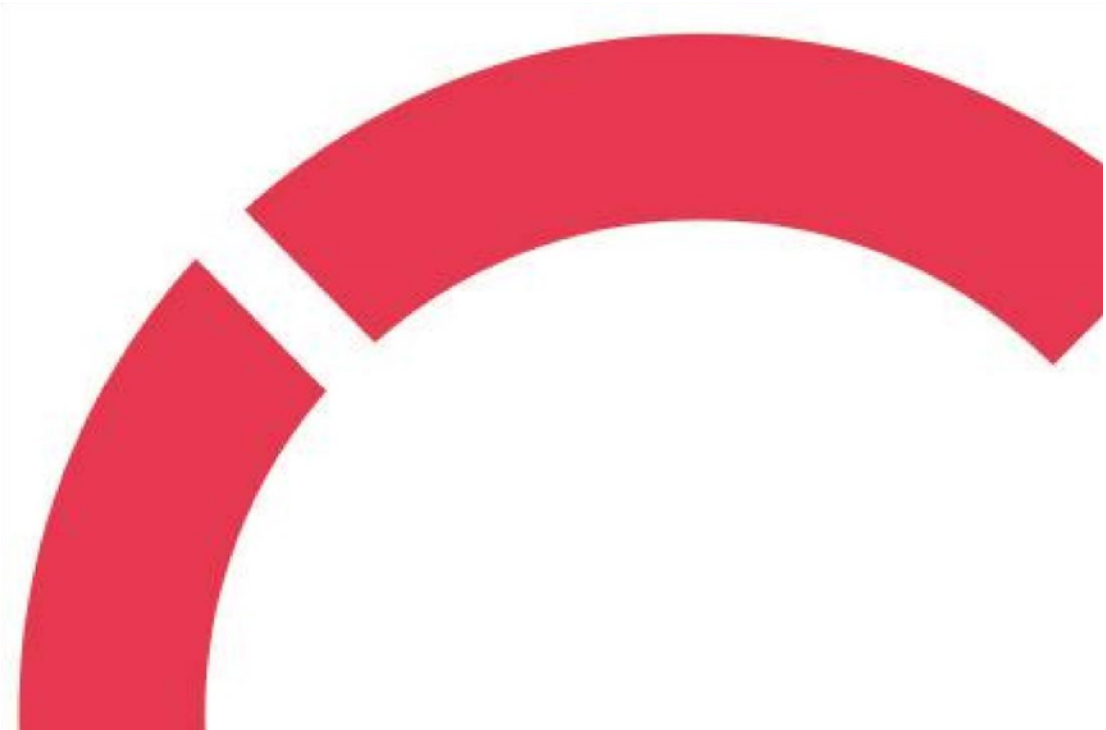
Arafat Adigun

DESIGNING A SIMPLE AI-POWERED CHATBOT FOR CUSTOMER SUPPORT

A User-Centric Approach

Thesis
CENTRIA UNIVERSITY OF APPLIED SCIENCES
Industrial Management
December 2024

centria
University of Applied Sciences



ABSTRACT

Centria University of Applied Sciences	**Date** December 2024	**Author** Arafat Adigun
Degree programme Industrial Management		
Name of thesis DESIGNING A SIMPLE AI-POWERED CHATBOT FOR CUSTOMER SUPPORT: A USER-CENTRIC APPROACH		
Centria supervisor Aliasghar Khavasi		**Pages** 38 + 2

This thesis explores the development of an AI-Powered chatbot designed to enhance customer service on WhatsApp Business. In short, the core concept lies in the development of a chatbot that can handle customer queries by responding aptly to improve user experience without human intervention in routine interactions. This thesis work also focused on the evaluation of a chatbot for efficiency in responses and solving queries by adopting a user-centric approach.

This thesis was supported by the theoretical framework of literature on chatbot technologies and NLP in customer service automation. The review of varied case studies has shown the existing limitations and the points which have the scope for improvement within the functionality of the chatbots.

A conversational chatbot was prototyped, tested, and enhanced through user feedback surveys and performance evaluation mechanisms. Certain methodologies undertaken in the research study involved the collection and analysis of real data with respect to user satisfaction about chatbot responses and their accuracy, among other factors. Performance was measured on how well it understands natural language queries and responds with relevant information, handled multiple scenarios around customer service.

The results showed that the bot could solve some simple inquiries, reduce response time, and raise efficiency in general, while several challenges were related to the inability of the chatbot to answer more complicated queries and give personalized responses. It was concluded that while the chatbot has done an excellent job for simple-level customer service, further development of the technology could be channelled toward tackling higher-level cases and allowing greater degrees of personalization to enhance user friendliness.

Key words
AI-Powered chatbot, customer service, machine learning, natural language processing, performance evaluation, user-centric design, WhatsApp Business.

CONCEPT DEFINITIONS

AI-POWERED CHATBOT

An artificial intelligence (AI) system designed to simulate human conversation and interact with users through text or speech, typically used in customer service.

NATURAL LANGUAGE PROCESSING (NLP)

A branch of AI that focuses on the interaction between computers and humans using natural language. It enables the chatbot to understand and respond to user inputs.

USER EXPERIENCE (UX)

The overall experience a user has when interacting with a product or system, in this case, the AI-powered chatbot, focusing on ease of use and satisfaction.

PROTOTYPE

An early sample or model of the chatbot developed to test and evaluate its functionalities and effectiveness in real-world scenarios.

PERFORMANCE EVALUATION

The process of assessing the chatbot's ability to understand queries, provide relevant responses, and improve customer service efficiency.

ABSTRACT CONCEPT DEFINITIONS CONTENTS

FIGURES

1 INTRODUCTION

The customer support industry is growing at an amazingly fast pace, and every other day, there is something new that makes communication faster and the service process more effective. Nowadays, more businesses look to accomplish speed, accuracy, and efficiency in customer support through automation, and precisely AI-powered chatbots. This project is intended for developing WhatsApp customer support chatbot that is designed to streamline interactions between businesses and their clients.

The chatbot was developed on natural language processing and machine learning methods to understand users' queries and intelligently respond, considering the context in which something has been asked. The chatbot, therefore, reduces human agent dependency to trivial queries through automated responses to often asked questions and common issues. The development of a user-friendly and responsive chatbot system is the aim of this project, enhancing customer satisfaction with the improvement of operational effectiveness.

With the increasing responsiveness and easy accessibility of customer support, AI-powered chatbots have been the core development in recent years (Baabdullah *et al.,* 2022). Along with growing expectations of speed and reliability in services, chatbots have reduced waiting for the customers with their apt responses. To this regard, the chatbot developed integrates with Ollama, a sophisticated AI language model that responds intelligently to queries that will be coming in from the customer via WhatsApp. The messaging via this chatbot is supported by Twilio, a cloud communication platform allowing users to interact seamlessly with the chatbot.
Other technology used in this project is Flask, a web framework powered by Python. It shall be deployed on Render for ease of access and scalability reasons. Unlike other customer support systems that are taken control of by human agents, the chatbot automatically responds to simple and frequently asked questions to help businesses improve their customer support processes.

This project involves developing an AI-powered chatbot for WhatsApp customer support. It deals with Twilio to handle messaging and Ollama to generate the AI-driven response. It is designed for basic inquiries by customers and to guide the user through diagnostic and information-related questions.
The major deliverables of the project are a fully functional, connected WhatsApp chatbot that will be integrated with Ollama to enable the generation of smart and contextually relevant responses. This would

be seamlessly deployed with the use of Flask and Render. This shall finally result in a tool that would help companies reduce human customer support agents to a minimum by automating many routine tasks.

This project will try to answer a few of the most vital questions through its building process: How can a WhatsApp chatbot help enhance customer support with the help of AI? How does it benefit from the usage of Ollama to generate intelligent responses? and How well is Twilio integrated with WhatsApp in handling messages? The project will finally investigate the challenges of developing and launching an AI-powered customer support chatbot. These questions are going to be explored and answered as development and assessment of the chatbot are being pursued.

2 THEORETICAL FRAMEWORK

The HCI is most central to the development of a chatbot for customer support settings, to judge the effectiveness of the chatbot. The major principles from HCI, such as usability, security, and performance, form the foundational pillars that ensure a chatbot application will be user-friendly, safe, and efficient in addressing user needs (Casheekar 2024). The word usability in the context of chatbots, means it should be an easy-to-use interface for customers without much learning or training. A good interface will look to make a chatbot experience as seamless as possible. Hence, the customers will feel that their concerns are understood and supported. Another important concern here is security since it also includes offering security features to protect customer data from unauthorised access or breaches. This will typically involve the use of encryption protocols, authentication processes, and adherence to laws such as Europe's GDPR in the protection of customer data privacy Lastly, performance concerns the speed and reliability of a response by a chatbot. A chatbot that produces fast answers without frequent errors will be satisfying to use and builds confidence in the technology. These three principles form the backbone of developing a customer-centric Chatbot (Hasal 2021).

Table 1 - HCI Principles and Their Relevance to Chatbots

HCI Principle	Definition	Relevance to Chatbots
Usability	All users will find the interface easy to use	Ensures Customer friendly interactions
Security	Maintaining privacy and safeguarding user data	increases adherence to rules and trust (e.g., GDPR)
Performance	Accurate, dependable, and prompt responses	Improves customer satisfaction and engagement

2.1 Chatbot Technology and Evolution

Chatbot technology has evolved much since its creation, moving from the first basic rule-based systems to today's highly advanced artificial intelligence (AI)-driven systems (Al- Amin 2024). The AI driven systems are capable of more complex interactions. Initially, chatbots followed scripted responses and could only answer pre-defined questions. With the development of AI and machine learning (ML), modern chatbots now go a step further; they recognize intent, learn from interactions, and provide contextually relevant responses. Various frameworks have been instrumental in the design and deployment of such AI chatbots, each having unique characteristics. For instance, Rasa is noted for its open-source architecture and extensive customization possibilities. This makes it a popular choice among developers looking for full control over chatbot functionality. On the other hand, the Microsoft Bot Framework offers a wide range of possibilities for integration with Microsoft apps, thus making it more appealing to companies that are already heavily invested in the Microsoft environment. Google's Dialogflow and Botpress offer pre-configured natural language processing instruments, thus increasing accessibility for enterprises that seek a rapid deployment of chatbots without the requirement for vast technical knowledge. The variability of chatbot technology is represented by these frameworks, showing how adaptable chatbots are in diverse organizational requirements and technical skills. (Rybaric 2023).

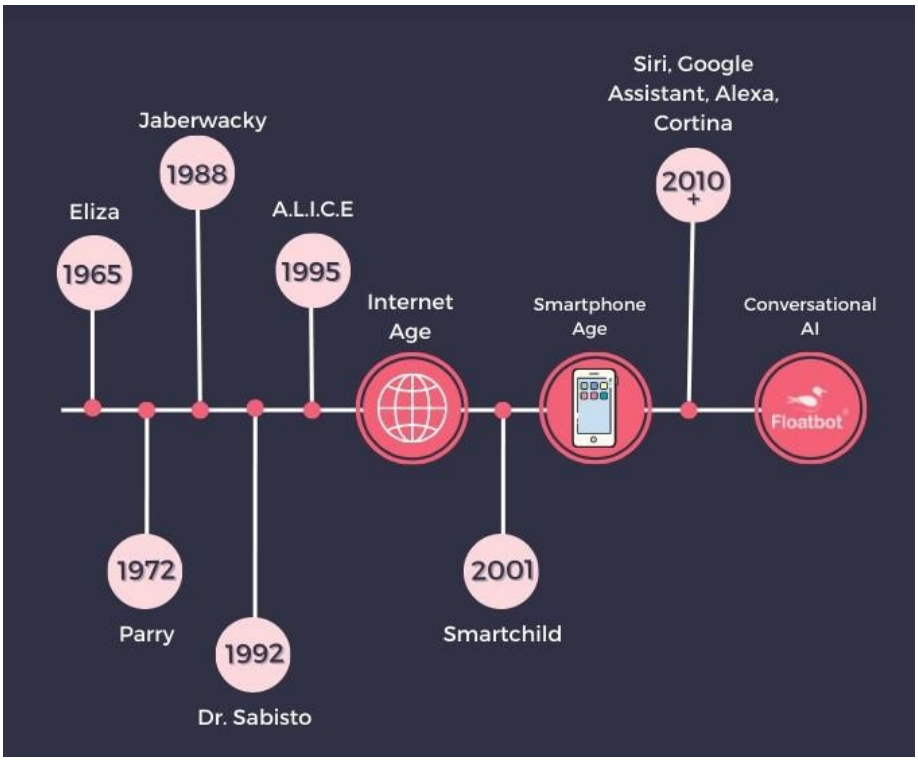


Figure 1: Schematic diagram showcasing the Evolution of Chatbot over the years (Floatboat,2023)

Figure 1 illustrates the evolution of chatbots, starting with Eliza in 1965 and progressing to ALICE in1995 and Dr. Sabisto in 1992. It also highlights significant milestones like the internet's rise in 2001 and the introduction of smartphone assistants like Siri, Google Assistant, and Alexa in the 2010s, culminating in conversational AI tools like Floatbot.

2.2 Combining Machine Learning with Chatbots

In machine learning, it is important to improving the intelligence of chatbots. Hence, the system improves its accuracy and effectiveness over time. Chatbots learn patterns of language and can recognize user intent more precisely through past conversation data analysis, thereby providing more proper responses. Machine learning enables a chatbot to handle variations in how people frame questions, from different accents, idioms, and phrases that might confuse a more basic system. Adaptability is very crucial in customer support, as understanding the diversified expressions of users and their requirements can really enhance the service quality. For example, machine learning algorithms enable chatbots to understand complex questions, which might require more than one answer or further specification by users. Using reinforcement learning and feedback systems, chatbots can effectively rank the answers that are most frequently used or successful. This aim in providing an enhanced response with respect to customer expectations in each interaction. Such models can be visualized by use of flowcharts on how machine learning does indeed support the intent recognition and response selection process of chatbots (Haile, 2022).

2.3 Cybersecurity Considerations

With the rise in importance of chatbots in customer service, cybersecurity has become most critical. Information of customers from basic contact details to extensive transaction records, often flows through these systems. Hence, the protection of this information is necessary. Standard security measures include data encryption. A process that masks data so it cannot be accessed, along with protection against malicious intrusions. This includes cross-site scripting and SQL injection via Web Application Firewalls (WAFs). The use of chatbots in line with stringent data privacy procedures, such as the EU's General Data Protection Regulation (GDPR), would be subjected to the observance of added controls. Each of these measures is helpful in ensuring customer interactions are still secure, which safeguards the trust in the chatbot and the organization it represents (Clifford 2024).

2.4 Effectiveness of Chatbot in Customer Support

Chatbots have been effective in their role of offering customer support, in e-commerce and services like WhatsApp Business. This is because chatbots automate the process of answering routine questions which now allows human representatives to concentrate on more complex or sensitive situations. Hence, it brings efficiency in service delivery. Examples of e-commerce chatbots would include handling queries concerning order status, product information, and return policies. This experience is further complemented by WhatsApp Business, which has a user-friendly interface that provides quick and familiar communication. Multitasking is also supported by chatbots which can handle more than one conversation at a time. A feat human agents cannot do. This 24/7 or all-day-round assistance is particularly invaluable to customers who are in different time zones or need help after regular business hours. Compared to human agents, chatbots have the advantages of speed and scalability. Although, this may be restricted in the handling of complex issues or tasks that require empathy and fine judgments (Kyrylenko 2024)

2.5 Effectiveness of Chatbot in Customer Support

An advantage of chatbot is clear in its consistent service provision because they can work around the clock without any breaks. As a result, it does not require any waiting time. It gives a better customer experience, especially for enterprises in which good support in the least time is important (Nicolescu 2022). They are also a cost-effective solution since they reduce the need for large customer support staff, thereby saving big on operational costs. Furthermore, chatbots possess the capability to manage many inquiries simultaneously. This thereby shows a degree of multitasking unattainable by human agents. Nevertheless, chatbots do meet constraints. One of this is particularly shown when confronted with complex questions necessitating contextual comprehension or emotional intelligence. Although advancements in machine learning and artificial intelligence have markedly improved the functionality of chatbots. But they may still struggle to understand specificity in language or emotional signals. This potentially results in customer dissatisfaction during more delicate circumstances. These limitations bring into focus the need for keeping human agents for more expert support, thus ensuring that chatbots act as a first line of contact rather than being an overall replacement for human interaction (Tudorache, 2022)

3 FUNCTIONAL, NON-FUNCTIONAL, AND DOMAIN SPECIFIC REQUIREMENTS

This chapter covers the design, development, and deployment of an AI-driven chatbot tailored for customer support in e-commerce. The process includes the integration of the Ollama AI model, which forms the core of the chatbot's natural language understanding and response generation. Additionally, the chatbot was seamlessly connected to WhatsApp Business using Twilio, enabling real-time. communication and enhanced accessibility for users. The chapter also delves into the tools, technologies, and methodologies utilized to ensure the chatbot's functionality, scalability, and ease of use.

3.1 Functional Specifications

Functional requirements outline the basic functions and actions of the chatbot, thus ensuring that it can effectively perform the duties of customer service. Natural Language Understanding (NLU) allows the chatbot to correctly understand the requests of users and to give relevant answers. Additionally, the chatbot has been designed to provide proper responses based on the input of users, answering those frequently asked questions, general questions, and specific requests from customers.
Example: "If a user sends the query 'Where is my order?,' the chatbot identifies the intent ('order tracking') and retrieves relevant information based on the user's order history."

The chatbot includes contextual continuity, which keeps the information from previous messages in that session to give more personalized responses to improve user experience. It works through a user interface on WhatsApp, making communication seamless for the users. Example: "If a user asks, 'What is the delivery fee?' after querying about product availability, the chatbot understands that the question relates to the previously discussed product."

Moreover, in case the chatbot cannot provide an accurate response, it should be able to escalate the question to human support so that the users can get the support needed.Example: "If a user asks, 'How do I get a refund for a damaged product?' and the chatbot cannot provide a resolution, it offers to connect the user to a live agent for further assistance."

3.2 Non-Functional Specifications

Non-functional requirements outline the quality attributes of the chatbot. Performance is one of the critical attributes, as the chatbot must provide answers within two seconds to maintain user interest and satisfaction. Reliability is another major attribute, as the chatbot must be able to function stably to minimize downtime and ensure service continuity. The scalability attribute is also essential for handling increased interaction volumes, especially during high-traffic periods. This is important to secure the users' data and maintain confidentiality in line with data protection regulations, ensuring that the users' interaction history and personal data are secured.

3.3 Domain Requirements

Domain requirements ensure the chatbot remains put with the specific customer support needs of the business. The chatbot gives proper responses and behaves in a manner that conforms to rules, guidelines, policies, and standards set by an organization. It also addresses common business-specific queries such as questions related to order status, product availability, and shipping information through proper responses thereby improving customer experience. Also, the tone and language of the chatbot are consistent with the brand's style and standards of communication, therefore maintaining brand consistency and further improving the user's perception of the business.For example: “The chatbot responds to product availability queries by retrieving stock information from the database.”

3.4 Architectural Framework and Design Considerations

The architectural framework and design of the chatbot involve both high-level and low-level considerations. This section provides an overview of the system's components, interactions, and detailed designs, including use case, activity, sequence, and class diagrams. These elements illustrate how the chatbot functions from a structural and operational perspective.

3.4.1 High-Level Architecture

I have a couple of components in this architecture; each part plays some kind of role within the system. The user interface is hosted on WhatsApp Business for easy and efficient interaction with the chatbot. The backend is developed using the Flask framework to handle user requests, route messages to the proper processes, and interact with the AI model. The AI model, Ollama, takes the natural language input and generates contextually proper responses. A communication layer using Twilio serves to connect WhatsApp with the backend, hence allowing smooth, continuous messaging between the chatbot and the users. To better understand these components, a part diagram is used to illustrate their interactions and how they help in information exchange.

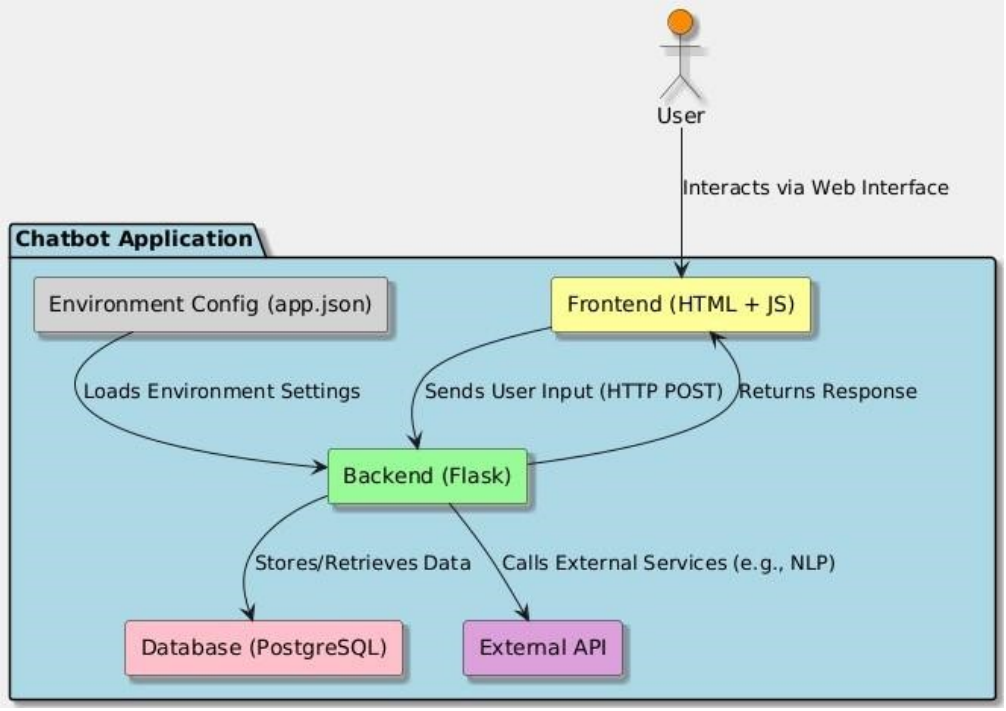


Figure 2: High-Level Component Diagram for a Chatbot Architecture Figure 2 shows the high-level architecture of a chatbot system, including its frontend, backend, database, and external API integrations. The frontend uses HTML and JavaScript for user interaction, while the backend uses Flask for request processing. The Chatbot Logic component handles input analysis, interacts with a PostgreSQL Database, and may call External APIs for natural language understanding. The interconnected system ensures efficient message processing and response generation.

3.4.2 Detailed Design

The design phase also contains detailed diagrams of the interaction and functionality of the chatbot in a more elaborate way. The Use Case Diagram depicts the way in which users interact with the chatbot: initiating a conversation, making queries on topics, accessing FAQs, and the possibility of escalation to human support when needed. The main functions and user interactions of the chatbot are explained through this diagram. The Sequence Diagram visually represents the step-by-step flow of interactions between users, WhatsApp, Twilio, the Flask backend, and the Ollama model in conveying the query initiation process, message relaying, processing, and response generation. The Activity Diagram captures the progression of actions from a user query to the response, inclusive of the error-handling steps and the escalation possibilities for unresolved queries. The Class Diagram shows the structural view of the components of the chatbot, elaborating on the main classes: User, Chatbot, Message, Session, and Response. All these classes have attributes and methods to support the operations of the chatbot in its sessions, response generation, and keeping the conversation flowing.

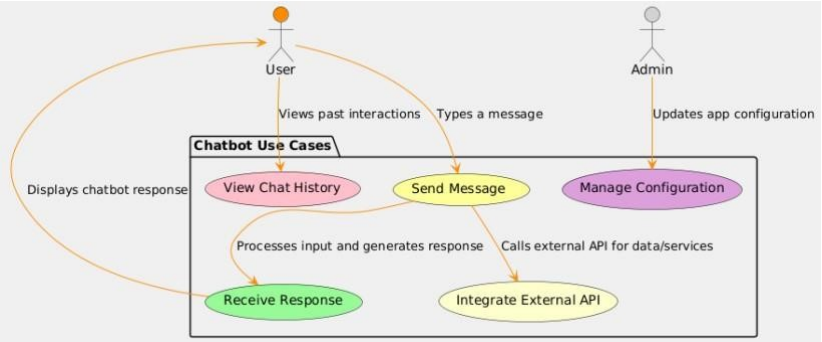


Figure 3: Use case Diagram Depicting user Interaction with the Chatbot

The Use Case Diagram in Figure 3 outlines user interactions with the chatbot system, focusing on two main actors: the User and the Admin. The User can send messages, receive responses, and view chat history, while the admin manages configuration by updating environment settings. The chatbot system also integrates with external services through the "Integrate External API" use case, allowing advanced processing through third-party APIs. This diagram provides a comprehensive overview of the application's functional requirements.

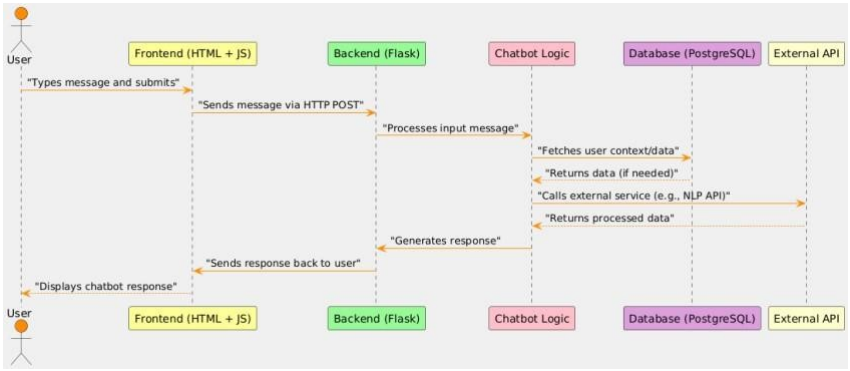


Figure 4: Sequence Diagram for a Chatbot System Showing Interaction Flow

The Sequence Diagram illustrates the interaction flow between the user and the chatbot system. The user sends a message through the Frontend, which captures input and forwards it to the Backend via HTTP POST. The Backend routes the request to the Chatbot Logic, which checks the Database for user context. If needed, the logic calls an External API for NLP processing. The response from the external service is combined with user data to generate a final response, which is then sent back to the Frontend.

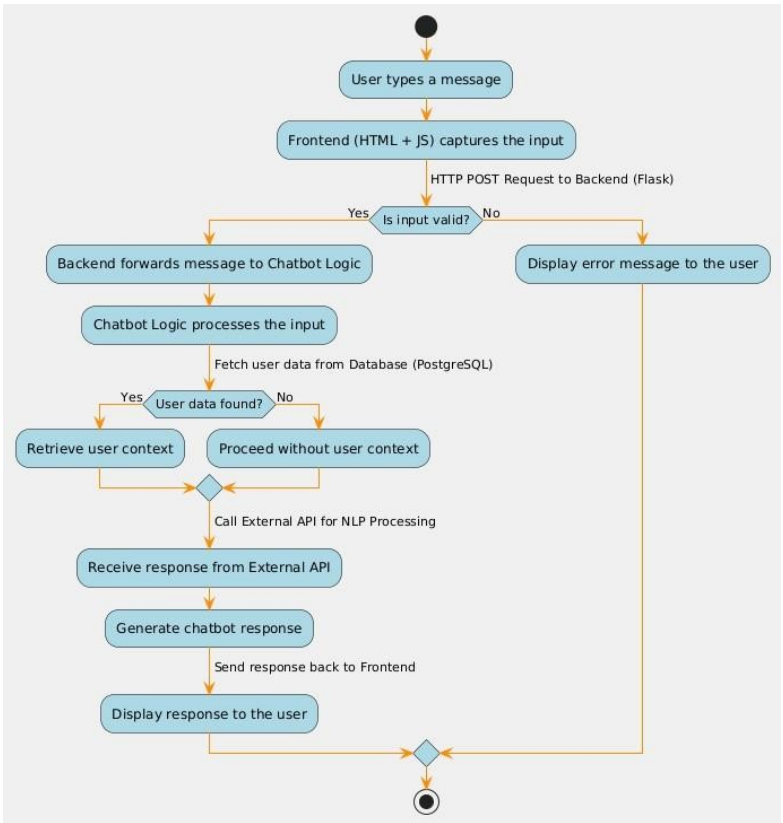


Figure 5: Activity Diagram Representing User Query Processing and Responses The Activity Diagram in Figure 5 illustrates the process of handling user queries in a chatbot system. The user inputs a message, which is captured by the Frontend. The input is sent to the Backend, which checks its validity before forwarding it to the Chatbot Logic for processing. The chatbot may interact with the Database to fetch user data or context. If further analysis is needed, the logic calls an External API for Natural Language Processing (NLP). The response is displayed to the user, and an error message is displayed if the input is invalid.

3.5 The Python Ecosystem and Libraries

The configuration of the development environment facilitates the effective integration of various components and promotes efficient testing and deployment processes. Python was selected due to its versatility and extensive library support, especially in the context of machine learning and artificial intelligence applications. A virtual environment was established utilizing Python's venv module to oversee dependencies and ensure compatibility. It installed essential libraries like Flask for backend operations, Twilio for integration with WhatsApp, and Ollama for AI-driven response generation within the virtual environment, to avoid potential conflicts with dependencies of other projects or system-wide libraries.

3.5.1 Flask Application Setup

Flask is a minimalistic web framework, used to create a clear interface that accepts user input and sends it to Ollama for processing; the response is returned accordingly. The application was set up with endpoints for handling incoming and outgoing messages, therefore providing a convenient interface. The user interface was simple and usable with the main goal of generating fast and effective answers to users' questions.

3.5.2 Ollama Model Integration

The Ollama model, known for its ability in natural language processing, was incorporated to generate contextually relevant responses. It enhances the capability of the chatbot to understand user queries,

maintain the context of a conversation, and respond accurately, which contributes much to the conversational abilities of the chatbot.

3.6 Organization of Code and Principal Components

The system consists of a chatbot module, a Flask API module, a Twilio Integration module, a context management module, a WhatsApp (via Twilio) module, an external model called the Olama Model, and a user interface. The chatbot module handles user input processing, conversation flow, and generating responses. It interacts with the Context Management module to fetch or save context, allowing the chatbot to provide context-aware responses. The Flask API module bridges the frontend and backend, managing HTTP routes and communication requests. The Twilio Integration module manages message transmission between WhatsApp and the Flask application, utilizing Twilio's messaging API. The context management module handles the storage and retrieval of conversation context, enhancing the chatbot's ability to provide context-aware responses. The user interface communicates with the backend via the Flask API module, facilitating seamless interaction between the user and the chatbot system. This modular architecture allows for clear separation of concerns, making the system more scalable and easier to maintain.

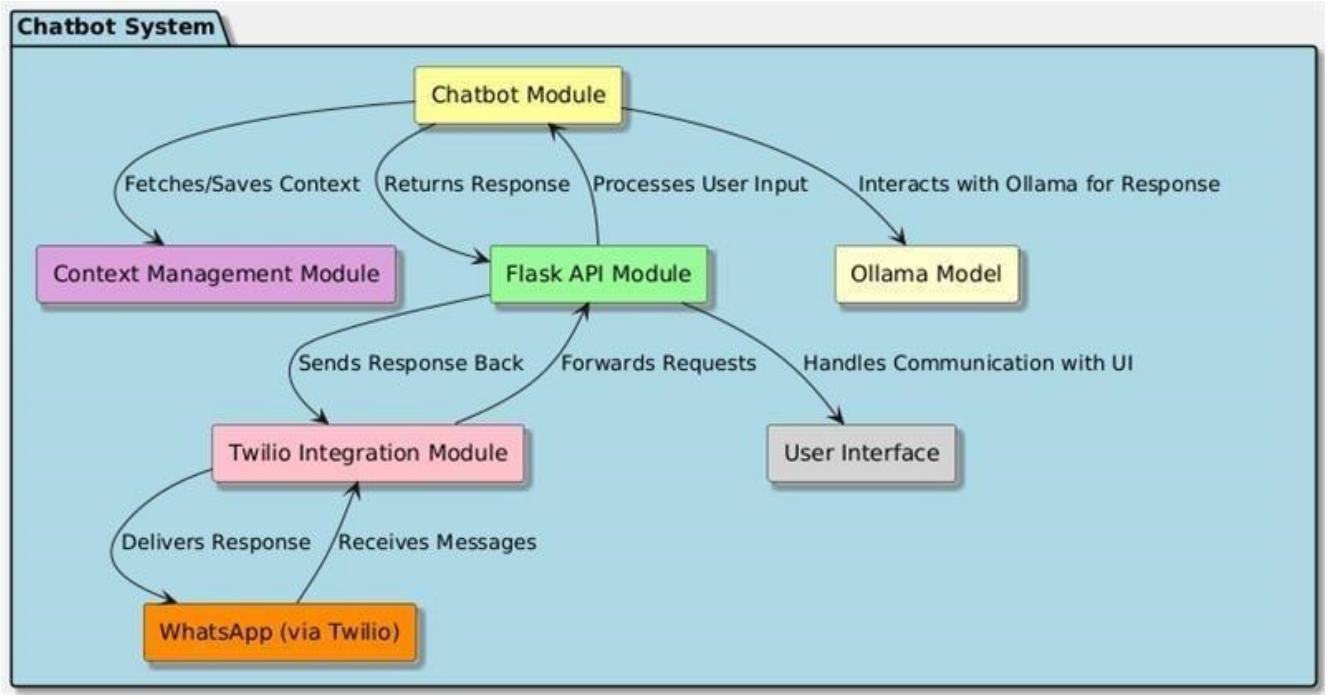


Figure 6: Code Organization and Principal Components

Figure 6 shows the chatbot system's code organization, utilizing a modular design to divide functionality into distinct modules for specific tasks, enhancing maintainability and testability.

3.7 Testing Approach

Unit testing was done on discrete functions and modules, like the response generator of the chatbot and the message-handling capabilities of the Flask API. This form of testing guarantees that each element functions accurately in isolation and thereby establishes a basis of dependability within the system. Integration testing confirmed that the Flask API, Twilio integration, and Ollama model worked harmoniously together. These tests proved that message flow, from the user's input within WhatsApp to Ollama's response generation, was working correctly and ensured all components of the chatbot interacted with one another.

The acceptance tests involved the checking of the chatbot's operational capabilities in real-life scenarios, such as handling normal customer queries, frequently asked questions, and procedures for escalation. This testing phase ensured the chatbot met user expectations, provided an acceptable experience, and functioned as designed while operating within the domain of customer support.

4 CHATBOT TESTING AND PROGRAM OUTPUT

This chapter elaborates in detail on the program output of the chatbot, with both functionality and performance at various testing phases. Each feature or interaction is presented with descriptive insights on how the chatbot addresses functional and non-functional requirements and aligns with domain specific needs for which it was developed. Testing results are discussed to prove the usability, accuracy of responses, and performance of the chatbot in various scenarios. These tests were a good step for the effectiveness of this chatbot because the tests were particularly dedicated to user experience and system reliability.

4.1 Implementation of Functional, Non-Functional and Domain Requirements

Demonstrations, in front of specified functional, non-functional, and domain-specific requirements, are done to confirm the capability of the chatbot in meeting the specifications. The chatbot's response accuracy and ability to interpret queries were the major functional requirements, while performance and security aspects were considered under the non-functional requirements. Each of the demonstrations in the front represents the critical functionalities that have been implemented by the chatbot and the ease of handling user interactions, continuity in user conversations, and responding to queries asked frequently by users.

The functional requirements of this chatbot were that it should be responsive, accurate, and user-friendly on WhatsApp Business. As an example, if any user initiates a conversation by asking about order tracking, then the chatbot understands the user's intention and has responded to track an order. This point identifies the language understanding capability in this chatbot, in which it processed the queries correctly and responded appropriately. Other examples are that the chatbot, if asked about delivery times, responds right away with the regular window when the products are usually delivered. It does this through answering the question in a manner that is vivid. Continuity of the context of the chatbot allows it to remember interactions that happened earlier in one session. For instance, if a user follows up and inquiries into costs, the chatbot keeps in context the previous order inquiry so that it may provide information accordingly without requiring the user to paraphrase his or her original question.

Non-functional requirements also played a significant role during performance assessment of the chatbot. As shown by the tests run, the performance standards were under two seconds per query. This is quite important in that an amazingly fast response keeps the users on track, especially when the user wants to get quick support. Moreover, the chatbot's reliability was confirmed since the tests run throughout the duration presented an uptime of the same. Security is one of the major non-functional requirements achieved by ensuring data encryption for all interactions via Twilio's secure channels, hence protecting user information against unauthorized access and maintaining privacy standards.

Further tuning of the chatbot for e-commerce was made by domain-specific requirements, in which its responses were to business standards and customer expectations. For this reason, it is so engineered to respond with precision for inquiries over product details, order tracking, and returns policies. By being domain-aligned, it ensures that responses are not only correct but also in line with the business guidelines on providing users with consistent and structured information, representative of the standards of communications within an organization.

4.2 Software Output and Screenshots of Functionality

The functioning of the chatbot has been documented with several screenshots showing its interaction with the user. In all the screenshots, a certain functionality of the chatbot is represented where the chatbot interacts with the users, responds to queries thrown at it, or shows session continuity during its conversation. For example, when the user initiates a query, the chatbot interprets it correctly and gives a relevant answer. Other proof is that the chatbot can give responses to frequently asked questions, which really aids the user in having a much better experience with the help of rapid and useful information. These views form proof of the efficiency, design, and compliance with requirements of the chatbot, thus proving good interaction capability within WhatsApp Business.

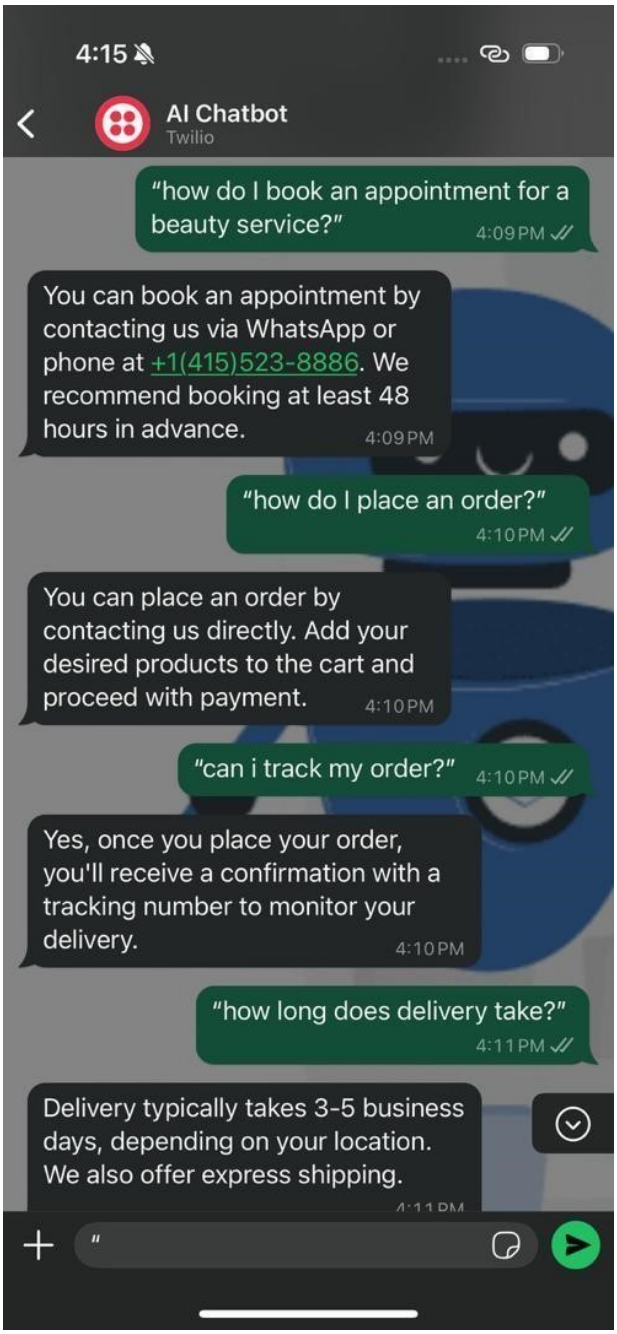


Figure 7: Response of the chatbot through WhatsApp Business for an order-tracking query. The figure above shows a chat between a user and a chatbot on WhatsApp. In the first chat, the user asks, "How do I place an order?" and the chatbot gives clear instructions. In the second chat, the user asks, "How long does the delivery take?" The chatbot remembers the earlier question about orders and gives the right answer. In the last chat, the user asks, "How do I track my order?" The chatbot explains the steps in a simple way.

Figure 8: Showing the interaction of the chatbot, answering one of the most common questions about the delivery times

The figures shows that a conversation between a user and an AI-powered customer support chatbot. In one chat, the user asks, "How long does delivery take?" The chatbot responds that delivery typically takes 3-5 business days and mentions express shipping as an option. In another chat, the user asks, "How much do your services cost?" The chatbot explains that prices vary by service and suggests contacting for specific details.

Figure 9: The ability of the chatbot to keep up continuity with the session upon follow-up question. Among all the significant aspects of the user interaction shown in these screenshots is session continuity. This concept means that the chatbot still remembers what was said before and, thus, it answers followup questions in relation to previous messages. This feature therefore enhances the smooth flow of conversations and minimizes users' need to repeat themselves. Screenshot also shows how the chatbot handles FAQs, which is a critical feature in customer support. The chatbot responds correctly to

commonly asked questions, hence making the user's experience easier to get information with less struggle.

4.3 Feedback collection and iterative improvement

These tests included functionality testing, a necessary process for the validation of functional and nonfunctional requirements, while it also covered the overall user experience about response accuracy and resilience of the system. The tests were useful in finding strengths and weaknesses, thus providing a basis for the ongoing optimization of the chatbot.

Usability testing was done to ascertain if users found it easy to interact with the system, clarity in responses, and the degree of accessibility. Different demographics were involved in the testing wherein users could give a wide variety of feedback about the usability of the chatbot in navigating WhatsApp intuitively. The overwhelming majority of the users believed the chatbot was quite easy to interact with, explaining that the responses were clear and easy to understand. Several participants did indicate that the chatbot needed refinement based on their responses to create a more interactive tool which relates specifically with the individual.

Along with usability, performance metrics were collected on how well the chatbot handled queries thrown at it by the user. On standard questions, such as product availability or the status of orders placed, users were satisfied with the responses thrown up by the chatbot. Complex queries, many times involving several layers or detail, occasionally led to misunderstandings. Such feedback indicated that this involved more subtle natural language processing so that the chatbot would be able to respond correctly to subtle or complex queries.

Feedback was systematically collected through user experience questionnaires, relevance to the responses, and shortfalls that were noted. For example, participants observed how conveniently the availability of a chatbot presence was assured around the clock, for simple inquiries out of usual business hours. They also liked the speed with which the chatbot responded, although they did note limitations in some cases with complex questions. From these comments, certain areas of improvement have been noted, which include enhancing multi-question interpretation by the chatbot and the use of a softer tone

of voice in sensitive questions. Iterative feedback enabled focused refinements that improved the response accuracy of the chatbot and increased user satisfaction.
Test scenarios were prepared according to the requirements for checking the robustness of the chatbot in the field conditions of the user for queries in a wide range from simple to complex. On simple questions, the chatbot performed very well and was able to give quite accurate answers and responded to the user's need. In open-ended or weakly framed queries, it did not give consistent results, which meant that further tuning of NLP would help a lot in enhancing the accuracy of the chatbot on unstructured queries. Further testing with other platforms showed that the chatbot is adaptable to multiple platforms and works within the WhatsApp Business channel.

Response time testing was done to measure how quickly users received responses to their queries. This aspect is paramount in maintaining user interests, as feedback was provided instantly to the users once the tasks were affected. Concisely, test results showed an average response time of less than two seconds, a speed perceived to be fast by most participants. The chatbot's speed, compared to traditional support methods which takes usually more time before getting responses, greatly contributed to efficient user experience.

4.4 Terminal output and confirmation of functionality of chatbot

Additional confirmation that the chatbot was up and functional was obtained during testing from the terminal output, as it provided insight into how well it was performing. The terminal logs showed that the running Flask application was set to debug mode and was accessible at the local address http://127.0.0.1:5000. Each incoming request was logged, including both GET requests for the main page and POST requests to the /whatsapp endpoint. This output validated that the requests were processed, and the status code 200 showed that the responses were processed accordingly. Real-time log data was an excellent way to track activity on this chatbot, so if there were any potential problems, they could be found through testing.

4.5 Summary of test results and improvements

The testing phase gave insight into the overall performance of the chatbot and how well it can be improved. The feedback indeed proved that the chatbot did its intended purpose of mostly understanding typical customer support queries and offering its assistance in a quick manner around the clock. Still, users were able to identify areas where the potential for improvement was there, especially regarding NLP accuracy when it came to multi-part or nuanced questions. This brought, upfront, the improvement by implementing this feedback into the tone, more conversational and with enhanced NLP capabilities, thereby seriously strengthening the functionality of the chatbot and improving the user experience of such a tool with AI-driven interaction versus the human touch of customer service

5 CONCLUSION AND RECOMMENDATIONS

This chapter provides a comprehensive summary of the chatbot development project, highlighting its overall performance, major findings, and areas for improvement. The chapter reviews the strengths demonstrated by the chatbot, such as its ability to handle simple customer queries efficiently and provide 24/7 support, while also identifying challenges like its limitations in addressing complex or context-dependent queries. Additionally, recommendations for future improvements are discussed, including enhancing the chatbot's natural language understanding, improving personalization, and expanding its integration with multiple platforms to optimize its effectiveness and user experience on e-commerce platforms. These insights pave the way for refining the chatbot to meet evolving user expectations and industry standards.

5.1 Major Findings

Findings that pertained to the performance of the chatbot in view of its usefulness concerning customer service were identified during the test and subsequent review. Major strengths and weaknesses, summarized, included the following: From the simplicity of handling basic enquiries about the status of an order, availability of a product, or details about shipment, this chatbot was strong in terms of the level of NLU required. The fact that these simple queries are processed with accuracy underlines the essential strength it possesses in supporting e-commerce customer service, handling routine queries efficiently.

Out of all the advantages that were made known, the chatbot could support multiple users without delays, something particularly valued during high-traffic situations, such as holiday sales, when responses may be often needed. What mattered most to most users was the fact that the chatbot was available anytime and could also provide answers outside regular working hours.

While performing well in the case of simple queries, many a time, when the questions were more complex or context-dependent, the chatbot responded inadequately. Occasionally, users received irrelevant answers to their questions if those depended a lot on more context or subtle interpretation. This weakness indicates one area of improvement concerning the capability for understanding subtler or multi-layered queries.

5.2 Identified Weaknesses and Proposed Enhancements

Moreover, the following are recommended to enhance the effectiveness of the chatbot on efficiency and users' satisfaction. Firstly, increasing the diversity and difficulty level in the training data related to higher-order queries of the chatbot will enhance it. Integrating more advanced datasets with mixed conversational patterns and contexts will help the chatbot understand complex queries put up by the customers and respond correctly. Personalization designs for recognition and the capability to compose responses based on user history can really make it personal. One could even greet a returning customer or refer to a previous query; such uses of the chatbot would go a long way in personalizing interactions closer to what most customers are used to experiencing with live customer support.

Voice recognition would make interaction with the chatbot easier for users, by using their voice commands, therefore aligning with emergent use of voice assistants. This may grant easier access to the chatbot for those who would prefer or need hands-free operation. Integrating the chatbot into other platforms, like social media or mobile apps, would extend its reach and provide touchpoints for customers. If one could use a chatbot through more channels, then it will make its presence even more pervasive and adaptable. This requires further optimization to sustain very heavy demands continuously while maintaining response speed. Strengthening the infrastructure of the chatbot, especially during peak hours, ensures that users have a uniform experience, while in promotional events or holiday seasons, demand is higher.

5.3 Future Work and Recommendations

Following are a few recommendations that can be implemented in the future to enhance the utility value of a chatbot in customer service applications. advanced machine learning models could be implemented, such as transformers and large language models. Examples include chatbots developed on BERT or GPT engines, which will allow it to process complex grammatical sentences and produce responses relevant in context. This will be an improvement in the capability toward emotion detection of the user through sentiment analysis and adapting the response in that respect. The further consequence of this will be an empathetic interaction-especially when dealing with frustrations and concerns of customers.

Periodic evaluation regarding usability and performance, including stress tests, will further ensure that responsiveness and efficiency scale as the chatbot does. Through continuous improvement in a periodic manner, it changes due to changes in user expectations and technological advancements. If only the chatbots would learn from them through continuous feedback to enable iterative improvement. That way, this would ensure feedback from any wrong responses informed subsequent training that the chatbot continually improved toward real usage patterns.

For this reason, while the present state of this chatbot is an incredibly good starting point, for this to realize maximum utility and effectiveness as a tool for customer service, a lot more development, testing, and refinement are necessary on the part of the users. With these recommendations, the chatbot could further adapt to evolving demands in electronic commerce and continue to give the best services to the customers.

5.4 Conclusion

The major objective of the project was to design a chatbot that supports customer service activities of ecommerce websites by responding quickly and effectively to users' queries. Therefore, it would be appropriate to infer from the project's outcomes that this objective has been broadly achieved, though the best strengths the chatbot has shown relate to simple inquiries and the possibility of offering its services on a 24/7 basis. Users appreciated speed and availability, with some mentioning that they would like its capability to tackle complex or contextual questions improved.

This development process instilled the need to have the user at the heart of designing and building a chatbot. User feedback has been immense in use for determining areas of improvement and has helped put work into developing the chatbot to enhance functionality. In fact, with further refinement in handling complex queries, more personalization, and adaptability to other platforms, it is turning out to be one powerful tool for e-commerce businesses to serve their customers sans glitches.

REFERENCES

Al-Amin, M., Ali, M.S., Salam, A., Khan, A., Ali, A., Ullah, A., Alam, M.N. and Chowdhury, S.K., 2024. History of generative Artificial Intelligence (AI) chatbots: past, present, and future development. *arXiv preprint arXiv:2402.05122*. Available at: https://arxiv.org/abs/2402.05122

Baabdullah, A.M., Alalwan, A.A., Algharabat, R.S., Metri, B. and Rana, N.P., 2022. Virtual agents and flow experience: An empirical examination of AI-powered chatbots. *Technological Forecasting and Social Change*, *181*, p.121772.

Casheekar, A., Lahiri, A., Rath, K., Prabhakar, K.S. and Srinivasan, K., 2024. A contemporary review on chatbots, AI-powered virtual conversational agents, ChatGPT: Applications, open challenges and future research directions. *Computer Science Review*, *52*, p.100632. Available at: https://www.sciencedirect.com/science/article/pii/S1574013724000169

Clifford, D., 2024. *Data protection law and emotion*. Oxford University Press. Available at:

Haile, R. K. (2022). Implementing a Low-Code AI-Based Chatbot on Azure combined with Boomi Integration. Available at:

https://www.theseus.fi/bitstream/handle/10024/787415/Haile_RedeatKassan.pdf?sequence=2

Hasal, M., Nowaková, J., Ahmed Saghair, K., Abdulla, H., Snášel, V. and Ogiela, L., 2021. Chatbots: Security, privacy, data protection, and social aspects. *Concurrency and Computation: Practice and Experience*, *33*(19), p.e6426. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6426

Kyrylenko, Y., 2024. Chatbots and virtual assistants vs. human agents in IT customer support. Available at: https://lauda.ulapland.fi/handle/10024/66071

Nicolescu, L., & Tudorache, M. T. (2022). Human-computer interaction in customer service: the experience with AI chatbots—a systematic literature review. *Electronics*, *11*(10), 1579. Available at: https://www.mdpi.com/2079-9292/11/10/1579

Rybaric, R., 2023. *Microsoft Power Platform Enterprise Architecture: Design tailor-made solutions for architects and decision makers to meet complex business requirements*.

APPENDIX 1

Questionnaire

I hope this finds you in great spirits and with the best health. I am researching the project "Design of a Simple AI-Powered Chatbot for Customer Support: A User-Centric Approach." To be able to reach this goal, I must learn about the present state, effectiveness, and user experience about the current AI chatbots in use-especially those serving for support applications on e-commerce websites and WhatsApp Business. This could enable me to attain a very pivotal position in contribution toward the design for an improvement of functionalities in chatbots.

Thus, I would like to invite you to take part in a short online questionnaire regarding this research study. The questions refer to your experience and level of satisfaction with AI-powered chatbots and point toward areas where improvement might be necessary. It is with your response to this survey that you will help shape how user-friendly and efficient the chatbot systems developed in the future are.

Note that participation is fully voluntary, and you are free to withdraw at any time without penalty. Information collected will only be used for academic purposes and will not be shared with any other party. All information shall be safely stored and reported as aggregates to protect the privacy of the individual responses.

Any other questions or for further information about the survey and/or research, do not hesitate to contact me at: arafat.adigun@centria.fi or +358465494147.

Your time and input are appreciated, and I thank you in advance for your participation.

Yours faithfully.

Arafat Adigun

Section A: Demographics

1. Age Group

☐ 19 -24 ☐ 25-34 ☐ 35-44 ☐ 45-54 ☐ 55+

2. Gender

☐ Male ☐ Female ☐ Prefer not to say

3. How frequently do you interact with online customer support?

☐ Daily ☐ Weekly ☐ Monthly ☐ Rarely ☐ Never

4. Which platform do you most often use for customer support?

☐ Website Chat ☐WhatsApp Business ☐ Email ☐ Phone ☐Other (please specify)

Section B: User Experience with Chatbot 5. How easy was it to interact with the chatbot?

☐ Very easy ☐ Easy ☐ Neutral ☐Difficult☐ Very difficult 6. How satisfied are you with the chatbot's user interface?

☐ Very satisfied ☐ Satisfied ☐ Neutral ☐ Dissatisfied ☐ Very dissatisfied

7. Did the chatbot respond quickly to your inquiries?

☐ Always ☐ Most of the time ☐ Sometimes ☐Rarely☐ Never

8. The chatbot's responses were clear and easy to understand.

☐Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree 9. How would you rate the overall user experience with the chatbot?

☐ Excellent ☐ Good ☐ Neutral ☐ Poor ☐ Very poor

Section C: Chatbot Performance and Accuracy 10. The chatbot understood my inquiries correctly.

☐Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

11. How often did the chatbot provide relevant responses?

☐ Always ☐ Most of the time ☐ Sometimes ☐Rarely☐ Never

12. Were you able to resolve your issue using the chatbot without needing human support?

☐ Yes ☐ No ☐ Partially

13. The chatbot is effective at solving simple customer inquiries.

☐Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

14. How would you compare the chatbot's performance to human customer support?

☐Much better ☐ Better ☐ About the same ☐ Worse ☐ Much worse

Section D: Chatbot Benefits and Limitations

15. The chatbot saves me time compared to traditional customer support methods.

☐Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree 16. The chatbot is available at times when I need customer support.

☐Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree 17. The chatbot can handle a wide variety of customer inquiries.

☐Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

18. I feel comfortable using the chatbot to handle personal or sensitive information.

☐Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

Section E: Areas for Improvement

19. The chatbot needs improvement in understanding complex queries.

☐Strongly agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly disagree

20. What is the biggest limitation of the chatbot in your opinion?

☐ Limited understanding ☐ Slow responses ☐ Lack of personalization ☐Requires too much input ☐ Other (please specify)

21. **Additional Feedback (Optional)**

If you have any other suggestions or feedback to improve the chatbot, please provide below: (Optional open-ended question for additional insights)

APPENDIX 2

User documentation manual

User Documentation for AI-Powered Customer Support Chatbot

Introduction

Welcome to the AF Beauty Castle AI-Powered Customer Support Chatbot! This chatbot is designed to provide users with instant answers to frequently asked questions and facilitate a seamless customer service experience through WhatsApp and the web interface. Powered by advanced AI technology, the chatbot uses predefined FAQs as well as the Ollama AI model for intelligent responses, ensuring you get the information you need quickly and efficiently.

Features

•Instant Responses: Get immediate answers to common questions about pricing, delivery, return policies, and more.

•Contextual Interaction: The chatbot remembers previous interactions to provide more relevant responses.

•User Feedback: Users can rate their experience, helping improve the chatbot's performance.

•WhatsApp Integration: Communicate with the chatbot directly through WhatsApp. •AI Integration: If an answer is not in the predefined FAQ, the chatbot uses the Ollama AI model to generate responses.

•Human Agent Escalation: If the chatbot can't answer a question, it offers to connect users with a human agent.

Getting Started

Step 1: Accessing the Chatbot To interact with the chatbot, use the following WhatsApp number: WhatsApp Number: +1 415 523 8886 You can also access the chatbot online at the following link: Web interface: https://my-chatbot-project.onrender.com Step 2: Joining the Sandbox To start using the chatbot in the Twilio sandbox environment:

1.Open WhatsApp on your device. 2.Send a message to the WhatsApp number provided above with the following text: join larger-modern

3. You will receive a confirmation message indicating that you can now interact with the chatbot.

Step 3: Sending Messages, you can send various inquiries to the chatbot. Here are some examples: • Pricing Inquiry: “How much do your services cost?”

• Delivery Inquiry: “How long does delivery take?”

• Return Policy: “What is your return policy?”
• Location Inquiry: “Where is AF Beauty Castle located?” Understanding Responses The chatbot utilizes a combination of predefined responses and AI-generated replies.

Here’s how it works:

•Predefined FAQ Responses: For common inquiries, the chatbot provides quick, accurate answers from a predefined list.

•AI Responses: For less common or more complex inquiries, the chatbot generates responses using the Ollama AI model. Escalation to a Human Agent If the chatbot cannot answer your question, it will offer to escalate the conversation to a human agent. Simply type “Talk to a human” at any time to request assistance from a live agent.

FAQ List and Responses

Here are some common questions and the chatbot’s predefined responses: General Information •Q: What is AF Beauty Castle? A: AF Beauty Castle is a premier beauty brand offering a wide range of beauty products and services, including skincare, haircare, and professional beauty treatments.

•Q: Where is AF Beauty Castle located? A: We are located at 28, Herman Renlundin katu 67200, Kokkola.

•Q: How can I contact AF Beauty Castle for more information? A: You can reach us via email at arafatbisola111@gmail.com or through our customer support line at +1 (415) 523 8886. Product Information
•Q: What products do you offer? A: We offer a variety of beauty products, including skincare, haircare, makeup, and beauty tools.

•Q: Can I get product recommendations for my skin type? A: Absolutely! Please provide your skin type, and we can recommend the best products for you. Service Information

•Q: What beauty services do you offer? A: We offer facials, hair treatments, manicures, pedicures, and makeup services for special occasions.

•Q: How do I book an appointment? A: You can book an appointment by contacting us through WhatsApp or by phone. Pricing and Promotions

•Q: How much do your services cost? A: Our service prices vary based on the treatment. Please contact us for specific pricing information.

•Q: Do you offer discounts or promotions? A: Yes, follow us on social media for the latest deals and promotions. Order Information

•Q: How do I place an order? A: You can place an order by contacting us directly via WhatsApp or through our online store.

•Q: How long does delivery take? A: Delivery typically takes between 3-5 business days, depending on your location.

•Q: What are the shipping charges? A: Shipping charges vary based on your location and selected delivery speed. Free shipping is available for orders above 50 euros. Returns and Refunds

•Q: What is your return policy? A: You can return items within 14 days, provided they are unused and in their original packaging.

•Q: How do I return a product? A: Contact us with your order number, and we'll guide you through the return process.

Limitations

•The chatbot operates within the limits of the Twilio sandbox environment, which allows a limited number of interactions. For production use, a Twilio account and setup are required.

•While the chatbot strives for accuracy, it may not have answers to all questions, especially those outside the predefined FAQ scope.

Technical Information (For Developers) For developers interested in the technical aspects of the chatbot:

•Framework: Built using Flask, a lightweight Python web framework.

•AI Technology: Utilizes the Ollama model for AI responses, enabling advanced natural language understanding.

•Deployment: Hosted on Render, with continuous deployment linked to a GitHub repository.

•Dependencies: Managed via Poetry and specified in requirements.txt.

Code

Repository The source code is available on GitHub: https://github.com/Bisola14/my-chatbot-project-.git Contact Information For further inquiries or support regarding the chatbot, please contact: •Email: arafatbisola111@gmail.com

•GitHub Issues: Use the issues section on the GitHub repository for bug reports and feature requests. Conclusion Thank you for using the AF Beauty Castle AI-Powered Customer Support Chatbot! We hope it enhances your customer service experience. Your feedback is valuable and helps us continually improve the chatbot's performance and capabilities.