATBOT VÀ NGUYÊN TẮC VẬN HÀNH CỦA CHATBOT.**

*Chương 2 đầu tiên sẽ tập trung làm rõ định nghĩa chatbot và các khái niệm cơ sở liên quan đến việc xây dựng chatbot; sau đó sẽ trình bày nguyên tắc vận hành của chatbot và các mô hình phổ biến được dùng để xây dựng chatbot hiện nay. Cuối cùng sẽ thực hiện tổng hợp, so sánh và đánh giá một số các nền tảng, công nghệ phổ biến được dùng để xây dựng chatbot hiện nay.*

### **1. Định nghĩa chatbot**

Theo trang [aws.amazon.com](https://aws.amazon.com) [3] chatbot được bắt nguồn từ “chat robot”, là chương trình cho phép nhiều trải nghiệm đàm thoại khác nhau thông qua giọng nói và văn bản, có thể được tùy chỉnh và sử dụng trên thiết bị di động, trình duyệt và trên các nền tảng trò chuyện phổ biến khác. Với sự ra đời của các công nghệ học sâu (deep-learning) như: chuyển văn bản thành giọng nói, nhận dạng giọng nói tự động và xử lý ngôn ngữ tự nhiên, các chatbot mô phỏng cuộc trò chuyện và đối thoại của con người.

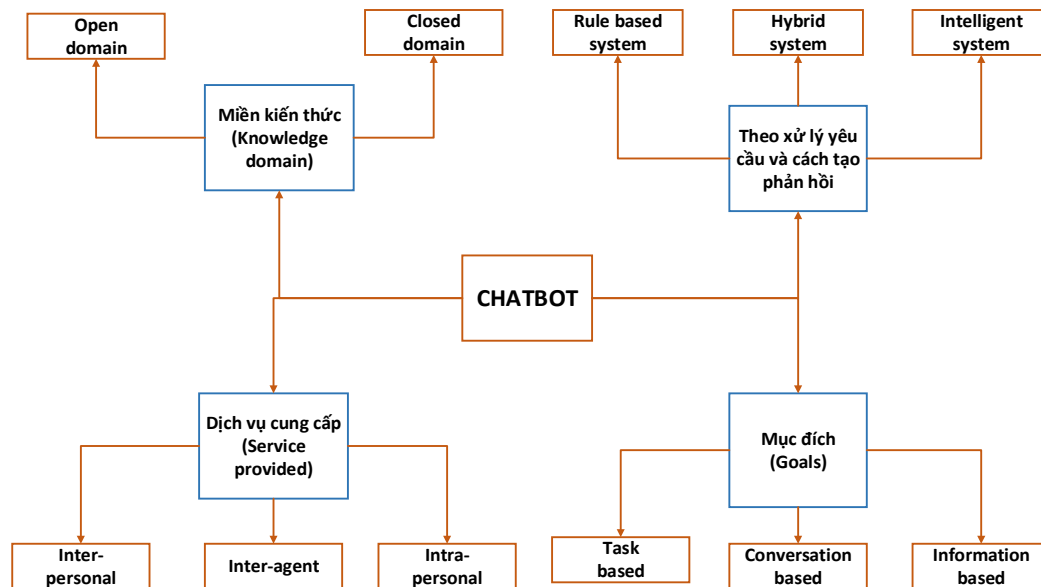
Định nghĩa một cách khác chi tiết hơn, theo Wikipedia [4], chatbot (còn được gọi là talkbot, chatterbot, tác nhân tương tác hoặc thực thể trò chuyện nhân tạo) là một chương trình máy tính hoặc trí thông minh nhân tạo thực hiện cuộc trò chuyện qua giọng nói hoặc văn bản. Các chương trình như vậy thường được thiết kế để mô phỏng cách con người cư xử như một đối tác đàm thoại. Một số chatbot sử dụng hệ thống xử lý ngôn ngữ tự nhiên phức tạp, nhưng nhiều hệ thống đơn giản chỉ quét từ khóa trong đầu vào, sau đó trả lời với từ khóa phù hợp nhất hoặc xây dựng câu trả lời từ cơ sở dữ liệu.

Ngoài hai định nghĩa trên, có nhiều nguồn tài liệu khác định nghĩa về chatbot một cách khác nhau. Tuy nhiên, sau khi tổng kết, chatbot có thể được định nghĩa như sau.

Chatbot là một chương trình được lập trình nhằm quản lý và mô phỏng lại cách thức con người thực hiện một cuộc trò chuyện thông qua giọng nói và văn bản, có thể được tùy chỉnh và sử dụng trên thiết bị di động, trình duyệt và trên các nền tảng trò chuyện phổ biến. Các chatbot được phát triển dựa trên những kỹ thuật khác nhau. Một số chatbot được xây dựng đơn giản như: quét các từ khóa trong câu truy vấn sau đó trích lọc và trả về câu trả lời hoặc so khớp với bộ dữ liệu đã xây dựng trước đó. Một số chatbot phức tạp hơn được xây dựng một hệ thống xử lý ngôn ngữ tự nhiên phức tạp, ứng dụng máy học hoặc các cơ chế phân tích cú pháp khác để có thể phân loại, phân tích câu truy vấn đầu vào của người dùng để đưa ra câu trả lời.

## 2. Phân loại chatbot

Khi phân loại chatbot, cần xác định tiêu chí chính xác để phân loại vì với mỗi tiêu chí khác nhau sẽ có những loại chatbot khác nhau. Theo [5], chatbot được chia vào 4 danh mục sau (hình 1):



Hình 1: Phân loại chatbot

- Theo miền kiến thức (Knowledge domain): chatbot được phân loại dựa trên lượng kiến thức được huấn luyện.

- Miền mở (Open domain): chatbot có thể thực hiện các cuộc trò chuyện với nhiều chủ đề khác nhau.
  - Miền đóng (Closed domain): các chatbot thuộc nhóm này chỉ tập trung vào một miền tri thức cụ thể và có thể không trả lời các câu hỏi nằm ngoài vùng tri thức này.
- Dịch vụ cung cấp (Service provided)
- Interpersonal: Những chatbot thuộc loại này sẽ không đồng hành cùng người dùng mà chỉ đơn thuần là lấy thông tin và trả về cho người dùng. Chatbot loại này không cần phải nhớ thông tin về người dùng.
  - Intrapersonal: các chatbot này mang tính phục vụ cá nhân như lên lịch hẹn, lưu giữ các ý kiến được ghi lại, chatbot loại này sẽ đồng hành cùng người dùng và hiểu người dùng giống như con người.
  - Inter-agent: chatbot dạng này thường xuất hiện trong các lĩnh vực IoT, bên trong sẽ có hai hệ thống thực hiện giao tiếp với nhau để hoàn thành một tác vụ. Sự kết hợp của hai chatbot Alexa-Cortana là một ví dụ về việc giao tiếp giữa các hệ thống chatbot với nhau.
- Mục đích (Goal):
- Cung cấp thông tin (informative): chatbot loại này được thiết kế để cung cấp thông tin cho người dùng dựa trên dữ liệu được lưu trữ trước đó hoặc có sẵn ở một nguồn cố định. Đây là loại chatbot thường được dùng để thay thế trang các câu hỏi thường gặp trên các trang website.
  - Đàm thoại (conversation based): các chatbot loại này “trò chuyện” với người dùng, mục đích của chúng là trả lời chính xác yêu cầu được đưa ra và tiếp tục cuộc hội thoại.
  - Công việc (task based): loại chatbot này được xây dựng để đáp ứng một công việc cụ thể như đặt vé máy bay, đặt phòng khách sạn,... Chatbot loại

này cần một danh sách các hành động cần thiết để hoàn thành một nhiệm vụ định trước.

- Theo cách xử lý yêu cầu đầu vào và cách tạo ra phản hồi (Input processing and response generation method)
  - Intelligent system: hệ thống thông minh sẽ dùng kỹ thuật xử lý ngôn ngữ tự nhiên để phân tích nội dung đầu vào. Các hệ thống này được sử dụng khi miền tri thức được giới hạn và có đủ dữ liệu để huấn luyện.
  - Rule-based system: hệ thống loại này sử dụng phương pháp so khớp mẫu. Loại chatbot này được đánh giá là cứng nhắc. Do đó chúng chỉ được sử dụng khi có kịch bản cố định.
  - Hybrid system: hệ thống là sự kết hợp giữa các các hệ thống rule-based và các hệ thống sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên.

Một chatbot không nhất thiết phải thuộc vào một loại cụ thể, đó có thể là sự kết hợp giữa một hoặc nhiều loại chatbot nằm ở các danh mục khác nhau. Với ứng dụng eBot được cài đặt; nếu phân loại chatbot theo mục đích thì ứng dụng sẽ vừa là chatbot cung cấp thông tin (informative) vừa là chatbot hỗ trợ công việc (task based) vì nó vừa giúp người dùng trả lời một số câu hỏi vừa giúp người dùng thực hiện một số hành động đơn giản; nếu phân loại chatbot theo cách xử lý yêu cầu đầu vào và cách tạo ra phản hồi thì ứng dụng thuộc vào nhóm hybrid system vì hệ thống giúp xử lý truy vấn đầu vào vừa sử dụng phương pháp so khớp mẫu, vừa sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên. Phần tiếp theo sẽ trình bày các khái niệm cơ sở liên quan đến việc xây dựng chatbot.

### **3. Các khái niệm cơ sở liên quan đến việc xây dựng chatbot**

Hiện nay chatbot được xây dựng để bắt chước con người theo cách tốt nhất bằng cách sử dụng các kiến thức về trí thông minh nhân tạo (artificial intelligent), máy học (machine learning) và xử lý ngôn ngữ tự nhiên (natural language processing). Trong phần sau của luận văn, các khái niệm này được lặp đi lặp lại nhiều

lần. Do đó việc hiểu được các khái niệm này sẽ giúp cho việc đọc hiểu luận văn trở nên dễ dàng hơn.

### **3.1. Giới thiệu khái niệm trí thông minh nhân tạo (artificial intelligent)**

Trí thông minh nhân tạo là một hệ thống phần mềm được con người thiết kế và phát triển để mô phỏng lại trí thông minh của con người với một mục đích nào đó nhất định.

Để một trí tuệ nhân tạo có thể học tập, suy luận với độ chính xác cao thì cần có một tập dữ liệu huấn luyện đủ lớn và tri thức của con người trên một miền kiến thức nhất định được mô hình hóa và lưu trữ dưới dạng mà máy có thể hiểu và xử lý được. Phương pháp được gọi là kỹ thuật tri thức (knowledge engineering). Các nhóm thông tin phức tạp được tổ chức và biên dịch thành cơ sở tri thức bởi các kỹ sư tri thức.

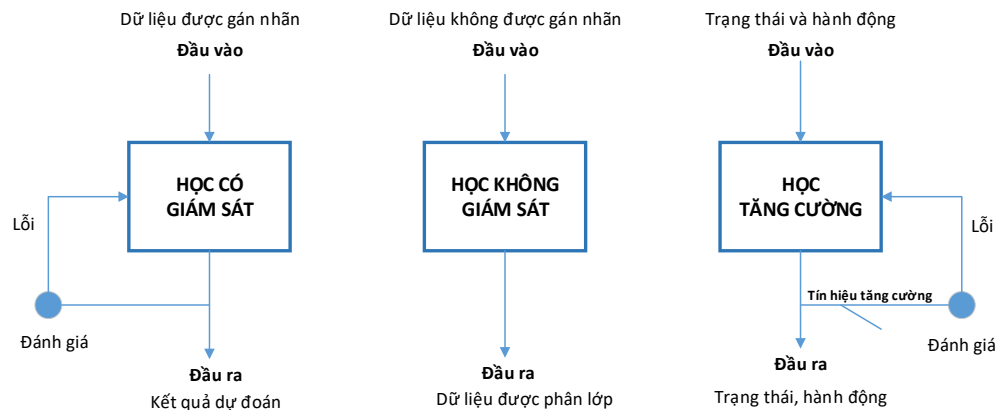
Một trong những phần nòng cốt của trí thông minh nhân tạo là các thuật toán máy học (machine learning) và xử lý ngôn ngữ tự nhiên (natural language processing). Đây là cũng là những thành phần cốt lõi để xây dựng chatbot thông minh hiện đại. Các khái niệm về máy học (machine learning) và xử lý ngôn ngữ tự nhiên (natural language processing) được trình bày ngay sau đây.

### **3.2. Giới thiệu khái niệm máy học (machine learning)**

Máy học là một kỹ thuật nền tảng dùng để huấn luyện các trí thông minh nhân tạo có một cơ sở tri thức nhất định dựa trên tập dữ liệu huấn luyện mẫu. Các thuật toán máy học giúp biến tập dữ liệu đầu vào thành các tri thức cho các trí thông minh nhân tạo học với mục đích cụ thể rõ ràng.

Các thuật toán máy học có thể được chia làm 3 nhóm như hình 2, phụ thuộc vào thông tin dùng để huấn luyện (xem thêm tại [6])





Hình 2: Các nhóm thuật toán máy học.

- Học có giám sát (supervised learning): là một kỹ thuật máy học để học tập từ tập dữ liệu được gán nhãn cho trước. Nhiệm vụ của học có giám sát là dự đoán đầu ra mong muốn dựa vào giá trị đầu vào để máy có thể phát hiện và sắp xếp được thông tin mới. Học có giám sát thường được áp dụng để giải quyết hai dạng bài toán là: bài toán dự đoán (regression problem) và bài toán phân lớp (classification). Một số thuật toán của mô hình học có giám sát là: Support Vector Machine, Naïve Bayes, Decision Trees,...
- Học không giám sát (unsupervised learning): là một kỹ thuật của máy học nhằm tìm ra một mô hình hay cấu trúc bị ẩn bởi tập dữ liệu không được gán nhãn cho trước. Ứng dụng phổ biến nhất của học không giám sát là gom cụm (cluster). Một số thuật toán phổ biến của mô hình học không giám sát có thể kể đến là: k-Means, Mean Shift, Adaptive Resonance Theory,...
- Học tăng cường (reinforcement learning): loại mô hình cải tiến từ mô hình học có giám sát. Hệ thống sẽ đưa ra một mục tiêu để làm theo, sau đó đánh giá và cung cấp phản hồi, không phân biệt phản hồi là tích cực hoặc tiêu cực. Mô hình học tăng cường phổ biến là Q-learning.

Các thuật toán máy học đóng vai trò rất quan trọng trong một hệ thống chatbot thông minh, chúng được dùng sử dụng để phân loại yêu cầu của người dùng vào các

ý định (intent) cụ thể và xác định thực thể (entity) để trích xuất dữ liệu có liên quan từ chúng. Thuật ngữ ý định (intent) và thực thể (entity) được trình bày phía sau.

### **3.3. Khái niệm xử lý ngôn ngữ tự nhiên (natural language processing)**

Xử lý ngôn ngữ tự nhiên đóng vai trò vô cùng quan trọng trong các hệ thống chatbot phức tạp. Xử lý ngôn ngữ tự nhiên nghĩa là kỹ thuật sử dụng các thuật toán máy học, phân tích cú pháp hay phân tích ngữ nghĩa giúp phân tích ngôn ngữ tự nhiên thành các dạng mà máy có thể hiểu được.

### **3.4. Các khái niệm về xử lý ngôn ngữ tự nhiên trong chatbot**

Phần này sẽ trình bày ba khái niệm quan trọng nhất về xử lý ngôn ngữ tự nhiên trong chatbot là: ý định (intent), thực thể (entity) và ngữ cảnh (context). Vì dùng các thuật ngữ tiếng Việt sẽ gây nhiều nhầm lẫn và khó hiểu trong lúc giải thích về khái niệm cũng như phần trình bày các nội dung phía sau, do đó sẽ dùng thuật ngữ “intent” thay cho thuật ngữ “ý định”, thuật ngữ “entity” thay cho thuật ngữ “thực thể” và thuật ngữ “context” thay cho thuật ngữ “ngữ cảnh” trong toàn bộ phần sau của luận văn.

#### **3.4.1. Khái niệm ý định (intent)**

Intent là đại diện cho một ánh xạ giữa thông điệp của người dùng và hành động mà phần mềm cần phải thực hiện. Có thể hiểu intent là tên chủ đề cho một thông điệp của người dùng, nó mô tả ngắn gọn mục đích hoặc mục tiêu trong câu nói của người dùng với một từ khóa ngắn gọn được quy định trước đó.

Ví dụ: Xét hai câu sau:

*Câu 1: “Điểm số môn học Cơ sở dữ liệu.”*

*Câu 2: “Tôi muốn biết điểm môn Trí tuệ nhân tạo”*

Cả 2 câu mẫu được cung cấp ở trên đều có nội dung cơ bản là để hỏi về điểm số. Vì vậy, ta có thể xếp chúng và các câu tương tự vào intent có tên là “*diem\_so*”.

Khi người dùng nhập vào một câu, hệ thống sẽ dùng thuật toán để tìm ra intent của câu. Biết được intent của một câu giúp lập hệ thống có thể hiểu được ý định của người dùng và xây dựng phản hồi thích hợp.

Trong một intent thường có các thành phần:

- **Phát biểu (utterance):** Những câu mẫu mà người dùng có thể nhập vào trong ứng dụng.
- **Phản hồi (response):** là câu trả lời được trả về khi một intent được kích hoạt.
- **Tham số (parameter):** là thành phần dùng để kết nối thông tin trong câu nói của người dùng với các entity. Thuật ngữ entity sẽ được trình bày phía sau đây.

### 3.4.2. Khái niệm thực thể (entity)

Entity được hiểu là đối tượng đại diện cho các khái niệm hoặc thông tin nhất định như người, địa điểm, tổ chức, sản phẩm,... Các từ và cụm từ có tính chất tương tự nhau có thể gặp phải trong thông điệp hay văn bản của người dùng được tập hợp lại thành một entity.

Các entity giúp trích thông tin từ các đoạn văn bản bất kỳ một cách dễ dàng và chính xác hơn. Bất kỳ thông tin quan trọng nào cần phải thu thập từ người dùng, cũng nên cần phải có entity tương ứng cho nó.

Tiếp tục xét tiếp ví dụ trên với câu: “*Tôi cần biết điểm môn Cơ sở dữ liệu.*”

Giả định đã xây dựng một entity tên là *@ten\_mon\_hoc* chứa tên các môn học đã đang và sẽ được giảng dạy tại trường Đại học Khoa học Tự nhiên - ĐHQG TP.HCM (xem hình 3).

ten\_mon\_hoc

Tên entity

Hệ điều hành năng cao	Hệ điều hành năng cao, HDH năng cao, HĐH năng cao
Thực tập hệ thống viễn thông	Thực tập hệ thống viễn thông
Giải tích BI	Giải tích BI
An ninh mạng	An ninh mạng
Nhập môn công nghệ phần mềm	Nhập môn công nghệ phần mềm, NM CNPM, NM Công nghệ phần mềm
Mạng máy tính năng cao	Mạng máy tính năng cao, MMT năng cao
Lý thuyết mạch số	Lý thuyết mạch số

Cụm từ có tính chất tương tự nhau đó là "tên môn học"

Các cụm từ đồng nghĩa với các cụm từ nằm ở cột bên trái

Hình 3: Minh họa một entity

Như phân tích ở ví dụ phía trên, intent của câu “Tôi cần biết điểm môn Cơ sở dữ liệu.” là “*diem\_so*” và môn học cụ thể cần lấy điểm cho người dùng xem là “*Cơ sở dữ liệu*”. Hệ thống sẽ lấy được tên môn học thông qua tham số *ten\_mon\_hoc* được kết nối với entity *@ten\_mon\_hoc*.

Việc nhận dạng entity giúp thu thập được các thông tin chi tiết trong thông điệp của người dùng để thuận tiện cho việc xử lý và xây dựng câu trả lời.

### 3.4.3. Khái niệm ngữ cảnh (context)

Context là bối cảnh hiện tại của cuộc hội thoại khi yêu cầu của người dùng được đưa ra. Khi một hệ thống chatbot thực hiện việc phân tích câu truy vấn đầu vào của người dùng, nó không nhất thiết phải dựa vào lịch sử của các câu hội thoại xuất hiện trước nó. Với ngữ cảnh được cung cấp, các chatbot có thể dễ dàng tìm hiểu ý định của câu hỏi hiện tại. Ngữ cảnh hữu ích trong việc phân biệt các cụm từ có ý nghĩa mơ hồ hoặc có ý nghĩa khác nhau tùy thuộc vào các tùy chọn của người dùng. Ngữ cảnh cũng có thể là những biến được xác định giá trị trước như thông tin về địa điểm (được xác định thông qua GPS), thông tin về hồ sơ cá nhân (nếu được phép truy cập),...

Ví dụ 1: Xét hai câu sau đây và giả sử hệ thống được phép truy cập thông tin hồ sơ cá nhân của người dùng.

*Người dùng:* “Hi”

*Chatbot:* “Hi, tôi có thể giúp gì được cho bạn?”

*Người dùng:* “Tôi cần biết điểm môn Cơ sở dữ liệu.” (1)

*Chatbot:* “Sau đây là điểm môn Cơ sở dữ liệu: 9”

Như đã phân tích ở hai ví dụ trong trong khái niệm ý định và khái niệm thực thể, intent của câu “*Tôi muốn biết điểm môn Cơ sở dữ liệu*” là “diem\_so”, entity cần thiết là “@ten\_mon\_hoc”. Để thực hiện yêu cầu (1), hệ thống cần biết mã số sinh viên của sinh viên. Tuy nhiên, hệ thống được phép truy cập hồ sơ cá nhân của người dùng và đã biết mã số sinh viên của người dùng nên không yêu cầu người dùng cung cấp lại nữa.

Ví dụ 2: Xét hai câu sau:

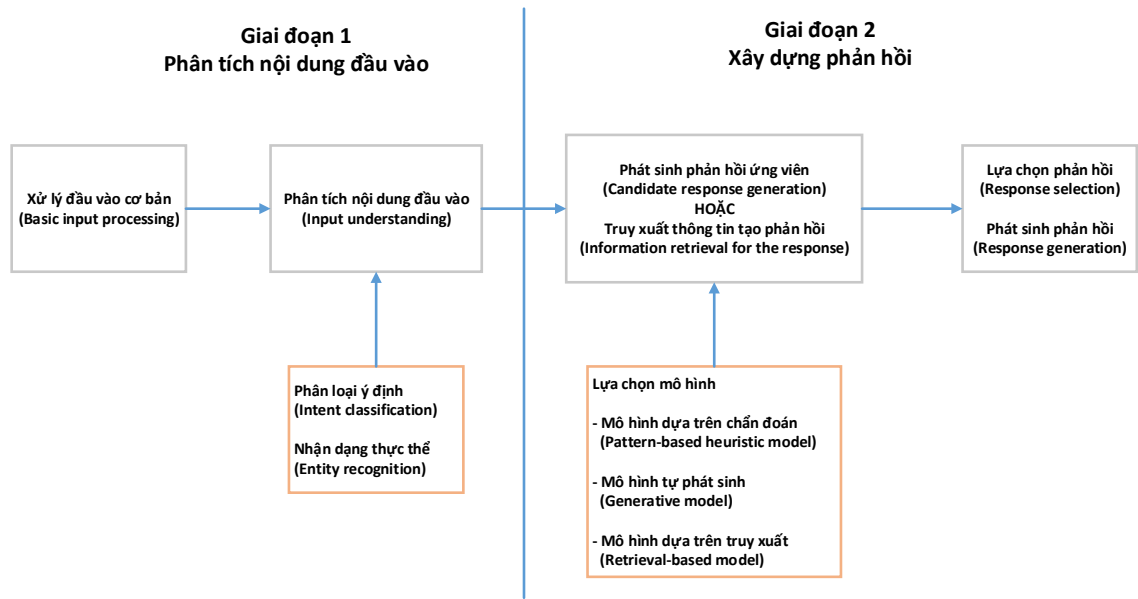
*Câu 1:* “Cầu thủ đá bóng vợt xà ngang.”

*Câu 2:* “Tôi cần bỏ thêm đá vào ly cà phê của tôi.”

Đây là hai câu với ý nghĩa hoàn toàn khác nhau. Nhưng trong câu 1, từ “đá” là một động từ, thể hiện một hành động trong khi từ “đá” ở câu 2 là một danh từ, chỉ một loại phụ gia bỏ thêm vào thức uống để tạo cảm giác lạnh. Khi không có ngữ cảnh để xác định sẽ rất khó khăn để hiểu nghĩa thích hợp của chúng.

Có thể tóm tắt các khái niệm về xử lý ngôn ngữ tự nhiên trong việc xây dựng chatbot như sau: các intent giúp hiểu được nội dung chủ đạo của thông điệp, entity giúp trích xuất những thông tin cần thiết và context hữu ích trong việc xác định ngữ cảnh của cuộc trò chuyện. Phần tiếp sau sẽ trình bày nguyên tắc vận hành của chatbot.

#### **4. Nguyên tắc vận hành của chatbot**



Hình 4: Kiến trúc chung của chatbot

Hình 4 mô tả kiến trúc chung của một chatbot. Về cơ bản, công việc xây dựng một chatbot được phân biệt thành hai nhiệm vụ chính đó là:

- *Phân tích nội dung đầu vào*: có thể có nhiều mô hình để xử lý yêu cầu, tuy nhiên, đối với các chatbot phức tạp thì cần phải ứng dụng kỹ thuật xử lý ngôn ngữ tự nhiên để có thể "hiểu" nội dung mà người dùng đang nhắc đến.
- *Xây dựng phản hồi dựa trên phát biểu của người dùng*: dựa trên mô hình hoạt động của chatbot mà ta có thể chia làm 2 hướng tiếp cận trong việc tạo phản hồi: dựa theo kịch bản có sẵn và dựa trên dữ liệu được trả về sau bước phân tích nội dung đầu vào.

#### 4.1. Phân tích nội dung đầu vào

Việc phân tích nội dung đầu vào có nghĩa là trích xuất nội dung từ thông điệp người dùng. Hai module chính trong việc xử lý thông điệp người dùng là: phân loại ý định và nhận dạng thực thể.

- Mô hình phân loại ý định (intent classification module): sẽ xác định intent từ thông điệp của người dùng.

- Mô hình nhận dạng thực thể xác định (named-entity recognition module): giúp rút trích tập các entity có trong thông điệp của người dùng.

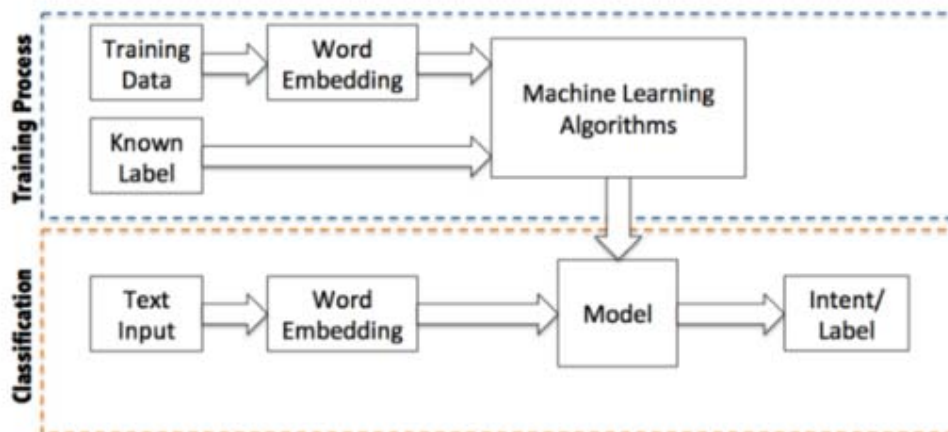
#### 4.1.1. Phân loại ý định (intent classification)

Một trong những nhiệm vụ chính của chatbot là phân loại ý định [7].

Dữ liệu đầu vào được xác định bởi một trình phân loại, trình phân loại này giúp liên kết thông tin được cung cấp với một intent cụ thể, đưa ra giải thích chi tiết về các từ để máy tính hiểu được. Trình phân loại này sẽ giúp phân loại các thành phần dữ liệu – trong trường hợp này là một câu – vào nhiều danh mục khác nhau. Có nhiều cách tiếp cận cho các nhà phát triển làm điều này:

- So khớp mẫu (pattern matching): liên quan đến việc sử dụng các biểu thức chính quy (regex) để tìm và phân loại các mẫu.
- Sử dụng các thuật toán máy học: sử dụng một hoặc kết hợp một vài các thuật toán máy học khác nhau để phân loại các mẫu.

Luận văn sử dụng công cụ hỗ trợ xử lý ngôn ngữ tự nhiên Dialogflow để hỗ trợ trong việc xử lý ngôn ngữ tự nhiên để phân tích nội dung đầu vào. Trình phân loại của công cụ này áp dụng các thuật toán máy học có giám sát cho việc phân loại các câu vào các danh mục khác nhau. Hình 5 minh họa cách phân loại sử dụng các thuật toán máy học có giám sát (đọc thêm tại [8])



*Hình 5: Phân loại văn bản bằng các thuật toán học có giám sát*

Quá trình phân loại intent dựa vào các thuật toán máy học có giám sát diễn ra theo hai giai đoạn:

- **Giai đoạn 1:** huấn luyện.
  - Dữ liệu dùng để huấn luyện phải được số hóa để máy có thể hiểu và hoạt động. Quá trình này gọi là quá trình nhúng từ (word embedding), được xây dựng dựa trên mô hình không gian vector (vector space model) – một mô hình cung cấp cách để biểu diễn các câu mà người dùng nhập vào thành các vector toán học một hoặc nhiều chiều. Việc thể hiện các từ dưới dạng vector toán học ngoài việc để có thể so sánh và tính toán còn giúp phát hiện ra mối quan hệ giữa các từ với nhau. Có rất nhiều cách để chuyển câu mà người dùng nhập vào thành các vector như: Average of Word2Vec, Weight Average of Word2Vec from TF-IDF, Doc2Vec, LSTM. Mỗi cách đều có ưu và nhược điểm của nó. Do đó, chọn cách nào tùy thuộc vào nhiệm vụ cần thực hiện với vector.
  - Tiếp theo, sử dụng các thuật toán máy học tính toán trên các vector tạo ra, phân lớp và tạo ra một mô hình. Mô hình này được sử dụng để phân loại dữ liệu cho các mẫu về sau. Một số các thuật toán máy học có thể được áp dụng: Naïve Bayes, Neural Network, Support Vector Machine,...
- **Giai đoạn 2:** phân loại
  - Dữ liệu vẫn sẽ trải qua quá trình số hóa để chuyển thành các vector giống như quá trình huấn luyện.
  - Kết quả từ quá trình nhúng từ sẽ đi qua mô hình được xây dựng ở giai đoạn huấn luyện, tính toán và đưa ra kết quả phân loại.

Phía trên đã trình bày tổng quan về quá trình phân loại dựa trên các thuật toán máy học có giám sát hoạt động. Ngoài ra còn có các cách phân loại khác dựa trên việc so khớp mẫu và các thuật toán máy học không giám sát. Tuy nhiên việc phân



loại dựa trên so khớp mẫu quá cứng nhắc còn việc phân loại dựa trên các thuật toán máy học không giám sát đòi hỏi nhiều kiến thức, đặc biệt là trong việc thiết kế quy trình huấn luyện. Phần tiếp theo sẽ trình bày một nhiệm vụ khác cực kì quan trọng của chatbot chính là nhận dạng thực thể.

#### 4.1.2. Nhận dạng thực thể (named-entity recognition)

Nhận dạng thực thể (còn gọi là khai thác thực thể) là một kỹ thuật giúp xác định các thực thể trong văn bản thành các danh mục được xác định trước như tên người, tổ chức, địa điểm. Việc xác định thực thể trong văn bản được thực hiện thông qua các chú thích (annotation) và danh sách các thể đã được phân loại (categoration tags) để trích xuất thông tin. Trong chatbot, hệ thống nhận dạng thực thể được dùng để phát hiện các entity. Xét ví dụ sau:

“Thầy **Cương** tên người sẽ đi **Hà Nội** địa điểm để dự hội thảo do trường **Đại học Khoa Học Tự Nhiên Hà Nội** tổ chức tổ chức vào vào **tháng 10** thời gian.”

Bảng 1 thể hiện việc trích xuất thông tin trong văn bản thành các danh mục được xác định trước.

Tên người	Cương
Địa điểm	Hà Nội
Tổ chức	Đại học Khoa Học Tự Nhiên Hà Nội
Thời gian	Tháng 10

Bảng 1: Thông tin được trích xuất từ ví dụ

Theo [9], các hệ thống nhận dạng thực thể có thể sử dụng một số phương pháp khác nhau để nhận dạng thực thể

- Regex extraction: phù hợp với việc trích xuất các thông tin liên quan đến số như: số điện thoại, số chứng minh nhân dân, địa chỉ email, số, URL, hashtag, số thẻ tín dụng,...

- Dictionary extraction: sử dụng từ điển các chuỗi và thông báo khi nào các chuỗi đó xuất hiện trong văn bản. Phương pháp này phù hợp để nhận dạng màu sắc, đơn vị, kích thước, sản phẩm, tên thương hiệu, tên thuốc,...
- Complex pattern-based extraction: phù hợp để trích xuất tên người, tên doanh nghiệp và trích xuất thông tin dựa trên ngữ cảnh.
- Statistical extraction: phương pháp sử dụng thống kê, phù hợp để trích xuất tên người, tên công ty, thực thể địa lý hoặc những thực thể chưa từng xuất hiện trước đó và những thực thể bên trong văn bản có cấu trúc rõ ràng như văn bản học thuật hoặc báo chí.

Một số mô hình mốc sử dụng kỹ thuật học sâu (deep learning) để nhận dạng đối tượng: CRF Classifier, Maximum Entropy Approach, FeedForward Neural Network, BILSTM CNNs, Multi Task Learning, Conditional Random Field, ... xem thêm tại [10]

#### **4.2. Xây dựng phản hồi**

Xây dựng phản hồi là một trong 2 chức năng chính và cơ bản nhất của chatbot. Tạo phản hồi là quá trình tạo ra câu trả lời cho người dùng sau khi phân tích và xử lý thông điệp đầu vào dựa trên các mẫu hoặc thuật toán máy học.

Thông thường, tùy vào mục đích sử dụng của chatbot mà cách thức phát sinh câu phản hồi có sự khác nhau, do đó mỗi một nhóm thông điệp khác nhau được người sử dụng nhập vào cần có một mô hình phản hồi chuyên biệt. Câu trả lời tĩnh là cách trả lời đơn giản nhất thường được sử dụng cho những câu hỏi đơn giản với câu trả lời cố định hoặc có tính lặp đi lặp lại và không có yêu cầu xử lý phức tạp. Tuy nhiên, với các chatbot phức tạp hơn, việc tạo phản hồi sẽ dựa trên các thông tin về intent, entity, context hay các tham số cần thiết và có thể thực hiện bằng nhiều cách: gọi các API bên ngoài để hỗ trợ, sử dụng thuật toán để tìm câu trả lời hay thậm chí nhờ sự can thiệp của con người trong một số tình huống nhất định.

Ví dụ một chatbot truy vấn thông tin về thời tiết sẽ sử dụng các API của các ứng dụng thời tiết để tạo câu trả lời. Trong khi đó, chatbot truy vấn thông tin trong lĩnh vực thương mại điện tử có thể cần phải tìm kiếm câu trả lời từ cơ sở dữ liệu và sự trợ giúp của con người.

Hai nhiệm vụ phân tích nội dung đầu vào và xây dựng phản hồi là hai vụ riêng biệt nhưng phụ thuộc lẫn nhau và chính hai nhiệm vụ này tạo ra các mô hình khác nhau để xây dựng chatbot. Phần tiếp sau đây sẽ trình bày các mô hình phổ biến được dùng để xây dựng chatbot.

#### **4.3. Các mô hình được dùng để xây dựng chatbot**

Việc xử lý được thông điệp từ người dùng và tạo phản hồi thường áp dụng theo một trong ba mô hình sau [11]: mô hình dựa trên chuẩn đoán (pattern-based heuristics), mô hình tự phát sinh (generative models) và mô hình dựa trên truy xuất (retrieval-based model).

##### **4.3.1. Mô hình dựa trên chuẩn đoán (pattern-based heuristic model)**

Ở mô hình này, việc dự đoán để lựa chọn một phản hồi có thể được thiết kế theo nhiều cách khác nhau, thường từ logic if-else, đến sử dụng biểu thức chính quy (regular expression) hoặc sử dụng một bộ quy tắc mà trong đó chứa các mẫu làm điều kiện cho các quy tắc. Ở hướng tiếp cận này, ngôn ngữ AIML và ChatScript được sử dụng rộng rãi nhất cho việc thiết kế mẫu và phản hồi.

AIML(Artificial Intelligence Mark-up Language) bắt nguồn từ XML cho phép người dùng nhập kiến thức cho những cuộc trò chuyện dành cho chatbot. AIML bắt đầu với một nút gốc là *<AIML>*. Tất cả các thành phần còn lại sẽ là con của nút gốc này. Đối tượng AIML được tạo thành từ các đơn vị được gọi là các danh mục (category) chứa các dữ liệu đã được phân tích cú pháp hoặc chưa được phân tích cú pháp.

Ví dụ: dưới đây là một đoạn kịch bản được viết bằng AIML.

```
<category>
<pattern>Bạn tên gì? </pattern>
<template>Tên đầy đủ của tôi là Hồ Thảo Hiền</template>
</category>
```

Nguyên tắc hoạt động của AIML là tất cả các dữ liệu được biên dịch trước vào một cấu trúc cây. Khi chatbot nhận được một câu, nó sẽ duyệt qua các danh mục (category) cho đến khi tìm thấy được khuôn mẫu (pattern) phù hợp với câu đầu vào của người dùng. Nếu tìm được khuôn mẫu khớp với câu của người dùng, nó sẽ dùng bản mẫu (template) tương ứng để tạo ra câu trả lời. Đọc thêm chi tiết về ngôn ngữ AIML tại mục 1 của phụ lục.

ChatScript là một ngôn ngữ kịch bản được thiết kế để chấp nhận câu người dùng nhập vào và tạo ra một phản hồi. Chương trình lấy một hoặc nhiều câu từ người dùng và xuất ra một hoặc nhiều câu trả lời trở lại. Đơn vị cơ bản của ChatScript là các quy tắc (rule). Một quy tắc đầy đủ bao gồm loại (kind), nhãn (label), khuôn mẫu (pattern) và đầu ra (output). Ngoài ra, ChatScript cũng gồm các thành phần khác: chủ đề (topic), khoảng trắng (white space) và chú thích (comment).

Ví dụ dưới đây là một rule đầy đủ trong ChatScript:

*?: HỌC (Bạn có thích học môn Toán không) Tôi thích học Toán lắm.*

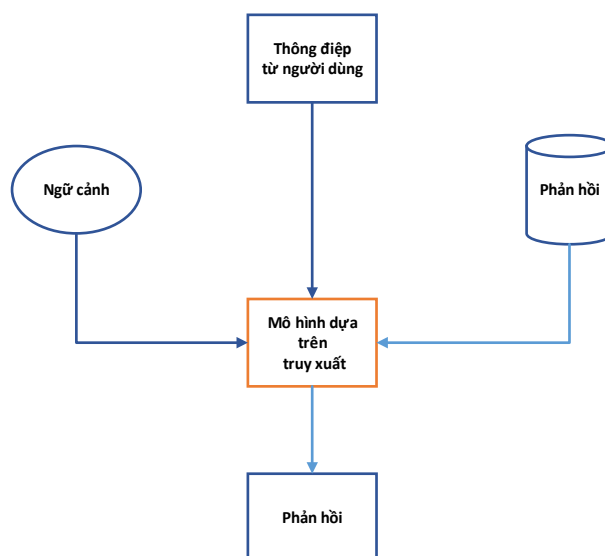
Trong ví dụ trên, loại quy tắc (rule kind) là ?: nghĩa là nó sẽ phản ứng với các câu có dạng là câu hỏi, nhãn (rule label) là HỌC, khuôn mẫu (rule pattern) là những từ được ghi trong dấu () và nó sẽ được dùng tìm kiếm cả cụm “Bạn có thích học toán không” trong câu đầu vào của người dùng, đầu ra (rule output) là “Tôi thích học toán lắm.”

Ngoài các quy tắc(rule), ChatScript còn có những thành phần khác. Đọc thêm chi tiết về các thành phần của ChatScript tại mục 2 của phụ lục.

Mô hình dựa trên truy xuất rất đơn giản và dễ tiếp cận, tuy nhiên nó cũng tồn tại nhiều hạn chế. Thứ nhất, việc xây dựng bộ dữ liệu được thực hiện thủ công, tốn nhiều thời gian. Thứ hai, nếu câu truy vấn của người dùng không nằm trong kịch bản, việc đưa ra câu trả lời có thể bị gián đoạn hoặc gây lỗi cho chương trình. Cuối cùng, vì không thể nhìn thấy tất cả các mẫu được cài đặt và quy tắc nào khớp với đầu vào nào nên khi bạn tham gia làm việc với một hệ thống sử dụng mô hình dựa trên chẩn đoán, việc viết mã nguồn, gỡ lỗi và bảo trì sẽ trở nên khó khăn hơn rất nhiều.

Tiếp theo sẽ trình bày một mô hình giúp chatbot xây dựng phản hồi một cách thông minh hơn so với mô hình dựa trên chẩn đoán.

#### 4.3.2. Mô hình dựa trên truy xuất (retrieval-based models)



Hình 6: Mô hình dựa trên truy xuất

Mô hình dựa trên truy xuất là mô hình sử dụng một kho lưu trữ các câu trả lời đã được xác định trước (predefined repository). Cách này sẽ giúp xây dựng chatbot dễ dàng hơn và câu trả lời có thể dự đoán được dựa trên quá trình thiết kế và lập trình. Đây là cách tiếp cận được nhiều nhà phát triển sử dụng vì cách tiếp cận này giúp chủ động hơn trong việc tạo ra phản hồi và tận dụng được các nguồn tài nguyên hay các API có sẵn. Chatbot sẽ sử dụng các kết quả phân tích từ phát biểu của người dùng

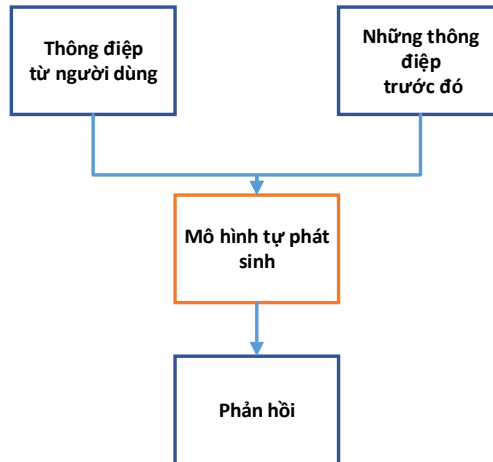
và ngữ cảnh có sẵn để chọn câu trả lời từ một danh sách các câu trả lời được xác định trước đó. Việc xây dựng mô hình này có thể rất đơn giản hoặc khá phức tạp tùy thuộc vào tình huống và giới hạn chức năng của chatbot.

Ví dụ: Giả sử, lập trình viên đã xây dựng trước câu trả lời cho mỗi intent. Trong đó, lập trình viên quy định rằng, khi intent là “*diem\_so*” sẽ thực hiện câu truy vấn “*select diem from diemsomonhoc where diemso = <ten\_mon\_hoc>*”. Khi người dùng gửi yêu cầu “*Tôi muốn xem điểm môn Cơ sở dữ liệu*”. Lập trình viên sẽ thay thế tham số tượng trưng là *<ten\_mon\_hoc>* trong câu truy vấn bằng từ khóa “*Cơ sở dữ liệu*” và thực hiện truy vấn trong cơ sở dữ liệu. Trong trường hợp câu trả lời không tồn tại trong cơ sở dữ liệu, lập trình viên có thể chuyển hướng sử dụng một API để tìm kiếm trên mạng một kết quả và trả về cho người dùng.

Mô hình này không phải xây dựng bộ dữ liệu một cách thủ công, tạo điều kiện dễ dàng hơn cho quá trình phát triển và bảo trì – cải thiện được những vấn đề vốn là nhược điểm của mô hình dựa trên chẩn đoán. Bên cạnh đó, mô hình này cũng giúp lập trình viên chủ động hơn cũng như có thể dự đoán được phản hồi được trả về cho người dùng. Tuy nhiên, nó vẫn chưa giải quyết được vấn đề khi câu truy vấn của người dùng nằm ngoài phạm vi được xác định trước.

Tiếp theo sẽ trình bày một mô hình giúp giải quyết vấn đề còn lại mà hai mô hình trên không giải quyết được. Nó là mô hình cấp cao, khiến chatbot trở nên thông minh hơn.

#### **4.3.3. Mô hình tự phát sinh (generative models)**



*Hình 7: Mô hình tự phát sinh*

Mô hình tự phát sinh là một loại mô hình không sử dụng bất kỳ kho lưu trữ câu trả lời được xác định trước. Mô hình này thường được xây dựng dựa theo mô hình xác suất hoặc các thuật toán máy học. Nó không có câu trả lời cố định và cần được huấn luyện để tạo ra một nội dung mới. Câu trả lời được tạo ra một cách tự động dựa trên thông điệp hiện tại và các thông điệp trước đó của người dùng.

Mô hình tự phát sinh được xem là mô hình tương lai của chatbot, nó giúp chatbot thông minh hơn, xử lý ngôn ngữ tự nhiên hơn. Trong trường hợp câu hỏi không nằm trong phạm vi được xác định trước, nó vẫn có thể đưa ra câu trả lời. Đây vốn là nhược điểm còn tồn đọng của hai mô hình trước.

Loại chatbot được xây dựng từ mô hình này thuộc mức nâng cao. Tuy nhiên, mô hình tự phát sinh gây khó khăn việc xây dựng và huấn luyện. Nghĩa là cần tập huấn luyện với kích thước rất lớn để có thể xây dựng một cuộc trò chuyện có chất lượng. Xây dựng chatobot với mô hình này khiến lập trình viên không thể dự đoán chắc chắn được câu trả lời sẽ như thế nào. Bên cạnh đó, việc phát sinh câu trả lời theo mô hình này khiến cho câu trả lời dễ mắc phải các lỗi ngữ pháp, đưa ra các câu trả lời chung chung, không liên quan đến câu hỏi. Vì những nhược điểm trên mà mô hình này ít được các lập trình viên lựa chọn để xây dựng chatbot trong thực tế

#### **4.4. Tổng kết và đánh giá các mô hình**

Phần này là tổng hợp về ưu và nhược về các mô hình được dùng để xây dựng phản hồi (bảng 2).

Mô hình	Mô hình dựa trên chẩn đoán	Mô hình dựa trên truy xuất	Mô hình tự phát sinh
Ưu điểm	Đơn giản, dễ tiếp cận.	Dễ dự đoán câu trả lời cho câu hỏi  Tận dụng được các nguồn lực có sẵn (các API bên ngoài)	Linh hoạt  Đảm bảo có câu trả lời dù nội dung câu truy vấn không được xác định trước
Khuyết điểm	Dữ liệu được lập trình thủ công.  Dự án xây theo mô hình này khó phát triển, bảo trì.  Không đảm bảo có câu trả lời khi nội dung câu truy vấn không được xác định trước.	Không đảm bảo có câu trả lời khi nội dung câu truy vấn không được xác định trước.	Khó xây dựng.  Cần tập dữ liệu huấn luyện lớn.  Không dự đoán được câu trả lời cho câu hỏi  Câu trả lời có xu hướng mắc lỗi ngữ pháp, chung chung hoặc không liên quan.

*Bảng 2: Bảng đánh giá các mô hình được dùng để xây dựng phản hồi cho chatbot*

Mô hình dựa trên chẩn đoán là một mô hình đơn giản, dễ tiếp cận. Tuy nhiên, vấn đề chính của mô hình dựa trên chẩn đoán là các mẫu cần phải được lập trình thủ công và đó không phải là công việc dễ dàng, đòi hỏi nhiều thời gian và công sức. Bên cạnh đó, chương trình xây dựng theo mô hình này khó phát triển, bảo trì và không



đảm bảo sẽ có câu trả lời cho câu truy vấn của người dùng nếu nó không nằm trong phạm vi được xác định trước.

Mô hình dựa trên truy xuất dù giải quyết được một số vấn đề của mô hình dựa trên truy xuất, nó giúp các lập trình viên dễ dàng phát triển, bảo trì dự án và dự đoán được câu trả lời cho mỗi câu truy vấn của người dùng. Tuy nhiên, nó vẫn chưa giải quyết được việc câu truy vấn của người dùng nằm ngoài phạm vi nội dung được xác định trước.

So sánh hai mô hình còn lại, mặc dù mô hình tự phát sinh linh hoạt hơn so với mô hình dựa trên truy xuất và mô hình dựa trên chẩn đoán nhưng nó lại không hoạt động tốt trong thực tế. Bởi vì mô hình này quá tự do trong việc sinh ra câu trả lời nên có xu hướng mắc lỗi ngữ pháp hoặc tạo ra các câu trả lời không liên quan, chung chung hoặc không phù hợp. Ngoài ra, mô hình tự phát sinh cần một lượng lớn dữ liệu để huấn luyện và rất khó để tối ưu hóa. Do đó, phần lớn các hệ thống hiện nay được các nhà phát triển dựa trên mô hình dựa trên truy xuất hoặc kết hợp mô hình dựa trên truy xuất và mô hình tự phát sinh.

Trong phần tiếp theo tiếp theo sẽ thực hiện tổng hợp, phân nhóm và so sánh một số các nền tảng, công nghệ phổ biến được dùng để xây dựng chatbot hiện nay.

## 5. Tổng hợp, so sánh một số nền tảng, công nghệ sử dụng xây dựng chatbot

Phần này sẽ thực hiện phân nhóm các nền tảng, công nghệ được dùng để xây dựng chatbot. Tiếp đến sẽ so sánh ra các nền tảng, công nghệ lớn được hỗ trợ bởi các tập đoàn lớn là Google, Facebook, Microsoft dựa trên các tiêu chí được đưa ra. Cuối cùng sẽ đưa ra kết luận về các nền tảng được thử nghiệm.

### 5.1. Các nhóm nền tảng, công nghệ xây dựng chatbot

Có thể chia các nền tảng, công nghệ xây dựng chatbot thành ba nhóm chính như trong bảng 3:

Không cần lập trình	Công cụ xử lý ngôn ngữ tự nhiên (NLP Engine)	Môi trường tích hợp
Chatfuel	Dialogflow (Google)	Recast
Manychat	Wit (Facebook)	Morph
FlowXO	Pandorabot	Meya
Octane	IBM Watson	Microsoft Bot Framework
Recime	...	...
...		

*Bảng 3: Phân loại các nền tảng xây dựng chatbot.*

- **Nhóm 1:** nhóm nền tảng, công nghệ dành cho người không biết lập trình. Nhóm các công cụ này ngày càng nhiều, giúp chúng ta có thể tạo được những chatbot đơn giản, không có cấu trúc quá phức tạp một cách nhanh chóng. Các nền tảng này cũng hỗ trợ kết nối với các nền tảng ứng dụng khác như Facebook Messenger, Slack, Skype,... một cách nhanh chóng và tiện lợi. Tuy nhiên nhược điểm của các công cụ này là không thể tùy biến câu trả lời như các công cụ ở nhóm 2 và nhóm 3 vì không cần phải lập trình.

- **Nhóm 2:** *nhóm những công cụ xử lý ngôn ngữ tự nhiên.* Đây là nhóm thuần là các công cụ cho phép xây dựng chatbot bằng cách huấn luyện các thông điệp từ người dùng. Ở nhóm công cụ này, chúng ta có thể xây dựng được những chatbot thông minh hơn so với các công cụ ở nhóm 1. Tuy nhiên, đây đơn thuần chỉ là các công cụ hỗ trợ xử lý ngôn ngữ tự nhiên hỗ trợ phân tích nội dung đầu vào của người dùng thành một dạng dữ liệu có cấu trúc để các lập trình viên có thể phân tích xây dựng câu trả lời.
- **Nhóm 3:** *môi trường tích hợp để xây dựng chatbot, bao gồm cả các công cụ xử lý ngôn ngữ tự nhiên, dành cho người có kỹ năng lập trình.* Đây là các môi trường tích hợp, ngoài việc cho phép xây dựng chatbot, phân tích nội dung đầu vào bằng các công cụ xử lý ngôn ngữ tự nhiên, nó còn cho phép kết nối, kiểm tra, triển khai và quản lý bot, tất cả trong cùng một nơi.

Tùy vào nhu cầu và điều kiện khác nhau mà các nhà phát triển sẽ chọn cho mình những nền tảng khác nhau để xây dựng, cài đặt và triển khai chatbot.

## 5.2. So sánh các nhóm nền tảng, công nghệ phổ biến dùng để xây dựng chatbot

Trong quá trình thực hiện đề tài, một số tiêu chí được đặt ra để chọn công cụ và nền tảng để thử nghiệm. Các tiêu chí cụ thể như sau:

- Phải cho phép lập trình và những ngôn ngữ mà nền tảng này hỗ trợ.
- Công cụ hay nền tảng đó phải miễn phí hoặc là mã nguồn mở.
- Tài liệu phải được hỗ trợ đủ tốt để có thể thực hiện theo.
- Khả năng xử lý ngôn ngữ tự nhiên mà công cụ, nền tảng cung cấp.
- Khả năng triển khai ứng dụng.

Dựa vào các tiêu chí trên chọn ba nền tảng lớn để thử nghiệm, với sự hỗ trợ của ba tập đoàn khổng lồ là:

- Microsoft Bot Framework và LUIS (công cụ xử lý ngôn ngữ tự nhiên của Microsoft Framework) của Microsoft,

- Wit.ai được hỗ trợ bởi Facebook
- Dialogflow sở hữu bởi Google.

Sau khi thử nghiệm, tôi rút ra được một số kết luận. Những kết luận này có thể đúng ở hiện tại và tùy vào khả năng của các công ty trong tương lai mà các nền tảng này có thể phát triển theo hướng khác.

Tiêu chí so sánh	Microsoft Bot Framework (Microsoft)	Dialogflow (Google)	Wit (Facebook)
Ngôn ngữ lập trình	NodeJS, Python, Java, Go, Ruby, C#, PHP	.NET, NodeJS, Python	NodeJS, Python, Ruby
Bản quyền	Mã nguồn mở	Miễn phí	Miễn phí
Tài liệu	Rất tốt.  Tài liệu sử dụng Luis và tài liệu hướng dẫn cài đặt rất đầy đủ. Phù hợp cho người mới bắt đầu tập xây dựng chatbot.	Tốt.  Tài liệu hướng dẫn sử dụng rất rõ ràng, tuy nhiên tài liệu về thiết lập cài đặt không được triển khai rõ ràng. Tuy nhiên có nhiều tài liệu hướng dẫn có sẵn trên mạng.	Chưa tốt.  Tài liệu hướng dẫn sử dụng chưa rõ ràng và tài liệu về cài đặt không được triển khai. Tuy nhiên có nhiều tài liệu hướng dẫn có sẵn trên mạng.

Công cụ xử lý ngôn ngữ tự nhiên	<p>Khá tốt.</p> <p>Giao diện của Luis (công cụ xử lý ngôn ngữ tự nhiên của Microsoft Bot Framework) thân thiện, tuy nhiên còn nhiều hạn chế như việc thiết lập môi trường khó khăn, giá cả dịch vụ đi kèm cao.</p>	<p>Tốt.</p> <p>Giao diện của Dialogflow thân thiện, hỗ trợ rất mạnh mẽ trong việc xây dựng câu trả lời.</p>	<p>Khá.</p> <p>Khá, giao diện ít trực quan.</p>
Khả năng triển khai ứng dụng	<p>Rất dễ</p> <p>Dễ để triển khai ứng dụng đầu tiên nhờ được mô tả rõ ràng trong tài liệu.</p>	<p>Khá khó.</p> <p>Khá khó khăn để triển khai ứng dụng và tùy biến chatbot vì điều này không được đề cập trong tài liệu.</p>	<p>Khá khó.</p> <p>Khá khó khăn để triển khai ứng dụng và tùy biến chatbot vì điều này không được viết trong tài liệu.</p>

*Bảng 4: So sánh đặc điểm của ba nền tảng lớn được dùng để xây dựng chatbot*

Trong ba nền tảng trên, Microsoft Bot Framework là nền tảng tốt nhất và thích hợp nhất cho người mới bắt đầu tập làm quen với việc xây dựng chatbot. Tuy nhiên, về sau nền tảng này có một số khó khăn trong việc đăng kí cùng với chi phí đắt đỏ cho các dịch vụ đi kèm.

Bên cạnh đó, Wit là một công cụ xử lý ngôn ngữ tự nhiên, tuy hỗ trợ được nhiều ngôn ngữ hơn so với Microsoft Bot Framework và Dialogflow nhưng về tài liệu, khả năng hỗ trợ huấn luyện và thời gian xử lý không bằng hai nền tảng còn lại.

Trong khi đó, Dialogflow là một nền tảng trung hòa được điểm mạnh của Microsoft Bot Framework và Wit. Về tài liệu, tuy không hỗ trợ tốt bằng Microsoft Bot Framework nhưng lại hơn hẳn Wit. Cùng với đó, nền tảng cung cấp khả năng hỗ trợ xây dựng phản hồi tốt và không gây khó khăn cho người dùng khi sử dụng các dịch vụ, công cụ đi kèm như Microsoft Bot Framework. Do đó, trong luận văn này, khi chọn nền tảng để xây dựng chatbot eBot hỗ trợ học tập, tôi chọn Dialogflow để hỗ trợ cho quá trình xử lý ngôn ngữ tự nhiên và phân tích yêu cầu người dùng.

Chương sau của luận văn sẽ trình bày các giải pháp cho việc xây dựng ứng dụng chatbot hỗ trợ học tập.

### CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ GIẢI PHÁP CHO ỨNG DỤNG CHATBOT HỖ TRỢ HỌC TẬP

*Trong chương này, nội dung sẽ trình bày tổng quan về các hướng xây dựng chatbot trong lĩnh vực giáo dục hiện nay. Qua đó xác định bài toán, ngữ cảnh, phạm vi và mục tiêu của ứng dụng. Tiếp đó sẽ trình bày kết quả của cuộc khảo sát để lấy yêu cầu người dùng để xây dựng tính năng cần có cho ứng dụng. Cuối cùng phân tích các giải pháp được sử dụng để xây dựng ứng dụng.*

#### 1. Tổng quan về các hướng xây dựng chatbot trong lĩnh vực giáo dục

Nội dung phần này sẽ trình bày những quan tâm mà đối với chatbot trong tương lai đối với lĩnh vực giáo dục và thống kê các xu hướng chính về lĩnh vực giáo dục mà chatbot đang tập trung giải quyết.

Đầu tiên sẽ nói về tiềm năng trong tương lai của chatbot trong lĩnh vực hỗ trợ học tập. Trong tương lai, chatbot có thể học một cách tự động nhờ khả năng máy học từ hàng trăm nguồn khác nhau, bao gồm cả các kiến thức của các trường đại học hàng đầu, tạo ra nhiều trải nghiệm học tập cho sinh viên trên toàn thế giới.

Khả năng ứng dụng chatbot trong các lĩnh vực của giáo dục rất lớn. Ngày nay học sinh, sinh viên dành nhiều thời gian hơn cho các ứng dụng tin nhắn trực tuyến và mạng xã hội. Đây chỉ có thể là vấn đề thời gian cho đến thời điểm mà mọi hoạt động hỗ trợ giảng dạy, trợ giảng hoặc sự phản hồi các bài luận được thực hiện bởi chatbot.

Theo [12], dưới đây là xu hướng chính về học tập mà chatbot đang tập trung giải quyết:

- *Chấm bài tự động*: phản hồi, nhận xét bài luận của mỗi cá nhân thường đòi hỏi nhiều thời gian, bằng cách dùng thuật toán máy học để học từ hàng ngàn bài luận khác nhau, hệ thống sẽ đưa ra những phản hồi có thể thay thế cho phản hồi của con người.

- *Giúp lặp lại các bài học cũ (space interval learning):* lặp lại bài học cũ ngay khi sắp quên nó là một cách học tối ưu. Vì vậy nên tập trung vào việc duy trì tri thức lâu dài bằng cách sử dụng phương pháp lặp lại kiến thức được học sau một khoảng thời gian hợp lí.
- *Hỗ trợ đánh giá môn học của sinh viên:* những đánh giá của sinh viên trong việc giảng dạy thường là nguồn thông tin có giá trị, rõ ràng và chúng cần được cải tiến. Sử dụng chatbot để thu thập các ý kiến phản hồi thông qua giao diện tin nhắn hội thoại. Cuộc trò chuyện và các câu hỏi tiếp theo có thể được điều chỉnh theo phản hồi và tính cách của người dùng.
- *Trợ giảng:* giúp trả lời những thắc mắc của sinh viên về môn học một cách nhanh chóng và chính xác. Sử dụng chatbot để trả lời các câu hỏi phổ biến của sinh viên về một môn học là một cách tiếp cận tốt.
- *Hỗ trợ giúp đỡ thông tin mà sinh viên cần biết khi học đại học:* giúp hỗ trợ sinh viên các thông tin trong môi trường đại học như cách nộp bài tập, cách đăng kí lớp cho học kì kế tiếp,... Đối với mỗi thế hệ sinh viên vào đại học, họ thường có một số câu hỏi tương tự nhau qua các năm. Hệ thống sẽ giúp giảm tải gánh nặng cho một số bộ phận của nhà trường khi họ không còn phải trả lời, giải thích những điều tương tự cho các sinh viên khác nhau.
- *Lắng nghe phản hồi của sinh viên:* mỗi người có tốc độ học và niềm yêu thích riêng đối với các lĩnh vực khác nhau. Cách tiếp cận đối với cùng một tốc độ và một loại giáo trình đối với những người học khác nhau không còn phù hợp. Một hệ thống lấy học sinh làm trung tâm, nơi cá tính và sở thích của học sinh là yếu tố quyết định khi nói đến cấu tạo của chương trình học rất quan trọng. Vì vậy lắng nghe phản hồi sẽ thực sự hiểu được những phản hồi của sinh viên về chương trình học để có thể cung cấp những chương trình học phù hợp với tốc độ và khả năng của từng người học riêng biệt. Bằng cách này, cả người học nhanh và học chậm đều có thể tiếp tục học với tốc độ của mình mà không bị nản lòng bởi các sinh viên khác.



Luận văn tập trung vào xu hướng *Hỗ trợ giúp đỡ thông tin mà sinh viên cần biết khi học đại học* để xây dựng chatbot eBot. Đầu tiên, ứng dụng theo xu hướng này tương đối dễ xây dựng nhưng mang lại hiệu quả cao đối với các trường học, nó giúp giảm tải gánh nặng cho một số bộ phận của nhà trường và tạo điều kiện cho học sinh, sinh viên tiếp cận với thông tin nhanh hơn. Thứ hai, việc xây dựng các ứng dụng chatbot phức tạp như: chatbot hỗ trợ chấm điểm tự động, chatbot trợ giảng,... cần dữ liệu lớn, quá trình nghiên cứu dài hạn, các vấn đề về kinh phí,... Để xây dựng được những ứng dụng chatbot như vậy cần có sự tập trung nguồn lực và tài chính nhất định. Thứ ba, thời gian và nguồn lực có hạn. Do đó thực hiện một ứng dụng hỗ trợ, giúp đỡ thông tin mà sinh viên cần biết khi học đại học là một quyết định hợp lý.

## **2. Tổng quan về ứng dụng**

Như đã trình bày phía trên, ứng dụng eBot được xây dựng nhằm mục đích hỗ trợ, và cung cấp những thông tin mà sinh viên cần biết khi học đại học. Thay vì phải xử lý một cuộc đối thoại dài, phạm vi của luận văn chỉ tập trung vào một công việc là trả lời một ngắn gọn câu hỏi cuối cùng của người dùng về một số vấn đề học tập và hỗ trợ người dùng thực hiện một số hành động đơn giản.

Các bước thiết kế giải pháp để xây dựng ứng dụng eBot:

- Thực hiện khảo sát lấy yêu cầu người dùng.
- Tổng hợp các tính năng cần có của ứng dụng dựa trên kết quả khảo sát.
- Sau khi đã xác định các tính năng cần có, xây dựng giải pháp để tạo phản hồi cho ứng dụng.

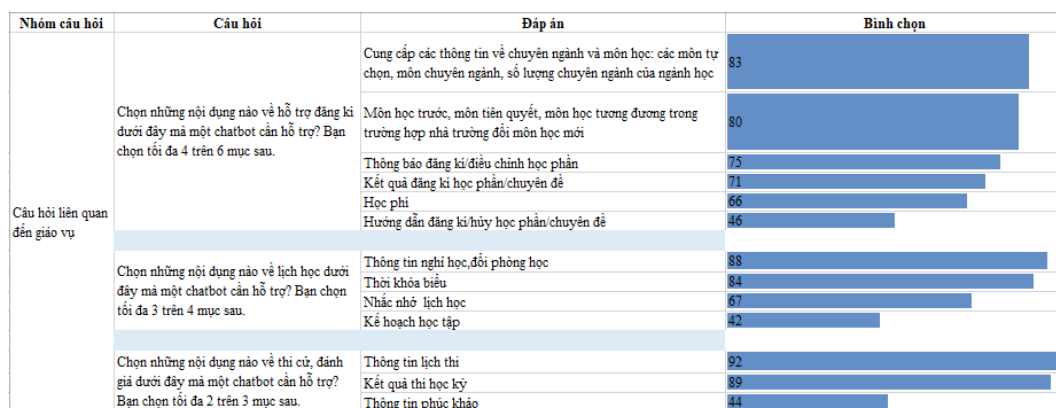
Sau đây mỗi phần sẽ trình bày tương ứng với các bước được nêu phía trên.

## **3. Yêu cầu và ngữ cảnh của ứng dụng**

Một cuộc khảo sát đã được thực hiện để lấy yêu cầu cho ứng dụng chatbot eBot. Bản khảo sát chia ra làm 3 nhóm câu hỏi. Ở mỗi câu hỏi yêu cầu người dùng chọn thông tin mà họ nghĩ mình cần được hỗ trợ. Đối với các mỗi câu hỏi chỉ yêu cầu người dùng chọn giới hạn một số lựa chọn được đưa ra, không được phép chọn tất

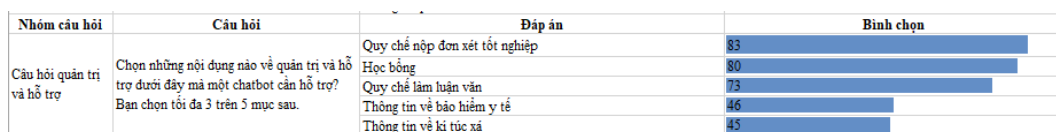
cả. Thực hiện khảo sát đối với hơn 100 bạn sinh viên trường đại học Khoa học tự nhiên TP.HCM. Trong số các bạn sinh viên tham gia khảo sát, có hơn 70% các bạn sinh viên học năm cuối. Dưới đây là chi tiết các nhóm câu hỏi và kết quả bình chọn.

- **Nhóm 1:** hỏi người dùng về những thông tin mà họ cần hỗ trợ khi đăng ký, những điều họ quan tâm về lịch học, thi cử và đánh giá. Danh sách các câu hỏi thuộc nhóm 1 và kết quả khảo sát được thể hiện trong hình 8.



Hình 8: Danh sách câu hỏi liên quan đến đến giáo vụ và kết quả bình chọn

- **Nhóm 2:** hỏi người dùng những nội dung cần được hỗ trợ về quản trị và hỗ trợ như: quy chế nộp đơn xét tốt nghiệp, học bổng, quy chế làm luận văn, thông báo về bảo hiểm y tế và thông tin về ký túc xá. Danh sách các câu hỏi thuộc nhóm 2 và kết quả khảo sát được thể hiện trong hình 9.



Hình 9: Danh sách câu hỏi liên quan đến đến quản trị, hỗ trợ và kết quả bình chọn

- **Nhóm 3:** hỏi người dùng những nội dung cần được hỗ trợ về nội dung học tập và hoạt động học tập. Danh sách các câu hỏi thuộc nhóm 3 và kết quả khảo sát được thể hiện trong hình 10.

Nhóm câu hỏi	Câu hỏi	Đáp án	Bình chọn
Câu hỏi liên quan đến nội dung học tập	Chọn những nội dung nào về trợ giúp tìm kiếm nội dung môn học dưới đây mà một chatbot cần hỗ trợ? Bạn chọn tối đa 3 trên 4 mục sau.	Hỗ trợ tìm kiếm tài nguyên môn học	86
		Nội dung chính của môn học	85
		Tìm nội dung buổi học tiếp theo để tham khảo	64
		Trả lời đơn giản một khái niệm môn học	43
	Chọn những nội dung nào về trợ giúp hoạt động học tập dưới đây mà một chatbot cần hỗ trợ? Bạn chọn tối đa 4 trên 6 mục sau.	Thông báo có deadline	83
		Hỗ trợ tìm kiếm nội dung có liên quan đến bài tập	84
		Bài tập	74
		Điểm và đánh giá môn học	66
		Nhắc nhở làm bài tập dựa trên hạn deadline	53
		Hỗ trợ nộp bài tập	41

*Hình 10: Danh sách câu hỏi liên quan đến nội dung môn học và kết quả bình chọn*

Từ kết quả khảo sát có thể rút ra những kết luận như sau:

- Nhu cầu được hỗ trợ về thông tin học tập của mỗi bạn sinh viên ở từng giai đoạn khác nhau là khác nhau. Ví dụ: sinh viên năm nhất và năm hai sẽ quan tâm nhiều hơn về các vấn đề như thông tin về kí túc xá, bảo hiểm y tế,... trong khi sinh viên năm ba và năm tư sẽ quan tâm hơn về các quy chế làm luận văn hay cách làm đơn xét tốt nghiệp,... Do đó kết quả này sẽ có thể khác nhau nếu tỉ lệ sinh viên tham gia khác nhau.
- Với kết quả khảo sát mà số lượng sinh viên tham gia chủ yếu là sinh viên năm ba và năm tư (năm ba 9,7% và năm tư 71,8%) thì chỉ có thể đưa ra kết luận về nhu cầu đối với nhóm sinh viên này.

Đối với mỗi câu hỏi, chọn trong mỗi nhóm các nội dung cần hỗ trợ có tỉ lệ bình chọn cao hơn 80%. Các nội dung cần được hỗ trợ được liệt kê bên trong bảng 5.

Nhóm nội dung	Yêu cầu cần hỗ trợ
Nhóm nội dung các yêu cầu cần hỗ trợ liên quan đến giáo vụ	Cung cấp các thông tin về chuyên ngành và môn học: các môn tự chọn, môn chuyên ngành, số lượng chuyên ngành của ngành học.

	Môn học trước, môn tiên quyết, môn học tương đương trong trường hợp nhà trường đổi môn học mới
	Thời khóa biểu
	Thông tin lịch thi
	Kết quả thi học kì
Nhóm nội dung các yêu cầu cần hỗ trợ về quản trị và hỗ trợ	Quy chế nộp đơn xét tốt nghiệp
	Học bổng
Nhóm nội dung cần hỗ trợ về nội dung học tập	Hỗ trợ tìm kiếm tài nguyên môn học
	Nội dung chính của môn học
	Hỗ trợ tìm kiếm nội dung có liên quan đến bài tập

*Bảng 5: Bảng danh sách các yêu cầu hỗ trợ có bình chọn trên 80%*

Các yêu cầu được hỗ trợ trong chatbot eBot sẽ được xây dựng dựa trên kết quả khảo sát. Tuy nhiên các tính năng này được xây dựng dựa trên tính chất của yêu cầu được đưa vào và cách mà chatbot eBot phản hồi yêu cầu. Do đó, các nội dung này cần phải được phân chia lại để phù hợp với từng tính năng riêng biệt. Bên cạnh các nội dung được bình chọn trên 80%, tôi cũng thêm một số nội dung khác để làm cho ứng dụng trở nên phong phú hơn. Các tính năng tương ứng của ứng dụng chatbot eBot được trình bày trong bảng 6 dưới đây.

STT	Tính năng	Mô tả
1	Trả lời những thông tin cố định	Tính năng này cho phép chatbot trả lời bằng những câu trả lời cố định được mặc định trước đó.

2	Trả lời những thông tin do nhà trường thông báo.	Tính năng này cho phép chatbot trả lời những thông tin được nhà trường đăng tải trên trang chủ của trường. Giới hạn của tính năng này là chỉ cho phép tìm kiếm trên trang chủ của nhà trường.
3	Trả lời những thông tin thường gặp, ít thay đổi được lưu trữ trong cơ sở dữ liệu.	Tính năng này cho phép chatbot trả lời những thông tin ít khi được thay đổi. Các thông tin này được lưu trữ trong cơ sở dữ liệu. Khi người dùng yêu cầu những thông tin này, chatbot eBot sẽ bắt đầu tìm kiếm trên cơ sở dữ liệu trước khi thực hiện khác thao tác khác.
4	Giúp người dùng thực hiện hành động.	Tính năng này cho phép chatbot eBot hỗ trợ người dùng thực hiện hành động đăng ký học phần. Công việc chính của tính năng này là kiểm tra thông tin và thực hiện thay đổi trên cơ sở dữ liệu.
5	Trả lời những thông tin từ kết quả tìm kiếm trên internet.	Tính năng này cho phép chatbot trả lời những câu hỏi bằng cách tìm kiếm từ nhiều nguồn khác nhau trên internet, không chỉ riêng trong trang chủ của trường. Tính năng này phục vụ khi thao tác tìm kiếm trên cơ sở dữ liệu không có câu trả lời.

*Bảng 6: Các tính năng của ứng dụng eBot*

Sau khi phân tích những nội dung cần được hỗ trợ và xây dựng được các tính năng cần có của ứng dụng, bước tiếp theo là xây dựng giải pháp xây dựng phản hồi cho từng tính năng.

#### **4. Giải pháp xây dựng phản hồi**

Việc xử lý ngôn ngữ tự nhiên và phân tích yêu cầu đầu vào đã được hỗ trợ bởi nền tảng Dialogflow. Chi tiết về công cụ Dialogflow vui lòng tham khảo mục 3 của phụ lục. Nguyên nhân chọn nền tảng Dialogflow được trình bày trong mục 5.2 của chương 2. Do đó, trong khóa luận này chỉ tập trung xây dựng giải pháp để tạo ra phản hồi.

Dựa vào các tính năng cần có của ứng dụng ở bảng 6, hệ thống sẽ thực hiện thiết kế các nội dung cần được hỗ trợ thành các intent. Sau đó phân loại các loại intent vào các danh mục. Đối với mỗi danh mục sẽ có cách phản hồi khác nhau. Danh sách các intent được chia làm 4 danh mục:

Chia intent làm 4 loại:

- Intent loại 1: thường là các intent mang chủ đề mà câu trả lời của nó không mang tính chất xử lý thông tin như: chào hỏi, tạm biệt. Đối với loại intent này, câu trả lời trả về sẽ là câu trả lời cố định, không thay đổi.
- Intent loại 2: là những intent mang chủ đề mà câu trả lời của nó không nằm trong cơ sở dữ liệu và cũng không thể có câu trả lời cố định, thường được thông báo theo lịch định kì của nhà trường: lịch thi học kì, cách đăng kí học phần, thông tin về học bổng.
- Intent loại 3: là những intent mang chủ đề mà câu trả lời của nó được xác định từ kết quả truy vấn cơ sở dữ liệu như: môn học trước, môn học thay thế, số lượng môn học chuyên ngành.

- Intent loại 4: là những intent mà yêu cầu của nó là thực hiện một hành động. Đối với loại intent này, hệ thống sẽ hỗ trợ người dùng thực hiện một loạt các hành động liên quan để có thể hoàn thành yêu cầu.

Khi hệ thống nhận một yêu cầu từ người dùng, nó sẽ xác định yêu cầu thuộc loại intent nào. Trong trường hợp hệ thống không nhận ra truy vấn của người dùng thuộc intent nào, sẽ có một intent dự phòng (fallback intent) trả thông báo về cho người dùng. Trường hợp hệ thống nhận ra intent của câu truy vấn thì sẽ xử lý như sau:

- Nếu là intent loại 1, hệ thống chỉ đơn giản trả về câu trả lời cố định được xây dựng mặc định trước đó.
- Nếu là intent loại 2, hệ thống sẽ sử dụng các API hỗ trợ bên ngoài để tìm kiếm thông tin từ trên trang web của nhà trường.
- Nếu là intent loại 3, hệ thống sẽ kiểm tra và tìm kiếm câu trả lời bằng cách truy vấn trên cơ sở dữ liệu. Nếu không có kết quả trả về từ cơ sở dữ liệu, hệ thống sẽ tìm kiếm câu trả lời trên internet thông qua API.
- Nếu là intent loại 4: hệ thống sẽ thực hiện những thay đổi trên cơ sở dữ liệu.

Việc tìm kiếm câu trả lời trên internet thông qua API có thể làm cho câu trả lời thiếu tính chính xác, tuy nhiên nó giúp người dùng hiểu được nỗ lực tìm kiếm câu trả lời của chatbot dành cho câu hỏi của họ và ít nhất người dùng cũng nhận được phản hồi từ các yêu cầu của họ dành cho chatbot.

Bên cạnh đó, để tạo ra câu trả lời cho câu hỏi của người dùng, hệ thống phải có đủ thông tin về câu hỏi. Do đó, việc hỏi thêm những câu hỏi liên quan đến khi nhận đủ thông tin là điều cần thiết.

Bằng việc xây dựng các tham số bắt buộc - tham số mà giá trị của nó không được rỗng và các nhắc nhở (prompt), hệ thống sẽ lấy đủ thông tin cần thiết để xây

dụng phản hồi. Các nhắc nhở này sẽ được kích hoạt khi giá trị của tham số bắt buộc mang giá trị rỗng.

Ví dụ: Với mỗi câu hỏi về điểm số của một môn học có hai tham số bắt buộc là *ma\_so\_sinh\_vien* và *ten\_mon\_hoc*. Khi sinh viên gửi đến ứng dụng câu truy vấn “*tôi muốn xem điểm*”. Lúc này, hai tham số *ma\_so\_sinh\_vien* và *ten\_mon\_hoc* mang giá trị rỗng, hệ thống sẽ kích hoạt các nhắc nhở “*mã số sinh viên của bạn là gì?*” và “*vui lòng cung cấp tên môn học mà bạn muốn xem điểm*” để người dùng có thể cung cấp đầy đủ thông tin cần thiết. Hình 11 dưới đây minh họa việc cung cấp các nhắc nhở cho các tham số bắt buộc.

The screenshot shows a chatbot interface with a header "Prompts for 'ten\_mon\_hoc'" and a close button. Below the header is a table with three columns: NAME, ENTITY, and VALUE. The NAME column contains "Tên tham số" and "ten\_mon\_hoc". The ENTITY column contains "@ten\_mon\_hoc". The VALUE column contains "\$ten\_mon\_hoc". Below the table is a list of prompts. The first prompt is "Vui lòng cung cấp tên môn học." and it is highlighted with a red box. To the right of the prompt is a red text label: "Nhắc nhở sẽ được kích hoạt khi tham số ten\_mon\_hoc có giá trị rỗng".

NAME	ENTITY	VALUE
Tên tham số		
ten_mon_hoc	@ten_mon_hoc	\$ten_mon_hoc

PROMPTS
1
Vui lòng cung cấp tên môn học.

Nhắc nhở sẽ được kích hoạt khi tham số ten\_mon\_hoc có giá trị rỗng

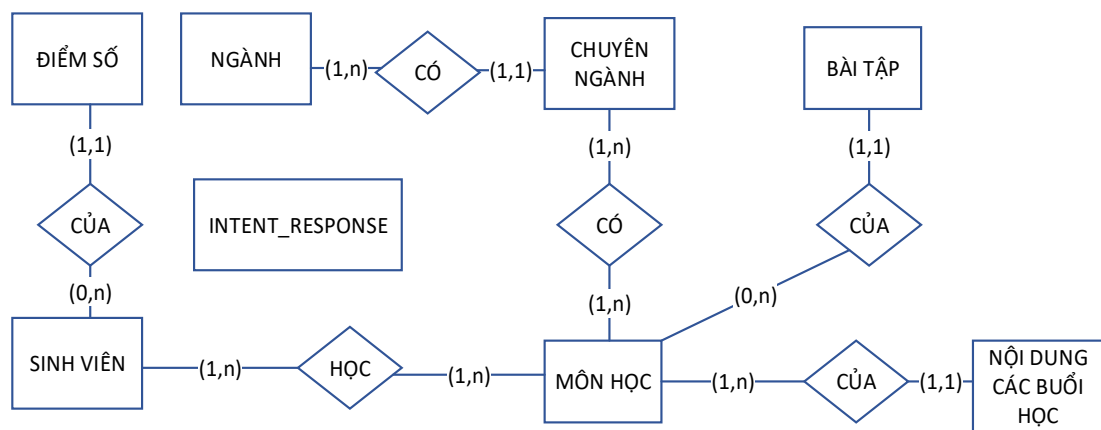
Hình 11: Giao diện cung cấp nhắc nhở đối với các tham số bắt buộc

Sau khi đã xây dựng giải pháp cho việc trả lời câu hỏi của người dùng, tiếp đến là xây dựng cơ sở dữ liệu cho ứng dụng.

## 5. Cấu trúc cơ sở dữ liệu dùng để xây dựng phản hồi

Để trả lời các câu hỏi liên quan đến loại intent thứ 3 và thực hiện các yêu cầu của intent loại 4, thực hiện xây dựng cơ sở dữ liệu và thực hiện truy vấn trên cơ sở dữ liệu để trả về kết quả cho người dùng. Hình 12 là lược đồ ER cơ sở dữ liệu của ứng dụng và bảng 7 thể hiện danh sách các bảng trong cơ sở dữ liệu được xây dựng để xây dựng phản hồi cho chatbot.





Hình 12: Lược đồ ER của cơ sở dữ liệu của ứng dụng eBot

STT	Tên bảng	Mô tả
1	INTENT_RESPONSE	Chứa danh sách các intent được phân loại và câu truy vấn tương ứng để truy vấn thông tin từ cơ sở dữ liệu.
2	NGANH	Chứa danh sách các ngành học.
3	CHUYENNGANH	Chứa danh sách các chuyên ngành của mỗi ngành học.
4	MONHOC	Chứa danh sách các môn học ngành Công nghệ thông tin của trường Đại học Khoa Học Tự Nhiên.

5	NOIDUNGCACBUOIHOC	Chứa nội dung tóm gọn các buổi học của từng môn học.
6	BAITAP	Chứa nội dung bài tập các bài tập của các môn học.
7	SINHVIEN	Cho biết thông tin sinh viên
8	DIEMSO	Chứa thông tin điểm số từng môn học của sinh viên.
9	MONHOCTRUOC	Cho biết các môn học trước của một môn học, tham chiếu đến bảng môn học.
10	MONHOCTHAYTHE	Cho biết các môn học thay thế của một môn học, tham chiếu đến bảng môn học.
11	SINHVIEN_MONHOC	Sinh viên đăng kí môn học nào.
12	CHUYENNGANH_MONHOC	Cho biết môn học nào thuộc chuyên ngành nào.

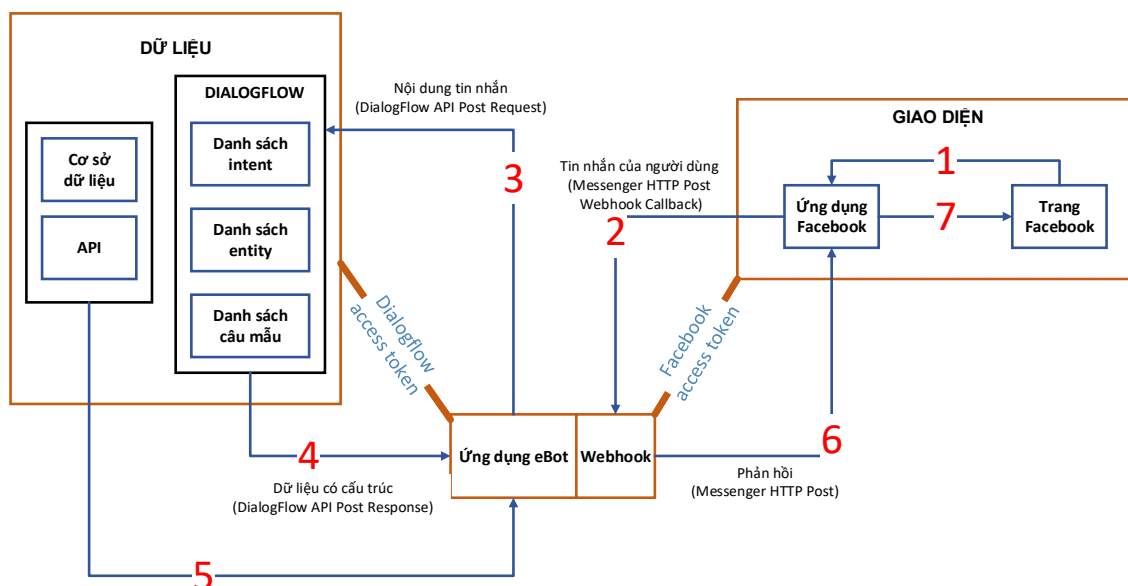
*Bảng 7: Danh sách các bảng trong cơ sở dữ liệu*

Chi tiết thuộc tính của từng bảng vui lòng tham khảo tại mục 5 của phụ lục.

## CHƯƠNG 4: CÀI ĐẶT ỨNG DỤNG CHATBOT HỖ TRỢ HỌC TẬP

Chương 4 sẽ trình bày tổng quan về quá trình cài đặt ứng dụng hỗ trợ học tập. Phần đầu tiên sẽ trình bày sơ đồ hoạt động của ứng dụng eBot trong thực tế. Tiếp theo là chi tiết về quá trình chuẩn bị và cài đặt chi tiết của các hàm trong chương trình.

### 1. Sơ đồ tổng quát hoạt động của ứng dụng



Hình 13: Sơ đồ hoạt động của ứng dụng chatbot eBot

Hình 13 biểu diễn sơ đồ hoạt động của chatbot trong thực tế. Quá trình khi người dùng gửi một tin nhắn đến chatbot đến lúc chatbot trả lời lại như sau:

- Bước 1: Khi người dùng gửi tin nhắn đến trang Facebook thông qua nền tảng Facebook Messenger, ứng dụng Facebook sẽ lắng nghe sự kiện này.
- Bước 2: Messenger API sẽ gửi một HTTP callback tới ứng dụng eBot.
- Bước 3: Ứng dụng sẽ tách lấy nội dung tin nhắn và gửi yêu cầu đến Dialogflow, một nền tảng hỗ trợ xử lý ngôn ngữ tự nhiên, để Dialogflow phân tích nội dung tin nhắn này.

- Bước 4: Sau khi phân tích xong, Dialogflow sẽ trả về một dạng dữ liệu có cấu trúc dưới dạng JSON. JSON là 1 định dạng hoán vị dữ liệu nhanh, với cấu trúc đơn giản là mỗi thông tin dữ liệu sẽ có 2 phần đó là key và value. Dữ liệu trả về bao gồm các thông tin như intent, entity, context và các tham số cần thiết.
- Bước 5: Sau khi nhận được phản hồi dưới dạng JSON của Dialogflow, ứng dụng sẽ thực hiện các thao tác bổ sung cần thiết như truy vấn trên cơ sở dữ liệu hoặc gọi API thích hợp để tạo ra phản hồi thích hợp.
- Bước 6: Phản hồi được trả về cho ứng dụng Facebook.
- Bước 7: Phản hồi sẽ thông qua ứng dụng Facebook hiển thị lên giao diện Facebook Messenger cho người dùng.

Phía trên đã trình bày sơ đồ hoạt động của ứng dụng chatbot eBot trong thực tế và quá trình truyền dữ liệu giữa các thành phần với nhau. Mục tiếp theo sẽ trình bày quá trình chuẩn bị và cài đặt ứng dụng.

## **2. Cài đặt ứng dụng**

### **2.1. Quá trình chuẩn bị**

Mục đích của quá trình chuẩn bị là để tạo ra các kết quả sau:

- Về dữ liệu:
  - Một danh sách các intent, entity và câu mẫu phục vụ cho việc huấn luyện mô hình trên Dialogflow - nền tảng hỗ trợ xử lý ngôn ngữ tự nhiên của Google.
  - Một cơ sở dữ liệu để có thể truy vấn.
- Về giao diện:
  - Một trang Facebook và ứng dụng Facebook. Liên kết trang Facebook với ứng dụng Facebook.
- Về ứng dụng:

- Một ứng dụng tối thiểu chạy trên localhost, ứng dụng được kết nối với Facebook và mô hình trên Dialogflow.
- Đường hàm liên kết giữa localhost và internet phục vụ cho quá trình kiểm thử ứng dụng.

Các mục sau sẽ trình bày chi tiết về các nội dung cần phải chuẩn bị.

### **2.1.1. Về dữ liệu**

Như mô hình ở hình 13, bộ dữ liệu được chia làm 2 phần là dữ liệu huấn luyện mô hình trên Dialogflow và cơ sở dữ liệu. Các phần sau đây trình bày quá trình chuẩn bị cho từng phần.

#### **2.1.1.1. Xây dựng dữ liệu và huấn luyện mô hình trên Dialogflow**

Để Dialogflow có thể phân tích yêu cầu của người dùng thành dạng dữ liệu có cấu trúc cần phải có bộ dữ liệu và tiến hành huấn luyện để Dialogflow có thể xây dựng được mô hình riêng.

##### **a) Giai đoạn tạo dữ liệu:**

Tạo một danh sách các intent, danh sách các entity và danh sách các câu mẫu. Phân loại các câu theo intent và xác định tham số trong mỗi câu. Bảng 8 và bảng 9 dưới đây là danh sách các intent và entity được dùng để huấn luyện tạo mô hình trên Dialogflow.

<b>Tên intent</b>	<b>Mô tả</b>
bai_tap	Dùng để nhận ra các câu hỏi về vấn đề bài tập.
chao_hoi	Dùng để nhận ra các câu mang tính chất chào hỏi.
dang_ki_mon_hoc	Dùng để nhận ra yêu cầu giúp đăng ký môn học.
diem_so	Dùng để nhận ra các câu hỏi thắc mắc về điểm số.
huy_mon_hoc	Dùng để nhận ra yêu cầu giúp hủy môn học.
mon_hoc_thay_the	Dùng để nhận ra các câu hỏi thắc mắc về môn học thay thế của một môn học.
mon_hoc_truoc	Dùng để nhận ra các câu hỏi thắc mắc về môn học trước của một môn học.
noi_dung_mon_hoc	Dùng để nhận ra các câu hỏi thắc mắc về nội dung môn học.
so_luong_chuyen_nganh	Dùng để nhận ra các câu hỏi thắc mắc về số lượng chuyên ngành của một ngành học.
so_luong_mon_hoc_chuyen_nganh	Dùng để nhận ra các câu hỏi thắc mắc về các môn học bắt buộc và môn học tự chọn của một chuyên ngành.
tai_lieu_mon_hoc	Dùng để nhận ra các câu hỏi thắc mắc về tài liệu môn học của một môn học.

tam_biet	Dùng để nhận ra các câu mang tính chất chào tạm biệt kết thúc cuộc trò chuyện.
thong_tin_khac	Dùng để nhận ra các câu hỏi liên quan đến các thông tin về: học bổng, thời khóa biểu, lịch thi, kết quả thi học kì, quy chế nộp đơn xét tốt nghiệp.
unknow	Dùng để nhận ra các câu hỏi nằm ngoài phạm vi được xác định.

*Bảng 8: Danh sách các intent được dùng để huấn luyện mô hình trên Dialogflow*

Tên entity	Mô tả
@ma so sinh vien	Chứa một vài mẫu mã số sinh viên
@ten mon hoc	Chứa danh sách các môn học của ngành Công nghệ thông tin trường đại học Khoa học tự nhiên TP.HCM
@ten nganh	Chứa danh sách các ngành học của trường đại học Khoa học tự nhiên TP.HCM
@ten chuyen nganh	Chứa danh sách tên chuyên ngành của mỗi ngành trường đại học Khoa học tự nhiên TP.HCM
@hinh thuc mon hoc	Chứa các hình thức môn học: tự chọn, bắt buộc.

*Bảng 9: Danh sách các entity được dùng để rút trích thông tin trong các câu mẫu trong mỗi intent trên Dialogflow*

## b) Giai đoạn huấn luyện:

Tạo một agent trên Dialogflow, để phân tích nội dung đầu vào của người dùng. Dialogflow được hỗ trợ bởi các kỹ thuật máy học để có thể phân loại các truy vấn của người dùng vào các intent, trích xuất các entity trong truy vấn của người dùng và nhận ra ngữ cảnh của cuộc trò chuyện. Để Dialogflow có thể phân tích được câu truy vấn của người dùng thì nó phải được huấn luyện với một bộ dữ liệu. Quá trình huấn luyện gồm hai phần là: phân nhóm nội dung truy vấn của người dùng và xác định tham số.

### - Phân nhóm nội dung truy vấn của người dùng

Phân nhóm các câu hỏi của người dùng (mỗi nhóm nên có từ 10 đến 15 câu mẫu) vào các intent thích hợp. Hình 14 mô tả cách nội dung được phân vào các intent.

**mon\_hoc\_truoc** Tên intent

Training phrases

- đầu học phần nào trước khi đăng kí môn học **Cơ sở dữ liệu**?
- có cần học chuyển đề nào trước khi học môn **Cơ sở dữ liệu** hay không?
- trước khi học môn **Cơ sở dữ liệu** cần phải học **trước** môn học nào
- học trước môn nào **trước** khi học môn học **Cơ sở dữ liệu**
- để học được môn **hệ điều hành** cần thi cần học môn gì **trước**
- trước khi học môn **Cơ sở dữ liệu** cần học học phần nào
- học trước học phần nào trước khi theo học môn học **Cơ sở dữ liệu**
- cần đăng kí học phần nào trước khi học môn **Cơ sở dữ liệu** hay không?
- trước khi đăng kí môn **Cơ sở dữ liệu** cần phải học **trước** môn học nào?
- cần học môn nào **trước** khi học môn **Cơ sở dữ liệu** hay không

Danh sách các câu hỏi liên quan đến môn học trước được mong đợi sẽ kích hoạt intent

**diem\_so** Tên intent

Training phrases

- điểm số của môn **Cơ sở dữ liệu**
- điểm của môn học **Cơ sở dữ liệu**
- minh muốn biết **điểm** môn **Cơ sở dữ liệu**
- điểm số của học phần **Cơ sở dữ liệu**
- điểm môn **Cơ sở dữ liệu**
- điểm môn **Cơ sở dữ liệu** của mình
- tui muốn biết **điểm** môn **trí tuệ nhân tạo**
- điểm môn **kỹ năng mềm**
- điểm môn học **Tâm lý đại cương**
- điểm môn **Trí tuệ nhân tạo** của tui

Danh sách các câu hỏi liên quan đến điểm số được mong đợi sẽ kích hoạt intent

Hình 14: Minh họa phân nhóm nội dung vào các intent

### - Xác định tham số

Đối với mỗi câu mẫu trong intent, xác định các tham số - thành phần kết nối thông tin người dùng với entity - trong mỗi câu hỏi.

Ví dụ: xác định 10-15 câu hỏi liên quan đến điểm số của một môn học, sau đó đưa toàn bộ những câu này vào một intent có tên là *diem\_so*. Với mỗi câu hỏi,



cần phải xác định tên của môn học cần biết điểm số. Do đó, tạo một tham số là *ten\_mon\_hoc*, tham số này được kết nối với một entity tên là *@ten\_mon\_hoc*, nơi chứa danh sách các tên môn học được xây dựng trước đó để lấy đúng thông tin tên môn học. Hình 15 minh họa việc xác định tham số trong câu hỏi của người dùng.

PARAMETER NAME	ENTITY	RESOLVED VALUE
ten_mon_hoc	@ten_mon_hoc	Cơ sở dữ liệu

Tham số - thành phần kết nối thông tin người dùng với các entity

Tên entity - Nơi chứa danh sách các môn học được xây dựng trước

Giá trị của tham số

Hình 15: Minh họa về việc xác định tham số trong câu hỏi

#### 2.1.1.2. Cơ sở dữ liệu

Viết script xây dựng cơ sở dữ liệu và tạo cơ sở dữ liệu. Để xây dựng cơ sở dữ liệu, sử dụng MySQL - một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở và MySQL Workbench - là một công cụ thiết kế cơ sở dữ liệu trực quan tích hợp phát triển SQL, môi trường phát triển tích hợp duy nhất cho hệ thống cơ sở dữ liệu MySQL. Cấu trúc cơ sở dữ liệu được trình bày ở mục 5 chương 3.

#### 2.1.2. Về giao diện

Về giao diện, ứng dụng eBot sẽ sử dụng Facebook Messenger – một ứng dụng và nền tảng tin nhắn được phát triển bởi Facebook - để làm giao diện người dùng.



*Hình 16: Giao diện tin nhắn hội thoại Messenger của Facebook.*

Để giải thích việc lựa chọn ứng dụng trò chuyện để tích hợp làm giao diện, cần xem qua số liệu báo cáo về lượng người dùng sử dụng các ứng dụng trò chuyện ở Việt Nam, được thực hiện bởi Di-Marketing [13] và được báo cáo vào tháng 6 năm 2016. Theo báo cáo, 94% người dùng tại Việt Nam sử dụng ứng dụng trò chuyện trong đó 42% người dùng sử dụng ứng dụng Messenger của Facebook một cách thường xuyên. Độ tuổi sử dụng ứng dụng Messenger nhiều nhất nằm ở khoảng 18-25. Điều này chứng tỏ Messenger của Facebook là nền tảng thích hợp để tích hợp với lượng người dùng hướng đến là học sinh, sinh viên. Nếu được tích hợp với Messenger sẽ giúp ứng dụng tiếp cận với học sinh, sinh viên một cách dễ dàng hơn vì không cần phải cài đặt thêm bất kỳ phần mềm nào.

Quá trình tạo giao diện diễn ra như sau:

- *Tạo một trang trên Facebook (Facebook fanpage).* Cần phải tạo một trang Facebook vì hiện nay Facebook chỉ cho phép kết nối trang Facebook với ứng dụng Facebook mà không cho phép các tài khoản cá nhân kết nối với ứng dụng

Facebook. Giao diện tương tác với người dùng chính là hộp tin nhắn Messenger đến trang Facebook.

- *Tạo một ứng dụng Facebook (Facebook app).* Ứng dụng Facebook là chương trình được thiết kế để tương tác với các nền tảng cốt lõi của Facebook thông qua API. Ứng dụng Facebook là ứng dụng liên kết với trang Facebook với webhook. Đồng thời đây cũng sẽ là nơi cấu hình các cài đặt khác nhau cho chatbot Messenger, tạo các mã truy cập và chọn những sự kiện nào được gửi tới webhook.
- *Liên kết ứng dụng Facebook với trang Facebook.* Sau khi liên kết và cấu hình webhook, nếu người dùng thực hiện nhắn tin tới chatbot Messenger của trang Facebook được liên kết với ứng dụng Facebook, tin nhắn được gửi dưới dạng sự kiện sẽ được gửi đến webhook. Từ đây, lập trình viên có thể lấy được dữ liệu mà người dùng gửi đến để xử lý.

### 2.1.3. Về cấu hình ứng dụng

#### 2.1.3.1. Tạo ứng dụng

Đầu tiên cần xây dựng ít nhất một chương trình tối thiểu và liên kết ứng dụng với Dialogflow và ứng dụng Facebook. Lí do sử dụng NodeJS để xây dựng ứng dụng chatbot eBot vì ứng dụng được xây dựng bằng NodeJS dễ thiết lập; bên cạnh đó, NodeJS thực thi mã theo cơ chế bất đồng bộ, phù hợp với ứng dụng tương tác thời gian thực (real-time) khi định dạng dữ liệu được lưu trữ và trao đổi dưới dạng JSON. (Xem thêm [14] )

```
1  var http = require('http');
2  var express = require('express');
3  var app = express();
4
5  app.set('port', process.env.OPENSIFT_NODEJS_PORT || process.env.PORT || 8080);
6  app.set('ip', process.env.OPENSIFT_NODEJS_IP || process.env.IP || "127.0.0.1");
7
8  var server = http.createServer(app);
9  server.listen(app.get('port'), app.get('ip'), function() {
10 |   console.log("Chat bot server listening at %s:%d ", app.get('ip'), app.get('port'));
11 | });
```

*Hình 17: Ứng dụng NodeJS tối thiểu*

Đoạn mã trong hình 17 là một chương trình NodeJS tối thiểu. Đoạn mã bắt đầu với việc tải hai module là HTTP và Express, đặt biến máy chủ thành localhost (127.0.0.1) và port là 8080. Sau đó tạo một máy chủ, yêu cầu máy chủ lắng nghe trên localhost với port 8080 và định nghĩa một hàm callback với nội dung in ra màn hình console một thông báo khi máy chủ bắt đầu lắng nghe.

Khi sử dụng lệnh **node** chạy đoạn mã ở hình 14 trên cửa sổ dòng lệnh, nếu máy chủ đã sẵn sàng thì màn hình sẽ có kết quả tương tự như hình 15.

```
node index.js
Chat bot server listening at 127.0.0.1:8080
```

*Hình 18: Chương trình chạy bằng lệnh node*

Tuy nhiên, mỗi lần có sự thay đổi mã lệnh, chương trình đều phải được khởi động lại. Điều này gây ra rất nhiều bất tiện cho việc phát triển. Do đó, NodeJS cung cấp một module là **nodemon**. Nodemon là một tiện ích sẽ giám sát mọi thay đổi trong mã nguồn và tự động khởi động lại máy chủ, giúp tiết kiệm nhiều thời gian cho lập trình viên. Nodemon được cài đặt toàn cục để có thể gọi được từ bất kì đâu trên máy tính. Hình 16 minh họa cho việc tự động giám sát mã nguồn và khởi động lại máy chủ khi có sự thay đổi mã nguồn.

```
nodemon index.js
[nodemon] 1.17.5
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node index.js`
Chat bot server listening at 127.0.0.1:8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Chat bot server listening at 127.0.0.1:8080
```

*Hình 19: Chương trình chạy bằng lệnh nodemon*

### 2.1.3.2. Liên kết agent trên Dialogflow và ứng dụng Facebook với ứng dụng chatbot eBot thông qua mã truy cập (token)

Mã truy cập (token) là thông tin xác thực được ứng dụng sử dụng để truy cập API. Mục đích của nó là để thông báo cho API rằng ứng dụng mang mã truy cập này đã được ủy quyền truy cập API và thực hiện các hành động cụ thể được chỉ định bởi phạm vi dịch vụ được cung cấp. Hình 20 minh họa việc đưa mã xác thực vào code.

```
const DIALOGFLOW_AI_TOKEN = '0c2fd7d8eca148709cbbcbce23843a3f7';  
const FACEBOOK_ACCESS_TOKEN = 'EAACQrYT690EBABFZBArg65TWKAUnq1XLyNmfyl
```

*Hình 20: Minh họa đưa mã truy cập vào code để xác thực*

### 2.1.3.3. Cấu hình webhook

Webhook là một HTTP callback [15], một cách để một ứng dụng có thể cung cấp thông tin theo thời gian thực cho những ứng dụng khác. Một webhook cung cấp thông tin cho các ứng dụng khác gần như ngay lập tức. Đối với webhook, nó không bắt buộc phải có yêu cầu, nó trả về dữ liệu khi dữ liệu có sẵn. Nó cho phép dịch vụ mà chúng ta sử dụng gọi tới ứng dụng của chúng ta thông qua một điểm cuối (endpoint) cụ thể. Ứng dụng sẽ lắng nghe những sự kiện xảy ra thay vì phải thăm dò các thay đổi có thể xảy ra.

Để triển khai một webhook phù hợp cho ứng dụng Facebook, mã nguồn phải được lưu trữ trên máy chủ HTTP công khai với những yêu cầu sau (xem thêm tại [16])

- Hỗ trợ HTTPS
- Có chứng chỉ SSL (Secure Socket Layer) - công nghệ bảo mật tiêu chuẩn để thiết lập liên kết được mã hóa giữa máy chủ web và trình duyệt.
- Cổng giao tiếp chấp nhận yêu cầu GET và POST.

Trong quá trình phát triển, việc lưu trữ mã nguồn trên một máy chủ HTTP công khai thực sự chưa cần thiết vì có thể phát sinh nhiều chi phí và gây khó khăn

trong quá trình cập nhật mã nguồn. Tuy nhiên cần phải tạo ra một webhook đáp ứng các nhu cầu trên để có thể thực hiện kiểm thử trong quá trình phát triển.

Giải pháp được đưa ra là sử dụng một công cụ tên là **Ngrok** - một công cụ tạo đường hầm (tunnel) giữa localhost và internet – để xây dựng webhook tới localhost một cách dễ dàng, thuận tiện cho quá trình kiểm thử. Ứng dụng khác trên mạng có thể truy cập được localhost thông qua tên miền ngẫu nhiên được tạo ra bởi của **Ngrok**. Hình 21 minh họa việc Ngrok tạo tên miền tùy chọn liên kết với localhost. Đường hầm từ webhook tới localhost được duy trì trong vòng 8 giờ đồng hồ tính từ lúc bắt đầu tạo đường hầm.

Forwarding	http://66b3f9ce.ngrok.io -> localhost:8080
Forwarding	https://66b3f9ce.ngrok.io -> localhost:8080

*Hình 21: Minh họa tên miền tùy chọn được tạo ra bởi Ngrok liên kết với localhost*

Khi tạo đường hầm thành công, dùng tên miền tùy chọn được tạo ra bởi Ngrok để đưa vào ứng dụng Facebook. Nền tảng Messenger sẽ gửi sự kiện đến webhook của để thông báo cho chatbot khi xảy ra nhiều tương tác hoặc sự kiện khác nhau, bao gồm cả khi một người gửi tin nhắn. Sự kiện webhook được nền tảng Messenger gửi như yêu cầu POST đến webhook. Sau đó yêu cầu POST này sẽ được chuyển tiếp về localhost.

Sau quá trình chuẩn bị sẽ có một cơ sở dữ liệu hoàn chỉnh, một chương trình NodeJS đơn giản đã được kết nối với ứng dụng Facebook và mô hình trên Dialogflow. Những bước chuẩn bị này sẽ phục vụ cho quá trình cài đặt chi tiết được trình bày ở phần sau.

## **2.2. Quá trình cài đặt chi tiết**

### **2.2.1. Cấu trúc thư mục của chương trình:**

Dưới đây là cấu trúc chung của chương trình bao gồm danh sách các tập tin, thư mục và mô tả chức năng của chúng. Tất cả được trình bày trong bảng 10.

Tập tin/thư mục	Mô tả
database.js	Chứa thông tin của cơ sở dữ liệu bao gồm: tên cơ sở dữ liệu, tên và mật khẩu đăng nhập, host, port kết nối, số lượng kết nối đối đa.
index.js	Mã chi tiết của chương trình được cài đặt ở đây.
package.json	NodeJS xây dựng trên nhiều module gọi là package, quản lý thông qua NPM - trình quản lý các package của NodeJS. package.json là tập tin cấu hình của NPM, chứa danh sách các package mà dự án phụ thuộc vào và phiên bản của chúng. Tập tin package.json giúp cho NPM hiểu cần phải cài đặt những gì cho dự án. Bên cạnh đó, tập tin package.json cũng chứa thông tin khác về ứng dụng như tên tác giả, mô tả,...
node_modules (thư mục)	Thư mục node_modules này được sinh ra một cách tự động. Các package sau khi tải về được đưa vào thư mục này. Để gọi và sử dụng được các package, chỉ cần thông qua cú pháp require (“tên_package”).
package-lock.json	Tập tin package-lock.json được sinh ra một cách tự động nhằm đảm bảo sự cài đặt nhất quán và tương thích. (xem thông tin chi tiết ở [17])

*Bảng 10: Danh sách file và thư mục được dùng để xây dựng ứng dụng*

### 2.2.2. Chi tiết cài đặt các hàm

### 2.2.2.1. Cài đặt và cấu hình cơ sở dữ liệu

Hình 18 mô tả mã nguồn cấu hình cơ sở dữ liệu được cài đặt trong tập tin **database.js**:

```
1  var mysql = require('mysql');
2  var conn = mysql.createConnection({
3    host      : '██████████',
4    user      : '██████',
5    password  : '██████',
6    database  : '████████████████████',
7    port      : █████,
8    connectionLimit : █████
9  });
10 conn.connect(function(err) {
11   if (err) throw err;
12   else console.log ("CONNECTED !!!")
13 });
14 module.exports = conn;
```

Hình 22: Nội dung tập tin dùng để cấu hình cơ sở dữ liệu

Trong đoạn mã nguồn ở hình 22 chỉ rõ việc cung cấp các thông tin về cơ sở dữ liệu bao gồm: host, tên người dùng và mật khẩu, tên cơ sở dữ liệu, cổng kết nối và số lượng kết nối tối đa. Nếu quá trình kết nối cơ sở dữ liệu không gặp bất kì vấn đề gì sẽ hiển thị thông báo đã được kết nối trên màn hình console. Tiếp đến là chi tiết về quá trình cài đặt webhook

### 2.2.2.2. Cài đặt webhook

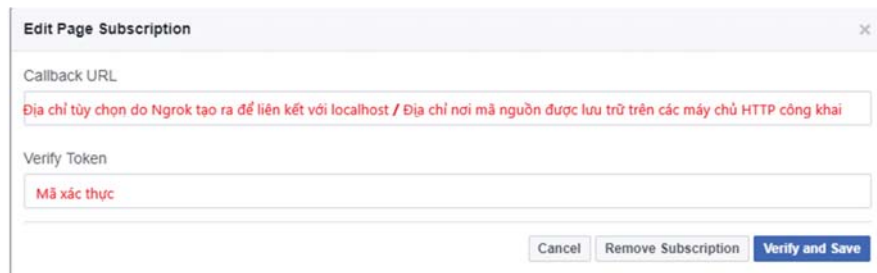
```
1  app.get('/webhook', function(req, res) {
2    const hubChallenge = req.query['hub.challenge'];
3    const hubMode = req.query['hub.mode'];
4    const verifyTokenMatches = (req.query['hub.verify_token'] === '██████████');
5    if (hubMode && verifyTokenMatches) {
6      res.status(200).send(hubChallenge);
7    } else {
8      res.status(403).end();
9    }
10 });
```

Hình 23: Đoạn mã dùng để đăng ký và xác thực webhook



Đoạn mã trong hình 23 được dùng để xác thực webhook. Nó bảo đảm webhook được xác thực và hoạt động đúng. Quá trình xác thực diễn ra như sau:

- Tạo một mã xác thực là một chuỗi ký tự ngẫu nhiên tự chọn.
- Cung cấp mã xác thực này cho nền tảng Messenger của Facebook khi đăng ký webhook của mình trên ứng dụng Facebook để nhận sự kiện. Hình 24 là giao diện của ứng dụng Facebook, nơi các lập trình viên cung cấp mã xác thực.



The image shows a dialog box titled "Edit Page Subscription". It has two main input fields: "Callback URL" and "Verify Token". The "Callback URL" field has a red text hint below it: "Địa chỉ tùy chọn do Ngrok tạo ra để liên kết với localhost / Địa chỉ nơi mã nguồn được lưu trữ trên các máy chủ HTTP công khai". The "Verify Token" field has a red text hint below it: "Mã xác thực". At the bottom right, there are three buttons: "Cancel", "Remove Subscription", and "Verify and Save".

*Hình 24: Giao diện để cung cấp mã xác thực*

- Nền tảng Messenger sẽ gửi một yêu cầu GET đến webhook với mã thông báo trong tham số *hub.verify\_token* của chuỗi truy vấn.
- Nếu mã xác thực được gửi khớp với mã được ghi trong mã nguồn thì trả về tham số *hub.challenge*.
- Nền tảng Messenger sẽ đăng ký webhook này với ứng dụng Facebook.

```

1 app.post('/webhook', function(req, res) {
2   var entries = req.body.entry;
3   for (var entry of entries) {
4     var messaging = entry.messaging;
5     for (var message of messaging) {
6       var senderId = message.sender.id;
7       if (message.message) {
8         if (message.message.text) {
9           sendTypingActionToUser(senderId)
10
11           const apiAiSession = apiAiClient.textRequest(message.message.text, {sessionId: 'rukawa10051996'});
12
13           apiAiSession.on('response', (response) => { ...
14             });
15
16           apiAiSession.on('error', error => console.log(error));
17           apiAiSession.end();
18         }
19       }
20     }
21   }
22   res.status(200).send("OK");
23 });

```

Đoạn code gửi nội dung tin nhắn đến người dùng đến Dialogflow để phân tích

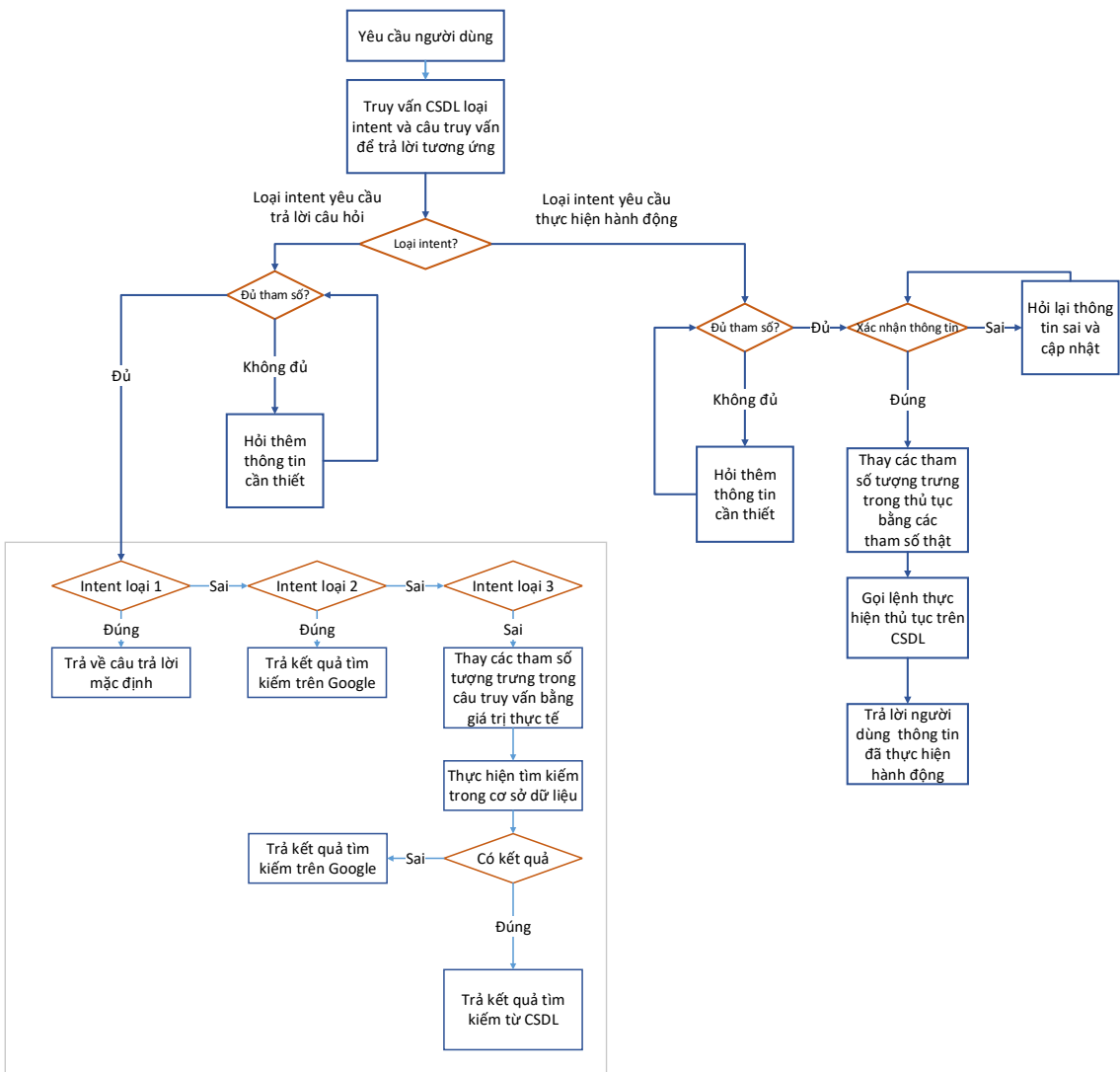
Đoạn code nhận phản hồi từ Dialogflow, phân tích và trả kết quả về cho người dùng.

Hình 25: Đoạn mã dùng để tạo điểm kết nối cuối để nhận yêu cầu POST

Đoạn mã ở hình 25 dùng để tạo một điểm cuối để nhận yêu cầu POST. Điểm cuối (end-point) này là nơi mà nền tảng Messenger sẽ gửi đến tất cả các sự kiện webhook. Sau khi nhận sự kiện webhook, kết quả sẽ được sử dụng khác nhau cho các công việc cụ thể.

- Đầu tiên, nội dung tin nhắn sẽ được gửi đến Dialogflow để phân tích nội dung. Dialogflow sau khi phân tích sẽ trả về kết quả dưới dạng JSON.
- Kết quả được trả về từ Dialogflow bao gồm intent, entity, context, các tham số cần thiết sẽ được xử dụng và xử lý để xây dựng phản hồi phù hợp và trả về cho người dùng

### 2.2.2.3. Cài đặt chức năng



Hình 26: Sơ đồ thực thi của ứng dụng

Hình 26 mô tả sơ đồ thực thi của ứng dụng. Dựa trên sơ đồ này để thiết kế các hàm cần thiết. Danh sách các hàm được cài đặt trong chương trình và chức năng của chúng được mô tả trong bảng 11 bên dưới. Những hàm này được viết chi tiết trong tập tin **index.js**.

Hàm	Chức năng
-----	-----------

sendTextMessageToUser	Gửi tin nhắn dưới dạng text đến người dùng.
sendTypingActionToUser	Gửi hành động đang gõ phím trả lời. Chức năng này cho cảm giác như đang trò chuyện với con người.
sendPayloadToUser	Gửi biểu mẫu xác nhận đến cho người dùng.
sendGoogleSearchResultToUser	Gửi kết quả tìm kiếm trên Google đến người dùng.
replaceParamValueInProcedure	Thay thế các giá trị tham số trong thủ tục được lấy lên từ cơ sở dữ liệu.
replaceParamValueInQuery	Thay thế các giá trị tham số trong câu truy vấn được lấy lên từ cơ sở dữ liệu.
confirmActionInfo	Xác nhận thông tin khi người dùng yêu cầu thực hiện hành động đăng ký hoặc hủy học phần.
doAction	Thực hiện hành động đăng ký hoặc hủy học phần và trả về kết quả.
responseToUser	Xác định loại intent, thực hiện gọi các hàm liên quan để trả về câu trả lời cho người dùng.

*Bảng 11: Danh sách các hàm được cài đặt trong chương trình*

Dưới đây sẽ trình bày cài đặt chi tiết của ba hàm quan trọng nhất là hàm *confirmActionInfo*, hàm *doAction* và hàm *responseToUser*

```

1  function confirmActionInfo (senderId, studentId, subject){
2      sendPayloadToUser (senderId, {
3          "type": "template",
4          "payload": {
5              "template_type": "button",
6              "text": "Bạn kiểm tra thông tin đúng không nhé" + "\nMSSV: "
7                  + studentId + "\nTên môn học:" + subject,
8              "buttons": [
9                  {
10                     "type": "postback",
11                     "title": "Đúng thông tin",
12                     "payload": "Ok"
13                 },
14                 {
15                     "type": "postback",
16                     "title": "Sai MSSV",
17                     "payload": "wrongStudentId"
18                 },
19                 {
20                     "type": "postback",
21                     "title": "Sai tên môn học",
22                     "payload": "wrongSubject"
23                 }
24             ]
25         }
26     })
27 }

```

Hình 27: Mã nguồn của hàm *confirmActionInfo*

Đoạn mã nguồn ở hình 27 được dùng để xác nhận thông tin trước khi giúp người dùng thực hiện hành động. Khi gửi biểu mẫu đến người dùng, sẽ có ba lựa chọn cho người dùng:

- Trường hợp đúng thông tin, hệ thống sẽ gửi về *Ok*.
- Trường hợp sai mã số sinh viên hệ thống sẽ trả về *wrongStudentId*.
- Trường hợp sai tên môn học hệ thống sẽ trả về *wrongSubject*.

```

1 function doAction (senderId, paramName, paramValue, payload){
2     //Thực hiện hành động
3     if (payload === 'Ok'){ ...
37    }
38    //Hỏi lại mã số sinh viên
39    else if (payload === 'wrongStudentId') {
40        actionIsCompleted = false
41        sendTextMessageToUser(senderId, 'Mã số sinh viên đúng của bạn là gì?');
42    }
43    }
44    //Hỏi lại tên môn học
45    else if (payload === 'wrongSubject') {
46        actionIsCompleted = false
47        sendTextMessageToUser(senderId, 'Bạn vui lòng cho mình biết' +
48            'tên môn học chính xác. (Ví dụ: Cơ sở dữ liệu)');
49    }
50 }

```

*Hình 28: Mã nguồn của hàm doAction*

Hình 28 là mã nguồn của hàm doAction. Theo mã nguồn được cài đặt:

- Trường hợp cờ lệnh là *Ok*, hệ thống sẽ thực hiện hành động và trả về kết quả.
- Trường hợp cờ lệnh là *wrongStudentId*, hệ thống sẽ hỏi lại mã số sinh viên.
- Trường hợp cờ lệnh là *wrongSubject*, hệ thống sẽ hỏi lại tên môn học.

Khi giúp đăng ký học phần, hệ thống sẽ không kiểm tra thông tin về mã số sinh viên hợp lệ hay không do giới hạn về dữ liệu trong việc xây dựng cơ sở dữ liệu.

```

1 var sql = "select intentType, response as res from intent_response where intent = ?";
2 db.query(sql, [intentName], function(err, sqlResponse) {
3     if (err) throw err;
4     Object.keys(sqlResponse).map(function(key) {
5         var row = sqlResponse[key];
6         sendTextMessageToUserBeforeAction(senderId, paramMessage, function(){
7             switch (row.intentType) {
8                 case 1: ...
14                case 2: ...
43                case 3: ...
45            }
46        });
47    });
48 })

```

*Hình 29: Một phần mã nguồn của hàm responseToUser*

Đoạn mã ở hình 29 là một trong những đoạn mã nguồn quan trọng nhất của chương trình. Đoạn mã nguồn này là nơi phân tích và xử lý loại intent để trả về câu

trả lời phù hợp. Đối với từng loại intent sẽ có cách phản hồi khác nhau. Cách phản hồi được mô tả trong hình 26.

Các hàm còn lại dùng để phục vụ cho quá trình truyền nhận dữ liệu, thay thế tham số trong câu truy vấn hoặc thủ tục. Chi tiết các hàm này vui lòng tham khảo mục 4 của phụ lục.

## CHƯƠNG 5: TỔNG KẾT, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

*Chương này sẽ tập trung trình bày về những kết quả tôi đúc kết được qua quá trình thực hiện luận văn, đồng thời đưa ra những đánh giá, nhận xét về những gì tôi đã làm được trong suốt thời gian qua.*

### 1. Kết quả đạt được và hạn chế

Sau nhiều tháng tìm hiểu và nghiên cứu đề tài với mục tiêu đề ra từ đầu là tìm hiểu về smart-chatbot và xây dựng ứng dụng hỗ trợ tra cứu tài nguyên học tập, tôi đã gặp không ít khó khăn. Nhưng bên cạnh những khó khăn đó, tôi cũng học được nhiều điều trong quá trình nghiên cứu đề tài: tiếp thu những kiến thức mới như cài đặt ứng dụng, tự xây dựng giải pháp trước một vấn đề được đưa ra, đặc biệt hiểu rõ hơn về cách thức xây dựng một quy trình chức năng phát triển phần mềm. Ngoài ra, tôi còn học hỏi được nhiều kiến thức bổ ích khác trong quá trình làm luận văn như: cách thức trình bày báo cáo, cách làm việc độc lập, khả năng chịu áp lực cao. Sau đây tôi sẽ trình bày kết quả sau quá trình nghiên cứu và xây dựng thử nghiệm hệ thống hỗ trợ tra cứu tài nguyên học tập.

#### 1.1. Về lý thuyết

- Hiểu được khái niệm chatbot, các lý thuyết cơ sở liên quan đến việc xây dựng chatbot và phân loại chatbot.
- Hiểu được nguyên lý vận hành của chatbot và một số mô hình phổ biến dùng để xây dựng chatbot cùng với ưu nhược điểm của từng loại.
- Tổng hợp và so sánh được các công nghệ, nền tảng phổ biến dùng để xây dựng chatbot.

#### 1.2. Về ứng dụng

- Khả năng phân tích và tìm hiểu được các công nghệ đang phát triển hiện nay để xây dựng ứng dụng một cách hiệu quả.
- Khả năng phân tích yêu cầu người dùng và tạo giải pháp.



- Có khả năng xây dựng một ứng dụng hoàn chỉnh, có kết nối cơ sở dữ liệu và sử dụng các API hỗ trợ bên ngoài.

### **1.3. Hạn chế**

- Do thời gian có hạn và chưa có nhiều kinh nghiệm trong lĩnh vực xử lý ngôn ngữ tự nhiên nên thay vì phải tự phân tích và thiết kế thuật toán để phân tích câu truy vấn của người dùng thì sử dụng các nền tảng có sẵn.
- Ứng dụng được xây dựng chưa thực sự thông minh và linh hoạt, cần tạo dữ liệu để huấn luyện lớn hơn để quá trình phân tích câu truy vấn của người dùng diễn ra một cách chính xác hơn.
- Việc sử dụng cơ sở dữ liệu quan hệ xây dựng trả lời câu hỏi còn khá cứng nhắc, trong tương lai có thể thay đổi sang một dạng cơ sở dữ liệu khác linh hoạt hơn.

## **2. Hướng phát triển**

Ứng dụng của đề tài chỉ là ứng dụng nhỏ, sơ khởi cho việc xây dựng chatbot. Ứng dụng sử dụng các thư viện, nền tảng bên ngoài để hỗ trợ phân tích nội dung đầu vào, không tự cài đặt các thuật toán để thực hiện điều này. Do đó, trong tương lai cần phát triển thêm phần nghiên cứu, cài đặt, đánh giá thuật toán và các kỹ thuật máy học để phân tích nội dung đầu vào. Bên cạnh đó, có thể tiếp tục nghiên cứu xây dựng phản hồi kết hợp giữa nhiều mô hình khác nhau để tạo phản hồi với độ chính xác cao hơn và linh động hơn so với việc sử dụng cơ sở dữ liệu quan hệ.

## PHỤ LỤC

*Phụ lục dùng để trình bày các khái niệm và công nghệ liên quan trực tiếp đến luận văn nhưng không được nêu chi tiết ở phần chính của luận văn.*

### 1. AIML

AIML (Artificial Intelligence Mark-up Language) bắt nguồn từ XML, ban đầu được phát triển bởi tiến sĩ Richard Wallace và cộng đồng phát triển phần mềm tự do trên toàn thế giới từ năm 1995-2002. AIML cho phép người dùng nhập kiến thức cho những cuộc trò chuyện dành cho chatbot.

AIML mô tả một lớp các đối tượng dữ liệu gọi là đối tượng AIML và mô tả một phần hành vi của chương trình máy tính xử lý chúng. AIML bắt đầu với một nút gốc là `<AIML>`. Tất cả các thành phần còn lại sẽ là con của nút gốc này. Đối tượng AIML được tạo thành từ các đơn vị được gọi là các danh mục (category) chứa các dữ liệu đã được phân tích cú pháp hoặc chưa được phân tích cú pháp.

AIML bao gồm rất nhiều thành phần. Trong đó những thành phần quan trọng nhất được trình bày bên dưới:

- **Danh mục (category):** được xem là đơn vị kiến thức cơ bản trong AIML, một danh mục sẽ bao gồm ít nhất hai thành phần khác là khuôn mẫu (pattern) và bản mẫu (template).

Ví dụ: dưới đây là một đoạn kịch bản mẫu được viết bằng AIML.

```
<category>
<pattern>Bạn tên gì? </pattern>
<template>Tên đầy đủ của tôi là Hồ Thảo Hiền</template>
</category>
```

Khi danh mục này được nạp, chatbot sử dụng đoạn AIML này sẽ trả lời “Tên đầy đủ của tôi là Hồ Thảo Hiền” cho câu hỏi “Bạn tên gì”.

- **Khuôn mẫu (pattern):** là một chuỗi các ký tự tương ứng với một hoặc nhiều đầu vào của người dùng.

Tiếp tục với đoạn kịch bản mẫu phía trên, khuôn mẫu “Bạn tên gì?” chỉ tương ứng phù hợp với một đầu vào. Tuy nhiên, các khuôn mẫu cũng có thể chứa các ký tự đại diện, các ký tự này có thể khớp với một hoặc nhiều từ. Ví dụ, với khuôn mẫu “\* tên gì?”, sau khi thế thay thế ký tự \* thành “anh ấy”, “cô ấy”, “con chó này”, “cửa hàng gần siêu thị”, ...sẽ tương ứng với nhiều đầu vào khác nhau.

Cú pháp của một khuôn mẫu trong AIML rất đơn giản, ít phức tạp hơn so với các biểu thức chính quy (regular expression). Để bù đắp cho các khả năng khớp mẫu đơn giản, trình thông dịch AIML có thể cung cấp các chức năng tiền xử lý để mở rộng các từ viết tắt, loại bỏ lỗi chính tả,...

- **Bản mẫu (template):** chỉ định phản hồi phù hợp cho một khuôn mẫu tương ứng. Một bản mẫu có thể đơn giản là một số văn bản chữ, như “Tên đầy đủ của tôi là Hồ Thảo Hiền” hoặc có thể sử dụng biến như “Tên đầy đủ của tôi là <bot\_name= “name”>”. Bản mẫu có thể được điều hướng đến các khuôn mẫu khác nhau. Xét ví dụ với kịch bản sau đây:

```
<category>
<pattern>Tên bạn là gì?</pattern>
<template><![CDATA[Tên tôi là <bot_name="name"/>.]></template>
</category>
<category>
<pattern>Tôi có thể gọi bạn là gì?</pattern>
<template>
<srai>Tên bạn là gì?</srai>
</template>
</category>
```

Danh mục đầu tiên chỉ đơn giản là trả lời một đầu vào với tên là “Bạn tên gì?”  
Danh mục thứ hai với đầu vào “Tôi có thể gọi bạn là gì nhỉ?” được điều hướng về danh mục thứ nhất, điều này nói lên rằng hai câu này có nghĩa giống nhau.

Nguyên tắc hoạt động của AIML là tất cả các dữ liệu được biên dịch trước vào một cấu trúc cây. Khi chatbot nhận được một câu, nó sẽ duyệt qua các danh mục cho đến khi tìm thấy được khuôn mẫu phù hợp với câu đầu vào của người dùng. Nếu tìm được khuôn mẫu khớp với câu của người dùng, nó sẽ dùng bản mẫu tương ứng để tạo ra câu trả lời.

Cấu trúc và cách hoạt động của AIML đơn giản nên nó phù hợp với những người mới tập làm quen xây dựng chatbot. Tuy nhiên vì đơn giản và nguyên tắc nên nó trở thành điểm yếu của AIML. Các điểm yếu của AIML như sau:

- AIML sẽ không thể nhận ra nếu đầu vào không khớp với tất cả các từ trong khuôn mẫu.
- AIML dựa trên XML, mà XML thì không thân thiện với người dùng.
- Vì không thể nhìn thấy tất cả các quy tắc được cài đặt trong AIML và quy tắc nào khớp với đầu vào nào nên khi bạn tham gia làm việc với một hệ thống sử dụng AIML, việc viết, gỡ lỗi và bảo trì sẽ trở nên khó khăn hơn rất nhiều. Mặc dù có thể dùng biểu thức chính quy để hỗ trợ, tuy nhiên rất ít người có thể đọc được biểu thức chính quy một cách nhanh chóng mà không cần sự trợ giúp của một chương trình hỗ trợ khác.

## 2. Chatscript

Theo [18], ChatScript là một ngôn ngữ kịch bản được thiết kế để chấp nhận câu người dùng nhập vào và tạo ra một phản hồi. Chương trình lấy một hoặc nhiều câu từ người dùng và xuất ra một hoặc nhiều câu trả lời trở lại. Một kịch bản được viết bằng ChatScript thường bao gồm các thành phần: quy tắc (rule), chủ đề (topic), khoảng trắng (white space) và chú thích (comment).

- **Quy tắc (rule):** đơn vị cơ bản của ChatScript là các quy tắc (rule). Một quy tắc đầy đủ bao gồm loại (kind), nhãn (label), khuôn mẫu (pattern) và đầu ra (output).

Ví dụ: dưới đây là một rule đầy đủ trong ChatScript:

*?: HOCTAP (Bạn có thích học Toán không) Tôi thích học Toán lắm.*

Trong ví dụ trên, loại quy tắc (rule kind) là *?:* nghĩa là nó sẽ phản ứng với các câu có dạng là câu hỏi, nhãn (rule label) là *HOCTAP*, mẫu (rule pattern) là những từ được ghi trong dấu ngoặc và nó sẽ được dùng tìm kiếm cả cụm “Bạn có thích học Toán không” trong câu đầu vào của người dùng, đầu ra (rule output) là “Tôi thích học Toán lắm.”

- **Loại quy tắc (rule kind):** Loại quy tắc sẽ hạn chế khi quy tắc được xét đến. Loại quy tắc có thể hạn chế được loại câu đầu vào (tuyên bố hoặc câu hỏi). Loại quy tắc là một chữ cái, đi theo sau là dấu hai chấm. Ví dụ, có một số loại quy tắc như sau:

*s: nghĩa là quy tắc sẽ phản ứng với các tuyên bố (statement)*

*?: nghĩa là quy tắc sẽ phản ứng với các câu hỏi*

*u: nghĩa là quy tắc sẽ phản ứng với cả câu hỏi và các tuyên bố.*

- **Nhãn (rule label):** nhãn là một lựa chọn không bắt buộc, nó chỉ được dùng khi quy tắc được gọi hoặc điều khiển bởi các quy tắc khác. Ngoài ra, nó giúp ghi lại tài liệu, giúp gỡ lỗi hoặc bảo trì trong tương lai.

- **Khuôn mẫu (rule pattern):** là tập hợp các câu, các điều kiện cụ thể hơn cho phép hoặc không cho phép thực hiện quy tắc nào đó. Thông thường khuôn mẫu sẽ được so khớp với các từ trong câu đầu vào của người dùng, nhưng đôi khi dùng để tính toán lịch sử của cuộc trò chuyện.
- **Đầu ra (rule output):** Đầu ra chính là phản hồi hoặc điều mà quy tắc này được phép thực hiện. Bởi vì mục tiêu của toàn bộ hệ thống là tạo ra phản hồi nên đầu ra thường là chuỗi câu trả lời. Đầu ra phức tạp có thể bao gồm các lời gọi hàm, vòng lặp hay điều kiện vòng lặp.

Hệ thống thường thực thi các quy tắc theo một thứ tự đã được chỉ định cho đến khi có một quy tắc nào đó khớp với các hạn chế được đặt ra và cho ra phản hồi cuối cùng phù hợp với người dùng.

- **Chủ đề (topics):** các quy tắc được chia theo chủ đề. Khi hệ thống thực thi một chủ đề, nó sẽ thực thi các quy tắc trong chủ đề đó cho đến khi nó tạo ra một phản hồi. Các chủ đề có thể gọi lẫn nhau. Cách xử lý đầu là kiểm soát một chủ đề với một đoạn lệnh điều khiển. Nó thường gọi các hàm và thực thi một cách có điều kiện các chủ đề khác nhau.
- **Khoảng trắng (whitespace):** ChatScript thường bỏ qua khoảng trắng vượt quá quy định như nhiều khoảng trắng hay cách dòng liên tục. Tuy nhiên phải có khoảng trắng để tách các thành phần khác nhau.
- **Chú thích (comment):** chú thích là các kí tự # dùng để chú thích ý nghĩa của một hay một đoạn script nào đó. Chú thích này tương tự như chú thích trong các ngôn ngữ lập trình C/C++, Java, C#,... trình biên dịch sẽ bỏ qua chú thích khi thực hiện biên dịch kịch bản ChatScript.

Chatscript loại bỏ sự cần thiết khi đầu vào phải khớp toàn bộ các từ trong khuôn mẫu được xây dựng sẵn. Các mẫu ChatScript nhỏ gọn và mạnh mẽ nhưng dễ hiểu, dễ đọc và dễ viết. Vì vậy, các mẫu có thể được gỡ rối và duy trì dễ dàng hơn so với AIML.

### 3. Dialogflow

Theo [19], Dialogflow là công nghệ hỗ trợ tương tác giữa người và máy tính dựa trên các cuộc trò chuyện bằng ngôn ngữ tự nhiên. Nó được Google mua lại vào tháng 9 năm 2016 với tên gọi ban đầu là API.AI và đổi tên thành Dialogflow vào tháng 10 năm 2017.

Khi còn tên là API.AI, công nghệ này được xây dựng với mục đích là cung cấp cho các nhà phát triển một API để hỗ trợ khả năng xử lý ngôn ngữ tự nhiên vào các ứng dụng của họ. Tuy nhiên, sau khi xây dựng các tính năng mới như công cụ hỗ trợ phân tích vào API.AI, Google đã quyết định đổi tên nó thành Dialogflow để phục vụ cho nhiều mục đích hơn.

Dialogflow cung cấp một giao diện web thân thiện cho cả những lập trình viên chuyển nghiệp lẫn những người không chuyên về lập trình để xây dựng và kiểm thử kịch bản các cuộc trò chuyện. Nền tảng này có khả năng tích hợp với các nền tảng khác như Skype, Slack, Whatapp, Messenger và những thiết bị như xe tự động, loa, điện thoại. Nó hỗ trợ hơn 14 ngôn ngữ bao gồm tiếng Anh, tiếng Tây Ban Nha, tiếng Trung quốc và nhiều ngôn ngữ khác.

Dialogflow được hỗ trợ bởi các kỹ thuật máy học để có thể phân loại các truy vấn của người dùng vào các intent, trích xuất các entity trong truy vấn của người dùng và nhận ra ngữ cảnh của cuộc trò chuyện.

Thuật toán cài đặt chi tiết của Dialogflow không được công bố. Tuy nhiên, theo tài liệu của Dialogflow, thuật toán sẽ được tùy chỉnh dựa theo dữ liệu của người dùng đưa vào huấn luyện. Dữ liệu từ các cuộc trò chuyện của người dùng trong quá trình sử dụng được lưu lại để làm nguồn dữ liệu huấn luyện tiếp tục cho chatbot.

#### 4. Chi tiết cài đặt các hàm trong chương trình

Nội dung phần này sẽ giải thích phần cài đặt chi tiết của các hàm trong chương trình.

##### - Hàm `sendTextMessageToUser`

```
1 function sendTextMessageToUser(senderId, text) {  
2     request({  
3         url: 'https://graph.facebook.com/v2.6/me/messages',  
4         qs: { access_token: FACEBOOK_ACCESS_TOKEN },  
5         method: 'POST',  
6         json: {  
7             recipient: { id: senderId },  
8             message: { text },  
9         }  
10    });  
11 }
```

Hình 30: Hàm gửi tin nhắn dạng text đến người dùng

Trong hàm dùng để gửi tin nhắn đến người dùng (mã nguồn hình 30) thực hiện việc gửi một yêu cầu POST đến máy chủ Facebook với nội dung tập tin là ID người gửi và nội dung muốn gửi. Nội dung này sẽ được trả về cho người dùng và hiển thị lên giao diện.

##### - Hàm `sendTypingActionToUser`

```
1 function sendTypingActionToUser (senderId, response, callback){  
2     request({  
3         url: 'https://graph.facebook.com/v2.6/me/messages',  
4         qs: { access_token: FACEBOOK_ACCESS_TOKEN },  
5         method: 'POST',  
6         json: {  
7             recipient: { id: senderId },  
8             sender_action: "typing_on"  
9         }  
10    });  
11    callback(senderId, response)  
12 }
```

Hình 31: Hàm gửi hành động đang gõ phím đến người dùng

Trong hàm dùng để gửi hành động gõ phím đến người dùng (mã nguồn hình 31) thực hiện một yêu cầu POST đến máy chủ Facebook với nội dung tập tin



là ID người gửi và hành động được thực hiện. Hành động này sẽ được gửi đến người dùng và hiển thị lên giao diện, giúp người dùng có cảm giác đang có người gõ phím, giúp cho cuộc trò chuyện trở nên sinh động hơn. Hàm này có gọi một callback, hàm callback này sẽ được thực hiện ngay sau khi hành động gõ phím được gửi tới cho người dùng.

- **Hàm sendPayloadToUser**

```
1 function sendPayloadToUser(senderId, attachment) {  
2   request({  
3     url: 'https://graph.facebook.com/v2.6/me/messages',  
4     qs: { access_token: FACEBOOK_ACCESS_TOKEN },  
5     method: 'POST',  
6     json: {  
7       recipient: { id: senderId },  
8       message: { attachment },  
9     }  
10  });  
11 }
```

*Hình 32: Hàm gửi biểu mẫu đến người dùng*

Mã nguồn trong hình 32 thực hiện một yêu cầu POST đến máy chủ Facebook với nội dung tập tin là ID người gửi và biểu mẫu tương ứng. Biểu mẫu sẽ được trả về cho người dùng và hiển thị lên giao diện

## - Hàm sendGoogleSearchToUser

```
1 | google.resultsPerPage = 2
2 | function sendGoogleSearchResultToUser (senderId, userQuery){
3 |     google( userQuery , function (err, res){
4 |         if (err) throw err
5 |         if (res.links.length === 0){
6 |             sendTextMessageToUser(senderId, 'Xin lỗi,'
7 |                                     + 'tôi không tìm thấy câu trả lời cho câu hỏi của bạn,'
8 |                                     + 'tôi rất lấy làm tiếc!');
9 |         }
10 |         else {
11 |             sendTextMessageToUserBeforeAction(senderId, 'Bạn tham khảo thêm thông tin'
12 |                                                  + 'trong các link dưới đây nhé.', function(){
13 |                 for (i = 0; i<res.links.length; i++){
14 |                     sendTextMessageToUser(senderId, res.links[i].href);
15 |                     console.log (res)
16 |                 }
17 |             });
18 |         }
19 |     })
20 | }
```

*Hình 33: Hàm dùng để tìm kiếm kết quả từ Google và trả về cho người dùng.*

Mã nguồn ở hình 33 được dùng để tìm kiếm kết quả từ Google và trả về cho người dùng. Trong đoạn mã ở dòng đầu tiên sử dụng một module có tên là **google**. Module này cho phép lấy kết quả tìm kiếm từ Google mà không sử dụng Google Search API. Tại dòng thứ hai cho phép lấy về số lượng kết quả tối đa mỗi lần thực hiện tìm kiếm. Kết quả tìm kiếm sẽ được trả về cho người dùng. Trong trường hợp không tìm thấy kết quả tìm kiếm nào, hệ thống sẽ xuất ra câu thông báo không tìm thấy kết quả tương ứng.

## - Hàm `confirmActionInfo`

```
1 function confirmActionInfo (senderId, studentId, subject){
2     sendPayloadToUser (senderId, {
3         "type": "template",
4         "payload": {
5             "template_type": "button",
6             "text": "Bạn kiểm tra thông tin đúng không nhé" + "\nMSSV: "
7                 + studentId + "\nTên môn học:" + subject,
8             "buttons": [
9                 {
10                    "type": "postback",
11                    "title": "Đúng thông tin",
12                    "payload": "Ok"
13                },
14                {
15                    "type": "postback",
16                    "title": "Sai MSSV",
17                    "payload": "wrongStudentId"
18                },
19                {
20                    "type": "postback",
21                    "title": "Sai tên môn học",
22                    "payload": "wrongSubject"
23                }
24            ]
25        }
26    })
27 }
```

Hình 34: Thông tin được gửi để xác nhận với người dùng

Đoạn mã nguồn ở hình 34 được dùng để xác nhận thông tin trước khi giúp người dùng thực hiện hành động. Khi gửi biểu mẫu đến người dùng, sẽ có ba lựa chọn cho người dùng:

- Trường hợp đúng thông tin, hệ thống sẽ gửi về *Ok*.
- Trường hợp sai mã số sinh viên hệ thống sẽ trả về *wrongStudentId*.
- Trường hợp sai tên môn học hệ thống sẽ trả về *wrongSubject*.

## - Hàm doAction

```
1 function doAction (senderId, paramName, paramValue, payload){
2     //Thực hiện hành động
3     if (payload === 'Ok'){ ...
37 }
38     //Hỏi Lại mã số sinh viên
39     else if (payload === 'wrongStudentId') {
40         actionIsCompleted = false
41         sendTextMessageToUser(senderId, 'Mã số sinh viên đúng của bạn là gì?');
42     }
43 }
44     //Hỏi Lại tên môn học
45     else if (payload === 'wrongSubject') {
46         actionIsCompleted = false
47         sendTextMessageToUser(senderId, 'Bạn vui lòng cho mình biết' +
48             'tên môn học chính xác. (Ví dụ: Cơ sở dữ liệu)');
49     }
50 }
```

*Hình 35: Hàm kiểm tra thông tin được xác nhận  
và thực hiện hành động giúp cho người dùng*

Mã nguồn trong hình 35 mô tả quá trình hỗ trợ người dùng thực hiện hành động.

- Trường hợp cờ lệnh là *Ok*, hệ thống sẽ thực hiện hành động
- Trường hợp cờ lệnh là *wrongStudentId*, hệ thống sẽ hỏi lại mã số sinh viên.
- Trường hợp cờ lệnh là *wrongSubject*, hệ thống sẽ hỏi lại tên môn học.

## - Hàm `responseToUser`

```
1 function responseToUser (senderId, resolvedQuery, speech, intentName, paramName, paramValue) {
2     var sql = "select intentType, response as res from intent_response where intent = ?";
3     db.query(sql, [intentName], function(err, response) {
4         if (err) throw err;
5         Object.keys(response).map(function(key) {
6             var row = response[key];
7             //Trường hợp intent Loại 1
8             if (row.intentType === 0){ ...
11         }
12         //Trường hợp intent Loại 2
13         else if (row.intentType === 1){ ...
17         }
18         //Trường hợp intent Loại 3
19         else if (row.intentType === 2) { ...
48         }
49         //Trường hợp intent Loại 4
50         else { ...
71         }
72     });
73 }
74 }
```

*Hình 36: Hàm trả về kết quả cho người dùng*

Đoạn mã ở hình 36 là đoạn mã quan trọng nhất của chương trình. Đoạn mã này là nơi phân tích và xử lý intent để trả về kết quả phù hợp. Đối với từng loại intent sẽ có cách phản hồi khác nhau. Các điều kiện về intent được mô tả trong sơ đồ khối trong hình 27.

## - Hàm replaceParamValueInQuery

```
1 function replaceParamValueInQuery (sqlString, paramName, paramValue){
2     for (i = 0; i < paramName.length; i++){
3         if (paramValue[i] === 'undefined' || paramValue[i] === ''){
4             sqlString = sqlString.replace(paramName[i], null)
5         }
6         else {
7             if (paramName[i] === 'hinh_thuc_mon_hoc'){
8                 if (paramValue[i] === 'bắt buộc'){
9                     paramValue[i] = 1
10                }
11                else {
12                    paramValue[i] = 2
13                }
14            }
15            console.log (paramName[i] + ' ' + paramValue[i])
16            if (isNaN(paramValue[i]) === true){
17                sqlString = sqlString.replace(paramName[i], "N'" + paramValue[i] + "%'")
18            }
19            else {
20                sqlString = sqlString.replace(paramName[i], paramValue[i])
21            }
22        }
23    }
24 }
25 console.log (sqlString)
26 return sqlString
27 }
```

Hình 37: Hàm thay đổi các tham số trong câu truy vấn.

## - Hàm replaceParamValueInProcedure

```
1 function replaceParamValueInProcedure (sqlString, paramName, paramValue){
2     for (i = 0; i < paramName.length; i++){
3         if (paramValue[i] === 'undefined' || paramValue[i] === ''){
4             sqlString = sqlString.replace(paramName[i], null)
5         }
6         else {
7             console.log (paramName[i] + ' ' + paramValue[i])
8             if (isNaN(paramValue[i]) === true){
9                 sqlString = sqlString.replace(paramName[i], "" + paramValue[i] + "")
10            }
11            else {
12                sqlString = sqlString.replace(paramName[i], paramValue[i])
13            }
14        }
15    }
16    console.log (sqlString)
17    return sqlString
18 }
```

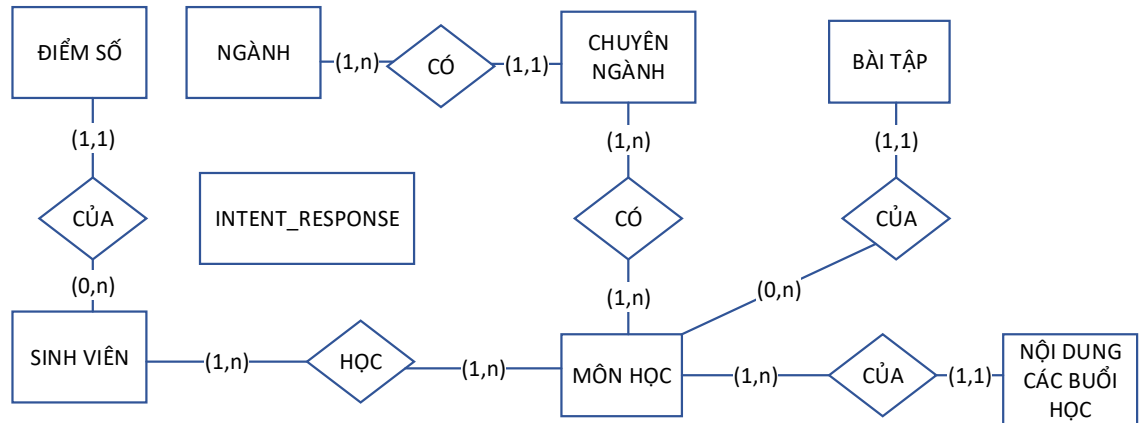
Hình 38: Hàm thay đổi các tham số trong procedure.

Đoạn mã ở hình 37 và hình 38 được dùng để thay thế các tham số tượng trưng trong câu truy vấn SQL hoặc thủ tục bằng các giá trị tham số thực tế được lấy về từ Dialogflow sau khi gửi nội dung tin nhắn của người dùng đến để phân tích.

## 5. Chi tiết thuộc tính các bảng trong cơ sở dữ liệu

Hình 39 dưới đây là lược đồ ER của cơ sở dữ liệu của ứng dụng eBot được cài đặt.

Chi tiết các thuộc tính của các bảng trong cơ sở dữ liệu được trình bày bên dưới.



Hình 39: Sơ đồ ER của cơ sở dữ liệu

Bảng: INTENT_RESPONSE			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Id	Int	Mã định danh
2	Intenttype	Int	Loại intent
3	Intent	String	Tên intent tương ứng với tên intent được thiết kế để huấn luyện trên Dialogflow
4	Response	String	Câu truy vấn chứa các tham số tương ứng với từng intent.

Bảng 12: Cấu trúc dữ liệu bảng INTENT\_RESPONSE

Bảng: <b>NGANH</b>			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Id	Int	Mã định danh
2	Tennganh	String	Tên ngành học

*Bảng 13: Cấu trúc dữ liệu bảng NGANH*

Bảng: <b>CHUYENNGANH</b>			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Id	Int	Mã định danh
2	Nganh	Int	Mã định danh của ngành học
3	Tenchuyennganh	String	Tên chuyên ngành

*Bảng 14: Cấu trúc dữ liệu bảng CHUYENNGANH*

Bảng: <b>MONHOC</b>			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Id	Int	Mã định danh
2	Tenmonhoc	String	Tên môn học
3	Noidungmonhoc	String	Nội dung môn học

*Bảng 15: Cấu trúc dữ liệu bảng MONHOC*



Bảng: <b>NOIDUNGCACBUOIHOC</b>			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Id	Int	Mã định danh
2	Monhoc	Int	Mã định danh môn học
3	Tuan	Int	Tuần thứ mấy học môn học
4	Thoigian	Datetime	Thời gian diễn ra buổi học
5	Noidungbuoihoc	String	Nội dung buổi học

*Bảng 16: Cấu trúc dữ liệu bảng NOIDUNGCACBUOIHOC*

Bảng: <b>BAITAP</b>			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Id	Int	Mã định danh
2	Monhoc	Int	Mã định danh của môn học
3	Noidungbaitap	String	Nội dung bài tập

*Bảng 17: Cấu trúc dữ liệu bảng BAITAP*

Bảng: <b>SINHVIEN</b>			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Id	Int	Mã định danh
2	MSSV	String	Mã số sinh viên
3	Tensinhvien	String	Tên sinh viên

*Bảng 18: Cấu trúc dữ liệu bảng SINHVIEN*

Bảng: <b>DIEMSO</b>			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Id	Int	Mã định danh
2	Sinhvien	Int	Mã số sinh viên
3	Monhoc	Int	Mã định danh môn học
4	Diem	Int	Điểm số môn học

*Bảng 19: Cấu trúc dữ liệu bảng DIEMSO*

Bảng: <b>MONHOCTHAYTHE</b>			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Monhoc	Int	Mã định danh môn học
2	Monhoctruoc	Int	Mã định danh của môn học trước

*Bảng 20: Cấu trúc dữ liệu bảng MONHOCTHAYTHE*

Bảng: MONHOCTHAYTHE			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Monhoc	Int	Mã định danh môn học
2	Monhocthaythe	Int	Mã định danh của môn học thay thế

*Bảng 21: Cấu trúc dữ liệu bảng MONHOCTRUOC*

Bảng: CHUYENNGANH_MONHOC			
Stt	Tên	Kiểu dữ liệu	Giải thích
1	Id	Int	Mã định danh
2	Chuyennganh	Int	Mã định danh của chuyên ngành
3	Monhoc	Int	Mã định danh của môn học
4	Hinhthucmonhoc	Int	Hình thức môn học với 1 là bắt buộc, 2 là tự chọn.

*Bảng 22: Cấu trúc dữ liệu bảng CHUYENNGANH\_MONHOC*

## TÀI LIỆU THAM KHẢO

- [1] “What are the benefits of using a chatbot,” [Trực tuyến]. Available: <https://www.marutitech.com/benefits-chatbot/>.
- [2] J. Lachs, “Will chatbots revolutionise education,” [Trực tuyến]. Available: <https://www.opencolleges.edu.au/informed/features/will-chatbots-revolutionise-education/>.
- [3] “What is a Chatbot,” [Trực tuyến]. Available: <https://aws.amazon.com/what-is-a-chatbot>.
- [4] “Chatbot,” [Trực tuyến]. Available: <https://en.wikipedia.org/wiki/Chatbot>.
- [5] Ketakee Nimavat, Tushar Chamapaneria, “Chatbots: An overview. Types, Architecture, Tool and Future Possibility,” trong *International Journal for Scientific Research & Development*, Ahmedabad, 2017.
- [6] M. T. Jones, “Models for machine learning,” 2017. [Trực tuyến]. Available: <https://www.ibm.com/developerworks/library/cc-models-machine-learning/index.html>.
- [7] “Intent classification and its significance in chatbot development,” [Trực tuyến]. Available: <https://blog.vsoftconsulting.com/blog/intent-classification-and-its-significance-in-chatbot-development>.
- [8] H. Sharatun, “Topic and intent classification from scratch,” 2018. [Trực tuyến]. Available: <https://medium.com/skyshidigital/topic-and-intent-classifier-from-scratch-83278fb8cf3>.
- [9] Paul, “Natural Language Processing (NLP) Techniques for Extracting Information,” [Trực tuyến]. Available: <https://www.searchtechnologies.com/blog/natural-language-processing-techniques>.
- [10] P. Shrivastava, “Named Entity Recognition: Milestone Models, Papers and Technologies,” 12 09 2017. [Trực tuyến]. Available: <https://blog.paralleldots.com/data-science/named-entity-recognition-milestone-models-papers-and-technologies/>.

- [11] P. Surmenok, “Chatbot Architechture,” 2016. [Trực tuyến]. Available: <https://medium.com/@surmenok/chatbot-architecture-496f5bf820ed>.
- [12] Hubert, “Ways Artificial Intelligence and Chatbots Are Changing Education,” 2017. [Trực tuyến]. Available: <https://chatbotsmagazine.com/six-ways-a-i-and-chatbots-are-changing-education-c22e2d319bbf>.
- [13] “Chat Apps Usage In Vietnam,” Di-Marketing, 2016.
- [14] P. Pal, “12 Benefits of Using Node.js For Web Applications,” 21 10 2016. [Trực tuyến]. Available: <https://think360studio.com/12-benefits-of-using-node-js-for-web-application/>.
- [15] “Webhook,” [Trực tuyến]. Available: <https://en.wikipedia.org/wiki/Webhook>.
- [16] Facebook, “Setting Up Your Webhook,” [Trực tuyến]. Available: [https://developers.facebook.com/docs/messenger-platform/getting-started/webhook-setup/?locale=en\\_US#verify\\_webhook](https://developers.facebook.com/docs/messenger-platform/getting-started/webhook-setup/?locale=en_US#verify_webhook).
- [17] J. Quigley, “Everything You Wanted To Know About package-lock.json But Were Too Afraid To Ask,” 12 08 2017. [Trực tuyến]. Available: [https://medium.com/@Quigley\\_Ja/everything-you-wanted-to-know-about-package-lock-json-b81911aa8ab8](https://medium.com/@Quigley_Ja/everything-you-wanted-to-know-about-package-lock-json-b81911aa8ab8).
- [18] B. Wilcox, ChatScript User’s Manual, 2011.
- [19] “Dialogflow,” [Trực tuyến]. Available: <https://en.wikipedia.org/wiki/Dialogflow>.