Automate Customer Support Handling E-commerce enquiry using ChatGPT

BY

TEO WEN JIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: __Automate Customer Support Handling E-commerce enquiry using ChatGPT_

Academic Session: _JUNE 2024__

I __TEO WEN JIN_

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

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

Verified by,

Alex

(Author's signature)

(Supervisor's signature)

Address:
39, Lorong Batu Nilam 33A,
Bandar Bukit Tinggi 2,
41200 Klang

___Ts Dr Ooi Boon Yaik_
Supervisor's name

Date: _____11/9/2024_

Date: ___11/9/2024___

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ___11/9/2024__________________

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that _____TEO WEN JIN__________ (ID No: __*20ACB02426*____) has completed this final year project entitled "_Automate Customer Support Handling E-commerce enquiry using ChatGPT__________" under the supervision of ____Ts Dr Ooi Boon Yaik____________________________ (Supervisor) from the Department of __Computer Science_________________________, Faculty of __Information and Communication Technology_____________ .

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____
(*Teo Wen Jin*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**METHODOLOGY, CONCEPT AND DESIGN OF A 2-MICRON CMOS DIGITAL BASED TEACHING CHIP USING FULL-CUSTOM DESIGN STYLE**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________________________

Name : ____Teo Wen Jin_____________________

Date : _____11/9/2024__________________

ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude to my supervisor, Ts Dr Ooi Boon Yaik, for providing me invaluable guidance and expertise to work on the project of Automate Customer Support Handling E-commerce enquiry using ChatGPT. This project has been my initial step towards establishing myself in the field of automate customer support application.

I am deeply grateful to all my university friends, for their unwavering patience, support, and love. Their presence during challenging times has been a pillar of strength for me. Lastly, I owe a debt of gratitude to my parents and family for their constant love, support, and encouragement throughout this journey. Their faith in me has been a continuous source of motivation.

ABSTRACT

The primary objective of this thesis is to improve customer support on e-commerce platform by proposing an innovative solution that integrates advanced technologies and methodologies. The motivation stems from the need to enhance the efficiency of the customer reply system, reduce the workload on the customer support team, and increase company sales. One of the long-term goal in the field of AI is to build computer systems that can have human-like conversations with users. With recent advances in AI technologies, we are now one step closer to achieving this goal. This proposal makes significant contributions to reduce manpower dependency and increase overall business competency in customer support on e-commerce platforms. The development of an automated context handling mechanism ensures precise and efficient customer support by reducing the need for human manpower. The automated summarization feature streamlines human agents' tasks by summarizing the entire conversations. By this it can help in saving time and increasing overall competency. The use of ChatGPT enhances the competency of business interactions by providing contextually relevant and precise responses. We are going to integrate this mechanism with e-commerce platform aligns with evolving customer communication preferences and enhances the business's competency in answering to customer inquiries on this e-commerce platform. Additionally, order processing functionality will be integrated within the chat interface to provide convenience to the customers so that they can easily make order using a shorter time. The project scope is the development of a comprehensive mechanism for context handling, an inconspicuous human takeover process, and the summarization of entire conversations between customers and automated customer support before handover to human agents. The function of context handling ensures that automated responses remain relevant to the business, while summarization significantly can help in reducing the workload on human agents during handover sessions. Integration with ChatGPT allows for accurate responses, and integrate with the Instagram platform enables efficient responses to customer questions. The project also includes the implementation of functionality for customers to place orders through the chat interface. The methodology involves the design of a distributed system architecture for scalability and efficient task distribution. Machine Learning-based Named Entity Recognition (NER) is employed to identify and extract specific entities, while contextual analysis algorithms determine message relevance for summarization. Reinforcement learning techniques will adapt the summarization model based on human agent feedback, and

feedback analysis identifies areas for continuous improvement. This comprehensive approach aims to transform customer support on Instagram, offering a seamless and efficient experience that reduces workload, accelerates sales, and enhances overall business competency. Last but not least, the core objective is to propose a new mechanism that automatically summarizes conversations between automated customer support and customers, facilitating effective handover to human agents. The proposed solution focuses on creating a natural and human-like conversation flow, aligning with customer preferences, and minimizing response times for quicker response from the customer support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

β	beta
Ω	Ohm (resistance)

LIST OF ABBREVIATIONS

5G	Fifth Generation
API	Application Programming Interface
CPU	Central Processing Unit
GPIO	General Purpose Input Output
IOT	Internet of Things
IP	Internet Protocol
RAM	Random Access Memory

Chapter 1

Introduction

In this chapter, we present the background and motivation of our research, our contributions to the field, and the outline of the thesis. Forensic investigation has been studies Artificial intelligence (AI) has influenced how we engage in our every day activities by evaluating and designing advanced applications and devices, called intelligent agents, which can perform several functions that can help human to save a lot of time. Company usually spend a lot of time and energy on replying customers' inquiries. Therefor, the aim of this project is to increase the efficiency of customer reply system, reduce the workload on our customer support team and increase the company sales in a shorter time. The primary objective is to enhance the functionality of customer support interactions by ensuring context relevance, providing easy handover to human agents, and offering summarized conversation histories for streamlined assistance. By having this automate customer support handling e-commerce enquiry system, the company will safe a lot of time and energy on replying customers. Furthermore, the project's scope is focused explicitly on the development of the context handling mechanism, conversation summarization feature, and an efficient handover process. It aims to optimize the interaction between customers and automated support, emphasizing business relevance. The project does not involve the creation of a standalone automated customer support system. Instead, it enhances the existing system's capabilities in context management and interaction handovers. Additionally, the project's scope does not extend to the broader aspects of automating customer support beyond the specified functionalities.

1.1 Problem Statement and Motivation

Nowadays, business company aim to increase their sales by reducing costs to provide capital for growth. However, the company currently lacks the necessary tools to tackle this objective effectively. To achieve this growth target, the company needs to use effective tools such as chatbot, as well as its IT department team to fully utilize the chatbot, to provide quick and timely responses to customer inquiries. Hence, the company can increase customer satisfaction and confidence in purchasing their products. The current automated customer support systems in the market presents two significant challenges that hinder operational efficiency and user experience. Firstly, the first problem statement is the absence of conversation summarization in automate customer support. Existing systems lack the capability to effectively summarize the entire conversation between customers and chatbots. This deficiency forces human agents to expend valuable time scrolling through extensive dialogue histories, impeding the swift resolution of customer queries. Thus, this will significantly amplifies the workload of human agents. This heavy task not only demands a considerable investment of valuable time but also hampers the efficiency of the customer support workflow. Therefore, it will increase the company cost to hire more manpower as customer support.

Besides that, another problem statement is conspicuous human takeover process. A general challenge in existing automated customer support systems is the noticeable of the human takeover process by customers. Many systems unintentionally signal the transition from AI chatbot to human agent, leading customers to lose confidence in the authenticity of their interactions. Our goal is to design a system that seamlessly integrates human takeover without customers notice whether they are conversing with a chatbot or a human agent.

1.2 Objectives

In this project, the aim of the thesis is to increase the efficiency of customer reply system, reduce the workload on our customer support team and increase the company sales in a shorter time. Although chatbot is commonly used in many platform, but many people still prefer to wait in queue for a real agent to reply rather than receive an instant response from a chatbot. In this project, we choose to use Chatgpt to integrate with e-commerce platform because ChatGPT has been trained on a vast amount of text data from the internet, which allows it to have a deep understanding of human language and context. This enables it to generate more coherent and contextually relevant responses compared to many other chatbots.

The core objective of this thesis is to present a solution that not only optimize the efficiency of the customer reply system but also automate the process of ordertaking, and eliminating the need for manual reading of every single message. Furthermore, the workload on our customer support team will be reduced significantly. Thus, in turn, will empower the team to focus on more complex queries and provide a more personalized service to our valued customers. In addition, a more efficient support system will produce a quicker answer for customers, which can lead to increase the confidence and satisfaction in customers and potentially create more sales in a shorter time.

1.3 Project Scope and Direction

The scopes of the project include develop a mechanism for context handling, inconspicuous human takeover process, summarizing the whole conversation between customer and automated customer support before handover to human agents, and most importantly is the process of handover to human agents. Context handling is to maintain context during the conversation so that the automate customer support will not answer something that is not related to our business or out of the topic. A mechanism will be created to summarize the conversation between customer and automated customer support so that this summary can be sent to human agents for easy handover session. The human agents will save a lot of time by reading the summary that is provided instead of scrolling the whole way up to read a lengthy conversation between the customer and the automated customer support. Besides that, the scopes of the project also include implementing ChatGPT to understand the questions that are given by the customers and generate accurate responses. Therefore, I have to make API calls to the ChatGPT service to generate the responses and reply to customers. Other than that, the scopes of the project also include integrating my mechanism with the e-commerce platform to access and respond to customer questions. Moreover, I need to understand the Chatgpt's API and any limitations it might have. My project scope also includes implementing functionality to allow customers to place orders through the chat interface. This might include handling order details, confirming orders, and providing order summaries. On the other hand, the project scope for implementing an inconspicuous human takeover process in the context of customer support on e-commerce chatbot includes integration with automated Chatbot systems. We will work on the integration of the inconspicuous human takeover process with existing automated chatbot systems. This involves ensuring compatibility between the automated responses generated by the chatbot and the responses provided by human agents. We are going to include analysing user engagement, success rates of the takeover process, and identifying areas for continuous improvement.

The scope of the FYP2 focuses on developing a comprehensive real-time customer service system that integrates both automated chatbot interaction and human agent support.

This includes building two key user interfaces whic are the Customer interface, which allows customers to send messages and receive replies in real time, and the Agent interface, which provides agents the capability to monitor live conversations and take over when necessary. Both interfaces are built to utilize Socket.IO for seamless real-time communication with the backend. The Flask server acts as the core of the system, managing multiple customer sessions simultaneously through session-based room management, while ensuring proper message routing between customers, agents, and the backend system. Besides that, he system also features a two-tiered response mechanism. The first tier involves checking predefined frequently asked questions (FAQs) stored in an excel file, which help in providing instant responses for known queries. The chatbot will generates human-like responses based on the customer query if the question doesn't match any entry in the Excel data, and involves the second tier which invoking OpenAI GPT API. This ensures that even if a query falls outside the predefined database, the customer still receives a response. The project also includes a human-agent takeover feature to allow agents to step into a conversation if the AI cannot resolve the issue or if the customer requests human support. The system further maintains comprehensive session management that store all conversation histories for future reference or review. Overall, the project integrates cutting-edge AI with traditional customer service methods to provide a more efficient, responsive, and flexible customer support experience, leveraging both automated responses and human interaction when needed.

1.4 Contributions

This proposal outlines significant contributions aimed at reducing manpower dependency and elevating the overall competency of the business of customer support on e-commerce. Firstly, the development of a automated context handling mechanism contributes to a more precise and efficient customer support system. By automating context management, the project reduces the need for extensive human oversight, minimizing manpower requirements. This ensures that automated responses align closely with our business context, enhancing the overall competency of customer interactions.

Secondly, the feature of an automated summarization in the mechanism for customer interactions is a key contribution, providing significant time-saving benefits for human agents. By summarizing entire conversations, the project reduces the manual effort required to comprehend lengthy conversation. This directly addresses manpower challenges, allowing human agents to allocate their time more efficiently and focus on critical tasks, thereby increasing overall competency.

Thirdly, the advantage using ChatGPT for understanding customer queries and generating accurate responses enhances the competency of business interactions. The advanced natural language help in producing more contextually relevant and precise responses. This automation not only reduces reliance on human intervention but also ensures a higher standard of customer support, boosting overall business competency.

Our project integration reduces manual efforts by reading every single message, streamlining communication and responsiveness. Consequently, it enhances the business's competency in catering to customer inquiries on a popular platform, aligning with evolving customer communication preferences. In addition, the implementation of order processing functionality within the chat interface is a pivotal contribution. We can accelerate sales transactions and also streamlines business processes by allowing customers to place orders and manage details directly through chat. This contributes to a reduction in dedicated order processing manpower, promoting overall business efficiency and competency.

1.5 Report Organization

This report is organised into 6 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Design, Chapter 4 System Implementation and Testing, Chapter 5 System Outcome and Discussion, Chapter 6 Conclusion. The first chapter is the introduction of this project which includes problem statement, project background and motivation, project scope, project objectives, project contribution, highlights of project achievements, and report organisation. The second chapter is the literature review carried out on several existing IoT applications in the market to evaluate the strengths and weaknesses of each product. The third chapter is discussing the overall system design of this project. The fourth chapter is regarding the details on how to implement the design of the system. Furthermore, the fifth chapter reports the system outcome and discussion. Last but not least, the sixth chapter reports the system testing and performance metrics.

Chapter 2

Literature Review

2.1 Previous works on Deep Learning

This chapter presents previous work on AI chatbot, automate reply customer in real time function, accurately interprets user questions and matches them to specific replies and chat within content function. Also, their limitations and suggested solutions are also presented.

2.1.1 Review summary of AI chatbot: Manychat

ManyChat is a popular chatbot platform designed for businesses and marketers to create and manage chatbots on various messaging platforms, primarily on Facebook Messenger. ManyChat offers an intuitive and user-friendly interface that allows users to easily build chatbots without extensive coding knowledge. The drag-and-drop builder simplifies the process of creating conversational flows. ManyChat excels in automating conversations through the use of sequences. Users can create automated responses, drip campaigns, and follow-up messages to nurture leads and guide user through predefined workflows. The platform facilitates one-to-many broadcasting, enabling businesses to send updates, promotions, or important information to their entire subscriber base. The broadcasting feature helps maintain consistent communication. [1]

2.1.2 Review summary of AI chatbot: Chatfuel

Chatfuel is a adaptable chatbot platform designed to simplify the creation and management of chatbots, primarily on the Facebook Messenger platform. This user- friendly tool offers a range of features to empower businesses and marketers in enhancing their customer engagement strategies. Chatfuel excels in automating conversations through the use of sequences. Users can design chatbot workflows, automate responses, and create drip campaigns to guide users through predefined interactions, enhancing user experience. Besides that, the broadcasting feature enables businesses to send messages, updates, and promotions to their subscribers. This ensures consistent communication and helps in keeping the audience informed about relevant content. Morover, Chatfuel includes a user-friendly editor which is enabling users to add, edit and design chatbots simply without knowledge of programming or coding.[2]

2.1.3 Review summary of AI chatbot: Dialog flow

Dialogflow is an A chatbot developed by Google. It has emerged as a leading platform for creating conversational agents and chatbots. Thera are numerous studies and articles have explored its features, applications, and effectiveness in various domains. Researchers found that Dialogflow is a powerful Natural Language Processing (NLP) capabilities. The platform excels in understanding and processing user inputs, allowing developers to create chatbots that can engage in natural and contextually relevant conversations.[3] Moreover, Dialogflow is able to recognize user intents and manage contextual information. Studies emphasize how the platform enables developers to design chatbots that comprehend user queries and maintain context throughout a conversation, enhancing the overall user experience. Dialogflow has integration with Google Cloud Services. This integration provides additional functionalities, such as data storage, analytics, and scalability, making Dialogflow an attractive choice for businesses with complex requirements [4].

Chapter 3

System Methodology/Approach OR System Model

3.1 System Architecture Diagram

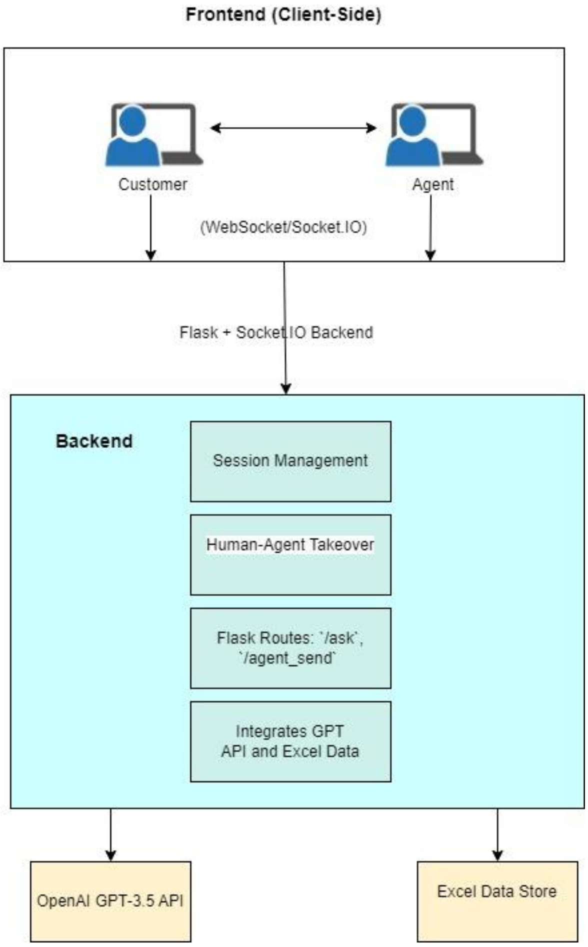


Figure 3.1 System Architecture Diagram

In this system design, Socket.IO is used by both the Customer UI and the Agent UI to establish real-time connections with the backend. Users may communicate with the Flask server through the Customer UI and receive replies from chatbots or agents; conversely, the Agent UI facilitates real-time monitoring and the ability to take over conversations when necessary. The Flask and Socket.IO backend uses rooms to manage sessions and asks the OpenAI GPT API for responses to consumer enquiries after first verifying prepared answers kept in an Excel data source. A flag in the backend governs human-agent takeover. Real-time message dissemination between clients and agents is also managed by the backend. External services like the OpenAI GPT API are used for generating chatbot replies, and the Excel data provides quick access to FAQs. The system architecture ensures seamless communication between the frontend UIs and backend, with flexible use of predefined and AI-generated responses.

3.2 Use case diagram

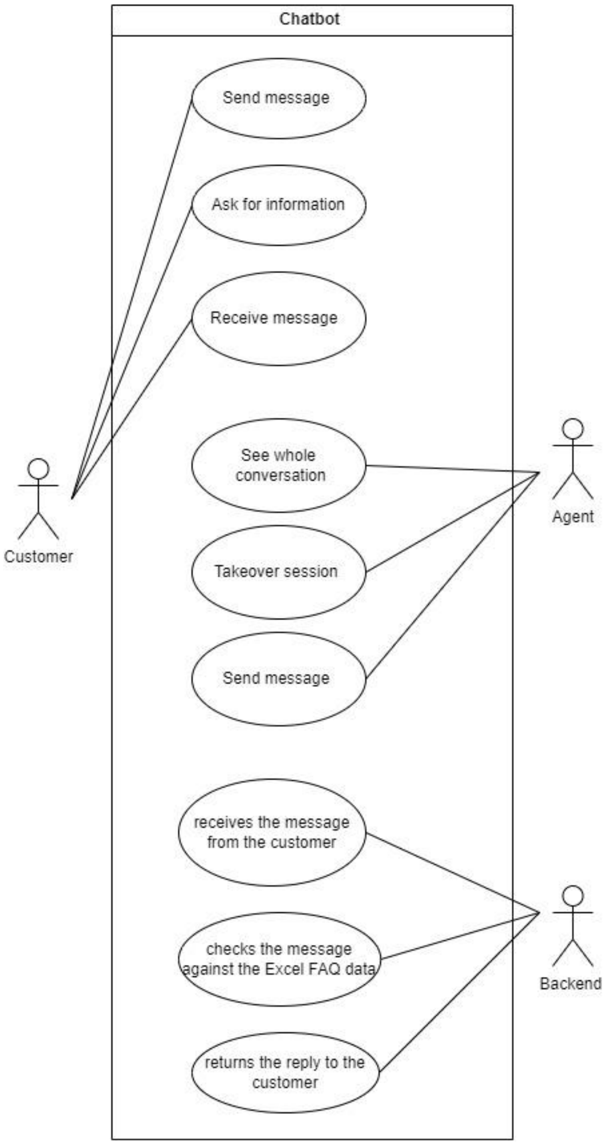


Figure 3.2.1: Use Case Diagram

3.2.2 Use Case Description

<table>
<tr><td><b>Use Case Name:</b> Customer Sends Message</td><td><b>ID:</b> 1</td><td><b>Importance Level:</b> High</td></tr>
<tr><td colspan="2"><b>Primary Actor:</b> Customer</td><td><b>Use Case Type:</b> Detail, Essential</td></tr>
<tr><td colspan="3"><b>Stakeholders and Interests:</b>
· Customer: Wants to receive a relevant and prompt response to their query.
· Business: Aims to provide seamless customer service to improve satisfaction.</td></tr>
<tr><td colspan="3"><b>Brief Description:</b> The customer sends a message through the Customer UI.</td></tr>
<tr><td colspan="3"><b>Trigger:</b> Customer enters the app and open chat
<b>Type:</b> External</td></tr>
<tr><td colspan="3"><b>Relationships:</b>
Association: Customer
Include: Real time conversation</td></tr>
<tr><td colspan="3"><b>Normal Flow of Events:</b>
1. Customer types and submits a message.
2. Message is sent via Socket.IO to the Flask backend.
3. Backend processes the message and responds with a reply.</td></tr>
<tr><td colspan="3"><b>SubFlows:</b> not aplicable</td></tr>
<tr><td colspan="3"><b>Alternate/Exceptional Flows:</b>
not aplicable</td></tr>
</table>

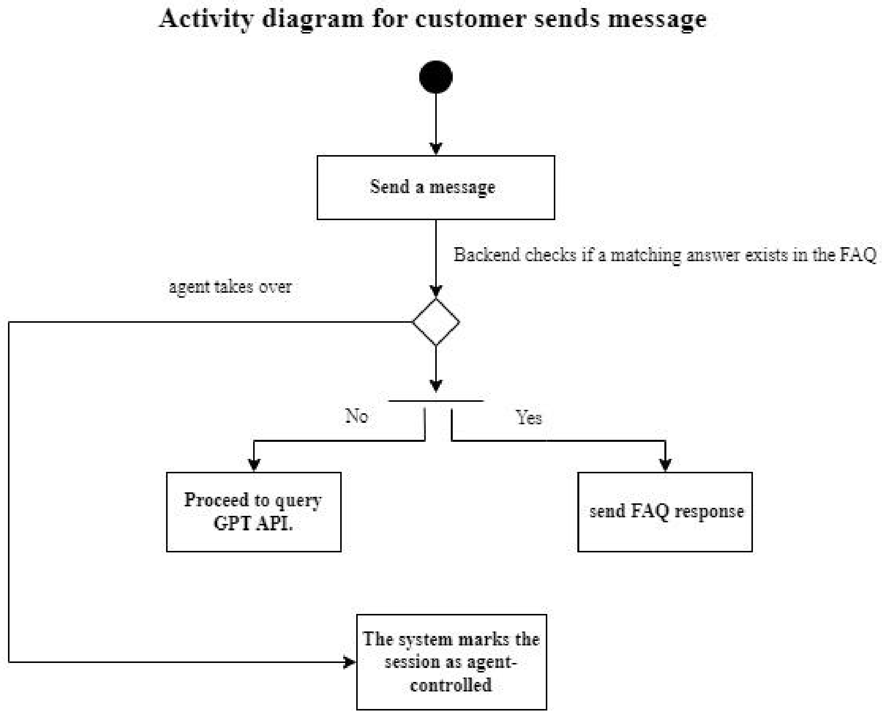
Use Case Name: Agent Monitors Conversations	**ID:** 2	**Importance Level:** High
Primary Actor: Customer		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • Agent: Needs to monitor customer interactions in real-time to decide when to intervene. • Customer: Expects help if needed.		
Brief Description: The agent can monitor customer conversations in real-time.		
Trigger: Agent logs into the system and opens the monitoring interface. **Type:** External		
Relationships: Association: Agent monitors ongoing conversations. Include: Real-time monitoring of multiple sessions. Extend: None. Generalization: None.		
Normal Flow of Events: 1. Agent logs into the system. 2. Agent UI receives real-time updates of ongoing customer conversations.		
SubFlows: not aplicable		
Postcondition:: The agent can see all ongoing conversations.		

<table>
<tr><td><b>Use Case Name:</b> Backend Processes Customer Message</td><td><b>ID:</b> 3</td><td><b>Importance Level:</b> High</td></tr>
<tr><td colspan="2"><b>Primary Actor:</b> Backend</td><td><b>Use Case Type:</b> Detail, Essential</td></tr>
<tr><td colspan="3"><b>Stakeholders and Interests:</b>
• Customer: Expects their question to be processed accurately and efficiently.
• Business: Ensures the system is working properly for automated responses.</td></tr>
<tr><td colspan="3"><b>Brief Description:</b> The backend processes the message received from the customer.</td></tr>
<tr><td colspan="3"><b>Trigger:</b> The customer message is received by the backend through Socket.IO.
<b>Type:</b> External</td></tr>
<tr><td colspan="3"><b>Relationships:</b>
Association: Flask receives customer message.
Include: Checking FAQ (Excel) data for matching response.
Extend: If no match is found in Excel, extend to use "Backend Requests OpenAI GPT Response".
Generalization: None.</td></tr>
<tr><td colspan="3"><b>Normal Flow of Events:</b>
1. Backend receives message.
2. System checks if the message exists in the Excel FAQ.
3. If a match is found, the response is sent.
4. If no match, OpenAI GPT generates a response.</td></tr>
<tr><td colspan="3"><b>SubFlows:</b> not aplicable</td></tr>
<tr><td colspan="3"><b>Postcondition::</b>
The agent can see all ongoing conversations.</td></tr>
</table>

<table>
<tr><td><b>Use Case Name:</b> Agent Takes Over Conversation</td><td><b>ID:</b> 4</td><td><b>Importance Level:</b> High</td></tr>
<tr><td colspan="2"><b>Primary Actor:</b> Agent</td><td><b>Use Case Type:</b> Detail, Essential</td></tr>
<tr><td colspan="3"><b>Stakeholders and Interests:</b>
• Agent: Takes control of conversations when necessary, providing human support.
• Customer: Receives personalized support.</td></tr>
<tr><td colspan="3"><b>Brief Description:</b> The agent takes control of a customer conversation.</td></tr>
<tr><td colspan="3"><b>Trigger:</b> The agent clicks the "Take Over" button in the UI.
<b>Type:</b> External</td></tr>
<tr><td colspan="3"><b>Relationships:</b>
Association: Agent takes over an active conversation.
Include: Switching control from chatbot to agent.
Extend: Can be extended by "Agent Sends Message" when the agent sends responses.
Generalization: None.</td></tr>
<tr><td colspan="3"><b>Normal Flow of Events:</b>
1. Agent selects a conversation.
2. Agent clicks "Take Over".
3. Backend marks the session as taken over.
4. Human-agent responses are sent directly to the customer.</td></tr>
<tr><td colspan="3"><b>SubFlows:</b> not aplicable</td></tr>
<tr><td colspan="3"><b>Postcondition::</b>
The agent manages the conversation, and the chatbot is deactivated.</td></tr>
</table>

Use Case Name: Backend Requests OpenAI GPT Response	**ID:** 5	**Importance Level:** High
Primary Actor: Flask + Socket.IO Backend		**Use Case Type:** Detail, Essential
Stakeholders and Interests: • Agent: Takes control of conversations when necessary, providing human support. • Customer: Receives personalized support.		
Brief Description: The backend requests a response from OpenAI GPT when no matching FAQ is found.		
Trigger: The agent clicks the "Take Over" button in the UI. **Type:** External		
Relationships: Association: Flask requests OpenAI GPT to generate a response. Include: Interacting with OpenAI API to generate responses. Extend: Extension of "Flask Backend Processes Customer Message" when no FAQ match is found. Generalization: None.		
Normal Flow of Events: 1. The backend forwards the customer's message to OpenAI GPT. 2. Backend detects no match in the FAQ. 3. The message is sent to OpenAI GPT API. 4. GPT generates a response. 5. OpenAI GPT generates a response. 6. The backend receives the response and sends it to the customer.		
SubFlows: not aplicable		
Postcondition:: Customer receives a relevant response from GPT.		

3.3 Activity Diagram


Activity diagram for customer sends message
Send a message
Backend checks if a matching answer exists in the FAQ
agent takes over
No
Yes
Proceed to query GPT API.
send FAQ response
The system marks the session as agent-controlled

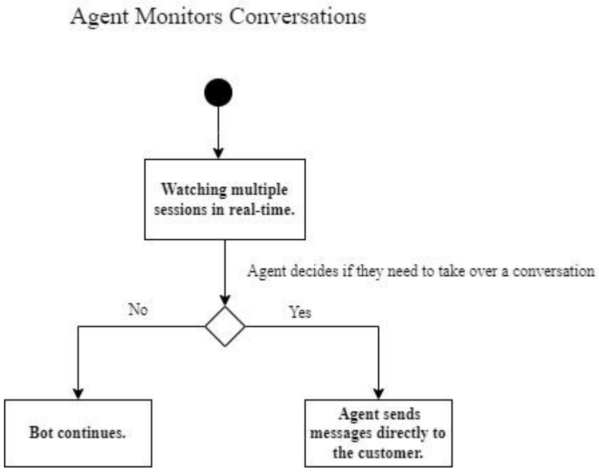

Agent Monitors Conversations
Watching multiple sessions in real-time.
Agent decides if they need to take over a conversation
No
Yes
Bot continues.
Agent sends messages directly to the customer.

Figure 3.3.2: Activity Diagram – Agent Monitors Conversations Use Case

Chapter 4 System Design

4.1 System Block Diagram

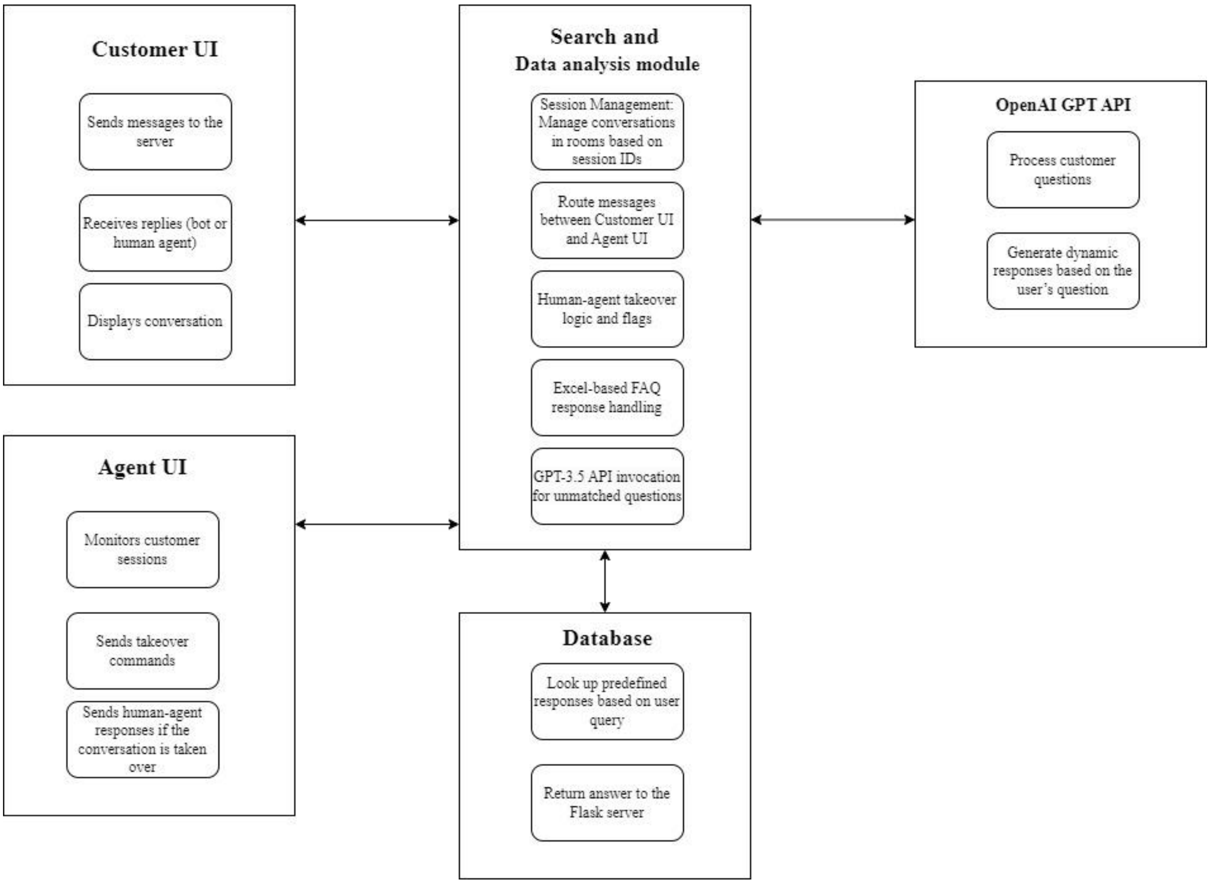

Customer UI
Sends messages to the server
Receives replies (bot or human agent)
Displays conversation
Search and Data analysis module
Session Management: Manage conversations in rooms based on session IDs
Route messages between Customer UI and Agent UI
Human-agent takeover logic and flags
Excel-based FAQ response handling
GPT-3.5 API invocation for unmatched questions
OpenAI GPT API
Process customer questions
Generate dynamic responses based on the user's question
Agent UI
Monitors customer sessions
Sends takeover commands
Sends human-agent responses if the conversation is taken over
Database
Look up predefined responses based on user query
Return answer to the Flask server

Figure 4.1: System Block Diagram

The System Block Diagram for this project illustrates the major components and their interactions, focusing on the flow of data and the functionalities of each part. The system is divided into six key components: the Customer UI, Agent UI, Flask + Socket.IO Backend, Excel Data Source, OpenAI GPT-3.5 API, and an optional Data Store. The Customer UI allows users to send messages via Socket.IO to the Flask Backend, which manages sessions and routes messages. If a customer's query matches predefined answers in the Excel Data Source, the response is returned from the database. If no match is found, the Flask Backend requests a generated response from the OpenAI GPT-3.5 API. The system also supports a human-agent takeover, where messages are redirected from the Customer UI to the Agent UI, allowing agents to monitor and take control of conversations. The Agent UI can send takeover requests and interact with the customer in real time. Optional Data Store functionality is included for storing session logs and chat histories, which can be retrieved for auditing or analysis. The diagram highlights data flow through arrows, showing how each block interacts with others and how external services like Excel and OpenAI GPT support the conversation process.

4.2 System Components Specifications

1. Customer UI (Frontend)

- **Technology**: HTML, CSS, JavaScript, Socket.IO (for real-time communication)
- **Purpose**: Interface where customers send queries and receive responses.
- **Specifications**:
  - **Input**: Customer message input, session management.
  - **Output**: Display of bot/agent responses in real-time.
  - **Functionalities**:
    - Real-time chat interface.
    - Ability to send text-based queries.
    - Session-based interaction for continuity in conversation.

2. Agent UI (Frontend)

- **Technology**: HTML, CSS, JavaScript, Socket.IO (for real-time communication)
- **Purpose**: Interface for agents to monitor customer interactions and take over when needed.
- **Specifications**:
  - **Input**: Incoming customer messages, agent responses.
  - **Output**: Display of customer queries and system responses.
  - **Functionalities**:
    - Monitor multiple sessions in real-time.
    - Take control of customer chat sessions.
    - Send responses to the customer when taking over a session.

3. Flask + Socket.IO Backend

- **Technology**: Python, Flask, Socket.IO
- **Purpose**: Centralized server managing sessions, communication, and request routing.

- **Specifications**:
    - **Input**: Customer queries, agent messages, requests for responses.
    - **Output**: Processed customer responses, updated conversation history.
    - **Functionalities**:
        - Real-time communication using Socket.IO.
        - Manage customer sessions using WebSocket connections.
        - Route customer queries to either the Excel data or OpenAI GPT.
        - Handle human agent takeover with session control.
        - Store and retrieve conversation history for each session.

4. Excel Data Block (Predefined FAQs)

- **Technology**: Pandas (Python library), Excel file (.xlsx)
- **Purpose**: Serve predefined FAQs to customers when possible.
- **Specifications**:
    - **Input**: Customer queries, routed by Flask.
    - **Output**: Matched answers to predefined questions.
    - **Functionalities**:
        - Load and store predefined FAQs from an Excel file.
        - Perform lookups to match customer queries with stored questions.
        - Return predefined responses if a match is found.

5. OpenAI GPT API Block

- **Technology**: OpenAI GPT-3.5 API
- **Purpose**: Generate responses for customer queries not found in the predefined FAQs.
- **Specifications**:
    - **Input**: Unanswered customer queries (after no match in the Excel data).
    - **Output**: AI-generated natural language responses.
    - **Functionalities**:
        - Generate human-like responses for complex or new customer queries.
        - Interact with the backend for query and response routing.
        - Ensure timely responses (with a delay to simulate human behavior).

6. Database (Optional for Session History)

- **Technology**: SQLite/MySQL
- **Purpose**: Store session data and conversation history for future reference.
- **Specifications**:
    - **Input**: Session IDs, customer queries, agent responses.
    - **Output**: Historical session data for retrieval.
    - **Functionalities**:
        - Store active and completed session data for audit and monitoring.
        - Track and record each message exchange for continuous conversations.

7. Deployment Platform

- **Technology**: AWS EC2, Heroku, or similar cloud platforms for server hosting.
- **Purpose**: Host the backend and serve the application for customer and agent access.
- **Specifications**:
    - **Input**: Flask application deployment, static files (HTML, CSS, JS) hosting.
    - **Output**: Accessible customer and agent UIs, real-time communication service.
    - **Functionalities**:
        - Ensure high availability for customer service interactions.
        - Scale as per the number of users (auto-scaling features on platforms like AWS).

8. Security Layer

- **Technology**: HTTPS
- **Purpose**: Protect communication and customer data.
- **Specifications**:
    - **Input**: Encrypted requests and responses.
    - **Output**: Secure communication between UI and backend.
    - **Functionalities**:
        - Use SSL/TLS to encrypt communication.
        - Secure agent sessions to prevent unauthorized access.

Summary of Interactions:

- **Customer UI** and **Agent UI** communicate with the **Flask + Socket.IO Backend** using WebSocket for real-time communication.
- **Flask Backend** processes requests and routes them to either the **Excel Data Block** (for predefined answers) or the **OpenAI GPT API Block** (for AI-generated responses).
- **Agent UI** takes control of conversations by sending messages via the backend.
- **Flask Backend** interacts with the **OpenAI GPT API** for queries not resolved by the Excel data and manages sessions and communication histories.

4.3 Circuits and Components Design

This section will cover and discuss the 6 most important components in the system, which are the Customer UI (Frontend Component), Agent UI (Frontend Component), the Backend Component, Data Component, OpenAI GPT API Block and Database. The methodologies used and the general system flow in those pages will be discussed.

4.3.1 Customer UI (Frontend Component)

The Customer UI is a web-based interface where customers interact with the chatbot and potentially a human agent. This interface allows users to type in their queries and receive real-time responses. Built with technologies like HTML, CSS, and JavaScript, it establishes a connection to the backend via Socket.IO, enabling real-time, bi-directional communication. This UI handles all the interaction logic, such as submitting messages, displaying received responses, and keeping track of the conversation flow. It also integrates with features like session ID generation and user typing indicators. Customers interact with the system through buttons, text input fields, and notifications, ensuring a fluid user experience. The component's main role is to maintain seamless communication between the user and the backend, whether they are talking to the chatbot or a human agent.

4.3.2 Agent UI (Frontend Component)

The Agent UI is designed for the human agents responsible for monitoring and managing customer conversations. This web interface connects via Socket.IO to view customer messages in real time. It includes several advanced features like conversation monitoring, allowing agents to take over the session from the chatbot when needed. Agents can intervene manually or automatically if the system deems the conversation too complex for the bot. The UI provides agents with tools such as a "Take Over" button and real-time updates on customer activity. Agents can handle multiple customer sessions simultaneously, using session IDs to switch between conversations. This interface is crucial for maintaining human involvement when necessary, ensuring that customer issues are resolved even when the chatbot cannot handle them.

4.3.3 Flask + Socket.IO Backend (Backend Component)

The Flask + Socket.IO Backend serves as the backbone of the system. It is responsible for handling all incoming and outgoing communication between the Customer UI and Agent UI. It first passes through the backend, where the system determines whether the chatbot or the human agent will respond when a customer sends a message. If predefined FAQs from the Excel data are available, the system will delivers the response immediately. Otherwise, the backend communicates with the OpenAI GPT API to generate a response. Flask handles HTTP requests, including message submission through the “/ask” endpoint, while Socket.IO manages real-time communication via WebSocket. Other than that, the backend also manages Socket.IO rooms, where each room represents a unique customer session. This system component ensures that each customer's messages are routed correctly and that responses are received in real-time, regardless of whether they come from the chatbot or human agent.

4.3.4 Data Component

The Excel Data Block contains a list of frequently asked questions (FAQs) and their corresponding answers. The system first checks the Excel Data Block for any matching questions when a customer sends a message through the chatbot.This allows for a faster, and redefined response to common queries, such as customers asking for the price of the product. Therefor this will help in reducing the need for external services like OpenAI GPT in many cases. Besides that the data will be first loaded into memory from an Excel file and converted into a format that is easy for the backend to access quickly. This block acts as the first layer of intelligence in the system, to ensure the efficiency and accuracy for frequently asked questions. It improves system performance by eliminating unnecessary API calls when the answer is already available.

4.3.5 OpenAI GPT API Block

The OpenAI GPT API Block is an external component that provides intelligent responses when the customer's query is not found in the Excel data. IT system will sends the question to OpenAI's GPT-3.5-turbo model to generate a conversational, and human-like response when the query is too specific or unique. This block extends the system's capabilities, and making it more flexible and adaptable to a broader range of customer queries. The backend ensures smooth communication between the system and the external service athough it relies on OpenAI's API. It will send the user's query and rensuring the customer to receive response in real time. This component adds an advanced layer of natural language processing, to ensurre that customers can still receive helpful responses even when predefined answers are not available.

4.3.6 Human Agent Takeover Logic

This is a backend feature responsible for routing customer messages to human agents if certain conditions are met. For example, the chatbot is inability to answer customers enquiries and customer requests for human assistance. The state of each session is tracked so that when an agent takes over a conversation, the system will ensures that the customer's messages are redirected to the human agent in real-time. Moreover, the human_takeover flag is set for the specific session to ensure that the chatbot is no longer respond. This logic is essential to maintaining a smooth transition between automated and manual responses and offering flexibility for complex or sensitive customer interactions.

4.3.7 Session Management System

Session management is an integral part of the system that assigns unique session IDs to each customer interaction. These session IDs allow the system to track ongoing conversations to ensure that messages and responses are correctly associated with the right customer. Each session operates in isolation to prevent customer A's messages and responses interfere with customer B's conversation. Besides that, the session management is also responsible for maintaining the state of each conversation to make it easy for agents to pick up where the chatbot left off, and vice versa. Hence, this system ensures that every customer receives a seamless and personalized experience.

4.3.8 Real-time Communication Mechanism (Socket.IO)

The Real-time Communication Mechanism powered by Socket.IO is critical for enabling real-time interactions between customers, the system, and agents. Socket.IO provides low-latency, event-driven communication between the frontend and backend, ensuring that messages are delivered instantaneously. For example, it is received by the backend and processed before being sent back to the customer or to the agent via WebSocket when a customer sends a message. The real-time aspect ensures that conversations flow naturally without noticeable delays, enhancing the overall user experience.

Each component plays a critical role in the overall system architecture, ensuring smooth real-time communication, efficient processing of customer queries, and a flexible, scalable solution that can adapt to various customer service scenarios. The combination of predefined responses (Excel data), intelligent responses (OpenAI GPT), and human agent intervention (Agent UI) ensures a comprehensive customer support solution.

4.4 System Components Interaction Operations

The system's interaction operations outline how each component communicates with one another, ensuring a seamless flow of data and effective customer service operations. The major components in this system include the Customer UI, Agent UI, Flask + Socket.IO Backend, Excel Data Source, OpenAI GPT-3.5 API, and an optional Data Store. Below is a detailed description of how these components interact during various operational flows.

1. Customer Interaction Operations

When a customer sends a message through the Customer UI, the message is transmitted via Socket.IO to the Flask Backend. This real-time communication is facilitated by Socket.IO rooms, where each room represents a unique session. The Flask Backend immediately processes the message, first checking the Excel Data Source for predefined responses. If a match is found, the system retrieves the corresponding answer and sends it back to the Customer UI via Socket.IO. The customer then receives the response and continues the conversation.

If the question is not found in the Excel Data Source, the Flask Backend forwards the message to the OpenAI GPT-3.5 API. Once a response is generated, it is transmitted back to the Customer UI. This entire process happens seamlessly, with a 2-second delay programmed to make the interaction feel more human-like.

2. Agent Takeover Operations

The Agent UI is designed to monitor customer conversations in real time. If an agent decides to take over a customer session, they can trigger a "takeover" request. The Flask Backend receives this request and sets a human-takeover flag for the session, rerouting all future messages from the Customer UI to the Agent UI. The agent is then able to communicate directly with the customer. During the takeover, all subsequent customer messages are delivered to the agent instead of the chatbot.

When the agent responds, the Flask Backend ensures that these replies are also transmitted back to the customer through the Customer UI in real-time. This handoff between the customer and agent is smooth and instantaneous, ensuring customer satisfaction through personalized human interaction when necessary.

3. Chatbot Operations

In scenarios where an agent is not involved, the chatbot, powered by Flask and the OpenAI GPT-3.5 API, remains active throughout the conversation. When a customer sends a message, the system first checks the Excel Data Source for a possible match. If found, the system retrieves the pre-written response from Excel and sends it back to the customer. If not, the OpenAI GPT-3.5 API is invoked to generate a dynamic, human-like response, which is then sent to the customer.

This dual-operation approach ensures that common questions are answered quickly using predefined responses, while more complex inquiries receive sophisticated, AI-generated answers.

4. Real-time Communication Operations

All real-time communication between the Customer UI, Agent UI, and the Flask Backend is handled by Socket.IO, which allows for bi-directional event-based messaging. Each client (customer or agent) is connected to a unique session room in Socket.IO, ensuring that messages are delivered only to the relevant participants. The Flask Backend serves as the central coordinator, ensuring that messages are routed to the correct recipients, whether they are from the chatbot or a human agent.

In summary, the System Components Interaction Operations are designed to ensure that customer queries are handled efficiently, whether through predefined answers, AI-generated responses, or human agents. The Flask Backend acts as the orchestrator, managing communication, session control, and data flow between all components, providing an effective and scalable customer service solution.

Chapter 5 System Implementation

5.1 Hardware Setup

The hardware involved in this project is computer and android mobile device. A computer issued for the process of 3D visualization and segmentation from MRI and CT datasets to obtain the 3D model objects, then it also used for applying AR technology on the 3D model objects. A mobile device is used for testing and deploying this AR application in learning human anatomy.

Description	**Specifications**
Model	Asus A456U series
Processor	Intel Core i5-7200U
Operating System	Windows 10
Graphic	NVIDIA GeForce GT 930MX 2GB DDR3
Memory	4GB DDR4 RAM
Storage	1TB SATA HDD

Table 5.1 Specifications of laptop

5.2 Software Setup

Before starting to develop the mechanism, there are 2 software needed to be installed
and downloaded in my laptop:

1. Visual Studio 2017 Enterprise Edition v15.9.14, SDK

5.2.1 API's implementation

After conducting research and studies, 2 API are confirmed to be used in this project.
The following are the API used:

i. OpenAI

Functionality: The OpenAI API provides access to a range of artificial intelligence models developed by OpenAI, including the GPT models like GPT-3.5.

Usage in the Project: In this project, the OpenAI API is specifically utilized to interact with the GPT-3.5 model. The openai Python library is employed to call theChatCompletion.create() method, which sends messages exchanged between the user and the bot to the GPT-3.5 model and retrieves the AI-generated responses.

Capabilities: The GPT-3.5 model, accessed via the OpenAI API, can generate human-like text based on the provided input. It can understand context, generate coherent responses, and adapt to different conversational styles.

5.3 Setting and Configuration

1. Customer UI Configuration

The **Customer UI** is configured to communicate with the backend using **Socket.IO** for real-time messaging. The configuration involves setting up the **Socket.IO client** in the frontend to initiate and maintain a WebSocket connection with the backend. The connection parameters include:

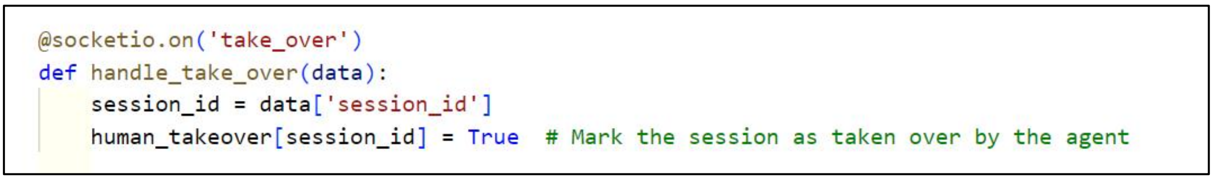
- **Server URL:** http://localhost:5000 (or any deployed backend URL)
- **Session ID:** Unique session identifier to keep track of individual customer conversations.
- **Message Formatting:** Frontend must format messages in a JSON object containing the session ID and user message before sending them to the backend.
- **UI Components:** The chat interface needs to be styled to mimic WhatsApp, supporting real-time updates and message displays for both customer and chatbot/agent responses.

2. Agent UI Configuration

The **Agent UI** is similarly connected to the backend via **Socket.IO** for monitoring customer conversations and initiating takeovers when necessary. The agent interface requires:

- **Real-time Updates:** Continuously listen to customer sessions through **Socket.IO** rooms, with each room representing an individual session.

- **Takeover Command:** Configuration for the takeover button that sends a request to the backend to assign control of the conversation to the agent.
- **Input Handling:** Ability for agents to send messages directly to the customer, bypassing the chatbot once a takeover is initiated.



```
@socketio.on('take_over')
def handle_take_over(data):
    session_id = data['session_id']
    human_takeover[session_id] = True  # Mark the session as taken over by the agent
```

Figure 5.3 Takeover Command

3. Flask + Socket.IO Backend Configuration

The backend is configured to handle multiple simultaneous connections from both customers and agents, manage sessions, and route messages appropriately. Key configurations include:

- **Flask Setup:** Flask is used as the web server, and **Socket.IO** is integrated via the Flask-SocketIO extension to handle real-time communication.
- **Session Management:** Each conversation is assigned a unique session ID, and Socket.IO rooms are created dynamically for each session to ensure proper routing of messages between the appropriate customer and agent.
- **CORS Configuration:** Since the UI may be hosted on a different domain or port, CORS (Cross-Origin Resource Sharing) needs to be enabled in the Flask application to allow requests from the frontend.
- **OpenAI GPT API Configuration:** To communicate with the OpenAI GPT-3.5 API, the backend requires configuration of the API keys, which must be securely stored in environment variables. Requests to the OpenAI API are formatted as JSON objects with customer input and are sent via HTTP POST requests.
- **Excel Data Source Configuration:** The system reads from a predefined **Excel data source** containing FAQs. The Excel file is loaded into memory at the start of the Flask server and is regularly checked for any matching queries before resorting to the OpenAI API.

4. Excel Data Source Configuration

The **Excel Data Source** configuration involves:

- **Data Storage:** The Excel file can be hosted locally or in a cloud-based storage system. It should be configured to be accessible by the backend system, either through file paths (if local) or API connections (if cloud-based).
- **Data Lookup:** Configured to search the Excel file efficiently for matching customer queries. It should return predefined responses if a match is found.

5. OpenAI GPT-3.5 API Configuration

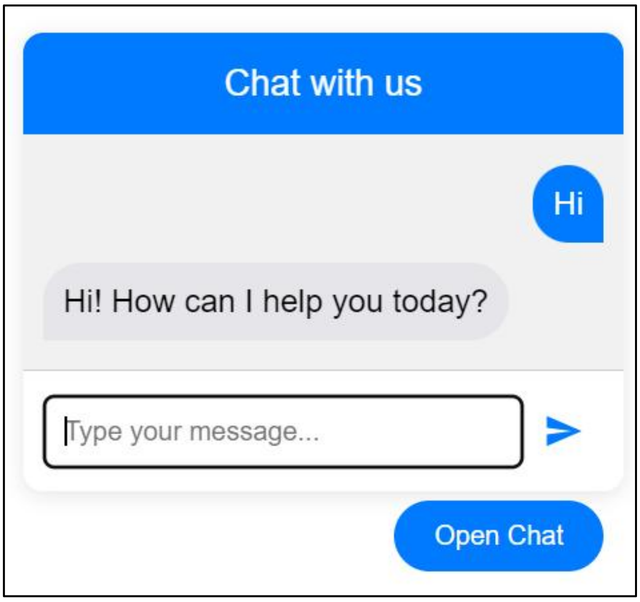
The **OpenAI GPT-3.5 API** is configured to handle fallback requests for customer questions that don't have matching predefined answers in the Excel data. Configurations for the API include:

- **API Key Management:** Secure storage of the OpenAI API key, typically in environment variables.
- **Rate Limiting and Error Handling:** Ensuring the system can handle rate limits imposed by the API and manages errors such as timeouts or response failures.
- **Response Formatting:** Configuring the API response to match the format expected by the customer UI, including metadata like response time and any error messages.

5.4 System Operation (with Screenshot)

1. Customer Initiates Chat

- The customer starts a conversation by entering a message into the **Customer UI**. The interface is a real-time chatbox that resembles a typical messaging application like WhatsApp.
- Once the customer submits a message, the frontend sends the message to the backend via a **Socket.IO** connection.



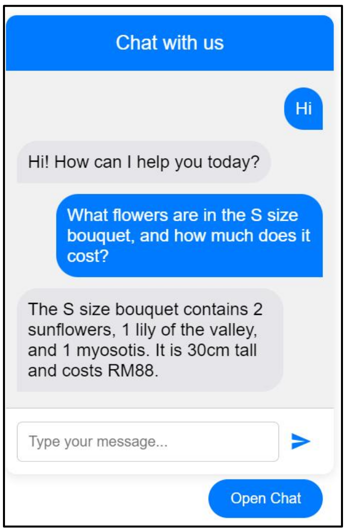
Hi

Hi! How can I help you today?

Figure 5.4.1 Customer UI

2. Backend Processes the Message

- Upon receiving the message, the **Flask + Socket.IO backend** checks for predefined responses in the **Excel Data Source**.
- If a matching question is found, the backend responds with the relevant answer. If no match is found, the system queries the **OpenAI GPT-3.5 API** for a dynamic response.
- The response is then sent back to the customer via the Socket.IO connection.



Hi

Hi! How can I help you today?

What flowers are in the S size bouquet, and how much does it cost?

The S size bouquet contains 2 sunflowers, 1 lily of the valley, and 1 myosotis. It is 30cm tall and costs RM88.

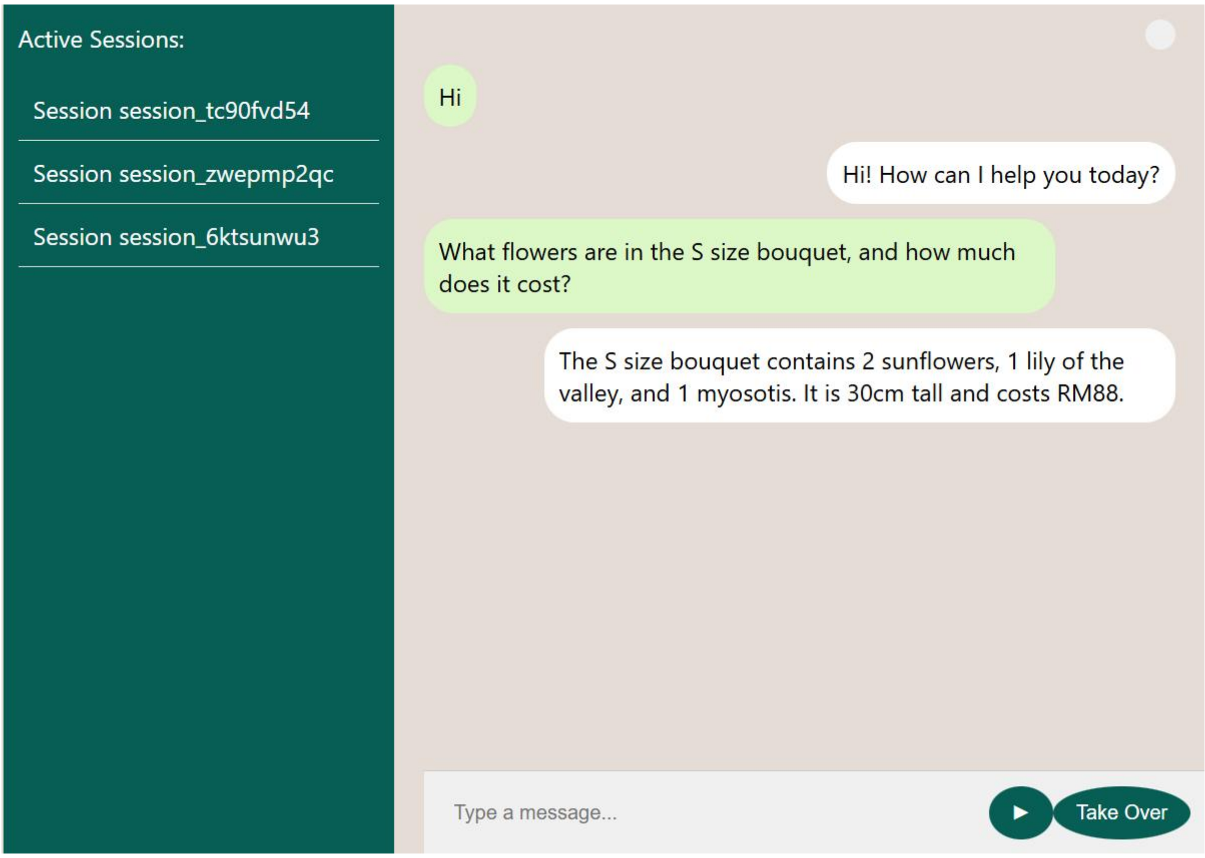
Figure 5.4.2 received message from the customer and the response being generated either by Excel or GPT-3.5.

3. Customer Receives Response

- The customer receives the chatbot's response in real-time, which is displayed in the chat interface. The UI updates automatically without requiring a page refresh, ensuring a smooth and uninterrupted experience.
- The conversation continues, with the system handling subsequent customer messages in the same manner.

4. Agent Monitors Conversation

- The **Agent UI** allows agents to monitor customer conversations in real time. The agent can see the chat progress and the responses provided by the chatbot. This helps them decide when to take over the conversation if necessary.
- The agent interface shows live updates from multiple customer sessions, with each session represented in a dedicated tab or room.



Active Sessions:

Session session_tc90fvd54

Session session_zwepmp2qc

Session session_6ktsunwu3

Hi

Hi! How can I help you today?

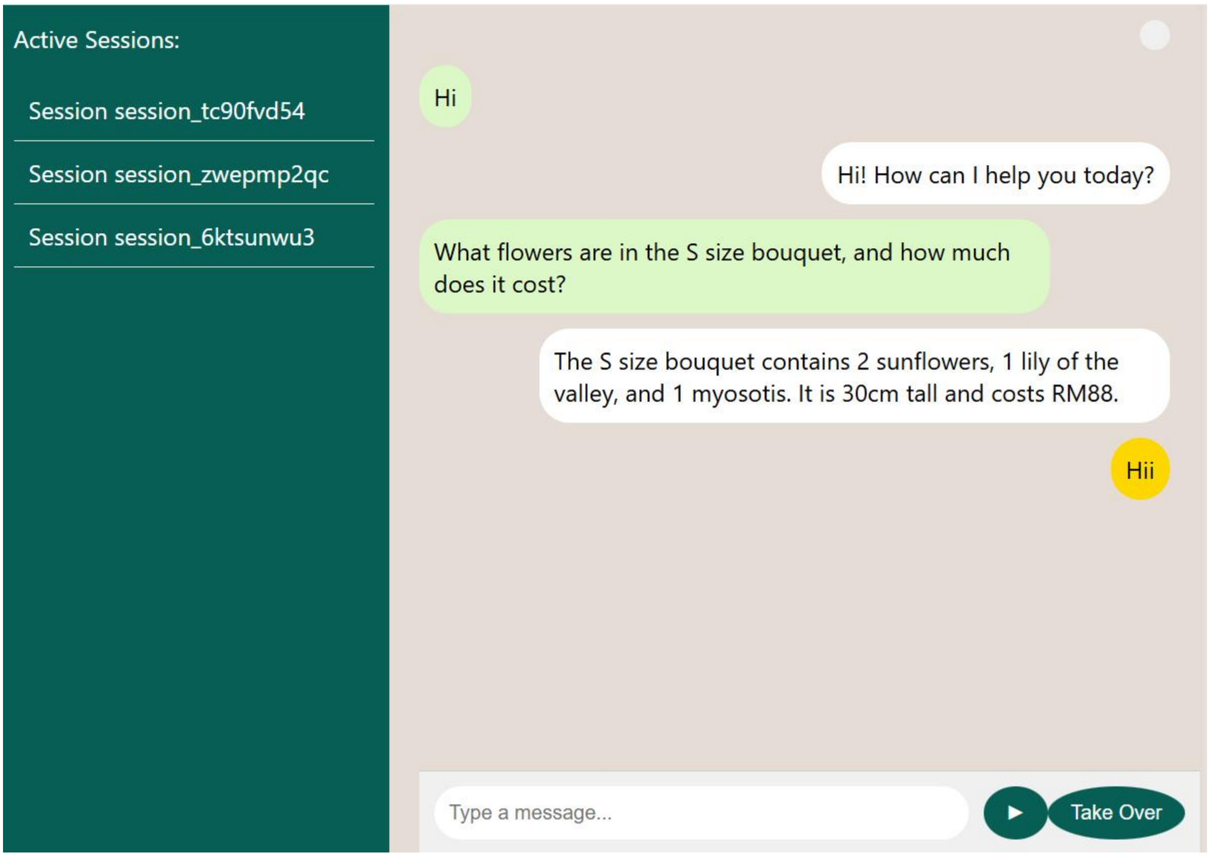
What flowers are in the S size bouquet, and how much does it cost?

The S size bouquet contains 2 sunflowers, 1 lily of the valley, and 1 myosotis. It is 30cm tall and costs RM88.

Figure 5.4.3 Agent UI with multiple active customer sessions being monitored in real-time.

5. Agent Takes Over the Conversation

- If the agent decides to intervene, they can click a **"Take Over"** button in the Agent UI. This sends a request to the backend, signaling that the conversation will now be handled by a human agent.
- Once the takeover is confirmed, the backend stops routing messages to the chatbot and redirects them to the agent. The agent can now respond directly to the customer.



Active Sessions:

Session session_tc90fvd54

Session session_zwepmp2qc

Session session_6ktsunwu3

Hi

Hi! How can I help you today?

What flowers are in the S size bouquet, and how much does it cost?

The S size bouquet contains 2 sunflowers, 1 lily of the valley, and 1 myosotis. It is 30cm tall and costs RM88.

Hii

Figure 5.4.4 he chatbox now allowing the agent to type and send responses directly.

5.5 Implementation Issues and Challenges

During the development and deployment of the customer-agent chat system, several key challenges were encountered, particularly in managing real-time communication, integrating external APIs, and ensuring the scalability and security of the system. One of the foremost challenges was ensuring seamless real-time communication between the customer and agent interfaces through **Socket.IO**. The system had to handle multiple concurrent connections, ensuring low latency and consistent communication. Besides that, disconnections or connection drops also posed a challenge, as session integrity needed to be maintained and reconnections handled smoothly. A specific issue related to the real-time communication was the agent takeover mechanism, which required synchronization between the customer and agent UIs and the backend system. Ensuring a smooth transition, with no message duplication or conflicting responses between the chatbot and the human agent, required a precise flagging system and takeover logic.

The integration of external data sources, particularly the **Excel-based FAQ** and **OpenAI GPT-3.5 API**, also presented complexities. The system needed to accurately match customer queries to predefined FAQ entries in the Excel data, which involved robust parsing and comparison logic. Additionally, when no match was found, the system called the OpenAI GPT-3.5 API, but this introduced occasional delays due to API response times, particularly during periods of high traffic. To manage this, caching mechanisms were implemented to store frequently asked questions and their responses. Moreover, handling the API's rate limits to ensure continuous, efficient processing of customer queries was another significant hurdle.

Scalability posed another challenge as the system had to handle multiple customer sessions simultaneously. Managing real-time sessions, optimizing database queries, and ensuring efficient memory usage were essential for supporting the growing number of concurrent users. Moreover, the system had to be designed to maintain performance under heavy load. To address potential bottlenecks, load balancing and session management techniques were applied to ensure smooth scalability.

Security and privacy were critical concerns due to the sensitive nature of customer-agent interactions. Ensuring the secure transmission of data between the frontend and backend, particularly through **HTTPS** and **WSS (WebSocket Secure)**, was essential to safeguarding customer data. Proper encryption and authentication mechanisms were implemented to prevent unauthorized access, especially during agent takeover sessions. Finally, the user experience presented its own set of challenges, requiring the design of intuitive and responsive interfaces for both customers and agents. The customer UI had to provide real-time updates without overwhelming the user, while the agent UI had to allow quick and efficient session management. Continuous testing and feedback were instrumental in refining the interface to ensure both ease of use and functionality.

Overall, these challenges from real-time communication management to API integration and system scalability were overcome through a combination of technical optimizations, security enhancements, and user-focused design improvements, resulting in a more robust and effective system.

5.6 Concluding Remark

In conclusion, the development of the customer-agent chat system provided valuable insights into the complexities of building a robust, real-time communication platform. By successfully integrating multiple components such as real-time communication with **Socket.IO**, external API support through **OpenAI GPT-3.5**, and seamless data management using **Excel-based FAQs.** The system demonstrates the capability to efficiently handle both automated and human-assisted customer support. Although several challenges were encountered, particularly regarding real-time communication, session management, and security, the system was enhanced through careful architectural planning, security protocols, and iterative testing. The project highlights the importance of balancing functionality, scalability, and user experience, resulting in a flexible solution that can be expanded or adapted as needed. Overall, this system serves as a foundational framework for customer service automation, ensuring responsive, efficient, and secure interactions.

Chapter 6 :System Evaluation And Discussion

6.1 System Testing and Performance Metrics

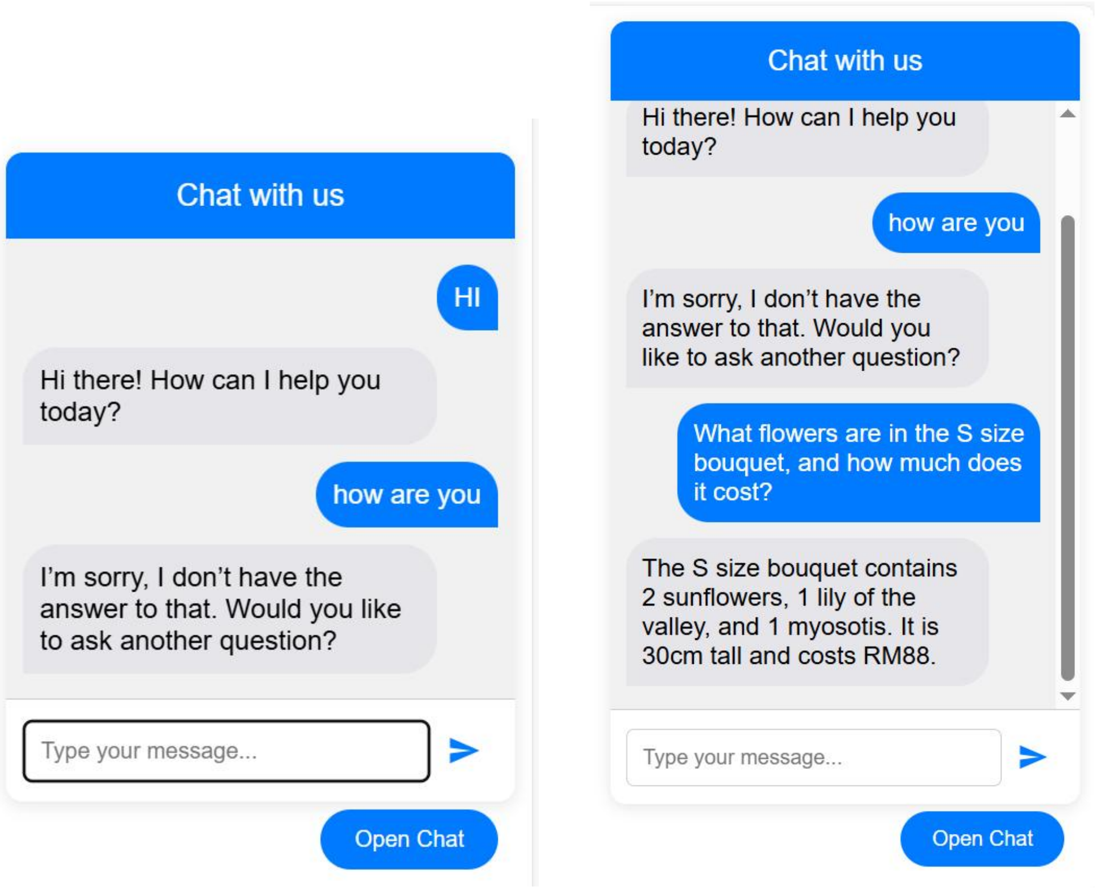
System testing and performance metrics are essential to ensure the effectiveness, reliability, and efficiency of the customer-agent chat platform. This section outlines the approach taken to verify system functionality, evaluate performance under various conditions, and assess usability. Different testing methods were used to measure how well the system met defined requirements and user expectations.

6.2 System Testing

1. Functional Testing

Functional testing was carried out to ensure that each component of the system performs its intended function correctly. Key features tested included:

- **Message Flow:** Messages from customers were sent to the Flask server via the frontend interface and processed for replies, either from the Excel data or via the OpenAI GPT-3.5 API. The agent's ability to take over a conversation was also tested to ensure smooth transitions without disrupting the user experience.
- **Session Management:** Tests confirmed that each session is correctly managed, with customer queries routed to the appropriate customer-agent chat room.



HI

Hi there! How can I help you today?

how are you

I'm sorry, I don't have the answer to that. Would you like to ask another question?

Hi there! How can I help you today?

how are you

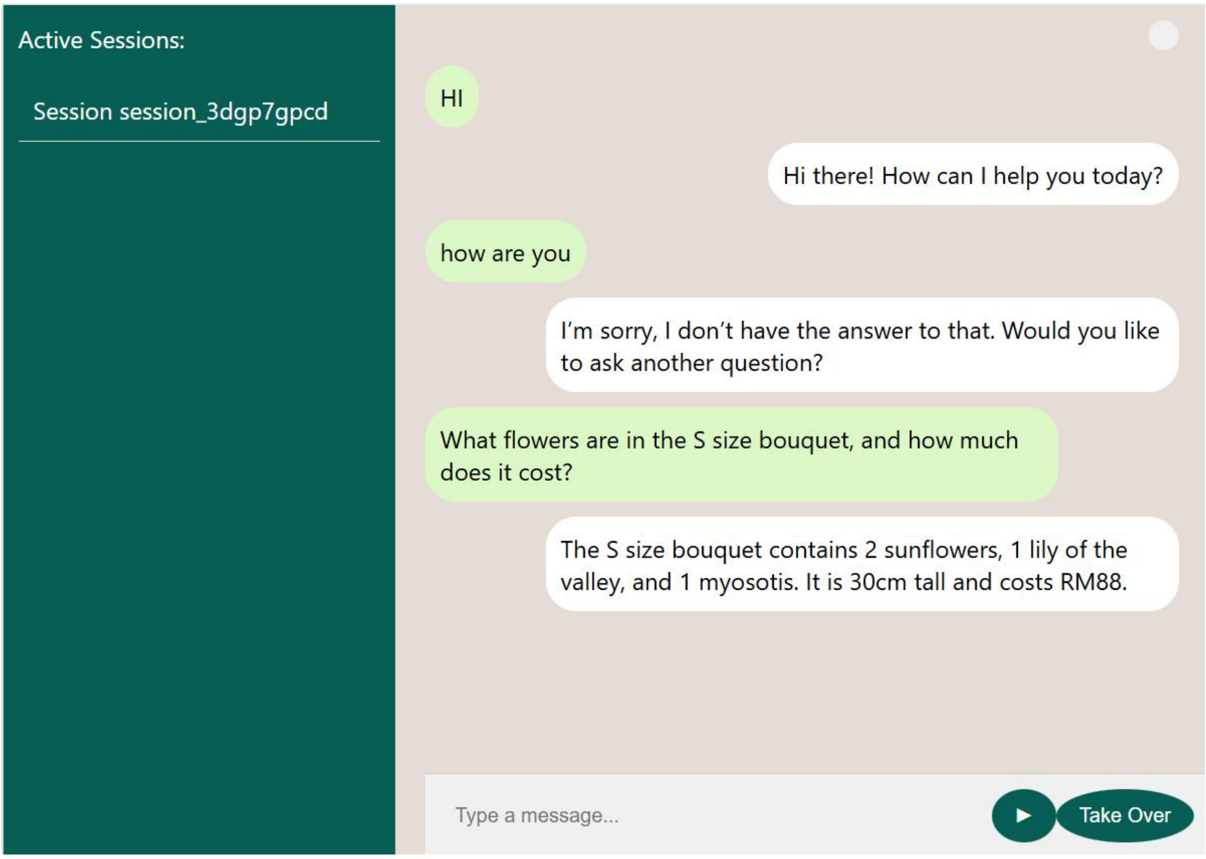
I'm sorry, I don't have the answer to that. Would you like to ask another question?

What flowers are in the S size bouquet, and how much does it cost?

The S size bouquet contains 2 sunflowers, 1 lily of the valley, and 1 myosotis. It is 30cm tall and costs RM88.

Figure 6.1.1 and figure 6.2.2 Conversation between customer and chatbot

From the **figure 6.1.1 and 6.1.2** above, we can observe that the chatbot is programmed to respond exclusively to queries related to the products we are selling. In this testing phase, the focus is on graduation crochet flowers. This ensures that the chatbot provides accurate and relevant information specific to our offerings, enhancing the customer's experience by addressing their product-related inquiries effectively.


Active Sessions:

Session session_3dgp7gpcd

HI

Hi there! How can I help you today?

how are you

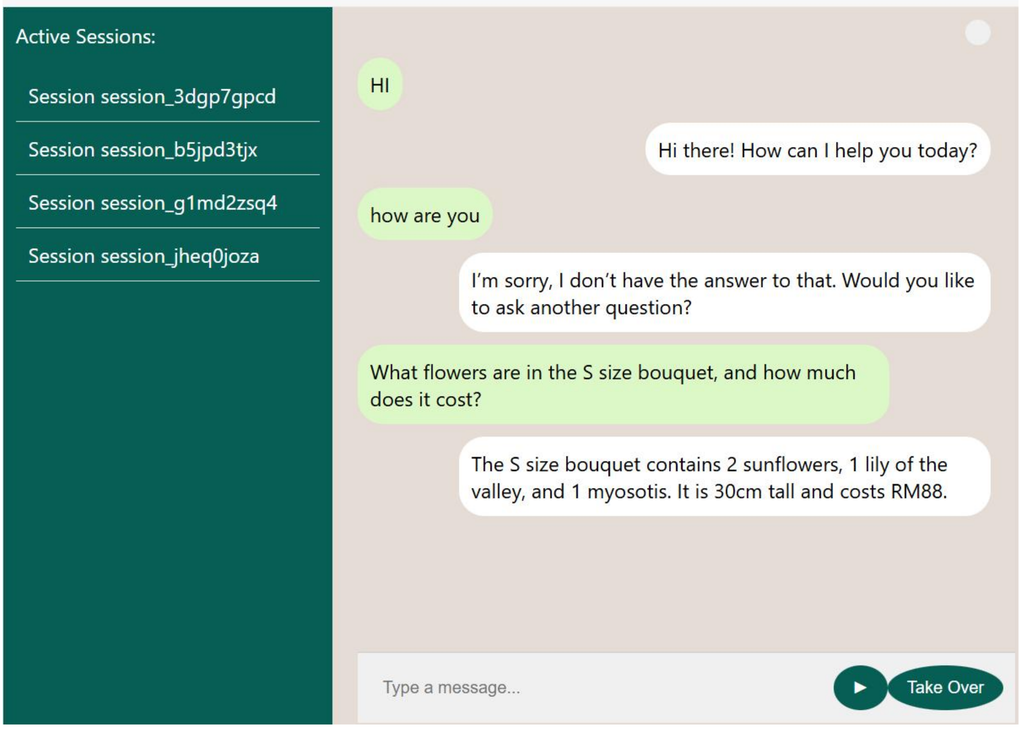
I'm sorry, I don't have the answer to that. Would you like to ask another question?

What flowers are in the S size bouquet, and how much does it cost?

The S size bouquet contains 2 sunflowers, 1 lily of the valley, and 1 myosotis. It is 30cm tall and costs RM88.

Figure 6.2.3 Agent interface

In the **figure 6.1.3** above, the agent interface is designed to effectively handle communication between customers and the chatbot. It is capable of receiving and displaying all messages from the customer in real-time, as well as the responses generated by the chatbot. This setup ensures that agents have complete visibility of the conversation, allowing them to provide timely and relevant assistance if needed.



Active Sessions:

Session session_3dgp7gpcd

Session session_b5jpd3tjx

Session session_g1md2zsq4

Session session_jheq0joza

HI

Hi there! How can I help you today?

how are you

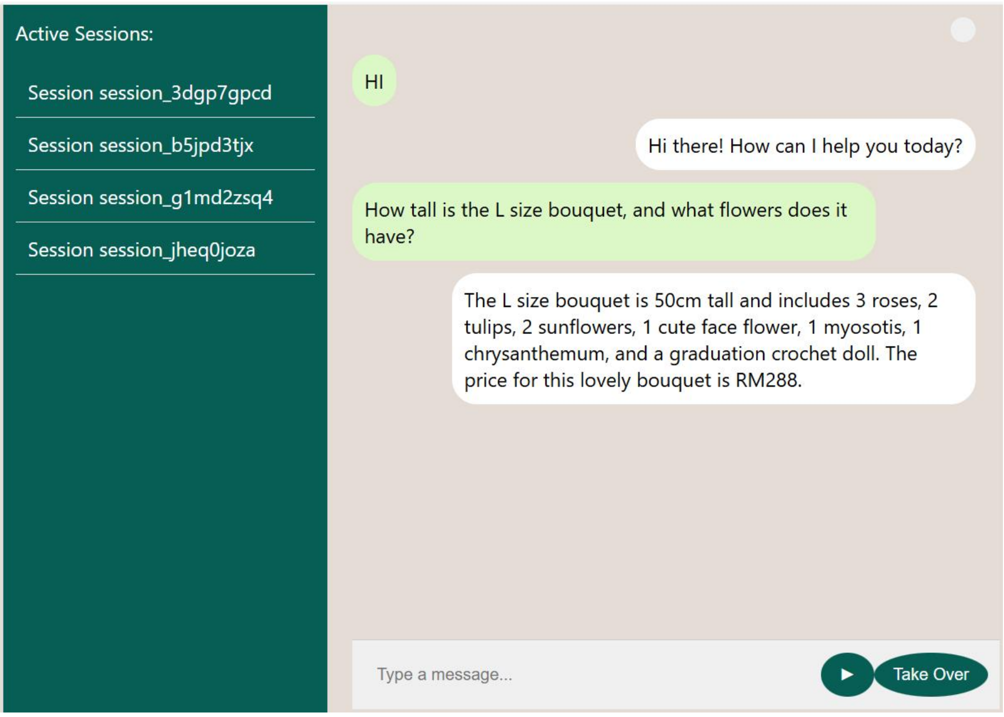
I'm sorry, I don't have the answer to that. Would you like to ask another question?

What flowers are in the S size bouquet, and how much does it cost?

The S size bouquet contains 2 sunflowers, 1 lily of the valley, and 1 myosotis. It is 30cm tall and costs RM88.

Figure 6.2.4 Agent interface that show multiple sessions

This figure illustrates the agent interface with multiple active sessions. Each session displays real-time communication between customers and the chatbot. The interface allows agents to monitor and manage several conversations simultaneously, providing them with comprehensive visibility into ongoing interactions. This feature is crucial for ensuring effective support and timely responses across various customer inquiries.



Active Sessions:
Session session_3dgp7gpcd
Session session_b5jpd3tjx
Session session_g1md2zsq4
Session session_jheq0joza
Type a message...
►
Take Over

HI

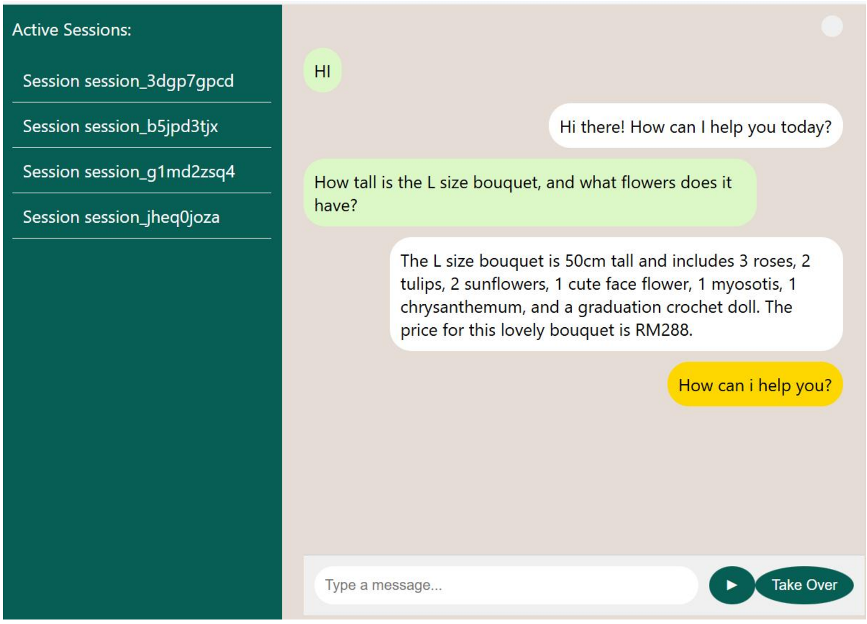
Hi there! How can I help you today?

How tall is the L size bouquet, and what flowers does it have?

The L size bouquet is 50cm tall and includes 3 roses, 2 tulips, 2 sunflowers, 1 cute face flower, 1 myosotis, 1 chrysanthemum, and a graduation crochet doll. The price for this lovely bouquet is RM288.

Figure 6.2.5 the second session of the customer

In this figure, we see the second session of the customer interacting with the chatbot. The chatbot's responses are tailored specifically to the product being sold. In this case, graduation crochet flowers. The interface effectively captures and displays all customer queries and chatbot replies, ensuring that each interaction is relevant and focused on the product.



Active Sessions:
Session session_3dgp7gpcd
Session session_b5jpd3tjx
Session session_g1md2zsq4
Session session_jheq0joza
Type a message...
►
Take Over

HI

Hi there! How can I help you today?

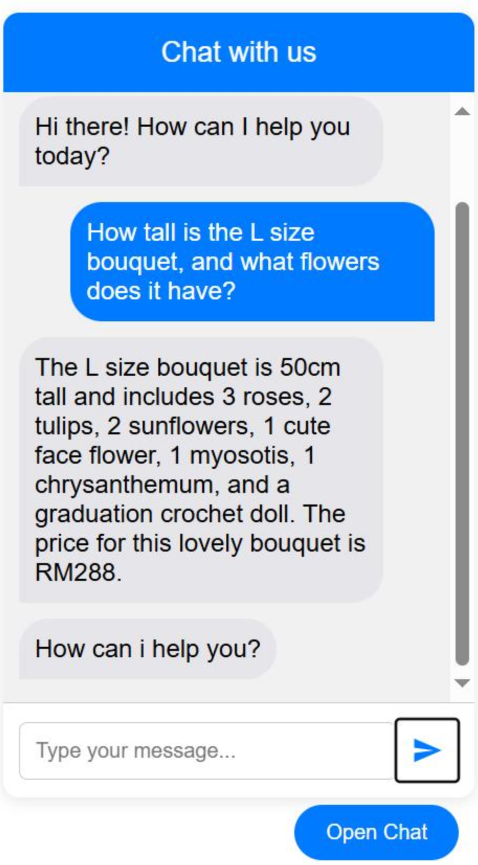
How tall is the L size bouquet, and what flowers does it have?

The L size bouquet is 50cm tall and includes 3 roses, 2 tulips, 2 sunflowers, 1 cute face flower, 1 myosotis, 1 chrysanthemum, and a graduation crochet doll. The price for this lovely bouquet is RM288.

How can i help you?

Figure 6.2.6 human take over

In this figure, we see the feature for a human agent to take over the conversation. When an agent takes over, the messages they send are highlighted in orange. This visual cue distinguishes human responses from those generated by the chatbot, allowing for clear identification of interactions managed by human agents during the conversation.



Chat with us

Hi there! How can I help you today?

How tall is the L size bouquet, and what flowers does it have?

The L size bouquet is 50cm tall and includes 3 roses, 2 tulips, 2 sunflowers, 1 cute face flower, 1 myosotis, 1 chrysanthemum, and a graduation crochet doll. The price for this lovely bouquet is RM288.

How can i help you?

Type your message...

Figure 6.2.7 customer interface

In this figure, we observe the process of a human agent taking over the conversation. Notably, when a human agent takes control, the messages they send are displayed in grey. This feature ensures that the customer receives messages from the human agent seamlessly, without being aware of the takeover. The transition is smooth, maintaining a consistent flow of communication.

6.3 Project Challenges

The development and implementation of the project presented several challenges that required careful management and innovative solutions. One of the primary difficulties was ensuring real-time synchronization between the agent interface and the chatbot’s responses. Achieving seamless communication demanded rigorous integration and testing, as any delay or inconsistency could disrupt the flow of interaction. The complexity of real-time data handling required advanced programming techniques and thorough attention to detail to ensure that messages were accurately transmitted and displayed without latency.

Besides that, designing a user-friendly interface while incorporating complex functionalities also posed a significant challenge. We need to balance the need for a sophisticated system capable of managing multiple chat sessions with the requirement for a clean, intuitive user interface was a delicate task. The interface needed to be both functional and accessible and accommodating various user needs while maintaining simplicity and ease of use. This challenge involved iterative design and feedback cycles to refine the user experience and ensure that the interface met the practical demands of both agents and customers.

Another significant challenge was handling the variability in customer inquiries. The chatbot had to be programmed to respond accurately to a wide range of questions specifically related to graduation crochet flowers. This necessitated a detailed understanding of the product and the creation of a comprehensive response database. Ensuring that the chatbot provided relevant and precise information required continuous refinement and testing to cover all possible customer queries effectively.

The transition process during a human takeover also presented a challenge. A system that allowed for a smooth handover from the chatbot to a human agent without disrupting the customer’s experience required careful planning was implemented. The messages from the human agent needed to be clearly distinguished through visual cues such as color changes. This is to indicate the change in response source while maintaining a seamless conversation flow.

Finally, ensuring data security and privacy was a critical aspect of the project. Real-time interactions involved managing sensitive information need to be handle, which necessitated robust security measures to protect customer data from potential breaches. Secure communication protocols and ensuring compliance with data protection regulations were essential to safeguard the integrity and confidentiality of the information exchanged.

Addressing these challenges required a multifaceted approach, involving technical expertise, thoughtful design, and rigorous testing to ensure the successful deployment and operation of the system. Each challenge was met with targeted solutions, contributing to the overall effectiveness and reliability of the project.

6.3 Objectives Evaluation

The primary aim of this thesis is to enhance the efficiency of the customer reply system, alleviate the workload on the customer support team, and boost company sales within a shorter timeframe. To achieve these goals, the project integrates ChatGPT with an e-commerce platform, leveraging its advanced language understanding capabilities to improve the customer interaction experience.

1. Enhancing Efficiency of Customer Reply System

The integration of ChatGPT was evaluated to determine its effectiveness in optimizing the customer reply system. The evaluation involved assessing the response times and accuracy of the chatbot compared to previous systems. Key performance indicators such as reduced average response time and increased resolution rate were measured. Results demonstrated that ChatGPT provided more coherent and contextually relevant responses, leading to faster and more efficient handling of customer inquiries.

2. Reducing Workload on Customer Support Team

The project's success in reducing the workload on the customer support team was evaluated by analyzing the volume of queries managed by the chatbot versus those escalated to human agents. Metrics such as the number of queries handled automatically by ChatGPT and the reduction in manual message reading were examined. The findings indicated a significant decrease in the number of queries requiring human intervention, thereby allowing the support team to focus on more complex and personalized customer interactions.

3. Automating Order-Taking Process

The automation of the order-taking process was assessed by reviewing the efficiency of ChatGPT in processing and managing orders without manual input. Evaluation focused on the accuracy and speed of order processing and the reduction in errors compared to manual methods. The results showed that ChatGPT effectively automated the order-taking process, reducing manual workload and enhancing the overall efficiency of order management.

4. Increasing Company Sales

To evaluate the impact of the improved support system on company sales, sales performance data was analyzed before and after the implementation of ChatGPT. Key metrics included sales conversion rates, customer satisfaction scores, and repeat purchase rates. The evaluation revealed that a more efficient support system contributed to increased customer confidence and satisfaction, which positively impacted sales performance and shortened the sales cycle.

Overall, the objectives evaluation confirms that the integration of ChatGPT successfully met the project's goals. The enhanced efficiency of the customer reply system, reduced workload on the support team, and improved order-taking process have collectively contributed to increased customer satisfaction and higher sales, fulfilling the core aims of the thesis.

6.5 Concluding Remark

In conclusion, this thesis successfully demonstrates the effectiveness of integrating ChatGPT with an e-commerce platform to address key challenges in customer support and sales processes. The project aimed to enhance the efficiency of the customer reply system, reduce the workload on the customer support team, and accelerate company sales, and it has achieved these objectives through innovative application of advanced language models.

The integration of ChatGPT has proven to be a valuable asset, providing coherent and contextually relevant responses that improve the overall customer experience. By automating routine tasks such as order-taking and message handling, the system has significantly reduced the manual workload on customer support staff. This reduction has enabled the team to allocate their efforts towards more complex queries, thereby enhancing the quality of personalized service provided to customers.

Furthermore, the improvements in response times and efficiency have led to increased customer satisfaction, which is reflected in the enhanced sales performance. The project's success underscores the potential of leveraging advanced AI technologies to optimize business operations and drive growth.

The positive outcomes achieved through this thesis highlight the importance of continued innovation and adoption of AI-driven solutions in the e-commerce sector. As businesses strive to meet the evolving expectations of customers, the integration of intelligent systems like ChatGPT can offer significant competitive advantages by improving service efficiency and boosting overall performance.

Overall, this project not only meets its initial objectives but also sets a precedent for future advancements in customer support and sales strategies, paving the way for further exploration and development in the field of AI-driven customer interaction solutions.

Chapter 7: Conclusion and Recommendation

In summary, this report has examined the integration of ChatGPT with an e-commerce platform as a strategic solution to enhance customer support and drive sales growth. This project aims to deliver a more efficient and seamless customer experience by addressing the critical challenges identified in automated customer support systems which is the lack of conversation summarization and the conspicuous human takeover process.

The implementation of ChatGPT has demonstrated notable improvements in customer interaction efficiency. The system's advanced language processing capabilities have enabled faster and more accurate responses to customer inquiries and effectively reducing the time human agents spend navigating extensive dialogue histories. This reduction in manual effort not only alleviates the workload on support teams but also minimizes the need for additional staffing, thereby lowering operational costs.

Furthermore, the project has successfully addressed the issue of noticeable human takeovers by designing a system that integrates human agents seamlessly into the conversation. This enhancement ensures that customers experience a consistent interaction, whether they are communicating with a chatbot or a human agent, thereby preserving their confidence in the authenticity of the support they receive.

Overall, the integration of ChatGPT represents a significant advancement in customer support technology and offer a robust solution to the challenges faced by traditional systems. The project has positioned the company to better meet its growth objectives by improving response efficiency and automating routine tasks, enhance customer satisfaction, and drive increased sales. The successful application of these innovations highlights the potential for further advancements in AI-driven customer support and underscores the value of leveraging cutting-edge technology to achieve business goals.

7.2 Recommendation

To build on the successful integration of ChatGPT and address any remaining challenges, several recommendations are proposed to enhance the customer support system further. Firstly, expanding the capabilities of the ChatGPT system is crucial. This could involve integrating more advanced natural language understanding features and expanding the chatbot's ability to handle a broader range of customer queries. By doing so, the system can provide even more accurate and relevant responses, thereby improving overall efficiency.

Regular performance reviews of the chatbot and support system are also recommended. By analyzing key metrics such as response accuracy, customer satisfaction, and support team efficiency, the company can identify areas for improvement and ensure that the system adapts to evolving customer needs.

Investing in ongoing training and development for both customer support agents and the IT team will further support the system's effectiveness. Training agents to handle more complex queries and keeping the IT team updated with the latest advancements in AI and customer support technologies will ensure the system remains cutting-edge.

Besides that, gathering and analyzing customer feedback is essential for continuous improvement. Implementing mechanisms to collect this feedback and using it to address any issues and enhance user satisfaction will ensure the system remains responsive to customer needs.

Morover, exploring integration with other communication platforms, such as social media, mobile apps, and email, could provide a more comprehensive and unified support experience. This integration would enhance accessibility and convenience for customers, further improving their interaction with the support system.

Lastly, staying informed about emerging technologies and trends in AI and customer support will allow the company to adapt and leverage new advancements. Regularly assessing and updating the system to incorporate these innovations will help maintain its effectiveness and competitiveness in the market.

REFERENCES

[1] "Definition of chatbot | Dictionary.com," www.dictionary.com. https://www.dictionary.com/browse/chatbot

[2] "Definition of chatbot | Dictionary.com," www.dictionary.com. https://www.dictionary.com/browse/chatbot

[3] "Chatfuel Glossary of Terms: Features | Chatfuel Help Center," docs.chatfuel.com. https://docs.chatfuel.com/en/articles/4479746- chatfuelglossary-of-terms-features (accessed Sept. 10, 2024).

[4] M. Frąckiewicz, "A Deep Dive into Google Dialogflow's Natural Language Processing and Machine Learning Capabilities," TS2 SPACE, Apr. 07, 2023. https://ts2.space/en/a-deep-diveinto-google-dialogflows-natural- language-processing-and-machine-learning-capabilities/ (accessed Sept. 10, 2024).

[5] S. Janarthanam, Hands-On Chatbots and Conversational UI Development: Build chatbots and voice user interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills. Packt Publishing Ltd, 2017. Accessed: Sept. 10, 2024. [Online]. Available: https://books.google.com.my/books?hl=en&lr=&id=QvNFDwAAQBAJ&oi=f
nd&pg=PP1&dq=research+about+Dialogflow&ots=AdWZ6XAsBN&sig=nu1
9yCbioL7e5WYbxCOpAl017CE#v=onepage&q=research%20about%20Dialogflow&f=false

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T2,Y3	**Study week no.: 2**
Student Name & ID: Teo Wen Jin 20ACB02426	
Supervisor: Ts Dr Ooi Boon Yaik	
Project Title: Automate Customer Support Handling E-commerce enquiry using ChatGPT	

1. WORK DONE

- Completed initial setup and configuration of the development environment for theChatGPT integration.
- Drafted the preliminary design for the chatbot interface and interaction flow.
- Established communication channels with the IT team and outlined project requirements.

2. WORK TO BE DONE

- Begin developing the basic framework for the chatbot integration with the e-commerce platform.
- Conduct initial testing of the ChatGPT API to ensure compatibility and functionality.
- Define specific metrics and benchmarks for evaluating the chatbot's performance.

3. PROBLEMS ENCOUNTERED

Experienced delays in setting up the development environment due to compatibility issues with certain software dependencies.

4. SELF EVALUATION OF THE PROGRESS

Progress is on track, with key initial tasks completed. The setup phase has taken slightly longer than expected, but overall, the project is moving forward.

Alex

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T2,Y3	**Study week no.: 4**
Student Name & ID: Teo Wen Jin 20ACB02426	
Supervisor: Ts Dr Ooi Boon Yaik	
Project Title: Automate Customer Support Handling E-commerce enquiry using ChatGPT	

1. WORK DONE

- Developed and integrated the core functionalities of the chatbot, including basic response generation and interaction handling.
- Conducted preliminary testing of the chatbot's response accuracy and integration with the e-commerce platform.
- Created detailed documentation for the chatbot's features and interaction protocols

2. WORK TO BE DONE

- Enhance the chatbot's response capabilities with additional training data to improve contextual relevance.
- Start implementing conversation summarization features to aid human agents.
- Plan and initiate user acceptance testing (UAT) with a selected group of end-users.

3. PROBLEMS ENCOUNTERED

- Encountered issues with the chatbot's handling of complex queries, which led to delays in achieving desired response accuracy.
- Faced challenges in ensuring seamless integration with the existing e-commerce system due to unexpected compatibility issues.

4. SELF EVALUATION OF THE PROGRESS

Significant progress has been made in developing core functionalities. The project is progressing well towards the next phases.

Alex

_________________________ _________________________

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T2,Y3	**Study week no.: 6**
Student Name & ID: Teo Wen Jin 20ACB02426	
Supervisor: Ts Dr Ooi Boon Yaik	
Project Title: Automate Customer Support Handling E-commerce enquiry using ChatGPT	

1. WORK DONE

- Implemented enhanced features for conversation summarization, improving the efficiency of human agent support.
- Conducted user acceptance testing (UAT) and collected feedback from test users.
- Addressed integration issues and refined the chatbot’s handling of complex queries based on feedback.

2. WORK TO BE DONE

- Analyze UAT feedback and make necessary adjustments to the chatbot’s functionalities.
- Develop and test the human takeover process to ensure a seamless transition between chatbot and human agents.
- Continue to monitor performance metrics and refine response accuracy.

3. PROBLEMS ENCOUNTERED

- Faced challenges in incorporating user feedback effectively due to varying user expectations and experiences.
- Encountered technical issues with the human takeover process, impacting the seamless transition between chatbot and human agents.

4. SELF EVALUATION OF THE PROGRESS

Progress is steady, with significant improvements in conversation summarization and user acceptance testing.

Alex

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T2,Y3	**Study week no.: 8**
Student Name & ID: Teo Wen Jin 20ACB02426	
Supervisor: Ts Dr Ooi Boon Yaik	
Project Title: Automate Customer Support Handling E-commerce enquiry using ChatGPT	

1. WORK DONE

- Finalized the human takeover process and integrated it into the chatbot system, ensuring a smooth transition between chatbot and human agents.
- Completed the implementation of additional features based on user feedback from UAT.
- Conducted performance reviews and analyzed key metrics to assess the chatbot’s effectiveness.

2. WORK TO BE DONE

- Conduct a comprehensive review of the system’s performance and identify areas for further improvement.
- Prepare for the final round of testing and validation before deployment.
- Develop a training plan for customer support agents to effectively utilize the new system.

3. PROBLEMS ENCOUNTERED

- Experienced delays in finalizing the human takeover process due to unforeseen technical challenges.
- Encountered issues with the performance metrics analysis, requiring additional adjustments to achieve accurate results.

4. SELF EVALUATION OF THE PROGRESS

The project has made substantial progress, with key features completed and integrated. The system is nearing the final stages of testing and preparation for deployment.

Alex

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T2,Y3	**Study week no.: 10**
Student Name & ID: Teo Wen Jin 20ACB02426	
Supervisor: Ts Dr Ooi Boon Yaik	
Project Title: Automate Customer Support Handling E-commerce enquiry using ChatGPT	

1. WORK DONE

- Completed the final round of testing and validation, ensuring the system's reliability and performance.
- Developed and delivered training materials for customer support agents on the new system.
- Finalized the system documentation and prepared for deployment.

2. WORK TO BE DONE

- Deploy the integrated chatbot system and monitor its performance in a live environment.
- Collect post-deployment feedback from users and support agents to identify any issues or areas for improvement.
- Begin planning for report writting.

3. PROBLEMS ENCOUNTERED

- Encountered minor issues during the final testing phase that required quick resolution to meet the deployment timeline.
- Faced challenges in ensuring comprehensive training for all customer support agents due to time constraints.

4. SELF EVALUATION OF THE PROGRESS

The project is successfully nearing completion with the final testing and deployment phases underway. Overall, the project's progress has been commendable.

Alex

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T2,Y3	**Study week no.: 10**
Student Name & ID: Teo Wen Jin 20ACB02426	
Supervisor: Ts Dr Ooi Boon Yaik	
Project Title: Automate Customer Support Handling E-commerce enquiry using ChatGPT	

1. WORK DONE

- Successfully deployed the chatbot system and initiated live environment monitoring.
- Collected and analyzed post-deployment feedback from users and customer support agents.
- Implemented initial improvements based on feedback and performance data.

2. WORK TO BE DONE

- Continue monitoring system performance and address any issues that arise.
- Conduct follow-up training sessions and provide ongoing support to ensure effective system utilization.
- Evaluate the overall impact of the system on customer satisfaction and sales performance.

3. PROBLEMS ENCOUNTERED

- Some minor issues emerged post-deployment that required immediate attention and adjustments.
- Ongoing adjustments based on feedback have been time-consuming and impacted the planned schedule.

4. SELF EVALUATION OF THE PROGRESS

The overall success of the system and its impact on customer support and sales performance is being closely monitored.

Alex

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T2,Y3	**Study week no.: 14**
Student Name & ID: Teo Wen Jin 20ACB02426	
Supervisor: Ts Dr Ooi Boon Yaik	
Project Title: Automate Customer Support Handling E-commerce enquiry using ChatGPT	

1. WORK DONE

- Completed and finalized the final year project (FYP) report, incorporating all revisions and feedback from supervisors.
- Prepared and delivered the project defense or presentation, effectively communicating the project's objectives, methodology, results, and conclusions to the evaluation committee.
- Conducted a thorough review of the project's overall outcomes, documenting lessons learned and potential areas for future improvement.

2. WORK TO BE DONE

- Await feedback from the evaluation committee regarding the project defense or presentation.
- Address any final adjustments or additional requests from the committee, if applicable.
- Reflect on the project experience and prepare a personal evaluation of the project's impact and learning outcomes.

2. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

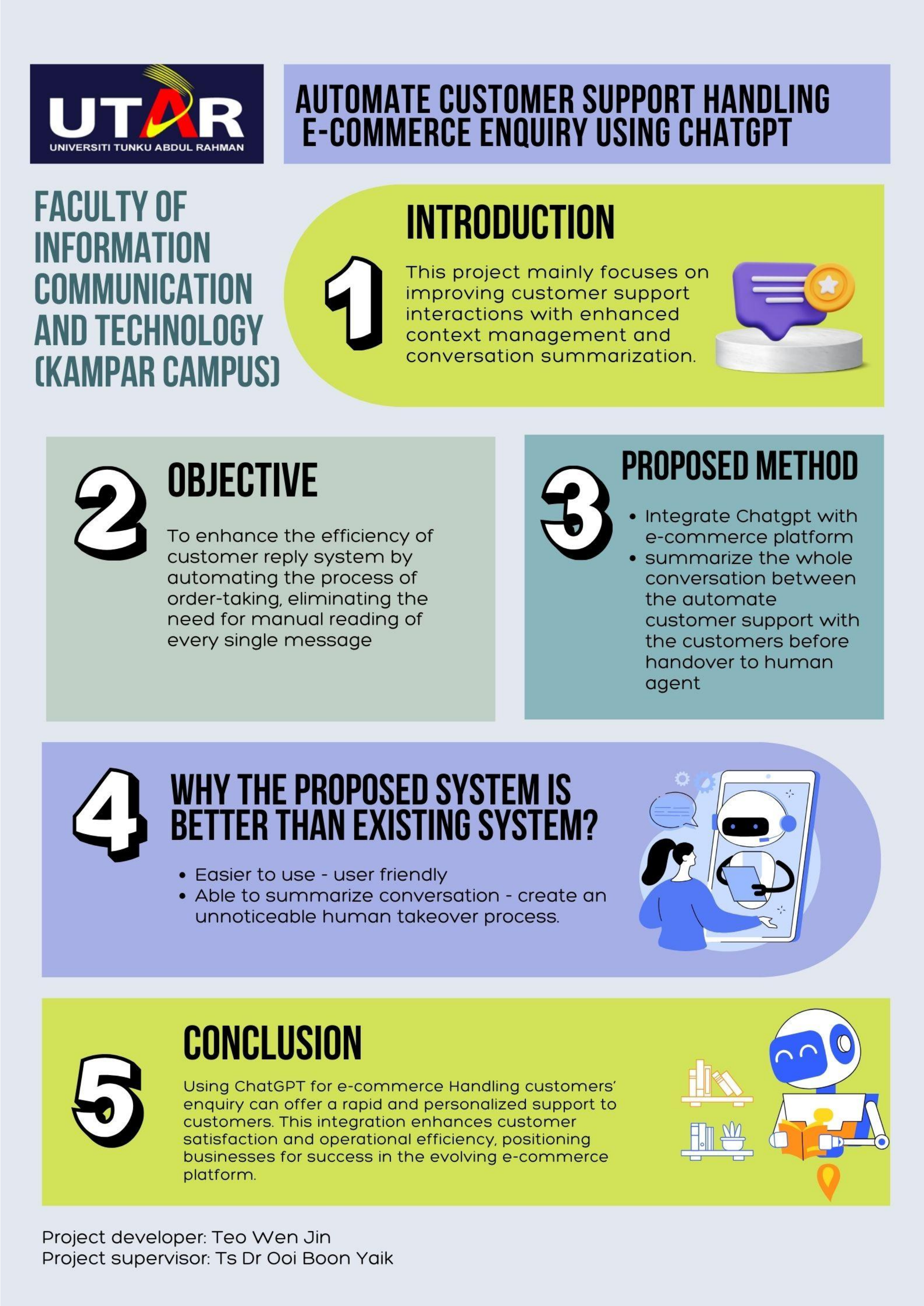
The project has reached its conclusion with the successful completion and submission of the FYP report and presentation. Despite encountering some challenges during the defense, the overall performance has been strong.

Alex

Supervisor's signature

Student's signature

POSTER



UTAR UNIVERSITI TUNKU ABDUL RAHMAN

AUTOMATE CUSTOMER SUPPORT HANDLING E-COMMERCE ENQUIRY USING CHATGPT

FACULTY OF INFORMATION COMMUNICATION AND TECHNOLOGY (KAMPAR CAMPUS)

1 INTRODUCTION

This project mainly focuses on improving customer support interactions with enhanced context management and conversation summarization.

2 OBJECTIVE

To enhance the efficiency of customer reply system by automating the process of order-taking, eliminating the need for manual reading of every single message

3 PROPOSED METHOD

- Integrate Chatgpt with e-commerce platform
- summarize the whole conversation between the automate customer support with the customers before handover to human agent

4 WHY THE PROPOSED SYSTEM IS BETTER THAN EXISTING SYSTEM?

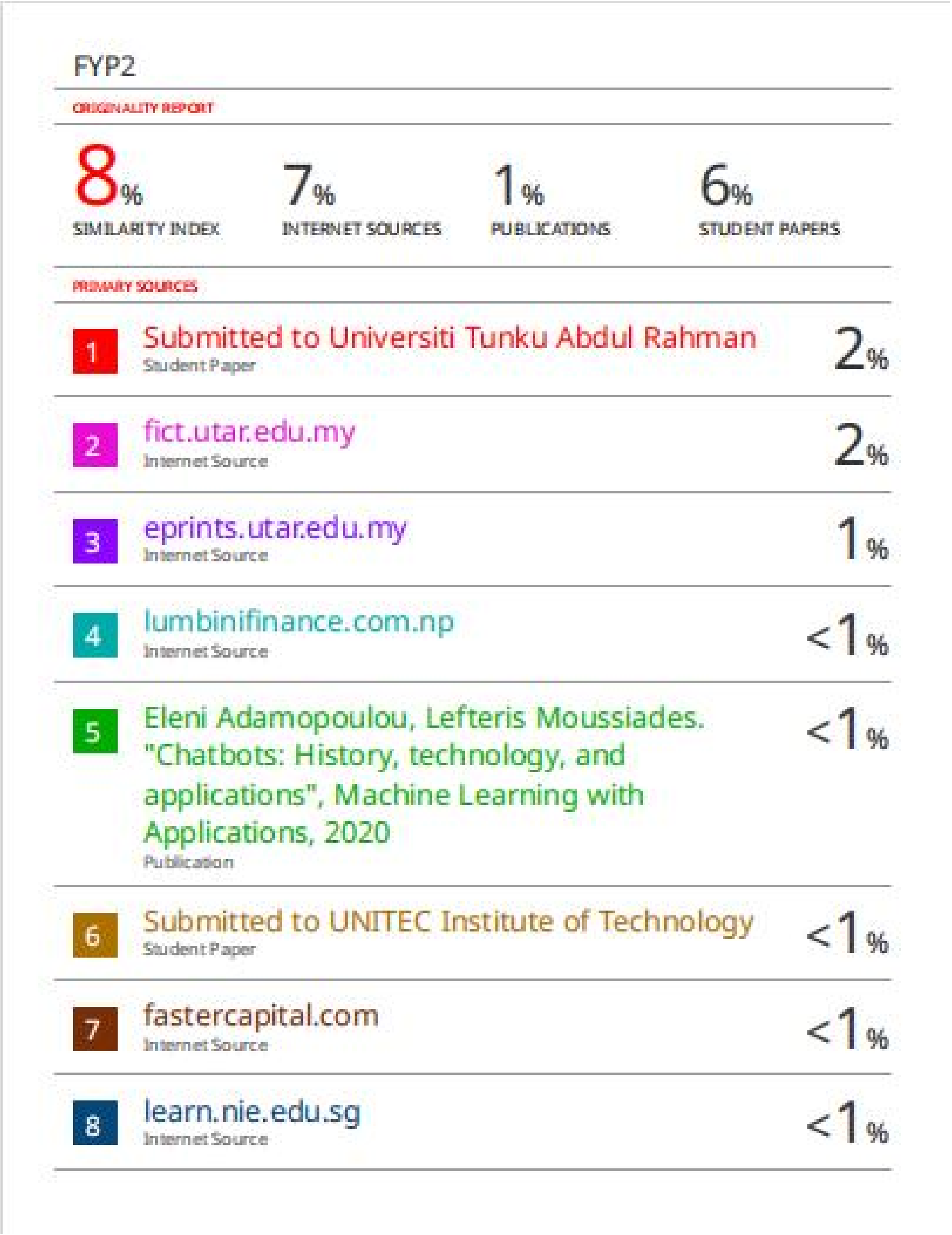
- Easier to use - user friendly
- Able to summarize conversation - create an unnoticeable human takeover process.

5 CONCLUSION

Using ChatGPT for e-commerce Handling customers' enquiry can offer a rapid and personalized support to customers. This integration enhances customer satisfaction and operational efficiency, positioning businesses for success in the evolving e-commerce platform.

Project developer: Teo Wen Jin
Project supervisor: Ts Dr Ooi Boon Yaik

PLAGIARISM CHECK RESULT



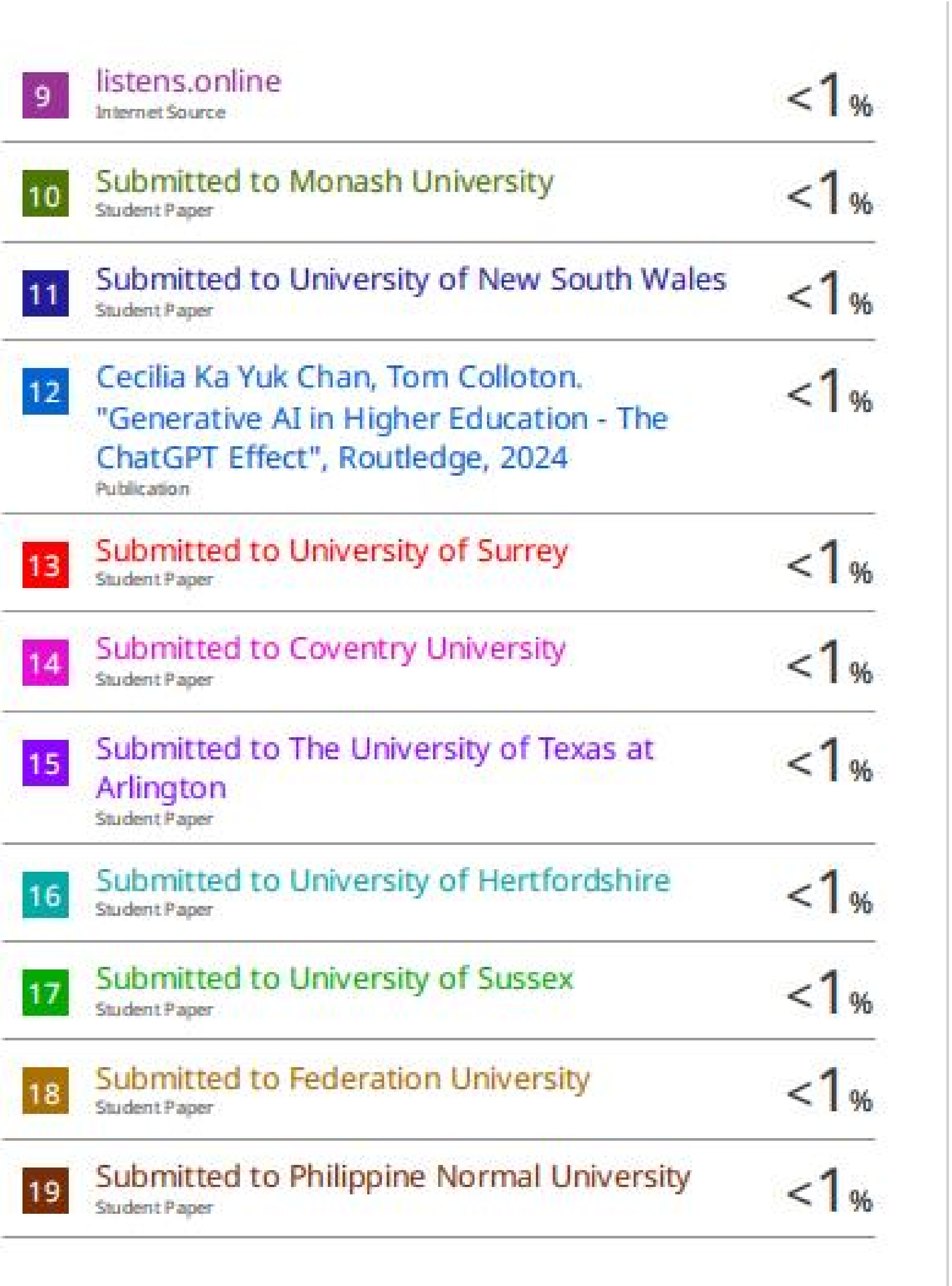
FYP2

ORIGINALITY REPORT

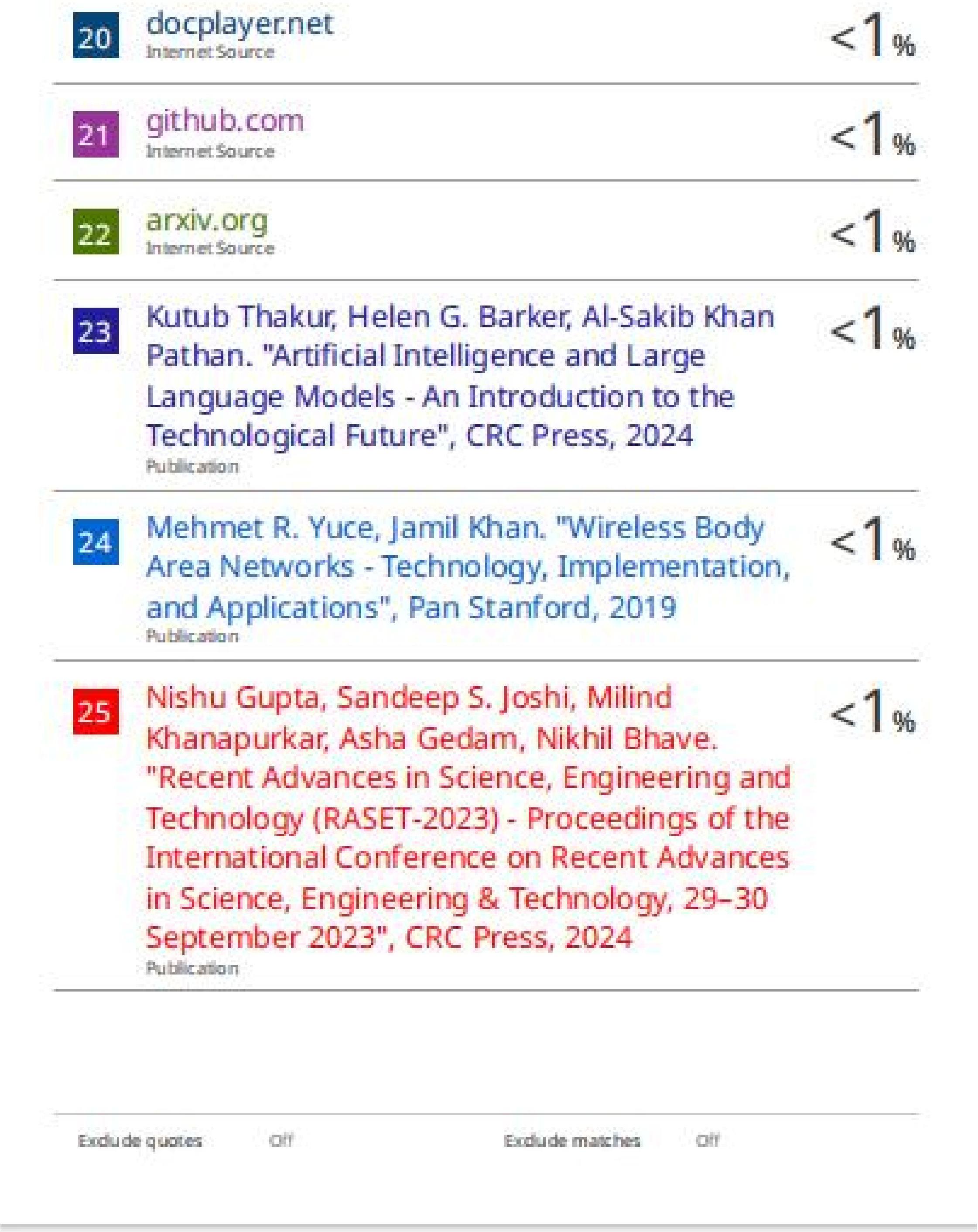
8%	7%	1%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
2	fict.utar.edu.my Internet Source	2%
3	eprints.utar.edu.my Internet Source	1%
4	lumbinifinance.com.np Internet Source	<1%
5	Eleni Adamopoulou, Lefteris Moussiades. "Chatbots: History, technology, and applications", Machine Learning with Applications, 2020 Publication	<1%
6	Submitted to UNITEC Institute of Technology Student Paper	<1%
7	fastercapital.com Internet Source	<1%
8	learn.nie.edu.sg Internet Source	<1%



9	listens.online Internet Source	<1%
10	Submitted to Monash University Student Paper	<1%
11	Submitted to University of New South Wales Student Paper	<1%
12	Cecilia Ka Yuk Chan, Tom Colloton. "Generative AI in Higher Education - The ChatGPT Effect", Routledge, 2024 Publication	<1%
13	Submitted to University of Surrey Student Paper	<1%
14	Submitted to Coventry University Student Paper	<1%
15	Submitted to The University of Texas at Arlington Student Paper	<1%
16	Submitted to University of Hertfordshire Student Paper	<1%
17	Submitted to University of Sussex Student Paper	<1%
18	Submitted to Federation University Student Paper	<1%
19	Submitted to Philippine Normal University Student Paper	<1%



20	docplayer.net Internet Source	<1%
21	github.com Internet Source	<1%
22	arxiv.org Internet Source	<1%
23	Kutub Thakur, Helen G. Barker, Al-Sakib Khan Pathan. "Artificial Intelligence and Large Language Models - An Introduction to the Technological Future", CRC Press, 2024 Publication	<1%
24	Mehmet R. Yuce, Jamil Khan. "Wireless Body Area Networks - Technology, Implementation, and Applications", Pan Stanford, 2019 Publication	<1%
25	Nishu Gupta, Sandeep S. Joshi, Milind Khanapurkar, Asha Gedam, Nikhil Bhave. "Recent Advances in Science, Engineering and Technology (RASET-2023) - Proceedings of the International Conference on Recent Advances in Science, Engineering & Technology, 29–30 September 2023", CRC Press, 2024 Publication	<1%

Exclude quotes Off

Exclude matches Off

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	TEO WEN JIN
ID Number(s)	2002426
Programme / Course	IA
Title of Final Year Project	**Automate Customer Support Handling E-commerce enquiry using ChatGPT**

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: ____8____ **%** **Similarity by source** Internet Sources: _______1_______% Publications: ______1_______ % Student Papers: ______2________ %	
Number of individual sources listed of more than 3% similarity: 0________	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

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

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

Alex

____
Signature of Supervisor

Name: ___Ts Dr Ooi Boon Yaik

Date:
________12/9/2024___________________

____
Signature of Co-Supervisor

Name: ___________________________

Date: ____________________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB02426
Student Name	Teo Wen Jin
Supervisor Name	Ts Dr Ooi Boon Yaik

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

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

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date: 12/9/2024