

MediBot: UTAR Hospital AI Health Companion

BY

TONG JIA SENG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

**BACHELOR OF INFORMATION SYSTEMS (HONOURS)
BUSINESS INFORMATION SYSTEMS**

Faculty of Information and Communication Technology

(Kampar Campus)

JANUARY 2025

COPYRIGHT STATEMENT

© 2025 Tong Jia Seng. All rights reserved.

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Business Information Systems at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I really would like to thanks to my supervisors, Dr Abdulkarim Kanaan Jebna who has given teach me a lot for my fyp project. Thank you.

Secondly, I want to thanks to my fellow friends and family on giving positive feedback and provide me supportive environment during facing difficulties.

ABSTRACT

This proposal introduces a project aimed at enhancing the user experience on UTAR Hospital's platform by developing an English-Chinese multilingual chatbot that provides personalized medical guidance through doctor recommendations and disease prediction. The chatbot leverages advanced technologies such as the LLaMA transformer model, Retrieval-Augmented Generation (RAG), and natural language processing (NLP) to interact with users in a natural, friendly, and informative way.

The core of the project lies in the chatbot's ability to understand user-described symptoms and predict the most likely disease category using machine learning techniques, such as Random Forest Classifier, Logistic Regression, Xgboost Classifier. Based on the prediction, the chatbot recommends suitable doctors from UTAR Hospital's Traditional and Complementary Medicine Centre for further consultation.

RAG plays a key role in generating human-like responses by combining retrieved information with natural language generation, ensuring the conversation feels more engaging and helpful. The chatbot's multilingual capability, supporting both English and Chinese, enables it to assist a wider and more diverse range of users, particularly in Malaysia's multicultural context.

Additionally, the system incorporates a similarity search mechanism using a temporary vector database to improve the accuracy and relevance of responses. It also features an integrated online appointment system to streamline consultation scheduling and reduce reliance on manual processes.

Overall, this project aims to enhance healthcare accessibility through a multilingual chatbot that supports a diverse user base by providing symptom-based disease prediction, personalized doctor recommendations, and streamlined appointment scheduling based on the predicted disease category.

Area of Study: Machine Learning, Web Development; **Keyword:** Bilingual Chatbot, RAG, Llama model, UTAR hospital, T&CM centre, Dashboard, Appointment System

Table of Contents

TITLE PAGE.....	I
COPYRIGHT STATEMENT	II
ACKNOWLEDGEMENTS	III
ABSTRACT	IV
LIST OF FIGURES	VIII
LIST OF TABLES	XI
LIST OF ABBREVIATIONS	XIII
CHAPTER 1: INTRODUCTION.....	1
1.0 Background	1
1.1 Problem Statement and Motivation	2
1.2 Research Objectives	4
1.3 Project Scope and Direction.....	6
1.4 Contributions.....	7
1.5 Report Organization	8
CHAPTER 2: LITERATURE REVIEWS	10
2.1 Previous works on Chatbot Application	10
2.1.1 Mount Sinai hospital chatbot.....	10
2.1.2 Zydus hospital chatbot (ZyE)	12
2.1.3 UCLA Health chatbot.....	13
2.2 Reviews on Technology.....	15
CHAPTER 3: SYSTEM METHODOLOGY	30
3.1 System Design Diagram.....	33
3.1.1 System Architecture Diagram	33
3.1.2 Use Case Diagram	34

3.1.3	Use Case Description	36
3.1.4	ERD Diagram	44
CHAPTER 4: SYSTEM DESIGN		46
4.1	System Block Diagram	46
4.2	System Components Specifications	47
4.2.1	RAG Development Flow	47
4.2.2	Classification Model Development Flow.....	48
4.2.3	Flow on connection between Google Colab and PHP Web Application.....	50
4.2.4	Chinese-English Translation Module Flow	51
CHAPTER 5: SYSTEM IMPLEMENTATION		52
5.1	Hardware Setup	52
5.2	Software Setup	52
5.3	Business Understanding	55
5.4	Data Understanding	56
5.5	Data Preparation.....	59
5.6	Modeling	62
5.7	Evaluation.....	66
5.8	Deployment.....	87
CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION		96
6.1	System Testing and Performance Metrics	96
6.1.1	Classification model evaluation	96
6.1.2	Chatbot Response Testing.....	97
6.1.3	Testing on functionality module	101
6.2	Project Challenges	103
6.3	Objectives Evaluation.....	104
CHAPTER 7: CONCLUSION AND RECOMMENDATION		107

7.1	Conclusion	107
7.2	Recommendation.....	107
REFERENCES.....		108
APPENDIX.....		1
POSTER.....		1

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.0.1	Statistic on Adults with English Proficiency comparison on health	1
Figure 2.1.1.1	Mount Sinai Chatbot Interface	10
Figure 2.1.1.2	Mount Sinai Chatbot Interface	10
Figure 2.1.1.2	Mount Sinai Chatbot Symptom Checker (Demo)	11
Figure 2.1.2.1	ZyE Interface	12
Figure 2.1.2.2	Zye Symptom Checker (Demo)	12
Figure 2.1.3.1	UCLA Health Chatbot Interface	13
Figure 2.2.1	System architecture of proposed chatbot	17
Figure 2.2.2	Architecture of the Proposed RAG-powered CancerBot	22
Figure 2.2.3	RAG workflow	29
Figure 3.1	CRISP-DM Methodology	30
Figure 3.1.1.1	System Architecture Diagram	33
Figure 3.1.2.1	Use Case Diagram	34
Figure 3.1.4.1	ERD diagram for Database in MySQL	44
Figure 3.1.4.2	ERD Diagram for Database in Supabase	45
Figure 4.1.1	System Block Diagram	46
Figure 4.2.1.1	RAG Development Flow Diagram	47
Figure 4.2.2.1	Classification Model Development Flow Diagram	48
Figure 4.2.3.1	Connection Diagram on two platforms	50
Figure 4.2.4.1	Chinese-English Translation Module Flow Diagram	51
Figure 5.2.1	Load Llama3 med42 8b model	55
Figure 5.4.1	Dataset example	57
Figure 5.4.2	Disease Categories with Encoded class	57
Figure 5.4.3	Practitioners categories by disease	57
Figure 5.4.4	Quantity of dataset based on disease categories	58
Figure 5.4.5	RAG dataset	58
Figure 5.4.6	RAG dataset stored example without tag	59

Figure 5.5.1	Oversampling dataset (Embedding)	59
Figure 5.5.2	Oversampling dataset (TF-IDF)	59
Figure 5.5.3	Dataset preprocessed	60
Figure 5.5.4	Tokenization Example	61
Figure 5.5.5	Embedding Example	61
Figure 5.5.6	TF-IDF Vector Representation Example	62
Figure 5.6.1	Data splitting	62
Figure 5.6.2	Adasyn oversampling on training dataset	63
Figure 5.6.3	GridSearch CV result without oversampling	63
Figure 5.6.4	Random Forest Classifier	63
Figure 5.6.5	Logistic Regression	64
Figure 5.6.6	Complement Naïve Bayes	64
Figure 5.6.7	Xgboost Classifier	65
Figure 5.7.1	Evaluation on Random Forest Classifier (Embedding)	66
Figure 5.7.2	Evaluation on Test case and Stability	67
Figure 5.7.3	Confusion Matrix of Random Forest Classifier	68
Figure 5.7.4	Evaluation on Random Forest Classifier (TF-IDF)	69
Figure 5.7.5	Evaluation on Test case and Stability	70
Figure 5.7.6	Confusion Matrix of Random Forest Classifier	71
Figure 5.7.7	Evaluation on Logistic Regression (Embedding)	72
Figure 5.7.8	Evaluation on Test case and Stability	73
Figure 5.7.9	Confusion Matrix of Logistic Regression	74
Figure 5.7.10	Evaluation on Logistic Regression (TF-IDF)	75
Figure 5.7.11	Evaluation on Test case and Stability	76
Figure 5.7.12	Confusion Matrix of Logistic Regression	77
Figure 5.7.13	Evaluation on Xgboost Classifier (Embedding)	78
Figure 5.7.14	Evaluation on Test case and Stability	79
Figure 5.7.15	Confusion Matrix of XgboostClassifier	80
Figure 5.7.16	Evaluation on Xgboost Classifier (TF-IDF)	81
Figure 5.7.17	Evaluation on Test case and Stability	82
Figure 5.7.18	Confusion Matrix of Xgboost Classifier (TF-IDF)	83

Figure 5.7.19	Evaluation on Naïve Bayes (TF-IDF)	84
Figure 5.7.20	Evaluation on Test case and Stability	85
Figure 5.7.21	Confusion Matrix of Naïve Bayes (TF-IDF)	86
Figure 5.8.1	Show case for English Language Chatbot Interaction	87
Figure 5.8.2	Show case for Chinese Language Chatbot Interaction	88
Figure 5.8.3	Home Page	88
Figure 5.8.4	Sign Up Page	89
Figure 5.8.5	Login Page	89
Figure 5.8.6	Forgot Password Page	90
Figure 5.8.7	Book appointment page	90
Figure 5.8.8	View/Cancel appointment page	91
Figure 5.8.9	Dashboard on disease predicted by category	92
Figure 5.8.10	Dashboard on user engagement with chatbot	92
Figure 5.8.11	Dashboard on doctor appointments distribution	93
Figure 5.8.12	User-Doctor chatting page	93
Figure 5.8.13	Print Report Page	94
Figure 5.8.14	Disease List Page	94
Figure 5.8.15	View Appointments Booked by user (Doctor Side)	95
Figure 6.1.2.1	Chatbot Test Case 1	97
Figure 6.1.2.2	Model Prediction Result	98
Figure 6.1.2.3	Chatbot Test Case 2	99
Figure 6.1.2.4	Model Prediction Result	100
Figure 6.1.2.5	Chatbot Test Case 3	101
Figure 6.1.2.6	Model Prediction Result	101

LIST OF TABLES

Table Number	Title	Page
Table 2.1.1	Strength and Weakness of Mount Sinai Chatbot	11
Table 2.1.2	Strength and Weakness of ZyE Chatbot	13
Table 2.1.3	Strength and Weakness of UCLA Health Chatbot	14
Table 2.1.4	Summary of Review Chatbot & Proposed Chatbot	14
Table 2.2.1	Tag explanation for AIML techniques	19
Table 2.2.2	Summary on Literature Reviews	24
Table 2.2.3	Vector Representation Method	27
Table 3.1.3.1	Use case description 1: Enquire symptom for disease prediction and doctor recommendation	36
Table 3.1.3.2	Use case description 2: View Disease List	37
Table 3.1.3.3	Use case description 3: Schedule Appointment	38
Table 3.1.3.4	Use case description 4: Print Report	39
Table 3.1.3.5	Use case description 5: Chat	40
Table 3.1.3.6	Use case description 6: View Appointment Booked by User	41
Table 3.1.3.7	Use case description 7: View Dashboard	42
Table 3.1.3.8	Use case description 8: Export Predicted Dataset	43
Table 5.1.1	Specifications of laptops	52
Table 5.2.1	Tool	52
Table 5.2.2	Programming Language	53
Table 5.2.3	Python Library	53
Table 5.2.4	Specification of Transformer model	54
Table 6.1.1.1	Summary on Classification Model Performance	96
Table 6.1.2.1	Top 5 retrieve texts	98
Table 6.1.2.2	Top 5 retrieve texts	99
Table 6.1.2.3	Top 5 retrieve texts	100
Table 6.1.3.1	Test Case on Admin Dashboard Module	101
Table 6.1.3.2	Test Case on Translation Module	102

Table 6.1.3.3	Test Case on Appointment Module	102
Table 6.1.3.4	Test Case on User-Doctor Chat Module	103

LIST OF ABBREVIATIONS

<i>UTAR</i>	University Tunku Abdul Rahman
<i>T&CM</i>	Traditional & Complementary Medicine
<i>NLP</i>	Natural Language Processing
<i>TF-IDF</i>	Term Frequency – Inverse Document Frequency
<i>CRISP-DM</i>	Cross Industry Standard Process for Data Mining
<i>HTML</i>	Hyper Text Markup Language
<i>CSS</i>	Cascading Style Sheet
<i>NLTK</i>	Natural Language Toolkit
<i>SGD</i>	Stochastic Gradient Descent
<i>GRU</i>	Gated Recurrent Unit
<i>LSTM</i>	Long-Short Term Memory
<i>RNN</i>	Recurrent Neural Network
<i>RAG</i>	Retrieval-augmented Generation
<i>SVG</i>	Stochastic Gradient Descent

CHAPTER 1: Introduction

1.0 Background

This chapter provides the background and motivation for our research, highlights our contributions to the field, and presents the structure of the thesis.

In today's digital world, the application of artificial intelligence (AI) in healthcare is expanding rapidly, offering innovative solutions to improve communication and access to medical services. One such solution involves the use of AI-powered chatbots, which can assist patients in obtaining health-related information, predicting possible diseases based on symptoms, and connecting with appropriate healthcare providers.

This project is motivated by real and pressing challenges observed in Malaysia's healthcare system, especially among the elderly and multilingual communities. One of the primary issues is the **language barrier** between patients and healthcare professionals. Figure 1.0.1 shows the statistical graph on adults with limited English proficiency are 15% more in fair or poor health compared to English proficient adults.

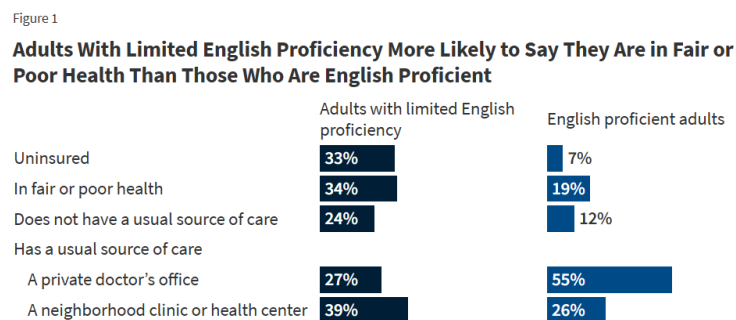


Figure 1.0.1 Statistic on Adults with English Proficiency comparison on health issue [8]

Elderly individuals with limited proficiency in Malay or English often encounter difficulties in describing their symptoms accurately. This can result in miscommunication, delayed diagnoses, and inappropriate treatments. In many cases, untrained staff or family members are relied upon for translation, which can lead to medical errors and reduced patient safety.

Another major concern is the **difficulty patients experience in identifying the correct healthcare provider** for their medical condition. In Malaysia, incidents have been reported where patients, due to uncertainty or lack of clear guidance, consulted the wrong practitioner, resulting in delayed or ineffective treatment. This issue is further reflected in the current UTAR Hospital appointment website, where practitioner information is presented in broad categories, without specific details about each practitioner's specialization, making it challenging for patients to know whom to consult.

Appointment scheduling remains a persistent issue in the healthcare system, significantly impacting patient access and satisfaction. Recent findings show that 61% of patients skipped medical care in the past year due to difficulties in booking appointments [3]. Despite investments in digital scheduling tools, 63% of patients felt these systems did not meet their needs, and 70% were ultimately redirected to phone calls [3]. As a result, 41% of patients switched healthcare providers due to poor digital experiences [3]. Additionally, inefficiencies such as mismatches between patient demand and provider availability, high no-show rates, and staff overwork continue to hinder clinic operations and the overall quality of care [4].

1.1 Problem Statement and Motivation

In Malaysia's multicultural society, **language barriers significantly hinder healthcare access for the elderly**, particularly among non-Malay or non-English-speaking populations. Elderly patients with limited proficiency in the national language often struggle to accurately convey their symptoms, leading to **misdiagnoses, inappropriate treatments, and increased hospital readmissions**. A study found that elderly patients with limited English proficiency are at a higher risk of medical errors due to miscommunication. The reliance on **ad-hoc translation methods**, such as using family members or untrained staff, further exacerbates misunderstandings and compromises patient safety [1].

In addition to language barriers, patients in Malaysia face the **risk of misdiagnosis due to consulting the wrong healthcare provider**. A notable case in December 2023

involved a 51-year-old man who was misdiagnosed with gastric issues despite presenting with chest pain and breathing difficulties, which eventually led to his death. This highlights the dangers of patients being uncertain about which healthcare provider to consult, often due to unclear symptom descriptions or confusion regarding which professional to visit [2].

Issues related to **appointment scheduling** also plague the healthcare system. A significant 61% of patients skipped medical care in the past year due to difficulties in booking appointments. Despite the investment in digital tools, many patients reported that these tools fell short of expectations, with 63% of patients indicating that digital scheduling didn't meet their needs and 70% being redirected to phone calls after attempting online booking. This has contributed to 41% of patients switching healthcare providers due to poor digital experiences [3]. Moreover, **inefficient scheduling systems**, such as mismatches in demand and supply, frequent patient no-shows, and overworked staff, continue to affect the quality of patient care and clinic operations [4]. At UTAR Hospital, users also face challenges with the **appointment website**, where the practitioner descriptions are overly general and fail to specify their areas of expertise, making it difficult for patients to identify the right healthcare provider for their needs.

Motivation

This project is motivated by several pressing issues observed within Malaysia's healthcare sector, particularly among elderly individuals and multilingual communities. One of the primary barriers is the difficulty in communication between patients and healthcare providers due to language differences. In Malaysia's multicultural society, many elderly patients may be fluent in dialects such as Mandarin or other Chinese variants and may have limited understanding of English. This communication gap can lead to miscommunication, incorrect symptom reporting, and ultimately, delayed or inappropriate medical treatment. The use of untrained translators or family members further increases the risk of misdiagnosis, placing the patient's safety at risk.

Another core issue which arises the motivation of project is the lack of guidance in selecting the appropriate healthcare provider based on symptoms. Patients may be

unaware of which practitioner or department to consult, leading them to book appointments with the wrong specialist.

1.2 Research Objectives

The objective is to develop a system included multilingual chatbot, designed to assist a diverse population in predicting a wide range of diseases, including gynecological, musculoskeletal, internal, liver, skin, ENT, heart, lung, kidney, gastrointestinal disorders. Furthermore, the chatbot will offer tailored doctor recommendations based on Universiti Tunku Abdul Rahman (UTAR) Hospital's list of qualified Traditional Chinese Medicine (T&CM) practitioners. The system would also build for managing appointment and dashboard visualization (admin side).

1. To develop a bilingual (English and Chinese) chatbot for UTAR Hospital that utilizes machine learning to predict potential diseases based on symptom described by users.

The chatbot will engage with users in both English and Chinese, provides predictions of possible diseases based on the symptoms they describe. This approach aims to improve accessibility for a diverse user base, allowing individuals from different linguistic backgrounds to easily access healthcare information. Additionally, the chatbot will act as an initial advisory tool, offering preliminary disease prediction to users based on their input, making healthcare services more inclusive and user-friendly in a multicultural setting.

2. To implement doctor recommendations from UTAR Hospital's T&CM Centre based on the disease category predicted.

This feature will enable the chatbot to recommend the most appropriate healthcare specialists for users based on the predicted disease categories from their symptoms. By utilizing machine learning and RAG, the system will provide doctor recommendations, making it easier for patients to connect with the right expert. This streamlined approach reduces delays in finding the right specialist, enhancing the overall healthcare experience and improving efficiency for both patients and the hospital.

- 3. To design an online appointment scheduling system for UTAR Hospital T&CM Centre that allows patients to book and cancel appointments, with email notifications for confirmations, cancellations, and reminders.**

The system is designed to offer a platform for patients of Universiti Tunku Abdul Rahman (UTAR) Hospital's Traditional and Complementary Medicine (T&CM) Centre to easily schedule and manage their appointments. Patients will be able to directly book consultations with suitable healthcare providers. The appointment process will be streamlined to reduce manual coordination and administrative burden. Additionally, automated email notifications for booking confirmations, cancellations, and appointment reminders will help improve patient engagement, reduce no-shows, and enhance the overall patient experience.

- 4. To develop a dashboard that allows administrators to monitor user engagement with the chatbot, visualize the diseases predicted by the chatbot using a bar chart that displays their frequency and present a pie chart showing the number of appointments made by users for each doctor.**

The dashboard will provide administrators at UTAR Hospital with a clear and interactive overview of chatbot-related activities. It will display user engagement trends, commonly discussed diseases, and doctor appointment activities using visual elements such as bar charts, line charts, and pie charts. Administrators will be able to filter the data by specific time periods using timestamp filters, allowing them to track changes over days, weeks, or months. This real-time monitoring supports better decision-making, early identification of common patient concerns, and more efficient management of healthcare services.

1.3 Project Scope and Direction

The aim of this project is to develop a system included multilingual chatbot (Chinese English Language Support) for the UTAR Hospital, designed to provide symptom-based disease predictions, recommend the most appropriate healthcare practitioners, streamline the appointment scheduling process and dashboard visualization. The system and chatbot aims to enhance the accessibility of healthcare services and improve the overall user experience.

The key features of the project include:

- The chatbot will engage with users in both English and Chinese, making it accessible to a diverse, multicultural population.
- The chatbot will provide disease predictions based on the symptoms described by users, acting as a preliminary tool to guide them towards potential medical conditions.
- The chatbot will recommend the most suitable doctors based on the predicted disease category, helping users find the right specialist without confusion.
- The chatbot will streamline the process for users to book appointments with the recommended doctors and link provision.
- The chatbot will be able to automatically detect the language of the user's query and respond in the appropriate language (either English or Chinese).
- The system will include an administrative dashboard that tracks usage period of chatbot, disease trends, and export user query, response, disease category as JSON file as dataset for model further training which could provide insights to optimize healthcare services.
- Chatting feature among doctor and user

1.4 Contributions

A chatbot provides valuable support by offering **24/7 access** to symptom-based disease guidance, doctor recommendations, and links to reliable information about various health conditions. From the customer's perspective, it helps users understand their symptoms and find the right doctor **without the need to wait or feel overwhelmed**. The chatbot can also direct users to the appropriate appointment booking page, making the process quicker and easier. For the hospital, this **reduces the workload on staff, lowers operational costs and improves overall service efficiency**. It ensures consistent support, enhances patient experience and allows the hospital to operate more effectively with fewer resources.

The project brings together advanced technologies to improve accessibility to healthcare services, providing a multilingual, symptom-based chatbot that delivers personalized disease predictions and doctor recommendations. The integration of a **multilingual feature (English and Chinese)** enhances inclusivity by catering to a broader, multicultural audience, helping individuals who may otherwise face language barriers in accessing healthcare information. This is particularly important in Malaysia's diverse population.

The **symptom-based disease prediction** helps users gain preliminary advice on potential conditions, which acts as a first step in managing their health. By allowing users to input their symptoms, the chatbot offers a personalized disease prediction, reducing uncertainty and assisting users in identifying potential health concerns early on. The chatbot will respond in the user's input language, providing the probability of the predicted disease category and relevant link related to the predicted disease.

Doctor recommendations based on the predicted disease further streamline the process by guiding users to the most suitable healthcare professional for consultation. This feature minimizes delays in receiving specialized care, improving the overall healthcare experience. The chatbot will response user the healthcare professionals which work in T&CM centre of UTAR Hospital based on predicted disease category.

CHAPTER 1

Additionally, the development of an **appointment scheduling system** ensures that patients can conveniently book consultations with the recommended practitioners, reducing the need for manual coordination and enhancing overall operational efficiency. The integration of this system significantly enhances the patient experience by simplifying appointment booking.

Lastly, the **administrative dashboard** is a unique aspect of the project, providing valuable insights into user behaviour and disease trends. This data of user query, response of chatbot, predicted category can extract into JSON file from database for further training on model which could optimize the system, improve healthcare services.

In conclusion, the project's contribution lies in combining **multilingual support, disease prediction, doctor recommendations, appointment scheduling, and data analytics** in a single platform. The chatbot offers 24/7 availability, providing instant disease diagnosis and guiding patients to relevant information and doctor recommendations. It also redirects users to the appropriate appointment booking page, simplifying the process. This not only eases the workload of hospital staff but also lowers operational costs, benefiting both users and healthcare providers, especially those from diverse cultural backgrounds.

1.5 Report Organization

This report is structured into seven chapters: Chapter 1 covers the Introduction, Chapter 2 presents the Literature Review, Chapter 3 outlines the System Methodology, Chapter 4 details the System Design, Chapter 5 describes the System Implementation, Chapter 6 discusses the System Evaluation and Discussion, and Chapter 7 concludes with the Conclusion and Recommendations.

Chapter 1 provides an introduction to the project by presenting the problem statement, motivation, objectives, scope, key contributions, and the overall structure of the report. Chapter 2 presents a comprehensive literature review, including a discussion of the technologies used and an analysis of existing systems related to chatbots. Chapter 3 explains the methodology and approach used in the system, supported by diagrams such as system architecture, use case, and use case description. Chapter 4 centers on the

CHAPTER 1

system design, covering block diagrams, component specifications, and the interactions within the system. Chapter 5 describes the system implementation process, including hardware, software setting, configuration, and operational screenshots, as well as the challenges encountered during development. Chapter 6 evaluates the system through testing, performance metrics, and a discussion of how well the objectives were achieved. Chapter 7 summarizes the report and offers recommendations for future enhancements.

CHAPTER 2: Literature Reviews

2.1 Previous works on Chatbot Application

2.1.1 Mount Sinai hospital chatbot [5]

Figure 2.1.1.1, Figure 2.1.1.2, Figure 2.1.1.3 show the interface of Mount Sinai Hospital Chatbot with some testing case.

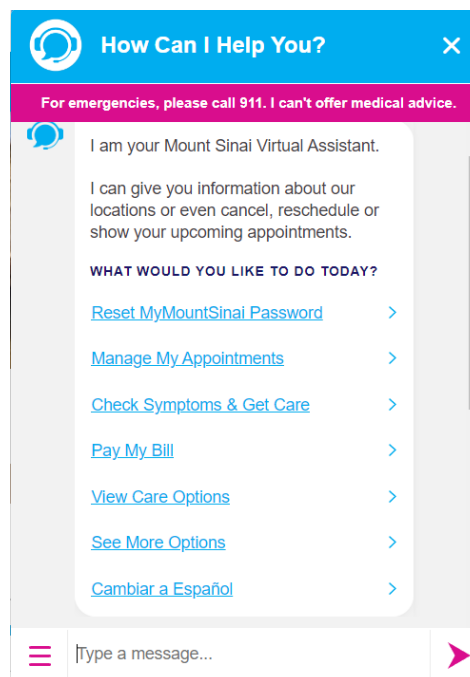


Figure 2.1.1.1 Mount Sinai Chatbot Interface

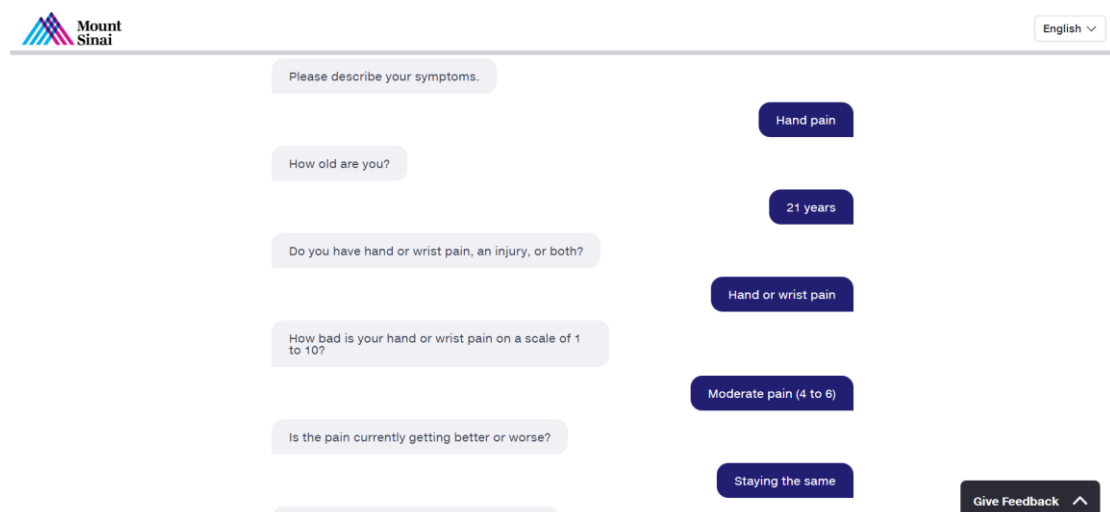


Figure 2.1.1.2 Mount Sinai Chatbot Interface

Mount Sinai

English ▾

How long have you had hand or wrist pain?

More than 7 days

Do you have any of these symptoms?

- ☐ Hand or finger numbness
- ☐ Hand or finger weakness (not just from the pain)
- ☐ Bruising of the hand or wrist
- ☐ Hand or wrist swelling
- ☐ Finger swelling
- ☐ None of the above

Send

Give Feedback ^

See below to schedule an appointment. You should see an orthopedic hand/wrist surgeon during normal office hours.

Figure 2.1.1.3 Mount Sinai Chatbot Symptom Checker (Demo)

Mount Sinai is a hospital placed in New York City. The hospital provided a chatbot for user to enquiry some questions. The user can make the appointment, check symptom (provided type of specialist can consult) and getting the information regarding Mount Sinai hospital in faster way as linking is provided.

Table 2.1.1 Strength and Weakness of Mount Sinai Chatbot

Strength	Weakness
<ul style="list-style-type: none"> - Enables users to quickly access information through provided links - Facilitates online appointment booking for user convenience - Helps users identify symptoms and suggests the appropriate type of specialist - Supports both Spanish and English languages for wider accessibility 	<ul style="list-style-type: none"> - Does not provide personalized doctor recommendations based on the user's specific disease or symptoms - Symptom checking is limited to predefined options; users cannot describe their symptoms freely

2.1.2 Zydus hospital chatbot (ZyE) [6]

Figure 2.1.2.1, Figure 2.1.2.2 show the interface of Zydus Hospital Chatbot with some testing case.

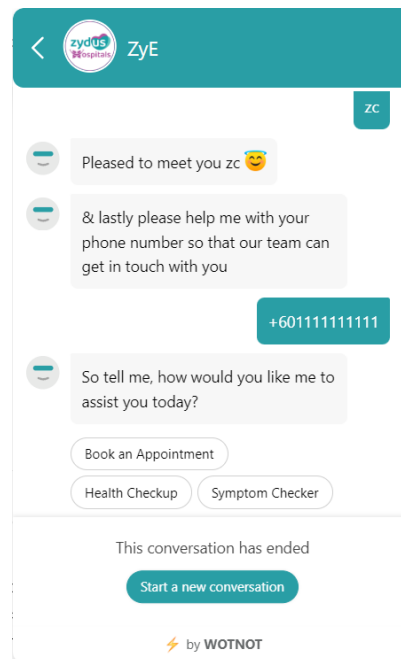


Figure 2.1.2.1 ZyE Interface

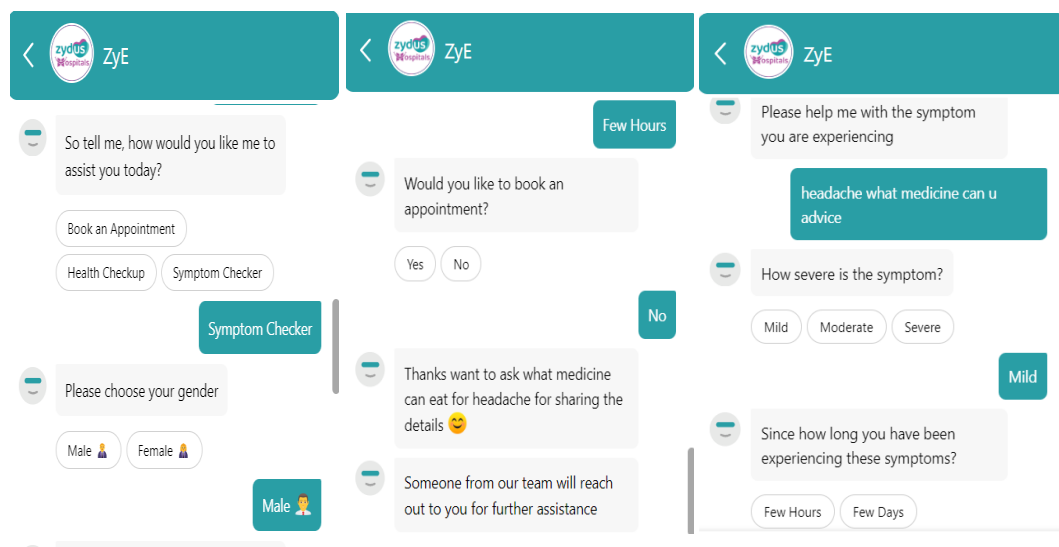


Figure 2.1.2.2 ZyE Symptom Checker (Demo)

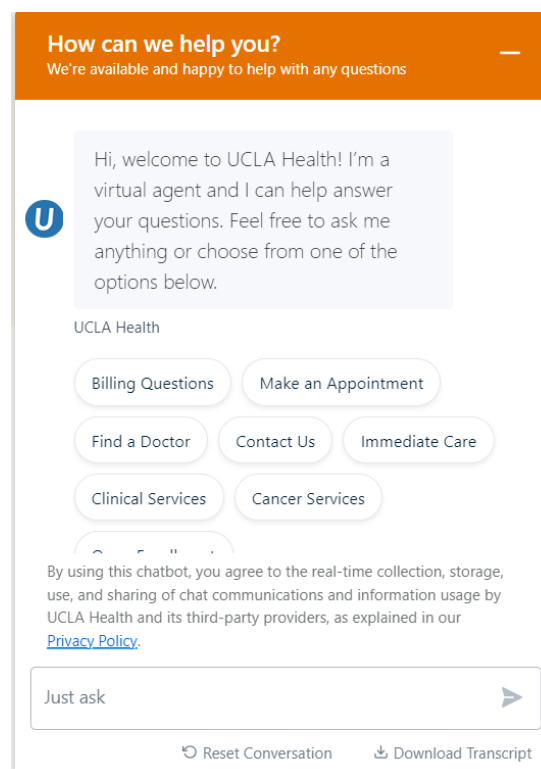
ZyE is a chatbot used by Zydus hospital which placed in India. ZyE allows users to make appointments, suggestions on health checkups based on gender and age. While the symptom checker is only based on what users describe and the chatbot will ask how long have suffered then ask user whether want to make appointment.

Table 2.1.2 Strength and Weakness of ZyE Chatbot

Strength	Weakness
<ul style="list-style-type: none"> - Enables users to book appointments online for greater convenience - Recommends suitable health checkup packages based on the user's age and gender, along with estimated pricing 	<ul style="list-style-type: none"> - Health checkup suggestions are limited to fixed answer choices; users cannot freely describe their needs - Does not provide general information about the hospital to the user

2.1.3 UCLA Health chatbot [7]

Figure 2.1.3.1 show the interface of UCLA Hospital Chatbot.

**Figure 2.1.3.1 UCLA Health Chatbot Interface**

UCLA health chatbot is used by UCLA health hospital. This chatbot allow user to consult on information about the hospital such as billing question, find doctor, immediate care, clinical service and cancer service.

Table 2.1.3 Strength and Weakness of UCLA Health Chatbot

Strength	Weakness
<ul style="list-style-type: none"> - Helps users quickly find doctors based on medical categories - Provides links for faster access to relevant information - Supports online appointment booking for user convenience 	<ul style="list-style-type: none"> - Lacks a symptom checker feature to help users identify potential health concerns

Table 2.1.4 Summary of Review Chatbots & Proposed Chatbot

Features	MountSinai hospital chatbot	Zydus hospital chatbot	UCLA health hospital chatbot	Proposed solution
Multilingual Support	Yes	Yes	No	Yes
Disease Diagnosis Based on Symptom Description	No	No	No	Yes
Doctor Recommendation	No	No	No	Yes
Direct Links for Providing Information	Yes	No	Yes	Yes
Appointment	Yes	Yes	Yes	Yes

2.2 Reviews on Technology

The paper by [9], emphasis on developing of healthcare chatbot using **natural language processing (NLP)** and neural network, specifically on **Multi-Layered Perceptron (MLP)**. The NLP technique that the author used for chatbot development is specifically based on **Natutal Language Toolkit (NLTK)**. The author used technique such as tokenization, removal of stop word, lemmatization to process user input and extract symptom of user. By comparing the symptoms with the dataset prepared, a final symptom list will be created. While regarding MLP, it is a three-layer neural network model consisting of input, hidden and output. Combination of input and output will be trained for the model. MLP can increase the accuracy compared to many machine learning algorithms. This paper may make improvement on adding more features such as doctor consultation, chat with voice and diagnosis via image.

The paper by [10], focuses on developing a health chatbot with Python library such as TensorFlow, TFLearn, NumPy, NLTK. JSON will be used as the format to store the data. TensorFlow is a software library for machine learning and deep neural networks research. While TFLearn is a deep learning library built on top of TensorFlow to accelerate neural network experimentation. **Lancaster Stemmer algorithms** enhance the processing speed as any word that came into the system will be reduced to its original word. For example, the word "Assigning, Assigned" will be reduce to Assign. After importing data from a JSON file, the process involves pickling for future use and tokenization to break down sentences into arrays of tags and responses. Stemming reduces vocabulary, and the "Bag of Words" concept is used for numerical representation because the neural network and machine learning algorithm require numeric input. Feed-forward neural network with two hidden layers, is trained to predict tags based on a bag of words. **The output layer will use softmax function** to ensure the highest probability neuron will return which will decide the accurate response. The threshold probability is set at 70% to filter out irrelevant responses.

The paper by [11] proposed implementation chatbot with deep learning technique, through past research from Anupam Mondal and Monalisa Dey, involves processing data using the random forest algorithm with issue of either no subsampling and too much sampling, the author addressing its consistency issues through the application of the Bag of Words Representation. To overcome challenges such as domain-specific answers and a rigid input format found from Question Answering system, the authors advocate for the use of a sequence2sequence model for text summarization. This model employs an encoder to convert input words into hidden vectors using deep neural network layers, and a decoder, similar to the encoder, to generate the next hidden vector based on the hidden states and the current word.

Sequence2sequence technique relies on RNNs. Given that the two RNNs are distinct, their lengths also differ. They can be instantiated using GRU and LSTM architectures. The initial state of the search decoder aligns with the above notation. This involves taking the output of the unrolled decoder network at each time step when words are generated as output. These outputs are then fed into a function, generating conditional probabilities for the words in the document used as input. The training process for chatbots is similar to training recurrent neural networks. The objective is to maximize the probability of generating the correct target sequence given the source sequence. Both RNNs are trained simultaneously. **SGD** will be used as optimistic technique to send back the error, their weight will optimize. **Beam search decoding technique** used in this paper, breadth-first search and a greedy approach to build a state-space tree. It identifies the most likely sequences by expanding reasonable next steps and retaining the top n sequences based on probabilities. At each level, it generates successors, sorts them by heuristic values, and stores only a predefined number of the best states for further expansion. If no response is found, the process repeats with an expanded beam.

The paper by [12] proposed development of healthcare system chatbot using machine learning techniques to help users regarding minor health information. Figure 2.2.1 shows the system architecture of the proposed chatbot.

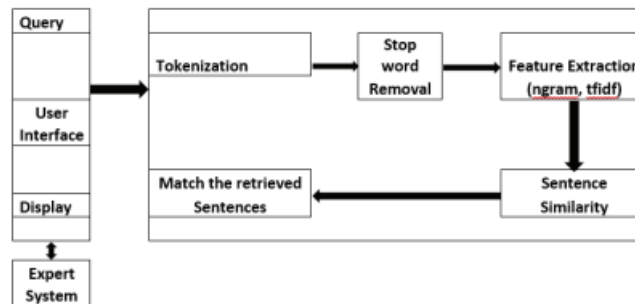


Figure 2.2.1 System architecture of proposed chatbot

The system implemented preprocessing step such a tokenization, removal of stop word for user query that receive through the user interface. Also, technique such as n-gram, TF-IDF, cosine likeness for feature extraction. The answer to the question that used to answer user query will be stored in the knowledge database. While domain expert is present in this system to answer the question of users if the answer of user query is not included in the knowledge database. The project aims to save users time in consulting doctor or experts for healthcare solution. **N-gram and TF-IDF** are used to extract and assign weights for user queries. A web interface is developed for users to input their queries and receive responses.

The text processing pipeline begins with tokenization, where text is split into individual words and punctuation is removed. Next, stop words (e.g., "an," "a," "the") are eliminated to highlight significant keywords, thus reducing computational complexity. Feature extraction employs N-gram TF-IDF, which ranks document features based on term frequency and inverse document frequency, emphasizing rare terms. N-grams help in text compression and keyword extraction. Finally, **cosine similarity** is used to measure the similarity between sentences, with values ranging from 0 to 1, facilitating the comparison of document relevance. The answer of the user query obtained after undergoes above process are retrieved and displayed to the user interface.

The paper by [13] aimed to develop a chatbot called "HSchatbot" to predict the intent classifications of high school students' inquiries. Utilizing Multinomial Naive-Bayes and Random Forest classifiers, the researchers improved performance through feature extractions, with **Random Forest, Multinomial Naive-Bayes**. Both classifiers achieved high accuracy scores exceeding 90% in all metrics. Recognizing the importance of providing instant support to high school students for their future career decisions, the study integrated advanced chatbot technology to categorize users' requests based on their intentions using Natural Language Processing (NLP).

The research builds on various studies showing chatbots aiding in software development, customer service, and education, particularly for learning languages and sciences, motivating student engagement, and answering frequently asked questions. The dataset comprised 505 inquiries collected from academic advisors and university websites. Essential **NLP preprocessing** steps included tokenization, lowercasing, stop words removal, and lemmatization, while feature extraction techniques like **CountVectorizer and TfidfTransformer** enhanced classifier performance. The CountVectorizer tokenized text to select the most frequent words, converting text to structured data, whereas TfidfTransformer weighed tokens to emphasize infrequent, valuable words. Future work will analyze the results to determine factors affecting the Multinomial Naive-Bayes classifier's performance and evaluate the model with a larger corpus of students' inquiries. The study underscores the potential of chatbots in supporting high school students through intent classification, leveraging key NLP techniques and machine learning algorithms.

The paper [14] discusses the development and implementation of a chatbot system using NLP to assist the new student admissions committee for university. The chatbot uses **TF-IDF and Dice Similarity algorithms** to match and respond to inquiries from prospective students. The evaluation of the chatbot involved posing 42 questions to measure its effectiveness. The results demonstrated a recall rate of 100% and a precision of 76.92%, indicating the chatbot's capability to accurately address questions. The study underscores the importance of integrating advanced technology in educational institutions to enhance service quality and improve customer satisfaction. Preprocessing of the text data involves several steps to make it suitable for analysis by

machine learning algorithms. These preprocessing stages include case folding, tokenization, stop word removal and stemming.

Paper by [15], proposed system is an interactive chatbot designed to handle university-related FAQs. It starts with the user greeting or asking a general question. The chatbot processes the inquiry to see if it matches any **AIML scripts**, which are predefined templates for common questions. The process involves three main steps: the user posts a query, the chatbot processes it to match predefined formats set by developers and then performs pattern matching between the query and its knowledge base to generate a relevant response.

This chatbot is specifically designed for the educational sector, allowing students or parents to ask about college admissions, university information, and other academic topics. It provides information on university rankings, available services, campus environment, and updates on campus activities. Implemented using AIML for Manipal University, the chatbot helps students access various types of academic information, improving their ability to get updates and details about the university. This method ensures that the chatbot can effectively address common queries by using pattern matching techniques and predefined templates.

For future work, the author suggested that we can develop a chatbot that combines **AIML and Latent Semantic Analysis (LSA)**. This will enable users to interact with the chatbot more naturally. Improve the chatbot by adding and updating patterns and templates for general queries using AIML, while LSA ensures the correct responses are given more often. Table 2.2.1 show the AIML tag explanation.

Table 2.2.1 Tag explanation for AIML techniques

AIML Tag	Explanation
< aiml > tag:	Root tag for AIML files, includes encoding and version attributes.
< category > tag:	Encloses knowledge units and contains pattern and template tags.

< pattern > tag	Defines user query patterns within a category.
< template > tag:	Contains the response to the user's query.
< srail > tag:	Targets multiple patterns for a single template to handle similar user inputs efficiently.
< random > and < li > tags:	Generate random responses for varied interactions.
< set > and < get > tags:	Manage variables to store and retrieve user data.
< that > tag:	Refers to the last response given by the chatbot for maintaining conversation context.
< topic > tag:	Groups related categories to maintain coherent conversations about specific subjects.
< think > tag:	Processes data internally without displaying it to the user.
< condition > tag:	Displays responses based on specific variable values.
< bot > tag:	Retrieves and displays chatbot properties during a conversation.

This paper [16] focuses on developing an advanced deep learning chatbot designed to provide comprehensive first aid guidance. The objective of the paper is to develop a chatbot that effectively understands and generates human-like responses by utilizing NLTK for enhanced language processing and Keras for constructing a neural network model to improve conversation comprehension and response generation.

Leveraging technologies such as the **Natural Language Toolkit** and the **Keras deep learning framework**, the chatbot utilizes the **first aid dataset from Kaggle**, which **includes intents, patterns, and responses related to various medical conditions**. The dataset is meticulously preprocessed through tokenization, removal stopword and sequence padding to standardize input data, with an Embedding layer used to transform

categorical input into dense vectors. Dense layers with ReLU activation facilitate feature extraction, and a softmax activation layer handles multi-class classification. The models, including LSTM networks, GRU, Bidirectional LSTM, and Dense Neural Networks, are compiled using dropout for regularization which could reduce overfitting, sparse categorical crossentropy loss function and the Adam optimizer for efficient optimization on parameter, all model trained over 150 epochs with a batch size of 16 except for DNN model which trained over batch size of 5 with 150 epochs. Accuracy, F1-score, recall and precision and user feedback is collected to evaluate the performance to assess real-world effectiveness. The implementation involves Python and TensorFlow for development, with a user-friendly web interface created using HTML, CSS, and JavaScript.

The chatbot, built using Dense Neural Networks (DNN), has effectively provided first aid information. Future enhancements include integrating advanced NLP techniques like BERT or GPT for better language understanding, implementing a context-aware system for more personalized interactions, and expanding the dataset. Additional improvements involve adding real-time assistance, collaborating with UX/UI designers, offering multilingual support, and ensuring robust security.

This paper [17] emphasis on **FAISS-based Retrieval-Augmented Generation (RAG)** methodology for medical assistance enhances **large language model (LLM)** response generation by integrating external knowledge sources. Specifically, the system leverages *The Gale Encyclopedia of Medicine* as its foundational knowledge base, which is extracted from PDF documents and segmented into 500-token chunks with a 25-token overlap to preserve context across sections. These text chunks are embedded using the **all-MiniLM-L6-v2 sentence-transformer model** from Hugging Face, enabling the creation of high-quality vector representations that are stored in a FAISS vector datastore for efficient similarity search operations.

When a user submits a query, the system conducts a similarity search on the FAISS datastore to retrieve the three most relevant text chunks. These retrieved chunks, combined with the user's query and existing chat history, are then provided as input to the **Mistral NeMo Instruct model** for response generation. This retrieval-augmented

approach improves the accuracy, relevance, and efficiency of medical information retrieval by grounding LLM outputs in verified external knowledge sources, thereby enhancing the overall quality of responses in a medical assistance context.

This paper [18] focus on a cancer-specific **Retrieval-Augmented Generation (RAG)** system, gathers data from medical and academic sources. Using an API key, PubMed was searched for cancer-related papers, resulting in the collection of 100 full-text PDFs covering diagnosis, treatments, advancements, and clinical trials. Additionally, seven books on cancer treatment, prevention, and patient care were sourced via Google to provide a more comprehensive knowledge base. Figure 2.2.2 show the architecture of the RAG powered CancerBot.

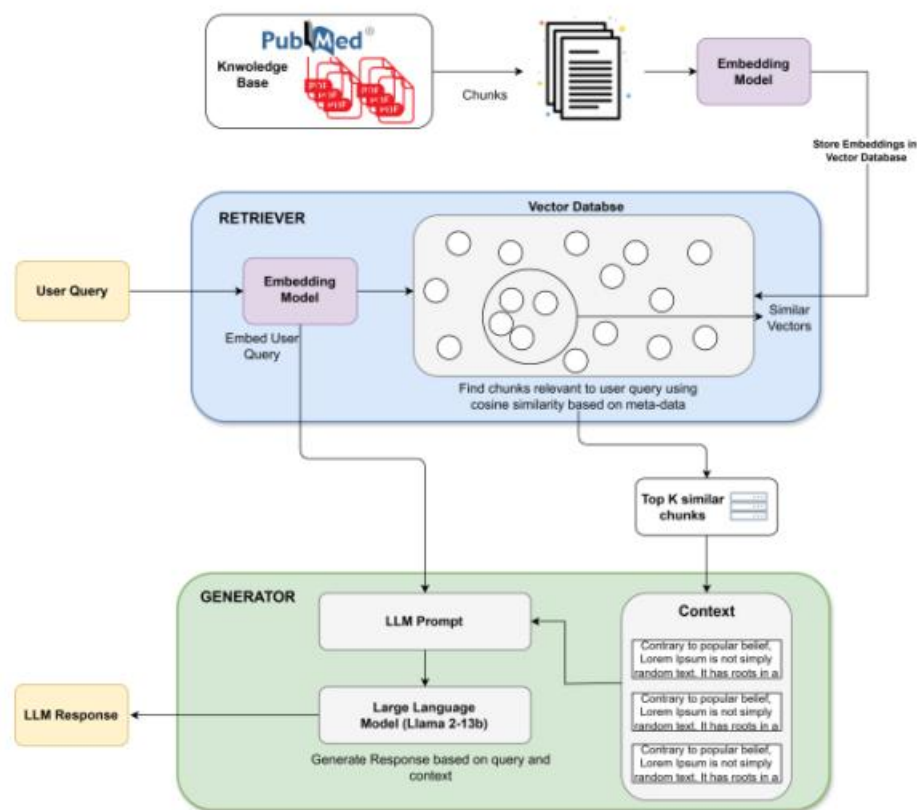


Figure 2.2.2 Architecture of the Proposed RAG-powered CancerBot

To prepare this data, **LangChain's ReadTheDocsLoader** was used for document processing. The texts were segmented into 800 characters chunks using **RecursiveCharacterTextSplitter** to maintain coherence. The **sentence transformer**

(all-mpnet-base-v2) model was applied to generate embeddings, enabling the system to retrieve contextually relevant information.

Pinecone Vector Database is utilized to efficiently store and retrieve text embeddings, enabling fast similarity searches across large datasets. To configure the database, an index with a dimensionality of 384 was created, matching the embeddings from the sentence-transformers (all-mpnet-base-v2) model. **Cosine similarity** was used as the metric to compare text embeddings based on semantic content. The system first checked for an existing index and created one if necessary, ensuring it was fully initialized before use.

The **Llama-2-13B model**, part of Meta's Llama-2 family, is a pre-trained generative model using a transformer architecture. It employs self-attention mechanisms to understand contextual relationships, with an encoder-decoder structure where the encoder processes input tokens and the decoder generates responses. The multi-head attention mechanism enhances its ability to recognize complex patterns. To optimize deployment on limited-memory hardware, quantization techniques were used, reducing model size while maintaining performance.

For the cancer chatbot, RAG enriches the knowledge base by extracting relevant information from medical literature stored in a vector database. This approach enables real-time updates without the need for costly model retraining, allowing the chatbot to expand its cancer-related knowledge as new data becomes available. At last, **Gradio** is used to build the web interface with local hosted.

Table 2.2.2 Summary on Literature Reviews

Paper	Chatbot Usage	Technique used	Future Improvement
[9]	Healthcare Chatbot	Machine learning <ul style="list-style-type: none"> • NLP Technique (Tokenization, Removal of stop word, Lemmatization) • Algorithms/Library/Others MLP, NLTK 	Adding features such as voice chat, doctor consultation, diagnosis via image
[10]	Health Chatbot	Machine learning <ul style="list-style-type: none"> • NLP Technique Lancaster Stemmer algorithm • Algorithms/Library/Others Tensorflow, TFLearn, Numpy, NLTK, JSON, Pickle file 	-
[11]	Deep Learning Chatbot	Machine learning <ul style="list-style-type: none"> • NLP Technique Bag of Word • Algorithms/Library/Others Random Forest Algorithm, Sequence2sequence Model, RNN, SGD, Beam Search Decoding 	Using voice-based query recognition
[12]	Healthcare Chatbot	Machine learning <ul style="list-style-type: none"> • NLP Technique (Tokenization, Removal of 	-

		<p>stop word, Removal of punctuation)</p> <ul style="list-style-type: none"> • Feature Extraction Technique N-gram TF-IDF • Algorithms/Library/Others Cosine similarity 	
[13]	High School Chatbot	<p>Machine learning</p> <ul style="list-style-type: none"> • NLP Technique (Tokenization, Removal of stop word, Lowercasing, Lemmatization) • Feature Extraction Technique CountVectorizer, TfidfTransformer • Algorithms Multinomial Naïve-Baiyes, Random Forest Classifier 	Using Deep Learning Algorithm
[14]	University Chatbot for New Student Admission	<p>Machine learning</p> <ul style="list-style-type: none"> • NLP Technique Case folding, Tokenization, Removal of stop word, Stemming • Feature Extraction Technique TF-IDF 	-

		<ul style="list-style-type: none"> • Algorithms Dice Similarity algorithm 	
[15]	University-related FAQs Chatbot	AIML Technique	Combine AIML and Latent Semantic Analysis together for chatbot development to enhance naturally
[16]	First-aid chatbot	Machine & Deep learning <ul style="list-style-type: none"> • NLP Technique Tokenization, Removal of Stopword, Sequence Padding • Algorithms/Library/Others NLTK, Keras, LSTM, Bidirectional LSTM, GRU, DNN • Frontend Technique Flask, HTML, CSS, Javascript 	Integrate BERT or GPT as advanced NLP technique for chatbot development Expand dataset, Multilingual support
[17]	Medical Chatbot	LLM and RAG technique <ul style="list-style-type: none"> • RAG • LLM Sentence transformer	-
[18]	Cancer Chatbot	LLM and RAG technique <ul style="list-style-type: none"> • Pinecone Vector Database • LLM • RAG 	-

		<ul style="list-style-type: none"> • Sentence transformer • Cosine similarity • Frontend Technique Gradio 	
--	--	---	--

Table 2.2.3 Vector Representation Method

Vector Transformer	Description
TF-IDF [19]	<p>TF-IDF is a way used to measure the significance of a word in a document compared to a collection of documents (corpus). It involves two key components:</p> <ul style="list-style-type: none"> • Term Frequency (TF) quantifies how often a word appears in a document. The formula is: $TF(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$ • Inverse Document Frequency (IDF) can reduce the weight of common words from multiple documents and gives more importance to less frequent terms. The formula is: $IDF(t, D) = \log \frac{\text{Total number of documents in corpus } D}{\text{Number of documents containing term } t}$ <p>Combining TF and IDF allows TF-IDF to emphasize words that are more in document but rare in the corpus, making it an effective tool for identifying important terms while disregarding common, less informative words.</p>
Sentence Transformer [20]	<p>Sentence Transformers are deep learning models in Natural Language Processing (NLP) that convert entire sentences into high-dimensional vectors, also known as embeddings. Unlike traditional word embeddings like Word2Vec or GloVe, which</p>

	only represent individual words without full context, sentence transformers capture the complete semantic meaning of a sentence. This makes them highly effective for tasks such as semantic search, sentence similarity, and text clustering. Their ability to preserve context and meaning has greatly improved the accuracy and effectiveness of various NLP applications, making them a valuable tool for deeper and more meaningful text analysis.
--	---

Llama index

LlamaIndex is a framework that can enhance Large Language Models (LLM) by enabling them to access private data. While LLMs are pre-trained on vast public datasets, their functionality is limited without personalized data. LlamaIndex solves this by allowing data ingestion from sources like APIs, databases, and PDFs through flexible connectors. It then indexes this data into optimized intermediate representations for LLMs. With query engines, LLM-powered agents and chat interfaces, users can perform natural language searches and conversations with their data without retraining the model, making large-scale private data interpretation seamless [21].

RAG Workflow

Data Collection – Gather relevant data. **Data Chunking** – Process of breaking large documents into smaller, topic-focused sections to improve retrieval efficiency. **Document Embeddings** – Convert text chunks into numerical vector (embedding) representations to capture semantic meaning. **Handling User Queries** – Transform user queries into embeddings and retrieve the most relevant document chunks using similarity measures. **Generating Responses with LLM** – Feed retrieved information and the query into a language model to generate an accurate and contextually relevant response [22].

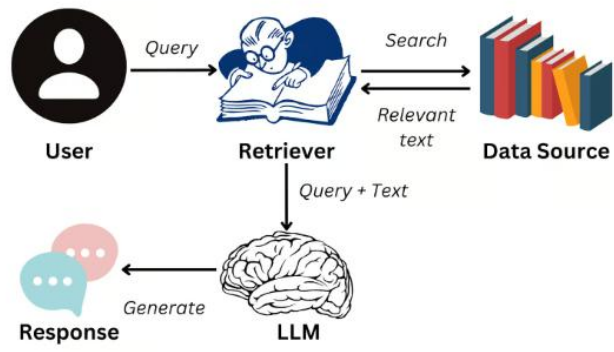


Figure 2.2.3 RAG workflow

CHAPTER 3: System Methodology

CRISP-DM is an open standard process model that provides organizations with a structured framework to navigate the complexities of data science projects, resulting in deeper insights, better decision-making, and ultimately, business success [23]. Figure 3.1 show the Crisp DM methodology which included six major phases which are Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment.

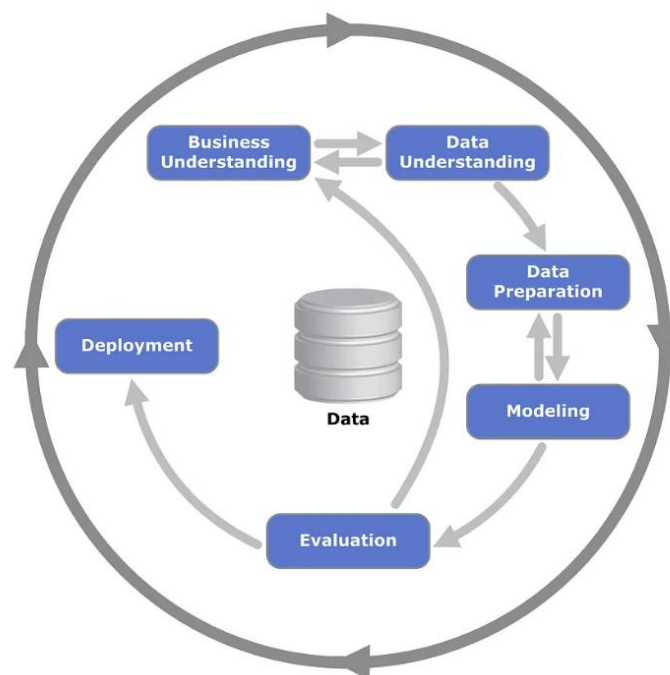


Figure 3.1 CRISP-DM Methodology

Business understanding

This phase involves analysing the current business problems and identifying the project's goals. As stated in Chapter 1, section 1.1 the problems have been identified, and the objectives have been clearly defined in section 1.2 to guide the development of a solution. The focus now is on aligning the project direction with these objectives to ensure the proposed system effectively addresses the key challenges and delivers meaningful improvements to healthcare access and service delivery. The data mining goal for the models used on chatbot development is to achieve an accuracy of minimum 80%.

Data understanding

The focus of this phase is on collecting, exploring and understanding about the data that will be used for the following phase such as data preparation and modelling. The primary tasks emphasis on process of gathering the data sources for the project. Understanding the structure of the dataset is also a crucial part for data understanding, which involves identifying data patterns and insight via visualization that could provide value information to undergo following phase. The data will collect from UTAR hospital flyer, UTAR hospital and clinical websites. The data is used to form sentences and used as dataset for model training and used in RAG for transformer model usage. In this section, some insight regarding dataset will show.

Data Preprocessing

The focus of this phase is to transform the initial dataset into the final dataset that will be used in the modelling phase. This includes various text preprocessing tasks such as removing noise and stop words, expanding contractions, converting text to lowercase, and applying lemmatization (an NLP technique). Besides, sentence embedding transformer will also use for data preprocessing. For the RAG dataset, the question and answer pairs will have their labels removed, retaining only the words themselves without any associated tags.

Modeling

The modeling phase involves several iterations to identify the most effective approach for disease prediction using text-based data. The final dataset will be separate into train and test set. Some machine learning classification algorithms such as Random Forest, Logistic Regression, and Naïve Bayes will be applied to training on data. These models are initially trained using default parameters, and GridSearchCV is used to tune defined hyperparameters to improve performance of the model. Cross validation will do combining with GridSearchCV. Oversampling method would implement to handle class imbalance issue and prevent model overfitting.

While transfer learning was attempted using the Qlora method for transformer-based LLaMA model to improve response generation related to dataset provided. However, the fine-tuning process was not successful due to the limited size of the dataset.

Evaluation

This phase is focus on using the success criteria that define in Business Understanding phase to evaluate is the model build achieves the business objectives. The measurement metric on classification model such as accuracy, confusion matrix could be a success criterion on evaluating the model achievement. While the model can achieve approximately 80% accuracy result mean that the classification model has the ability on making disease category prediction task. The model has ability to achieve the objective that declare.

Deployment

The creation of a model does not mark the end of the project. It is important to present the results in a way that customers can understand and use effectively. Since most users are not data analysts, the deployment phase focuses on making the model accessible and user-friendly. Depending on the project's needs, deployment can range from simple report generation to the development of a full web application. In this project, a website will be developed to serve as the user interface, with separate access for administrators and general users. The website will include chatbot page and some key features of hospital such as an appointment management page (for booking and cancelling appointments), admin dashboard page to monitor and manage system activities, chatting with doctor page and related disease information page. This approach ensures that both users and administrators can easily interact with the system and benefit from the model's capabilities.

3.1 System Design Diagram

3.1.1 System Architecture Diagram

Figure 3.1.1.1 show the system architecture diagram. There are 3 main roles for this system. Admin has the highest authority (can access all page), Doctor (can access all page except dashboard page) and User (can access the page that listed below). PHP is the language used for the interface and MYSQL database is used to store and retrieve data.

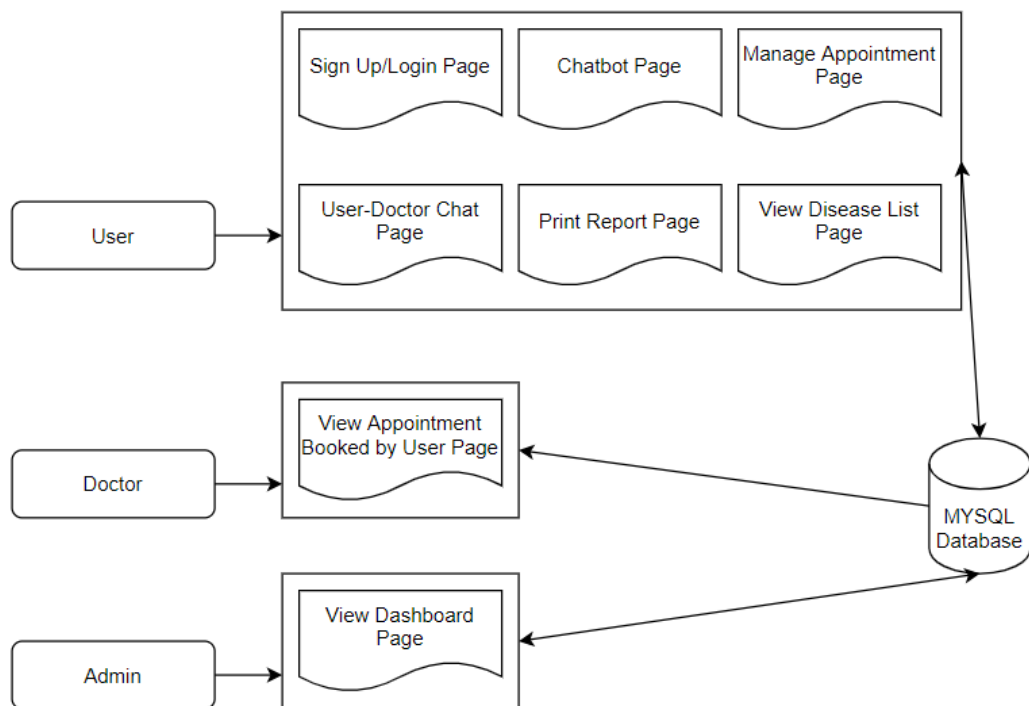


Figure 3.1.1.1 System Architecture Diagram

3.1.2 Use Case Diagram

Figure 3.1.2.1 show the use case diagram, the feature included in the system

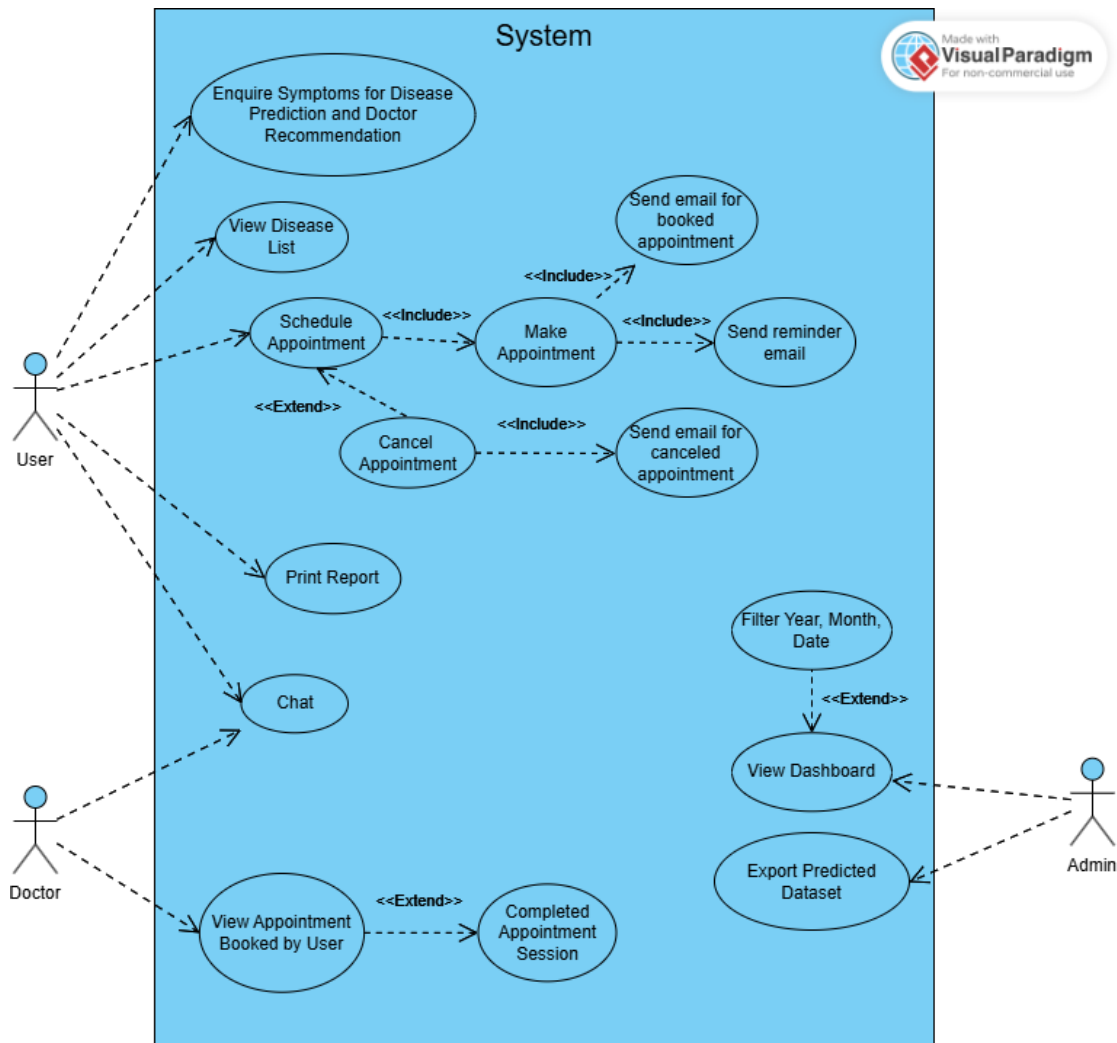


Figure 3.1.2.1 Use Case Diagram

The system is designed to support three key user roles: users, doctors and administrators (admins), each with distinct functionalities tailored to their needs.

Users are the primary individuals who interact with the system through a web interface. One of the main features provided to users is the ability to submit queries by describing their symptoms. The system then uses a machine learning model to analyse these symptoms and predict the most likely disease category from ten predefined health conditions. Based on the predicted disease, the chatbot recommends suitable doctors from the UTAR Hospital TCM Centre who can assist with the user's condition.

In addition to querying, users can manage their medical consultation appointments through the system. Users could schedule new appointments, view existing bookings, and cancel appointments if needed. Each appointment, along with relevant details such as date, time, and assigned practitioner, users who make appointment or cancel appointment would get email notification, reminder notification will send to user if the booked appointment date is current date. For doctor side, doctor is allowed to view who have made appointment with him/her, if session done. The record can be removed from the database.

Furthermore, the users could view the disease list (more information about the disease with link provided). Report printing is supported by the system as well. Users could print the report regarding the question asked to the chatbot. This could benefit users from the perspective if they want to do manual booking at UTAR hospital with the chatbot response. The admission can be based on the report to identify what problem are the users faced and provide more suitable doctor for them. The system included with the chat feature allows the communication between the users and doctors. Therefore, any update can be made directly among the users and his/her doctor.

Admins are typically staff or representatives from the organization responsible for overseeing and managing the backend of the system. They are granted access to an administrative dashboard that presents various analytics and operational insights. For example, admins can view reports that highlight which diseases are most frequently mentioned in user queries based on the chatbot's predictions. They can also monitor how often the chatbot is used and identify which doctors are in highest demand for appointments, allowing them to make data-driven decisions about resource allocation and service planning.

Furthermore, the system supports the export of key data including user queries, chatbot responses, and predicted disease categories—in a structured JSON format. This exported data can be used further enhance and retrain the underlying machine learning model, ensuring continuous improvement of the chatbot's accuracy and relevance over time.

3.1.3 Use Case Description

Table 3.1.3.1 Use case description 1: Enquire symptom for disease prediction and doctor recommendation

Use Case Name: Enquire symptom for disease prediction and doctor recommendation	ID: 1	Important Level: High
Primary Actor: User	Use Case Type: Detail, Essential	
Stakeholders and Interests: User – Make disease diagnosis using chatbot		
Brief Description: This use case describes on how user makes disease diagnosis and doctor recommendation from the chatbot		
Trigger: User wants to make disease diagnosis Type: External		
Relationships: Association: User Include: - Extend: - Generalization: -		
Normal Flow of Events: <div>1. User input some symptoms about what they are facing.</div> <div>2. System passes user input to machine learning model to make prediction</div> <div>3. System response back user with predicted disease category along with the doctor recommendation.</div>		
Sub Flows: -		
Alternate/Exceptional Flows: 1. The system display “I did not understand” if the chatbot cannot predicted any patterns that related to user enquires.		

Table 3.1.3.2 Use case description 2: View Disease List

Use Case Name: View Disease List	ID: 2	Important Level: High
Primary Actor: User	Use Case Type: Detail, Essential	
Stakeholders and Interests: User wants to know more information about disease		
Brief Description: This use case describes on how to view more information		
Trigger: User wants to know more information about disease performance Type: External		
Relationships: Association: User Include: - Extend: - Generalization: -		
Normal Flow of Events: <div><div>-</div>User selects disease category</div> <div><div>-</div>System display link regarding the disease</div> <div><div>-</div>User clicks on link to redirect to page related to the link to gather more information</div>		
Sub Flows: -		
Alternate/Exceptional Flows: -		

Table 3.1.3.3 Use case description 3: Schedule Appointment

Use Case Name: Schedule Appointment	ID: 3	Important Level: High
Primary Actor: User	Use Case Type: Detail, Essential	
Stakeholders and Interests: User – Schedule appointment using chatbot		
Brief Description: This use case describes on how user schedule appointment using chatbot		
Trigger: User wants to schedule appointment Type: External		
Relationships: Association: User Include: Make appointment Extend: Cancel appointment Generalization: -		
Normal Flow of Events: <div><div>- User login using email and password</div><div><div>1. User makes appointment</div><div><div>a. User enters name</div><div>b. System used login email</div><div>c. User selects doctor to consult</div><div>d. User selects date, time available</div><div>e. User enters reason for appointment</div><div>f. User clicks book appointment button</div><div>g. User’s booking information store in MYSQL database</div><div>h. System sends booking information to user’s email</div></div></div><div><div>2. User cancels appointment</div><div><div>a. System display appointment booked after user login</div><div>b. User click cancel button to cancel appointment</div><div>c. System sends cancel information to user’s email</div><div>d. Appointment deletes from MYSQL database</div></div></div></div>		
Sub Flows: -		

Alternate/Exceptional Flows:

1. The system will display 'Appointment booked by others' to user if the dates, time for the respective doctor already booked by other person.
2. The system will display 'Invalid email/password' if the authentication account for login does not exist.

Table 3.1.3.4 Use case description 4: Print Report

Use Case Name: Print Report	ID: 4	Important Level: High
Primary Actor: User	Use Case Type: Detail, Essential	
Stakeholders and Interests: User wants print report that been chat with the chatbot		
Brief Description: This use case describes on how to print the report that been chat with the chatbot		
Trigger: User wants print report that been chat with the chatbot Type: External		
Relationships: Association: User Include: - Extend: - Generalization: -		
Normal Flow of Events: <div><div>- User login using email and password</div><div>1. Print whole report by disease category</div><div>- User selects the disease category and click the print button to print the report</div><div>2. Print report for search input</div><div>- User inputs query he/she chats before with the chatbot and click the print button to print the report</div></div>		
Sub Flows: -		
Alternate/Exceptional Flows: The system will display ‘No record found’ if user inputs query is no found for report printing.		

Table 3.1.3.5 Use case description 5: Chat

Use Case Name: Chat	ID: 5	Important Level: High
Primary Actor: User, Doctor	Use Case Type: Detail, Essential	
Stakeholders and Interests: User wants to chat with doctor		
Brief Description: This use case describes on how user chat with doctor		
Trigger: User wants to chat with doctor Type: External		
Relationships: Association: User, Doctor Include: - Extend: - Generalization: -		
Normal Flow of Events: User Side <ul style="list-style-type: none">- User login using email and password- User selects the doctor want to chat with- User queries input to chat with doctor Doctor Side <ul style="list-style-type: none">- Doctor logins using email and password- Doctor receives notification and user query		
Sub Flows: -		
Alternate/Exceptional Flows: User and Doctor side can click on end session button to clear all message they typed.		

Table 3.1.3.6 Use case description 6: View Appointment Booked by User

Use Case Name: View Appointment Booked by User	ID: 6	Important Level: High
Primary Actor: Doctor	Use Case Type: Detail, Essential	
Stakeholders and Interests: Doctor wants to view his/her time slot that booked by user		
Brief Description: This use case describes on how doctor view his/her time slot that booked by user		
Trigger: Doctor wants to view his/her time slot that booked by user Type: External		
Relationships: Association: Doctor Include: - Extend: Completed Appointment Session Generalization: -		
Normal Flow of Events: <div><div>-</div>Doctor logins using email and password</div> <div><div>-</div>Doctor view booked user details with time</div> <div><div>-</div>After done session with respective user, doctor can click done button</div>		
Sub Flows: -		
Alternate/Exceptional Flows: -		

Table 3.1.3.7 Use case description 7: View Dashboard

Use Case Name: View Dashboard	ID: 7	Important Level: High
Primary Actor: Admin	Use Case Type: Detail, Essential	
Stakeholders and Interests: Admin – want to investigate on the user query, predicted disease category, peak time on chatbot usage, popular disease and doctor booked by user via the usage of chatbot and appointment system		
Brief Description: This use case describes on visualization of data the ingest from chatbot interface and appointment system		
Trigger: Admin want to analyse the data via the graph visualization Type: External		
Relationships: Association: Admin Include: - Extend: Filter Year, Month, Date Generalization: -		
Normal Flow of Events: - Admin login using email and password 1. System displays few tabs for admin 2. Admin select tabs a. When the admin selects Tab A , a bar chart is displayed showing the most common diseases predicted based on user queries. b. When the admin selects Tab B , a line chart is displayed illustrating the frequency of user interactions with the chatbot for inquiries. c. When the admin selects Tab C , a pie chart is displayed showing the distribution of doctors booked by users through the chatbot. d. When the admin selects Tab D , a table is displayed listing the symptoms described by users that were predicted by the chatbot, which may correspond to multiple disease categories.		
Sub Flows: Admin can select filter such as select year, month, date to look into more specific graph display		

Alternate/Exceptional Flows: -

Table 3.1.3.8 Use case description 8: Export Predicted Dataset

Use Case Name: Export Predicted Dataset	ID: 8	Important Level: High
Primary Actor: Admin	Use Case Type: Detail, Essential	
Stakeholders and Interests: Admin – extract dataset from database that make based on user query and model prediction		
Brief Description: This use case describes on how to export predicted dataset		
Trigger: Admin want to extract dataset for further model training to increase model performance Type: External		
Relationships: Association: Admin Include: - Extend: - Generalization: -		
Normal Flow of Events: <ul style="list-style-type: none">- Admin login using email and password- Admin click the extract button- System extracts the information from database and format into question, answer and category into JSON file.- JSON file download into Admin computer.		
Sub Flows: -		
Alternate/Exceptional Flows: -		

3.1.4 ERD Diagram

Figure 3.1.4.1 show the ERD diagram, the table used for the system

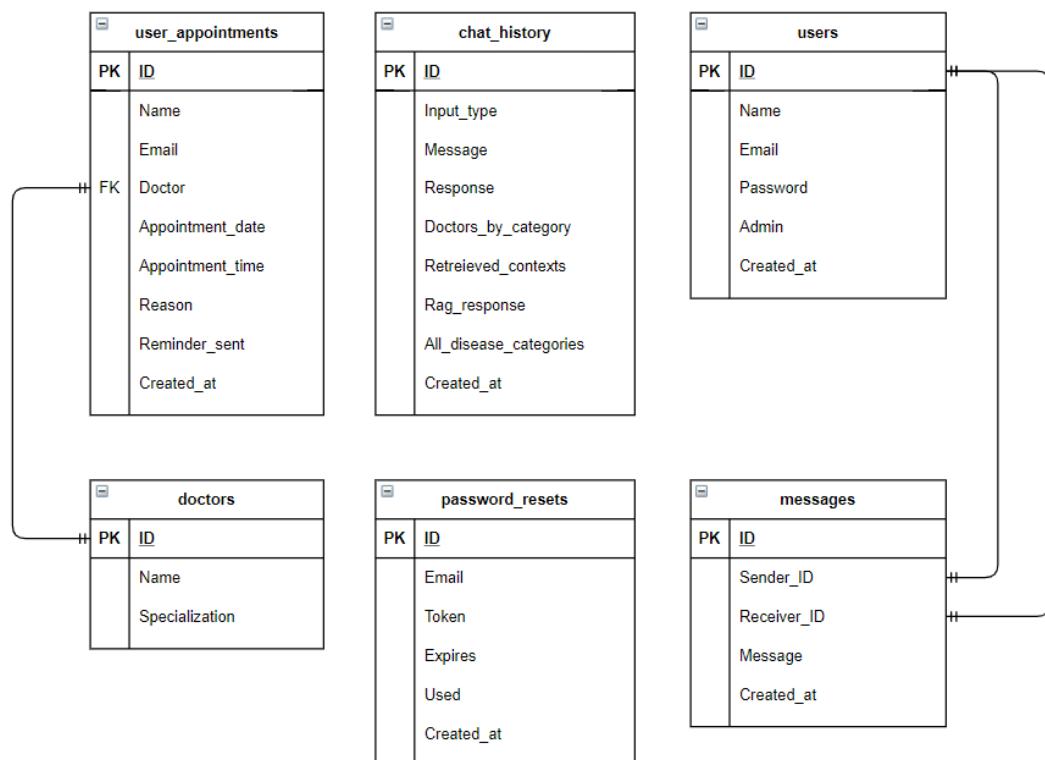


Figure 3.1.4.1 ERD diagram for Database in Mysql

Figure 3.1.4.1 show six table which are user_appointments, chat_history, users, doctors, password_resets and messages. The user_appointments table include attributes such as ID, Name, Doctor, Appointment Date, Appointment Time, Reason of booking, Reminder_sent. These informations are required for user to make appointments. The reminder sent attribute is used to send reminder to user. The chat_history table include attributes such as Input Type (text), Message (user query from chatbot interface), Response (all chatbot engine responded answer), Doctor By Category (Combination of recommended doctor and disease categories), Retrieved_contexts (the retrieve text that get when the chatbot response for generating response to user), Rag_response (part of chatbot engine responded answer), All_disease_categories (disease category predicted by the chatbot engine). These informations will be used for the dashboard visualization, report generation.

CHAPTER 3

The users table include attribute such as ID, Name, Email, Password, Admin (number to identify role). The doctors table include attribute such as ID, Name (Doctor Name), Specialization (Doctor Specialist). The password reset table includes attribute such as ID, Email, Token, Expires, Used. This table is used for password reset purpose. Last, the message table include ID, sender_ID, receiver_ID, Message attribute. This table is to store the query made when communication between users and doctor.

Figure 3.1.4.2 show the ERD diagram, the table that store the link of ngrok API link

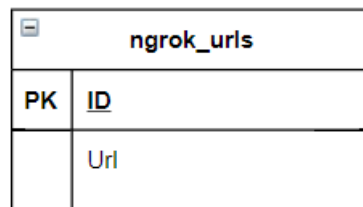


Figure 3.1.4.2 ERD Diagram for Database in Supabase

Figure 3.1.4.2 show a table name ngrok_urls, the table consist of attributes ID, and Url. The purpose of this table is used to store the Url that generated from ngrok API which allowing to connect Google Colab to local machine.

CHAPTER 4: System Design

4.1 System Block Diagram

Figure 4.1.1.1 show the system architecture diagram of the system. Include what tool is used and what page included.

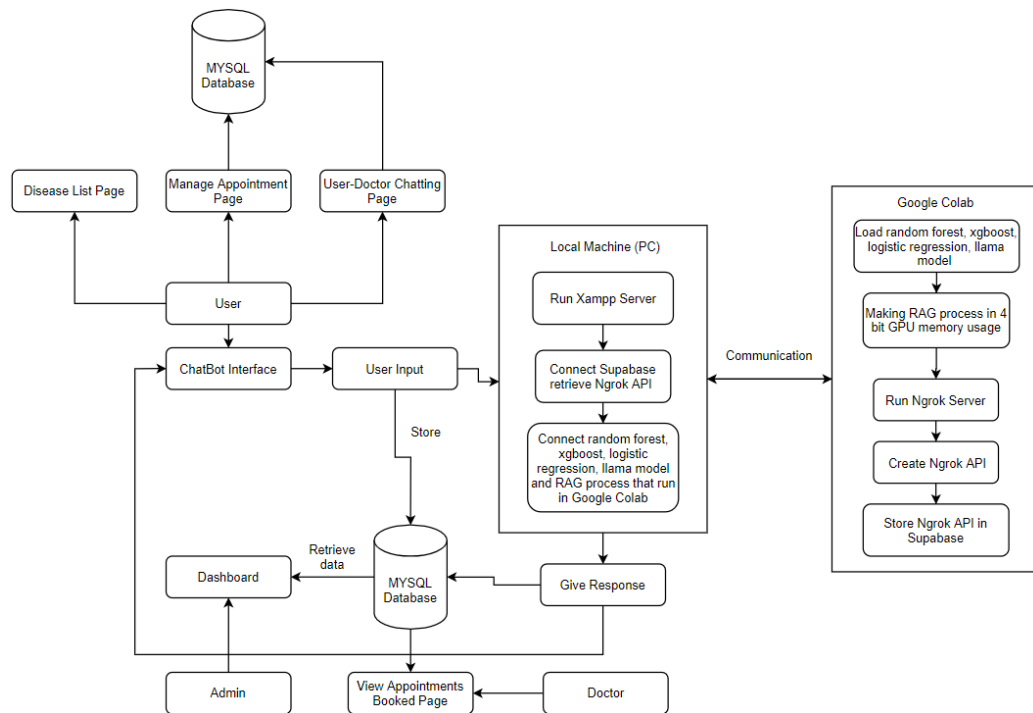


Figure 4.1.1 System Block Diagram

The Google Colab will on the Ngrok server to connect to local user interface. As free Ngrok account user cannot create sub domain with specific name, whenever restart the Ngrok server, a new sub domain will be provided. Therefore, supabase is used to store the sub domain. After that, at local machine run xampp server to run PHP, connected supabase to retrieve the Ngrok API sub domain, so that can run the RAG process on the Google Colab. The reason not running the RAG process on local machine is because not enough GPU. Then, user can provide input via the local machine interface and then the user query will send to Google Colab to do disease prediction using the model and underdo RAG process and return response to user. Besides that, user could also navigate manage appointment page (book/cancel appointment), user-doctor chatting page (chat with doctor), disease list page (more information about disease with link). Doctor can see the appointment booked by user. All the data will be store and retrieved from the Mysql database.

4.2 System Components Specifications

4.2.1 RAG Development Flow

Figure 4.2.1.1 show the diagram of RAG flow

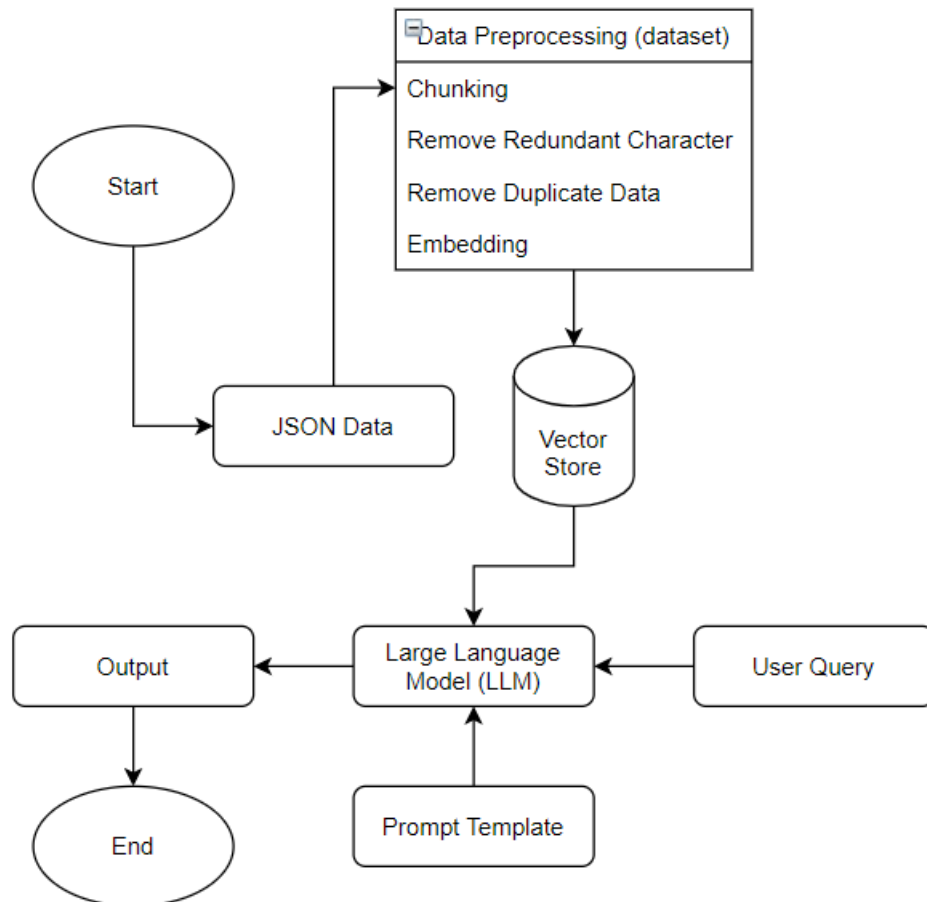


Figure 4.2.1.1 RAG Development Flow Diagram

The process for this set of RAG is help with the usage of llama index framework. The JSON dataset is loaded and doing on the data preprocessing such as chunking and embedding. Chunking means the process of making large dataset to smaller dataset with nodes that have respective id for the smaller dataset. For this chunking process, the JSON NodeParser is used chunking the dataset into smaller set with respective id. Embedding means the process of converting the text representation word to numeric vector representation that can easily understand by the computer. In this case, the sentence transformer () is used to convert the word to numeric vector representation. After that, the datasets that have done chunking and embedding are store in a vector

store. VectorStoreIndex is used to make the process of storing, indexing and embedding using the predefined embedding model (the sentence transformer). Then, when the user makes a query, the user query will also turn into vector numeric representation using the sentence transformer, based on the user query find the most relevant response from the vector store and fed both user query and retrieval response based on user query to the large language model (llama3 med42 8b). Then, the large language model will be based on the prompt template format to response to user query.

4.2.2 Classification Model Development Flow

Figure 4.2.2.1 show the diagram Classification Model Flow

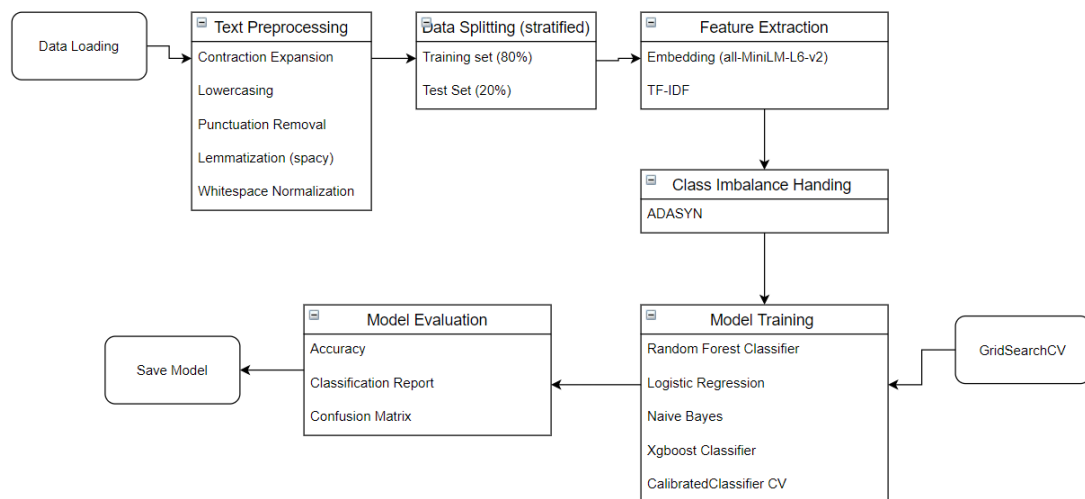


Figure 4.2.2.1 Classification Model Development Flow Diagram

Figure 4.2.2.1 shows the machine learning pipeline for building and evaluating four models. The workflow begins with data processing, text cleaning (such as re, spaCy, and contractions). The dataset, containing medical-related questions and their categories, is loaded from a JSON file. Each question undergoes a cleaning process: expanding contractions, converting to lowercase, removing noise, and lemmatizing using spaCy to reduce words to their root form.

The cleaned data is then split into features (X) and labels (y), and stratified sampling is applied to maintain the original class distribution in both training and test sets.

ADASYN is used to handle class imbalances in the training set by generating more samples for underrepresented classes.

For feature extraction, both TF-IDF vectorization and SentenceTransformer embeddings (all-MiniLM-L6-v2) are implemented. These features are passed into four classifiers: Random Forest Classifier, Logistic Regression, and Complement Naive Bayes, XgBoost Classifier. Each model is fine-tuned using GridSearchCV to find better hyperparameter, and the outputs are calibrated using sigmoid scaling for better probability interpretation.

To evaluate model, cross-validation is applied using StratifiedKFold, ensuring that each fold maintains the original class distribution. The oversampling method is applied to only the training folds when cross validation test. Five folds is undergone and four be training folds and one be test fold and iteration is made. The models are assessed using accuracy scores, classification report with the mean accuracy and standard deviation providing insights into their stability. This is followed by visualizations such as confusion matrices to help interpret model performance. After model evaluation, the models will be saved.

4.2.3 Flow on connection between Google Colab and PHP Web Application

Figure 4.2.3.1 illustrates the system architecture for integrating a chatbot module built on Google Colab with a PHP web application.

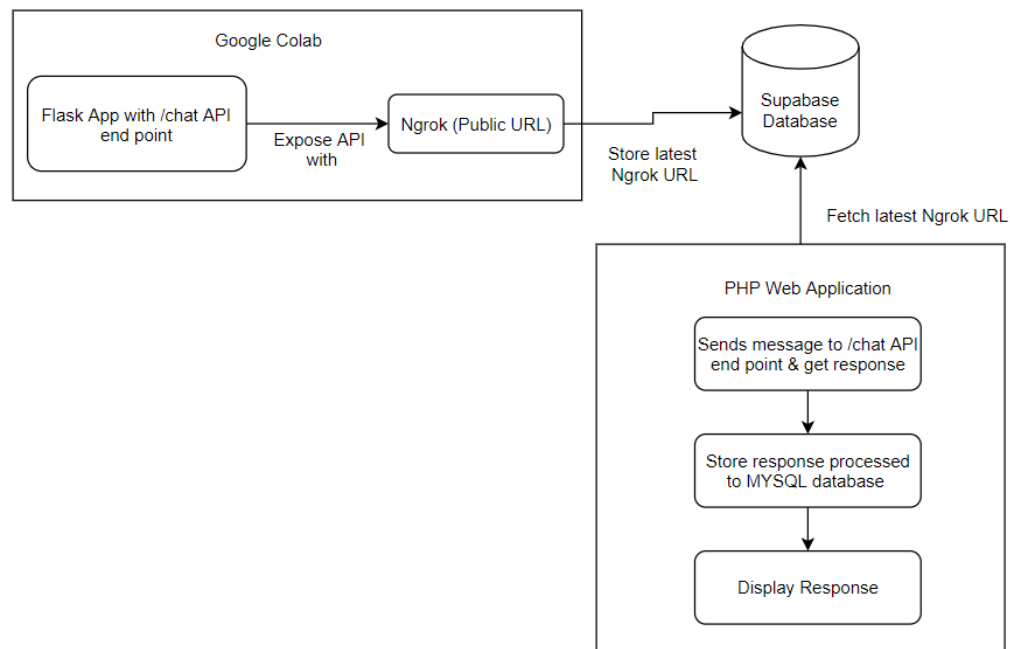


Figure 4.2.3.1 Connection Diagram on two platforms

The chatbot is built using a Flask app that provides a /chat API endpoint. Since Google Colab does not provide a permanent public URL, Ngrok is used to expose the Flask app to the internet by generating a temporary public URL. This URL changes every time the Colab environment is restarted. To handle this, the latest Ngrok URL is automatically stored in a Supabase database.

On the PHP web application side, the system first fetches the most recent Ngrok URL from the Supabase database. It then uses this URL to send user messages to the /chat API endpoint hosted in Colab. After receiving the response from the chatbot, the PHP application stores the processed response in a MySQL database. Finally, the stored response is displayed to the user through the PHP web interface. This setup allows seamless communication between the PHP frontend and the Flask-based chatbot module building that running in Google Colab, while also managing the dynamic nature of the Ngrok URL.

4.2.4 Chinese-English Translation Module Flow

Figure 4.2.4.1 illustrate on the translation module working flow

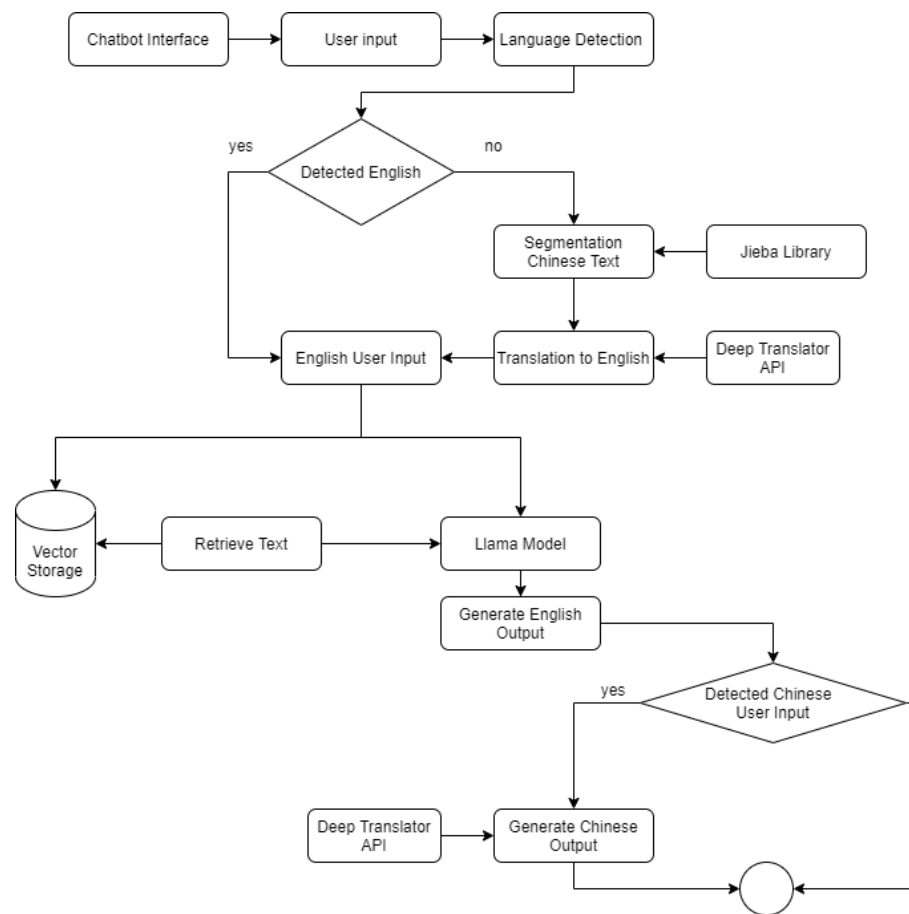


Figure 4.2.4.1 Chinese-English Translation Module Flow Diagram

When the user input Chinese language, Language detection module will detect it is Chinese or English. If the English language detected. The user input will direct pass to making text retrieved from the vector storage and based on the top similar node score sentence feed together the user input and retrieved text to the llama model. Then, the llama transformer model will generate output based on the input given. On the other hand, if Chinese language user input, the user input undergoes Chinese segmentation to make the translation more accurate when passed to the deep translator API for translation on Chinese to English language. Then, remaining flow will be same as above mentioned flow. Then, the output from the llama model will be transform to the Chinese Language with the deep translator API.

CHAPTER 5: System Implementation

5.1 Hardware Setup

Laptop is the hardware used throughout the project to develop system for UTAR hospital.

Table 5.1.1 Specifications of laptops

Description	Specifications
Model	Nitro AN515-43
Processor	AMD Ryzen7 3750H
Operating System	Windows 11
Graphic	NVIDIA GeForce GTX 1650
Memory	20GB DDR4 RAM
Storage	512GB SSD

5.2 Software Setup

The software and technology that will be used for this project is Visual Studio Code, XAMPP, MYSQL database, Supabase, Google Colab. The programming languages are PHP and Python. Below table show description on the software used.

Table 5.2.1 Tool

Tool Used	Description
Visual Studio Code [24]	Visual Studio Code serve as a code editor that seamlessly blends simplicity with powerful developer tools like IntelliSense code completion and debugging. It is available on macOS, Linux, and Windows. Visual Studio Code will be used as the editor for development of chatbot by using python language.
XAMPP [25]	XAMPP is a popular PHP development environment that offers a free and easy-to-install Apache distribution, which includes MariaDB, PHP, and Perl. As an open-source package, XAMPP is designed for ease of use, making setup and development straightforward.
Google Colab	Google Colab is platform by Google allowed write and execute Python code in your browser. Specifically, it enables you to run Jupyter notebooks without

[26]	the need to worry about hardware requirements or software installations on your computer.
Ngrok [27]	Ngrok is a cross-platform tool that allows developers to easily make their local development server accessible on the Internet. It does this by creating a secure tunnel to your local server and assigning it a public URL under the ngrok.com domain. This means you don't need a public IP address or your own domain name to share your local server online. While similar results can be achieved using Reverse SSH Tunneling, ngrok offers a much simpler setup without the need for managing a remote server.

Table 5.2.2 Programming Language

Programming Language	Description
Python [28]	Python is used as development programming language for this project. Python offer some useful library such as Tensorflow, Numpy, NLTK which help in the process of development of chatbot.
PHP [29]	PHP is a language commonly used in web development. Renowned for its speed, flexibility, and efficiency, it supports a wide range of applications from personal blogs to some of the world's most heavily trafficked websites.

Table 5.2.3 Python Library

No	Framework/Library
1	Llama-index
2	Transformers
3	Torch

4	Deep Translator
5	Jieba
6	Spacy
7	Flask
8	Pyngrok
9	Joblib
10	Supabase

Transformer model

- **Llama 3 med42 8b**

Table 5.2.4 Specification of Transformer model [30]

Specification	Value
Model name	<u>Llama3-Med42-8B</u>
Developer	M42 Health AI Team
Base model	Llama3 - 8B
Model Type	Clinical large language model (LLM)
Parameter	8 billion
Context length	8k token
Intended use	<ul style="list-style-type: none"> • Medical question answering • Patient record summarization • Aiding medical diagnosis • General health Q&A

Figure 5.2.1 show the transformer model loading method from Hugging Face using AutoTokenizer function and a token is needed for authentication purpose. 4-bit quantization is used to reduce the GPU usage when model loading.

```
tokenizer_llm = AutoTokenizer.from_pretrained("m42-health/Llama3-Med42-8B", token=HF_TOKEN)
stopping_ids = [tokenizer_llm.eos_token_id, tokenizer_llm.convert_tokens_to_ids("<|eot_id|>")]

try:
    quantization_config = BitsAndBytesConfig(
        load_in_4bit=True,
        bnb_4bit_compute_dtype=torch.float16,
        bnb_4bit_quant_type="nf4",
        bnb_4bit_use_double_quant=True
    )
    llm = HuggingFaceLLM(
        model_name="m42-health/Llama3-Med42-8B",
        model_kwargs={"token": HF_TOKEN, "quantization_config": quantization_config},
        generate_kwargs={"do_sample": True, "temperature": 0.6, "top_p": 0.9},
        tokenizer_name="m42-health/Llama3-Med42-8B",
        tokenizer_kwargs={"token": HF_TOKEN},
        stopping_ids=stopping_ids,
    )
    print("Loaded LLaMA with 4-bit quantization")
except Exception as e:
    print(f"4-bit quantization failed: {e}. Falling back to bfloat16.")
    llm = HuggingFaceLLM(
        model_name="m42-health/Llama3-Med42-8B",
        model_kwargs={"token": HF_TOKEN, "torch_dtype": torch.bfloat16},
        generate_kwargs={"do_sample": True, "temperature": 0.6, "top_p": 0.9},
        tokenizer_name="m42-health/Llama3-Med42-8B",
        tokenizer_kwargs={"token": HF_TOKEN},
        stopping_ids=stopping_ids,
    )
```

Figure 5.2.1 Load Llama 3 med42 8b model

5.3 Business Understanding

In Malaysia's multicultural society, elderly patients who speak Chinese or dialects often face language barriers when seeking healthcare, leading to miscommunication, misdiagnosis, and delayed treatment. Patients also struggle to identify the right healthcare provider due to unclear practitioner roles, increasing the risk of inappropriate consultations. Additionally, UTAR Hospital's current appointment system lacks clarity and usability, with many patients abandoning digital scheduling due to inefficiencies. These challenges highlight the need for a solution that improves communication, guidance, and access to medical services.

This project aims to develop a bilingual (English and Chinese) chatbot for UTAR Hospital's T&CM Centre that predicts possible diseases based on user-described symptoms, recommends suitable doctors, and streamlines appointment scheduling with

automated email notifications. An admin dashboard will also be implemented to visualize user interactions and appointment data, supporting better healthcare management. The chatbot targets a minimum disease prediction accuracy of 80%, with the overall goal of enhancing patient experience and service delivery in a multilingual environment.

5.4 Data Understanding

The dataset is collected from UTAR hospital website, flyer and Cleveland clinic website. The dataset from the flyer is extracted using OneNote, Google Len. The flyer (Appendix A-1, Appendix A-2) has column Practitioners, Category and Details. The Practitioners column shows the name of the practitioners. The Category column shows the main disease, for example Internal Disease, ENT, Lung System Disorder. The Details column show the keyword on disease or symptoms for the main disease that the practitioner can diagnose.

From the UTAR hospital flyer and website, data such as symptom or sub disease of disease category and doctor for respective disease category can be retrieved. From the Cleveland clinic website, the symptom of the sub disease can be found. An example of sub disease of musculoskeletal disease, kyphosis is place at Appendix A-3. Via the symptom of sub disease, sentence is formed as dataset. The dataset formed in format of “Question”, “Answer”, “Category”. The question is sentence that will likely ask by user. The answer is the sub disease of the category. The category is main disease problem. Figure 5.4.1 show the example dataset that used to train model. Figure 5.4.2 show the 10 disease categories with respective encoded class. Figure 5.4.3 show the example of UTAR hospital T&CM centre practitioners with respective disease categories. Figure 5.4.4 show the dataset quantity.

Dataset for machine learning model

```
{
  "question": "I feel a tight sensation around my ribcage, especially when I move.",
  "answer": "Tightness in the ribcage, especially during movement, might be related to intercostal neuralgia.",
  "category": "musculoskeletal problem"
},
{
  "question": "I have a throbbing pain in my lower abdomen during my period.",
  "answer": "That sounds like dysmenorrhea.",
  "category": "gynaecological disease"
},
}
```

Figure 5.4.1 Dataset example

```
Classes: ['ENT disease' 'Heart problem disease' 'Liver problem disease'
'Lung problem disease' 'Skin problem disease' 'gastrointestinal disease'
'gynaecological disease' 'internal disease' 'kidney problem disease'
'musculoskeletal problem']
Encoded Classes: [0 1 2 3 4 5 6 7 8 9]
```

Figure 5.4.2 Disease Categories with Encoded class

```
"gynaecological disease": [
  "Ms. Goh Wui Yee (Internal Medicine)",
  "Ms. Lai Phooi Yan (Acupuncture)"
],
"musculoskeletal problem": [
  "Mr. Aw Chi Min - Upper Limb Disorder (Tuina)",
  "Mr. Lim Yuan Khay - Lower Limb Disorder (Acupuncture) & [TCM Neurology]",
  "Ms. Lai Phooi Yan - Upper Limb Disorder (Acupuncture)",
  "Mr. Choo Zi Xian - Bone, Joint & Muscle System Disorder (Internal Medicine) & [Oncological Issues]",
  "Mr. David Koh Vui En - Lower Limb Disorder"
],
"internal disease": [
  "Mr. Chia Yi Keong (Acupuncture) & [TCM Psychological Issue]",
  "Dr. Te Kian Keong (Internal Medicine) - Oncological Issues"
],
"Liver problem disease": [
  "Ms. Wong Zi Xin (Internal Medicine)"
],
"Skin problem disease": [
  "Ms. Wong Zi Xin (Internal Medicine)",
  "Ms. Ng Zhi Yee (Acupuncture)"
],
}
```

Figure 5.4.3 Practitioners categories by disease

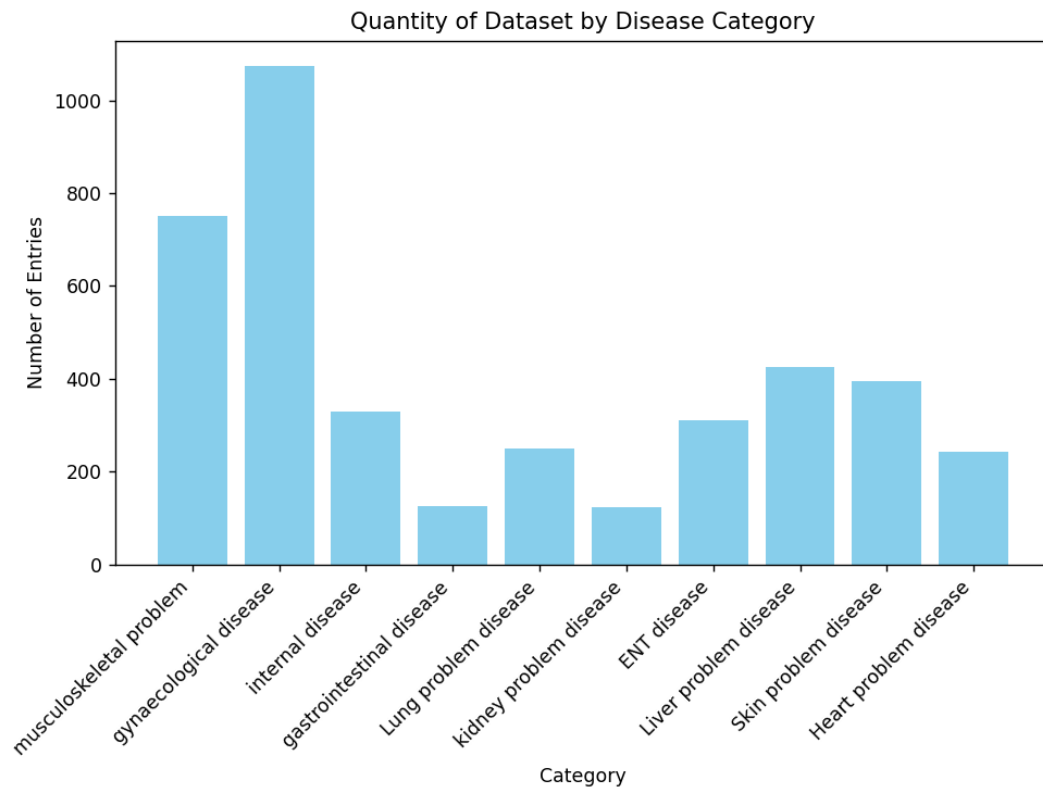


Figure 5.4.4 Quantity of dataset based on disease categories

Figure 5.4.4 show an imbalance in class distribution, which is because the flyer includes more sub-diseases under certain categories, while others have fewer sub-diseases listed. This bar graph showed dataset before data splitting.

Dataset for RAG process and transformer model usage

Figure 5.4.5 show the dataset stored in the vector storage. RAG dataset formed using the question and answer but not the category. Example datasets are showed in Figure 5.4.6. The question and the answer tag are not stored.

```
[Document(id_='5f0b9ca5-797b-4a18-b4d3-bd1f75807f2d', embedding=None, metadata={},
excluded_embed_metadata_keys=[], excluded_llm_metadata_keys=[], relationships={},
metadata_template='{key}: {value}', metadata_separator='\n',
text_resource=MediaResource(embeddings=None, data=None, text='question My knees feel stiff in
the morning but loosen up after some movement\nanswer Morning stiffness that improves with
movement could indicate joint pain. Do you also experience swelling or aching?', path=None,
url=None, mimetype=None), image_resource=None, audio_resource=None, video_resource=None,
text_template='{metadata_str}\n\n{content}')
```

Figure 5.4.5 RAG dataset

Text: I struggle to last long enough for both of us to finish. Difficulty lasting long enough to satisfy both partners can be a sign of male sexual dysfunction.

Text: I am unable to maintain an erection long enough for intercourse. Inability to maintain an erection during intercourse may be related to male sexual dysfunction.

Text: I ejaculate too soon during intercourse. That could be a sign of male sexual dysfunction.

Text: I notice I need more effort to keep an erection during intercourse. That may indicate male sexual dysfunction.

Text: I often struggle to keep an erection throughout intercourse. That could be a sign of male sexual dysfunction.

Figure 5.4.6 RAG dataset stored example without tag

5.5 Data Preparation

Data Preparation for machine learning model

Figure 5.5.1 show the oversampling dataset that using Adasyn for embedding. Figure 5.5.2 show the oversampling dataset that using Adasyn for TF-IDF.

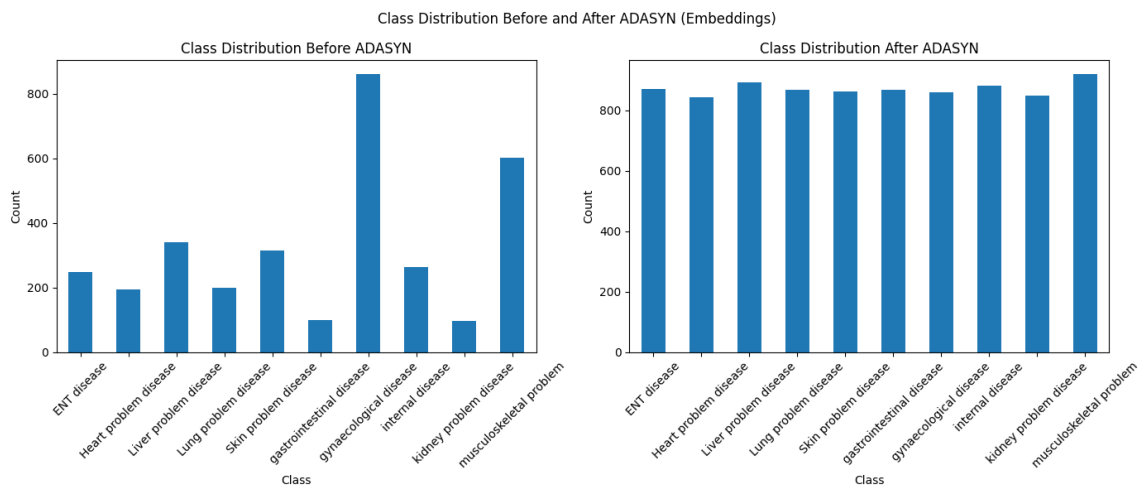


Figure 5.5.1 Oversampling dataset (Embedding)

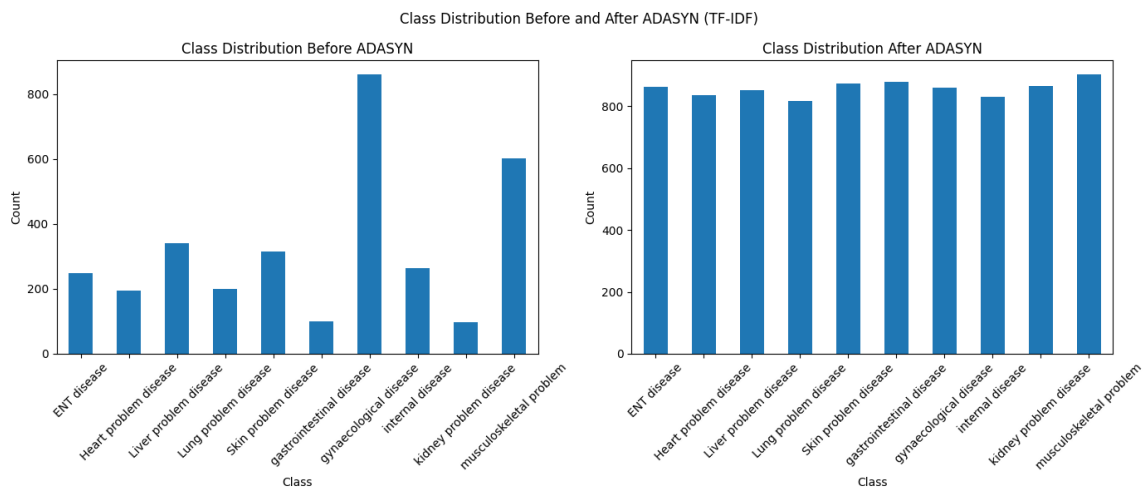


Figure 5.5.2 Oversampling dataset (TF-IDF)

Figure 5.5.1 and Figure 5.5.2 show datasets. Left side is the original dataset while right side is the dataset are oversampling using Adasyn method to make the class dataset balance, this can ensure that during model training can prevent the model overfitting to class that have more data.

Figure 5.5.3 show the example of dataset that preprocessed. The technique used for data preprocessing included removing noise and stop words, expanding contractions, converting text to lowercase, and applying lemmatization. Figure 5.5.4 show sentence with tokenize. Token ID will be given to each word. Figure 5.5.5 show the embedding example of sentence using the sentence transformer. Figure 5.5.6 show TF-IDF vector representation sentence.

=== Example of Original and Preprocessed Data ===

Sample 1:

Original: I feel like my shoulders are always rounded forward.

Preprocessed: I feel like my shoulder be always round forward

Sample 2:

Original: I have noticed a hump forming in my upper back.

Preprocessed: I have notice a hump form in my upper back

Sample 3:

Original: My hamstrings always feel tight, even when I stretch.

Preprocessed: my hamstring always feel tight even when I stretch

Figure 5.5.3 Dataset preprocessed

The sentence for example, 'my hamstring always feel tight even when I stretch' is convert to lowercase 'My' to 'my' and ',' is removed. While for the sentence 'I have noticed a hump forming in my upper back' is convert from 'noticed' to 'notice' and 'forming' to 'form'

Tokenization

```
Tokenized Output for Sample Questions:  
Sample 1: back pain when standing  
Tokens: ['[CLS]', 'back', 'pain', 'when', 'standing', '[SEP]']  
Token IDs: [101, 2067, 3255, 2043, 3061, 102]  
Sample 2: curved spine causing pain  
Tokens: ['[CLS]', 'curved', 'spine', 'causing', 'pain', '[SEP]']  
Token IDs: [101, 9203, 8560, 4786, 3255, 102]
```

Figure 5.5.4 Tokenization Example

Figure 5.5.4 show sentence like back pain when standing. The token is assigned to 2067 as back, 3255 as pain.

Embedding

```
=== Example Embeddings ===  
  
Sample 1:  
Text: I notice my tinnitus be loud when I be in a quiet space...  
Embedding (first 5 values): [ 0.10559759 -0.08583464  0.05321611  0.03240304 -0.02438306]  
Embedding shape: (384,)  
  
Sample 2:  
Text: the low part of my child leg do not align with the upper part...  
Embedding (first 5 values): [-0.02180514  0.00020488  0.02810107 -0.03117116 -0.01863489]  
Embedding shape: (384,)  
  
Sample 3:  
Text: my skin feel more sensitive and I experience irritation frequently...  
Embedding (first 5 values): [ 0.08221792 -0.07089128  0.0101326  0.07881148  0.08586723]  
Embedding shape: (384,)
```

Figure 5.5.5 Embedding Example

Figure 5.5.5 show examples of sentence embeddings, which are numerical representations of the meaning of entire sentences. For each sample, the image displays the first 5 values of the embedding and notes that the embedding shape is (384,), meaning each sentence is converted into a vector with 384 numbers. These embeddings help machines understand and compare the meanings of sentences.

TF-IDF vector representation

```

=== Example TF-IDF Vectors ===

Sample 1:
Full Preprocessed Text: I notice my tinnitus be loud when I be in a quiet space
Top TF-IDF features: [('be loud', np.float64(0.3578)), ('my tinnitus', np.float64(0.3578)), ('tinnitus', np.float64(0.3578)), ('loud when', np.float64(0.3449)),

Sample 2:
Full Preprocessed Text: the low part of my child leg do not align with the upper part
Top TF-IDF features: [('part', np.float64(0.4363)), ('not align', np.float64(0.264)), ('align with', np.float64(0.264)), ('upper part', np.float64(0.2545)), ('w

Sample 3:
Full Preprocessed Text: my skin feel more sensitive and I experience irritation frequently
Top TF-IDF features: [('experience irritation', np.float64(0.3598)), ('sensitive and', np.float64(0.3468)), ('more sensitive', np.float64(0.3367)), ('and experi

```

Figure 5.5.6 TF-IDF Vector Representation Example

Figure 5.5.6, each sample contains a preprocessed sentence followed by a list of its top TF-IDF features. These features include word or phrase combinations like "be loud", "my tinnitus", or "experience irritation", each paired with a numerical value that represents its importance. Higher values indicate terms that are more relevant to the individual sample and less common in the other samples, making them useful for identifying key topics or symptoms being described.

5.6 Modeling

Figure 5.6.1 show data splitting function, Figure 5.6.2 show the adasyn oversampling function on training data set, Figure 5.6.3 show the best cross validation score on gridsearchcv test on original dataset, Figure 5.6.4 until Figure 5.6.7 show the functions of classification model

```

RANDOM_STATE = 42
TEST_SIZE = 0.2
CV_FOLDS = 5

# --- Split Data ---
X_train, X_test, y_train, y_test = train_test_split(
    X, y_encoded, test_size=TEST_SIZE, random_state=RANDOM_STATE, stratify=y_encoded
)

```

Figure 5.6.1 Data splitting

The data set is split into X and Y, X is the question and Y is the categories. 80% is the training dataset and 20% is the test set. Stratify is used to ensure all class is distributed equally.

```
# --- Apply ADASYN Oversampling ---
adasyn = ADASYN(random_state=RANDOM_STATE)
X_train_resampled_emb, y_train_resampled_emb = adasyn.fit_resample(X_train_embeddings, y_train)
X_train_resampled_tfidf, y_train_resampled_tfidf = adasyn.fit_resample(X_train_tfidf, y_train)
print("\nAfter ADASYN (Embeddings), training set class distribution:\n", pd.Series(label_encoder.inverse_transform(y_train_resampled_emb)).value_counts())
print("\nAfter ADASYN (TF-IDF), training set class distribution:\n", pd.Series(label_encoder.inverse_transform(y_train_resampled_tfidf)).value_counts())
```

Figure 5.6.2 Adasyn oversampling on training dataset

After the dataset is undergo embedding of TF-IDF, convert into vector representation, the training dataset undergoes oversampling.

```
=== GridSearchCV for Naive Bayes (TF-IDF) with TF-IDF Tuning ===
Best Parameters: {'nb__alpha': 0.5, 'nb__fit_prior': True, 'tfidf__max_df': 0.8, 'tfidf__max_features': 10000, 'tfidf__min_df': 1, 'tfidf__ngram_range': (1, 2)}
Best Cross-Validation Accuracy: 0.7722605806731186

Best TF-IDF Parameters from Naive Bayes GridSearch: {'max_features': 10000, 'ngram_range': (1, 2), 'min_df': 1, 'max_df': 0.8}

=== GridSearchCV for Random Forest (Embeddings) ===
Best Parameters: {'class_weight': 'balanced', 'max_depth': None, 'max_features': 'log2', 'min_samples_split': 10, 'n_estimators': 300}
Best Cross-Validation Accuracy: 0.7998757763975155

=== GridSearchCV for Random Forest (TF-IDF) ===
Best Parameters: {'rf__class_weight': 'balanced', 'rf__max_depth': None, 'rf__max_features': 'log2', 'rf__min_samples_split': 5, 'rf__n_estimators': 200}
Best Cross-Validation Accuracy: 0.7772300062593288

=== GridSearchCV for Logistic Regression (Embeddings) ===
Best Parameters: {'C': 10.0, 'class_weight': 'balanced', 'max_iter': 1000, 'penalty': 'l2', 'solver': 'liblinear'}
Best Cross-Validation Accuracy: 0.799571958206943

=== GridSearchCV for Logistic Regression (TF-IDF) ===
Best Parameters: {'lr__C': 10.0, 'lr__class_weight': 'balanced', 'lr__max_iter': 1000, 'lr__penalty': 'l2', 'lr__solver': 'liblinear'}
Best Cross-Validation Accuracy: 0.7989489142471953
```

Figure 5.6.3 GridSearch CV result without oversampling

Figure 5.6.3 show the best cross validation accuracy with 5 folds set. The range are between 0.77 until 0.79, the random forest classifier done the best over other two classifier model. While logistic regression classification model also done quick well compared to naïve bayes. XGboost classifiers not applied with Gridsearch CV because the session always crashes when using Google Colab to find best hyperparamter.

```
# Random Forest (Embeddings)
rf_calibrated_emb = CalibratedClassifierCV(
    RandomForestClassifier(**rf_best_params_emb, random_state=RANDOM_STATE),
    method='sigmoid', cv=5, n_jobs=-1
)
rf_calibrated_emb.fit(X_train_resampled_emb, y_train_resampled_emb)

# Random Forest (TF-IDF)
rf_calibrated_tfidf = CalibratedClassifierCV(
    RandomForestClassifier(**rf_best_params_tfidf, random_state=RANDOM_STATE),
    method='sigmoid', cv=5, n_jobs=-1
)
rf_calibrated_tfidf.fit(X_train_resampled_tfidf, y_train_resampled_tfidf)
```

Figure 5.6.4 Random Forest Classifier

The best parameter is trained on the randomforest classifier for two cases, Embedding and TF-IDF. The calibratedclassifierCV is used to adjust predicted probabilities to better reflect the true likelihood of each class.

```
# Logistic Regression (Embeddings)
lr_calibrated_emb = CalibratedClassifierCV(
    LogisticRegression(**lr_best_params_emb, random_state=RANDOM_STATE),
    method='sigmoid', cv=5, n_jobs=-1
)
lr_calibrated_emb.fit(X_train_resampled_emb, y_train_resampled_emb)

# Logistic Regression (TF-IDF)
lr_calibrated_tfidf = CalibratedClassifierCV(
    LogisticRegression(**lr_best_params_tfidf, random_state=RANDOM_STATE),
    method='sigmoid', cv=5, n_jobs=-1
)
lr_calibrated_tfidf.fit(X_train_resampled_tfidf, y_train_resampled_tfidf)
```

Figure 5.6.5 Logistic Regression

The best parameter is trained on the logistic regression classifier for two cases, Embedding and TF-IDF. The calibratedclassifierCV is used to adjust predicted probabilities to better reflect the true likelihood of each class.

```
# Naïve Bayes
nb_model = ComplementNB(**nb_best_params)
nb_model.fit(X_train_resampled_tfidf, y_train_resampled_tfidf)
```

Figure 5.6.6 Complement Naïve Bayes

The best parameter is trained on the the complement naïve bayes for two cases, Embedding and TF-IDF. The complement naïve bayes is used because it can handle better for imbalance class compared to multinomial naïve bayes.

```

# XGBoost (Embeddings)
xgb_best_emb = xgb.XGBClassifier(
    n_estimators=200,
    max_depth=6,
    learning_rate=0.1,
    subsample=0.8,
    colsample_bytree=0.8,
    objective='multi:softprob',
    random_state=RANDOM_STATE,
    n_jobs=-1
)
xgb_calibrated_emb = CalibratedClassifierCV(
    xgb_best_emb,
    method='sigmoid', cv=5, n_jobs=-1
)
xgb_calibrated_emb.fit(X_train_resampled_emb, y_train_resampled_emb)

# XGBoost (TF-IDF)
xgb_calibrated_tfidf = CalibratedClassifierCV(
    xgb.XGBClassifier(
        n_estimators=200,
        max_depth=6,
        learning_rate=0.1,
        subsample=0.8,
        colsample_bytree=0.8,
        objective='multi:softprob',
        random_state=RANDOM_STATE,
        n_jobs=-1
    ),
    method='sigmoid', cv=5, n_jobs=-1
)
xgb_calibrated_tfidf.fit(X_train_resampled_tfidf, y_train_resampled_tfidf)

```

Figure 5.6.7 Xgboost Classifier

The best parameter is trained on the xgbosst classifier for two cases, Embedding and TF-IDF. The calibratedclassifierCV is used to adjust predicted probabilities to better reflect the true likelihood of each class. Above code also shows the hyperparameter used for the model training.

5.7 Evaluation

Random Forest Classifier (Embedding)

Figure 5.7.1 show the classification report and top 3 accuracy evaluation. A Random Forest machine learning model was used to predict different types of diseases based on patient information. It applied word embeddings to better understand symptom-related terms. Tested on 806 samples, the model achieved 82% accuracy in correctly identifying unseen dataset.

```

=== Random Forest (Embeddings) Evaluation ===
Random Forest (Embeddings) Classification Report:
              precision    recall  f1-score   support

    ENT disease           0.80      0.90      0.85         62
    Heart problem disease  0.55      0.57      0.56         49
    Liver problem disease  0.76      0.75      0.76         85
    Lung problem disease   0.72      0.52      0.60         50
    Skin problem disease   0.92      0.97      0.94         79
    gastrointestinal disease 0.83      0.76      0.79         25
    gynaecological disease  0.85      0.89      0.87        215
    internal disease       0.75      0.71      0.73         66
    kidney problem disease  1.00      0.44      0.61         25
    musculoskeletal problem 0.89      0.93      0.91        150

    accuracy                   0.82        806
    macro avg           0.81      0.75      0.76        806
    weighted avg        0.82      0.82      0.81        806

Random Forest (Embeddings) Accuracy: 0.818
Random Forest (Embeddings) Top-3 Accuracy: 0.963

```

Figure 5.7.1 Evaluation on Random Forest Classifier (Embedding)

The classification report shows an overall accuracy of 82%, with a high top 3 accuracy of 96.3%, indicating the model often ranks the correct class within its top three predictions. The weighted F1-score of 0.81 suggests balanced performance across classes, while the macro F1-score of 0.76 reveals that some classes, particularly those with fewer samples, are less accurately predicted. High-performing classes include skin disease (F1-score: 0.94) and musculoskeletal problems (F1-score: 0.91), reflecting strong precision and recall. In contrast, the model struggles with kidney, heart, and lung diseases. Kidney disease shows perfect precision (1.00) but low recall (0.44), meaning many true cases were missed. Heart and lung problems also show low F1-scores (around 0.56–0.60), suggesting confusion with other classes.

Figure 5.7.2 show the test set and model stability evaluation. The model perform stable as the standard deviation is low and test set is displayed with prediction result.

```
Random Forest (Embeddings) Top-3 Predictions (Test Set):
Sample 1: my leg appear to be of different length...
True Label: musculoskeletal problem
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.93)), ('gynaecological disease', np.float64(0.035)), ('kidney problem disease', np.float64(0.015))]

Sample 2: I be unable to get pregnant despite be in good health...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.971)), ('Liver problem disease', np.float64(0.007)), ('kidney problem disease', np.float64(0.006))]

Sample 3: some period last one day other up to eight day...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.972)), ('kidney problem disease', np.float64(0.006)), ('Liver problem disease', np.float64(0.005))]

Sample 4: my skin feel very tight especially around my neck...
True Label: Skin problem disease
Top-3 Predictions: [('Skin problem disease', np.float64(0.317)), ('musculoskeletal problem', np.float64(0.262)), ('gynaecological disease', np.float64(0.156))]

Sample 5: I notice that my leg look puffy than usual...
True Label: kidney problem disease
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.575)), ('kidney problem disease', np.float64(0.211)), ('gynaecological disease', np.float64(0.078))]

Random Forest (Embeddings) Stability (Cross-Validation on Original Data):
Mean Accuracy: 0.812
Standard Deviation: 0.017
Accuracy Range: [0.779, 0.845]
```

Figure 5.7.2 Evaluation on Test case and Stability

The Random Forest model with embeddings demonstrates in classifying disease-related symptoms. In the top 3 predictions for test samples, the correct label appeared within the top 3 suggestions in all cases and was ranked first in four out of five samples. This highlights the model's reliability in suggesting accurate diagnoses, even if the top prediction isn't always correct. Misclassifications occurred between musculoskeletal, kidney, and gynaecological diseases, likely due to overlapping symptoms but still predicted in top 3 classes. Additionally, cross-validation using adasyn on training folds results show a mean accuracy of 81.2% with a low standard deviation of 0.017, indicating consistent performance across different data splits. The accuracy ranged from 77.9% to 84.5%, confirming the model's robustness and making it suitable for use in medical decision-support systems where multiple suggestions can guide further examination.

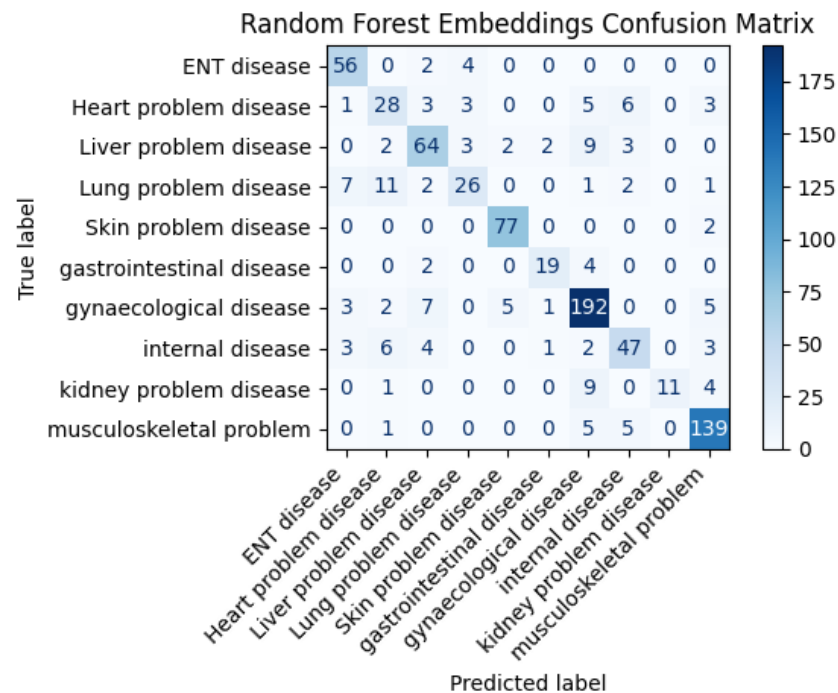


Figure 5.7.3 Confusion Matrix of Random Forest Classifier

Figure 5.7.3 show the confusion matrix evaluation on the random forest classifier model. The model performs quick well for ENT disease (56/62), skin problem disease (77/79), gastrointestinal disease (19/25), gynaecological disease (192/215) and musculoskeletal problem (139/150). However, lung problem disease and kidney problem disease show more confusion. For instance, lung disease is often misclassified as heart disease (11 times) and kidney problem disease suffers from misclassification as gynaecological disease (9 times) across several classes due to its small sample size. The confusion between kidney disease, heart, and lung issues suggest overlapping features that the model struggles to separate. This can be due to the reason; similar symptoms exist between the classes cause misclassification.

Random Forest Classifier (TF-IDF)

Figure 5.7.4 show the classification report and top 3 accuracy evaluation. A Random Forest machine learning model using TF-IDF features was used to predict different types of diseases based on patient information. Tested on 806 samples, the model achieved 80% accuracy in correctly identifying unseen dataset.

```

=== Random Forest (TF-IDF) Evaluation ===
Random Forest (TF-IDF) Classification Report:

```

	precision	recall	f1-score	support
ENT disease	0.83	0.79	0.81	62
Heart problem disease	0.56	0.37	0.44	49
Liver problem disease	0.76	0.81	0.78	85
Lung problem disease	0.56	0.54	0.55	50
Skin problem disease	0.92	0.91	0.92	79
gastrointestinal disease	0.89	0.68	0.77	25
gynaecological disease	0.80	0.93	0.86	215
internal disease	0.72	0.67	0.69	66
kidney problem disease	0.94	0.60	0.73	25
musculoskeletal problem	0.88	0.91	0.89	150
accuracy			0.80	806
macro avg	0.79	0.72	0.75	806
weighted avg	0.80	0.80	0.80	806

```

Random Forest (TF-IDF) Accuracy: 0.801
Random Forest (TF-IDF) Top-3 Accuracy: 0.943

```

Figure 5.7.4 Evaluation on Random Forest Classifier (TF-IDF)

The classification report shows an overall accuracy of 80%, with a high top 3 accuracy of 94.3%, indicating strong performance in ranking the correct disease within the top three predictions. The model performed particularly well for classes with more samples, such as skin, gynaecological and musculoskeletal problems, both achieving high recall and F1-scores (above 0.85). However, performance dropped for less represented or more symptomatically overlapping classes, such as heart and lung diseases, with F1-scores of 0.44 and 0.55, respectively. The macro average F1-score of 0.75 suggests moderate performance across all classes, while the weighted average of 0.80 reflects the model's overall balance in handling class imbalance. This shows the model is reliable for many classes but struggles with some minority or ambiguous cases.

Figure 5.7.5 show the test set and model stability evaluation. The model perform stable as the standard deviation is low and test set is displayed with prediction result.

```
Random Forest (TF-IDF) Top-3 Predictions (Test Set):
Sample 1: my leg appear to be of different length...
True Label: musculoskeletal problem
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.867)), ('gynaecological disease', np.float64(0.122)), ('Liver problem disease', np.float64(0.003))]

Sample 2: I be unable to get pregnant despite be in good health...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.988)), ('Liver problem disease', np.float64(0.005)), ('kidney problem disease', np.float64(0.002))]

Sample 3: some period last one day other up to eight day...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.976)), ('musculoskeletal problem', np.float64(0.016)), ('Lung problem disease', np.float64(0.002))]

Sample 4: my skin feel very tight especially around my neck...
True Label: Skin problem disease
Top-3 Predictions: [('Skin problem disease', np.float64(0.936)), ('musculoskeletal problem', np.float64(0.034)), ('gynaecological disease', np.float64(0.023))]

Sample 5: I notice that my leg look puffy than usual...
True Label: kidney problem disease
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.661)), ('gynaecological disease', np.float64(0.164)), ('Liver problem disease', np.float64(0.145))]

Random Forest (TF-IDF) Stability (Cross-Validation on Original Data):
Mean Accuracy: 0.790
Standard Deviation: 0.019
Accuracy Range: [0.752, 0.827]
```

Figure 5.7.5 Evaluation on Test case and Stability

The Random Forest model with TF-IDF feature extraction demonstrates in classifying disease-related symptoms. In the top 3 predictions for test samples, the correct label appeared within the top 3 suggestions in 4 cases except for sample 5. Misclassifications occurred in sample 5 may be due to the reason TF-IDF vector representation not based on the whole sentence but rare word or common word causing the model cannot catch the pattern well on identifying disease. Additionally, cross-validation using adasyn on training folds results show a mean accuracy of 79% with a low standard deviation of 0.019, indicating consistent performance across different data splits. The accuracy ranged from 75.2% to 82.7%.

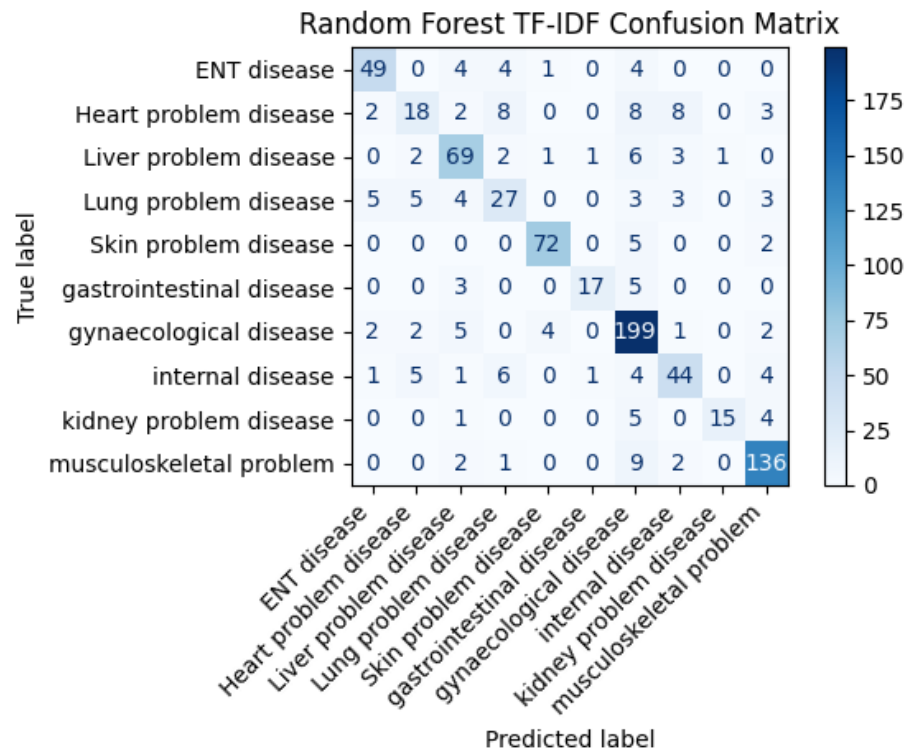


Figure 5.7.6 Confusion Matrix of Random Forest Classifier

Figure 5.7.6 show the confusion matrix evaluation on the random forest classifier model. The model performs quick well for skin problem disease (72/79), gynaecological disease (199/215) and musculoskeletal problem (136/150). The confusion between classes suggest overlapping features that the model struggles to separate. This can be due to the reason; similar symptoms exist between the classes cause misclassification. This proves that the project could have improvement in data preparation and containing similar symptoms in different diseases is also possible.

Logistic Regression (Embedding)

Figure 5.7.7 show the classification report and top 3 accuracy evaluation. A Logistic Regression machine learning model was used to predict different types of diseases based on patient information. It applied word embeddings to better understand symptom-related terms. Tested on 806 samples, the model achieved 78% accuracy in correctly identifying unseen dataset.

```

=== Logistic Regression (Embeddings) Evaluation ===
Logistic Regression (Embeddings) Classification Report:

```

	precision	recall	f1-score	support
ENT disease	0.78	0.85	0.82	62
Heart problem disease	0.44	0.51	0.47	49
Liver problem disease	0.72	0.66	0.69	85
Lung problem disease	0.54	0.52	0.53	50
Skin problem disease	0.86	0.99	0.92	79
gastrointestinal disease	0.61	0.88	0.72	25
gynaecological disease	0.92	0.79	0.85	215
internal disease	0.62	0.64	0.63	66
kidney problem disease	0.79	0.76	0.78	25
musculoskeletal problem	0.89	0.91	0.90	150
accuracy			0.78	806
macro avg	0.72	0.75	0.73	806
weighted avg	0.79	0.78	0.78	806

```

Logistic Regression (Embeddings) Accuracy: 0.777
Logistic Regression (Embeddings) Top-3 Accuracy: 0.937

```

Figure 5.7.7 Evaluation on Logistic Regression

The classification report shows an overall accuracy of 78%, with a high top 3 accuracy of 93.7%, indicating the model often ranks the correct class within its top three predictions. The weighted F1-score of 0.78 model performs well overall, especially on frequent diseases, while the macro F1-score of 0.73 reveals that some classes, particularly those with fewer samples, are less accurately predicted. High-performing classes include skin disease (F1-score: 0.92) and musculoskeletal problems (F1-score: 0.90), reflecting strong precision and recall. In contrast, the model struggles with heart, and lung diseases. Heart and lung problems also show low F1-scores (around 0.47–0.53), suggesting confusion with other classes.

Figure 5.7.8 show the test set and model stability evaluation. The model perform stable as the standard deviation is low and test set is displayed with prediction result.

```

Logistic Regression (Embeddings) Top-3 Predictions (Test Set):
Sample 1: my leg appear to be of different length...
True Label: musculoskeletal problem
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.696)), ('Liver problem disease', np.float64(0.278)), ('kidney problem disease', np.float64(0.016))]

Sample 2: I be unable to get pregnant despite be in good health...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.876)), ('Liver problem disease', np.float64(0.1)), ('Heart problem disease', np.float64(0.021))]

Sample 3: some period last one day other up to eight day...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.986)), ('kidney problem disease', np.float64(0.009)), ('Liver problem disease', np.float64(0.003))]

Sample 4: my skin feel very tight especially around my neck...
True Label: Skin problem disease
Top-3 Predictions: [('Skin problem disease', np.float64(0.892)), ('kidney problem disease', np.float64(0.043)), ('musculoskeletal problem', np.float64(0.032))]

Sample 5: I notice that my leg look puffy than usual...
True Label: kidney problem disease
Top-3 Predictions: [('kidney problem disease', np.float64(0.58)), ('Heart problem disease', np.float64(0.323)), ('musculoskeletal problem', np.float64(0.08))]

Logistic Regression (Embeddings) Stability (Cross-Validation on Original Data):
Mean Accuracy: 0.789
Standard Deviation: 0.010
Accuracy Range: [0.769, 0.810]

```

Figure 5.7.8 Evaluation on Test case and Stability

The Logistic Regression with embeddings demonstrates in classifying disease-related symptoms. In the top 3 predictions for test samples, the correct label appeared within the top 3 suggestions in all cases. This highlights the model's reliability in suggesting accurate diagnoses. Cross validation using adasyn on training folds results show a mean accuracy of 78.9% with a low standard deviation of 0.01, indicating consistent performance across different data splits. The accuracy ranged from 76.9% to 81%. However, even test set is fully predicted but the overall accuracy does not meet 80%, improvement can be made by adding the dataset.

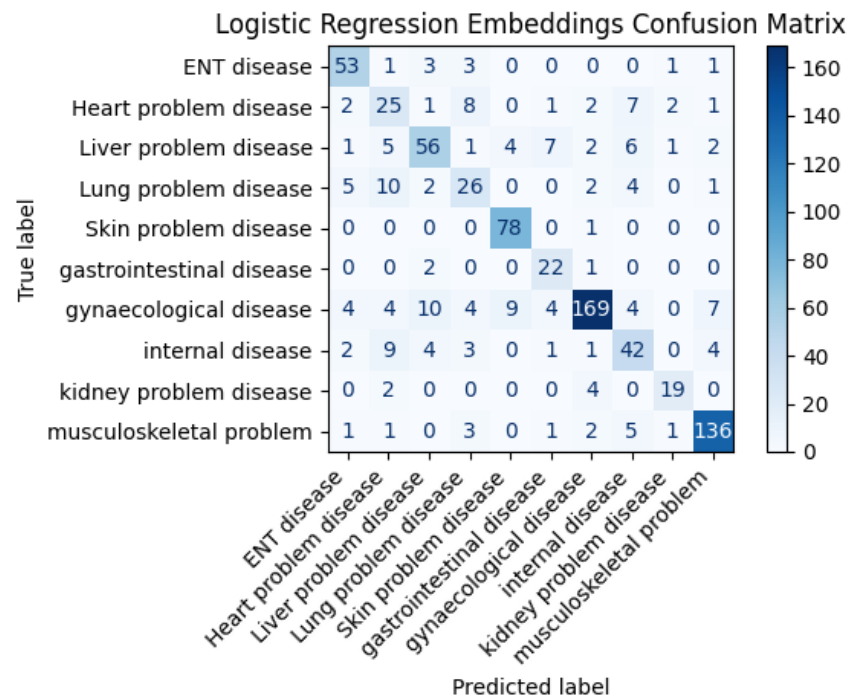


Figure 5.7.9 Confusion Matrix of Logistic Regression

Figure 5.7.9 show the confusion matrix evaluation on the logistic regression model. The model performs quick well for skin problem disease (78/79) and musculoskeletal problem (136/150). The confusion between classes suggests overlapping features that the model struggles to separate. This can be due to the reason; similar symptoms exist between the classes cause misclassification. For example, the model missclassificate 10 gynaecological disease cases into liver disease, this could be due to the reason of disease like hormone imbalance symptom such as fatigue and numbness. This proves that the project could make improvements in data preparation as containing similar symptoms in different diseases is also possible.

Logistic Regression (TF-IDF)

Figure 5.7.10 show the classification report and top 3 accuracy evaluation. A Logistic Regression machine learning model using TF-IDF features was used to predict different types of diseases based on patient information. Tested on 806 samples, the model achieved 80% accuracy in correctly identifying unseen dataset.

```

=== Logistic Regression (TF-IDF) Evaluation ===
Logistic Regression (TF-IDF) Classification Report:

```

	precision	recall	f1-score	support
ENT disease	0.80	0.84	0.82	62
Heart problem disease	0.54	0.41	0.47	49
Liver problem disease	0.72	0.68	0.70	85
Lung problem disease	0.59	0.54	0.56	50
Skin problem disease	0.93	0.99	0.96	79
gastrointestinal disease	0.70	0.76	0.73	25
gynaecological disease	0.84	0.87	0.86	215
internal disease	0.65	0.62	0.64	66
kidney problem disease	0.83	0.76	0.79	25
musculoskeletal problem	0.89	0.93	0.91	150
accuracy			0.80	806
macro avg	0.75	0.74	0.74	806
weighted avg	0.79	0.80	0.79	806

```

Logistic Regression (TF-IDF) Accuracy: 0.797
Logistic Regression (TF-IDF) Top-3 Accuracy: 0.927

```

Figure 5.7.10 Evaluation on Logistic Regression (TF-IDF)

The classification report shows an overall accuracy of 80%, with a high top 3 accuracy of 92.7%, indicating the model often ranks the correct class within its top three predictions. The weighted F1-score of 0.79 model performs well overall, especially on frequent diseases, while the macro F1-score of 0.74 reveals that some classes, particularly those with fewer samples, are less accurately predicted. High-performing classes include skin disease (F1-score: 0.96) and musculoskeletal problems (F1-score: 0.91), reflecting strong precision and recall. In contrast, the model struggles with heart, and lung diseases. Heart and lung problems also show low F1-scores (around 0.47–0.56), suggesting confusion with other classes.

Figure 5.7.11 show the test set and model stability evaluation. The model perform stable as the standard deviation is low and test set is displayed with prediction result.

```
Logistic Regression (TF-IDF) Top-3 Predictions (Test Set):
Sample 1: my leg appear to be of different length...
True Label: musculoskeletal problem
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.965)), ('kidney problem disease', np.float64(0.019)), ('gynaecological disease', np.float64(0.012))]

Sample 2: I be unable to get pregnant despite be in good health...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.995)), ('Liver problem disease', np.float64(0.002)), ('kidney problem disease', np.float64(0.001))]

Sample 3: some period last one day other up to eight day...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.993)), ('musculoskeletal problem', np.float64(0.005)), ('gastrointestinal disease', np.float64(0.001))]

Sample 4: my skin feel very tight especially around my neck...
True Label: Skin problem disease
Top-3 Predictions: [('Skin problem disease', np.float64(0.99)), ('musculoskeletal problem', np.float64(0.009)), ('kidney problem disease', np.float64(0.0))]

Sample 5: I notice that my leg look puffy than usual...
True Label: kidney problem disease
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.953)), ('kidney problem disease', np.float64(0.028)), ('gynaecological disease', np.float64(0.014))]

Logistic Regression (TF-IDF) Stability (Cross-Validation on Original Data):
Mean Accuracy: 0.800
Standard Deviation: 0.020
Accuracy Range: [0.759, 0.840]
```

Figure 5.7.11 Evaluation on Test case and Stability

The Logistic Regression with TF-IDF extraction demonstrates in classifying disease-related symptoms. In the top 3 predictions for test samples, the correct label appeared within the top 3 suggestions in all cases and was ranked first in four out of five samples. This highlights the model's reliability in suggesting accurate diagnoses, even if the top prediction isn't always correct. Misclassifications occurred between musculoskeletal, kidney, and gynaecological diseases, likely due to overlapping symptoms but still predicted in top 3 classes. Additionally, cross-validation using adasyn on training folds results show a mean accuracy of 80% with a low standard deviation of 0.02, indicating consistent performance across different data splits. The accuracy ranged from 75.9% to 84%, confirming the model's robustness and making it suitable for use in medical decision-support systems where multiple suggestions can guide further examination.

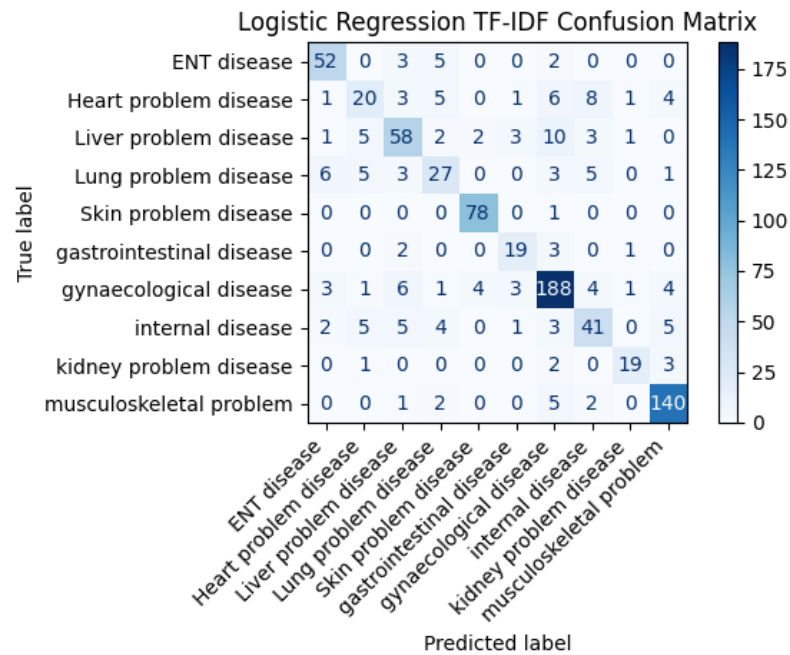


Figure 5.7.12 Confusion Matrix of Logistic Regression

Figure 5.7.12 show the confusion matrix evaluation on the logistic regression model. The model performs quick well for skin problem disease (78/79) and musculoskeletal problem (140/150). The confusion between classes suggests overlapping features that the model struggles to separate. This can be due to the reason; similar symptoms exist between the classes cause misclassification. This proves that the project could have improvement in data preparation and containing similar symptoms in different diseases is also possible.

XGBoost Classifier (Embedding)

Figure 5.7.13 show the classification report and top 3 accuracy evaluation. A Xgboost Classifier machine learning model was used to predict different types of diseases based on patient information. It applied word embeddings to better understand symptom-related terms. Tested on 806 samples, the model achieved 84% accuracy in correctly identifying unseen dataset.

```

=== XGBoost (Embeddings) Evaluation ===
XGBoost (Embeddings) Classification Report:

```

	precision	recall	f1-score	support
ENT disease	0.82	0.90	0.86	62
Heart problem disease	0.58	0.59	0.59	49
Liver problem disease	0.80	0.75	0.78	85
Lung problem disease	0.72	0.58	0.64	50
Skin problem disease	0.92	0.96	0.94	79
gastrointestinal disease	0.74	0.80	0.77	25
gynaecological disease	0.90	0.90	0.90	215
internal disease	0.72	0.70	0.71	66
kidney problem disease	0.91	0.84	0.88	25
musculoskeletal problem	0.89	0.93	0.91	150
accuracy			0.84	806
macro avg	0.80	0.80	0.80	806
weighted avg	0.83	0.84	0.83	806

```

XGBoost (Embeddings) Accuracy: 0.836
XGBoost (Embeddings) Top-3 Accuracy: 0.932

```

Figure 5.7.13 Evaluation on Xgboost Classifier (Embedding)

The classification report shows an overall accuracy of 84%, with a high top 3 accuracy of 93.2%, indicating the model often ranks the correct class within its top three predictions. The weighted F1-score of 0.83 suggests balanced performance across classes, while the macro F1-score of 0.80 performs consistently across all disease types, treating each equally. It shows balanced classification, even for less common classes. High-performing classes include skin disease (F1-score: 0.94) and musculoskeletal problems (F1-score: 0.91), reflecting strong precision and recall. In contrast, the model struggles with heart, and lung diseases. Heart and lung problems also show low F1-scores (around 0.59–0.64), suggesting confusion with other classes.

Figure 5.7.14 show the test set and model stability evaluation. The model perform stable as the standard deviation is low and test set is displayed with prediction result.

```
XGBoost (Embeddings) Top-3 Predictions (Test Set):
Sample 1: my leg appear to be of different length...
True Label: musculoskeletal problem
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.967)), ('gynaecological disease', np.float64(0.011)), ('kidney problem disease', np.float64(0.006))]

Sample 2: I be unable to get pregnant despite be in good health...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.973)), ('kidney problem disease', np.float64(0.007)), ('internal disease', np.float64(0.004))]

Sample 3: some period last one day other up to eight day...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.973)), ('kidney problem disease', np.float64(0.007)), ('internal disease', np.float64(0.004))]

Sample 4: my skin feel very tight especially around my neck...
True Label: Skin problem disease
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.384)), ('Heart problem disease', np.float64(0.179)), ('internal disease', np.float64(0.116))]

Sample 5: I notice that my leg look puffy than usual...
True Label: kidney problem disease
Top-3 Predictions: [('kidney problem disease', np.float64(0.959)), ('gynaecological disease', np.float64(0.012)), ('musculoskeletal problem', np.float64(0.006))]

XGBoost (Embeddings) Stability (Cross-Validation on Original Data):
Mean Accuracy: 0.808
Standard Deviation: 0.019
Accuracy Range: [0.771, 0.846]
```

Figure 5.7.14 Evaluation on Test case and Stability

The Xgboost Classifier model with embeddings demonstrates in classifying disease-related symptoms. In the top 3 predictions for test samples, the correct label appeared within the top 3 suggestions in all cases and was ranked first in four out of five samples. This highlights the model's reliability in suggesting accurate diagnoses, even if the top prediction isn't always correct. Misclassifications occurred between musculoskeletal, heart, and internal diseases for sample 4, likely due to overlapping symptoms especially the sentence 'feel very tight especially around my neck' cause the misclassification to 'musculoskeletal problem', the possible reason can be due to the dataset is still not enough. Additionally, cross-validation using adasyn on training folds results show a mean accuracy of 80.8% with a low standard deviation of 0.019, indicating consistent performance across different data splits. The accuracy ranged from 77.1% to 84.6%, confirming the model's robustness and making it suitable for use in medical decision-support systems where multiple suggestions can guide further examination.

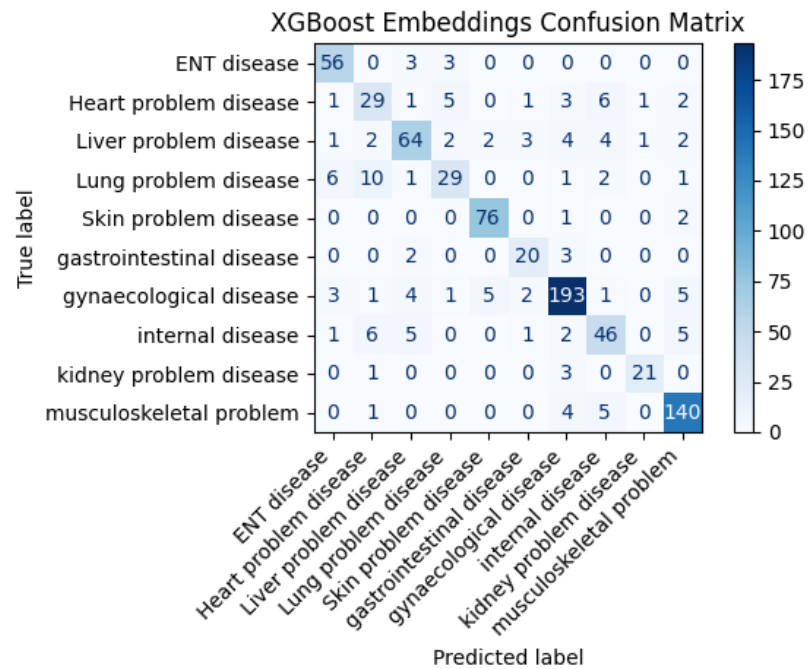


Figure 5.7.15 Confusion Matrix of XgboostClassifier

Figure 5.7.15 show the confusion matrix evaluation on the xgboost classifier model. The model performs quick well for skin problem disease (76/79), gastrointestinal disease (20/25), gynaecological disease (193/215), kidney problem disease (21/25) and musculoskeletal problem (140/150). The confusion between classes suggests overlapping features that the model struggles to separate. This can be due to the reason; similar symptoms exist between the classes cause misclassification.

XGBoost Classifier (TF-IDF)

Figure 5.7.16 show the classification report and top 3 accuracy evaluation. A Xgboost Classifier machine learning model using TF-IDF features was used to predict different types of diseases based on patient information. Tested on 806 samples, the model achieved 73% accuracy in correctly identifying unseen dataset.

```

=== XGBoost (TF-IDF) Evaluation ===
XGBoost (TF-IDF) Classification Report:

```

	precision	recall	f1-score	support
ENT disease	0.85	0.73	0.78	62
Heart problem disease	0.49	0.39	0.43	49
Liver problem disease	0.58	0.74	0.65	85
Lung problem disease	0.50	0.38	0.43	50
Skin problem disease	0.92	0.89	0.90	79
gastrointestinal disease	0.76	0.64	0.70	25
gynaecological disease	0.75	0.79	0.77	215
internal disease	0.56	0.68	0.62	66
kidney problem disease	0.82	0.56	0.67	25
musculoskeletal problem	0.85	0.83	0.84	150
accuracy			0.73	806
macro avg	0.71	0.66	0.68	806
weighted avg	0.73	0.73	0.73	806

```

XGBoost (TF-IDF) Accuracy: 0.727
XGBoost (TF-IDF) Top-3 Accuracy: 0.901

```

Figure 5.7.16 Evaluation on Xgboost Classifier (TF-IDF)

The classification report shows an overall accuracy of 73%, with a high top 3 accuracy of 90.1%, indicating the model often ranks the correct class within its top three predictions. A macro F1-score of 0.68 means the model's performance across all classes is moderate, with some classes likely performing poorly. The weighted F1-score of 0.73 shows slightly better overall performance, mainly because the model does better on common classes. This suggests the model is not well-balanced and struggles more with less frequent or harder-to-classify diseases. High-performing classes include skin disease (F1-score: 0.90). In contrast, the model struggles with heart, and lung diseases. Heart and lung problems also show low F1-scores (F1-score: 0.43), suggesting confusion with other classes.

Figure 5.7.17 show the test set and model stability evaluation. The model perform stable as the standard deviation is low and test set is displayed with prediction result.

```
XGBoost (TF-IDF) Top-3 Predictions (Test Set):
Sample 1: my leg appear to be of different length...
True Label: musculoskeletal problem
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.887)), ('gynaecological disease', np.float64(0.053)), ('Liver problem disease', np.float64(0.015))]

Sample 2: I be unable to get pregnant despite be in good health...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.957)), ('internal disease', np.float64(0.008)), ('Liver problem disease', np.float64(0.007))]

Sample 3: some period last one day other up to eight day...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.956)), ('internal disease', np.float64(0.008)), ('Heart problem disease', np.float64(0.007))]

Sample 4: my skin feel very tight especially around my neck...
True Label: Skin problem disease
Top-3 Predictions: [('skin problem disease', np.float64(0.661)), ('musculoskeletal problem', np.float64(0.249)), ('Liver problem disease', np.float64(0.024))]

Sample 5: I notice that my leg look puffy than usual...
True Label: kidney problem disease
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.885)), ('Liver problem disease', np.float64(0.043)), ('Heart problem disease', np.float64(0.027))]

XGBoost (TF-IDF) Stability (Cross-Validation on Original Data):
Mean Accuracy: 0.775
Standard Deviation: 0.016
Accuracy Range: [0.744, 0.807]
```

Figure 5.7.17 Evaluation on Test case and Stability

The Xgboost Classifier model with TF-IDF extraction demonstrates in classifying disease-related symptoms. In the top 3 predictions for test samples, the correct label appeared within the top 3 suggestions in all cases and was ranked first in four out of five samples. This highlights the model's reliability in suggesting accurate diagnoses, even if the top prediction isn't always correct. Misclassifications occurred between musculoskeletal, heart, and liver diseases for sample 5, likely due to overlapping symptoms especially the sentence 'I notice that my leg look puffy than usual' cause the misclassification to 'musculoskeletal problem', the possible reason can be due to the dataset is still not enough. Additionally, cross-validation using adasyn on training folds results show a mean accuracy of 77.5% with a low standard deviation of 0.016, indicating consistent performance across different data splits. The accuracy ranged from 74.4% to 80.7%.

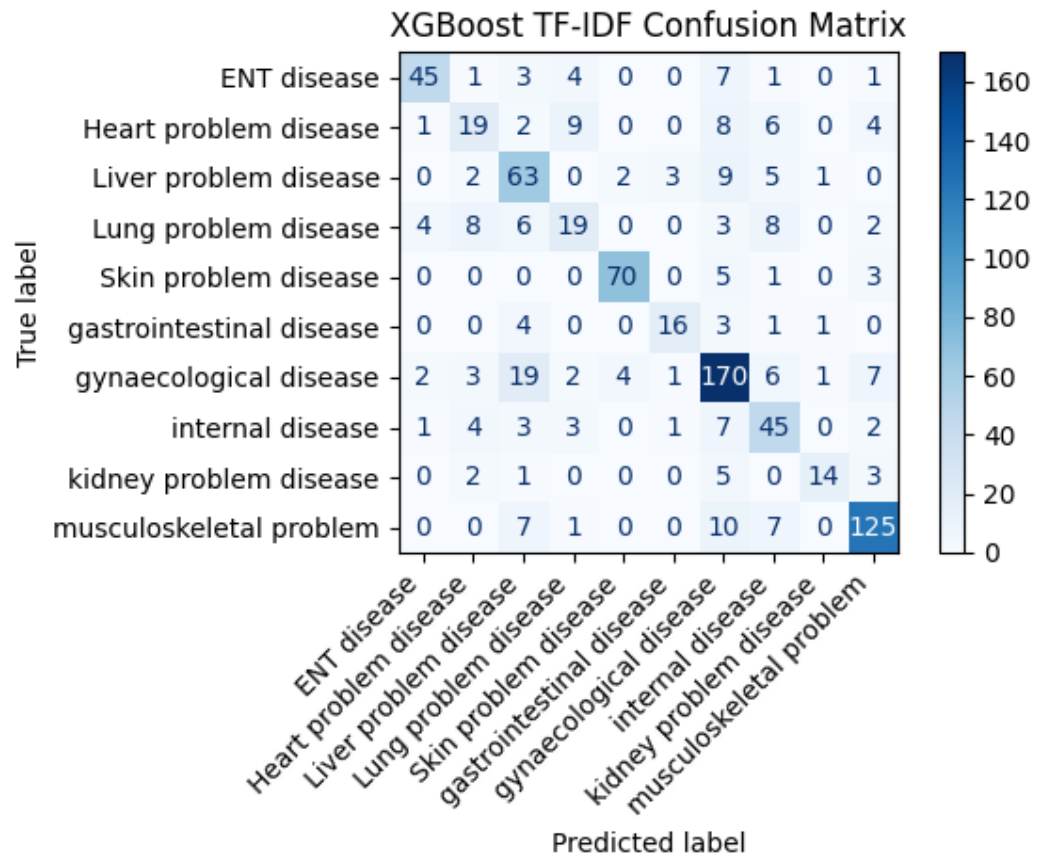


Figure 5.7.18 Confusion Matrix of Xgboost Classifier

Figure 5.7.18 show the confusion matrix evaluation on the xgboost classifier model. The model performs quick well for skin problem disease (70/79). The confusion between classes suggests overlapping features that the model struggles to separate. This can be due to the reason; similar symptoms exist between the classes cause misclassification.

Naïve Bayes (TF-IDF)

Figure 5.7.19 show the classification report and top 3 accuracy evaluation. A Naïve Bayes machine learning model using TF-IDF features was used to predict different types of diseases based on patient information. Tested on 806 samples, the model achieved 72% accuracy in correctly identifying unseen dataset.

```

=== Naive Bayes Evaluation ===
Naive Bayes Classification Report:
              precision    recall  f1-score   support

    ENT disease           0.69      0.77      0.73         62
    Heart problem disease  0.39      0.45      0.42         49
    Liver problem disease  0.72      0.66      0.69         85
    Lung problem disease   0.45      0.40      0.43         50
    Skin problem disease   0.74      1.00      0.85         79
    gastrointestinal disease 0.49      0.84      0.62         25
    gynaecological disease  0.95      0.69      0.80        215
    internal disease        0.59      0.58      0.58         66
    kidney problem disease  0.53      0.96      0.69         25
    musculoskeletal problem 0.88      0.83      0.86        150

              accuracy
    macro avg           0.64      0.72      0.67        806
    weighted avg        0.75      0.72      0.72        806

Naive Bayes Accuracy: 0.722
Naive Bayes Top-3 Accuracy: 0.924

```

Figure 5.7.19 Evaluation on Naïve Bayes (TF-IDF)

The classification report shows an overall accuracy of 72%, with a high top 3 accuracy of 92.4%, indicating the model often ranks the correct class within its top three predictions. A macro F1-score of 0.68 means the model's performance across all classes is moderate, with some classes likely performing poorly. The weighted F1-score of 0.73 shows slightly better overall performance, mainly because the model does better on common classes. This suggests the model is not well-balanced and struggles more with less frequent or harder-to-classify diseases. Heart and lung problems show low F1-scores (around 0.42 - 0.43), suggesting confusion with other classes. And suprisely, for the skin problem disease, recall is 1.0 which mean all test case is predicted correctly.

Figure 5.7.20 show the test set and model stability evaluation. The model perform stable as the standard deviation is low and test set is displayed with prediction result.

```
Naive Bayes Top-3 Predictions (Test Set):
Sample 1: my leg appear to be of different length...
True Label: musculoskeletal problem
Top-3 Predictions: [('musculoskeletal problem', np.float64(0.235)), ('kidney problem disease', np.float64(0.107)), ('gynaecological disease', np.float64(0.103))]

Sample 2: I be unable to get pregnant despite be in good health...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.284)), ('Liver problem disease', np.float64(0.211)), ('kidney problem disease', np.float64(0.109))]

Sample 3: some period last one day other up to eight day...
True Label: gynaecological disease
Top-3 Predictions: [('gynaecological disease', np.float64(0.308)), ('musculoskeletal problem', np.float64(0.089)), ('Heart problem disease', np.float64(0.083))]

Sample 4: my skin feel very tight especially around my neck...
True Label: Skin problem disease
Top-3 Predictions: [('Skin problem disease', np.float64(0.56)), ('musculoskeletal problem', np.float64(0.067)), ('ENT disease', np.float64(0.053))]

Sample 5: I notice that my leg look puffy than usual...
True Label: kidney problem disease
Top-3 Predictions: [('kidney problem disease', np.float64(0.151)), ('gynaecological disease', np.float64(0.149)), ('musculoskeletal problem', np.float64(0.116))]

Naive Bayes Stability (Cross-Validation on Original Data):
Mean Accuracy: 0.742
Standard Deviation: 0.023
Accuracy Range: [0.697, 0.788]
```

Figure 5.7.20 Evaluation on Test case and Stability

The Xgboost Classifier model with TF-IDF extraction demonstrates in classifying disease-related symptoms. In the top 3 predictions for test samples, the correct label appeared within the top 3 suggestions in all cases. This highlights the model's reliability in suggesting accurate diagnoses. Cross validation using adasyn on training folds results show a mean accuracy of 74.2% with a low standard deviation of 0.023, indicating consistent performance across different data splits. The accuracy ranged from 69.7% to 78.8%.

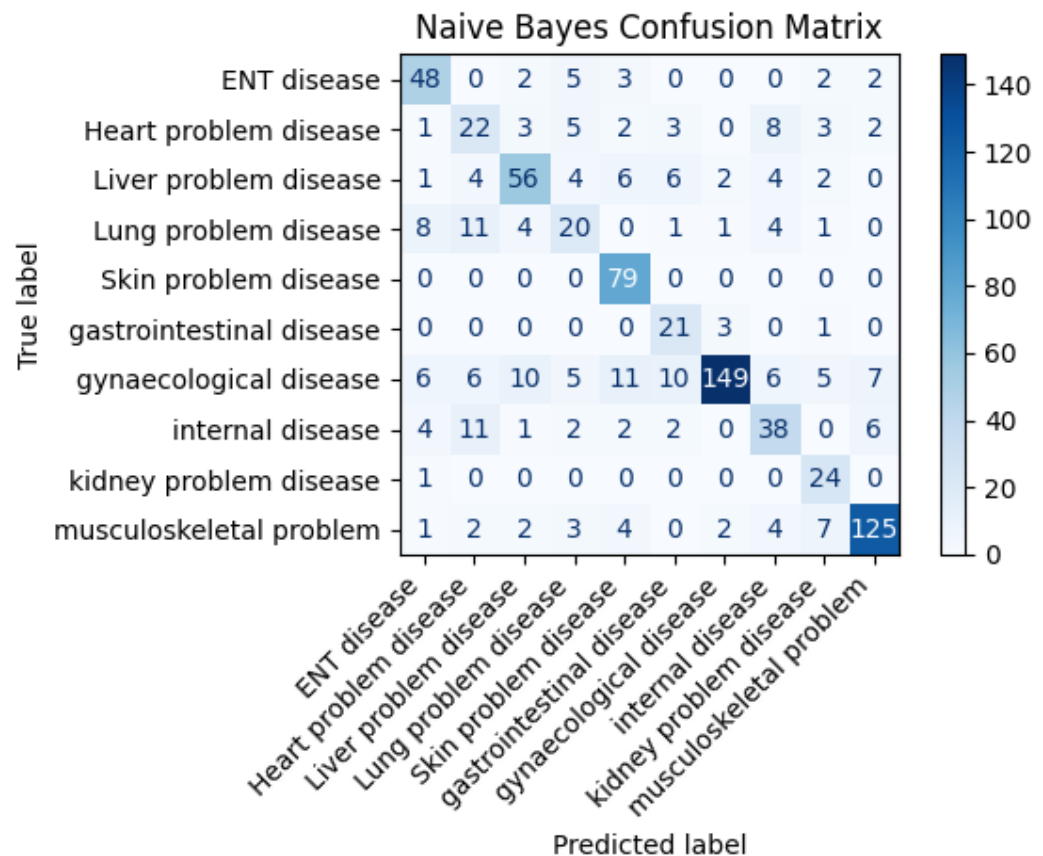


Figure 5.7.21 Confusion Matrix of Naïve Bayes (TF-IDF)

Figure 5.7.21 show the confusion matrix evaluation on the naïve bayes model. The model performs well for skin problem disease (79/79), kidney problem disease (24/25) and gastrointestinal disease (21/25). This insight are meaningful as even when the training set and test set is not that much, Naïve bayes have the ability to identify correct class for disease. However there still existed confusion between classes which overlapping features that the model struggles to separate. This can be due to the reason; similar symptoms exist between the classes cause misclassification.

5.8 Deployment

Figure 5.8.1 show the chatbot interface. The user input ‘sometime got ringing sound in my ear but not sound is around’. The chatbot reply with the help of RAG and transformer model. The disease ‘ENT disease’ that show to user is based on the machine learning model and display respective practitioners. User is enabled to click the link to navigate to appointment page to make appointment, and the link for related disease is based on the keyword extracted from the transformer model rag response.

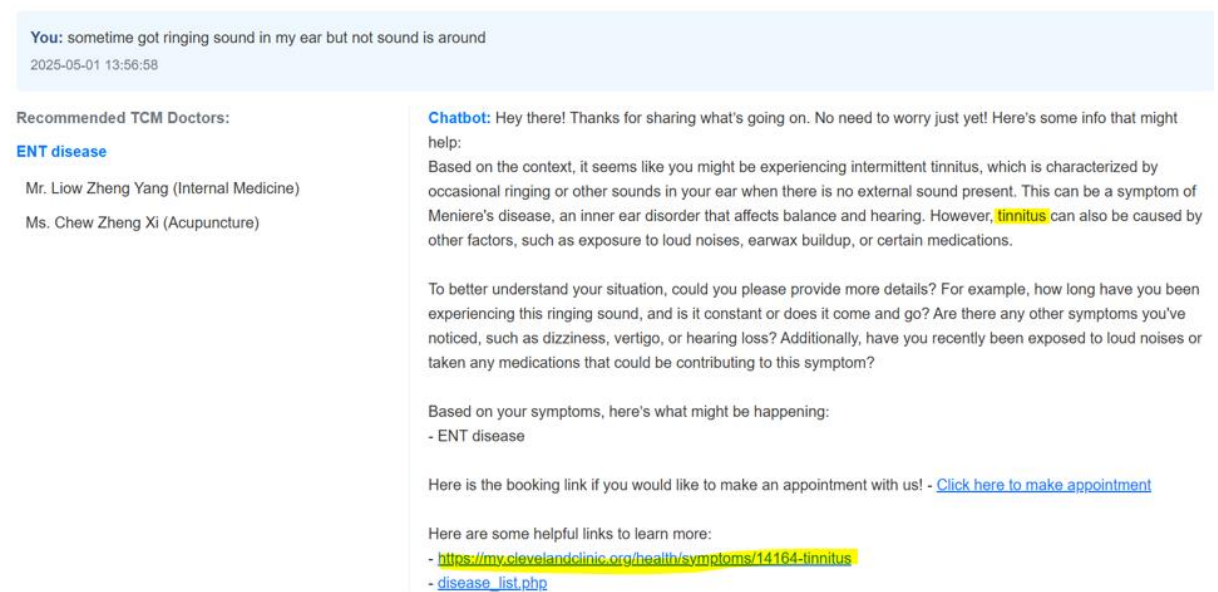


Figure 5.8.1 Show case for English Language Chatbot Interaction

CHAPTER 5

Figure 5.8.2 show the chatbot interface. The user input ‘why my eye ball is yellow’. The translation module will trigger to display Chinese response to user.



Figure 5.8.2 Show case for Chinese Language Chatbot Interaction

Figure 5.8.3 to Figure 5.8.14 are the function/page regarding hospital system. The chatbot can interaction with the page developed.

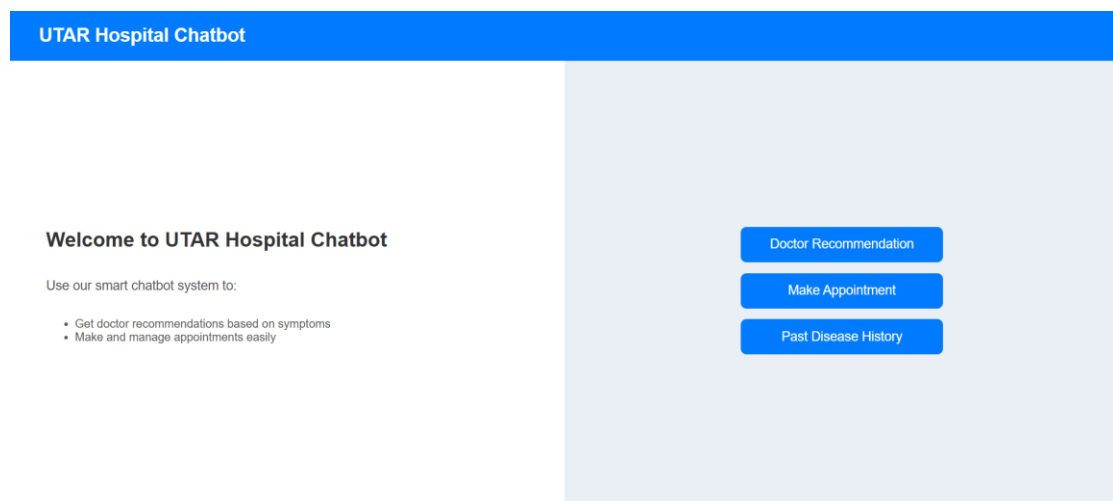


Figure 5.8.3 Home Page

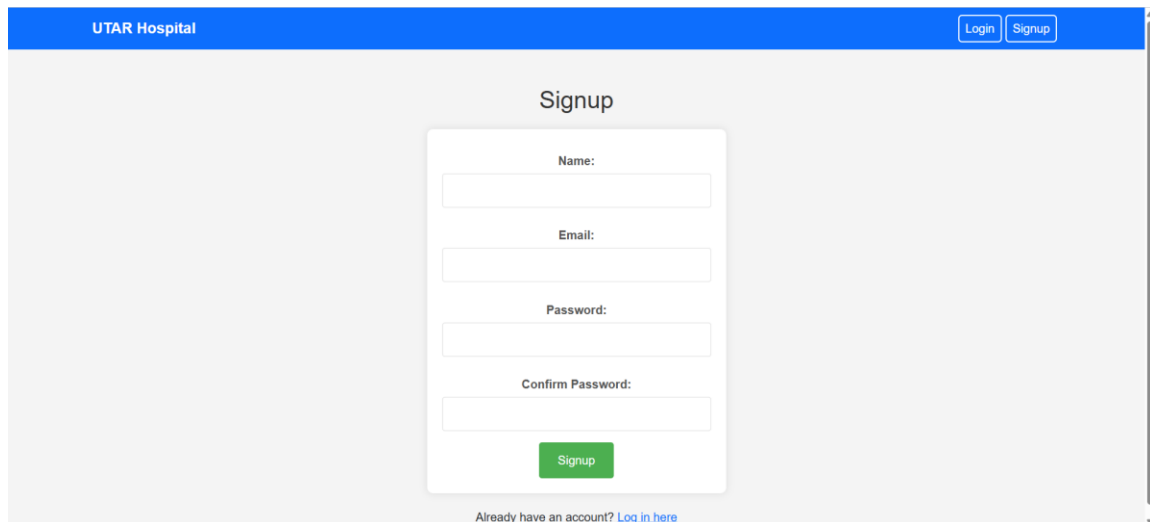


Figure 5.8.4 Sign Up Page

User could sign up with the name, email and password. If the confirmation password is not correct as password. Error messages will prompt.

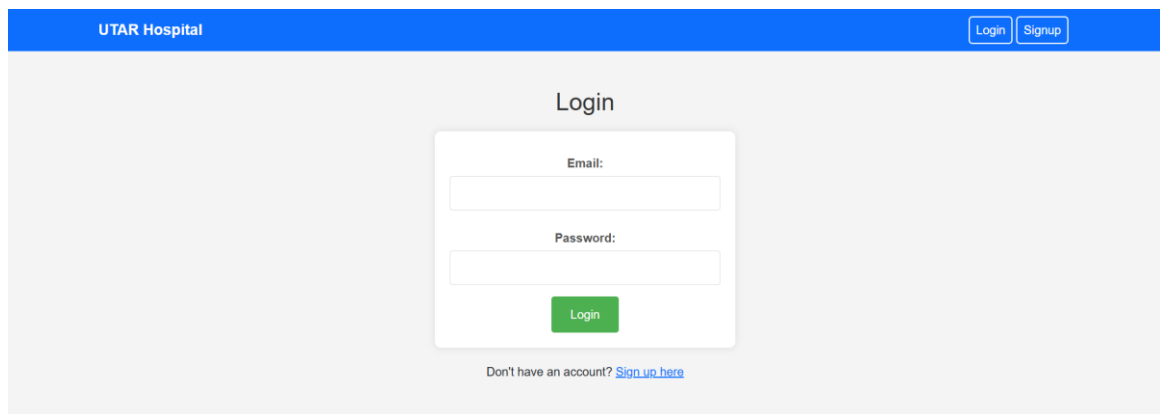


Figure 5.8.5 Login Page

After sign up, user allows to login. If login details not existed in database, the users are not allowed to login, error message will prompt.

Login here'." data-bbox="222 83 828 213"/>

Figure 5.8.6 Forgot Password Page

If user forgot password, they could enter their email, and a reset password email will send to the user for password reset purpose.

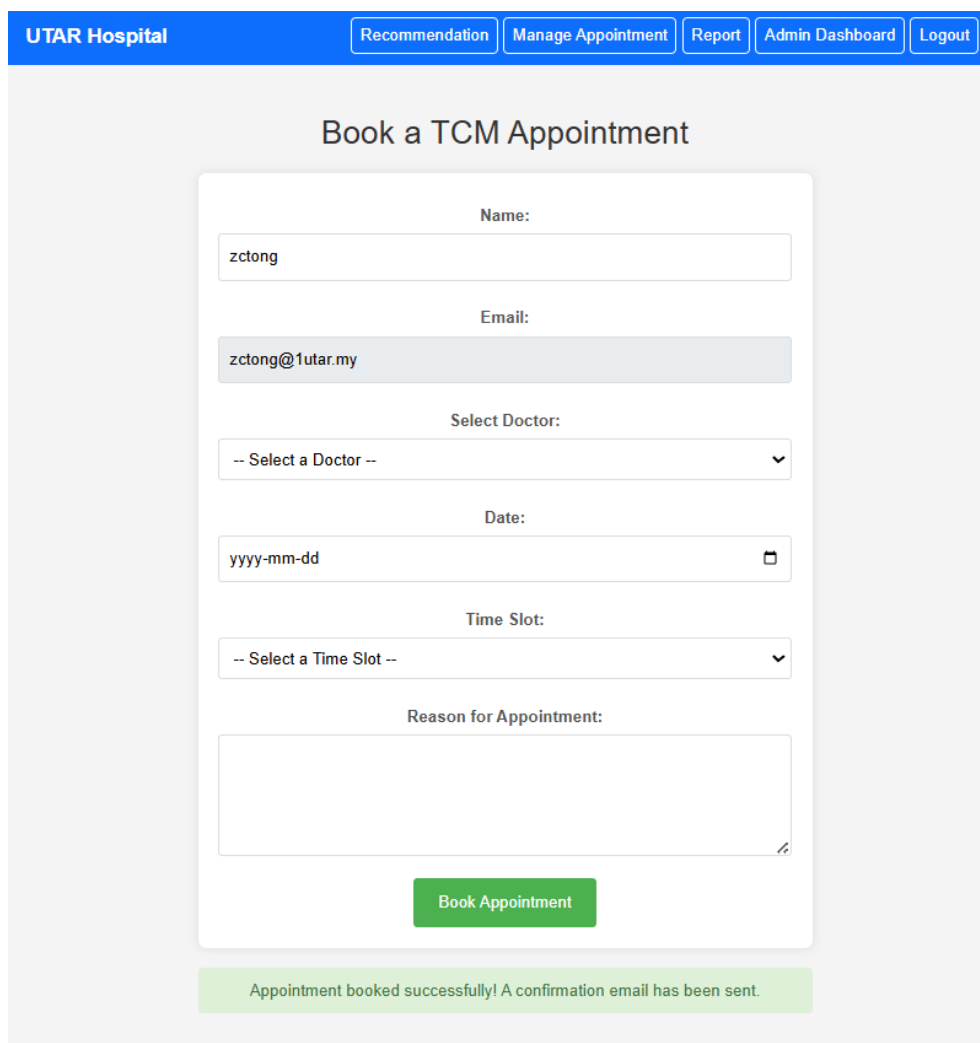
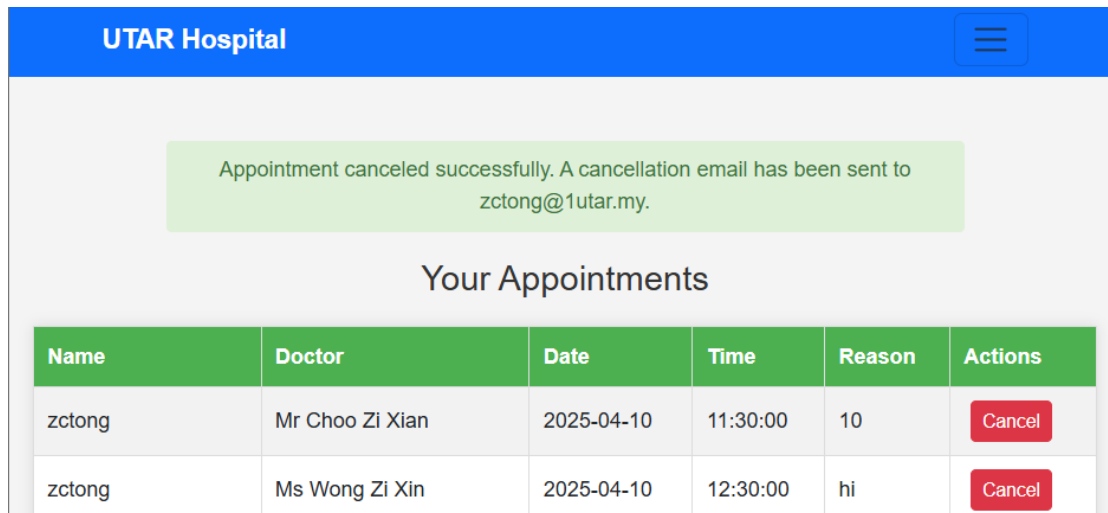


Figure 5.8.7 Book appointment page

User could make appointments after they login. The name and email will auto fill up by the system, user can select doctor, date, time and state reason to make appointment, and an email notification will send to user. Reminder will send to the user, if the current date is the booked appointment date. In other hands, if the timeslot is not available, non-successfully booked message will prompt to user.



Name	Doctor	Date	Time	Reason	Actions
zctong	Mr Choo Zi Xian	2025-04-10	11:30:00	10	Cancel
zctong	Ms Wong Zi Xin	2025-04-10	12:30:00	hi	Cancel

Figure 5.8.8 View/Cancel appointment page

User could view and cancel their appointment if they login. If they cancel their appointment, the cancel email will send to user as well.

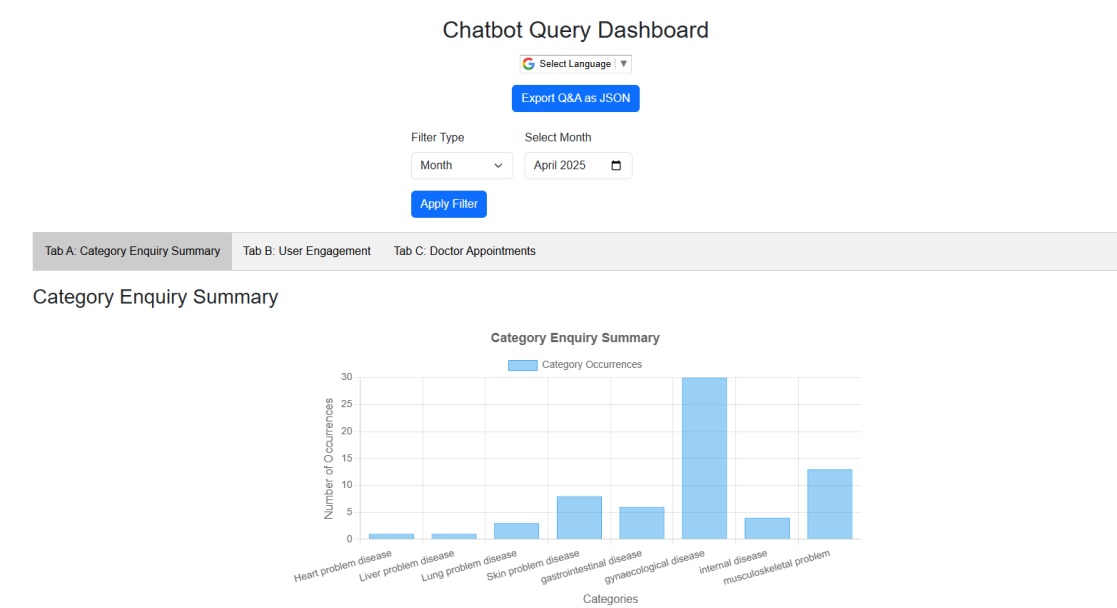


Figure 5.8.9 Dashboard on disease predicted by category

The admin can view the bar graph for the disease category, see the trend of disease. Besides that, admin can also filter based on year, month and date to display respective graph based on selected slot. The Export JSON file button can reduce the workload of finding dataset, as RAG method can use the dataset to generate output. So, using user query and monitoring on the answer and category by professional can make the chatbot generate more user accurate result to user.

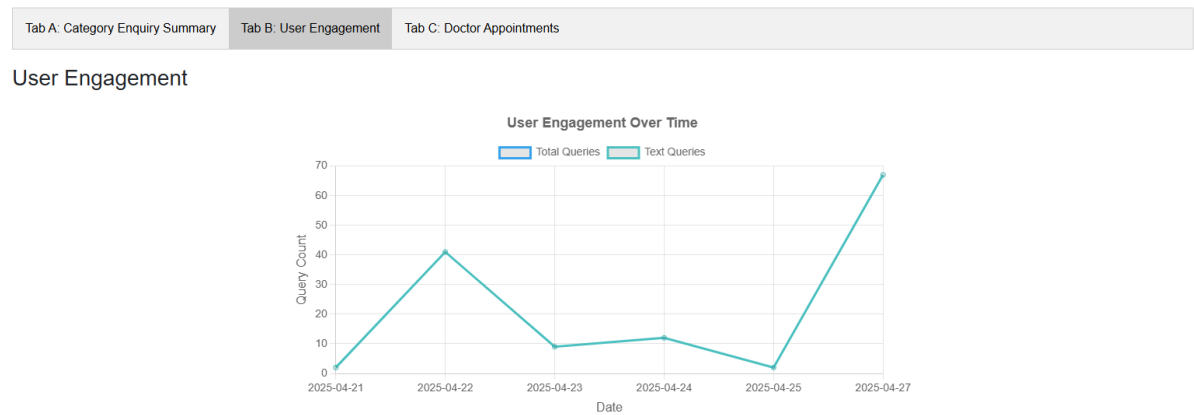


Figure 5.8.10 Dashboard on user engagement with chatbot

The admin can view the line chart on the user engagement with chatbot.

CHAPTER 5

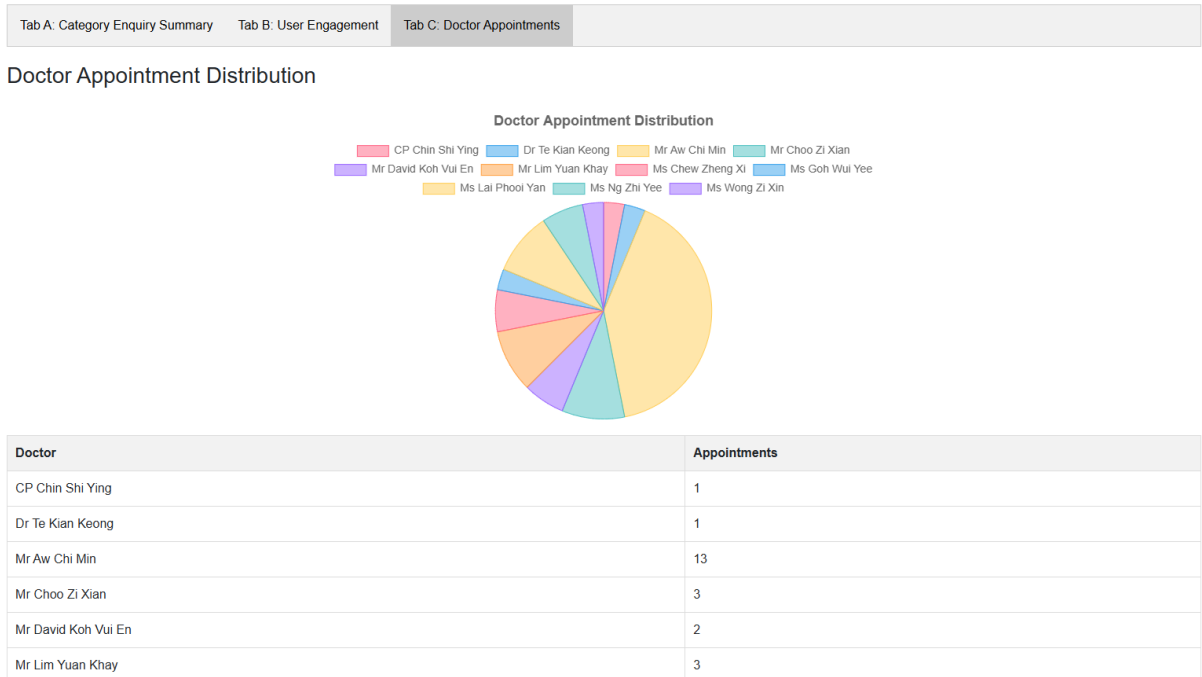


Figure 5.8.11 Dashboard on doctor appointments distribution

The dashboard allows the admin to view the number of appointments made by users with each respective doctor.

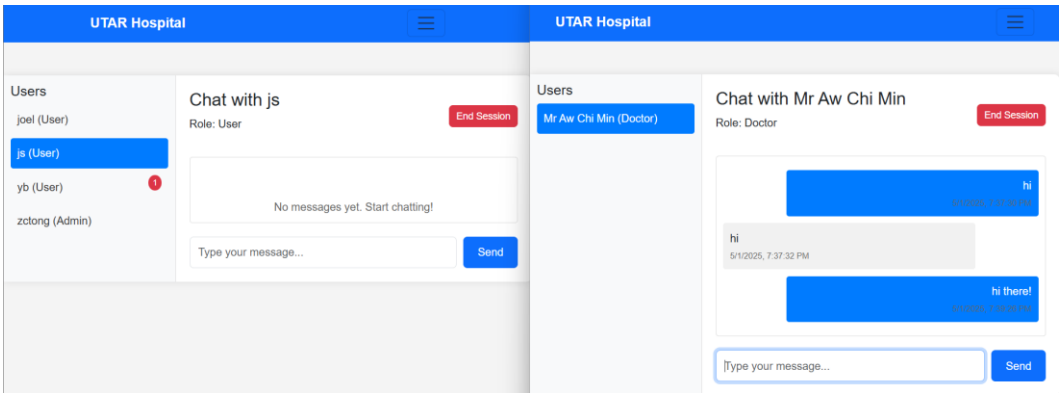


Figure 5.8.12 User-Doctor chatting page

This feature enables easy communication between the user and doctor for convenient follow-up and further access.

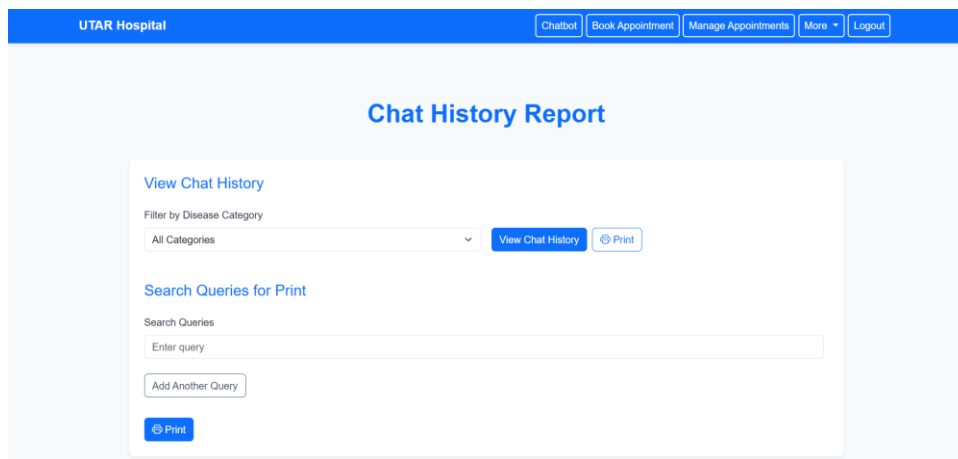


Figure 5.8.13 Print Report Page

The chat history report allows users to print a record of all previous interactions with the chatbot. Users can also search for specific queries they made to include in the report. This makes it easier for the admission team to find suitable practitioners for the user during physical admission.

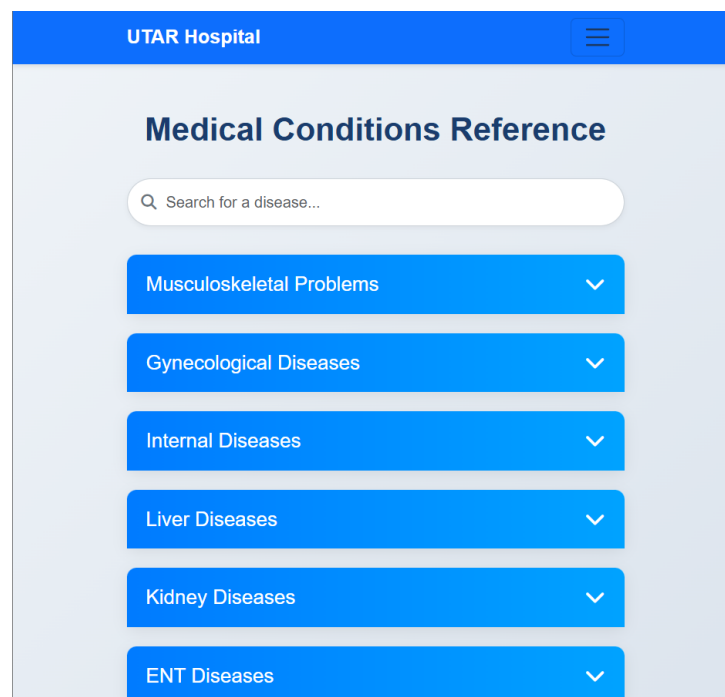


Figure 5.8.14 Disease List Page

User can get more information about respective disease, the link is provided to user for the disease list page

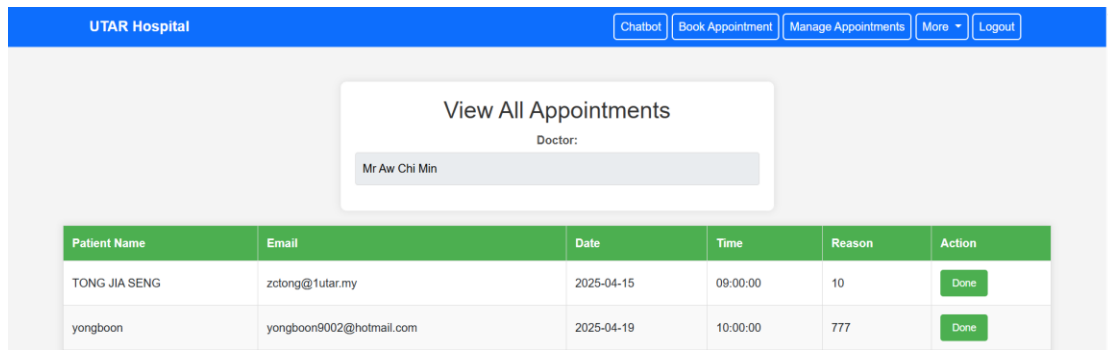


Figure 5.8.15 View Appointments Booked by user (Doctor Side)

The doctor enables to see who have booked with him/her. Once session done, doctor can click on the done button to remove the record.

CHAPTER 6: System Evaluation And Discussion

6.1 System Testing and Performance Metrics

6.1.1 Classification model evaluation

For the classification model evaluation, metrics such as accuracy, top-3 accuracy, classification report (precision, recall, f1-score), cross validation score and confusion matrix are used. Table 6.1.1.1 show summary on the accuracy, cross validation score (5 folds, using training dataset - unseen data) and top-3 accuracy (top 3 predicted classes containing the true classes). More detail evaluation explanations can refer to **Chapter 5 – Section 5.7 Evaluation**.

Table 6.1.1.1 Summary on Classification Model Performance

Classification Model	Accuracy	Cross Validation Score	Top-3 accuracy
Random Forest Classifier (Embedding)	82%	81.2%	96.3%
Random Forest Classifier (TF-IDF)	80%	79%	94.3%
Logistic Regression (Embedding)	78%	78.9%	93.7%
Logistic Regression (TF-IDF)	80%	80%	92.7%
Xgboost Classifier (Embedding)	84%	80.8%	93.2%
Xgboost Classifier (TF-IDF)	73%	77.5%	90.1%
Naïve Bayes (TF-IDF)	72%	74.2%	92.4 %

From all the trained models, only three classification models (which is almost 80% or > 80%) are used for making prediction [Random Forest Classifier (Embedding), Logistic Regression (TF-IDF), Xgboost Classifier (Embedding)] and combine with RAG. The insight get from the model training are TF-IDF work better in Logistic Regression Classification, while the Embedding work better in Random Forest and Xgboost Classification.

6.1.2 Chatbot Response Testing

The testing is to evaluate the chatbot responses whether are relevant to retrieve text, and the response show should include the disease diagnosed, doctor recommendation and respective disease link. Besides, also testing on whether the chatbot response could be in bilingual language (English/Chinese). The testing results shows the information displayed are relevant to the retrieve text, and others information also display correctly as well. Therefore, the response from chatbot feature meet the expectation.

Test Case 1:

You: my discharge is yellowish why
2025-05-03 16:07:19

Recommended TCM Doctors:

[gynaecological disease](#)

Ms. Goh Wui Yee (Internal Medicine)

Ms. Lai Phooi Yan (Acupuncture)

Chatbot: Hey there! Thanks for sharing what's going on. No need to worry just yet! Here's some info that might help:

A yellowish discharge may indicate abnormal leucorrhoea, which could be due to an infection. It's important to note that a change in the color or consistency of your discharge can be a sign that something is off and should be evaluated by a healthcare provider. Both bacterial vaginosis and trichomoniasis, which are forms of abnormal leucorrhoea, can cause thick yellow discharge along with discomfort. It's best to consult with a healthcare provider to determine the cause of your symptoms and receive appropriate treatment.

Based on your symptoms, here's what might be happening:

- gynaecological disease

Here is the booking link if you would like to make an appointment with us! - [Click here to make appointment](#)

Here are some helpful links to learn more:

- <https://my.clevelandclinic.org/health/symptoms/4719-vaginal-discharge>
- [disease_list.php](#)

Figure 6.1.2.1 Chatbot Test Case 1

Table 6.1.2.1 show the top 5 retrieve texts from vector storage that feeded to the llama transformer model.

Table 6.1.2.1 Top 5 retrieve texts

"text": "My vaginal discharge has turned yellowish. A yellow or green discharge may be a sign of abnormal leucorrhoea, often due to an infection."	"text": "My discharge has changed to a yellow color. Yellow discharge may be an indication of abnormal leucorrhoea, often due to an infection."	"text": "I have yellow, thick discharge and feel discomfort. Thick yellow discharge may indicate bacterial vaginosis or trichomoniasis, both forms of abnormal leucorrhoea. "	"text": "I have yellow, thick vaginal discharge that causes discomfort. Thick yellow discharge can be a sign of infection, such as bacterial vaginosis or trichomoniasis, both of which are types of abnormal leucorrhoea."	"text": "My vaginal discharge is pale yellow with a noticeable odor and I feel discomfort while sitting. A pale yellow discharge with odor and discomfort may indicate abnormal leucorrhoea due to an infection. "
--	---	---	---	--

Figure 6.1.2.2 show disease prediction based on the classification model. The top predicted disease will be get from the three models and combined with RAG response with keyword detection. Based on the tcm doctors list, output doctor for user.

```
Setting `pad_token_id` to `eos_token_id`:128001 for open-end generation.
INFO:werkzeug:127.0.0.1 - - [03/May/2025 08:07:18] "POST /chat HTTP/1.1" 200 -
Setting `pad_token_id` to `eos_token_id`:128001 for open-end generation.
Model Predicted Categories with Probabilities:
Random Forest (Embeddings): gynaecological disease (62.53%), Liver problem disease (9.14%), musculoskeletal problem (8.01%)
Logistic Regression (TF-IDF): gynaecological disease (62.58%), Liver problem disease (20.66%), Lung problem disease (10.71%)
XGBoost (Embeddings): gynaecological disease (63.94%), musculoskeletal problem (22.92%), Liver problem disease (3.70%)
```

Figure 6.1.2.2 Model Prediction Result

Test Case 2:

You: 我的眼睛很干呀怎么办

2025-05-03 15:51:52

Recommended TCM Doctors:

ENT disease

Mr. Liow Zheng Yang (Internal Medicine)

Ms. Chew Zheng Xi (Acupuncture)

Chatbot: 嘿！感谢您分享发生的事情。暂时不用担心！这里有一些可能会有所帮助的信息：
基于上下文和用户查询，可能的响应是：

如果您经历了持久的眼睛干燥，频繁眨眼或过度撕裂，尽管感觉干燥，则可能正在处理干眼症。为了减轻症状，请尝试以下步骤：

1. 通过喝大量水来保持水分。脱水会恶化干眼。
2. 用人造眼泪润滑您的眼睛。可以提供非处方选项，或咨询您的医生处方选项。
3. 避免刺激性烟雾，风或化学物质。如有必要，请穿防护眼镜。
4. 考虑使用加湿器，尤其是在干燥或低湿度环境中。
5. 如果您的症状持续或恶化，请咨询眼保健专业人员以进行进一步评估和治疗。

请记住，如果您遇到其他任何异常症状或视力变化，请随时寻求医疗护理。

根据您的症状，这可能是发生的事情：

- ENT疾病

如果您想与我们预约，这是预订链接！ - [单击此处进行约会](#)

以下是一些有用的链接以了解更多信息：

- <https://my.clevelandclinic.org/health/diseases/24479-dry-eye>
- [disease_list.php](#)

Figure 6.1.2.3 Chatbot Test Case 2

Table 6.1.2.2 show the top 5 retrieve texts from vector storage that feeded to the llama transformer model.

Table 6.1.2.2 Top 5 retrieve texts

"text": "My eyes water too much, even though they feel dry. Excessive tearing despite feeling dry may indicate dry eye."	"text": "My eyes feel dry and irritated all the time. Persistent eye dryness could be a symptom of dry eye."	"text": "I feel like I need to blink a lot to see clearly. Frequent blinking to clear vision can be a sign of dry eye."	"text": "My eyes feel worse in dry or low-humidity environments. Low humidity levels can trigger or worsen dry eye symptoms."	"text": "I feel like my eyes are always watery, but they still feel dry. Excess tearing with a dry sensation may be related to dry eye."
--	--	---	---	--

CHAPTER 6

Figure 6.1.2.4 show disease prediction based on the classification model. The top predicted disease will be get from the three models and combined with RAG response with keyword detection. Based on the tcm doctors list, output doctor for user.

```
INFO:werkzeug:127.0.0.1 - - [03/May/2025 07:51:06] "POST /chat HTTP/1.1" 200 -
Segmented Chinese Input for Translation: 我的眼睛很干呀 怎么办
Translated Query: My eyes are very dry. What should I do
Model Predicted Categories with Probabilities:
Random Forest (Embeddings): ENT disease (81.30%), Skin problem disease (7.82%), gynaecological disease (3.57%)
Logistic Regression (TF-IDF): ENT disease (86.65%), gynaecological disease (12.22%), Skin problem disease (0.99%)
XGBoost (Embeddings): ENT disease (53.91%), Skin problem disease (39.87%), gynaecological disease (2.53%)
```

Figure 6.1.2.4 Model Prediction Result

Test Case 3:

You: and my shoulde look uneven and i also has hunch back

2025-05-03 16:05:27

Recommended TCM Doctors:

musculoskeletal problem

Mr. Aw Chi Min - Upper Limb Disorder (Tuina)

Mr. Lim Yuan Khay - Lower Limb Disorder (Acupuncture) & [TCM Neurology]

Ms. Lai Phooi Yan - Upper Limb Disorder (Acupuncture)

Mr. Choo Zi Xian - Bone, Joint & Muscle System Disorder (Internal Medicine) & [Oncological Issues]

Mr. David Koh Vui En - Lower Limb Disorder

Chatbot: Hey there! Thanks for sharing what's going on. No need to worry just yet! Here's some info that might help:

Based on the symptoms you've described, it seems like you might be experiencing scoliosis, which is a curvature of the spine that can cause one shoulder to appear higher than the other, as well as a lopsided appearance when wearing fitted clothes. The hunching appearance and uneven waist you've mentioned can also be indicative of scoliosis. To confirm the diagnosis and determine the appropriate treatment, I recommend consulting an orthopedic specialist or a primary care physician for a thorough evaluation. They may perform X-rays or other imaging tests to assess the degree and location of the curvature. Depending on the severity, they may recommend physical therapy, bracing, or surgery to correct the curvature and alleviate any associated pain or discomfort.

Based on your symptoms, here's what might be happening:

- musculoskeletal problem

Here is the booking link if you would like to make an appointment with us! - [Click here to make appointment](#)

Here are some helpful links to learn more:

- <https://my.clevelandclinic.org/health/diseases/15837-scoliosis>
- [disease_list.php](#)

Figure 6.1.2.5 Chatbot Test Case 3

Table 6.1.2.3 show the top 5 retrieve texts from vector storage that feeded to the llama transformer model.

Table 6.1.2.3 Top 5 retrieve texts

"text": "My back looks lopsided when I wear fitted clothes. Lopsided back appearance could	"text": "People tell me I look like I'm hunching over. A hunching appearance may	"text": "My waist looks uneven, and one side appears higher than the other. An uneven waist	"text": "My clothes do not seem to fit evenly on both sides. Uneven clothing	"text": "My upper body seems to lean slightly forward. Forward leaning may suggest scoliosis-
--	--	---	--	---

indicate scoliosis."	be due to kyphosis."	may suggest scoliosis."	fit might indicate scoliosis."	related posture issues."
-------------------------	-------------------------	----------------------------	-----------------------------------	-----------------------------

Figure 6.1.2.6 show disease prediction based on the classification model. The top predicted disease will be get from the three models and combined with RAG response with keyword detection. Based on the tcm doctors list, output doctor for user.

```
Setting `pad_token_id` to `eos_token_id`:128001 for open-end generation.
INFO:werkzeug:127.0.0.1 - - [03/May/2025 08:04:31] "POST /chat HTTP/1.1" 200 -
Model Predicted Categories with Probabilities:
Random Forest (Embeddings): musculoskeletal problem (85.16%), gynaecological disease (7.90%), Heart problem disease (1.95%)
Logistic Regression (TF-IDF): musculoskeletal problem (91.99%), gynaecological disease (5.09%), gastrointestinal disease (0.99%)
XGBoost (Embeddings): musculoskeletal problem (93.73%), gynaecological disease (2.21%), Heart problem disease (1.48%)
```

Figure 6.1.2.6 Model Prediction Result

6.1.3 Testing on functionality module

Table 6.1.3.1 Test Case on Admin Dashboard Module

Admin Dashboard Module	Test Scenario	Expected Output	Actual Output	Evaluation
Filter function	Filter based on year, month, date	Dashboard Visualization for all tabs change depend on selected filter	Dashboard Visualization for all tabs change depend on selected filter	Pass
Extract JSON file	Clicked extract JSON file button	Extract data from the database and form JSON file content in format question, answer and category	Extract data from the database and form JSON file content in format question, answer and category	Pass

Table 6.1.3.2 Test Case on Translation Module

Translation Module	Test Scenario	Expected Output	Actual Output	Evaluation
	User input english language input	System output english response	System output english response	Pass
	User input chinese language input	System output chinese response	System output chinese response	Pass

Table 6.1.3.3 Test Case on Appointment Module

Appointment Module	Test Scenario	Expected Output	Actual Output	Evaluation
Book Appointment	User fill in details and clicked send button	Success booking, and send email with booked details to user	Success booking, and send email with booked details to user	Pass
	User fill in details which the slot booked by others and clicked send button	Failed to book	Failed to book	Pass
	User book appointment for next day	Success booking, and send email with booked details to user. At the next day, send a reminder email to the user	Success booking, and send email with booked details to user. At the next day, send a reminder email to the user	Pass
Cancel Appointment	User click cancel button to	Success cancel appointment,	Success cancel appointment,	Pass

	cancel appointment	and send cancel email with booked details to user	and send cancel email with booked details to user	
--	-----------------------	--	--	--

Table 6.1.3.4 Test Case on User-Doctor Chat Module

User-Doctor Chat Module	Test Scenario	Expected Output	Actual Output	Evaluation
Notification	User no clicked the user tab	Raise notification	Raise notification	Pass
	User clicked the user tab	No Raise notification	No Raise notification	Pass
	User A chat with User B, User A no clicked User B tab	Raise notification and notification number increase	Raise notification and notification number increase	Pass
Chat	User A chat with User B	Message sending between users within 5 second	Message sending between users within 5 second	Pass
End Session	User clicked end session button	All message clean from chat	All message clean from chat	Pass

6.2 Project Challenges

1. Google Colab Limitation with Local Database

Google Colab cannot directly connect to local databases such as MySQL from XAMPP. This becomes a barrier to the interaction between the Flask API hosted in Colab and the local PHP web application. To get around this, Supabase as a cloud service is used to store and retrieve the latest Ngrok URL so that the local PHP system can consume the Flask API through a public URL.

2. Hardware Limitations

Running and fine-tuning large language models, including transformer-based models or Retrieval-Augmented Generation (RAG), requires high-performance GPU resources. Local machines lack the computational power to handle the tasks efficiently. Therefore, the model is executed in Google Colab, and the local application sends requests through a publicly exposed API using Ngrok.

3. Scalability Issues

Handling multiple user queries is challenging with limited hardware, especially when relying on a single local GPU. While cloud platforms like AWS or Azure offer scalable resources, they often come with additional costs, which may not be suitable for all projects.

4. Quality of Retrieved Information

RAG models depend heavily on document retrieval. Sometimes, irrelevant or contextually incorrect documents are fetched, leading to inaccurate or misleading responses from the model. This issue requires additional tuning during development, such as implementing node re-ranking and filtering techniques to improve the quality and relevance of the retrieved content.

6.3 Objectives Evaluation

The project successfully achieved its four main objectives, integrating machine learning, multilingual support, and healthcare service features into a comprehensive chatbot with system for Universiti Tunku Abdul Rahman (UTAR) Hospital's Traditional and Complementary Medicine (T&CM) centre.

Objective 1: To develop a bilingual (English and Chinese) chatbot for UTAR Hospital that utilizes machine learning to predict potential diseases based on symptom described by users.

The project developed a bilingual (English and Chinese) chatbot for UTAR Hospital that uses a combination of machine learning models to predict potential diseases based on user-described symptoms, achieving an approximate accuracy of 80%. The models

work together to identify the most likely disease for the user. A translator API and a Chinese segmentation library were integrated to handle language processing challenges. This enhances the chatbot's accessibility in Malaysia, where both languages are commonly spoken.

Additionally, the chatbot integrates Retrieval-Augmented Generation (RAG) approach with the Llama Transformer model to refine disease predictions. This combination improves accuracy and generates more human-like, supportive responses, providing better overall user experience.

Objective 2: To implement doctor recommendations from UTAR Hospital's T&CM Centre based on the disease category predicted.

The project successfully integrated a doctor recommendation feature into the chatbot. The disease category is predicted using machine learning techniques, combined with RAG (Retrieve and Generate) response keyword extraction. Based on the predicted disease category, the system automatically recommends the most appropriate Traditional Chinese Medicine (TCM) practitioner at UTAR Hospital. This personalized approach simplifies the process for patients to find the right specialist, eliminating the need for manual searching. It also reduces waiting times and enhances efficiency by quickly matching patients with the right doctor based on their condition. Therefore, this objective is considered achieved.

Objective 3: To design an online appointment scheduling system for UTAR Hospital T&CM Centre that allows patients to book and cancel appointments, with email notifications for confirmations, cancellations, and reminders.

The project introduced appointment scheduling system, allowing patients to book and cancel appointments. Once an appointment is booked, an email notification is sent with the booking details, while reminders and cancellation notifications are also sent as needed. This feature enhances user convenience by automating appointment management and reducing the administrative workload on hospital staff. Therefore, this objective is considered achieved.

Objective 4: To develop a dashboard that allows administrators to monitor user engagement with the chatbot, visualize the diseases predicted by the chatbot using a bar chart that displays their frequency and present a pie chart showing the number of appointments made by users for each doctor.

The dashboard features a line graph to track user engagement with the chatbot over time, a bar chart to visualize the frequency of diseases predicted by the chatbot, and a pie chart that illustrates the number of appointments made for each doctor. With these components, the objective has been successfully achieved.

CHAPTER 7: Conclusion and Recommendation

7.1 Conclusion

This project solves several problems in the healthcare sector, such as language barriers, misdiagnosis because of unclear symptom descriptions, and inefficient appointment scheduling. Through the development of a bilingual chatbot in English and Chinese, the system bridges communication by providing preliminary disease predictions based on patient-described symptoms. With the model achieving an 80% accuracy rate, the goal of efficient disease prediction has been met. The system also improves the referral process by recommending appropriate doctors from UTAR Hospital's T&CM Centre based on predicted disease categories. An online appointment system with automated email reminders allows for a convenient scheduling of appointments, and an admin dashboard gives insights into the rates of engagement, trends in diseases through bar charts, and appointment rates per doctor through a pie chart. and present pragmatic answers to the communications and business challenges prevalent within the healthcare field.

7.2 Recommendation

To further enhance the UTAR Hospital chatbot system, it is recommended to fine-tune a transformer-based model such as Llama, which can significantly improve the chatbot's ability to understand complex medical inquiries and provide more accurate and personalized responses. Additionally, incorporating a voice chat feature would make the system more accessible to a wider range of users, including those who may have difficulties typing or prefer speaking. Implementing a report analysis system would allow the chatbot to interpret and provide feedback based on users' submitted reports, offering a more comprehensive and supportive healthcare experience. Finally, continuously expanding the dataset with the assistance of healthcare professionals and domain experts can ensure that the chatbot remains up-to-date with the latest medical knowledge and provides reliable and high-quality support to users.

REFERENCES

- [1] Open Medscience, "The Impact of Language Barriers on Elderly Healthcare," *Open MedScience*, Jan. 28, 2025. <https://openmedscience.com/the-impact-of-language-barriers-on-elderly-healthcare/>
- [2] "Health Ministry Probing Alleged Misdiagnosis Leading to Patient's Death." *The Star*, 9 Nov. 2023, www.thestar.com.my/news/nation/2023/12/09/health-ministry-probing-alleged-misdiagnosis-leading-to-patient039s-death.
- [3] Plescia, Marissa. "Survey: Scheduling Troubles Force 61% of Patients to Skip Medical Care." *MedCity News*, 16 Nov. 2022, medcitynews.com/2022/11/survey-scheduling-troubles-force-61-of-patients-to-skip-medical-care/. Accessed 6 May 2025.
- [4] "3 Main Patient Appointment Scheduling Problems and How to Solve Them." *Www.medesk.net*, 15 May 2024, www.medesk.net/en/blog/scheduling-issues-in-healthcare/.
- [5] Mount Sinai, "Mount Sinai Health System," *Mount Sinai Health System*, 2019. <https://www.mountsinai.org/> (accessed Sep. 02, 2024).
- [6] "Best Multispeciality Hospitals in Ahmedabad - Zydus Hospital," *zydushospitals.com*. <https://zydushospitals.com/> (accessed Sep. 02, 2024).
- [7] UCLA Health, "UCLA Health: High Quality Health Care Services, Top Health Care Specialists, Best Doctors - UCLA, Los Angeles, CA," *Uclahealth.org*, 2010. <https://www.uclahealth.org/> (accessed Sep. 02, 2024).
- [8] Gonzalez-Barrera, Ana, et al. "Language Barriers in Health Care: Findings from the KFF Survey on Racism, Discrimination, and Health." *KFF*, 16 May 2024, www.kff.org/racial-equity-and-health-policy/poll-finding/language-barriers-in-health-care-findings-from-the-kff-survey-on-racism-discrimination-and-health/.
- [9] M. K. Ogirala, R. Tallapaneni, S. M. Chalamcharla and A. Chinta, "A Medical Diagnosis and Treatment Recommendation Chatbot using MLP," *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India, 2023, pp. 495-500, doi: 10.1109/ICAAIC56838.2023.10141211

REFERENCES

- [10] P. Kandpal, K. Jasnani, R. Raut and S. Bhorge, "Contextual Chatbot for Healthcare Purposes (using Deep Learning)," *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, London, UK, 2020, pp. 625-634, doi: 10.1109/WorldS450073.2020.9210351.
- [11] S. P. Reddy Karri and B. Santhosh Kumar, "Deep Learning Techniques for Implementation of Chatbots," *2020 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2020, pp. 1-5, doi: 10.1109/ICCCI48352.2020.9104143.
- [12] L. Athota, V. K. Shukla, N. Pandey and A. Rana, "Chatbot for Healthcare System Using Artificial Intelligence," *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, 2020, pp. 619-622, doi: 10.1109/ICRITO48877.2020.9197833.
- [13] S. K. Assayed, K. Shaalan, and M. Alkhatib, "A Chatbot Intent Classifier for Supporting High School Students," *ICST Transactions on Scalable Information Systems*, vol. 10, no. 3, p. e1, Dec. 2022, doi: <https://doi.org/10.4108/eetsis.v10i2.2948>.
- [14] M. R. A. Prasetya and A. M. Priyatno, "Dice Similarity and TF-IDF for New Student Admissions Chatbot," *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 1, no. 1, pp. 13–18, Jul. 2022, doi: <https://doi.org/10.31004/riggs.v1i1.5>.
- [15] B. R. Ranoliya, N. Raghuwanshi and S. Singh, "Chatbot for university related FAQs," *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, India, 2017, pp. 1525-1530, doi: 10.1109/ICACCI.2017.8126057.
- [16] T. Ranjith Kumar, A. Akarapu, S. Allam, N. Gattepally, and P. Yashaswi Nellutla, "CHATBOT APPLICATION USING NLTK AND KERAS," Apr. 2024. Accessed: Sep. 02, 2024. [Online]. Available: <https://www.irjet.net/archives/V11/i4/IRJET-V11I4418.pdf>
- [17] T, Archana, et al. "DocBot: Integrating LLMs for Medical Assistance." *SSRN Electronic Journal*, 2025, papers.ssrn.com/sol3/papers.cfm?abstract_id=5089136, <https://doi.org/10.2139/ssrn.5089136>. Accessed 6 May 2025.

REFERENCES

- [18] Azam, Ayesha, et al. *CancerBot: A Retrieval-Augmented Generation Based Cancer Chatbot Using Large Language Models*. 26 Dec. 2024, pp. 1–6, <https://doi.org/10.1109/icosst64562.2024.10871155>. Accessed 4 Apr. 2025.
- [19] Geeksforgeeks. “Understanding TF-IDF (Term Frequency-Inverse Document Frequency).” *GeeksforGeeks*, 20 Jan. 2021, www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/.
- [20] “What Is a Sentence Transformer?” *Marqo.ai*, 2024, www.marqo.ai/course/introduction-to-sentence-transformers.
- [21] Awan, Abid Ali. “LlamaIndex: Adding Personal Data to LLMs.” *Datacamp.com*, DataCamp, 27 July 2023, www.datacamp.com/tutorial/llama-index-adding-personal-data-to-llms. Accessed 6 May 2025.
- [22] Selvaraj, Natassha. “What Is Retrieval Augmented Generation (RAG)?” *Datacamp.com*, DataCamp, 5 Jan. 2024, www.datacamp.com/blog/what-is-retrieval-augmented-generation-rag.
- [23] P. Wainaina, “Mastering Data Science with CRISP-DM Methodology: A Step-by-Step Guide,” *Medium*, Feb. 11, 2024. <https://medium.com/@wainaina.pierre/mastering-data-science-with-crisp-dm-methodology-a-step-by-step-guide-a976b71257b5#:~:text=%E2%80%9C%20CRISP%20DDM%20remains%20the%20most> (accessed Sep. 02, 2024).
- [24] Visual Studio Code. “Visual Studio Code.” *Visualstudio.com*, 14 Apr. 2016, code.visualstudio.com/Download.
- [25] Apache Friends. “XAMPP Installers and Downloads for Apache Friends.” *Www.apachefriends.org*, 2022, www.apachefriends.org/.
- [26] Melanie. “Google Colab: The Power of the Cloud for Machine Learning.” *Data Science Courses / DataScientest*, 17 Apr. 2024, datascientest.com/en/google-colab-the-power-of-the-cloud-for-machine-learning.
- [27] PubNub. “What Is Ngrok? - PubNub - Medium.” *Medium*, Medium, 24 Nov. 2023, medium.com/@PubNub/what-is-ngrok-9f35088c6dcf.

REFERENCES

- [28] AI, Hands on. “Build a Chat Bot from Scratch Using Python and TensorFlow.” *Medium*, 12 Apr. 2023, handsonai.medium.com/build-a-chat-bot-from-scratch-using-python-and-tensorflow-fd189bcfae45.
- [29] The PHP Group. “PHP: Hypertext Preprocessor.” *Php.net*, 4 Apr. 2019, www.php.net/.
- [30] “SandLogicTechnologies/Llama3-Med42-8B-GGUF · Hugging Face.” *Huggingface.co*, 2025, huggingface.co/SandLogicTechnologies/Llama3-Med42-8B-GGUF. Accessed 6 May 2025.

APPENDIX

Flyer For Dataset

推拿 Tuina		
医师 Practitioner	科系 Category	详情 Details
胡后曼主治医师(男) Mr. Aw Chi Min (M) 硕士学位于广西中医药大学 MMed GXUTCM	中医姿势问题 TCM Posture Problem	脊柱侧弯、肩高低、驼背、骨盆倾斜等 Scoliosis, uneven shoulder, hunched back, pelvic tilt, bow legs, etc.
	中医上肢关节疾病 Upper Limbs Disorder	颈部、肩部、上肢关节疼痛、肩周炎、网球肘等 Neck, shoulder, upper limbs, musculoskeletal pain, frozen shoulder, tennis elbow, etc.
黄德国医师(男) Mr. Ng De Quade (M) 学士学位于拉曼大学 BChinMed (Hons) UTAR	中医下肢关节疾病 Lower Limbs Disorder	下背部、臀部、腿部、膝盖、脚后跟肌肉骨骼疼痛、脚踝扭伤和运动损伤 Lower back, hip, legs, knees, heels musculoskeletal pain, ankle sprain and other sports injuries.
	中医下肢关节疾病 Lower Limbs Disorder	
针灸 Acupuncture		
医师 Practitioner	科系 Category	详情 Details
林元凯主治医师(男) Mr. Lim Yuan Khay (M) 硕士学位于广州中医药大学 MMed GZUTCM	中医神经科 TCM Neurology	头痛头胀、偏头痛、面三叉神经痛、眼痛、下肢麻痹、神经痛、中风等 Headache tension, migraine, trigeminal neuralgia, ophthalmalgia, crural paralysis, neuralgia, stroke, etc.
	中医下肢关节疾病 Lower Limbs Disorder	腰腿下肢久慢性疼痛、坐骨神经痛、神经损伤后疼痛、肋间神经痛等 Chronic pain in lower limbs, sciatica, pain after nerve injury, intercostal neuralgia, etc.
	其他 Others	糖尿病、肥胖 Diabetes, obesity
赖佩欣主治医师(女) Ms. Lai Phooi Yan 学士学位于英迪国际大学 BChinMed (Hons) Inti	中医上肢关节疾病 Upper Limbs Disorder	肩颈上肢久慢性疼痛、弹响指、肩周炎、网球肘、上肢麻痹、癌性疼痛等 Chronic pain in upper limbs, snapping finger, frozen shoulder, tennis elbow, brachial palsy, cancer pain, etc.
	中医妇科疾病 TCM Gynaecological Disease	痛经、经期腰痛、慢性盆腔炎、月经失调、白带异常、不孕症、产后腹胀/身痛/抑郁、更年期综合症等 Dysmenorrhea, menstrual pain, chronic pelvic pain, menstrual disorder, abnormal leucorrhoea, infertility, post partum stomach bloating/body ache/depression, menopausal syndrome, etc.
	中医美容 TCM Cosmetology	
郑裕强主治医师(男) Mr. Chia Yi Keong (M) 硕士学位于广州中医药大学 MMed GZUTCM	中医内科疾病 TCM Internal Disease	心悸心慌、心胸痛、胃脘痛、下腹疼痛、腹泻、咳嗽、哮喘等 Palpitation, chest pain, epigastric pain, hypogastralgia, diarrhea, cough, asthma, etc.
	中医急性疼痛 TCM Acute pain	四肢筋伤病情少过1个月或1个月以内、软组织损伤 Limbs tendon injury (more than one month), soft tissue injury
	中医心理问题 TCM Psychological Issue	焦虑、抑郁、睡眠障碍 Anxiety, depression, sleep disorder
中医康复科 Rehabilitation		
医师 Practitioner	科系 Category	详情 Details
王朝正主治医师(男) Mr. Wang Chaozheng 中文硕士学位于拉曼大学 MA (Chinese Studies) UTAR	中医上肢关节疾病 Upper Limbs Disorder	颈部、肩部、上肢关节疼痛、肩周炎、网球肘等 Neck, shoulder, upper limbs, musculoskeletal pain, frozen shoulder, tennis elbow, etc.
	中医下肢关节疾病 Lower Limbs Disorder	下背部、臀部、腿部、膝盖、脚后跟肌肉骨骼疼痛等 Lower back, hip, legs, knees, heels musculoskeletal pain, etc.
	中医骨、关节和肌肉系统疾病 Bone, Joint & Muscle System Disorder	关节疼痛、肌肉痉挛、麻木、震颤等 Joint pain, muscle spasm, numbness, tremor, etc.

中医内科 TCM Internal Medicine		
医师 Practitioner	科系 Category	详情 Details
郑建强主任医师(男) Dr. Te Jian Keong (M) 博士学位于南京中医药大学 PhD NUCM	中医内科疾病 TCM Internal Disease	肿瘤、新冠肺炎、新冠肺炎后遗症、癌症问题 Tumor, Covid-19, Post Covid-19, cancer problem
	中医杂病 TCM Miscellaneous Disease	疑难杂病 Difficulty and miscellaneous diseases
廖政阳主治医师(男) Mr. Liow Zheng Yang (M) 硕士学位于广州中医药大学 MMed GZUTCM	中医脾胃系疾病 TCM Spleen & Gastrointestinal System Disorder	胃胀、胃痛、反酸、呕吐、便秘、腹泻、吐血等 Stomach bloating, stomach pain, acid reflux, vomiting, constipation, diarrhoea, haematemesis, etc.
	中医肾系疾病 TCM Kidney System Disorder	水肿、阳痿、夜尿、早泄、尿血、前列腺、泌尿道问题等 Edema, impotence, nocturnal, premature ejaculation, prostate, urinary problem, etc.
	其他 Others	糖尿病、肥胖 Diabetes, obesity
	中医五官问题 TCM ENT	耳鸣、梅尼埃综合征、过敏性鼻炎、鼻窦炎、急性慢性咽喉炎、眼睛干涩 Tinnitus, Meniere's syndrome, allergic rhinitis, sinusitis, acute, and chronic pharyngitis, dry eyes, etc.
朱子贤医师(男) Mr. Choo Zi Xian (M) 学士学位于上海中医药大学 BMed SHUTCM	中医骨、关节和肌肉系统疾病 Bone, Joint & Muscle System Disorder	关节疼痛、肌肉痉挛、麻木、震颤等 Joint pain, muscle spasm, numbness, tremor, etc.
伍惠仪医师(女) Ms. Goh Wui Yee (F) 学士学位于上海中医药大学 BMed SHUTCM	中医妇科疾病 TCM Gynaecological Disease	痛经、月经失调、白带异常、经前综合症、更年期综合症、不孕症、子宫内膜异位症、多囊卵巢综合症、子宫肌瘤、外阴瘙痒、乳腺疾病、产后治疗与调理、妇科荷尔蒙失调、孕吐、先兆流产等 Dysmenorrhea, menstrual disorders, abnormal leucorrhoea, premenstrual syndrome, menopausal syndrome, infertility, endometriosis, polycystic ovary syndrome, uterine fibroid, genital itching breast disease, post partum treatment and care, gynaecological hormonal imbalance, morning sickness, threatened abortion, etc.
黄子敏医师(女) Ms. Wong Zixin (F) 学士学位于拉曼大学 BChinMed (Hons) UTAR	中医肝系疾病 TCM Liver System Disorder	肝炎、肝硬化、黄疸、胆结石、甲状腺等 Hepatitis, cirrhosis, jaundice, gallstone, thyroid disease, etc.
	中医心理问题 TCM Psychological Issue	双向情感障碍、焦虑、抑郁等 Bipolar disorder, anxiety, depression, etc.
	中医皮肤问题 TCM Dermatology	痤疮、皮肤瘙痒、皮疹、湿疹、带状疱疹、紫癜 Acne, itchy skin, rash, eczema, purpura, etc.
陈展庆主治医师(男) Mr. Tan Chan Qing (M) 硕士学位于广州中医药大学 MMed GZUTCM	中医肺系疾病 TCM Lung System Disorder	感冒、咳嗽、哮喘、肺结核、鼻出血、咳血等 Common Cold, cough, asthma, TB, nose bleeding, hemoptysis, etc.
	中医心系疾病 TCM Heart System Disorder	失眠、心悸、胸痛、心力衰竭、癫痫、痴呆等 Insomnia, heart palpitations, chest pain, heart failure, epilepsy, dementia, etc.
	中医儿科 TCM Pediatrics	
心理咨询 Counselling		
医师 Practitioner	科系 Category	详情 Details
吴恺康辅导员(女) Ms. Ng Kai Yean (F) 马来亚大学辅导学士 BCoun UM	心理问题 Psychological Issue	焦虑、忧郁、睡眠困难、关系与沟通、边缘性人格困扰、心理健康、个人成长 Anxiety, depression, sleep difficulties, relationship and communication, borderline personality disturbance, mental health, personal growth

Example of Symptom Get from Cleveland Clinic Website

What are the symptoms of kyphosis?

The main symptoms of kyphosis include:

- Rounded shoulders.
- A curve or hump in your upper back.
- Tight [hamstrings](#) (muscles in the back of your thighs).

Severe kyphosis may cause the following symptoms:

- Pain or stiffness in your back and shoulder blades.
- [Numb](#), weak or tingling legs.
- Extreme [fatigue](#).
- [Balance issues](#).
- Bladder [incontinence](#) or bowel incontinence.
- Shortness of breath or difficulty breathing.

POSTER



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

MEDIBOT UTAR HOSPITAL AI HEALTH COMPANION

INTRODUCTION

- THIS PROJECT AIMS TO DEVELOP A SYSTEM COMING WITH CUSTOMIZED CHATBOT FOR UTAR HOSPITAL TO ADDRESS THE NEED FOR EFFICIENT HEALTHCARE SOLUTIONS.

OBJECTIVES

- TO DEVELOP A BILINGUAL (ENGLISH AND CHINESE) CHATBOT FOR UTAR HOSPITAL THAT UTILIZES MACHINE LEARNING TO PREDICT POTENTIAL DISEASES BASED ON SYMPTOM DESCRIBED BY USERS.
- TO IMPLEMENT DOCTOR RECOMMENDATIONS FROM UNIVERSITI TUNKU ABDUL RAHMAN'S T&CM CENTER BASED ON THE DISEASE CATEGORY PREDICTED.
- TO DESIGN AN ONLINE APPOINTMENT SCHEDULING SYSTEM FOR UTAR HOSPITAL T&CM CENTRE WITH EMAIL NOTIFICATIONS FOR CONFIRMATIONS, CANCELLATIONS, AND REMINDERS.
- TO DEVELOP A DASHBOARD TO MONITOR DISEASE TREND, USER ENGAGEMENT, APPOINTMENT MONITORING

CONTRIBUTION

ENHANCES INCLUSIVITY BY BRIDGING LANGUAGE GAPS AND IMPROVING ACCESS TO HEALTHCARE INFORMATION WITH DISEASE DIAGNOSIS, DOCTOR RECOMMENDATION CHINESE-ENGLISH LANGUAGE CHATBOT AND DASHBOARD OFFERS KEY INSIGHTS INTO USER BEHAVIOR AND DISEASE TRENDS.



FURTHER IMPROVEMENT

- VOICE FEATURE
- AUTOMATE REPORT ANALYSIS SYSTEM
- MORE LANGUAGE SUPPORT



PROJECT DEVELOPER: TONG JIA SENG (20ACB02867)
PROJECT SUPERVISOR: DR ABDULKARIM KANAAN JEBNA