

**A CONVERSATIONAL QUESTION AND ANSWER CHATBOT
FOR UNIVERSITY**

By
THAM ZHAO JIN

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONS)
Faculty of Information and Communication Technology
(Kampar Campus)

JANUARY 2020

Student ID: 16ACB01219
Supervised By: DR. PRADEEP A/L ISAWASAN

REPORT STATUS DECLARATION FORM

Title: A CONVERSATIONAL QUESTION AND ANSWER
CHATBOT FOR UNIVERSITY

Academic Session: JAN 2020

I THAM ZHAO JIN
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

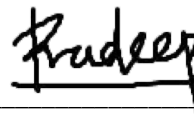
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.



(Author's signature)

Verified by,



(Supervisor's signature)

Address:

52A, JLN PERS ARTISTIK 33/1,
BDR COUNTRY HOMES,
48000 RAWANG, SELANGOR.

PRADEEP ISAWASAN

Supervisor's name

Date: 18 April 2020

Date: 19/04/2020

**A CONVERSATIONAL QUESTION AND ANSWER CHATBOT
FOR UNIVERSITY**

By
THAM ZHAO JIN


A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONS)
Faculty of Information and Communication Technology
(Kampar Campus)

JANUARY 2020

Student ID: 16ACB01219
Supervised By: DR. PRADEEP A/L ISAWASAN

DECLARATION OF ORIGINALITY

I declare that this report entitled “**A CONVERSATIONAL QUESTION AND ANSWER CHATBOT FOR UNIVERSITY**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :  _____

Name : THAM ZHAO JIN

Date : 18 April 2020

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Pradeep A/L Isawasan for the continuous support of my undergraduate study and final year project, for his patience, motivation, enthusiasm and immense knowledge. His guidance helped me in all the time of final year project and writing of this report. He had given me this bright opportunity to engage in an IT project. It is my first step to establish a career in IT field. A million thanks to you.

My sincere thanks also goes to my fellow coursemates – Fan Wei Cong, Kheng Jia Shen, Khiu Zhen Hung, Khoo Chia Hong, Loh Zhan Heng, Ng Jiunn, Soh Lun Hao, Tan Hoai Thang and Tin Ley Ter, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last three years. In particular, I am grateful to all who I had engaged in my undergraduate study.

Last but not the least, I would like to thank my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

University admissions have been increasing over each intake. Students and parents who attend University Open Days for information querying are caught in long queues. In addition, University FAQ sites do not always resolve the problems of students and parents. Research has shown that automating information querying using chatbot can resolve operational inefficiencies and FAQ site ineffectiveness. This project proposes to develop “A Conversational Question and Answer Chatbot for University” to reduce manual work done by University marketing and promotion teams in handling information querying. Based on existing chatbot systems review, the AI Chatbot Development Methodology was adopted, and Naïve Bayes approach was chosen as the NLP Engine for the chatbot development implementation. Deployment of the chatbot demonstrated that the chatbot was able to handle most conversations in a limited domain. Further stabilization is required to improve the NLP Engine for the chatbot so that all possible user interactions could be handled.

TABLE OF CONTENTS

FRONT COVER	i
REPORT STATUS DECLARATION FORM	ii
TITLE PAGE	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	xi
LIST OF TABLES	xiv
LIST OF ABBREVIATIONS	xvi
CHAPTER 1: INTRODUCTION	1
1.1 Problem Statement	1
1.2 Background and Motivation	2
1.2.1 Background Information	2
1.2.2 Motivation.....	4
1.3 Project Scope	5
1.4 Project Objectives	6
1.4.1 Chatbot Module	6
1.4.2 ML Module	8
1.5 Impact, Significance and Contribution	10
1.6 Report Organization.....	11
CHAPTER 2: LITERATURE REVIEW	12
2.1 Introduction.....	12
2.2 Nubitel Nubibot	14
2.2.1 Brief	14

2.2.2 Strengths	15
2.2.3 Weakness	17
2.2.4 Recommendation	17
2.3 Microsoft QnA Maker.....	18
2.3.1 Brief	18
2.3.2 Strengths	19
2.3.3 Weakness	20
2.3.4 Recommendation	21
2.4 Lincoln University AskLive	22
2.4.1 Brief	22
2.4.2 Strength	23
2.4.3 Weakness	24
2.4.4 Recommendation	25
2.5 Tesco Chatbot	26
2.5.1 Brief	26
2.5.2 Strength	27
2.5.3 Weakness	28
2.5.4 Recommendation	28
2.6 Table of Systems Comparison and Discussions	29
CHAPTER 3: SYSTEM DESIGN.....	30
3.1 System Framework	30
3.2 System Flow.....	32
3.3 Use Case Diagram.....	33
3.4 Flow of Events	34
3.5 Activity Diagram	38
3.6 Entity Relationship Diagram.....	40
3.7 Implementation Issues and Challenges.....	41

3.8 Timeline	42
CHAPTER 4: PROPOSED METHOD/APPROACH AND IMPLEMENTATION	43
4.1 Introduction.....	43
4.2 AI Chatbot Development Methodology.....	44
4.2.1 Envisioning	44
4.2.2 Planning	46
4.2.3 Developing.....	50
4.2.4 Deploying.....	52
4.2.5 Stabilization	52
4.3 Technologies and Tools Involved.....	53
4.3.1 Technologies Involved.....	53
4.3.2 Tools Involved	56
4.4 Modules.....	57
4.4.1 Chatbot Module	57
4.4.2 ML Module	67
4.4.3 Others	76
CHAPTER 5: SYSTEM TESTING	78
5.1 Usability Testing.....	78
5.1.1 Chat Module.....	78
5.1.2 Stories Module	80
5.1.3 Login Module.....	82
5.1.4 Intents Module	84
5.1.5 Intents – Create / Update Module	86
5.1.6 Intents – Agents Module	88
5.1.7 Intents – Create / Update Agents Module.....	90
5.1.8 Mail Module.....	92

5.1.9 Mail – View Detailed Module	94
5.2 Unit Testing	96
5.2.1 Chat Module.....	96
5.2.2 Stories Module	98
5.2.3 Login Module.....	100
5.2.4 Intents Module	103
5.2.5 Intents – Create / Update Module	108
5.2.6 Intents – Agents Module.....	112
5.2.7 Intents – Create / Update Agents Module.....	116
5.2.8 Mail Module.....	118
5.2.9 Mail – View Detailed Module	120
5.3 Performance Measure	123
CHAPTER 6: CONCLUSION.....	125
6.1 Project Review, Discussions and Conclusions	125
6.2 Novelties and Contributions	125
6.3 Future Work.....	126
BIBLIOGRAPHY	127

LIST OF FIGURES

Figure 2.1: Nubitel Nubibot Homepage	14
Figure 2.2: Nubibot List of Features.....	15
Figure 2.3: Nubibot Greetings Intent.....	15
Figure 2.4: Enquiring Nubitel Solutions.....	15
Figure 2.5: Expressing Dissatisfaction	16
Figure 2.6: Requesting to Talk to Live Agent	16
Figure 2.7: Microsoft QnA Maker Homepage.....	18
Figure 2.8: QnA Maker Portal	18
Figure 2.9: Creating a Knowledge Base	19
Figure 2.10: Lincoln University Homepage	22
Figure 2.11: Lincoln University Dashboard	23
Figure 2.12: Lincoln University AskLive.....	24
Figure 2.13: Lincoln University AskLive Close-up View.....	24
Figure 2.14: Tesco Malaysia Homepage	26
Figure 2.15: Tesco Chatbot Greetings Intent.....	27
Figure 2.16: Tesco List of Services	27
Figure 2.17: Enquiring the Price of Ice Cream.....	28
Figure 2.18: Enquiring the Price of Ice Cream (cont.)	28
Figure 3.1: Logical Process for System Framework.....	30
Figure 3.2: Chatbot and Emailbot Flow Diagram.....	32
Figure 3.3: Use Case Diagram for Chatbot System.....	33
Figure 3.4: Activity Diagram for Chatbot System.....	39

Figure 3.5: ER Diagram for Chatbot System.....	40
Figure 4.1: MSF Model	44
Figure 4.2: Simplified View of Chatbot Flow Diagram	47
Figure 4.3: Conversation Flow Diagram based on Chatbot Intents.....	49
Figure 4.4: Chatroom for Chat (Web-View)	57
Figure 4.5: Chatroom for Chat (Mobile-View) and Live Chat.....	57
Figure 4.6: Dashboard for Stories (Web-View) – 1.....	58
Figure 4.7: Dashboard for Stories (Web-View) – 2.....	58
Figure 4.8: Dashboard for Stories (Web-View) – 3.....	59
Figure 4.9: Dashboard for Stories (Mobile-View).....	59
Figure 4.10: Login for Staff.....	60
Figure 4.11: Console for Intents	60
Figure 4.12: Diagnostics for Test console	61
Figure 4.13: Delete Intent	61
Figure 4.14: Delete Multiple Intents.....	62
Figure 4.15: Delete All Intents	62
Figure 4.16: Create Intent, Utterance and Response	63
Figure 4.17: Edit Intent, Utterance and Response	63
Figure 4.18: Console for Agents.....	64
Figure 4.19: Edit Agent	64
Figure 4.20: Console for Mail.....	65
Figure 4.21: Detailed Mail – 1	65
Figure 4.22: Detailed Mail – 2.....	66
Figure 4.23: Training Set for Sentiments.....	67

Figure 4.24: Chatlog with Sentiment Scoring	67
Figure 4.25: Emaillog with Sentiment Scoring	68
Figure 4.26: Training Set for Utterances	68
Figure 4.27: Training Set for Intents	68
Figure 4.28: JSON Results for Intent Classification.....	69
Figure 4.29: Preprocessed Email	69
Figure 4.30: Conditional Probability Formula.....	70
Figure 4.31: Bayesian Rule Formula	70
Figure 4.32: Laplacian Smoothing Formula	73
Figure 4.33: Training Set for Chat Spam.....	75
Figure 4.34: Training Set for Email Spam.....	75
Figure 4.35: Setting Up Facebook Page via Facebook for Developers	76
Figure 4.36: Chatting with Quro via Facebook Messenger	77
Figure 5.1: Classification Report for Retrieval-based Model.....	123
Figure 5.2: Confusion Matrix for Retrieval-based Model	124

LIST OF TABLES

Table 2.1: Table of Systems Comparison	29
Table 3.1: Estimated Timeline for Deliverables and Milestones.....	42
Table 4.1: Examples of User Questions.....	45
Table 4.2: Design Ideas of the Chatbot.....	48
Table 4.3: System Testing Activities	51
Table 4.4: Hardware Specification for Laptop.....	53
Table 4.5: Strengths and Weaknesses of Naïve Bayes Approach.....	55
Table 4.6: Tools Involved	56
Table 4.7: Example of Vocabulary List.....	71
Table 5.1: Chat Usability Test Case.....	79
Table 5.2: Stories Usability Test Case	81
Table 5.3: Login Usability Test Case.....	83
Table 5.4: Intents Usability Test Case	85
Table 5.5: Intents – Create / Update Usability Test Case	87
Table 5.6: Intents – Agents Usability Test Case	89
Table 5.7: Intents – Create / Update Agents Usability Test Case.....	91
Table 5.8: Mail Usability Test Case.....	93
Table 5.9: Mail – View Detailed Usability Test Case	95
Table 5.10: Chat Unit Test Case	97
Table 5.11: Stories Unit Test Case.....	99
Table 5.12: Login Unit Test Case	102
Table 5.13: Intents Unit Test Case.....	107

Table 5.14: Intents – Create / Update Unit Test Case.....	111
Table 5.15: Intents – Agents Unit Test Case	115
Table 5.16: Intents – Create / Update Agents Unit Test Case	117
Table 5.17: Mail Unit Test Case	119
Table 5.18: Mail – View Detailed Unit Test Case	122

LIST OF ABBREVIATIONS

<i>AI</i>	Artificial Intelligence
<i>API</i>	Application Programming Interface
<i>cont.</i>	Continued
<i>CSS</i>	Cascading Style Sheets
<i>ER</i>	Entity Relationship
<i>FAQ</i>	Frequently Asked Questions
<i>FBF</i>	Faculty of Business and Finance
<i>FICT</i>	Faculty of Information and Communication Technology
<i>HTML</i>	Hypertext Markup Language
<i>IT</i>	Information Technology
<i>MSF</i>	Microsoft Solutions Framework
<i>NER</i>	Named-Entity Recognition
<i>NLP</i>	Natural Language Processing
<i>NLTK</i>	Natural Language Toolkit
<i>QnA</i>	Questions and Answers
<i>RDBMS</i>	Relational Database Management System
<i>SDLC</i>	Software Development Life Cycle
<i>SQL</i>	Structured Query Language
<i>UAT</i>	User Acceptance Testing
<i>UI</i>	User Interface
<i>URL</i>	Uniform Resource Locator

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

Students and parents who attend University Open Days for information querying are caught in long queues.

University marketing and promotion teams would encounter a bottleneck situation during Open Days. This is because their working procedures are still largely dependent on manual works. By way of explanation, they have only limited manpower and intellectuals to handle information querying among those attending Open Days. In addition, increasing visitor numbers have a significant impact on the workload of marketing and promotion teams. The waiting time and expenses are greatly increased. Hence, visitors could get frustrated easily when waiting in line for a long time. This will lead to operational inefficiencies. In response to this problem, the study of this project involves automating the information querying using chatbot to reduce the workload of marketing and promotion teams during Open Days.

University FAQ sites do not always resolve the problems of students and parents.

Poorly crafted FAQ site has a significant impact on the user experience. This is because the FAQs only cover limited areas. The content on the site lacks expertise because it is wholly designed by someone in the marketing team. Moreover, the selected FAQs do not always reflect on the purpose of visitor queries. Some of them may not even assure of what problems they are facing in the first place. They normally formulate their problems when reading through the FAQs. This is time-consuming and thus lead to FAQ site ineffectiveness. In response to this problem, the study of this project involves incorporating the FAQ site with chatbot to increase the user experience by upholding the purpose of visitor queries.

1.2 Background and Motivation

1.2.1 Background Information

Automation is the innovation done by humans to perform tasks that are deemed too complex or labour-intensive for them to handle manually. Hence, automation has yielded technological advancements to various use cases of industry today. For instance, cost reduction, productivity and availability are the notable contributions to society (Cameron, n.d.).

First and foremost, automation reduces operating costs significantly without diminishing the performance. By way of illustration, automation has slashed the costs by nearly half in the IT field (Nizri 2018). At the same time, automation boosts productivity because it outperforms the humans for such metrics as throughput and human error reduction. Furthermore, automation contributes in maintaining the availability of online systems. It has facilitated automatic backup and recovery services of these systems. These decrease the likelihood of catastrophic events such as permanent data loss and security intrusion. Therefore, automation has become an increasingly commonplace given the fact that various use cases of industry have been benefited from it.

Chatbot has simplified the human-to-human and human-to-machine interactions (Chandrayan 2019). NLP, a component of AI, has empowered the chatbots to understand human natural languages. This is because human natural languages are high-level languages. They are difficult to be interpreted by machines. Human-answered responses act as a benchmark to formulate chatbot responses so that they are relevant to user query. Besides that, chatbots can effectively handle the fast pace of simultaneous conversation online. Various use cases of industry such as customer support, education and hospitality have been proliferated by it. Therefore, chatbots have enhanced the quality of communication experience.

Chatbots will continue to become an inherent part of daily life considering that they facilitate problem-solving and decision-making processes. The popularity of chatbots is growing due to simplicity of building them. Back in 2016, Facebook released the Messenger Platform which encouraged the creation of chatbots. Dialogflow, Wit.ai and other platforms also provide the groundworks of chatbots. For example, modern chatbots such as Siri, Cortana, Google Assistant and Alexa can handle online

conversations to reduce human intervention. Chatbots are normally built to sustain informational conversations so that they can automate repeatedly answers provided by humans. In addition, they can also automate daily procedures of humans such as ordering items and making reservations online to reduce the workload. The future of chatbots and NLP is coming one way or another that is inevitable.

In education use cases, QnA connects the students with academic staff to discuss matters regarding to university studies and general questions. Nevertheless, emails, phone calls and FAQ sites are inefficient QnA approaches that cause annoyance when clarifying student queries. Students would face frustration due to unread emails, busy phonelines and lateness of addressing their problems. These approaches will become unreliable in the future considering that university admissions increase gradually.

1.2.2 Motivation

The motivation of this project is to resolve operational inefficiencies and FAQ site ineffectiveness. Firstly, the chatbot can support informative counters to handle information querying during Open Days. This is because information querying is made to be fast and easy. More visitor queries can be accommodated and resolved at the same time. Hence, the chatbot will be able to overcome limited manpower faced by marketing and promotion teams. Furthermore, the chatbot provides the perfect complement to FAQ site. The visitors will be able to receive specific answers to their respective questions directly from the chatbot. This facilitates information querying by reducing the time and efforts spent on it.

1.3 Project Scope

The final deliverable of this project proposes an automated QnA AI-powered chatbot built for university marketing and promotion teams. The chatbot is easily accessible in an Internet environment. Moreover, the chatbot handles inquiry-oriented conversations and regulates its responses based on user query. For example, the user may ask questions such as “What happens during Orientation Week?” or “What if I fail an exam?”. The chatbot is expected to provide specific answer to corresponding questions, instead of providing lists of documents. Nevertheless, this project is limited to targeted audiences which are students and parents. The context on the conversations only covers queries related to university studies and general questions. In addition, the language supported by the chatbot is limited to English only since it is by far the most common language used by universities in Malaysia. Therefore, the design of the chatbot will be covering Chatbot Module and ML Module. The Chatbot Module holds the conversation online, whereas the ML Module powers the chatbot to understand the purpose of user query.

1.4 Project Objectives

The project objectives are stated as follows:

1.4.1 Chatbot Module

- To develop an automated QnA AI-powered chatbot in reducing the workload of university marketing and promotion teams within the boundary condition of problems related to university studies and general questions.

Intent Classification

The chatbot interfaces with the user to accept the user query as input. Upon receiving new input, the chatbot is expected to identify the purpose of query. The goal of the user trying to complete such as enquiry or problem-solving is identified since the context on the chatbot is limited to inquiry-oriented conversations. Intents are labelled based on the names of possible user intentions. In addition, the intents are trained with utterance patterns and their associated responses that will be used in the subsequent Response Generation stage. This stage will be powered by the ML model.

Knowledge Management

The sets of questions and answers regarding to university studies and general questions are obtained from the marketing and promotion teams. Data preparation activities are carried out to build the dataset that will be used in Intent Classification and Response Generation stages. This is because the dataset constructed is used as data sources to intent, utterance patterns and response. Subsequently, the dataset will be stored in the knowledge base for optimized database querying.

Response Generation

In response to user query, the chatbot must possess the ability to reply. Furthermore, the replies must be coherent according to the context on the conversations. This issue can be tackled using retrieval-based method whereby the most appropriate response is provided. This method simply relies on the Knowledge Management stage which provides the data sources for response. Optimized database request is made to retrieve the utterance patterns beforehand. The most appropriate response is returned to the user by matching the query with related patterns. Response Generation stage enables the chatbot to regulate every answer and thus avoiding inappropriate responses.

Rich Messages

Rich messages are implemented to the chatbot to promote a more engaging conversation experience. This encourages the user to use the chatbot. For example, rich messages are such as text, quick replies, images and buttons. In addition, the implementation of rich messages increases the click-through rate. This reduces the need for the user to type out their query. Hence, intuitive conversation flow can be designed to ensure higher response rate instead of fallbacks.

Emailbot

An emailbot using the chatbot's ML model is built to further extend the services of the chatbot. By reasoning, email is still a default and universal channel of communication considering the rise of messaging applications such as Facebook Messenger, WhatsApp, WeChat, etc. It has always been a headache to integrate all messaging applications unless they provide integration through API calls. Nevertheless, integration with their respective APIs is a troublesome process. Hence, the emailbot will automate the communication by means of email channel. It is simple and familiar way to communicate, send an email – get instant response.

1.4.2 ML Module

- To construct a ML Python framework for the chatbot in resolving the challenges of university FAQ site ineffectiveness to understand the purpose of visitor queries.

Retrieval-Based Model

The prepared dataset is used to construct a ML model for the chatbot. The ML model will power Intent Classification stage to enhance the performance in terms of accuracy. In addition to pattern matching, the intent score is introduced by labelling it to each intent. Hence, greater similarity measure between the user query and patterns will contribute to a higher intent score. This facilitates the chatbot to retrieve the most appropriate response based on the highest intent score.

Sentiment Analysis

Sentiment analysis is implemented into the chatbot and emailbot. This feature analyzes the user query and understand the emotional state behind a series of words and phrases. It is widely used in marketing and customer service sectors. This is because sentiment analysis helps businesses to understand the impact of their products and services. In addition, they can harness the sentiment information to fine tune their business strategies in accordance with the user requirements. For instance, positive sentiments imply that the chatbot and emailbot are performing well; whereas negative sentiments imply that they are performing poorly. Hence, live support agent intervention is needed to resolve this issue.

Custom-Built Personality

This stage refers to the inclusion of small talk in conversations. Briefly, small talk is an informal type of conversations that does not relate to the problem domain. It can cover topics ranged from greetings to art and entertainment. This is because the user might test the chatbot by going off topic and pushing its knowledge to the limit. Appropriate answers should be provided to these queries even though they are not listed in the problem domain. Otherwise, the chatbot will fall flat and thus ending the conversation every time it is tested. In addition, the chatbot is expected to address the user query in a genuine manner. For instance, “Thanks for your patience”, instead of “Sorry, I did not understand” should be returned to the user. Therefore, this gives a feeling of the chatbot is genuinely listening and responding to the user queries.

1.5 Impact, Significance and Contribution

The final deliverable of this project will contribute to the society considering that chatbots leverage information querying in various use cases of industry today. The greater growth of student numbers in universities justifies the necessity of adopting competent QnA approach. Traditional QnA approaches such as emails, phone calls and FAQ sites should be automated using chatbots to accommodate them. By doing so, problem-solving and decision-making processes are facilitated. Hence, universities which apply the approach derived from the outcomes of this project will be able to promote a more effective and efficient QnA experience. This project is therefore intended to introduce the perceived degree of automating QnA approach using chatbot impacts productivity in information querying.

1.6 Report Organization

This report consists of 6 parts. The first part – Chapter 1: Introduction – gives an overview about the problem statement, background and motivation, project scope, project objectives, and impact, significance and contribution. The second part – Chapter 2: Literature Review – is a detailed review of related works on the strengths and weaknesses of their solutions, and recommendation to resolve the weaknesses. The third part – Chapter 3: System Design – gives the top-down level chatbot system diagrams. The fourth part – Chapter 4: Proposed Method/Approach and Implementation – gives an overview about the methodology involved in this project, detailed description on the technology and tools involved, and associated system modules. The fifth part – Chapter 5: System Testing – comprises the Usability Testing and Unit Testing for each UI and system function respectively. The last part – Chapter 6: Conclusion – summarizes the report on project review, novelties and future work.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter provides an overview of current practice or existing research prior to the project. It introduces chatbots for the case study that comprises the primary focus of project described in this report. This is in order to scope out what have other researchers done to resolve the problem, strengths and weaknesses of their solutions, and how these weaknesses can be resolved. Various techniques have been implemented to address these issues. Although the literature review covers various techniques, this review will only focus on five major research themes, namely intent classification, knowledge management, entity recognition, pattern matching and sentiment analysis.

These research themes are studied because they contribute to the fundamentals of chatbots. A collection of secondary resources is acquired to identify related works for review. For example, websites and articles are selected as key bibliographic materials. Analysis of related works establishes the framework for the chatbot. Useful insight derived from the resources helps to determine the functional requirements of the chatbot. Therefore, the goal of this chapter is to study the research themes mentioned above and their association in existing chatbots.

Intent and utterance are repeatedly mentioned in the context on the chatbots (McGrath 2017). Intent represents the purpose of user query. It is defined for each type of use case. Moreover, the user query is better known as utterance that represents the user sayings. For example, the user may ask question such as “What is the weather today?” or “Order a chicken pepperoni pizza from Domino’s”. The utterance will trigger subsequent chatbot action to retrieve appropriate response based on its associated intent. Hence, chatbots are trained with sets of utterance patterns and responses directed to respective intents to handle inquiry-oriented conversations.

Traditional chatbots use pattern matching to classify user query into associated intent based on its related pattern. The patterns are variations of user query in terms of grammar and sentence structure. The more alike between the query and patterns, the more likely the associated response to be retrieved. This kind of chatbots is commonly known as rule-based chatbots (Shridhar 2017). The accuracy of rule-based chatbots depends heavily on the availability and number of patterns trained for each intent. If the query does not match with trained patterns at all, then no response will be provided.

Huge dataset of patterns is required to alleviate this issue. However, the data generation and training phases can be time-consuming and require IT intellectuals to consistently maintain the data model.

There are two types of knowledge base which are machine-readable and human-readable. The literature review will focus on human-readable knowledge base. As its name implies, human-readable knowledge base is accessed by humans to store and retrieve datasets of information. For instance, business use cases have been benefited from it in providing valuable information such as products and services, FAQs, and troubleshooting procedures to their customers. Although this kind of knowledge base is more user friendly, manual works are required to monitor and maintain corresponding information.

NER is a technique applied in various use cases of NLP that extracts and classifies known entities from user query into predefined categories (Banerjee, S 2018). Entities such as name, location, and currency are useful to understand the context on the query. This technique aids in addressing the relevance of real-world problems, namely customer support, search algorithms, and news content classification. By reasoning, NER has contributed in limiting the domain of query and facilitating the collection of insight. Hence, decision-making process will be improved to better satisfy user requirements and achieve business objectives.

2.2 Nubitel Nubibot

2.2.1 Brief

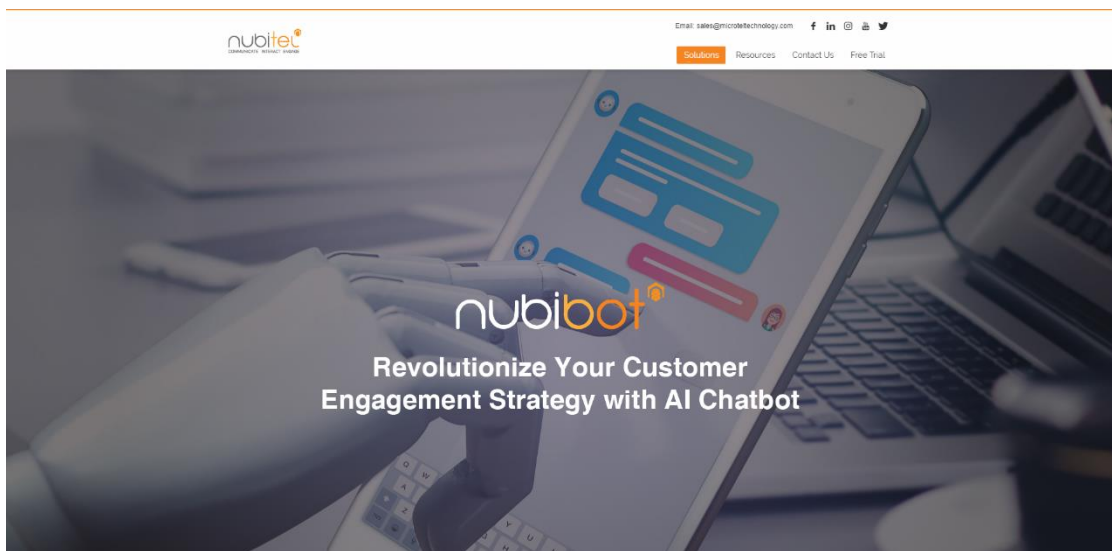


Figure 2.1: Nubitel Nubibot Homepage

Nubitel has developed an AI-powered chatbot, namely Nubibot (nubibot 2019) that implements virtual assistants in business use cases such as contact centre, knowledge base and IT helpdesk. It has contributed three notable benefits which are operating cost reduction, staying relevant and increase of response time. Firstly, Nubibot can reduce the customer service costs significantly without affecting the performance. This is because it has automated the daily operations of contact centre staff. Moreover, Nubibot enables businesses to stay relevant in this rapidly changing world. Staying relevant is the key advantage to maintain competitiveness in the changing business environment. In addition, Nubibot can increase the response rate of contact centre customer support. This facilitates the customer engagement strategies that will lead to customer satisfaction. To summarize, Nubibot allows businesses to promote a seamless customer engagement experience to their customers.

2.2.2 Strengths

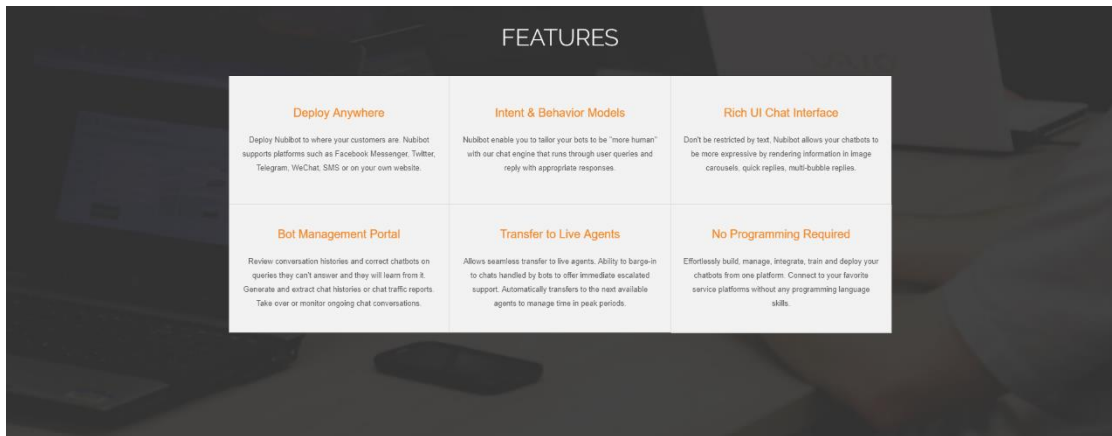


Figure 2.2: Nubibot List of Features

Figure 2.3 and Figure 2.4 show the chat conversations on enquiring Nubitel solutions.

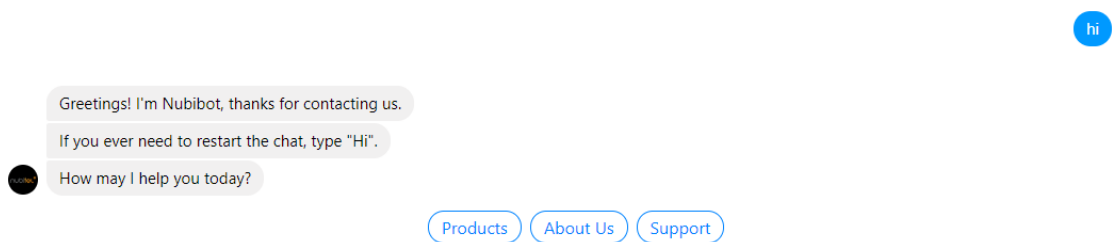


Figure 2.3: Nubibot Greetings Intent

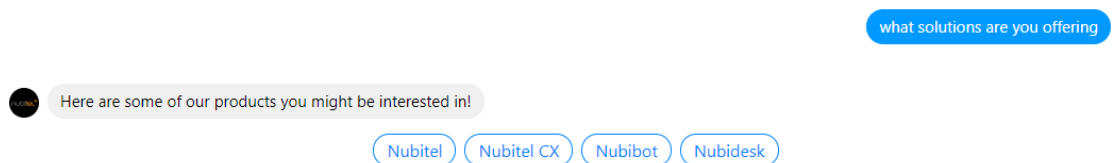


Figure 2.4: Enquiring Nubitel Solutions

The strengths of this chatbot are described as follows. First and foremost, Nubibot has implemented intent classification to determine the intention of customer query. It is trained with sets of questions and answers relevant to respective business use cases. Hence, it can understand what type of questions requires what type of answers. In addition, Nubibot uses pattern matching to classify customer query into appropriate intent. It retrieves the answer that corresponds to the query. This can be done so because the query is related to its associated patterns. Therefore, intent classification has facilitated information querying in contact centres.

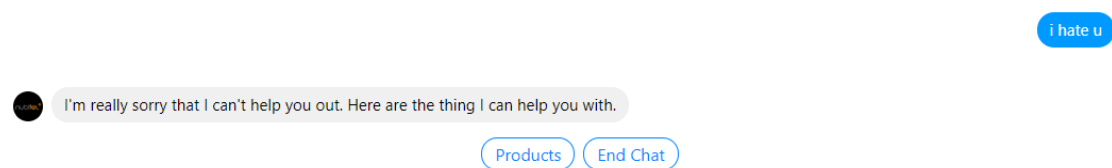


Figure 2.5: Expressing Dissatisfaction



Figure 2.6: Requesting to Talk to Live Agent

Furthermore, Nubibot has also implemented sentiment analysis as the analytical complement to intent classification. This feature measures the quality of customer experience based on a series of words and phrases. By way of explanation, sentiment analysis can process the query to collect insight on the underlying emotional tone. This insight is used to further improve the customer engagement strategies. For instance, positive sentiments are conducive for strengthening customer relationships, whereas negative sentiments are conducive for improving customer service. In addition, sentiment analysis supports Nubibot in transferring the online conversation to a live agent whenever it cannot resolve the customer query. This is in order to prevent enraging the customer with impersonal answer. Therefore, sentiment analysis has employed to cover for Nubibot in text-based emotion analysis and routing to live agents.

2.2.3 Weakness

The weakness of this chatbot is explained as follows. Pattern matching used by Nubibot is prone to inaccuracy due to the number and distribution of utterance patterns. Supposedly, it should be trained with an even distribution of patterns for each intent to avoid bias. For example, variations of “What is i-Akaun?” are trained for Intent A, while variations of “I forgot my i-Akaun password” are trained for Intent B. If Intent A dominates the distribution in terms of pattern numbers, then the result will go astray. When the customer asks, “What is my i-Akaun password?”, Nubibot will retrieve the wrong answer from Intent A even though the corresponding question should be directed to Intent B. By reasoning, it has wrongly labelled the word, “i-Akaun” as the trigger to Intent A. Therefore, it is impossible to maintain an even distribution of utterance patterns and define them for all possible scenarios at the same time.

2.2.4 Recommendation

Pattern matching used by Nubibot can be swapped with retrieval-based approach. This approach performs more intelligent than pattern matching. By way of justification, pattern matching works by matching the customer query with trained utterance patterns based on sequences of words and phrases. This method is bound to fail when no patterns are trained on corresponding query. It is also unable to handle complex query. In contrast, retrieval-based approach works by labelling each intent with a score and the total scores represent the probability of 1. Higher similarity measure between the query and patterns would result in a higher intent score. Thus, the answer with highest intent score will be returned to the customer. The corresponding answer is most likely to suit the context on the conversation. This ensures the customer to receive the best possible answer so that the query is resolved. Nonetheless, retrieval-based approach has plenty of room for improvement because the dataset used is predefined, whereas the human natural languages come with all kinds of possibility.

2.3 Microsoft QnA Maker

2.3.1 Brief

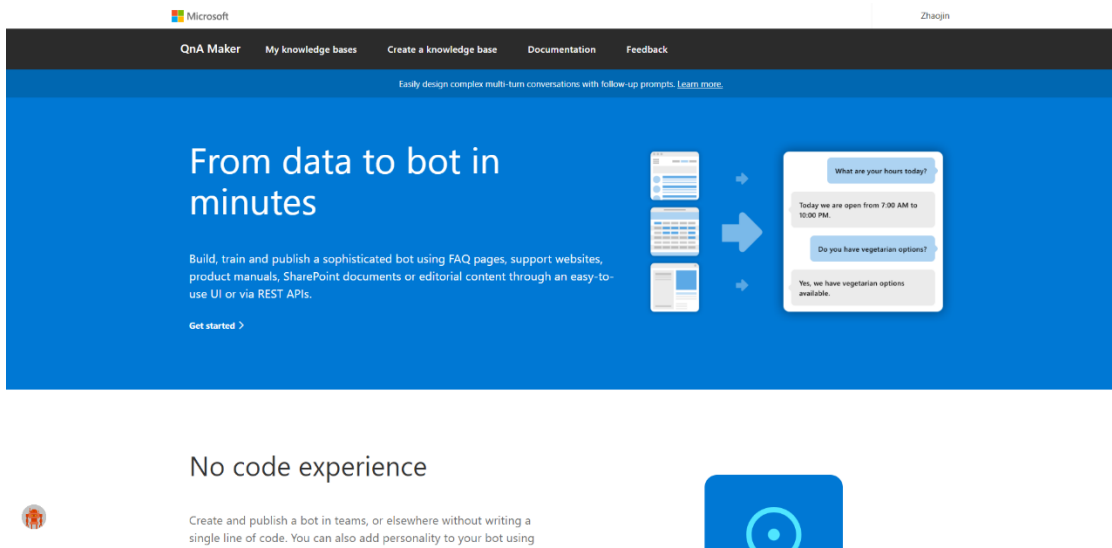


Figure 2.7: Microsoft QnA Maker Homepage

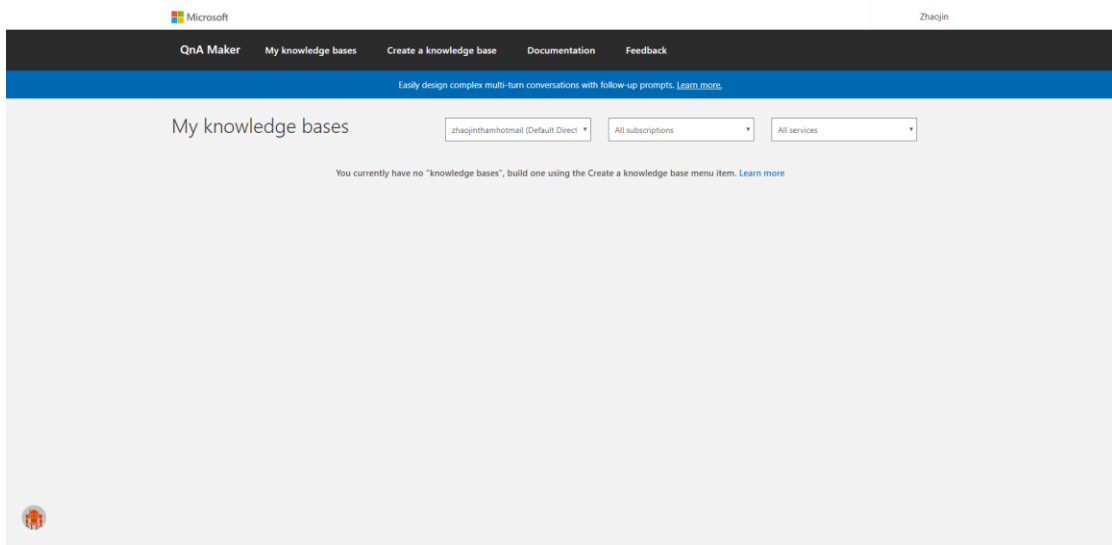


Figure 2.8: QnA Maker Portal

Microsoft has developed a collection of Cognitive Services that provides AI solutions to developers in the form of APIs. QnA Maker (QnA Maker 2019) is one of the Cognitive Services aimed at businesses seeking to improve their FAQ sites. It operates on cloud-based basis by legally scrapping sets of questions and answers from FAQ sites. The questions and answers can be further customized to better suit the voice of brand. Besides scrapping from FAQ sites, it also supports product manuals, brochures and documents. In addition, QnA Maker enables businesses to integrate it into their existing chat applications such as Facebook, Skype and email. Microsoft Bot Platform provides the framework of chatbots to perform corresponding integration. To sum up, QnA Maker allows customers to make enquiries or complaints to businesses by means of online chat.

2.3.2 Strengths

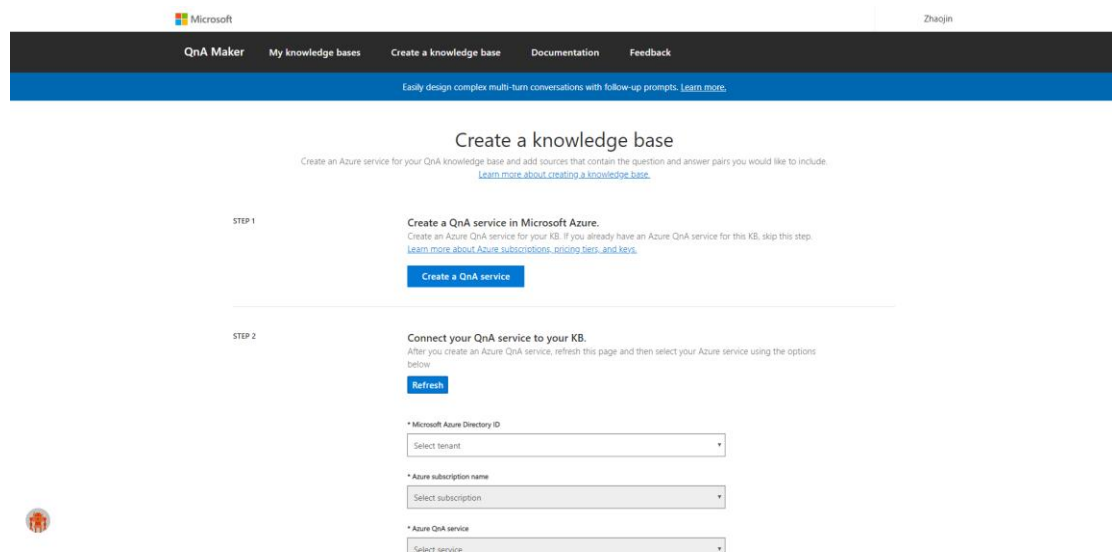


Figure 2.9: Creating a Knowledge Base

The strengths of this service are described as follows. First and foremost, QnA Maker has implemented knowledge management by creating the knowledge base from sets of questions and answers. The knowledge base contributes to the extraction and matching phases of QnA Maker. Extraction phase involves the mapping of sets of questions and answers. The answers are then trained with variations of corresponding questions. Each answer will have respective question list. They are stored in the knowledge base for the next phase usage. Matching phase involves the querying of knowledge base according to similarity between the user question and question list. A confidence score is computed based on the corresponding similarity to aid the matching process. Hence, the user will receive the best possible answer with highest confidence score.

Furthermore, QnA Maker has also implemented entity recognition to further complement the extraction process. By way of explanation, entity recognition can extract and label interested information such as name, location and date. This information is also known as metadata that describes the context of user question. Each question and answer pair will have respective metadata list. The metadata is useful to determine the domain of question so that the subsequent matching phase can be narrowed down. For example, location can differ such as KL and Ipoh when the geographical location of the restaurant branch is different. Thus, the user question will have a variety of answers depending on the metadata tags.

2.3.3 Weakness

The weakness of this service is stated as follows. Pattern matching used by QnA Maker relies too heavily on the sentence structure. Although it has shallow NLP abilities to identify user question that is slightly different than expected, the answer may be incorrect depending on the utterance patterns trained. For example, the question, “Can I bring my vodka?” and associated answer, “Alcohol is not allowed” are listed as question and answer pair. When the user asks, “Can I bring my tea?”, QnA Maker would still respond to it. By reasoning, QnA Maker has wrongly interpreted the phrase, “Can I bring my” as the trigger. Hence, the confidence score fails to aid the matching process despite showing a great deal of similarity. This weakness also applies to the case study of Nubibot such that the number and distribution of utterance patterns have a significant impact on the accuracy of answers returned.

2.3.4 Recommendation

Pattern matching used by QnA Maker should be swapped with retrieval-based approach. Keywords matching can also be implemented to overcome the vulnerability of confidence score. By way of justification, the answer should be trained with multiple variations of user question in terms of grammar style and sentence structure. Each variation includes the keywords which are directed to corresponding answer. For example, “beer”, “wine” and “whiskey” are the keywords representing the alcohol group. Variations of “Can I bring my vodka?” with different alcohol examples can alleviate the inaccurate answer returned. Keywords that are not originated from the alcohol group will not trigger the answer.

2.4 Lincoln University AskLive

2.4.1 Brief

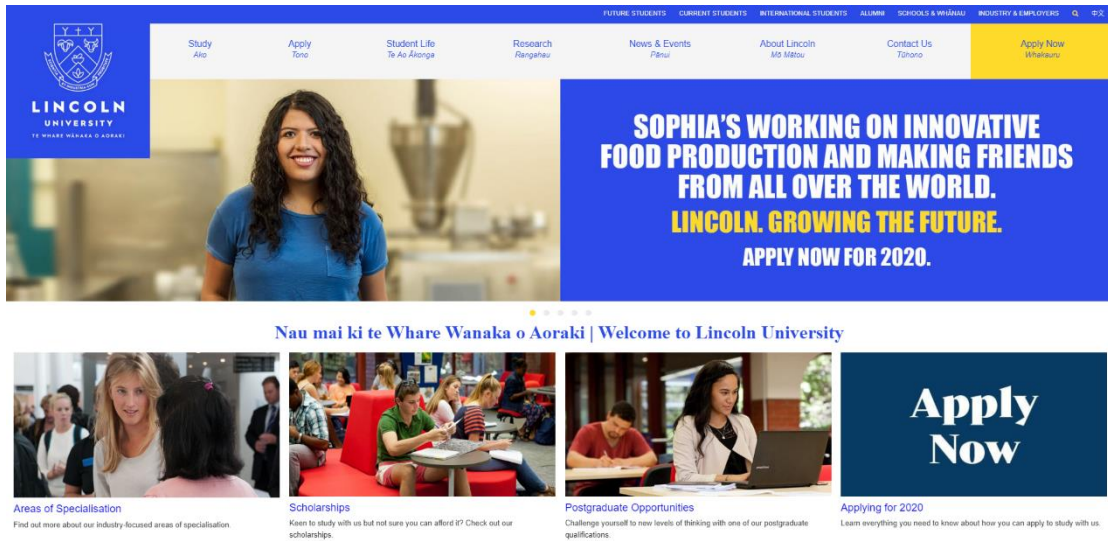


Figure 2.10: Lincoln University Homepage

Lincoln University is a study and research institution based in New Zealand. It offers a range of specialized fields such as agriculture, commerce and environment. In addition, it also offers a wide variety of services and expertise to help students through implementing Library, Teaching and Learning (LTL). For example, the core functions of LTL include study and learning, research, and careers and employment.

2.4.2 Strength

Figure 2.11 shows the embedded AskLive function in the dashboard of Lincoln University.

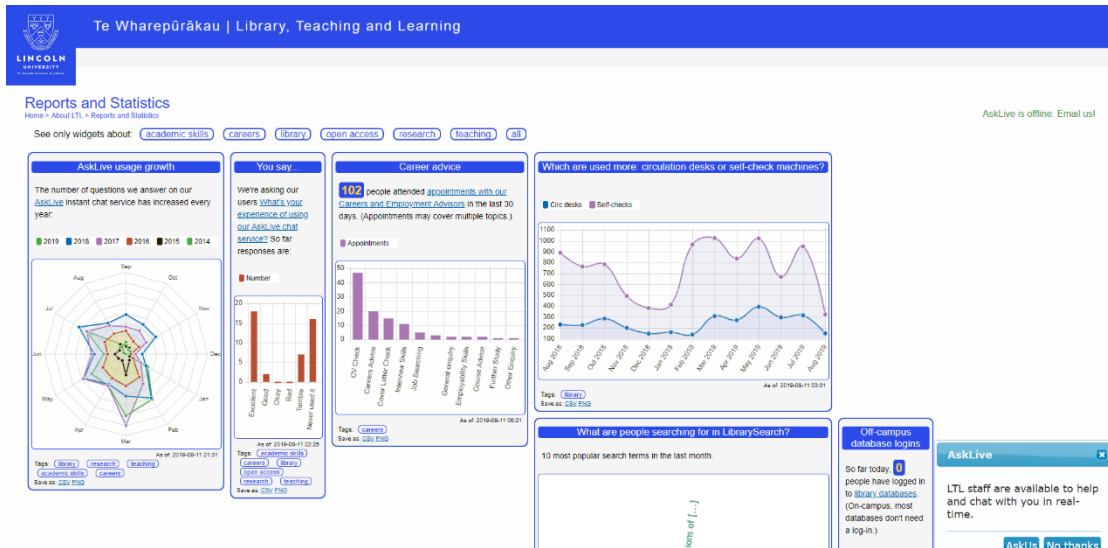


Figure 2.11: Lincoln University Dashboard

The strength of this function is described as follows. AskLive (Lincoln University, n.d.) allows site visitors to chat directly with an available staff online. They can make enquiries or complaints regarding to academics, careers and library resources. Hence, the staff will be able to provide relevant answers to their queries.

2.4.3 Weakness

Figure 2.12 and Figure 2.13 show the chat conversations on greeting.

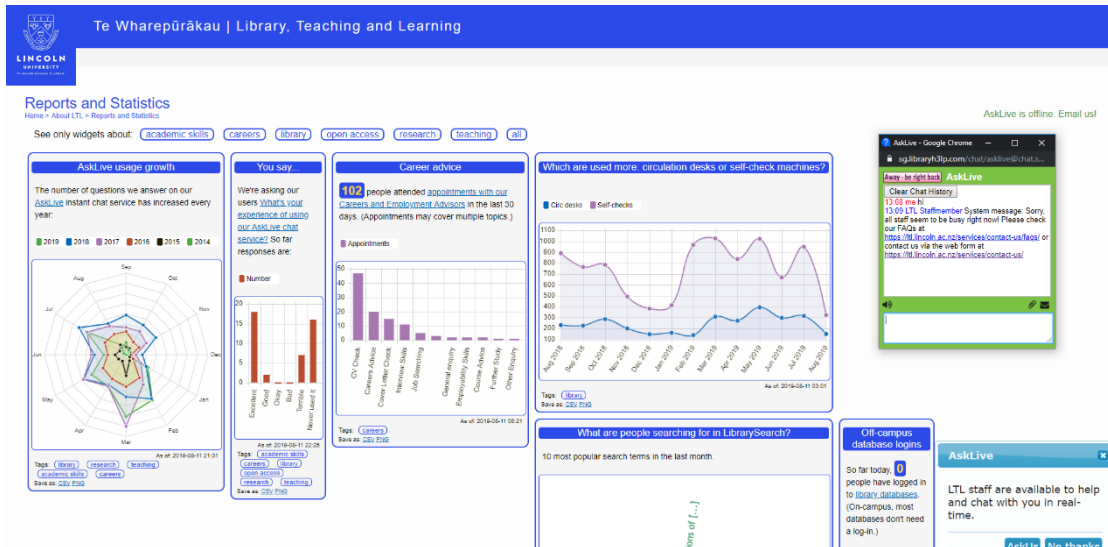


Figure 2.12: Lincoln University AskLive

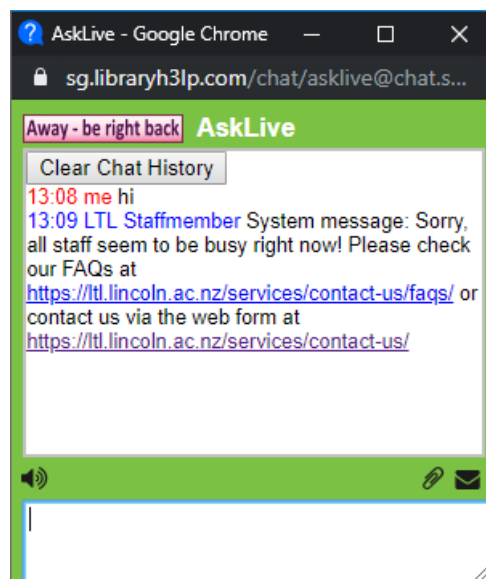


Figure 2.13: Lincoln University AskLive Close-up View

The weakness of this function is stated as follows. Once no staff is available to handle the online chat, the site visitors would get frustrated easily due to delays in responding. For instance, they are redirected to FAQ site to browse through the FAQs manually.

This is time-consuming and thus lead to FAQ site ineffectiveness.

2.4.4 Recommendation

The recommendation to the above-mentioned weakness is discussed as follows. Chatbot should be integrated into AskLive so that they can reduce the workload of staff in handling information querying online. In addition, it guarantees high response rate such that the site visitors do not need to wait for a long time to get their query addressed.

2.5 Tesco Chatbot

2.5.1 Brief

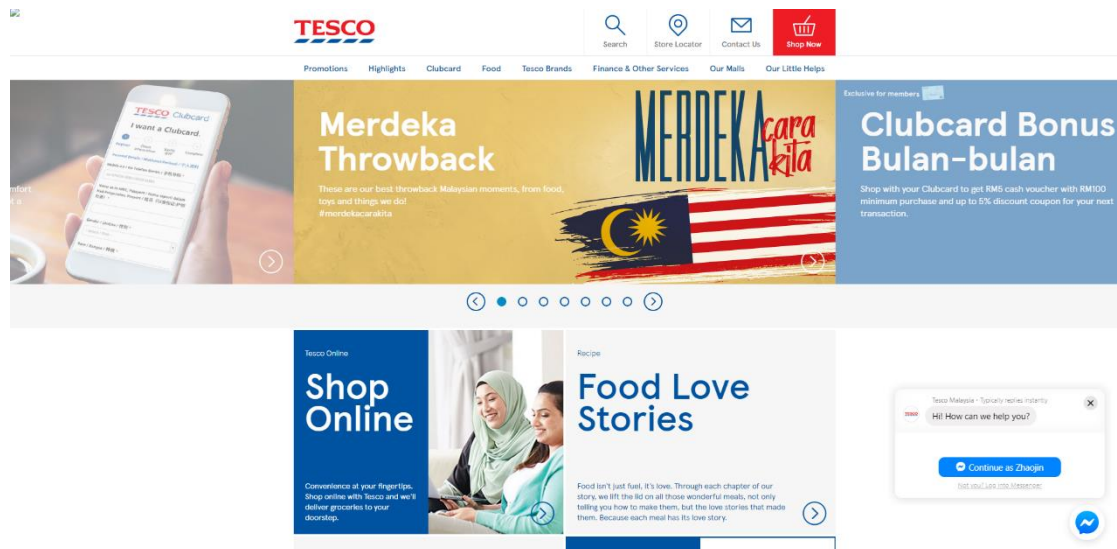


Figure 2.14: Tesco Malaysia Homepage

Tesco Malaysia is a groceries and general merchandise retailer with headquarters based in England, United Kingdom. It provides a diverse retailing of clothing, electronics, furniture, etc.

2.5.2 Strength

MON 16:36

hi



Figure 2.15: Tesco Chatbot Greetings Intent

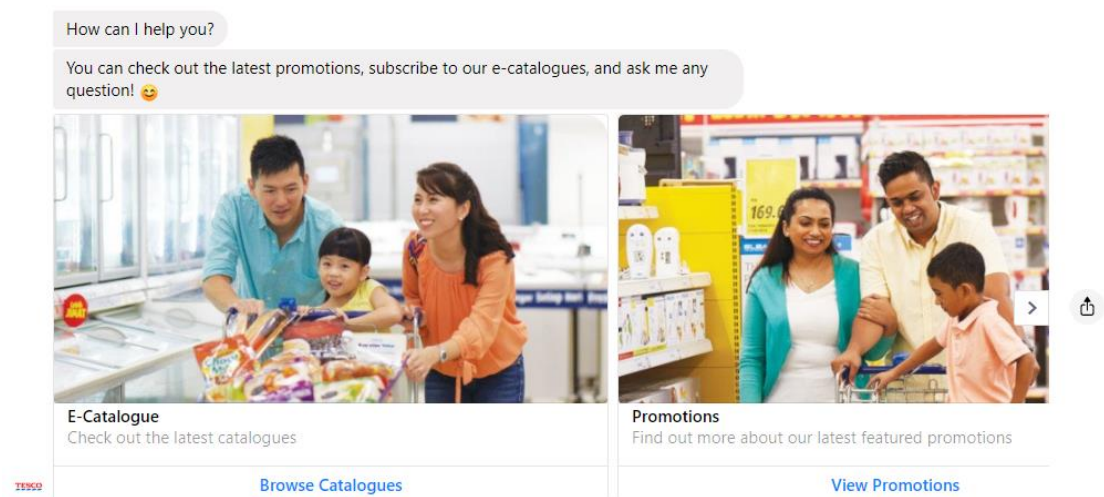


Figure 2.16: Tesco List of Services

The strength of this system is described as follows. Tesco website (Tesco 2017) is integrated with chatbot so that the workload of staff in handling online information querying can be reduced significantly. The chatbot also offers rich messages such as text, image and carousel. This has greatly improved the user experience when interacting with it. In addition, the chatbot converses in such a way that it is cheerful and friendly.

2.5.3 Weakness

Figure 2.17 and Figure 2.18 show the chat conversations on enquiring the price of ice cream.

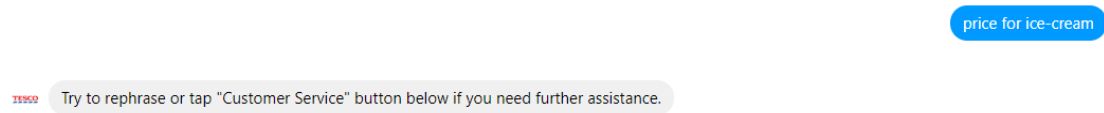


Figure 2.17: Enquiring the Price of Ice Cream

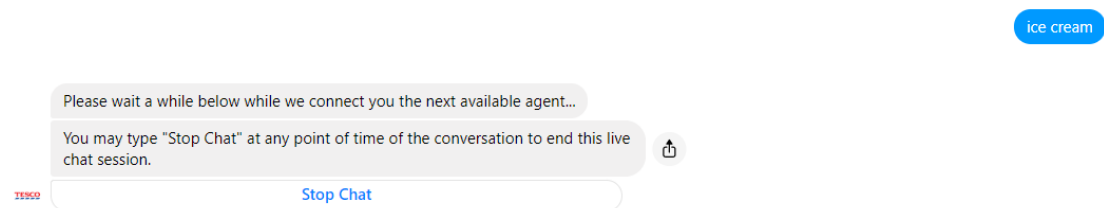


Figure 2.18: Enquiring the Price of Ice Cream (cont.)

The weakness of this system is stated as follows. There is no ML model implemented into the chatbot at all. It fails to capture the intentions of user query and thus providing the wrong response. For example, the chatbot does not understand the enquiry on the price of ice cream. Nevertheless, it is a sensible countermeasure for transferring the current conversation to an available live agent. This is in order to prevent enraging the user with impersonal answer.

2.5.4 Recommendation

The recommendation to the above-mentioned weakness is discussed as follows. Pattern matching should be implemented into the chatbot to capture the intentions of user query. This is because it can retrieve the correct response based on the related utterance patterns. “ice cream” will be the trigger to subsequent chatbot action.

2.6 Table of Systems Comparison and Discussions

Module	Feature/System	Nubibot	QnA Maker	Lincoln University	Tesco Chatbot	Proposed Solution
Chatbot	Intent Classification	✓	✓		✓	✓
	Knowledge Management	✓	✓		✓	✓
	Response Generation	✓	✓		✓	✓
	Rich Messages	✓			✓	✓
	Keywords Matching	✓				
	Entity Recognition	✓	✓			
	Emailbot					✓
ML	Pattern Matching	✓	✓			
	Retrieval-Based Model					✓
	Sentiment Analysis	✓				✓
	Custom-Built Personality	✓	✓		✓	✓
Others	Multiple Languages Support	✓	✓	✓		
	Multiple Platforms Support	✓	✓			✓
	Route to Live Agents	✓		✓	✓	✓

Table 2.1: Table of Systems Comparison

Based on the literature review, the functional requirements of the chatbot have been determined. For instance, fundamental techniques such as intent classification, knowledge management, response generation, rich messages, emailbot, retrieval-based model, sentiment analysis and custom-built personality will be included in the project scope and objectives. Quick replies are implemented as the rich messages feature to streamline the conversation process. In addition, the chatbot supports both web and mobile views so that it is convenient for the users to use it. Facebook is selected as the social platform integration channel. The chatbot can also direct the conversations with users to staff available online.

CHAPTER 3: SYSTEM DESIGN

3.1 System Framework

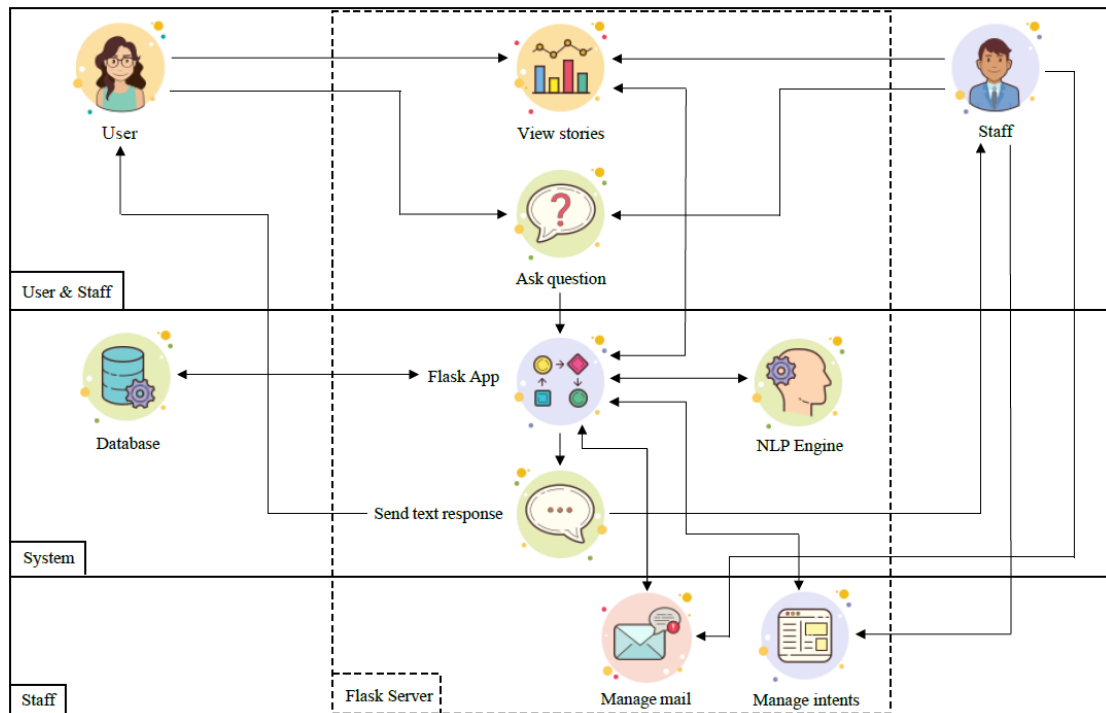


Figure 3.1: Logical Process for System Framework

As shown in Figure 3.1, the Chatbot System Framework depicts the fundamental practices by which QnA process can be fulfilled. The system dynamically manages user privileges such as User & Staff, System and Staff, as well as associated general system activities. First and foremost, the Flask Server houses two main components which are the Flask App and NLP Engine. It is also associated with activities such as View stories, Ask question, Send text response and Manage intents. The system will cover the Flask App and NLP Engine components, Send text response activity, as well as the Database.

Furthermore, both user and staff can view stories and ask question through the Flask Server. The Flask App provides reporting and analytics feature such as recent chats and chat sentiments by means of View stories activity. Next, the question submitted will be passed to the Flask App for text preprocessing. The Flask App queries the Database for training parameters and then passed to the NLP Engine for modelling and training. The training results are used by the Flask App to predict the appropriate intent based on the preprocessed question through Send text response activity. Lastly, the predicted intent

will be used to retrieve the associated text response and then return to the user and staff accordingly.

Moreover, the staff have the privilege to manage intents through the Flask Server. For instance, the staff can create, edit or delete the training parameters such as intents, utterances and responses. The changes will reflect on the Database through the Flask App. In addition, the staff can update the ML model with updated training parameters by sending them to the Flask App. Then, the Flask App will pass them to the NLP Engine for modelling and training.

In addition, the staff also have the privilege to manage mail through the Flask Server. For example, the staff can monitor the incoming emails received from user and outgoing replies sent by the emailbot. An email log is kept in the Database through the Flask App for reference and analysis purposes.

3.2 System Flow

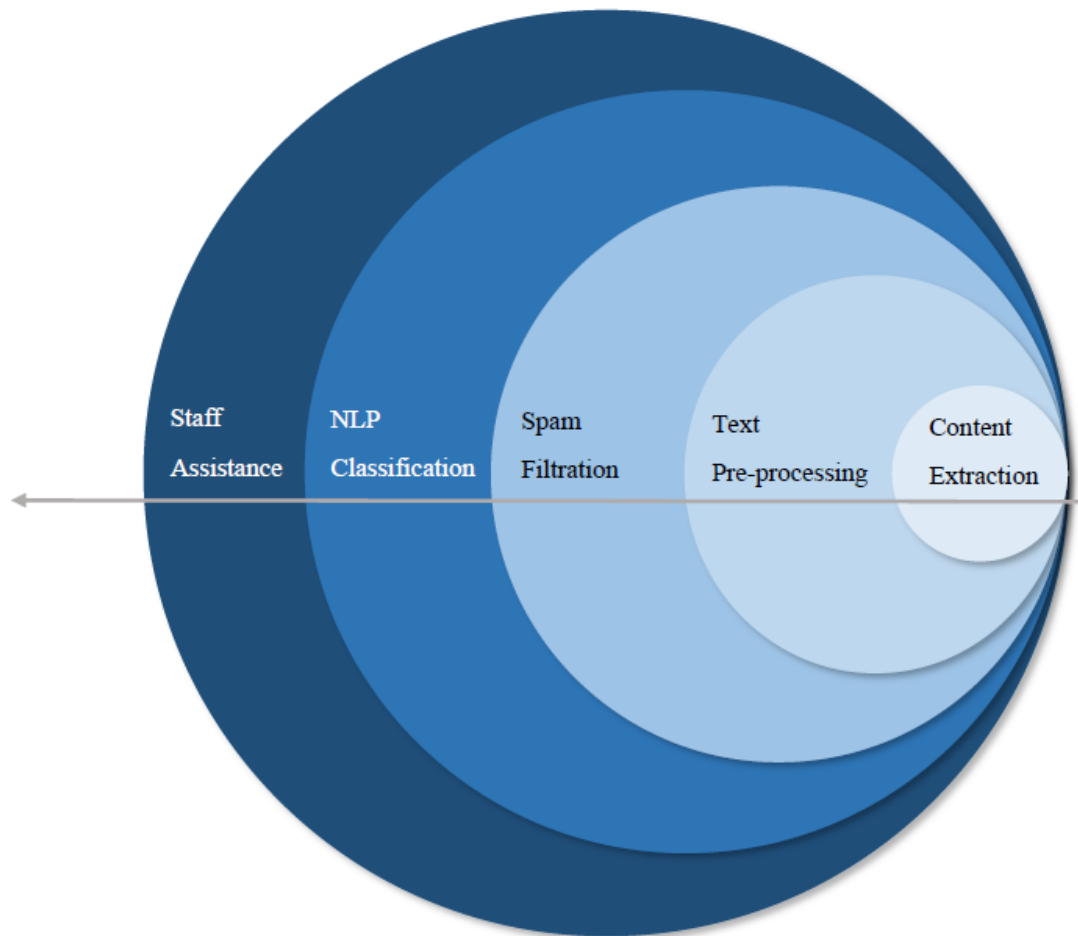


Figure 3.2: Chatbot and Emailbot Flow Diagram

Based on Figure 3.2, the Chatbot and Emailbot Flow Diagram depicts the layers that a user query went through to retrieve the appropriate response. Noted that Staff Assistance acts as a last resort. For instance, Live Chat feature enables staff to intervene the conversations with users; whereas Staff Follow-up Mail feature enables staff to send follow-up emails to users. It is important for staff intervention when the chatbot or emailbot is not performing as the users desired.

3.3 Use Case Diagram

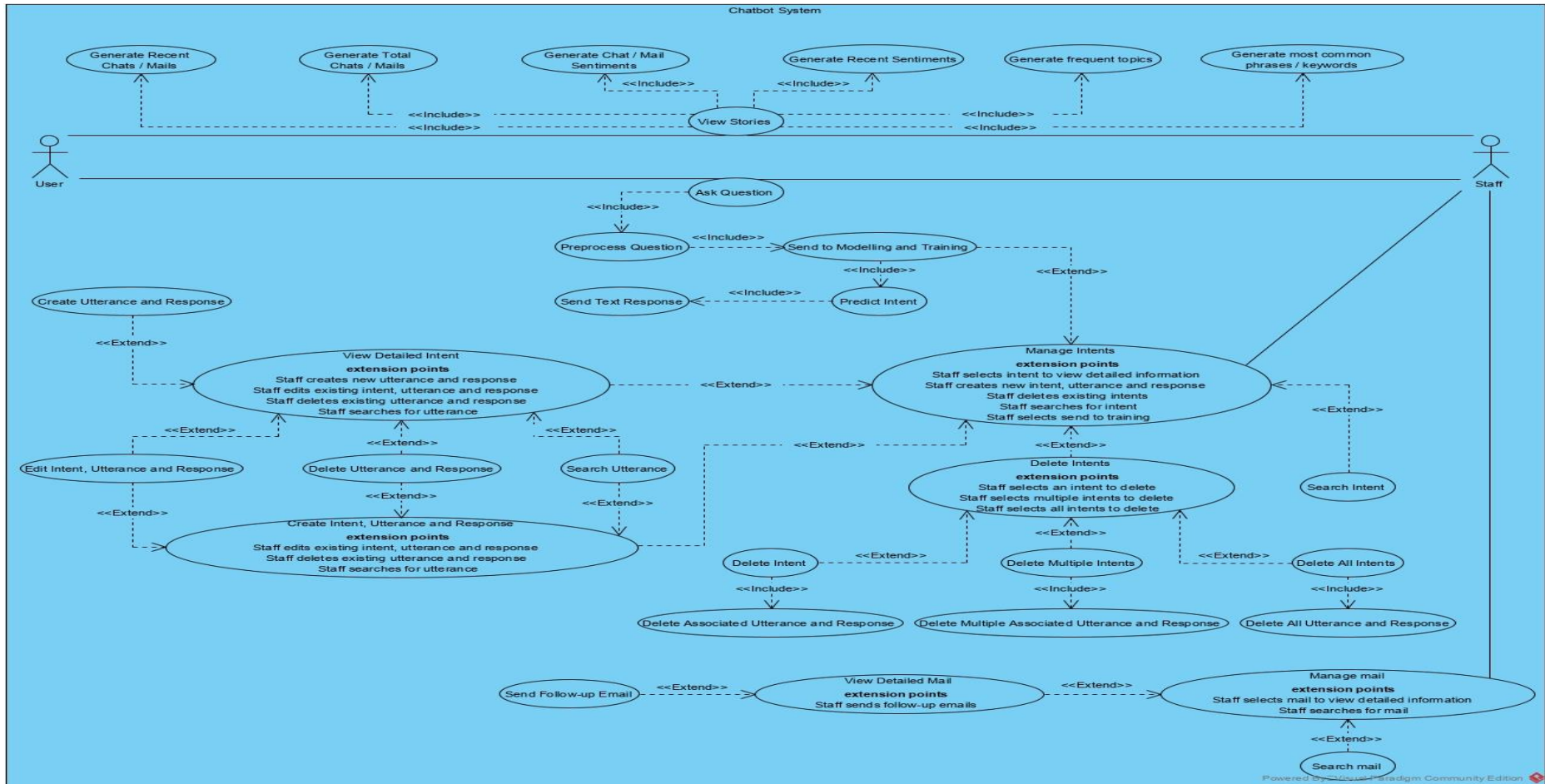


Figure 3.3: Use Case Diagram for Chatbot System

3.4 Flow of Events

Normal Flow of Events:

1. User and Staff can choose whether to ask question or view stories, whereas only Staff can manage intents.

If User and Staff selects Stories:

1. System generates recent chats, total chats, chat sentiments, recent sentiments, frequent topics and most common phrases charts.
2. System displays View stories screen.

If User and Staff selects Home:

1. System displays Ask question screen.
2. User and Staff input question.
3. System preprocesses question.
4. System retrieves training parameters from Database and send them to NLP model for modelling and training.
5. System predicts intent based on the preprocessed question.
6. System retrieves text response from Database based on predicted intent.
7. System returns text response to User and Staff.

If Staff selects Intents,

the S-1: Manage intents subflow is performed.

If Staff selects Mail,

the S-2: Manage mail subflow is performed.

Subflow of Events:

S-1: Manage intents

1. System displays Manage intents screen.
2. Staff can choose whether to view detailed intent, create intent, utterance and response, delete intent, search intent or send to modelling and training.

If Staff selects an intent,

the S-1-1: View detailed intent subflow is performed.

If Staff clicks CREATE button,
the S-1-2: Create intent, utterance and response subflow is performed.

If Staff selects intent(s) to delete,
the S-1-3: Delete intents subflow is performed.

If Staff inputs intent name in search bar:

If intent match found,
System displays intent(s) match found.

If no intent match found,
System displays none.

If Staff clicks TRAIN button,
System retrieves updated training parameters from Database and send them to NLP model for modelling and training.

S-2: Manage mail

1. System displays Manage mail screen.
2. Staff can choose whether to view detailed mail or search mail.

If Staff selects a mail,
the S-2-1: View detailed mail subflow is performed.

If Staff inputs mail name in search bar:

If mail match found,
System displays mail(s) match found.

If no mail match found,
System displays none.

S-1-1: View detailed intent

1. Staff can choose whether to create utterance and response, edit intent, utterance and response, delete utterance and response, search utterance or save intent, utterance and response.

If Staff inputs utterance and response,
System creates new utterance and response views.

If Staff modifies intent, utterance and response,
System updates existing intent, utterance and response views.

If Staff selects utterance and response to delete,
System removes existing utterance and response views.

If Staff inputs utterance name in search bar:
If utterance match found,
System displays utterance(s) match found.
If no utterance match found,
System displays none.

If Staff clicks SAVE button:
If no missing values found,
System stores intent, utterance and response into
Database.
If missing values found,
System prompts Staff to input values into missing fields.

S-1-2: Create intent, utterance and response

1. Staff can choose whether to edit intent, utterance and response, delete utterance and response, search utterance or save intent, utterance and response.

If Staff modifies intent, utterance and response,
System updates existing intent, utterance and response views.

If Staff selects utterance and response to delete,
System removes existing utterance and response views.

If Staff inputs utterance name in search bar:
If utterance match found,
System displays utterance(s) match found.
If no utterance match found,
System displays none.

If Staff clicks SAVE button:
If no missing values found,
System stores intent, utterance and response into
Database.
If missing values found,
System prompts Staff to input values into missing fields.

S-1-3: Delete intents

1. Staff can choose whether to delete intent, delete multiple intents or delete all intents.

If Staff selects an intent to delete,

System prompts Staff for delete confirmation.

If Staff accepts delete confirmation,

System deletes intent and associated utterance and response.

If Staff declines delete confirmation,

System displays Manage intents screen.

If Staff selects multiple intents to delete,

System prompts Staff for multiple delete confirmation.

If Staff accepts multiple delete confirmation,

System deletes multiple intents and associated utterance and response.

If Staff declines multiple delete confirmation,

System displays Manage intents screen.

If Staff selects all intents to delete,

System prompts Staff for delete all confirmation.

If Staff accepts delete all confirmation,

System deletes all intents and associated utterance and response.

If Staff declines delete all confirmation,

System displays Manage intents screen.

S-2-1: View detailed mail

1. Staff can send follow-up mail.

If Staff clicks SEND button:

If no missing values found,

System sends follow-up mail to current User and stores follow-up subject and follow-up body into Database.

If missing values found,

System prompts Staff to input values into missing fields.

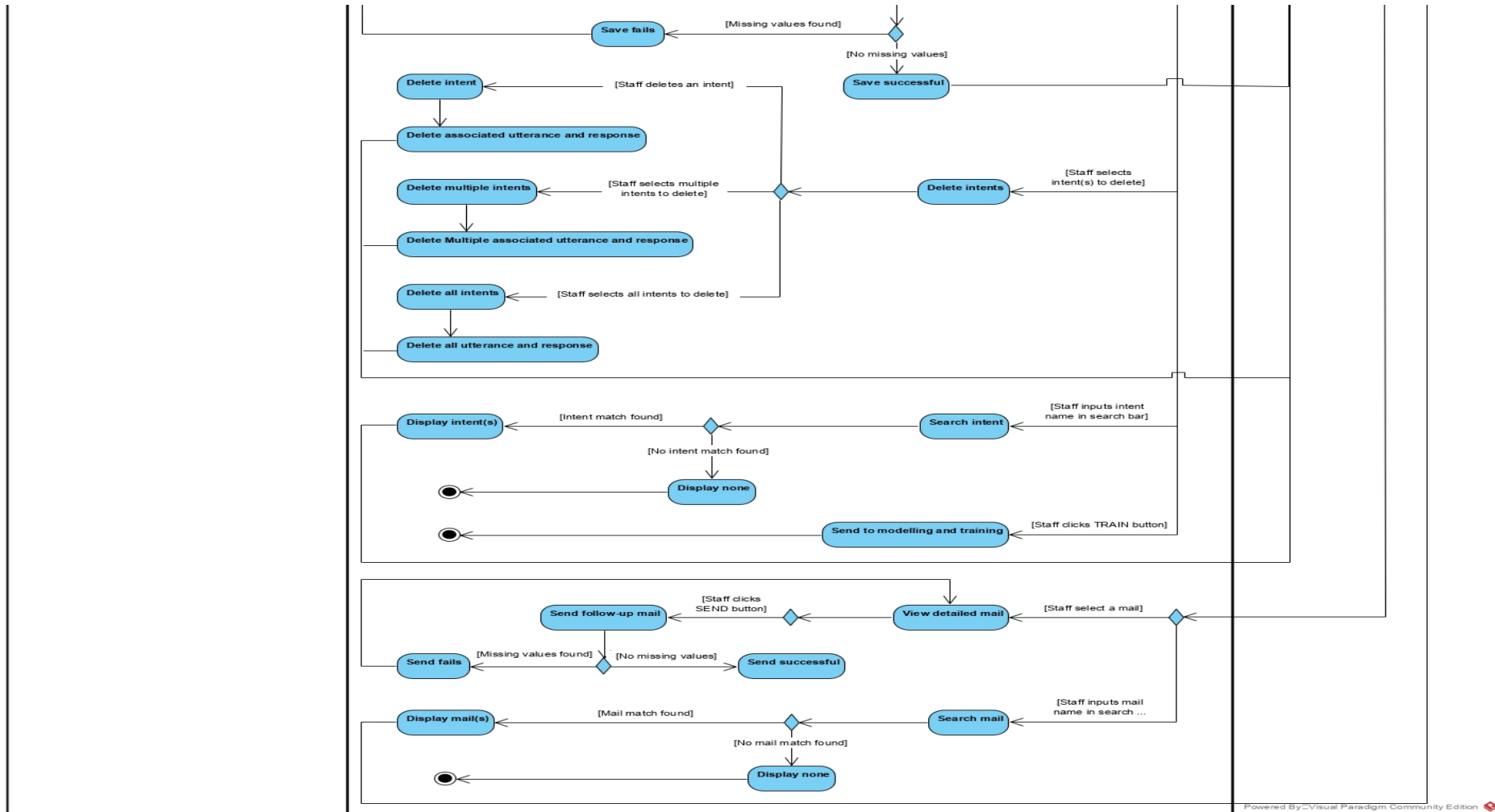


Figure 3.4: Activity Diagram for Chatbot System

3.6 Entity Relationship Diagram

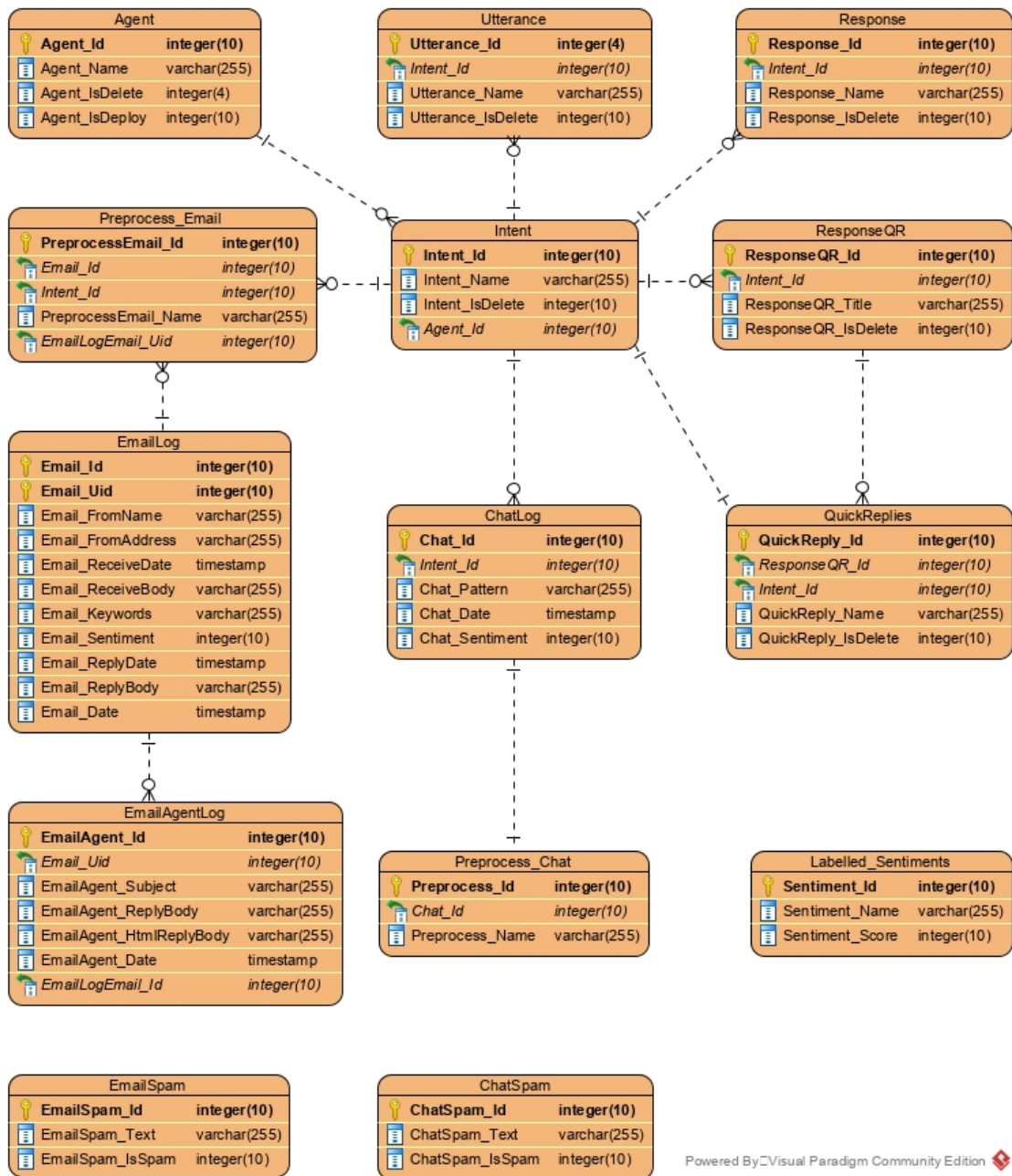


Figure 3.5: ER Diagram for Chatbot System

3.7 Implementation Issues and Challenges

The implementation issues encountered are described as follows. Tensorflow was selected as the backend to Keras, a Python Deep Learning library. Nevertheless, Tensorflow cannot run on Python ≥ 3.7 . In addition, Keras was unable to run in Flask, a micro web framework written in Python on debug mode. Furthermore, the implementation challenges faced are discussed as follows. The dataset for both sentiments and training parameters such as intents, utterances and responses are small and less variations. It is not suitable for Deep Learning approach. Hence, the chatbot ML model was changed from Keras to Naïve Bayes approach. Moreover, the intents should not have identical words or phrases which made up the utterance. Otherwise, the intent prediction might go astray, resulting in the wrong response being returned to the user and staff.

3.8 Timeline

Task	Start Date	Week(s) To Complete	End Date	Start Date	Week(s) To Complete	End Date	16/10/2019	23/10/2019	30/10/2019	6/11/2019	13/11/2019	20/11/2019	15/1/2020	22/1/2020	29/1/2020	5/2/2020	12/2/2020	19/2/2020	26/2/2020	4/3/2020	11/3/2020	18/3/2020	25/3/2020	1/4/2020	8/4/2020	15/4/2020		
Envisioning																												
Define Project Title	16/10/2019	1	23/10/2019				█																					
Define Problem Statement and Motivation	16/10/2019	1	23/10/2019				█																					
Define Project Scope and Objectives	16/10/2019	1	23/10/2019				█																					
Define Impact, Significance and Contribution	16/10/2019	1	23/10/2019				█																					
Define Background Information	16/10/2019	1	23/10/2019				█																					
Define Chatbot Role	16/10/2019	1	23/10/2019				█																					
Define Chatbot Scenario Scope	16/10/2019	1	23/10/2019				█																					
Planning																												
Define System Requirement	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
Literature Review	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
Determine Methodology	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
Determine Technologies and Tools Involved	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
System Framework	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
Use Case Diagram	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
Flow of Events	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
Activity Diagram	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
Entity Relationship Diagram	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
Determine Conversation Flow	23/10/2019	1	30/10/2019					█																				
Determine Chatbot Character	23/10/2019	1	30/10/2019					█																				
Plan NLP Engine approach	23/10/2019	1	30/10/2019	15/1/2020	2	29/1/2020		█					█															
Developing																												
Database Configuration	30/10/2019	2	13/11/2019	29/1/2020	6	11/3/2020			█					█														
UI Design	30/10/2019	2	13/11/2019	29/1/2020	6	11/3/2020			█					█														
Coding	30/10/2019	2	13/11/2019	29/1/2020	6	11/3/2020			█					█														
Chatbot Training	30/10/2019	2	13/11/2019	29/1/2020	6	11/3/2020			█					█														
System Testing	30/10/2019	2	13/11/2019	29/1/2020	6	11/3/2020			█					█														
Conversation Testing	30/10/2019	2	13/11/2019	29/1/2020	6	11/3/2020			█					█														
Deploying																												
Deployment	13/11/2019	1	20/11/2019	11/3/2019	1	18/3/2019					█											█						
Stabilizing																												
Coding	13/11/2019	1	20/11/2019	11/3/2020	5	15/4/2020					█																	
Corrective Maintenance	13/11/2019	1	20/11/2019	11/3/2020	5	15/4/2020					█																	
Perfective Maintenance	13/11/2019	1	20/11/2019	11/3/2020	5	15/4/2020					█																	
Chatbot Training	13/11/2019	1	20/11/2019	11/3/2020	5	15/4/2020					█																	
System Testing	13/11/2019	1	20/11/2019	11/3/2020	5	15/4/2020					█																	
Conversation Testing	13/11/2019	1	20/11/2019	11/3/2020	5	15/4/2020					█																	

Table 3.1: Estimated Timeline for Deliverables and Milestones

CHAPTER 4: PROPOSED METHOD/APPROACH AND IMPLEMENTATION

4.1 Introduction

This chapter provides an overview of methodology and technologies used for the realization of the project. This project will be adopting the AI Chatbot Development Methodology considering that its circular sequence life cycle promotes understandability and ease of use (Luzhanskiy, n.d.). It follows the MSF Model which breaks a project into five stages, namely Envisioning, Planning, Developing, Stabilizing and Deploying (Microsoft 2002).

The project scope and objectives set the project boundaries that describe what is included in the project and what is not. This facilitates the requirements gathering process to ensure the user requirements are within the project scope and minimize the changes during development. The project duration would span across two semesters. Hence, the estimated timeline for deliverables and milestones is drafted prior to next semester so that the development can be accommodated into the schedule of corresponding semester. In addition, documentation of each phase is done prior to next semester due to time limitations in development. The client will have the opportunity to preview the system because approximately 30% of final deliverable is expected to be completed at the end of first semester. Therefore, a fully workable system is expected to be delivered to the client at the end of second semester.

4.2 AI Chatbot Development Methodology

Figure 4.1 shows the AI Chatbot Development Methodology based on MSF Model.



Figure 4.1: MSF Model

4.2.1 Envisioning

The aim of Envisioning phase is to communicate with marketing and promotion teams to gather and analyze user requirements. This phase is consisted of two different activities, namely Requirements Gathering and Analysis, and Requirements Specification (Basics of Software Life Cycle, n.d.).

First and foremost, Requirements Gathering activity is done to capture all user questions relevant to the problem domain that the students and parents may ask. Corresponding answers may be included as well. Questions which are not cover in the project scope would be trimmed. This is in order to clearly define the requirements before proceeding to the subsequent phase so that changes made during development can be minimized. Moreover, the functional and non-functional requirements which describe the behavior of the chatbot are determined. For example, the functional requirements are the user questions mentioned above with the inclusions of UI requirements, chatbot attributes and database requirements. In contrast, the non-functional requirements are imposed on the design and implementation of the chatbot

that include properties such as availability, reliability and performance. Hence, the industrial-strength operation of the chatbot can be realized.

Furthermore, Requirements Specification activities which encompass documentation and data preparation will be done when all the requirements are properly understood. During documentation activity, the requirements are systematically arranged and organized into this report. In addition, data preparation activity takes place to construct the dataset that will be used in the subsequent Planning stage. For instance, data preparation activities such as data cleaning, combining data from multiple sources and transforming data into more useful variables are performed. The data model constructed is expected to meet the database schemes that will be used in this project. The technologies and tools involved in this project are identified at this stage.

The user questions are the subsets of the main user problem to address which is “What user questions should the chatbot be able to handle?”. The corresponding questions should be kept simple and within the project scope so that the chatbot would be able to resolve them without any human intervention. Hence, a few user questions are stated as follows.

User Question	Description
Programs Offered	- The user would like to know the programs which are offered on a full-time basis and, evening and weekend basis.
Admission	- The user enquires about application, supporting documents and status of admission.
Entry Requirements	- The user would like to know the general requirements for entry.
Intakes	- The user enquires about intake periods.
Fees	- The user would like to know the payment mode of program fees.

Table 4.1: Examples of User Questions

In addition, the roles of the chatbot are identified in this stage, namely Informant and Helper. For instance, Informant facilitates data search from different sources about the interested services, whereas Helper assists users in problem-solving process and answers on FAQ. In this project, the data sources will be limited to university domain. Besides that, the business cases of the chatbot development implementation are defined. The chatbot should optimize information management and automate manual work. Therefore, better user engagement can be promoted.

4.2.2 Planning

The goal of Planning stage is to transform the user requirements specified in Requirement phase into the design of the chatbot while taking the data model constructed into consideration. This phase is comprised of Logical and Physical Designs. The designs are very important because they define the Developing stage and outcomes from the final deliverable. In addition, the chatbot architecture is illustrated before proceeding to the corresponding designs. The project scope and objectives are also illustrated in the designs.

Firstly, the Logical Design of the chatbot is prepared by illustrating top-down system design diagrams. For example, the corresponding diagrams are such as chatbot flow diagram, UI design, classes and methods used, and database schematic diagram. Hence, a simplified chatbot flow diagram is illustrated as follows.

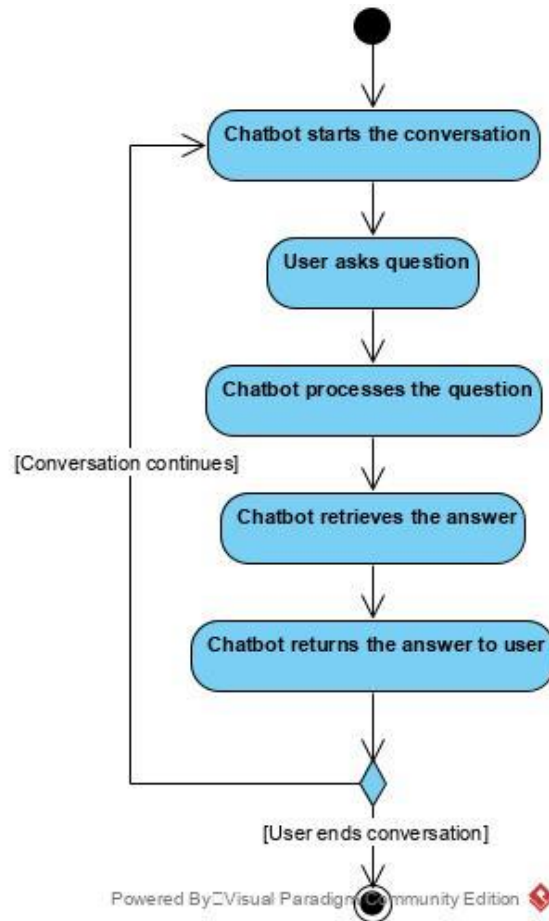


Figure 4.2: Simplified View of Chatbot Flow Diagram

Furthermore, the Logical Design of the chatbot is converted into the Physical Design while taking the hardware and software technologies into consideration (Wong, n.d.). A few design ideas with respect to the project scope and objectives are listed as follows.

Scope	Design Ideas
Intent Classification	- Automatically extract information from the user question.
Retrieval-Based Model	- Process extracted information into a suitable format that meets the ML model. - Prepare a ML pipeline that fits specifically to the format of the extracted information.
Intent Classification, Knowledge Management, Response Generation and Retrieval-Based Model	- Retrieve and return response to the user in real-time.
Custom-Built Personality	- Chatbot conversations reflect on the voice of university. - Support enjoyable and lighthearted conversations for social engagement.

Table 4.2: Design Ideas of the Chatbot

Nonetheless, the design ideas must adhere to certain constraints so that they are successfully translated into source code. For instance, the Logical and Physical Designs of the chatbot should be implemented module-by-module to allow subsequent individual testing. This is because they set up the groundworks of the chatbot for the sequential development. Thus, the designs of the chatbot are carried forward to the subsequent Developing stage.

In addition, the conversation flow of the chatbot is built in this stage so that it is straightforward and can handle exceptions. Besides that, identification of intents is performed to label user questions as shown in Table 4.1. Technical intents such as welcome intent and fallback intent are default existing in the chatbot. For example, fallback intent is triggered in the event of the chatbot has low confidence level in its understanding. Hence, the chatbot should fallback in politely manner without enraging users with wrong responses. Moreover, the chatbot is named as “Quro” and given with a cheerful personality. This is because the chatbot character will reflect on the

university branding as the users communicate with it. An NLP engine is planned which powers the chatbot to handle conversations.

Figure 4.3 illustrates the conversation flow diagram constructed for the chatbot.

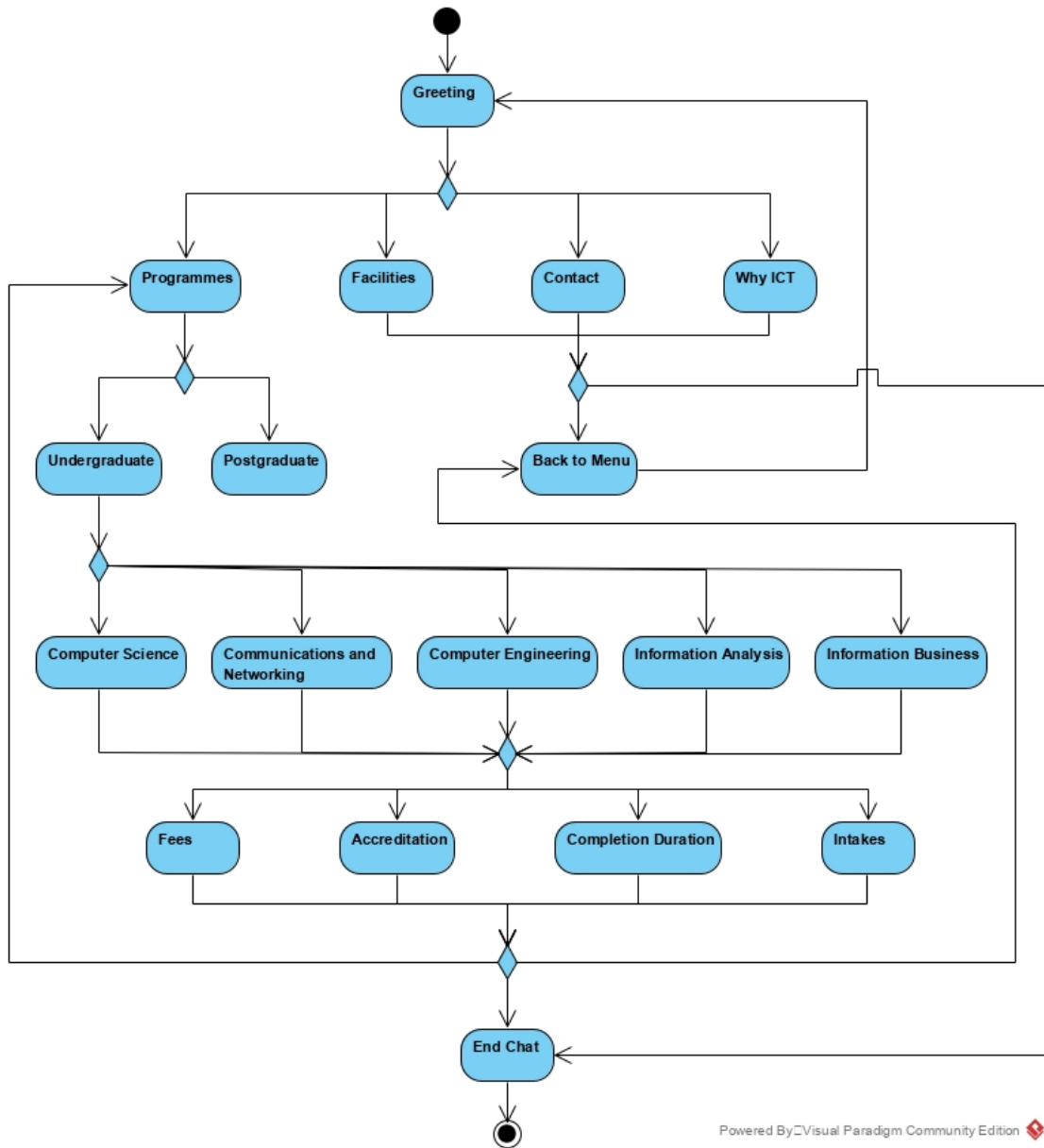


Figure 4.3: Conversation Flow Diagram based on Chatbot Intents

4.2.3 Developing

The purpose of Developing stage is to realize the design ideas by coding the actual program (Bassil 2012). Based on the documents prepared in Envisioning and Planning stages, the functional requirements of the chatbot are implemented. The technologies and tools identified in Envisioning stage are used to implement them. For example, an executable program, database and UI design are the main considerations when coding the chatbot. This stage is where the chatbot is prepared for the production environment.

In addition, the content for chatbot conversations is written in this stage. The user questions defined in Requirement/Envisioning are transformed into an approach for how to create the content to meet university marketing and promotion goals and solve the problems that students and parents will meet. The answers are customized so that they reflect on the personality of the chatbot. Besides that, the NLP engine is trained with a list of utterances for each intent. This is one of the most challenging tasks whereby all possible ways of communicating with the chatbot should be considered. For instance, finding out the correct utterances representing the intents while not having experience on how a user will communicate with the chatbot could be a problem. The utterances should not be identical between intents. Otherwise, the accuracy and performance of the chatbot will be affected due to the intents having similar training parameters clashing with each other.

Test Plans consisting the Usability Testing and Unit Testing are drafted to examine whether all aspects of the chatbot are performing as expected. This stage acts as the final checkpoint before deploying the chatbot to the production environment. Firstly, Usability Testing is done to learn if the users achieve their goal in chatbot conversations, how satisfied the users are when interacting with the chatbot and identifying possible modifications to improve user experience. For instance, the chatbot should be able to resolve user questions, while simulating the real-world chat context such as Messenger and WhatsApp. In addition, Unit Testing is done to evaluate the performance of each module in isolation. Based on the project scope and objectives, each feature should be performing as intended to address the problem statement specified in this project. Bugs and errors are found, debugged and fine-tuned accordingly at this stage. Once the Usability Testing and Unit Testing are completed, the Module Integration and System Testing will be done.

The Module Integration is performed and tested incrementally to produce an integrated system. Moreover, the System Testing is done when the Module Integration is completed to ensure that the chatbot conforms to the user requirements specified in Requirement phase (Basics of Software Life Cycle, n.d.).

Type of System Testing	Description
Alpha-Testing	- The development personnel perform this testing using fake data to examine if every function of the chatbot are performing as expected.
Beta-Testing	- The volunteers representing the students and parents perform this testing using real-world data which is regarding to university studies and general questions. This is in order to collect feedback on whether the chatbot resolves their problems successfully.
Acceptance Testing	- The students and parents perform this testing after the chatbot delivery to evaluate the chatbot's compliance with the user requirements specified in Requirement phase.

Table 4.3: System Testing Activities

In addition, the test cases for NLP are defined in this stage. This is done to validate the chatbot behavior in understanding intents and thus responding accurately to users. For example, the chatbot should be able to recognize misspelled words and synonyms by users. Therefore, conversation testing is performed to monitor the test cases results.

Hence, the chatbot is ready to be deployed to the production environment after all test cases and respective results are completed and documented.

4.2.4 Deploying

The chatbot is deployed at this stage that makes it available for use. The marketing and promotion teams who applied the chatbot to automate information querying will gain benefits. For example, they can streamline their workflow and optimize efficiency to deliver better information querying service to the students and parents.

4.2.5 Stabilization

Maintenance activities are performed after the completion of the project life cycle. They are mainly focused on Corrective Maintenance and Perfective Maintenance (Thakur, n.d.). Corrective Maintenance involves the debugging of faults or defects found in the design and operation of the chatbot, whereas Perfective Maintenance involves the improvement of existing functions or implementation of new user requirements. These are done so to meet the non-functional requirements of the chatbot defined earlier in Requirement phase which are availability, reliability and performance.

In addition, the launch approach for the chatbot is defined in the AI Chatbot Development Methodology's Stabilization stage whereby the deployment is done incrementally, starting with a small task. In this project, the scope of the chatbot scenario will be within FICT FAQs. Besides that, the test cases for NLP defined in Testing/Developing are carried out continuously through conversation testing. This is to make sure that the intent classification works as expected. Finally, the chatbot will be ready to have its own dedicated environment for NLP engine once it is stable. For instance, development, UAT and production environments.

4.3 Technologies and Tools Involved

4.3.1 Technologies Involved

Laptop Specification

Hardware Component	Specification
Processor	Intel Core i5-8250U
Memory	12 GB DDR4-2400
Disk Storage	128 GB SSD 1 TB HDD
Graphics Processor	NVIDIA GeForce 940MX Intel UHD Graphics 620
Display	1920x1080 FHD

Table 4.4: Hardware Specification for Laptop

Programming Language

Python is chosen as the main programming language to develop the chatbot and construct the ML model. This is because it offers a range of prolific ML libraries that facilitate continuous data processing. Industrial-strength companies are renowned for working with ASP.NET framework and C# as the main programming language. Nevertheless, Python is considered as a simpler option because it resembles the English language which makes the coding process easier. In addition, HTML and CSS programming languages serve as the template to create the chat interface. SQL programming language is also included to manage the data held in the RDBMS.

ML Model

The chatbot uses Retrieval-Based Model for functioning. Firstly, the Retrieval-Based Model depends on Intent Classification, Knowledge Management and Response Generation defined in the project scope and objectives. An NLP framework is constructed for this model that provides text analytics and classification.

Naïve Bayes

The NLP Engine is based on classification predictive modelling problems. The features that will be used to predict the results of the label are utterance patterns. For example, “When is the next intake?” and “Who is the dean of FICT?” are potential features to be trained with, whereas the labels involved are “FICT.Intake” and “FICT.Dean”. Hence, the features will help the NLP Engine to classify user problems.

Naïve Bayes is chosen as the primary approach for the NLP Engine. It is a statistical classification technique based on Bayes Theorem (Navlani 2018). This approach works relatively well for text classification and NLP, especially at the early stages of the bot launch. By reasoning, the “young” chatbot need to be trained with a lot of conversations. Otherwise, the chatbot is not good enough to handle most conversations. This is so called the chatbot’s learning phase. The strengths and weaknesses of Naïve Bayes approach are illustrated as follows.

Variable	Description
Strengths	<ul style="list-style-type: none"> - Simple Supervised Learning. - Fast and accurate. - Works relatively well on a large dataset. - Low computation cost. - Works relatively well on discrete variable compared to continuous variable. - Supports multiple class prediction problems.
Weaknesses	<ul style="list-style-type: none"> - Low accuracy if number of classes is too many. - Class predictions fail if no utterance patterns are trained on a particular class.

Table 4.5: Strengths and Weaknesses of Naïve Bayes Approach

Naïve Bayes approach is known as “naïve” because it assumes the features are independent to each other. Specifically, each word making up the utterance patterns is treated independently in the event of text classification. For example, there is no connection between “When”, “next” and “intake” in the feature “When is the next intake?”. The order of the words in an utterance pattern makes no difference but occurrences of words do matters. In addition, the features are converted into bag-of-words format before modelling and training. In NLP context, bag-of-words format is a simplified representation of each word in discrete variable. Thus, this approach works relatively well on them. Nevertheless, the tradeoff is Naïve Bayes approach does not understand the meaning behind the user problems when text classification underwent.

4.3.2 Tools Involved

The tools identified in Requirement/Envisioning are used to realize the technologies involved in Implementation phase. Hence, they are discussed as follows.

Tools	Description
Visual Studio Code	- Set up Python development environment for the Flask web application and ML Model (Microsoft 2019).
Flask Pocoo	- Provide the framework to host the chatbot and ML model. Embed with a built-in development server and fast debugger (Flask Python Framework, n.d.).
MSSQL	- Regarded as the RDBMS for this project to store and retrieve data as requested by the chatbot (Introducing SQL Server 2019).
NLTK	- Provide NLP libraries in Python for performing text analytics on human natural languages (Natural Language Processing, n.d.).
Sklearn Multinomial Naïve Bayes	<ul style="list-style-type: none"> - Text classification based on word counts with discrete features (scikit-learn developers 2019). - Support multiple class prediction problems.

Table 4.6: Tools Involved

4.4 Modules

4.4.1 Chatbot Module

User and Staff Sides

Chat

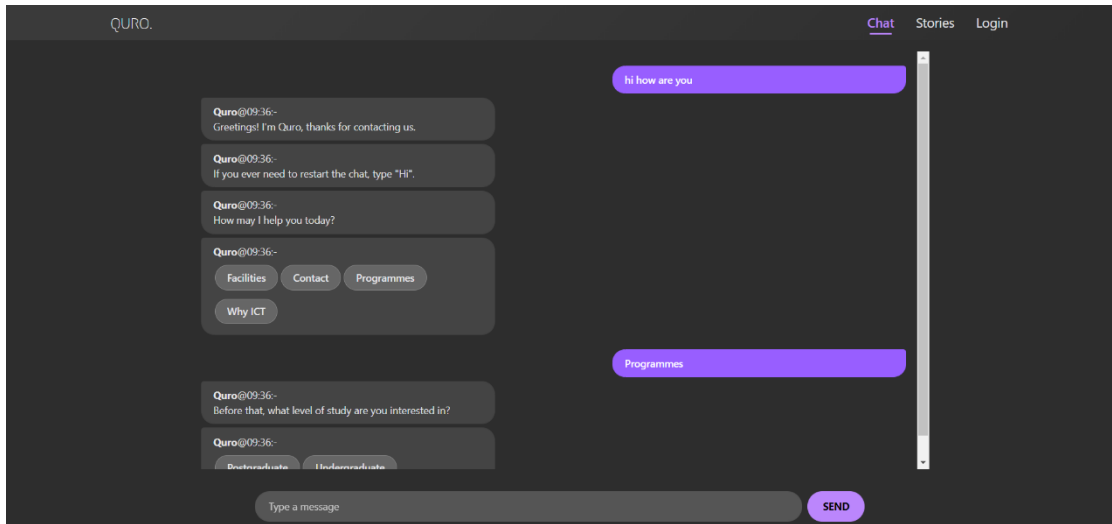


Figure 4.4: Chatroom for Chat (Web-View)

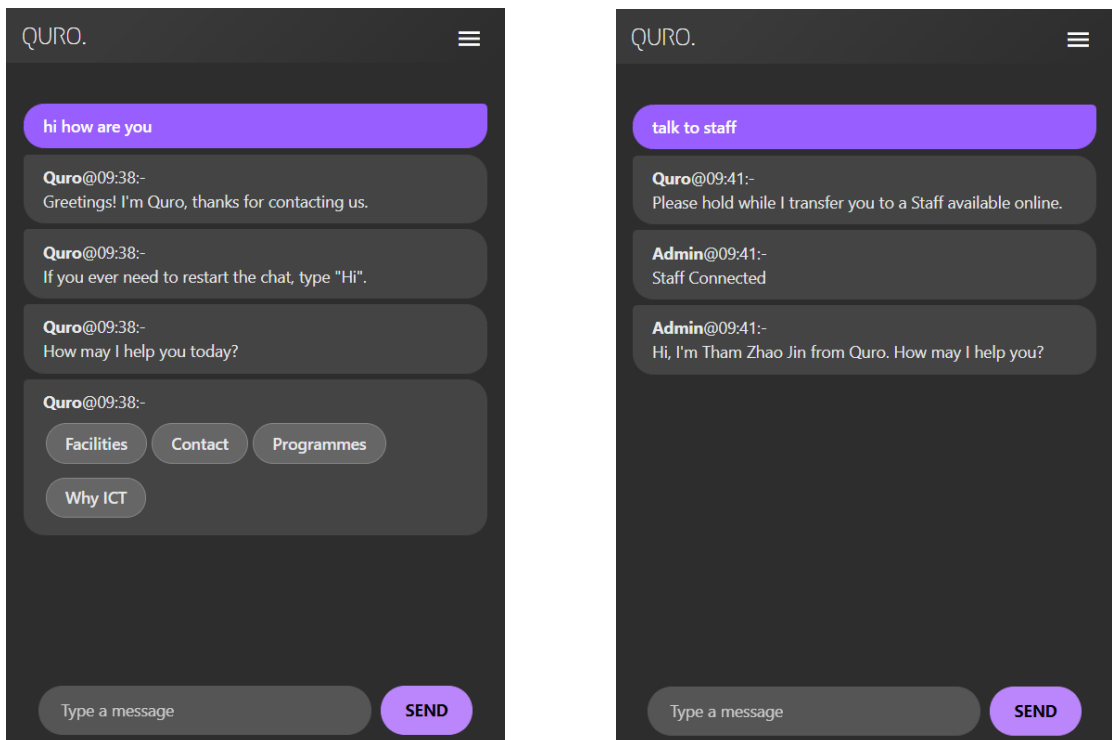


Figure 4.5: Chatroom for Chat (Mobile-View) and Live Chat

Based on Figure 4.4 and Figure 4.5, the Chat screen supports both web and mobile views. It provides a chatroom for users to ask questions directed to Quro the Chatbot. In addition, the text responses provided by Quro are customized to create a more human-like conversation with users. The conversations are powered by Retrieval-Based Model of ML Module. Quick replies are implemented to keep the users in the context of conversation. They reduce the need for users to type out their query.

Stories

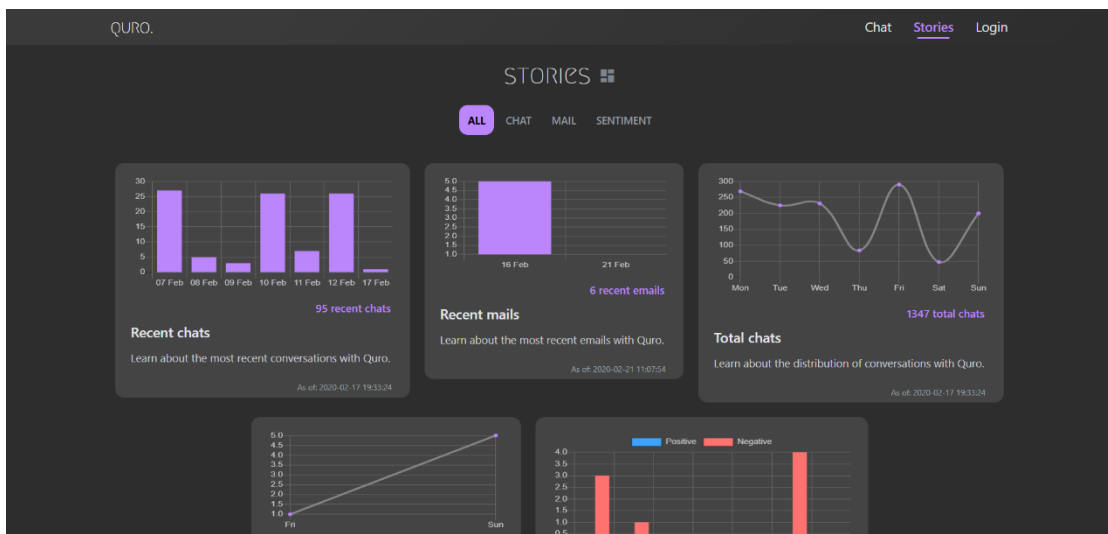


Figure 4.6: Dashboard for Stories (Web-View) – 1



Figure 4.7: Dashboard for Stories (Web-View) – 2

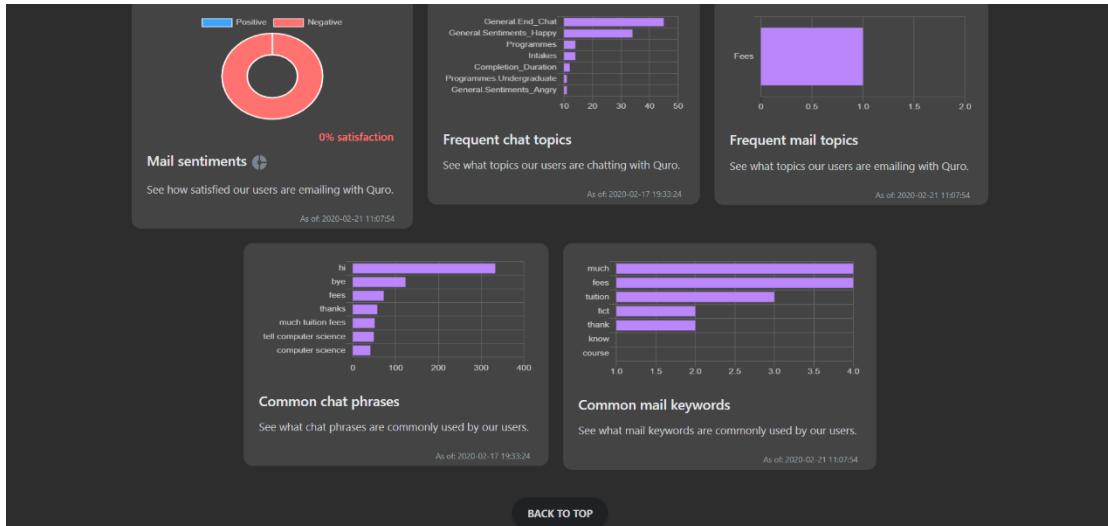


Figure 4.8: Dashboard for Stories (Web-View) – 3

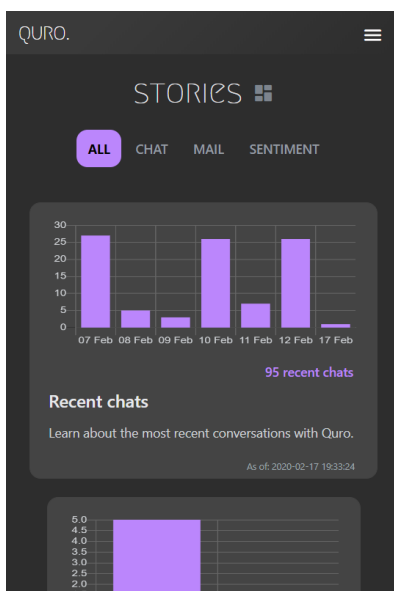


Figure 4.9: Dashboard for Stories (Mobile-View)

Based on Figure 4.6, Figure 4.7, Figure 4.8 and Figure 4.9, the Stories screen supports both web and mobile views. It provides a dashboard for users to explore the analytics running behind the scenes. Conversations between the users and Quro are stored as chatlog for analysis purpose, whereas emails between them are stored as emaillog. In addition, the users can filter the chart views by making use of the tags, namely “All”, “Chat”, “Mail” and “Sentiment”. The sentiment charts are powered by Sentiment Analysis ML Model of ML Module. Noted that the data involved to generate the charts for Stories screen is for testing purpose.

Staff Side

Login

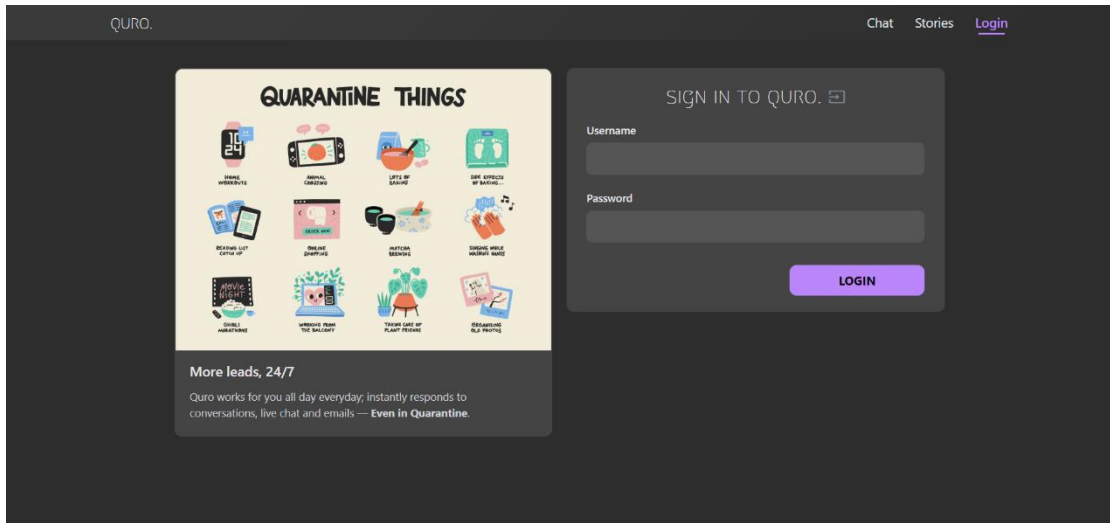


Figure 4.10: Login for Staff

Based on Figure 4.10, staff is required to login with admin credentials in order to gain access to Intents and Mail screens. For testing purpose, staff can login with “admin” as the username and “123” as the password.

Intents

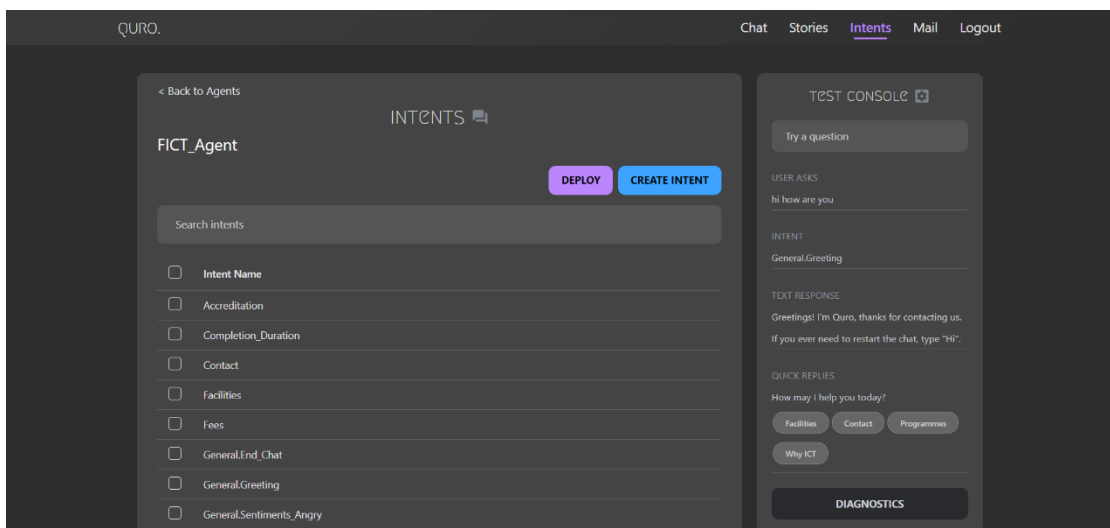


Figure 4.11: Console for Intents

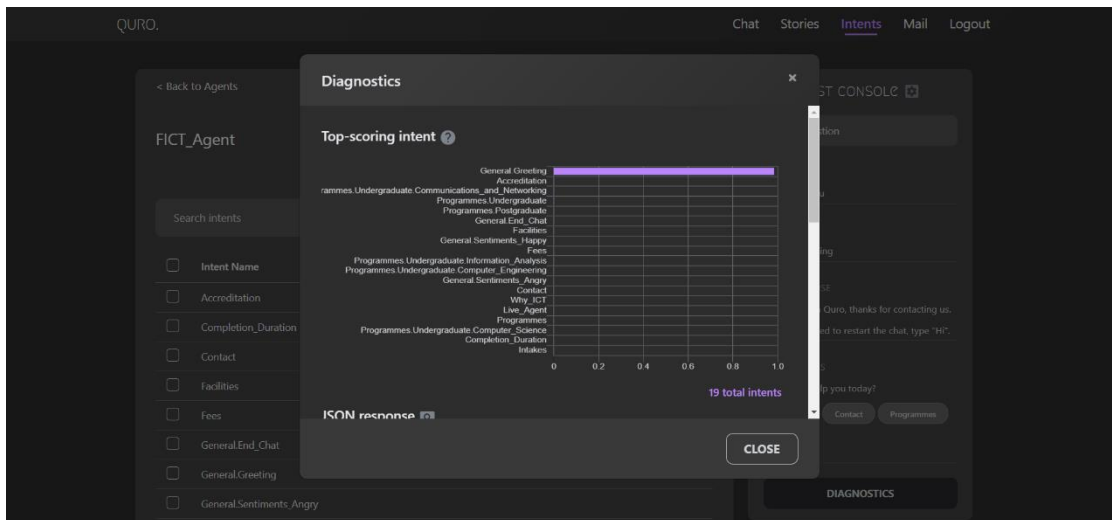


Figure 4.12: Diagnostics for Test console

Based on Figure 4.11, the Intents screen provides a console for staff to manage the training parameters such intents, utterances and responses. For instance, they can create, edit or delete the training parameters. Besides, the Intents screen provides a search function for staff to filter the list of intents based on its keyword. Once the staff is satisfied with the training parameters, they can update Quro’s ML model by clicking on Deploy button.

Furthermore, the Test console simulates the chatroom in Chat screen whereby the staff can test it out and diagnose it as shown in Figure 4.12.

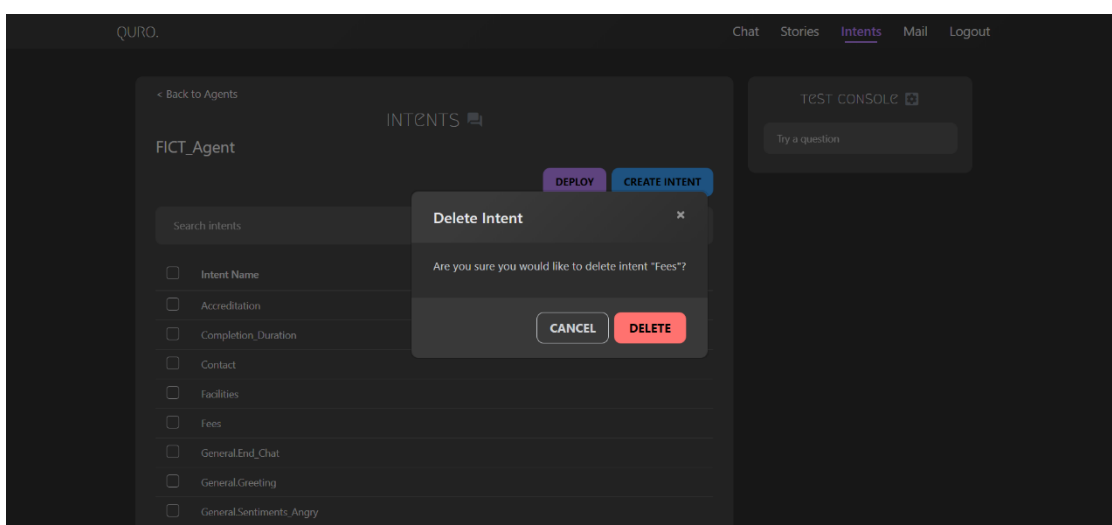


Figure 4.13: Delete Intent

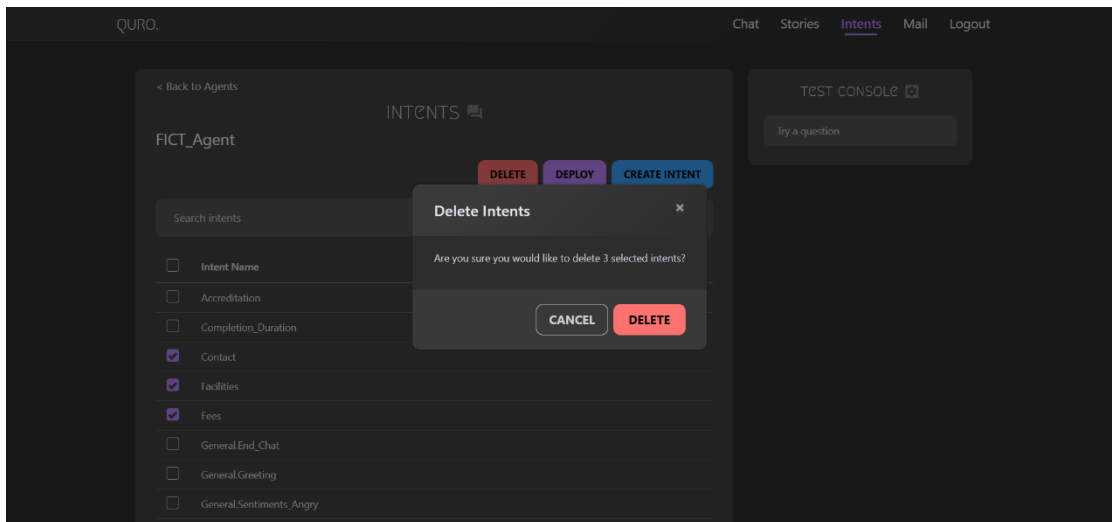


Figure 4.14: Delete Multiple Intents

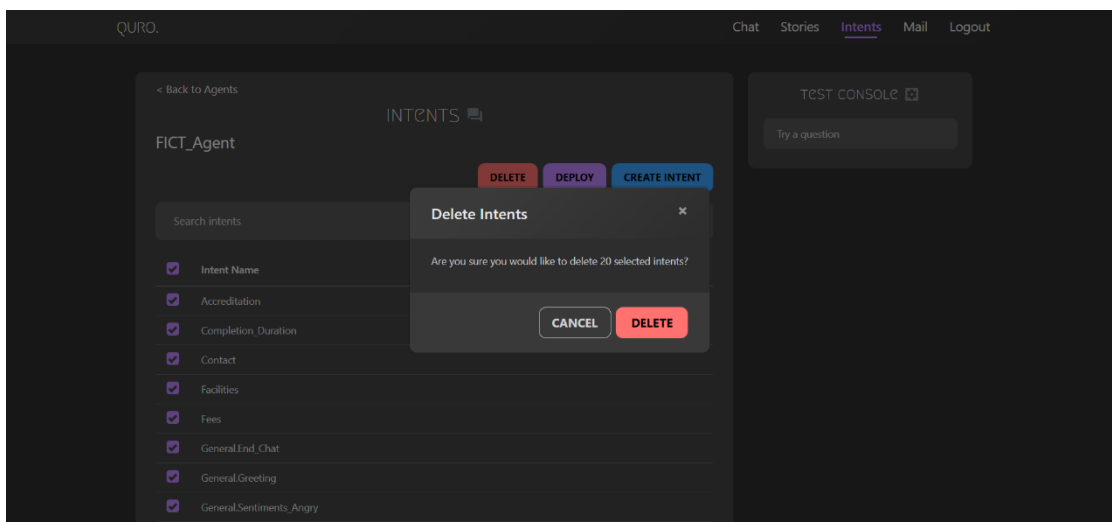


Figure 4.15: Delete All Intents

Based on Figure 4.13, Figure 4.14 and Figure 4.15, the staff can delete an intent, delete multiple intents or delete all intents. They are prompted for delete confirmation before making any changes to the Database. This is because the associated utterances and responses will be affected as well.

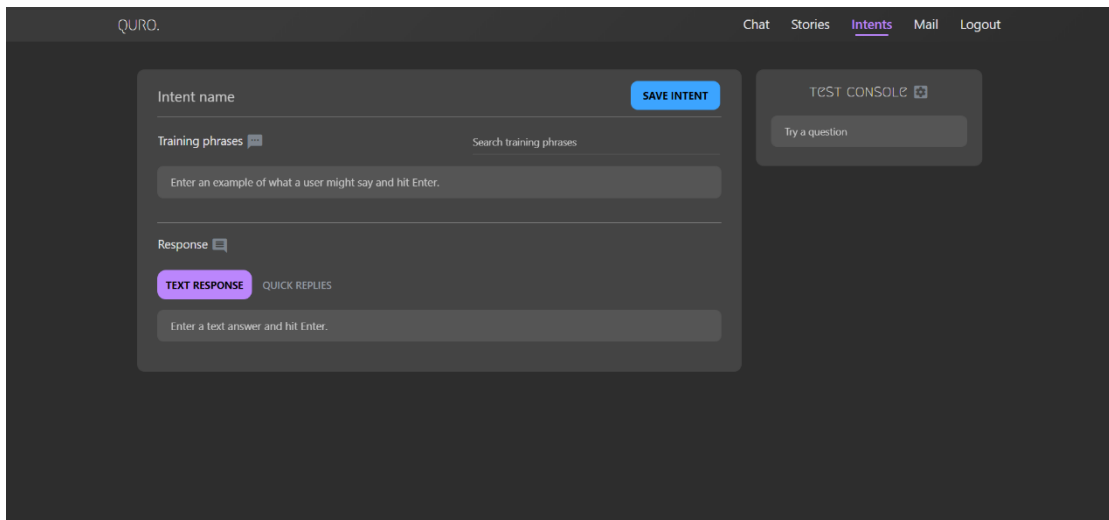


Figure 4.16: Create Intent, Utterance and Response

Based on Figure 4.16, the staff can create new intent with its utterances and responses.

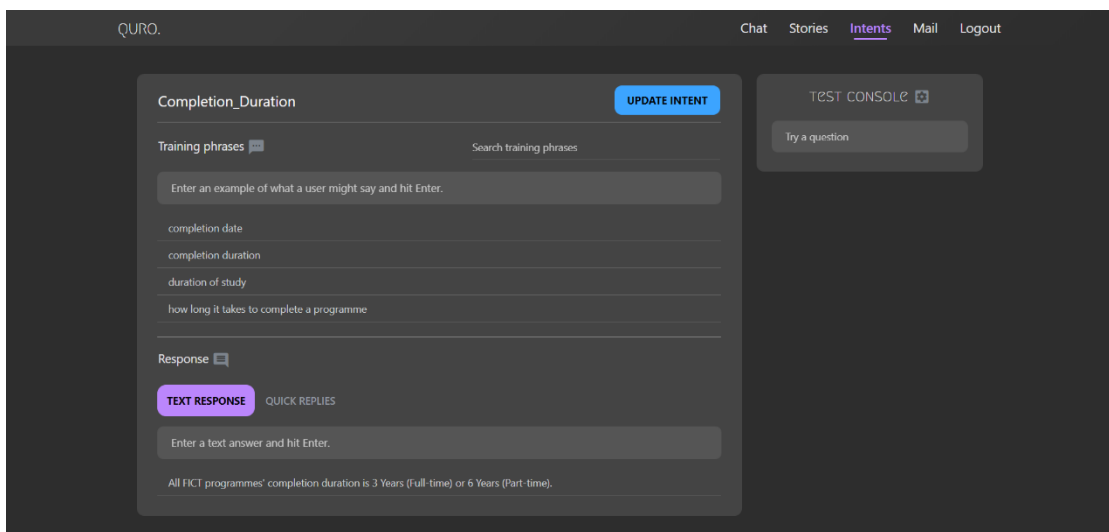


Figure 4.17: Edit Intent, Utterance and Response

Based on Figure 4.17, the staff can view detailed intent information by selecting the desire intent as shown in Figure 4.11. Then, they can update its utterances and responses. In addition, the staff can delete unwanted utterances and responses by clicking on delete icon. The delete icon will be shown when they are hovering on the utterances and responses.

Intents – Agents

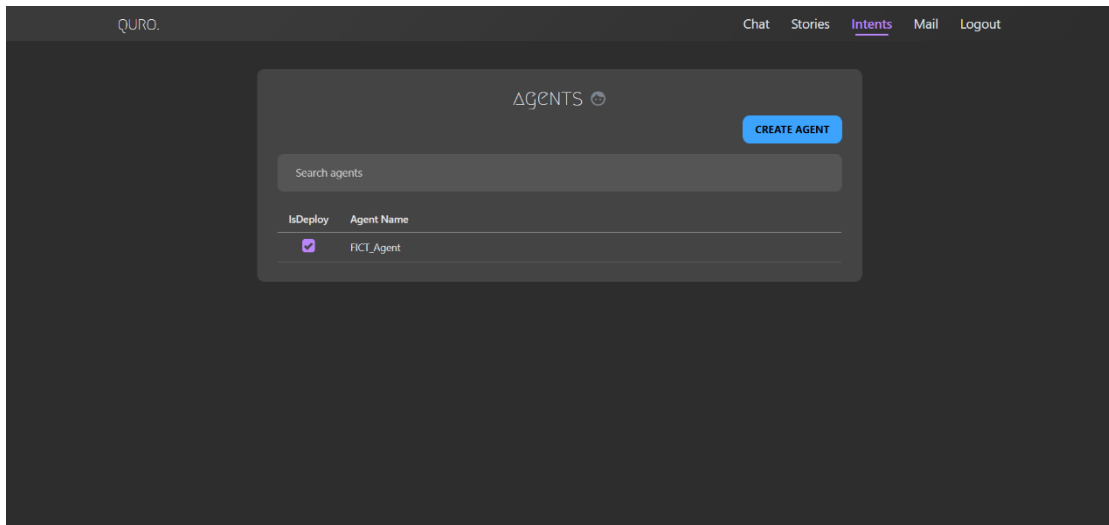


Figure 4.18: Console for Agents

Based on Figure 4.18, the staff can view Agents screen by clicking on “< Back to Agents” as shown in Figure 4.11. Agents screen provides the bot model selection functionality to staff. It tackles the problem of having a large vocabulary list. This is due to the larger the vocabulary list, the lower the uniqueness of words and phrases. Hence, the staff can configure and deploy different versions of the chatbot.

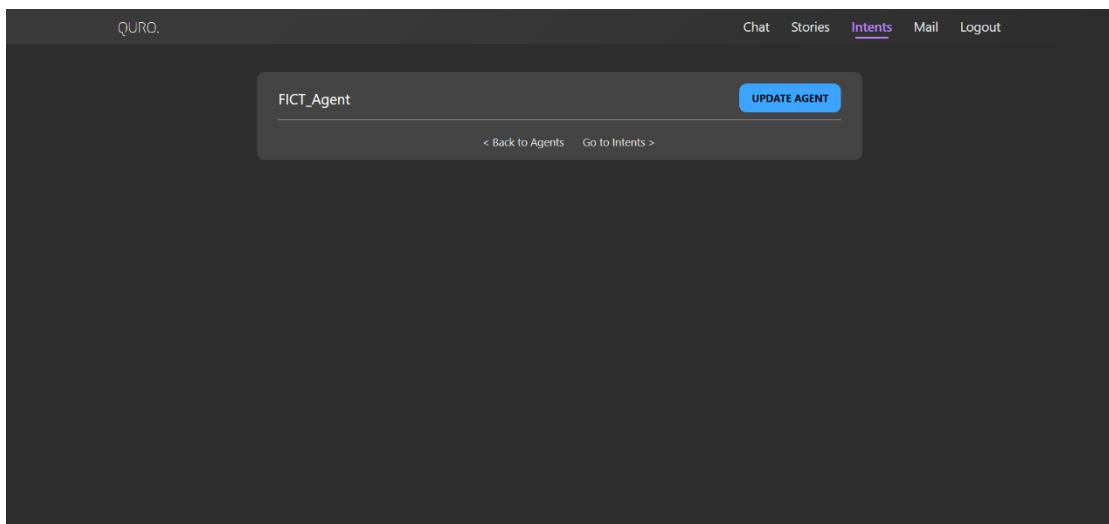


Figure 4.19: Edit Agent

Mail

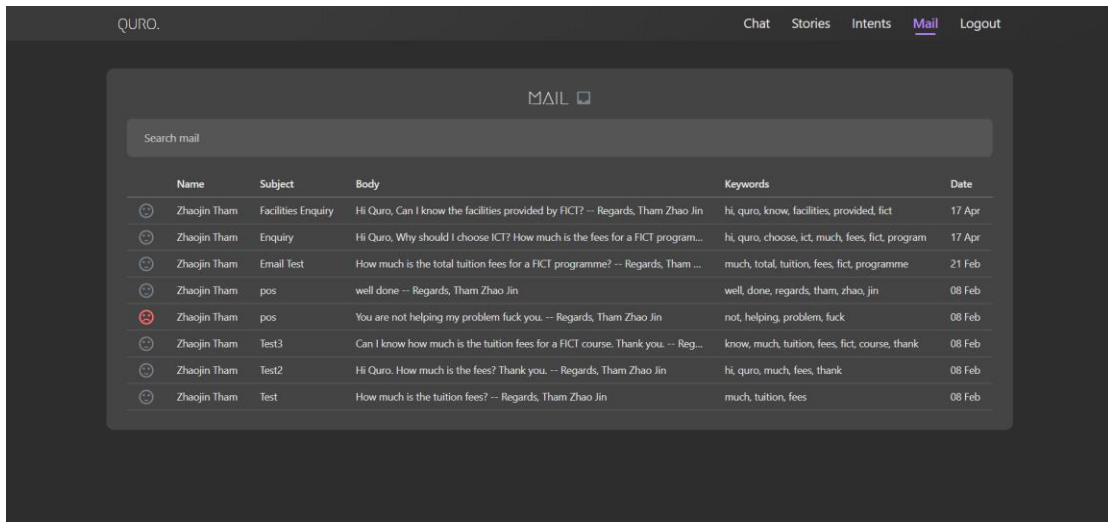


Figure 4.20: Console for Mail

Based on Figure 4.20, the Mail screen provides a console for staff to monitor incoming emails directed to Quro’s inbox. For testing purpose, Quro’s email address is “quroqu17@gmail.com”. This screen is powered by the emailbot, whereas the text analytics and classification are still handled by the ML Module of the chatbot. In addition, the leftmost column represents the sentiment for the corresponding email. The email sentiment assists staff in following-up with users.

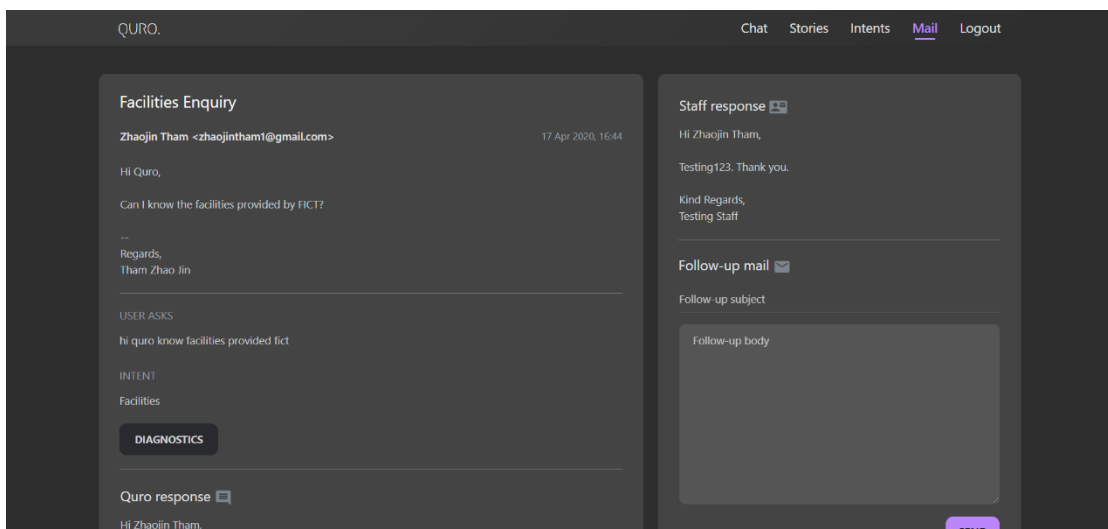


Figure 4.21: Detailed Mail – 1

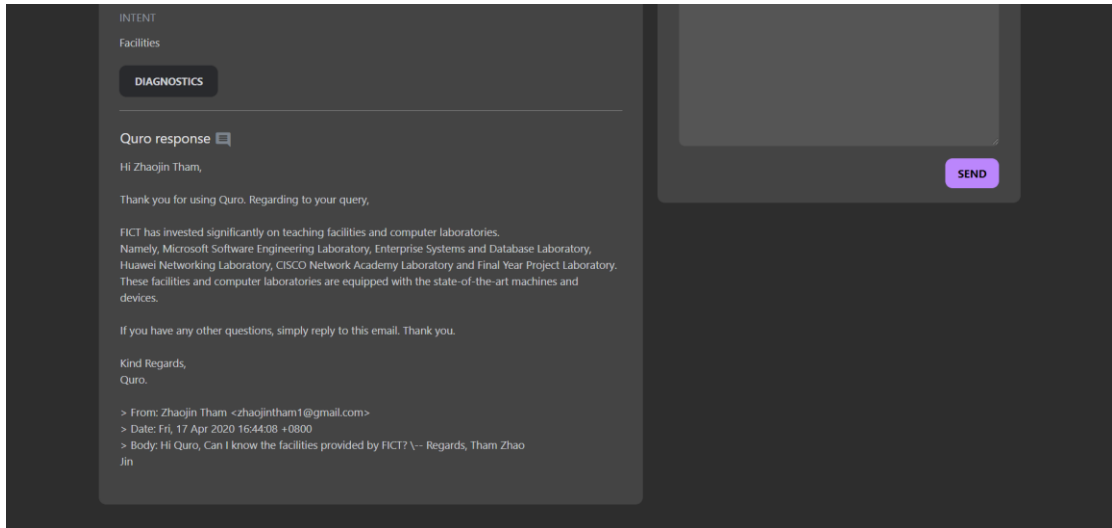


Figure 4.22: Detailed Mail – 2

Based on Figure 4.21 and Figure 4.22, the staff can view how the email content is being analyzed and its subsequent response provided by Quro. If anything goes wrong, the staff can follow-up with the corresponding user by sending a follow-up mail. This is considered as the last resort since emails are much more complex than chats.

4.4.2 ML Module

Sentiment Analysis

	Sentiment_Id	Sentiment_Name	Sentiment_Score
10	10	i'm satisfied	1
11	11	happy	1
12	12	i'm so happy	1
13	13	good job	1
14	14	wonderful	1
15	15	marvelous	1
16	16	well done	1
17	17	you have done a splendid work	1
18	18	brilliant	1
19	19	you are awesome	1
16	36	shit	-1
17	37	you are annoying	-1
18	38	i want to talk to someone	-1
19	39	i am angry	-1
20	40	sad	-1
21	41	i'm so dissapointed that you can't help me	-1
22	65	what a waste of time	-1
23	66	you are misleading	-1
24	67	poor performance	-1
25	68	bad quality	-1
26	69	service very bad	-1

Figure 4.23: Training Set for Sentiments

The Sentiment Analysis ML model will be using Naïve Bayes approach. It is a supervised learning that maps the Sentiment_Name input to Sentiment_Score output based on example input-output pairs as shown in Figure 4.23. The Sentiment_Name input is converted into bag-of-words format before modelling and training. Sentiment classification will be based on bag-of-words that depict occurrence of words within a document.

	Chat_Id	Intent_Id	Chat_Pattern	Chat_Date	Chat_Sentiment
55	110	3	well done bro	2019-11-06 12:04:44.893	1
56	112	3	well done bro	2019-11-06 12:07:36.660	1
57	136	3	well done bro	2019-11-06 12:24:54.223	1
58	114	3	well done bro	2019-11-06 12:08:16.130	1
59	137	3	well done bro	2019-11-08 10:46:01.950	1
60	119	1	who are you	2019-11-06 12:11:55.593	0
61	138	3	hi	2019-11-12 14:55:07.230	0
62	139	3	hi	2019-11-12 14:55:20.710	0
63	140	4	you are not helping at all	2019-11-12 16:28:05.310	-1
64	141	3	good job	2019-11-12 16:28:17.313	1
65	142	4	you are not solving my problem	2019-11-12 16:28:32.137	-1

Figure 4.24: Chatlog with Sentiment Scoring

Email_Id	Email_Uid	Email_Subj...	Email_FromName	Email_FromAddress	Email_ReceiveDate	Email_ReceiveBody	Email_Keywords	Email_Sentiment	En	
1	37	19	Test3	Zhaojin Tham	zhaojintham1@gmail.c...	2020-02-08 15:54:57.0...	Can I know how much is the tuition fees for a FI...	know, much, tuition, fees, fict, course, tha...	0	2C
2	38	18	pos	Zhaojin Tham	zhaojintham1@gmail.c...	2020-02-08 16:08:21.0...	well done -- Regards, Tham Zhao Jin	well, done, regards, tham, zhao, jin	0	2C
3	39	17	pos	Zhaojin Tham	zhaojintham1@gmail.c...	2020-02-08 16:03:05.0...	You are not helping my problem fuck you. -- ...	not, helping, problem, fuck	-1	2C
4	40	15	Test2	Zhaojin Tham	zhaojintham1@gmail.c...	2020-02-08 15:53:28.0...	Hi Quro. How much is the fees? Thank you. -- ...	hi, quro, much, fees, thank	0	2C
5	41	12	Test	Zhaojin Tham	zhaojintham1@gmail.c...	2020-02-08 15:11:46.0...	How much is the tuition fees? -- Regards, Th...	much, tuition, fees	0	2C
6	42	20	Email Test	Zhaojin Tham	zhaojintham1@gmail.c...	2020-02-21 11:07:30.0...	How much is the total tuition fees for a FICT pro...	much, total, tuition, fees, fict, programme	0	2C

Figure 4.25: Emaillog with Sentiment Scoring

Based on Figure 4.24, the chatlog will be updated with respective sentiment scoring. In addition, the Chat_Sentiment with values of 0 indicates neutral sentiment or untrained positive/negative sentiment. This metric captures the mood of users interacting with Quro based on a series of words and phrases. Hence, sentiment analysis is useful in conversation monitoring as it allows staff to gain insights on user behavior. Same goes for the emaillog as shown in Figure 4.25.

Retrieval-Based Model

	Utterance_Id	Intent_Id	Utterance_Name	Utterance_IsDelete
22	376	65	careers in ict	0
23	377	65	jobs in ict	0
24	378	65	why should i consider ict	0
25	379	65	why should i choose ict	0
26	380	66	how long it takes to complet...	0
27	381	66	duration of study	0
28	382	66	completion duration	0
29	383	66	completion date	0

Figure 4.26: Training Set for Utterances

	Intent_Id	Intent_Name	Intent_IsDelete	Agent_Id
4	57	General.End_Chat	0	22
5	63	Programmes	0	22
6	64	Intakes	0	22
7	65	Why_ICT	0	22
8	66	Completion_Duration	0	22
9	67	Programmes.Undergraduate	0	22
10	68	Programmes.Postgraduate	0	22
11	69	Fees	0	22
12	70	Facilities	0	22

Figure 4.27: Training Set for Intents

The Retrieval-Based Model will be using Naïve Bayes approach. It is a supervised learning that maps the Utterance_Name input to Intent_Id output based on example input-output pairs as shown in Figure 4.26 and Figure 4.27. The Utterance_Name input will undergo tokenization, stop words and punctuations removal, and convert into bag-of-words format before modelling and training. Similar to Sentiment Analysis ML Model, intent classification will be based on bag-of-words as well.

```

{
  "modelAccuracy": "0.4482758620689655",
  "question": "why should i choose ict",
  "questionPreprocessed": "choose ict",
  "results": [
    {
      "intent": "FICT.Why_ICT",
      "response": [
        "ICT is one of the area with the highest market demand for graduates in Malaysia today due to the vast expanding opportunities in ICT.",
        "Furthermore, ICT graduates will have a long list of work positions that are available to be chosen from upon graduation due to the various skills in ICT that they possess."
      ],
      "score": "0.9937011464862265"
    },
    {
      "intent": "FICT.Career_Prospect",
      "response": [
        "ICT is applicable to almost every industry out there - Software Engineer, Web Application Developer, System Analyst, etc."
      ],
      "score": "0.005517204489339947"
    },
    {
      "intent": "FICT.Location",
      "response": [
        "FICT is located at UTAR Kampar Campus."
      ],
      "score": "6.061251622026425e-05"
    }
  ]
}

```

Figure 4.28: JSON Results for Intent Classification

Furthermore, the question submitted by users will undergo text preprocessing before sending it for intent prediction. Stop words and punctuations are removed as they affect the accuracy and performance of intent classification. Based on Figure 4.28, the appropriate text response with highest intent scoring will be returned to the users. Noted that the question is reduced so that the vocabulary used for intent classification is kept short and meaningful. In addition, the preprocessed question also contributes to the common chat phrases and mail keywords charts as shown in Figure 4.8.

	PreprocessEmail_Id	Email_Id	Intent_Id	PreprocessEmail_Name
1	130	37	8	know much tuition fees fict course
2	131	37	5	thank

Figure 4.29: Preprocessed Email

Based on Figure 4.25 and Figure 4.29, the email content with Email_Id = 37 is analyzed sentence by sentence. Hence, there might be multiple queries stated in the email. The user will then receive a response for each query. In addition, the emailbot will not send out a response if the intent confidence score of the query is less than the threshold = 0.4. This is to avoid sending out the wrong response. Thus, complex query with low confidence score requires clarification from staff through follow-up mail.

The workings behind Naïve Bayes approach are explained as follows.

Conditional Probability

Based on Probability Theory, conditional probability is the measure of probability that the event will occur considering that another event has already occurred.

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Figure 4.30: Conditional Probability Formula

Bayesian Rule

Based on Figure 4.30, Bayesian Rule is formulated when $P(A \cap B)$ is substituted with $P(A|B) * P(B)$. Hence, Bayesian Rule formula is shown as follows.

$$P(B|A) = \frac{P(A|B) * P(B)}{P(A)}$$

Figure 4.31: Bayesian Rule Formula

Naïve Bayes Theorem

Naïve Bayes approach is based on Bayesian Rule with the independence assumptions between features. In chatbot context, the features used will be the utterance patterns. For example, how to classify the user question “When is the next intake?” according to the vocabulary list built for intent classification as shown in the table below.

Utterance_Id	programme	next	fees	intake	Intent_Name
1	1	0	1	0	FICT.Fees
2	0	1	0	1	FICT.Intake
3	0	0	0	1	FICT.Intake
4	1	1	0	0	FICT.Programme
5	0	0	1	0	FICT.Fees

Table 4.7: Example of Vocabulary List

Simplified version of the user question: (0, next=1, 0, intake=1) is obtained based on the vocabulary list with 1 denoting its existence. Let $X = (0, 1, 0, 1)$, there are three possibilities that need to be classified, the Intent_Name predicted could be “FICT.Fees”, “FICT.Intake” or “FICT.Programme”.

Possibility #1:

If Intent_Name = “FICT.Fees”, the result is calculated as follows.

$$P(X|\text{Intent_Name} = \text{“FICT.Fees”}) = P(\text{Intent_Name} = \text{“FICT.Fees”}|X) * P(X) / P(\text{Intent_Name} = \text{“FICT.Fees”})$$

$P(\text{Intent_Name} = \text{“FICT.Fees”}|X) = ?$ since right side of equation consisting independent features first, the individual probability for each feature is calculated and then multiple all of them to obtain the final result.

$$P(\text{programme}=0|\text{Intent_Name} = \text{"FICT.Fees"}) = 1/2$$

$$P(\text{next}=1|\text{Intent_Name} = \text{"FICT.Fees"}) = 0$$

$$P(\text{fees}=0|\text{Intent_Name} = \text{"FICT.Fees"}) = 0$$

$$P(\text{intake}=1|\text{Intent_Name} = \text{"FICT.Fees"}) = 0$$

Since all of them are independent features:

$$P(X|\text{Intent_Name} = \text{"FICT.Fees"}) = P(\text{programme}=0|\text{Intent_Name} = \text{"FICT.Fees"}) *$$

$$P(\text{next}=1|\text{Intent_Name} = \text{"FICT.Fees"}) * P(\text{fees}=0|\text{Intent_Name} = \text{"FICT.Fees"}) *$$

$$P(\text{intake}=1|\text{Intent_Name} = \text{"FICT.Fees"})$$

$$P(X|\text{Intent_Name} = \text{"FICT.Fees"}) = (1/2) * 0 * 0 * 0 = 0$$

$$P(\text{Intent_Name} = \text{"FICT.Fees"}) = 2/5$$

$$P(\text{Intent_Name} = \text{"FICT.Fees"}|X) = 0 * (2/5) = 0$$

Hence, it is impossible for the user question to be classified as "FICT.Fees".

Possibility #2:

If Intent_Name = "FICT.Intake", the result is calculated as follows.

$$P(X|\text{Intent_Name} = \text{"FICT.Intake"}) = P(\text{Intent_Name} = \text{"FICT.Intake"}|X) * P(X) /$$

$$P(\text{Intent_Name} = \text{"FICT.Intake"})$$

$$P(X|\text{Intent_Name} = \text{"FICT.Intake"}) = P(\text{programme}=0|\text{Intent_Name} = \text{"FICT.Intake"})$$

$$* P(\text{next}=1|\text{Intent_Name} = \text{"FICT.Intake"}) * P(\text{fees}=0|\text{Intent_Name} = \text{"FICT.Intake"})$$

$$* P(\text{intake}=1|\text{Intent_Name} = \text{"FICT.Intake"})$$

$$P(X|\text{Intent_Name} = \text{"FICT.Intake"}) = 1 * (1/2) * 1 * 1 = 1/2$$

$$P(\text{Intent_Name} = \text{"FICT.Intake"}) = 2/5$$

$$P(\text{Intent_Name} = \text{"FICT.Intake"}|X) = (1/2) * (2/5) = 1/5$$

Hence, it has 1/5 chances for the user question to be classified as "FICT.Intake".

Possibility #3:

If Intent_Name = “FICT.Programme”, the result is calculated as follows.

$$P(X|Intent_Name = \text{“FICT.Programme”}) = P(Intent_Name = \text{“FICT.Programme”}|X) * P(X) / P(Intent_Name = \text{“FICT.Programme”})$$

$$P(X|Intent_Name = \text{“FICT.Programme”}) = P(\text{programme}=0|Intent_Name = \text{“FICT.Programme”}) * P(\text{next}=1|Intent_Name = \text{“FICT.Programme”}) * P(\text{fees}=0|Intent_Name = \text{“FICT.Programme”}) * P(\text{intake}=1|Intent_Name = \text{“FICT.Programme”})$$

$$P(X|Intent_Name = \text{“FICT.Programme”}) = 0 * 1 * 1 * 0 = 0$$

$$P(Intent_Name = \text{“FICT.Programme”}) = 1/5$$

$$P(Intent_Name = \text{“FICT.Programme”}|X) = 0 * (1/5) = 0$$

Therefore, Naïve Bayes approach classifies the user question “When is the next intake?” as “FICT.Intake” with the probability of 1/5, whereas the other intents have 0 probabilities.

Laplacian Smoothing

When no utterance patterns are trained for a particular intent, zero probabilities will occur as shown in the example above. Thus, Laplacian Smoothing can be applied to overcome zero probabilities. Based on the example below, Laplace Smoothing parameter is set to 1.

$$P(x_i|v_j) = \frac{n_i + 1}{n + |vocabulary|}$$

Figure 4.32: Laplacian Smoothing Formula

Utterance pattern #1: “How much is the fees?”

Utterance pattern #2: “When is the next intake?”

Vocabulary list: (fees, next, intake) -> Stop words are removed

Let $X =$ “What are the fees for next intake?” be the user question, chances the user question to be matched with the utterance patterns are $1/2$ (Utterance pattern #1 or Utterance pattern #2).

Based on Figure 4.32,

$$|\text{vocab}| = 3$$

$$n_{\text{Utterancepattern1}} = 1 \text{ (Stop words are removed)}$$

$$n_{\text{Utterancepattern2}} = 2 \text{ (Stop words are removed)}$$

X will have “fees”, “next” and “intake” words found in the vocabulary list.

Possibility #1:

If $n_{\text{Utterancepattern1}}$ is matched,

$$P(n_{\text{Utterancepattern1}}|X) = P(n_{\text{Utterancepattern1}}) * P(\text{fees}|n_{\text{Utterancepattern1}}) *$$

$$P(\text{next}|n_{\text{Utterancepattern1}}) * P(\text{intake}|n_{\text{Utterancepattern1}}) = (1/2) * [(1+1)/(1+3)] *$$

$$[(0+1)/(1+3)] * [(0+1)/(1+3)] = 0.015625$$

Noted that without Laplace Smoothing, Utterance pattern #1 will face zero probabilities.

Possibility #2:

If $n_{\text{Utterancepattern2}}$ is matched,

$$P(n_{\text{Utterancepattern2}}|X) = P(n_{\text{Utterancepattern2}}) * P(\text{fees}|n_{\text{Utterancepattern2}}) *$$

$$P(\text{next}|n_{\text{Utterancepattern2}}) * P(\text{intake}|n_{\text{Utterancepattern2}}) = (1/2) * [(0+1)/(2+3)] *$$

$$[(1+1)/(2+3)] * [(1+1)/(2+3)] = 0.016$$

Therefore, the user question “What are the fees for next intake?” has a higher probability to be match with Utterance pattern #2 compared to Utterance pattern #1 ($0.016 > 0.015625$). Laplace Smoothing is required for multiple class prediction problems because as the number of classes increases, the probability becomes smaller in decimals. Hence, chances of facing zero probabilities will become higher.

Spam Filtration

	ChatSpam_Id	ChatSpam_Text	ChatSpam_IsSpam
1	9	Go until jurong point, crazy.. Available only in bugis n gre...	0
2	10	Ok lar... Joking wif u oni...	0
3	11	Free entry in 2 a wkly comp to win FA Cup final tkts 21st ...	1
4	12	U dun say so early hor... U c already then say...	0
5	13	Nah I don't think he goes to usf, he lives around here tho...	0
6	14	FreeMsg Hey there darling it's been 3 week's now and n...	1
7	15	Even my brother is not like to speak with me. They treat ...	0
8	16	As per your request 'Melle Melle (Oru Minnaminunginte ...	0

Figure 4.33: Training Set for Chat Spam

	EmailSpam_Id	EmailSpam_Text	EmailSpam_IsSpam
1	52	Subject: naturally irresistible your corporate identity I...	1
2	53	Subject: the stock trading gunslinger fanny is merrill...	1
3	54	Subject: unbelievable new homes made easy im w...	1
4	55	Subject: 4 color printing special request additional i...	1
5	56	Subject: do not have money , get software cds from ...	1
6	58	Subject: here ' s a hot play in motion homeland sec...	1
7	65	Subject: wall street phenomenon reaps rewards sm...	1
8	70	Subject: security alert - confirm your national credit u...	1

Figure 4.34: Training Set for Email Spam

Sentiment Analysis and Intent Classification which make use of Naïve Bayes approach for the ML Module are regarded as the NLP Classification layer. Before the user query goes through this layer, it requires to pass through a Spam Filtration layer beforehand. The Spam Filtration layer also uses the same approach for classifying whether the query is spam or not. As shown in Figure 4.33 and Figure 4.34, it is a supervised learning that maps the ChatSpam_Text input to ChatSpam_IsSpam output based on example input-output pairs; whereas for email, it maps EmailSpam_Text input to EmailSpam_IsSpam output based on example input-output pairs. Hence, the chatbot and emailbot will not accept the query if the query is labelled as spam.

4.4.3 Others

Access Tokens
Create New Page

Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page.

Pages ↑	Tokens
<div style="display: flex; align-items: center;"> <div style="margin-left: 5px;"> <p>Quro.</p> <p>110796970526363</p> </div> </div>	—
Generate Token	

Add or Remove Pages ⓘ

Webhooks

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Callback URL

Verify Token

Validation requests and Webhook notifications for this object will be sent to this URL.

Token that Facebook will echo back to you as part of callback URL verification.

Edit Callback URL
Show Recent Errors

Pages ↑	Webhooks
<div style="display: flex; align-items: center;"> <div style="margin-left: 5px;"> <p>Quro.</p> <p>110796970526363</p> </div> </div>	<p>4 Fields</p> <p>messages, messaging_postbacks, messaging_optins, message_deliveries</p>
Edit	

Add or Remove Pages ⓘ

Figure 4.35: Setting Up Facebook Page via Facebook for Developers

Based on Figure 4.35, page access token, verify token and callback URL are the attributes needed to integrate the Flask App and Facebook Messenger. Page access token is generated by Facebook, whereas verify token is self-defined in the Flask App.

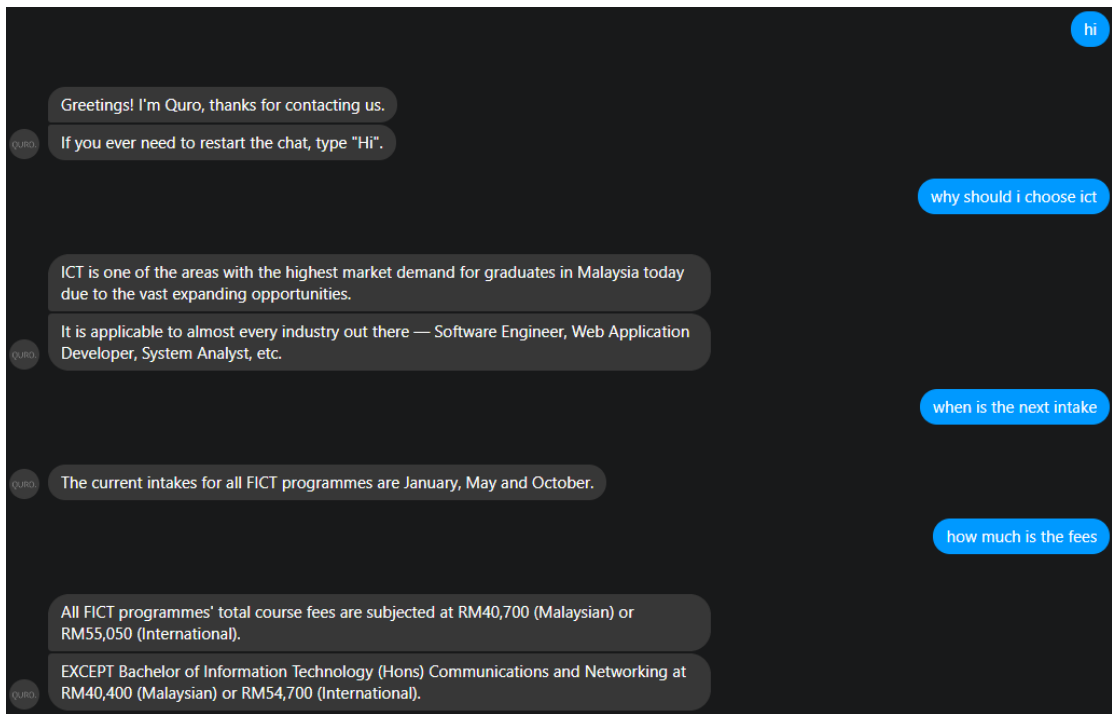


Figure 4.36: Chatting with Quoro via Facebook Messenger

Based on Figure 4.36, it shows the conversation undergoing as expected, indicating that the integration between the Flask App and Facebook Messenger is successfully done.

CHAPTER 5: SYSTEM TESTING

5.1 Usability Testing

5.1.1 Chat Module

Chat Usability Test Case

Purpose: The system should always fulfil UI & UX criteria

Step No.	Review Checklist	Yes / No / Not Applicable	Comments
1.1	Is there a consistency in font size and font family?	Yes	
1.2	Is the system easy to use and understand?	Yes	
1.3	Is there a clean and simple presentation in the system?	Yes	
1.4	Is the system can be navigated easily?	Yes	
1.5	Is the system provide useful information?	No	
1.6	Is the system can be explored by trial and error?	Yes	
1.7	Is the system can be used by inexperienced user?	Yes	
1.8	Is there any clear instruction when using the system?	No	

1.9	Do the system provide any helpful information to correct user?	No
1.10	Is there a consistency in using term throughout the system?	Yes
1.11	Is the position of message on screen consistent?	Yes
1.12	Is the organization of information clear?	Yes

Table 5.1: Chat Usability Test Case

5.1.2 Stories Module

Stories Usability Test Case

Purpose: The system should always fulfil UI & UX criteria

Step No.	Review Checklist	Yes / No / Not Applicable	Comments
1.1	Is there a consistency in font size and font family?	Yes	
1.2	Is the system easy to use and understand?	Yes	
1.3	Is there a clean and simple presentation in the system?	Yes	
1.4	Is the system can be navigated easily?	Yes	
1.5	Is the system provide useful information?	Yes	
1.6	Is the system can be explored by trial and error?	Yes	
1.7	Is the system can be used by inexperienced user?	Yes	
1.8	Is there any clear instruction when using the system?	Yes	
1.9	Do the system provide any helpful information to correct user?	No	
1.10	Is there a consistency in using term throughout the system?	Yes	

1.11	Is the position of message on screen consistent?	Yes
1.12	Is the organization of information clear?	Yes

Table 5.2: Stories Usability Test Case

5.1.3 Login Module

Login Usability Test Case

Purpose: The system should always fulfil UI & UX criteria

Step No.	Review Checklist	Yes / No / Not Applicable	Comments
1.1	Is there a consistency in font size and font family?	Yes	
1.2	Is the system easy to use and understand?	Yes	
1.3	Is there a clean and simple presentation in the system?	Yes	
1.4	Is the system can be navigated easily?	Yes	
1.5	Is the system provide useful information?	No	
1.6	Is the system can be explored by trial and error?	No	
1.7	Is the system can be used by inexperienced user?	Yes	
1.8	Is there any clear instruction when using the system?	No	
1.9	Do the system provide any helpful information to correct user?	No	
1.10	Is there a consistency in using term throughout the system?	Yes	

1.11	Is the position of message on screen consistent?	Yes
1.12	Is the organization of information clear?	Yes

Table 5.3: Login Usability Test Case

5.1.4 Intents Module

Intents Usability Test Case

Purpose: The system should always fulfil UI & UX criteria

Step No.	Review Checklist	Yes / No / Not Applicable	Comments
1.1	Is there a consistency in font size and font family?	Yes	
1.2	Is the system easy to use and understand?	Yes	
1.3	Is there a clean and simple presentation in the system?	Yes	
1.4	Is the system can be navigated easily?	Yes	
1.5	Is the system provide useful information?	Yes	
1.6	Is the system can be explored by trial and error?	Yes	
1.7	Is the system can be used by inexperienced user?	Yes	
1.8	Is there any clear instruction when using the system?	Yes	
1.9	Do the system provide any helpful information to correct user?	Yes	
1.10	Is there a consistency in using term throughout the system?	Yes	

1.11	Is the position of message on screen consistent?	Yes
1.12	Is the organization of information clear?	Yes

Table 5.4: Intents Usability Test Case

5.1.5 Intents – Create / Update Module

Intents – Create / Update Usability Test Case

Purpose: The system should always fulfil UI & UX criteria

Step No.	Review Checklist	Yes / No / Not Applicable	Comments
1.1	Is there a consistency in font size and font family?	Yes	
1.2	Is the system easy to use and understand?	Yes	
1.3	Is there a clean and simple presentation in the system?	Yes	
1.4	Is the system can be navigated easily?	Yes	
1.5	Is the system provide useful information?	Yes	
1.6	Is the system can be explored by trial and error?	No	
1.7	Is the system can be used by inexperienced user?	Yes	
1.8	Is there any clear instruction when using the system?	Yes	
1.9	Do the system provide any helpful information to correct user?	Yes	
1.10	Is there a consistency in using term throughout the system?	Yes	

1.11	Is the position of message on screen consistent?	Yes
1.12	Is the organization of information clear?	Yes

Table 5.5: Intents – Create / Update Usability Test Case

5.1.6 Intents – Agents Module

Intents – Agents Usability Test Case

Purpose: The system should always fulfil UI & UX criteria

Step No.	Review Checklist	Yes / No / Not Applicable	Comments
1.1	Is there a consistency in font size and font family?	Yes	
1.2	Is the system easy to use and understand?	Yes	
1.3	Is there a clean and simple presentation in the system?	Yes	
1.4	Is the system can be navigated easily?	Yes	
1.5	Is the system provide useful information?	Yes	
1.6	Is the system can be explored by trial and error?	Yes	
1.7	Is the system can be used by inexperienced user?	Yes	
1.8	Is there any clear instruction when using the system?	Yes	
1.9	Do the system provide any helpful information to correct user?	Yes	
1.10	Is there a consistency in using term throughout the system?	Yes	

1.11	Is the position of message on screen consistent?	Yes
1.12	Is the organization of information clear?	Yes

Table 5.6: Intents – Agents Usability Test Case

5.1.7 Intents – Create / Update Agents Module

Intents – Create / Update Agents Usability Test Case

Purpose: The system should always fulfil UI & UX criteria

Step No.	Review Checklist	Yes / No / Not Applicable	Comments
1.1	Is there a consistency in font size and font family?	Yes	
1.2	Is the system easy to use and understand?	Yes	
1.3	Is there a clean and simple presentation in the system?	Yes	
1.4	Is the system can be navigated easily?	Yes	
1.5	Is the system provide useful information?	No	
1.6	Is the system can be explored by trial and error?	No	
1.7	Is the system can be used by inexperienced user?	Yes	
1.8	Is there any clear instruction when using the system?	No	
1.9	Do the system provide any helpful information to correct user?	No	
1.10	Is there a consistency in using term throughout the system?	Yes	

1.11	Is the position of message on screen consistent?	Yes
1.12	Is the organization of information clear?	Yes

Table 5.7: Intents – Create / Update Agents Usability Test Case

5.1.8 Mail Module

Mail Usability Test Case

Purpose: The system should always fulfil UI & UX criteria

Step No.	Review Checklist	Yes / No / Not Applicable	Comments
1.1	Is there a consistency in font size and font family?	Yes	
1.2	Is the system easy to use and understand?	Yes	
1.3	Is there a clean and simple presentation in the system?	Yes	
1.4	Is the system can be navigated easily?	Yes	
1.5	Is the system provide useful information?	Yes	
1.6	Is the system can be explored by trial and error?	Yes	
1.7	Is the system can be used by inexperienced user?	Yes	
1.8	Is there any clear instruction when using the system?	Yes	
1.9	Do the system provide any helpful information to correct user?	Yes	
1.10	Is there a consistency in using term throughout the system?	Yes	

1.11	Is the position of message on screen consistent?	Yes
1.12	Is the organization of information clear?	Yes

Table 5.8: Mail Usability Test Case

5.1.9 Mail – View Detailed Module

Mail – View Detailed Usability Test Case

Purpose: The system should always fulfil UI & UX criteria

Step No.	Review Checklist	Yes / No / Not Applicable	Comments
1.1	Is there a consistency in font size and font family?	Yes	
1.2	Is the system easy to use and understand?	Yes	
1.3	Is there a clean and simple presentation in the system?	Yes	
1.4	Is the system can be navigated easily?	Yes	
1.5	Is the system provide useful information?	Yes	
1.6	Is the system can be explored by trial and error?	No	
1.7	Is the system can be used by inexperienced user?	Yes	
1.8	Is there any clear instruction when using the system?	No	
1.9	Do the system provide any helpful information to correct user?	Yes	
1.10	Is there a consistency in using term throughout the system?	Yes	

1.11	Is the position of message on screen consistent?	Yes
1.12	Is the organization of information clear?	Yes

Table 5.9: Mail – View Detailed Usability Test Case

5.2 Unit Testing

5.2.1 Chat Module

Chat Unit Test Case				
Test Case Filename: main.py		Date Created: 10 October 2019		
Tester(s): Tham Zhao Jin		Access Path: ..\Chatbot\main.py		
Transactions: 1. / → /stories 2. / → /login				
Purpose of Test Case: To ensure that the user can chat with the system and changes are reflected directly to the database. To test for errors in page navigation.				
Step No.	Input Values to Execute Test Case	Expected Results	Actual Results	Pass / Fail - Comments
1.	User inputs query and clicks on the “Send” button at the bottom of Chat screen or presses “Enter” key.	The user query will be added to the database and displayed in the chatroom. Then, a response provided by the chatbot will also be displayed shortly after.	The system adds the user query in database and displays the query and corresponding response provided by the chatbot in the chatroom.	Pass

2.	User clicks on the quick reply button in the chatroom.	The text in quick reply button will be displayed in the chatroom. Then, a response provided by the chatbot will also be displayed shortly after.	The system displays the text in quick reply button and corresponding response provided by the chatbot in the chatroom.	Pass
3.	User clicks on the “Chat” navigation button on the top navigation bar.	User will remain in the same page (/chat).	User remains in Chat screen.	Pass
4.	User clicks on the “Stories” navigation button on the top navigation bar.	User will be redirected to the next page (/stories).	System redirects the user to Stories screen to view dashboard.	Pass
5.	User clicks on the “Login” navigation button on the top navigation bar.	User will be redirected to the next page (/login).	System redirects the user to Login screen to perform login action for staff.	Pass

Table 5.10: Chat Unit Test Case

5.2.2 Stories Module

Stories Unit Test Case

Test Case Filename: main.py **Date Created:** 10 October 2019

Tester(s): Tham Zhao Jin **Access Path:** ..\Chatbot\main.py

Transactions: 1. /stories → /
2. /stories → /login

Purpose of Test Case: To provide useful information about the interactions with the chatbot. To test for errors in page navigation.

Step No.	Input Values to Execute Test Case	Expected Results	Actual Results	Pass / Fail - Comments
1.	User clicks on the “All”, “Chat”, “Mail” or “Sentiment” tag below the “Stories” header.	The chart views of the dashboard are filtered based on the tag clicked by user.	The system filters and displays relevant charts according to the tag clicked by user.	Pass
2.	User clicks on the “Chat” navigation button on the top navigation bar.	User will be redirected to the next page (/chat).	System redirects the user to Chat screen to chat with the system.	Pass
3.	User clicks on the “Stories” navigation button on the top navigation bar.	User will remain in the same page (/stories).	User remains in Stories.	Pass

4.	User clicks on the “Login” navigation button on the top navigation bar.	User will be redirected to the next page (/login).	System redirects the user to Login screen to perform login action for staff.	Pass
----	---	--	--	------

Table 5.11: Stories Unit Test Case

5.2.3 Login Module

Login Unit Test Case

Test Case Filename: main.py **Date Created:** 10 October 2019

Tester(s): Tham Zhao Jin **Access Path:** ..\Chatbot\main.py

Transactions: 1. /login → /
 2. /login → /stories
 3. /login → /intents

Purpose of Test Case: To identify the possible conditions that may cause the system to behave abnormally and making sure that the users will be redirected to the next page if correct username and password were inputted. To test for errors in page navigation.

Step No.	Input Values to Execute Test Case	Expected Results	Actual Results	Pass / Fail - Comments
1.	User inputs valid username and password.	User will be redirected to the next page (/intents) successfully.	User was able to proceed to the Intents screen.	Pass

2.	User inputs valid username but invalid password.	User will not be able to proceed, and an error message will be popped up: <i>Wrong username and password! Please try again.</i>	No error message popped out, but user remains on the Login screen.	Fail
3.	User inputs valid password but invalid username.	User will not be able to proceed, and an error message will be popped up: <i>Wrong username and password! Please try again.</i>	No error message popped out, but user remains on the Login screen.	Fail
4.	User leaves the username and password fields blank.	User will not be able to proceed, and an error message will be popped up: <i>Wrong username and password! Please try again.</i>	No error message popped out, but user remains on the Login screen.	Fail
5.	User clicks on the “Login” navigation button on the top navigation bar.	User will remain in the same page (/login).	User remains in Login screen.	Pass
6.	User clicks on the “Chat” navigation button on the top navigation bar.	User will be redirected to the next page (/chat).	System redirects the user to Chat screen to chat with the system.	Pass

- | | | | |
|----|---|--|---|
| 7. | User clicks on the “Stories” navigation button on the top navigation bar. | User will be redirected to the next page (/stories). | System redirects the user to Pass Stories screen to view dashboard. |
|----|---|--|---|

Table 5.12: Login Unit Test Case

5.2.4 Intents Module

Intents Unit Test Case				
Test Case Filename: main.py		Date Created: 10 October 2019		
Tester(s): Tham Zhao Jin		Access Path: ..\Chatbot\main.py		
Transactions:				
1. /intents → /				
2. /intents → /stories				
3. /intents → /mail				
4. /intents → /logout				
Purpose of Test Case: To ensure that the staff can manage the intents and changes are reflected directly to the database. To test for errors in page navigation.				
Step No.	Input Values to Execute Test Case	Expected Results	Actual Results	Pass / Fail - Comments
1.	Staff clicks on the “< Back to Agents” navigation button on the top Intents card.	Staff will be redirected to the next card view (Agents card) successfully.	Staff was able to proceed to the Agents card.	Pass

- | | | | | |
|----|--|--|--|------|
| 2. | Staff clicks on the “Deploy” button on the Intents card. | Staff will be able to activate the deployment of all listed intents into an ML model. Changes are updated to the database. | System deploys all listed intents into an ML model and update the changes in database. | Pass |
| 3. | Staff clicks on the “Create Intent” button on the Intents card. | Staff will be redirected to the next card view (Create Intent card) successfully. | Staff was able to proceed to the Create Intent card. | Pass |
| 4. | Staff inputs a valid intent name in the “Search intents” search bar on the Intents card to search for an intent. | Intent(s) for that specific intent name will be automatically displayed. | System displays the intent(s) based on intent name search criteria. | Pass |
| 5. | Staff inputs an invalid intent name in the “Search intents” search bar to search for an intent. | No intent will be displayed. | System displays none. | Pass |
| 6. | Staff clicks on the intent name on the Intents card. | Staff will be redirected to the next card view (Update Intent card) successfully. | Staff was able to proceed to the Update Intent card. | Pass |

- | | | | |
|--|---|--|------|
| 7. Staff clicks on the delete icon when hovering on the intent name to delete an intent. | A confirmation message will be popped up:

<i>Are you sure you would like to delete intent "<intent name>"?</i>

If staff clicks on the "Delete" button, then the intent will be deleted and removed from the Intents card; whereas if staff clicks on the "Cancel" button, then the confirmation message will be closed. | System deletes the intent when staff clicks on the "Delete" button, whereas system closes the confirmation message when staff clicks on the "Cancel" button. | Pass |
|--|---|--|------|

- | | | | | |
|-----|---|---|---|------|
| 8. | Staff selects multiple checkboxes or all checkboxes to delete multiple intents or all intents respectively. Then, staff clicks on the “Delete” button appeared on the Intents card. | A confirmation message will be popped up:
<i>Are you sure you would like to delete <number of selected intents> selected intents?</i>

If staff clicks on the “Delete” button, then the intents will be deleted and removed from the Intents card; whereas if staff clicks on the “Cancel” button, then the confirmation message will be closed. | System deletes the intents when staff clicks on the “Delete” button, whereas system closes the confirmation message when staff clicks on the “Cancel” button. | Pass |
| 9. | Staff inputs a user query in the “Try a question” input field on the Test console card and presses “Enter” key to test an intent. | Diagnostic information based on the user query will be displayed to staff. | System displays relevant diagnostic information according to the user query. | Pass |
| 10. | Staff clicks on the “Diagnostics” button on the Test console card to view further diagnostic results. | Top-scoring intent chart and JSON response based on the user query will be displayed to staff. | System displays relevant top-scoring intent chart and JSON response according to the user query. | Pass |

11.	Staff clicks on the “Chat” navigation button on the top navigation bar.	Staff will be redirected to the next page (/chat).	System redirects the staff to Chat screen to chat with the system.	Pass
12.	Staff clicks on the “Stories” navigation button on the top navigation bar.	Staff will be redirected to the next page (/stories).	System redirects the staff to Stories screen to view dashboard.	Pass
13.	Staff clicks on the “Intents” navigation button on the top navigation bar.	Staff will remain in the same page, but the page refreshes (/intents).	Staff remains in Intents screen, but system refreshes the screen.	Pass
14.	Staff clicks on the “Mail” navigation button on the top navigation bar.	Staff will be redirected to the next page (/mail).	System redirects the staff to Mail screen to manage the emails.	Pass
15.	Staff clicks on the “Logout” navigation button on the top navigation bar.	Staff will be redirected back to the first page (/login).	System redirects the staff back to Login screen.	Pass

Table 5.13: Intents Unit Test Case

5.2.5 Intents – Create / Update Module

Intents – Create / Update Unit Test Case

Test Case Filename: main.py

Date Created: 10 October 2019

Tester(s): Tham Zhao Jin

Access Path: ..\Chatbot\main.py

Transactions:

1. /intents → /
2. /intents → /stories
3. /intents → /mail
4. /intents → /logout

Purpose of Test Case: To determine the possible conditions that will be faced by the staff in the process of creating an intent. To test for errors in page navigation.

Step No.	Input Values to Execute Test Case	Expected Results	Actual Results	Pass / Fail - Comments
1.	Staff inputs a valid training phrase name in the “Search training phrases” search bar on the Create Intent card to search for a training phrase.	Training phrase(s) for that specific training phrase name will be automatically displayed.	System displays the training phrase(s) based on training phrase name search criteria.	Pass

- | | | | | |
|----|---|--|--|------|
| 2. | Staff inputs an invalid training phrase name in the “Search training phrases” search bar to search for a training phrase. | No training phrase will be displayed. | System displays none. | Pass |
| 3. | Staff clicks on the “Text Response” or “Quick Replies” button on the Create Intent card to switch view accordingly. | Staff will be able to add text answer(s) when “Text Response” button is clicked, whereas staff will be able to add quick replies when “Quick Replies” button is clicked. | System switches different view between adding a text answer or quick reply. | Pass |
| 4. | Staff leaves all the fields on the Create Intent card blank. Then, staff clicks on the “Save Intent” button. | Staff will not be able to create the intent. | System remains in the same card view (Create Intent card) and no new intent created. | Pass |
| 5. | Staff leaves either field – intent name, training phrases or responses on the Create Intent card blank. Then, staff clicks on the “Save Intent” button. | Staff will not be able to create the intent. | System remains in the same card view (Create Intent card) and no new intent created. | Pass |

- | | | | | |
|-----|---|--|--|------|
| 6. | Staff enters the details for intent name, training phrases and responses on the Create Intent card. Then, staff clicks on the “Save Intent” button. | All intent details will be saved, and changes are updated to the database. Then, the Intents screen will be refreshed, and the Intents card will be displayed. | System inserts all intent details to the database. Then, system refreshes the Intent screen and displays the Intents card. | Pass |
| 7. | Staff inputs a user query in the “Try a question” input field on the Test console card and presses “Enter” key to test an intent. | Diagnostic information based on the user query will be displayed to staff. | System displays relevant diagnostic information according to the user query. | Pass |
| 8. | Staff clicks on the “Diagnostics” button on the Test console card to view further diagnostic results. | Top-scoring intent chart and JSON response based on the user query will be displayed to staff. | System displays relevant top-scoring intent chart and JSON response according to the user query. | Pass |
| 9. | Staff clicks on the “Chat” navigation button on the top navigation bar. | Staff will be redirected to the next page (/chat). | System redirects the staff to Chat screen to chat with the system. | Pass |
| 10. | Staff clicks on the “Stories” navigation button on the top navigation bar. | Staff will be redirected to the next page (/stories). | System redirects the staff to Stories screen to view dashboard. | Pass |
| 11. | Staff clicks on the “Intents” navigation button on the top navigation bar. | Staff will remain in the same page, but the page refreshes (/intents). | Staff remains in Intents screen, but system refreshes the screen. | Pass |

12.	Staff clicks on the “Mail” navigation button on the top navigation bar.	Staff will be redirected to the next page (/mail).	System redirects the staff to Mail screen to manage the emails.	Pass
13.	Staff clicks on the “Logout” navigation button on the top navigation bar.	Staff will be redirected back to the first page (/login).	System redirects the staff back to Login screen.	Pass

Table 5.14: Intents – Create / Update Unit Test Case

5.2.6 Intents – Agents Module

Intents – Agents Unit Test Case				
Test Case Filename: main.py		Date Created: 10 October 2019		
Tester(s): Tham Zhao Jin		Access Path: ..\Chatbot\main.py		
Transactions:				
1. /intents → /				
2. /intents → /stories				
3. /intents → /mail				
4. /intents → /logout				
Purpose of Test Case: To ensure that the staff can manage the agents and changes are reflected directly to the database. To test for errors in page navigation.				
Step No.	Input Values to Execute Test Case	Expected Results	Actual Results	Pass / Fail - Comments
1.	Staff clicks on the “Create Agent” button on the Agents card.	Staff will be redirected to the next card view (Create Agent card) successfully.	Staff was able to proceed to the Create Agent card.	Pass

- | | | | | |
|----|--|--|---|------|
| 2. | Staff inputs a valid agent name in the “Search agents” search bar on the Agents card to search for an agent. | Agent(s) for that specific agent name will be automatically displayed. | System displays the agent(s) based on agent name search criteria. | Pass |
| 3. | Staff inputs an invalid agent name in the “Search agents” search bar to search for an agent. | No agent will be displayed. | System displays none. | Pass |
| 4. | Staff clicks on the agent name on the Agents card. | Staff will be redirected to the next card view (Update Agent card) successfully. | Staff was able to proceed to the Update Agent card. | Pass |
| 5. | Staff clicks on the “Go to Intents >” navigation button on the bottom Agents card. | Staff will be redirected to the next card view (Intents card) successfully. | Staff was able to proceed to the Intents card. | Pass |

- | | | | | |
|----|---|--|--|------|
| 6. | Staff clicks on the delete icon when hovering on the agent name to delete an agent. | <p>A confirmation message will be popped up:</p> <p><i>Are you sure you would like to delete agent "<agent name>"?</i></p> <p>If staff clicks on the “Delete” button, then the agent will be deleted and removed from the Agents card; whereas if staff clicks on the “Cancel” button, then the confirmation message will be closed.</p> | <p>System deletes the agent when staff clicks on the “Delete” button, whereas system closes the confirmation message when staff clicks on the “Cancel” button.</p> | Pass |
| 7. | Staff clicks on the “Chat” navigation button on the top navigation bar. | Staff will be redirected to the next page (/chat). | System redirects the staff to Chat screen to chat with the system. | Pass |
| 8. | Staff clicks on the “Stories” navigation button on the top navigation bar. | Staff will be redirected to the next page (/stories). | System redirects the staff to Stories screen to view dashboard. | Pass |
| 9. | Staff clicks on the “Intents” navigation button on the top navigation bar. | Staff will remain in the same page, but the page refreshes (/intents) and Intents card will be displayed. | Staff remains in Intents screen, but system refreshes the screen and displays the Intents card. | Pass |

10.	Staff clicks on the “Mail” navigation button on the top navigation bar.	Staff will be redirected to the next page (/mail).	System redirects the staff to Mail screen to manage the emails.	Pass
11.	Staff clicks on the “Logout” navigation button on the top navigation bar.	Staff will be redirected back to the first page (/login).	System redirects the staff back to Login screen.	Pass

Table 5.15: Intents – Agents Unit Test Case

5.2.7 Intents – Create / Update Agents Module

Intents – Create / Update Agents Unit Test Case				
Test Case Filename: main.py		Date Created: 10 October 2019		
Tester(s): Tham Zhao Jin		Access Path: ..\Chatbot\main.py		
Transactions:				
1. /intents → /				
2. /intents → /stories				
3. /intents → /mail				
4. /intents → /logout				
Purpose of Test Case: To determine the possible conditions that will be faced by the staff in the process of creating an agent. To test for errors in page navigation.				
Step No.	Input Values to Execute Test Case	Expected Results	Actual Results	Pass / Fail - Comments
1.	Staff leaves the agent name field on the Create Agent card blank. Then, staff clicks on the “Save Agent” button.	Staff will not be able to create the agent.	System remains in the same card view (Create Agent card) and no new agent created.	Pass

2.	Staff enters the details for agent name on the Create Agent card. Then, staff clicks on the “Save Agent” button.	All agent details will be saved, and changes are updated to the database. Then, the Intents screen will be refreshed, and the Intents card will be displayed.	System inserts all agent details to the database. Then, system refreshes the Intent screen and displays the Intents card.	Pass
3.	Staff clicks on the “Chat” navigation button on the top navigation bar.	Staff will be redirected to the next page (/chat).	System redirects the staff to Chat screen to chat with the system.	Pass
4.	Staff clicks on the “Stories” navigation button on the top navigation bar.	Staff will be redirected to the next page (/stories).	System redirects the staff to Stories screen to view dashboard.	Pass
5.	Staff clicks on the “Intents” navigation button on the top navigation bar.	Staff will remain in the same page, but the page refreshes (/intents) and Intents card will be displayed.	Staff remains in Intents screen, but system refreshes the screen and displays the Intents card.	Pass
6.	Staff clicks on the “Mail” navigation button on the top navigation bar.	Staff will be redirected to the next page (/mail).	System redirects the staff to Mail screen to manage the emails.	Pass
7.	Staff clicks on the “Logout” navigation button on the top navigation bar.	Staff will be redirected back to the first page (/login).	System redirects the staff back to Login screen.	Pass

Table 5.16: Intents – Create / Update Agents Unit Test Case

5.2.8 Mail Module

Mail Unit Test Case

Test Case Filename: main.py **Date Created:** 10 October 2019

Tester(s): Tham Zhao Jin **Access Path:** ..\Chatbot\main.py

Transactions: 1. /mail → /
 2. /mail → /stories
 3. /mail → /intents
 4. /mail → /logout

Purpose of Test Case: To ensure that the staff can manage the emails. To test for errors in page navigation.

Step No.	Input Values to Execute Test Case	Expected Results	Actual Results	Pass / Fail - Comments
1.	Staff inputs a valid email details in the “Search mail” search bar on the Mail card to search for an email.	Email(s) for that specific email details based on subject or body will be automatically displayed.	System displays the email(s) based on email details search criteria.	Pass

2.	Staff inputs an invalid email details in the “Search mail” search bar to search for an email.	No email will be displayed.	System displays none.	Pass
3.	Staff clicks on the email on the Mail card.	Staff will be redirected to the next card view (View Detailed Mail card) successfully.	Staff was able to proceed to the View Detailed Mail card.	Pass
4.	Staff clicks on the “Chat” navigation button on the top navigation bar.	Staff will be redirected to the next page (/chat).	System redirects the staff to Chat screen to chat with the system.	Pass
5.	Staff clicks on the “Stories” navigation button on the top navigation bar.	Staff will be redirected to the next page (/stories).	System redirects the staff to Stories screen to view dashboard.	Pass
6.	Staff clicks on the “Intents” navigation button on the top navigation bar.	Staff will be redirected to the next page (/intents).	System redirects the staff to Intents screen to manage the intents.	Pass
7.	Staff clicks on the “Mail” navigation button on the top navigation bar.	Staff will remain in the same page, but the page refreshes (/mail) and Mail card will be displayed.	Staff remains in Mail screen, but system refreshes the screen and displays the Mail card.	Pass
8.	Staff clicks on the “Logout” navigation button on the top navigation bar.	Staff will be redirected back to the first page (/login).	System redirects the staff back to Login screen.	Pass

Table 5.17: Mail Unit Test Case

5.2.9 Mail – View Detailed Module

Mail – View Detailed Unit Test Case				
Test Case Filename: main.py		Date Created: 10 October 2019		
Tester(s): Tham Zhao Jin		Access Path: ..\Chatbot\main.py		
Transactions:				
1. /mail → /				
2. /mail → /stories				
3. /mail → /intents				
4. /mail → /logout				
Purpose of Test Case: To determine the possible conditions that will be faced by the staff in the process of creating a follow-up email if needed. To test for errors in page navigation.				
Step No.	Input Values to Execute Test Case	Expected Results	Actual Results	Pass / Fail - Comments
1.	Staff leaves all the fields on the View Detailed Mail card blank. Then, staff clicks on the “Send” button.	Staff will not be able to create and send the follow-up email.	System remains in the same card view (View Detailed Mail card) and no new follow-up email created and sent.	Pass

- | | | | | |
|----|--|---|--|------|
| 2. | Staff leaves either field – follow-up subject or follow-up body on the View Detailed Mail card blank. Then, staff clicks on the “Send” button. | Staff will not be able to create and send the follow-up email. | System remains in the same card view (View Detailed Mail card) and no new follow-up email created and sent. | Pass |
| 3. | Staff enters the details for follow-up subject and follow-up body on the View Detailed Mail card. Then, staff clicks on the “Send” button. | All follow-up email details will be saved, and changes are updated to the database. Then, the Mail screen will be refreshed, and the Mail card will be displayed. | System inserts all follow-up email details to the database. Then, system refreshes the Mail screen and displays the Mail card. | Pass |
| 4. | Staff clicks on the “Chat” navigation button on the top navigation bar. | Staff will be redirected to the next page (/chat). | System redirects the staff to Chat screen to chat with the system. | Pass |
| 5. | Staff clicks on the “Stories” navigation button on the top navigation bar. | Staff will be redirected to the next page (/stories). | System redirects the staff to Stories screen to view dashboard. | Pass |
| 6. | Staff clicks on the “Intents” navigation button on the top navigation bar. | Staff will be redirected to the next page (/intents). | System redirects the staff to Intents screen to manage the intents. | Pass |
| 7. | Staff clicks on the “Mail” navigation button on the top navigation bar. | Staff will remain in the same page, but the page refreshes (/mail) and Mail card will be displayed. | Staff remains in Mail screen, but system refreshes the screen and displays the Mail card. | Pass |

- | | | | | |
|----|---|---|--|------|
| 8. | Staff clicks on the “Logout” navigation button on the top navigation bar. | Staff will be redirected back to the first page (/login). | System redirects the staff back to Login screen. | Pass |
|----|---|---|--|------|

Table 5.18: Mail – View Detailed Unit Test Case

5.3 Performance Measure

This section provides the results of quantifying the quality of the chatbot's ML module, specifically the Retrieval-based Model. This model uses Naïve Bayes approach to hold the conversations. The intent classification task is examined as follows.

	precision	recall	f1-score	support
Accreditation	0.00	0.00	0.00	1
Completion_Duration	1.00	1.00	1.00	1
Contact	1.00	1.00	1.00	2
Fees	1.00	1.00	1.00	1
General.Sentiments_Angry	0.00	0.00	0.00	1
General.Sentiments_Happy	0.00	0.00	0.00	0
Programmes	1.00	1.00	1.00	2
Programmes.Postgraduate	0.00	0.00	0.00	3
Programmes.Undergraduate	0.00	0.00	0.00	1
Programmes.Undergraduate.Communications_and_Networking	1.00	1.00	1.00	1
Programmes.Undergraduate.Computer_Engineering	0.00	0.00	0.00	1
Programmes.Undergraduate.Computer_Science	0.00	0.00	0.00	1
Programmes.Undergraduate.Information_Analysis	0.00	0.00	0.00	0
Programmes.Undergraduate.Information_Business	0.00	0.00	0.00	1
Why_ICT	1.00	1.00	1.00	1
accuracy			0.47	17
macro avg	0.40	0.40	0.40	17
weighted avg	0.47	0.47	0.47	17

Figure 5.1: Classification Report for Retrieval-based Model

Figure 5.1 shows the quality of predictions from Retrieval-based Model based on each label in terms of precision, recall and f1-score. Briefly, precision is the accuracy concerning positive predictions. Recall is the fraction of positives that were correctly identified. F1-score is the weighted harmonic mean of precision and recall whereby 1.00 is the best score and 0.00 is the worst. Furthermore, an accuracy of 0.47 or 47% is achieved due to the dimensionality of training samples. Naïve Bayes approach works best on binary predictions. Nevertheless, this approach is still feasible in this model that involves multiclass predictions. In addition, the deployment of this model is dynamic whereby the staff can manage the labels and corresponding training data to meet the ever-changing user requirements. Not much training data will be needed to train the model because Naïve Bayes approach performs considerably well if the features are well-engineered.

For the intent classification, the test dataset is used to derive the confusion matrix of Retrieval-based Model. The intent classifier is trained within FICT FAQs scope.

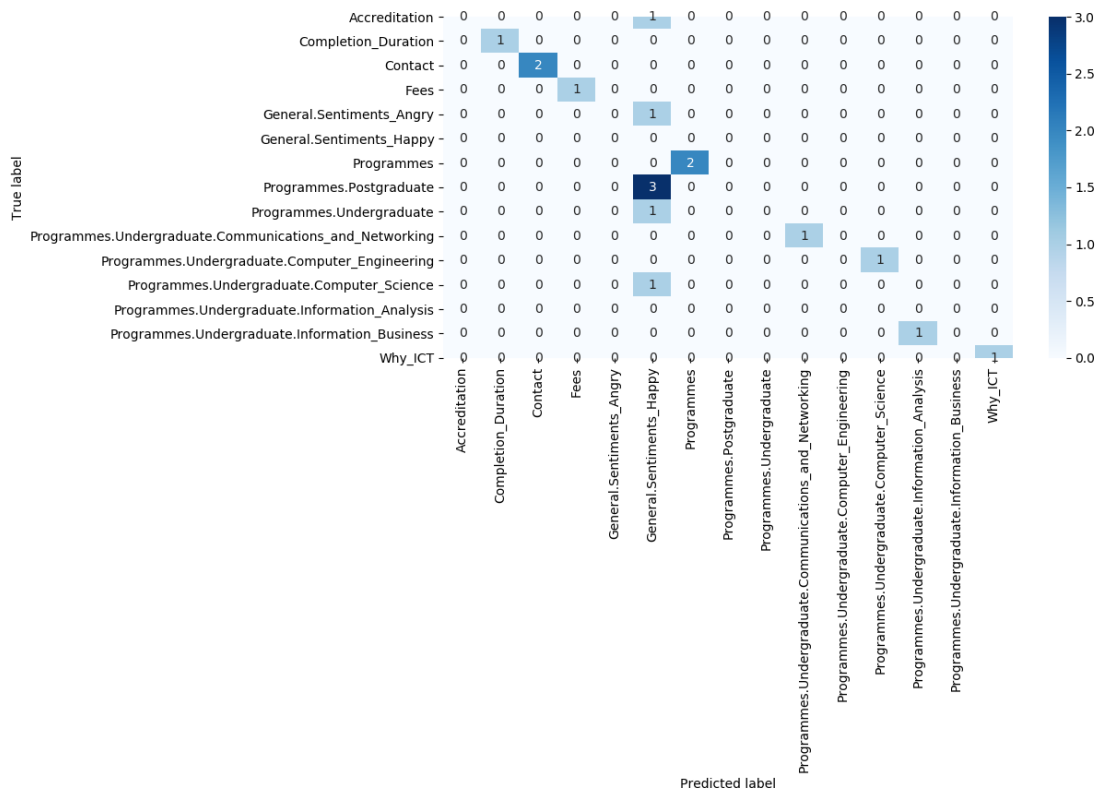


Figure 5.2: Confusion Matrix for Retrieval-based Model

The FICT agent seems to perform poorly in correctly identifying most users’ intents. For example, it seems to confuse some “Programmes.Postgraduate” samples for the “General.Sentiments_Happy” class. This could be due to two scenarios – human errors in labelling the samples or the samples are not unique enough to differentiate accurately between the two classes. In addition, the test dataset is considerably small after train-test split on the training samples. Hence, the agent will be running in a mockup environment considering it is not robust enough to be used in a real university QnA environment yet. Nonetheless, although the performance measure for the Retrieval-based Model not being as good as it should be, the chatbot is still able to handle concrete user queries in practice. Careful engineering the features will improve the results of this model. For instance, maintaining the uniqueness of each feature.

CHAPTER 6: CONCLUSION

6.1 Project Review, Discussions and Conclusions

This project aimed to develop an automated AI-powered chatbot for University marketing and promotion teams. Starting from the project scope and objectives, the system design was laid down on paper in Chapter 3. Schematic diagrams were provided to illustrate the connection of all the components and how the chatbot operates. Furthermore, in Chapter 4, a scalable software solution was designed and developed for training and deploying the chatbot. The chatbot design was decomposed into several modules later in the chapter. Moreover, in Chapter 5, a manual way of testing the chatbot on usability and functionality was presented.

Based on the chatbot development implementation in response to reducing manual work done for information querying, it can be concluded that operational inefficiencies and FAQ site ineffectiveness are reduced. The deployment demonstrates that the chatbot is still in learning phase whereby not all possible interactions can be handled. Hence, the NLP Engine using Naïve Bayes approach should be improved.

Overall, the chatbot solution developed is considered satisfying due to ease of development and deployment. It is hosted in a dedicated Python environment using Flask Pycoco. Nevertheless, the solution may not be perfect but considering the requirements, it works well in practice. There might be a few loopholes whereby the chatbot does not operate as intended. Those who apply the approach derived from the outcomes of this project would need to take note.

6.2 Novelties and Contributions

Project II works are discussed as follows. Besides using text format as response to users, rich messages such as quick replies were included to promote a more engaging chat experience. Furthermore, a Bot Model Selection screen was provided to allow staff deploying specific bot model such as “FICT_Agent”, “FBF_Agent”, etc. This reduces the vocabulary size used for predicting intents. Hence, better performance and higher accuracy of the chatbot can be achieved. Moreover, live chat, email automation and social platform integration (Facebook) were done to extend the services of the chatbot.

6.3 Future Work

Several improvements are suggested in this section to extend the realm of possibilities for the chatbot. They are regarded more as design ideas than actual implementation to follow. The ideas are not confirmed whether they are feasible in the university domain.

First and foremost, recommendation of relevant answers based on what users are typing. This reduces the need for users to type out their query in full sentence. In addition, Bahasa Malaysia to English translation and speech recognition could be introduced. To conclude, these design ideas are merely suggestions. Hence, they are excluded from this project considering their feasibility.

BIBLIOGRAPHY

- Amareesan, S n.d., *What Is a Knowledge Base, and Why Do You Need One?* Available from: <<https://blog.hubspot.com/service/what-is-a-knowledge-base>>. [20 July 2019].
- Banerjee, S 2018, *Introduction to Named Entity Recognition*. Available from: <<https://medium.com/explore-artificial-intelligence/introduction-to-named-entity-recognition-eda8c97c2db1>>. [20 July 2019].
- Basics of Software Life Cycle and Waterfall Model*, n.d. Available from: <<https://nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Soft%20Engg/pdf/m02L03.pdf>>. [10 August 2019].
- Bassil, Y 2012, *A Simulation Model for the Waterfall Software Development Life Cycle*. Available from: <<https://arxiv.org/ftp/arxiv/papers/1205/1205.6904.pdf>>. [10 August 2019].
- Cameron, P n.d., *Automated Operations: 5 Benefits of Automation*. Available from: <<https://www.helpsystems.com/resources/guides/automated-operations-5-benefits-your-organization>>. [6 July 2019].
- Chandrayan, P 2019, *Understanding AI Chatbots, Challenges, Opportunities & Beyond*. Available from: <<https://towardsdatascience.com/understanding-ai-chatbots-challenges-opportunities-beyond-fb657fa3e0da>>. [6 July 2019].
- Definition of chatbot in English*, n.d. Available from: <<https://www.lexico.com/en/definition/chatbot>>. [6 July 2019].
- Flask Python Framework*, n.d. Available from: <<https://quintagroup.com/cms/python/flask>>. [10 August 2019].
- Gupta, S 2018, *Sentiment Analysis: Concept, Analysis and Applications*. Available from: <<https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17>>. [13 July 2019].

- Introducing SQL Server 2019, 2019*. Available from: <<https://www.microsoft.com/en-us/sql-server/sql-server-2019>>. [10 August 2019].
- Lincoln University – Reports and Statistics, n.d.* Available from: <<https://library2.lincoln.ac.nz/dashboard/>>. [9 August 2019].
- Luzhanskiy, I n.d., *AI Chatbot Project Management*. Available from: <<https://chatbots.studio/wp-content/uploads/2019/04/methodology-.pdf>>. [18 November 2019].
- McGrath, C 2017, *Chatbot Vocabulary: 10 Chatbot Terms You Need to Know*. Available from: <<https://chatbotsmagazine.com/chatbot-vocabulary-10-chatbot-terms-you-need-to-know-3911b1ef31b4>>. [13 July 2019].
- Meaning of FAQ in English, n.d.* Available from: <<https://dictionary.cambridge.org/dictionary/english/faq>>. [6 July 2019].
- Microsoft 2002, *Microsoft Solutions Framework*. Available from: <<http://download.microsoft.com/download/8/7/e/87eeff7e-05d2-418a-900d-4896ae4e20db/msf%20team%20model%20v.3.1.pdf>>. [18 November 2019].
- Microsoft 2019, *Visual Studio Code*. Available from: <<https://code.visualstudio.com/>>. [19 November 2019].
- Moppel, J 2018, *Socratic chatbot*. Available from: <https://comserv.cs.ut.ee/home/files/moppel_informaatika_2018.pdf?study=ATILoputoo&reference=993918BCC4306ABB070B26E697983ABAA9E409A8>. [6 July 2019].
- Natural Language Processing with Python, n.d.* Available from: <<https://www.nltk.org/book/>>. [10 August 2019].
- Navlani, A 2018, *Naive Bayes Classification using Scikit-learn*. Available from: <<https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn>>. [19 November 2019].
- Nizri, G n.d., *Using Intelligent Automation to Reduce Operating Costs*. Available from: <<https://ayehu.com/using-intelligent-automation-reduce-operating-costs/>>. [6 July 2019].

- nubibot – Revolutionize Your Customer Engagement Strategy with AI Chatbot 2019.* Available from: <<https://nubitel.co/nubibot>>. [13 July 2019].
- Peters, F 2018, *Design and implementation of a chatbot in the context of customer support.* Available from: <https://matheo.uliege.be/bitstream/2268.2/4625/6/Thesis_PETERS_Florian.pdf>. [6 July 2019].
- QnA Maker – From data to bot in minutes 2019.* Available from: <<https://www.qnamaker.ai/>>. [13 July 2019].
- scikit-learn developers 2019, *sklearn.naive_bayes.MultinomialNB.* Available from: <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html>. [19 November 2019].
- Sentiment Analysis Explained, n.d.* Available from: <<https://www.lexalytics.com/technology/sentiment-analysis#how-used>>. [13 July 2019].
- Shridhar, K 2017, *Rule based bots vs AI bots.* Available from: <<https://medium.com/botsupply/rule-based-bots-vs-ai-bots-b60cdb786ffa>>. [13 July 2019].
- Tesco 2017.* Available from: <<https://www.tesco.com.my/>>. [9 August 2019].
- Thakur, D n.d., *Types of Software Maintenance.* Available from: <<http://ecomputernotes.com/software-engineering/types-of-software-maintenance>>. [10 August 2019].
- What Is Automation? n.d.* Available from: <<https://www.isa.org/about-isa/what-is-automation/>>. [6 July 2019].
- Wong, YT n.d., *Software Development Life Cycle.* Available from: <https://www.cse.ust.hk/~rossiter/independent_studies_projects/software_development/software_development_report.pdf>. [10 August 2019].

FINAL YEAR PROJECT WEEKLY REPORT

(*Project I / Project II*)

Trimester, Year: Trimester 3, Year 3	Study week no.: 1
Student Name & ID: THAM ZHAO JIN 16ACB01219	
Supervisor: DR. PRADEEP A/L ISAWASAN	
Project Title: A Conversational Question and Answer Chatbot for University	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Plan project 2 works.
- Build Agents module.

2. WORK TO BE DONE

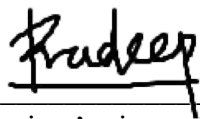
- Update Intents and Chat modules.
- Implement quick replies.

3. PROBLEMS ENCOUNTERED

- None.

4. SELF EVALUATION OF THE PROGRESS

- In progress.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 3
Student Name & ID: THAM ZHAO JIN 16ACB01219	
Supervisor: DR. PRADEEP A/L ISAWASAN	
Project Title: A Conversational Question and Answer Chatbot for University	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Update Intents and Chat modules.
- Implement quick replies.

2. WORK TO BE DONE

- Build and train new model for the chatbot.
- Build emailbot.

3. PROBLEMS ENCOUNTERED

- Lack of main system and front-end framework. The chatbot application should be an embedded function to the main system.

4. SELF EVALUATION OF THE PROGRESS

- In progress.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 5
Student Name & ID: THAM ZHAO JIN 16ACB01219	
Supervisor: DR. PRADEEP A/L ISAWASAN	
Project Title: A Conversational Question and Answer Chatbot for University	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Build and train new model for the chatbot.
- Build emailbot.

2. WORK TO BE DONE

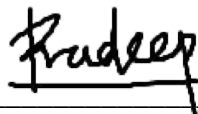
- Update Stories module.
- Integrate Facebook Graph API.
- Build Login module and live chat.

3. PROBLEMS ENCOUNTERED

- None.

4. SELF EVALUATION OF THE PROGRESS

- In progress.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 7
Student Name & ID: THAM ZHAO JIN 16ACB01219	
Supervisor: DR. PRADEEP A/L ISAWASAN	
Project Title: A Conversational Question and Answer Chatbot for University	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Update Stories module.
- Integrate Facebook Graph API.
- Build Login module and live chat.

2. WORK TO BE DONE

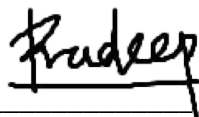
- UI redesign.
- Testing and debugging.

3. PROBLEMS ENCOUNTERED

- None.

4. SELF EVALUATION OF THE PROGRESS

- In progress.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(*Project I / Project II*)

Trimester, Year: Trimester 3, Year 3	Study week no.: 9
Student Name & ID: THAM ZHAO JIN 16ACB01219	
Supervisor: DR. PRADEEP A/L ISAWASAN	
Project Title: A Conversational Question and Answer Chatbot for University	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- UI redesign.
- Testing and debugging.

2. WORK TO BE DONE

- Documentation.

3. PROBLEMS ENCOUNTERED

- None.

4. SELF EVALUATION OF THE PROGRESS

- In progress.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(*Project I / Project II*)

Trimester, Year: Trimester 3, Year 3	Study week no.: 11
Student Name & ID: THAM ZHAO JIN 16ACB01219	
Supervisor: DR. PRADEEP A/L ISAWASAN	
Project Title: A Conversational Question and Answer Chatbot for University	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Documentation.

2. WORK TO BE DONE

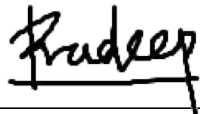
- Presentation.

3. PROBLEMS ENCOUNTERED

- MCO.

4. SELF EVALUATION OF THE PROGRESS

- Done.



Supervisor's signature



Student's signature

POSTER



A CONVERSATIONAL QUESTION AND ANSWER CHATBOT FOR UNIVERSITY

Done By: Tham Zhao Jin

Supervised By: Dr. Pradeep A/L Isawasan

「INTRODUCTION」

This project is built and hosted in Flask Pocoo. The chatbot is powered by an NLP (Natural Language Processing) Engine to automate conversations. The conversations are limited to University domain.

「METHODS」

AI Chatbot Development Methodology based on MSF (Microsoft Solutions Framework) is adopted. Microsoft Visual Studio Code and Microsoft SQL Server are used for development.



「DISCUSSION」

The features of this project include Intent Classification, Quick Replies, Dashboard, Sentiment Analysis, Live Chat, Email Automation and Facebook Integration.

「CONCLUSION」

In future work, the chatbot can be enhanced with more features such as Relevent Answers Recommendation, Multi-language Support and Speech Recognition.

SIGNED TURNITIN REPORT

Turnitin Originality Report Document Viewer

Processed on: 13-Apr-2020 09:20 +08
ID: 128605942
Word Count: 18113
Submitted: 2
Chatbot_FYP2 By Tham Zhao Jin

Similarity Index	Similarity by Source
7%	Internet Sources: 1%
	Bibliography: 1%
	Student Papers: 5%

[exclude quoted](#) [exclude bibliography](#) [exclude small matches](#) mode: [quickview \(classic\) report](#) | [Change mode](#) [print](#) [download](#)

1% match (student papers from 29-May-2014) Submitted to Curtin University of Technology on 2014-05-29
1% match (student papers from 14-Sep-2010) Submitted to University of York on 2010-09-14
1% match (student papers from 24-Apr-2015) Submitted to Whitireia Community Polytechnic on 2015-04-24
1% match (student papers from 07-Mar-2011) Submitted to Drexel University on 2011-03-07
<1% match (student papers from 23-Jun-2014) Submitted to North East Surrey College of Technology, Surrey on 2014-06-23
<1% match (student papers from 21-Jun-2019) Submitted to Asia Pacific Institute of Information Technology on 2019-06-21
<1% match (Internet from 01-Jan-2020) http://prints.utar.edu.my
<1% match (student papers from 06-Dec-2012) Submitted to CSU, Fullerton on 2012-12-06
<1% match (student papers from 10-Feb-2020) Submitted to Multimedia University on 2020-02-10
<1% match (student papers from 24-May-2016) Submitted to University of York on 2016-05-24
<1% match (student papers from 09-Apr-2017) Submitted to Aston University on 2017-04-09
<1% match (student papers from 05-Sep-2018) Submitted to University of East London on 2018-09-05
<1% match (student papers from 07-Aug-2009) Submitted to Middlesex University on 2009-08-07
<1% match (student papers from 17-Feb-2011)

Chatbot_FYP2

ORIGINALITY REPORT

7 %	1 %	1 %	7 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of York Student Paper	1 %
2	Submitted to Curtin University of Technology Student Paper	1 %
3	Submitted to Whitireia Community Polytechnic Student Paper	1 %
4	Submitted to Drexel University Student Paper	1 %
5	Submitted to North East Surrey College of Technology, Surrey Student Paper	<1 %
6	Submitted to Asia Pacific Institute of Information Technology Student Paper	<1 %
7	Submitted to University of Greenwich Student Paper	<1 %
8	eprints.utar.edu.my Internet Source	<1 %

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	THAM ZHAO JIN
ID Number(s)	16ACB01219
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONS)
Title of Final Year Project	A CONVERSATIONAL QUESTION AND ANSWER CHATBOT FOR UNIVERSITY

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u>7</u> % Similarity by source Internet Sources: <u>1</u> % Publications: <u>1</u> % Student Papers: <u>7</u> %	
Number of individual sources listed of more than 3% similarity: <u>0</u>	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: PRADEEP ISAWASAN

Date: 19/04/2020

Signature of Co-Supervisor

Name: _____

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	16ACB01219
Student Name	THAM ZHAO JIN
Supervisor Name	DR. PRADEEP A/L ISAWASAN

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Front Cover
✓	Signed Report Status Declaration Form
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

<p>I, the author, have checked and confirmed all the items listed in the table are included in my report.</p> <div style="text-align: center; margin-top: 20px;"> </div> <p>_____ (Signature of Student) Date: 18 April 2020</p>	<p>Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.</p> <div style="text-align: center; margin-top: 20px;"> </div> <p>_____ (Signature of Supervisor) Date: 19/04/2020</p>
--	---

